

Detection of Covid-19 through Cough and Breathing Sounds using CNN

Evangeline D¹, Sumukh M Lohit², Tarun R³, Ujwal K C⁴, Sai Viswa Sumanth D⁵

Assistant Professor, Department of ISE, M S Ramaiah Institute of Technology, Bangalore, India¹

Student, Department of ISE, M S Ramaiah Institute of Technology, Bangalore, India^{2,3,4,5}

Abstract—Covid-19 is declared a global pandemic by WHO due to its high infectivity rate. Medical attention is required to test and diagnose those with Covid-19 like symptoms. They are required to take an RT-PCR test which takes about 10-15 hours to obtain the result, and in some cases, it goes up to 3 days when the demand is too high. Majority of victims go unnoticed because they are not willing to get tested. The commonly used RT-PCR technique requires human contact to obtain the swab samples to be tested. Also, there is a shortage of testing kits in some areas and there is a need for self-diagnostic testing. This solution is a preliminary analysis. The basic idea is to use sound data, in this case, cough sounds, breathing sounds and speech sounds to isolate its characteristics and deduce if it belongs to a person who is infected or not, based on the trained model analysis. An Ensemble of Convolution neural networks have been used to classify the samples based on cough, breathing and speech samples, the model also considers symptoms exhibited by the person such as fever, cold, muscle pain etc. These Audio samples have been pre-processed and converted into Mel spectrograms and MFCC (Mel Cepstral Coefficients) are obtained that are fed as input to the model. The model gave an accuracy of 88.75% with a recall of 71.42 and Area Under Curve of 80.62%.

Keywords—Coronavirus; cough sounds; mel frequency cepstral coefficients; convolutional neural network; reverse transcription-polymerase chain reaction (RT-PCR)

I. INTRODUCTION

Covid19 is caused by the SARS-CoV2 Virus and was declared a pandemic across the world on the 11th of February 2020. Majority of Covid-19 patients experienced fever, dry cough and fatigue. Other symptoms experienced by the patients include aches and pains, sore throat, diarrhoea, conjunctivitis, headache, loss of taste or smell, a rash on skin, or discoloration of fingers or toes. This pandemic has affected more than 17 crores worldwide and has resulted in the death of 38 lakhs as of June 2021. Testing has become one of the most important requirements for starting the treatment, allocation of Beds, procurement of specific medicines etc. The current methods which are RT-PCR tests are conducted and samples are sent to a lab for disease detection. While Lateral Flow Tests (LFTs) can diagnose Covid-19 immediately, it is not as precise as RT-PCR. Antibody tests also cannot effectively detect Covid-19, but can determine an individuals' immunity to Covid-19. There could be false negatives up to 30% in RT-PCR tests. It means the presence of an infection could be done in far better way than giving a patient the all-

clear negative report. There is possibility of false positive results because of detecting dead and deactivated viruses in the body of a patient recovered from Covid-19.

The paper focuses on using deep learning techniques to detect Covid 19 using methods that don't involve any incision or wound to the patient while still finding ways to detect Covid-19. Cough and breathing based analysis to isolate cough sound snippets and convert this data that can be used for analysis and training a model based on the characteristics obtained. Convolutional neural network is used to analyze the images which are derived after preprocessing and Ensemble of models are used to increase the accuracy. For testing performed the performance will be measured based on the recall and accuracy rates. The applications would include testing methods within web applications, and sophisticated testing without one's aid.

The rest of the paper is organized as follows: Section II highlights the contemporary works carried out for Covid-19 detection in various countries. Section III discusses the materials and methods used in this work. Section IV gives the results obtained on carrying out the work. Section V summarizes the work carried out for Covid-19 detection.

II. LITERATURE REVIEW

In [1], the authors had analyzed the features for cough breathing and voice of the patients. Long term dependencies can be remembered in LSTM. Dataset consists of 60 healthy speakers and 20 Covid-19 patients. In comparison between the three through performance metrics, it is found that patients cough and breathing sounds are effective to diagnose infection as they have high recall. The Limitations of the papers include inefficient preliminary results due to time constraints, collected data is small and lacks control on other patients suffering from other kinds of respiratory illness.

In [2], a CNN model to detect Covid-19 from breath and cough sounds was proposed. Spectrogram extraction to obtain visuals of audio frequencies against time. CNN variant CIdER is based on ResNets which alleviates the vanishing gradient problem. The output is later given to a sigmoid layer. A score, thus obtained can be used to determine if a person is Covid-19 positive or not. Dataset used is 517 coughing and breathing recordings from 355 people, of which 62 participants had tested positive within 14 days of the recording. The Technique used is CNN-ResNet. The prime limitation of this study is size and demographics of the dataset.

The proposed work in [3] identified coughing is not just one predominant symptom of Covid-19 but a symptom of more than 100 diseases. Machine learning techniques can be applied on global smartphone recordings to detect Covid-19. Dataset used for training is the Coswara dataset and the Sarcos dataset. Techniques used are Logistic regression, SCM, MLP, LSTM, Resnet50.

