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

A Deep and Recurrent Architecture for Primate Vocalization Classification

Robert Müller, Steffen Illium and Claudia Linnhoff-Popien

Mobile and Distributed Systems Group,
LMU Munich
Germany

{robert.mueller,steffen.illium,linnhoff}@ifi.lmu.de

Abstract

Wildlife monitoring is an essential part of most conservation efforts where one of the many building blocks is acoustic monitoring. Acoustic monitoring has the advantage of being non-invasive and applicable in areas of high vegetation. In this work, we present a deep and recurrent architecture for the classification of primate vocalizations that is based upon well proven modules such as bidirectional Long Short-Term Memory neural networks, pooling, normalized softmax and focal loss. Additionally, we apply Bayesian optimization to obtain a suitable set of hyperparameters. We test our approach on a recently published dataset of primate vocalizations that were recorded in an African wildlife sanctuary. Using an ensemble of the best five models found during hyperparameter optimization on the development set, we achieve a Unweighted Average Recall of 89.3% on the test set. Our approach outperforms the best baseline, an ensemble of various deep and shallow classifiers, which achieves a UAR of 87.5%.

Index Terms: Deep Audio Classification, Recurrent Neural Networks

1. Introduction

To reverse the trend of declining biodiversity, more accurate and reliable wildlife monitoring tools are crucial. Moreover, wildlife monitoring is an essential part of most conservation efforts and is used to track movement patterns, population demographics and social dynamics, for disease control and to prevent poaching. Obtaining this information is particularly important to save endangered species. In general, wildlife monitoring enables insights into the spatial and temporal scales over which individuals and populations interact [1, 2].

One of the many building blocks for wildlife monitoring is acoustic monitoring. Compared to visual monitoring, it has the advantage of being usable even in areas where high vegetation is present (e.g. dense rain forest) and that it is usually cheaper to deploy [1]. A real-world wildlife monitoring system combines measurements from various sensors such as cameras, autonomous recording units and motion detectors in order to maximize the expressiveness and reliability of the obtained information. Weaknesses is any of these sub-systems leads to a deterioration of the wildlife monitoring system's performance.

In this work, we focus on the classification of primate vocalizations, a task that fits well into the larger picture stated above. In previous work, acoustic primate classification and related tasks are tackled by extracting Mel Frequency Cepstral Coefficients and using them as input to traditional classifiers such as Support-Vector Machines [2, 3, 4] and Multi-Layer Perceptrons [5].

While deep learning is ubiquitous in audio signal processing, it has received little attention in the context of primate vocalization classification. Consequently, we propose a deep learning approach based on bidirectional recurrent neural networks [6] that mimics traditional feature extraction from mel-spectrograms (e.g. functionals) but does so in a fully differentiable way. Moreover, natively supports recordings of variable length. We combine multiple well proven modules such as pooling, normalized softmax [7] and focal loss [8] into a novel approach to primate classification.

Since our approach introduces a variety of hyperparameters, we resort to Bayesian hyperparameter optimization [9] instead of naively tuning them by hand.

We test our approach on a recently introduced dataset for primate classification [10] which was released in context of the ComparE 2021 Challenge [11] and achieve an Unweighted Average Recall (UAR) of 89.4% on the development set and 89.3 on the test set using an ensemble of the five best models found during hyperparameter optimization. Our approach outperforms the best baseline from [11] which achieves a UAR of 87.5% on the test set.

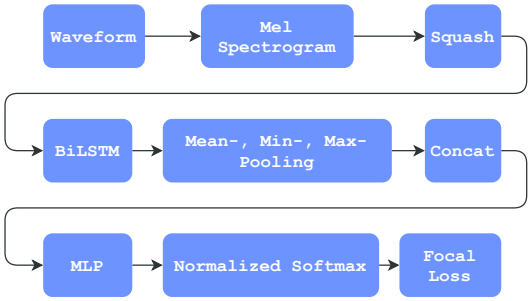


Figure 1: *High-level overview of the proposed workflow.*

2. Proposed Approach

Before the advent of deep learning, it was common to manually extract features of the underlying signal. For these features to remain meaningful and informative, they were usually extracted using a small sliding window over the signal. To obtain a description of the whole signal one would aggregate the extracted features into a few statistical measurements such as mean, median and standard deviation or the minimum and maximum feature values. In fact, this process is still being used today and may serve as a competitive baseline [12, 10].

