



Introducing Auxiliary Text Query-modifier to Content-based Audio Retrieval

Daiki Takeuchi, Yasunori Ohishi, Daisuke Niizumi, Noboru Harada, and Kunio Kashino

NTT Corporation, Japan

daiki.takeuchi.ux@hco.ntt.co.jp

Abstract

The amount of audio data available on public websites is growing rapidly, and an efficient mechanism for accessing the desired data is necessary. We propose a content-based audio retrieval method that can retrieve a target audio that is similar to but slightly different from the query audio by introducing auxiliary textual information which describes the difference between the query and target audio. While the range of conventional content-based audio retrieval is limited to audio that is similar to the query audio, the proposed method can adjust the retrieval range by adding an embedding of the auxiliary text query-modifier to the embedding of the query sample audio in a shared latent space. To evaluate our method, we built a dataset comprising two different audio clips and the text that describes the difference. The experimental results show that the proposed method retrieves the paired audio more accurately than the baseline. We also confirmed based on visualization that the proposed method obtains the shared latent space in which the audio difference and the corresponding text are represented as similar embedding vectors.

Index Terms: content-based audio retrieval, contrastive learning, crossmodal representation learning, deep neural network

1. Introduction

A massive amount of audio data is available on public websites, and it will continue to increase. Audio retrieval and environmental sound recognition by deep learning have been widely explored as way to use audio data effectively [1–16]. Audio retrieval is particularly essential for extracting desired audio data from the massive amount of available data.

The audio retrieval methods using an audio query are called *content-based audio retrieval*. Those methods retrieve audio data with acoustic features similar to the query [14, 15, 17–19]. The methods can retrieve any audio data as long as it is similar to the audio query. However, it is necessary to prepare an audio query similar to the target audio data in advance.

On the other hand, the audio retrieval methods using a text query retrieve audio data paired with the text query [8–10, 20]. Some methods retrieve audio clips associated with the text query, and others retrieve audio clips whose embedding in the latent space is similar to that of the text query. Compared to content-based audio retrieval, we can easily prepare and edit text queries. However, it is not always easy to describe the audio contents precisely in text.

To overcome the above issues, we propose a new content-based audio retrieval framework that combines the auxiliary text query modifier with the given audio query. With this approach, one possible scenario could be as follows: First, we retrieve the initial audio by a method such as a text-query-based one, in which case the initial audio may not be close enough to the target one. Next, we find another audio clip by using a query that

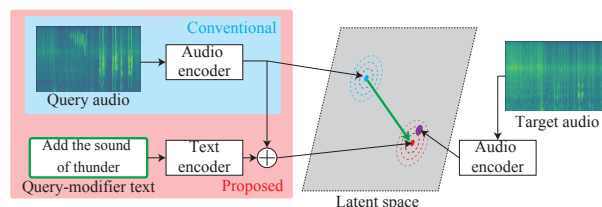


Figure 1: *Conceptual diagram of proposed method. In conventional content-based audio retrieval, a query audio becomes an embedding in a range (blue) in the latent space, roughly similar to the target audio (purple dot). The proposed method adds the embedding of the query-modifier text (green) to the query audio embedding, adjusting the retrieval range (red) closer to the target audio (purple dot).*

combines the initial audio and an additional text that describes the difference between the target and the initial audio. This combines the content-based and the text-query-based methods. We expect that the second search result gets closer to the target audio; then, we can further repeat the second step until the result becomes acceptable.

We implemented the framework as a method using neighborhood search in the latent space where the audio and difference described by text are embedded together. The method learns the representations in the latent space using crossmodal contrastive learning. We also introduced multi-task learning with a loss function that learns to classify audio contents to enhance differences in audio clips. To evaluate the proposed method, we built a dataset comprising two sounds with a difference and a description of the difference. Then, we conducted a comparison experiment with a baseline method that only uses sample audio as the query. We evaluated the retrieval accuracy and visualized the embeddings in the shared latent space and verified that the proposed method can adjust the retrieval range by means of the text query-modifier.

