





DCNN-LSTM Based Audio Classification Combining Multiple Feature Engineering and Data Augmentation Techniques

Md. Moinul Islam¹ , Monjurul Haque², Saiful Islam³,
Md. Zesun Ahmed Mia^{4,5} , and S. M. A. Mohaiminur Rahman¹

¹ Chittagong University of Engineering and Technology, Chittagong, Bangladesh

² Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh

³ Ahsanullah University of Science and Technology, Dhaka, Bangladesh

⁴ Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh

⁵ University of Liberal Arts Bangladesh (ULAB), Dhaka, Bangladesh

Abstract. Everything we know is based on our brain's ability to process sensory data. Hearing is a crucial sense for our ability to learn. Sound is essential for a wide range of activities such as exchanging information, interacting with others, and so on. To convert the sound electrically, the role of the audio signal comes into play. Because of the countless essential applications, audio signal & their classification poses an important value. However, in this day and age, classifying audio signals remains a difficult task. To classify audio signals more accurately and effectively, we have proposed a new model. In this study, we've applied a brand-new method for audio classification that combines the strengths of Deep Convolutional Neural Network (DCNN) and Long-Short Term Memory (LSTM) models with a unique combination of feature engineering to get the best possible outcome. Here, we have integrated data augmentation and feature extraction together before fitting it into the model to evaluate the performance. There is a higher degree of accuracy observed after the experiment. To validate the efficacy of our model, a comparative analysis has been made with the latest conducted reference works.

Keywords: DCNN-LSTM · Spectrograms · Short Time Fourier Transform · Data augmentation · Spectral feature extraction · MFCC · Melspectrogram · Chroma STFT · Tonnetz

1 Introduction

Digital and analog audio signals both use a varying amount of electrical voltage to delineate sound. Our daily lives depend heavily on audio signals of various origins. No one would be able to hear anything without it. Audio signals are now required not just by humans, but also by man-made machines. Human-like sound comprehension has several uses, involving intelligent machine control and monitoring, acoustic information use, acoustic surveillance, and categorization and information extraction applications such as exploring audio archives

and audio-assisted multimedia assets [9]. For many years, categorizing audio or sound has been an important area of research. Intending to achieve this classification, multiple models and features have been tried and experimented with over the years, all of which have proved to be helpful and accurate in the process of classifying and separating audio and sound. Many possible applications exist in the area of sound detection and classification, including matrix factorization, the categorization of music genres, wavelet filterbanks, automated music tagging, dictionary learning, bird song classifications, IoT embedded automated audio categorization, and emotion recognition [1–3, 6, 8, 12]. Since deep learning was introduced, it has boosted research in various fields and swiftly superseded traditional machine learning algorithms by exhibiting superior performance on numerous tasks. With or without Artificial Intelligence, there are countless possible approaches for developing audio recognition and classification models that use various audio feature extraction procedures. The detection and categorization of ambient sound is a fascinating subject with several applications, ranging from crime detection to environmental context-aware analysis. For audio classification, prominent classifier models include those that use sensible artificial intelligence or linear predictive coding, as well as those Deep Neural Networks, Decision Tree Classifier, and Random Forest.

A few contributions have been made to the field of audio categorization. In recent research studies, convolutional neural networks were shown to be very efficient in classifying brief audio samples of ambient noises. The authors in [11] used the publicly accessible ESC-10, ESC-50, and Urbansound8K data sets and enhanced them by adding arbitrary temporal delays to the original tracks and conducted class-dependent time stretching and pitch shifting on the ESC-10 training set, as well as extracted Log-scaled mel-spectrograms from all recordings, to develop a model composed of two convolutional ReLU layers with max-pooling, two fully connected layers of each ReLU, and a softmax output layer trained on a low-level audio data representation. The authors used 300 epochs for the short segment version and 150 epochs for the long segment variant and tested the model using fivefold cross-validation (ESC-10 and ESC-50) and tenfold cross-validation (UrbanSound8K) with a single training fold to show that CNN outperformed solutions based on manually-engineered features. Palanisamy et al. [10] showed that standard deep CNN models trained on ImageNet might be used as strong foundation networks for audio categorization. They claimed that just by fine-tuning basic pre-trained ImageNet models with such a single set of input character traits for audio tasks, they could achieve cutting-edge results on the UrbanSound8K and ESC-50 datasets, as well as good performance on the GTZAN datasets, and to define spectrograms using qualitative visuals, CNN models might learn the bounds of the energy distributions in the spectrograms. Abdoli et al. [4] presented a method for classifying ambient sound that uses a 1D Convolutional Neural Network (CNN) to attempt to acquire a representation straight from the audio input in order to capture the signal's precise temporal characteristic. The performance of their proposed end-to-end approach for detecting ambient noises was found to be 89% accurate. The suggested

end-to-end 1D design for ambient sound categorization employs lesser parameters than the bulk of previous CNN architectures while reaching a mean accuracy of 11.24% to 27.14% greater than equivalent 2D designs.

