Detection of Covid-19 using Cough Sounds

1st Abhinav
dept. of Computer Science
Bharati Vidyapeeth's College of Engineering
New Delhi, India
abhi2000.rtk@gmail.com

2nd Apoorv
dept. of Computer Science
Bharati Vidyapeeth's College of Engineering
New Delhi, India
apoorvpeb1@gmail.com

3rd Himanshu Mangal
dept. of Computer Science
Bharati Vidyapeeth's College of Engineering
New Delhi, India
himanshumangal09@gmail.com

4th Naman Aggarwal
dept. of Computer Science
Bharati Vidyapeeth's College of Engineering
New Delhi, India
aggarwalnaman025@gmail.com

5th Prof. Harsh Taneja
dept. of Computer Science)
Bharati Vidyapeeth's College of Engineering
New Delhi, India
harshtaneja88@gmail.com

***Abstract*—Coronavirus, the pandemic due to which about 4 million have lost their lives and counting is still on. Many Scientists, Researchers are trying to find out the ways to detect coronavirus as soon as possible in the human body so that they can start their medication and precaution as soon as possible but due to lack of lab facilities the RT-PCR is taking more than 3 days to give the report and in the meanwhile patient get serious and life in danger. So in this paper, we proposed an audio-based coronavirus detection technique in which we can get results in minutes. Coronavirus is a respiratory disease the sound produces while breathing can tell us about the presence of coronavirus. Audio-based detection was already used for the detection of asthma, pneumonia. The technique uses machine learning and deep learning and it has an accuracy of 78% and an f1 score of 71%. This technique can be used as a starting point for just audio data to diagnose diseases and save lives.**

***Index Terms*—COVID-19, Cough, Voice, Machine Learning, VGG-19, Classifiers, KNN, Audio Analysis**

I. Introduction

Covid-19 also known as Coronavirus [1] was declared as a global pandemic by the World Health Organisation in March 2020 [2]. Since then due to the devastating impact of Covid-19 and the tragic loss of lives, leaders all across the world have been making policies trying to stop the spread of this deadly disease. It has become a topic of utmost concern to develop methods for early detection of Covid-19 which may help limit its spread. Medical workers have been working day and night to develop various vaccines and trying to improve their efficiency. In many countries like India, USA, and UK, mass production of vaccines has already begun and people have been given vaccines for Covid-19. Meanwhile, scientists have been trying to come up with different ways to detect Covid-19 in patients at early stages using Machine Learning and Artificial Intelligence. Some of them have come up with methods that have very high accuracy such as detection of Covid-19 using chest X-Rays and CT scans, and respiratory sound data [3]. In this paper, we will be discussing our research on the Detection of Covid-19 using Audio analysis of respiratory data.

Early symptoms seen in patients include breathlessness, high fever, tiredness, and dry cough [1]. Out of these dry cough is a symptom seen in over 65% of the cases at a very early stage. Cough audio sounds can be used to extract information of pulmonary health with the help of Machine Learning and Artificial Intelligence. We are using 9 different audio samples of individuals including cough sounds(shallow and heavy) and breathing(shallow and deep) and others. The dataset used is provided by IISC Banglore [4]. These audio files in the dataset are cleaned using Librosa library [5]. To perform feature extraction the audio data needs to be converted to images for better understanding. This is done using Mel Spectrogram [6]. For feature extraction we use a VGG model with 19 layers of weight which provides us with approximately 25,000 features [7]. To avoid overfitting we use PCA(Principal Component Analysis) to reduce features from 25088 to 200 [7]. At last to detect whether the tested individual is infected or not we use XGBoost Classifier [8]. XGBoost provides us with a F1 score of 71.17%. Further by ensembling XGBoost Classifier with Linear Discriminant Classifier we achieved an accuracy of 78%.

This paper is organised as follows: Section II covers Related Works. Section III provides a brief about the Audio Dataset Used. Section IV discusses the Working Methodology and Pipeline Structure of ML model. Finally in Section V we give the Results and Conclusion of our research.

