COMBINING MULTI-SCALE FEATURES USING SAMPLE-LEVEL DEEP CONVOLUTIONAL NEURAL NETWORKS FOR WEAKLY SUPERVISED SOUND EVENT DETECTION

Jongpil Lee[1]*, Jiyoung Park*[1]*, Sangeun Kum*[1]*, Youngho Jeong*[2]*, Juhan Nam*[1]*,*

[1] Graduate School of Culture Technology, KAIST, Korea,
[2] Realistic AV Research Group, ETRI, Korea,
{richter, jypark527, keums, juhannam}@kaist.ac.kr, yhcheong@etri.re.kr

ABSTRACT

This paper describes our method submitted to large-scale weakly supervised sound event detection for smart cars in the DCASE Challenge 2017. It is based on two deep neural network methods suggested for music auto-tagging. One is training sample-level Deep Convolutional Neural Networks (DCNN) using raw waveforms as a feature extractor. The other is aggregating features on multi-scaled models of the DCNNs and making final predictions from them. With this approach, we achieved the best results, 47.3% in F-score on subtask A (audio tagging) and 0.75 in error rate on subtask B (sound event detection) in the evaluation. These results show that the waveform-based models can be comparable to spectrogram-based models when compared to other DCASE Task 4 submissions. Finally, we visualize hierarchically learned filters from the challenge dataset in each layer of the waveform-based model to explain how they discriminate the events.

Index Terms— Sound event detection, audio tagging, weakly supervised learning, multi-scale features, sample-level, convolutional neural networks, raw waveforms

1. INTRODUCTION

Understanding the sounds of everyday life has received great attention in recent years due to its practical applications such as the hearing impaired, smart cars and smart appliances [1, 2, 3, 4, 5]. Among others, Sound Event Detection (SED) is a particularly challenging task because it predicts not only possible descriptive words of environment sounds but also their start and end times. Most SED systems are based on hard annotated data where both event classes and their timestamps are present [4, 6, 7, 8, 9]. However, it is time-consuming and expensive to construct a large dataset with such labels and so this has limited the use of highly data-driven learning algorithms such as deep neural networks. To take account of this problem, Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2017 has opened a new task [5]: Large-scale weakly supervised sound event detection for smart cars where subtask A is audio tagging and subtask B is sound event detection. Especially for subtask B, the objective is to construct novel SED system based on a dataset without timestamps.

Recent deep learning based SED systems that use timestamps information can be divided into two approaches. One is using the sequence information to predict the order of the timestamps, for example, using Recurrent Neural Networks (RNN) [6, 7]. The other is dividing an audio clip into the same length of small segments (e.g. 1 second long) and using the segments as input for the models, for example, using Deep Neural Networks (DNN) [4] or Convolutional Neural Networks (CNN) [9]. This segmentation-based approach does not use sequence information, but it can capture local audio characteristics well [4, 9]. The effectiveness was shown in many audio tagging tasks [2, 10, 11, 12]. The main difference is that segments in the SED systems have their own labels depending on the presence of events at the moment whereas those in audio tagging systems are annotated with the same labels as long as they are from the same audio file.

For the weakly supervised SED task, we mix the two settings. That is, in training phase, we use the same labels for all segments within an audio file whereas, in test phase, we regard the outputs for segments as separate event predictions. With this setting, we can apply some of the methods developed primarily for audio tagging to the weakly supervised SED task. In the following sections, we describe the methods and show that the approach is effective in our target task.

2. PROPOSED METHOD

2.1. Combination of Multi-Scale Features

Event sounds have different timbre patterns in terms of feature hierarchy and time-scales [13, 11]. For example, bicycle and motorcycle sounds are generated as a repetition of specific sound sources. They tend to be local and repetitive within an audio clip. On the other hand, car and train sounds are more sustained or ambient. They are relatively more global and require longer audio segments to discriminate them. We previously addressed the issue by using multiple CNNs, each of which covers different time scales [11, 14]. The proposed method is performed in three steps: feature learning by multiple CNNs, feature aggregation, and final classification. The CNNs are trained with the sound labels, taking different input sizes to capture both local and global characteristics of the sounds. We then use these trained networks as a feature extractor. Since these feature extractors are trained with different input sizes, these can capture different audio characteristics. After the features are extracted, we summarize them for the given task-specific format. For example, for the audio tagging task, we summarize segment-level features to audio-clip-level by averaging the whole segment features. For the SED task, segment-level features are averaged every second. Lastly, the final prediction is performed using a fully-connected neural network for each subtask.

