CHT+NSYSU SOUND EVENT DETECTION SYSTEM WITH MULTISCALE CHANNEL ATTENTION AND MULTIPLE CONSISTENCY TRAINING FOR DCASE 2021 TASK 4

Technical Report

Yih-Wen Wang[1], *Chia-Ping Chen*[1], *Chung-Li Lu*[2], *Bo-Cheng Chan*[2]

[1]National Sun Yat-Sen University, Taiwan, m083040011@student.nsysu.edu.tw, cpchen@cse.nsysu.edu.tw
[2]Chunghwa Telecom Laboratories, Taiwan, {chungli,cbc}@cht.com.tw

ABSTRACT

In this technical report, we describe our submission system for DCASE 2021 Task4: sound event detection and separation in domestic environments. The proposed system is based on mean-teacher framework of semi-supervised learning and neural networks of CRNN and CNN-Transformer. We employ consistency training of interpolation (ICT), shift (SCT), and clip-level (CCT) to enhance the generalization and representation. A multiscale CNN block is applied to extract various features to mitigate the influence of the event length diversity for the network. An efficient channel attention network (ECA-Net) and exponential softmax pooling enable the model to obtain definite sound event predictions. To further improve the performance, we use data augmentation including mixup, time shift, and time-frequency masks. Our ensemble system achieves the PSDS-scenario1 of 40.72% and PSDS-scenario2 of 80.80% on the validation set, significantly outperforming that of the baseline score of 34.2% and 52.7%, respectively.

Index Terms— sound event detection, CRNN, transformer, semi-supervised learning, consistency training, mean-teacher model, channel attention, pooling function

1. INTRODUCTION

This technical report describes our submission system for DCASE 2021 Task4: Sound Event Detection (SED) and separation in domestic environments. The goal of this task is to build a SED system to detect sound events and time boundaries in Scenario1 (react fast) and Scenario2 (avoid class confusion) by using a large amount of weakly labeled and unlabeled data. In this task, we employ two neural networks and multiple strategies as below:

- CRNN [1] and CNN-Transformer model [2, 3],
- multiscale CNN blocks [4] to extract various features,
- consistency training of interpolation (ICT) [5], shift (SCT) [6], and clip-level (CCT) [7] to enhance model robustness,
- efficient channel attention network (ECA-Net) [8] to pay more attention to important features,
- exponential softmax pooling function [9] to let the weight of frame-level probability be exponential instead of learning.

To further improve the performance, we implement:

- data augmentation methods including mixup [10], time shift, and time-frequency masks [11] to increase data diversity,
- adaptive post-processing to effectively smooth network output,
- score fusion to ensemble the advantages of each single system.

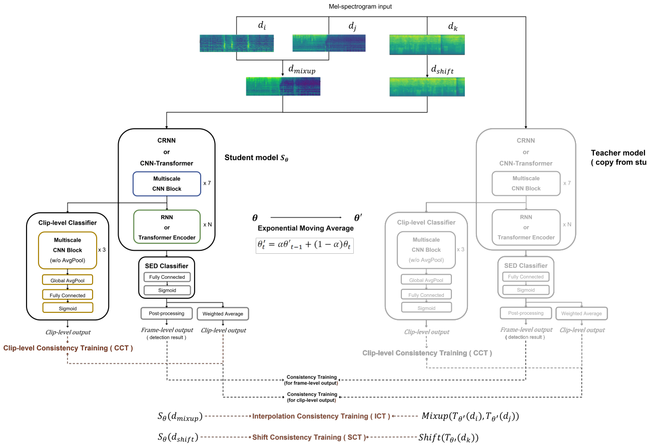

Mel-spectrogram input
d_i
d_j
d_k
d_{mixup}
d_{shift}
CRNN or CNN-Transformer
Multiscale CNN Block
x 7
RNN or Transformer Encoder
x N
Student model S_θ
Teacher model $T_{\theta'}$ (copy from student)
θ
θ'
Exponential Moving Average
$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$
Clip-level Classifier
Multiscale CNN Block (w/o AvgPool)
x 3
Global AvgPool
Fully Connected
Sigmoid
SED Classifier
Fully Connected
Sigmoid
Post-processing
Weighted Average
Clip-level output
Frame-level output (detection result)
Clip-level output
Clip-level Consistency Training (CCT)
Consistency Training (for frame-level output)
Consistency Training (for clip-level output)
$S_\theta(d_{mixup})$
Interpolation Consistency Training (ICT)
$Mixup(T_{\theta'}(d_i), T_{\theta'}(d_j))$
$S_\theta(d_{shift})$
Shift Consistency Training (SCT)
$Shift(T_{\theta'}(d_k))$

Figure 1: The proposed sound event detection system structure.