In our research, we've introduced an entirely new technique to audio classification strategy by integrating two separate models: deep CNN and LSTM. Before we train the data in our newly proposed model, we used a unique combination of feature engineering methods to discover the best results. There are three phases to sound classification: audio signal preprocessing, spectral feature extraction, and classification of the corresponding audio signal. The Urbansound8K dataset has been utilized for audio categorization by our team. There are 8732 audio slices in total that have been tagged in this dataset. There are ten groups in the audio file which entails air conditioning, car horns, children's laughter, dog barks, engine idle, gunshots, jackhammers, sirens, and street music and all of them are examples of ambient noise. Data augmentation is first used to improve the model's training results so that it can yield good results. Three data augmentation methods were investigated: time-stretching, noise introduction, and pitch shifting. To convert audio data to numerical values, we used the NumPy array in python. The audio was then transformed using spectral features via Fourier Transform from the time domain to the frequency domain. In addition to Zero Crossing Rate, Chroma STFT (Short-Time Fourier Transform), MFCC (Mel-frequency Cepstral Coefficient), Mel spectrogram, RMS, and Tonnetz, we have also computed a number of feature extraction approaches like these. Spectral feature extraction approaches are being combined to create a new model. After that, the 34928 numerical data with a total field of 5867904 have been integrated using data augmentation and spectral feature extraction before training the data into the model. We trained with 80% of the data, tested with 10%, and validated with 10%. Finally, we've trained the data with our recommended model, a hybrid of deep CNN and LSTM. There are three layers in a deep CNN. We have used Adam optimizer for improved optimization. Hyperparameter tuning uses batch normalization, maximum pooling, and dropout all at once. ReLU and Softmax were used to fit the model, and Softmax was also employed for the output layer. The LSTM model's input layer receives data from the output layer. The LSTM model makes use of two levels. As with deep CNN, we used Adam optimizer and activation functions like ReLU and Softmax to fit the model better and improve tuning. However, in this case, dropout was used for hyperparameter tuning. After that, the accuracy of audio classification was significantly enhanced. Finally, our novel model has been compared to the models from other recent reference works in order to highlight its worth.

2 Methodology

The overall methodology of our suggested audio classification model is described in this section. We have used a benchmark dataset UrbanSound8K [13] for validating our model. This dataset contains 8732 brief audio samples (with a duration of 4s or less) taken from a variety of urban recordings, including air condi-

tioners, vehicle horns, kids playing, barking dogs, drilling, engine revving, gunshots, jackhammers, sirens, and street music, among other things. This dataset is divided into ten (ten) classes, as stated above. It was found that vehicle horns, gunshots, and siren noises were not uniformly dispersed throughout the classrooms.

2.1 Data Augmentation

Data augmentation is a simple technique for generating synthetic data with variations from the current or existing samples to offer the model with larger data samples with more variety, allowing the model to prevent overfitting and be more generalized. There are several augmentation methods in audio, such as Noise Injection, Time Shifting, Pitch Shifting, Changing Speed, Time Stretching, and others.

This research has adopted three data augmentation techniques: Background Noise Injection, Pitch Shifting, and Time Stretching. In Noise Injection, the sample data was merged with a separate recording that includes external noise from a variety of acoustics. Each data was generated by,

$$m = x_i(1 - w) + wy_i \quad (1)$$

where x_i is the original audio sample of the dataset, y_i is the background noise that is injected, and w is the weighted parameter chosen randomly for each merge within a range of 0.001 to 0.009. During Pitch Shifting, the pitch of the audio samples is either increasingly or decreasingly shifted based on a particular value. Each data was pitch-shifted by $[-2, -1, 1, 2]$. Time stretching is an audio processing technique that lengthens or shortens the duration of a sample without altering its pitch. The augmentation techniques were applied using the Librosa library. Figure 1a, 1b and 1c illustrate the data augmentation techniques applied in the dataset.

