INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Paralinguistic Classification of Mask Wearing by Image Classifiers and Fusion

Jeno Szep, Salim Hariri

Dept. of Electrical and Computer Engineering, The University of Arizona
szep@arizona.edu, hariri@arizona.edu

Abstract

In this study, we address the ComParE 2020 Paralinguistics Mask sub-challenge, where the task is the detection of wearing surgical masks from short speech segments. In our approach, we propose a computer-vision-based pipeline to utilize the capabilities of deep convolutional neural network-based image classifiers developed in recent years and apply this technology to a specific class of spectrograms. Several linear and logarithmic scale spectrograms were tested, and the best performance is achieved on linear-scale, 3-Channel Spectrograms created from the audio segments. A single model image classifier provided a 6.1% better result than the best single-dataset baseline model. The ensemble of our models further improves accuracy and achieves 73.0% UAR by training just on the 'train' dataset and reaches 80.1% UAR on the test set when training includes the 'devel' dataset, which result is 8.3% higher than the baseline. We also provide an activation-mapping analysis to identify frequency ranges that are critical in the 'mask' versus 'clear' classification.

Index Terms: spectrogram, convolutional neural networks (CNN), image-classification, ensemble learning, computational paralinguistics

1. Introduction

Due to the Covid-19 pandemic, the benefits of automatic detection, if the speaker wears a face mask, became more critical. The wear of a surgical mask among doctors was assessed [1], but no automated detection solution was proposed so far. In this paper, we propose a computer-vision-based pipeline for classifying the speech segments if the speaker wears a surgical mask or not.

In the last few years, the rapid development of computer vision resulted in a set of robust convolutional neural network-based image classifiers that became common in many fields. These new classifiers also became an essential part of the toolset of acoustic signal analysis. The technology utilizes the possibility that spectral images created from the audio signal can be analyzed by the deep convolutional neural networks. There are multiple approaches and combinations of them in this field, that are used for speaker identification [2,3,4], baby cry detection [5,6], sentiment analysis [7], and animal sound detection [8,9]. One of the best practice approaches is using pre-trained image classifier networks for feature extraction and then processes them further. This method has performed well in several areas, e.g., in snore classification [10]. Such feature sets were also used to set up the baseline for the current challenge [11]. As another approach, it is also possible to train the classifiers directly for a specific purpose, without making feature extraction [12,13,14,15]. In this paper, we focus on this latter approach.

The organizers provided a sufficiently sizeable speech-segment dataset, with good quality sound recordings, and they also provided best-practice feature sets, as well as the baseline of the challenge. The dataset of the challenge has three cohorts of one-second-long speech segments, one for training ('train'), one for development ('devel'), and one for testing ('test').

In our approach, we transformed the speech-segments into spectrogram images and then trained image classifiers to identify whether the speaker wore a surgical mask or not. The parameters in the process of creating the spectrograms play a crucial role in the accuracy of the final results. Therefore, we analyzed different approaches to understand how the frequency range, plotting scale, and the Fourier window-size affects the efficiency of the classification. We concluded that in this task, the linear-scale spectrograms are more beneficial than the logarithmic-scale ones. We also utilized a new idea of constructing 3-Channel Spectrograms with different parameters for each channel that has improved the accuracy by approximately two percent. The transfer learning concept was used: we start with image classifiers already pre-trained on ImageNet, and we finetune them for this specific task. First, we trained multiple image classifiers on the training data and validated the performance on the 'devel' dataset. Our best result with a single model on a single spectrogram set outperforms the best result on a single dataset of the baseline by 6.1% UAR. In the second step, we trained the image classifiers on both the 'train' and the 'devel' set. Here we used the K-fold cross-validation technique, without the possibility of reliable validation due to the dependency between speech segments within the validation dataset. We also applied ensemble methods to utilize the versatility of multiple models and multiple spectrogram sets. As a result, in the first step, we achieved 73.0% AUR on the development set by training solely on the training set, and in the second step, we achieved 80.1% AUR on the test set.

