Computers in Biology and Medicine 141 (2022) 105153

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

COVID-19 detection in cough, breath and speech using deep transfer learning and bottleneck features

Madhurananda Pahar [a,*], Marisa Klopper [b], Robin Warren [b], Thomas Niesler [a,**]

[a] Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa
[b] SAMRC Centre for Tuberculosis Research, DSI-NRF Centre of Excellence for Biomedical Tuberculosis Research, Division of Molecular Biology and Human Genetics, Faculty of Medicine and Health Sciences, Stellenbosch University, South Africa

ARTICLE INFO

Keywords:
COVID-19
Cough
Breath
Speech
Transfer learning
Bottleneck features

ABSTRACT

We present an experimental investigation into the effectiveness of transfer learning and bottleneck feature extraction in detecting COVID-19 from audio recordings of cough, breath and speech. This type of screening is non-contact, does not require specialist medical expertise or laboratory facilities and can be deployed on inexpensive consumer hardware such as a smartphone. We use datasets that contain cough, sneeze, speech and other noises, but do not contain COVID-19 labels, to pre-train three deep neural networks: a CNN, an LSTM and a Resnet50. These pre-trained networks are subsequently either fine-tuned using smaller datasets of coughing with COVID-19 labels in the process of transfer learning, or are used as bottleneck feature extractors. Results show that a Resnet50 classifier trained by this transfer learning process delivers optimal or near-optimal performance across all datasets achieving areas under the receiver operating characteristic (ROC AUC) of 0.98, 0.94 and 0.92 respectively for all three sound classes: coughs, breaths and speech.

This indicates that coughs carry the strongest COVID-19 signature, followed by breath and speech. Our results also show that applying transfer learning and extracting bottleneck features using the larger datasets without COVID-19 labels led not only to improved performance, but also to a marked reduction in the standard deviation of the classifier AUCs measured over the outer folds during nested cross-validation, indicating better generalisation.

We conclude that deep transfer learning and bottleneck feature extraction can improve COVID-19 cough, breath and speech audio classification, yielding automatic COVID-19 detection with a better and more consistent overall performance.

1. Introduction

COVID-19 (**CO**rona **VI**rus **D**isease of 20**19**) was declared a global pandemic on February 11, 2020 by the World Health Organisation (WHO). Caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), this disease affects the respiratory system and includes symptoms like fatigue, dry cough, shortness of breath, joint pain, muscle pain, gastrointestinal symptoms and loss of smell or taste [1,2]. Due to its effect on the vascular endothelium, the acute respiratory distress syndrome can originate from either the gas or vascular side of the alveolus which becomes visible in a chest x-ray or computed tomography (CT) scan for COVID-19 patients [3,4]. Among the patients infected with SARS-CoV-2, between 5% and 20% are admitted to an intensive care unit (ICU) and their mortality rate varies between 26% and 62% [5]. Medical lab tests are available to diagnose COVID-19 by analysing exhaled breaths [6]. This technique was reported to achieve an accuracy of 93% when considering a group of 28 COVID-19 positive and 12 COVID-19 negative patients [7]. Related work using a group of 25 COVID-19 positive and 65 negative patients achieved an area under the ROC curve (AUC) of 0.87 [8].

Previously, machine learning algorithms have been applied to detect COVID-19 using image analysis. For example, COVID-19 was detected from CT images using a Resnet50 architecture with 96.23% accuracy in Ref. [9]. The same architecture also detected pneumonia due to COVID-19 with an accuracy of 96.7% [10] and COVID-19 from x-ray images with an accuracy of 96.30% [11].

* Corresponding author.
** Corresponding author.
E-mail addresses: mpahar@sun.ac.za (M. Pahar), marisat@sun.ac.za (M. Klopper), rw1@sun.ac.za (R. Warren), trn@sun.ac.za (T. Niesler).

https://doi.org/10.1016/j.compbiomed.2021.105153
Received 17 November 2021; Received in revised form 24 November 2021; Accepted 14 December 2021
Available online 17 December 2021

The automatic analysis of cough audio for COVID-19 detection has also received recent attention. Coughing is a predominant symptom of many lung ailments and its effect on the respiratory system varies [12,13]. Lung disease can cause the glottis to behave differently and the airway to be either restricted or obstructed and this can influence the acoustics of the vocal audio such as cough, breath and speech [14,15]. This raises the prospect of identifying the coughing audio associated with a particular respiratory disease such as COVID-19 [16,17]. Researchers have found that a simple binary machine learning classifier can distinguish between healthy and COVID-19 respiratory audio, such as coughs gathered from crowdsourced data, with an AUC above 0.8 [18]. Improved performance was achieved using a convolutional neural network (CNN) for cough and breath audio, achieving an AUC of 0.846 [19].

In our previous work, we have also found that automatic COVID-19 detection is possible on the basis of the acoustic cough signal [20]. Here we extend this work firstly by considering whether breath and speech audio can also be used effectively for COVID-19 detection. Secondly, since the COVID-19 datasets at our disposal are comparatively small, we apply transfer learning and extract bottleneck features to take advantage of other datasets that do not include COVID-19 labels. To do this, we use publicly available as well as our own datasets that do not include COVID-19 labels to pre-train three deep neural network (DNN) architectures: a CNN, a long short-term memory (LSTM) and a 50-layer residual-based architecture (Resnet50), which uses convolutional layers with skip connections. For subsequent COVID-19 classifier evaluation, we used the Coswara dataset [21], the Interspeech Computational Paralinguistics ChallengE (ComParE) dataset [22] and the Sarcos dataset [20], all of which do contain COVID-19 labels. We report further evidence of accurate discrimination using all three audio classes and conclude that vocal audio including coughing, breathing and speech are all affected by the condition of the lungs to an extent that they carry acoustic information that can be used by existing machine learning classifiers to detect signatures of COVID-19. We are also able to show that the variability in performance of the classifiers, as measured over the independent outer folds of nested cross-validation, is strongly reduced by the pre-training, despite the absence of COVID-19 labels in the pre-training data. We can therefore conclude that the application of transfer learning enables the COVID-19 classifiers to perform both more accurately and with greater greater consistency. This is key to the viability of the practical implementation of cough audio screening, where test data can be expected to be variable, depending for example on the location and method of data capture.

Sections 2 and Section 3 summarise the datasets used for experimentation and the primary feature extraction process. Section 4 describes the transfer learning process and Section 5 explains the bottleneck feature extraction process. Section 6 presents the experimental setup, including the cross-validated hyperparameter optimisation and classifier evaluation process. Experimental results are presented in Section 7 and discussed in Section 8. Finally, Section 9 summarises and concludes this study.

2. Data

2.1. Datasets without COVID-19 labels for pre-training

Audio data with COVID-19 labels remain scarce, which limits classifier training. We have therefore made use of five datasets without COVID-19 labels for pre-training. These datasets contain recordings of coughing, sneezing, speech and non-vocal audio. The first three datasets (TASK, Brooklyn and Wallacedene) were compiled by ourselves as part of research projects concerning cough monitoring and cough classification. The last two (Google Audio Set & Freesound and Librispeech) were compiled from publicly available data. Since all five datasets were compiled before the start of the COVID-19 pandemic, they are unlikely to contain data from COVID-19 positive subjects. All datasets used for pre-training include manual annotations but exclude COVID-19 labels.

2.1.1. TASK dataset

This corpus consists of spontaneous coughing audio collected at a small tuberculosis (TB) clinic near Cape Town, South Africa [23]. The dataset contains 6000 recorded coughs by patients undergoing TB treatment and 11 393 non-cough sounds such as laughter, doors opening and objects moving. This data was intended for the development of cough detection algorithms and the recordings were made in a multi-bed ward environment using a smartphone with an attached external microphone. The annotations consist of the time locations and labels of sounds, including coughs.

2.1.2. Brooklyn dataset

This dataset contains recordings of 746 voluntary coughs by 38 subjects compiled for the development of TB cough audio classification systems [24]. Audio recording took place in a controlled indoor booth, using a RØDE M3 microphone and an audio field recorder. The annotations include the start and end times of each cough.

2.1.3. Wallacedene dataset

This dataset consists of recordings of 1358 voluntary coughs by 51 patients, also compiled for the development of TB cough audio classification [25]. In this case, audio recording took place in an outdoor booth located at a busy primary healthcare clinic. Recording was performed using a RØDE M1 microphone and an audio field recorder. This data has more environmental noise and therefore a poorer signal-to-noise ratio than the Brooklyn dataset. As for the Brooklyn dataset, annotations include the start and end times of each cough.

2.1.4. Google Audio Set & Freesound

The Google Audio Set dataset contains excerpts from 1.8 million Youtube videos that have been manually labelled according to an ontology of 632 audio event categories [26]. The Freesound audio database is a collection of tagged sounds uploaded by contributors from around the world [27]. In both datasets, the audio recordings were contributed by many different individuals under widely varying recording conditions and noise levels. From these two datasets, we have compiled a collection of recordings that include 3098 coughing sounds, 1013 sneezing sounds, 2326 speech excerpts and 1027 other non-vocal sounds such as engine noise, running water and restaurant chatter. Previously, this dataset was used for the development of cough detection algorithms [28]. Annotations consist of the time locations and labels of the particular sounds.

2.1.5. LibriSpeech

As a source of speech audio data, we have selected utterances by 28 male and 28 female speakers from the freely available LibriSpeech corpus [29]. These recordings contain very little noise. The large size of the corpus allowed easy gender balancing.

2.1.6. Summary of data used for pre-training

In total, the data described above includes 11 202 cough sounds (2.45 h of audio), 2.91 h of speech from both male and female participants, 1013 sneezing sounds (13.34 min of audio) and 2.98 h of other non-vocal audio. Hence sneezing is under-represented as a class in the pre-training data. Since such an imbalance can detrimentally affect the

performance of neural networks [30,31], we have applied the synthetic minority over-sampling technique (SMOTE) [32]. SMOTE oversamples the minor class by creating additional synthetic samples rather than, for example, random oversampling. We have in the past successfully applied SMOTE to address training set class imbalances in cough detection [23] and cough classification [20] based on audio recordings.

