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

Qingkai WEI, Yanfang LIU, Xiaohui RUAN

Beijing Kuaiyu Electronics Co., Ltd., Beijing, PRC.
{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. In detail, different audio features are used, i.e. raw waveforms are for 1D-ConvNet, while frequency domain features, such as mfcc, log-mel spectrogram, multi-resolution log-mel spectrogram and spectrogram, are utilized 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 0.967 on the public leaderboard, ranking 5/558, while score 0.942 on the private leaderboard ranking 11/558.

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

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 extraction 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].

However, there is no universally accepted conclusion about which neural network and audio feature are best. 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 which 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, 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 ensemble techniques are used, predictions of 1D-ConvNet and 2D-ConvNet are combined with rank averaging method. More model ensemble techniques like stacking should be tested in future;
- Training and validation based on DCASE 2018 task 2 dataset verify the effectiveness of the proposed methods.

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 original 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 would lead to more computationally expensive resources.

For 2D-ConvNet, we study the performances of different frequency domain features. The features 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) with pre-fft Hamming window. As it demonstrated above, same length of waveforms are randomly extracted from raw data firstly, transformed to T frames. For four different features, other 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

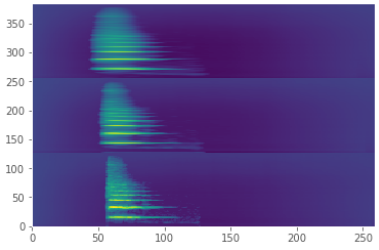


Figure 1: Example of multi-resolution log-mel spectrogram feature.

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
conv1d, 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

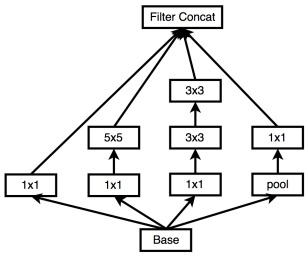
Table 1: Architectures of 1D-ConvNet with time-domain waveform inputs [21]. $[...] \times k$ denotes the k stacked layers. Double layers in a bracket denotes a residual block [7]. Convolutional layers are followed with BN and ReLU, which are not shown in the table.

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 important 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.

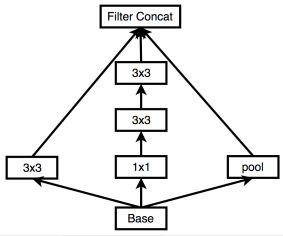
2D-ConvNet (parameters: 7,664,969)
Input: $299 \times 299 \times 3$ frequency-domain features
conv2d, kernel 3×3, stride 2, 32
conv2d, kernel 3×3, stride 1, 32
conv2d, kernel 3×3, stride 1, 64
max pool, 3, stride 2
[inception block A as Fig. 2(a)] $\times 3$
[inception block B as Fig. 2(b)] $\times 1$
[inception block C as Fig. 2(c)] $\times 3$
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.

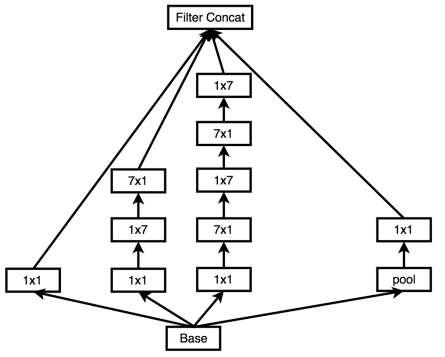
For 2D-ConvNet, frequency domain audio features are used as input. As Sec. 2.1 described, the features' size can be $T \times 40$, $T \times 128$, $T \times (128 * 3)$ or $T \times 513$. Here, T is set as 299, about



(a) Inception block A.



(b) Inception block B.



(c) Inception block C.

Figure 2: Details of inception blocks.

3.5 s. To match the neural network, features are resized to the shape $(299, 299, 3)$. The input size is 3 channel, when we accidentally use 3 channel input, the score increased a lot than that of 1 channel input. The may because 3 channel neural network has better ability to represent. 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. Categories of sound event include musical instruments, human sounds, domestic sounds, animals. Recording scenarios and techniques can be very different as sounds are uploaded by users all around the world. 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 [25, 26]. The train set includes 9473 samples while the number of audio samples per category ranges from 94 to 300. 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, 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 score decrease from above 0.95 to below 0.70, which shows a much worse performance. So for the final submissions, the original training 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, some categories can be fully representative, e.g. most samples of 'Laugher' is a 'evil' type in train set;
- Some samples can be with multi-label.