Our approach is inspired by the described workflow but aims to make the process fully differentiable so that we can leverage the power of neural-networks (NNs) which were shown to outperform the manual feature extraction paradigm in terms of audio

http://dx.doi.org/10.21437/Interspeech.2021-1274

classification performance. Moreover, this shifts the focus from finding a good set of features to finding a suitable NN architecture and the right hyperparameters while following the same conceptual considerations.
In this work, we assume that audio signals are already preprocessed into their time-frequency representation (in most cases a mel-spectrogram). Hence, training dataset $\mathcal{D}$ that includes n mel-spectrograms is given as follows:

$$\mathcal{D} = \{(X_i, L_i) \in \mathbb{R}^{F \times T_i} \times \mathbb{N}\}_{i=1}^{n} \tag{1}$$

where F is the number of frequency-bins, T_i the number of time-frames and L_i and integer that indicates the label.Note that because each mel-spectrogram X_i has a different number of time frames T_i, the mel-spectrograms in $\mathcal{D}$ are of variable length. This is a very common property in real world audio datasets for which we aim to develop a model that natively supports variable length mel-spectrograms.
Lastly, one can interpret each X_i as a sequence of T_i column vectors with a dimensionality of F.
Under this interpretation we can facilitate NNs that were specifically developed to work with sequential data. The best known members of this flavor of NNs are Long Short-term Memory NNs [6] (LSTMs) and Transformers [13]. In our case, we have found LSTMs to be easier to train and we will therefore use LSTM throughout the rest of this work. For an in-depth discussion of LSTMs, see [6].
Here we use LSTMs to compute contextualized latent representations for each input vector (time frame). To reduce the sequence length and thereby also reduce the number of timesteps the gradient must be back-propagated through, we stack two consecutive mel-spectrogram frames on top of each other. Doing so, halves the sequence length and doubles the feature dimension. Since this is only possible for mel-spectrograms with an even number of time frames, we simply repeat the mel-spectrogram's last time frame once if the number of time frames is odd. Under these assumptions the following `einops` [1] code rearranges some mel-spectrogram X_i accordingly:

$$X_i' \in \mathbb{R}^{2F \times (T_i/2)} \tag{2}$$

$$X_i' = \texttt{rearrange}(X_i, \text{"}F\ (t\ k) \rightarrow (k\ F)\ t\text{"},\ k = 2) \tag{3}$$

Here we use a bidirectional LSTM (`BiLSTM`) that extracts a hidden state for each time frame in X_i. The `BiLSTM` simply concatenates the H-dimensional hidden states obtained by feeding X_i in normal and reversed order through an LSTM. Doing so results in T_i highly contextualized hidden states $\in \mathbb{R}^{2H}$, i.e. each hidden state is informed about the hidden states of all the other time frames. Formally:

$$\texttt{BiLSTM} : \mathbb{R}^{2F \times (T_i/2)} \rightarrow \mathbb{R}^{2H \times (T_i/2)} \tag{4}$$

$$\mathbf{x_i} \in \mathbb{R}^{(3*2*H)} \tag{5}$$

$$\tilde{X}_i = \texttt{BiLSTM}(X_i') \tag{6}$$

$$\mathbf{x_i} = [\texttt{mean}(\tilde{X}_i), \texttt{max}(\tilde{X}_i), \texttt{min}(\tilde{X}_i)] \tag{7}$$

