DCASE 2020 TASK 3: A SINGLE STAGE FULLY CONVOLUTIONAL NEURAL NETWORK FOR SOUND SOURCE LOCALIZATION AND DETECTION

Technical Report

Sohel J. Patel[1]*, *Maciej Zawodniok*[1], *Jacob Benesty*[2]

[1]Missouri University of Science & Technology, Department of Electrical Engineering, Missouri, USA
{sjp432,mjzx9c}@mst.edu
[2]INRS-EMT, University of Quebec, Montreal, Canada
benesty@emt.inrs.ca

ABSTRACT

In this report, we present our approach for DCASE 2020 Challenge Task3: Sound event localization and detection. We use a single step training method using SELDNet like models but using fully convolutional architectures. We consider the joint optimization of both event detection and doa estimation. For the metrics that evaluate the performance of the model consider interdependence of both parameters performance unlike independent performance like DCASE 2019 challenge. We use all the sound event classes and corresponding cartesian co-ordinates for each class to create an image like label for reference and make this an image to image mapping problem. The best model could get DOA error of around 13.5^0 and error rate of 0.55.

Index Terms— DCASE 2020, SELDNet, Fully convolutional networks, sound event localization and detection, skip connections

1. INTRODUCTION

Sound event Localization and detection has been an interesting topic for research for a long time. Previously it was a very challenging to achieve satisfactory performance mainly because of their implementation was based on pure signal processing algorithms. Recently with availability of comprehensive databases and computational resources several interesting performances have been observed. Researchers have proposed several deep learning algorithms that combined solve the localization and detection problem. However, most of the proposed approaches consider raw spectrograms as an input feature to neural networks. Such kind of features may be okay for application involving speech enhancement, dereverberation and few other problems where the networks learn some kind of patterns like formants, pitch etc.. Such features are however not applicable to sound events detection most of the time for they do not consider human speech as their only inputs.

A very recent implementation of SELD has been described in [1] where a Convolutional Recurrent Neural Network (CRNN) is trained using magnitude and phase spectrograms to predict active sound events and their location w.r.t the microphone array. In DCASE 2019 challenge [2], many research teams have proposed models with state of the art performances with reduced feature sizes. For example in [3] mel-scale spectrograms and phase transformed generalized cross correlation (GCC-PHAT) have been extracted from the spectrograms which contain the necessary and sufficient information for the model to learn patterns to predict active sound events class and their location.

Most of the previous approaches make use of ensemble models [4] and average them [3] in the end to further reduce the training error and avoid overfitting. This method has an advantage when running the evaluation dataset on these trained models to achieve reasonable performance. However, in our approach we do not train such ensemble models for they require lot of hyperparameters tuning.

2. FEATURE AND LABEL EXTRACTION

The TAU-NIGENS Spatial Sound Events 2020 dataset [5] consists of two recordings format: first order ambisonic (foa) and 4 channels from a microphone array (mic). We use both microphone array (MIC) and first order ambisonics (FOA) format in our experiments. The development dataset consists of a total of 600 recordings each one minute long sampled at 24000 Hz.

The labels represent the location of each sound source and their corresponding class for every 0.1s. Therefore, for each recording there are 600 labels. For hop length of 0.02 seconds and window length 0.04 seconds, a complex spectrogram of size 3000x512 can be derived. This means the labels represent ground truth for every 5 spectrogram frames. Processing the spectrogram this size can be time consuming and redundant.

For this kind of application it has been shown the instead of using raw spectrograms for training neural networks, we can extract useful features like mel spectrograms [3]. For both FOA and MIC dataset, the mel spectrograms are calculated. In addition to this the GCC-PHAT are also calculated for the mel bands. We use the 64 mel frequency bands as suggested in the baseline thus reducing the input feature mel-spectrogram to 3000x64 with a total of 17 channels (10 MIC and 7 FOA).

There are a total of 14 sound events with 2 or less sound events active at any given time. Hence the sound event detection (SED) Labels can be structured as a vector of length 14 for 14 classes with any particular sound event as active or not by a binary 0 *or* 1. Similarly, we have another 14 reference labels for each x, y and z co-ordinates where any particular active sound event is given its co-ordinates rest being zero. We use the cartesian co-ordinate system over spherical for all the reference labels lie between (-1,1). This way each corresponding label now has 56 labels of which first 14 are binary 0/1

*Research reported in this publication was partially supported by Navy SBIR program under award number N182-100.

SED labels and next 42 are x, y and z DOA labels.

since the hop length (0.02 s) is $1/5^{th}$ of the time for which labels have been provided (0.1 s), we upsample the labels by copy the labels 5 times to match the input feature size in time steps. For example, for every 300 input frames, there would be (300/5=60) reference labels. After upsampling, there are labels for each time frame. As a result, for input feature size of (300 x 64) the labels matrix size is (300 x 56). This can be considered as image to image mapping problem since all the reference labels are within (-1,1)

3. ARCHITECTURE

The baseline architecture consists of a convolutional recurrent neural network with 3 convolutional layers followed by bidirectional GRUs and Dense layers. The SED task is considered as classification problem and the DOA is considered regression problem. In our approach, we consider the entire problem as regression and train image to image network using Fully convolutional layers with GRUs.