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

3.2. Baseline

The baseline method is provided, giving 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. The feature size is (25, 64, 1), following with three convolutional and pooling layers. Details can be found in the paper [26]. 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

The parameters of training with 1D-ConvNet and 2D-ConvNet are set as below.

For 1D-ConvNet, the loss function is a categorical cross entropy with predicted values (0$\sim$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 factor 0.5, while the minimum learning rate is 1e-6. Training is stopped early when 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 ensemble.

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 a 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 ensemble.

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), 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 ensemble, ensemble predictions with waveforms of 3 s get a higher score.

data length	val acc	LB score	stopping epoch
2 s	0.7031	0.870	58
3 s	0.7142	0.873	83
4 s	0.7205	0.869	59
5 s	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 get higher validation accuracy. So we use log-mel spectrogram as features for the final model ensemble. The highest public leaderboard score attained by 2D-ConvNet with log-mel spectrogram is 0.961, whose private score is 0.938.

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

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

Model ensemble is a very effective technique to increase accuracy on machine learning tasks. A good ensemble contains high performing models which are less correlated. Model ensemble methods include rank ensemble, bagging, boosting and stacking. Ranking averaging is used with predictions of 1D-ConvNet and 2D-ConvNet combined with different weighs. For our submissions, the best private leaderboard score is 0.942, while the public is 0.967, score on the whole test set is 0.947. Those scores are attained by submission 2 to challenge, details of ensemble are shown below.

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

With the newly released groundtruth of test set, per-class score on the whole, private and public test set can be attained as Table. 5. The accuracy for 'Scissors', 'Telephone' and 'Squeak' are too low, while overfitting is obvious for the 301 samples of public test set.

category	score(whole)	score(private)	score(public)
Oboe	0.9881	0.9853	1
Bass_drum	1	1	1
Saxophone	0.9803	0.9754	1
Chime	0.8736	0.8472	1
Electric_piano	0.9063	0.8846	1
Shatter	0.9828	0.9792	1
Bark	0.9821	0.9783	1
Acoustic_guitar	0.9667	0.9583	1
Scissors	0.7667	0.7083	1
Double_bass	1	1	1
Knock	0.9872	0.9844	1
Telephone	0.7674	0.7564	0.8148
Violin_or_fiddle	1	1	1
Gunshot_or_gunfire	0.9259	0.9379	0.8750
Burping_or_eructation	1	1	1
Clarinet	1	1	1
Computer_keyboard	0.9808	0.9762	1
Flute	0.9727	0.9659	1
Cello	0.9815	0.9773	1
Tambourine	0.9833	1	0.9167
Drawer_open_or_close	0.8793	0.8542	1
Snare_drum	1	1	1
Fart	1	1	1
Meow	0.9828	0.9792	1
Trumpet	0.9324	0.9500	0.8571
Fireworks	0.8698	0.8782	0.8333
Bus	0.8400	0.8000	1
Keys_jangling	0.9107	0.8913	1
Applause	1	1	1
Harmonica	0.9091	0.9074	0.9167
Cough	1	1	1
Gong	0.9730	0.9667	1
Glockenspiel	0.9023	0.8819	1
Tearing	0.9630	0.9545	1
Writing	0.8621	0.8542	0.9000
Squeak	0.5230	0.5069	0.6000
Microwave_oven	0.9310	0.9167	1
Laughter	0.9737	0.9677	1
Finger_snapping	1	1	1
Hi_hat	0.9829	1	0.9048
Cowbell	1	1	1

Table 5: Per-class scores on whole, private and public test set.

4. CONCLUSION

In this paper, inspired by the neural network evolutions in computer vision, we apply two deeper CNN in the DCASE 2018 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 ensemble technique, rank averaging is used, which improved the leaderboard score slightly. More fine tuning and model ensemble techniques 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] 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.

[26] E. Fonseca, M. Plakal, F. Font, D. P. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," *arXiv preprint arXiv:1807.09902*, 2018.