# SUPERVISED AND SELF-SUPERVISED PRETRAINING BASED COVID-19 DETECTION USING ACOUSTIC BREATHING/COUGH/SPEECH SIGNALS

*Xing-Yu Chen*[1*], *Qiu-Shi Zhu*[1*], *Jie Zhang*[1,2], *Li-Rong Dai*[1]

[1]NEL-SLIP, University of Science and Technology of China (USTC), Hefei, China
[2]State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China

## ABSTRACT

A rapid-accurate detection method for COVID-19 is rather important for avoiding its pandemic. In this work, we propose a bi-directional long short-term memory (BiLSTM) network based COVID-19 detection method using breath/speech/cough signals. Three kinds of acoustic signals are taken to train the network and individual models for three tasks are built, respectively, whose parameters are averaged to obtain an average model, which is then used as the initialization for the BiLSTM model training of each task. It is shown that such an initialization method can significantly improve the detection performance on three tasks. This is called supervised pre-training based detection. Besides, we utilize an existing pre-trained wav2vec2.0 model and pre-train it using the DiCOVA dataset, which is utilized to extract a high-level representation as the model input to replace conventional mel-frequency cepstral coefficients (MFCC) features. This is called self-supervised pre-training based detection. To reduce the information redundancy contained in the recorded sounds, silent segment removal, amplitude normalization and time-frequency masking are also considered. The proposed detection model is evaluated on the DiCOVA dataset and results show that our method achieves an area under curve (AUC) score of 88.44% on blind test in the fusion track. It is shown that using high-level features together with MFCC features is helpful for diagnosing accuracy.

***Index Terms***— COVID-19, binary classification, supervised pre-training, self-supervised pre-training, respiratory diagnosis.

## 1. INTRODUCTION

Since the outbreak of COVID-19, it quickly becomes pandemic all over the world, even at every conner. This unknown disease has brought a serious influence to all countries ranging from global health, economy, education, trade, cultural exchange, etc. Due to the fact that disease diagnosis is an important step for controlling the transmission, it is worthy studying how to detect the COVID-19 rapidly and efficiently. Nucleic acid test is a common and traditional COVID-19 detection method, but patients have to be tested in specific place like hospitals, and the test is comparatively expensive and time-consuming. With the development of artificial intelligence (AI) techniques in recent years, machine learning based methods

have been frequently used in disease detection. Compared to conventional medical diagnosis approaches, the AI-based methods have an obvious superiority, as the detection can be performed fully online and the time and economic expenses are much lower. The obtained diagnostic result can be used as supplementary information for doctors to make a more accurate clinical discrimination.

It thus deserves to investigate *whether AI techniques can also be utilized for diagnosing COVID-19, where only medical images [1] or sounds [2] of patients will be recorded.* Recently, several AI-based techniques have been developed for the detection of COVID-19. For example, a ResNet50 based method was proposed in [3] based on the use of computed tomography (CT) images, which achieves an accuracy of 96.23%. Compared to the CT-based method, where the CT images are collected offline, it is more appealing if we can use sound signals (e.g., cough, speech, breath) to perform detection, as the latter can be recorded remotely and avoid the people gathering.

In principal, the detection of COVID-19 is a binary classification task (i.e., positive or negative). In order to make use of sound data for classification, feature extraction is required as a preprocessing step. Mel-frequency cepstral coefficients (MFCC) and mel-frequency spectrogram that can reflect the nonlinear perceptive characteristics of human hearings to frequency are commonly used as features for sound activity analysis [4, 5]. In addition, zero crossing rate (ZCR), kurtosis, log energy, spectral centroid, roll-off frequency can also be used [6, 7]. In [8], it was shown that the positive testee of COVID-19 have different acoustical parameters compared with the negative. In [9], Coppock et al extracted the spectrogram feature and used a ResNet based CNN to detect cough sounds, which achieves an area under curve (AUC) of 0.846. In [10], the support vector machine (SVM) was employed for the detection of COVID-19 in combination of voice signals and symptoms. In addition, other classifiers like long short-term memory (LSTM) [6], k-nearest neighbour (kNN) [11], random forest [12] and light gradient boosting machine (LightGBM) [13] can also be utilized in line with classical machine learning approaches.

