DETECTING PRESENCE OF SPEECH IN ACOUSTIC DATA OBTAINED FROM BEEHIVES

Pascal Janetzky[1], *Padraig Davidson*[1], *Michael Steininger*[1], *Anna Krause*[1], *Andreas Hotho*[1]

[1] Computer Science Dept., University of Würzburg, Würzburg, Germany
{janetzky,davidson,steininger,anna.krause,hotho}@informatik.uni-wuerzburg.de

ABSTRACT

Sound recorded from beehives is important to understand a colony's state. This fact is used in the *we4bee*[1] project, where beehives are equipped with sensors (among them microphones), distributed to educational institutions and set up to record colony characteristics at the communication level. Due to data protection laws, we have to ensure that no human is recorded besides the bees' sound. However, detecting the presence of speech is challenging since the frequencies of human speech and the humming of bees largely overlap. Despite having access to only a limited amount of labeled data, in this initial study we show how to solve this problem using Siamese networks. We find that using common convolutional neural networks in a Siamese setting can strongly improve the ability to detect human speech in recordings obtained from beehives. By adding train-time augmentation techniques, we are able to reach a recall of up to $100\,\%$, resulting in a reliable technique adhering to privacy regulations. Our results are useful for research projects that require written permits for acquiring data, which impedes the collection of samples. Further, all steps, including pre-processing, are calculated on the GPU, and can be used in an end-to-end pipeline, which allows for quick prototyping.

***Index Terms*—** Audio classification, Siamese networks, Speech detection, Deep learning

1. INTRODUCTION

With the introduction of the General Data Protection Regulation (GDPR) [1] in the European Union, publicly recording sound at any time, even for scientific purposes, requires written agreements and the immanent possibility to stop recording from the user's side. Smart home devices ensure this by only recording data after a signal word. Uploading recorded data is allowed, if no speaker can be recognized individually from the recording (i.e., distortion) or the recording device ensures no privacy concerned data is contained in the file (i.e., speech detection). Despite these challenges, it is desirable to use this data since sound is rich in information and enables more fundamental understandings of communicating organisms, such as bee colonies.

In this paper we focus on the task of detecting the presence of speech in audio signals obtained from beehives of our *we4bee* project. The we4bee project distributed over 100 *Top Bar Hives* (TBH), mainly to educational institutions in Germany. Within these "smart" beehives (see Figure 1), we can continuously monitor the state of the colony, and even analyse the auditory communication of the bees. Bee monitoring systems are an important tool for apiarists [2], and especially sound recordings are inevitable to fully understand the beehive's state [3]. Especially in swarming prediction, sound recordings have shown great success, in contrast to simpler monitoring, such as observing the temperature [4], which we used in prior work [5]. Our contributions are as follows: 1) An algorithm which achieves high recall, allowing allows us to detect human speech in a challenging environment, 2) a privacy-regulation conform recording approach without distorting the signals, and 3) to the best of our knowledge, the first study on human speech detection for smart apiculture.

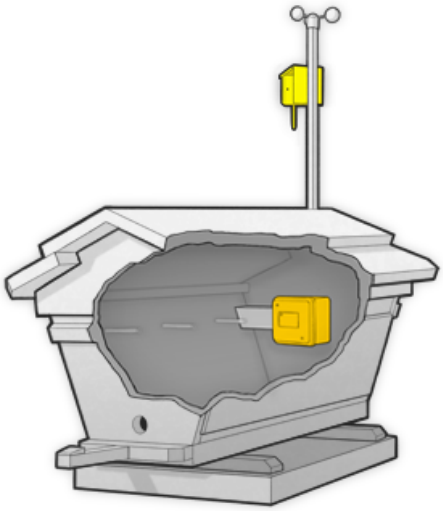

Figure 1: Sensor placement of the TBH. This cutaway view of the beehive shows the sensor placement of all sensors. The placement of the two microphones (one inside, one outside), is highlighted in yellow. Both mono sources are merged as a stereo-signal and uploaded as such. Image taken from we4bee[1]

2. RELATED WORK

As one of the first, [6] uses a convolutional neural network (CNN) to classify sound. This model is trained on the Environmental Sound Classification dataset (ESC-50) [7], consisting of 50 classes with 40 samples each. All samples are $5\,\mathrm{s}$ long and split into overlapping spectrogram segments. The length of the extracts is $950\,\mathrm{ms}$ (short variant) and $2.3\,\mathrm{s}$ (long variant), Lastly, the probabilities of all segment-level predictions are taken into account to obtain a final prediction. The authors find that using the longer samples improves the classification accuracy, reaching a score of $64.5\,\%$.