In [4], the authors suggested that the only screening method for Covid-19 is a thermometer but only 45% of the mild-moderate patients had fever, their study suggested that Covid patients who are especially asymptomatic could be classified with a good accuracy as positive or negative with just the forced-cough recordings. The data was collected from opensigma.mit.edu and it had 5320 cough recordings. The cough recordings were converted using Mel Frequency Cepstral Coefficient (MFCC) and later on, fed into CNN containing one Poisson biomarker layer with biomarkers like Muscular degradation, Vocal cords, Sentiments, Lungs and Respiratory Tract and 3 pre-trained ResNet50's in parallel; 4256 recordings were used for training and the remaining 1064 was used for testing. The model had a recall of 98.5% and a specificity of 94.2%. For asymptomatic persons, it achieved recall of 100% with a specificity of 83.2%.

It was proposed in [5] that the lung volume and oxygenation can be modelled and approximated with a good accuracy. Energy of acoustic signal of respiration in each phase of airflow to/from the lungs from the breathing sounds is considered. Rhonchus, squawk, and stridor are also considered in the inspiration phase. All of the above characteristics are extracted from the lung function augmentation graph which is calculated against time and amplitude.

In [6], the authors have analyzed the computation of Mel spectrogram from cough sounds. Deep transfer learning based multi class / binary classifier along with classical machine learning based multi class classifier were applied to differentiate cough due to Covid-19 from cough due to other respiratory infections.

In [7], the authors have proposed a Radiographic scoring model applied by assessing disease severity using severity score. Score was given based on the extent of lung infection. Statistical analysis was performed by applying Student's T test, Mann Whitney's test, Chi-square test and Fisher's exact test. Some limitations were lack of retrospective analysis and correlation between CXR severity score and patient co morbidities.

The authors in [8] have made a comparative study of twelve deep learning algorithms using a multi-center dataset, including open-source and in-house developed algorithms. The 12 methods which are Benchmarked and compared include Lung segmentation for severe pathologies, 3D Lung segmentation Lobe segmentation, CT Angel software for lung segmentation and binary lesion, CovidENet, 2D Unet, 3D multiclass segmentation, Inf-Net, WASS, UNWM and Majority voting.

In [9], it was proposed that RT-PCR is probably a more accurate and sensitive strategy. GradCAM mappings and 3-D

model were applied. Pretest background prevalence determines test success metrics. Testing practices may differ depending upon exposure rates and pandemic phases. The work was limited in truly evaluating the generalizability of this model to an independent population, because positive and negative cases were obtained from separate populations.

In [10], the authors have proposed some Non Clinical techniques such as machine learning, data mining, expert system and other artificial intelligence techniques must play significant roles in diagnosis and containment of the Covid-19 pandemic to detect asymptomatic cases and for accelerating the testing process. Supervised ML models were developed using decision tree, logistic regression, Naive Bayes, SVM and ANN.

Detection of Covid-19 by using both cough recording and the uttering of the vowel sounds was proposed in [11]. Many classifiers like decision trees, support vector machines, K-Nearest-Neighbor, Random Forest (RF) and XGBoost were used. The best performance was shown by the weighted XGBoost classifier. A larger number of X-ray images with a wider distribution could have been used.

The authors in [12] have proposed detection of Covid-19 using cough and breathing sounds. Crowdsourced dataset was used for analysis. The model was trained using two feature sets - one was the handcrafted features like the MFCC (Mel Frequency Cepstral Coefficients) and other statistical features and the other feature set was obtained from the pre-trained models [14] [17][21].

III. MATERIALS AND METHODS

A. Materials

The proposed solution provides a web application that takes cough, breathing and speech sounds along with the symptoms shown by the person as input and predicts the likelihood of him/her having Covid-19. The training process of the machine learning model begins with pre-processing techniques like loading the audio samples and removing null values. The features of the audio samples like MFCCs and Mel Spectrogram images are further extracted from the audio samples and given to the input generator function. Mel-Frequency Cepstrum Coefficients (MFCCs) are coefficients that provide a representation of the short-term power spectrum of a sound.

A Mel spectrogram is a spectrogram where the frequencies are transformed to Mel scale. This converts the audio samples to image form. The input generator function divides the training data into batches and shuffles the order of the examples so that batches between epochs do not look alike. The data generated from the input generator function is then fed to the machine learning model. Next, an ensemble of three convolutional neural network models and four dense neural network models are used to analyze the pre-processed data and predict the desirable output. Each CNN model consists of 3 2D convolutional layers, average pooling, batch normalization, Relu activation function and a dropout model. Further, the input is flattened using the Flatten class. Glorot Uniform initializer is used to initialize layer weights. Each dense neural network consists of two dense layers using Relu

activation function and Glorot Uniform initializer and a dropout layer. The concatenated ensemble model is then fed through two hidden layers and an output layer to finally generate the output.

The outline of the overall Workflow of the model from start to finish is as follows: The flow starts with the user recording his voice and the symptoms shown by him, the application after receiving the sample runs separate pre-processing steps for voice and categorical features, audio features are converted to Mel spectrograms, MFCC, along with the binary symptoms are fed into the ensemble model which tells if the patient is positive or not.

