PEPE: PLAIN EFFICIENT PRETRAINED EMBEDDINGS FOR SOUND EVENT DETECTION

Technical Report

Yongqing Wang[‡]*, Heinrich Dinkel*[‡]*, Zhiyong Yan, Junbo Zhang, Yujun Wang*

Xiaomi Corporation, Beijing, China
{wangyongqing3,dinkelheinrich,yanzhiyong,zhangjunbo1,wangyujun}@xiaomi.com

ABSTRACT

This paper is a system description of the XiaoRice team submission to the DCASE 2023 Task 4 challenge. In light of the increasing availability of pretrained audio embedding models, our research addresses the need for efficient utilization of these resources, taking into account their environmental impact. Our method named plain efficient pretrained (audio) embeddings (PEPE) integrates a linear classifier or a bidirectional gated recurrent network (BiGRU) with those embeddings while prioritizing energy efficiency, training speed and minimizing carbon emissions. By employing a streamlined approach, we demonstrate that a linear classifier with 52K parameters surpasses the challenge baseline for PSDS-2 scores, highlighting the potential of eco-friendly solutions in achieving superior performance. We achieve a polyphonic sound detection score (PSDS)-1 score of 53.44 via a 6-way ensemble and a PSDS-2 score of 88.60 with a simple linear classifier using PEPE. Through our work, we aim to emphasize the adoption of environmentally conscious practices in the field.

Index Terms— Semi-supervised learning, Weakly supervised learning, Transformers, Linear classification.

1. INTRODUCTION

This paper presents a system developed for the DCASE 2023 Task 4 challenge, which focuses on modeling audio signals for sound event detection (SED). In SED, the primary objective is to classify or tag an event, along with its corresponding onset and offset timings.

Currently, SED can be used for a variety of applications, such as an aid for the hearing impaired, smart cities and homes [1], audio-to-text retrieval [2], voice activity detection [3, 4] and audio captioning [5, 6]. Most current approaches within SED utilize neural networks, in particular convolutional neural networks [7, 8] (CNN), convolutional recurrent neural networks [9, 10, 11] (CRNN) and transformers [12, 13].

The paper is structured as follows. Section 2 describes our core system idea. Further, Section 3 introduces the experimental setup and Section 4 displays our achieved results. Finally, Section 5 concludes the work.

2. SYSTEM

Since the use of external data is allowed in this challenge, our objective is to harness large-scale pretraining to extract high-level embeddings for sound event detection (SED). In contrast to previous approaches that involve training large extensive models specifically designed for SED [13, 14], our work relies exclusively on pretrained embeddings as the primary input. The process of extracting embeddings involves mapping the raw audio data into a latent space representation. This latent space preserves important characteristics and discriminative features of the audio, enabling effective classification for downstream tasks such as SED. The use of pretraining allows us to leverage a large amount of data, enabling our model's to learn robust representations that generalize well to unseen audio samples. This system offers the advantages of overall low complexity (when disregarding pretraining costs), fast training speed, and a reduced carbon footprint. Our proposed embeddings are extracted from a variety of Vision Transformer (ViT) [15] models, which have been tailored and optimized for audio-based tasks.

We overall utilize two neural network-based classifiers for the embeddings: One is capable of predicting on- and offsets by using a bidirectional gated recurrent unit (BiGRU) network, while the other uses a simple linear classifier. Each model is optimized towards one of the challenge metrics being polyphonic sound detection score (PSDS) [16] 1 (on- and offset sensitive) and 2 (tagging). The model architecture optimized for PSDS-1 can be seen in Table 1. It operates on an input embedding of size $T \times D$, where T denotes the number of tokens and D represents the embedding dimension. Our approach begins by interpolating T to a target output resolution of T_{tar} and subsequently reducing the dimensionality to 128. The resulting 128-dimensional embedding is then fed into a BiGRU, responsible for predicting time stamps for each sound event. Lastly, we leverage an attention-based pooling method to compute the average output scores for individual sound events, enabling the utilization of weakly labelled data.

