A REPORT ON AUDIO TAGGING WITH DEEPER CNN, 1D-CONVNET AND 2D-CONVNET

Technical Report

Qingkai WEI, Yanfang LIU, Xiaohui RUAN

Beijing Kuaiyu Electronics Co., Ltd., Beijing, PRC.
weiqingkai1@163.com, {wqk, liuyf, rxh}@kuaiyu.com

ABSTRACT

General-purpose audio tagging is a newly proposed task in DCASE 2018, which can provide insight towards broadly-applicable sound event classifiers. In this paper, two systems (named as 1D-ConvNet and 2D-ConvNet in this paper) with small kernel sizes, multiple functional modules, deeper CNN (convolutional neural networks) are developed to improve performance in this task. Different audio features are used: raw waveforms are for 1D-ConvNet; frequency domain features such as mfcc, log-mel spectrogram, multi-resolution log-mel spectrogram and spectrogram, are compared as the 2D-ConvNet input. Using DCASE 2018 Challenge task 2 dataset to train and evaluate, the best single model with 1D-ConvNet and 2D-ConvNet are chosen, whose kaggle public leaderboard score are 0.877 and 0.961 respectively. In addition, a better ensemble rank averaging prediction get a score 0.968 on the public leaderboard, ranking 5/556.

Index Terms— DCASE 2018, Audio tagging, Convolutional neural networks, 1D-ConvNet, 2D-ConvNet, Model ensembling

1. INTRODUCTION

In recent years, computer vision techniques such as object detection and segment, are applied in monitoring, surveillance and autonomous driving. In the process of these techniques' performance improved from laboratory to applications, development of neural network architectures played an important role. Along with the appearance of LeNet [1], Alexnet [2], VGG Net[3], GoogLeNet (Inception V1 and following V3, V4) [4, 5, 6], Deep Residual Net [7], Squeeze-and excitation networks [8], neural networks become much deeper, together with the ingenious modules such an inception modules, factorizing convolutions, residual blocks and so on.

Similar to vision, audio also takes lots of unique information, which can help people recognize their surroundings together with vision or tactile information. However, corresponding techniques such as sound event detection and specific sound extractions have not been brought to general applications.

Sound event detection is a system to automatically detect and classify emergency sound events. In 1st DCASE challenge (DCASE 2013, IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events), sound event detection was firstly focused together with audio scene classification [9]. Then in DCASE 2016 challenge, audio tagging was introduced as a new task. Audio tagging aims at putting one or several sound events tags on a sound clip, like "domestic", "musical instruments", "animals", "human sounds", "speech". This task can provide insight to broadly-applicable sound event classifiers, with increasing amount of sound event categories. And it can be applied in areas such as audio surveillance [10], information retrieval [11], automatic description of multimedia.

Since 2013, the algorithms on audio tagging and sound event detection have been mainly shifted from traditional classifier approaches (mfcc-gmm, HMM: hidden Markov model, NMF: nonnegative matrix factorization, random forests) [9, 12] to deep learning methods such as DNN [13, 14, 15], CNN [16, 17], RNN [18].

As to audio features, many frequency domain features such as mfcc (mel-frequency cepstrum coefficients) [15], mel-spectrogram [13] and spectrogram [19] have been used in similar tasks. Moreover, raw waveform is also applied as the input to classifiers in some recent work about acoustic scene recognition and speech recognition [19, 20, 21].

In this audio tagging task, inspired by the process of neural network evolutions in computer vision, we applied two deeper convolutional neural networks (1D-ConvNet with raw waveforms as input, 2D-ConvNet with frequency domain features as input) to improve the performance. Several techniques work well in computer vision are applied effectively in this audio tagging task:

- The neural network architectures are much deeper (1D-ConvNet 18 layers, 2D-ConvNet 32 layers), with inception modules, factorizing convolutions, residual blocks applied, which lead to much better performance;
- For 2D-ConvNet, different frequency domain audio features are compared with the same model preliminarily, including mfcc, log-mel spectrogram, multi-resolution log-mel spectrogram and spectrogram;
- Data augmentation methods such as mixup, random erase are used, which are effective to overcome overfitting;
- Model ensembling techniques are used, predictions of 1D-ConvNet and 2D-ConvNet are combined with rank averaging method. More model ensembling techniques like stacking should be test in future.

