Recognising Covid-19 from coughing using ensembles of SVMs and LSTMs with handcrafted and deep audio features

Vincent Karas, Björn W. Schuller

Angaben zur Veröffentlichung / Publication details:

Karas, Vincent, and Björn W. Schuller. 2021. "Recognising Covid-19 from coughing using ensembles of SVMs and LSTMs with handcrafted and deep audio features." In *Interspeech 2021, Brno, Czechia, 30 August - 3 September 2021*, edited by Hynek Heřmanský, Honza Černocký, Lukáš Burget, Lori Lamel, Odette Scharenborg, and Petr Motlicek, 911–15. Baixas: International Speech Communication Association (ISCA). https://doi.org/10.21437/interspeech.2021-1267.

INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Recognising Covid-19 from Coughing using Ensembles of SVMs and LSTMs with Handcrafted and Deep Audio Features

Vincent Karas[1], *Björn W. Schuller*[1,2]

[1]EIHW–Chair for Embedded Intelligence for Healthcare and Wellbeing, University of Augsburg, Germany
[2]GLAM–Group on Language, Audio and Music, Imperial College London, UK
vincent.karas@informatik.uni-augsburg.de

Abstract

As the Covid-19 pandemic continues, digital health solutions can provide valuable insights and assist in diagnosis and prevention. Since the disease affects the respiratory system, it is hypothesised that sound formation is changed, and thus, an infection can be automatically recognised through audio analysis. We present an ensemble learning approach used in our entry to Track 1 of the DiCOVA 2021 Challenge, which aims at binary classification of Covid-19 infection on a crowd-sourced dataset of 1 040 cough sounds. Our system is based on a combination of handcrafted features for paralinguistics with deep feature extraction from spectrograms using pre-trained CNNs. We extract features both at segment level and with a sliding window approach, and process them with SVMs and LSTMs, respectively. We then perform least-squares weighted late fusion of our classifiers. Our system surpasses the challenge baseline, with a ROC-AUC on the test set of 78.18 %.
Index Terms: COVID-19, acoustics, coughing, machine learning, respiratory diagnosis, healthcare

1. Introduction

Respiratory diseases can affect sound formation, with different patterns depending on the underlying pathology [1]. In addition to traditional medical diagnostics using e. g. , stethoscopes, researchers have developed digital health systems that analyse speech, breathing or coughing with machine learning methods. Applications include the detection of pertussis [2], tuberculosis [3], and early signs of cardiovasculary disease [4].

Currently, the Covid-19 pandemic has sparked much research interest into digital health [5], driven by the need to develop solutions that deliver results quickly and can be deployed at scale. The effect of Covid-19 on the lungs has been documented by thoracic scans [6], and early research has indicated that severity of illness, sleep quality, fatigue, and anxiety of patients can be derived from the voice [7]. Another study based on clinically validated data found that Covid-19 can be distinguished from other diseases by coughing [8]. In addition to data collection in clinical settings, crowdsourcing recordings from smartphones via a web app is a cost-effective way to assemble databases, and multiple such projects have already been launched [9], [10], [11], [12]. To compare various research efforts in a standardised setting, [13] introduced the Interspeech 2021 ComParE Challenge, and [14] introduced the DiCOVA 2021 challenge.

This paper presents an ensemble learning approach used in our entry into the DiCOVA 2021 challenge. We show that the ensemble boosts the performance of the individual classifiers, surpassing the challenge baseline by a wide margin.

The rest of the paper is structured as follows: In section 2, we list related work for diagnosing Covid-19 from coughing. Our method is presented in detail in section 3. An overview of the dataset and a report on our results is given in section 4. We discuss our findings in section 5. Finally, we summarise our work and provide an outlook on future research directions in section 6.

