CLAP4SED: TRAINING-FREE MULTIMODAL FEW-SHOT RETRIEVAL FOR REAL-TIME SOUND EVENT DETECTION ON EMBEDDED DEVICES

Wei-Cheng Lin, Irtsam Ghazi, Ajit Belsarkar, Luca Bondi, Samarjit Das, Ho-Hsiang Wu

Robert Bosch LLC, USA
wei-cheng.lin@us.bosch.com

ABSTRACT

Implementing real-time *sound event detection* (SED) on embedded devices poses significant challenges, primarily related to generalizability and complexity. Existing SED models are predominantly suited for closed-form recognition, making adaptation to new or unseen sound classes difficult. While recent advancements in *audio foundation models* (AFM) such as CLAP offer potential for open-form sound event classification, they often come with substantial model complexity, rendering them impractical to deploy on embedded devices for real-time tracking. In this study, we introduce the CLAP4SED framework, a training-free, real-time SED solution derived from CLAP that can be flexibly deployed across various open-ended scenarios on embedded devices. Our experimental results conducted on three publicly available datasets demonstrating the competitive SED accuracy with less than 100ms latency under Ambarella CV22 camera chip setup.

Index Terms— sound event detection, few-shot prompt engineering, audio foundation models, embedded AI systems

1. INTRODUCTION

Audio has become a popular sensory modality for monitoring our environment, it complements vision in better handling of occlusions and can support omnidirectional signal. Audio sensors have been deployed to real-world environment for applications such as noise monitoring in urban areas [1], tracking avian diversities [2] and bird migrations [3]. These *sound event detection* (SED) [4] solutions are usually deployed as embedded systems with computational resource constraints, requiring constant monitoring and handling of input signal streams, and supporting diverse characteristics of sounds such as gunshot [5], glass breaking, baby crying, and screaming [6], etc.

Recent advancements in *audio foundation models* (AFM) provide a promising solution to bridge the generalization gaps encountered with unseen acoustic events or conditions. There are two main campaigns for building AMF: First, *contrastive language-audio pretraining* (CLAP) [7], trained with large amount of audio captioning data [8, 9] contrastively, sometimes with the aid of ChatGPT-assisted caption generation [10]. Second, audio encoders are trained to adapt towards *large language models* (LLMs) such as Pengi [11], *listen, think, and understand* (LTU) [12], Qwen-Audio [13], and SALMONN [14]. These AFM unlock free-form natural language interactions with audio data and provide new avenues for embedded audio AI solutions. There has also been a paradigm shift from collecting data tailored for specific downstream tasks and training models in a supervised manner to utilizing these AFM for rapid prototyping with zero-shot capabilities, and further adapting with few-shot examples [15,16]. However, AFM typically rely on computational heavy model architectures, especially when they accompany with additional language models. This imposes another critical challenge for utilizing AFM under the embedded device setups [17].

Recently, it has been a surge of interest in adapting the CLAP model for offline audio analytic techniques, such as zero-shot audio classification or retrieval via natural language prompts [18, 19]. However, there is a noticeable gap in utilizing CLAP for on-device real-time applications such as SED. To bridge this gap, we propose CLAP4SED in this study, which is a method that utilizes a pretrained lightweight CLAP model for real-time SED tasks. This approach is designed to be executable on embedded devices, facilitating flexible adaptation of SED to handle various deployment environments. More specifically, we decouple the query step from the original CLAP inference stage and devise an offline multimodal few-shot retrieval pipeline to achieve real-time SED. We experiment with several prompting strategies from zero-shot to few-shot scenarios and discuss corresponding constraints in practical applications. We also highlight several design choices and trade-offs deploying these SED models to real-world embedded devices. The main contributions of this study are:

- We propose a training-free, real-time SED solution based on the novel multimodal retrieval framework, which aims to be executable on embedded devices for practical deployment.
- We provide comprehensive experimental results and highlight the design choices between the model performance and complexity for CLAP4SED.
- To best of our knowledge, we are the first work that explicitly leverages CLAP to perform on-device real-time SED.