In total, therefore, a dataset containing 10.29 h of audio recordings annotated with four class labels (cough, speech, sneeze, noise) was available to pre-train the neural architectures. The composition of this dataset is summarised in Table 1. All recordings used for pre-training were downsampled to 16 kHz.

2.2. Datasets with COVID-19 labels for classification

Three datasets of coughing audio with COVID-19 labels were available for experimentation.

2.2.1. Coswara dataset

This dataset is specifically developed with the testing of classification algorithms for COVID-19 detection in mind. Data collection is web-based, and participants contribute by using their smartphones to record their coughing, breathing and speech. Audio recordings were collected of both shallow and deep breaths as well as speech uttered at a normal and fast pace. However, since the deep breaths consistently outperformed the shallow breaths in our initial experiments, the latter will not be presented in our experiments. At the time of writing, the data included contributions from participants located on five different continents [20,21,33].

Figs. 1 and 2 show examples of Coswara breaths and speech respectively, collected from both COVID-19 positive and COVID-19 negative subjects. It is evident that breaths have more higher-frequency content than speech and interesting to note that COVID-19 breaths are, on average, 30% shorter than non-COVID-19 breaths (Table 2). All audio recordings were pre-processed to remove periods of silence to within a margin of 50 ms using a simple energy detector.

2.2.2. ComParE dataset

This dataset was provided as a part of the 2021 Interspeech Computational Paralinguistics ChallengE (ComParE) [22]. The ComParE dataset contains recordings of both coughs and speech, where the latter is the utterance 'I hope my data can help to manage the virus pandemic' in the speaker's language of choice.

2.2.3. Sarcos dataset

This dataset was collected in South Africa as part of this research and currently contains recordings of coughing by 18 COVID-19 positive and 26 COVID-19 negative subjects. Audio was pre-processed in the same way as the Coswara data. Since this dataset is very small, we have used it in our previous work exclusively for independent validation [20]. In this study, however, it has also been used to fine-tune and evaluate pre-trained DNN classifiers by means of transfer learning and the extraction of bottleneck features.

2.2.4. Summary of data used for classification

Table 2 shows that the COVID-19 positive class is under-represented in all datasets available for classification. To address this, we again apply SMOTE during training. We also note that the Coswara dataset contains the largest number of subjects, followed by ComParE and finally Sarcos. As for pre-training, all recordings were downsampled to 16 kHz.

3. Primary feature extraction

From the time-domain audio signals, we have extracted mel-frequency cepstral coefficients (MFCCs) and linearly-spaced log filterbank energies, along with their respective velocity and acceleration coefficients. We have also extracted the signal zero-crossing rate (ZCR) [34] and kurtosis [34], which are indicative respectively of time-domain signal variability and tailedness, i.e. the prevalence of higher amplitudes.

MFCCs have been very effective in speech processing [35], but also in discriminating dry and wet coughs [36], and recently in characterising COVID-19 audio [37]. Linearly-spaced log filterbank energies have proved useful in several biomedical applications, including cough audio classification [24,25,38].

Features are extracted from overlapping frames, where the frame overlap δ is computed to ensure that the audio signal is always divided into exactly $\mathcal{S}$ frames, as illustrated in Fig. 3. This ensures that the entire audio event is always represented by a fixed number of frames, which allows a fixed input dimension to be maintained for classification while preserving the general overall temporal structure of the sound. Such fixed two-dimensional feature dimensions are particularly useful for the training of DNN classifiers, and have performed well in our previous experiments [20].

The frame length ($\mathcal{F}$), number of frames ($\mathcal{S}$), number of lower order MFCCs ($\mathcal{M}$) and number of linearly spaced filters ($\mathcal{B}$) are regarded as feature extraction hyperparameters, listed in Table 3. The table shows that in our experiments each audio signal is divided into between 70 and 200 frames, each of which consists of between 512 and 4096 samples, corresponding to between 32 msec and 256 msec of audio. The number of extracted MFCCs ($\mathcal{M}$) lies between 13 and 65, and the number of linearly-spaced filterbanks ($\mathcal{B}$) between 40 and 200. This allows the spectral information included in each feature to be varied.

The input feature matrix to the classifiers has the dimension of $(3\mathcal{M}+2, \mathcal{S})$ for $\mathcal{M}$ MFCCs along with their $\mathcal{M}$ velocity and $\mathcal{M}$ acceleration coefficients, as shown in Fig. 3. Similarly, for linearly spaced filters, the dimension of the feature matrix is $(3\mathcal{B}+2, \mathcal{S})$.

Table 1
Summary of the Datasets used in Pre-training. Classifiers are pre-trained on 10.29 h audio recordings annotated with four class labels: cough, sneeze, speech and noise. The datasets do not include any COVID-19 labels.

Type	Dataset	Sampling Rate	No of Events	Total audio	Average length	Standard deviation
Cough	TASK dataset	44.1 kHz	6000	91 min	0.91 s	0.25 s
	Brooklyn dataset	44.1 kHz	746	6.29 min	0.51 s	0.21 s
	Wallacedene dataset	44.1 kHz	1358	17.42 min	0.77 s	0.31 s
	Google Audio Set & Freesound	16 kHz	3098	32.01 min	0.62 s	0.23 s
	Total (Cough)	—	11 202	2.45 h	0.79 s	0.23 s
Sneeze	Google Audio Set & Freesound	16 kHz	1013	13.34 min	0.79 s	0.21 s
	Google Audio Set & Freesound + SMOTE	16 kHz	9750	2.14 h	0.79 s	0.23 s
	Total (Sneeze)	—	10 763	2.14 h	0.79 s	0.23 s
Speech	Google Audio Set & Freesound	16 kHz	2326	22.48 min	0.58 s	0.14 s
	LibriSpeech	16 kHz	56	2.54 h	2.72 min	0.91 min
	Total (Speech)	—	2382	2.91 h	4.39 s	0.42 s
Noise	TASK dataset	44.1 kHz	12 714	2.79 h	0.79 s	0.23 s
	Google Audio Set & Freesound	16 kHz	1027	11.13 min	0.65 s	0.26 s
	Total (Noise)	—	13 741	2.79 h	0.79 s	0.23 s

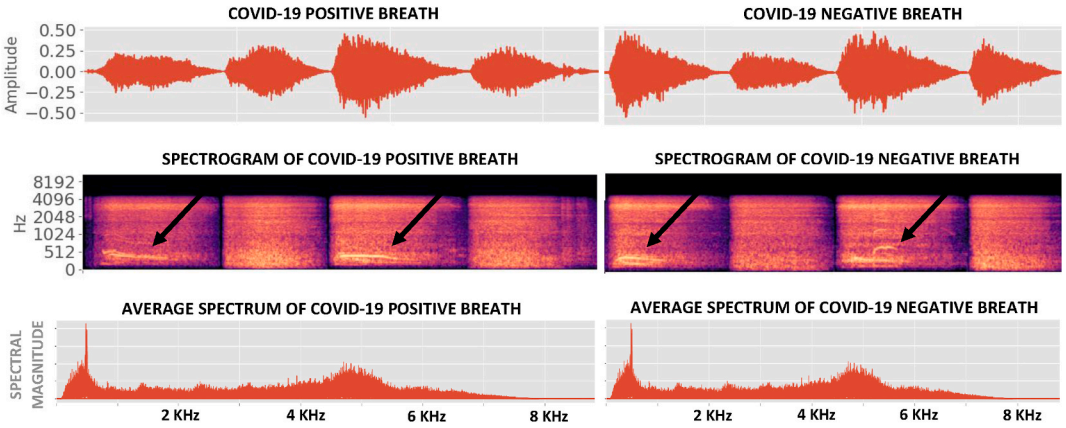

Fig. 1. Pre-processed breath signals from both COVID-19 positive and COVID-19 negative subjects in the Coswara dataset. Breaths corresponding to inhalation are marked by arrows, and are followed by an exhalation.

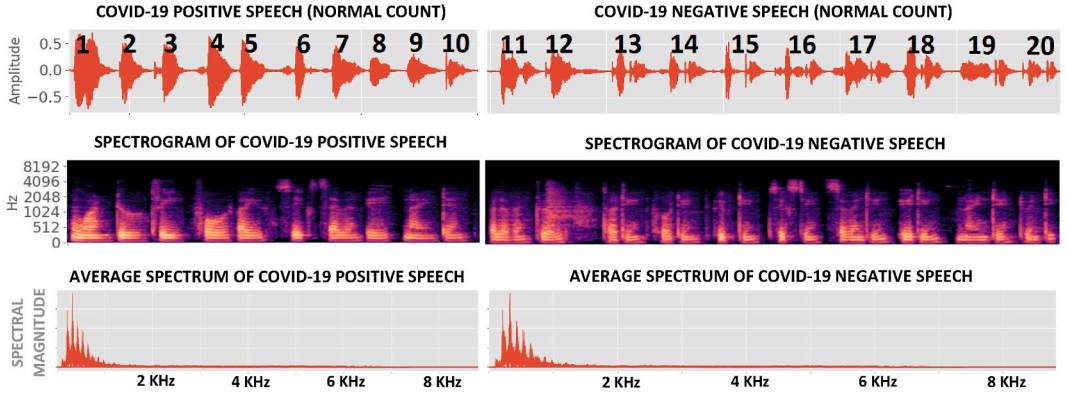

Fig. 2. Pre-processed speech (counting from 1 to 20 at a normal pace) from both COVID-19 positive and COVID-19 negative subjects in the Coswara dataset. In contrast to breath (Fig. 1), the spectral energy in this speech is concentrated below 1 kHz.

Table 2
Summary of the datasets used for COVID-19 classification. Cough, breath and speech signals were extracted from the Coswara, ComParE and Sarcos datasets. COVID-19 positive subjects are under-represented in all three.

