Semi-supervised Sound Event Detection with Consistency Training by Unsupervised Data Augmentation

Arkaprava Biswas(20204407)
Priyanshu Kumar(190651)

***Abstract*—In this work, we've implemented consistency training with Unsupervised Data Augmentation along with supervised training for sound event detection in domestic environment. We've used DCase 2020 Task-4 dataset for this work.**

Keywords: semi-supervised learning, unsupervised data augmentation, consistency training, sound event detection

I. Introduction

Sound event detection recently emerging as a very pivotal problem day by day and recently DCase competitions resulted in increasing interests. After Valpola's remarkable work[1], mean-teacher models became benchmark for SED. The guided learning framework[3] has also got benchmark performances for SED. But the problem of these two frameworks is that it takes very large computations to train. In 2020, Google Research published their work[5], which utilised unlabelled data in a small model using consistency training with unsupervised data augmentation. In the resultant year's DCase competition, Dinkel et. al.[3] used that Google's work and got comparable performance with state-of-the-art mean-teacher and guided-learning frameworks.

II. Dataset

Data has been adopted from DCASE 2022 Task-4 Dataset where D_{syn} contains 2045 synthetic audios of which we've frame level informations. D_{weak} contains 1336 clip level labelled files and D_{unlab} contains 6752 unlabelled files. Both of these are real dataset. i.e we have: $|D_{syn}| = 2045, \{x_n(t), y_n(t)\}$; $|D_{weak}| = 1336, \{x_n, y_n\}$; $|D_{unlab}| = 6752, \{x_n\}$;

Each audio is 10 seconds long. Log Mel Spectrograms features have been taken. Nfft window length = 40 ms and hopLength = 20 ms

III. Model

The Convolutional Recurrent Neural Network model has been used which takes in Log Mel Spectrogram (inputSize, 64, 501) as input and outputs Frame Level(inputSize, 501,10) and Clip Level(inputSize, 10) prediction probabilities for 10 classes as shown in the figure 3 and figure 4.

Each convolutional block has three layers stacked as shown in the figure 2.

High-Level Feature Representation of the Log Mel Spectrogram is produced after passing through the convolutional blocks and average pooling blocks which have been applied only to the frequency features.

These High-Level Features when passed through 2 LSTM layers and dense_ sed layers give the class probability of each frame hence have output size (inputSize, 501,10). Further, the frame-level predictions have been aggregated to give clip-level predictions(inputSize, 10) Thus our model finally outputs {Frame Level, Clip Level} class probabilities as shown in figure 4.

Model: "model_audio_1"

Layer (type)	Output Shape	Param #
cnn_block_1 (CNN_block)	multiple	448
cnn_block_2 (CNN_block)	multiple	37504
cnn_block_3 (CNN_block)	multiple	148096
cnn_block_4 (CNN_block)	multiple	148096
cnn_block_5 (CNN_block)	multiple	148096
average_pooling2d_1 (AveragePooling2D)	multiple	0
average_pooling2d_2 (AveragePooling2D)	multiple	0
average_pooling2d_3 (AveragePooling2D)	multiple	0
dropout_1 (Dropout)	multiple	0
bidirectional_2 (Bidirectional)	multiple	2229248
bidirectional_3 (Bidirectional)	multiple	394240
time_distributed_1 (TimeDistributed)	multiple	65792
time_distributed_2 (TimeDistributed)	multiple	2570
aggregrate_1 (aggregrate)	multiple	0 (unused)
aggregrate_2 (aggregrate)	multiple	0