Layer	Output size
InputEmbed	$T \times D$
Interpolation	$T_{tar} \times D$
Linear	$T_{tar} \times 128$
BiGRU	$T_{tar} \times 256$
Attentionpool	$T_{tar} \times 10$

Table 1: The proposed model for PSDS-1 optimization. T_{tar} is the (interpolated) target length of the output.

Additionally, the model optimized for coarse tagging performance (PSDS-2) is showcased in Table 2. In this architecture, a basic linear transformation of the input is computed by taking the average across the time dimension T. Subsequently, this transformed representation is fed into a linear classifier, responsible for predicting the presence of a sound event.

[‡] equal contribution.

Layer	Output size
InputEmbed	$T \times D$
Linear	$T \times 256$
Mean	256
Linear	10

Table 2: The proposed model for PSDS-2 optimization.

2.1. Embeddings

In the following, we describe the embeddings we used for the ensembles of our submissions. If not otherwise stated, the pretrained models utilize 64-dimensional log-Mel-spectrograms extracted with a window size of 32 ms at a rate of 10 ms at 16kHz. All transformer models use a patch-size of 16×16, with no overlap between patches, which results in $T = 248$ (4 tokens in frequency and 62 in time) for each model. We further also include the baseline BEATs embedding [17], which has $T = 496$ tokens, due to the use of 128-dimensional Mel-spectrograms.

Embeddingname	Size	Backbone	mAP
Tiny	248×192	ViT-Tiny	44.56
Small	248×384	ViT-Small	46.25
Base	248×768	ViT-Base	47.54
BEATs	496×768	ViT-Base	*45.40
Large	248×1024	Vit-Large	48.32

Table 3: A summary of the embeddings utilized in this study. AUnless otherwise specified, all embeddings have been pretrained on Audioset. The corresponding mAP scores on Audioset are also provided for each embedding. The size is indicated as $T \times D$. Results marked with $\star$ indicate finetuning on a semi-supervised level, making them not directly comparable.

These embedding extractors were all pre-trained on Audioset, whereas we provide the mean average precision (mAP) for each model in Table 3. It is worth mentioning that our proposed embeddings were not obtained through teacher-student training, distinguishing them from BEATs [17].

2.2. Stacking Embeddings

One of the simplest ways to improve performance is to concatenate or stack embeddings together over the embedding dimension D. However, when dealing with embeddings of different token lengths T, an interpolation strategy is employed to address the mismatch between different token lengths. Since the number of tokens T differs between the embeddings, we opt for a simple interpolation strategy. To accomplish this, we first pool the frequency-token dimension for each embedding, resulting in 62 tokens (160 ms per token) for a given 10-second input. Then, we stack all embeddings across the embedding dimension, which yields a comprehensive high-level embedding. In this work, we utilize three stacked embeddings, which are introduced in Table 4.

2.3. Training framework

We follow our previous work [14], where we utilize Mean Teacher (MT) [18] for all PSDS-1 optimized models and unsupervised data augmentation (UDA) [19] for PSDS-2 optimized models. Since our work focuses on utilizing pretrained embeddings to their fullest, we do not submit a non pretrained model.

Embeddingname	Size	Source-Embeddings
BeST	62×1344	BEATs, Small, Tiny
LBST	62×2368	Large, Base, Small, Tiny
BeLBST	62×3136	BEATs, Large, Base, Small, Tiny

Table 4: The stacked embeddings used in this work.

1. SINGLE is a monolithic model approach, where a single model is utilized for the task.
2. SED is specifically designed and optimized to achieve high performance on the PSDS-1 metric.
3. L-TAG (Linear Tag) is specifically designed and optimized to achieve high performance on the PSDS-2 metric with as few resources as possible.

