Deep Convolutional Neural Network with Mixup for Environmental Sound Classification

Zhichao Zhang, Shugong Xu(✉), Shan Cao, and Shunqing Zhang

Shanghai Institute for Advanced Communication and Data Science,
Shanghai University, Shanghai 200444, China
{zhichaozhang,shugong,cshan,shunqing}@shu.edu.cn

Abstract. Environmental sound classification (ESC) is an important and challenging problem. In contrast to speech, sound events have noise-like nature and may be produced by a wide variety of sources. In this paper, we propose to use a novel deep convolutional neural network for ESC tasks. Our network architecture uses stacked convolutional and pooling layers to extract high-level feature representations from spectrogram-like features. Furthermore, we apply mixup to ESC tasks and explore its impacts on classification performance and feature distribution. Experiments were conducted on UrbanSound8K, ESC-50 and ESC-10 datasets. Our experimental results demonstrated that our ESC system has achieved the state-of-the-art performance (83.7%) on UrbanSound8K and competitive performance on ESC-50 and ESC-10.

Keywords: Environmental sound classification
Convolutional neural network · Mixup

1 Introduction

Sound recognition is a front and center topic in today's pattern recognition theories, which covers a rich variety of fields. Some of sound recognition topics have made remarkable research progress, such as automatic speech recognition (ASR) [9,10] and music information retrieval (MIR) [4,31]. Environmental sound classification (ESC) is an another important branch of sound recognition and is widely applied in surveillance [21], home automation [33], scene analysis [3] and machine hearing [14]. However, unlike speech and music, sound events are more diverse with a wide range of frequencies and often less well defined, which make ESC tasks more difficult than ASR and MIR. Hence, ESC still faces critical design issues in performance and accuracy improvement.

Traditional ASR techniques such as MFCC, LPC, PLP are applied directly to ESC fields in previous works [7,13,16,28]. However, state-of-the-art performance has been achieved when using more discriminative representations such as Mel filterbank features [5], Gammatone features [34] and wavelet-based features [8].

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11257, pp. 356–367, 2018.
https://doi.org/10.1007/978-3-030-03335-4_31

These features were modeled with some typical machine learning algorithms such as SVM [32], GMM [17] and KNN [20] for ESC tasks. However, the performance gain introduced by these approaches is still unsatisfying. One main reason is that traditional classifiers do not have feature extraction ability.

Over the past few years, deep neural networks (DNNs) have made great success in ASR and MIR [10,25]. For audio signals, DNNs have ability to extract features from raw data or hand-draft feature. Therefore, some DNN-based ESC systems [12,15] were proposed and performed much better than SVM-based ESC system. However, deep fully-connected architecture of DNNs is not robust for transformative features [22]. Some new researchs find convolutional neural networks (CNNs) have strong abilities to explore inherit and hidden patterns through huge amount of training data. Several attempts that apply CNN to ESC have received performance boosts by learning spectrogram-like features from environment sounds [19,23,35]. However, the existing networks for ESC mostly use shallow architecture, such as 2 convolutional layers [19,35] and 3 convolutional layers [23]. Getting a more discriminative and powerful information usually requests a deeper model. Therefore in this paper, we propose an enhanced CNN architecture with a deeper network based on VGG Net [26]. The main contributions of this paper includes

- We propose a novel CNN network based on VGG Net. We find that simply using stacked convolutional layers with 3×3 convolution filters is unsatisfying in our tasks. So we redesign a novel CNN architecture in our ESC system. Instead of 3×3 convolution filters, We use 1-D convolution filters to learn local patterns across frequency and time, respectively. And our method performs better than CNN using 3×3 convolution filters with same depth of network.
- Mixup is applied in our ESC system for ESC tasks. Every training sample is created by mixing two examples randomly selected from original training dataset when using mixup. And the training target is also changed to the mix ratio. The effectiveness of mixup on classification performance and feature distribution is then explored further.
- Experiments were conducted on UrbanSound8K, ESC-50 and ESC-10 datasets, the result of which demonstrated that our ESC system has achieved the state-of-the-art performance (83.7%) on UrbanSound8K and competitive performance on ESC-50 and ESC-10.

The rest of this paper is organized as follows. Recent related works of ESC are introduced in Sect. 2. Section 3 provides detailed introduction of our methods. Section 4 presents the experiments settings on ESC-10, ESC-50 and UrbanSound8K datasets, and Sect. 5 gives both experimental results and detailed discussions of our results. Finally, Sect. 6 concludes the paper.

