Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Mobile Solution for Early Detection of Heart Diseases using Artificial Intelligence and Novel Digital Stethoscope

Devang Sharma*, Saurabh Sahu*, Dr. Amol Pande*
*Department of Computer Engineering, Datta Meghe College of Engineering,
University of Mumbai

Abstract – **As per the World Health Organization (WHO), cardiovascular diseases (CVD) are the number one cause of death globally, more people die annually from heart diseases than from any other cause. Moreover, in India mortality rates in rural areas is very high as compared to urban areas due to lack of proper health care facility and qualified health workers. Therefore, this paper proposes to develop a mobile, affordable system that will be used by amateur health-workers for early detection of cardiac abnormalities like Murmurs. We have designed a novel digital stethoscope to record heart sounds in real-time and upload it to our mobile application for concurrent analysis on the Cloud. To detect abnormalities, heart sounds are pre-processed via algorithms like audio slicing and segmentation, after which they will be converted into high-quality spectrograms and then classified by our pre-trained Cloud-based Convolution Neural Networks (CNN). The CNN model is trained using datasets provided by the University of Michigan, PhysioNet Challenge 2016 and PASCAL's Classifying Heart Sounds Challenge.**

***Keywords** – Heart Sound Classification, Murmur, Heart Sound Segmentation, Spectrograms, Convolution Neural Networks (CNN), Transfer Learning, Inception v3, Digital Stethoscope, Amazon Web Services (AWS), Python Flask Server, AWS Amplify, Android.*

I. INTRODUCTION

Heart diseases have emerged as the primary cause of deaths worldwide [4]. A study of the Medical Council of India's historical data conducted in 2017 shows that there were only 4.8 practicing doctors per 10,000 of the population [7]. As a result, a majority of people living in rural areas depend on informal healthcare workers who provide ~ 75 percent of primary-care but have no formal medical training, thereby worsening the cardiac risks [6]. One of the most common forms of heart disorders is murmur. Expert healthcare professionals are able to differentiate between abnormal and normal heart sounds and make an assessment for referrals to more advanced and expensive testing such as ECG. However, informal healthcare workers lack the expertise for cardiac auscultation; hence, they are not able to catch early signs of Murmurs. Our study aims to understand whether a low-cost system can be developed to detect early signs of valvular heart disorders.

At the beginning of a cardiac cycle, the heart produces electrical activity and later these electrical activity causes atrial and ventricular contractions. This, in turn, causes the blood to flow between the different chambers of the heart and all around the body [16]. The opening and closure of the heart valves resulting in accelerations-decelerations of blood give rise to vibrations of the cardiac structure (the heart sounds and murmurs). These vibrations can be heard at the chest wall, and listening for the specific pattern can give an indication of the health of the heart. Cardiac auscultation, i.e. listening to the heart sounds via stethoscope, is one of the main diagnostic techniques used by doctors to discern the presence or absence of cardiac abnormalities such as Murmurs [17].

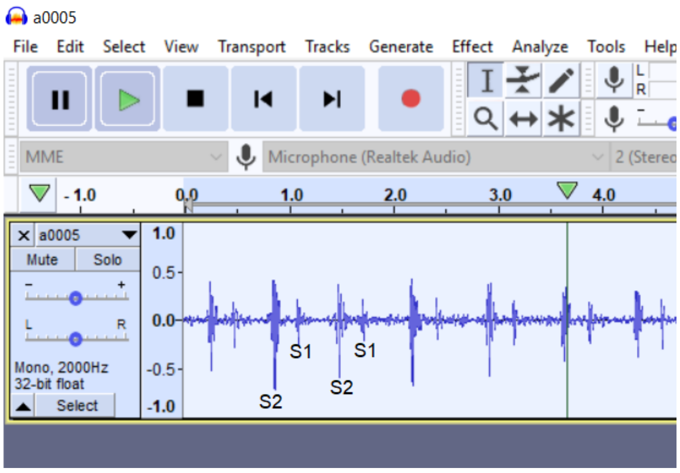


Fig 1. – Heartbeats visualized with Audacity Software

Figure 1. illustrates heartbeats as phonocardiograms (PCG) using Audacity. The phonocardiograms (PCG) are the graphical representations of the heart sound recordings. Usually, the fundamental heart sounds (FHS) include the first (S1) and second (S2) heart sounds. The S1 sound is heard at the start of the isovolumetric ventricular contraction when the mitral and tricuspid valves close as a result of a rapid increase in pressure within the ventricles [3]. Whereas, S2 happens at the start of diastole with the closing of the aortic and pulmonic valves. Even though FHS is the most recognizable sounds of the heart cycle, other audible sounds may also be present due to the mechanical activity of the heart, like the third heart sound (S3), systolic ejection click (EC), diastolic sound or opening snap (OS), mid-systolic click (MC), as well as heart murmurs.