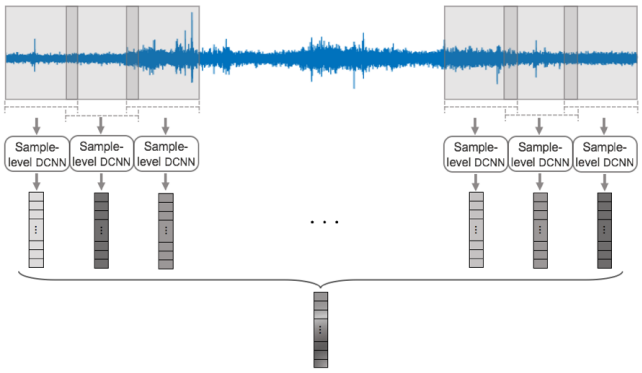

Figure 1: Feature aggregation method for subtask A of Task 4 (audio tagging). The features of models with different input sizes are concatenated.

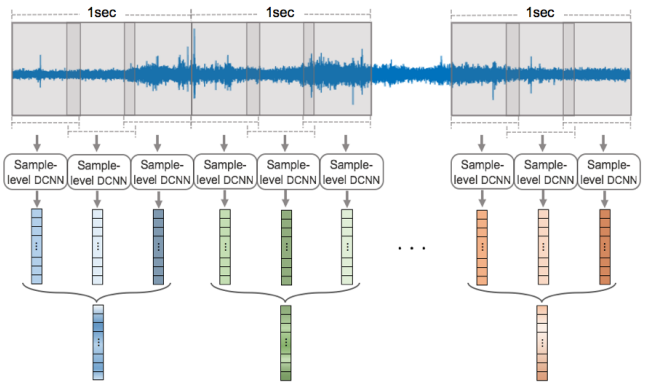

Figure 2: Feature aggregation method for subtask B of Task 4 (sound event detection). The features of models with different input sizes are concatenated.

2.2. Sample-level Deep Convolutional Neural Networks

Using raw audio as input allows the network to learn very low-level features. Generally, in audio classification tasks, raw waveforms are converted to a time-frequency representation before used as input to the system. However, in this preprocessing stage, short-time Fourier transform (STFT) parameters, such as hop size or window size, are often ignored in parameter optimization even though optimal parameters for each sound class may vary [15, 16]. To take account of this and also to avoid exhausting parameter search, we used the previously proposed network as a feature extractor which learns from raw waveforms with very small sample-level filters [14].

3. EXPERIMENTS

3.1. Datasets

The DCASE Challenge 2017 Task4 uses a subset of AudioSet [2]. This subset consists of 17 sound events and the classes are unbalanced and multi-labeled. The task setup comes with training, testing and evaluation set. The split includes 51172, 488, and 1103 audio clips, respectively. Because the evaluation set is saved for the challenge evaluation, we split the training set by randomly selecting 10% of audio clips for each class and using them as a validation set. Since the audio clips are multi-labeled, we in fact selected more than 10% audio clips per class. As a result, the sub-training set consists of 45313 clips and the validation set contains 5859 clips.

3.2. CNN Models

We followed CNN model configuration and training settings in our previous work [15]. For example, all audio clips are segmented according to the network input size and each segment is used as a single sample for training with its corresponding event labels. We used a total of eight CNN models with different lengths of waveforms as inputs from 372ms, 557ms, 627ms, 743ms, 893ms, 1486ms, 2678ms and up to 3543ms. After the networks are trained, they can directly predict the results of subtask A and subtask B. This is termed as Sample level Deep Convolutional Neural Networks (SDCNN) in our experiment.