2. Related work

In content-based audio retrieval, audio data with features similar to the audio query in the latent space are retrieved. Wold et al. attempted to retrieve audio on the basis of acoustic features, including loudness, pitch, brightness, and bandwidth [17]. Guo et al. proposed using Mel-frequency cepstral coefficients as a feature and the latent space trained by a support vector machine [21]. Other methods used DNNs as embedding models to construct the latent space. They trained the embedding models based on contrastive learning [14] and unsupervised learning [12]. The advantage of the content-based method is that it can search for similar audio at the level of acoustic features, making it possible to retrieve audio closer to the query in terms

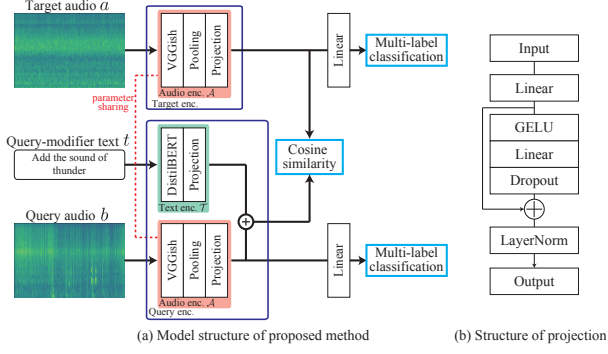


Figure 2: Illustration of proposed method. The parameters of both audio encoders are shared. VGGish and DistilBERT pass the output of the last convolutional layer and the CLS token.

of the acoustic feature representation. On the other hand, the retrieval range is limited to audio similar to the query audio. Since editing the speech waveform is a complex task, it is difficult to adjust the retrieval range by editing the query.

Audio retrieval methods using a text query train the common latent space using audio-text paired data annotated in advance and retrieve the audio data whose embedding on the trained latent space is similar to the embedding of the text query. Audio tags written in text [8] and text captions of audio data [9–11] have been used as the queries in these methods. Ikawa et al. proposed a method using onomatopoeia as a query [13]. The range of audio retrieval with a text query can be easily adjusted by editing the text query. However, collecting many pairs consisting of audio and text data for training is time-consuming, and words and phrases that are not included in the training data cannot be used to edit the text query meaningfully.

Our work is also related to crossmodal audio retrieval. [22] proposed a method to retrieve audio data corresponding to a video query. [23, 24] proposed methods to retrieve audio captions that explain the content of a video query. Guzhov et al. proposed a method to obtain the latent space integrating three modalities: audio, visual, and text [25]. Unlike these methods, which use a query that consists of a single modality, we use a query that consists of two modalities: an audio sample and an auxiliary text. We believe that the retrieval range can be adjusted by means of the auxiliary text.

3. Proposed method

We propose a content-based audio retrieval method which can select the content of the retrieved audio data by adding a difference between the target audio data and the sample as auxiliary textual information when sample audio data close to the target one is given. The proposed method uses sample audio and auxiliary text that describes the difference between the sample audio and the target audio as a search query. We construct the method with a model structure and training method that can handle two modal queries.

Consider the problem of retrieving audio data b from all candidate data by using audio query a and text query t . This can be regarded as a neighborhood search problem in the latent space, and can be treated as a problem of retrieving b satisfying the following equation:

$$\min_b d(\mathcal{F}_q(a, t), \mathcal{F}_t(b)), \quad (1)$$

where \mathcal{F}_q is an embedding model for queries a and t , \mathcal{F}_t is an embedding model for target b , and d is a distance function in the latent space. The embedding model in Eq. (1), \mathcal{F}_q and \mathcal{F}_t , with an audio encoder and text encoder is as follows:

$$\mathcal{F}_q(a, t) = \mathcal{A}(a) + \mathcal{T}(t), \quad \mathcal{F}_t(b) = \mathcal{A}(b), \quad (2)$$

where \mathcal{A} is an audio encoder, and \mathcal{T} is a text encoder. The parameters of the audio encoder \mathcal{A} in \mathcal{F}_q and \mathcal{F}_t are shared. The structure of the audio encoder and text encoder is illustrated in Fig. 2(a). In the audio encoder, the output from the last convolutional layer of the pre-trained VGGish [26] is flattened in the channel direction, and the sum of its max pooling and mean pooling [27] is input to the projection block. In the text encoder, the text t is input to the pretrained DistilBERT [28], and the output corresponding to the CLS token, which is a special token prefixed with input text, is input to the projection block. The projection block consists of two linear layers, the Gaussian error linear unit (GELU) function [29], layer normalization, and a dropout layer as shown in Fig. 2(b). The cosine similarity is used for the distance function d . Thus, Eq. (1) is rewritten as

$$\max_b d_{\text{cosim}}(\mathcal{A}(a) + \mathcal{T}(t), \mathcal{A}(b)), \quad (3)$$

where $d_{\text{cosim}}(x, y) = xy/\|x\|_2\|y\|_2$ is cosine similarity and $\|\cdot\|_2$ is ℓ^2 norm.