2 Related Work

In this section, we introduce the recent deep learning methods for environmental sound classification. Piczak [20] proposed to apply CNNs to the log mel spec-

trogram which is calculated for each frame of audio and represents the squared magnitude of each frequency area. Piczak created a two-channel feature by applying log mel spectrogram and its delta information as the input of his CNN model and gave a 20.5% improvement over Random Forest method on ESC-50 dataset. Takahashi et al. [27] also used log mel spectrogram and their delta and delta-delta information as a three-channel input in a manner similar to the RGB inputs of the image. Agrawal et al. [1] used gammatone spectrogram and a similar CNN architecture as Piczak [19] and claimed that they achieved 79.1% and 85.34% accuracy on ESC-50 and UrbanSound8K dataset, respectively. However, since their results were not reproducible, we contacted with the author and realized that the results achieved by them didn't follow the official cross validation methods, which means they used different training data and validation data than main published papers and not comparable. So we will not compare our results with the results from [1].

Some researchers also proposed to train model directly from raw waveforms. Dai et al. [6] proposed a deep CNN architecture (up to 34 layers) with 1-D convolutional layers using 1-D raw data as input and they showed competitive accuracy with CNN using log mel spectrogram inputs [20]. Tokozume et al. [29] proposed a end-to-end network named EnvNet using raw data as inputs and reported EnvNet could extract a discriminative feature that complements the log mel features. In [30], they constructed a deeper recognition network based on EnvNet, referred as EnvNet-v2, and achieved better performance.

In addition, some researchers proposed to use external data for sound recognition. Mun et al. [18] proposed a DNN based transfer learning method for ESC. They first trained a DNN model using merged different web accessible environmental sound datasets. Then, they transferred the parameters of the pre-trained model and adapted the sound recognition system for target domain task using additional layers. Aytar et al. [2] proposed to learn rich sound representations from large amounts of unlabeled sound and videos dataset. They transferred the knowledge of pre-trained visual recognition network into the sound recognition network. Then, they used a linear-SVM classifier to classify the feature which is the output of the hidden layer of the sound recognition network to the target task.

3 Methods

3.1 Convolutional Neural Network

CNN is a stack of multi-layer neural networks including a group of convolutional layers, pooling layers and a limited number of fully connected layers. In this section, we propose a novel CNN as our ESC system model inspired by VGG Net [26], the architecture of which is presented in Table 1. The proposed CNN architecture is comprised of eight convolutional layers and two fully connected layers. We first use 2 convolutional layers with large filter kernals as a basic feature extractor. Then, we learn local patterns across frequency and time using 3×1 and 1×5 convolution filters, respectively. Next, we use small convolution

filters (3×3) to learn joint time-frequency patterns. Batch normalization [11] is applied to the output of convolutional layers to speed up training. We use the Rectified Linear Units (ReLU) to model the non-linearly for the output of each layer. After every two convolutional layers, a pooling layer is used to reduce the dimensions of the convolutional features maps, where maximum pooling is chosen in our network. To reduce the risks of overfitting, the dropout technique is applied after the first fully connected layers, with the probability of 0.5. L2-regularization is applied to the weights of each layer with the coefficient 0.0001. In the output layer, softmax function is used as the activation function which outputs probabilities of all classes.

Table 1. Configuration of proposed CNN. Out shape represents the dimension in (frequency, time, channel). Batch normalization is applied for each convolutional layer.

Layer	Ksize	Stride	Nums of filters	Out shape
Input	-	-	-	(128, 128, 2)
Conv1	(3, 7)	(1, 1)	32	(128, 128, 32)
Conv2	(3, 5)	(1, 1)	32	(128, 128, 32)
Pool1	(4, 3)	(4, 3)	-	(32, 43, 32)
Conv3	(3, 1)	(1, 1)	64	(32, 43, 64)
Conv4	(3, 1)	(1, 1)	64	(32, 43, 64)
Pool2	(4, 1)	(4, 1)	-	(8, 43, 64)
Conv5	(1, 5)	(1, 1)	128	(8, 43, 128)
Conv6	(1, 5)	(1, 1)	128	(8, 43, 128)
Pool3	(1, 3)	(1, 3)	-	(8, 15, 128)
Conv7	(3, 3)	(1, 1)	256	(8, 15, 256)
Conv8	(3, 3)	(1, 1)	256	(8, 15, 256)
Pool4	(2, 2)	(2, 2)	-	(4, 8, 256)
FC1	-	-	512	(512,)
FC2	-	-	Nums of classes	(Nums of classes,)