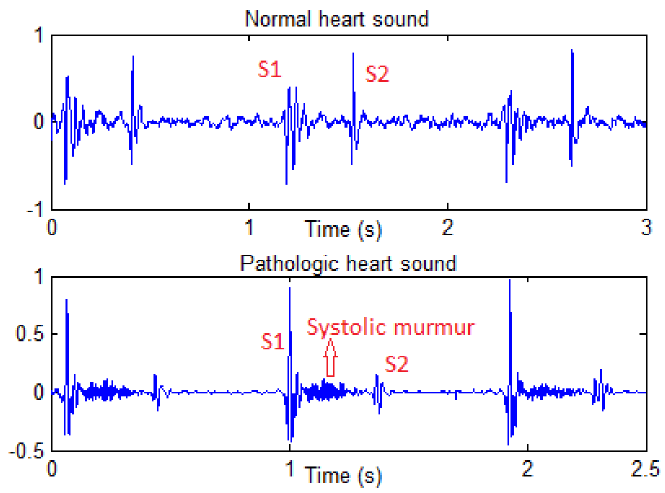


Fig 2. Normal and Abnormal Heart Sounds

Figure 2. illustrates the difference between the PCG of a normal and abnormal heart sound. A heart murmur is a blowing, whooshing, or rasping sound heard during a heartbeat caused by the turbulent, high-velocity flow of blood through the heart valves or near the heart [15]. Murmurs are evaluated on a scale of 1 to 6 based on their intensity, location and when they occur within the cardiac cycle [17]. The system proposed in this paper will facilitate the initial diagnosis of murmurs which can later be used to refer the patients for more advanced tests like ECG.

This study does not take into consideration the effects of blood pressure, diabetes or other diseases on the heart and is only concerned with the classification of heart sounds collected from a diverse variety of people under the diverse conditions and the presence of noise in these recordings.

II. SYSTEM ARCHITECTURE

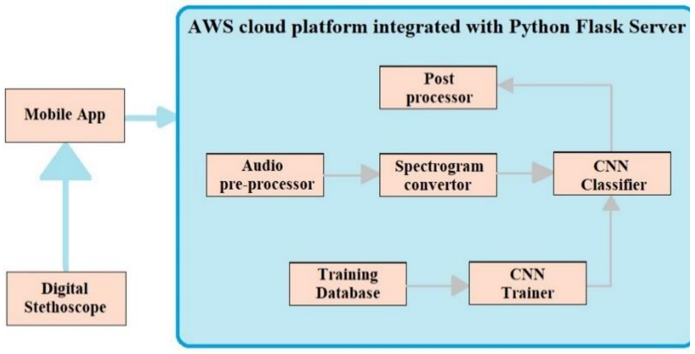


Fig 3. System Architecture and components

As per figure 3. the system consists of the following components –

A. Digital Stethoscope –

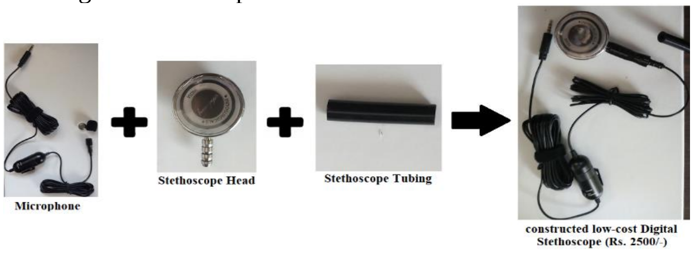


Fig 4. Digital Stethoscope Design

Digital Stethoscope is built using an analog stethoscope head (IndoSurgicals Silvery II-SS), an 8mm lapel mic (Omni-directional electrical condenser with a signal-to-noise ratio of 74dB, the sensitivity of -30dB, the frequency range of 65Hz-18kHz), and a stethoscope tubing to provide a sound-insulated connection. For recording the low amplitude and lower frequency band heart sound, we had to maximize the sensitivity by using a low pass filter, and a 32bit floating point sampling at the rate of 44KHz while recording. Furthermore, we used Audacity for an experimental view of spectrograms on a laptop.

B. Mobile Application –

It contains a simplified frontend for the users to capture heart sound by interfacing with the stethoscope through the microphone jack. Saves the heart sound as a digital audio signal (.wav) and then uploads it to Amazon Web Services (AWS) S3 storage integrated with an AWS Lambda function for analysis. It then returns the diagnosis to the user whether the heart sound is normal or abnormal. We used Android Studio, AWS Amplify and Python Flask server to develop the application.

C. Audio pre-processor –

Python program to reduce noise and then slice the heart sound into fixed-length segments. The first and last slices are dropped from analysis as the first slice usually contains noise and the last slice is usually incomplete.

D. Spectrogram Convertor –

Python program to transform the audio segments into spectrograms (time-based frequencies on the y-axis and linear time on the x-axis) with the help of several helper libraries such as Scipy and Matlabplot. It performs Short Time Fourier Transform (STFT) of the audio signal and plots the spectrogram. We have also appraised the properties such as the Hanning window “overlap factor” and frequency “bin size” to get a better-quality spectrogram for best results in CNN classification.

E. CNN Trainer –

A cloud-based program to train the CNN model (Inception v3) to classify spectrograms as either normal or abnormal. CNN model is primarily trained with normal and abnormal heart sounds of three open-source repositories.

F. CNN classifier –

A program that classifies input spectrograms as abnormal or normal using the trained model.

G. Post–Processor –

Calculates the mean of all classification scores to give an overall result.

H. Training Dataset –

We primarily used 3 open datasets to train our CNN model:

1.) PASCAL’s Classifying Heart Sounds Challenge:

It consists of data gathered from two sources: Dataset A – recorded via the iStethoscope Pro iPhone app from the general public via, and Dataset B – recorded using the digital stethoscope DigiScope from a clinical trial in hospitals. Dataset A consists of 176 audio files in WAV format out of which 31 are normal, 34 are murmur and remaining are other cardiac diseases. Dataset B consists of 656 files in WAV format out of which 320 are normal and 95 are murmur.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Dataset B also consists of noisy samples for attributing a wide variety of scenarios [2].