In 2017, Stowell et al. hosted the Bird Audio Detection challenge (BADc) [8]. For this challenge, the task is to detect the presence

[1] https://we4bee.org

Table 1: Overview of the dataset.

Dataset	negative samples	positive samples
Train	119	16
Validation	27	3
Test	25	10
Overall	171	29

of any bird sound in short (mostly 10 s) audio recordings. The development data and test data come from different sources, which requires methods that generalize to unseen recording conditions. The highest scoring approach uses CNNs on spectrogram inputs and cyclical time shifting to classify the data [9]. With their submission named `bulbul`, which works on spectrograms calculated over 14 s, they reach an area under ROC score (AUROC) of 88.7 %.

Building on this architecture [10] classify sound excerpts of audio data recorded from beehives. The excerpts are labeled as containing bee-related sound or containing external sounds. Mel spectrograms are then calculated and used as input features to the network, with random pitch and time shifting augmenting the training data. Using a wide receptive field of 30 s, the classifier network reaches an AUROC score of 90.1 %.

In [11], Manocha et al. use a Siamese network [12] to compare the similarity of input pairs. On several audio datasets, among them ESC-50, they study the problem of retrieving semantically similar audio clips. In their setup, log-scaled spectrograms are calculated for the data, and feed to Siamese networks to obtain dense embedding vectors. Using a k nearest neighbor search on the embeddings, the authors achieve a mean precision of up to 78.4 %, indicating that the learned representations successfully capture similarity.

In this work, we use the mentioned CNNs in a Siamese setting to detect speech presence in audio recordings obtained from beehives.

3. DATASET

In order to enable continuous sound monitoring with we4bee, we obtained permission to record audio at one location. Since the data for this feasibility study currently originates from a single beehive only, we plan to add more recording stations in the future. Recording started in May 2020, and is still running. For the purpose of this paper, we used one day each in August and September for training and validation data, and a separate day in October for final testing.

Each audio sample is recorded at 44.1 kHz, 16 bits resolution for 60 s. We let one voluntary annotator manually label a small, random amount of these recordings on file level. Each sample is hand-labeled as class 0, no human speech present (negative samples), or as class 1, meaning the presence of human speech (positive samples). Additionally, to collect more positive samples, we followed a simple heuristic: Every time speech was detected in the current sample, we searched in a ± 10 min interval for more positive samples. An exemplary spectrogram of a recording with speech segments can be seen in Figure 2. Table 1 lists the complete dataset statistics.

This dataset poses two challenges: First, human speech is only sparsely present, both in terms of absolute numbers of samples and relative time within the samples. Analysing shorter windows would yield more samples but also lead to higher class imbalance towards class 0 which is why we kept the 60 s windows from the recordings. These samples come from a broad range of situations, from children playing far away (hardly audible) to adults talking next to the beehives (clearly audible). Second, the general limited number of annotated samples increases the difficulty. Large supervised audio datasets such as the ones used for the BADc have 24 000 annotated samples and more. In contrast, our labeled dataset only consists of 200 samples in total. The challenge is therefore to learn a model to classify a diverse, unbalanced, and small dataset.

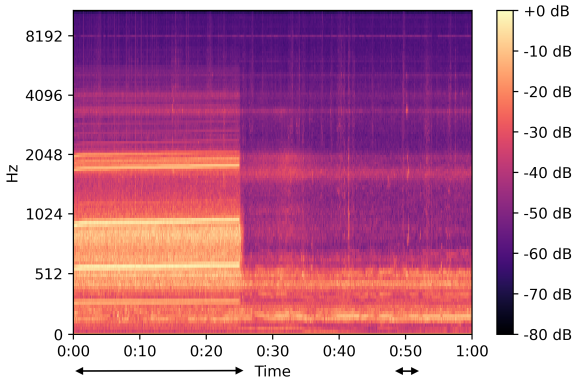

Figure 2: Sample audio file containing speech. A high noise level is created by a motor moving until 25 s, marked by the large arrow. The speech, located at 50 s (small arrow), is covered by the background noise and not visually discernible.

4. METHODS

4.1. Siamese Neural Networks

Siamese neural networks [12] are a class of networks that learn the similarity of an input pair. Each sample is fed into the neural network to obtain a dense representation, termed embedding. A distance metric (i.e., Euclidean distance) is then used to calculate the distance between the embeddings. To interpret the result, one often applies a sigmoid activation function, forcing the values to lie between $[0, 1]$. An output close to 0 indicates highly similar samples, conversely values near 1 indicate high dissimilarity. The term "siamese" refers to the fact that the same set of weights is used to calculate each embedding of the input pair.