II. Related Work

In the past few years, studies have shown that various respiratory diseases such as tuberculosis, pneumonia, asthma, and bronchitis can be detected using acoustic characteristics of audio voice samples. Some of the studies and related works have been discussed below.

An integrated model of audio and image processing [1] [9] was used by the healthcare division for the detection of diseases like tuberculosis with high accuracy. It was achieved by recording verbal communication of patients. This model identifies the pain in a patient's voice using machine learning tools like CNN(Convolutional Neural Network)

Damage in pulmonary and vocal regions due to respiratory diseases can be easily identified by audio analysis of cough and breathing sounds [1]. The differences in cough sounds of patients suffering from asthma, bronchitis, and pneumonia can be analysed using speech recognition techniques [10]. Using a combination of vocal features based on parameters obtained from cough sounds alone a sensitivity of 94% and specificity of 75% can be achieved [10].

Difference between dry and wet cough can be identified using audio signal spectral energy, temporal envelope, and time-independent waveform [11]. When trained against 536 samples, a recall of 55% and specificity of 93% were obtained. In the same way, audio cough samples [12] show that patients suffering from asthma have higher energy signatures when compared to non-asthmatic patients. Reference [13] provides a comprehensive review of the detection and analysis of respiratory diseases using audio and speech analysis so far.

As the surge of Covid-19 cases increased worldwide researchers with the help of machine learning and deep learning started studying different methods to detect the presence of Covid-19 at early stages to prevent the virus from spreading. [3] gave a brief overview of initiatives taken by researchers so far to detect Covid-19 using Machine Learning and AI.

As the number of cases of Coronavirus disease increased, it was visible that cough was a common symptom in 65%. Hence several researches have been done on the detection of Covid-19 using cough sound analysis. But cough is also a symptom for over Thirty more non-COVID-19 diseases [14]. Therefore this made it even more difficult to use cough as a differentiator for COVID-19. A literature review of work done on detection of Covid-19 using cough analysis is presented in [15].

Ali Imran et al [14] began to investigate the pathomorphological alterations in the respiratory system caused by the COVID-19 infection and compared them to other non-COVID 19 diseases. In order to deal with the shortage of data, they used transfer learning [16]. They employed three different models along with a mediator to avoid false-positive results. They used Deep Learning-based classifiers [16] [14] and Classical ML-based classifiers under the hood [14]. The model also has a mediator which returns inconclusive results if any of the output from three classifiers mismatches. Their model is able to identify covid 19 coughs from several other coughs. Due to the lack of audio data, they shifted to a domain aware approach and began to investigate the distinctness of the pathomorphological alterations thus overcoming the lack of data [14].

Neeraj Sharma et al. [4] created a database, called Coswara, a respiratory for cough, breath, and voice sounds. They used a website application for collecting sound samples via worldwide crowdsourcing. The dataset is divided into nine different categories these are i)breathing shallow, ii)breathing deep iii)cough shallow iv) cough heavy v) sustained vowel phonation a, vi) sustained vowel phonation e, vii) sustained vowel phonation-o, viii) one to twenty digits counting in normal pace vii) one to twenty digit counting in fast-paced. They have used a random forest classifier [17] with default parameters and 30 trees and the model was able to achieve an accuracy of 66.74% on the test dataset.

Researchers [18] started collecting voice sounds(cough and breath sounds) for the detection of covid -19. So they came up with a cross-platform application for the collection of crowdsourced data. They built a dataset consisting of voice sounds to distinguish between COVID-19, asthma, and healthy people [18]. They used three binary classifiers to distinguish between COVID-19 patients and healthy people, distinguish COVID-19 patients having cough from healthy people who have a cough, distinguish between COVID-19 patients with cough from those having asthma who had a cough. Their dataset consisted of around 7000 unique people out of which more than 200 were detected as COVID-19 positive [18]. Due to the less amount of data, the researchers applied standard audio augmentation techniques to increase the sample size of the dataset. They tried out three classifiers - Logistic Regression [19], Gradient Boosting Trees [20], and SVM [21] and got more than 70% AUC score in all three binary classification tasks. When they used only breath sounds for classification they got an AUC score of about 60% but when the researchers combined the cough and breathing sound for classification, the AUC score improved to 80% [18].