2. Related Work

[15] used Mel-Frequency Cepstral Coefficients (MFCCs) and pretrained convolutional neural networks (CNNs) to classify phone recordings collected through a website. [8] combined MFCCs, Mel-spectrograms and Linear Predictive Coding Spectrum (LPCS) coefficients into Tensors processed with CNNs. [16] created an app based on a combination of support vector machines (SVMs) and CNNs processing MFCCs and Mel-spectrograms respectively, to distinguish Covid-19, pertussis, bronchitis, and healthy cough recordings. [12] used MFCCs with features extracted from Mel-spectrograms by a pretrained CNN to detect Covid-19 from coughing and breathing sounds gathered from a crowd-sourced dataset.

3. Methods

Our approach employs ensemble learning, combining several classifiers to boost overall performance. The individual models are trained on a variety of features handcrafted for paralinguistics and deep features extracted from spectrograms with pre-trained CNNs. The feature selection is inspired by the systems presented by Brown et al. [12] and Schuller et al. [13] for the tasks of Covid Cough and Speech analysis. We extract a total of 15 feature sets, each both at segment level and in chunks with sliding windows. Those feature sets are then used to train Support Vector Machines (SVMs) and Long Short-Term Memory Networks (LSTMs), respectively. Finally, we use a least-squares weighted late fusion to combine the models. The metric used for rating model performance is the receiver-operator characteristic area under the curve (ROC-AUC), which is obtained by plotting true-positive rate (TPR) against false-positive rate (FPR) for varying decision thresholds. An AUC of 50 % indicates chance level performance.

3.1. Feature Extraction

We extract 15 different feature sets, both handcrafted and produced from deep networks. Furthermore, we extract each feature at the level of the entire segments for the SVMs, and as a sequence of chunks produced by sliding windows for the LSTMs. The latter uses a window width of 1.0 s and a hopsize of 0.5 s. This results in a total of 30 feature sets.

http://dx.doi.org/10.21437/Interspeech.2021-1267

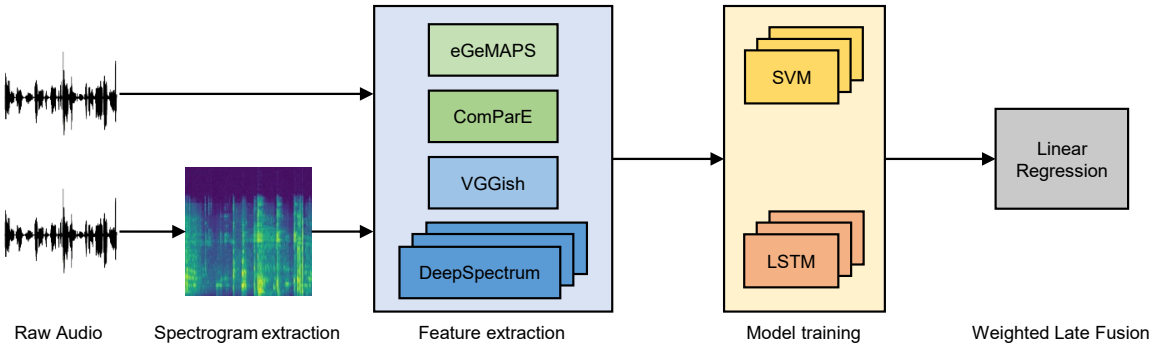

Figure 1: *Illustration of our method. Audio files are converted into spectrograms for* VGGish *and* DeepSpectrum*, as well as processed by* openSMILE *to return* eGeMAPS *and* ComParE *features. Optionally, PCA is applied at this step. Features are then used to train SVMs and LSTMs, one model per feature type. Finally, a linear regression model is used for weighted late fusion on the trained classifiers' validation predictions to combine them into an ensemble and improve performance.*

Table 1: *Features extracted for the challenge. For each feature, segment level and chunked sliding window representations are extracted, and PCA reduction to 90 % and 95 % variance is applied to* DeepSpectrum *and* ComParE *sets. DS =* DeepSpectrum.

