TUC Media Computing at BirdCLEF 2022: Strategies in identifying bird sounds in a complex acoustic environments

Arunodhayan Sampathkumar[1], Danny Kowerko[1]

[1]Technische Universität Chemnitz , Str. der Nationen 62, 09111 Chemnitz

Abstract

Birds play an essential role in monitoring the quality of the environment, pollution, and climate changes. Advancements in convolutional neural networks allow us to recognize birds, hence assisting researchers in monitoring the bird population and biodiversity in an ecosystem. This research paper proposed a pipeline that deals with data augmentation strategies and Sound Event Detection (SED) methods to recognize birds in complex environments. Our proposed solution achieved 69th rank among 807 teams at the BirdCLEF 2022 challenge hosted in Kaggle.

Keywords

Acoustic environment, Convolutional Neural Network, Sound Event Detection, Data Augmentation, Ensemble, Birdcall Identification

1. Introduction

The BirdCLEF 2022 challenge proposes to identify rare/endangered bird species from soundscape recordings recorded in Hawaii's location. The challenge was hosted from February 15, 2022 to May 24, 2022 [1].

Dataset- The training dataset was populated with 14,853 short audio recordings of 152 bird species uploaded by users of Xeno-canto [2]. Additionally, the training data was supported with metadata containing the information of recording location, type of birds chirp (bird call or song), etc. The test set was populated with approximately 5,500 recordings each 1 minute long to be used for scoring, 21 endangered bird species were used for scoring on the test set (**note** that participant cannot access these audios). These audio files are sampled to 32 kHz in Ogg format. The preprocessing and the process of generating Mel-spectrograms are discussed in section 3.1.

Problem- The training set consists of long Ogg recordings with multiple bird species present, which denotes multilabel classifications. The competition evaluation (F1 score) relies on identifying 21 rare bird species (not 152 bird species), some bird species contain only one

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

arunodhayan.sampath-kumar@informatik.tu-chemnitz.de (A. Sampathkumar); danny.kowerko@informatik.tu-chemnitz.de (D. Kowerko)
https://arunodhayan.github.io/ (A. Sampathkumar); https://www.tu-chemnitz.de/informatik/mc/professor.php.en (D. Kowerko)
0000-0002-2654-7050 (A. Sampathkumar); 0000-0002-4538-7814 (D. Kowerko)

CEUR Workshop Proceedings (CEUR-WS.org)

recording sample which is 1 to 3 seconds in length as presented in the Figure 1. The most challenging and motivation of the competition are the weakly labeled train data, and there are multiple distribution domain shifts present, namely shifts in input space, shifts in the prior probability of labels, and shifts in the function which connects train and test recordings. Domain shifts in this competition are large differences in data characteristics between train (clean recordings) and test (noisy recordings) generalizing models on unseen data difficult, as visualized in the Figure 2.

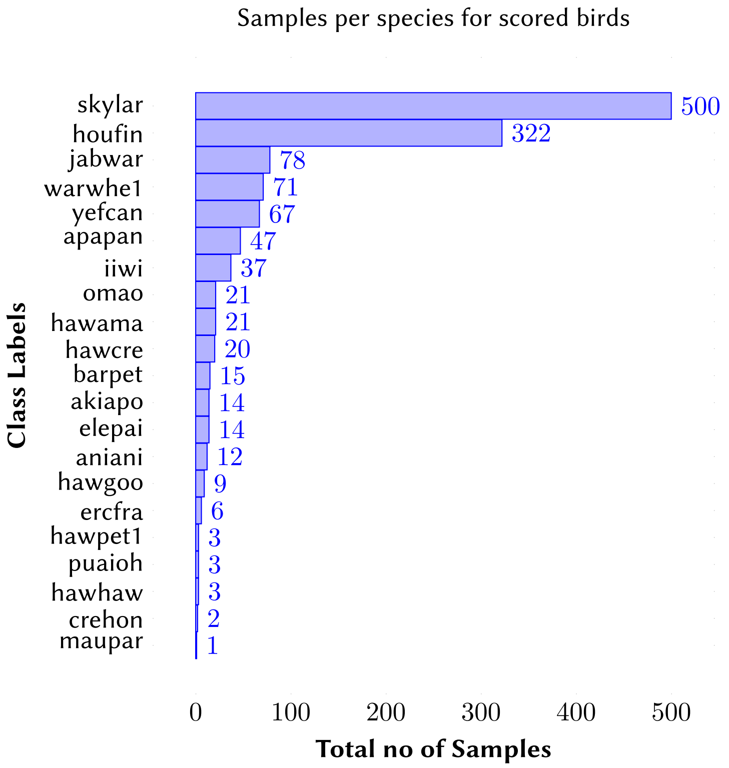
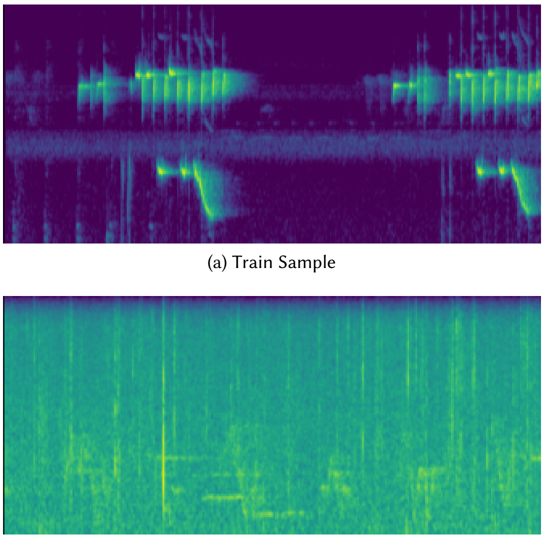