2. Experimental Framework

In this chapter, we describe the data, the tools, and the technological steps performed in the experiment.

2.1. Data

The dataset of the Mask Sub-Challenge is the Mask Augsburg Speech Corpus (MASC) that contains 36,554 short one-second-long speech segments of 32 German native speakers speaking with and without operation masks. The dataset is segmented into three parts for training, development, and testing. We assume the organizers divided the 32 speakers into three groups and created the three cohorts from the speeches of the three groups of people to ensure that different cohorts do not contain the speech of the same person. This ensures, that the samples of the three cohorts are independent. As a result, the samples between cohorts are independent, but samples within cohorts can be dependent, since two samples may come from the same

http://dx.doi.org/10.21437/Interspeech.2020-2857

sentence of a speaker. Since, in general, the validation dataset and the training dataset should be independent, the standard K-fold cross-validation technique cannot be used while training machine learning models on these datasets, due to the lack of reliable validation accuracy.

2.2. Spectrograms

Spectrograms (SP) have been used for speech analysis for decades [16] because they provide meaningful information on the voice in the form of an image. In speech analysis, both the frequency spectrum and the time-dependence of that are of critical importance. While creating a spectrogram, many parameters should be considered, and one of the most basic ones is the window-size, which determines the length of the domain of the Fourier transform. A small window-size provides good time-resolution but limited frequency-resolution. On the other hand, larger window-size provides proper spectral resolution aven at low frequencies, but limited time-resolution. At a given window-size, usually, it is not possible to have both. Therefore, we generated color spectrograms. The three RGB channels of the image were generated with different window-sizes resulting in colored spectrograms that contain both high frequency-resolution and high time-resolution channels, as in Fig. 1.

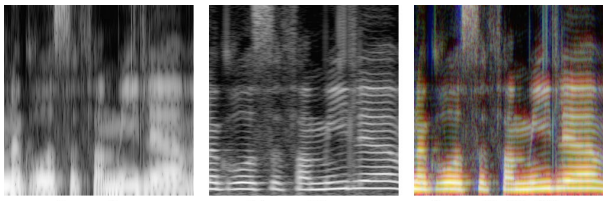

Figure 1, Linear-scale spectrograms with different Hann-window sizes (ws). a) and b) are single-channel images, c) is a 3-Channel Spectrogram.

In our experiments, the 3-channel composite spectrograms provided slightly better accuracies than the single window-size versions.

We have used several spectrogram types and tested their performance on the dataset. These included classical spectrograms with linear and quasi-logarithmic frequency-scale, Mel-Scale Spectrograms, Constant-Q Spectrograms [17], and Mel-frequency Cepstral Coefficient Cepstrograms [18]. We have used the Parselmouth API [19] Python library for the Praat software [20], and also the Librosa toolkit [21] for Python.

In the following, we present the results obtained with the following spectrograms:

1. 1CH: Linear frequency scale from 20 to 7000 Hz, time_step = 3 ms, frequency_step = 20 Hz, single-channel images, in three versions with Hann-windows of either 8, 15, or 30 ms.
2. 3CH-0: Linear frequency scale from 20 to 7000 Hz, time_step = 3 ms, frequency_step = 20 Hz, Hann-window sizes = (8, 15, 30) ms for the narrow version (N3CH) and (5, 15, 50) ms for the wide version (W3CH).
3. 3CH-75 and 3CH-60: Same parameters as of 3CH-0, but there is a cutoff at -75 dB / -60 dB in order to reduce the effects of the background noise.
4. MSS: Mel-Scale Spectrograms, with FFT windows of (2.6, 5.1, 10.5) ms.
5. ConstQ: Spectrograms based on the Constant-Q transform [8]. The frequency range is seven octaves starting at C1, with (12, 24, 48) bins per octave for the three channels of the image.

All spectrograms were scaled to 320x320 pixel size.