As the amount of DiCOVA data is limited, we mainly aim at exploring whether additional pre-training can improve the detection performance in this work. For this, we explore a supervised pre-training method, a self-supervised pre-training method and model enemble method. Specifically, for supervised pre-training, we utilize breath, cough and speech sounds to train models, respectively, then we can obtain models for these three different tasks. The model parameters are averaged to obtain an average model. We treat the average model as an initialization model, which is taken to initialize the model on different tasks. Experimental results show that this initialization method can significantly improve the performance on three tasks, which surpasses the official baseline results. For the self-supervised pre-training method, we pre-train the public pre-trained wav2vec2.0 model [14] on the DiCOVA dataset [15] and use the

This paper presents a detailed introduction for the COVID-19 detection system from the team NELSLIP-USTC, which obtains the first place in tracks 3&4 (i.e., speech & fusion) of DiCOVA-ICASSP 2022 contest and the second place in track 1 (i.e., breathing).
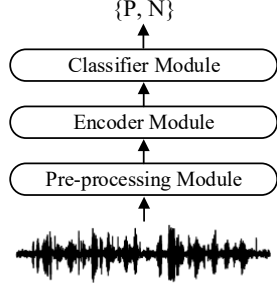
**Fig. 1**: An illustration of the proposed DiCOVA diagnosing model.



**Fig. 2**: The supervised pre-training process.

pre-trained model as a high-level feature extractor. We utilize the pre-trained model to extract high-level features, which are input into the diagnosing model to replace the classic MFCC feature. Experimental results reveal that the developed self-supervised pre-training method also outperforms the official baseline, but it is worse than the supervised pre-training method. It is worth mentioning that ensembling the two methods can obtain a better result in the same task, which shows that ensemble different levels of information can improve the performance. More importantly, by fusing the output probabilities, we obtain the best performance in the fusion task.

## 2. METHODOLOGY

### 2.1. Model structure

The diagram of the proposed classification model that we utilize in each subtask is shown in Fig. 1, which mainly consists of a pre-processing module, an encoder module and a classifier module. In the encoder module, two bi-directional LSTM (BiLSTM) [16] layers was utilized as the encoder. Each BiLSTM layer contains 128 hidden units and the dropout rate is set to be 0.1. Two fully connected feed-forward layers are utilized in the classifier module, which consists of two linear transformations with a ReLU activation in between. As the detection of COVID-19 is a 0-1 classification problem, we use the binary classification loss function for training.

### 2.2. Supervised pre-training method

For the supervised pre-training, no extra data are utilized except the officially provided DiCOVA data. The total DiCOVA dataset consist of breathing, cough and speech labeled data. We obtain a breathing/cough/speech model using labeled breathing/cough/speech data through supervised pre-training, respectively. Specifically, the breathing model $Model_{breath}$ can be obtained using the labeled breathing data $X_{breath}$ through supervised pre-training, and $Model_{cough}$ and $Model_{speech}$ can be built similarly as

$$Model_{breath} = BiLSTM(X_{breath}), \quad (1)$$

$$Model_{cough} = BiLSTM(X_{cough}), \quad (2)$$

$$Model_{speech} = BiLSTM(X_{speech}). \quad (3)$$

Then, we average the BiLSTM weights of the three models as the average model as

$$Model_{ave} = ave(Model_{breath}, Model_{cough}, Model_{speech}). \quad (4)$$

For each task, the average model is considered as an initialization model, and we re-train the breathing/cough/speech model in the corresponding task. The pre-training process is shown in Fig. 2. As we initialize the model using the average model, it will be easier for the encoder to find the optimal solution as well as for the classifier.
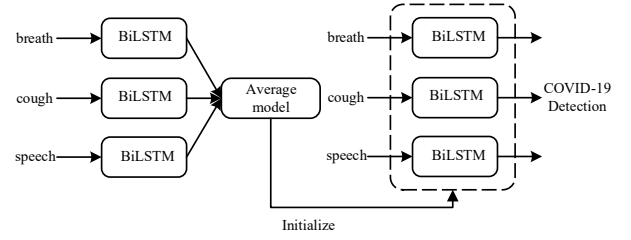
### 2.3. Self-supervised pre-training method