We train the Siamese networks to minimize the distance between audio pairs from the same class and to maximize the distance for opposite pairs. For this, we randomly draw an audio sample, and pair it both with a random sample from the same class (this pair is labeled as 0), and with a random sample from the opposite class (labeled as 1). The learned embeddings are used to train a kNN classifier [13, 14]. To obtain class predictions for the test samples, we first extract embeddings for the test data and then query the classifier.

4.2. Base Neural Networks

Motivated by the frequent usage of mel-scaled spectrograms (Mel spectrograms from now on) as input features, we utilize the two networks briefly introduced in Section 2 (the networks trained on the ESC dataset and submitted to the BADc, respectively) , and one published in an introductory article on sound classification with CNNs. We adapt each network to accept the raw audio and compute the

Mel spectrograms as part of the forward pass, using the `kapre` [15] package. With this modification, the computation is accelerated by the GPU, making a separate on-CPU pre-processing step obsolete.

Saeed The first network is from work published by Aaqib Saeed [16], which we name `Saeed` after its author. The input Mel spectrogram features are merged with their deltas, local estimates of the derivative that capture the transition dynamics of sound. The resulting two-channel representation is then framed, splitting the vector into small excerpts. These excerpts are processed by four sets of the following stack: convolution → batch normalization [17] → ReLU activation [18, 19] → max pooling. Additionally to the proposed architecture, we add a drop dropout [20] layer before each pooling operation and replace the max- with global max-pooling in the last stack. Afterwards, we add a further dropout operation prior to two ReLU dense layers. We modify the final layer to have a single neuron only and replace the softmax with a sigmoid activation.

Bulbul The highest-scoring submission in the aforementioned BADc is a CNN named Bulbul [9]. The key feature of this network is its wide receptive field, enabling it to find short local events. The input Mel spectrograms are normalized with batch normalization, followed by four sets of the following stack: convolution → leaky ReLU → max pooling layers. The output is then flattened, and followed by two blocks of dropout → dense → leaky ReLU layers. The final layer has a single neuron with sigmoid activation.

ESC The third model we used is the network introduced for the ESC dataset (see Section 2 and [7, 6]), which we call `ESC`. The Mel spectrogram is stacked with its delta features. The computation of these features is followed by a single stack of convolution → dropout → max-pooling. After another convolution and max pooling layer, the tensor is flattened and processed by two dense layers, each with dropout. The output activation is a binary sigmoid.

5. EXPERIMENTAL SETUP

For all experiments, we utilize the Adam optimizer [21] with default parameter values: A learning rate of 0.001, β_1 of 0.9, and β_2 of 0.999. Each model is trained and evaluated on our dataset five times, and the results are averaged. We use `librosa` [22] to load and downsample the audio to $22\,050\,\mathrm{Hz}$. As a metric function, we follow [9] and report the area under the receiver-operating curve (AUROC) score [23, 24, 25]. This metric first calculates recall versus fall-out at various threshold levels, yielding the ROC curve. The area under this curve captures the performance of a classifier in a single metric. A value of 0.5 equals random guessing, a value of 1.0 is equal to a perfect classifier. For our imbalanced two-class dataset, we use the AUROC metric, as opposed to misleading accuracy scores. Further, for the baseline networks, we interpret the binary output as class 1 if it is above the default threshold of 0.5, and as class 0 otherwise. Similarly, for the Siamese networks the default threshold is 0.5. Additionally, we report the recall for class 1 (speech). Since samples of class 1 might contain sensitive information, we are interested in particularly high recall. Therefore, regarding privacy, a false negative is more severe than a false positive.

5.1. Baselines

We initially tried various clustering algorithms, which scored only slightly better than random guessing and where thus not further evaluated. We therefore used the three CNNs introduced in Section 4.2 as baseline networks, without using a Siamese setting. For the training we follow the approach of [9]: We train the network for 100 epochs, use EarlyStopping [26] with a patience of 20 epochs and a batch size of 16. We reduce the learning rate by a factor of 10, if the area-under-curve score has not improved for ten consecutive epochs.

Table 2: Baseline networks on the test set, averaged over five runs.

Network	AUROC	Recall speech
Saeed	0.6125 ± 0.0400	0.0
Bulbul	0.7525 ± 0.0100	0.0
ESC	0.5017 ± 0.0300	0.0

5.2. Siamese Network

We use all networks introduced in Section 4.2 in a Siamese setting. For all networks, we replace the final hidden dense and any subsequent layer with a single dense layer of 128 neurons, which is interpreted as the embedding vector. All embedding vectors are normalized using $L2$ normalization.

