Deep learning for COVID-19 classification using cough audio.

Sarthak Vajpayee[a,*], Madhurananda Pahar[b], and Thomas Niesler[c]

[a]*Department of Electrical and Electronics Engineering, University of Stellenbosch, South Africa.*[1]
[b]*Department of Electrical and Electronics Engineering, University of Stellenbosch, South Africa.*
[c]*Department of Electrical and Electronics Engineering, University of Stellenbosch, South Africa.*

Abstract

We present a deep learning based COVID-19 cough classifier, which can discriminate between audio recordings of coughing by COVID-19 positive subjects and COVID-19 non-positive (COVID-19 negative and healthy) subjects. This classifier represents means of early screening for COVID-19 that does not require the subject to be present at the testing location. Since it can be made available remotely on a mobile device such as a smartphone, the risk of COVID-19 transmission due to contact while testing can be eliminated. The experimental evaluation is based on a blend of two crowdsourced and publicly available datasets. The first is COSWARA, compiled by IISC-Bangalore and containing the audio recordings of 1184 subjects who are either healthy or have recovered from COVID-19 and 107 subjects who are COVID-19 positive. The second is COUGHVID, compiled by EPFL and containing audio recordings of 1133 subjects who are healthy and 397 subjects who are COVID-19 positive. The machine learning algorithms considered for the classification of the audio are a multilayer perceptron (MLP), a GRU-based recurrent neuralnetwork (RNN), a convolutional neural network (CNN), an OxfordNet (VGG-16) and a ResNet-50. Results show that the pre-trained ResNet-50 neural network architecture outperformed the other candidates, achieving a specificity of 90%, a sensitivity of 93%, an AUC score of 0.96, and an accuracy of 92.78%

Keywords: `COVID-19, cough, audio, deep-learning, classification, Multilayer Perceptron (MLP), Recurrent Neural Network (GRU), Convolutional Neural Network (CNN), OxfordNet (VGG-16), ResNet-50, Transfer learning.`

1. Introduction

COVID-19 (COronaVirus Disease of 2019), is a contagious disease caused by SARS-CoV-2 (Severe Acute Respiratory Syndrome Corona Virus 2). With the first case identified in Wuhan, China in December 2019, it has since spread around the world and was classified as a global pandemic on 11th February 2020 by the WHO (World Health Organization) [1]. The SARS-CoV-2 virus spreads through direct contact with respiratory droplets and aerosols produced by an infected person, generated as they breathe, cough, sneeze or speak. Some common symptoms of COVID-19 include dry coughing, fever, fatigue, breathing difficulties, loss of taste and smell, and muscle pain [2] [3]. Because this virus has the ability to mutate and generate new strains, it is difficult to develop a vaccine or medical treatment. Therefore, the WHO has recommended increased testing and social distancing in regions affected by the COVID-19 pandemic.

For diagnosis, RT-PCR (Reverse Transcription Polymerase Chain Reaction) [4] is a standard diagnostic method adopted to detect the COVID-19 infection in suspected individuals. As of 2nd October 2021, more than 564 billion COVID-19 tests have been performed around the world [5]. Although RT-PCR has been globally adopted as a gold standard, its limitations include a long turnaround time (the test takes between 4 and 48 hours to give the results [6]) and reported false negative rates as high as 33%, equating to sensitivity of 67% [7]. The equipment required to perform the test also has a high cost, making it less affordable to economically weaker regions. In the past 22 months, more than 235 million cases of COVID-19 have been reported worldwide, resulting in more than 4.8 million deaths. In several cases COVID-19 testing facilities have been overloaded [8].

The last decade has seen a rise in the application of machine learning based approaches across various domains due to the increased availability of data. Since the onset of COVID-19, researchers have applied such techniques to detect early symptoms. For example, a ResNet-50 based network was trained on CT (Computed Tomography) images of lungs to detect COVID-19 [9, 10, 11]. Similarly, X-ray images of lungs have been used to detect COVID-19 with the help of a CNN (Convolutional Neural Network) [12, 13, 14, 15]. One study shows that machine learning models can be used to predict whether a patient suffers from bronchitis, bronchiolitis, or per-

*Corresponding author. Tel: +91-9919191103
Email address: `itssarthakvajpayee@gmail.com` (Sarthak Vajpayee)
[1]Work performed while a collaborator with Stellenbosch University.

tussis based on audio recordings of their coughing sounds with accuracy over 89% [16, 17, 18, 19]. Other studies have used similar approaches to determine whether these coughing sounds are a symptom of COVID-19 or not [20, 21, 22]. For example, a crowdsourced dataset of respiratory sounds produced by healthy individuals and by sufferers of COVID-19 was compiled in [23]. A binary classifier was shown to achieve an AUC (area under the ROC curve) above 0.8 across various tasks. Another study used two crowdsourced datasets (COSWARA and Sarcos) of cough audio recordings to train a ResNet-50 classifier that distinguishes between the coughs of COVID-19 and healthy subjects. A Resnet-50 architecture was shown to achieve an AUC of 0.976 and an accuracy of 95.3% while an LSTM (Long Short Term Memory) recurrent architecture subjected to SFS (Sequential Forward Search) achieved an AUC of 0.938 on the held-out Sarcos dataset [24].

Collecting cough audio from COVID-19 patients is challenging and many of the resulting datasets are not publicly available. Among the freely available datasets, project NoCoCoDa includes annotated coughing audio from COVID-19 subjects collected through public interviews [25]. Project Coswara, executed by IISC-Bangalore, is a regularly-updated crowdsourced open-access dataset of respiratory sounds including cough and breath, as well as speech [26]. At the time of writing, this dataset contains audio recordings from 1184 healthy participants and 107 COVID-19 subjects. Project Coughvid, executed by EPFL, also strives to develop a validated crowdsourced COVID-19 cough audio dataset [27]. At the time of writing, this dataset contained more than 20,000 audio recordings of which more than 2,000 had been labeled by experienced pulmonologists to diagnose medical abnormalities present in the coughs. This makes Coughvid one of the largest expert-labeled cough datasets.

This study is based on the audio cough recordings found in the two publicly available crowdsourced datasets Project Coswara and Project Coughvid. Both datasets are imbalanced and contain a much smaller number of COVID-19 coughs than other coughs. After preprocessing, the Coswara dataset includes 107 COVID-19 positive and 1184 COVID-19 negative samples. The Coughvid dataset contains 397 COVID-19 positive and 1133 COVID-19 negative samples. For our work, the two datasets were merged to create a single dataset with a total of 2821 samples, of which 504 corresponded to COVID-19 positive and 2317 to COVID-19 negative subjects. To balance this dataset, the Synthetic Minority Oversampling Technique (SMOTE) was used [28].

Five deep-learning architectures, MLP, RNN, CNN, VGG-16, and ResNet-50, were trained, optimised and tested using K-fold cross-validation. Model performance was compared based on specificity, sensitivity, AUC and accuracy. ResNet-50 was seen to outperform the other models, achieving a specificity of 90%, a sensitivity of 93%, an AUC of 0.96 and an accuracy of 92.78%.

2. Data description

2.1. The Coswara dataset

The Coswara project, executed by the Indian Institute of Science (IISc) in Bangalore, aims to build a diagnostic tool for COVID-19 detection based on respiratory, cough, and speech audio recordings. Crowd-sourcing is used to collect the data via a web-based interface. The Coswara interface records the user's heavy and shallow coughing, fast and slow breathing, spoken English vowels, digits from one to ten and metadata including geographical location, age, gender and pre-existing medical conditions. Finally, the current COVID-19 health status must be specified according to the following seven classes:

- Healthy
- No respiratory illness
- Respiratory illness not identified
- COVID-19 positive with no symptoms
- COVID-19 positive with mild symptoms
- COVID-19 positive with moderate symptoms
- Fully recovered from COVID-19

Data collection began in April of 2020 and the dataset is being regularly annotated and updated. Audio recordings are sampled at 44.1 kHz, but in order to ensure uniformity with the Coughvid data, these recordings are downsampled to 22.4 kHz. Each sample in this dataset had a shallow and a heavy cough audio. The shallow coughs generally had a lower amplitude than the heavy coughs, as shown in the histogram presented in Figure 1. The figure shows that the amplitude distribution of the heavy coughs in the Coswara dataset follow the distribution of Coughvid coughs more closely. Therefore, for the experiments presented in this work, we have made use of the "heavy cough" recordings in the Coswara dataset.

The 'Healthy' coughs were taken as COVID-19 negative class, while the 'Positive mild COVID-19 symptoms', 'Positive asymptomatic COVID-19 symptoms' and 'Positive moderate COVID-19 symptoms' were taken as the COVID-19 positive class. An analysis on the Coswara dataset is presented in Figures 2 and 3.