2.) PhysioNet Challenge 2016:
Heart sound recordings were taken from around the world, in either a clinical or nonclinical environment, from both healthy subjects and diseased patients. Training set – comprising five databases (A to E) with a total of 3,126 heart sound recordings, ranging from 5 seconds to 120 seconds. Validation set – a copy of 300 records from the training set [3].

3.) University of Michigan:
It consists of 23 audio files of approximately one-minute length taken from 4 different sections, i.e., Apex Area - Supine, Apex Area - Left Decubitus, Aortic Area - Sitting; Listening with the bell of the stethoscope, Pulmonic Area - Supine; Listening with the diaphragm of the stethoscope [1].

III. IMPLEMENTATION METHODOLOGIES

A. Pre-processing and Noise Reduction –

The heart sounds recorded by our digital stethoscope will contain background noise and also sounds from other body organs. Therefore, the first step of our pre-processing module deals with reducing the above-mentioned disturbances from the recorded heart sounds. This will help us to increase our classification accuracy and allow us to record and use our system in different environments with adequate resistance to noise. In order to implement this functionality, we have made use of a Low-Pass Filter (LPF). An LPF passes the signals with a frequency lower than the cut-off frequency and attenuates signals with higher frequencies. The heart sounds have a frequency range of 20Hz to 150 Hz [18], therefore, we used a cut-off frequency of 195Hz where higher frequency signal will begin to be reduced by 6dB per octave (doubling in frequency). Figure 5. illustrates a randomly selected heart sound recording before and after performing the pre-processing step.

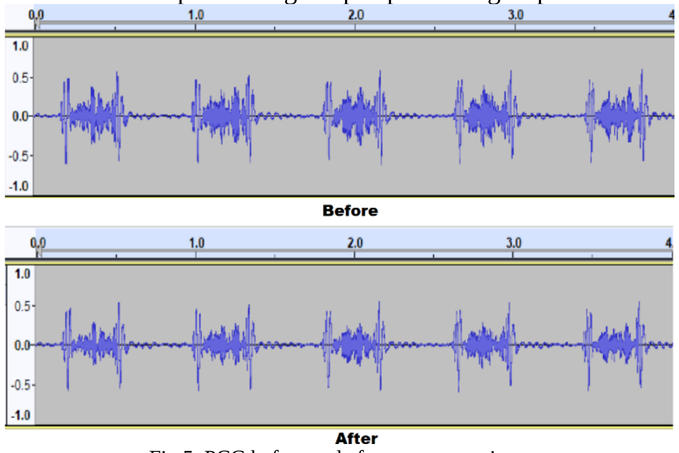


Fig 5. PCG before and after pre-processing

After applying an LPF to the heart sound, the next important pre-processing step is to ensure that the audio signal is a mono channel and not stereo. A mono channel audio signal uses only one channel, whereas stereo signals use more than one (typically two) channels. This step is performed because the spectrogram conversion process requires only a single channel of the audio signal to create a visual representation of the signal. We used Python’s Pydub library to perform this task. It is also important to note that the audio files must be in ‘.wav’ format.

Finally, while working with the Dataset B of the PASCAL dataset and the PhysioNet dataset, the sampling rate of the audio signals was 4000 Hz and 2000 Hz respectively. Therefore, we had to resample these audio signals at 44100 Hz and 32-bit floating-point.

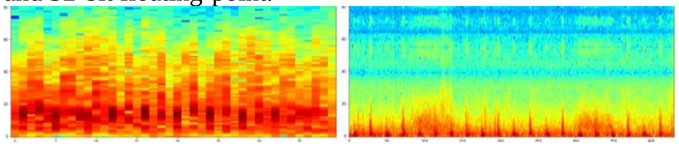


Fig 6.a. Original Fig 6.b. Resampled

Figure 6.a illustrates a spectrogram of an audio signal from the PASCAL B dataset originally when the sampling rate was 4000 Hz whereas, Figure 6.b represents the spectrogram of the same audio signal after it was resampled at 44100 Hz. Thus, it is clear from the spectrograms that at 44100 Hz, the quality of spectrograms is better as at a higher sampling rate, we are able to capture information of higher frequency components.

B. Segmentation –

The input to the system consists of a continuous, unsegmented audio stream, whereas the spectrogram converter requires segments of relatively short length as input. Therefore, for such an application, an effective method to segment audio streams into homogenous (periodic) segments is required. Segmentation involves dividing a particular object (audio) into separate parts based on a well-defined set of properties. It allows us to extract more information from the input by breaking it down into fixed-length components.

We experimented with two different methods to segment heart sounds and compared their results in order to use the method with higher accuracy.

Method 1: Designing an Artificial Neural Network (ANN) to Segment Heart Sounds:

In our case, we had to first create training data from the raw files, which was accomplished by going through all the raw files and extracting a part of the audio along with its respective label. When we created this data, a bit of pre-processing was required, i.e., we had to make all the extracted samples of the same shape, normalize the data and then we create appropriate X and Y for our deep learning model. The ANN comprised two pair of Convolution 1D (filters=50, kernel size=10, activation function = ReLU) and Max Pooling 1D (strides = 8) layer, followed by a fully-connected layer and an output layer with Softmax activation function.