When feeding images to classifiers for training, we used a label-preserving limited augmentation (max rotation = 3°, max zoom = 140%, max lighting change = 30%, max warp = 0.03).

2.3. Image classifiers

In the last decade, the importance of computer vision has been the driving force of the development of advanced deep convolutional neural networks for image classification. The ImageNet challenge provides the baselines. In this research, we used the common practice of transfer learning when models pre-trained for the ImageNet challenge are retrained for specific classification purposes. This technique utilizes the fact that the front layers of an image classifier usually do not need retraining; therefore, the training process for a specific purpose can be relatively short. We have used three different image classifier architectures: VGGNet, ResNet, and DenseNet. All models used a final sigmoid layer.

2.3.1. VGGNet

The VGGNet architecture uses 3x3 convolutional layers stacked on top of each other in increasing depth, followed by two fully connected layers and a softmax classifier [22]. We have used the VG19 version of this architecture.

2.3.2. ResNet

The ResNet residual-network architecture is a deep convolutional neural network that has a fundamental building block where a previous layer is merged into the following layer. This forces the network to learn residuals (the difference between a previous layer and the current one) [23]. We experimented with the ResNet-50 and the ResNet-101 models.

2.3.3. DenseNet

The DenseNet architecture is an extension of ResNet, where all fundamental blocks are connected with the concatenation of the feature maps [24]. We have used the DenseNet-121 pre-trained model in the experiment.

2.4. Ensemble learning

Ensemble learning provides an advantage over the single-model approach [25]. Utilizing different models and different spectrogram types, we can obtain a more accurate and more robust classification system. For the fusion of predictions, we employed averaging and majority voting. The outputs of the models are class probabilities; so that we averaged the probabilities of multiple predictions for each sample to decide which class the sample belongs to. In the case of majority voting, each prediction has one vote for the class, and the final classification is decided by the tally.

In this research, we also experimented with adding predictions based on the tabular feature sets that were provided by the organizers, as an additional modality that might add accuracy to the image-based classification by blending.

3. Experimental results

3.1. 'Training at daylight'

As a first step, the image classifiers described in section 2.4 were trained on the spectrograms of the 'train' dataset and validated on the spectrograms of the 'devel' dataset. Because the 'devel' dataset is independent, we can accurately validate the models while training so that we can tune the models appropriately. We called this 'training at daylight' because we were able to see the models' accuracy while training.

Table 1 shows the performance of the classifiers on the different types of spectrograms. The unweighted average recall (UAR) values in the table are the average results of two independent trainings.

Regarding the spectrograms, the best results were achieved on the 3CH-0 and 3CH-75 versions. The use of the Constant-Q Spectrograms and the Mel-Scale Spectrograms resulted in 4-5% lower UAR values, for probable reasons explained later.

Table 1: *The UAR values of classification on the spectrograms trained on the 'train' set and validated on the 'devel' set.*

Spectr. set	ResNet 50	**ResNet 101**	**DenseNet 121**	**VG19**	Avg.
1CH-30	0.664	0.690	0.683	0.687	0.681
N3CH-0	0.683	**0.692**	**0.687**	**0.694**	0.689
N3CH-75	0.688	**0.700**	**0.698**	**0.693**	0.695
N3CH-60	0.659	0.655	0.662	0.664	0.660
W3CH-0	0.664	**0.695**	**0.688**	**0.686**	0.683
W3CH-75	0.680	**0.705**	**0.692**	**0.690**	0.692
W3CH60	0.661	0.671	0.664	0.667	0.665
MSS	0.661	0.662	0.645	0.653	0.655
ConstQ	0.586	0.667	0.653	0.655	0.640
Avg.	0.661	0.682	0.675	0.677	

We trained the models for 12 epochs with a gradually decreasing learning rate starting with 1.5e-3 and finishing with 2.0e-5. The 12 epochs had a distribution of 7 epoch training the model frozen up-to-the average-pooling layer, followed by 3 epochs training when only the first three convolutional layers are frozen, and 2 epochs training on all weights. Optimized with the Adam optimizer (beta = 0.9, 0.99), weight decay = 0.02, loss function = Cross-Entropy.

