Classification of Lung Sounds and Disease Prediction using Dense CNN Network

Suyash Lakhani, Ridhi Jhamb

Abstract: Respiratory illnesses are a main source of death in the world and exact lung sound identification is very significant for the conclusion and assessment of sickness. Be that as it may, this method is vulnerable to doctors and instrument limitations. As a result, the automated investigation and analysis of respiratory sounds has been a field of great research and exploration during the last decades. The classification of respiratory sounds has the potential to distinguish anomalies and diseases in the beginning phases of a respiratory dysfunction and hence improve the accuracy of decision making. In this paper, we explore the publically available respiratory sound database and deploy three different convolutional neural networks (CNN) and combine them to form a dense network to diagnose the respiratory disorders. The results demonstrate that this dense network classifies the sounds accurately and diagnoses the corresponding respiratory disorders associated with them.

Keywords: Respiratory sounds, Classification, Dense Network, Audio files, Respiratory cycles

I. INTRODUCTION

The lungs are quite possibly the most essential organs in our body, however one that we frequently underestimate. Respiratory disorders such as Chronic Obstructive Pulmonary Disease (COPD), Lower Respiratory Tract Infection (LRTI), Upper Respiratory Tract Infection (URTI), Pneumonia and Asthma are one of the leading causes of deaths worldwide [1]. Lung auscultation is a technique to diagnose the functioning of the lungs and abnormalities. This technique basically depends on the doctor. Utilizing a stethoscope, the doctors may hear ordinary breathing sounds, diminished or missing breath sounds, and unusual breath sounds (e.g., rale, rhonchus, cackle, wheeze, rub). Auscultation is a straightforward, patient-accommodating and non-intrusive strategy which is generally utilized. However, it still remains vulnerable to false interpretations because of instrument limitations and physicians [2]. The latest technology which is experimenting on this problem is machine learning. Over the years, many algorithms have been developed and since the development of deep learning, the error rate has almost become close to negligible.

Different deep learning techniques like deep neural organizations, convolutional deep neural organizations have been applied to fields like computer vision, speech recognition, audio processing, sound analysis and bioinformatics where they have delivered cutting edge results on different undertakings. The machines have successfully reached the human levels of detection [3,4,5,6]. Lung sounds are generally non-stationary and thus non-linear signals makes it difficult to distinguish and analyse the sounds. In the recent years, the use of electronic stethoscope has made automated analysis efficient and easy [7]. Recently, a dataset containing various respiratory sounds was made available to encourage the analysis and development of different algorithms that can identify types of sounds and the diseases associated with it [8]. This data has sounds which can be classified into wheezes and crackles. Crackles are intermittent sounds, ordinarily under 20 ms, related with lung fibrosis (fine crackles) or chronic airway obstruction (coarse crackles). Wheezes are shrill sounds that typically last in excess of 100 ms that show obstructive airway conditions, for example, asthma and COPD. An aggregate of 6898 respiratory cycles (a solitary inward breath and exhalation) were recorded and explained by specialists as wheezes, crackles, both, or no adventitious sounds. There is currently no publication which uses dense CNN network to classify the diseases based on sounds. The database was extracted and stored in the form of audio and csv files. The data was then pre-processed, analysed and fed to custom CNN models which were then stacked to form a dense network for better accuracy. All of these steps will be explained in details in the following sections. The goal of this project is to improve the classification accuracy of the already existing CNN models by customizing them and forming a dense network.

II. LITERATURE SURVEY

Murphy et al. fabricated a framework for accurately giving an exact conclusion dependent on an analysis of recorded lung sounds. As of late, research on the utilization of CNNs in lung sound analysis has started to be distributed [4, 5, 6]. In 2017, one paper inferred that spectrogram picture characterization with CNN algorithm functions just as the SVM algorithm, and given the enormous measure of data, CNN and SVM machine learning algorithms can precisely order and pre-diagnose respiratory audio [4]. Notwithstanding, this paper utilized a similar CNN for 13 distinct respiratory sound sorts, all of which have distinctive recurrence and span profiles. The lone other distribution of CNN characterization of lung sounds in 2018 likewise made a solitary model to characterize 7 distinctive sound sorts [5].

Their end was that "CNN outperformed the feature based classifiers specifically SVM, k-closest neighbour and the Gaussian mixture models [5]. We utilised the basic CNN models used in these papers and customised them to form a dense network for better results.