However, due to limit of challenge time, architectures and parameters of these two neural networks are not fine tuned enough, which would be our further work. In this paper, these two neural networks used to get task 2 challenge results are briefly introduced. The rest of this paper is organized as follows. Section 2 describes the features, data augmentation methods, architectures and parameters of these two neural networks. Section 3 shows the experiment setup and performances with DCASE 2018 task2 dataset. Submissions and conclusions are presented in Section 4.

http://www.kuaiyu.com/en, the leading enterprise in audio security monitoring industry in China.

2. METHODS

The architectures of two neural networks, 1D-ConvNet and 2D-ConvNet, are shown in Table. 1 and 2. For 1D-ConvNet, raw waveforms with normalization are set as input directly. While for 2D-ConvNet, the features including mfcc, log-mel spectrogram, multi-resolution log-mel spectrogram, spectrogram are extracted from raw waveforms. The output of the neural network is the probabilities of 41 classes, between 0 and 1, with sum as 1. Details about feature extraction, data augmentation and neural networks are described then.

2.1. Features and data augmentation

For 1D-ConvNet, the raw time-domain waveforms are directly used as input at 44100 Hz. The data length of train and test samples are range from 300 ms to 30 s. To get input for 1D-ConvNet, waveforms of a few seconds are randomly (with random offset) extracted from the raw waveforms. The length of extracted waveforms are set as 2s, 3s, 4s, 5s, to compare the performances in this task. It should be noticed that longer extracted waveforms can lead to much more computationally expensive.

For 2D-ConvNet, we study the performances of different frequency domain features. The features we selected are mfcc, log-mel spectrogram, multi-resolution log-mel spectrogram [22] and spectrogram. The basic parameters are same: sample frequency 44100 Hz, window size 2048 samples (46.44 ms), hop size 512 samples (11.61 ms), pre-fft Hamming window. As it demonstrated above, same length of waveforms are randomly extracted from raw data firstly, transform to T frames. For the above four different features, other different parameters are list below:

mfcc:

Number of mfccs 40, feature size $T \times 40$.

log-mel spectrogram:

Number of mel filters 128, feature size $T \times 128$.

multi-resolution log-mel spectrogram:

It's concluded that log mel-band energy extracted in multi-resolution windows give considerable improvement [22]. We wish to examine its effect with deeper CNN, so the window sizes are 2048, 8192 and 16384 samples, with feature size $T \times (128 * 3)$ shown as Fig. 1

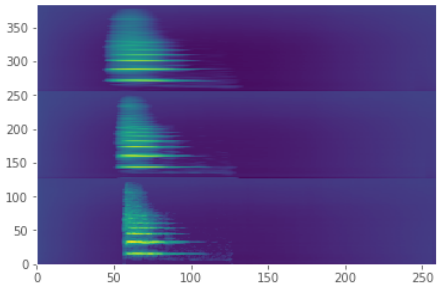


Figure 1: Example of multi-resolution log-mel spectrogram feature.

spectrogram:

Number of frequency bins is 513, feature size $T \times 513$.

Data augmentation methods such as mixup [23] and random erasing [24], are applied to the frequency domain features, which help eliminating overfitting effectively. Preprocessing methods like silence trim are also examined, which did not show improvement of performance.

2.2. Neural networks

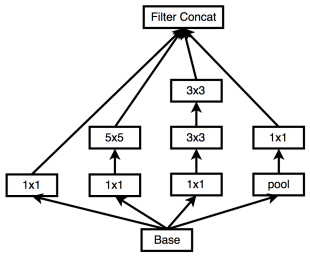
1D-ConvNet (parameters: 2,099,801)
Input: 44100·t 1D time-domain waveform
conv, kernel 80, stride 4, 48
max pool, 4, stride 4
$\begin{bmatrix} \text{conv1d, kernel 3, stride 1, 48} \\ \text{conv1d, kernel 3, stride 1, 48} \end{bmatrix} \times 2$
max pool, 4, stride 4
$\begin{bmatrix} \text{conv1d, kernel 3, stride 1, 96} \\ \text{conv1d, kernel 3, stride 1, 96} \end{bmatrix} \times 2$
max pool, 4, stride 4
$\begin{bmatrix} \text{conv1d, kernel 3, stride 1, 192} \\ \text{conv1d, kernel 3, stride 1, 192} \end{bmatrix} \times 2$
max pool, 4, stride 4
$\begin{bmatrix} \text{conv1d, kernel 3, stride 1, 384} \\ \text{conv1d, kernel 3, stride 1, 384} \end{bmatrix} \times 2$
Global average pooling (output: 41)
Softmax

Table 1: Architectures of 1D-ConvNet with time-domain waveform inputs [21]. $[...] \times k$ denotes the k stacked layers. Double layers in a bracket denotes a residual block [7]. Convolutional layers are followed with BN and ReLU, which are not shown in the table.