Figure 1: Distribution of sample frequencies of 21 endangered bird species in Hawaii location.

Standard terms are defined which are used in this research paper:

1. LB - Leadership Board
2. PR-LB - Private Leadership Board (84% of test data (4620 recordings))
3. PU-LB - Public Leadership Board (16% of test data (880 recordings))
4. CV-BG - Cross Validation samples embedded with background noise (pink noise)
5. CV - Simple Cross Validation Samples
6. SED - Sound Event Detection
7. EfficientNet-b0-ns - EfficientNet-b0-Noisy Student



(a) Train Sample

(b) Test Sample

Figure 2: Illustration of the domain shift between a) train set with low noise and b) test set with high noise level.

2. Related work

Previous BirdCLEF (2020, 2021) challenges proposed problems related with recognition of bird events in a complex acoustic environments [3],[2]. Researchers from the previous challenges proposed deep learning techniques based on Convolutional Neural Network (CNN) to recognize birds on a large scale [4]. State-of-the-art CNNs which performed better to recognize birds when combined with different augmentation strategies, metadata information (latitude, longitude, type of call, etc.) improves the performance [4],[5], [6]. Furthermore, Pretrained Audio Neural Network (PANN) from the DCASE 2021 audio challenge provided a better generalization capability in predicting the audio events when compared with previous audio-related CNN models [7]. Sound Event Detection (SED) approaches usually employs two-dimensional CNNs to extract useful time and frequency information from the given audio sample, then combine the information with an attention head to predict the bird events, here the attention head can be either a CNN or Recurrent Neural Network (RNN) [8], [9], [10], [11].

3. Proposed methods

The section explains the main components of our solution to the BirdCLEF 2022 Birdcall Identification Challenge. The section comprises dataset preprocessing, data augmentation, Training setup, and Model architecture.

3.1. Data preprocessing

The raw audio files from the training set of random 5 seconds length were converted to Mel-spectrograms using *torchaudio* library [12]. The following parameters were selected to generate Mel-spectrograms: sample rate 32 kHz, Mel bins 224, minimum frequency 0 Hz, maximum frequency 16000 Hz, fast Fourier transform window (n-fft) 2048, and hop-length to 512. The visual inspection of Mel-spectrograms was performed on a random 5 seconds clip to investigate the non-bird events and hence determine the threshold to have a noise (non-bird events) or silence. The Mel-spectrograms containing noise or silence were removed and treated as noise for the augmentation process. The training data set was split into 5 different stratified folds using Scikit-learn library [13].

3.2. Data augmentation

We implemented 8 different augmentation techniques to improve the generalization and robustness of the models. The following augmentation techniques were applied to the raw audio recordings.