2. PROPOSED METHODS

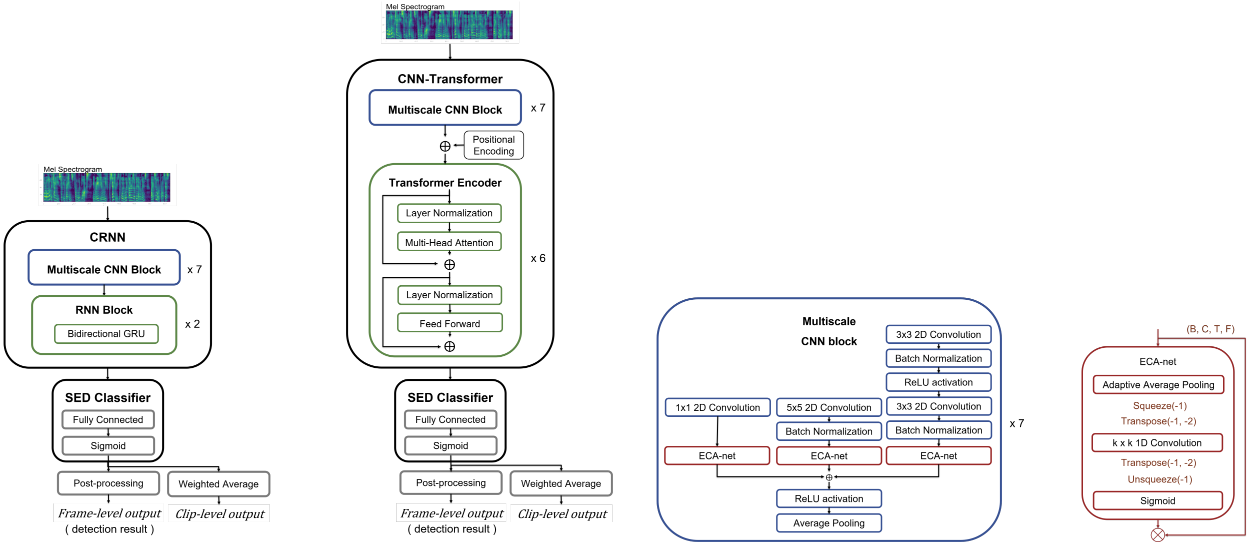
2.1. Network architecture

2.1.1. CRNN

The convolutional recurrent neural network (CRNN) is similar to DCASE 2021 Task4 baseline architecture, which consists of 7 layers of CNN blocks and 2 layers of bidirectional gated recurrent unit (GRU), as shown in 2(a). A CNN block contains the convolutional layer, batch normalization (BN), Rectified Linear Unit (ReLU) activation, and average-pooling (AvgPool) layer. The input mel-spectrogram passes learnable convolution kernels and output the feature maps. BN and ReLU activation are intended to speed up and stabilize training. AvgPool calculates the average for each patch of the feature map and downsamples feature dimensions along both the time axis and the frequency axis. Then, RNN layers capture the long-term contextual information. Finally, the SED classifier consists of a fully connected layer and sigmoid function to discriminate the sound event types.

2.1.2. CNN-Transformer

Transformer [12] allows parallel computation and achieves state-of-the-art performance on many tasks [13, 14, 15, 16, 17]. Hence, we implement CNN-Transformer network [2, 3] for SED, as shown in Figure 2(b). Positional encoding is used to enhance the output features from the CNN blocks with order information before the trans-



(a) CRNN (b) CNN-Transformer (c) Multiscale CNN block (d) ECA-Net

Figure 2: The network structure of CRNN, CNN-Transformer, multiscale CNN block, and efficient channel attention network (ECA-Net).

former blocks. A transformer encoder block has layer normalization, multi-head attention, and feed-forward layer. The multi-head attention estimates the similarity between query and key, and extracts value as a weighted sum. The mechanism allows the model to jointly pay attention to the information from different positions. The fully-connected feed-forward layer with ReLU activation is applied to each position identically. For regularization, we adopt pre-layer normalization (Pre-LN) [18] and residual connection.