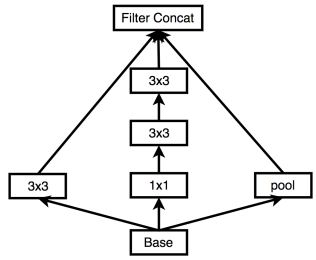
1D-ConvNet takes time-domain waveforms as input, which are represented as a long 1D vector. The neural network is same as that in the paper [21], details are shown as Table. 1. For t seconds long waveforms, the input layer is a 44100·t 1D vector. To build this deep CNN, small kernel sizes are used for convolutional layers. Basic modules like batch normalization, rectified linear units are applied following each convolutional layer. Network depth is very importance to get better accuracy. However, with the depth of network increasing, accuracy can get saturated and degrade. To construct effective deeper network, residual blocks can help a lot [7]. In 1D-ConvNet, two convolutional layers in a bracket denotes a residual block.

2D-ConvNet (parameters: 7,664,969)
Input: 299 × 299 × 3 frequency-domain features
conv2d, kernel 3 × 3, stride 2, 32
conv2d, kernel 3 × 3, stride 1, 32
conv2d, kernel 3 × 3, stride 1, 64
max pool, 3, stride 2
[inception block A as Fig. 2(a)] × 3
[inception block B as Fig. 2(b)] × 1
[inception block C as Fig. 2(c)] × 3
Global average pooling
Dense 1024 (output: 41)
Softmax

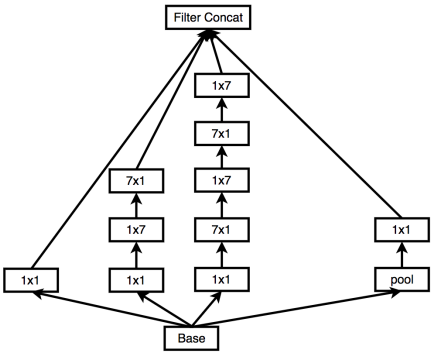
Table 2: Architectures of 2D-ConvNet network for frequency-domain features. $[...] \times k$ denotes the k stacked layers. Details of inception blocks can be seen in Fig. 2. Convolutional layers are followed with BN and ReLU, which are not shown in the table.



(a) Inception block A.



(b) Inception block B.



(c) Inception block C.

Figure 2: Details of inception blocks.

For 2D-ConvNet, frequency domain audio features are used as input. As Sec. 2.1 described, the features' size can be $T \times 40$, $T \times 128$, $T \times (128 * 3)$ or $T \times 129$. T is set as 299, about 3.5 s. To match the neural network, features are resized to the shape $(299, 299, 3)$. The input size is 3 channel now, because when we accidentally use 3 channel input, the score increase a lot than that of 1 channel input. As it concluded in [5], inception modules can widen the network with multiple sizes of kernel in the same layer and factorizing convolutions decrease parameters a lot. They are applied in 2D-ConvNet, played an important role in the improvement of performance. Details are shown in Table. 2.

3. SETUP AND PERFORMANCE EVALUATION

3.1. Dataset and evaluation metric

DCASE 2018 task 2 dataset is used to train and evaluate above two neural networks. The audio samples are from Freesound [25, 26] annotated using a vocabulary of 41 labels from Googles AudioSet Ontology. Categories of sound event include musical instruments, human sounds, domestic sounds, animals, which can be seen from task 2 webpage [27].

As it describes on the webpage, recording scenarios and techniques can be very different as sounds are uploaded by users all around the world, thought all the samples are provided as uncompressed PCM 16 bit, 44.1 kHz, mono audio. The labeling of the samples is a mapping from Freesound tags to AudioSet Ontology categories, which may not so match with the content of samples. So, a data validation process was carried out with people listening to the samples and manually verifying the tags [28].

Instruments	Instruments Continued	Human	Actions	Other
Hi-hat	Bass_drum	Laughter	Knock	Gunshot_or_gunfire
Saxophone	Harmonica	Finger_snapping	Drawer_open_or_close	Bus
Trumpet	Gong	Fart	Computer_keyboard	Telephone
Glockenspiel	Double_bass	Cough	Tearing	Squeak
Cello	Tambourine	Applause	Shatter	Scissors
Clarinet	Cowbell	Burping_or_eructation	Keys_jangling	Microwave_oven
Snare_drum	Electric_piano		Writing	Bark
Oboe	Acoustic_guitar			Meow
Flute	Voilin_or_fiddle			Fireworks
Chime				

Figure 3: Categories of the samples.