Fig. 1 gives a brief overview of the Machine Learning model employed in the proposed work. It is an ensemble model of 3 CNN models and 4 dense neural network models. The 3 CNN models take Mel Spectrogram images as input, 3 dense neural network models take MFCCs as inputs and the other dense neural network takes symptoms as input. The ensemble model is then fed through the two hidden layers and an output layer to derive the output. The model consists of an ensemble model of 3 CNN models and 4 dense neural network models. The 3 CNN models each consist of 3 Conv-2D layers,

3 Average Polling 2D layers, 3 Batch Normalization layers, 3 dropout layers and 3 Relu activation functions. The CNN models flatten the input before sending the input through the final dropout layer, dense layer and activation function. The 4 dense neural network models consist of 2 dense layers and 2 dropout layers [16],[18],[26].

Three CNN models were selected because out of the 7 inputs,

The inputs were image inputs i.e Mel spectrogram images of Cough, Breathing and Speech. Since CNN's are very effective in Image classifications because Images have higher dimensions and CNN is very good at reducing the number of parameters without reducing the quality makes it the best option to consider as the model. The 3 CNN models take images as input, 3 dense neural network models take MFCCs as input and the other dense neural network models take symptoms as input. Glorot Uniform is used as the initializer for each of the models. The ensemble model is then passed through 2 dense hidden layers which then feed the input into the dense output layer. The hidden layers use Relu activation function. The output layer finally makes the prediction for the input. The output layer uses the sigmoid activation function.

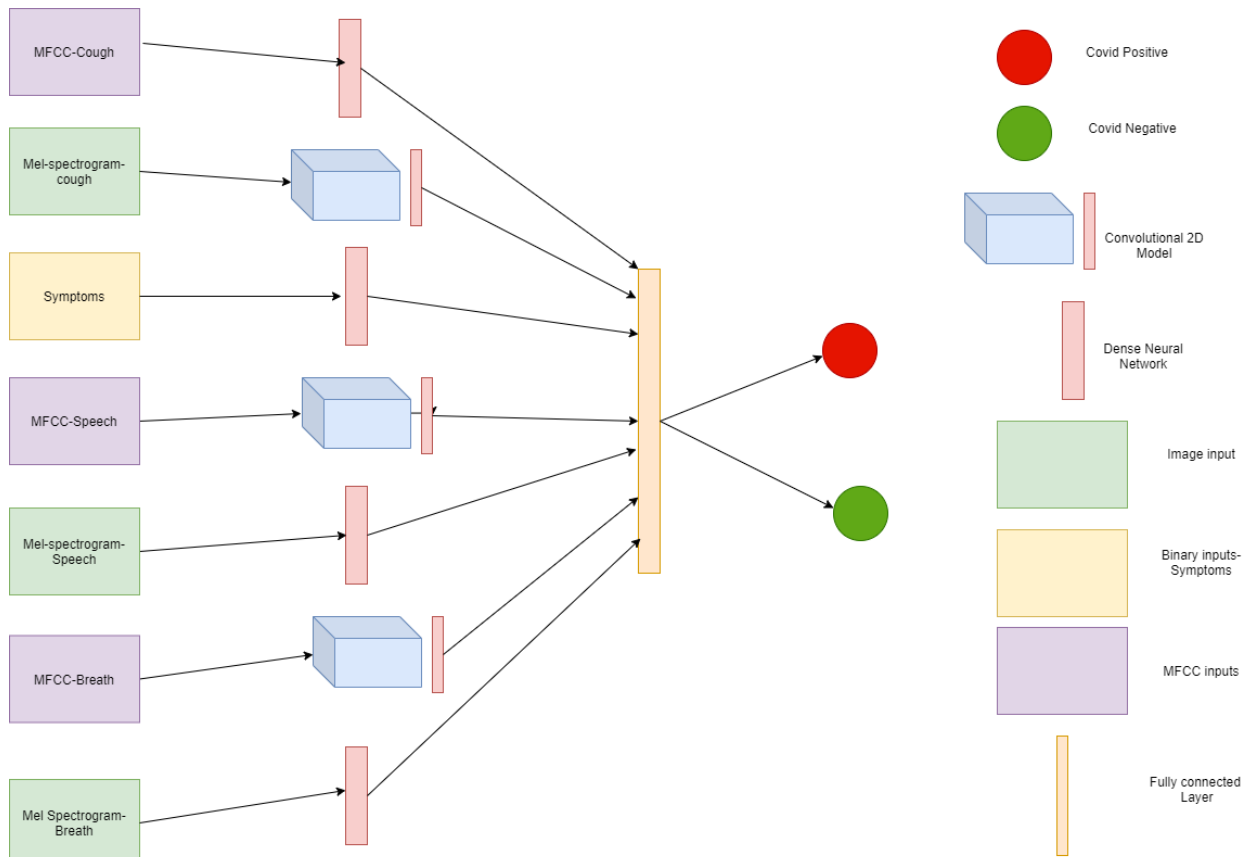


Fig. 1. Brief overview of the Model.