3.2 Mixup

Mixup is an simple but effective method to generate training data [36]. Figure 1 shows the pipeline of mixup. Different from traditional augmentation approaches, mixup constructs virtual training samples by mixing training samples. Normally, a model is optimized by using a mini-batch optimization method, such as mini-batch SGD, and each mini-batch data is selected from the whole original training data. In mixup, however, each data and label of a mini-batch is generated by mixing two training samples, which are determined by

$$\begin{cases} \hat{\mathbf{x}} = \lambda x_i + (1-\lambda)x_j \\ \hat{\mathbf{y}} = \lambda y_i + (1-\lambda)y_j \end{cases} \tag{1}$$

where x_i and x_j are two samples randomly selected from training data, and y_i and y_j are their one-hot labels. The mix factor λ is decided by a hyper-parameter α and $\lambda \sim \text{Beta}(\alpha, \alpha)$. Therefore, mixup extends the training data distribution by mixing various training data within or without the same class by a linear way, leading to a linear interpolation of the associated targets. Note that we do not use mixup for testing phase.

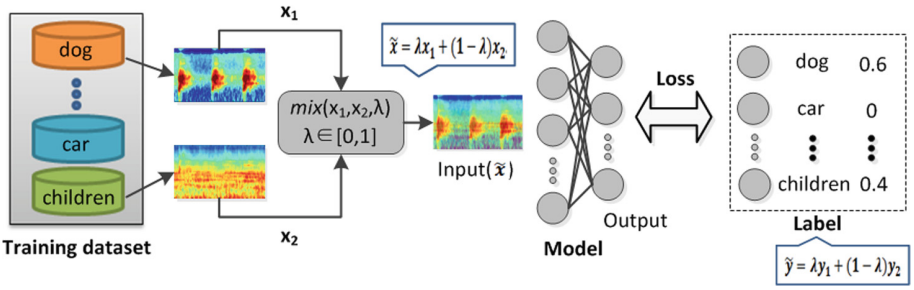


Fig. 1. Pipeline of mixup. Every training sample is created by mixing two examples randomly selected from original training dataset. We use the mixed sound to train the model and the train target is the mixing ratio.

4 Experiments

4.1 Dataset

Three publicly available datasets are used for model training and performance evaluation of the proposed approach, including ESC-10, ESC-50 [20] and UrbanSound8K [24], the detailed information of which is shown in Table 2.

The ESC-50 dataset consists of 2000 short environmental records which are divided into 50 classes in 5 major categories, including *animals, natural soundscapes and water sounds, human non-speech sounds, interior/domestic sounds*, and *exterior/urban noises*. All audio samples are 5 seconds with 44.1 kHz sampling frequency.

The ESC-10 dataset is a subset of 10 classes (400 samples) selected from the ESC-50 dataset (*dog bark, rain, sea waves, baby cry, clock tick, person sneeze, helicopter, chainsaw, rooster, fire crackling*).

The UrbanSound8K dataset is a collection of 8732 short (up to 4 s) audio clips of urban sound areas. And the audio clips are prearranged into 10 folds. The dataset is divided into 10 classes: *air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren*, and *street music*.

Table 2. Information of datasets.

Datasets	Classes	Nums of samples	Duration
UrbanSound8K	10	8732	9.7 h
ESC-50	50	2000	2.8 h
ESC-10	10	400	33 min

4.2 Preprocessing

We use a 44.1 kHz sampling rate for ESC-10, ESC-50, UrbanSound8K datasets. All audio samples are normalized into a range from −1 to 1. In order to avoid overfitting and to effectively utilize the limited data, we use Time Stretch [23] and Pitch Shift [23] deformation methods to generate new audio samples. We use two spectrogram-like representations, log mel spectrogram (Mels) and gammatone spectrogram (GTs). Both features are extracted from all recordings with hamming window size of 1024, hop length of 512 and 128 bands. Then, the resulting spectrograms are converted into logarithmic scale. In our experiments, we use a simple energy-based silence drop algorithm to drop silence regions. Finally, the spectrograms are split into 128 frames (approximately 1.5 s) length with 50% overlap. The delta information of the original spectrogram is calculated, which is the first temporal derivative of the spectrogram feature. Then, we use the segments with their deltas as a two-channel input to the network.

