TOWARDS LEARNING A DIFFERENCE-AWARE GENERAL-PURPOSE AUDIO REPRESENTATION

Daiki Takeuchi, Masahiro Yasuda, Daisuke Niizumi, Noboru Harada

NTT Corporation, Japan

ABSTRACT

General-purpose audio representations with self-supervised learning have shown promising results on diverse tasks. Methods such as BYOL-A try to learn semantically robust representation by ignoring differences between two data computed using data augmentations that simulate semantically similar data from the same input. However, some audio-difference-related tasks require representations that are sensitive to slight semantic differences while maintaining robustness to similar data. This study investigates how to learn difference-aware audio representations. We propose subtraction-consistent representation learning in which mixed sounds are separable by subtracting representations in latent space. In the proposed method, an additional network extending BYOL-A learns the difference between a sound sample and its down-mix with another sound sample. Experiments confirmed that the proposed method improves the accuracy of difference-aware audio tasks while maintaining the general-purpose audio representation performance.

***Index Terms*—** general-purpose audio representation, audio difference, self-supervised learning

1. INTRODUCTION

General-purpose audio representations with self-supervised learning have shown promising results on diverse tasks [1–4]. Some of the self-supervised learning methods try to semantically robust learn representations by ignoring differences between two data augmentations applied to the same input. Data augmentations, such as time shifting, pitch shifting, and mixing other audio samples or noise, are designed and selected to emulate divisions to be ignored to obtain semantically similar representations in the latent space. As a result, learned representation will be robust to the difference between semantically similar data.

However, some difference-aware audio tasks, such as audio retrieval with auxiliary information [5], require representations that are sensitive to slight semantic differences while maintaining robustness to similar data. Existing general-purpose representation learning methods do not sufficiently solve this kind of task.

To address the lack of difference awareness in conventional self-supervised learning, we propose subtraction-consistent representation learning in which mixed sounds are separable by subtracting representations in latent space. The overview of the proposed method is shown in Fig. 1. The proposed method is implemented as an extension of BYOL-A [3]. Subtraction-consistent representation learning is based on the hypothesis that the semantic information present in a mixture of two sounds at similar sound pressure levels is equivalent to the combined semantic information of the two sounds before mixing. Our training method subtracts the representation of one mixed audio sample from the representation of the mixture and maximizes the agreement between the remaining representation of the subtraction and the representation of the other mixed audio sample. Multitask learning of BYOL-A and subtraction-consistent representation learning losses are performed during training. BYOL-A learns semantically robust audio representation, while subtraction-consistent representation learning makes that representation aware of differences. As a result, our method should learn a difference-aware general-purpose audio representation.

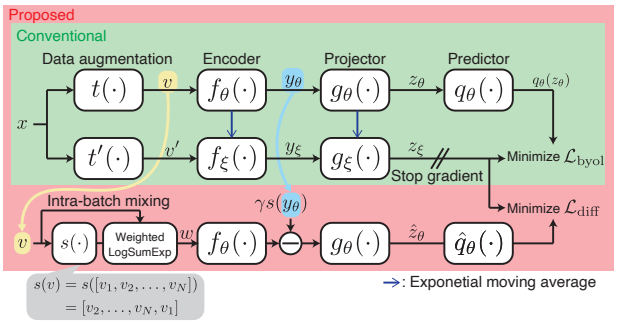


Figure 1: Overview of the proposed method and BYOL-A (conventional method). The proposed method (colored in red) extends BYOL-A (colored in green), mixing the augmented view v among other batch data to make a mixed input w. We train the proposed method to predict the BYOL-A target network output z_ξ from the difference between the encoder outputs of v and w.

Experiments confirm the learned representation by the proposed method improves the performance on two difference-aware audio tasks: environmental sound classification under noisy conditions and audio retrieval with auxiliary information. We also evaluate the learned representations in various downstream tasks and confirm that the performance was comparable to that learned by conventional BYOL-A. Therefore, the proposed method learns the difference-aware audio representation without degrading the general-purpose audio representation performance.

