A SEQUENTIAL SYSTEM FOR SOUND EVENT DETECTION AND LOCALIZATION USING CRNN

Technical Report

Rohit Singla

Samsung Research Institute, Bengaluru
Sound Intelligence
Bengaluru, 560037, India
rohit.singla@samsung.com

Sourabh Tiwari

Samsung Research Institute, Bengaluru
Sound Intelligence
Bengaluru, 560037, India
sourabh.t@samsung.com

Rajat Sharma

Samsung Research Institute, Bengaluru
Sound Intelligence
Bengaluru, 560037, India
raj.sharma@samsung.com

ABSTRACT

In this technical report, we describe our method for DCASE2020 task 3: Sound Event Localization and Detection. We use a CRNN SELDnet-like single output models which run on the features extracted from audio files using log-mel spectrogram. Our model uses CNN layers followed by RNN layers followed by predicting sound event classes: Sound Event Detection (SED) and then giving the output of SED to estimate Direction Of Arrival (DOA) for those sound events and then the final output is given as a concatenation of SED and DOA. The proposed approach is evaluated on the development set of TAU Spatial Sound Events 2020 – First-Order Ambisonics (FOA).

Index Terms— CNN, RNN, DCASE2020, log-mel, Sound Event Detection

1. INTRODUCTION

Sound Event Localization and Detection (SELD) is a complex task in which along with predicting the sound events that are occurring in our surroundings, the direction of arrival of those sounds is also estimated. It is very useful in learning the surrounding environments and to alert people in case of some accidents which produce sounds that can be recognized by systems. SELDnet introduced in [1], which is given as a baseline system to the participants is a reasonably good system that gives an SELD score of 0.47. In our work, the task of SELD is performed as Sound Event Detection (SED) which is given as input to Direction of Arrival (DOA) and then we concatenate the outputs of both for the purpose of evaluation.

2. FEATURES EXTRACTION

We have been given two formats of audio input to use: first order ambisonics (FOA) and tetrahedral microphone array. We have used first order ambisonics format for our model. We are given 800 audio recording files. There are 4 channels in each of the 800 audio files. Each recording is approximately 1-minute-long with sampling rate of 24kHz. Figure 1(a) shows a wave-form corresponding to one of the audio recordings from our dataset.

We use short time Fourier transform (STFT) with Hanning window to generate spectrograms. We use window of length 0.04s and hop of length 0.02s in STFT to transform a raw audio associated to each FOA channel into a spectrogram of size 3000x513. Figure 1(b) shows the spectrogram for the same audio recording. From each recording we acquire 4 standardized amplitude spectrograms in decibel scale and 4 standardized phase spectrograms corresponding to 4 FOA channels. We reduce these 4 channel-spectrograms a single log-mel spectrogram of size 3000x256. Figure 1(c) shows the log-mel spectrogram for the same audio recording. This log-mel spectrogram is stacked with intensity vectors of size 3000x64 for 3 directions of Cartesian Coordinates. For each audio recording we get a 3000x448 size array which is then normalized using Standardized Scalar.