4.3 Training Settings

All models are trained using mini-batch stochastic gradient descent (SGD) with Nesterov momentum of 0.9. We used a learning rate decrease schedule with a initial learning rate of 0.1, and then divided the learning rate by 10 every 80 epoch for UrbanSound8K and 100 epoch for ESC-10 and ESC-50. Every batch consists of 200 samples randomly selected from training set without repetition. The models are trained for 200 epochs for UrbanSound8K and 300 epochs for ESC-50 and ESC-10. We initialize all the weights to zero mean Gaussian noise with a standard deviation of 0.05. We use cross entropy as the loss function, which is typically used for multi classification task.

In the test stage, feature extraction and audio cropping patterns are the same as those used in the training stage. Prediction probability of a test audio sample is the average of predicted class probability of each segment. The predicted label of the test audio sample is the class with the highest posterior possibility. The classification performance of the methods is evaluated by the K-fold cross-validation. For the ESC-50 and ESC-10 dataset, K is set to 5, while for the UrbanSound8K dataset, K is set to 10.

All models are trained using Keras library with TensorFlow backend on an Nvidia P100 GPU with a 12 GB memory.

5 Results and Analysis

The classification accuracy of the proposed method compared with recent related works is shown in Table 3. It can be observed that our method achieved the state-of-the-art performance (83.7%) on UrbanSound8K dataset and competitive performance (91.7%, 83.9%) on ESC-10 and ESC-50. The average classification accuracy of our methods with Mels outperformed PiczakCNN [19] (baseline) by 10.8%, 17.6%, 9.9% on ESC-10, ESC-50 and UrbanSound8K datasets, respectively. Data augmentation is an important technique for increasing performance for limited dataset, which gave an improvement of 1.1%, 3.3% and 5.3% on ESC-10, ESC-50 and UrbanSound8K, respectively. In addition, GTs improved by 0.4%, 1.4% and 1.1% over Mels on ESC-10, ESC-50 and UrbanSound8K, respectively. We can see that classification accuracy with GTs is always better than accuracy with Mels on ESC-10, ESC-50 and UrbanSound8K datasets, which indicates that feature representation is a critical factor for classification performance. What's more, mixup is a powerful way to improve performance which can always perform better results than that without mixup. In our experiments, Mixup gave an improvement of 1.5%, 2.4% and 2.6% with Mels on ESC-10, ESC-50, UrbanSound8k datasets, respectively. As mentioned in Sect. 3, mixup trains a network using a linear combination of training examples and their labels and leads to a regularization for neural network and generalization for unseen data. For the effect of mixup, we do a further exploration in the following parts.

Table 3. Classification accuracy (%) of different ESC systems. In our ESC system, we compare two different features with augmentation and without augmentation. 'aug' stands for augmentation, including Pitch Shift, Time Stretch. Note that we will not compare with the results of Agrawal [1] which was discussed in Sect. 2.

Model	Acc (%)			
	Feature	ESC10	ESC50	UrbanSound8K
PiczakCNN [19]	Mels	80.5	64.9	72.7
D-CNN [37]	Mels	-	68.1	81.9
SoundNet [2]	-	**92.1**	74.2	-
Envnet-v2 [29]	Raw data	91.4	**84.9**	78.3
proposedCNN	Mels	88.7	76.8	74.7
	GTs	89.2	78.9	77.4
proposedCNN + mixup	Mels	90.2	79.2	77.3
	GTs	90.7	80.7	79.8
proposedCNN + aug + mixup	Mels	91.3	82.5	82.6
	GTs	**91.7**	83.9	**83.7**
Human performance	-	95.7	81.3	-
Agrawal [1]	GTs	-	79.10	85.34

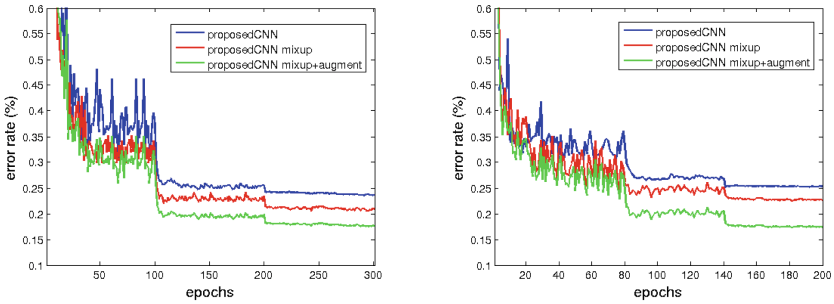


Fig. 2. Training curves of our proposed CNN on (a) ESC-50 and (b) UrbanSound8K datasets.