III.DATA SET

For this project, we used the respiratory sound database which was created by two research teams in Portugal and Greece [8]. This database was presented in the International conference of Biomedical and Health informatics in the year 2017. This is the first sound dataset to be made available to the public. It was collected using heterogeneous equipment and it includes 920 annotated recordings of varying length - 10s to 90s. These recordings were taken from 126 patients. There are a total of 5.5 hours of recordings containing 6898 respiratory cycles, 1864 contain crackles, 886 contain wheezes and 506 contain both crackles and wheezes. The data includes clean respiratory sounds as well as noisy recordings that simulate the real life conditions. Every annotation file contains four segments: the start and end of each respiratory cycle in the recording, presence of crackle or not and presence of wheeze or not. The patients span all age groups - children, adults and the elderly. This data set includes:

- 920 .wav sound files
- 920 annotation .txt files
- A text file listing the diagnosis for each patient
- A text file explaining the file naming format
- A text file containing 91 names (filename_differences.txt)
- A text file containing demographic information for each patient.

To summarize, most of the work in the project was related to pre-processing the audio files and extracting features from file names and tables. After this process, the model was trained using convolutional neural networks.

IV.METHODOLOGY

A. Reading And Utilising Provided Data

The text file in the data set which has the annotation for each sound (.wav) files has very useful information like Start time of the cycle, the end time, indication of crackles or weezels (or both), the patient ID and the acquisition mode. This can also be seen in the .txt file which contains the file naming format. The segregation of the text file based on its name was done using python to extract the above mentioned features.

Table 1 : Snippet of the table after extraction of info from file

	Start	End	crac kles	weez els	pid	mod e	filename
0	1.86 2	5.71 8	0	1	160	mc	160_1b3_Al_mc_ AKGC417L
1	5.71 8	9.72 5	1	1	160	mc	160_1b3_Al_mc_ AKGC417L

Then, this newly formed table was merged with the file containing diagnosis (corresponding disease) of each patient.

B. Processing audio files

After pre-processing the file names and extracting useful information, the processing of audio files need to be done. We extracted the parts from the whole audio file which contains the respiratory cycles. This was done by utilizing the specific start and end time of cycle from Table 1. After this process, the extracted audio files were of different sizes. To be fed to the CNN, the length of the audio files must be same (i.e Start – end). Using Data visualization techniques, the best length was found out to be 6. If the length was less than 6, zero padding was done (adding silence) to get the required length. Librosa module in python was used for loading the audio files and SoundFile modules for writing it to the output path.

C. Handling class imbalance

In the patient diagnosis text file, we found that the data was imbalanced. So the data was split into train and test in a stratified manner to help the model learn and validate different classes and not just the class with majority count. To achieve this, we extracted the Patient Id from the audio files and merged it with its respective class label (Disease Type).
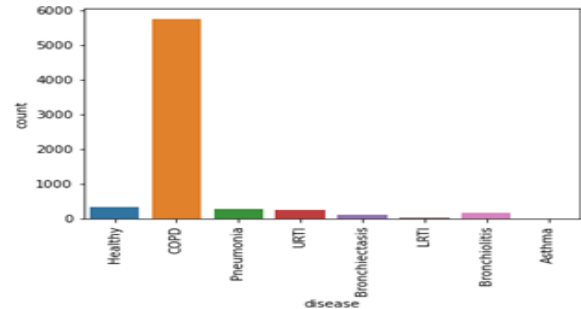

Figure 1: Skewed (Imbalanced) class labels

D. MFCCS

Mel Frequency Cepstral Coefficient (MFCC) is a feature widely used in audio recognition. In this process the audio files and signals are converted into corresponding shapes which can then be used as features to train the CNN Model. These features can help us accurately determine what type of sounds are being produced. There are other sound features such as Chroma and Mel-spectrogram which will be used for model creation. The extraction of features was done using the Librosa module in python which converts the audio files into different features.
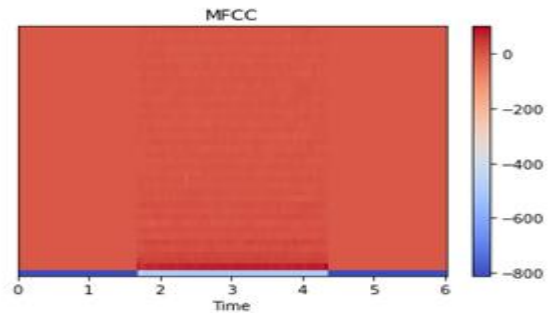

Figure 2: MFCC of a sample audio file of a patient

E. Feature Extraction

After converting the Audio files into MFCC features, firstly, the model was tested and was giving a good accuracy. So we also decided to uses chroma and mel-spectrogram features one by one and test them as well. All of them gave a very good accuracy. So, we decided to use all the features and make a dense CNN Network.

The features were extracted one by one by iterating through the train and test data and using filenames. The features were then converted into numpy array. The shape of the numpy arrays were (20, 259) (12, 259) (128, 259) for MFCC, Chroma and mel-spectrogram respectively. This would act as the input shape for our model.