2.1.3. Multiscale CNN

From strongly labeled training data, we estimate duration of each sound event as below. 0∼2s: `alarm/bell/ringing`, `cat`, `dishes`, `dog`, and `speech`. 4∼6s: `blender` and `running water`. 7∼10s: `electric shaver/toothbrush`, `frying`, and `vacuum cleaner`. The length of sound events is various and cause the model to work with inconsistent accuracy for the event of different scales. Thus, we refer to [4] to apply different kernel sizes to build a multiscale CNN block to capture the richer features, as Figure 2(c). A multiscale CNN block contains the kernel size of 1x1, 3x3, 5x5 and uses addition to integrate features of different scales.

2.1.4. Efficient Channel Attention

The effect of the acoustic feature extraction largely determines the model ability to predict different sound events and affects the final classification result. However, the attention mechanism can make the model pay more attention to areas which may be important features, and improve the model ability to distinguish features of sound events. We combine the efficient channel attention network (ECA-Net) [8] in multiscale CNN blocks before adding features of different scales, as shown in Figure 2(c). ECA-Net is composed of adaptive average pooling (A-AvgPool) layer, 1D convolutional (1D-CNN) layer, and sigmoid function, as shown in Figure 2(d). A-Avgpool is applied along the channel axis and 1D-CNN calculate the attention of each channel. The kernel size of 1D-CNN is defined by

$$k = \left| \frac{\log_2(C) + b}{\gamma} \right|_{odd} \tag{1}$$

where k and C denote kernel size and channel dimensional, γ and b are set to 2. Clearly, high-dimensional channels have longer range interaction, vice versa.

2.1.5. Pooling Function

[9] compared five different types of pooling functions in the multiple instance learning (MIL) framework for SED, namely attention pooling, max pooling, average pooling, linear softmax, and exponential softmax. The attention pooling estimates the weights for each frame are learned with a dense layer in the network. The max pooling simply take the large probability in all frames. The average pooling assigns an equal weight for all frames. The linear softmax assigns weights equal to the frame-level probability, while the exponential softmax assigns a weight of exponential to the frame-level probability. Baseline uses attention pooling to transform frame-level into clip-level. However, with different application scenarios, there should be a relatively appropriate pooling function to replace.

2.2. Semi-Supervised Learning

In this work, we employ the mean-teacher framework [19] for semi-supervised learning, and use the Mean Square Error (MSE) loss for the consistency cost. The MSE loss function is defined by

$$\mathrm{MSE}(y, \hat{y}) = (y - \hat{y})^2 \tag{2}$$

where y and $\hat{y}$ denote the target and the prediction, respectively. Following, we propose multiple consistency criteria to regularize/direct how the SED system should learn from unlabeled or weakly-labeled data.

2.2.1. *Interpolation Consistency Training*

Recently, the interpolation consistency training (ICT) [5] has been proposed for semi-supervised learning. ICT encourages the prediction at an interpolation of unlabeled data points to be consistent with the interpolation of the prediction at these data points. Learning from interpolation samples can help the model discriminate ambiguous samples to improve the generalization ability. We define the ICT loss function by

$$\begin{aligned} L_{ICT} = \text{MSE}(&S_{\theta}(\lambda d_i + (1-\lambda)d_j), \\ &\lambda T_{\theta'}(d_i) + (1-\lambda)T_{\theta'}(d_j)) \end{aligned} \tag{3}$$

where S_{θ} and $T_{\theta'}$ denote a student model and a teacher model, d_i and d_j denote data points, and λ is randomly sampled from a Beta distribution.

2.2.2. *Shift Consistency Training*

Inspired by ICT, we consider time-shift as another way to enhance consistency which is similar to proposed by [6], called shift consistency training (SCT). We define the SCT loss function by

$$L_{SCT} = \text{MSE}(S_{\theta}(\text{shift}(d_k)), \text{shift}(T_{\theta'}(d_k))) \tag{4}$$

SCT encourages the prediction of time-shift input to be consistent with time-shift prediction. In theory, it allows the model to learn shift-invariance and temporal localization of sound events.