1. **Gaussian noise**- Adding Gaussian noises with randomly chosen weights to our audio signal and re-normalize the results.
2. **Pink noise**- The background noise was generated as pink noise. The pink noise was generated using the colored noise library [14]. Adding background noise is a mixup augmentation where labels of background noise are neglected.
3. **Tanh distortion**- This technique is adding distortion to recordings. The tanh() function can give a rounded "soft clipping" kind of distortion, and the distortion amount is proportional to the loudness of the input and the pre-gain. Tanh is symmetric, so the positive and negative parts of the signal are squashed in the same way.
4. **Denoise transform**- The denoising steps are the following. Apply the fft to the signal followed by computing the frequencies associated with each coefficient hence keeping only the coefficients that have a low enough frequency (in absolute). Compute the inverse fft [15].
5. **Gain**- Multiply the audio by a random amplitude factor to reduce or increase the volume. This technique can help a model become somewhat invariant to the overall gain of the input audio.
6. **Loudness normalization**- Apply a constant amount of gain to match a specific loudness.
7. **Mixup** - The training examples for mixup can be constructed using the following formula:

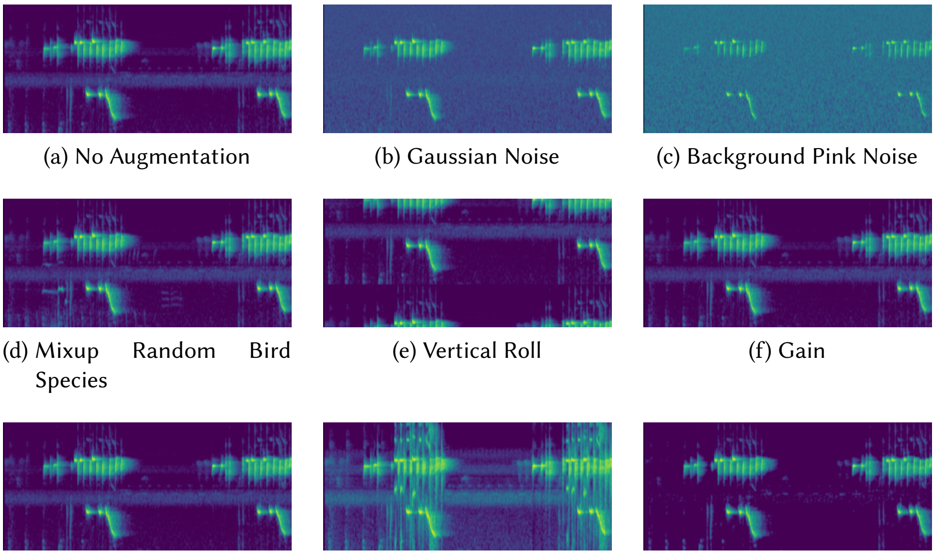
   $$x = x_i + (1 - \tau) \times x_j \tag{1}$$

   $$y = y_i + (1 - \tau) \times y_j \tag{2}$$

   where (x_i,y_i) and (x_j,y_j) are the two ramdomly selected examples from the training data where τ is the mixed ratio. τ is set as 0.1.
8. **Vertical roll**- This method is applied on spectrograms. Roll the spectrogram with respect to height * α, where α is the roll factor (eg.0.05).

ID	Data Augmentation	Time Domain	Spectro-gram	Time Taken per epoch (min)		
				DenseNet-121	EfficientNet-b0-ns	ResNet-50
1	Gaussian Noise	✓		6	4	6
2	Pink Noise	✓		6	4	6
3	Tanh distortion	✓		6	4	6
4	Denoise Transform	✓		6	5	7
5	Gain	✓		4	4	4
6	Loudness Normalization	✓		4	4	4
7	Mixup Random Bird Species		✓	9	5	9
8	Vertical Roll		✓	8	5	8

Table 1
List of augmentation strategies, their domain of application (time vs. spectral), and respective IDs.



(a) No Augmentation

(b) Gaussian Noise

(c) Background Pink Noise

(d) Mixup Random Bird Species

(e) Vertical Roll

(f) Gain

(g) Loudness

(h) Tanh Distortion

(i) Denoise transform

Figure 3: Illustration of different augmentation techniques on time domain, frequency domain and spectrogram.