In the proposed method, the audio encoder and the text encoder are trained by a crossmodal contrastive loss and an audio content classification loss. The crossmodal contrastive loss connects the audio difference to its textual representation. The content classification loss trains the audio encoder so that the audio embedding has information about the type of content. The use of this loss is intended to induce a clear representation of the type of difference in the content in the audio embedding differences. Let us consider the set of data $\{a_n, b_n, t_n, v_n, w_n\}_{n=1}^B$, where a_n and b_n are similar audio clips with some differences, t_n is the description of the differences, v_n and w_n are the labels of the acoustic events contained in a_n and b_n , n is a data index, and B is batch size. The crossmodal contrastive loss function is the same as that used in [30] and is written as

$$\mathcal{L}_{\text{cont}} = -\frac{1}{2} \left(\sum_{i=1}^B \log \frac{e^{z_{i,i}}}{\sum_{j=1}^B e^{z_{i,j}}} + \sum_{i=1}^B \log \frac{e^{z_{i,i}}}{\sum_{j=1}^B e^{z_{j,i}}} \right), \quad (4)$$

where

$$z_{k,l} = d_{\text{cosim}}(\mathcal{A}(a_k) + \mathcal{T}(t_k), \mathcal{A}(b_l)) \cdot e^\tau, \quad (5)$$

τ is a temperature parameter which is fixed to 0 for simplicity in the proposed method, and i and j are the indexes of data in a batch. The audio content classification loss is written as

$$\mathcal{L}_{\text{classif}} = \sum_{i=1}^B \{\text{BCE}(\mathcal{C}(\mathcal{A}(a_i)), v_i) + \text{BCE}(\mathcal{C}(\mathcal{A}(b_i)), w_i)\}, \quad (6)$$

where BCE is the binary cross entropy loss and \mathcal{C} is the linear layer with the sigmoid function. Finally, the sum of the two losses is the overall loss for training: $\mathcal{L} = \mathcal{L}_{\text{cont}} + \rho\mathcal{L}_{\text{classif}}$, where ρ is a weighting parameter.

In addition, the parameters of VGGish and DistilBERT are fixed in training. Thus, only the parameters of the projection blocks are updated.

Table 1: Labels used to synthesize APwD-Dataset

	<i>Rain</i>	<i>Traffic</i>
background (from FSD50K)	rain	car_passing_by
event (from ESC-50)	dog, chirping_birds thunder, footsteps	dog, chirping_birds car_horn, church_bells

4. Experiment

In the experiment, we focused on three types of difference: an increase/decrease in a background sound, addition/removal of a sound event, and an increase/decrease of a sound event. We built a dataset consisting of a pair of sounds with the above differences and the corresponding text and evaluated the proposed method by training and testing with its dataset.

4.1. Audio Pair with Difference Dataset

We built an Audio Pair with Difference Dataset (APwD-Dataset) to evaluate the proposed method. The APwD-Dataset was a set of two similar audio clips synthesized using the FSD50K [2] and ESC-50 [31] audio data, and an auxiliary text describing the differences between the similar audios. Scaper [32] was used to synthesize the similar audio data. In this experiment, the dataset was created by setting up two scenes: *Rain* and *Traffic*¹. To synthesize a *Rain* scene, data labeled “rain” in FSD50K was used as background, and data labeled “dog”, “chirping_bird”, “thunder”, or “footsteps” in ESC-50 was used as events added to background. To synthesize a *Traffic* scene, data labeled “car_passing” in FSD50K was used as background, and data labeled “dog”, “chirping_birds”, “car_horn”, or “church_bells” in ESC-50 was used as events. The sounds with specific labels used for synthesis in the FSD50K and ESC-50 shown in Table 1. The development set and evaluation set for each scene contain 50,000 and 1,000 data, respectively.