2. PROPOSED FRAMEWORK

The proposed full framework consists of two main steps: A). building a backbone AFM optimized for operation on embedded devices, B). the multimodal few-shot retrieval system to perform real-time SED predictions.

2.1. CLAP Pretraining

We implement the CLAP model as our AFM for the first step. The CLAP training involves in audio $f_A(\cdot)$ and text $f_T(\cdot)$ encoders to process incoming pairs of audio sequence X_a and the corresponding caption descriptions X_t. This results in the audio $E_a = f_A(X_a)$ and text $E_t = f_T(X_t)$ embeddings, respectively. The model is then trained to optimize the symmetric similarity contrastively (Eq. 1) in a joint multimodal space for audio-text pairs containing within a mini batch size B, where η is a temperature parameter to scale the output ranges. More details can be found in [7].

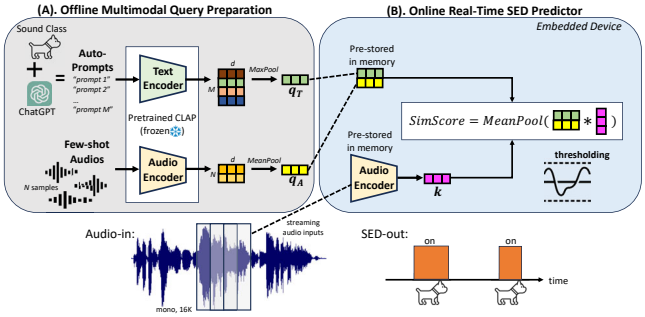


Figure 1: Overview of the proposed CLAP4SED framework for real-time SED on embedded device.

$$\mathcal{L} = \frac{1}{2B}\sum_{B}[\log\, diag(softmax(\eta(E_a \cdot E_t^{\top}))) + \log\, diag(softmax(\eta(E_t \cdot E_a^{\top})))] \quad (1)$$

However, existing pretrained CLAP models [10, 20] are majorly focusing on recognition performance without much consideration of model complexity, which might not be affordable to deploy on embedded devices. To accommodate this restriction, we substitute the conventional audio encoder from Transformers-based (e.g., HTSAT [21]) to lightweight CNN-based (PANNs [22]) family architecture. Since compare or compete embedded *machine learning* (ML) approaches is not our paper focus, we choose a relatively naïve method to obtain lightweight encoder for simplicity of the proposed framework. While beyond the scope of this study, it's worth noting that various advanced model compression techniques such as quantization or distillation [23] could be considered to further improve the backbone AFM performance. As it is unavoidable to balance model efficiency with performance, we present Table 1 to benchmark our retrained lightweight CLAP encoder, providing a reference for this trade-offs. For other training configurations, we follow closely the standard recipe of CLAP works [7, 20] and discuss in Section 3.1.

2.2. CLAP4SED: Multimodal Few-shot Retrieval for SED

The core idea to leverage a retrieval-based AFM for SED is to decouple the query component from the original CLAP inference stage. Figure 1 provides an overview of this framework. Specifically, the desired queries are calculated offline and stored in advance on the embedded devices. This approach eliminates the model complexity of the entire text modality (i.e., LLMs), thereby substantially lowering memory and computational demands and enabling operation on small embedded devices. However, the robustness and representativeness of pre-computed queries emerge as the most crucial factors for accurate SED predictions.

A). Offline Query Preparation: we propose to utilize multimodal information for obtaining effective query prototypes (see the top-left gray box in Figure 1), assuming N few-shot audio samples are available on hand per interested sound event. For the audio part, the trained audio encoder $f_A(\cdot)$ is applied to extract audio embeddings from the given few-shot samples. Following by a mean-pooling operation to summarize the audio prototypical vector as the final audio query $q_A \in \mathbb{R}^{1\times d}$, where d represents the dimension of hidden space that performs audio-text contrastive learning in the CLAP model. As for the text query preparation, we first employ GPT-4[1] to rewrite the convention CLAP retrieval template (e.g., "*this is the sound of [class label]*") for enriching text expressions [10] into M different prompts. These gpt-generated retrieval prompts are then fed into the trained text encoder $f_T(\cdot)$ to obtain embeddings. An audio-informed max-pooling operation over prompts is conducted, which only returns the most relevant (i.e., maximum dot-product similarity) prompt embedding to the given few-shot audio embeddings. This results in the final text query $q_T \in \mathbb{R}^{1\times d}$. From the high-level standpoint, audio query guides the specificity of retrieval outcomes while text query enhances additional diversity from different modality perspective for better robustness.

