Integrating VGGSK and BEATs for Enhanced Sound Event Detection: A Semi-Supervised GRU-Based System with Weak Labels and Synthetic Soundscapes

Po Cheng Chan[1,2], Wei-Yu Chen[1], Chung Li Lu[1], Hsiang-Feng Chuang[1], Yu-Han Cheng[1], and Jia Ching Wang[2]

[1]Advanced Technology Laboratory, Chunghwa Telecom Laboratories, Taoyuan, Taiwan
[2]Department of Computer Science and Information Engineering, National Central University
[1]Email: *cbc, weiweichen, chungli, gotop, henacheng@cht.com.tw*
[2]Email: *jcw@csie.ncu.edu.tw*

***Abstract*—This paper presents the architecture we developed for the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 Challenge, specifically Task 4 on Sound Event Detection using Weak Labels and Synthetic Soundscapes. We integrated embeddings from VGGSK and BEATs, and employed a GRU-based model to classify sound events in each time frame. The system employs thresholding and smoothing techniques in its post-processing phase. For semi-supervised learning, we used the mean teacher approach with an Exponential Moving Average (EMA) strategy to update the teacher model's parameters. Pseudo-labels, generated by the student model, help leverage unlabeled data. Additionally, we applied data augmentation methods including mix-up, Gaussian noise, and embedding masking. With additional training data, our system achieved a Polyphonic Sound Detection Score (PSDS) of 0.529 for PSDS1 and 0.78 for PSDS2 on the validation dataset.**

I. Introduction

The objective of sound event detection is to identify the occurrence of specific sound events within an audio clip, including their onset and termination times. A significant challenge in employing supervised learning for this task lies in the high cost and potential bias associated with manually annotating sound data after its collection, as annotations can vary greatly among different reviewers. To address this, the DESED dataset [1] comprises audio files collected via two approaches: authentic ambient recordings and synthetic sounds. The dataset categorizes labels into three types: strong, weak, and unlabeled. Participants in the challenge are prompted to utilize the DESED dataset and are permitted to incorporate external datasets or pre-trained embeddings, although one submitted system must rely solely on DESED without using these additional resources, and at least one system should avoid using ensemble methods.

For enhancing the system that can utilize external datasets or pre-trained embeddings, the Bidirectional Encoder Representation from Audio Transformers (BEATs) [2] based on the Transformer architecture is employed to derive embedding features. Conversely, the system exclusively trains on the VGGSK model [3][4], a CNN-based architecture. The Exponential Moving Average (EMA) strategy is employed to update the VGGSK and Gated Recurrent Unit (GRU) [5] components during the teacher-student model update process. Data augmentation techniques such as mix-up [6], Gaussian noise, and ICT [7] are implemented on VGGSK inputs, while masking is applied to BEATs embeddings to enhance feature extraction. The primary aim is to boost predictive accuracy without extending inference time.

II. Methodology

This section will discuss the methods we used. Before employing the mean-teacher approach, we will first conduct supervised training on the CRNN and BEATs-VGGSK models. Subsequently, we will train the teacher model using a semi-supervised method to enhance the model's performance.

A. Feature Extraction

Due to variations in sampling rates, channel numbers, and audio file lengths in the DESED dataset, we employ the librosa library to normalize all audio files to a consistent 16000 Hz sampling rate and convert them to a mono channel format. Each audio file is padded to reach a standard length of 10 seconds through zero-padding. Subsequently, these audio files are converted into Mel-spectrograms by transforming the waveform signals and applying logarithmic scaling. We conduct a Short-Time Fourier Transform (STFT) on these signals using a window size of 2048 and a hop length of 256. The Mel-spectrograms produced are of dimensions 768 (time) and 128 (frequency), created using a bank of 128 Mel filters.

B. Baseline

The baseline model, a convolutional recurrent neural network (CRNN) [8], integrates elements of both convolutional neural networks (CNNs) and recurrent neural networks (RNNs). This CRNN structure features a CNN component consisting of seven blocks, each equipped with filters in a

sequence of 16, 32, 64, 128, 128, 128, and 128. Every block uses a 3 × 3 kernel size and employs average-pooling operations with configurations of [2, 2], [2, 2], [2, 1], [2, 1], [2, 1], [2, 1], [2, 1] across the layers. The RNN segment of the architecture includes two layers of 128 bidirectional gated recurrent units (Bi-GRUs) [5]. An attention pooling layer follows the RNN portion, featuring a linear layer with softmax functions followed by multiplication with another linear layer that utilizes sigmoid activations.