As the amount of COVID-19 audio data in each sub-task is still limited, the traditional MFCC feature might be not sufficiently representative for classification tasks. Hence, we propose to utilize a high-level feature to replace the traditional MFCC feature. It is worth mentioning that there are many effective methods that were proposed to solve these low-resource tasks [17–21]. In the speech domain, the wav2vec2.0 [14] model is a representative self-supervised pre-training framework for learning speech representation. The public self-supervised pre-trained wav2vec2.0 [14] model is utilized in the stage of self-supervised pre-training. We pre-train the public pre-trained wav2vec2.0 model on the DiCOVA data and use the pre-trained model as a high-level feature extractor. For example, we use DiCOVA breathing data to pre-train the wav2vec2.0 model. After finishing pre-training, we utilize the pre-trained model to extract high-level breathing feature, which is then input into the diagnosing model to replace the MFCC feature.

In order to guide the reader, we will briefly review the wav2vec 2.0 model in this section. The structure of wav2vec 2.0 model is shown in Fig. 3, including a CNN-based feature encoder $f : X \mapsto Z$ and a transformer encoder $g : Z \mapsto C$. In detail, the input raw waveform $X$ is downsampled to the latent speech representation $Z$ by the feature encoder. The transformer encoder then models the contextualized representation $C$ and extracts a high-level feature from the input $Z$. A quantization module $Z \mapsto Q$ discretizes the output of the feature encoder to $q_t$ as targets in the contrastive objective.

The quantization module first maps the latent speech representation $Z$ to logits $l \in \mathbb{R}^{G \times V}$, given $G$ codebooks with $V$ entries. The Gumbel softmax operation [22] is then used to select discrete codebook entries in a fully differentiable way. For a given frame $Z_t$, we can therefore select one entry from each codebook and concatenate the resulting vectors $e_1, ..., e_G$ and apply a linear transformation to obtain $q_t$. The weighted loss function is thus given by

$$L = L_m + \alpha L_d + \beta L_f, \quad (5)$$

where

$$L_m = -\log \frac{\exp(sim(c_t, q_t)/\kappa)}{\sum_{\tilde{q} \sim Q_t} \exp(sim(c_t, \tilde{q})/\kappa)}, \quad (6)$$

$$L_d = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \overline{p}_{g,v} \log \overline{p}_{g,v}, \quad (7)$$

$$\overline{p}_{g,v} = \frac{\exp(\overline{l}_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} \exp(\overline{l}_{g,k} + n_k)/\tau}. \quad (8)$$

It is clear that the total loss function is the weighted summation of three terms $L_m$, $L_d$ and $L_f$ parameterized by $\alpha$ and $\beta$. In (6), $L_m$ is the contrastive loss, which enables the model distinguishable between the true quantized latent speech representation $q_t$ and a set
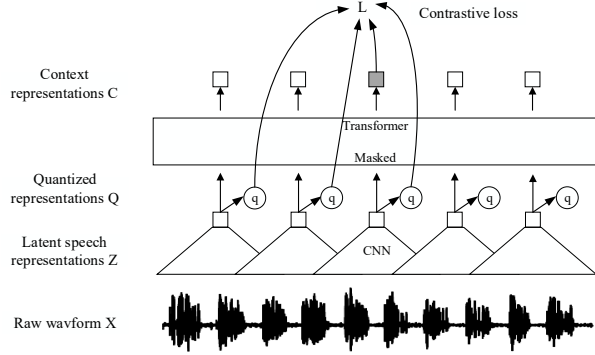
562