Type	Dataset	Sampling Rate	Label	Subjects	Total audio	Average per subject	Standard deviation
Cough	Coswara	44.1 kHz	COVID-19 Positive	92	4.24 min	2.77 s	1.62 s
			Healthy	1079	0.98 h	3.26 s	1.66 s
			Total	1171	1.05 h	3.22 s	1.67 s
	ComParE	16 kHz	COVID-19 Positive	119	13.43 min	6.77 s	2.11 s
			Healthy	398	40.89 min	6.16 s	2.26 s
			Total	517	54.32 min	6.31 s	2.24 s
	Sarcos	44.1 kHz	COVID-19 Positive	18	0.87 min	2.91 s	2.23 s
			COVID-19 Negative	26	1.57 min	3.63 s	2.75 s
			Total	44	2.45 min	3.34 s	2.53 s
Breath	Coswara	44.1 kHz	COVID-19 Positive	88	8.58 min	5.85 s	5.05 s
			Healthy	1062	2.77 h	9.39 s	5.23 s
			Total	1150	2.92 h	9.126 s	5.29 s
Speech	Coswara (normal)	44.1 kHz	COVID-19 Positive	88	12.42 min	8.47 s	4.27 s
			Healthy	1077	2.99 h	9.99 s	3.09 s
			Total	1165	3.19 h	9.88 s	3.22 s
	Coswara (fast)	44.1 kHz	COVID-19 Positive	85	7.62 min	5.38 s	2.76 s
			Healthy	1074	1.91 h	6.39 s	1.77 s
			Total	1159	2.03 h	6.31 s	1.88 s
	ComParE	16 kHz	COVID-19 Positive	214	44.02 min	12.34 s	5.35 s
			Healthy	396	1.46 h	13.25 s	4.67 s
			Total	610	2.19 h	12.93 s	4.93 s

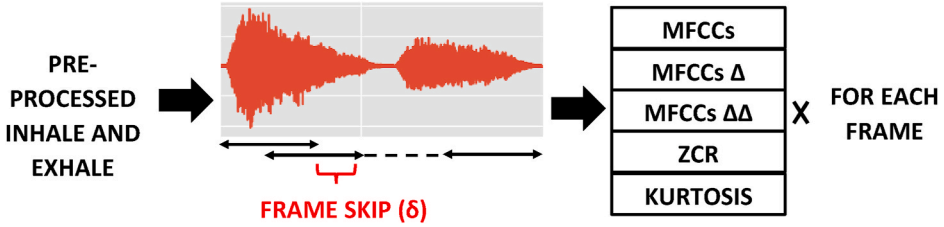

Fig. 3. Feature extraction process for a breath audio. The frame overlap δ is calculated to ensure that the entire recording is divided into $\mathcal{S}$ segments. For $\mathcal{M}$ MFCCs, for example, this results in a feature matrix with dimensions $(3\mathcal{M}+2, \mathcal{S})$.

Table 3
Primary feature (PF) extraction hyperparameters. We have used between 13 and 65 MFCCs and between 40 and 200 linearly spaced filters to extract log energies.

Hyperparameters	Description	Range
MFCCs ($\mathcal{M}$)	lower order MFCCs to keep	$13 \times k$, where $k = 1, 2, 3, 4, 5$
Linearly spaced filters ($\mathcal{B}$)	used to extract log energies	40 to 200 in steps of 20
Frame length ($\mathcal{F}$)	into which audio is segmented	2^k where $k = 9, 10, 11, 12$
Segments ($\mathcal{S}$)	number of frames extracted from audio	$10 \times k$, where $k = 7, 10, 12, 15, 20$

We will refer to the features described in this section as **primary features** (PF) to distinguish them from the bottleneck features (BNF) described in Section 5.

4. Transfer learning architecture

Since the audio datasets with COVID-19 labels described in Section 2.2 are small, they may lead to overfitting when training deep architectures. Nevertheless, in previous work we have found that deep architectures perform better than shallow classifiers when using these as training sets [20]. In this work, we consider whether the classification performance of such DNNs can be improved by applying transfer learning.

To achieve this, we use the datasets described in Section 2.1 containing 10.29 h of audio, labelled with four classes: cough, sneeze, speech and noise, but that do not include COVID-19 labels (Table 1 in Section 2.1). This data is used to pre-train three deep neural architectures: a CNN, an LSTM and a Resnet50. The feature extraction hyperparameters: $\mathcal{M} = 39, \mathcal{F} = 2^{10}$ and $\mathcal{S} = 150$ delivered good performance in our previous work [20] and thus have also been used here (Table 4).

The CNN consists of three convolutional layers, with 256, 128 and 64 (2×2) kernels respectively, each followed by (2,2) max-pooling. The LSTM consists of three layers with 512, 256 and 128 LSTM units respectively, each including dropout with a rate of 0.2. A standard Resnet50, as described in Table 1 of [39], has been implemented with 512-dimensional dense layers.

During pre-training, all three networks (CNN, LSTM and Resnet50) are terminated by three dense layers with dimensionalities 512, 64 and finally 4 to correspond to the four classes mentioned in Table 1. Relu activation functions were used throughout, except in the four-dimensional output layer which was softmax. All the above architectural hyperparameters were chosen by optimising the four-class classifiers during cross-validation (Table 4).

After pre-training on the datasets described in Section 2.1, the 64 and 4-dimensional dense layers terminating the network were discarded from the CNN, the LSTM and the Resnet50. This left three trained deep neural networks, each accepting the same input dimensions and each with a 512-dimensional relu output layer. The parameters of these three pre-trained networks were then fixed for the remaining experiments.

In order to obtain COVID-19 classifiers by transfer learning, two dense layers are added after the 512-dimensional output layer of each of the three pre-trained deep networks. The final layer is a two-dimensional softmax, to indicate COVID-19 positive and negative classes respectively. The dimensionality of the penultimate layer was also considered to be a hyperparameter and was optimised during nested k-fold cross-validation. Its optimal value was found to be 32 for all three architectures. The transfer learning process for a CNN architecture is illustrated in Fig. 4.

Table 4
Hyperparameters of the pre-trained networks: Feature extraction hyperparameters were adopted from the optimal values in previous related work [20], while classifier hyperparameters were optimised on the pre-training data using cross-validation.

FEATURE EXTRACTION HYPERPARAMETERS		
Hyperparameters		Values
$\mathcal{M}$	MFCCs	39
$\mathcal{F}$	Frame length	$2^{10} = 1024$
$\mathcal{S}$	Segments	150
CLASSIFIER HYPERPARAMETERS		
Hyperparameters	**Classifier**	**Values**
Convolutional filters	CNN	256 & 128 & 64
Kernel size	CNN	2
Dropout rate	CNN, LSTM	0.2
Dense layer (for pre-training)	CNN, LSTM, Resnet50	512 & 64 & 4
Dense layer (for fine-tuning)	CNN, LSTM, Resnet50	32 & 2
LSTM units	LSTM	512 & 256 & 128
Learning rate	LSTM	$10^{-3} = 0.001$
Batch Size	CNN, LSTM, Resnet50	$2^7 = 128$
Epochs	CNN, LSTM, Resnet50	70

5. Bottleneck features

The 512-dimensional output of the three pre-trained networks described in the previous section has a much lower dimensionality than the $(3\mathcal{M}+2, \mathcal{S})$ i.e. $(3 \times 39 + 2) \times 150 = 17\,850$ dimensional input matrix consisting of primary features (Table 4). Therefore, the output of this layer can be viewed as a bottleneck feature vector [40–42]. In addition to fine-tuning, where we add terminating dense layers to the three pre-trained networks and optimise these for the binary COVID-19 detection task as shown in Fig. 4, we have trained logistic regression (LR), support vector machine (SVM), k-nearest neighbour (KNN) and multilayer perceptron (MLP) classifiers using these bottleneck features as inputs. Bottleneck features computed by the CNN, the LSTM or the Resnet50 were chosen based on the one which performed better in the corresponding transfer learning experiments. Since the Resnet50 achieved higher development set AUCs than the CNN and the LSTM during transfer learning, it was used to extract bottleneck features on which the LR, SVM, KNN and MLP classifiers were trained.

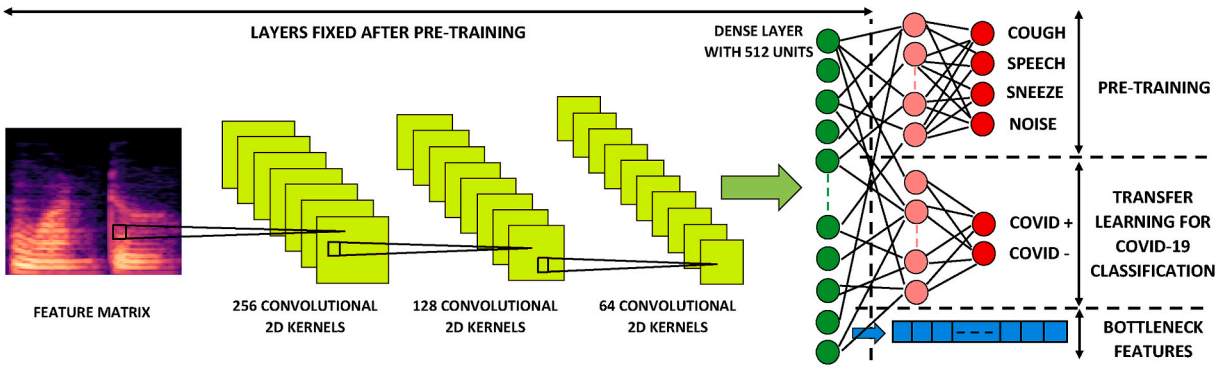

Fig. 4. CNN Transfer Learning Architecture. Cross-validation on the pre-training data determined the optimal CNN architecture to have three convolutional layers with 256, 128 and 64 (2×2) kernels respectively, each followed by (2,2) max-pooling. The convolutional layers were followed by two dense layers with 512 and 64 relu units each, and the network was terminated by a 4-dimensional softmax. To apply transfer learning, the final two layers were removed and replaced with a new dense layer and a terminating 2-dimensional softmax to account for COVID-19 positive and negative classes. Only this newly added portion of the network was trained for classification on the data with COVID-19 labels. In addition, the outputs of the third-last layer (512-dimensional dense relu) from the pre-trained network were used as bottleneck features.