Name	Type	Dimension
ComParE	handcrafted	6 373
eGeMAPS	handcrafted	88
VGGish	CNN	128
DS VGG16	CNN	4 096
DS VGG19	CNN	4 096
DS ResNet50	CNN	2 048
DS InceptionResNetV2	CNN	1 536
DS Xception	CNN	2 048
DS DenseNet121	CNN	1 024
DS DenseNet169	CNN	1 664
DS DenseNet201	CNN	1 920
DS MobileNet	CNN	1 024
DS MobileNetV2	CNN	1 280
DS NASNetLarge	CNN	4 032
DS NASNetMobile	CNN	1 056

For extracting handcrafted features, we use the openSMILE toolkit [17]. The feature sets based on expert knowledge it provides have been applied to a variety of paralinguistics tasks [18],[19],[20]. We choose the eGeMAPS and ComParE 2016 feature sets, which have 88 and 6 373 dimensions, respectively.

For the deep features, we compute Mel-spectrograms from the audio and process them with pre-trained CNNs. We use two tools for this, VGGish and DeepSpectrum.

VGGish [21] uses an architecture based on VGG16. It is pretrained with the audioset database, a large collection of sound recordings collected from YouTube. The VGGish preprocessing pipeline extracts Mel-spectrograms with 64 frequency bands on 960 ms audio chunks. While the original implementation produces non-overlapping chunks, we change the hopsize to 0.5 s to match the other features. The CNN transforms the 96x64 inputs to 128-dimensional features.

DeepSpectrum [22] is a Python-based toolkit that can extract various spectrograms from audio and process them with a selection of popular CNN architectures, pre-trained on the Imagenet dataset. We use the default settings for the spectrograms (128 Mel-bands, viridis colormap), and extract features using the following architectures: VGG16, VGG19 [23], ResNet50, [24] InceptionResNetV2, [25] Xception, [26] DenseNet121, DenseNet169, DenseNet201, [27] MobileNet, MobileNetV2, [28] NASNetLarge, and NASNetMobile [29]. The different feature types and their dimensionality are summarised in table 1.

Table 2: *Hyperparameters and their ranges used for crossvalidation during LSTM training.*

LSTM Hyperparameters	
Parameters	Values
Architecture	
cells layer 1	32, 64, 128, 256
cells layer 2	16, 32, 64, 128
bidirectionality	True, False
dropout	0.2, 0.3, 0.4, 0.5
Training	
learning rate	$[10^{-4}$ - $10^{-3}]$
epochs	[10 - 60]

3.2. Pre-processing

The sample sequences are clipped to a length of 5.0 s, shorter samples are zero-padded. We employ principal component analysis (PCA) to reduce the number of features of the higher-dimensional representations. For this, we use the PCA implementation from scikit-learn [30], and set the explained variance to 90 % and 95 %. Features are scaled to zero mean and unit standard deviation before LSTM training. For SVM training, we apply a min max scaling to transform features into the range [-1, 1].

3.3. SVM

For SVM training, we use the SVC implementation from scikit-learn [30], setting the classifiers to output probabilities. Since the dataset is imbalanced towards negative samples, we set the positive class weight proportionally higher. All models use the linear kernel.

Table 3: *Validation AUC in % for the trained SVM and LSTM classifiers. Shown are the mean ROC-AUCs across 5 validation folds for each of the feature sets and with PCA at different variance explanation ratios.*