2.2. The Coughvid dataset

The Coughvid project is executed by by EPFL (École Polytechnique Fédérale de Lausanne). Like the Coswara project, it also uses crowdsourcing for data collection by means of a web-based interface. Recorded metadata includes the subject's

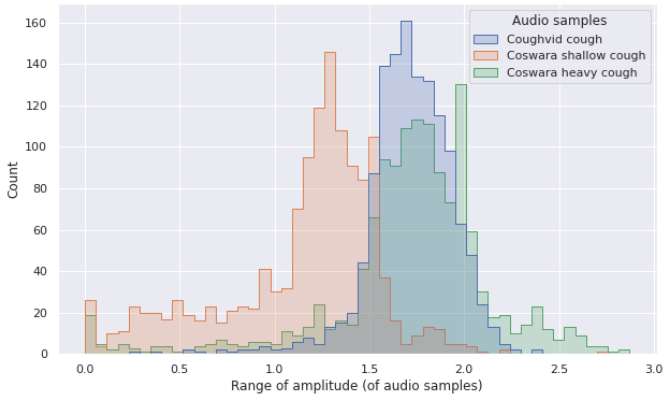

Figure 1

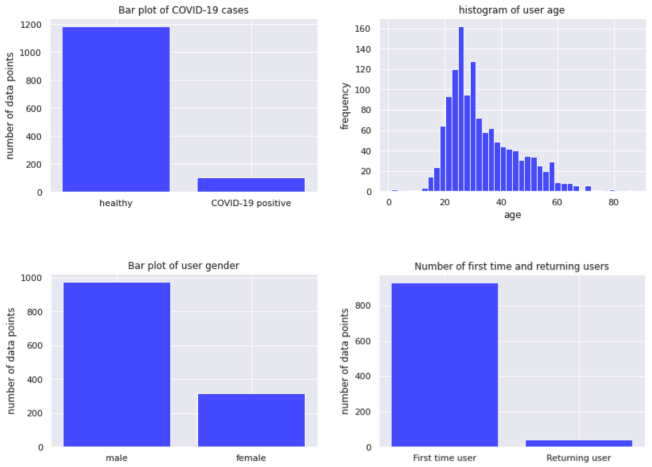

Figure 2: The Coswara dataset includes data from 107 Covid-19 positive and 1184 healthy subjects. There are 974 male and 317 female subjects most of whom are aged between 20 and 45 and reside in India.

geographical location, age, gender, other respiratory conditions, fever or muscle pain. The current COVID-19 health status must be specified according to the following three classes.

- Healthy
- COVID-19 symptomatic (not diagnosed)
- COVID-19 positive (diagnosed)

Coughvid data collection began in April of 2020 and remains ongoing. At the time of writing, there were more that 20,000 audio recordings in the dataset, all made at a sampling rate of 22.4 kHz. A subset of approximately 2,000 of the collected cough recordings have been examined and annotated by expert pulmonologists, and these were used in our experiments. All recordings labeled as COVID-19 positive by at least one of the three experts were assumed to belong to the 'COVID-19 positive' class, and the remaining were assumed to belong to the 'COVID-19 negative' class.

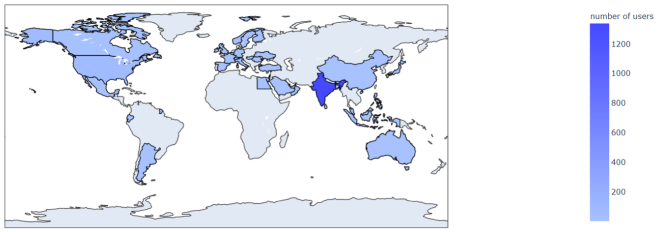

Figure 3: Coswara dataset: Geographical location of the contributing subjects shown on a world map. Most of the contributors are from India, followed by USA, Canada and, France.

An analysis on the Coughvid dataset is shown in Figure 4 and Figure 5.

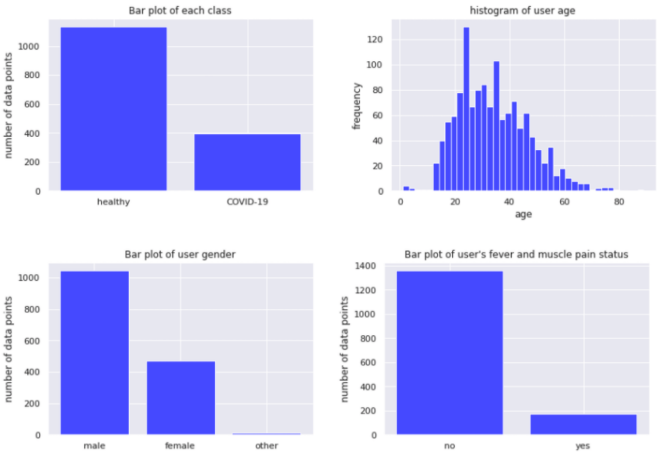

Figure 4: The Coughvid dataset has 397 covid-19 positive users and 1133 healthy users. Most of the users have an age between 20 and 50. There are 1045 male users, 473 female users, and 12 other.

2.3. *Dataset summary*

The data from both sources were combined to create a larger pool of audio samples which was later preprocessed and used for feature extraction and model training. Table 1 describes the number of samples from each source and some statistics on the duration of these samples. The final pool contains 2821 samples with total duration of 4.04 hours, average duration of 5.16 seconds and standard deviation of 2.64 seconds.

3. Data Preprocessing

3.1. *Data segmentation*

All data was in the form of PCM encoded audio at a sample rate of 22.4 kHz. Regions of silence were removed from this data using a simple energy-based threshold. Figure 6 shows an audio signal before and after this silence removal.

Table 1: Summary of the Coswara and Coughvid data. The combined data was used for preprocessing, feature extraction and classifier training and evaluation. The number of subjects was not available for the Coughvid data (indicated as 'na').

	Status	Number of subjects	Number of audio samples	Total duration	Average duration	Standard deviation of duration
Coswara	COVID-19 positive	98	107	0.16 hours	5.45 seconds	2.97 seconds
	COVID-19 negative	1027	1184	2.12 hours	6.45 seconds	3.44 seconds
	Total	1125	1291	2.28 hours	6.36 seconds	3.4 seconds
Coughvid	COVID-19 positive	na	397	0.44 hours	4.02 seconds	1.82 seconds
	COVID-19 negative	na	1133	1.32 hours	4.18 seconds	2.05 seconds
	Total	na	1530	1.76 hours	4.14 seconds	1.99 seconds
Combined	COVID-19 positive	na	504	0.6 hours	4.32 seconds	2.06 seconds
	COVID-19 negative	na	2317	3.44 hours	5.34 seconds	2.76 seconds
	Total	na	2821	4.04 hours	5.16 seconds	2.64 seconds

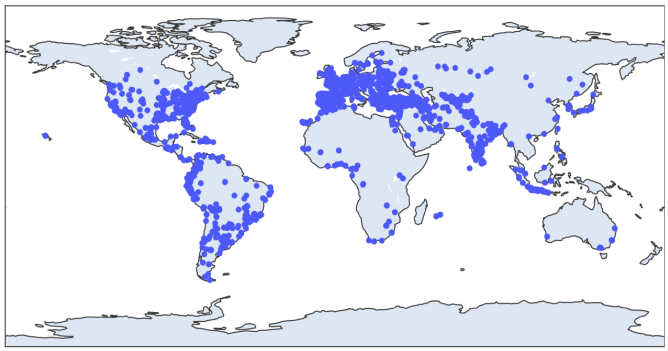

Figure 5: Coughvid dataset: Location of the users on the world map in violet dots. The data is gathered from users located in all the 6 major continents.

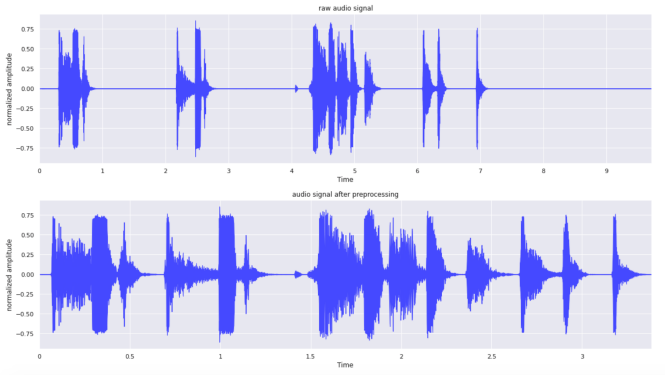

Figure 6: An example of a raw audio signal from the Coswara dataset (a) before and (b) after silence removal.