6. Experimental method

We have evaluated the effectiveness of transfer learning (Section 4) and bottleneck feature extraction (Section 5) using CNN, LSTM and Resnet50 architectures in improving the performance of COVID-19 classification based on cough, breath and speech audio signals. In order to place these results in context, we provide two baselines.

1. As a first baseline, we train the three deep architectures (CNN, LSTM and Resnet50) directly on the primary features extracted from data containing COVID-19 labels (as described in Section 2.2) and hence skip the pre-training. Some of these baseline results were developed in our previous work [20].
2. As a second baseline, we train shallow classifiers (LR, SVM, KNN and MLP) on the primary input features (as described in Section 3), also extracted from the data containing COVID-19 labels (described in Section 2.2).

The performance of these baseline systems will be compared against:

1. Deep architectures (CNN, LSTM and Resnet50) trained by the transfer learning process. The respective deep architectures are pre-trained (as described in Section 4), after which the final two layers are fine-tuned on the data containing COVID-19 labels (as described in Section 2.2).
2. Shallow architectures (LR, SVM, KNN and MLP) trained on the bottleneck features extracted from the pre-trained networks.

Table 5
Classifier hyperparameters, optimised using leave-*p*-out nested cross-validation.

Hyperparameters	Classifier	Range
Regularisation Strength (α_1)	LR, SVM	10^i where, $i = -7, -6, \ldots, 6, 7$
$l1$ penalty (α_2)	LR	0 to 1 in steps of 0.05
$l2$ penalty (α_3)	LR, MLP	0 to 1 in steps of 0.05
Kernel Coefficient (α_4)	SVM	10^i where, $i = -7, -6, \ldots, 6, 7$
No. of neighbours (α_5)	KNN	10 to 100 in steps of 10
Leaf size (α_6)	KNN	5 to 30 in steps of 5
No. of neurons (α_7)	MLP	10 to 100 in steps of 10
No. of convolutional filters (β_1)	CNN	3×2^k where $k = 3, 4, 5$
Kernel size (β_2)	CNN	2 and 3
Dropout rate (β_3)	CNN, LSTM	0.1 to 0.5 in steps of 0.2
Dense layer size (β_4)	CNN, LSTM	2^k where $k = 4, 5$
LSTM units (β_5)	LSTM	2^k where $k = 6, 7, 8$
Learning rate (β_6)	LSTM, MLP	10^k where, $k = -2, -3, -4$
Batch Size (β_7)	CNN, LSTM	2^k where $k = 6, 7, 8$
Epochs (β_8)	CNN, LSTM	10 to 250 in steps of 20

6.1. Hyperparameter optimisation

Hyperparameters for three pre-trained networks have already been described in Section 4 and are listed in Table 4. The remaining hyperparameters are those of the baseline deep classifiers (CNN, LSTM and Resnet50 without pre-training), the four shallow classifiers (LR, SVM, KNN and MLP), and the dimensionality of the penultimate layer for the deep architectures during transfer learning.

With the exception of Resnet50, all these hyperparameters optimisation and performance evaluation has been performed within the inner loops of nested k-fold cross-validation scheme [43]. Due to the excessive computational requirements of optimising Resnet50 metaparameters within the same cross-validation framework, we have used the standard 50 skip layers in all experiments [39]. Classifier hyperparameters and the values considered during optimisation are listed in Table 5. A five-fold split, similar to that employed in Ref. [20], was used for the nested cross-validation.

6.2. Classifier evaluation

Receiver operating characteristic (ROC) curves were calculated within both the inner and outer loops of the nested cross-validation scheme described in the previous section. The inner-loop ROC values were used for the hyperparameter optimisation, while the average of the outer-loop ROC values indicates final classifier performance on the independent held-out test sets. The AUC score indicates how well the classifier performs over a range of decision thresholds [44]. The threshold that achieves an equal error rate (γ_{EE}) was computed from these curves.

We note the mean per-frame probability that an event such as a cough is from a COVID-19 positive subject by $\widehat{P}$:

$$\widehat{P} = \frac{\sum_{i=1}^{K} P(Y = 1|X_i, \theta)}{\mathcal{S}} \tag{1}$$

where $\mathcal{S}$ indicates the number of frames in an event and $P(Y = 1|X_i, \theta)$ is the output of the classifier for feature vector X_i and parameters θ for the ith frame. Now we define the indicator variable C as:

$$C = \begin{cases} 1 & \text{if } \widehat{P} \geq \gamma_{EE} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We then define two COVID-19 index scores CI_1 and CI_2 in Equations (3) and (4) respectively, with N_1 the number of events from the subject in the recording and N_2 the total number of frames of the events gathered from the subject. Here, $N_2 = \mathcal{S} \times N_1$.

Table 6
COVID-19 cough classification performance. For the Coswara, Sarcos and ComParE datasets the highest AUCs of 0.982, 0.961 and 0.944 respectively were achieved by a Resnet50 trained by transfer learning in the first two cases and a KNN classifier using 12 primary features determined by sequential forward selection (SFS) in the third. When Sarcos is used exclusively as a validation set for a classifier trained on the Coswara data, an AUC of 0.954 is achieved.

Dataset	ID	Classifier	Best Feature Hyperparameters	Best Classifier Hyperparameters (Optimised inside nested cross-validation)	Performance				
					Spec	Sens	Acc	AUC	σ_{AUC}
Coswara	*C1*	*Resnet50 + TL*	Table 4	*Default Resnet50 (Table 1 in* Ref. [39]*)*	*97%*	*98%*	*97%*	*0.982*	2×10^{-3}
	C2	CNN + TL	”	Table 4	92%	98%	95%	0.972	3×10^{-3}
	C3	LSTM + TL	”	”	93%	95%	94%	0.964	3×10^{-3}
	C4	MLP + BNF	”	$\alpha_3 = 0.35$, $\alpha_7 = 50$	92%	96%	94%	0.963	4×10^{-3}
	C5	SVM + BNF	”	$\alpha_1 = 10^4$, $\alpha_4 = 10^1$	89%	93%	91%	0.942	3×10^{-3}
	C6	KNN + BNF	”	$\alpha_5 = 20$, $\alpha_6 = 15$	88%	90%	89%	0.917	7×10^{-3}
	C7	LR + BNF	”	$\alpha_1 = 10^{-1}$, $\alpha_2 = 0.5$, $\alpha_3 = 0.5$	84%	86%	85%	0.898	8×10^{-3}
	C8	Resnet50 + PF [20]	Table 4 in [20]	Default Resnet50 (Table 1 in Ref. [39])	98%	93%	95%	0.976	18×10^{-3}
	C9	CNN + PF [20]	”	Table 4 in [20]	99%	90%	95%	0.953	39×10^{-3}
	C10	LSTM + PF [20]	”	”	97%	91%	94%	0.942	43×10^{-3}
Sarcos	*C11*	*Resnet50 + TL*	Table 4	*Default Resnet50 (Table 1 in* Ref. [39]*)*	*92%*	*96%*	*94%*	*0.961*	3×10^{-3}
	C12	LSTM + TL	”	Table 4	92%	92%	92%	0.943	3×10^{-3}
	C13	CNN + TL	”	”	89%	91%	90%	0.917	4×10^{-3}
	C14	MLP + BNF	”	$\alpha_3 = 0.75$, $\alpha_7 = 70$	88%	90%	89%	0.913	7×10^{-3}
	C15	SVM + BNF	”	$\alpha_1 = 10^{-2}$, $\alpha_4 = 10^4$	88%	89%	89%	0.904	6×10^{-3}
	C16	KNN + BNF	”	$\alpha_5 = 40$, $\alpha_6 = 20$	85%	87%	86%	0.883	8×10^{-3}
	C17	LR + BNF	”	$\alpha_1 = 10^{-3}$, $\alpha_2 = 0.4$, $\alpha_3 = 0.6$	83%	86%	85%	0.867	9×10^{-3}
Sarcos (val only)	*C18*	*Resnet50 + TL*	”	*Default Resnet50 (Table 1 in* Ref. [39]*)*	*92%*	*96%*	*94%*	*0.954*	–
	C19	LSTM + PF [20]	Table 5 in [20]	Table 5 in [20]	73%	75%	74%	0.779	–
	C20	LSTM + PF + SFS [20]	”	”	96%	91%	93%	0.938	–
ComParE	C21	Resnet50 + TL	Table 4	Default Resnet50 (Table 1 in Ref. [39])	89%	93%	91%	0.934	4×10^{-3}
	C22	LSTM + TL	”	Table 4	88%	92%	90%	0.916	4×10^{-3}
	C23	CNN + TL	”	”	86%	90%	88%	0.898	4×10^{-3}
	C24	MLP + BNF	”	$\alpha_3 = 0.25$, $\alpha_7 = 20$	85%	90%	88%	0.912	5×10^{-3}
	C25	SVM + BNF	”	$\alpha_1 = 10^{-3}$, $\alpha_4 = 10^2$	85%	90%	88%	0.903	6×10^{-3}
	C26	KNN + BNF	”	$\alpha_5 = 70$, $\alpha_6 = 20$	85%	86%	86%	0.882	8×10^{-3}
	C27	LR + BNF	”	$\alpha_1 = 10^4$, $\alpha_2 = 0.3$, $\alpha_3 = 0.7$	84%	86%	85%	0.863	8×10^{-3}
	C28	*KNN + PF + SFS*	$\mathcal{B} = 60, \mathcal{F} = 2^{11}, \mathcal{S} = 70$	$\alpha_5 = 60$, $\alpha_6 = 25$	*84%*	*90%*	*92%*	*0.944*	9×10^{-3}
	C29	KNN + PF	$\mathcal{B} = 60, \mathcal{F} = 2^{11}, \mathcal{S} = 70$	$\alpha_5 = 60$, $\alpha_6 = 25$	78%	80%	80%	0.855	13×10^{-3}
	C30	MLP + PF	$\mathcal{M} = 13, \mathcal{F} = 2^{10}, \mathcal{S} = 100$	$\alpha_3 = 0.65$, $\alpha_7 = 40$	76%	80%	78%	0.839	14×10^{-3}
	C31	SVM + PF	$\mathcal{B} = 80, \mathcal{F} = 2^9, \mathcal{S} = 70$	$\alpha_1 = 10^{-4}$, $\alpha_4 = 10^{-1}$	75%	78%	77%	0.814	12×10^{-3}
	C32	LR + PF	$\mathcal{B} = 140, \mathcal{F} = 2^{11}, \mathcal{S} = 70$	$\alpha_1 = 10^{-2}$, $\alpha_2 = 0.6$, $\alpha_3 = 0.4$	69%	73%	71%	0.789	13×10^{-3}

$$\mathrm{CI}_1 = \frac{\sum_{i=1}^{N_1} C}{N_1} \tag{3}$$

$$\mathrm{CI}_2 = \frac{\sum_{i=1}^{N_2} P(Y=1|X_i)}{N_2} \tag{4}$$

Hence Equation (1) computes a per-event average probability while Equation (4) computes a per-frame average probability. The use of one of Equations (3) and (4) was considered an additional hyperparameter during cross-validation, and it was found that taking the maximum value of the index scores consistently led to the best performance.