Feature	Crossvalidation Results					
	SVM ROC-AUC [%]			LSTM ROC-AUC [%]		
	full	PCA 95%	PCA 90%	full	PCA 95%	PCA 90%
ComParE	67.3	61.6	57.9	73.9	62.9	62.9
eGeMAPS	66.8	66.8	66.8	69.1	69.1	69.1
VGGish	54.9	54.9	54.9	61.4	61.4	61.4
VGG16	70.1	55.2	54.3	73.0	66.1	**72.8**
VGG19	53.5	50.6	54.9	70.0	66.9	67.8
ResNet50	64.9	61.0	60.4	72.3	67.0	67.8
InceptionResNetV2	66.3	65.4	62.6	69.2	66.3	65.8
Xception	59.1	52.9	54.6	**75.3**	69.8	72.6
DenseNet121	70.2	64.4	65.1	74.2	70.1	68.5
DenseNet169	69.7	**68.1**	**70.0**	74.8	65.2	69.4
DenseNet201	66.2	64.0	62.7	75.0	**70.1**	69.1
MobileNet	**70.4**	57.5	62.5	75.2	69.9	69.5
MobileNetV2	64.7	51.3	50.8	71.7	67.0	68.2
NASNetLarge	67.1	53.8	52.4	69.7	67.6	70.1
NASNetMobile	67.8	62.1	67.4	72.4	68.8	68.9

Table 4: *Composition of the DiCOVA dataset. Not shown is the held-out test set. Note that the dataset is unbalanced, with negative samples largely outnumbering positives.*

Subset	Dataset composition		
	Positives	Negatives	Total
Dataset	75	965	1 040
Train fold 1–5	50	772	822
Val fold 1–5	25	193	218

We optimise the hyperparameter C across the range $[10^{-5}, 10^{2}]$ through crossvalidation on the provided validation folds. To predict the test set, we then train an optimal SVM on the full training set.

3.4. LSTM

We attempt to leverage temporal information contained in the cough sounds by processing them with recurrent neural networks. For implementation, we use both PyTorch for preliminary experiments and TensorFlow for the final training. Our basic model has an architecture of 2 LSTM layers, followed by a fully connected layer with 32 neurons, a dropout layer, and a final layer with sigmoid activation for classification.

In order to optimise our model, we vary the following hyperparameters: The number of cells in the LSTM layers, using bidirectional layers, the dropout rate, the learning rate, and the number of epochs. The hyperparameters and their ranges are given in section 3.1.

The best hyperparameters for each classifier are determined through random search with crossvalidation on the provided folds. 10 hyperparameter combinations per feature are sampled. The networks are trained using the Adam optimiser [31] with binary crossentropy loss, and early stopping is added to reduce the risk of overfitting.

3.5. Weighted Late Fusion

To fuse our predictions into the final scores, we choose a least-squares approach that weighs the contribution of each classifier. We employ the LinearRegression [30] implementation from scikit-learn. The regression is trained on the concatenated validation predictions of each classifier, and the result is used to combine the test predictions of our classifiers for the final ROC-AUC score.

4. Experiments and Results

We now give an overview over the challenge dataset and present the results of our systems on the validation and test sets. Crossvalidation results are given at one decimal place to reflect the small sample size, but we report the final scores at two decimal places for compatibility with the challenge baselines.

4.1. Dataset

The dataset used for our experiments was derived from the COSWARA dataset [11]. It contains recordings of a total of 1 040 subjects (one recording per subject), of whom 965 are Covid-19 negative but may have other respiratory conditions, and 75 are Covid-19 positive [14]. Thus, the dataset is imbalanced, with negatives outnumbering positives by a factor of approximately 12.9. Recordings have an average duration of 4.72 s and are resampled to 44.1 kHz. We summarise the dataset in table 4.

The dataset is divided into a crossvalidation split mandated by the challenge organisers, with each training fold containing 50 positive and 772 negative samples, and each validation fold containing 25 positive and 193 negative samples, respectively. The test set consists of 233 recordings. Since the labels are withheld, we cannot report the ratio of positive and negative samples.