2. RELATED WORK

2.1. Self-supervised learning for audio representation

The general-purpose audio representation with self-supervised learning is effective for diverse tasks, including environmental sounds, music, and speech. BYOL-A [3] combines the self-supervised learning method Bootstrap Your Own Latent [6] (BYOL) with audio data augmentation. It learns representations invariant to differences in background noise and changes in the pitch and duration of audio. COLA [1] uses contrastive learning to learn representations that become closer to the segments cropped from the same audio clip and farther among the segments from the different

audio clips, making the representations of an audio clip invariant to the cropping location. Fonseca et al. [2], and DeLoRes [4] also learn representations invariant to audio differences produced by data augmentation.

While they learn representations robust to changes produced by data augmentation and differences in segment cropping locations, they do not explicitly learn to encode information about differences in audio. This study investigates the learning of a general-purpose audio representation with awareness of audio differences by introducing the difference-based loss created by mixing sounds.

2.2. Difference-aware audio tasks

The recognition and retrieval tasks related to audio differences have also been studied. In [5], audio retrieval with auxiliary information was proposed. The content-based audio retrieval with text-query modifier [5] enables us to search an audio clip from an audio sample and the description of the difference. This method uses the common latent space between audio clips and descriptions of differences.

The methods to generate text explaining the difference between two sounds have also been studied [7, 8]. In [8], self-supervised learning focusing on the fact that input two audio clips are similar but slightly different is applied for learning the audio difference encoder. For the audio captioning system, the training method using the difference between the audio representation of before and after mixing is proposed in [9]. This study fixed the parameters of the encoder model that outputs acoustic representations and utilized the differences to train the text generation model. Unlike this study, we use differences to learn the parameters of the encoder model that outputs the audio representation.

3. BACKGROUND: BYOL-A

BYOL-A [3] is the method to obtain general-purpose audio representation by self-supervised training based on the BYOL framework [6]. The green area in Fig. 1 shows the overview of the BYOL training procedure. BYOL framework uses online and target networks with parameters θ and ξ, respectively. The online network has encoder f_θ, projector g_θ, and predictor q_θ. The target network has encoder f_ξ and projector g_ξ. The parameter of the target network ξ is the exponential moving average of the parameter of the online network θ. In the online network, compute v by data augmentation t to input x, then pass through the encoder, projector, and predictor to obtain $q_\theta(z_\theta)$. In the target network, compute v' by another data augmentation t' to input x, then pass through the encoder and projector to obtain z_ξ. After that, the normalized mean squared error of $q_\theta(z_\theta)$ and z_ξ is used for training loss:

$$\begin{aligned}\mathcal{L}_{\mathrm{byol}} &= ||l_2(q_\theta(z_\theta)) - l_2(z_\xi)||_2^2 \\ &= 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z_\xi \rangle}{||q_\theta(z_\theta)||_2 \cdot ||z_\xi||_2}, \end{aligned} \tag{1}$$

where $l_2(\cdot)$ is l_2-normalization, and $\langle x, y \rangle$ indicates the inner product of x and y. Thus, the BYOL framework can obtain the feature representation robust to the data augmentation t and t', and designing the data augmentation is one of the important elements to obtain better representation.

BYOL-A uses mel-spectrogram to preprocess the audio signal and three data augmentation methods that consider the nature of the audio signal: Mixup, random resize crop (RRC), and random linear fader (RLF). Mixup randomly adds another sound as background sound, RRC performs shifts and stretches in the axis of time and frequency randomly, and RLF makes random changes of temporal amplitude, which simulates fade in or out. BYOL-A applies Mixup, RRC, and RLF to input x sequentially and outputs the data-augmented views v and v'.

4. PROPOSED METHOD

The proposed method adds self-supervised learning to represent the relation between the audio signals before and after the mixture through differences in feature representations in the training procedure of BYOL-A. The training procedure of the proposed method is shown in red in Fig. 1. The proposed method is structured to include BYOL-A and in addition to a conventional loss $\mathcal{L}_{\mathrm{byol}}$, it learns to predict the target network output z_ξ from the difference between audio representations before and after the mixture. The computational procedure of the proposed method branches from the input v after data augmentation, following the conventional BYOL-A. First, the mixture w is obtained by intra-batch mixing v with its index-shifting $s(v)$ and weighted log-sum-exp:

$$w = \log(\gamma \exp(s(v)) + (1 - \gamma) \exp(v)), \tag{2}$$

where, γ is the mixing rate, s is the intra-batch shift operator, $s(v) = s([v_1, v_2, \ldots, v_N]) = [v_2, \ldots, v_N, v_1]$ and v_n is n-th data of v. Then, the difference between the encoder output of the mixture $f_\theta(w)$ and encoder output of the sound before mixing multiplied by the mixing ratio $\gamma s(y_\theta)$ is calculated and input into the projector g_θ and another predictor $\hat{q}_\theta$ to compute $\hat{q}_\theta(\hat{z}_\theta)$. Finally, we get the difference loss $\mathcal{L}_{\mathrm{diff}}$, a normalized mean squared error between $\hat{q}_\theta(\hat{z}_\theta)$ and z_ξ:

$$\begin{aligned}\mathcal{L}_{\mathrm{diff}} &= ||l_2(\hat{q}_\theta(\hat{z}_\theta)) - l_2(z_\xi)||_2^2 \\ &= 2 - 2 \cdot \frac{\langle \hat{q}_\theta(\hat{z}_\theta), z_\xi \rangle}{||\hat{q}_\theta(\hat{z}_\theta)||_2 \cdot ||z_\xi||_2}. \end{aligned} \tag{3}$$

The training step backpropagates the weighted sum of two loss $(1 - \lambda)\mathcal{L}_{\mathrm{byol}} + \lambda\mathcal{L}_{\mathrm{diff}}$, where λ is the weight parameter.

5. EXPERIMENTS

We conducted the following experiments to evaluate the audio representations learned by the proposed method, and we used BYOL-A [3] as the baseline method.

5.1. Pre-training Setup

All audio data was transformed into a mel-spectrogram with a sampling frequency of 16,000 Hz, window size of 25 ms, hop size of 10 ms, and mel-spaced frequency bins $F = 64$ in the range of 50 to 8,000 Hz. The pre-training dataset was a random sample of 200,000 files from AudioSet [10]. Note that it was approximately 1/10 of the original size. The pre-training only utilized audio files without employing any labels. The same setup for data augmentation, exponential moving average, and model structures was used as the conventional method [3]. Adam [11] was used as the optimizer with a learning rate 0.001. The number of epochs was set to 100. The weight parameter λ, which decides the balance between $\mathcal{L}_{\mathrm{byol}}$ and $\mathcal{L}_{\mathrm{diff}}$ was set to 0, 0.1, 0.2, 0.5, or 0.8. Note that $\lambda = 0$ corresponds to the baseline method. The mixing rate of the intra-batch mixing γ is randomly sampled uniformly between 0.4 and 0.6 for each input.

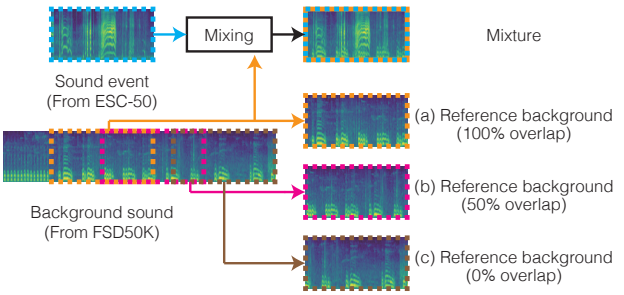


Figure 2: Procedure to generate BgKnown ESC-50. We mix the ESC-50 samples with the FSD50K sample as background noise to create a mixture and three reference background sounds.

Table 1: BgKnown ESC-50 results (%). A larger λ learned more from $\mathcal{L}_{\text{diff}}$ improves accuracy, validating that the proposed approach achieved the difference-aware property.

Method	λ	Mix	(a) 100%	(b) 50%	(c) 0%
Baseline	0	47.25	55.29	52.21	47.92
Proposed	0.1	47.39	55.54	52.46	48.50
Proposed	0.2	**47.42**	57.54	53.37	48.67
Proposed	0.5	47.33	58.96	54.63	50.50
Proposed	0.8	45.84	**59.96**	**55.50**	**51.96**

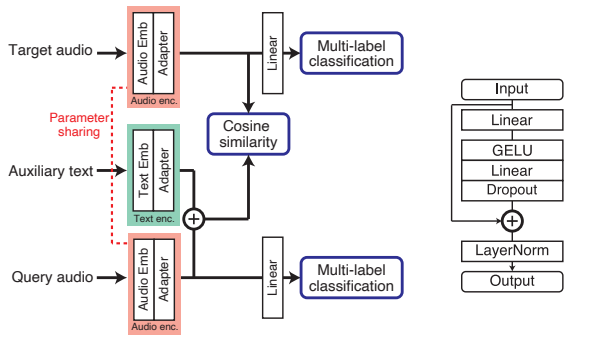
5.2. Evaluation: Background-known ESC-50

This experiment verified that the audio representation learned by the proposed method holds effective information about the audio differences for solving a task. To do so, we created a dataset, Background-known ESC-50 (BgKnown ESC-50), and tested the pre-trained models.