3. EXPERIMENTAL SETUP

3.1. Dataset

The DCASE 2023 Task 4 dataset is split into a development (used for training) and an evaluation section. The development set is further split into training and validation sections. The training section contains three datasets $\mathcal{D}_{weak}, \mathcal{D}_{syn}, \mathcal{D}_{un}$:

$$\begin{aligned} \mathcal{D}_{weak} &= \{(x_1, y_2), (x_2, y_2), \ldots, (x_N, y_N)\}, \\ \mathcal{D}_{syn} &= \{(x_1, y_2), (x_2, y_2), \ldots, (x_M, y_M)\}, \\ \mathcal{D}_{un} &= \{x_1, \ldots, x_P\}. \end{aligned}$$

The $\mathcal{D}_{weak}$ and $\mathcal{D}_{syn}$ datasets are labeled and $\mathcal{D}_{un}$ only consists of audio data in a matching domain with $\mathcal{D}_{weak}$.

3.2. Training hyperparameters

Further, we denote use $\mathcal{R}$ to denote each model's output-label frame resolution. During training, if segments are shorter than 10 seconds, we zero-pad the input to the longest sample within a batch. During inference, we use a batch size of 1, such that padding has no effect.

All experiments start with a learning rate of 0.001 and are run for at most 200 epochs, with a linear warmup duration of 5000 iterations $\approx$ 50 epochs the Adam optimizer. Batch sizes are set to be 12 for weak and synthetic data and 24 for unlabeled data. The available weak training data is split into a 90% training and a 10% cross-validation portion. Cross-validation is done on the 10% held-out weak subset with the additional synthetic validation data. The training objective is the sum of the weak F1 and the intersection-F1 score, whereas training is stopped if the model did not improve for 15 epochs. Pytorch [20] was used as the neural network backbone.

For training, we use the standard binary cross-entropy (BCE) criterion. The following losses are employed during training:

$$\mathcal{L}_{\text{sup}} = \text{BCE}(\hat{y}, y), \{y, \hat{y}\} \in \mathcal{D}_{weak}, \quad (1)$$

$$\mathcal{L}_{\text{syn}} = \text{BCE}(\hat{y}_t, y_t), \{y_t, \hat{y}_t\} \in \mathcal{D}_{syn}, \quad (2)$$

$$\mathcal{L}_{\text{UDA}} = \mathcal{L}_{\text{Cstcy}}(\hat{y}^{\dagger}, \hat{y}) + \mathcal{L}_{\text{Cstcy}}(\hat{y}_t^{\dagger}, \hat{y}_t), x \in \mathcal{D}_{un}. \quad (3)$$

$$\mathcal{L}_{\text{MT}} = \mathcal{L}_{\text{Cstcy}}(\hat{y}^{\mu}, \hat{y}) + \mathcal{L}_{\text{Cstcy}}(\hat{y}_t^{\mu}, \hat{y}_t), x \in \mathcal{D}_{un}. \quad (4)$$

$$\mathcal{L}_{\text{unsup}}(x) = \begin{cases} \mathcal{L}_{\text{UDA}}(x) & \text{if UDA} \\ \mathcal{L}_{\text{MT}}(x) & \text{if MT} \end{cases}, \quad (5)$$

where $\hat{y}^{\dagger}$ is the model prediction of an augmented sample $x^{\dagger} = \text{Aug}(x)$ and $\hat{y}^{\mu}$ is the mean-teacher predicted label for a sample. For mean-teachers we follow the public DCASE2023 Task 4 baseline approach, while UDA is applied according to [10]. If not further stated we use BCE as the consistency loss $\mathcal{L}_{\text{Cstcy}}$. Each network is optimized using the sums of all introduced losses seen in Equation (6).

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{syn}} + \mathcal{L}_{\text{unsup}} \quad (6)$$

Note that we do not use data augmentation, since we believe that a simple linear layer requires little regularization. For all experiments in SED, we set $T_{tar} = 156$, to be identical to the baseline frame resolution.

3.3. Post-processing

If not further stated, we use the default median-filtering approach with a length of approximately 320ms.

3.4. SINGLE/SED

The models used for SED are introduced in Table 5. We provide information regarding the amount of trainable parameters (# Params), number of multiply–accumulate operations (MACs) and each model's respective target resolution $\mathcal{R}$. Further, our SINGLE model is chosen to be the best-performing model within the proposed models S1-S6. For all models, we train a 2-layer, 128-dimensional BiGRU (see Table 1), which is attached to the input embedding.