We decided to proceed to the second step of the experiment with the 12 combinations of the better-performing three models and four datasets only, that are marked with bold characters in Table 1.

3.2. 'Training in the darkness'

In the second step, we wanted to operate on a broader training set to obtain better predictions for the 'test' set. Therefore, we combined the 'train' and the 'devel' datasets. We also wanted to take advantage of the K-fold cross-validation approach on different folds. However, as we discussed in section 2.1 a standard K-fold CV is not feasible on this dataset. Therefore, we made a *'K-fold in the darkness'* cross-validation, when the selected models were trained with the selected spectrogram types on the union of the 'train' and 'devel' datasets in a K-fold scheme. The expression *'in the darkness'* refers to the fact, described in section 2.1, that we do not know the real accuracy of the validation due to a specific property of the datasets. In this case, the measured UAR values are supposedly larger than they would have been on an independent dataset. Since we cannot trust the validation accuracy, we cannot optimize the training parameters. Therefore we trained the models with the same parameter settings that were proved to be the best set in the first step of the experiment. Since the observed UAR values at this training are higher than the ones we would obtain on an independent test set, we marked these values with parentheses in Table 2. The meaning of these numbers is what the UAR is when predicting from speech segments when the training set includes other speech segments of the same person.

Table 2: *The UAR values of classification on the spectrograms trained in 5-fold cross-validation on both the 'train' set and the 'devel' set.*

Spectr. set	ResNet-101	DenseNet-121	VG19	Avg.
N3CH-0	*(0.903)*	*(0.878)*	*(0.887)*	*(0.889)*
N3CH-75	*(0.910)*	*(0.881)*	*(0.894)*	*(0.895)*
W3CH-0	*(0.899)*	*(0.876)*	*(0.886)*	*(0.887)*
W3CH-75	*(0.905)*	*(0.887)*	*(0.889)*	*(0.894)*
Avg.	*(0.904)*	*(0.880)*	*(0.889)*	

In the second step, we trained the models with 5-fold cross-validation. This approach has the advantage that prediction on the 'test' set is calculated at each fold, and the averaging of these five predictions can yield better accuracy on the 'test' set.

The number of epochs and the learning rates were set to the same values that were used at the training at 'daylight'.

3.3. Data fusion

In the blending, we use the predictions of the classifiers as feature vectors for calculating the final prediction. We use classical averaging and majority voting, as described in section 2.4. for the models and datasets presented in Table 2. The ensemble results for training at 'daylight' are summarized in Table 3.

Table 3: *The UAR and accuracy values obtained by fusion methods on the 'devel' dataset.*

	UAR		Accuracy	
	Averaging	Voting	Averaging	Voting
'Daylight'	0.730	0.728	0.730	0.730
'Darkness'	(0.925)	(0.924)	(0.926)	(0.925)

The 'Daylight' predicted values have 74.3% specificity, 69.9% sensitivity, and the confusion matrix is shown in Figure 2.

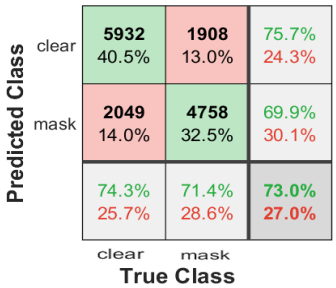

Figure 2: Confusion matrix of the validation on the 'devel' set when training on the 'train' set only.

As expected, the 'training in the darkness' process provided models with higher accuracy than 'daylight' training. The ensemble prediction on the 'test' set resulted in **80.1 % UAR**, which is 8.3% higher than the baseline UAR of 71.8%. This result also shows the difference between the UARs if the training set includes speech segments from the same people as the validation (92.5% - when validating on the 'devel' set) or not (80.1% - validating on the independent 'test' set).