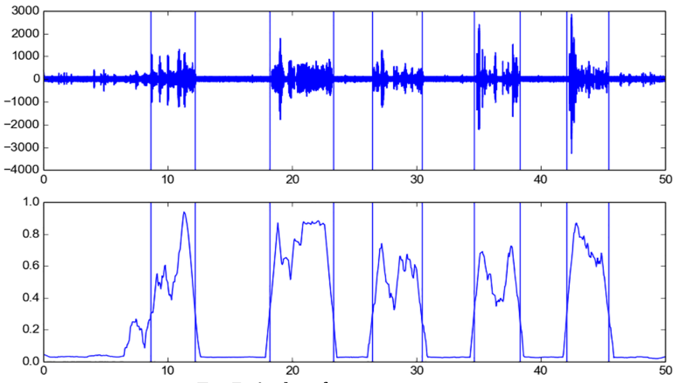


Fig 7. Audio after segmentation

Figure 7. illustrates the segments obtained after testing the ANN model. This method had several issues; firstly, the overall classification accuracy was poor (60%). Secondly, ANN led to increased complexity in the pre-processing stage and thus lengthened computation time. Thirdly, this solution was based on Silence Removal Principle and hence was not suitable for low frequency, low amplitude heart sounds as intermediate murmurs would not be captured.

Method 2: Using a Novel Elementary Audio Splitting Algorithm (Higher Accuracy):

In order to reduce the complexity, we came up with a simplified Audio Splitting Algorithm to segment the heart sounds into fixed-length chunks. We used Python's pydub library to work with the audio files. Firstly, the audio file (stored in .wav format) is read and then using split functionality of the pydub library, it is segmented based on the start, end and overlap variable values. We experimented with the overlap factor to check if by overlapping the segments is there an increase in the accuracy. Initially, we kept an overlap factor of 1500 milliseconds and then we removed the overlap factor. The accuracy in the above two cases was almost equal. So, we decided to remove the overlap factor as it may cause the CNN classifier to overfit. Using this algorithm, we segmented heart sounds into 5 seconds segments. The first segment and the last segment were discarded as the first segment contains a high percentage of noise and the last segment is generally less than 5 seconds.

Due to the simplicity and improved accuracy from Method 2, we implemented it for our final model.

C. Spectrogram Conversion –

While using Artificial Intelligence (AI) for the processing of audio signals, it is crucial to decide how the audio signals should be represented before they can be processed by CNN. According to recent studies, audio signal translated as spectral images is ideal for representation before being classified by a CNN. Classification of heart sounds involves a time series analysis of the frequency component of the audio signal. Spectrograms are 2-dimensional or 3-dimensional images representing sequences of spectra with time along the x-axis, frequency along the y-axis, and brightness or color representing the intensity or strength of a frequency component at each time frame. So, spectrograms can be considered as a detailed image of the heart sounds represented on a graph according to time and frequency with brightness or color giving the amplitude.

Spectrograms can be created either using a Fourier Transform of the time-based signal or approximated with a series of band-pass filter banks. Given our digitally sampled audio data, we use the Fast Fourier Transform (FFT) method. Our spectral image consists of slices of overlapping images ("windowing") with each slice representing the frequency components and strength at the time. This method is called Short-Time-Fourier Transform (STFT). The size and shape of the windowing slices can be varied providing us with tunable parameters for our spectrogram image. The different parameters affecting the spectrogram's precision are FFT length, window type, hop size, window length. If we use a shorter time window, it gives better timing precision but deteriorates frequency precision and vice versa. The FFT length determines the amount of spectral oversampling. The different window types are Hanning, Gaussian, Rectangular, etc. controls side-lobe suppression. In our study, we used the Hanning window and experimented with other parameters to get sharper spectrograms and improve the overall classification score.

Method 1: Using specgram() method of scipy.io library:

The scipy.io library in python includes a specgram() method that allows us to create spectrograms after reading an audio file.

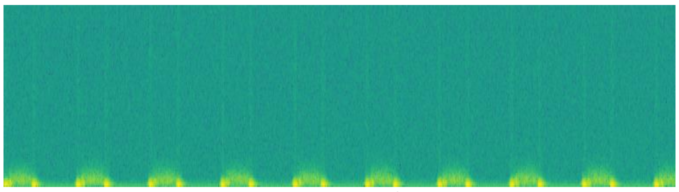


Fig 8. Spectrogram obtained from Method 1.

Issue – Firstly, this method did not provide much flexibility to experiment with the window type, size, and other trade-off parameters. Secondly, it produced poor quality bluish spectrogram and thus the overall classification score was low.

Method 2: Using Short Time Fourier Transform (STFT) and log scaling:

In this method, we designed python functions to the read audio file, perform Short Time Fourier Transform (STFT)of the audio and then perform log scaling and finally create an image file (.png). STFT is one of the most frequently used tools in audio analysis and processing. It describes the evolution of frequency component with time and in order to compute it, the time signal is divided into shorter chunks of equal length and then Fourier transform is computed on each segment separately. On plotting the changing spectra as a function of time, we get a spectrogram.

In our case, since the audio signal is a continuous-time signal to transform it, the function is multiplied by a window function which is a nonzero value over a short period of time. Then the Fourier transform of the signal is computed with the window being moved along the time axis, resulting in a two-dimensional representation of the signal. Mathematically, it is expressed as:

$$\mathbf{STFT}\{x(t)\}(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}\,dt$$

where $w(t)$ is the window function (Hanning Window) centered around zero and x(t) is our audio signal which is to be transformed.