B. Dataset

DiCOVA challenge employs a dataset for respiratory health diagnosis by speech and audio processing [13]. In this paper, the data used is taken from Coswara data which was collected by Indian Institute of Science (IISc) Bangalore. The dataset includes voice samples including fast and slow breathing sounds, deep and shallow cough sounds, phonation of sustained vowels, and counting numbers at slow and fast pace. Collected metadata includes the participant's age, gender, location country, state/ province of the participant, current health status that could be healthy, exposed, cured or infected and the presence of comorbidities like pre-existing medical conditions. The dataset consists of 1645 samples of individuals. The data was sampled from all the continents except Africa and more than 88% of the samples were from India. 74% of samples are male and around 26% are females. 76.8% of samples are healthy individuals; 8.4% of samples are positive and the rest can't be identified. Majority of the samples are those between 20-30 years of age followed by 30-40 years and followed by 40-50 years. Fig. 2 shows Pearson's correlation heatmap for the dataset. It can be inferred from the above figure that the most important symptoms of Covid-19 status are asthma, fever, cough, sore throat, breathing difficulty, cold, fatigue, muscle pain, loss of smell and pneumonia in that order.

C. Data Preprocessing

Sound, represented as an audio signal possesses characteristics like frequency, bandwidth, decibel, etc. Usually, such signals are given as a function of amplitude and time. Audio processing involves extraction of acoustics features relevant to a task. Librosa [19][23][24], a Python for music and audio analysis facilitates various options to construct music information retrieval systems. A Mel

spectrogram is a spectrogram where the frequencies are transformed to the Mel scale. The Fourier transform maps continuous time into a frequency spectrum, but an inverse is performed over its log to make it perceptible by humans. General data preprocessing is finally done using Python libraries [22]. In brief, the process of obtaining Mel – Spectrogram is as follows:

- 1) The samples of air pressure are collected at different instances of time and the same digitally represents an audio signal.
- 2) The audio signal is transformed from time domain to frequency domain using the Fast Fourier transform.
- 3) Frequency on y axis is converted to a log scale and amplitude is converted to decibels and the spectrogram is subsequently formed.
- 4) Frequency is again mapped onto Mel scale and Mel spectrogram is obtained.

D. Mel-Frequency Cepstral Coefficients (MFCCs) and Feature Extraction

MFCCs consist of 10–20 features usually. These features give the general shape of a spectral envelope and models the characteristics of the human voice. To get MFCC, DCT on the Mel-spectrogram is computed. To obtain MFCC, the following steps could be carried out.

- 1) A windowed excerpt of a signal is taken and Fourier Transform is applied on it.
- 2) Powers of the spectrum thus obtained are mapped onto Mel scale using triangular overlapping windows.
- 3) Logarithms of the powers at each Mel frequency are computed.
- 4) DCT of such MEL log powers is calculated.
- 5) Amplitudes of the resulting spectrum give MFCCs