The architecture we use is a modified version of U-NET [6] with skip connections replaced by convolutional and GRU layers as seen in Fig. 1. Each path towards the output with/without skip connections serves similar to SELDNet like CRNN model. The intuition behind this architecture is to avoid the use of ensemble models and stack all the models in the final stage. In total there are 4 paths serving as 4 SELDNet like models. The main path consists of the following sequence of layers. The squeezing path is given by:
encoder1 $\rightarrow$ encoder2 $\rightarrow$ encoder3 $\rightarrow$ encoder4 $\rightarrow$ GRU4 $\rightarrow$ decoder1 $\rightarrow$ decoder2 $\rightarrow$ decoder3 $\rightarrow$ decoder4 $\rightarrow$ output.

Throughout the training we use (3 x 3) convolutional filters with padding on both sides unless specified. We specifically use padding and max pooling (stride=(2 x 2)) to reduce feature maps instead of stride to change the shape of the output image to that of the reference label. The layers in the main path and their hyper-parameters with corresponding outputs are as described Table.1. Encoder1 and 2 blocks have not been zero padded along third dimension (frequency) while performing convolutions. As such after the convolution and max-pooling their size is reduced to (32 x 300 x 62) and (32 x 150 x 31) respectively instead of (32 x 300 x 64) and (32 x 150 x 32) if done with zero padding along third dimension. Similarly for Encoder2 the output is reduced to (64 x 75 x 14). Encoder3 and 4 follow usually with zero padding along both the dimensions.

The output of GRU4 is then upsampled by doing transposed convolutions to get (128 x 37 x 7). This output is concatenated with encoder3 output and again upsampled. However, the output of decoder3 has (150 x 28) feature maps as opposed to (150 x 31) of encoder 1. Hence in encoder1, we crop the first and last 2 features across last dimension and perform upsampling to achieve (150 x 28) features. The last layer does not need to any concatenation and is directly upsampled to achieve the predicted labels. Unlike U-NET that performs all symmetric reduction and upsampling of features, this approach has asymmetric skipped operations.

There are in total three skipped connections in this architecture. Each connection consists of repeated convolutions followed by Bidirectional GRUs. The (conv x 5), (conv x 4) and (conv x 3) represents performing 5, 4 and 3 repeated convolutions. The layers details can be found in Table.2. The convolutions are performed with same input and output features such that their dimensions remain same in order to concatenate with upsampled features.

Stage	Output	Layers
input	17x300x64	input features
encoder1	32x150x31	conv,BN,pReLu,maxpool,pad(1,0)
encoder2	64x75x14	conv,BN,pReLu,maxpool,pad(1,0)
encoder3	128x37x7	conv,BN,pReLu,maxpool,pad(1,1)
encoder4	256x18x3	conv,BN,ReLu,maxpool,pad(1,1)
GRU 4	256x18x3	Bidirectional 2 layer,768 GRU units
decoder1	128x37x7	convTranspose,BN, ReLu
decoder2	64x75x14	convTranspose,BN, ReLu
decoder3	32x150x28	convTranspose,BN, ReLu
decoder4	1x300x56	convTranspose,BN, TanH

Table 1: Encoder Decoder Architecture

Stage	Output	Layers
conv x 5	32x150x31	5 repeated convolutions conv,BN,ReLu
GRU 1	32x150x31	Bidirectional 2 layer,992 GRU units
conv x 4	64x75x14	4 repeated convolutions conv,BN,ReLu
GRU 2	64x75x14	Bidirectional 2 layer,896 GRU units
conv x 3	128x37x7	3 repeated convolutions conv,BN,ReLu
GRU 3	128x37x7	Bidirectional 2 layer,896 GRU units

Table 2: Skip connection/concatenation layers

4. RESULTS

The trained models are used to predict the outcomes of the validation and test dataset. All GRU layers are set with dropout of 0.2. The approach used in this study is very specific to the input feature dimensions. For other previous approaches like SELDNet models [1] and ensemble approaches, the model can accept variable inputs. But in our case, since we are performing asymmetric skip connections in the so called fully convolutional architecture and with image to image learning and the size of input and output image being different. The entire model needs to be changed in order to be compatible with different input feature shape.

We consider input feature to have 300 frames or 300 time steps with 64 mel bands. The image is then passed through the network to output a (300 x 56) image with predicted SED+DOA (14+42) values. From this output, the SED corresponding values are rounded to their nearest integer. Since the reference labels have been provided for every 5 frames we squeeze out one set of labels for every 5 time steps to calculate the SELD metrics.