3.3. Models Architecture

In Table 4 we discuss the model architectures used in our experiments. Single models had reasonable better results when validated using our stratified 5 fold cross-validation. Our validation data were augmented with pink noise to replicate the test set, having strong cross-validation will yield a better performance of the model. From the previous BirdCLEF challenges, Sound Event Detection (SED) models played a crucial role in achieving the top performances. We used the SED based model adopted from DCASE 2021 [16]. The encoder part of SED models is DenseNet-121, ResNet-50, Noisy student EfficientNet-b0(EfficientNet-b0-ns) whereas the decoder part is the attention head. The output embedding of the encoder is average-pooled along the frequency dimension before being fed into the SED decoder. For the SED decoder, we use 2 layers, 1D CNN and temporal network bidirectional GRU with a hidden size of 256 as presented in Figure 4. The SED problem was formulated as a multi-label multi-class classification task for bird events. We employ 2 fully connected (FC) layers to produce the SED output. The first FC layer comprises 512 hidden units, which is followed by a sigmoid activation function to produce the posterior probabilities of the bird events.

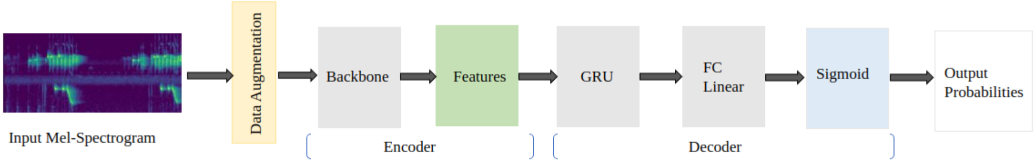


Figure 4: Example of a multilabel SED classification model.

3.4. Training Details

A Titan-RTX GPU with 24 GB VRAM has been used to train our models, the reports of training time for each epoch are discussed in the Table 1. Models were trained for 30 epochs using a batch size of 64 and the Adam Optimizer. The loss used was based on Binary Cross-Entropy (BCR) based focal loss. The pytorch-based sampler was used [17] to balance the train set. Additionally, we use a cosine annealing-based Learning Rate Scheduler with a base Learning Rate (LR) of 0.001. We track the loss function, F1 score, recall during our training process.

4. Experimental Results

Table 3 presents the best augmentation combinations, the focus was to develop a model to recognize bird sounds effectively without adding any soundscape recordings to the training samples. The best single augmentation from set 1 had an absolute increase of 8% in the F1 score, when compared with baseline. The best combination augmentations are tanh distortion, mixup different bird species, Gaussian noise, denoise transformation, and loudness normalization achieved a 12% increase in the score when compared with the baseline.

Set	Description
Set 1	Single augmentation
Set 2	Ranked augmentations 1 and 2 are combined, followed by ranks 3, 4, 5 and 6, 7, 8 are combined.
Set 3	The top augmentation combination from set 2 are combined with other augmentation combinations from set 2.

Table 2
Description of each set from Table 3.

Tables 5 and 4 summarize the results. Data augmentation improves the model robustness against anthropogenic noise from the test set as discussed in the Table 4. The ensemble technique used in this competition was bagging. The bagging of 5 models with different augmentation strategies achieves better performance. The single model results are summarized in Table 4.

	1	2	3	4	5	6	7	8	F1 score
Baseline									0.55
Set 1	✓								0.61
		✓							0.60
			✓						**0.63**
				✓					0.61
					✓				0.59
						✓			0.59
							✓		0.62
								✓	0.58
Set 2.1			✓				✓		**0.65**
Set 2.2	✓	✓		✓					0.63
Set 2.3					✓	✓		✓	0.60
Set 3.1	✓	✓	✓	✓	✓	✓	✓	✓	0.61
Set 3.2			✓		✓	✓	✓	✓	0.64
Set 3.3	✓	✓	✓	✓			✓		**0.67**

Table 3
Different augmentation combinations and their respective F1 scores. The model architecture used here was SED-DenseNet-121. For the set descriptions, be referred to Table 2. The numbers in the heading refer to the IDs in Table 1

.

Table 4 presents single model results of SED-DenseNet-121, SED-ResNet-50, SED-EfficientNet-b0-ns. The models were trained along with the Set 3 data augmentation combination. A detailed description is documented in Table 3. The best results were obtained when these augmentation combinations were trained along with SED-EfficientNet-b0-ns and CV-BG which achieved an F1 score of 0.74 in PU-LB and 0.69 in PR-LB.