3.2. Data Balancing

After pooling the Coswara and Coughvid data, the dataset contains 2317 audio recordings of coughs by healthy subjects and 504 audio recordings of coughs by COVID-19 positive subjects. Thus, it is highly imbalanced in favour of samples from healthy subjects. To address this imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied [28], as also proposed in [24]. SMOTE synthetically oversamples the minority class by choosing a random example x_{NN} and determining its k nearest neighbours. One of these neighbours x is chosen at random and a synthetic example x_{SMOTE} is calculated by choosing a point between x and x_{NN} according to Equation 1

$$x_{SMOTE} = x + \mu \cdot (x_{NN} - x) \tag{1}$$

here μ is uniformly distributed between 0 and 1. During preliminary experimentation, we have found that $k = 5$ produced best results and note that this value also agrees with the ratio between the minority and majority classes in our dataset.

3.3. Data Padding

The preprocessed audio signals generally have different lengths because the audio recordings were of different duration. To accommodate the fixed input dimensionality demanded by some of the classifiers, data padding was applied [29]. The 90$^{\mathrm{th}}$ percentile of the durations of the audio signals in the dataset was determined and used as a threshold $D^{threshold}$. All preprocessed signals with a duration greater than $D^{threshold}$ were truncated to this duration and all signals with a duration less than $D^{threshold}$ were padded with zeros. After padding, each audio signal had a duration of $D^{threshold}$ which in our experiment was 9.5 seconds.

4. Feature Extraction

After preprocessing, features were extracted form the audio data. These features were used to train and evaluate the classifiers. Six types of feature were considered in this study:

- Mel spectrogram features.
- Mel frequency cepstral coefficients (MFCCs).
- MFCC delta features, also known as velocity features.
- MFCC delta-delta features, also known as acceleration features.
- Time-domain zero crossing rate (ZCR).
- Time-domain root mean squared energy (RMSE).

4.1. Mel Spectrogram

The mel scale is a non-linear transformation of frequency based on the human perception of pitch. Frequencies that are equidistant on the mel scale will be perceived as equidistant by a human listener. A mel spectrogram is a spectrogram indicating frequencies on the mel scale, for example Figure 7. The mel spectrogram has been successfully used in audio recognition, music detection and environmental sound classification [30, 31]. In this study, mel-scaled spectrograms were calculated for all audio samples with amplitudes in decibels.

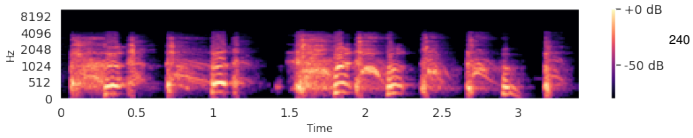

Figure 7: Mel Spectrogram representation of a COVID-19 positive cough audio signal.

4.2. Mel Frequency Cepstral Coefficients (MFCCs):

Mel frequency cepstral coefficients (MFCCs) are derived from the cepstral representation of an audio signal based on frequencies that are equally spaced on the mel scale, as computed using Equation 2.

$$f_{mel} = 2595 \times log(1 + \frac{f}{700}) \tag{2}$$

MFCC features have been widely and successfully applied in audio processing tasks like automatic speech recognition. Recently, MFCCs have also been applied to COVID-19 cough classification [24, 32, 33, 34]. An example of MFCC features extracted from the cough audio signal of a COVID-19 positive patient are shown in Figure 8.

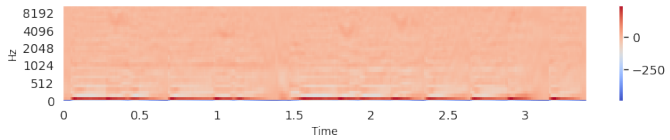

Figure 8: MFCC feature representation of a COVID-19 positive cough audio signal.

4.3. Delta features of Mel Frequency Cepstral Coefficients:

Delta features (also known as velocity features) capture the dynamic behaviour of a signal by determining the first difference of the feature vector in time. When appended to the MFCC, delta features can offer improved performance in speech recognition tasks and also in COVID-19 cough classification [24, 35]. The delta coefficient $d[t]$ at time t is defined using Equation 3.

$$d[t] = \frac{\sum_{\theta=1}^{\Theta} \theta(c[t+\theta] - c[t-\theta])}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{3}$$

Here, $c[t-\theta]$ and $c[t+\theta]$ are static coefficients from the MFCC feature of audio, and the value of Θ was set to be 2.

The delta features were calculated for the MFCC feature vectors extracted from the audio signals. An example of the delta-MFCC features for the cough audio signal of a COVID-19 positive patient are shown in Figure 9.

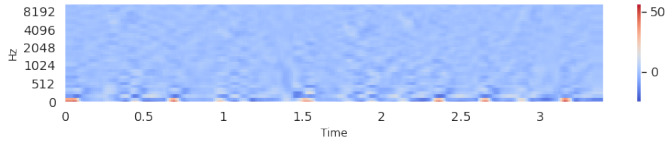

Figure 9: Delta features of the signal in Figure 8.

4.4. Delta-Delta features of Mel Frequency Cepstral Coefficients:

Delta-delta features (also known as acceleration features) are also useful in capturing the dynamic behaviour of a signal. Like velocity features, acceleration features can lead to improved performance in tasks involving audio classification, such as COVID-19 cough classification [24, 36]. Acceleration features are calculated from the second difference of a feature sequence, or equivalently the first difference of the velocity feature sequence. The delta-delta or acceleration coefficient $a[t]$ at time t is defined using Equation 4.

$$a[t] = \frac{\sum_{\theta=1}^{\Theta} \theta(d[t+\theta] - d[t-\theta])}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{4}$$

Here, $d[t-\theta]$ and $d[t+\theta]$ are delta coefficients from the MFCC feature of audio, and the value of Θ was set to be 2.

An example of the delta-delta MFCC features extracted from a cough audio signal are shown in Figure 10.

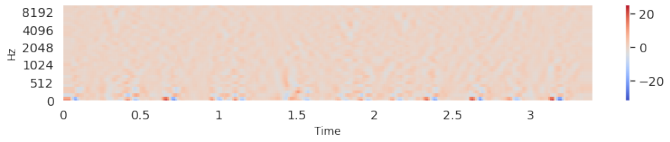

Figure 10: Delta-delta features of the signal in Figure 8.

4.5. Zero Crossing Rate (ZCR):

Zero crossing rate (ZCR) is the number of times a signal changes polarity within a fixed time interval [37]. ZCR has been widely used in audio classification [24, 38, 39]. ZCR is calculated using Equation 5.

$$ZCR = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{\mathbb{R}_{<0}}(\mathbf{s}[t]\cdot\mathbf{s}[t+1]) \tag{5}$$

where, $\mathbf{s}[t]$ is the signal at time t and $1_{\mathbb{R}_{<0}}$ is an indicator function such that it returns 1 when input is less than 0 and 0 otherwise.

This feature was calculated on the padded samples of preprocessed audio signals. The ZCR (Zero Crossing Rate) feature extracted from an audio signal is shown in Figure 11.

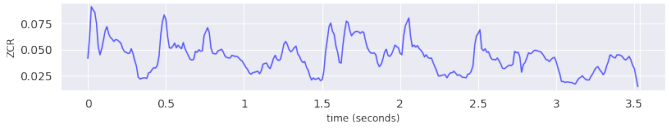

Figure 11: The zero crossing rate (ZCR) of a COVID-19 positive cough audio signal.

4.6. Root Mean Squared Energy (RMSE):

The root mean squared energy (RMSE) is a feature indicating the energy in a frame of the audio signal. It is a widely used feature in audio analysis tasks like speech recognition [40]. The RMSE value of a signal $s[n]$ is calculated using Equation 6.

$$RMSE[n] = \frac{1}{N}\sqrt{\sum_{n=0}^{N-1}\mathbf{s}[n]^2} \tag{6}$$

This feature was calculated on the padded samples of preprocessed audio signals. The Root Mean Squared Energies (RMSE) feature extracted from an audio signal is shown in Figure 12.

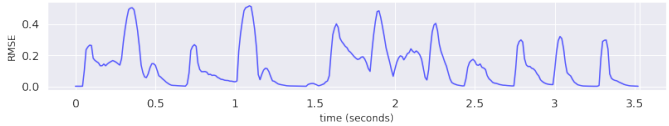

Figure 12: RMSE (Root Mean Squared Energies) of a COVID-19 positive cough audio signal.