4. Discussion

4.1. Class Activation Mapping (CAM)

Activation Mapping (CAM) is a technique for producing heat maps to highlight class-specific regions of images [26]. We use this method to identify frequency ranges that are critical in the 'mask' versus 'clear' classification

The last convolutional layer of the ResNet-101 model is a 2048x10x10 size tensor, where the 10x10 indices (7x7 at VG19) are from the convolutions; therefore, they represent locations on the spectrogram images. The activation values of these cells are used by the average-pooling layer before the prediction on the output layer, in other words, the activation values of the tensor *A(z,x,y)* of the last convolutional layer determine feature *z* based on its values in the *x-y* plane of the image. In the case of a trained model, the average activation values of these cells tell us what part of the spectrogram the CNN model is focused on when deciding in which class the image belongs. Averaging tensor *A* for *z* we get an *I(x,y)* activation intensity map, that can be considered as a marker representing the importance of the area *(x,y)* of the image for the classification decision on the sample if it is a 'mask' or 'clear'. In Fig. 3. the activation values are displayed as a heatmap in the right-side images, over the semi-transparent original image. The spectrograms are displayed in the left-side images.

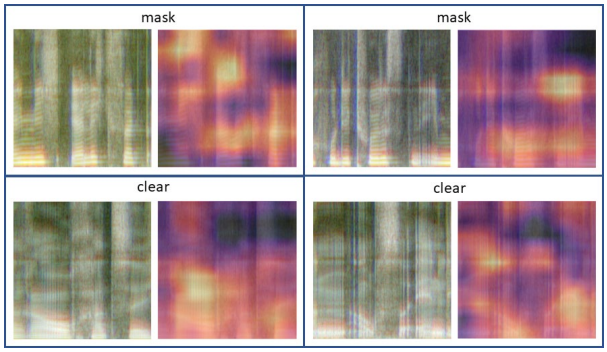

Figure 3: *Four randomly chosen examples of spectrograms and corresponding activation heatmaps of a trained ResNet-101 classifier. The horizontal axis is time, and the vertical axis is the frequency from 20-7000 Hz. On the heatmaps, the brightness represents the activation value of the last convolutional layer of the specific region.*

The CAM technique traditionally is used for identifying the critical regions on individual images. In our case, the structure of the spectrograms is the same in all images, therefore by also averaging the values over the spectrograms, we got an 'importance map' of the different regions on the spectrograms that is characteristic for the given class of spectrograms. When averaging the activations over 1000 samples in a trained model and then also averaging the activation values over the trained models in a model class, a pattern can be observed. Fig. 4. shows the obtained average activation values mapped on the frequency spectrum of the spectrograms, with an added spline line for more visibility.

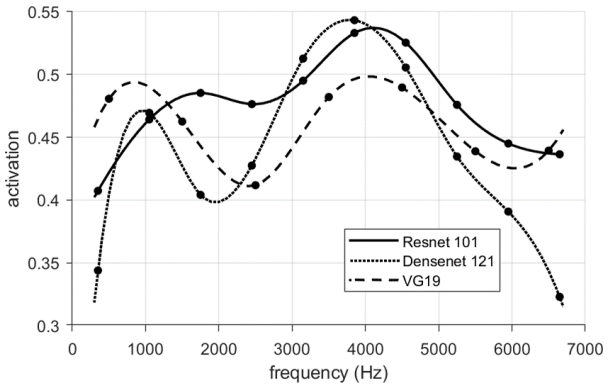

Figure 4: *Average activation values marked with dots and plotted with a fitted spline for three different trained classifiers on the 3CH spectrogram sets. The activation values play a role in deciding if the sample belongs to the 'clear' or the 'mask' classes.*

As Fig. 4 shows, the curves always show a peak in the 3-5 kHz range and also a secondary peak around or below 1 kHz. A similar analysis with the spectrograms with a logarithmic frequency scale verified the above assumption.

