SOUND EVENT LOCALIZATION AND DETECTION USING CRNN ARCHITECTURE WITH *MIXUP* FOR MODEL GENERALIZATION

Pranay Pratik [1*], *Wen Jie Jee* [2*†], *Srikanth Nagisetty*[1], *Rohith Mars*[1], *Chong Soon Lim*[1],

[1] Panasonic R&D Center Singapore
{pranay.pratik, srikanth.nagisetty, rohith.mars, chongsoon.lim}@sg.panasonic.com
[2] Nanyang Technological University Singapore,
wjee001@e.ntu.edu.sg

ABSTRACT

In this paper, we present the details of our solution for the IEEE DCASE 2019 Task 3: Sound Event Localization and Detection (SELD) challenge. Given multi-channel audio as input, goal is to predict all instances of the sound labels and their directions-of-arrival (DOAs) in the form of azimuth and elevation angles. Our solution is based on Convolutional-Recurrent Neural Network (CRNN) architecture. In the CNN module of the proposed architecture, we introduced rectangular kernels in the pooling layers to minimize the information loss in temporal dimension within the CNN module, leading to boosting up the RNN module performance. Data augmentation *mixup* is applied in an attempt to train the network for greater generalization. The performance of the proposed architecture was evaluated with individual metrics, for sound event detection (SED) and localization task. Our team's solution was ranked 5^{th} in the DCASE-2019 Task-3 challenge with an F-score of 93.7% & Error Rate 0.12 for SED task and DOA error of $4.2°$ & frame recall 91.8% for localization task, both on the evaluation set. This results showed a significant performance improvement for both SED and localization estimation over the baseline system.

***Index Terms*—** DCASE-2019, SELD, SED, Localization, CRNN, *mixup*

1. INTRODUCTION

Sound event localization and detection (SELD) is a challenging and well-researched topic in the field of acoustic signal processing. There are two sub-tasks for SELD, first: the sound event detection (SED), second: the sound source's direction estimation. An ideal SELD system would be able to detect & classify multiple sound events and for each detected sound event determines its direction of arrival. Signal processing algorithms have been traditionally employed to address this challenging task. However performance achieved by such methods are still limited under practical conditions.

In recent research, deep learning based techniques have been applied individually for both SED and localization part of the SELD task. In [1, 2], it has been shown that CNN based network can detect and classify sound events with high accuracy. In [3], 1D-CNN has been applied for solving the sound localization task. The recent trend in this field has been about developing deep learning techniques for joint localization and classification of multiple sound sources. In [4] authors proposed 2D-CNN based network for joint sound source localization and classification. In [5] authors introduced convolutional recurrent neural network architecture (CRNN), where the CNN module learns the audio spectral information followed by the RNN module, that learn the temporal information. This network architecture has been set as the baseline model in the DCASE2019 Task 3 challenge - Sound Event Localization & Detection. In [6] authors introduced two-stage training approach, which shows improvement in the overall performance over [5]. In this approach the training of the network is split into two branches, i.e., the SED branch and the localization branch.

In this paper, we proposed two deep CRNN architectures with log-mel spectrogram and generalized cross-correlation phase transforms (GCC-PHATs) as input features to the network. In the CNN module of one of the proposed network architecture, we restricted pooling in the frequency domain, this helps in preserving temporal information, boosting the performance of RNN module. Data augmentation technique *mixup* was used in an attempt to generalize the network. We investigated the effect of *mixup* on each of the sub-task, SED and localization and compared our results with baseline system provided by the DCASE-2019 challenge and with other prior-arts.

The rest of the paper is organized as follows. In Section 2, we presented the details on feature extraction, data augmentation technique and our proposed CRNN architectures. In Section 3, we discuss experiments setup & compare our results with prior-arts. Finally, conclusion and future work is presented in Section 4.

2. METHODOLOGY

In this section, we present our methodology starting with input feature extraction description followed by CRNN architecture description. In addition, we also discuss the data augmentation step used during training for improving model generalization. For training the network we adopted the strategy proposed in [6], where the model is first trained on SED task, then on localization task using the same network architecture.