4.7. Combining the features:

For each audio signal in the dataset, the above features were extracted and represented as feature matrices which were used for training and evaluation of the classification models. The first four features (mel spectrogram, MFCCs, MFCC deltas, MFCC delta-deltas) were represented as a matrix with time on the horizontal axis and coefficients on the vertical axis. The remaining features (ZCR, RMSE) were represented as one-dimensional matrices (vectors) with time again on the horizontal axis. These six feature matrices were stacked vertically (row wise) to from a single feature matrix with time on the horizontal axis and features on the vertical axis.

Since a number of different hyperparameters were involved in calculating this feature matrix, their dimensions were different for different classifiers. The process of selecting optimal hyperparameters is discussed in Section 6 and the resulting dimensions of the feature matrices is discussed in Section 5.

5. Model Architectures

In this study, we have considered five different deep learning models for classification. These are a multilayer perceptron (MLP), a GRU based recurrent neural network (RNN), a convolutional neural network (CNN), an OxfordNet (VGG-16) and a ResNet-50. Each of these is briefly described below.

5.1. Multilayer Perceptron (MLP):

A multilayer perceptron (MLP) is a type of artificial neural network (ANN) with multiple hidden layers between the input and output [41]. These models can learn non-linear relationships between input and output data and have a wide scope of application in classification and regression tasks, such as handwritten digit classification and house price prediction [42, 43]. They have also successfully been applied to cough audio classification [44, 45, 46]. Each node in MLP classifier calculates the output from the input data using Equation 7.

$$\mathbf{a}_j^l = \phi(b_j^l + \sum_k \mathbf{W}_{jk}^l \mathbf{a}_k^{l-1}) \tag{7}$$

In this equation $\mathbf{W}_{jk}^l$ is the trainable weight matrix of current layer l, $\mathbf{a}_k^{l-1}$ is the activation vector from the previous layer

$l-1$, b_j^l is the bias term of current layer and, ϕ is the non-linear activation function. Supervised learning is accomplished by updating the trainable parameters (weights and bias terms) by applying gradient backpropagation and gradient descent. We have applied adaptive momentum (ADAM) as the optimization algorithm and sparse categorical cross-entropy as the loss function. The architecture and important hyperparameters used while training the model are shown in Figure 13.

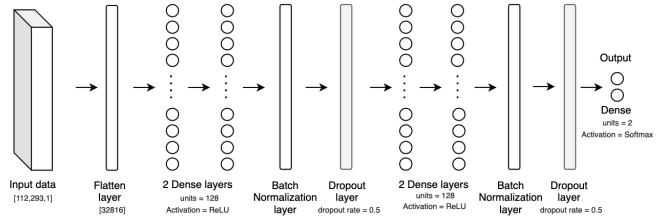

Figure 13: The model architecture of the MLP network. The network has four hidden layers with alternating pairs of batch normalization and dropout layers.

5.2. Recurrent Neural Network (RNN):

Recurrent neural networks (RNN) are a type of artificial neural network widely used in tasks involving time-series data. In contrast to MLPs, RNNs include feedback loops in their recurrent layer, allowing them to use information from previously seen inputs. This internal memory makes them effective at modelling time series or sequential data [47] [48]. Gated recurrent units (GRU) are a simplified version of the architecture used in the more complex long short term memory (LSTM) units and have been found to be easier to train. The inclusion of explicit internal update and reset gates allows previous information to be remembered over extended periods [49]. The output of a GRU as denoted by Equations 8, 9, 10 and 11.

$$\mathbf{h}[n] = (1 - \mathbf{z}[n]) \odot \mathbf{h}'[n] + \mathbf{z}[n] \odot \mathbf{h}[n-1] \tag{8}$$

$$\mathbf{h}'[n] = tanh(\mathbf{x}[n]\mathbf{W}_{xh} + (\mathbf{r}[n] \odot \mathbf{h}[t-1])\mathbf{W}_{hh} + \mathbf{b}_h) \tag{9}$$

$$\mathbf{r}[n] = \sigma(\mathbf{x}[n]\mathbf{W}_{xr} + \mathbf{h}[n-1]\mathbf{W}_{hr} + \mathbf{b}_r) \tag{10}$$

$$\mathbf{z}[n] = \sigma(\mathbf{x}[n]\mathbf{W}_{xz} + \mathbf{h}[n-1]\mathbf{W}_{hz} + \mathbf{b}_z) \tag{11}$$

In these equations,, $\mathbf{x}[n]$ is the time sequence input vector, $\mathbf{h}[n]$ is the output vector, $\mathbf{h}'[n]$ is the candidate activation vector, $\mathbf{r}[n]$ is the reset gate vector, $\mathbf{z}[n]$ is the update gate vector, $\mathbf{W}_{xh}$, $\mathbf{W}_{hh}$, $\mathbf{W}_{xr}$, $\mathbf{W}_{hr}$,$\mathbf{W}_{xz}$,$\mathbf{W}_{hz}$ are the trainable weight matrices and, $\mathbf{b}_h$, $\mathbf{b}_r$, $\mathbf{b}_z$ are the trainable bias vectors. For training the model, adaptive momentum (ADAM) was used as the optimization function and sparse categorical cross-entropy was used as the loss function. The architecture and important hyper-parameters used while training the model are defined in Figure 14.

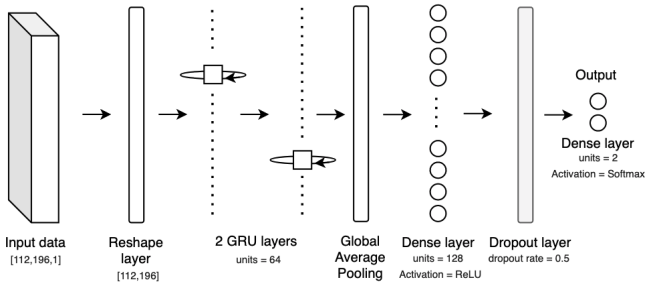

Figure 14: The model architecture of the GRU network. The network has three hidden layers of which two are GRU layers and one is a dense layer.

5.3. Convolutional Neural Network (CNN):

A convolutional neural network (CNN) is a type of artificial neural network that is particularly effective in image classification tasks, such as face recognition and instance segmentation [50, 51, 52]. CNNs have recently been applied successfully to speech signals since these signals can be represented as two-dimensional images in the form of a spectrogram. In this way they have also been shown to be an effective tool for cough detection [53]. A CNN is able to assign importance to various regions of an image by means of trainable kernels. Equation 12 describes the calculation of the output of a CNN layer from an input matrix.

$$\begin{aligned}\mathbf{X}^l[m,n] &= \phi(b + \mathbf{W}^l[m,n] * \mathbf{X}^{l-1}[m,n]) \\ &= \phi(b + \sum_j \sum_k \mathbf{W}^l[j,k]\mathbf{X}^{l-1}[m-j, n-k])\end{aligned} \tag{12}$$

Here, $\mathbf{X}^{l-1}$ is the activation matrix from previous layer, $\mathbf{W}^l$ is the trainable weight matrix of current layer, b is the bias term, ϕ is the non linear activation function and, $*$ is the convolution operation. For training the model, adaptive momentum (ADAM) was used as the optimization function and sparse categorical cross-entropy was used as the loss function. The architecture and important hyperparameters used while training the model are defined in Figure 15.

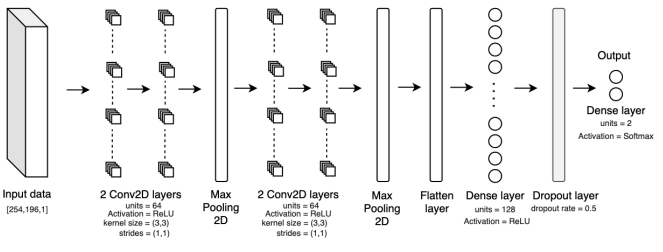

Figure 15: The model architecture of the CNN network The five hidden layers consist of four convolutional layers and one dense layer. The features extracted from the cough audio signals are stacked vertically to create a vector input.

5.4. OxfordNet (VGG-16):

VGG-16 (also called OxfordNet) is a convolutional neural network architecture named after the Visual Geometry Group at Oxford University [54]. The network is 16 layers deep and is widely used in the computer vision domain for transfer learning and efficiently handling complex tasks like image classification with many classes [55]. The architecture and important hyperparameters used while training the model are defined in Figure 16.

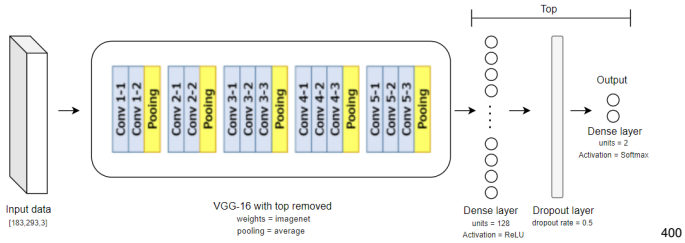

Figure 16: The model architecture of the OxfordNet (VGG-16) neural network. The last three fully connected layers of VGG-16 were replaced by a single dense layer followed by a dropout layer and a a final two-dimensional softmax layer.