2.2 Spectral Feature Extraction

When using feature extraction, the acoustic signal is transformed into a series of acoustic feature vectors that accurately describe the input audio sound. The goal is to condense the several massive amounts of data in each file into a considerably smaller collection of characteristics with a known number. We have used spectral characteristics to solve our classification problem, which involves utilizing the Short-Time Fourier Transformation to transform the enhanced audio samples from time domain to frequency domain displayed in Fig. 2a. There are numerous spectral features. Among them, we have employed six: Zero Crossing Rate, Chroma STFT, MFCC, Mel spectrogram, Tonnetz, and computing RMS value for each frame. Figure 2b, 2c and 2d represent the plotting of spectrogram for each feature extraction technique.

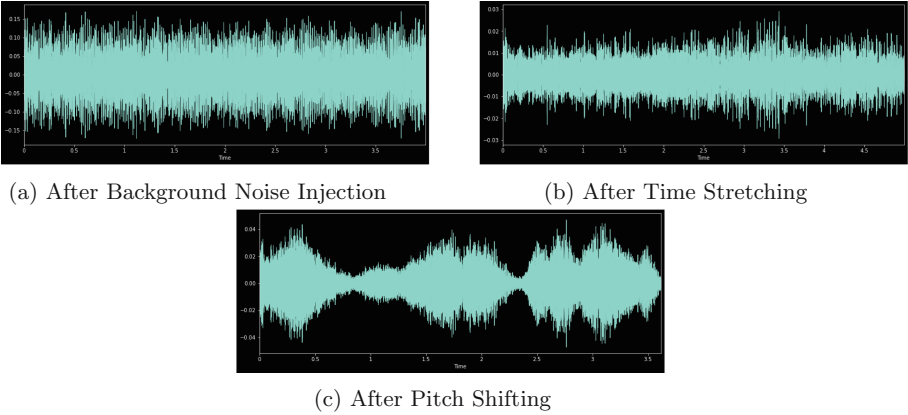


Fig. 1. Data augmentation illustration

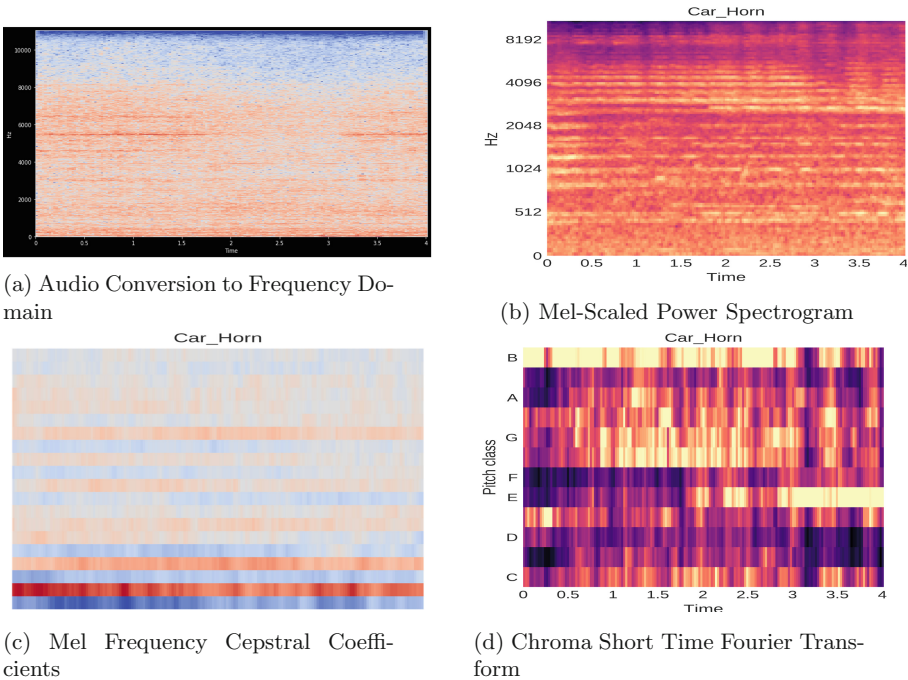


Fig. 2. Audio conversion & spectral feature extraction

Zero-Crossing Rate indicates that how many times the signal shifts from positive to negative and vice-versa, and that will be divided by the frame duration [7], where *sgn* is the sign function.

$$Z_i = \frac{1}{2w_L} \sum_{n=1}^{w_L} |sgn[x_i(n)] - sgn[x_i(n-1)]| \quad (2)$$