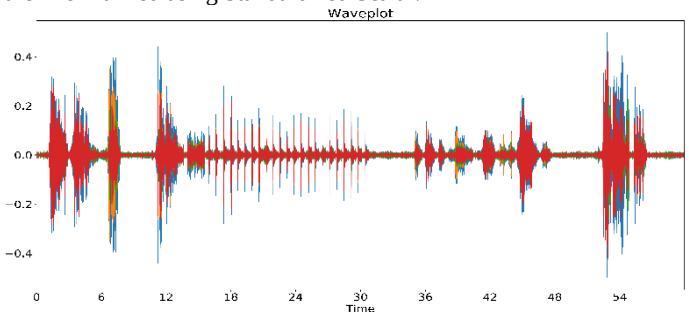

Figure 1(a): Waveform

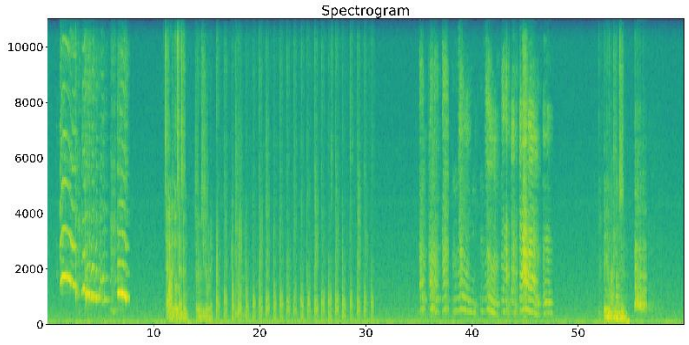

Figure 1(b): Spectrogram

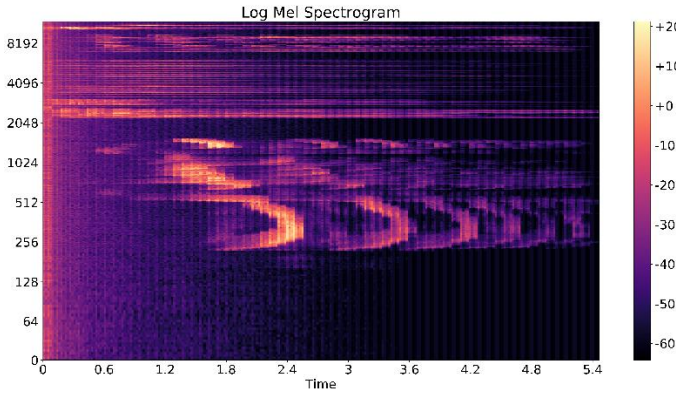

Figure 1(c): Log Mel-Spectrogram

3. LABEL EXTRACTION

Each record has been labelled in this format: -
`<frame, class, index, elevation, azimuth>`
The 1-minute audio recording is converted into 600 frames of 100ms each. So, `frame` is in [0,599] with 0-indexing.
`Class` corresponds to one of the 14 sound event classes with 0-indexing
The following sound classes of the spatialized events are used:

1. alarm
2. crying baby
3. crash
4. barking dog
5. running engine
6. female scream
7. female speech
8. burning fire
9. footsteps
10. knocking on door
11. male scream
12. male speech
13. ringing phone
14. piano

`Index` corresponds to the index of a particular class within frame starting from 0, if the source for that class has moved within that frame, leading to change in DOA.
`Elevation` and `Azimuth` are polar coordinates of source's spatial location. To feed our model, we have converted them into Cartesian coordinates (x, y, z).

These labels are extracted as an array of (600,56) for each of the train and test set labels file.

Here 600 corresponds to number of frames of 100ms in a 1-minute audio recording. The first 14 columns are {0,1}. 1 if the event corresponding to each class has happened in that frame; 0 otherwise. Rest of the 42 columns correspond to the Cartesian coordinates (x, y, z) for all the 14 classes. They are marked as 0 for the event classes that have not occurred in the frame.

4. DATA AUGMENTATION

After running baseline model, we analyzed its class-wise performance. There were few classes for which occurrences in training and validation dataset were very less. Hence, the model was performing very poor on those classes. Those classes along with their performance on validation dataset are shown in Table 1

Class	F	DE	DE_F
5 (Running Engine)	0.07	33.76	0.41
6 (Female Scream)	0.00	55.71	0.77
9 (Footsteps)	0.00	55.87	0.49
10 (Knocking on door)	0.01	35.35	0.55
11 (Male scream)	0.00	56.12	0.37
12 (Male speech)	0.18	21.45	0.32

Table 1: Classes with poor performance

We filtered only those frames in audio files which were having occurrences of any of these classes. This resulted in 559 out of 600 files of train + validation dataset. Some of these files were of ov1 category (no overlapping events) and others were of ov2 category (2 events overlapping at some time-point).
As removing other classes resulted in most of the parts of the audio as empty. Thus, we used only those files from ov2 in which at least half of the time-frames (300 out of 600) were active (having some sound events). This resulted in 20 such files, which we used after applying time-stretch augmentation with 1.07 and 0.81. So, we used 40 files from here.
For ov1 category, we decided to overlap signals among files within same fold and same room and use only those files to make our system more robust towards overlapping sounds. We overlapped sound signals within the same fold and same room pairwise and used only those which resulted in overlap of at least 5 seconds (50 frames). This resulted in 254 such files.
Overall, we augmented our training data with 294 more files of ov2 category.