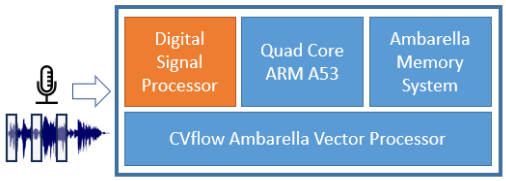


Figure 2: High-level structure of the embedded hardware setup based on Ambarella CV22 chip.

B). Online SED Predictor: only the lightweight audio encoder (Sec. 2.1) and modality-specific query vectors need to be pre-stored in the embedded device, as depicted in the top-right blue box in Figure 1. Upon receiving an input audio streaming data chunk (with window size L), encoder $f_A(\cdot)$ extracts it to generate key embeddings $k \in \mathbb{R}^{1\times d}$, which is then used to calculate a predefined similarity criteria with the prepared queries, thereby forming the averaged decision score across modalities. Finally, a simple binary thresholding is applied to determine the activity of sound event for that specific timeframe. The minimum real-time prediction time grid, denoted as τ, depends on the overall latency of the prediction process.

3. EXPERIMENTAL SETUPS

3.1. Embedded System, Pretraining and Configurations

We use the Ambarella CV22 chip [24] to construct the embedded system environment, which is typically used for IP cameras. The CV22 chip comes equipped with a quad core ARM A-53 Linux enabled processor, 1MB L2 cache, an Neon SIMD accelerator for *digital signal processing* (DSP), and a *computer vision* (CV) flow vector processor for deep learning matrix operations. The Neon chip can effectively accelerate the *Fast-Fourier transform* (FFT) for spectrogram computations. Figure 2 shows a high-level structure of the hardware components we used for running computational cost analysis in Section 4.3.

For the CLAP model pretraining, we use Adam (lr=0.0001) to optimize the standard contrastive loss (Eq. 1) based on the AudioCaps, Clotho, FSD50K, MACS, and WavCaps [10] train datasets. The default audio encoder $f_A(\cdot)$ is PANN10 [22] architecture unless specified in the results. We use the pretrained CLIP [25] text encoder to extract caption embeddings, the encoder $f_T(\cdot)$ is frozen all

[1]https://openai.com/gpt-4

the time during the training process. The hidden dimension d of the joint contrastive space is 512, temperature η is 0.07, and 128 batch size B training on a single NVIDIA-TESLA-V100 32GB GPU device. All the models are implemented in PyTorch.

For the real-time SED configurations, the streaming audio input is a 1 sec length (i.e., the sliding window size L), 16K, mono and 16-bit data chunk. Note that longer lengths require to register more memory buffer and increase the computational latency. The time grid of producing SED outputs is set to 0.1 secs (i.e., the window hop size τ), since our maximum prediction latency can be less than 100ms under the proposed framework. We assume 5-shot examples (N) are available per sound event in default unless specified. These few-shot samples are randomly selected from the corresponding validation or train data. We prompt GPT-4 with "*what are the sounds of [class label] ?*" to produce 30 (M) diversified but relevant enough sound descriptions for each target retrieval class. Cosine similarity is set as the criteria to measure the decision score.

Since our proposed framework is to perform real-time SED under practical industrial setup (e.g., security camera), the model is not receiving the full clip (global)-level context to compute advanced offline metrics such as PSDS scores [26]. Instead, the model only receives local segment input (e.g., 1 sec streaming chunk) during each inference period. Therefore, we calculate the *area under curve* (AUC) for segment-based precision-recall as our system evaluation metric, which is more suitable to evaluate on-device real-time SED and can comprehensively compare overall performance for the full threshold space.