Based on this analysis, we hypothesize that the reason why the linear scale spectrograms performed better than the others is that the 3-5 kHz frequency range, which is crucial for this classification task is more compressed on the logarithmic scale, while it is better detailed in the linear scale spectrograms. In our linear-scale spectrogram, this range occupies about one-third of the spectrogram area, while on a Mel-scale spectrogram, it is only about 15%.

4.2. Tabular data

The organizers of the challenge provided a variety of state-of-the-art tabular feature sets [11], extracted from the speech segments. Two of the tabular data sets, the AuDeep [27], and the DeepSpectrum [7], are both spectrogram-based, and the latter one uses a pre-trained ResNet 50 classifier for feature extraction. In the SVM-based baseline classification of the baseline paper [11], the latter feature set performed the best on the test set.

We also tried to add to our ensemble the predictions based on the tabular datasets. We used the regressor version of SVM for four datasets (ComParE, BoAW2000, DeepSpectrumR50, AuDeep-Fused) and added the predictions to our 12-member ensemble. This additional fusion slightly increased the 'daylight' UAR from 73.0 % to 73.3%.

5. Conclusions

We proposed a computer vision pipeline that performs a spectrogram-based classification with deep convolutional networks, that have been developed for image classification but are relatively simple to retrain for other classification methods. On the 'devel' dataset our best single model on a single spectrogram set reached 70.5% UAR which is higher than the baseline's best single dataset result of 64.4%, and our ensemble on the 'test' set reached 80.1% UAR, which is higher than the baseline's 71.8%.

6. References

[1] Ahmad M, Mohmand M H, Ahmad T. 'A Survey among Plastic Surgeons Wearing a Mask in Operating Room.' *World J Plast Surg* 2019; vol. 8(1) pp. 93-96. doi: 10.29252/wjps.8.1.93. 2019

[2] Joon Son Chung, Arsha Nagrani, Andrew Zisserman, 'VoxCeleb2: Deep Speaker Recognition,' *Proc. Interspeech 2018*, pp. 1086-1090, 2018.

[3] Sarthak Yadav, Atul Rai, 'Frequency and Temporal Convolutional Attention for Text-Independent Speaker Recognition,' *arXiv:*1910.07364v2 [cs.SD], 2019.

[4] Arsha Nagraniy, Joon Son Chungy, Andrew Zisserman, 'VoxCeleb: a large-scale speaker identification dataset,' *in Proc. Interspeech 2017 August 20–24, Stockholm*, Sweden, pp. 2616-2620, 2017.

[5] L. Le, A. N. M. H. Kabir, C. Ji, S. Basodi, and Y. Pan, 'Using Transfer Learning, SVM, and Ensemble Classification to Classify Baby Cries Based on Their Spectrogram Images,' *IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW),* Monterey, CA, pp. 106-110, 2019.

[6] Rami Cohen, Dima Ruinskiy, Janis Zickfeld, Hans Ijzerman, and Yizhar Lavne, 'Baby Cry Detection: Deep Learning and Classical Approaches,' *PsyArXiv*, retrieved in April 2020 from https://psyarxiv.com 2019.

[7] Shahin Amiriparian, Nicholas Cummins, Sandra Ottl, Maurice Gerczuk, and Björn Schuller, 'Sentiment Analysis Using Image-based Deep Spectrum Features,' *Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 978-1-5386-0680-3/17/$31.00 c 2017 IEEE, 2017

[8] Harvey, M. et al., 'Acoustic detection of humpback whales using a convolutional neural network,' https://ai.googleblog.com /2018/10/acoustic-detection-of-humpback-whales.html, retrieved April 2020, 2018

[9] Himawan, I., Towsey, M., Law, B. & Roe, P. 'Deep learning techniques for koala activity detection.' In *Proc. Interspeech 2018*, pp. 2107–2111, 2018

[10] Shahin Amiriparian, Maurice Gerczuk, Sandra Ottl, Nicholas Cummins, Michael Freitag, Sergey Pugachevskiy, Alice Baird, and Björn Schuller, 'Snore sound classification using image-based deep spectrum features.' In *Proc. Interspeech 2017, ISCA*, Stockholm, Sweden, pp. 3512–3516, 2017.