C. DataAugmentation

The mixed output is combined by audio and Gaussian noise that RMS of the noise generated:

$$y[n] = x[n] + \text{RMS}_{\text{noise}} \tag{1}$$

$$\text{RMS}_{\text{noise}} = \sqrt{\frac{P_{\text{signal}}}{10^{\text{SNR}/10}}} \tag{2}$$

Mixup [6] is a technique that performs linear combinations of pairs of data points and their labels. The proportion of each data point in the combination is sampled from a Beta distribution, using a parameter alpha to control the interpolation strength between the two samples:

$$y[n] = \alpha \cdot x_1[n] + (1 - \alpha) \cdot x_2[n] \tag{3}$$

where y is a mixture features of x_1 and x_2 are log-mel-filterbank outputs of class ID. We also multiply the one-hot encoded label with α.

Embedding masking in the context of neural networks is a technique used to prevent over-fitting and improve generalization by randomly setting elements of the embedding vector to zero during training. This can be represented mathematically as follows:

$$\mathbf{E}' = \mathbf{E} \odot \mathbf{M} \tag{4}$$

where $\odot$ denotes the element-wise product. Each element of $\mathbf{M}$ is generated independently based on a certain probability p, which is a hyperparameter determining the likelihood that an element of the embedding vector is masked. Typically, each m_{ij} in $\mathbf{M}$ is sampled from a β distribution:

$$m_{ij} \sim \beta(1 - p) \tag{5}$$

where $\mathbf{E}$ is the original embedding matrix. $\mathbf{M}$ is the mask matrix with the same shape as $\mathbf{E}$. $\mathbf{E}'$ is the masked embedding matrix, used during the forward pass in training. p is the probability of an element being set to zero (masked).

D. BEATS

The BEATs model [2], trained on extensive datasets such as Audioset [9], represents a significant advancement in the field of sound event classification. Central to the BEATs architecture is the innovative use of acoustic tokenizers, a component designed to transform raw audio signals into a series of discrete tokens. This transformation is achieved by extracting meaningful audio features from the raw data, which are then quantized into tokens that capture the fundamental acoustic properties of the input signals. This tokenization process simplifies the audio data, converting it into manageable and semantically rich units.

E. VGGSK

The network architecture primarily utilizes VGGSK [3] [4]and BEATs. In this configuration, data augmentation is conducted during the preprocessing phase of VGGSK. Embeddings are derived from the BEATs model, which masking is applied to enhance the embedding representation. The VGGSK component of the architecture includes a VGG block and four residual blocks that employ selective kernels (SK). For the supervised learning, Binary Cross-Entropy (BCE) serves as the loss function to assess model performance.

F. Interpolation Consistency Training

The interpolation consistency training (ICT) [7] is to perform interpolation calculation on the prediction results of the model. ICT involves mixing any two data samples from the dataset to create a new input for the student model, which then makes predictions based on this mixed data. Simultaneously, the original two data samples are used as inputs for the teacher model, which also makes predictions. These predictions are then blended together. Finally, the predictions made by the student model on the mixed data are regularized for consistency with the blended predictions of the teacher model, and an additional ICT loss value is calculated to measure this consistency.

G. Semi-supervised learning

The Mean Teacher model, employed within a semi-supervised learning framework, utilizes a sophisticated dual-model architecture to enhance model performance and stability using both labeled and unlabeled data. This framework consists of two main components: a student model and a teacher model, which share identical network structures but differ significantly in their parameter updating methods. The student model is conventionally trained through gradient descent, processing both labeled and unlabeled data. For unlabeled data, it generates predictions that are used as pseudo labels for training the teacher model. This phase not only optimizes the student model's parameters but also prepares it to update the teacher model using the Exponential Moving Average (EMA) strategy [10]. This EMA strategy smooths the parameters of the teacher model, reducing the impact of volatile updates from the student model and ensuring a more stable parameter evolution. Moreover, the teacher model is not trained directly. Instead, its parameters are continuously refined based on the EMA of the student model's parameters, enhancing the stability and reducing discrepancies between the models, which are quantified using Mean Square Error (MSE). To further improve recognition accuracy, Inter-Class Training (ICT) is integrated into the loss function, leveraging the structured yet flexible nature of this framework to effectively harness the potential of both labeled and unlabeled datasets in semi-supervised learning scenarios.

The Exponential Moving Average (EMA) is used to stabilize and update model parameters during the training process as follow:

$$\theta_{\text{ema}} = \beta \cdot \theta_{\text{ema}} + (1 - \beta) \cdot \theta \tag{6}$$

where $\theta_e ma$ is the exponential moving average of the model parameters at the current step. β is the decay factor, controlling the extent to which the previous values of ema_θ influence the new value. θ is the current parameter value.