2.1. Features

Input features plays a crucial role in training deep neural network. In this work, the raw data is in the form of four-channel audio signal, recorded at 48kHz sampling rate using a microphone array and was provided by DCASE Task-3 organizers [7]. The time domain multi-channel signals were first down-sampled to 32 kHz and then used

*Both authors contributed equally.
†This work was done during an internship at Panasonic R&D Center Singapore.

https://doi.org/10.33682/gbfk-re38

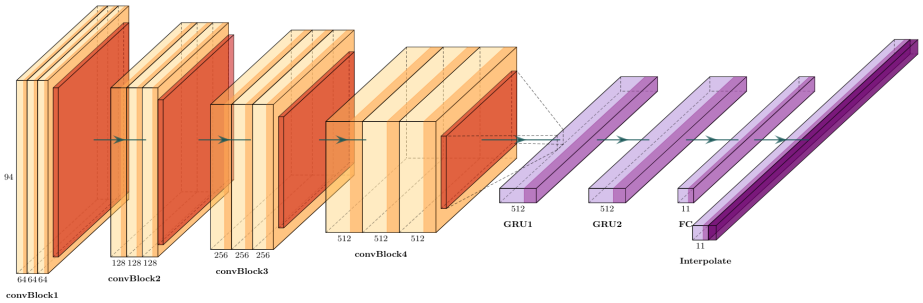

Figure 1: Base architecture **TS-C2Rnn**. Each convBlock contains three Conv2D layers followed by (2×2) average pooling. Each CNN layer is followed by batch normalization and ReLU activation. convBlock1 receives the input features.

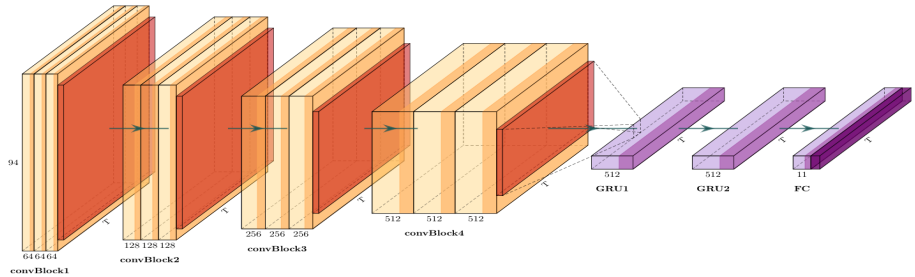

Figure 2: Proposed architecture **TS-C2Rnn-P**, with (2×1) average pooling after each convBlock, with no interpolation layer.

to extract log-mel spectrogram and GCC-PHAT features.

The log-mel spectrogram is commonly used as input feature in speech recognition [8] because of its similarity to frequency decomposition of the human auditory system. To obtain the log-mel spectrogram, time domain audio signal is converted to the time-frequency (TF) domain using short-time Fourier transform (STFT). This ensures that both the temporal and spectral characteristics of the audio data are utilized. After the frequency domain conversion, we extracted the log-mel spectrogram corresponding to each audio clip using 96 Mel bands. For the STFT parameters, we employ a frame length of 1024 points, with a hop length of 10 ms.

GCC-PHAT is widely used for estimation of time difference of arrival (TODA) of acoustic signal between two microphones. A basic frequency domain cross-correlation is estimated by the inverse-FFT of cross power spectrum. GCC is the improved version of cross-correlation, it is estimated by adding a windowing (or filtering) function prior to the inverse transform to improve the estimation of the time delay, depending on the specific characteristics of the signals and noise. GCC-PHAT is the phase-transformed version of GCC, which eliminate the influence of amplitude in the cross power spectrum, hence only preserving the phase information [9].

2.2. CRNN Architecture

The base network architecture introduced in this work is inspired from [6] and named as TS-C2Rnn as shown in figure 1 . The extracted audio features are provided as input to a CRNN architecture. CNN module of TS-C2Rnn consist of 4 convolutional blocks, named convBlock1 to convBlock4. Each convBlock is followed by an average pooling layer. Within each convBlock there are 3 convolutional layers, followed by batch normalization and ReLU activation. For convolutional layers in the convBlocks, 3×3 kernel is used, with stride and padding fixed to 1. The number of filters used in convBlock1 to convBlock4 are $\{convBlock1 : 64, convBlock2 : 128, convBlock3 : 256, convBlock4 : 512\}$. For performing average pooling in convBlocks, we used 2×2 window, with a stride of 2×2 . The CNN module of the network is followed by RNN module, which has two GRU layers, GRU-1 and GRU-2. The output of the GRU-2 layer is fed into fully connected (FC) layer of size N, where N is the number of sound event classes. FC layer is followed by interpolate layer to ensure the final number of the time frames is approximately equal to the original number of time frames of the input clip. This is necessary due to the presence of square kernels in the pooling layers in each convBlock. The output of the interpolate layer contains N class scores, azimuth and elevation values corresponding to each T time frames, where T varies from clip to clip.