5.1 Comparison of Network Architecture

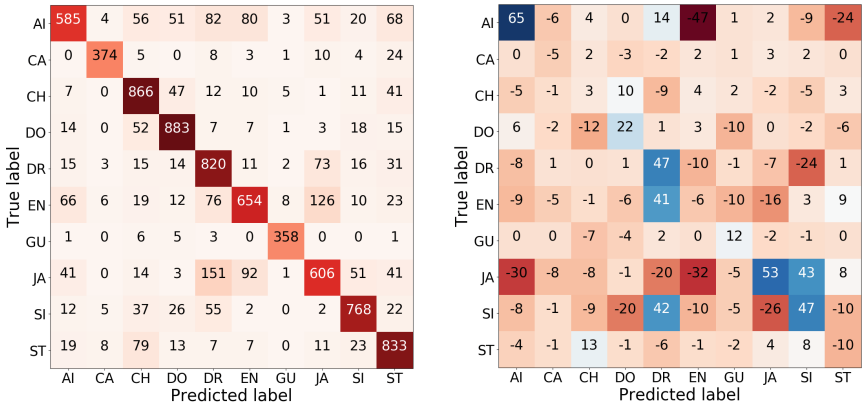
We compare our proposed CNN with a VGG network architecture with same depth of network. This VGG network has same network parameters with our proposed CNN except for replacing to use 3×3 convolution filters and 2×2 stride pooling and we refer to this architecture as VGG10. In Table 4, we provide classification accuracy of proposedCNN and VGG10 on ESC-10, ESC-50 and UrbanSound8K datasets. The results shows that our proposed CNN always performs better than VGG10 on three datasets.

Table 4. Comparison between proposed CNN and VGG10 Net (%).

Model	ESC-10	ESC-50	UrbanSound8K
proposedCNN	88.7	76.8	74.7
VGG10	87.5	73.3	73.2

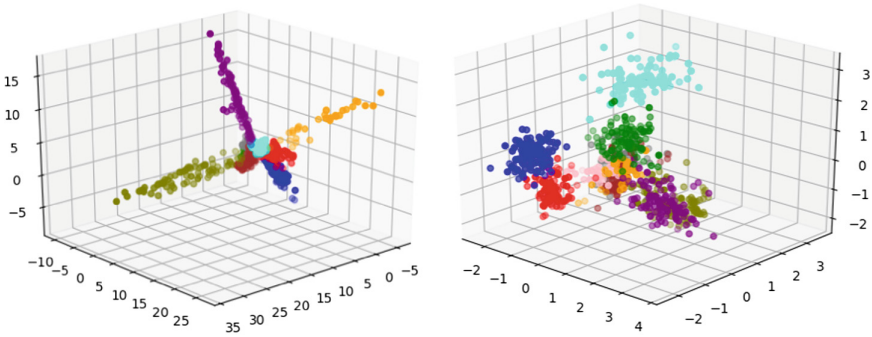
5.2 Effects of Mixup

Analysis. The confusion matrix by the proposed CNN with Mels and mixup for the UrbanSound8K dataset is given in Fig. 3(a). We can observe that the most misrecognition happened between two noise-like classes, such as *jackhammer* and *drilling*, *engine idling* and *jackhammer*, and *air conditioner* and *engine idling*. In Fig. 3(b), we provide the difference of the confusion for the proposed CNN method with and without mixup. We see that mixup gives an improvement for most classes, especially for *air conditioner, drilling, jackhammer* and *siren*. However, mixup also has a slightly harmful effect on the accuracy for some classes and increases confusion between some specific pairs classes. For example,



(a) (b)

Fig. 3. (a) Confusion matrix for UrbanSound8K dataset using the proposed CNN model applying to Mels with mixup augmentation methods. (b) Different between the confusion matrix for UrbanSound8K dataset using the proposed CNN and Mels with mixup and without mixup: the negative values (brown) mean the confusion is decreased with mixup, the positive (blue) values mean the confusion is increased with mixup. Classes are air conditioner (AI), car horn (CA), children playing (CH), dog barking (DO), drilling (DR), engine idling (EN), gun shot (GU), jackhammer (JA), siren (SI) and street music (ST). (Color figure online)



(a) (b)

Fig. 4. Visualization of the feature distribution at the output of FC1 using PCA (a) without mixup and (b) with mixup.

although mixup reduces the confusion between *jackhammer* and *engine idling*, it increases the confusion between *jackhammer* and *siren*.