H. Loss Function

The supervised loss $L_{\text{supervised}}$ is formulated by summing the binary cross-entropy losses from two distinct segments of the dataset, each processed by separate parameterized models. Specifically, we define the loss as:

$$L_{\text{supervised}} = \text{BCE}(\theta_s(X_s), Y_s) + \text{BCE}(\theta_w(X_w), Y_w) \tag{7}$$

where θ_s and θ_w denote the model parameters tailored to specific subsets of data, denoted as X_s and X_w respectively. The Y_s and Y_w represent the true binary labels corresponding to these subsets. The Binary Cross-Entropy (BCE) loss is utilized to measure the discrepancy between the predicted probabilities and the actual labels across both model configurations

$$L_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{8}$$

The consistency loss $L_{\text{consistency}}$ in our framework is defined using a combination of weighted mean squared errors (MSE) that compare predictions made by the baseline model parameters θ_s and θ_w, with those made by their perturbed counterparts θ_s and θ_w. This formulation is expressed as:

$$L_{\text{consistency}} = W \cdot \{\text{MSE}(\theta_s(X), \theta'_s(X')) + \text{MSE}(\theta_w(X), \theta'_w(X'))\} \tag{9}$$

where X and X denote the original and perturbed versions of the input data. θ_s and θ_w represent the original model parameters, whereas θ_s and θ_w are their respective perturbed versions. W serves as a scalar that adjusts the impact of the MSE terms on the total loss value.

III. EXPERIMENTS

A. DataSet

The dataset consists of a training set along with validation and evaluation components, as listed in Table I. The occurrences of events within these components are listed in Table II. The training set includes: Weakly Labeled Training Set: This subset comprises 1578 clips with 2244 class occurrences, where each audio clip is labeled with the class of audio events but without specific timestamps. Unlabeled In-Domain Training Set: This subset includes 14412 clips, significantly larger in size compared to the weakly labeled data, and it does not contain labels. Synthetic Strongly Labeled Set: Composed of 10000 clips created using the Scaper soundscape synthesis and augmentation library, this subset is strongly labeled with timestamps indicating the sound events. Real Strongly Labeled Training Set: This subset contains 3470 audio clips sourced from Audioset, each strongly labeled with timestamps of sound events. This set is regarded as an external dataset. Subset of Audioset: Selected from Audioset, this segment includes 32975 clips. Labels were removed, and the clips are used as an external dataset. The validation dataset contains 1168 clips, each strongly labeled with timestamps. The evaluation dataset comprises 699 clips. As shown in Figure 1, it illustrates the key distinction between strong and weak labels in audio event annotation. Strong labels provide precise timestamps marking the start and end points of specific sound events within an audio track, as shown in the waveform and spectrogram. This labeling is detailed, indicating exactly when events like speech or the sound of a blender occur. On the other hand, weak labels only confirm the presence of these sound events within the entire audio track without specifying their temporal boundaries. The visual representation highlights that strong labels are time-bound and specific, whereas weak labels are more general and only indicate occurrence.

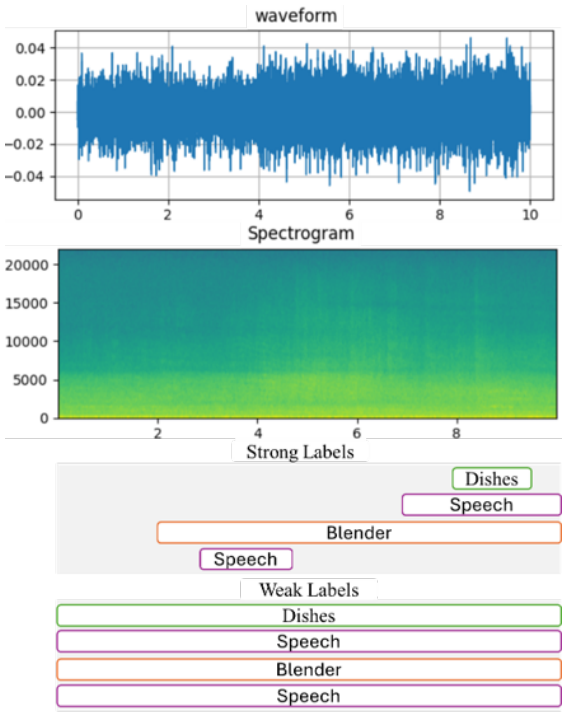

Fig. 1. The differences between strong and weak labels in audio event annotation.