The average specificity, sensitivity and accuracy, as well as the AUC together with its standard deviation (σ_{AUC}) are shown in Tables 6–8 for cough, breath and speech events respectively. These values have all been calculated over the outer folds during nested cross-validation. Hyperparameters producing the highest AUC over the inner loops have been noted as the ‘best classifier hyperparameter’.

7. Experimental results

COVID-19 classification performance based on cough, breath and speech is presented in Tables 6–8 respectively. These tables include the performance of baseline deep classifiers without pre-training, deep classifiers trained by transfer learning (TL), shallow classifiers using bottleneck features (BNF) and baseline shallow classifiers trained directly on the primary features (PF). The best performing classifiers appear first for each dataset and the baseline results are shown towards the end. Each system is identified by an ‘ID’.

7.1. Coughs

We have found in our previous work [20] that, when training a Resnet50 on only the Coswara dataset, an AUC of 0.976 ($\sigma_{AUC} = 0.018$) can be achieved for the binary classification problem of distinguishing COVID-19 coughs from healthy coughs. These results are reproduced as baseline systems C8, C9 and C10 in Table 6. The improved results achieved by transfer learning are indicated by systems C1 to C7 in the same table. Specifically, system C1 shows that, by applying transfer learning as described in Section 4, the same Resnet50 architecture can achieve an AUC of 0.982 ($\sigma_{AUC} = 0.002$). The entries for systems C2 and C3 show that pre-training also improves the AUCs achieved by the deep CNN and LSTM classifiers from 0.953 (system C9) to 0.972 (system C2) and from 0.942 (system C10) to 0.964 (system C3) respectively. Of particular note

Table 7
COVID-19 breath classifier performance: For breaths, the best performance was achieved by an SVM using bottleneck features (AUC = 0.942). The Resnet50 classifier trained by transfer learning achieves a similar AUC of 0.934.

Dataset	ID	Classifier	Best Feature Hyperparameters	Best Classifier Hyperparameters (Optimised inside nested cross-validation)	Performance				
					Spec	Sens	Acc	AUC	σ_{AUC}
Coswara	B1	Resnet50 + TL	Table 4	Default Resnet50 (Table 1 in Ref. [39])	87%	93%	90%	0.934	3×10^{-3}
	B2	LSTM + TL	”	Table 4	86%	90%	88%	0.927	3×10^{-3}
	B3	CNN + TL	”	”	85%	89%	87%	0.914	3×10^{-3}
	B4	*SVM + BNF*	”	$\alpha_1 = 10^2$, $\alpha_4 = 10^{-2}$	*88%*	*94%*	*91%*	*0.942*	4×10^{-3}
	B5	MLP + BNF	”	$\alpha_3 = 0.45$, $\alpha_7 = 50$	87%	93%	90%	0.923	6×10^{-3}
	B6	KNN + BNF	”	$\alpha_5 = 70$, $\alpha_6 = 10$	87%	93%	90%	0.922	9×10^{-3}
	B7	LR + BNF	”	$\alpha_1 = 10^{-4}$, $\alpha_2 = 0.8$, $\alpha_3 = 0.2$	86%	90%	88%	0.891	8×10^{-3}
	B8	Resnet50 + PF	$\mathcal{M} = 39, \mathcal{F} = 2^{10}, \mathcal{S} = 150$	Default Resnet50 (Table 1 in Ref. [39])	92%	90%	91%	0.923	34×10^{-3}
	B9	LSTM + PF	$\mathcal{M} = 26, \mathcal{F} = 2^{11}, \mathcal{S} = 120$	$\beta_3 = 0.1$, $\beta_4 = 32$, $\beta_5 = 128$, $\beta_6 = 0.001$, $\beta_7 = 256$, $\beta_8 = 170$	90%	86%	88%	0.917	41×10^{-3}
	B10	CNN + PF	$\mathcal{M} = 52, \mathcal{F} = 2^{10}, \mathcal{S} = 100$	$\beta_1 = 48$, $\beta_2 = 2$, $\beta_3 = 0.3$, $\beta_4 = 32$, $\beta_7 = 256$, $\beta_8 = 210$	87%	85%	86%	0.898	42×10^{-3}

Table 8
COVID-19 speech classifier performance: For the Coswara (fast and normal speech) and the ComParE speech the highest AUCs were 0.893, 0.861 and 0.923 respectively and achieved by a Resnet50 trained by transfer learning in the first two cases and an SVM using with bottleneck features in the third case.

Dataset	ID	Classifier	Best Feature Hyperparameters	Best Classifier Hyperparameters (Optimised inside nested cross-validation)	Performance				
					Spec	Sens	Acc	AUC	σ_{AUC}
Coswara normal speech	*S1*	*Resnet50 + TL*	Table 4	*Default Resnet50 (Table 1 in* Ref. [39]*)*	*90%*	*85%*	*87%*	*0.893*	3×10^{-3}
	S2	LSTM + TL	”	Table 4	88%	82%	85%	0.877	4×10^{-3}
	S3	CNN + TL	”	”	88%	81%	85%	0.875	4×10^{-3}
	S4	MLP + BNF	”	$\alpha_3 = 0.25$, $\alpha_7 = 60$	83%	85%	84%	0.871	8×10^{-3}
	S5	SVM + BNF	”	$\alpha_1 = 10^{-6}$, $\alpha_4 = 10^5$	83%	85%	84%	0.867	7×10^{-3}
	S6	KNN + BNF	”	$\alpha_5 = 50$, $\alpha_6 = 10$	80%	85%	83%	0.868	6×10^{-3}
	S7	LR + BNF	”	$\alpha_1 = 10^2$, $\alpha_2 = 0.6$, $\alpha_3 = 0.4$	79%	83%	81%	0.852	7×10^{-3}
	S8	Resnet50 + PF	$\mathcal{M} = 26, \mathcal{F} = 2^{10}, \mathcal{S} = 120$	Default Resnet50 (Table 1 in Ref. [39])	84%	80%	82%	0.864	51×10^{-3}
	S9	LSTM + PF	$\mathcal{M} = 26, \mathcal{F} = 2^{11}, \mathcal{S} = 150$	$\beta_3 = 0.1$, $\beta_4 = 32$, $\beta_5 = 128$, $\beta_6 = 0.001$, $\beta_7 = 256$, $\beta_8 = 170$	84%	78%	81%	0.844	51×10^{-3}
	S10	CNN + PF	$\mathcal{M} = 39, \mathcal{F} = 2^{10}, \mathcal{S} = 120$	$\beta_1 = 48$, $\beta_2 = 2$, $\beta_3 = 0.3$, $\beta_4 = 32$, $\beta_7 = 256$, $\beta_8 = 210$	82%	78%	80%	0.832	52×10^{-3}
Coswara fast speech	*S11*	*Resnet50 + TL*	Table 4	*Default Resnet50 (Table 1 in* Ref. [39]*)*	*84%*	*78%*	*81%*	*0.861*	2×10^{-3}
	S12	LSTM + TL	”	Table 4	83%	78%	81%	0.860	3×10^{-3}
	S13	CNN + TL	”	”	82%	76%	79%	0.851	3×10^{-3}
	S14	MLP + BNF	”	$\alpha_3 = 0.55$, $\alpha_7 = 70$	78%	83%	81%	0.858	7×10^{-3}
	S15	SVM + BNF	”	$\alpha_1 = 10^4$, $\alpha_4 = 10^{-2}$	78%	83%	81%	0.856	8×10^{-3}
	S16	KNN + BNF	”	$\alpha_5 = 60$, $\alpha_6 = 15$	77%	83%	81%	0.854	8×10^{-3}
	S17	LR + BNF	”	$\alpha_1 = 10^{-3}$, $\alpha_2 = 0.4$, $\alpha_3 = 0.6$	77%	82%	80%	0.841	11×10^{-3}
	S18	LSTM + PF	$\mathcal{M} = 26, \mathcal{F} = 2^{11}, \mathcal{S} = 120$	$\beta_3 = 0.1$, $\beta_4 = 32$, $\beta_5 = 128$, $\beta_6 = 0.001$, $\beta_7 = 256$, $\beta_8 = 170$	84%	80%	82%	0.856	47×10^{-3}
	S19	Resnet50 + PF	$\mathcal{M} = 39, \mathcal{F} = 2^{10}, \mathcal{S} = 150$	Default Resnet50 (Table 1 in Ref. [39])	82%	78%	80%	0.822	45×10^{-3}
	S20	CNN + PF	$\mathcal{M} = 52, \mathcal{F} = 2^{10}, \mathcal{S} = 100$	$\beta_1 = 48$, $\beta_2 = 2$, $\beta_3 = 0.3$, $\beta_4 = 32$, $\beta_7 = 256$, $\beta_8 = 210$	79%	77%	78%	0.810	41×10^{-3}
ComParE	S21	Resnet50 + TL	Table 4	Default Resnet50 (Table 1 in Ref. [39])	84%	90%	87%	0.914	4×10^{-3}
	S22	LSTM + TL	”	Table 4	82%	88%	85%	0.897	5×10^{-3}
	S23	CNN + TL	”	”	80%	88%	84%	0.892	5×10^{-3}
	S24	*SVM + BNF*	”	$\alpha_1 = 10^{-1}$, $\alpha_4 = 10^3$	*84%*	*88%*	*86%*	*0.923*	4×10^{-3}
	S25	MLP + BNF	”	$\alpha_3 = 0.3$, $\alpha_7 = 60$	80%	88%	84%	0.905	6×10^{-3}
	S26	KNN + BNF	”	$\alpha_5 = 20$, $\alpha_6 = 15$	80%	86%	83%	0.891	7×10^{-3}
	S27	LR + BNF	”	$\alpha_1 = 10^2$, $\alpha_2 = 0.45$, $\alpha_3 = 0.7$	81%	85%	83%	0.890	7×10^{-3}
	S28	MLP + PF + SFS	$\mathcal{M} = 26, \mathcal{F} = 2^{11}, \mathcal{S} = 150$	$\alpha_3 = 0.35$, $\alpha_7 = 70$	82%	88%	85%	0.912	11×10^{-3}
	S29	MLP + PF	$\mathcal{M} = 26, \mathcal{F} = 2^{11}, \mathcal{S} = 150$	$\alpha_3 = 0.35$, $\alpha_7 = 70$	81%	85%	83%	0.893	14×10^{-3}
	S30	KNN + PF	$\mathcal{B} = 100, \mathcal{F} = 2^{10}, \mathcal{S} = 120$	$\alpha_5 = 70$, $\alpha_6 = 15$	80%	84%	82%	0.847	16×10^{-3}
	S31	SVM + PF	$\mathcal{B} = 80, \mathcal{F} = 2^{11}, \mathcal{S} = 120$	$\alpha_1 = 10^{-2}$, $\alpha_4 = 10^{-3}$	79%	81%	80%	0.836	15×10^{-3}
	S32	LR + PF	$\mathcal{B} = 60, \mathcal{F} = 2^{10}, \mathcal{S} = 100$	$\alpha_1 = 10^4$, $\alpha_2 = 0.35$, $\alpha_3 = 0.65$	69%	72%	71%	0.776	18×10^{-3}