Junaid Shuja et al [22] introduced a new approach in which they used a portable smartphone enabled with a spirometer with automated disease classification using CNN. This approach was found to have good results for the classification of other diseases so authors tried to use the same technique for COVID-19 detection. They proposed a system consisting of three basic modules [22]. A fleisch type airflow tube for capturing the breathing sound using a differential pressure-based approach, a Bluetooth enabled microcontroller for data processing, and lastly, an Android application with a pre-trained CNN model to classify breath sounds. The authors examined various classifiers such as stacked AutoEncoders, Long Short Term Memory Network, and CNN and used them for lung diseases. In some cases, 1D CNN classifiers performed well with higher accuracy than other ML classifiers so they thought to use the same concept for the classification of COVID-19. In Order to improve the results of the existing applications more audio data is required from COVID-19 patients [22].

III. Description of Audio Data

In this paper, we have used the Open Source Coswara dataset which is uploaded by IISC Banglore [4]. To collect this dataset they have made a website that supports both laptop and mobile application.The average time a user devotes to recording audio in this website is about 5-7 minutes. In this dataset there are 7 sound categories namely Healthy,positive mild, positive asymptomatic,positive moderate ,respiratory illness not identified,no respiratory illness exposed,recovered full of which a user has to choose it condition.In every sub categories there are nine audio classes , namely, cough heavy,cough shallow,breathing shallow, breathing deep, sustained vowel a ,vowel e,vowel o,one to twenty digit counting normal,one to twenty digit counting fast paced.These audio samples are recorded at a sampling frequency of 48 kHz [4].

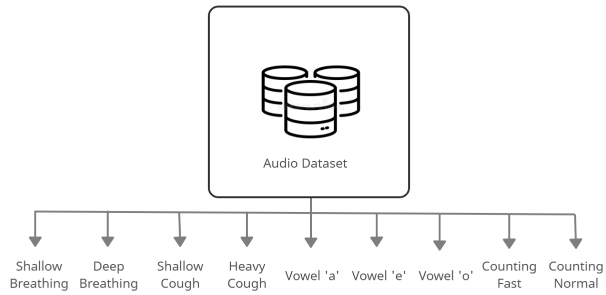

Fig. 1. Dataset.

IV. Methodology

The complete pipeline of our work can be divided into a total of 4 sets of phases that is cleaning the audio, audio conversion into images using Mel spectrogram [23], image feature extraction using VGG [7], and finally applying image classifiers to get the results.

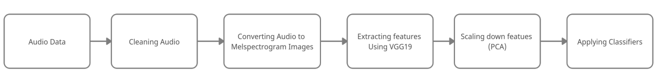

Fig. 2. Complete Pipeline of Experimentation

A. Audio Cleaning

In the first phase of our methodology, cleaning part of the audio is done. We are using librosa library [5] to read the audio file . After reading some basic audio cleaning techniques are applied like making every audio file of same length in terms of time that is of 5 seconds. If length of audio file is greater than 5 sec then we trim the audio file and if audio file is less than 5 seconds then we add little bit of padding to it. The ultimate aim for this step is to make to audio files of the same size.