The APwD dataset was synthesized using the following procedure. First, the audio data with the labels were extracted from FSD50K and ESC-50. The data assigned to the training and validation split of FSD50K and folds 1–4 of ESC-50 were used to synthesize the development set of the APwD-dataset, and the data assigned to the evaluation split of FSD50K and fold 5 of ESC-50 were used to synthesize the evaluation set. Afterwards, audio data containing audible noise and other audio events were manually excluded.

Next, we synthesized pairs of similar audio samples, α and β , which only consist of background audio. Two pieces of data were cropped from the same audio file for 10 s at random locations and assigned to α and β .

Then, the difference was given between the pair of sound data by adding another background audio and/or event audio to α and β . We consider the following six types of differences listed below and describe how they were synthesized. Note that because it is difficult to semantically reduce or eliminate background audio and event audio, they are simulated by addition.

- (a) Increase volume of background audio of α : Randomly select the background sound data, cut out 10s, and add it to audio sample β
- (b) Decrease volume of background audio of α : Apply operation (a) and replace α and β
- (c) Add sound of audio event to α : Randomly select the data used in the audio event and add it to audio sample β

- (d) Remove sound of audio event from α : Apply operation (c) and replace α and β
- (e) Increase volume of audio event of α : Randomly select two audio data with the same label for the audio event and normalize them. After that, the amplitude of one data is reduced. The one with the larger amplitude is added to audio sample β ; the one with the smaller amplitude is added to audio sample α .
- (f) Decrease volume of audio event of α : Apply operation (e) and replace α and β

We provided one or two types of differences for each pair of audio samples.

Finally, a description corresponding to the difference was assigned. This description was written in the form of an imperative sentence, whose content described how to change the audio sample α to the audio β . For example, if difference (a) is given to the paired data of the *Rain* scene, the description is “increase the sound of rain”; if (f) is given for car.horn and (c) for dog, the description is “make car horn lower and add dog bark.”

4.2. Experimental conditions

We used 10% of the development set for validation. The optimizer is Adam [33]. The number of epochs was set to 300, and the model with the smallest loss of validation data 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 within the K th rank of the retrieval result.

We compared three models: the proposed method without *classif*. loss ($\rho = 0$), the proposed method with *classif*. loss ($\rho = 1$), and the baseline method. For the baseline, we used the method of retrieving the target audio using only the sample audio without auxiliary text. The parameters of the audio encoder were the same as in the proposed method. Therefore, $z_{k,l} = d_{\text{cosim}}(\mathcal{A}(a_k), \mathcal{A}(b_l))$ was applied to Eq. (6) instead of Eq. (5) In a preliminary study, we examined another method that takes an existing captioning system and a text query-modifier. However, it was not as effective as the baseline method using a sample audio query, so we did not include it in the comparison.

4.3. Results

4.3.1. Comparison between proposed method and baseline

We conducted an experiment to compare the proposed method with the baseline, the results of which are shown in Table 2. Bold font indicates the highest scores. The retrieval accuracy of the proposed method was higher than that of the baseline method in all conditions. Table 3 shows the Recall@1 for each background and event sound that was given a difference. “background” is “rain” in the *Rain* scene and “car_passing_by” in the *Traffic* scene. The results show that the proposed method outperformed the baseline method for all background and event sounds. The retrieval accuracy of the differences in the background audio of each scene, “rain” and “car_passing”, was significantly lower than the others. Thus, dealing with differences in the background audio was more difficult than dealing with one in the audio event.

4.3.2. UMAP visualization of embedding vector

To verify that the proposed method handles difference information appropriately, we visualized the embedding vectors in the latent space. For visualization, we created 100 datasets each

¹The dataset is available at <https://github.com/nttclab/apwd-dataset>

Table 2: Comparison between proposed and baseline method

Method	Text info.	Classif. loss	Rain			Traffic		
			R@1	R@5	R@10	R@1	R@5	R@10
(a) Baseline	×	×	0.256	0.611	0.699	0.260	0.500	0.593
(b) Proposed w/o classific. loss	✓	×	0.388	0.681	0.745	0.361	0.590	0.675
(c) Proposed w/ classific. loss	✓	✓	0.445	0.721	0.769	0.391	0.622	0.695