We proposed another network architecture TS-C2Rnn-P which is a modified version of TS-C2Rnn architecture as shown in Figure 2. In the CNN module of TS-C2Rnn the 2×2 pooling across time and frequency domain reduces the information both in frequency and temporal dimension of feature maps. In order to preserve the time domain information which may be critical for GRU performance, we introduced 2×1 rectangular kernels in the CNN module pooling layers for TS-C2Rnn-P architecture. This results in restricting the pooling of feature maps in the frequency dimension.

Both the proposed networks TS-C2Rnn & TS-C2Rnn-P), were first trained on SED task and then on the localization task. In the first stage, all features are fed into the network to train for SED task and only the loss of SED is minimized. After SED have been trained, the learned weights from the convBlocks in the SED branch is transferred to the convBlocks in the localization branch to train

for the localization task using the reference event labels to mask the localization predictions.

2.3. Data Augmentation: *MIXUP*

For better model generalization, we adopted *mixup* [10] a data augmentation technique which is a popular for image classification tasks. It has been illustrated in [10] that mixup scheme helps in alleviating undesirable behaviors of the deep neural network such as memorization and sensitivity to adversarial examples. *Mixup* is a data-agnostic data augmentation routine. It makes decision boundaries transit linearly from class to class, providing a smoother estimate of uncertainty.

The idea behind *mixup* is that of risk minimization. We wish to determine a function f that describes the relationship between input x_i and target y_i , and follows the joint distribution $P(x, y)$. We can minimize the average of the loss function ℓ (or expected risk) over P in the following manner

$$R(f) = \int \ell(f(x), y) dP(x, y)\,,$$

where $f(x)$ is a function that describes the relationship between input vector x and target vector y, ℓ is the loss function that penalizes the difference between the output of $f(x)$ and target y. While P is unknown is most practical cases, it can be approximated. There are two such approximations raised in [10] namely *empirical risk minimization* [11] and *vicinal risk minimization* [12]. While the vicinal virtual input-target pairs are generated by addition of Gaussian noise in [12], Zhang et al. [10] proposed the generation of virtual input and target pairs as such,

$$\begin{aligned} X &= \lambda \times x_1 + (1 - \lambda) \times x_2, \\ Y &= \lambda \times y_1 + (1 - \lambda) \times y_2, \end{aligned} \tag{1}$$

where λ is a weight drawn from the beta distribution with parameters $\alpha, \beta = 0.2$ and x_1, x_2, y_1 and y_2 are two pairs of input-target pairs drawn randomly from the dataset. The parameters α and β are chosen such that the probability density is denser in the domain $0 < \lambda < 0.1$ and $0.9 < \lambda < 1.0$ which can be seen in Figure 3. The average of the loss function can then be minimized over this probability distribution approximation.

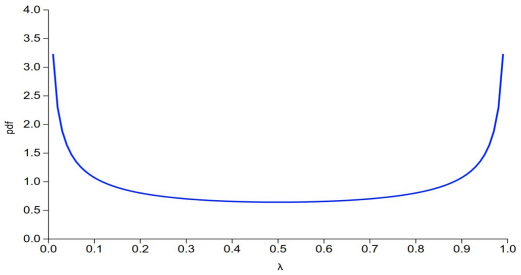

Figure 3: Beta distribution with $\alpha, \beta = 0.2$

3. EXPERIMENTS AND RESULTS

DCASE2019 Task-3 organizers has provided two datasets [7], TAU Spatial Sound Events 2019: Ambisonic, Microphone Array datasets of an identical sound scene with only difference in the format of the audio. In this work, we only used TAU Spatial Sound Events 2019: Microphone array dataset for all our experiments. The dataset consist of multiple audio recordings from 4 channel, directional microphones arranged in a tetrahedral array configuration with overlapping sound events recorded in different environments. Dataset is divided into two sets, development set and evaluation set. The development set consists of 400, one minute long recordings sampled at 48kHz, divided into four cross-validation splits of 100 recordings each. The evaluation set consists of 100, one-minute recordings. There are total 11 isolated classes of the sound events. We trained our network using this 4 pre-defined cross-validation folds and the final results are the overall aggregated from the test data of all 4 folds in the development set. The performance of the architecture is evaluated with individual metrics, for SED F-score and error rate (ER) was considered and for localization task, direction of arrival (DOA) error and frame recall (FR) were used. We trained our network with an objective to achieve lower DOA error & ER and higher FR & F-score.