**Fig. 3**: An illustration of the involved wav2vec 2.0 model.

of $K + 1$ quantized candidate representations $\tilde{\mathbf{q}} \in \mathbf{Q}_t$. The quantized candidate representation $\tilde{\mathbf{q}}$ contains $\mathbf{q}_t$ and $K$ distractors, and the latter are uniformly sampled from other masked time steps of the same utterance. In (5), the diversity loss $L_d$ aims to increase the use of quantized codebook representation, and $L_f$ is an $\ell_2$ penalty over the outputs of the feature encoder. In (6), sim stands for the cosine similarity between two vectors and $\kappa$ is a temperature. In (7), $\overline{p}_{g,v}$ represents the probability of choosing the $v$-th codebook entry for group $g$ across a batch of utterances, where $\tau$ is a temperature. In (8), $\overline{l}_{g,v}$ stands for the average logits $\mathbf{l}$ across utterances in a batch. More details on the wav2vec 2.0 model can be found in [14].

### 2.4. Model ensemble method

For the supervised pre-training method, the model uses MFCC feature as input, and for the self-supervised pre-training method, the model uses high-level features extracted by the pre-trained wav2vec2.0 model. As local features and high-level features contain different information, in order to combine the superiority of different features, we use model ensemble to improve the detection performance. Specifically, for the same task we utilize the proposed supervised and self-supervised methods to train two models and obtain score$_{\text{sup}}$ and score$_{\text{self}-\text{sup}}$ on the test set, respectively. Then, we can ensemble the scores obtained by the two models, which is given by

$$\text{score}_{\text{emsemble}} = \mu * \text{score}_{\text{sup}} + (1 - \mu) * \text{score}_{\text{self}-\text{sup}}, \quad (9)$$

where $\mu$ is chosen by the user. The weighted summation score$_{\text{emsemble}}$ is taken as the detection probability.

For the track-4 fusion task, we utilize the best breathing, cough and speech models to obtain an appropriate fusion model as

$$\text{score}_{\text{fusion}} = \theta * \text{score}_{\text{bre}} + \gamma * \text{score}_{\text{cou}} + \phi * \text{score}_{\text{spe}}, \quad (10)$$

where the parameter $\theta, \gamma, \phi$ need to satisfy $\theta + \gamma + \phi = 1$. In principal, the breathing, cough and speech signals have a different importance in terms of the COVID-19 information, $\theta, \gamma, \phi$ should be chosen in line with the signal qualities.

## 3. PERFORMANCE EVALUATION

### 3.1. Datasets

The proposed model is evaluated on the DiCOVA-ICASSP 2022 challenge dataset [15], which is derived from the crowd-sourced Coswara dataset [23] and collected from volunteers with different

health conditions. Volunteers were advised to record their sound in a quiet environment using a web-application. The audio streams are sampled at a sampling frequency of 44.1 kHz and in an FLAC format. The durations of the sound recordings range from about 1 second up to 29 seconds. The dataset consists of a total of 965 samples including 793 negative samples and 172 positive samples, where each sample includes cough, speech and breath sounds. The blind test set provided by the organizer includes 471 samples without labels. The training set and test set are both provided by the organizer, and there is no duplicate data between them.

### 3.2. Model configuration

In the pre-processing module, the amplitude of the raw waveform data is normalized between -1 to 1 through a normalization operator. As there are many silent segments in the speech/cough/breath sound signals, the speech activity detector (SAD) is applied to cut off these silent segments. The sound data is downsampled to 16 kHz. Forty dimensional MFCC and delta-delta coefficients are extracted with a window of 25 msec audio samples and a hop of 10 msec. Due to the small size of the training data, we use SpecAugment [24] time-frequency mask to augment the data, as it was shown that data augmentation is effective to improve the performance, particularly in low-resource cases. The time mask length is 20 and the frequency masking length is 50. In the encoder module, two BiLSTM [16] layers are utilized as the encoder. The BiLSTM layer dimension is 128 and the dropout rate is set to be 0.1. Two fully connected feed-forward layers are utilized in the classifier module, which consists of two linear transformations with a ReLU activation in between. The dimension of feed-forward layers is 256, which is finally mapped to 1-dimension for binary classification. As the detection of COVID-19 is a 0-1 classification problem, we use the binary classification loss function for training. We follow the baseline system given by the organizer to train the model for 5-fold cross-validation in each task and then decode on the official blind test set to obtain the final test results, separately. The CNN model is also utilized as another baseline system in experiments.