Total params: 3,174,090
Trainable params: 3,173,002
Non-trainable params: 1,088

Fig. 1: Model Architecture with parameters

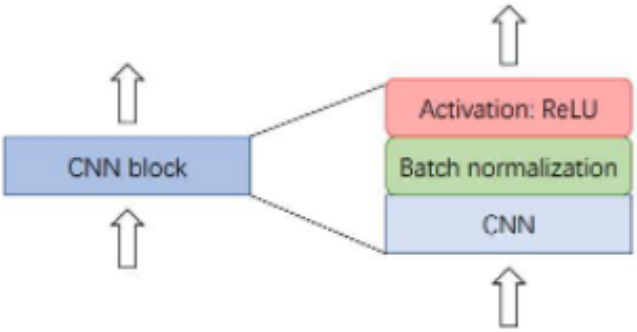

Fig. 2: CNN Block

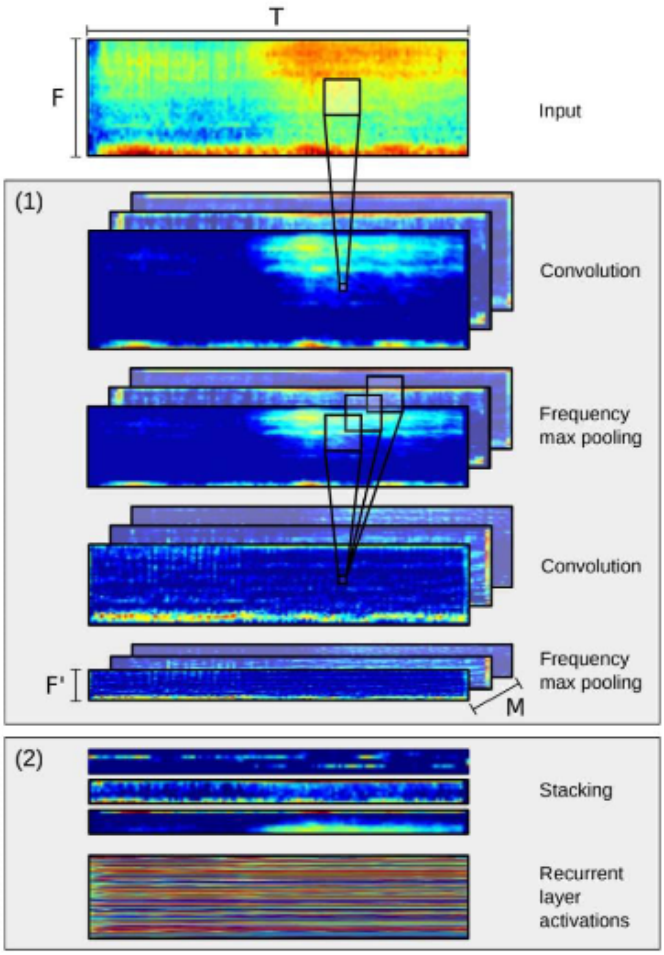

Fig. 3: Detailed Architecture of the Convolutional Recurrent Neural Network showing the various layers of the model.

A. Aggregate Layer

To produce Clip Level class probabilities from the frame-level class probabilities, we tried two approaches: CDur and EATP. But EATP was not giving satisfactory results so we moved on with the CDur approach. The mathematical details have been shown below:

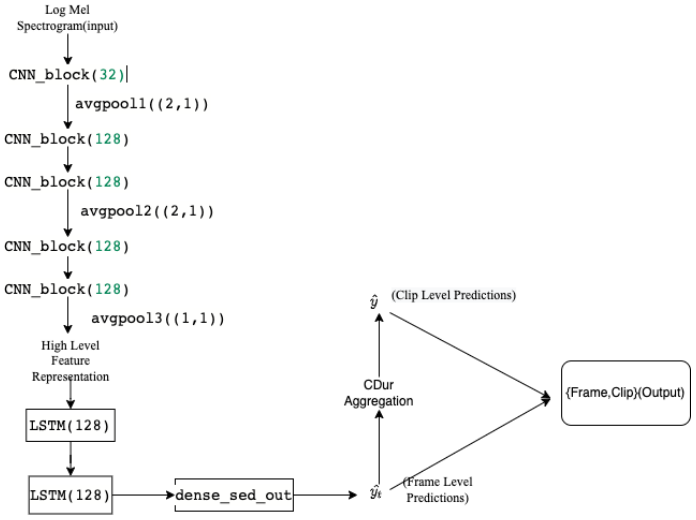

Fig. 4: Model Flowchart

1) LinSoft: This is the aggregate layer we used. It takes in frame level class probabilities and produces clip level class probabilities.