B. Spectrogram generation

In order to implement image classification techniques, raw audio data is first converted to a spectrogram representation. The Mel spectrogram is acting like a transformation that details the frequency composition of a signal over time. In order to classify audio files we are converting audio into image files and then using image classification techniques like CNN's to classify between positive and negative covid

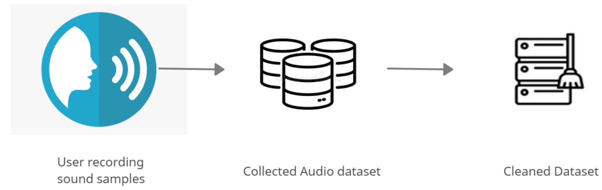

Fig. 3. Cleaning Audio Data

victims [24]. The recorded audio signal is amplitude of air pressure over time.we can work on this waveform but the resulting plot is mess and both human and machine have difficulty to understand [22]. So to overcome this we can use fourier transform to transform time domain into frequency domain.The fourier transform is computed on overlapping window segments of the signal and we get a spectrogram [4].The sampling frequency of the recorded is 16000 kHz and there is a well known fact that human can detect difference in lower frequency better than higher frequency.So,there is mel scale to convert f hertz into m mels [25].

$$m = 2595\log_{10}\left(1 + \frac{f}{700}\right)$$

So after using this formula we are getting mel spectrograms of each audio and then saving each mel spectrogram in respective image folder.Then these images are further used for later processing. After this we're able to apply image classification techniques [23].Now as a consequence, the audio signal is represented as an image.This allow us to apply various image classifiers into it.

C. Feature Extraction

Further, We are using a pre-trained VGG model to extract features from the generated spectrogram images.The Visual Geometry Group at the University of Oxford proposed the VGG convolutional neural network model for image recognition, where VGG16 refers to a VGG model with 16 weight layers and VGG19 refers to a VGG model with 19 weight layers.The architecture of VGG 19 [26] is shown in Fig. 4: the input layer accepts an image of size (224 x 224 x 3), and the output layer is a 1000-class softmax prediction. The feature extraction part of the model runs from the input layer to the last max pooling layer (labelled by 7 x 7 x 512), while the classification part of the model runs from the input layer to the last max pooling layer (labelled by 7 x 7 x 512).

VGG is a convolutional neural network that is 50 layers deep. We can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals [27] [4]. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.We need to convert every

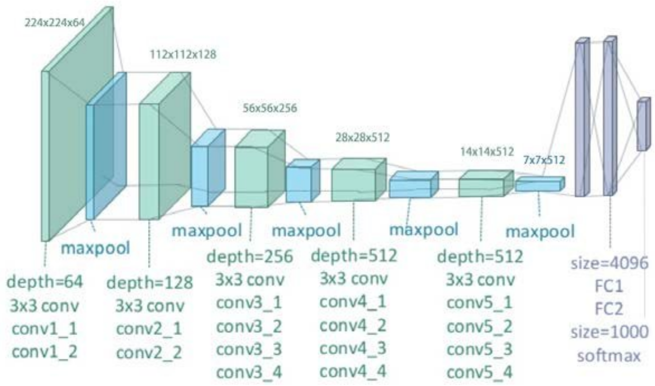

Fig. 4. Architecture of VGG 19

image into a fixed sized vector which can then be fed as input to the final classifier.Our purpose here is not to classify the image but just get fixed-length informative vector for each image.After this we encode our data in which we first preprocess the image and than pass it from the model as input and than store it as a feature vector.Finally we have to reshape our feature vector into 4d tensor because the input given to model is in the form of batches like this (1,224,224,3) and currently it is 3d..while preprocessing we also have to take care of Normalisation in which we use preprocessinput as function of keras [28]. This preprocessinput is done by resnet because resnet is trained on this kind of input,Pixels are clipped in Range 0 to 255 basically it has subtracted the channel mean from all of its pixels.