In this study, we have used VGG-16 for transfer learning and the model was initialized with weights pre-trained on the ImageNet data. The default VGG-16 architecture has a 1000-dimnesional output layer for classifying the ImageNet data hence, the final three layers, denoted as top in Figure 16, were replaced with a dense layer, a dropout layer, and a final two-dimensional softmax layer for our binary classification task.

5.5. Residual Networks (ResNet-50):

Residual network (ResNet-50) is a convolutional neural network architecture that contains residual connections, also known as skip connections, between its layers [56]. The network is 50 layers deep and has been found to outperform many other deep neural network architectures. It is widely applied and has been successful in detecting COVID-19 from CT scan images of lungs and also from cough audio [9, 10, 11, 24]. We have used ResNet-50 for transfer learning, since it is pre-trained on the ImageNet data. Since the output layer of the default pre-trained ResNet-50 has 1000 dense units, we have replaced it with two pairs of dense and dropout layers and a final two-dimensional softmax layer for binary classification. The newly-added layers are trained by adaptive momentum (ADAM) with sparse categorical cross-entropy as the loss function. The architecture and important hyperparameters used while training the model are presented in Figure 17.

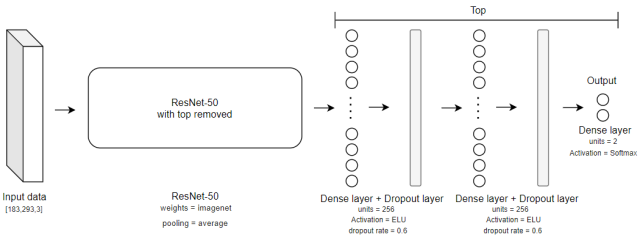

Figure 17: The model architecture of the ResNet-50 neural network. The final dense layers of ResNet-50 have been replaced by two sets of dense layer followed by dropout layer, and finally a two-dimensional softmax later.

6. Experimental Method

6.1. Evaluation Metrics

To compare different model architectures, we assess their performance on the basis of the evaluation metrics accuracy, sensitivity, specificity and area under the receiver operating characteristic curve (ROC AUC).

6.1.1. Accuracy:

Accuracy is defined as a ratio of the number of correctly classified samples to the total number of classified samples and is usually denoted as a percentage. Accuracy is stated in Equation 13 in terms of the confusion matrix of a binary classifier illustrated in Figure 18.

$$\begin{aligned} Accuracy &= \frac{\text{Number of correctly classified points}}{\text{Total number of classified points}} \\ &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \end{aligned} \tag{13}$$

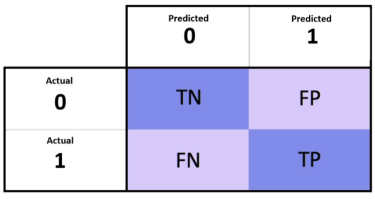

Figure 18: Confusion Matrix represented in terms of actual class labels and predicted class labels.

In this figure, TP corresponds to the number of true positives, the positive samples that were correctly classified as positive, TN corresponds to the number of true negatives, the negative samples that were correctly classified as negative, FP corresponds to the number of false positives, the negative samples that were incorrectly classified as positive, and FN corresponds to the number of false negatives, the positive samples that were incorrectly classified as negative.

6.1.2. Equal Error Rate (EER):

EER is a rate at which both false positive rate (FPR) and false negative rate (FNR) are equal. The FPR is defined as the ratio of negative samples that were incorrectly classified to the total number of negative samples. In contrast, the FNR is defined as the ratio of positive samples that were incorrectly classified to the total number of positive samples. To calculate the EER, the FPR and FNR are calculated at different decision thresholds and the threshold at which the difference between FPR and FNR is zero is identified. It can be formulated as a value of threshold th that minimizes the absolute difference between FPR and FNR, as shown in Equation 14.

$$\arg\min_{th\in(0,1)} \mid FPR_{th} - FNR_{th} \mid \tag{14}$$

6.1.3. Area under the ROC curve (AUC):

The receiver operating characteristic (ROC) graphically indicates the tradeoff between sensitivity (TPR) and specificity (1 - FPR) for a binary classifier as the decision threshold varies [57]. The area under this ROC curve (AUC) is a useful figure of merit that reflects the performance of the classifier over the entire range of decision thresholds.

In our binary classification problem, cough audio by COVID-19 positive subjects is labeled as the positive class and cough audio by healthy or COVID-19 negative subjects is labeled as the negative class. The primary objective of the classifier is to reduce both false negatives (subjects who have COVID-19 but are classified as healthy) and false positives (healthy or COVID-19 negative subjects who are classified as COVID-19 positive). These errors are balanced at the equal error rate (EER).

$$\begin{aligned} y_{pred} &= 0 \quad if\ P < P_{EER} \\ &= 1 \quad otherwise \end{aligned} \tag{15}$$

Equation 15 indicates the classification decision, where P is the probability of the audio input being from a COVID-19 positive subject (as calculated by the classifier) and P_{ERR} is the threshold that achieves the equal error rate.

6.1.4. Sensitivity and specificity:

Sensitivity and specificity are measures of the performance of a binary classifier. Sensitivity, also known as the true positive rate (TPR), is a measure of how well a classifier can identify the positive class. It is given by the ratio of correctly classified positive samples to the total number of positive samples, as shown in Equation 16.

$$\begin{aligned} Sensitivity\ (TPR) &= \frac{\text{Correctly classified positive samples}}{\text{Total number of positive samples}} \\ &= \frac{TP}{TP + FN} \end{aligned} \tag{16}$$

Specificity, also known as true negative rate (TNR), is a measure of how well a classifier can identify the negative class. It is given by the ratio of correctly classified negative samples to the total number of negative samples, as shown in Equation 17.

$$\begin{aligned} Specificity\ (TPR) &= \frac{\text{Correctly classified negative samples}}{\text{Total number of negative samples}} \\ &= \frac{TN}{TN + FP} \end{aligned} \tag{17}$$

In our evaluation, we have calculated sensitivity and specificity at the equal error rate (EER).

6.2. Cross Validation

Cross-validation was used for model training, hyperparameter optimisation and classifier evaluation. We used a two step nested process for fine-tuning the classifiers, and evaluating their performance on unseen data. First, the dataset is split into multiple non-overlapping optimization and test sets, as illustrated in Figure 19. The test sets were reserved as unseen data for final independent evaluation of the performance of optimized classifiers. This split is used in the outer loop of cross validation, as shown in Figure 21. The optimization set is further divided in an inner loop into multiple folds of non-overlapping training and validation sets, as shown in Figure 20. These are used for training the classifier and optimising the hyperparameters respectively.

The two steps, namely, *hyperparameter optimization* and *performance on unseen data*, are discussed briefly below.

6.2.1. Hyperparameter optimization

K-fold cross validation is a statistical method that employs a dataset resampling procedure to train, optimise and evaluate machine learning models [58]. The procedure has a parameter k that denotes the number of 'folds' (partitions) into which the dataset is divided. This division ensures that the samples from a particular patient all occur in the same fold and hence that the

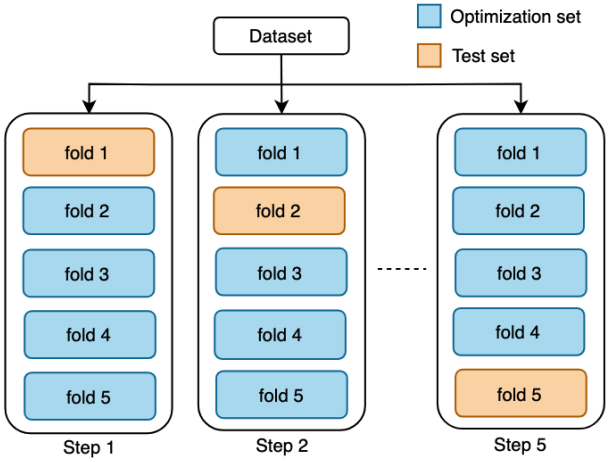

Figure 19: Five step data splitting for model evaluation on unseen data. At each step, training and validation set is used for hyperparameter optimization and test set is used for evaluating the performance on unseen data.