ID	Embedding-Name	#Params	MACs (M)	$\mathcal{R}$ (ms)
-	Baseline	2.6M	930	64
S1	Tiny	524K	81.25	64
S2	Small	549K	86.04	64
S3	Base	598K	93.75	64
S4	Large	630K	98.95	64
S5	BeST	671K	105.20	64
S6	BeLBST	901K	141.04	64
SED	Ensemble	3.8M	606	64

Table 5: Introduction to the models used for SED. The back-bone of each model follows Table 1.

3.5. TAG

For our PSDS-2 optimized submission, we focus on coarse-scale predictions of 2/5/10s respectively. Our L-TAG ensemble components are described in Table 6. Note that the majority of models have a label resolution of 10s, which can have a negative impact on performance since some very short bursts of sound events might not be detected. Thus we additionally use pseudo strong labels (PSL) [21], to predict labels on a scale of 2s/5s respectively. Note that the BEATs embedding uses 1-dimensional average pooling over the token dimension to align with the target resolution T_{tar}, while our proposed embeddings use standard nearest-neighbor interpolation, leading to a smaller amount of MACs.

ID	EmbeddingName	#Params	MACs (M)	$\mathcal{R}$ (s)
-	Baseline	2.6M	930	64
T1	Tiny	52K	12.50	10
T2	Small	101K	22.91	10
T2/PSL2s	Small	101K	22.91	2
T2/PSL5s	Small	101K	22.91	5
T3	Base	200K	47.91	10
T4	Large	264K	64.58	10
T5	BEATs	200K	97.91	10
T6	LBST	610K	37.50	10
T7	BeST	350K	20.83	10
L-TAG	-	2.0M	350	-

Table 6: Introduction to the models used for TAG. The back-end for each model can be seen in Table 2.

3.6. Ensemble

During model ensemble, the outputs from different models might output at different resolutions. In order to average these predictions, we nearest-neighbour upsample all model predictions to the highest resolution within an ensemble. Post-processing is applied after score averaging.

4. RESULTS

We report our results in terms of the two main challenge metrics denoted as PSDS-1 and PSDS-2 [22], where this year's challenge calculates threshold-independent PSDS [16]. Note that all results represent the performance on the held-out official development dataset.

4.1. System-1 (SINGLE)

The results regarding our system-1 submission can be seen in Table 7. Here we only use the previously introduced S5 model Table 5.

ID	**PSDS-1**	**PSDS-2**	Kwh
Baseline	50.00	72.60	1.82
SINGLE (S5)	**52.75**	**78.96**	**0.19**

Table 7: Results for our SINGLE system (submission 1), where no external data is used. Best results are in bold.

4.2. System-2 (SED)

Results regarding our proposed SED system can be seen in Table 8.

ID	**PSDS-1**	**PSDS-2**	Kwh
Baseline	50.00	76.20	1.82
S1	44.99	70.04	0.18
S2	48.77	76.01	0.15
S3	46.14	73.77	0.24
S4	46.38	71.37	0.16
S5	52.75	78.96	0.19
S6	51.31	73.18	**0.11**
SED	**53.44**	**81.10**	1.03

Table 8: Results for our SED model (submission 2), emphasizing accurate on- and offsets. Best results are displayed in bold.

4.3. System-3 (L-TAG)

Our results for the L-TAG model can be seen in Table 9. If during testing clips longer than 10s are provided we split these samples into 10s chunks and individually estimate scores for each chunk.

ID	**PSDS-1**	**PSDS-2**	Kwh
Baseline	**50.00**	76.20	1.82
T1	9.52	86.82	0.10
T2	8.89	87.55	0.11
T2/PSL2s	10.39	83.22	**0.06**
T2/PSL5s	11.67	87.00	0.11
T3	8.55	87.20	0.12
T4	9.04	86.92	0.16
T5	9.30	86.76	0.08
T6	8.90	87.65	0.16
T7	9.40	88.13	0.10
L-TAG	10.24	**88.60**	1.00

Table 9: Results for our L-TAG model (submission 3), focusing on coarse performance. The best models for each respective metric are in bold.