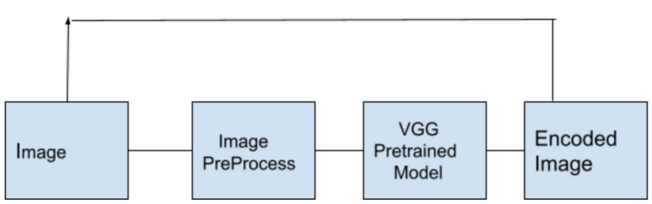

Fig. 5. Feature Extraction

As we have used VGG19 as feature extraction technique we got approximately twenty five thousand features from a single image so to avoid overfitting and improve the results we used PCA to reduce the features from 25088 to just 200 and maintaining variance of more than 80% [4]. PCA is an unsupervised linear transformation technique that is commonly used in a variety of fields, with the most common applications being feature extraction and dimensionality reduction [29]. PCA is also used for exploratory data processing and signal de-noising in stock market trading, as well as the analysis of genome data and gene expression levels in the field of bioinformatics. PCA seeks to find the highest variance directions in high-dimensional data and project them onto a new subspace with the same or less dimensions as the original one. We further use several classifiers like XGBoost classifier [7], KNN [29], Decision Trees [30], Random Forest Classifier [10], Support Vector Machines, Quadratic Discriminant Analysis [31], Logistic Regression and Bayes Theorem on the extracted 200 features to classify the image between healthy and positive [30] [29].

D. Classifiers

Due to the diversity and sophistication of image content, the image classification issue has become one of the most critical research directions in image processing and has been the subject of research for many years. Since current image classification models fail to fully exploit image information, this paper proposes a novel image classification approach that combines two outstanding classifiers: the Convolutional Neural Network (CNN) [32] [4] and eXtreme Gradient Boosting (XGBoost).After,all these data processing steps we tried XGBoost classifier to evaluate our results. By combining pretrained CNN as a trainable feature extractor to automatically obtain features from input and XGBoost as a recognizer in the top level of the network to generate performance, the presented CNN-XGBoost model provides more precise output [29].

XGBoost is an ensemble tree-based supervised learning algorithm [7]. Its aim is to find the best solution for a cost objective function that consists of a loss function (d) and a regularisation term (β):

$$\Omega(\theta) = \underbrace{\sum_{i=1}^{n} d(y_i, \hat{y}_i)}_{\text{Loss}} \; + \; \underbrace{\sum_{k=1}^{K} \beta(f_k)}_{\text{Regularization}} , \qquad (1)$$

where $\hat{y}_i$ represents the predictive value, n represents the number of instances in the training set, K represents the number of trees to be generated, and f_k represents a tree from the ensemble trees. The term ”regularisation” is described as follows [7]:

$$\beta(f_t) = \gamma T + \frac{1}{2}\left[\alpha \sum_{j=1}^{T} |c_j| + \lambda \sum_{j=1}^{T} c_j^2\right] , \qquad (2)$$

where c is the weight associated with each leaf, λ is a regularisation term on the weight, and γ is the minimum split loss reduction. Let $f_t(x_i) = c_{q(xi)}$, where q is in [1,T], where T is the number of leafs. A greedy approach is performed to select the split that increases the most the gain [7].

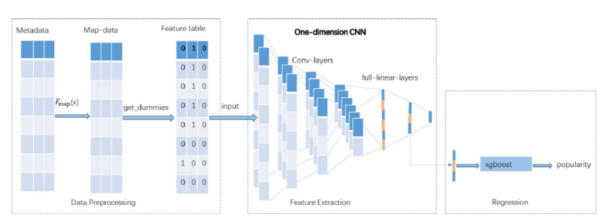

Fig. 6. Overall Workflow

Our architecture is made up of three sections, as seen in the diagram above: data preprocessing, feature extraction, and regression analysis.In the final phase We use XGBoost directly in the regression analysis process to make popularity predictions based on the high-level features extracted by pretrained models [33].

Considering the fact that ensembling generally improves the model [34] we created an ensemble of SVM, XGB, and KNN. This ensemble model gave the best accuracy so far but the F1 score was very low. Since we used F1 score as a measure to determine best classifier, hence we didn't use it [35].

V. Results and Conclusion

As explained in the previous part of the paper we used VGG19 to extract features from the image data(converted from audio) and then fed those features into different classifiers like XGB , Support Vector Machine, Random forest to name a few. We know due to the lack of sufficient data we are using an unbalanced dataset i.e the number of covid positive cases are not in equal proportion to the number of covid negative cases. Thus we cannot explicitly use accuracy as a parameter to determine the best classifier out of all those mentioned in Table 1.