The difference from the previous work is that the audio sampling rate increased by a factor of 2 (i.e. 44100 Hz) and the model size is expanded accordingly (11 to 16 convolution layers and 128 to 512 filters). Also, since we measure detection performance rather than ranking, we predicted the presence of tags with a threshold value. In the SDCNN model, we used 0.1 for the tagging task and 0.5 for the SED task.

3.3. Feature Aggregation and Final Classification

We also used the trained CNN models as a feature extractor instead of using their prediction results, following the multi-level and multi-scale feature aggregation approach [11]. By multi-level features, we mean to use top three hidden layers in the sample-level CNN models. The purpose of using multi-level features is considering various abstraction levels of the sound tags. Although the tag descriptions are limited to smart cars and so the diversity in feature hierarchy is not strong, we put the multi-level concatenation in our experiment.

For the tagging task, the features of all segments are averaged into a single feature vector for each model as shown in Figure 1. We then combined multi-scaled features and fed them into a fully connected layer for final decision. For the SED task, we summarized the features every second as depicted in Figure 2. If the input length of the CNN model is less than one second, we computed the number of segments by dividing one second by the input size and rounding it up, and overlapped adjacent segments such that all segments fit within one second. We then averaged the features from segments as a single vector. If the input length exceeds one second, we extracted a single feature only. We then move the model by one second for next event detection. Finally, we fed the features into a fully connected layer to make a final decision for each period. We term this setup as Multi-Level Multi-Scale (MLMS) model. In all MLMS models, we used a threshold of 0.2 for tagging predictions and 0.5 to SED predictions to make a final decision.

3.4. List of Submissions

Based on the experimental setup above, we submitted four settings of models or DCASE Challenge 2017 Task 4 (Large-scale weakly supervised sound event detection for smart cars) as follows:

- SDCNN: Sample-level Deep Convolutional Neural Networks that takes 893ms of audio as input. This is one of the models used as a feature extractor for the rest submissions.

Table 1: The class-wise performance of submitted systems and their comparisons on the development set. In the middle section, we show the results with multi-level only (termed as ML) to observe the sensitivity of tag prediction to different input sizes (the numbers after ML). When the performance of a tag has a trend according to the input size, we highlighted the tag and the value of the optimal input size.