3.2. Datasets

One important feature of the proposed framework is that we can quickly adapt the trained AFM towards different application scenarios without involving additional finetuning or retraining efforts on the embedded devices (i.e., training-free approach). Here, we showcase this flexibility by evaluating it on three diverse datasets for the domestic environments, urban sounds, and aggression events monitoring, respectively.

- **DESED** [27]: is composed of 10 domestic event classes (i.e., alarm/bell/ringing, blender, cat, dog, dishes, electric shaver/toothbrush, frying, running water, speech and vacuum cleaner) originating from the AudioSet [28]. We only utilize its evaluation set to report the system performance, which has 692 audio files (fixed 10 secs length for each) in total.
- **Urban-SED** [29]: synthesizes strong-labeled soundscapes from UrbanSound8K dataset using the SCAPER [29] tool, which consists of 10 city sounds (i.e., air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren and street music). Its evaluation set contains 2,000 audio files and each is 10 secs long.
- **Aggression-SED**: is our own curated evaluation subset by defining the violent or aggressive relevant events out of the AudioSet. We define 8 classes to be included (i.e., smoke/fire/car alarm, ambulance/defense/truck siren, explosion, fire, gunshot, screaming, shouting and smash/crash/breaking sound), which has a total of 1,495 audio files extracting from the AudioSet strong evaluation partition[2].

[2]https://research.google.com/audioset/download_strong.html

Table 1: Performance summary of our retrained lightweight CLAP encoders comparing to existing SOTA models. We evaluate the zero-shot classification (ZS) on UrbanSound8k (US8K) and ESC-50 based on macro F1 score (F1), as well as the text-to-audio (T2A) and audio-to-text (A2T) retrieval on Clotho using recall at 10 (R@10) metrics. All the results are in percentage scale.

	US8K (F1)	**ECS-50** (F1)	**Clotho** (R@10)	
	ZS	*ZS*	*T2A*	*A2T*
PANN14 [7]	73.2	82.6	-	-
HTSAT-LAION [20]	77.0	91.0	54.4	65.7
HTSAT-WavCaps [10]	80.6	94.8	50.9	56.6
PANN6 (ours)	68.1	68.9	33.8	35.1
PANN10 (ours)	72.5	78.0	37.8	42.3
PANN14 (ours)	77.7	85.3	42.5	48.1

3.3. Ablation Baselines

We want to highlight that our approach is incomparable to existing SED models, since we do not rely on any labeled data (except for a very limited few-shot audio examples) nor a particular training framework for SED. Instead, we conduct comparisons against ablation baselines focusing on the query design component to demonstrate the advantage of leveraging multimodal information for retrieval-based SED. Specifically, four single-modality baselines are compared by preparing the query vector q_T or q_A in different ways while everything else remains the same. These single-modality retrieval approaches are also commonly adopted from previous literatures.

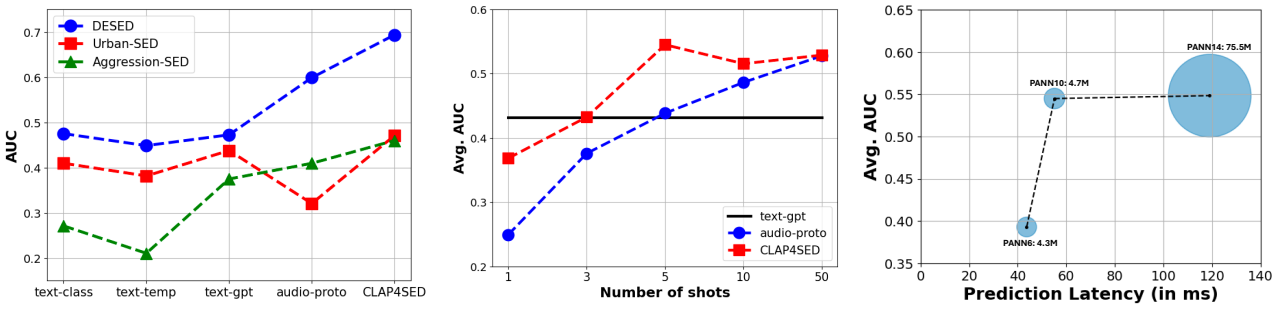
- **Class Prompt**: zero-shot audio retrieval using raw class labels (i.e., "*[class label]*") as input prompt for generating the text-only query q_T, denoted as *text-class*.
- **Template Prompt** [7]: zero-shot audio retrieval appending with natural language-alike template (i.e., "*this is the sound of [class label]*") to produce the text-only query q_T, denoted as *text-temp*.
- **GPT Prompt** [10]: same as we depicted in Figure 1 (the gray box) but only considers the text-only query q_T. We use mean-pooling operation instead of audio-informed max-pooling to summarize the query embedding, since we do not have available audio samples under the single-modality setting to compute the most relevant prompt. We denote this baseline as *text-gpt*.
- **Audio Prototypes** [30]: same as we depicted in Figure 1 (the gray box) but only considers the audio-only query q_A to form an audio-to-audio retrieval task. As q_A serves as the prototypical vector of audios, we denote it as *audio-proto*.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. System Performance Comparison