Table 5 presents ensemble results. The 5 model ensemble of SED-EfficientNet-b0-ns achieved a better F1 score of 0.75 in PU-LB and 0.71 in PR-LB. The difference between the CV-BG and PU-LB for SED-DenseNet-121, ResNet-50 was - 8 to 12% whereat for SED-EfficientNet-b0-ns was - 3 to 5%. The ensemble result achieved the 69th place out of 807 teams, as shown in Figure 5.

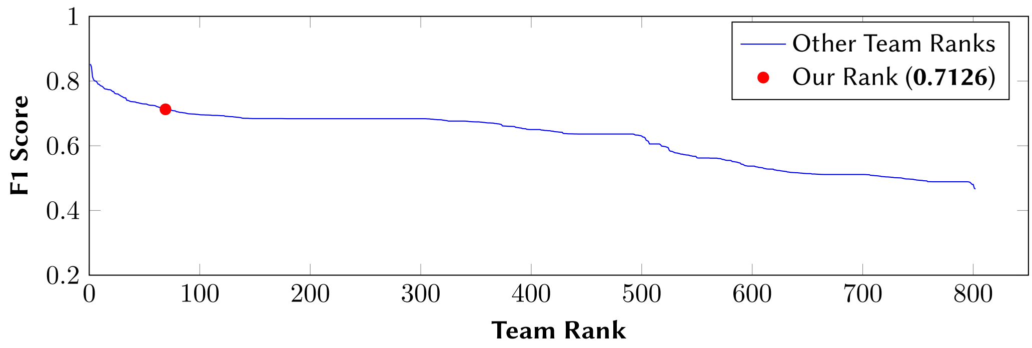


Figure 5: Leadership board rank based on PR-LB.

ID	Model	Augmentation Set	CV	CV-BG	PU-LB	PR-LB
1	SED-DenseNet-121	Set 3.1	0.74	-	0.59	0.57
2	SED-DenseNet-121	Set 3.1	-	0.73	0.61	0.59
3	SED-DenseNet-121	Set 3.2	0.72	-	0.60	0.57
4	SED-DenseNet-121	Set 3.2	-	0.75	0.64	0.61
5	SED-DenseNet-121	Set 3.3	0.72	-	0.65	0.63
6	**SED-DenseNet-121**	**Set 3.3**	**-**	**0.73**	**0.67**	**0.64**
7	SED-ResNet-50	Set 3.1	0.76	-	0.67	0.65
8	SED-ResNet-50	Set 3.1	-	0.78	0.70	0.66
9	SED-ResNet-50	Set 3.2	0.80	-	0.69	0.64
10	SED-ResNet-50	Set 3.2	-	0.75	0.70	0.64
11	SED-ResNet-50	Set 3.3	0.73	-	0.69	0.65
12	**SED-ResNet-50**	**Set 3.3**	**-**	**0.76**	**0.71**	**0.69**
13	SED-EfficientNet-bo-ns	Set 3.1	0.76	-	0.72	0.67
14	SED-EfficientNet-bo-ns	Set 3.1	-	0.78	0.73	0.68
15	SED-EfficientNet-bo-ns	Set 3.2	0.80	-	0.71	0.66
16	SED-EfficientNet-bo-ns	Set 3.2	-	0.75	0.72	0.67
17	SED-EfficientNet-bo-ns	Set 3.3	0.77	-	0.73	0.68
18	**SED-EfficientNet-bo-ns**	**Set 3.3**	**-**	**0.79**	**0.74**	**0.69**

Table 4
Classification results using the single model approach with 3 different models (SED-Densenet-121, SED-EfficientNet-bo-ns, SED-ResNet-50), whereat each is combined with 3 augmentations sets and 2 cross-validation strategies

Models	PU-LB	PR-LB
1+2+3+4+5+6	0.69	0.65
7+8+9+10+11+12	0.72	0.69
18+16+14+17+15+13	0.74	0.70
18+16+14+17+15	**0.75**	**0.71**

Table 5
Bagging ensemble results using combinations of models shown in Table 4.

5. Conclusion and Discussion

The presented research helps to automate the monitoring of bird species to improve the quality of life and keep track of endangered bird species and climate changes. Replicating test set background noise (set soundscape noise) was difficult, in the proposed system the noise-based augmentation strategies without soundscape noise achieve a better F1 score on the test set. The BirdCLEF 2022 focuses on recognizing endangered species, particularly in the Hawaii location where some classes in the train set had a very limited number of samples which made the competition more challenging. In all BirdCLEF challenges, the major problem was the domain shift between train and test set were the recordings of train set were distinct concerning test set since both the dataset was recorded using different microphones. This year's competition focused on recognizing 21 major endangered species in Hawaii, whereat the distribution of samples was majorly unbalanced and some classes even had only one sample.