The train set includes 9473 samples among 41 categories listed in Fig. 3, while the number of audio samples per category ranges from 94 to 300. The duration of samples ranges from 300ms to 30s, while the distribution of durations ranges from 200ms to 30s, length of 4151 samples is less than 5s. 3710 of 9473 annotations of samples is manually verified while the others are not. The test set includes 9400 samples, with about 1.6k manually-verified annotations with a similar category distribution, while the other about 7.8k as padding annotations. These 1.6k samples are used for evaluating the system.

We tried to do manually-verify to the rest of train set, and used verified labels to train 2D-ConvNet. However, the leaderboard score decrease from above 0.95 to below 0.7, which shows such a bad performance. So in the final submissions, the origin train labels are used.

When doing the manually verify, we found several tips that make this task difficult:

- Some categories are really hard to classify even by people, for example (Chime, Cowbell, Glockenspiel) or (Flute, Clarinet);
- With below 300 samples, if the specific category can be fully representative, e.g. most samples of 'Laugher' is a 'evil' type in train set;
- Some samples can be with multi-tag, e.g. with one tag at about 1 s and the other at about 2 s.

To evaluate each developed system, submissions should be uploaded to kaggle platform and will be evaluated with the Mean Average Precision @ 3 metric. The kaggle platform will give a public leaderboard score with approximately 19% of the test data (about 300 samples). The final results will be based on the rest 81%. We worry about that if public and private test data are independent identically distributed, while public test data have about 300 samples of 41 categories.

3.2. Baseline

The baseline method is provided on webpage [27], gives a sense of performance possible with the above dataset. The baseline system implements a CNN classifier, with frames of log-mel spectrogram as input features. The window fft size is 25 ms and hop size is 10 ms, with 64 mel bins, while the feature window size is 0.25 s. The feature size is (25, 64, 1), following with three convolutional and pooling layers. Adam optimizer is used to train the model, with a learning rate 1e-4, the batch size is 64. Details of the parameters and neural network architect can be found on webpage. The kaggle leaderboard score can be 0.704 with 5 epochs, while we trained for more epochs, it can reach 0.798.

3.3. Parameter setup

With lots of tests, the parameters of training with 1D-ConvNet and 2D-ConvNet are set as below.

For 1D-ConvNet, the loss function is a binary cross entropy with predicted values (0∼1) and correct values (0 or 1). Adam is used as optimizer and the size of a mini-batch is set to 128. The learning rate is initially set as 1e-3. It decays when the validation accuracy does not increase for last 3 epochs with decay factor 0.5 while the minimum learning rate is 1e-6. Training is stopped early when a validation accuracy has stopped increasing for 10 epochs. The model weights with highest validation accuracy will be saved for following predictions. For the single model, 5-fold cross validation is used to tune the parameters. 5 prediction files for test set are generated and used to do model ensembling.

For 2D-ConvNet, the loss function is same as above. Adam is used as optimizer and the size of a mini-batch is set to 16. The learning rate is initially set as 1e-3. It decays when the validation accuracy does not increase for last 4 epochs with decay factor 0.5 while the minimum learning rate is 1e-6. Training is stopped early when a validation accuracy has stopped increasing for 24 epochs. The model weights with hight validation accuracy will be stored for following predictions. For the single model, 7-fold cross validation is used to tune the parameters. 7 prediction files for test set are generated and used to do model ensembling.

3.4. Results and discussion

For 1D-ConvNet, 2 s, 3 s, 4 s, 5 s length of waveforms are extracted randomly as input. The validation accuracy (average of 5-fold CV), leaderboard score and early stopping epoch numbers are listed in Table 3. As the Table shows, length of input affects little while longer waveforms as train input lead to bit better performances. For the final model ensembling, ensemble predictions with waveforms of 3 s get a higher score.

data length	val acc	LB score	stopping epoch
2s	0.7031	0.870	58
3s	0.7142	0.873	83
4s	0.7205	0.869	59
5s	0.7252	0.877	72

Table 3: Results of 1D-ConvNet with different time length input.