Figure 3(a) summarizes the evaluation results of single-modality baselines (Sec. 3.3) versus proposed CLAP4SED method across three datasets (Sec. 3.2). There are three major points to focus on:

First, we can observe that the text-gpt approach generally obtains higher performance comparing to other text-only query methods (i.e., text-class and text-temp), especially for the Aggression-SED. It might due to LLMs can effectively enrich language diver-



(a) (b) (c)

Figure 3: Comparison of system performance under different scenarios and model configurations. (a) Performance across SED scenarios for different queries. (b) Influence of few-shot numbers. (c) Performance and complexity trade-offs.

sity to incorporate a broader spectrum of common sense knowledge into the query space. This augmentation contributes to a more comprehensive coverage of sound event scenarios, increasing the model's generalizability.

Second, we can see that the audio-proto approach can outperform text-only schemes in the cases of DESED and Aggression-SED. However, its performance significantly deteriorates in the Urban-SED scenario, indicating a notable issue with robustness. While few-shot references can offer advantages for specific prototypes (i.e., retrieval specifications), they also impose limitations on generalization capability as these collected few samples on hand might not be sufficient to represent the full event space. This compromises the model's robustness against diverse scenarios, limiting its practical applicability.

Last, the proposed CLAP4SED framework effectively reconciles the trade-offs inherent in both text-only and audio-only approaches by leveraging the benefits of multimodal fusion. The text-gpt query q_T enhances model generalization, addressing potential representational gaps in the audio-proto q_A. Meanwhile, the audio-proto offers supplementary guidance on recognition precision, thereby complementing each other's information. As a result, CLAP4SED consistently achieves the best system performance across three datasets over all the single-modality approaches.

4.2. Few-shot Capability Analysis

This section discusses how the few-shot number (N) impacts on system performance that involves in utilizing audio samples as query (i.e., audio-proto and CLAP4SED). Figure 3(b) illustrates the averaged AUC results across three datasets, and we pick text-gpt as the benchmark representative since it obtains the most competitive performance among the text-only query approaches. We can observe a consistent trend where increasing the number of few-shot audios leads to an improvement in overall SED performance. Upon gathering 5-shot examples, both audio-proto and CLAP4SED outperform the text-only query approaches. Interestingly, the proposed CLAP4SED demands significantly fewer audio samples compared to the audio-proto approach. Its performance with 5-shots can achieve comparable results to 50-shots for audio-proto (this trend holds true for 1-shot and 3-shots cases as well). This characteristic has significant importance for practical applications, as it is often infeasible to gather as many supervised shots in most cases. With the advantage to collect just 5 examples for new environments or undefined sound events, CLAP4SED can rapidly deploy and adapt to various real-world scenarios by simply updating the query vectors without additional training steps, resulting in an effective training-free solution.