For the supervised pre-training, different seeds are set to train different models. The pre-training model is trained using 2 RTX3090ti-24G GPUs with 50 epochs. For the self-supervised pre-training, the wav2vec2.0 pre-training model is implemented using the fairseq toolkit [25]. The feature encoder contains seven blocks, where each block has 512 temporal convolution channels with strides (5, 2, 2, 2, 2, 2, 2) and kernel widths (10, 3, 3, 3, 3, 3, 2, 2). Thus, the interval between two sequential samples in the feature encoder output $Z$ is around 20 ms and the receptive audio field is 25 ms. The models contain 12 transformer encoder blocks with a dimension of 512, a feed forward module with a dimension of 2048 and 8 attention heads. The pre-training process is optimized with Adam [26]. We set $G = 2$ and $V = 320$ for the quantization module and each entry with a size of 128. The temperature $\kappa$ is set to be 0.1 and $\tau$ is annealed from 2 to 0.5 by a factor of 0.999995 over iterations. For the contrastive loss, $\alpha$ and $\beta$ are set to be 0.1 and 10, respectively. We use $K = 100$ distractors and the total number of pre-training epochs are 200. After self-supervised pre-training, the pre-trained model parameters are frozen. Then, the pre-trained model is utilized as a high-level feature extractor, which generates high-level features from raw waveforms.

### 3.3. Results

In experiments, we use 5-fold cross-validation to evaluate our model. For comparison, we also test CNN, LSTM and official baseline

**Table 1**: The AUC score of different methods on test/validation sets.

| Method | Model | ROC-AUC score (%) | |
|---|---|---|---|
| | | **Test** | **Validation** |
| **Track-1 breath** | | | |
| Official baseline | - | 84.50 | 77.63 |
| Baseline 1 | BiLSTM | 84.17 | 76.26 |
| Baseline 2 | CNN | 84.10 | 75.84 |
| Supervised pre-train | BiLSTM | 86.41 | 80.05 |
| Self-supervised pre-train | BiLSTM | 86.22 | 79.04 |
| Model ensemble | BiLSTM | **86.72** | **80.05** |
| **Track-2 cough** | | | |
| Official baseline | - | 74.89 | 75.88 |
| Baseline 1 | BiLSTM | 75.04 | 76.18 |
| Baseline 2 | CNN | 73.70 | 76.06 |
| Supervised pre-train | BiLSTM | 76.05 | 78.92 |
| Self-supervised pre-train | BiLSTM | 75.55 | 78.56 |
| Model ensemble | BiLSTM | **76.36** | **78.92** |
| **Track-3 speech** | | | |
| Official baseline | - | 84.26 | 82.24 |
| Baseline 1 | BiLSTM | 83.68 | 82.15 |
| Baseline 2 | CNN | 83.38 | 81.96 |
| Supervised pre-train | BiLSTM | 85.02 | 81.00 |
| Self-supervised pre-train | BiLSTM | 84.35 | 80.74 |
| Model ensemble | BiLSTM | **85.21** | **81.30** |

**Table 2**: The AUC score of track-4 tasks on test and validation sets.

| Track-4 fusion | Fusion weight | ROC-AUC score (%) | |
|---|---|---|---|
| | | **Test** | **Validation** |
| Official baseline | - | 84.50 | 77.63 |
| Fusion1 | (1/3,1/3,1/3) | 87.01 | 82.56 |
| Fusion2 | (0.4,0.2,0.4) | **88.44** | **82.93** |
| Fusion3 | (0.5,0.1,0.4) | 88.44 | 82.71 |