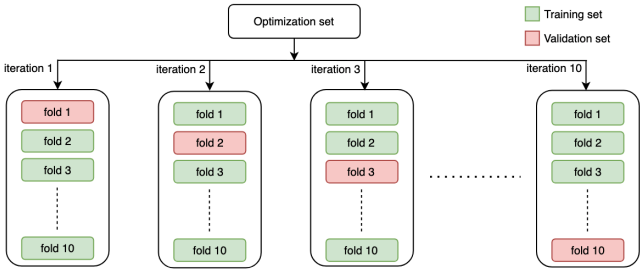

Figure 20: k-Fold cross validation with k=10. In each iteration 9 data folds are used for model training and the remaining fold for validation.

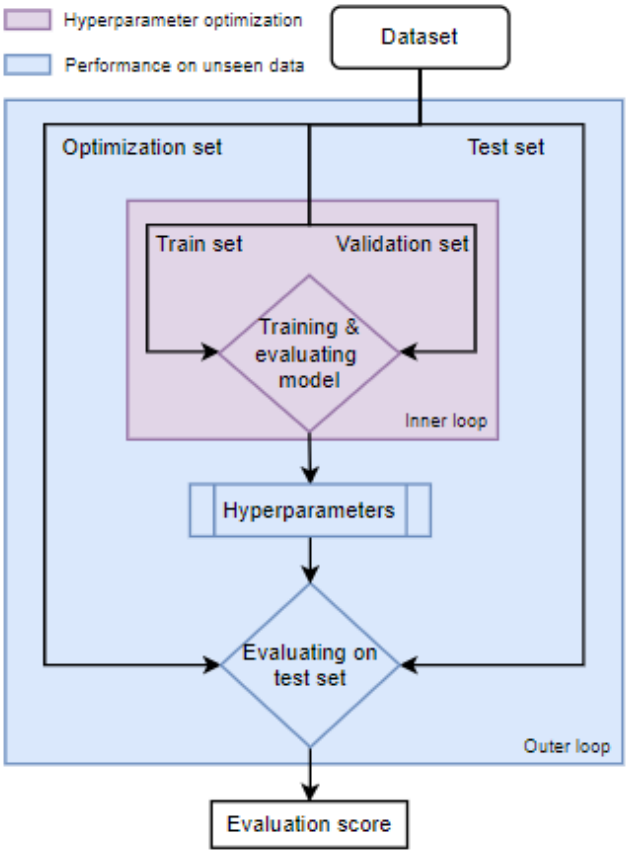

Figure 21: Nested cross validation for hyperparameter optimization and classifier performance evaluation.

folds are independent. Once the dataset has been divided into k folds, the algorithm iterates over these folds. At each iteration one of the k folds is chosen as a validation set and the remaining $k-1$ as training set. This process is illustrated as inner loop in Figure 21.

Figure 20 illustrates this for $k = 10$, as also used in our experiments.

A number of hyperparameters are implicit in feature extraction, classification model architecture and training. These hyperparameters, as well as the values considered in their optimisation, are listed in Tables 2 and 3. All these hyperparameters were optimised by means of nested cross validation. The optimal values found in this way are listed in Table 4. The optimised model architecture hyperparameters have also been specified in Figures 13, 14, 15, 16, and 17.

6.2.2. *Classifier performance*

Classifier performance is evaluated on unseen test data in the outer loop of nested cross validation as shown in Figure 21. The final classification performance is determined by calculating the average over the outer loop test sets.

7. Results

The five neural network architectures described in Section 5 were trained and optimised as described in Section 6. In the following section, we present the results of the optimal classifier of each architecture using the metrics.

The performance of the classifiers is presented in Table 4 and illustrated in Figure 22. The accuracy, AUC, specificity and, sensitivity values are the mean values over the test folds, as described in Section 6.2.2. Receiver Operating Characteristics are presented in Figure 23.

The results show that the ResNet-50 outperformed all other models, achieving AUC, sensitivity, specificity and accuracy values of 0.96, 93%, 90% and, 92.78% respectively. The ROC plots confirm that the deep neural networks pre-trained on imagenet data (VGG-16 and ResNet-50) achieve better performance than the other neural networks (CNN, RNN and MLP) over a wide range of decision thresholds.

Table 2: The best hyperparameters tested for designing and fine-tuning the classifier models

	Classifier type	Hyperparameter	Values tested
Classifier architecture hyperparameters	MLP	Number of hidden layers	$1, 2, 3, 4, 5, 6$
		Number of units in each layer	2^k, where $k = 4, 5, 6, 7, 8, 9$
		Dropout rate	$0.2, 0.3, 0.4, 0.5, 0.6, 0.7$
	RNN	Number of recurrent layers	$1, 2, 3, 4, 5, 6$
		Number of GRU units in each layer	2^k, where $k = 4, 5, 6, 7$
		Number of dense layers	$1, 2, 3$
		Number of units in dense layers	2^k, where $k = 4, 5, 6, 7, 8, 9$
		Dropout rate	$0.2, 0.3, 0.4, 0.5, 0.6, 0.7$
	CNN	Number of convolutional layers	$1, 2, 3, 4, 5, 6$
		Kernel size in each convolutional layer	$3, 4, 5, 6, 7$
		Number of dense layers	$1, 2, 3$
		Number of units in dense layers	2^k, where $k = 4, 5, 6, 7, 8, 9$
		Dropout rate	$0.2, 0.3, 0.4, 0.5, 0.6, 0.7$
	VGG-16, ResNet-50	Number of dense layers at the end	$1, 2, 3, 4$
		Number of units in dense layers	2^k, where $k = 4, 5, 6, 7, 8, 9$
		Dropout rate	$0.2, 0.3, 0.4, 0.5, 0.6, 0.7$
Classifier training hyperparameters	All classifiers	Batch size	2^k, where $k = 6, 7, 8$
		Epochs	$20 \times k$, where $k = 1, 2, 3, 4, 5$
		Optimization function	SGD, Adagrad, RMSProp, ADAM
		Learning rate	10^{-k}, where $k = 1, 2, 3, 4$

Table 3: The hyperparameters tested during data-preparation and feature extraction using k-fold cross validation. Note that k=0 was used while testing, which means independent features and combinations where one or more features were absent were also tested.

Processes	Hyperparameters	Description	Range
Data segmentation	Frame length	The number of samples per analysis frame	2^k, where $k = 7, 8, 9, 10, 11, 12$
	Hop length	The number of samples between analysis frame	$256 \times k$, where $k = 1, 2, 3, 4$
Root Mean Square Energy (RMSE), Zero Crossing Rate (ZCR)	Frame length	The number of samples per analysis frame	$k = 0, 2^i$, where $i = 7, 8, 9, 10, 11, 12$
	Hop length	The number of samples between analysis frame	$256 \times k$, where $k = 1, 2, 3, 4$
Mel Spectrogram	n_mels	number of Mel bands to generate	$32 \times k$, where $k = 0, 1, 2, 3, 4, 5$
MFCC, MFCC delta, MFCC delta-delta	n_mfcc	Number of MFCCs to return	$13 \times k$, where $k = 0, 1, 2, 3, 4, 5, 6$

Table 4: Model performances observed on unseen data using optimized hyperparameters. ResNet-50 yields the best results with the highest AUC, specificity, sensitivity and, accuracy values.

Model	Hyperparameters							Results			
	Feature extraction hyperparameters				Model training hyperparameters						
	n_mels	n_mfcc	frame length	hop length	batch-size	epochs	learning rate	AUC	Sensitivity	Specificity	Accuracy
MLP	32	26	1024	512	64	40	0.001	0.73	70%	68%	69.42%
RNN	32	26	2048	768	64	40	0.001	0.84	81%	75%	78.84%
CNN	96	52	2048	768	64	80	0.0001	0.89	86%	80%	84.24%
VGG-16	64	39	1024	512	64	100	0.0001	0.92	88%	82%	86.24%
ResNet-50	64	39	1024	512	64	100	0.0001	0.96	93%	90%	92.78%

8. Conclusion

In this study, we have presented COVID-19 classification using coughing audio data and various machine learning architectures. For experimentation we have used two crowdsourced and publicly available datasets, COUGHVID by IISC Bangalore and COSWARA by EPFL respectively. Together, these datasets contain 2821 audio samples. We trained five neural network architectures for classification: a deep multilayer perceptron (MLP), a GRU based Recurrent Neural Network (RNN), a convolutional neural network (CNN), a VGG-16 and a ResNet-50 on this data using mel-spectrogram, MFCC, zero crossing rate (ZCR) and root mean square energy (RMSE) as features. The best performing architecture is based on ResNet-50 and achieves an AUC of 0.96, a sensitivity of 93%, a specificity of 90% and an accuracy of 92.78%. In terms of sensitivity and specificity, this performance is on par with that achieved by an RT-PCR test COVID-19 test. Therefore, COVID-19 screening by machine learning based cough audio classification appears to be a promising method of screening. In comparison with existing biological tests, it has the advantage of being cost-effective and easily accessible by means of internet-enabled mobile devices such as smartphones. It also can eliminate or strongly reduce the risk of COVID-19 transmission during testing.