Below is the list of prior arts and proposed architectures used for experiments and evaluations.

- **Baseline**, which is the benchmark model [5] released by DCASE-2019 Task-3 organizers. This network is based on the CRNN architecture, and take magnitude & phase spectrogram as input features.
- **SELDNet**, this network has the same architecture as in Baseline, but instead of magnitude & phase spectrogram, it takes log-mel spectrograms & GCC-PHAT as input features.
- **Two-Stage (TS)**, this network has CRNN architecture and is based on two stage training methodology [6] .
- **TS-CRnn**, same as our base network architecture TS-C2Rnn except only 1 GRU layer used as the RNN.
- **TS-C2Rnn**, our base network architecture as illustrated in Figure 1 and explained in section 2.2 .
- **TS-C2Rnn-P**, the modified version of our base network architecture TS-C2Rnn, which has 2×1 kernel size for pooling layers, as illustrated in Figure 2 and explained in section 2.2.

Table 1 presents the performance results on development set w.r.t to prior-arts for SED and localization task, with the effect of data augmentation technique, *mixup*. Compared with Baseline, our base network TS-C2Rnn without *mixup* shows 12.6% and 50.2% improvement on ER and F-score respectively for the SED task, while for localization task it show 20° and 2.6% improvement on DOA error and FR respectively. This result shows that deep CRNN based architectures improves the performance for SELD task compared to CNN based architecture.

In addition, TS-C2Rnn-P architecture which uses average pooling with kernel size of 2×1 in the CNN module, shows the best improvements with the best score across all evaluation metrics. For the SED task, TS-C2Rnn-P achieved an error rate of 0.149 and an F-score of 91.9%. For the localization evaluation metrics, it achieved a DOA error of 4.588 °and frame recall of 0.896. It shows improvement of 13% and 4° respectively on ER and DOA error, over the state-of-art Two-Stage(TS) network. This result infers that 2×1 pooling in the CNN module of TS-C2Rnn-P, helps it to learn the spectral information efficiently, and at the same time minimize the loss of information in the temporal dimension. In turn there is more information available to the RNN module, which helps in effectively learning the temporal information. This lead to boosting up

	no *mixup*				with *mixup*			
	ER	**F-score**	**DOA(°)**	**FR**	**ER**	**F-score**	**DOA(°)**	**FR**
Baseline	0.350	0.800	30.800	0.840	—	—	—	—
SELDNet	0.213	0.879	11.300	0.847	—	—	—	—
Two-Stage (TS)	0.166	0.908	9.619	0.863	0.194	0.888	8.901	0.839
TS-CRnn	0.186	0.897	9.450	0.857	0.200	0.888	7.866	0.841
TS-C2Rnn	0.174	0.901	8.881	0.862	0.176	0.903	7.236	0.856
TS-C2Rnn-P	0.147	0.916	5.631	0.902	0.149	0.919	4.588	0.896

Table 1: Performance evaluation of the proposed network architecture on the development set comparing with prior-arts

	DCASE-2019 Task-3 Evaluation-result				
Team Name	**Rank**	**ER**	**F-score**	**DOA(°)**	**FR**
Kapka_SRPOL [13]	1	0.08	0.947	3.7	0.968
Cao_Surrey [14]	2	0.08	0.955	5.5	0.922
Xue_JDAI [15]	3	0.06	0.963	9.7	0.923
He_THU [16]	4	0.06	0.967	22.4	0.941
Jee_NTU (our)	5	0.12	0.937	4.2	0.918

Table 2: Comparison of top 5 results of DCASE-2019 Task-3.

of the overall performance of the proposed TS-C2Rnn-P network.

Table 1 also illustrate the effect of data augmentation i.e. *mixup* on the performance of above mention networks. Comparing the results we can infer that upon applying the *mixup* the F-score slightly dropped while the DOA error improved. We realized that the *mixup* is having positive effect on improving the localization task performance but at the same time it is showing a slight drop or no change in performance for the SED task. We applied a new training strategy of applying *mixup* only on the localization task during training, as there is no effect of *mixup* on SED task.

	mixup on localization task only			
	ER	**F-score**	**DOA(°)**	**FR**
Two-Stage	0.175	0.903	8.056	0.861
TS-C2Rnn	0.171	0.903	7.486	0.861
TS-C2Rnn-P	0.144	0.904	4.746	0.902

Table 3: Results from using *mixup* in localization branch training only.