Dataset: Background-known ESC-50
BgKnown ESC-50 extends ESC-50 [12], an environmental sound classification task with 50 classes, by mixing the FSD50K audio files as background noise to the ESC-50 audio files. As shown in Fig. 2, we created a mixture (an ESC-50 audio contaminated with noise) and three reference backgrounds. While solving a task using only the mixture is challenging due to the noise, we made one of the reference backgrounds available; the more effectively the solver utilizes the difference between a mixture and a reference, the higher the task performance.

We randomly selected the FSD50K sample with 10 seconds or longer and cropped (a) a 5-second long clip, (b) a 5-second long clip with 50% overlap with (a), and (c) a 5-second long clip without overlap from (a), and mixed (a) into ESC-50 sample with a random SNR between 0 to 3 dB using Scaper [13]. We kept the labels unchanged. Among the split folds of ESC-50, we assigned 1, 2, and 3 to the training set (1200 files) and 4 and 5 to the test set (800 files). We used the FSD50K development and evaluation sets as the background noise for the training and test sets, respectively.

Experimental setup
We conducted a linear evaluation using feature differences on BgKnown ESC-50. First, we used the pre-trained encoder f_θ to obtain representations y_{mix} and y_{bg} of the mixture and reference background and obtained the difference representation $y_{\text{diff}} = y_{\text{mix}} - y_{\text{bg}}$. Then, we conducted a linear evaluation using the y_{diff} on the three problem settings (a) to (c).



(a) Model structure for audio retrieval with auxiliary information (b) Structure of adapter

Figure 3: Model and adapter structure for audio retrieval with auxiliary information. We train the system using a contrastive learning and classification task. Audio Emb and Text Emb indicate audio and text embedding layers, respectively. GELU is the Gaussian error linear unit [14].

Table 2: APwD-Dataset results (%). The proposed method improves the audio encoder, performing better than the conventional and baseline, with the best results using λ of 0.2 to 0.5.

		Rain			**Traffic**		
Method	λ	**R@1**	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
Conventional [5]	-	44.5	72.1	76.9	39.1	62.2	69.5
Baseline	0	50.23	71.66	75.26	36.86	58.59	67.93
Proposed	0.1	51.96	71.63	74.83	37.96	59.73	66.9
Proposed	0.2	**53.99**	**72.06**	**76.00**	37.70	60.06	68.00
Proposed	0.5	52.76	71.73	75.73	**39.73**	60.63	**68.30**
Proposed	0.8	51.66	70.63	74.86	39.56	**61.36**	68.16

We followed the standard linear evaluation procedure in the conventional method [3] that trains a single linear layer, taking the difference representation y_{diff} as input. We set the training epochs for 200 with early stopping based on the validation loss value, assigned 10% of the training set as the validation set, and used the Adam optimizer with a learning rate of 0.001. We ran the experiments with different random seeds three times and averaged the results.

Results
Table 1 shows the results of BgKnown ESC-50. In addition to the (a) to (c), we also tested y_{mix} as is in the linear evaluation, denoted as "Mix". The results show that the proposed method improved accuracy with larger λ for the (a) to (c) when using the difference representation y_{diff}. In contrast, the results stayed around 47% for the Mix when we used the representation of the y_{mix} as it is instead of y_{diff}. These results demonstrate that the representation of the proposed method holds effective information about the audio differences.

Notably, the (c) 0% results show improvement despite no direct overlap with the mixed background noise. The segments cropped from the same audio clip share the background sounds (or sound scene of the clip), indicating that the information about the audio difference represents the clip-level (or semantic-level) information of the audio clip.

5.3. Evaluation: Audio retrieval with auxiliary information

We validated the effectiveness of the difference-aware representation for the difference-aware audio task. We evaluated the represen-

Table 3: Linear evaluation results on audio classification tasks (%) with 95% CI. The results in bold are the best scores in each task. Many underlined results within the 95% confidence interval of the baseline show that our models maintain baseline performance.