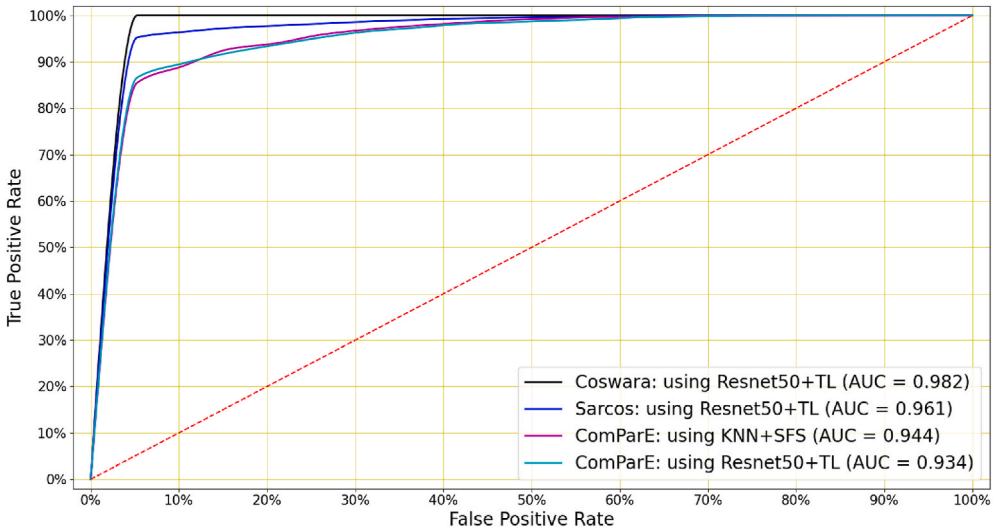

Fig. 5. COVID-19 cough classification: A Resnet50 classifier with transfer learning achieved the highest AUC in classifying COVID-19 coughs for the Coswara and Sarcos datasets (0.982 and 0.961 respectively). For the ComParE dataset, AUCs of 0.944 and 0.934 were achieved by a KNN classifier using 12 features identified by SFS and by a Resnet50 classifier trained by transfer learning respectively.

in all these cases is the substantial decrease in the standard deviation of the AUC (σ_{AUC}) observed during cross-validation when implementing transfer learning. This indicates that pre-training leads to classifiers with more consistent performance on the unseen test data.

The Sarcos dataset is much smaller than the Coswara dataset and too small to train a deep classifier directly. For this reason, it was used only as an independent validation dataset for classifiers trained on the Coswara data in our previous work [20]. It can however be used to fine-tune pre-trained classifiers during transfer learning, and the resulting performance is reflected by systems C11 to C17 in Table 6. Previously an AUC of 0.938 (system C20) was achieved when using Sarcos as an independent validation data and applying sequential forward selection (SFS) [20]. Here, we find that transfer learning applied to the Resnet50 model results in an AUC of 0.961 (system C11) and a lower standard deviation ($\sigma_{AUC} = 0.003$). As an additional experiment, we apply the Resnet50 classifier trained by transfer learning using the Coswara data to the Sarcos data, thus again using the latter exclusively as an independent validation set. The resulting performance is indicated by system C18, while the previous baselines are repeated as systems C19 and C20 [20]. System C18 achieves an AUC of 0.954, which is only slightly below the 0.961, achieved by system C11 where the pre-trained model used the Sarcos data for fine-tuning, and slightly higher than the AUC of 0.938 achieved by system C20 which is the baseline LSTM trained on Coswara without transfer learning but employing SFS [45]. This supports our earlier observation that transfer learning appears to lead to more robust classifiers that can generalise to other datasets. Due to the extreme computational load, we have not yet been able to evaluate SFS within the transfer learning framework.

For the ComParE dataset, we have included shallow classifiers trained directly on the primary input features (KNN + PF, MLP + PF, SVM + PF and LR + PF). These are the baseline systems C29 to C32 in Table 6. The best-performing shallow classifier is C29, where a KNN used 60 linearly spaced filterbank log energies as features. System C28 is the result of applying SFS to system C29. In this case, SFS identifies the top 12 features based on the development sets used during nested cross-validation, and results in the best-performing shallow system with an

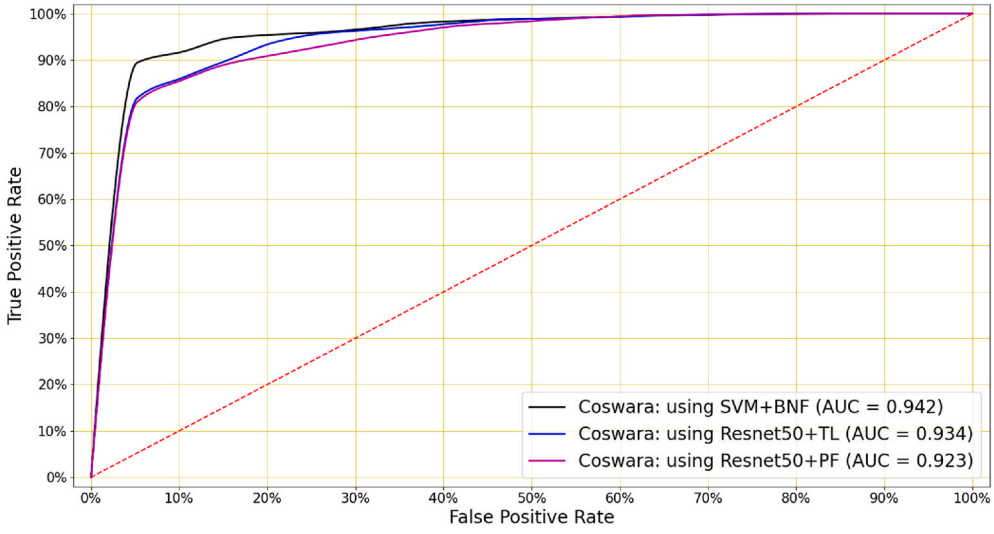

Fig. 6. COVID-19 breath classification: An SVM classifier using bottleneck features (BNF) achieved the highest AUC of 0.942 when classifying COVID-19 breath. The Resnet50 with and without the transfer learning has achieved AUCs of 0.934 and 0.923 respectively, with higher σ_{AUC} for the latter (Table 7).

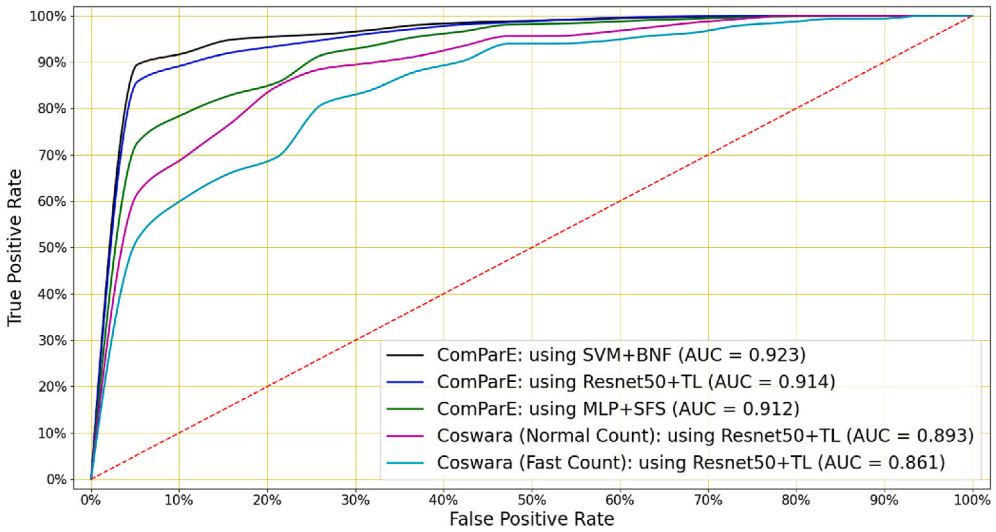

Fig. 7. COVID-19 speech classification: An SVM classifier using bottleneck features (BNF) achieved the highest AUC of 0.923 when classifying COVID-19 speech in ComParE dataset. A Resnet50 trained by transfer learning achieves a slightly lower AUC of 0.914. Speech (normal and fast) in the Coswara dataset can be used to classify COVID-19 with AUCs of 0.893 and 0.861 respectively using a Resnet50 trained by transfer learning.