	Model	Fitting time	Scoring time	Accuracy	Precision	Recall	F1_score	AUC_ROC
7	K-Nearest Neighbors	0.012046	0.037497	0.783623	0.680773	0.570658	0.740229	0.630130
1	Logistic Regression	0.068904	0.009447	0.739831	0.616446	0.605142	0.733837	0.613913
4	Linear Discriminant Analysis	0.096662	0.010695	0.744324	0.614932	0.595233	0.733191	0.636763
0	XGB Classifier	1.067243	0.009896	0.787101	0.614124	0.533871	0.712665	0.626001
8	Bayes	0.004879	0.006084	0.716546	0.582561	0.557693	0.705452	0.598683
3	Support Vector Machine	1.082017	0.025556	0.782778	0.516248	0.511558	0.692992	0.707307
6	Random Forest	0.969429	0.023752	0.779493	0.414633	0.502500	0.683840	0.582441
5	Quadratic Discriminant Analysis	0.052469	0.011379	0.778382	0.389191	0.500000	0.681389	0.538613
2	Decision Tree	0.190894	0.005614	0.655966	0.533744	0.540233	0.664635	0.540233

TABLE I
Implementation Results

Thus to deal with this unbalanced data we calculated a few parameters through which we can judge our classifier to find the best one. As per our use case one wants to maximize recall so that we can have the least number of false negatives so our main priority is to maximize recall but as we know there is always a tug of war between precision and recall we need to make sure that we have an even balance. Thus for this we reach out to calculate F1 score which takes an even balance of both of these parameters. Thus finally we would be using F1 score to rank our models.

Figure 7 shows all the models tested on the data, out of all the models applied, the XGB classifier has the maximum accuracy but considering the above facts we rely on the F1 score which is only 71.17% comparatively lesser than other models.

According to the F1 score K-Nearest Neighbor classifier gives the best results thus relying on this fact we can say it's the best classifier out of the others with an F1 score of 74.03%.

Considering the fact that ensembling generally improves the model accuracy we created an ensemble of all the models we used so far. The results of the ensembling both hard and soft are shown in Table 2 and Figure 8.

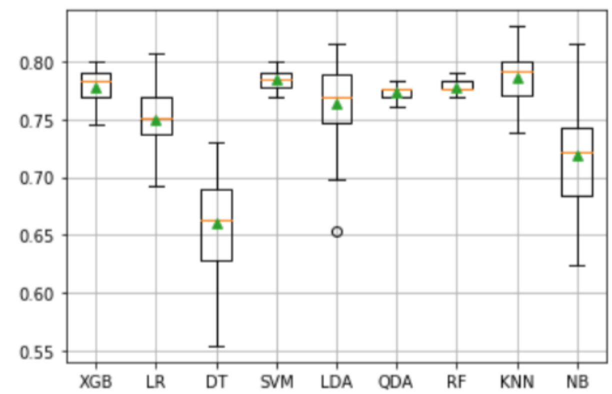

Fig. 7. Overall Claasifier result

	Model	Accuracy	Precision	Recall	F1_score
0	Ensebling_hard	0.784615	0.750	0.034884	0.066667
1	Ensembling_soft	0.784615	0.625	0.058140	0.106383

TABLE II
Ensembling Results

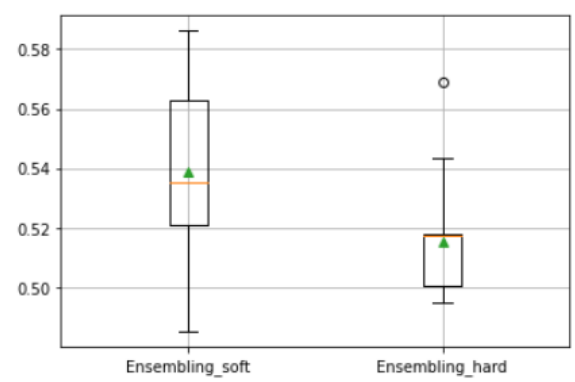

Fig. 8. Overall Ensembling Claasifier Result

We can see that we are able to achieve an accuracy of about 78% by ensembling but the f1 score is terrible.So when we keep f1 score the priority the KNN is best classifier.