Fig. 2. Pearson Correlation Heatmap for the Dataset

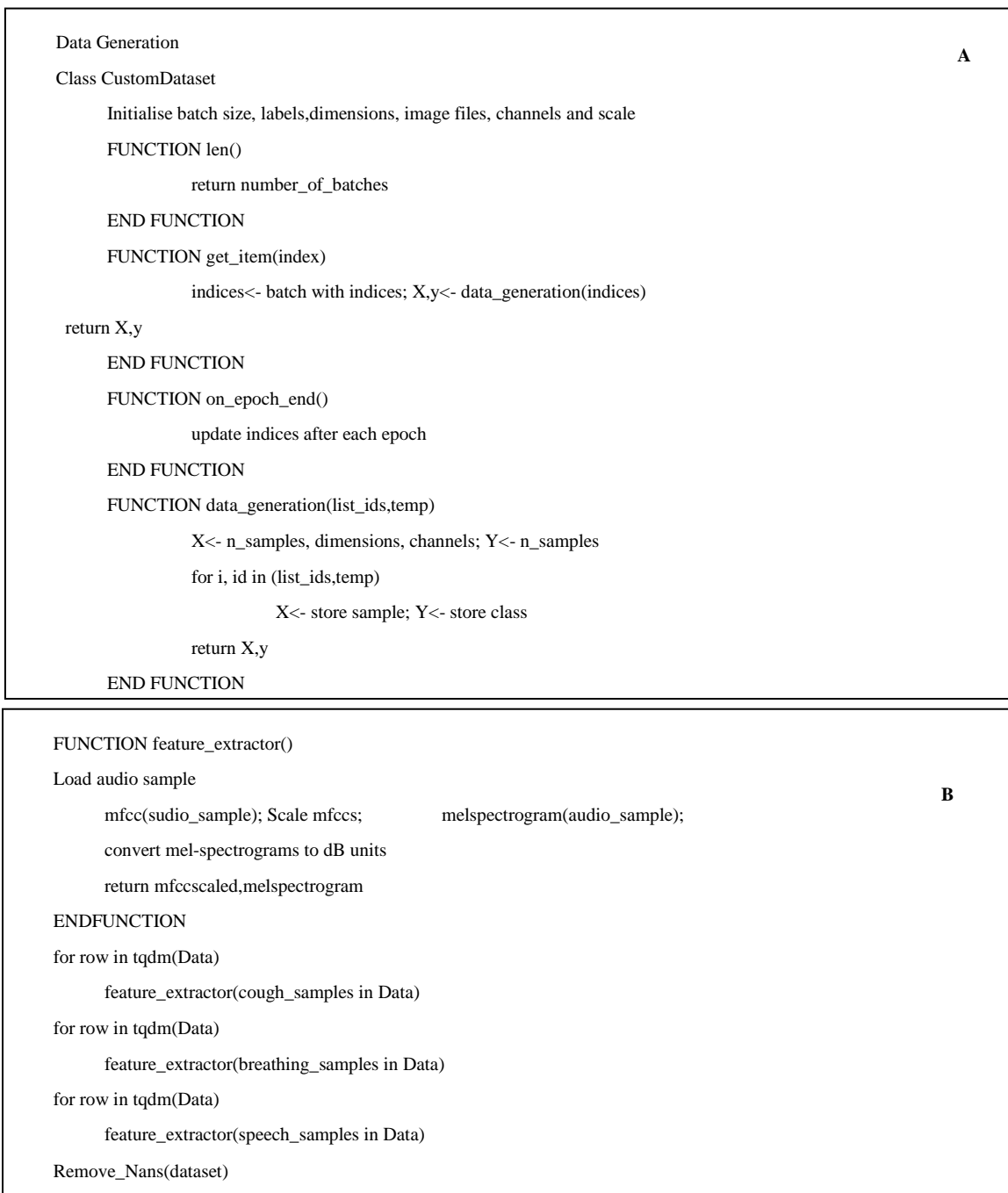


Fig. 3. Psuedocodes (A) Data Generation (B) Feature Extraction.

The following is the six step process to obtain the features.

- 1) The signals are split into short-time frames.
- 2) Windowing is applied.
- 3) An NN-point FFT on each frame is applied and frequency spectrum also called Short-Time Fourier-Transform (STFT) is obtained.
- 4) Filter Banks are applied. Actually, a set of 20–40 triangular filters is employed.

- 5) Logarithm of these spectrogram values is applied to get log filter bank energies.
- 6) DCT (Discrete Cosine Transform) is applied.

E. Convolution Neural Network

In Convolutional 2D filters, Keras Conv2D parameter determines the number of kernels to convolve with the input volume resulting in a 2D activation map. Average Pooling calculates the average value for patches of a feature map and creates a down sampled (pooled) feature map.

```
Model-training
FUNCTION build_model()
    //First, third, fifth and seventh model,
    Input()
    Dense(activation='relu',kernel_initializer='GlorotUniform')
    Dropout()
    Dense()
    Dropout()
//Second, Fourth and sixth model
    Input()
    Conv2D()
    AveragePooling2D()
    BatchNormalization()
    Activation('relu')
    Conv2D()
    AveragePooling2D()
    BatchNormalization()
    Activation('relu')
    Conv2D()
    AveragePooling2D()
    BatchNormalization()
    Activation('relu')
    merge<- Concatenation of seven models
    hidden1<-Dense(activation='relu')(merge)
    hidden2<-Dense(activation='relu')(hidden1)
    output<-Dense(activation='sigmoid')(hidden2)
for all splits:
    build_model()
    Compile_model(metric='AUC',optimizer = 'Adam',loss = 'BinaryCrossentropy')
    Split data into train,validation and test sets of ratio 0.7:0.15:0.15
    SevenInputGenerator(train_data)
    SevenInputGenerator(validation_data)
    SevenInputGenerator(test_data)
    model.fit(training_data)
END FUNCTION
predict(model,train_data); predict(model,validation_data); predict(model,test_data)
Evaluate metrics on train, test and validation data; Visualise the results
```

Fig. 4. Psuedocode (C) Model Training.

In Average Pooling layer the average for each block is computed rather than a max value, which is the case with max pooling. Batch Normalization (BN) is performed as a solution to speed up the training phase of deep neural networks through the introduction of internal normalization of the input values within the neural network layer. Dropouts can prevent overfitting in the model and can be added to randomly switch

some percentage of neurons of the network. When the neurons are switched off, the incoming and outgoing connection to those neurons is also switched off. The dense layer is a deeply connected neural network layer with each neuron in the dense layer receiving input from all neurons of the preceding layer. Thus the dense layer results in an 'm' dimensional vector.

```
Class CustomPipeline
  Initialise batch size, labels,dimensions, image files, channels and scale
  FUNCTION len()
    return number_of_batches
  END FUNCTION
  FUNCTION get_item(index)
    indices<- batch with indices
    X,y<- data_generation(indices)
    return X,y
  END FUNCTION
  FUNCTION on_epoch_end()
    update indices after each epoch
  END FUNCTION
  FUNCTION data_generation(list_ids,temp)
    X<- n_samples, dimensions, channels
    Y<- n_samples
    for i, id in (list_ids,temp)
      X<- store sample
      Y<- store class
    return X,y
  END FUNCTION
Class SevenInputGenerator
  X1<- CustomPipeline(X1,Y)
  X2<- CustomDataset(X2,Y)
  X3<- CustomPipeline(X3,Y)
  X4<- CustomDataset(X4,Y)
  X5<- CustomPipeline(X5,Y)
  X6<- CustomDataset(X6,Y)
  X7<- CustomPipeline(X7,Y)
  FUNCTION get_item(index)
    X1_batch, Y_batch = X1.getitem(index)
    X2_batch, Y_batch = X2.getitem(index)
    X3_batch, Y_batch = X3.getitem(index)
    X4_batch, Y_batch = X4.getitem(index)
    X5_batch, Y_batch = X5.getitem(index)
    X6_batch, Y_batch = X6.getitem(index)
    X7_batch, Y_batch = X7.getitem(index)
    X_batch=[X1_batch,X2_batch,X3_batch,X4_batch,X5_batch,X6_batch,X7_batch]
  return X_batch_Y_batch
  END FUNCTION
```