Subtask A F-score	SDCNN893	ML372	ML557	ML627	ML743	ML893	ML1486	ML2678	ML3543	MLMS5	MLMS3	MLMS8
Train horn	48.7	22.8	32.4	36.8	26.3	28.5	33.3	36.8	27.0	47.6	41.0	41.0
Air horn, truck horn	43.9	27.7	27.7	27.8	31.5	35.9	22.8	17.1	37.8	35.0	36.8	41.0
Car alarm	27.7	0.0	6.4	0.0	6.4	0.0	0.0	0.0	0.0	6.4	0.0	0.0
Reversing beeps	40.0	6.5	18.1	12.5	23.5	**28.6**	**28.6**	18.2	12.5	18.1	33.3	18.2
Ambulance (siren)	40.0	21.6	24.4	**40.9**	29.2	34.1	15.8	21.6	10.8	50.9	27.9	36.4
Police car (siren)	44.6	38.6	43.6	43.2	44.4	46.3	41.3	44.9	47.1	42.9	46.6	43.9
Fire engine, fire truck (siren)	40.8	43.5	43.4	42.0	44.0	42.4	44.9	42.8	46.9	46.8	40.4	42.2
Civil defense siren	67.4	78.8	77.1	76.7	**80.0**	77.7	77.6	74.6	72.7	77.8	73.2	76.7
Screaming	52.6	40.9	41.6	47.8	48.9	**50.0**	48.9	36.7	44.9	53.1	39.1	48.0
Bicycle	42.5	58.1	58.0	52.3	55.1	48.5	56.1	55.7	44.8	53.1	45.6	61.0
Skateboard	71.4	71.1	70.0	72.4	75.0	**80.0**	72.4	73.3	71.2	77.2	73.7	71.4
Car	23.1	30.7	30.3	32.3	30.7	32.9	32.1	33.5	31.8	32.8	33.8	35.0
Car passing by	19.0	5.7	4.9	12.7	14.6	13.6	23.2	12.8	**23.5**	13.0	10.0	16.6
Bus	29.6	26.1	37.0	38.2	33.3	37.0	31.7	34.5	32.5	34.6	30.0	33.3
Truck	32.9	42.8	40.7	41.1	42.9	41.1	41.8	44.5	43.9	43.0	42.7	40.4
Motorcycle	53.8	**61.0**	54.2	52.8	52.6	46.6	49.1	54.5	46.1	53.3	46.4	57.6
Train	61.2	58.4	62.2	64.4	62.9	64.4	**65.2**	63.7	62.5	67.3	68.8	68.1
Subtask B ER												
Train horn	0.85	0.90	0.93	0.90	0.92	0.85	0.93	0.91	0.90	0.84	0.91	0.87
Air horn, truck horn	0.82	0.86	0.91	0.86	0.90	0.81	0.93	0.88	0.92	0.82	0.89	0.85
Car alarm	0.97	0.94	0.97	0.98	0.96	0.97	0.92	0.98	0.97	0.95	0.96	0.95
Reversing beeps	0.91	0.98	0.92	0.92	0.92	0.92	0.92	0.91	0.90	0.91	0.94	0.88
Ambulance (siren)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Police car (siren)	1.12	1.14	1.09	1.16	1.15	1.04	1.23	1.16	1.13	1.15	1.11	1.09
Fire engine, fire truck (siren)	0.98	0.95	0.99	1.00	0.99	0.98	0.98	1.00	1.00	1.01	0.99	0.97
Civil defense siren	0.58	0.60	0.61	0.62	0.63	0.58	0.63	0.62	0.64	0.62	0.59	0.59
Screaming	0.93	0.95	0.95	0.95	0.95	0.91	0.94	0.94	0.98	0.92	0.91	0.91
Bicycle	0.87	0.87	0.90	0.86	0.85	0.91	0.96	0.96	0.95	0.90	0.88	0.91
Skateboard	0.80	0.78	0.80	0.75	0.78	0.80	0.81	0.83	0.81	0.73	0.79	0.72
Car	3.32	3.29	3.66	2.83	3.42	3.03	3.62	3.29	2.82	3.00	3.00	2.85
Car passing by	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Bus	1.03	1.00	1.01	1.02	1.01	1.00	1.00	1.02	0.99	1.03	1.01	0.98
Truck	0.96	0.90	0.97	0.95	0.97	0.92	0.97	0.95	0.95	0.92	0.98	0.95
Motorcycle	0.92	**0.90**	0.97	0.91	0.91	0.94	0.99	0.91	0.96	0.95	0.91	0.93
Train	0.93	0.92	0.95	0.95	0.95	0.94	0.94	0.90	**0.89**	0.94	0.93	0.92

- MLMS5: Multi-Level and Multi-Scale features extracted from models taking 372ms, 557ms, 627ms, 743ms and 893ms as input.
- MLMS3: Multi-Level and Multi-Scale features extracted from models taking 1486ms, 2678ms, and 3543ms as input.
- MLMS8: Multi-Level and Multi-Scale features extracted from models taking 372ms, 557ms, 627ms, 743ms, 893ms, 1486ms, 2678ms, 3543ms as input.

Details about the models can be found in our DCASE submission webpage link[1].

4. RESULTS AND DISCUSSION

4.1. Evaluation on the Development set

We report the performance of the proposed method in Table 2 (tagging) and Table 3 (SED). From the results, we can find that the feature aggregation and final classification stage improve performance compared to the direct result of SDCNN. Also, as the number of model combinations increases, the performance is generally improved as well.