[11] Björn W. Schuller, Anton Batliner, Christian Bergler, Eva-Maria Messner, Antonia Hamilton, Shahin Amiriparian, Alice Baird, Georgios Rizos, Maximilian Schmitt, Lukas Stappen, Harald Baumeister, Alexis Deighton MacIntyre, Simone Hantke: 'The INTERSPEECH 2020 Computational Paralinguistics Challenge: Elderly Emotion, Breathing & Masks*', Proc. Interspeech 2020*, Shanghai, China, 2020.

[12] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, Kevin Wilson, 'CNN Architectures for Large-Scale Audio Classification,' *arXiv:*1609.09430v2 [cs.SD] 10 Jan 2017

[13] Mangalam Sankupellay and Dmitry Konovalov, 'Bird Call Recognition using Deep Convolutional Neural Network, ResNet-50.' *Proceedings of ACOUSTICS 2018*, 7-9 November, Adelaide, Australia, 2018

[14] Mark Thomas, 'Towards a Novel Data Representation for Classifying Acoustic Signals.' *Advances in Artificial Intelligence: 32nd Canadian Conference on Artificial Intelligence,* Canadian AI 2019, Kingston, ON, Canada, May 28–31, pp. 601-604, 2019.

[15] Christian Bergler, Hendrik Schröter, Rachael Xi Cheng, Volker Barth, MichaelWeber, Elmar Nöth, Heribert Hofer, Andreas Maier, 'ORCA-SPOT: An Automatic Killer Whale Sound Detection Toolkit Using Deep Learning' *ScientificReports,* 9:10997 | https://doi.org/10.1038/s41598-019-47335-w, 2019

[16] J L Flanagan, *Speech Analysis, Synthesis and Perception, Chapter 7*, Springer- Verlag, New York, 1972

[17] Schoerkhuber, Christian, and Anssi Klapuri. 'Constant-Q transform toolbox for music processing.' *7th Sound and Music Computing Conference*, Barcelona, Spain. 2010.

[18] S. B. Davis and P. Mermelstein, 'Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences,' *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[19] Yannick Jadoul and Bill Thompson and Bart de Boer, 'Introducing Parselmouth: A Python interface to Praat', *Journal of Phonetics,* vol. 71, pp. 1--15, https://doi.org/10.1016/j.wocn. 2018.07.001, 2018

[20] Paul Boersma and David Weenink, 'Praat: doing phonetics by Computer program,' Version 6.0.37, retrieved 3 February 2018. http://www.praat.org/, 2018

[21] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenbergk, Oriol Nieto, 'librosa: Audio and Music Signal Analysis in Python,' *Proc. of the 14th Python in Science Conf. (Scipy 2015)*, pp. 18-24, 2015.

[22] Karen Simonyan, Andrew Zisserman, 'Very Deep Convolutional Networks for Large-Scale Image Recognition,' *arXiv*:1409.1556 [cs.CV], 2015

[23] Kaiming He, Xiangyu Zhang, Shaoqing, Ren Jian Sun, 'Deep Residual Learning for Image Recognition,' *arXiv*:1512.03385v1 [cs.CV] 10 Dec 2015.

[24] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, 'Densely Connected Convolutional Networks.' *arXiv:*1608.06993v5, 2018.

[25] Lior Rokach, *Ensemble Learning – Pattern Classification Using Ensemble Methods, Series in Machine Perception and Artificial Intelligence*, New Jersey: World Scientific, 2019.

[26] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, 'Learning Deep Features for Discriminative Localization,' accessed in April 2020, from http://openaccess.thecvf.com/content_cvpr_2016/papers/Zhou_Learning_Deep_Features_CVPR_2016_paper.pdf, 2016

[27] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, 'auDeep: Unsupervised Learning of Representations from Audio with Deep Recurrent Neural Networks,' *Journal of Machine Learning Research*, vol. 18, pp. 1–5, 2018.