model without pre-training. The official baseline comes from the website[1]. The obtained results are shown in Table 1 and Table 2. Experimental results show that the methods with pre-training achieve a higher AUC than the baseline. That is, pre-training can provide more information for classifiers, particularly in case the dataset is small-sized. Supervised pre-training models can achieve a great enhancement of performance compared with the baseline system. In breath/cough/speech task, supervised pre-training models increase the AUC from 84.50/74.89/84.26 to 86.41/76.05/85.02, respectively. Self-supervised pre-training methods obtain the AUC of 86.22/75.55/84.35 in the breath/cough/speech task. Despite achieving a smaller improvement in performance than supervised counterparts, self-supervised pre-training models still surpass the baseline in all tasks. Compared to the supervised and self-supervised methods, the proposed ensemble model can further improve the detection performance, which obtains the best performance for all tasks, and the corresponding AUCs are 86.72/76.36/85.21 in cough/speech/breath tasks, respectively. These results indicate that high-level feature

---

[1]https://competitions.codalab.org/competitions/34801#results
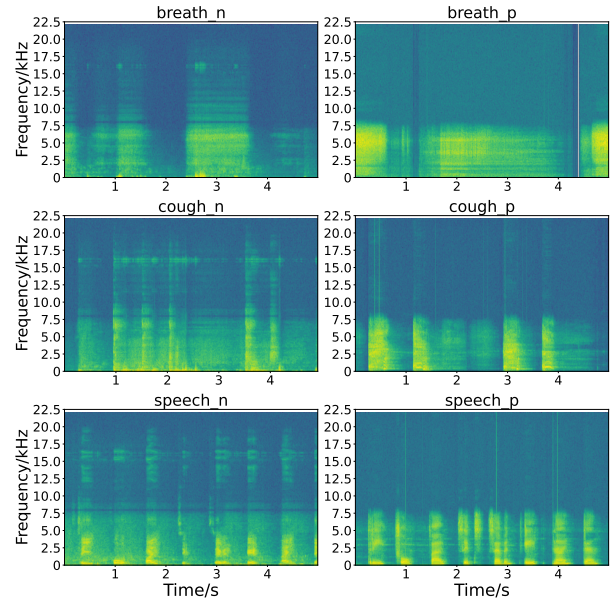


**Fig. 4**: The spectrograms of the positive (right) and negative (left) breathing/cough/speech audio samples.

pre-trained by wav2vec2.0 and low-level MFCC feature are complementary. For the fusion task, the final probability is calculated as a weighted summation over three best individual tasks. In case the cough/speech/breath weights are set to be 0.4/0.2/0.4, we achieve the highest AUC of 88.44 for the fusion task.

## 4. CONCLUSION AND DISCUSSION

In this work, we presented an ensemble model based on BiL-STM for diagnosing COVID-19 using acoustic signals. Due to the small size of the training data, we used supervised pre-training and self-supervised pre-training methods, and more importantly both achieved a better performance than the baseline. Using the ensemble model of supervised pre-training and self-supervised pre-training, the AUC score was further improved, showing that the high-level feature pre-trained by wav2vec2.0 and low-level MFCC feature are complementary. The proposed model was evaluated on the DiCOVA challenge dataset and achieved an AUC score of 88.44% in the blind test set for the fusion task, which reaches the first place in tracks 3&4 of the DiCOVA-ICASSP 2022 contest.

As it was shown by experiments that in case of using the same classifier the high-level representation obtained by the wav2vec 2.0 model achieves a better performance than the MFCC feature, it tells that feature extraction is a vital step for the COVID-19 detection. In order to more clearly see the difference in the features of positive and negative audio samples, we show the spectrograms of the positive and negative breathing/cough/speech audio samples in Fig. 4. It is clear that for the positive samples, the energy of breathing/cough/speech signals is almost concentrated on low-frequency bands, while for negative samples the energy is distributed over full-frequency bands, particularly for the negative cough and speech signals. It reveals that high-level features are more beneficial for diagnosing COVID-19 using acoustic signals. In the future, we will therefore investigate an effective combination of the wav2vec 2.0 based acoustic representation, spectrograms, MFCC as a more representative feature.

## 5. REFERENCES

[1] M. M. Islam, F. Karray, R. Alhajj, and J. Zeng, "A review on deep learning techniques for the diagnosis of novel coronavirus (COVID-19)," *IEEE Access*, vol. 9, pp. 30 551–30 572, 2021.