4.2. Results

The mean ROC-AUC values across the validation folds specified by the challenge organisers for each of our SVM classifiers are summarised in table 3. The classifier trained on the Mo-

Table 5: *Results of the late fusion for SVM, LSTM, and combined SVM+LSTM systems. Shown are the area under the curve (AUC) scores in % for the fused validation and test predictions, as well as test specificity (TNR) at 80 % sensitivity. Also listed are the challenge baseline scores.*

Fusion results			
Classifier	**ROC-AUC**		**TNR**
	Val [%]	Test [%]	Test [%]
SVM			
SVM_{full}	94.82	75.82	52.60
$SVM_{PCA\ 95\%}$	76.58	64.49	39.58
$SVM_{PCA\ 90\%}$	92,70	76,80	53.12
LSTM			
$LSTM_{full}$	80.48	74.92	59.90
$LSTM_{PCA\ 95\%}$	76.58	69.77	43.23
$LSTM_{PCA\ 90\%}$	78.69	75.65	-
SVM + LSTM			
$LSTM_{full}$ + $SVM_{PCA\ 90\%}$	94.34	76.35	57.81
$LSTM_{PCA\ 90\%}$ + $SVM_{PCA\ 90\%}$	94.31	**78.18**	58.85
Challenge Baseline			
Logistic Regression	66.97	61.97	-
Multilayer Perceptron	68.81	69.91	-
Random Forest	70.63	67.59	-

BILENET features performed best at 70.4 %.

The mean ROC-AUCs on the validation set achieved by our LSTM classifiers per feature are also reported in table 3. XCEPTION without PCA achieved the highest value of 75.3 %. In most cases, applying PCA reduced the validation AUC for both SVM and LSTM models.

Fusing our SVMs resulted in AUC of 75.82 % on the test set and test specificity (TNR) at 80 % sensitivity of 52.60 % using the feature set without PCA. Applying PCA with 90 % variance explained increased the test AUC to 76.80 % and TNR to 53.12 %. However, PCA with 95 % variance explained decreased the test AUC to 64.49 and TNR to 39.58 %.

Weighted late fusion of our LSTMs achieved a test AUC of 74.92 % and 59.90 % TNR with the original feature files. The models trained on features transformed with PCA for 95 % and 90 % variance result in test AUC scores of 69.77 % with 43.23 % TNR and 75.53 % AUC with no valid TNR score respectively.

Finally, late fusion of the predictions of both our LSTM and SVM models give a test measure of 78.18 % AUC and 58.85 % TNR, with a corresponding validation score of 94.31 % AUC. We summarise the results of our fusion algorithm, including ROC-AUC scores as well as test specificity (TNR) at 80 % sensitivity, along with the challenge baseline, in table 5.

5. Discussion

Our results demonstrate that the proposed ensemble approach with models trained on 15 handcrafted and deep features is capable of surpassing the challenge baseline for both SVM and LSTM classifiers. Using an even larger ensemble by combining SVMs and LSTMs for a total of 30 classifiers leads to a further improvement.

We note that while applying PCA with 90 % variance explained improved the test set performance for both SVMs and LSTMs, while PCA with 95 % variance explained deteriorated the test set performance below the level achieved with training on the original feature set. One would expect that, if PCA negatively impacted performance, the effect would be more pronounced when less variance is explained by the learnt representation. Another notable observation from our individual model validation scores is that DEEPSPECTRUM features appear to surpass VGGISH features, despite their CNNs not being pre-trained on audio.

We have conducted our experiments on a limited dataset containing a small number of positive samples, with no data augmentation. Using a larger dataset or a combination of datasets may allow us to further improve performance. While our results surpass the challenge baseline, they are not yet sufficient to deploy the model in a real application. The current system is limited to binary classification of cough sounds, but it could be extended to processing speech and breathing, or classifying other pathologies than Covid-19.

6. Conclusions

We have presented an ensemble learning approach for detecting Covid-19 infection from coughing, based on SVMs and LSTMs trained on a combination of handcrafted features and deep feature extraction from Mel-spectrograms with pre-trained CNNs. Applying it to the dataset of the DiCOVA 2021 challenge yielded a test score of 78.18 % ROC-AUC, surpassing the baseline.