We train our Siamese network for 100 epochs and use EarlyStopping on the validation AUROC score with a patience of 20 epochs. Since our audio samples are quite large, we use small batch sizes for the training. The default value is 4, which means that four pairs are created, using 8 individual audio samples in total.

To obtain better scores, we also try train-time augmentations on the raw audio. For this, we used the `audiomentations`[2] package. With a probability of $50\,\%$ each, we add gaussian noise, use time-shifting of $\pm 30\,\mathrm{s}$, and shift the pitch ± 2 semitones.

Further, we experiment with more epochs (500) and try different EarlyStopping offsets (300 and 500), which is the number of guaranteed update steps done before the counter begins. We try different values for the number of neighbors $k \in \{1, 3, 5\}$.

6. RESULTS & DISCUSSION

As summarized in Table 2, of the baseline networks, the `ESC` network reaches the lowest AUROC score, with $50.17\,\%$. The next best network, `Saeed`, scores more than 10 points higher, reaching $61.25\,\%$, and the best network, `Bulbul`, reaches $75.25\,\%$. However, our primary indicator of performance, the recall of the speech class, is drastically lower; all three networks achieve a recall of 0. Using a Siamese setting, we can improve the score for all three networks (Table 3), and our strongest candidate reaches $94\,\%$ speech recall and an AUROC score of $96.88\,\%$. The `ESC` network does not benefit from a Siamese setup, it reaches only slightly better scores compared to the non-Siamese setting. Generally, a higher number of neighbors when classifying the test data via kNN improves scores.

When using train-time augmentations, as described in Section 5.2, we observe two primary effects: First, the scores are worse compared to no augmentation, as shown in Table 4. Secondly, training takes considerably longer since the computation is done on the CPU. On examination of the validation AUROC curve we noticed that the scores heavily oscillate in the beginning. Before the scores stabilize and increase, either the EarlyStopping patience prematurely terminates the run or the maximum number of epochs (100) is reached.

These instabilities can be overcome by using an offset for EarlyStopping and by training for more epochs. We find that an offset of 300 is sufficient when training for 500 epochs. Combined

[2] `https://git.io/JcQJQ`

Table 3: Scores for the Siamese networks on the test dataset, averaged over five runs, without augmentation.

	Saeed		Bulbul		ESC	
k	AUROC	Recall speech	AUROC	Recall speech	AUROC	Recall speech
1	0.8633 ± 0.1600	0.76 ± 0.31	0.9500 ± 0.0500	0.90 ± 0.01	0.5317 ± 0.0500	0.18 ± 0.05
3	0.8988 ± 0.1400	0.76 ± 0.31	0.9696 ± 0.5000	0.90 ± 0.10	0.5258 ± 0.0600	0.18 ± 0.05
5	0.9046 ± 0.1300	0.74 ± 0.29	0.9688 ± 0.0500	0.94 ± 0.10	0.5537 ± 0.0600	0.04 ± 0.05

Table 4: Scores for the Siamese networks on the test dataset, averaged over five runs, with augmentation. Augmenting the training data stops the training prematurely, as explained in Section 6.

	Saeed		Bulbul		ESC	
k	AUROC	Recall speech	AUROC	Recall speech	AUROC	Recall speech
1	0.6092 ± 0.1900	0.26 ± 0.34	0.4867 ± 0.0200	0.04 ± 0.05	0.5192 ± 0.0400	0.18 ± 0.11
3	0.6775 ± 0.1500	0.26 ± 0.34	0.4838 ± 0.5000	0.04 ± 0.05	0.5254 ± 0.0600	0.18 ± 0.12
5	0.6837 ± 0.1400	0.28 ± 0.31	0.5083 ± 0.0500	0.04 ± 0.09	0.4908 ± 0.1000	0.00 ± 0.00

Table 5: Scores for the Siamese networks on the test dataset, averaged over five runs, with augmentation. The training is done with an offset of 300 epochs for EarlyStopping. Compared to Table 4, using such an offset can lead to improved results.

	Saeed		Bulbul		ESC	
k	AUROC	Recall speech	AUROC	Recall speech	AUROC	Recall speech
1	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.495 ± 0.020	0.04 ± 0.09
3	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.51 ± 0.05	0.04 ± 0.09
5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	0.52 ± 0.07	0.06 ± 0.13

with augmentation, we reach perfect AUROC and recall scores (100 %) with both `Bulbul` and `Saeed`. These perfect scores indicate overfitting, which might be due to the relatively small dataset. We plan to address this in further research by increasing both the dataset's size and diversity. Nonetheless, for this initial study the results are very encouraging. A sample embedding is visualized with t-SNE [27, 28] in Figure 3, which shows how the classes are separated well in space. As before, the `ESC` network does not benefit. We suspect that this is due to the relatively shallow architecture, which may prevent the network from learning meaningful features.