Different audio features are compared preliminarily with the same neural network, 2D-ConvNet. As shown in Table. 4, model trained with log-mel spectrogram and multi-resolution log-mel spectrogram can get higher validation accuracy. So we use log-mel spectrogram as features for the final model ensembling. The highest public leaderboard score attained by 2D-ConvNet with log-mel spectrogram is 0.961.

feature	val acc
mfcc	0.7834
log-mel spectrogram	0.8662
multi-resolution log-mel spectrogram	0.8647
spectrogram	0.7878

Table 4: Results with 2D-ConvNet with different audio features.

Model ensembling is a very effective technique to increase accuracy on machine learning tasks. A good ensemble contains high performing models which are less correlated. Model ensembling methods include rank ensemble techniques, bagging, boosting and stacking techniques. Ranking averaging method is used here, with predictions of 1D-ConvNet and 2D-ConvNet combined with different weighs. The best public leaderboard score is 0.968. Details of two submissions for challenge are introduced in following Sec. 4.

4. SUBMISSIONS AND CONCLUSION

In DCASE 2018 task 2, we submitted 2 predictions based on above frameworks.

- For submission 1, we took the ensemble of 5 predictions (higher validation accuracy) from 7 folds CV with 2D-ConvNet and 3 predictions (higher validation accuracy) from 5 folds CV with 1D-ConvNet, with weights 2:2:2:2:2:1:1:1. The public leaderboard score is 0.968.
- For submission 2, we took the ensemble of 5 predictions (higher validation accuracy) from 7 folds CV with 2D-ConvNet and 3 predictions (higher validation accuracy) from 5 folds CV with 1D-ConvNet, with weights 3:3:2:2:2:1:1:1. The public leaderboard score is 0.967.

In this paper, inspired by the neural network evolutions in computer vision, we apply two deeper CNN in the DCASE2018 task 2 - audio tagging. Though these two neural networks (1D-ConvNet and 2D-ConvNet) are not fine tuned enough till now, they showed competitive potential in this field. For 2D-ConvNet, with the same neural networks, log-mel spectrogram performs better as the input. Data augmentation like mixup, random erase are effective to overcome overfitting in this task. An easy model ensembling technique, rank averaging is used, which improved the leaderboard score slightly from 0.961 to 0.968.

Next step, more fine tuning and model ensembling technique like stacking should be applied to get better performance, which can improve the performance and take these sound techniques to applications.

5. ACKNOWLEDGMENT

Thanks to Daisuke Niizumi, who shared lots of useful tips on Kaggle. Thanks to Zafarullah Mahmood, who gave a concise and effective kernel as a framework of this task.

6. REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, vol. 7, 2017.

[9] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[10] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE, 2007, pp. 21–26.

[11] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE multimedia*, vol. 3, no. 3, pp. 27–36, 1996.

[12] H. Phan, L. Hertel, M. Maass, P. Koch, and A. Mertins, "Car-forest: Joint classification-regression decision forests for overlapping audio event detection," *arXiv preprint arXiv:1607.02306*, 2016.

[13] Q. Kong, I. Sobieraj, W. Wang, and M. Plumbley, "Deep neural network baseline for dcase challenge 2016," *Proceedings of DCASE 2016*, 2016.

[14] I. Choi, K. Kwon, S. H. Bae, and N. S. Kim, "Dnn-based sound event detection with exemplar-based approach for noise reduction," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, 2016, pp. 16–19.

[15] Y. Xu, Q. Huang, W. Wang, P. J. Jackson, and M. D. Plumbley, "Fully dnn-based multi-label regression for audio tagging," *arXiv preprint arXiv:1606.07695*, 2016.

[16] T. Lidy and A. Schindler, "Cqt-based convolutional neural networks for audio scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, vol. 90. DCASE2016 Challenge, 2016, pp. 1032–1048.

[17] E. Cakır, T. Heittola, and T. Virtanen, "Domestic audio tagging with convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2016)*, 2016.

[18] T. H. Vu and J.-C. Wang, "Acoustic scene and event recognition using recurrent neural networks," *Detection and Classification of Acoustic Scenes and Events*, vol. 2016, 2016.

[19] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 3461–3466.

[20] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[21] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 421–425.

[22] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," *arXiv preprint arXiv:1710.02997*, 2017.

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

[24] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[25] https://freesound.org/.

[26] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: a platform for the creation of open audio datasets," in *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[27] http://dcase.community/challenge2018/task-general-purpose-audio-tagging.

[28] J. F. Gemmeke, D. P. 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 *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 776–780.