AUC of 0.944. This represents a substantial improvement over the AUC of 0.855 achieved by the same system without SFS (system C29). Systems C21 to C27 in Table 6 are obtained by transfer learning using the ComParE dataset. These show improved performance over the shallow classifiers without SFS. In particular, after transfer learning, the Resnet50 achieves almost the same AUC as the best ComParE system (system C28) with a lower σ_{AUC}.

When considering the performance of the shallow classifiers trained on the bottleneck features across all three datasets in Table 6, we see that a consistent improvement over the use of primary features with the same classifiers is observed. The ROC curves for the best-performing COVID-19 cough classifiers are shown in Fig. 5.

7.2. Breath

Table 7 demonstrates that COVID-19 classification is also possible on the basis of breath signals. The baseline systems B8, B9 and B10 are trained directly on the primary features, without pre-training. By comparing these baselines with B1, B2 and B3, we see that transfer learning leads to a small improvement in AUC for all three deep architectures. Furthermore, systems B4 to B7 show that comparable performance can be achieved by shallow classifiers using the bottleneck features. The best overall performance (AUC = 0.942) was achieved by an SVM classifier trained on the bottleneck features (system B4). However, the Resnet50 trained by transfer learning (system B1) performed almost equally well (AUC = 0.934). The ROC curves for the best-performing COVID-19 breath classifiers are shown in Fig. 6. As it was observed for coughs, the standard deviation of the AUC (σ_{AUC}) is consistently lower for the pre-trained networks.

7.3. Speech

Although not as informative as cough or breath audio, COVID-19 classification can also be achieved on the basis of speech audio recordings. For Coswara, the best classification performance (AUC = 0.893) was achieved by a Resnet50 after applying transfer learning (system S1). For the ComParE data, the top performer (AUC = 0.923) was an SVM trained on the bottleneck features (system S24). However, the Resnet50 trained by transfer learning performed almost equally well, with an AUC of 0.914 (system S21). Furthermore, while good performance was also achieved when using the deep architectures without applying the transfer learning process (systems S8–S10, S18–S20 and S28–S32), this again was at the cost of a substantially higher standard deviation σ_{AUC}. Finally, for the Coswara data, performance was generally better when speech was uttered at a normal pace rather than a fast pace. The ROC curves for the best-performing COVID-19 speech classifiers are shown in Fig. 7.

8. Discussion

Previous studies have shown that it is possible to distinguish between the coughing sounds made by COVID-19 positive and COVID-19 negative subjects by means of automatic classification and machine learning. However, the fairly small size of datasets with COVID-19 labels limits the effectiveness of these techniques. The results of the experiments we have presented in this study show that larger datasets of other vocal and respiratory audio that do not include COVID-19 labels can be leveraged to improve classification performance by applying transfer learning [46]. Specifically, we have shown that the accuracy of COVID-19 classification based on coughs can be improved by transfer learning for two datasets (Coswara and Sarcos) while almost optimal performance is achieved on a third dataset (ComParE). A similar trend is seen when performing COVID-19 classification based on breath and speech audio. However, these two types of audio appear to contain less distinguishing information, since the achieved classification performance is a little lower than it is for cough. Our best cough classification system has an area under the ROC curve (AUC) of 0.982, despite being trained on what remains a fairly small COVID-19 dataset with 1171 participants (92 COVID-19 positive and 1079 negative). Other research reports a similar AUC but using a much larger dataset with 8380 participants (2339 positive and 6041 negative) [47]. While our experiments also show that shallow classifiers, when used in conjunction with feature selection, can in some cases match or surpass the performance of the deeper architectures; a pre-trained Resnet50 architecture provides consistent optimal or near-optimal performance across all three types of audio signals and datasets. Due to the very high computational cost involved,

we have not yet applied such feature selection to the deep architectures themselves, and this remains part of our ongoing work.

Another important observation that we can make for all three types of audio signals is that transfer learning strongly reduces the variance in the AUC (σ_{AUC}) exhibited by the deep classifiers during cross-validation (Tables 6–8). This suggests that transfer learning leads to more consistent classifiers that are less prone to over-fitting and better able to generalise to unseen test data. This is important because robustness to variable testing conditions is essential in implementing COVID-19 classification as a method of screening.

An informal listening assessment of the Coswara and the ComParE data indicates that the former has greater variance and more noise than the latter. Our experimental results presented in Tables 6–8 found that, for speech classification on noisy data, fine-tuning a pre-trained networks demonstrates better performance, while for cleaner data, extracting bottleneck features and then applying a shallow classifier exhibits better performance. It is interesting to note that MFCCs are always the features of choice for this noisier dataset, while the log energies of linear filters are often preferred for the less noisy data. Although all other classifiers have shown the best performance when using these log-filterbank energy features, MLP classifiers performed best when using MFCCs and were best at classifying COVID-19 speech. A similar conclusion was drawn in Ref. [24], where coughs were recorded in a controlled environment with little environmental noise. A larger number of frames in the feature matrix also generally leads to better performance as it allows the classifier to find more detailed temporal patterns in the audio signal.

Finally, we note that, for the shallow classifiers, hyperparameter optimisation selected a higher number of MFCCs and also a more densely populated filterbank than what is required to match the resolution of the human auditory system. This agrees with an observation already made in our previous work that the information used by the classifiers to detect COVID-19 signature is at least to some extent not perceivable by the human ear.

9. Conclusions

In this study, we have demonstrated that transfer learning can be used to improve the performance and robustness of the DNN classifiers for COVID-19 detection in vocal audio such as cough, breath and speech. We have used a 10.29 h audio data corpus, which does not have any COVID-19 labels, to pre-train a CNN, an LSTM and a Resnet50. This data contains four classes: cough, sneeze, speech and noise. In addition, we have used the same architectures to extract bottleneck features by removing the final layers from the pre-trained models. Three smaller datasets containing cough, breath and speech audio with COVID-19 labels were then used to fine-tune the pre-trained COVID-19 audio classifiers using nested *k*-fold cross-validation.

Our results show that a pre-trained Resnet50 classifier that is either fine-tuned or used as a bottleneck extractor delivers optimal or near-optimal performance across all datasets and all three audio classes. The results show that transfer learning using the larger dataset without COVID-19 labels led not only to improved performance, but also to a much smaller standard deviation of the classifier AUC, indicating better generalisation to unseen test data. The use of bottleneck features, which are extracted by the pre-trained deep models and therefore also a way of incorporating out-of-domain data, also provided a reduction in this standard deviation and near-optimal performance. Furthermore, we see that cough audio carries the strongest COVID-19 signatures, followed by breath and speech. The best-performing COVID-19 classifier achieved an area under the ROC curve (AUC) of 0.982 for cough, followed by an AUC of 0.942 for breath and 0.923 for speech.

We conclude that successful classification is possible for all three classes of audio considered. However, deep transfer learning improves COVID-19 detection on the basis of cough, breath and speech signals, yielding automatic classifiers with higher accuracies and greater robustness. This is significant since such COVID-19 screening is inexpensive, easily deployable, non-contact and does not require medical expertise or laboratory facilities. Therefore it has the potential to decrease the load on the health care systems.

As a part of ongoing work, we are considering the application of feature selection in the deep architectures, the fusion of classifiers using various audio classes like cough, breath and speech, as well as the optimisation and adaptation necessary to allow deployment on a smartphone or similar mobile platform.

Declaration of competing interest

We confirm that there is no conflict of interest statement to be declared.

Acknowledgements

This research was supported by the South African Medical Research Council (SAMRC) through its Division of Research Capacity Development under the Research Capacity Development Initiative as well as the COVID-19 IMU EMC allocation from funding received from the South African National Treasury. Support was also received from the European Union through the EDCTP2 programme (TMA2017CDF-1885). The content and findings reported are the sole deduction, view and responsibility of the researcher and do not reflect the official position and sentiments of the SAMRC or the EDCTP.

We would also like to thank South African Centre for High Performance Computing (CHPC) for providing computational resources on their Lengau cluster to support this research, and gratefully acknowledge the support of Telkom South Africa.

We also especially thank Igor Miranda, Corwynne Leng, Renier Botha, Jordan Govendar and Rafeeq du Toit for their support in data collection and annotation.

References

[1] A. Carfì, R. Bernabei, F. Landi, et al., Persistent symptoms in patients after acute COVID-19, JAMA 324 (6) (2020) 603–605.

[2] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, et al., Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China, JAMA 323 (11) (2020) 1061–1069.

[3] J.J. Marini, L. Gattinoni, Management of COVID-19 respiratory distress, JAMA 323 (22) (2020) 2329–2330.

[4] D. Aguiar, J.A. Lobrinus, M. Schibler, T. Fracasso, C. Lardi, Inside the lungs of COVID-19 disease, Int. J. Leg. Med. 134 (2020) 1271–1274.

[5] D.R. Ziehr, J. Alladina, C.R. Petri, J.H. Maley, A. Moskowitz, B.D. Medoff, K. A. Hibbert, B.T. Thompson, C.C. Hardin, Respiratory pathophysiology of mechanically ventilated patients with COVID-19: a cohort study, Am. J. Respir. Crit. Care Med. 201 (12) (2020) 1560–1564.

[6] C. E. Davis, M. Schivo, N. J. Kenyon, A breath of fresh air–the potential for COVID-19 breath diagnostics, EBioMedicine 63.

[7] S. Grassin-Delyle, C. Roquencourt, P. Moine, G. Saffroy, S. Carn, N. Heming, J. Fleuriet, H. Salvator, E. Naline, L.-J. Couderc, et al., Metabolomics of exhaled breath in critically ill COVID-19 patients: a pilot study, EBioMedicine 63 (2021) 103154.