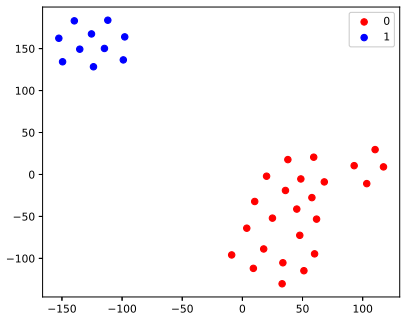

Figure 3: t-SNE plot (perplexity=12.5) of the learned embeddings, obtained from the Siamese `Bulbul` network for the test dataset. The network can perfectly separate the data.

7. CONCLUSION

Sound recordings of beehives are an important source of information for modeling the behavior of bees but it is not trivial to record publicly accessible hives in a privacy preserving manner. To allow for GDPR-compliant sound recordings of bee colonies we considered multiple approaches for presence of speech detection, allowing us to detect and remove human speech before storing the sound data. In this initial feasibility study, we used three convolutional neural networks to detect presence of speech in these challenging recordings. By using them in a Siamese setting, we achieve high recall. Motivated by the good results, we then used augmentation techniques and increased the number of epochs to achieve perfect recall and AUROC scores. For our small datasets, these results are promising, but open the opportunity for future work in several directions:

First, our current dataset is limited to recordings from a single beehive. In prospective work, it can be enriched with recordings from more beehives. This would capture more locations and characteristics, allowing to better examine the ability to generalize. Second, the code can be adapted for on-device inference. Currently, for the beehives we have permission to record, we upload the audio data to the cloud. Only then do we check for the presence of speech. However, this step can be greatly simplified by running the detection directly on the Raspberry Pi which powers all beehive sensors.

Acknowledgements All audio recordings were taken from Prof. Hotho's TBH with his written consent allowing the recording. We thank Albin Zehe for his helpful commentary on this paper.

8. REFERENCES

[1] General data protection regulation (gdpr). European Commission. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[2] S. Cecchi, S. Spinsante, A. Terenzi, and S. Orcioni, "A smart sensor-based measurement system for advanced bee hive monitoring," *Sensors*, vol. 20, no. 9, p. 2726, 2020.

[3] A. Terenzi, S. Cecchi, and S. Spinsante, "On the importance of the sound emitted by honey bee hives," *Veterinary Sciences*, vol. 7, no. 4, p. 168, 2020.

[4] S. Ferrari, M. Silva, M. Guarino, and D. Berckmans, "Monitoring of swarming sounds in bee hives for early detection of the swarming period," *Computers and electronics in agriculture*, vol. 64, no. 1, pp. 72–77, 2008.

[5] P. Davidson, M. Steininger, F. Lautenschlager, K. Kobs, A. Krause, and A. Hotho, "Anomaly detection in beehives using deep recurrent autoencoders," in *Proceedings of the 9th International Conference on Sensor Networks (SENSORNETS 2020)*, no. 9. SCITEPRESS – Science and Technology Publications, Lda., 2020, pp. 142–149.

[6] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[7] ——, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[8] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 368–380, 2019.

[9] T. Grill and J. Schlüter, "Two convolutional neural networks for bird detection in audio signals," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 1764–1768.

[10] I. Nolasco and E. Benetos, "To bee or not to bee: Investigating machine learning approaches for beehive sound recognition," *arXiv preprint arXiv:1811.06016*, 2018.

[11] P. Manocha, R. Badlani, A. Kumar, A. Shah, B. Elizalde, and B. Raj, "Content-based representations of audio using siamese neural networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 3136–3140.

[12] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, pp. 737–744, 1993.

[13] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review/Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.

[14] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[15] K. Choi, D. Joo, and J. Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," in *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning*. ICML, 2017.

[16] A. Saeed. (2016) Urban Sound Classification. [Online]. Available: http://aqibsaeed.github.io/2016-09-24-urban-sound-classification-part-2/

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[19] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

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

[22] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[23] D. K. McClish, "Analyzing a portion of the roc curve," *Medical decision making*, vol. 9, no. 3, pp. 190–195, 1989.

[24] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[25] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.

[26] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

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

[28] 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.