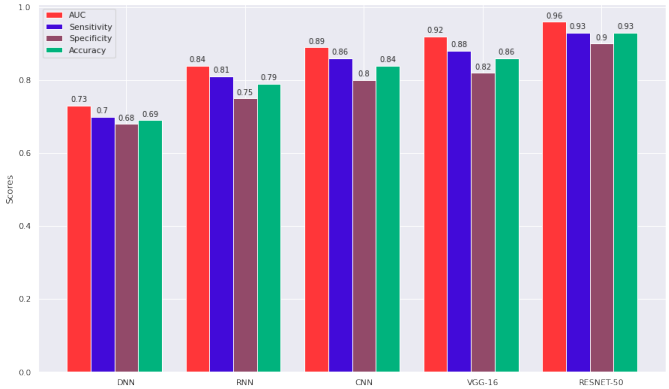

Figure 22: The accuracy, AUC, specificity and sensitivity of different optimised classifier. The ResNet-50 outperforms all the other architectures, achieving specificity of 0.9, sensitivity of 0.93, AUC of 0.96 and an accuracy of 92.78%.

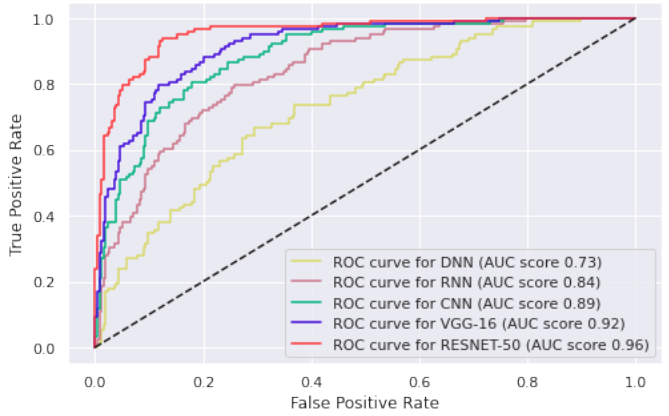

Figure 23: Receiver Operating Characteristic (ROC) curves of different classification models. Resnet-50 outperformed the other classifiers by obtaining a AUC score of 0.96

9. Future work

Although the results are promising, the setup requires more data in order to more rigorously test the robustness of the classifiers. To address this data from other COVID-19 coughing audio datasets is being considered. Furhermore, we are developing a cloud-based API that could be accessed by internet-enabled devices and allow the online classification of cough audio collected by mobile devices such as smartphones.

References

[1] H. Harapan, N. Itoh, A. Yufika, W. Winardi, S. Keam, H. Te, D. Megawati, Z. Hayati, A. L. Wagner, M. Mudatsir, Coronavirus disease 2019 (covid-19): A literature review, Journal of Infection and Public Health 13 (5) (2020) 667–673. doi:https://doi.org/10.1016/j.jiph.2020.03.019.
URL https://www.sciencedirect.com/science/article/pii/S1876034120304329

[2] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, Y. Zhao, Y. Li, X. Wang, Z. Peng, Clinical Characteristics of 138 Hospitalized Patients With 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China, JAMA 323 (11) (2020) 1061–1069.

[3] A. Carfì, R. Bernabei, F. Landi, Persistent Symptoms in Patients After Acute COVID-19, JAMA 324 (6) (2020) 603–605.

[4] S. Deepak, K. Kottapalli, R. Rakwal, G. Oros, K. Rangappa, H. Iwahashi, Y. Masuo, G. Agrawal, Real-Time PCR: Revolutionizing Detection and Expression Analysis of Genes, Curr Genomics 8 (4) (2007) 234–251.

[5] O. W. in Data, Coronavirus (covid-19) testing (2021).
URL https://ourworldindata.org/coronavirus-testing

[6] B. Giri, S. Pandey, R. Shrestha, K. Pokharel, F. S. Ligler, B. B. Neupane, Review of analytical performance of covid-19 detection methods, Analytical and Bioanalytical Chemistry 413 (1) (2021) 35–48. doi:10.1007/s00216-020-02889-x.
URL https://doi.org/10.1007/s00216-020-02889-x

[7] I. Arevalo-Rodriguez, D. Buitrago-Garcia, D. Simancas-Racines, P. Zambrano-Achig, R. Del Campo, A. Ciapponi, O. Sued, L. Martinez-García, A. W. Rutjes, N. Low, P. M. Bossuyt, J. A. Perez-Molina, J. Zamora, False-negative results of initial rt-pcr assays for covid-19: A systematic review, PLOS ONE 15 (12) (2020) 1–19. doi:10.1371/journal.pone.0242958.
URL https://doi.org/10.1371/journal.pone.0242958

[8] J. Hopkins, COVID-19 Dashboard by the center for systems science and engineering (csse) at johns hopkins university (2021).
URL https://coronavirus.jhu.edu/map.html

[9] T. C. Kwee, R. M. Kwee, Chest ct in covid-19: What the radiologist needs to know, RadioGraphics 40 (7) (2020) 1848–1865, pMID: 33095680. arXiv:https://doi.org/10.1148/rg.2020200159, doi:10.1148/rg.2020200159.
URL https://doi.org/10.1148/rg.2020200159

[10] I. Ozsahin, B. Sekeroglu, M. S. Musa, M. T. Mustapha, D. Uzun Ozsahin, Review on diagnosis of covid-19 from chest ct images using artificial intelligence, Computational and Mathematical Methods in Medicine 2020 (2020) 9756518. doi:10.1155/2020/9756518.
URL https://doi.org/10.1155/2020/9756518

[11] S. Sharma, Drawing insights from covid-19-infected patients using ct scan images and machine learning techniques: a study on 200 patients, Environmental science and pollution research international 27 (29) (2020) 37155–37163, 32700269[pmid]. doi:10.1007/s11356-020-10133-3.
URL https://pubmed.ncbi.nlm.nih.gov/32700269

[12] B. Sekeroglu, I. Ozsahin, Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks, SLAS Technol 25 (6) (2020) 553–565.

[13] T. B. Chandra, K. Verma, B. K. Singh, D. Jain, S. S. Netam, Coronavirus disease (COVID-19) detection in Chest X-Ray images using majority voting based classifier ensemble, Expert Syst Appl 165 (2021) 113909.

[14] R. Jain, M. Gupta, S. Taneja, D. J. Hemanth, Deep learning based detection and analysis of covid-19 on chest x-ray images, Applied Intelligence 51 (3) (2021) 1690–1700. doi:10.1007/s10489-020-01902-1.
URL https://doi.org/10.1007/s10489-020-01902-1

[15] A. Abbas, M. M. Abdelsamea, M. M. Gaber, Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network, Applied Intelligence 51 (2) (2021) 854–864. doi:10.1007/s10489-020-01829-7.
URL https://doi.org/10.1007/s10489-020-01829-7

[16] R. X. A. Pramono, S. A. Imtiaz, E. Rodriguez-Villegas, A cough-based algorithm for automatic diagnosis of pertussis, PLOS ONE 11 (9) (2016) 1–20. doi:10.1371/journal.pone.0162128.
URL https://doi.org/10.1371/journal.pone.0162128

[17] A. Windmon, M. Minakshi, P. Bharti, S. Chellappan, M. Johansson, B. A. Jenkins, P. R. Athilingam, TussisWatch: A Smart-Phone System to Identify Cough Episodes as Early Symptoms of Chronic Obstructive Pulmonary Disease and Congestive Heart Failure, IEEE J Biomed Health Inform 23 (4) (2019) 1566–1573.

[18] R. V. Sharan, U. R. Abeyratne, V. R. Swarnkar, P. Porter, Automatic croup diagnosis using cough sound recognition, IEEE Transactions on Biomedical Engineering 66 (2) (2019) 485–495. doi:10.1109/TBME.2018.2849502.

[19] G. Rudraraju, S. Palreddy, B. Mamidgi, N. R. Sripada, Y. P. Sai, N. K. Vodnala, S. P. Haranath, Cough sound analysis and objective correlation with spirometry and clinical diagnosis, Informatics in Medicine Unlocked

19 (2020) 100319. doi:https://doi.org/10.1016/j.imu.2020.100319.
URL https://www.sciencedirect.com/science/article/pii/S2352914819304071

[20] G. Deshpande, B. Schuller, An Overview on Audio, Signal, Speech, & Language Processing for COVID-19, arXiv e-prints (2020) arXiv:2005.08579arXiv:2005.08579.