2.2.3. *Clip-level Consistency Training*

In addition to ICT and SCT, we also implement clip-level consistency training (CCT) [7]. We define the CCT loss function by

$$L_{CCT} = \text{MSE}(\text{NN}(d_x), \text{ClipLevel}(f_x)) \tag{5}$$

where $\text{NN}(d_x)$ is the weighted average pooling of the CRNN or CNN-Transformer frame-level network output of data d_x, and $\text{ClipLevel}(f_x)$ is obtained by feeding the feature map f_x of the final CNN block to a clip-level classifier. As shown in Figure, the clip-level classifier consists of 3 extra multiscale CNN blocks, a global average pooling, and a fully connected layer.

2.2.4. *Overall Consistency Training*

In summary, the overall loss is

$$L = L_0 + L_{ICT} + L_{SCT} + L_{CCT} \tag{6}$$

where L_0 denotes the loss without the proposed consistency.

2.3. Data Augmentation

- Mixup [10]. It mixes two randomly selected samples from the original training data and uses λ sampled from Beta distribution to control the strength of interpolation between two samples. The linear interpolation technique can enhance the data diversity and robustness of the network.
- Shift [11]. It shifts a feature sequence on the time axis, and overrun frames are concatenated with the opposite side of the sequence. The usage helps the network learn temporal localization information of the sound event.
- Masks [11]. It creates artificial data by masking a block of consecutive time steps or frequency channels on the mel-spectrogram instead of the raw audio. It can help the network learn the beneficial features to be robust to partial loss of spectral information or speech segments.

2.4. Adaptive Post-Processing

The frame-level network output is further post-processed to become the final output. First, thresholding operation converts probabilistic outputs to binary outputs. Then, the binary output sequences are further smoothed by median filters to avoid spurious detection. As sound classes may have varying temporal characteristics, we untie median filter sizes in the post-processing of the different sound classes. Following [2], we search the median filter size from 1 to 51 in increments of 1 with data from DCASE 2021 Task 4.

2.5. Score Fusion

To improve generalization performance, we perform score fusion as a model ensemble technique. We utilize different data augmentation methods to build several single systems based on CRNN and CNN-Transformer models with different strategies. Then, we average the raw posterior outputs of the multiple models and perform adaptive post-processing to smooth the network output.

3. EXPERIMENTS

3.1. Dataset and Signal Preprocessing

The DESED dataset of DCASE 20201 Task 4 is comprised of 10-sec audio clips and 10 classes of sound events. The data are in two domains: real data (44.1kHz) extracted from AudioSet [20] and synthetic data (16kHz) generated by Scaper [21]. Each audio clip can be strongly labeled with the sound events and their time boundaries annotated, weakly labeled with only the sound events annotated, or unlabeled without any annotation. All dataset is divided into 4 subsets: weakly labeled (1,578 clips), unlabeled (14,412 clips), strongly labeled (10,000 clips), and validation set (1,168 clips). Audio signals are resampled to 16kHz sampling rate at first by librosa tool [22]. From the resampled signals, 128-channel mel-spectrogram is extracted with window size of 2048 and hop size of 256. The mel-spectrogram of a clip is normalized to zero mean and unit variance. Consequently, the size of the input acoustic features to the deep neural network is 626×128.

3.2. Network Setting

The 7 layers of multiscale CNN blocks have the number of filters:[16, 32, 64, 128, 128, 128, 128] and pooling size:[[2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2]]. The 6 layers of transformer encoder blocks have multi-head attention with 256 units and 8 heads and a feed-forward layer with 2048 units. For ICT and mixup augmentation, the parameter λ is sampled from $\text{Beta}(\alpha, \alpha)$ and α from 0.1 to 0.7 in increments of 0.1. For SCT and shift augmentation, we choose the amount of time-shift by sampling from a normal distribution with a zero mean and a standard deviation of 90. For masks augmentation, the size of time-mask and frequency-mask are sampled from a uniform distribution from 0 to 30 and 40, respectively.