5. SUBMISSIONS

Of the four submissions first two are FCn based and the other two are CRNN architecture based similar to SELDNet. Although the performance of FCn based model has better performance but it is only slight improvement in terms of DOA error as compared to submission 3 and 4. The layers for latter model are given in Table.4. The SED labels are considered as classification problem with and DOA are considered as regression while calculating losses. Unlike the FCn model where entire model output considered as regression

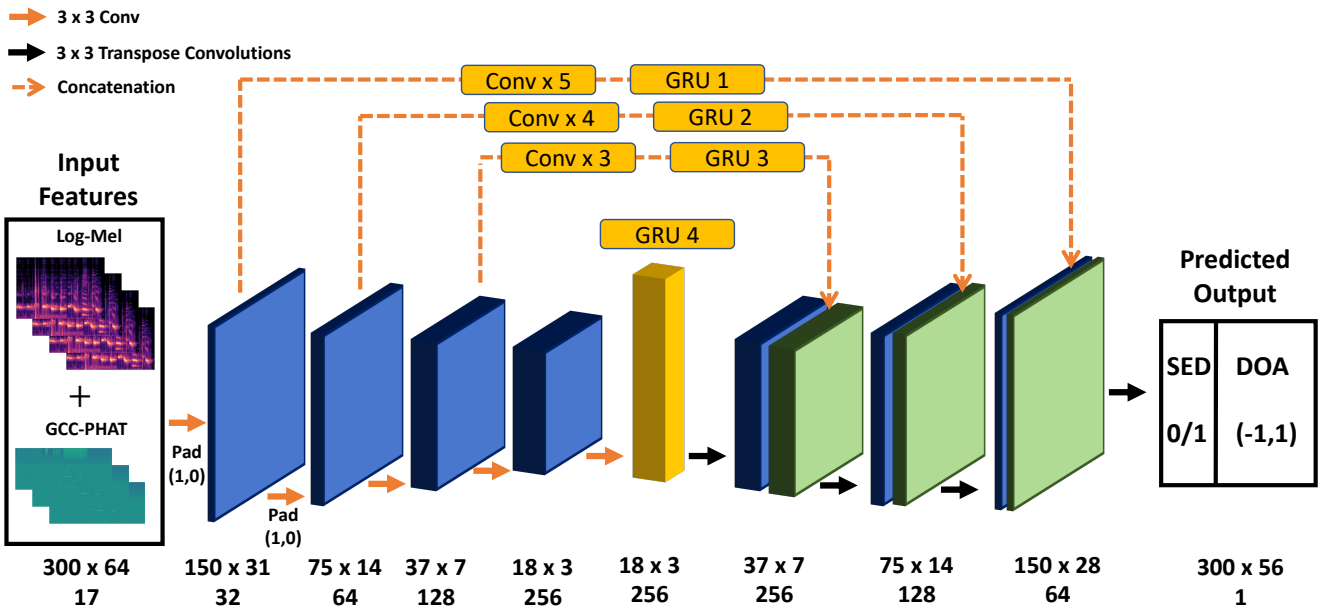

Figure 1: Fully convolutional architecture with skip conv-GRU layers for SELD

Submission	ER_{20^0}	F_{20^0}	LE_{CD}	LR_{CD}	SELD
1	0.55	54.2	13.6^0	63.6	0.35
2	0.56	53.7	14^0	62.6	0.37
3	0.55	55.4	14.9^0	66.5	0.35
4	0.54	55.6	15.2^0	67.2	0.35
baseline	0.72	37.4	22.8^0	60.7	0.49

Table 3: Results on development dataset

problem. For all submissions the learning rate was fixed throughout training and was optimized using Adam optimizer. For submissions (1,3) and (2,4), the learning rate was 0.0001 and 0.00005 respectively.

6. CONCLUSION

We conclude that the performance of our approach outperform baseline methods. But there definitely is further room for improvement. More complex and deeper architectures need to be experimented with like ResNET, Inception models among others. Also ensemble methods might lead to further reduction in training losses and better generalization for evaluation dataset.

7. REFERENCES

[1] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping

Stage	Output	Layers
input	17x300x64	input features
conv1	32x300x64	conv,BN,ReLu
conv2	64x300x32	conv,BN,ReLu,maxpool(1,2)
conv3	128x300x16	conv,BN,ReLu,maxpool(1,2)
conv4	256x300x8	conv,BN,ReLu,maxpool(1,2)
conv5	512x300x4	conv,BN,ReLu,maxpool(1,2)
conv6	512x60x2	conv,BN,ReLu,maxpool(5,2)
GRU	60x1024	2 layer,1024 GRU units
dense1	60x512	Fully connected layer,dropout
dense SED	60x14	Fully connected layer
dense2	60x512	Fully connected layer,dropout
dense DOA	60x42	Fully connected layer

Table 4: Alternate SELDNet like CRNN architecture

sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, March 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8567942

[2] http://dcase.community/challenge2019/.

[3] Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. Plumbley, "Polyphonic sound event detection and localization using a two-stage strategy," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 30–34.

[4] S. Kapka and M. Lewandowski, "Sound source detection, localization and classification using consecutive ensemble of crnn models," *arXiv preprint arXiv:1908.00766*, 2019.

[5] A. Politis, S. Adavanne, and T. Virtanen, "A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection," *arXiv e-prints: 2006.01919*, 2020. [Online]. Available: https://arxiv.org/abs/2006.01919

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.