4.3. Computational Trade-Offs

We also provide the computational trade-offs of CLAP4SED based on the configured embedded system setup (Sec 3.1) as a reference for future development. Figure 3(c) visualizes the result for PANN6, PANN10 and PANN14 architectures. The radius of a circle indicates the model size in number of parameters, x-axis represents overall prediction latency in milliseconds and y-axis shows the corresponding SED performance in averaged AUC. We can see that there is a clear sweet point of using the PANN10 encoder. It significantly improves the overall recognition accuracy from PANN6 with acceptable model size (4.3M to 4.7M) and latency (45ms to 55ms) increases. On the other hand, PANN14 only brings a very limited performance improvement from PANN10, but drastically escalates the computational requirements (e.g., latency increases from 55ms to 120ms). Prediction latency is a critical factor that cannot be compromised in real-time detection setups. Therefore, PANN10 emerges as the most recommended encoder for the CLAP4SED framework, striking a balance between recognition performance and computational efficiency.

5. CONCLUSION

In this study, we introduce the CLAP4SED framework, which utilizes the CLAP foundation model to enable real-time SED on embedded devices. The core innovation lies in decoupling the query component from the CLAP retrieval pipeline. This allows for significant reduction in model complexity and flexible adaptation to various SED scenarios, resulting in an efficient training-free solution. Notably, our experimental results showcase the effectiveness of the proposed few-shot multimodal query approach, which effectively combines the advantages of both text and audio modalities, thereby bridging modality gaps. Additionally, we provide comprehensive design choices and trade-offs analysis as a reference for future development endeavors.

6. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "SONYC: A system for monitoring, analyzing, and mitigating urban noise pollution," *Communications of the ACM*, vol. 62, no. 2, pp. 68–77, 2019.

[2] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck, "BirdNET: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021.

[3] V. Lostanlen, A. Cramer, J. Salamon, A. Farnsworth, B. M. Van Doren, S. Kelling, and J. P. Bello, "BirdVox: Machine listening for bird migration monitoring," *bioRxiv*, pp. 2022–05, 2022.

[4] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[5] D. Mares and E. Blackburn, "Acoustic gunshot detection systems: a quasi-experimental evaluation in st. louis, mo," *Journal of experimental criminology*, vol. 17, pp. 193–215, 2021.

[6] A. Suliman, B. Omarov, and Z. Dosbayev, "Detection of impulsive sounds in stream of audio signals," in *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*. IEEE, 2020, pp. 283–287.

[7] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "CLAP: Learning audio concepts from natural language supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[9] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[10] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3339–3354, 2024.

[11] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. R. Glass, "Listen, think, and understand," in *The Twelfth International Conference on Learning Representations*, 2024.

[13] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou, "Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv:2311.07919*, 2023.

[14] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. MA, and C. Zhang, "SALMONN: Towards generic hearing abilities for large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[15] Z. Lin, S. Yu, Z. Kuang, D. Pathak, and D. Ramanan, "Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 325–19 337.

[16] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, B. Patra, *et al.*, "Language is not all you need: Aligning perception with language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[17] Y. Chen, B. Zheng, Z. Zhang, Q. Wang, C. Shen, and Q. Zhang, "Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.

[18] J. Liang, X. Liu, H. Liu, H. Phan, E. Benetos, M. D. Plumbley, and W. Wang, "Adapting language-audio models as few-shot audio learners," in *Proc. INTERSPEECH 2023*, 2023, pp. 276–280.

[19] W.-C. Lin, S. Ghaffarzadegan, L. Bondi, A. Kumar, S. Das, and H.-H. Wu, "CLAP4Emo: Chatgpt-assisted speech emotion retrieval with natural language supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2024)*, Seoul, Korea, April 2024, pp. 11 791–11 795.

[20] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[21] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.

[22] 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 Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[23] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *International Conference on Learning Representations*, 2018.

[24] Ambarella Inc., "Product brief on ambarella cv22: Computer vision soc for ip cameras," 2021.

[25] 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," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[26] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 61–65.

[27] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[28] 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[29] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[30] S. Deshmukh, B. Elizalde, and H. Wang, "Audio retrieval with WavText5K and CLAP training," in *Proc. INTERSPEECH 2023*, 2023, pp. 2948–2952.