Fig. 5. Psuedocode (D) Processing.

Units are the fundamental parameter that takes positive integers usually. This parameter determines the size of the weight matrix along with the bias vector. The activation parameter helps to apply element-wise activation function in a

dense layer. By default, Linear Activation is used but any of the options that Keras [16][18][24] provides can be used for this. In the model Relu has been used as the activation function and it is used for activation to avoid gradient descent,

It also avoids vanishing gradient problem from sigmoid function. The rectified linear activation function or ReLU for short is a piecewise linear function that maps input to output for positive values. The initializer parameter deals with initialization of values in the initialization layer. In the Dense Layer, the weight matrix and bias vector has to be initialized. The initializer used here called glorot_uniform draws samples from a uniform distribution within [-limit, limit] where, $limit = \sqrt{6/fan_in + fan_out}$ where, fan_in is the number of input units in the weight tensor and fan_out is the number of output units. The algorithms employed in this work are mentioned in Fig. 3, Fig. 4 and Fig. 5.

IV. RESULTS AND DISCUSSION

Through the analysis of the Pearson's correlation matrix we found that a few symptoms are particularly highly correlated and other symptoms didn't have much weightage so only highly influential symptoms were provided as the option. The inclusion of the 7 inputs in itself has the information equal to that of many inputs because the inputs like Mel spectrogram and MFCC have a lot of parameters associated with them, each parameter represent different feature which makes it so that we are considering enough inputs without getting to the scale of overfitting. The 7 inputs include Mel spectrogram and MFCCs of cough, speech and breathing samples respectively along with another input that included symptoms. MFCC inputs further have 39 coefficients each.

The user interface for the web application is developed [15] [20]. It shows the input given to the Covisound-Covid detection website. Symptoms are given as inputs. The symptoms considered are fever, muscle pain and respiratory problems (asthma, breathing difficulty, cold and cough). It also shows the result displayed on the Covisound-Covid detection website. The website displays the result of the Covid-19 test, negative in this case and how likely it is that the person has Covid-19 is 5.82%.

Fig. 6 shows the input given to the Covisound-Covid detection website. Breathing sounds, cough sounds, speech sounds and symptoms are given as inputs. The application shows the result displayed on the Covisound-Covid detection website. 60% of the samples were used for training, 20% for testing and 20% for validation. The total number of samples considered after preprocessing are 1605. Therefore, the number of training and testing samples is 963 and 321 respectively. The website displays the result of the Covid-19 test, negative in this case and the likelihood of the person having Covid-19 is 72.37%. Evaluation metrics considered here include accuracy, recall, AUC Score, ROC Curve, False Positive Rate and False Negative Rate [25]. The five models used for analysis have a few attributes and parameter changes between them; this is performed to see how the prediction varies based on the values set. We set a different training environment for the data to deduce the best possible trade-off over different variations.

Five models are trained by having a binary cross entropy loss function with different metrics for optimising the model. The first model uses Area under the curve as the metric for optimising the model and thereby maximising the same parameter. For imbalance data handling, class weights

parameter was provided in the ratio of 1:10 (each negative sample gets ten times the weightage of a positive sample). A high AUC value is obtained here. In our second model the metric that was tuned used is accuracy so that the model focuses on the accuracy of each function and it maximises the same in each iteration. Here the trade-off made is recall which was very low and not favourable for this particular paper as the main aim is to have a high recall to reduce false negatives. In the Third model, the metric tuned was true negatives, the model gave considerable high accuracy and true negatives but it had one of the lowest recall when compared to all the other models. The false negatives rates obtained were also high, so this model was not very favourable. In the fourth model the metric used for optimising the model is recall, this attribute describes the number of positive predictions determined out of all True positive predictions. Recall also shows how well the positive class is covered in the predictions. The fifth model uses Area under the curve as the metric for optimising the model and maximises the parameter. The reason being this parameter shows the true variance between true positives and true negatives which leads to better distinguishability between them. Higher AUC means the model is able to deduce the right class it belongs to. Scoring metrics for different models are tabulated below in Table I. In Fig. 7A, ROC curves of a few models trained on the dataset in different epochs is shown. It also shows the AUC for the respective models. Fig. 7B shows the variation in AUC values vs different epochs for the model. The graph shows a steady improvement in AUC value as the epoch number increases.