5. MODEL ARCHITECTURE

We are given 800 audio recording files as 8 folds of 100 files each. They are split into train, valid and test set as shown in Table 2:

Train (Train Model)	Fold 2,3,4,5,6
Valid (Test Model)	Fold 1
Test (Submission)	Fold 7,8

Table 2: Dataset splits

Feature and label sequence lengths have been kept as a segment of $^1/_{10}\ th$ of the original audio
Feature sequence length $= 300$
Label sequence length $= 60$
Batch size is taken to be 256

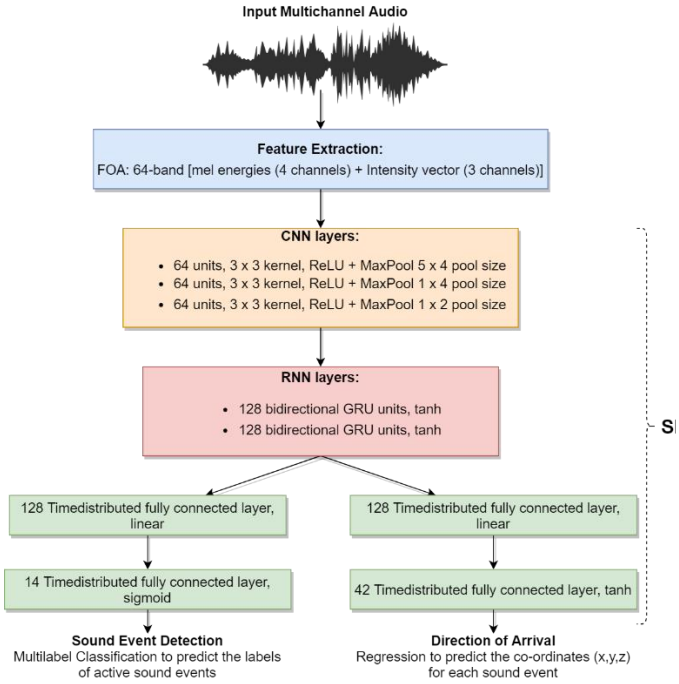

Figure 2(a): Baseline Model Architecture

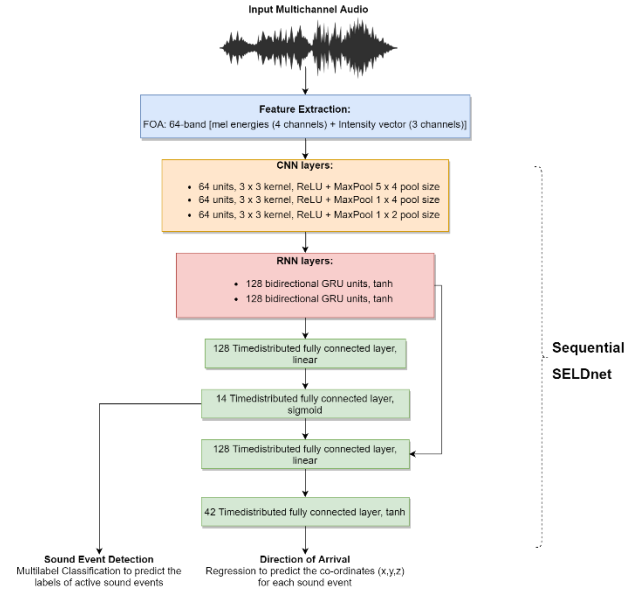

Figure 2(b): Sequential SED+DOA Model Architecture

Feature batch sequence length = $256 * 300 = 76800$
Label batch sequence length = $256 * 60 = 15360$
Number of files in train set = 500
Number of batches in train set = $(500 * 3000)/76800 = 19$
Number of channels = 7 (4 corresponding to channels of audio input + 3 for the 3 axes of Cartesian coordinates)
Number of mel-bins = 64

So dimensions of the input data to model is (256,7,300,64) and that of output of the model is [(256,60,14), (256,60,42)]
Here (256,60,14) and (256,60,42) corresponds to SED and DOA parts respectively.

The model is run for 50 epochs.

The baseline SELDnet model [1] as given for this task has been shown in Figure 2(a). The Input layer is followed by 3 CNN layers, each with kernel size of (3,3), ReLU activation function and MaxPooling layer. These are then followed by 2 bidirectional GRUs with 128 units. Then SED and DOA part of the models run in parallel, independent of each other as 2 time-distributed fully connected layers each. The outputs of SED and DOA are then concatenated and given as final output of the model for evaluation.

We tried to run the SED part and DOA part of the model sequentially as shown in Figure 2(b). The advantage of this approach is that we are giving output of SED part which predicts the sound events in time frames as input to the DOA part of the model along with the final RNN layer output that was initially given to the model. So, the DOA part of the model now predicts the direction of arrival of events given the events that have occurred. We also tried out keeping a dropout rate of 0.1 for all layers in this sequential SED+DOA model.