[2] G. Deshpande, A. Batliner, and B. W. Schuller, "AI-Based human audio processing for COVID-19: A comprehensive overview," *Pattern Recognition*, vol. 122, p. 108289, 2022.

[3] L. Nwosu, X. Li, L. Qian, S. Kim, and X. Dong, "Semi-supervised learning for COVID-19 image classification via ResNet," *EAI Endorsed Transactions on Bioengineering and Bioinformatics*, vol. 1, no. 3, 2021.

[4] V. Karas and B. W. Schuller, "Recognising covid-19 from coughing using ensembles of SVMs and LSTMs with hand-crafted and deep audio features," in *Proc. Interspeech*, 2021, pp. 911–915.

[5] K. V. S. Ritwik, S. B. Kalluri, and D. Vijayasenan, "COVID-19 detection from spectral features on the DiCOVA dataset," in *Proc. Interspeech*, 2021, pp. 936–940.

[6] M. Pahar, M. Klopper, R. Warren, and T. Niesler, "COVID-19 cough classification using machine learning and global smartphone recordings," *Computers in Biology and Medicine*, no. 2, p. 104572, 2021.

[7] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo, "Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data," in *Proc. of the ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, 2020, pp. 3474–3484.

[8] M. Asiaee, A. V. Azimi, S. S. Atashi, A. Keramatfar, and M. Nourbakhsh, "Voice quality evaluation in patients with COVID-19: An acoustic analysis," *Journal of Voice*, 2020.

[9] H. Coppock, A. Gaskell, P. Tzirakis, A. Baird, L. Jones, and B. Schuller, "End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study," *BMJ Innovations*, vol. 7, no. 2, pp. 356–362, 2021. [Online]. Available: https://innovations.bmj.com/content/7/2/356

[10] J. Han, C. Brown, J. Chauhan, A. Grammenos, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo, "Exploring automatic covid-19 diagnosis via voice and symptoms from crowdsourced data," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8328–8332.

[11] A. Anupam, N. J. Mohan, S. Sahoo, and S. Chakraborty, "Preliminary diagnosis of COVID-19 based on cough sounds using machine learning algorithms," in *IEEE Int. Conf. on Intelligent Computing and Control Systems*, 2021, pp. 1391–1397.

[12] I. Södergren, M. Nodeh, P. C. Chhipa, K. Nikolaidou, and G. Kovács, "Detecting COVID-19 from audio recording of coughs using Random Forests and Support Vector Machines," in *Proc. Interspeech*, 2021, pp. 916–920.

[13] M. Kamble, J. Gonzalez Lopez, T. Grau, J. Espín López, L. Cascioli, Y. Huang, A. Gomez-Alanis, J. Patino, R. Font, A. Peinado, A. Gomez, N. Evans, M. Zuluaga, and M. Todisco, "PANACEA cough sound-based diagnosis of COVID-19 for the DiCOVA 2021 challenge," in *Proc. Interspeech*, 2021, pp. 906–910.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[15] N. K. Sharma, S. R. Chetupalli, D. Bhattacharya, D. Dutta, P. Mote, and S. Ganapathy, "The Second DiCOVA Challenge: Dataset, task, and baseline system for COVID-19 diagnosis using acoustics," https://dicovachallenge.github.io/docs/Second_DiCOVA2_baselineSystem_doc.pdf.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[18] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM Tran. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2041–2053, 2019.

[19] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. Interspeech*, 2019, pp. 146–150.

[20] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "Wav2vec: unsupervised pre-training for speech recognition," in *Proc. Interspeech*, 2019, pp. 3465–3469.

[21] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[22] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[23] N. Sharma, P. Krishnan, R. Kumar, S. Ramoji, S. Chetupalli, R. Nirmala, P. Kumar Ghosh, and Ganapathy, "Coswara - A database of breathing, cough, and voice sounds for COVID-19 diagnosis," in *Proc. Interspeech*, 2020, pp. 4811–4815.

[24] D. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. Cubuk, and Q. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[25] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "Fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. Conf. the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 48–53.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.