Furthermore, the output of the STFT is complex-valued and even though the spectrum is a vector, the STFT output is a matrix. As a result, we cannot directly visualize it, instead, STFTs are visualized using log-spectra 20log10($X(h, k)$). This 2D log-spectra is visualized via our spectrogram.

We performed two experiments with different parameter values to get the ideal parameter setting for high-quality spectrograms.

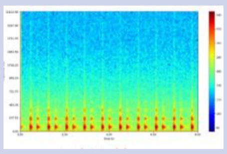
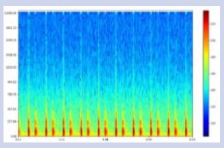
Parameter	Expt. B.1	Expt. B.2
Bin size	2**12	2**10
Overlap factor	0.9	0.5
Scaling factor	2	20
Quality	Low	High
Accuracy	75.2%	94%
Spectrogram		

Table 1.

As per table 1, in Experiment B.1 we a got poor quality spectrogram in comparison to the spectrogram obtained from Experiment B.2. As a result, the overall classification accuracy was better when using spectrograms developed using Experiment B.2 parameter settings.

Hence for the final system, we used method 2 with the parameter settings established from Experiment B.2.

D. CNN Classifier –

After the audio signal is sliced and each slice is converted into a spectrogram, CNN's are the best candidate to identify patterns within the image and classify them. CNNs are specialized Artificial Neural Networks (ANN) proposed by Yann LeCun in 1988. They are able to capture both Temporal and Spatial dependencies within an image by applying relevant filters. In order to achieve high classification accuracy, we experimented with 2 different approaches to classify spectrograms using CNN. In the first method, we designed our own CNN and in the second method, we used the concepts of Transfer Learning and applied Inception v3 architecture to classify the heart sound images.

Method 1: Using our CNN architecture:

We built a CNN model comprising 3 pairs of Convolution and Max Pooling layers followed by a flattening layer, a fully connected layer, and the output layer.

Input image – Spectrograms transformed into (64 x 64 x 3) input shape.

Convolution layer 1 – Kernel size = 3x3, number of Kernels = 32, stride = 1, activation = ReLU, padding = same.

The purpose of the Convolution layer is to perform the convolution operation so as to extract high-level features such as edges from the input image. This is done by moving the filter to the right with a certain stride value over the input image and multiplying the kernel value with the image pixel values after which they are added and a single value is obtained. The filter continuously moves towards the right-hand side till it parses the complete width of the input image after which it moves down the column to the leftmost side of the image with the same stride and continues the process till the entire image is parsed. Convolution operation tends to reduce the dimensionality of the input image, hence, in order to preserve the dimensionality, we make use of padding and use the property padding = same. Finally, so as to add non-linearity without which the network will not be able to model the response variable, we use the ReLU activation function.

Max pooling layer 1, 2 and 3 – pool size = (3 x 3)

These layers are used to reduce the spatial size of the Feature maps so that the computational power required for processing is reduced. It is also useful for extracting dominant features, i.e., positional and rotational invariance. Max Pooling returns the highest value from the part of the feature map covered by the kernel. Since Max Pooling also discards the noisy activations and perform de-noising in addition to dimensionality reduction, it is better than Average Pooling.

Convolution layer 2 and 3 – It is the same as the first convolution layer with the only difference that the number of Kernels = 64.

After three pairs of Convolution and Max Pooling layers, we enabled the model to understand the features and now flatten the final output and feed it to a fully connected layer. The fully connected layer contains 128 units and is followed by a dropout value of 0.25. Dropout is used to prevent over-fitting as the neurons may develop co-dependency during training which restrains the individual power of each neuron thereby causing over-fitting on the training set.

Finally, the output layer consists of a single neuron that employees the Sigmoid activation function since we have to classify the input image between two classes (Normal, Murmur).

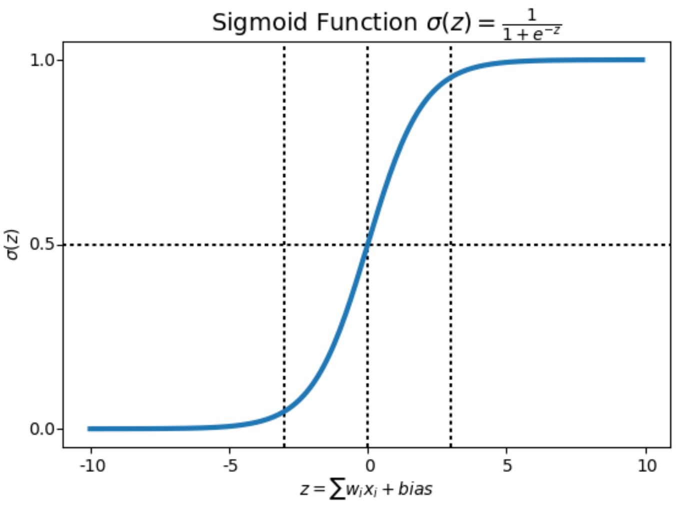


Fig 9. Sigmoid Function

The CNN model is compiled using 'adam' as the optimizer and 'binary cross-entropy' as the loss function. The model is then trained over different training datasets. Moreover, while testing models trained using this method, we found that the CNN model was overfitting on the training dataset. Hence there was a need for a better classifier model.

Method 2: Using Transfer Learning (Inception v3) for classification.

In this method, instead of building our model from scratch, we used the concepts of transfer learning as we have a relatively small dataset and limited computational resources. We used the Inception v3 model built by Google Brain Team, which is

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

a pre-trained model. It has been trained on a very big dataset and we are able to transfer weights which were learned through long training hours on multiple high-powered Graphics Processing Unit (GPU). Transfer Learning helped us to reduce training time, and increase performance.