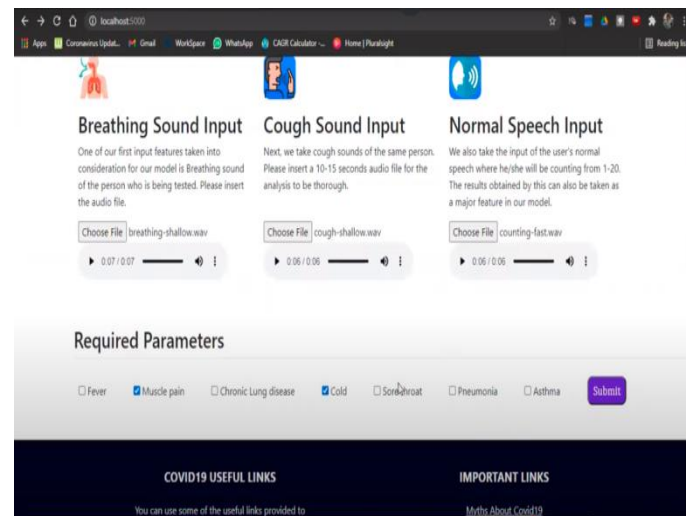


Fig. 6. Covisound- Covid Detection Website Inputs-Positive Sample.

TABLE I. SCORING METRICS FOR DIFFERENT MODELS

Model	Accuracy	Recall	AUC-Score	False positive rate	False negative rate
Model-1	88.75%	71.42%	80.62%	10.18%	28.57%
Model-2	95%	44.44%	71.77%	0.9%	55%
Model-3	92.91%	60%	77.95%	4%	40%
Model-4	83.75%	75%	79.86%	15.27%	25%
Model-5	90.42%	63.16%	77.96%	7.23%	36.84%

V. CONCLUSION

The proposed work showcases the possibility of using an ensemble of Convolutional Neural Networks as a testing method for Covid-19. An encouraging result was obtained in the paper by taking speech, cough and breathing sounds as inputs. Coswara data has been used in this work. MFCCs and Mel spectrogram images were obtained to extract features from the audio samples.

Seven inputs were fed to the model- MFCC cough, MFCC speech, MFCC breath, Mel spectrogram cough, Mel spectrogram breath, Mel spectrogram speech and symptoms. An ensemble of three CNN models and four dense neural network models was used for the purpose of classification. The machine learning model is able to identify Covid-19 patients with a recall of 71.42% and an AUC of 80.62%. The model performs much better than the baseline model in which had an AUC of 70%. An easy-to-use web application with a friendly user interface has also been developed as part of the work. The web application predicts whether the individual is Covid-19 positive or negative and also determines the likelihood of the individual having Covid-19. The tradeoffs between recall and the overall accuracy have also been compared in this paper. Some models had high accuracy but low recall whereas some had high recall but low accuracy. So, the best model was chosen in such a way that both recall and accuracy had decent values. Use of class weights was also attempted as part of the work to improve data imbalance handling. This work is able to reduce false negative rate and improve recall appreciably and is a good preliminary analysis tool for distinguishing Covid-19 affected individuals from healthy individuals.

Recall and false negative rate can be improved further. Dataset used is relatively small and imbalanced; a larger dataset may lead to improved results. As the dataset size increases, the model can be retrained with new data. Dataset was heavily imbalanced with positive samples contributing to less than 9% of the total data and less than 150 samples. Imbalanced data handling can be further improved in the future.

REFERENCES

- [1] Abdelfatah Hassan, Ismail shahin, mohamed Bader Alsabek. Covid 19 Detection using RNN (through breathing, cough sounds and talking. 2020.
- [2] Alexander Gaskell, Panagiotis Tzirakis, Alice Baird, Lyn Jones, Björn W. Schuller. End-2-End COVID-19 Detection from Breath & Cough Audio, Harry Coppock. 2021.
- [3] Madhurananda Pahar, Marisa Klopper, Robin Warren, and Thomas Niesler. COVID-19 Cough Classification using Machine Learning and Global Smartphone Recordings. 2020.
- [4] Jordi Laguarda, Ferran Hueto, Brian Subirana. COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings. 29 September 2020.
- [5] Miad Faezipour and Abdelshakour Abuzneid. Smartphone-Based Self-Testing of COVID-19 Using Breathing Sounds. 1 Oct 2020.
- [6] Ali ImranabIryna, Posokhovabc, Haneya, N.Qureshia, Usama, Masooda, Muhammad, Sajid, Riaza Kamran Ali, d, Charles, N. Johna MD, Iftikhar Hussain, be Muhammad Nabeel. AI enabled preliminary diagnosis for COVID-19 from cough samples via an app. 2020.
- [7] Rabab Yaasin and Walaa Gouda. Chest X-ray findings monitoring COVID-19 disease course and severity. 2020.