5. CONCLUSION

This paper presents the XiaoRice submission, named plain efficient pretrained embeddings (PEPE), for the DCASE2023 Task4 challenge. Our approach focuses on leveraging simple classifiers with pretrained audio transformer embeddings. PEPE represents one of the simplest yet effective approaches to achieve strong performance on the DCASE Task4 datasets. The SINGLE system obtains a PSDS-1 score of 52.75 and PSDS-2 score of 78.96, respectively without utilizing an ensemble. Second, our main SED submission to the challenge achieves a PSDS-1 score of 53.28, surpassing the baseline approach. Finally, our L-TAG ensemble method, utilizing a straightforward linear classification layer, attains an impressive PSDS-2 score of 88.60.

6. REFERENCES

[1] J. P. Bello, C. Mydlarz, and J. Salamon, *Sound Analysis in Smart Cities*. Cham: Springer International Publishing, 2018, pp. 373–397. [Online]. Available: https://doi.org/10.1007/978-3-319-63450-0{_}13

[2] S. Lou, X. Xu, M. Wu, and K. Yu, "Audio-text retrieval in context," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4793–4797.

[3] Y. Chen, H. Dinkel, M. Wu, and K. Yu, "Voice activity detection in the wild via weakly supervised sound event detection," *Proc. Interspeech 2020*, pp. 3665–3669, 2020.

[4] H. Dinkel, S. Wang, X. Xu, M. Wu, and K. Yu, "Voice Activity Detection in the Wild: A Data-Driven Approach Using Teacher-Student Training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1542–1555, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9405474/

[5] M. Wu, H. Dinkel, and K. Yu, "Audio caption: Listen and tell," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 830–834.

[6] X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2022 challenge task 6: Audio captioning with audio-text retrieval pre-training," DCASE2022 Challenge, Tech. Rep., July 2022.

[7] J. Yan, Y. Song, L.-R. Dai, and I. McLoughlin, "Task-Aware Mean Teacher Method for Large Scale Weakly Labeled Semi-Supervised Sound Event Detection," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings 2020*. Institute of Electrical and Electronics Engineers (IEEE), apr 2020, pp. 326–330.

[8] K. He, X. Shu, S. Jia, and Y. He, "Semi-supervised sound event detection system for dcase 2022 task 4," DCASE2022 Challenge, Tech. Rep., June 2022.

[9] H. Dinkel, M. Wu, and K. Yu, "Towards duration robust weakly supervised sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 887–900, 2021.

[10] H. Dinkel, X. Cai, Z. Yan, Y. Wang, J. Zhang, and Y. Wang, "A lightweight approach for semi-supervised sound event detection with unsupervised data augmentation," in *Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021), Online*, 2021, pp. 15–19.

[11] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," DCASE2022 Challenge, Tech. Rep., June 2022.

[12] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-augmented transformer for semi-supervised sound event detection," DCASE2020 Challenge, Tech. Rep., June 2020.

[13] K. Li, Y. Song, L.-R. Dai, I. McLoughlin, X. Fang, and L. Liu, "Ast-sed: An effective sound event detection method based on audio spectrogram transformer," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] H. Dinkel, Z. Yan, Y. Wang, M. Song, J. Zhang, and W. Wang, "A large multi-modal ensemble for sound event detection," DCASE2022 Challenge, Tech. Rep., June 2022.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words:

Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1021–1025.

[17] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[18] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 1195–1204.

[19] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised Data Augmentation for Consistency Training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2019, pp. 6256–6268. [Online]. Available: http://arxiv.org/abs/1904.12848

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8026–8037.

[21] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, and Y. Wang, "Pseudo strong labels for large scale weakly supervised audio tagging," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 336–340.

[22] C. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "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, may 2019, pp. 61–65. [Online]. Available: http://arxiv.org/abs/1910.08440