To gain further insights to the effect of mixup, we visualized the feature distributions for UrbanSound8K with mixup and without mixup using PCA in Fig. 4. The feature dots represent the high-level feature vectors obtained at the output of the first fully connected layer (FC1). We can observe that it is

quite different between feature distributions with and without mixup. Figure 4(a) shows the feature distributions of different classes with mixup. Some classes have a large within-class variance of the feature distribution, while some have a small within-class variance. In addition, the between-class distances of different pairs of classes are also varied, which may make models more sensitive to some classes. However, features of most classes distribute within a small space with a relative smaller within-class variance and the boundary of most classes is clear as shown in Fig. 4(b).

Hyper-parameter α Selected. In order to achieve a better performance for our system on ESC, the effect of mixup hyper-parameter α is further explored. Figure 5 shows the change of accuracy with different α ranging from $[0.1, 0.5]$. We see that when $\alpha = 0.2$, the best accuracy is achieved on all three datasets.

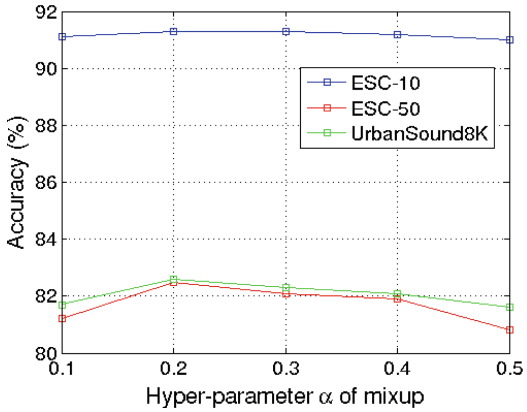


Fig. 5. Curves of an accuracy with different α for ESC-10, ESC-50, UrbanSound8K

6 Conclusion

In this paper, we proposed a novel deep convolutional neural network architecture for environmental sound classification. We compared our proposed CNN with VGG10 and results showed that our proposed CNN always performed better. To further improve the classification accuracy, mixup was applied in our ESC system. As a result, the proposed ESC system achieved state-of-the-art performance on UrbanSound8K dataset and competitive performance on ESC-10 and ESC-50 dataset. Furthermore, we explored the impacts of mixup on the classification accuracy and feature space distribution of different classes on UrbanSound8K dataset. The results showed that mixup is a powerful method to improves classification accuracy. Our future work will focus on the network design and exploration for using mixup method for specific classes.