[1]https://github.com/jongpillee/dcase2017submission

Table 2: Instance-based results of submitted systems for subtask A of Task 4 (audio tagging)

	Development set			Evaluation set		
	F-score	Prec.	Rec.	**F-score**	Prec.	Rec.
SDCNN	37.8%	26.7%	64.8%	40.3%	31.3%	56.7%
MLMS5	44.3%	38.8%	51.7%	47.3%	48.0%	46.6%
MLMS3	42.2%	39.0%	45.9%	47.2%	49.6%	45.0%
MLMS8	43.8%	39.2%	49.5%	47.1%	48.5%	45.9%

Table 3: Instance-based results of submitted systems for subtask B of Task 4 (sound event detection)

	Development set		Evaluation set	
	ER	F-score	**ER**	F-score
SDCNN	0.88	28.1%	0.82	39.4%
MLMS5	0.86	30.7%	0.78	42.6%
MLMS3	0.86	31.2%	0.78	44.2%
MLMS8	0.84	34.2%	0.75	47.1%

We report class-wise performance as well on Table 1. From the class-wise tagging results, we can find the sensitivity of tags to different time scales. For example, tags such as *Reversing beeps*, *Ambulance (siren)*, *Screaming*, *Civil defense siren* and *Skateboard* are optimal around one second. On the other hand, *Bicycle* and *Mo-*

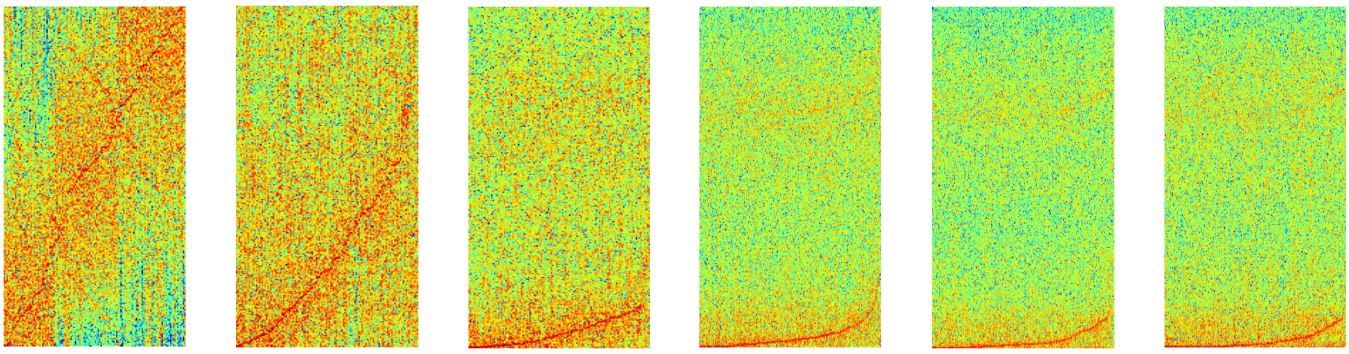

Figure 3: Spectrum of the filters in the sample-level convolution layers which are sorted by the frequency at the peak magnitude. The x-axis represents the index of the filters and the y-axis represents the frequency. The visualization was performed using a gradient ascent method to obtain the input waveform that maximizes the activation of a filter in the layers [15].

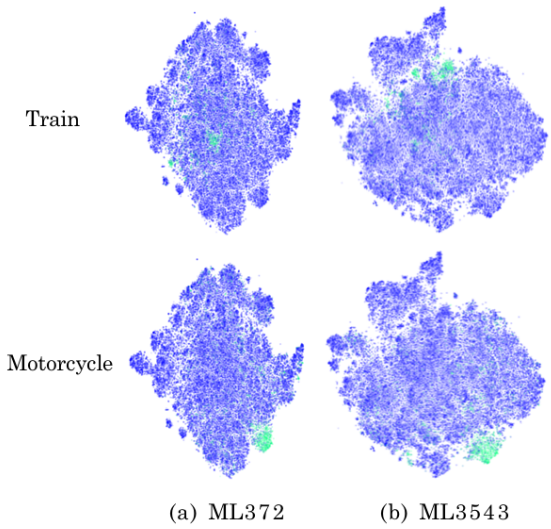

Figure 4: Visualization of aggregated features with *Train* tags and *Motorcycle* tags using t-SNE in the training set. Each dot corresponds to one audio clip. The green dots indicate those belonging to the tag denoted in the left side. ML372 indicates the model with multi-level features and 372ms as input.