Table 3: R@1 comparison for different audio events

Method	Scene	background* ¹	dog* ²	chirping_bird* ²	thunder* ²	footsteps* ²	car_horn* ²	church_bells* ²
(a) Baseline	Rain	0.035	0.304	0.205	0.346	0.295	N/A	N/A
(b) Proposed w/o classific. loss	Rain	0.061	0.448	0.360	0.490	0.450	N/A	N/A
(c) Proposed w/ classific. loss	Rain	0.061	0.538	0.438	0.534	0.522	N/A	N/A
(a) Baseline	Traffic	0.056	0.327	0.164	N/A	N/A	0.369	0.294
(b) Proposed w/o classific. loss	Traffic	0.092	0.429	0.298	N/A	N/A	0.470	0.396
(c) Proposed w/ classific. loss	Traffic	0.115	0.502	0.327	N/A	N/A	0.495	0.435

*¹The background audio is given a difference of increase or decrease. “bg” is “rain” in the *Rain* scene, and “car_passing_by” in the *Traffic* scene.

*²The event audio is given a difference of an addition, deletion, increase, or decrease.

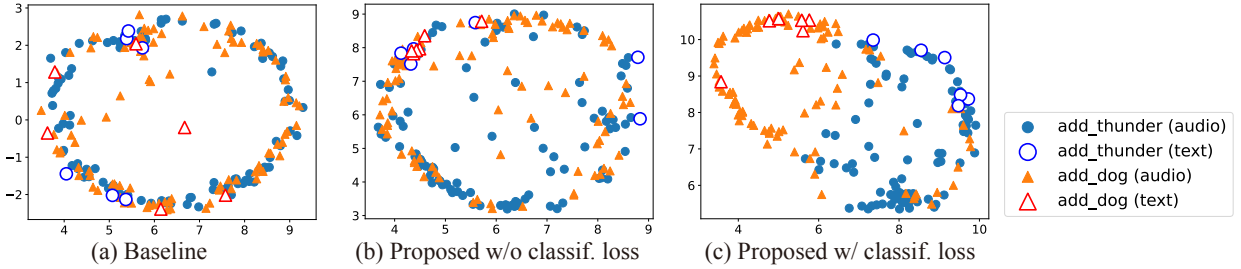


Figure 3: UMAP visualization of the difference in audio embedding vectors and text embedding vector. In (c), the proposed method with classification loss, the audio difference of “the addition of thunder” (blue circle) and that of “the addition of dog” (orange triangle) form different distributions for each type of difference. In addition, the text embedding (thunder: blue circle outline, dog: red triangle outline) belongs to the same distribution as the corresponding difference in audio embedding vectors.

for the difference between “the addition of thunder” and “the addition of dog”. The difference in the embedding vectors of audio data pairs, $\mathcal{A}(b) - \mathcal{A}(a)$, and the embedding vector of text, $\mathcal{T}(t)$, were visualized using UMAP [34].

The visualization results are shown in Fig. 3, where (a), (b), and (c) show the embedding vectors for the baseline, the proposed method without classification loss, and the proposed method with classification loss, respectively. In the baseline, embedding vectors were placed regardless of the type of difference. In the proposed method without classification loss, the distributions of the embedding vectors of “the addition of thunder” and “the addition of dog” were slightly biased to the lower right and upper left, respectively. In the proposed method with classification loss the distributions of the embedding vectors of “the addition of thunder” and “the addition of dog” were clearly biased to the lower right and upper left, respectively. Focusing on the text embedding vectors, they were placed such that they belonged to the same distribution as the corresponding difference of audio embedding vectors in the proposed method with classification loss. Therefore, the training classification task for audio data should enhance the recognition of differences.

5. Conclusion

We proposed a content-based audio retrieval method that can retrieve target audio that is similar to but slightly different from the query audio by introducing an auxiliary text-query modifier which describes the difference between the query and the target audio. The proposed method adjusts the retrieval range to obtain the target sound by adding the embedding vector of the auxiliary text-query modifier to that of the query audio in shared latent space. We experimentally verified that the proposed method can obtain audio data with the difference from the query sound by utilizing the information in the introduced auxiliary text. After visualizing the embedding vectors in the latent space, we also verified that the proposed method learns the relation between the audio difference and its textual representation.

Future work includes improving the accuracy of the search for increases and decreases in background so that it as accurate as the search for differences in audio events. We also aim to collect data from actual recordings annotated by humans to build a model that can handle a wider variety of descriptions and differences.