References

1. Agrawal, D.M., Sailor, H.B., Soni, M.H., Patil, H.A.: Novel teo-based gammatone features for environmental sound classification. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1809–1813. IEEE (2017)
2. Aytar, Y., Vondrick, C., Torralba, A.: Soundnet: learning sound representations from unlabeled video. In: Advances in Neural Information Processing Systems, pp. 892–900 (2016)
3. Barchiesi, D., Giannoulis, D., Dan, S., Plumbley, M.D.: Acoustic scene classification: classifying environments from the sounds they produce. IEEE Signal Process. Mag. **32**(3), 16–34 (2015)
4. Casey, M.A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-based music information retrieval: current directions and future challenges. Proc. IEEE **96**(4), 668–696 (2008)
5. Chu, S., Narayanan, S., Kuo, C.C.J.: Environmental sound recognition with time-frequency audio features. Institute of Electrical and Electronics Engineers Inc., (2009)
6. Dai, W., Dai, C., Qu, S., Li, J., Das, S.: Very deep convolutional neural networks for raw waveforms. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 421–425. IEEE (2017)
7. Eronen, A.J., et al.: Audio-based context recognition. IEEE Trans. Audio Speech Lang. Process. **14**(1), 321–329 (2006)
8. Geiger, J.T., Helwani, K.: Improving event detection for audio surveillance using gabor filterbank features. In: Signal Processing Conference, pp. 714–718 (2015)
9. Graves, A., Mohamed, A.R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
10. Hinton, G., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. **29**(6), 82–97 (2012)
11. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift, pp. 448–456 (2015)
12. Kons, Z., Toledo-Ronen, O.: Audio event classification using deep neural networks. In: Interspeech, pp. 1482–1486 (2013)
13. Lee, K., Ellis, D.P.: Audio-based semantic concept classification for consumer video. IEEE Trans. Audio Speech Lang. Process. **18**(6), 1406–1416 (2010)
14. Lyon, R.F.: Machine hearing: an emerging field [exploratory dsp]. Signal Process. Mag. IEEE **27**(5), 131–139 (2010)
15. McLoughlin, I., Zhang, H., Xie, Z., Song, Y., Xiao, W.: Robust sound event classification using deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(3), 540–552 (2015)
16. McLoughlin, I.V.: Line spectral pairs. Signal Process. **88**(3), 448–467 (2008)
17. Mesaros, A., et al.: Detection and classification of acoustic scenes and events: outcome of the dcase 2016 challenge. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(2), 379–393 (2018)
18. Mun, S., Shon, S., Kim, W., Han, D.K., Ko, H.: Deep neural network based learning and transferring mid-level audio features for acoustic scene classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 796–800. IEEE (2017)

19. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: IEEE International Workshop on Machine Learning for Signal Processing, pp. 1–6 (2015)
20. Piczak, K.J.: ESC: dataset for environmental sound classification. In: ACM International Conference on Multimedia, pp. 1015–1018 (2015)
21. Radhakrishnan, R., Divakaran, A., Smaragdis, P.: Audio analysis for surveillance applications. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 158–161 (2005)
22. Sainath, T.N., Mohamed, A.R., Kingsbury, B., Ramabhadran, B.: Deep convolutional neural networks for LVCSR. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8614–8618. IEEE (2013)
23. Salamon, J., Bello, J.: Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Process. Lett. **PP**(99), 1 (2016)
24. Salamon, J., Jacoby, C., Bello, J.P.: A dataset and taxonomy for urban sound research. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 1041–1044. ACM (2014)
25. Schedl, M., Gómez, E., Urbano, J., et al.: Music information retrieval: recent developments and applications. Found. Trends® Inf. Retr. **8**(2–3), 127–261 (2014)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Takahashi, N., Gygli, M., Pfister, B., Van Gool, L.: Deep convolutional neural networks and data augmentation for acoustic event detection. arXiv preprint arXiv:1604.07160 (2016)
28. Temko, A., Monte, E., Nadeu, C.: Comparison of sequence discriminant support vector machines for acoustic event classification. In: 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings, vol. 5, p. V. IEEE (2006)
29. Tokozume, Y., Harada, T.: Learning environmental sounds with end-to-end convolutional neural network. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2721–2725. IEEE (2017)
30. Tokozume, Y., Ushiku, Y., Harada, T.: Learning from between-class examples for deep sound recognition. arXiv preprint arXiv:1711.10282 (2018)
31. Typke, R., Wiering, F., Veltkamp, R.C.: A survey of music information retrieval systems. In: Proceedings of the 6th International Conference on Music Information Retrieval, pp. 153–160. Queen Mary, University of London (2005)
32. Uzkent, B., Barkana, B.D., Cevikalp, H.: Non-speech environmental sound classification using svms with a new set of features. Int. J. Innov. Comput. Inf. Control **8**(5), 3511–3524 (2012)
33. Vacher, M., Serignat, J.F., Chaillol, S.: Sound classification in a smart room environment: an approach using gmm and hmm methods. In: SpeD, vol. 1 (2014)
34. Valero, X., Alias, F.: Gammatone cepstral coefficients: biologically inspired features for non-speech audio classification. IEEE Trans. Multimedia **14**(6), 1684–1689 (2012)
35. Zhang, H., Mcloughlin, I., Song, Y.: Robust sound event recognition using convolutional neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 559–563 (2015)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: Mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
37. Zhang, X., Zou, Y., Shi, W.: Dilated convolution neural network with LeakyReLU for environmental sound classification. In: 2017 22nd International Conference on Digital Signal Processing (DSP), pp. 1–5. IEEE (2017)