F. Model Creation

We created a CNN model for every feature and combined them to form a dense network for better accuracy. The CNN model architecture is described in detail as follows.

- **MFCC Model**
  - MFCC input shape – (20,259,1)
  - First convolutional layer with filter size 5 and 32 feature maps
  - Max pool layer of pool size 2x2
  - Second convolutional layer with filter size 3 and 64 feature maps
  - Max pool layer of pool size 2x2
  - Third convolutional layer with filter size 2 and 96 feature maps
  - Max pool layer of pool size 2x2
  - Fourth convolutional layer with filter size 2 and 128 feature maps
  - Global Max pool layer

- **Croma Model**
  - MFCC input shape – (12,259,1)
  - First convolutional layer with filter size 5 and 32 feature maps
  - Max pool layer of pool size 2x2
  - Second convolutional layer with filter size 3 and 64 feature maps
  - Max pool layer of pool size 2x2
  - Third convolutional layer with filter size 2 and 128 feature maps
  - Global Max pool layer

- **Mel- Spectrogram Model**
  - MFCC input shape – (20,259,1)
  - First convolutional layer with filter size 5 and 32 feature maps
  - Max pool layer of pool size 2x2
  - Second convolutional layer with filter size 3 and 64 feature maps
  - Max pool layer of pool size 2x2
  - Third convolutional layer with filter size 2 and 96 feature maps
  - Max pool layer of pool size 2x2
  - Fourth convolutional layer with filter size 2 and 128 feature maps
  - Global Max pool layer

All of the models use a ReLU activation function and have a batch normalization layer after every convolutional layer.

For the final Dense network, all of these 3 layers were concatenated and fed to 3 dense layers with 50,28 and 8 feature maps respectively. For the last dense layer, a softmax function was used to predict the class labels. After every dense layer, a drop out layer was used to prevent overfitting. Furthermore, a NAdam optimizer was used with a learning rate of 0.001 and a default batch size. Sparse categorical cross entropy was used to compute the loss. Number of epochs used were 100 with a call back function with a patience of 5.
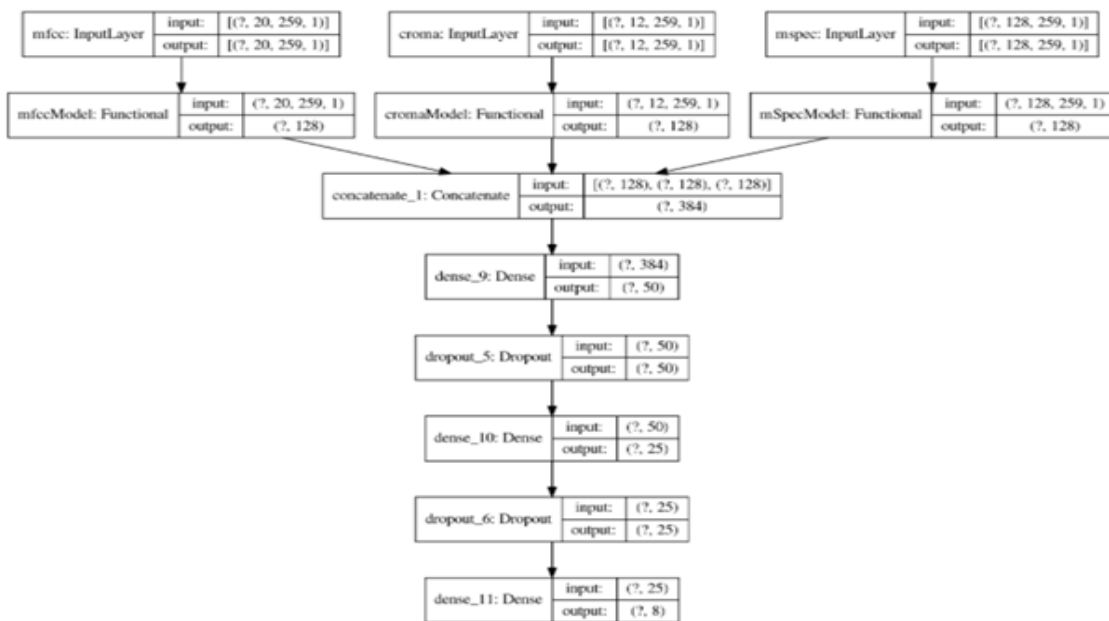