$\tilde{X}_i$ is the sequence of hidden states for X_i. In Equation 7 we aggregate the hidden states by using meanpooling, minpooling and maxpooling across the temporal dimension in $\tilde{X}_i$ and concatenate the pooled vectors to obtain statistical descriptions $\mathbf{x_i}$ of X_i. This is done to resemble the feature extraction paradigm we discussed earlier. The pooling operations were chosen due to their simplicity. Future work might investigate upon more sophisticated variants.
We can now simply place a small Multi-Layer-Perceptron (MLP) on top of $\mathbf{x_i}$ to predict class membership.
The most common approach to training a neural classifier is to use the softmax activation function at the MLP's last layer in order to convert from raw scores to probabilities. However, using softmax has the drawback that its definition does not encourage sharp decision boundaries. Additionaly, it promotes well-separated features to have bigger magnitudes which results in an over-reliance on easy examples as shown in [14]. To alleviate this problem, we use the normalized Softmax [14, 7].
Let $W \in \mathbb{R}^{d \times c}$ be the weights of the last linear layer of some NN. Further, let W_i be the ith column vector of W and $v \in \mathbb{R}^d$ the features the NN outputs up until the last layer (often referred to as embedding), then the conditional probability of class j given v is given as follows:

$$P(j \mid v) = -\log\left(\frac{\exp(sim(v^T, W_j)/\tau)}{\Sigma_{i=1}^{c}\exp(sim(v^T, W_i)/\tau)}\right) \tag{8}$$

where sim is the cosine-similarity and τ scales the volume of the hypersphere the embeddings are mapped into. An overview of the most crucial building blocks of our approach can be found in Figure 1.

3. Technical Details and Experiments

In this section we first briefly discuss the dataset we used for primate vocalization classification. Then we provide a technical discussion of the preprocessing and training procedure. Finally, we present the experiments we carried out to investigate upon the effectiveness of our approach.

3.1. Dataset

To study the efficacy of our approach for the classification of vocalisations from primates, we use a recently published dataset [10] that consists of annotated recordings from four different primate species (chimpanzees, guenons, mandrills and redcaps) and background noises (comparable to natural forests) that were recorded in a wildlife sanctuary in Cameroon. Thus, the task is a 5-way classification problem. The dataset loosely represents the class distribution one would obtain during wildlife monitoring and is therefore highly imbalanced. This is made more evident in Figure 2 c).

3.2. Preprocessing and Training Details

3.2.1. Mel-Spectrogram generation

All recordings are down-sampled to 16kHz and converted to mel-spectrograms with a FFT window length of 1024, a hop length of 512 and 128 mel-bands. Subsequently, the mel-spectrograms are squashed according to Equation 2.

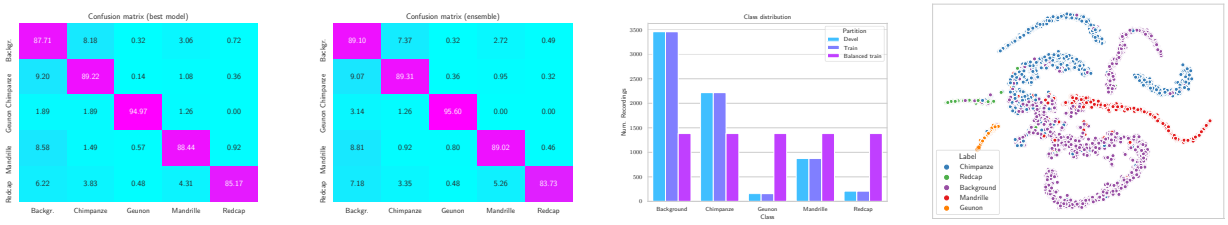
3.2.2. Model details

The squashed mel-spectrogram time frames are directly passed to a LSTM with hidden dimension h and l layers. If $l > 0$ we apply dropout between the layers with probability p_{dropout}. The pooled LSTM output is fed through a MLP with three linear layers ($3h \rightarrow 2h$, $2h \rightarrow 2h$ and $2h \rightarrow h$). We use dropout with p_{drop} and ELU activation functions between each layer. On top of that, a normalized softmax layer [7] ($h \rightarrow 5$) with temperature τ predicts class membership.