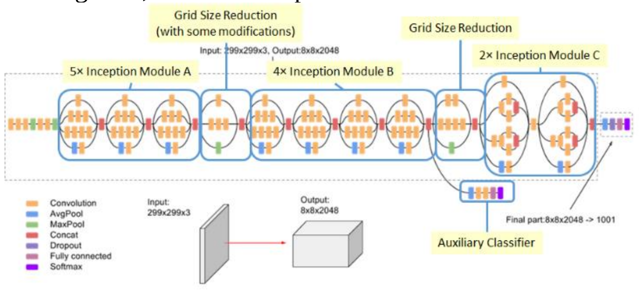


Fig 10. Inception v3 Architecture

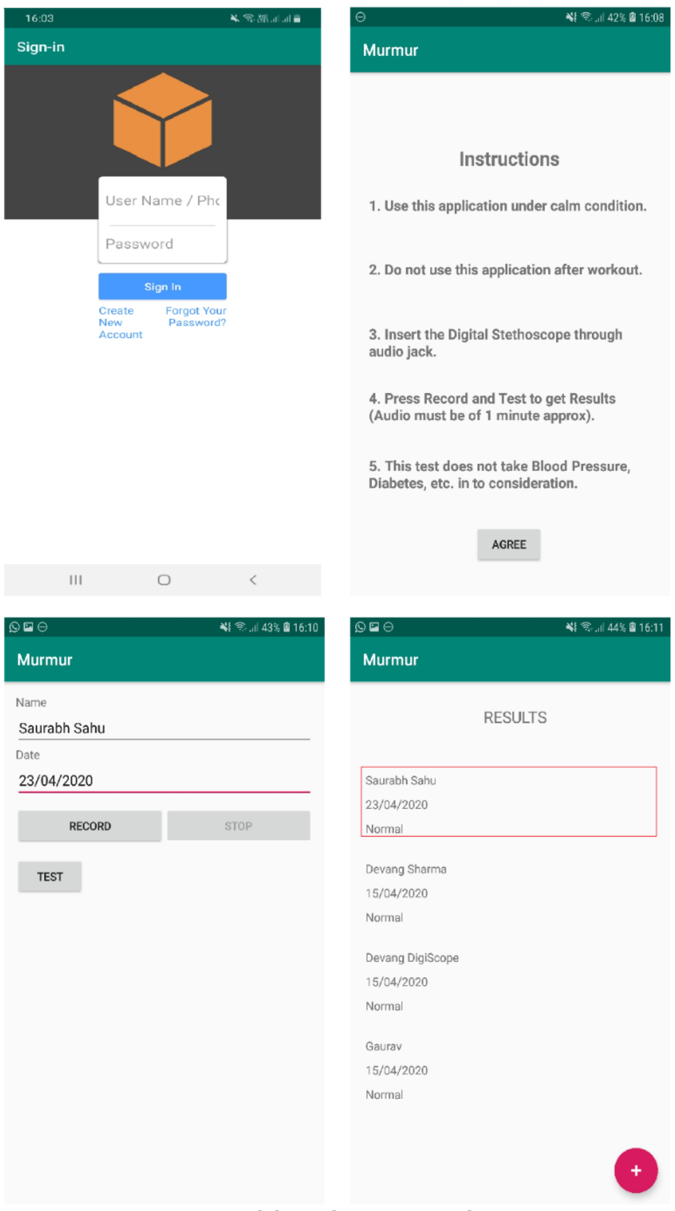
Inception v3 is a 42-layer deep learning network that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset [13]. It is in accordance with the paper written by Szegedy, et al: "Rethinking the Inception Architecture for Computer Vision". The model constitutes of symmetric and asymmetric building blocks, containing convolutions, average pooling, max pooling, dropouts, and fully connected layers. Batchnorm has been used widely in the model and applied to activation inputs.

For our system, we are going to use all the layers of the Inception v3 model except for the last fully connected layer as it is aimed at the ImageNet competition and all the previous layers are made non-trainable. Retraining of some of the lower layers is avoided to prevent overfitting. We used binary cross-entropy as the loss function and RMSprop as our optimizer with a learning rate of 0.0001.

Due to improved performance, we used models trained using the second method, i.e. Transfer Learning for our final system.

E. Mobile Application –

To provide a portable system for analysis, we developed a mobile application using Android Studio in integration with Amazon Web Services (AWS) Amplify, AWS storage (S3), AWS Lambda, AWS GraphQL, AWS DynamoDB, AWS Cognito, and a Flask API. We used AWS cloud services because of its numerous cost-effective and scalable services. The application is compatible with devices having Android 6.0 (Marshmallow) and above.



16:03
Sign-in
User Name / Phc
Password
Sign In
Create New Account
Forgot Your Password?