Figure 3: Architecture of the dense network in use

V. RESULTS

The model had a training accuracy of 95.42% and a loss of 0.1102 and a validation accuracy of 94.84% and a loss of 0.1955. This was already performing better than the CNNs previously used on this dataset. This result clearly shows us that by taking all the features into consideration and making a dense network, we can build a highly accurate model which can be very beneficial. This model can also achieve a higher accuracy if the number of samples are increased.
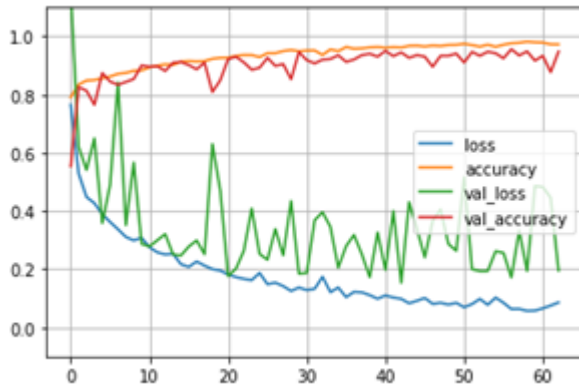

Figure 4: Graph showing the Train, validation loss and their corresponding accuracy.

VI. CONCLUSION AND FUTURE WORK

The results clearly give us an idea of the performance of this model. It demonstrates that concatenation of CNNs with different features can help us achieve very high accuracy even for limited number of samples. Future work includes testing the model with more number of samples to get an idea of its functioning with large datasets and in the real world scenario. If no data is available, we would consider data augmentation using time wrap to increase the number of samples for training and testing. This is one of the many data augmentation techniques for dealing with sound data. It can also be performed using other techniques which we will also consider in our future work. However, this will be considered as a last resort if we are not able to find any more data because of its less effectiveness and computational expenses. Lastly, we are also looking forward to optimise the model using hyper parameters such as the number of frames for the spectrogram.

REFERENCES

1. Murat Aykanat, Ozkan Kilic, Bahar Kurt, S. Saryal. "Classification of lung sounds using convolutional neural networks" EURASIP Journal on Image and Video Processing, Article. 65, 2017.
2. Vinita Shivakumar. "Classification of Respiratory sounds" Stanford University, 2019
3. R. Palaniappan, K. Sundaraj, N.U. Ahamed. "Machine learning in lung sound analysis: A systematic review," Biocybernetics and Biomedical Engineering, vol. 33, no. 3, pp. 129-135, 2013.
4. M. Aykanat, O. Kilic, B. Kurt, S. Saryal. "Classification of lung sounds using convolutional neural networks," Eurasip Journal on Image and Video Processing, vol. 65, 2017.
5. D. Bardou, K. Zhang, S.M. Ahmad. "Lung sounds classification using convolutional neural networks," Artificial Intelligence in Medicine, vol. 88, pp. 58-69, 2018.
6. C. Jacome, J. Ravn, E. Holsbo, J.C. Aviles-Solis, H. Melbye, L.A. Bongo. "Convolutional Neural Networks for Breathing Phase Detection in Lung Sounds," Sensors (Basel), vol. 19, no. 8, pp. 1798, 2019.
7. M. Gronnesby, J.C. Aviles-Solis, E. Holsbo, H. Melbye, L.A. Bongo. "Feature Extraction for Machine Learning Based Crackle Detection in Lung Sounds from a Health Survey," 2017.
8. B.M. Rocha, D. Filos, L. Mendes, I. Vogiatzis, E. Perantoni, E. Kaimakamis, et al. "A Respiratory Sound Database for the Development of Automated Classification," Precision Medicine Powered by Health and Connected Health, pp. 33– 37, 2018.
9. M. Sarkar, I. Madabhavi, N. Niranjan, M. Dogra. "Auscultation of the respiratory system," Annals of Thoracic Medicine, vol. 10, no. 3, pp. 158-168, 2015.
10. B.M. Rocha, D. Filos, L. Mendes, G. Serbes, S. Ulukaya, Y.P. Kahya, et al. "An open access database for the evaluation of respiratory sound classification algorithms," Physiological Measurement, vol. 40, no. 3, 2019.
11. G. Serbes, S. Ulukaya, Y.P. Kahya. "An Automated Lung Sound Preprocessing and Classification System Based On Spectral Analysis Methods," Precision Medicine Powered by Health and Connected Health, vol. 66, pp. 45-49, 2018.

AUTHORS PROFILE

Suyash Lakhani, Student of Vellore Institute of Technology, Vellore, Tamil Nadu, persuing B-tech, CSE department. Mail ID- lakhani.suyash@gmail.com
.

Ridhi Jhamb, Student of Vellore Institute of Technology, Vellore, Tamil Nadu, persuing B-tech, CSE Department. Mail ID- ridhijhamb1802@gmail.com