6. References

- [1] J. F. Gemmeke, D. P. 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)*. IEEE, 2017, pp. 776–780.
- [2] 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.
- [3] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. Conf. N. Am. Chapter Assoc. Comput. Linguist.*, 2019, pp. 119–132.
- [4] K. Drossos, S. Adavanne, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 736–740.
- [5] D. Takeuchi, Y. Koizumi, Y. Ohishi, N. Harada, and K. Kashino, "Effects of word-frequency based pre- and post- processings for audio captioning," in *Proc. Detect. Classif. Acoust. Scenes Events Workshop (DCASE)*, November 2020.
- [6] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," in *Proc. Interspeech*, 2021, pp. 571–575.
- [7] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.
- [8] B. Elizalde, S. Zarar, and B. Raj, "Cross modal audio search and retrieval with joint embeddings based on text and audio," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2019, pp. 4095–4099.
- [9] A.-M. Oncescu, A. Koepke, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," in *Proc. Interspeech*, 2021.
- [10] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Trans. Multimedia*, 2022.
- [11] H. Xie, S. Lipping, and T. Virtanen, "Dcase 2022 challenge task 6b: Language-based audio retrieval," *arXiv preprint arXiv:2206.06108*, 2022.
- [12] P. Panyapanuwat, S. Kamonsantiroj, and L. Pipanmaekaporn, "Unsupervised learning hash for content-based audio retrieval using deep neural networks," in *Proc. 11th Int. Conf. Knowl. Smart Technol. (KST)*. IEEE, 2019, pp. 99–104.
- [13] S. Ikawa and K. Kashino, "Acoustic event search with an onomatopoeic query: measuring distance between onomatopoeic words and sounds," in *Proc. Detect. Classif. Acoust. Scenes Events (DCASE) Workshop*, 2018, pp. 59–63.
- [14] P. Manocha, R. Badlani, A. Kumar, A. Shah, B. Elizalde, and B. Raj, "Content-based representations of audio using siamese neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2018, pp. 3136–3140.
- [15] B. Kim and B. Pardo, "Improving content-based audio retrieval by vocal imitation feedback," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2019, pp. 4100–4104.
- [16] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap: Learning audio concepts from natural language supervision," *arXiv preprint arXiv:2206.04769*, 2022.
- [17] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE multimedia*, vol. 3, no. 3, pp. 27–36, 1996.
- [18] S. Sundaram and S. Narayanan, "Audio retrieval by latent perceptual indexing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2008, pp. 49–52.
- [19] T. Mäkinen, S. Kiranyaz, J. Raitoharju, and M. Gabbouj, "An evolutionary feature synthesis approach for content-based audio retrieval," *EURASIP J. Audio Speech Music Process.*, vol. 2012, no. 1, pp. 1–23, 2012.
- [20] S. Kim, P. Georgiou, S. Narayanan, and S. Sundaram, "Using naïve text queries for robust audio information retrieval," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2010, pp. 2406–2409.
- [21] G. Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE trans. Neural Netw.*, vol. 14, no. 1, pp. 209–215, 2003.
- [22] D. Surís, A. Duarte, A. Salvador, J. Torres, and X. Giró-i Nieto, "Cross-modal embeddings for video and audio retrieval," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, pp. 711–716.
- [23] A. W. Boggust, K. Audhkhasi, D. Joshi, D. Harwath, S. Thomas, R. S. Feris, D. Gutfreund, Y. Zhang, A. Torralba, M. Picheny, and J. Glass, "Grounding spoken words in unlabeled video," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 29–32.
- [24] A. Rouditchenko, A. Boggust, D. Harwath, B. Chen, D. Joshi, S. Thomas, K. Audhkhasi, H. Kuehne, R. Panda, R. Feris, B. Kingsbury, M. Picheny, A. Torralba, and J. Glass, "AVLNet: Learning audio-visual language representations from instructional videos," in *Proc. Interspeech*, 2021.
- [25] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "AudioCLIP: Extending clip to image, text and audio," *arXiv preprint arXiv:2106.13043*, 2021.
- [26] 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. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 131–135.
- [27] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.
- [28] 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*, 2019.
- [29] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.
- [30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.
- [31] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proc. 23rd Annual ACM Conf. Multimedia*. ACM Press, pp. 1015–1018.
- [32] 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)*. IEEE, 2017, pp. 344–348.
- [33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. in Int. Conf. Learn. Represent. (ICLR)*, 2014.
- [34] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.