42% 16:08
Murmur
Instructions
1. Use this application under calm condition.
2. Do not use this application after workout.
3. Insert the Digital Stethoscope through audio jack.
4. Press Record and Test to get Results (Audio must be of 1 minute approx).
5. This test does not take Blood Pressure, Diabetes, etc. in to consideration.
AGREE

43% 16:10
Murmur
Name
Saurabh Sahu
Date
23/04/2020
RECORD
STOP
TEST

44% 16:11
Murmur
RESULTS
Saurabh Sahu
23/04/2020
Normal
Devang Sharma
15/04/2020
Normal
Devang DigiScope
15/04/2020
Normal
Gaurav
15/04/2020
Normal
+

Fig 11. Mobile Application Screenshots

As per the user interface shown in Fig 11, The mobile application users (amateur health workers in rural areas) are required to download the application first and then using our low-cost Digital Stethoscope to record the patient's heart sounds. These heart sounds are stored as .wav file on the mobile device as well as the AWS S3 storage. On uploading the audio to the AWS S3 bucket, a trigger is executed which runs AWS Lambda function which makes a REST API call to our Flask API server. The API takes the audio signal from the S3 bucket using Python's 'boto3' library and performs all the pre-processing steps discussed above and then classifies it using our trained CNN classifier. After post-processing, the result is returned by the API and stored in the DynamoDB along with other patient information. These results are then displayed on the Mobile Application. For better classification accuracy, the users are asked to input the heart sounds of above 24 seconds.

IV. RESULTS

Experiment 1 – University of Michigan Dataset.

The Michigan Dataset consisted of 23 audio files and is very small to train and validate a robust CNN model, therefore, this dataset was used to identify ideal parameter settings for a low-pass filter, audio splitting, and spectrogram conversion. The experiments and methods presented in the Implementation Methodology section were performed on the University of Michigan Dataset. By experimenting with this dataset, we selected a low-pass filter having a cut-off frequency of 195Hz. Audio segments length (5 secs) and overlap factor (none) were also realized using this dataset. Lastly, the ideal spectrogram parameter as listed in Table 1 were identified in order to get a high-quality spectrogram for better classification accuracy.

Furthermore, the University of Michigan Dataset was also used to demonstrate the feasibility of our idea, which is using CNN to classify spectrograms generated from the heart sounds. The audio files from the dataset were pre-processed, segmented, and converted into a spectrogram and a training set consisting of a total of 171 images and a validation set consisting of 22 images was prepared. The training set and test set were then used to model the inception v3 CNN model. After 10 epochs we obtained a training set accuracy of 96.61% and validation accuracy of 86.84%. The model seemed to have over-fitted on the Michigan Dataset and was unable to classify spectrograms that were new, all of which were classified into a single class, that is Murmur. Thus, we needed to use a larger open-source dataset (PASCAL and PhysioNet) to get accurate classification results and prevent over-fitting.

Experiment 2 – PASCAL Dataset.

PASCAL dataset A: 31 normal and 34 abnormal heart sound files. After pre-processing, segmentation, and spectrogram conversion training set containing 47 images and a validation set containing 11 images was created. After 25 epochs we got a validation accuracy of 81.82%.

PASCAL dataset B: 320 normal and 95 murmur files. After pre-processing, segmentation, and spectrogram conversion training set containing 91 images and a validation set containing 21 images was created. After 25 epochs we got a validation accuracy of 80.95%. Few anomalies were encountered due to significant levels of noise in the audio clips.

Combining PASCAL datasets (A and B) with the University of Michigan dataset: Training set contained 314 images and the validation set contained 70 images. After 20 epoch, validation accuracy = 97.14%.

Experiment 3 – PhysioNet Dataset.

It had 3216 audio files in the training set. After pre-processing, segmentation, and spectrogram conversion training set containing 11,274 images and validation set containing 600 images (300 normal, 300 murmur) was created. After 5 epochs we got a validation accuracy of 79.83%.

Experiment 4 – Real life heart sounds collected by our stethoscope

The digital stethoscope developed as part of the project was used to record heart signals of people ranging from 15 years to 70 years. These heart sounds were then classified using the CNN model trained with open-source datasets.

Name	**Segment**	**Segment classification**	**Overall classification**
Pramod	lowPramodMono.wavchunk1.wav.png	Murmur [0.]	Normal [0.67]
	lowPramodMono.wavchunk2.wav.png	Normal [1.]	
	lowPramodMono.wavchunk3.wav.png	Normal [1.]	
Daksha	lowDakshaMono.wavchunk1.wav.png	Normal [1.]	Normal [0.75]
	lowDakshaMono.wavchunk2.wav.png	Normal [1.]	
	lowDakshaMono.wavchunk3.wav.png	Murmur [0.]	
	lowDakshaMono.wavchunk4.wav.png	Normal [1.]	
Gaurav	lowGauravMono.wavchunk1.wav.png	Murmur [0.]	Normal [0.67]
	lowGauravMono.wavchunk2.wav.png	Normal [1.]	
	lowGauravMono.wavchunk3.wav.png	Normal [1.]	
Devang (14/4/20)	lowDigiScopeMono.wavchunk1.wav.png	Murmur [0.]	Murmur [0.]
	lowDigiScopeMono.wavchunk2.wav.png	Murmur [0.]	
	lowDigiScopeMono.wavchunk3.wav.png	Murmur [0.]	
Kamla	lowKamlaMono.wavchunk1.wav.png	Normal [1.]	Normal [0.67]
	lowKamlaMono.wavchunk2.wav.png	Murmur [0.]	
	lowKamlaMono.wavchunk3.wav.png	Normal [1.]	
Saurabh	low1586950993258Mono.wavchunk1.wav.png	Murmur [0.06]	Normal [0.76]
	low1586950993258Mono.wavchunk2.wav.png	Normal [1.]	
	low1586950993258Mono.wavchunk3.wav.png	Normal [0.99]	
	low1586950993258Mono.wavchunk4.wav.png	Normal [1.]	
Devang (11/5/20)	lowDevangMono.wavchunk1.wav.png	Normal [1.]	Normal [1.]
	lowDevangMono.wavchunk2.wav.png	Normal [1.]	