[1] `www.einops.rocks`

Table 1: *Comparison of our approach with other baselines. Performance is measured in terms of the average unweighted recall (UAR). The best scores on the development and test set are marked in bold, the second-best are underlined.*

	End2You	Deep Spectrum	OpenSMILE	OpenXBOW	AuDeep	Fusion	Ours 1st	Ours 2nd	Ours 3rd
Devel	72.70	81.3	82.4	83.3	84.6	-	<u>89.1</u>	88.9	**89.4**
Test	70.8	78.8	82.2	83.9	86.1	87.5	88.1	<u>88.7</u>	**89.3**



(a) (b) (c) (d)

Figure 2: *(a) Confusion matrix on the development set for the best model (b) Confusion matrix on the development set for an ensemble of the five best model. (c) Class distribution of the train, development and balanced development set. (d) TSNE plot of the embeddings obtained from the development set.*

Table 2: *Summary of all hyperparameters, their priors and the best hyperparameter settings.*

Hyperparameters	Prior	**Best setting**
Optimization		
learning rate lr	$\log \mathcal{U}(10^{-6}, 10^{-1})$	$3.373 * 10^{-4}$
batch size bs	$32, 64, 96 \ldots 256$	32
temperature τ	$\mathcal{U}(0.01, 1)$	0.6203
weight decay λ	$\mathcal{U}(10^{-6}, 10^{-1})$	0.0578
learning rate decay β	$0.8, 0.82, \ldots 1.0$	0.82
Focal loss γ	$\mathcal{U}\{1, 5\}$	3
Augmentations		
frequency masking p_{fm}	$0, 0.05, \ldots 0.5$	0.0
time masking p_{tm}	$0, 0.05, \ldots 0.5$	0.0
random apply p_{r}	$\mathcal{U}(0, 1)$	/
Model		
LSTM layers l	$\mathcal{U}\{1, 4\}$	2
LSTM hidden dim. h	$32, 64, 96 \ldots 256$	256
dropout p_{drop}	$0, 0.05 \ldots 0.8$	0.05
# trainable parameters		$\approx$ 3.5M

3.2.3. Data Augmentation

For data-augmentation purposes, frequency masking is applied on p_{fm} percent of the mel-bins and time masking on p_{tm} percent of the time frames (but at least one frame). Augmentations are applied with a probability of p_{r}. Augmentations help to synthetically create more "views" of the data and thereby diversify the data at hand. This often leads to less overfitting and increased classification performance [15, 16].

3.2.4. Optimization

Since naively training deep neural networks on highly imbalanced data may often result in the network taking a shortcut and predicting only the most frequent classes, we randomly resample (up and down-sampling with replacement) the dataset in each training epoch to follow a uniform class distribution. However, we observed that at least in the early stages of training, using standard cross-entropy causes easy examples to overwhelm the loss and dominate the gradient. Note that this does not apply equally to all datasets but we have found it to be the case for the datset discussed in Section 3.1. Hard examples are more informative and are crucial for good generalization and classification performance. Hence, we use the focal loss [8] instead of the standard cross-entropy to train the network. On a high-level, the focal loss down-weighs easy examples and up-weighs hard examples. The larger the focusing parameter γ, the more the focal loss focuses on hard examples and the less it focuses on easy examples. In our case, this considerably speeds up training and gives a small performance boost.
The network is optimized using AdamW [17] with weight decay λ, learning rate lr and a batch size of bs. The learning rate is multiplied by β each epoch. Training is carried out for at most 300 epochs.

3.2.5. Hyperparameter Optimization

In total, there are 12 hyperparameters $(lr, bs, \tau, \lambda, \beta, \gamma, p_{\text{fm}}, p_{\text{tm}}, p_{\text{r}}, l, h, p_{\text{drop}})$ we need to tune. Finding a good set of hyperparameters by hand is time consuming and may lead to lead unsatisfactory solutions as the process is often unstructured. Thus, we resort to bayesian optimization to find a suitable set of hyperparameters. That is, we use a a Tree-Structured Parzen Estimator (TPE) [9] to optimize the hyperparameters on the development set. We optimize over 400 trials and prune a single trial if the trial's best intermediate result is worse than median of intermediate results of previous trials at the same epoch [18]. The prior for each hyperparameter was determined through initial experiments.
A summary of all hyperparameters, their priors as well as the best configuration can be found in Table 2.