[8] D.M. Ruszkiewicz, D. Sanders, R. O'Brien, F. Hempel, M.J. Reed, A.C. Riepe, K. Bailie, E. Brodrick, K. Darnley, R. Ellerkmann, et al., Diagnosis of COVID-19 by analysis of breath with gas chromatography-ion mobility spectrometry-a feasibility study, EClinicalMedicine 29 (2020) 100609.

[9] S. Walvekar, D. Shinde, et al., Detection of COVID-19 from CT images using Resnet50, in: 2nd International Conference on Communication & Information Processing (ICCIP) 2020, 2020.

[10] H. Sotoudeh, M. Tabatabaei, B. Tasorian, K. Tavakol, E. Sotoudeh, A.L. Moini, Artificial intelligence empowers radiologists to differentiate pneumonia induced by COVID-19 versus influenza viruses, Acta Inf. Med. 28 (3) (2020) 190.
[11] M. Yildirim, A. Cinar, A deep learning based hybrid approach for COVID-19 disease detections, Trait. Du. Signal 37 (3) (2020) 461–468.
[12] T. Higenbottam, Chronic cough and the cough reflex in common lung diseases, Pulm. Pharmacol. Therapeut. 15 (3) (2002) 241–247.
[13] A. Chang, G. Redding, M. Everard, Chronic wet cough: protracted bronchitis, chronic suppurative lung disease and bronchiectasis, Pediatr. Pulmonol. 43 (6) (2008) 519–531.
[14] K.F. Chung, I.D. Pavord, Prevalence, pathogenesis, and causes of chronic cough, Lancet 371 (9621) (2008) 1364–1374.
[15] J. Knocikova, J. Korpas, M. Vrabec, M. Javorka, Wavelet analysis of voluntary cough sound in patients with respiratory diseases, J. Physiol. Pharmacol. 59 (Suppl 6) (2008) 331–340.
[16] 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, Inform. Med. Unlocked 20 (2020) 100378.
[17] J. Laguarta, F. Hueto, B. Subirana, COVID-19 artificial intelligence diagnosis using only cough recordings, IEEE Open J. Eng. Med. Biol. 1 (2020) 275–281, https://doi.org/10.1109/OJEMB.2020.3026928.
[18] 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, 2020, pp. 3474–3484.
[19] H. Coppock, A. Gaskell, P. Tzirakis, A. Baird, L. Jones, B. Schuller, End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study, BMJ Innovations 7 (2)..
[20] M. Pahar, M. Klopper, R. Warren, T. Niesler, COVID-19 cough classification using machine learning and global smartphone recordings, Comput. Biol. Med. 135 (2021) 104572, https://doi.org/10.1016/j.compbiomed.2021.104572.
[21] 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, Proc. Interspeech 2020 (2020) 4811–4815, https://doi.org/10.21437/Interspeech.2020-2768.
[22] B.W. Schuller, A. Batliner, C. Bergler, C. Mascolo, J. Han, I. Lefter, H. Kaya, S. Amiriparian, A. Baird, L. Stappen, S. Ottl, M. Gerczuk, P. Tzirakis, C. Brown, J. Chauhan, A. Grammenos, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, M. R. Leon, J.J. Zwerts, J. Treep, C. Kaandorp, The INTERSPEECH 2021 computational Paralinguistics challenge: COVID -19 Cough, COVID -19 Speech, escalation & primates, in: Proceedings INTERSPEECH 2021, 22nd Annual Conference of the International Speech Communication Association, ISCA, Brno, Czechia, 2021 to appear.
[23] M. Pahar, I. Miranda, A. Diacon, T. Niesler, Deep neural network based cough detection using bed-mounted accelerometer measurements, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP), 2021, pp. 8002–8006, https://doi.org/10.1109/ICASSP39728.2021.9414744.
[24] G. Botha, G. Theron, R. Warren, M. Klopper, K. Dheda, P. Van Helden, T. Niesler, Detection of tuberculosis by automatic cough sound analysis, Physiol. Meas. 39 (4) (2018), 045005.
[25] M. Pahar, M. Klopper, B. Reeve, R. Warren, G. Theron, T. Niesler, Automatic cough classification for tuberculosis screening in a real-world environment, Physiol. Meas. 42 (2021) 105014, https://doi.org/10.1088/1361-6579/ac2fb8.
[26] J.F. Gemmeke, D.P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R.C. Moore, M. Plakal, M. Ritter, Audio set: an ontology and human-labeled dataset for audio events, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2017, pp. 776–780.
[27] F. Font, G. Roma, X. Serra, Freesound technical demo, in: Proceedings of the 21st ACM International Conference on Multimedia, 2013, pp. 411–412.
[28] I.D. Miranda, A.H. Diacon, T.R. Niesler, A comparative study of features for acoustic cough detection using deep architectures, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2019, pp. 2601–2605.
[29] V. Panayotov, G. Chen, D. Povey, S. Khudanpur, Librispeech: an ASR corpus based on public domain audio books, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2015, pp. 5206–5210, https://doi.org/10.1109/ICASSP.2015.7178964.
[30] J. Van Hulse, T.M. Khoshgoftaar, A. Napolitano, Experimental perspectives on learning from imbalanced data, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 935–942.
[31] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, Prog. Artificial Intell. 5 (4) (2016) 221–232.
[32] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[33] 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, arXiv preprint arXiv: 2103.09148..
[34] R. Bachu, S. Kopparthi, B. Adapa, B.D. Barkana, Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy, in: Advanced Techniques in Computing Sciences and Software Engineering, Springer, 2010, pp. 279–282.
[35] M. Pahar, L.S. Smith, Coding and decoding speech using a biologically inspired coding system, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 3025–3032, https://doi.org/10.1109/SSCI47803.2020.9308328.
[36] H. Chatrzarrin, A. Arcelus, R. Goubran, F. Knoefel, Feature extraction for the differentiation of dry and wet cough sounds, in: IEEE International Symposium on Medical Measurements and Applications, IEEE, 2011, pp. 162–166.
[37] M.B. Alsabek, I. Shahin, A. Hassan, Studying the similarity of COVID-19 sounds based on correlation analysis of MFCC, in: 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), IEEE, 2020, pp. 1–5.
[38] S. Aydın, H.M. Saraoğlu, S. Kara, Log energy entropy-based EEG classification with multilayer neural networks in seizure, Ann. Biomed. Eng. 37 (12) (2009) 2626.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[40] A. Silnova, P. Matejka, O. Glembek, O. Plchot, O. Novotnỳ, F. Grezl, P. Schwarz, L. Burget, J. Cernockỳ, BUT/Phonexia Bottleneck Feature Extractor, Odyssey, 2018, pp. 283–287.
[41] Y. Song, I. McLoughLin, L. Dai, Deep bottleneck feature for image classification, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015, pp. 491–494.
[42] Q.B. Nguyen, J. Gehring, K. Kilgour, A. Waibel, Optimizing deep bottleneck feature extraction, in: The 2013 RIVF International Conference on Computing & Communication Technologies-Research, Innovation, and Vision for Future (RIVF), IEEE, 2013, pp. 152–156.
[43] S. Liu, Leave-*p*-Out cross-validation test for uncertain verhulst-pearl model with imprecise observations, IEEE Access 7 (2019) 131705–131709.
[44] T. Fawcett, An introduction to ROC analysis, Pattern Recogn. Lett. 27 (8) (2006) 861–874.
[45] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, 1982.
[46] D. Grant, I. McLane, J. West, Rapid and scalable COVID-19 screening using speech, breath, and cough recordings, in: 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), IEEE, 2021, pp. 1–6, https://doi.org/10.1109/BHI50953.2021.9508482.
[47] J. Andreu-Perez, H. Pérez-Espinosa, E. Timonet, M. Kiani, M.I. Giron-Perez, A. B. Benitez-Trinidad, D. Jarchi, A. Rosales, N. Gkatzoulis, O.F. Reyes-Galaviz, et al., A generic deep learning based cough analysis system from clinically validated samples for point-of-need Covid-19 test and severity levels, IEEE Trans. Serv. Comput. (2021), https://doi.org/10.1109/TSC.2021.3061402, 1–1.

Madhurananda Pahar received his BSc in Mathematics from the University of Calcutta, India, and his MSc in Computing for Financial Markets followed by his PhD in Computational Neuroscience from the University of Stirling, Scotland. Currently, he is working as a post-doctoral fellow at Stellenbosch University, South Africa. His research interests are in machine learning and signal processing for audio signals and smart sensors in bio-medicine. Currently, he is involved in the application of deep learning to the detection and classification of tuberculosis and COVID-19 coughs in real-world environments as well as the monitoring of patient behaviour using smart sensors such as an accelerometer.

Marisa Klopper is a researcher at the Division of Molecular Biology and Human Genetics of Stellenbosch University, South Africa. She holds a PhD in Molecular Biology from Stellenbosch University and her research interest is in TB and drug-resistant TB diagnosis, epidemiology and physiology. She has been involved in cough classification for the last 6 years, with application to TB and more recently COVID-19.

Robin Warren is the Unit Director of the South African Medical Research Council's Centre for Tuberculosis Research and Distinguished Professor at Stellenbosch University. He has a B2 rating by the National Research Council (NRF) and is a core member of the DSI-NRF Centre of Excellence for Biomedical Tuberculosis Research and head the TB Genomics research thrust. He has published over 320 papers in the field of TB and have an average H-index (Scopus, Web of Science and Google Scholar) of 65.

Thomas Niesler obtained the B.Eng (1991) and M.Eng (1993) degrees in Electronic Engineering from the University of Stellenbosch, South Africa and a Ph.D. from the University of Cambridge, England, in 1998. He joined the Department of Engineering, University of Cambridge, as a lecturer in 1998 and subsequently the Department of Electrical and Electronic Engineering, University of Stellenbosch, in 2000, where he has been Professor since 2012. His research interests lie in the areas of signal processing, pattern recognition and machine learning.