Method	λ	ESC-50	US8K	SPCV2	VC1	VF	CRM-D	GTZAN	NSynth	Surge	Average
Baseline	0	82.70 ± 1.76	79.43 ± 0.73	93.16 ± 0.18	**57.17 ± 0.97**	**93.39 ± 0.38**	**61.81 ± 2.30**	**67.24 ± 3.93**	74.80 ± 0.22	37.82 ± 0.17	**71.95**
Proposed	0.1	82.12 ± 1.37	**79.85 ± 0.33**	**93.22 ± 0.16**	56.90 ± 0.12	93.22 ± 1.10	60.67 ± 0.00	**67.24 ± 0.86**	**76.30 ± 0.37**	37.82 ± 0.02	71.93
Proposed	0.2	**82.77 ± 0.85**	79.72 ± 0.48	93.15 ± 0.31	56.75 ± 0.14	93.38 ± 0.09	61.21 ± 1.64	**67.24 ± 3.09**	74.23 ± 0.39	37.68 ± 0.29	71.79
Proposed	0.5	82.37 ± 1.56	78.99 ± 0.50	92.96 ± 0.24	55.86 ± 0.03	92.78 ± 0.79	60.65 ± 0.33	66.90 ± 1.48	74.76 ± 0.26	38.17 ± 0.91	71.49
Proposed	0.8	80.80 ± 2.00	78.62 ± 0.12	92.82 ± 0.23	55.19 ± 0.15	92.10 ± 0.45	61.36 ± 1.45	66.78 ± 1.98	74.25 ± 0.29	**38.72 ± 0.92**	71.18

tations using an audio retrieval task with auxiliary information [5], one of the practical tasks utilizing semantic differences.

Experimental setup

This experiment used the APwD-Dataset [5], which consists of a set of two similar audio clips and an auxiliary text describing the differences between these audio clips. The task is to search for a target audio that best matches the query audio and auxiliary text. The audio clip is a mixture of ESC-50 audio event samples (foreground sound with class labels) and an FSD50K acoustic scene sample (background sound). This dataset contains two scenes, "Rain" and "Traffic," distinguished by their background sounds, consisting of 50,000/1,000 samples for training and testing sets. In addition, class labels are available for an extra classification task.

We followed [5] for the system and the training/test details. Fig. 3 shows the system that inputs a query audio and a query-modifier text (auxiliary information), and searches the target audio using cosine similarity. During training, it learned through contrastive learning and multi-label classification tasks. We used the encoder pre-trained by the proposed method as the audio embedding layer in the shared audio encoder blocks and DistilBERT [15] as text embedding layer in the text encoder block. We froze all audio/text encoder parameters. We trained the adapter and linear layers for 300 epochs using the Adam [11] optimizer. We assigned 10% of the training samples for validation, and the model with the smallest validation loss was used for evaluation. We used recall@K(R@K) to evaluate the accuracy of audio retrieval. R@K is the rate at which the ground-truth audio files are included in the Kth rank of the selected candidates. We ran the evaluation with three random seeds and averaged the results to obtain the final score.

Results

Table 2 shows that the audio encoder pre-trained by the proposed method improves the audio retrieval performance. The results contain the conventional method [5] using VGGish [16] as audio embedding, the baseline using BYOL-A, and the proposed methods. The "Rain" results show that the proposed method improved to 53.99% for R@1 from the baseline of 50.23% and the conventional 44.5%. The "Traffic" results also show that the proposed method improved to 39.73% for R@1 from the baseline of 36.86% and the conventional 39.1%. These results validate the effectiveness of the proposed subtraction-consistent representation learning for the difference-aware audio task.

5.4. Evaluation: General-purpose audio representation

We validated that the proposed subtraction-consistent representation learning maintains a general-purpose audio representation performance without the impact of learning the difference-aware ability. We followed BYOL-A [3] to assess the performance in a linear evaluation on various tasks, including environmental sound, music, and speech.

Experimental setup

The tasks for linear evaluation include ESC-50 [12], Urban Sound 8K [17] (US8K), Speech Command V2 [18] (SPCV2), VoxCeleb1 [19] (VC1), VoxForge [20] (VF), CREMA-D [21] (CRM-D), GTZAN [22], NSynth [23], and the Pitch Audio Dataset (Surge synthesizer) [24] (Surge). The training/test details follow BYOL-A [3], such as the training epochs 200 with early stopping based on the validation loss. We ran the evaluation with three random seeds and averaged the results with 95% CI.