6. EVALUATION METRICS

Model has been evaluated on both SED and DOA parts.

For SED part,
DE_F = F-score for events detected without considering any location
DE = average distance between system detected and ground truth events without considering any location threshold

For DOA part considering threshold of 20 degrees of angle for direction of arrival,
ER = Error rate, out of reference events, how many are: S (swap errors), I (insertions) or D (Deletions)
F = F-score for events detected

Final SELD scores is calculated as mean of DE, 1-DE_F, EF, 1-F

7. RESULTS AND SUBMISSIONS

We are doing 4 submissions for this Task 3 of SELD. These have been described in Table 3(a) and their results on validation split have been described in Table 3(b).

Submission Name	Description
Singla_SRIB_ task3_1	Baseline Model
Singla_SRIB_ task3_2	Sequential SED+DOA Model
Singla_SRIB_ task3_3	Dropout rate of 0.1 over Sequential Model
Singla_SRIB_ task3_4	Data Augmentation over Baseline Model

Table 3(a): Submission Descriptions

Submission	ER	F	DE	DE_F	SELD
Singla_SRIB_task3_1	0.73	34.2	24.3	65.8	0.47
Singla_SRIB_task3_2	**0.72**	**36.2**	**23.4**	**67.7**	**0.45**
Singla_SRIB_task3_3	0.78	27.1	25.6	62.3	0.51
Singla_SRIB_task3_4	0.83	25.5	26.9	56.9	0.54

Table 3(b): Submission Scores on Validation Dataset

Thus, it can be inferred that our sequential SED+DOA model is giving best results on validation dataset. Our results are marginally better than the baseline model on all evaluation parameters.

The visualization of output for one of the audio files is shown in Figure 3

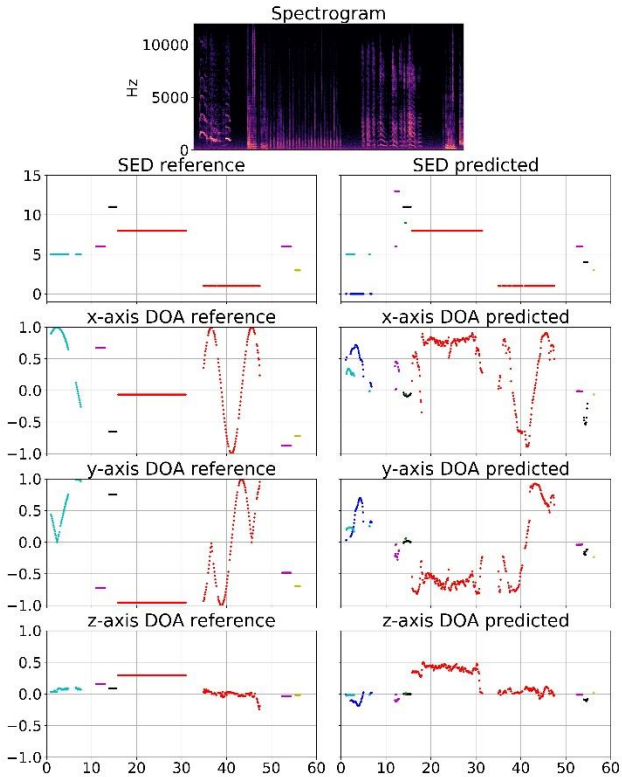

Figure 3: Output Visualization

8. REFERENCES

[1] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. IEEE Journal of Selected Topics in Signal Processing, 13(1):34–48, March 2018.

[2] http://dcase.community/challenge2020/task-sound-event-localization-and-detection

[3] Ivo Trowitzsch, Jalil Taghia, Youssef Kashef, and Klaus Obermayer (2019). The NIGENS general sound events database. Technische Universität Berlin, Tech. Rep. arXiv:1902.08314 [cs.SD]

[4] 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

[5] Archontis Politis, Sharath Adavanne, and Tuomas Virtanen. A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection. arXiv e-prints: 2006.01919, 2020. URL: https://arxiv.org/abs/2006.01919

[6] Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. IEEE Journal of Selected Topics in Signal Processing, 13(1):34–48, March 2018. URL: https://ieeexplore.ieee.org/abstract/document/8567942