The larger increase in performance was obtained by data augmentation methods. Augmentations like tanh, denoise, Gaussian, and pink noise boosted the performance along with mixing up different bird species. The results of single augmentation scored in the range of 0.58 - 0.63 (F1 score) which were better than baseline performance. The best-combined augmentation was set 3.3 which performed 12% better than the baseline. Some augmentation strategies such as time stretch and pitch shift turned out to be ineffective in our training pipeline as they consume significantly longer time for training the model.

The test set focused on predicting 21 classes. Unfortunately training only those 21 classes failed, since the dataset was unbalanced as discussed in Figure 1. Increasing the length of the audio samples to 10 or 30 seconds chunks had a bad influence on the model performance and even consumed more training time.

References

[1] S. Kahl, T. Denton, H. Klinck, Birdclef 2021: Bird call identification in soundscape recordings, in: CLEF 2022 - Conference and Labs of the Evaluation Forum, 2022.

[2] S. Kahl, M. Clapp, W. Hopping, H. Goëau, H. Glotin, R. Planqué, W.-P. Vellinga, A. Joly, Overview of birdclef 2020: Bird sound recognition in complex acoustic environments, 2020.

[3] S. Kahl, T. Denton, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of birdclef 2021: Bird call identification in soundscape recordings, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, 2021.

[4] M. Lasseck, Audio-based bird species identification with deep convolutional neural networks, in: CLEF, 2018.

[5] M. Lasseck, Bird species identification in soundscapes, in: CLEF, 2019.

[6] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, Birdnet: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236. URL: https://www.sciencedirect.com/science/article/pii/S1574954121000273. doi:https://doi.org/10.1016/j.ecoinf.2021.101236.

[7] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, M. D. Plumbley, Panns: Large-scale pretrained audio neural networks for audio pattern recognition, CoRR abs/1912.10211 (2019). URL: http://arxiv.org/abs/1912.10211. arXiv:1912.10211.

[8] T. Lin, P. Goyal, R. B. Girshick, K. He, P. Dollár, Focal loss for dense object detection, CoRR abs/1708.02002 (2017). URL: http://arxiv.org/abs/1708.02002. arXiv:1708.02002.

[9] G. Huang, Z. Liu, K. Q. Weinberger, Densely connected convolutional networks, CoRR abs/1608.06993 (2016). URL: http://arxiv.org/abs/1608.06993. arXiv:1608.06993.

[10] V. Lostanlen, J. Salamon, A. Farnsworth, S. Kelling, J. P. Bello, Robust sound event detection in bioacoustic sensor networks, PLOS ONE 14 (2019) 1–31. URL: https://doi.org/10.1371/journal.pone.0214168. doi:10.1371/journal.pone.0214168.

[11] E. Fonseca, M. Plakal, D. P. W. Ellis, F. Font, X. Favory, X. Serra, Learning sound event

classifiers from web audio with noisy labels, CoRR abs/1901.01189 (2019). URL: http://arxiv.org/abs/1901.01189. arXiv:1901.01189.

[12] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, Y. Shi, Torchaudio: Building blocks for audio and speech processing, 2021. URL: https://arxiv.org/abs/2110.15018. doi:10.48550/ARXIV.2110.15018.

[13] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[14] J. Timmer, Koenig, Generate gaussian distributed noise with a power law spectrum, 1995. URL: https://pypi.org/project/colorednoise/.

[15] A. Defossez, G. Synnaeve, Y. Adi, Real time speech enhancement in the waveform domain, 2020. URL: https://arxiv.org/abs/2006.12847. doi:10.48550/ARXIV.2006.12847.

[16] S. Adavanne, A. Politis, J. Nikunen, T. Virtanen, Sound event localization and detection of overlapping sources using convolutional recurrent neural networks, IEEE Journal of Selected Topics in Signal Processing 13 (2018) 34–48. URL: https://ieeexplore.ieee.org/abstract/document/8567942. doi:10.1109/JSTSP.2018.2885636.

[17] I. D. S. pytorch, Imbalanced dataset sampler, 2019. URL: https://github.com/ufoym/imbalanced-dataset-sampler.