4. EVALUATION RESULTS

The evaluation of DCASE 2021 Task4 contains PSDS-scenario 1 for time boundaries accuracy and PSDS-scenario 2 for sound class accuracy. From Table 1, whether neural network is CRNN or CNN-Transformer, the incorporation of ICT, SCT, and CCT has significantly achievement on two scenarios. The multiple consistency training strategies on CRNN improved PSDS 1 from 34.04% to 37.86%, PSDS 2 from 53.30% to 60.87%, and on CNN-Transformer, PSDS 1 from 33.46% to 37.33%, PSDS 2 ranges from 48.77% to 55.87%. From Table 2 and Table 3, we found that multiscale CNN blocks and ECA-Net can help the model obtain specific features of sound events so that CRNN can reach 65.54% and CNN-Transformer can reach 61.10% for PSDS 2. From Table 4, both types of neural networks are best when using attention pooling at PSDS 1 and using exponential softmax at PSDS 2, especially CRNN has a significant improvement. We consider that attention pooling learns weights from the network so that they have a time series relationship. Therefore, it has better performance under stricter evaluation standards with time requirements. Exponential softmax uses exponentials as weights to conform to monotonicity. The higher the prediction probability of the time point, the higher the weight, so the performance is better under the stricter evaluation criteria for the correctness of the category.

We combine two models with proposed strategies to build three single systems so that PSDS 1 and PSDS 2 can have the best performance: *(A) CNN-Transformer + ICT, SCT, CCT, Multiscale*, *(B) CRNN + ICT, SCT, CCT, Multiscale*, *(C) CRNN + ICT, SCT, CCT, Multiscale, ECA-Net, Exponential Softmax*. We apply different data augmentation methods to build several systems for fusion based on the three single systems above. From Table 5, our ensemble systems can achieve 40.72% of PSDS 1 and 80.80% of PSDS 2.

Table 1: Results of different consistency training on CRNN and CNN-Transformer.

Consistency Training	Model	PSDS 1	PSDS 2
-	CRNN	34.04%	53.30%
	CNN-Transformer	33.46%	48.77%
ICT	CRNN	36.38%	55.87%
	CNN-Transformer	33.39%	50.07%
ICT, SCT	CRNN	**37.86%**	59.47%
	CNN-Transformer	35.61%	52.01%
ICT, SCT, CCT	CRNN	37.64%	**60.87%**
	CNN-Transformer	**37.33%**	**55.87%**

Table 2: Results of different CNN blocks on CRNN and CNN-Transformer with ICT, SCT, and CCT.

CNN blocks	Model	PSDS 1	PSDS 2
3x3 CNN	CRNN	**37.64%**	60.87%
	CNN-Transformer	**37.33%**	55.87%
Multiscale CNN	CRNN	36.70%	**63.50%**
	CNN-Transformer	34.75%	**61.10%**

Table 3: Results of ECA-Net on CRNN and CNN-Transformer with ICT, SCT, CCT, and Multiscale CNN.

Efficient Channel Attention	Model	PSDS 1	PSDS 2
-	CRNN	**36.70%**	63.50%
	CNN-Transformer	34.75%	**61.10%**
ECA-Net	CRNN	34.71%	**65.54%**
	CNN-Transformer	**35.13%**	60.27%

Table 4: Results of different pooling function on CRNN and CNN-Transformer with ICT, SCT, CCT, and Multiscale CNN.

Pooling Function	Model	PSDS 1	PSDS 2
Attention	CRNN	**36.70%**	63.50%
	CNN-Transformer	**34.75%**	61.10%
Max pooling	CRNN	36.10%	64.59%
	CNN-Transformer	31.73%	59.77%
Average pooling	CRNN	5.34%	73.95%
	CNN-Transformer	4.53%	60.41%
Linear Softmax	CRNN	26.75%	60.17%
	CNN-Transformer	4.21%	60.57%
Exponential Softmax	CRNN	5.82%	**75.35%**
	CNN-Transformer	4.13%	**61.31%**

Table 5: Fusion results of different data augmentation on CRNN and CNN-Transformer with different strategies above. α means the parameter of beta distribution.