Opportunities for future work include applying our approach to a larger dataset to further improve performance, as well as extending it to speech and breathing and classifying different pathologies. From a technical point of view, other fusion schemes such as working on an earlier level appear promising. Also, self-supervision could be beneficial given the small size of the training data.

7. Acknowledgements

The authors would like to thank Shahin Amiriparian, Alice Baird, Adria Mallol-Ragolta, and Zhao Ren for their advice.

The authors acknowledge funding from the DFG (German Research Foundation) Reinhart Koselleck-Project AUDI0NOMOUS under grant agreement No. 442218748.

We express our deepest sorrow for those who left us due to COVID-19; they are not numbers, they are lives. We further express our highest gratitude and respect to the clinicians and scientists, and anyone else these days helping to fight against COVID-19, and at the same time help us maintain our daily lives.

8. References

[1] L. Lee, R. G. Loudon, B. H. Jacobson, and R. Stuebing, "Speech breathing in patients with lung disease," *American Review of Respiratory Disease*, vol. 147, pp. 1199–1199, 1993.

[2] R. X. A. Pramono, S. A. Imtiaz, and E. Rodriguez-Villegas, "A cough-based algorithm for automatic diagnosis of pertussis," *PloS one*, vol. 11, no. 9, 2016.

[3] G. Botha, G. Theron, R. Warren, M. Klopper, K. Dheda, P. Van Helden, and T. Niesler, "Detection of tuberculosis by automatic cough sound analysis," *Physiological Measurement*, vol. 39, no. 4, p. 045005, 2018.

[4] A. Windmon, M. Minakshi, P. Bharti, S. Chellappan, M. Johansson, B. A. Jenkins, and P. R. Athilingam, "Tussiswatch: A smart-phone system to identify cough episodes as early symptoms of chronic obstructive pulmonary disease and congestive heart failure," *IEEE J. Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1566–1573, 2018.

[5] G. Deshpande and B. Schuller, "An overview on audio, signal, speech, & language processing for covid-19," *arXiv preprint arXiv:2005.08579*, 2020.

[6] N. Islam, J.-P. Salameh, M. Leeflang, L. Hooft, T. McGrath, C. Pol, R. Frank, S. Kazi, R. Prager, S. Hare, C. Dennie, R. Spijker, J. Deeks, J. Dinnes, K. Jenniskens, D. Korevaar, J. Cohen, A. Van den Bruel, Y. Takwoingi, J. de Wijgert, J. Wang, and M. McInnes, "Thoracic imaging tests for the diagnosis of covid-19," *Cochrane Database of Systematic Reviews*, no. 11, 2020.

[7] J. Han, K. Qian, M. Song, Z. Yang, Z. Ren, S. Liu, J. Liu, H. Zheng, W. Ji, T. Koike, X. Li, Z. Zhang, Y. Yamamoto, and B. W. Schuller, "An early study on intelligent analysis of speech under covid-19: Severity, sleep quality, fatigue, and anxiety," *arXiv preprint: arXiv 2005.00096*, 2020.

[8] J. Andreu-Perez, H. Perez-Espinosa, E. Timonet, M. Kiani, M. I. Giron-Perez, A. B. Benitez-Trinidad, D. Jarchi, A. Rosales, N. Gkatzoulis, O. F. Reyes-Galaviz, A. Torres, C. A. Reyes-Garcia, Z. Ali, and F. Rivas, "A generic deep learning based cough analysis system from clinically validated samples for point-of-need covid-19 test and severity levels," *IEEE Transactions on Services Computing*, pp. 1–1, 2021.

[9] "CMU sounds for COVID Project," https://node.dev.cvd.lti.cmu.edu/, 2020, [Online; accessed 07-Aug-2020].

[10] L. Orlandic, T. Teijeiro, and D. Atienza, "The coughvid crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms," *arXiv preprint arXiv:2009.11644*, 2020.