Table 2.

As per the results presented in Table 2. we achieved 6 correct classifications out of 7 tests performed (6/7 = 85.7%). Due to the current COVID-19 situation, we were able to test only a few people. Once the condition alleviates, we will begin large scale testing of our system.

V. CONCLUSION

In conclusion, we have developed a cost-effective, mobile system that can provide initial screening for cardiovascular diseases like Murmur by amateur health workers, especially in rural areas. The feasibility and accuracy of the system are demonstrated by the results obtained on testing the system with different opensource heart sound datasets. We have innovated on previous techniques on multiple fronts and the novel Digital Stethoscope developed by us is 15-20% cheaper as compared to other stethoscopes. Furthermore, our system developed by us can be easily adapted to test other common diseases that depend on auscultation as a primary diagnosis technique, like respiratory disorders. Our system also allows continuous expansion of the dataset so as to improve the accuracy over a diverse array of patients. Moreover, we are aware of the fact that there is a lot of scope for improvement and hence we plan to continue our work on this system to come up with a more effective noise cancellation technique and keep testing the model on real-time recordings collected from clinical trials. Furthermore, we plan to supplement this system with a Framingham risk score so as to provide a detailed overall heart health profile. The Hard-Coronary Framingham model uses age, cholesterol, blood pressure, etc as the input parameter to calculate the risk score [10]. This model in integration with our system would be able to give complete heart profile.

REFERENCES

[1] R. Judge and R. Mangrulkar, “Heart Sound and Murmur Library”, 2015, Retrieved from Open.Michigan Educational Resources Website: https://open.umich.edu/find/open-educational-resources/medical/heart-sound-murmur-library
[2] P. Bentley, G. Nordehn, M. Coimbra and S. Mannor, “The PASCAL classifying heart sounds challenge 2011” Website: http://www.peterjbentley.com/heartchallenge/index.html
[3] PhysioNet Dataset: https://physionet.org/content/challenge-2016/1.0.0/
[4] World Health Organization Cardiovascular diseases (CVDs). [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)
[5] Heart Diseases in India: What Statistics Show. [Online]. Available: https://www.livemint.com/Politics/fKmvnJ320JOkR7hX0lbdKN/Rural-India-surpasses-urban-in-heart-diseaserelated-deaths.html
[6] On the quack track: Over 50% 'doctors' in country practicing without formal degree. [Online]. Available: https://www.dnaindia.com/health/report-on-the-quack-track-2289951
[7] B. Potnuru, “Aggregate availability of doctors in India: 2014-2030,” in Indian Journal of Public Health, 2017, pp 182-187
[8] M. N. Krishnan, “Coronary heart disease and risk factors in India – On the brink of an epidemic?” in Indian Heart Journal, 2012, pp 364–367
[9] Heart Disease Facts. [Online]. Available: https://www.cdc.gov/heartdisease/facts.htm
[10] Framingham Heart Study. [Online]. Available: https://framinghamheartstudy.org/fhs-risk-functions/hard-coronary-heart-disease-10-year-risk/
[11] Audio Spectrograms. [Online]. Available: https://www.dsprelated.com/freebooks/sasp/Audio_Spectrograms.html
[12] https://www.googlesciencefair.com/projects/2018/7b8a1383b3af5aa357d2d4ec74bd9fa52a418f0fc3e09938d24e914aafe60c8b
[13] Inception v3: https://cloud.google.com/tpu/docs/inception-v3-advanced
[14] AWS Amplify Documentation: https://docs.amplify.aws/start/getting-started/nextsteps/q/integration/android
[15] Heart Murmurs: https://medlineplus.gov/ency/article/003266.htm
[16] Heart sounds: https://www.kdnuggets.com/2020/02/audio-file-processing-ecg-audio-python.html
[17] Cardiac auscultation and murmur: https://www.merckmanuals.com/professional/cardiovascular-disorders/approach-to-the-cardiac-patient/cardiac-auscultation
[18] “Frequency shifting approach towards textual transcription of heartbeat sounds” https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3396354/
[19] A. Raghu, D. Praveen, D. P. Peiris, L. Tarassenko and G. Clifford, “Engineering a mobile health tool for resource-poor settings to assess and manage cardiovascular disease risk: SMARThealth study,” in BMC Medical Informatics and Decision Making, 2015
[20] H. Uguz, “A Biomedical System Based on Artificial Neural Network and Principal Component Analysis for Diagnosis of the Heart Valve Diseases,” in Journal of Medical Systems, 2010, pp 61-72
[21] S. Ari, K. Hembram and G. Saha, “Detection of cardiac abnormality from PCG signal using LMS based least square SVM classifier,” in Expert Systems with Applications, 2010, pp 8019-8026
[22] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei and K. Sricharan, “Recognizing Abnormal Heart Sounds Using Deep Learning,” in KHD@IJCAI, 2017
[23] S. Latif, M. Usman, R. Rana and J. Qadir, “Phonocardiographic Sensing using Deep Learning for Abnormal Heartbeat Detection,” in CoRR, 2018
[24] M. Nabih-Ali, E. El-Dahshan and A. Yahia “Heart Diseases Diagnosis Using Intelligent Algorithm Based on PCG Signal Analysis,” in Circuits and Systems 8, 2017, pp 184-190