B. Evaluation Metric

The evaluation of our systems was based on the recently introducedthreshold-independent [11] implementation of the polyphonic sound event detection scores (PSDS).

C. Single System

Table 1 shows the performance of each stage of our single model in the submission system. The baseline model initially utilized both the CRNN [8] and BEATs models [2]. However, it

TABLE I
DESED DATASET

Dataset	Label Type	Audio Clips	Sampling Rate(kHz)	Type
Training Set	**Strong Label**	13,470	44.1/16	**Record / Synthetic**
	Weak Label	1,578	44.1	Record
	Unlabeled	10,000	44.1	Record
Validation	**Strong Label**	1,168	44.1	Record
Evaluation		699	44.1	Record

TABLE II
OCCURRENCES OF DESED EVENT CLASSES

Class	Occurrences
Alarm bell ringing	2143
Blender	313
Cat	781
Dishes	2576
Dog	1949
Electric shaver toothbrush	279
Frying	620
Running water	833
Speech	9998
Vacuum Cleaner	178

was modified by replacing the CRNN model with the VGGSK model. This change resulted in an improvement in PSDS1, increasing it from 0.500 to 0.517. After incorporating the strong real dataset into the training process, an improvement was observed in PSDS2, with the value increasing from 0.764 to 0.775. By incorporating the ICT method, significant improvements were achieved in the results. PSDS1 improved to 0.529, indicating a substantial enhancement, while PSDS2 saw a remarkable improvement to 0.780.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODEL CONFIGURATIONS ON PSDS1 AND PSDS2 METRICS.

Model Configuration	PSDS1	PSDS2
CRNN+BEATs (Baseline)	0.500	0.762
VGGSK+BEATs	0.517	0.764
+Strong Real Dataset	0.516	0.775
+ICT	0.529	0.780

D. Ensemble System

As listed in Table ,the performance of System 2, 3 and 4 based on BEATs-VGGSK model in validation set with external data. Systems 3 and 4 were developed using the same model structure but were differentiated by employing distinct data augmentation techniques. System 3 achieved the highest PSDS1 score of 0.552. Similarly, System 4 recorded the highest PSDS2 score of 0.799.

IV. CONCLUSION

Our evaluation of the single and ensemble systems has significant improvement with down stream task of pre-trained model and data augmentation techniques. Initially, our single model system, the BEATs-VGGSK model leads an increase in PSDS1 from 0.500 to 0.517. Further incorporation of the strong real dataset significantly enhanced PSDS2, elevating it from 0.764 to 0.775. The application of the ICT method subsequently marked substantial improvements, with PSDS1 reaching 0.529 and PSDS2 peaking at 0.780. The ensemble systems, specifically Systems 3 and 4, built on the BEATs-VGGSK model and differentiated by data augmentation strategies, achieved robust performance. In conclusion, the updates to model architecture and the integration of advanced data augmentation and training techniques like ICT significantly contributed to the improvements in model performance.

TABLE IV
PERFORMANCE OF VARIOUS SYSTEMS ON PSDS1 AND PSDS2 METRICS WITH OPTIONAL EXTRA DATA

System	Extra data	PSDS1	PSDS2
CRNN (Baseline)		0.359	0.562
CRNN+BEATs (Baseline)	✓	0.500	0.762
System 1		0.424	0.633
System 2	✓	0.529	0.780
System 3	✓	0.552	0.794
System 4	✓	0.542	0.799

REFERENCES

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

[2] S. Chen, Y. Wu, C. Wang, S. Liu, Z. C. D. ompkins, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," in *arXiv:2212.09058*, 2018.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *arXiv:1409.1556.*, 2014.

[4] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 510–519.

[5] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, and F. B. et.al, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *In Proceeding of Empirical Methods in Natural Language Processing*, 2019, pp. 1724–1734.

[6] H. Zhang, M. Cisse, Y. N. Dauphin, and D. L. Paz, "Mixup: Beyond empirical risk minimization," in *arXiv:1710.09412*, 2017.

[7] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, and Y. B. et.al, "Interpolation consistency training for semi-supervised learning," in *arXiv:1903.03825*, 2019, pp. 271–350.

[8] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4 technical report," in *Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge*, 2019.

[9] A. Tarvainen and H. Valpola, "Audio set: An ontology and human-labeled dataset for audio events," in *In Proceeding of IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2017, pp. 1195–1204.

[10] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *In Proceeding of Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1195–1204.

[11] J. Ebbers, R. H. Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *In Proceeding of IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 1021–1025.