[11] N. Sharma, P. Krishnan, R. Kumar, S. Ramoji, S. R. Chetupalli, N. R., P. K. Ghosh, and S. Ganapathy, "Coswara – a database of breathing, cough, and voice sounds for COVID-19 diagnosis," in *Proceedings INTERSPEECH 2020, 21nd Annual Conference of the International Speech Communication Association*, Shanghai, China, September 2020.

[12] 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. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2020, p. 3474–3484.

[13] 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, L. J. M. Rothkrantz, J. Zwerts, J. Treep, and C. Kaandorp, "The interspeech 2021 computational paralinguistics challenge: Covid-19 cough, covid-19 speech, escalation & primates," *arXiv preprint: arXiv:2102.13468*, 2021.

[14] A. Muguli, L. Pinto, N. R., N. Sharma, P. Krishnan, P. K. Ghosh, R. Kumar, S. Ramoji, S. Bhat, S. R. Chetupalli, S. Ganapathy, and V. Nanda, "Dicova challenge: Dataset, task, and baseline system for covid-19 diagnosis using acoustics," *arXiv preprint: arXiv:2103.09148*, 2021.

[15] J. Laguarta, F. Hueto, and B. Subirana, "Covid-19 artificial intelligence diagnosis using only cough recordings," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 275–281, 2020.

[16] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, M. S. Riaz, K. Ali, C. N. John, M. I. Hussain, and M. Nabeel, "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app," *Informatics in Medicine Unlocked*, vol. 20, p. 100378, 2020.

[17] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1459–1462.

[18] B. Schuller, S. Steidl, A. Batliner, P. B. Marschik, H. Baumeister, F. Dong, S. Hantke, F. B. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, S. Amiriparian, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis, and S. Zafeiriou, "The interspeech 2018 computational paralinguistics challenge: atypical and self-assessed affect, crying and heart beats," in *Proceedings INTERSPEECH 2018, 19th Annual Conference of the International Speech Communication Association, September 2-6, Hyderabad, India: Speech Research for Emerging Markets in Multilingual Societies*, B. Yegnanarayana, C. C. Sekhar, S. Narayanan, S. Umesh, S. R. M. Prasanna, H. A. Murthy, P. Rao, P. Alku, and K. Ghosh, Eds., September 2018, pp. 122 – 126.

[19] B. Schuller, A. Batliner, C. Bergler, F. B. Pokorny, J. Krajewski, M. Cychosz, R. Vollmann, S.-D. Roelen, S. Schnieder, E. Bergelson, A. Cristia, A. Seidl, A. S. Warlaumont, L. Yankowitz, E. Nöth, S. Amiriparian, S. Hantke, and M. Schmitt, "The interspeech 2019 computational paralinguistics challenge: Styrian dialects, continuous sleepiness, baby sounds & orca activity," in *Proceedings INTERSPEECH 2019, 20nd Annual Conference of the International Speech Communication Association*, G. Kubin and Z. Kačič, Eds. Graz, Austria: ISCA, September 2019, pp. 2378 – 2382.

[20] B. W. Schuller, A. Batliner, C. Bergler, E.-M. Messner, A. Hamilton, S. Amiriparian, A. Baird, G. Rizos, M. Schmitt, L. Stappen *et al.*, "The interspeech 2020 computational paralinguistics challenge: Elderly emotion, breathing & masks," in *Proceedings INTERSPEECH 2020, 21nd Annual Conference of the International Speech Communication Association*. Shanghai, China: ISCA, September 2020, pp. 2042–2046.

[21] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 131–135.

[22] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, "Snore sound classification using image-based deep spectrum features," in *Proceedings INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association*. ISCA, 2017, pp. 3512–3516.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence*, ser. AAAI'17. San Francisco, California, USA: AAAI Press, 2017, p. 4278–4284.

[26] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[29] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

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