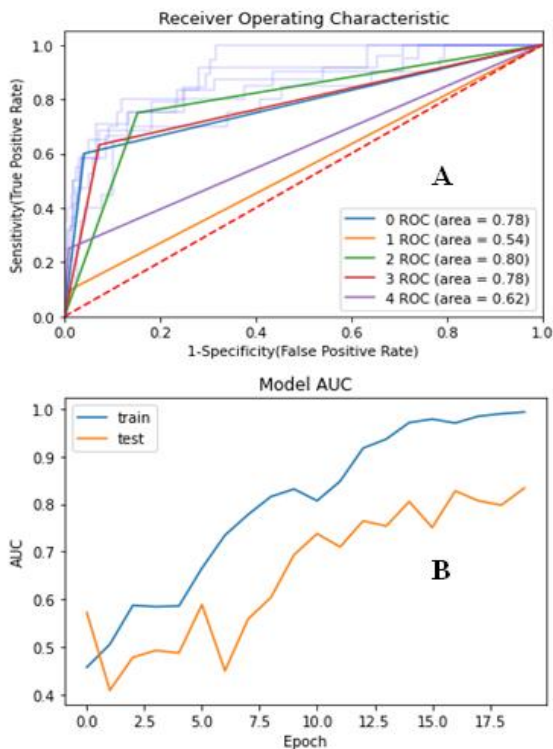


Fig. 7. (A) ROC Curves of a Few Trained Models (B) AUC Value vs epoch Numbers.

The following are the observations made after analysing the results: The best model for the paper was chosen based on AUC and recall. Model 1 gave the best recall and AUC values of 71.42% and 80.6% respectively. It gives an accuracy of 88.75%, a false negative rate of 28.57% and a false positive rate of 10%. A slight trade-off between accuracy and recall was observed while choosing the best model. Some of the models had an accuracy of 96.67% and 97% but had low recall whereas some of the other models had a high recall of >80% and close to 90% but they had low accuracy. So, the best model had to be chosen in such a way that it gave decent values for both accuracy and recall. AUC was chosen as the metric while training the model. Training AUC of 99.8%, validation AUC of 91% and a test AUC of around 81% were observed.

Binary cross entropy was chosen as the loss function while training the model. As the epoch number increased, the AUC value increased and the loss value-binary cross entropy decreased for test, train as well as the validation data. Choosing class weights while training increased the overall performance of the models but the best model was obtained while training without the class weights. The model performs much better than the baseline model in which had an AUC of 70%. The work shows that Covid-19 can be determined using the characteristics of an individual related to speech, breathing and cough sounds. The symptoms and existing conditions can further help in determining Covid-19 status. The model is able to predict the Covid-19 status of an individual reasonably and also the likelihood of the individual having Covid-19.

- [8] Sofie Tilborghs, Ine Dirks, Lucas Fidona, Siri Willems, Tom Eelbode, Jeroen Bertels, Bart Ilsen, Arne Brys, Adriana Dubbeldam, Nico Buls, Panagiotis Gonidakis, Sebastian Amador Sanchez, Annemiek Snoeckx, Paul M. Parizel, Johan de Mey, Dirk Vandermeulen, Tom Vercauteren, David Robben, Dirk Smeets, Frederik Maes, Jef Vandemeulebroucke, Paul Suetens. Comparative study of deep learning methods for the automatic segmentation of lung, lesion and lesion type in CT scans of COVID-19 patients. 2020.
- [9] Stephanie A. Harmon, Thomas H. Sanford, Sheng Xu, Evrim B. Artificial intelligence for the detection of COVID-19 pneumonia on chest CT using multinational datasets. 2020.
- [10] L. J. Muhammad, Ebrahim A. Algehyne, Sani Sharif Usman, Abdulkadir Ahmad, Chinmay Chakraborty & I. A. Mohammed. Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset. 2020.
- [11] Pauline Mouawad, Tammuz Dubnov, Shlomo Dubnov. Robust Detection of COVID-19 in Cough Sounds. 12th Jan 2021.
- [12] Chloë Brown, Jagmohan Chauhan, Andreas Grammenos, Jing Han, Apinan Hasthanasombat Dimitris Spathis, Tong Xia, Pietro Cicuta Cecilia Mascolo. Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data. 18th Jan 2021.
- [13] A. Muguli, L. Pinto, N. Sharma, P. Krishnan, P. K. Ghosh, R. Kumar, S. Ramoji, S. Bhat, S. R. Chetupalli, S. Ganapathy et al. Dicova Challenge: Dataset, task, and baseline system for COVID-19 diagnosis using acoustics. 2021.
- [14] <https://www.sciencedirect.com/topics/computer-science/cepstral-coefficient>.
- [15] <https://flask-doc.readthedocs.io/en/latest/>.
- [16] https://keras.io/api/layers/convolution_layers/convolution2d/.
- [17] <https://towardsdatascience.com/learning-from-audio-the-mel-scale-mel-spectrograms-and-mel-frequency-cepstral-coefficients-f5752b6324a8>
- [18] https://keras.io/api/layers/core_layers/dense/.
- [19] <https://librosa.org/doc/latest/index.html>.
- [20] <https://blog.getbootstrap.com/>.
- [21] <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>.
- [22] <https://scikit-learn.org/stable/modules/preprocessing.html>.
- [23] <https://towardsdatascience.com/how-to-apply-machine-learning-and-deep-learning-methods-to-audio-analysis-615e286fcbbc>.
- [24] <https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly>.
- [25] <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38>.
- [26] <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>.