torcycle favor shorter seconds, and *Police car (siren)*, *Car passing by*, *Bus* and *Train* prefer longer seconds. These trends can also be observed in Figure 4 where we displayed 2-D embedding space of aggregated features in the ML models using t-Distributed Stochastic Neighbor Embedding (t-SNE). We can see that audio clips with *Train* tags are more closely clustered in ML3543 whereas those with *Motorcycle* tags are more in ML372. This may explain why combining multi-scale features improves the performance. Also, in Table 1, we can find that SCDNN shows good results in class-wise performance. Especially when the sound is alarming ones, for example, *Car alarm*, *Reversing beeps*, *Ambulance (siren)* and *Police car (siren)*. However, the MLMS models achieve better performance on instance-based metrics as shown in Table 2 and 3. This is probably because about half of the dataset have car tags and the MLMS models tend to improve the performance for those with the car tags significantly.

4.2. Comparison with other submissions in DCASE 2017

Nine teams submitted their algorithms to subtask A and seven teams to subtask B in the DCASE2017 Task 4. Our team was ranked at the 5th for subtask A and at the 3rd for subtask B. Most submitted algorithms used mel-scaled spectrogram as input and CNN as a classifier. These results show that our model using raw waveform as input can be comparable those using spectrogram.

4.3. Filter Visualization of SDCNN

We visualize learned filters on each layer in the sample-level CNN. Figure 3 shows the filers obtained from a gradient ascent method [15] and sorted with the frequency at the peak magnitude. We can observe that they are sensitive to more log-scaled in frequency as the layer goes up. Compared to the learned filters from music audio 3, these filters tend to have more low-frequency concentration and less complex patterns.

5. CONCLUSIONS

In this paper, we presented sample-level DCNN models using raw waveforms and multi-scale feature aggregation method developed for the DCASE Challenge 2017. We showed that our proposed method is comparable to CNN-based models using spectrogram as input. Class-wise performance and feature visualization indicate that audio clips with different tags are optimal in different time scales. Combining the multi-scaled features improves overall performance. We also visualized hierarchically learned filters in the sample-level CNN. They showed the spectral patterns are adapted to the characteristic of the acoustic scene sounds.

6. ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (1711055381, Development of Human Enhancement Technology for auditory and muscle support).

7. REFERENCES

[1] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 1128–1132.

[2] J. F. Gemmeke, D. P. W. 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 *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[3] T. Heittola, A. Mesaros, and T. Virtanen, "DCASE2016 baseline system," DCASE2016 Challenge, Tech. Rep., September 2016.

[4] Q. Kong, I. Sobieraj, M. Plumbley, and W. Wang, "Deep neural network baseline for DCASE challenge 2016," DCASE2016 Challenge, Tech. Rep., September 2016.

[5] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE2017 challenge setup: Tasks, datasets and baseline system," *In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017). November 2017.*

[6] S. Adavanne, G. Parascandolo, P. Pertil, T. Heittola, and T. Virtanen, "Sound event detection in multichannel audio using spatial and harmonic features," DCASE2016 Challenge, Tech. Rep., September 2016.

[7] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, June 2017.

[8] B. Elizalde, A. Kumar, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, "Experiments on the dcase challenge 2016: Acoustic scene classification and sound event detection in real life recording," *arXiv preprint arXiv:1607.06706*, 2016.

[9] A. Gorin, N. Makhazhanov, and N. Shmyrev, "DCASE2016 sound event detection system based on convolutional neural network," DCASE2016 Challenge, Tech. Rep., September 2016.

[10] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[11] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using pre-trained convolutional neural networks for music auto-tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, 2017.

[12] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6964–6968.

[13] Y. Xu, Q. Huang, W. Wang, and M. D. Plumbley, "Hierarchical learning for dnn-based acoustic scene classification," *arXiv preprint arXiv:1607.03682*, 2016.

[14] J. Lee and J. Nam, "Multi-level and multi-scale feature aggregation using sample-level deep convolutional neural networks for music classification," *International Conference on Machine Learning (ICML), Machine Learning for Music Discovery Workshop*, 2017.

[15] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," *Sound and Music Computing Conference (SMC)*, pp. 220–226, 2017.

[16] K. Choi, D. Joo, and J. Kim, "Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras," *arXiv preprint arXiv:1706.05781*, 2017.