3.3. Comparision with Baselines

To evaluate the performance of our approach we compare it with the best baselines taken from [11]; namely:

i OpenSMILE [12]
A set of 6373 handcrafted features resulting from the computation of statistics over low-level descriptor (LLD) contours.

ii Deep Spectrum [19]
The mel-spectrograms are converted to RGB images and subsequently fed trough a DenseNet121 [20]. The activations after the average pooling layer are used to obtain a 2048 dimensional feature vector for each recording.

iii AuDeep [21]
In the first stage four recurrent Autoencoders are independently trained on mel-spectrograms where the power levels were clipped below the thresholds -30dB, -45dB, -60dB and -75dB. The features from the final hidden states from each autoencoder are concatenated to form a feature vector for each recording.

iv OpenXBOW [22]
For each recording 65 LLDs and their deltas based on the COMPARE [23] feature set are extracted an quantized using two codebooks of size 1000, one for the recordings and one for the deltas. The termfrequency weighted histograms of both representations are concatenated to form the final representation.

v End2You [24]
Three layers of a 1d convolutional neural network extract features from the raw waveform that are subsquently fed through Gated Recurrent Units to classify recordings. This model is trained end to end.

vi Fusion [11]
This is an ensemble of the five aforementioned approaches that predicts class membership via majority voting.

Unless mentioned otherwise a SVM is used for classification for the approaches presented above. For a more detailed discussion of the baselines as well as their hyperparameters, see [11].
Concerning the proposed model, we evaluate the following three variants on the train and test set:

i Ours $1^{\mathbf{st}}$
Here we use the best model we found during hyperparameter optimization according to Table 2.

ii Ours $2^{\mathbf{nd}}$
Here we use the second best model we found during hyperparameter optimization according to Table 2.

iii Ours 3^{rd}
Inspired by the Fusion baseline, here we use the five best models found during hyperparameter optimization and apply majority voting to obtain a final class membership prediction.

Since the dataset is highly imbalanced and consists of multiple classes, the *Unweighted Average Recall* (UAR) is used as an evaluation measure. The UAR is computed by taking the mean of the recall of each class where the recall is the number of correctly classified examples of a class divided by the total number of examples of that class.
The confusion matrices for the predictions on the development set are depicted in Figure 2 (a) and Figure 2 (b).

3.4. Manifold Analysis

Since the use of the normalized softmax allows for an intuitive geometric interpretation, i.e. examples are mapped into a hypersphere with a fixed volume, we investigate upon the quality of the learnt embeddings (activations from the MLP's penultimate layer) by visualizing them using TSNE [25] (Figure 2 (d)) . Each class is given a distinct color and embeddings are computed on the development set.

4. Results

When comparing all three variants of our model in Table 1, we can clearly observe that the ensemble achieves the best UAR on the development set (89.4% vs. 88.9% and 89.1%) and test set (89.3% vs. 88.1% and 88.7%). Diversity between the different models in the ensemble is introduced through the usage of different hyperparameters. Through majority voting, mistakes of a single model are less severe and can be compensated by the other members of the ensemble. This is made evident when comparing the diagonals in Figure 2 (a) and Figure 2 (b). The ensemble improves upon the classification accuracy for all but the redcap class. The drop in performance for the redcap class stems from misclassifying redcaps as mandrilles and background noises more often. For all our models, redcap is the class which is the most difficult to classify. We attribute this to a worse signal-to-noise-Ratio on of redcap recordings since they are a quieter species when compared to e.g. chimpanzees. Vocalizations from mandrills, guenons and chimpanzees are most often mistaken for background noise (8.81%, 3.14% and 9.07%). On the other hand, background noise is most often confused with chimpanzees. When inspecting the TSNE projected embeddings in Figure 2 (d), these findings are confirmed. Most points are easily separable. Occasionally, points from other classes land in the area where the background class dominates. We observe that many of these mistakes can be explained by recordings where the vocalizations are difficult to hear, i.e. vocalizations are made in the far distance from the recording unit. Hence, in future work one might might also incorporate background noise removal tools.
However, our approach considerably outperforms the best baselines. It achieves a UAR of 89.4% vs 84.6% (AuDeep) and 89.3% vs 87.5% (Fusion) on the development and test set, respectively. These results clearly indicate that our approach is well suited for primate vocalization classification.
In terms of the best hyperparameters from Table 2 we can see that the temperature parameter τ is considerably higher than recommended in the original work (0.05). The normalized softmax was originally proposed for face verification where the number of classes is orders of magnitude larger and consequently a more compact hypersphere suffices in our case. Another interesting observation is that data augmentation was completely turned off and dropout was set to a very low value ($\tau = 0.05$). This suggests these augmentations act as a source of confusion rather than promote learning. We hypothesize that these augmentations are better suited for the inductive bias of convolutional neural networks than for LSTMs.