Results

Table 3 shows that the proposed method slightly degrades the general-purpose performance, while most results are within the 95% confidence interval. The average result degrades from 71.95% for the baseline to 71.18% for $\lambda = 0.8$. However, most task results of the proposed method are marked with underline, i.e., within the range of 95% confidence interval of the baseline results. The most significant degradation of VC1 is -1.98 from 57.17%, which should be a slight drop considering the confidence interval range is ± 0.97. These results confirm that the performance degradation caused by the proposed subtraction-consistent representation learning is generally insignificant.

We confirm that the large λ changes the characteristics of the learned representations as the APwD-Dataset results in Section 5.3. While using $\lambda = 0.8$ degrades the general-purpose performance most in Table 3, using $\lambda = 0.5$ or 0.8 improves the Traffic performance of APwD-Dataset in Table 2. In addition, Surge, a pitch classification of musical instruments, improves as λ becomes larger, suggesting the representation contains more pitch information. These observations suggest a tradeoff of task performance by the use of learning tasks.

6. CONCLUSION

This study investigates how to learn difference-aware audio representations. We propose a self-supervised learning method called subtraction-consistent representation learning. With the obtained representation, mixed sounds are separable by subtracting representations in latent space. In the proposed method, an additional network extending BYOL-A learns the difference between a sound sample and its down-mix with another sound sample. Experiments confirmed that the proposed method improves the accuracy of audio signal retrieval with text auxiliary information utilizing semantic differences in sounds. It was also confirmed that the performance of the proposed method does not degrade significantly in the linear evaluation of various traditional audio classification tasks that require general-purpose audio representation.

7. ACKNOWLEDGEMENT

This work partially supported by JST Strategic International Collaborative Research Program (SICORP), Grant Number JPMJSC2306, Japan.

8. REFERENCES

[1] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021, pp. 3875–3879.

[2] E. Fonseca, D. Ortego, K. McGuinness, N. E. O'Connor, and X. Serra, "Unsupervised contrastive learning of sound event representations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021, pp. 371–375.

[3] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Exploring Pre-trained General-purpose Audio Representations," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 31, p. 137–151, 2023.

[4] S. Ghosh, A. Seth, and S. Umesh, "Decorrelating feature spaces for learning general-purpose audio representations," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1402–1414, 2022.

[5] D. Takeuchi, Y. Ohishi, D. Niizumi, N. Harada, and K. Kashino, "Introducing auxiliary text query-modifier to content-based audio retrieval," in *Proc. Interspeech*, 2022.

[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Proc. Conf. Workshop Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[7] S. Tsubaki, Y. Kawaguchi, T. Nishida, K. Imoto, Y. Okamoto, K. Dohi, and T. Endo, "Audio-change captioning to explain machine-sound anomalies," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2023, pp. 201–205.

[8] D. Takeuchi, Y. Ohishi, D. Niizumi, N. Harada, and K. Kashino, "Audio difference captioning utilizing similarity-discrepancy disentanglement," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2023, pp. 181–185.

[9] T. Komatsu, Y. Fujita, K. Takeda, and T. Toda, "Audio difference learning for audio captioning," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2024.

[10] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 776–780.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[12] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proc. ACM Multimed.*, 2015, pp. 1015–1018.

[13] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, 2017, pp. 344–348.

[14] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv::1910.01108*, vol. abs/1910.01108, 2019.

[16] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "Cnn architectures for largescale audio classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 131–135.

[17] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. ACM Multimed.*, 2014, pp. 1041–1044.

[18] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv preprint arXiv::1804.03209*, 2018.

[19] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[20] K. MacLean, *"Voxforge"*, 2018, available at http://www.voxforge.org/home.

[21] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE Trans. Affective Comput.*, vol. 5, no. 4, 2014.

[22] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, 2002.

[23] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *ICML*, 2017.

[24] J. Turian, J. Shier, G. Tzanetakis, K. McNally, and M. Henry, "One billion audio sounds from GPU-enabled modular synthesis," in *Proc. Int. Conf. Digit. Audio Eff. (DAFx)*, 2021.