[21] T. D. Pham, A comprehensive study on classification of covid-19 on computed tomography with pretrained convolutional neural networks, Scientific Reports 10 (1) (2020) 16942. doi:10.1038/s41598-020-74164-z.
URL https://doi.org/10.1038/s41598-020-74164-z

[22] B. W. Schuller, D. M. Schuller, K. Qian, J. Liu, H. Zheng, X. Li, COVID-19 and Computer Audition: An Overview on What Speech & Sound Analysis Could Contribute in the SARS-CoV-2 Corona Crisis, arXiv e-prints (2020) arXiv:2003.11117arXiv:2003.11117.

[23] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, C. Mascolo, Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery& Data Mining, KDD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 3474–3484. doi:10.1145/3394486.3412865.
URL https://doi.org/10.1145/3394486.3412865

[24] M. Pahar, M. Klopper, R. Warren, T. Niesler, COVID-19 Cough Classification using Machine Learning and Global Smartphone Recordings, arXiv e-prints (2020) arXiv:2012.01926arXiv:2012.01926.

[25] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, M. S. Riaz, K. Ali, C. N. John, M. I. Hussain, M. Nabeel, Ai4covid-19: Ai enabled preliminary diagnosis for covid-19 from cough samples via an app, Informatics in Medicine Unlocked 20 (2020) 100378. doi:https://doi.org/10.1016/j.imu.2020.100378.
URL https://www.sciencedirect.com/science/article/pii/S2352914820303026

[26] N. Sharma, P. Krishnan, R. Kumar, S. Ramoji, S. R. Chetupalli, N. R., P. K. Ghosh, S. Ganapathy, Coswara — a database of breathing, cough, and voice sounds for covid-19 diagnosis, Interspeech 2020 (Oct 2020). doi:10.21437/interspeech.2020-2768.
URL http://dx.doi.org/10.21437/Interspeech.2020-2768

[27] L. Orlandic, T. Teijeiro, D. Atienza, The coughvid crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms (2020). arXiv:2009.11644.

[28] R. Blagus, L. Lusa, Smote for high-dimensional class-imbalanced data, BMC Bioinformatics 14 (1) (2013) 106. doi:10.1186/1471-2105-14-106.
URL https://doi.org/10.1186/1471-2105-14-106

[29] M. Dwarampudi, N. V. Subba Reddy, Effects of padding on LSTMs and CNNs, arXiv e-prints (2019) arXiv:1903.07288arXiv:1903.07288.

[30] B.-Y. Jang, W.-H. Heo, J.-H. Kim, O.-W. Kwon, Music detection from broadcast contents using convolutional neural networks with a mel-scale kernel, EURASIP Journal on Audio, Speech, and Music Processing 2019 (1) (2019) 11. doi:10.1186/s13636-019-0155-y.
URL https://doi.org/10.1186/s13636-019-0155-y

[31] Y. Su, K. Zhang, J. Wang, K. Madani, Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion, Sensors (Basel) 19 (7) (Apr 2019).

[32] Wei Han, Cheong-Fat Chan, Chiu-Sing Choy, Kong-Pang Pun, An efficient mfcc extraction method in speech recognition, in: 2006 IEEE International Symposium on Circuits and Systems, 2006, pp. 4 pp.–. doi:10.1109/ISCAS.2006.1692543.

[33] T. Ratanpara, N. Patel, Singer identification using mfcc and lpc coefficients from indian video songs, in: S. C. Satapathy, A. Govardhan, K. S. Raju, J. K. Mandal (Eds.), Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) Volume 1, Springer International Publishing, Cham, 2015, pp. 275–282.

[34] Z. Wanli, L. Guoxin, The research of feature extraction based on mfcc for speaker recognition, in: Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, 2013, pp. 1074–1077. doi:10.1109/ICCSNT.2013.6967289.

[35] K. Kumar, C. Kim, R. M. Stern, Delta-spectral cepstral coefficients for robust speech recognition, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 4784–4787. doi:10.1109/ICASSP.2011.5947425.

[36] M. A. Hossan, S. Memon, M. A. Gregory, A novel approach for mfcc feature extraction, in: 2010 4th International Conference on Signal Processing and Communication Systems, 2010, pp. 1–5. doi:10.1109/ICSPCS.2010.5709752.

[37] R. Bachu, S. Kopparthi, B. Adapa, B. Barkana, Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy, in: K. Elleithy (Ed.), Advanced Techniques in Computing Sciences and Software Engineering, Springer Netherlands, Dordrecht, 2010, pp. 279–282.

[38] Yiu-Kei Lau, Chok-Ki Chan, Speech recognition based on zero crossing rate and energy, IEEE Transactions on Acoustics, Speech, and Signal Processing 33 (1) (1985) 320–323. doi:10.1109/TASSP.1985.1164503.

[39] S. K. Park, R. M. Kil, Y.-G. Jung, M.-S. Han, Zero-crossing-based feature extraction for voice command systems using neck-microphones, in: D. Liu, S. Fei, Z.-G. Hou, H. Zhang, C. Sun (Eds.), Advances in Neural Networks – ISNN 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 1318–1326.

[40] G. Saha, S. Chakroborty, S. Senapati, A new silence removal and endpoint detection algorithm for speech and speaker recognition applications, 2006.

[41] H. Taud, J. Mas, Multilayer Perceptron (MLP), Springer International Publishing, Cham, 2018, pp. 451–455. doi:10.1007/978-3-319-60801-3_27.
URL https://doi.org/10.1007/978-3-319-60801-3_27

[42] S. Knerr, L. Personnaz, G. Dreyfus, Handwritten digit recognition by neural networks with single-layer training, IEEE Transactions on Neural Networks 3 (6) (1992) 962–968. doi:10.1109/72.165597.

[43] W. T. Lim, L. Wang, Y. Wang, Q. Chang, Housing price prediction using neural networks, in: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016, pp. 518–522. doi:10.1109/FSKD.2016.7603227.

[44] B. H. Tracey, G. Comina, S. Larson, M. Bravard, J. W. López, R. H. Gilman, Cough detection algorithm for monitoring patient recovery from pulmonary tuberculosis, Annu Int Conf IEEE Eng Med Biol Soc 2011 (2011) 6017–6020.

[45] L. Sarangi, M. N. Mohanty, S. Pattanayak, Design of mlp based model for analysis of patient suffering from influenza, Procedia Computer Science 92 (2016) 396–403, 2nd International Conference on Intelligent Computing, Communication& Convergence, ICCC 2016, 24-25 January 2016, Bhubaneswar, Odisha, India. doi:https://doi.org/10.1016/j.procs.2016.07.396.
URL https://www.sciencedirect.com/science/article/pii/S1877050916316520

[46] J. Liu, M. You, Z. Wang, G. Li, X. Xu, Z. Qiu, Cough detection using deep neural networks, in: 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2014, pp. 560–563. doi:10.1109/BIBM.2014.6999220.

[47] Z. C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning (2015). arXiv:1506.00019.

[48] R. Rana, Gated recurrent unit (gru) for emotion classification from noisy speech (2016). arXiv:1612.07778.

[49] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling (2014). arXiv:1412.3555.

[50] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6. doi:10.1109/ICEngTechnol.2017.8308186.

[51] M. Coskun, A. Ucar, O. Yildirim, Y. Demir, Face recognition based on convolutional neural network, in: 2017 International Conference on Modern Electrical and Energy Systems (MEES), 2017, pp. 376–379. doi:10.1109/MEES.2017.8248937.

[52] F. Sultana, A. Sufian, P. Dutta, Evolution of image segmentation using deep convolutional neural network: A survey, Knowledge-Based Systems 201-202 (2020) 106062. doi:10.1016/j.knosys.2020.106062.
URL http://dx.doi.org/10.1016/j.knosys.2020.106062

[53] J. Amoh, K. Odame, Deepcough: A deep convolutional neural network in a wearable cough detection system, in: 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), 2015, pp. 1–4. doi:

10.1109/BioCAS.2015.7348395.

[54] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2015). arXiv:1409.1556.

[55] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, Li Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[56] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition (2015). arXiv:1512.03385.

[57] J. N. Mandrekar, Receiver operating characteristic curve in diagnostic test assessment, Journal of Thoracic Oncology 5 (9) (2010) 1315–1316. doi:https://doi.org/10.1097/JTO.0b013e3181ec173d.
URL https://www.sciencedirect.com/science/article/pii/S1556086415306043

[58] S. Raschka, Model evaluation, model selection, and algorithm selection in machine learning (2020). arXiv:1811.12808.