$$\hat{y} = \frac{\sum_{t=1}^{T} \hat{y_t}^2}{\sum_{t=1}^{T} \hat{y_t}} \tag{1}$$

B. Clip Smoothing

$$\hat{y_t} = \hat{y_t} \odot \hat{y} \tag{2}$$

IV. Model Training

We've frame level information for synthetic data, and from that we can also get clip level information. And for real data, we've a few of them clip level annotated. So, we've supervised learning for synthetic data for both frame and clip level and weak data for clip level.
$M(x) \Rightarrow \hat{y}, \hat{y_t}$

$$L_{sup} = L(y_t, \hat{y_t}) + L(y, \hat{y}), x \in D_{syn} or x \in D_{weak} \tag{3}$$

And we've around 7000 unlabelled data. Which we've used by consistency training with unsupervised data augmentation. We've also used frame level outputs from weak data for consistency training.
So, we have : $x \xrightarrow{aug} x^+$ and $M(x^+) \Rightarrow \hat{y^+}, \hat{y_t^+}$

$$L_{UDA} = L(\hat{y_t^+}, \hat{y_t}) + L(\hat{y^+}, \hat{y}), x \in D_{unlab} or x \in D_{weak} \tag{4}$$

$$L_{total} = L_{sup} + \lambda L_{UDA} \tag{5}$$

We've used BCE loss for most of the loss functions, and have taken λ as 1 in most of our experiments.

V. Experiments and Results

We've conducted experiments with batch size 64 with augmentation in raw audio as well as the mel-spectogram. We're reporting our results in terms of micro and macro event-f1 score and have reported the best result we've got. We've conducted experiments by augmenting raw audios with Gain, Polarity Inversion, and Time Masking, frequency masking. We've got best performance yet for unlabelled data using frequency mask augmentation with probability = 0.2.

TABLE I: Results

Dataset used	**Micro E-F1**	**Macro E-F1**
Synth	44.24	49.10
+ Weak	41.37	43.08
++ Unlab	43.04	46.00

VI. Conclusion and Future Work

This work shows there's a lot to discover, if we'd be able to fully utilise unlabelled data. But, we need to cautiously use unlabelled data. In future direction, we'd look for more ways to fully utilise unlabelled data for sound event detection.

References

[1] Harri Valpola, and Antti Tarvainen *LaTeX: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results*, Advances in Neural Information Processing Systems 30 (NIPS 2017)

[2] Emre, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen *LaTeX: Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection*, IEEE Transactions on Audio, Speech and Language Processing, Feb. 2017

[3] Liwei Lin, Xiangdong Wang, Hong Liu, and Yueliang Qian *LaTeX: GUIDED LEARNING FOR WEAKLY-LABELED SEMI-SUPERVISED SOUND EVENT DETECTION*, ICASSP, 2020

[4] Hao Yen, Pin-Jui Ku, Ming-Chi Yen, Hung-Shin Lee, and Hsin-Min Wang *LaTeX: JOINT TRAINING OF GUIDED LEARNING AND MEAN TEACHER MODELS FOR SOUND EVENT DETECTION*, DCase Proceedings 2020, Japan

[5] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, Quoc V. Le *Unsupervised Data Augmentation for Consistency Training*,34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada

[6] Heinrich Dinkel, Xinyu Cai, Zhiyong Yan, Yongqing Wang, and Junbo Zhang *LaTeX: THE SMALLRICE SUBMISSION TO THE DCASE2021 TASK 4 CHALLENGE: A LIGHTWEIGHT APPROACH FOR SEMI-SUPERVISED SOUND EVENT DETECTION WITH UNSUPERVISED DATA AUGMENTATION*, DCase Technical Report, 2021

[7] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz *MIXUP: Beyond Empirical Risk Minimization*, conference paper at ICLR 2018

[8] HDaniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le *SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition*, Google Brain, 2019