5. Conclusion

We proposed a deep and recurrent architecture based on recurrent neural networks for primate vocalization classification that manages to outperform all of its direct competitors. In future work, we plan to investigate upon the effects of pre-training the network with auxiliary data from various domains [26].

6. References

[1] D. T. Blumstein, D. J. Mennill, P. Clemins, L. Girod, K. Yao, G. Patricelli, J. L. Deppe, A. H. Krakauer, C. Clark, K. A. Cortopassi *et al.*, "Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus," *Journal of Applied Ecology*, vol. 48, no. 3, pp. 758–767, 2011.

[2] S. Heinicke, A. K. Kalan, O. J. Wagner, R. Mundry, H. Lukashevich, and H. S. Kühl, "Assessing the performance of a semi-automated acoustic monitoring system for primates," *Methods in Ecology and Evolution*, vol. 6, no. 7, pp. 753–763, 2015.

[3] P. Fedurek, K. Zuberbühler, and C. D. Dahl, "Sequential information in a great ape utterance," *Scientific Reports*, vol. 6, no. 1, pp. 1–11, 2016.

[4] D. J. Clink, M. C. Crofoot, and A. J. Marshall, "Application of a semi-automated vocal fingerprinting approach to monitor bornean gibbon females in an experimentally fragmented landscape in sabah, malaysia," *Bioacoustics*, vol. 28, no. 3, pp. 193–209, 2019.

[5] A. Mielke and K. Zuberbühler, "A method for automated individual, species and call type recognition in free-ranging animals," *Animal Behaviour*, vol. 86, no. 2, pp. 475–482, 2013.

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

[7] A. Zhai and H.-Y. Wu, "Classification is a strong baseline for deep metric learning," *arXiv preprint arXiv:1811.12649*, 2018.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[9] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *25th annual conference on neural information processing systems (NIPS 2011)*, vol. 24. Neural Information Processing Systems Foundation, 2011.

[10] J. A. Zwerts, J. Treep, C. S. Kaandorp, F. Meewis, A. C. Koot, and H. Kaya, "Introducing a central african primate vocalisation dataset for automated species classification," *arXiv preprint arXiv:2101.10390*, 2021.

[11] 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, and 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*. Brno, Czechia: ISCA, September 2021, to appear.

[12] 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*, 2010, pp. 1459–1462.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[14] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049.

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

[16] S. Illium, R. Müller, A. Sedlmeier, and C. Linnhoff-Popien, "Surgical Mask Detection with Convolutional Neural Networks and Data Augmentations on Spectrograms," in *Proc. Interspeech 2020*, 2020, pp. 2052–2056. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1692

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[19] 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 *Proc. Interspeech 2017*, 2017, pp. 3512–3516. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-434

[20] 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.

[21] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, and B. Schuller, "audeep: Unsupervised learning of representations from audio with deep recurrent neural networks," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6340–6344, 2017.

[22] M. Schmitt and B. Schuller, "Openxbow: introducing the passau open-source crossmodal bag-of-words toolkit," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3370–3374, 2017.

[23] F. Weninger, F. Eyben, B. W. Schuller, M. Mortillaro, and K. R. Scherer, "On the acoustics of emotion in audio: what speech, music, and sound have in common," *Frontiers in psychology*, vol. 4, p. 292, 2013.

[24] P. Tzirakis, S. Zafeiriou, and B. W. Schuller, "End2you–the imperial toolkit for multimodal profiling by end-to-end learning," *arXiv preprint arXiv:1802.01115*, 2018.

[25] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[26] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020.