VI. Future Works and Enhancements

This paper showcases the high potential cough sounds have in detection of Covid19. There have been various researches regarding the use of audio(cough) in detection of diseases like bronchitis, tuberculosis, pneumonia in which cough is a major symptom. Considering the above fact the cough sounds of other diseases can also be used to better generalise the model and get better accuracy. Many models developed so far lack sufficient amount of covid19 positive patients data

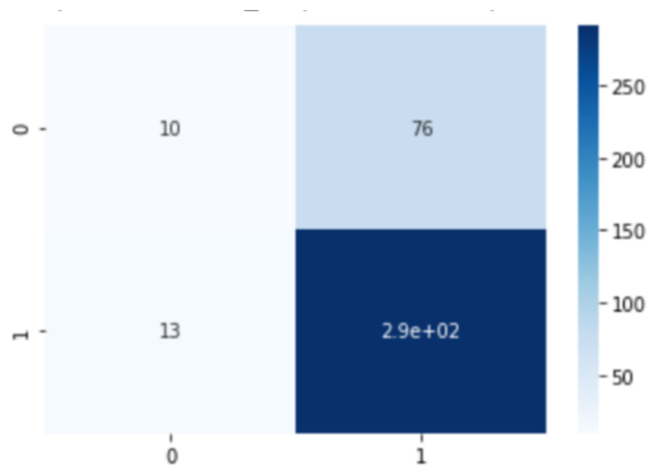

Fig. 9. Confusion Matrix of KNN

if the data can be increased then generalization can be made better. Selective extraction of features from the MFCCs can be used to get better generalization for the model. This analysis can be further expanded to other speech sounds like vowel sounds. Moreover the lack of data can be covered by the using the phone call data but this requires large amounts of preprocessing as one needs to filter other sounds and can capture only relevant audio data. A number of researches have been made regarding the same and its feasibility. This research paper will ignite further research in this area and will act as a guide for the endless possibilities we can have with the Audio data!

References

[1] Mouawad, Pauline, Tammuz Dubnov, and Shlomo Dubnov. "Robust Detection of COVID-19 in Cough Sounds." SN Computer Science 2.1 (2021): 1-13.

[2] World Health Organization. "World Health Organization coronavirus disease 2019 (COVID-19) situation report." (2020).

[3] Deshpande, Gauri, and Björn Schuller. "An overview on audio, signal, speech, & language processing for covid-19." arXiv preprint arXiv:2005.08579 (2020).

[4] Sharma, Neeraj, et al. "Coswara–A Database of Breathing, Cough, and Voice Sounds for COVID-19 Diagnosis." arXiv preprint arXiv:2005.10548 (2020). https://github.com/iiscleap/Coswara-Data.

[5] McFee, Brian, et al. "librosa: Audio and music signal analysis in python." Proceedings of the 14th python in science conference. Vol. 8. 2015.

[6] Dalmazzo, David, and Rafael Ramirez. "Mel-spectrogram Analysis to Identify Patterns in Musical Gestures: a Deep Learning Approach." MML 2020: 1.

[7] Mateen, Muhammad, et al. "Fundus image classification using VGG-19 architecture with PCA and SVD." Symmetry 11.1 (2019): 1.

[8] Cherif, Iyad Lahsen, and Abdesselem Kortebi. "On using eXtreme gradient boosting (XGBoost) machine learning algorithm for home network traffic classification." 2019 Wireless Days (WD). IEEE, 2019.