Comparing between the performance of Two-Stage, TS-C2Rnn and TS-C2Rnn-P in Table 1 (no *mixup*) and Table 3 (*mixup* on localization task only), an improvement could be seen across all four evaluation metrics for all of the networks. In contrast while comparing the results of these network in Table 1 (with *mixup*) and Table 3, only three metrics showed an improvement while DOA error increased. This abnormality tell us that, training for localization task is built upon the trained weights of the SED task, therefore for improving the results in the localization branch, SED results are also essential. Although *mixup* appear to slightly drop in the performance score for SED predictions, but its negative effect on SED score, appears to have a positive performance surge on the localization task. The negative effects of *mixup* on the SED branch appeared to be suppressed by increasing the number of layers. This can be seen from the F-score converging to 0.904 for networks tested with *mixup* applications in Table 3.

The DOA error could be improved further by learning from the trained weights of a *mixup*-applied SED task instead of a non-*mixup*-applied SED task although that would adversely affect the results of SED predictions. Thus, a balance must be found in the use of *mixup*, depending on the use case and the allowance for error in SED and DOA predictions.

Table 2 presents the top 5 teams results on the DCASE-2019 Task 3 evaluation set. In this Table *"Jee_NTU"* refers to the results of TS-C2Rnn-P architecture proposed in this work. From the table we can infer that our DOA error performance, which is the rank 5 system has given a positive improvement over rank 2-4 systems. In addition our overall F-score for the SED task is comparable with other systems. With the usage of both the data sets provided by DCASE-2019 Task 3 and including mixup, we believe TS-C2Rnn-P can yield similar results as the top system.

4. CONCLUSION & FUTURE WORK

In this paper, we proposed CRNN architecture with *mixup* as data augmentation technique for SELD task. Experimentally, we have shown that using *mixup* helps in improving the localization performance. In addition, usage of rectangular kernels for the pooling layers helps in overall performance of SED and localization. Experimental results show that our proposed network architecture TS-C2Rnn-P with *mixup* is shown to significantly outperform the baseline system for both SED and localization task. For future studies, the changing of parameters α and β in *mixup* can be investigated. The parameters were chosen so as not to create too many vastly different virtual input-target pairs. There might be a beneficial improvement if the λ is less heavily weighted to one side of the input-target pair.

5. REFERENCES

[1] A. Kumar and B. Raj, "Deep cnn framework for audio event recognition using weakly labeled web data," *arXiv preprint arXiv:1707.02530*, 2017.

[2] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[3] J. Vera-Diaz, D. Pizarro, and J. Macias-Guarasa, "Towards end-to-end acoustic localization using deep learning: From audio signals to source position coordinates," *Sensors*, vol. 18, no. 10, p. 3418, 2018.

[4] W. He, P. Motlicek, and J.-M. Odobez, "Deep neural networks for multiple speaker detection and localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 74–79.

[5] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, March 2019.

[6] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," *arXiv preprint arXiv:1905.00268*, 2019. [Online]. Available: https://arxiv.org/pdf/1905.00268.pdf

[7] S. Adavanne, A. Politis, and T. Virtanen, "A multi-room reverberant dataset for sound event localization and uetection," in *Submitted to Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, 2019. [Online]. Available: https://arxiv.org/abs/1905.08546

[8] B. Logan *et al.*, "Mel frequency cepstral coefficients for music modeling." in *ISMIR*, vol. 270, 2000, pp. 1–11.

[9] J. Hassab and R. Boucher, "Performance of the generalized cross correlator in the presence of a strong spectral peak in the signal," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 549–555, 1981.

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

[11] V. N. Vapnik, "Statistical learning theory," vol. 1, 1998.

[12] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Advances in neural information processing systems*, 2001, pp. 416–422.

[13] S. Kapka and M. Lewandowski, "Sound source detection, localization and classification using consecutive ensemble of crnn models," DCASE2019 Challenge, Tech. Rep., June 2019.

[14] Y. Cao, T. Iqbal, Q. Kong, M. Galindo, W. Wang, and M. Plumbley, "Two-stage sound event localization and detection using intensity vector and generalized cross-correlation," DCASE2019 Challenge, Tech. Rep., June 2019.

[15] W. Xue, T. Ying, Z. Chao, and D. Guohong, "Multi-beam and multi-task learning for joint sound event detection and localization," DCASE2019 Challenge, Tech. Rep., June 2019.

[16] J. Zhang, W. Ding, and L. He, "Data augmentation and prior knowledge-based regularization for sound event localization and detection," DCASE2019 Challenge, Tech. Rep., June 2019.