#	Model	Strategies	Data Augmentation	PSDS 1	PSDS 2
0	CRNN	-	Mixup ($\alpha = 0.2$)	34.04%	53.30%
1	CNN-Transformer	ICT, SCT, CCT, Multiscale	Mixup ($\alpha = 0.2$)	**34.75%**	61.10%
2			Shift	31.39%	55.05%
3			Masks	33.24%	59.04%
4			Mixup ($\alpha = 0.2$)+Shift	33.43%	58.68%
5			Mixup ($\alpha = 0.2$)+Masks	34.29%	**61.52%**
6			Shift+Masks	33.64%	55.46%
7	CRNN	ICT, SCT, CCT, Multiscale	Mixup ($\alpha = 0.1$)	37.69%	63.00%
8			Mixup ($\alpha = 0.2$)	37.51%	62.63%
9			Mixup ($\alpha = 0.4$)	36.71%	64.82%
10			Mixup ($\alpha = 0.5$)	36.84%	64.18%
11			Mixup ($\alpha = 0.6$)	36.55%	61.85%
12			Mixup ($\alpha = 0.7$)	36.70%	63.91%
13			Shift	35.71%	61.29%
14			Masks	36.96%	64.84%
15			Mixup ($\alpha = 0.2$)+Shift	37.03%	63.02%
16			Mixup ($\alpha = 0.2$)+Masks	**38.13%**	**65.32%**
17	CRNN	ICT, SCT, CCT, Multiscale, ECA-Net, Exp.Softmax	Mixup ($\alpha = 0.1$)	**6.81%**	75.59%
18			Mixup ($\alpha = 0.2$)	5.71%	76.16%
19			Mixup ($\alpha = 0.7$)	5.37%	**76.29%**
20			Shift	4.46%	72.16%
21			Masks	5.29%	75.07%
22			Mixup ($\alpha = 0.2$)+Shift	5.12%	76.19%
23			Mixup ($\alpha = 0.2$)+Masks	4.82%	75.45%
24			Shift+Masks	4.83%	76.08%
7∼16	-	-	-	**40.72%**	70.25%
17∼24	-	-	-	6.08%	**80.80%**
1∼16	-	-	-	38.79%	67.18%
1∼24	-	-	-	37.02%	72.42%

5. CONCLUSION

In this technical report, the proposed system is based on the neural network of CRNN and CNN-Transformer, which is trained with the mean-teacher framework of semi-supervised learning using multiple consistency criteria. Among them, interpolation consistency training (ICT) helps the model discriminate the ambiguous samples to enhance the generalization ability, shift consistency training (SCT) assists the model to learn better temporal information, clip-level consistency training (CCT) promotes the model feature representation power. In additional, a multiscale CNN block is applied to extract richer features to alleviate the influenct of the diversity of event length for the model. An efficient channel attention network (ECA-Net) and exponential softmax pooling assist model to obtain more definite sound event predictions. We employ the mixup, shift, and masks of data augmentation to further improve the model performance. Finally, our ensemble sound event detection system achieves the PSDS-scenario 1 of 40.72% and PSDS-scenario 2 of 80.80% on the validation set, considerably outperforming that of the baseline score of 34.2% and 52.7%, respectively.

6. REFERENCES

[1] N. Turpault, R. Serizel, J. Salamon, and A. P. Shah, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," 2019.

[2] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution augmented transformer for semi-supervised sound event detection," in *Proc. Workshop Detection Classification Acoust. Scenes Events (DCASE)*, 2020, pp. 100–104.

[3] ——, "Weakly-supervised sound event detection with self-attention," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 66–70.

[4] M. Tang, L. Guo, Y. Zhang, W. Yan, and Q. Zhao, "Multi-scale residual crnn with data augmentation for dcase 2020 task 4."

[5] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *arXiv preprint arXiv:1903.03825*, 2019.

[6] C.-Y. Koh, Y.-S. Chen, Y.-W. Liu, and M. R. Bai, "Sound event detection by consistency training and pseudo-labeling with feature-pyramid convolutional recurrent neural networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 376–380.

[7] L. Yang, J. Hao, Z. Hou, and W. Peng, "Two-stage domain adaptation for sound event detection."

[8] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks, 2020 ieee," in *CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE*, 2020.

[9] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[11] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[13] L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[14] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[15] S. V. Katta, S. Umesh, *et al.*, "S-vectors: Speaker embeddings based on transformer's encoder for text-independent speaker verification," *arXiv preprint arXiv:2008.04659*, 2020.

[16] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[17] J. Wang and S. Li, "Self-attention mechanism based system for dcase2018 challenge task1 and task4," *Proc. DCASE Challenge*, pp. 1–5, 2018.

[18] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 524–10 533.

[19] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *arXiv preprint arXiv:1703.01780*, 2017.

[20] 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.

[21] R. Serizel, N. Turpault, A. Shah, and J. Salamon, "Sound event detection in synthetic domestic environments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 86–90.

[22] librosa, "librosa," https://github.com/librosa/librosa, 2020.