[9] Santosh, K. C. "Speech Processing in Healthcare: Can We Integrate?." Intelligent Speech Signal Processing. Academic Press, 2019. 1-4.

[10] Abeyratne, Udantha R., et al. "Cough sound analysis can rapidly diagnose childhood pneumonia." Annals of biomedical engineering 41.11 (2013): 2448-2462.

[11] Swarnkar, Vinayak, et al. "Automatic identification of wet and dry cough in pediatric patients with respiratory diseases." Annals of biomedical engineering 41.5 (2013): 1016-1028.

[12] Al-Khassaweneh, Mahmood, and Ra'ed Bani Abdelrahman. "A signal processing approach for the diagnosis of asthma from cough sounds." Journal of medical engineering & technology 37.3 (2013): 165-171.

[13] Boyanov, Boyan, and Stefan Hadjitodorov. "Acoustic analysis of pathological voices. A voice analysis system for the screening of laryngeal diseases." IEEE Engineering in Medicine and Biology Magazine 16.4 (1997): 74-82.

[14] Imran, Ali, et al. "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app." Informatics in Medicine Unlocked 20 (2020): 100378.

[15] Lella, Kranthi Kumar, and P. J. A. Alphonse. "A literature review on COVID-19 disease diagnosis from respiratory sound data." AIMS Bioengineering 8.2 (2021): 140-153.

[16] Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." IEEE Transactions on knowledge and data engineering 22.10 (2009): 1345-1359.

[17] Livingston, Frederick. "Implementation of Breiman's random forest machine learning algorithm." ECE591Q Machine Learning Journal Paper (2005): 1-13.

[18] Brown, Chloë, et al. "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data." arXiv preprint arXiv:2006.05919 (2020).

[19] Wright, Raymond E. "Logistic regression." (1995).

[20] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189-1232.

[21] Noble, William S. "What is a support vector machine?." Nature biotechnology 24.12 (2006): 1565-1567.

[22] Faezipour, Miad, and Abdelshakour Abuzneid. "Smartphone-based self-testing of covid-19 using breathing sounds." Telemedicine and e-Health 26.10 (2020): 1202-1205.

[23] Zhou, Quan, et al. "Cough Recognition Based on Mel-Spectrogram and Convolutional Neural Network." Frontiers in Robotics and AI 8 (2021).

[24] DeCarlo, Lawrence T. "On the meaning and use of kurtosis." Psychological methods 2.3 (1997): 292.

[25] Christodoulou, Evangelia, et al. "A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models." Journal of clinical epidemiology 110 (2019): 12-22.

[26] Sarangi, Lokanath, Mihir Narayan Mohanty, and Srikanta Pattanayak. "Design of MLP Based Model for Analysis of Patient Suffering from Influenza." Procedia Computer Science 92 (2016): 396-403.

[27] Le Cessie, Saskia, and Johannes C. Van Houwelingen. "Ridge estimators in logistic regression." Journal of the Royal Statistical Society: Series C (Applied Statistics) 41.1 (1992): 191-201.

[28] Liu, Jia-Ming, et al. "Cough detection using deep neural networks." 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2014.

[29] Amoh, Justice, and Kofi Odame. "DeepCough: A deep convolutional neural network in a wearable cough detection system." 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS). IEEE, 2015.

[30] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.

[31] Tsuruoka, Yoshimasa, Jun'ichi Tsujii, and Sophia Ananiadou. "Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty." Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. 2009.

[32] Lawrence, Steve, et al. "Face recognition: A convolutional neural-network approach." IEEE transactions on neural networks 8.1 (1997): 98-113.

[33] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017.

[34] Qi, Xingqun, Tianhui Wang, and Jiaming Liu. "Comparison of support vector machine and softmax classifiers in computer vision." 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE). IEEE, 2017.

[35] Yamashita, Hiroshi, and Hiroshi Yabe. "An interior point method with a primal-dual quadratic barrier penalty function for nonlinear optimization." SIAM Journal on Optimization 14.2 (2003): 479-499.