The Chroma rate of an audio signal depicts the strength of each of the audio signal's twelve distinct pitch classes. They can be used to distinguish between the pitch class profiles of audio streams. Chroma-STFT contains information regarding pitch and signal structure categorization and uses short-term Fourier transform to generate Chroma properties. MFCC stands for Mel Frequency Cepstral Coefficients are concise representations of the spectrum. By transforming the conventional frequency to Mel Scale, MFCC takes into consideration human perception for sensitivity at correct frequencies. Mel spectrogram is a combination of Mel scale and spectrogram, whereas Mel scale denotes the frequency scale's nonlinear transformation. The y-axis indicates Mel scale, while the x-axis depicts time. Tonnetz detects harmonic shifts in audio recordings to calculate tonal centroid features. It is an infinite planar representation of pitch relationships in an audio sample.

For feature scaling purposes in our proposed method, we have utilized two standard techniques, 'One Hot Encoding' and 'Standard Scaler'. One hot encoding replaces the label encoded categorical data with numbers. It is a standardization technique to scale the independent features to bring them in the same fixed range. The Standard Normal Distribution (SND) is followed by StandardScaler. That's why the mean is set to 0, and the data is scaled to unit variance.

2.3 Deep CNN-LSTM Model Architecture

In the DCNN-LSTM design, CNN layers for feature extraction on input data are integrated with LSTMs to provide sequence prediction, resulting in a highly efficient feature extraction system. We combined CNN and LSTM models, both of which use spectrograms as their input. In order to generate a DCNN-LSTM model, Deep CNN layers on the front end were combined with LSTM layers and a Dense layer on the output. In this architecture, two sub-models are used for feature extraction and feature interpretation across a large number of iterations: the Deep CNN Model for extracting features and the LSTM Model for feature interpretation (Fig. 3).

We presented a model that consists of three layers of 2D convolutional networks, and two layers of MaxPooling2D arranged into a stack of the desired depth. These layers assess the spectral properties of the spectrograms, while the pooling layers help solidify the interpretation. The Conv2D layers interpret the spectrum characteristics of the spectrograms, and the pooling layers consolidate the interpretation. The first Conv2D layer that processes the input shape uses 64 filters, a kernel size of 5, and a stride of 1 before applying a MaxPooling layer to decrease the size of the input shape. Our framework utilizes a 5x5 filtered matrix as the argument defines the kernel's window's size. Due to stride being set to 1 in the first layer, the filter moves one unit to converge around the input volume. Using the 'same padding' technique, this convolutional layer yields the

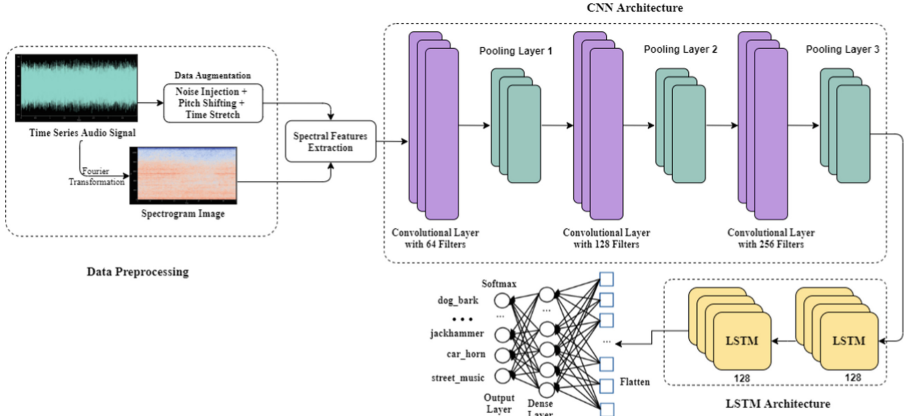


Fig. 3. Overall proposed methodology

same height and weight as the original. We chose ReLU as the activation function for this layer instead of sigmoid units because of its many advantages over more conventional units, including efficient gradient propagation and faster calculation than sigmoid units. Also, in order to minimize overfitting, there is a dropout of 0.3 in the layers that uses the same padding method and activation function (ReLU) as the other two Conv2D layers.

To stack the LSTM layers, we first created two LSTM layers with a total hidden unit count of 128 for each layer, and then we set the return sequence to true in order to stack the layers. To avoid overfitting, the output of both LSTM layers requires a 3D array followed by Time Distributed Dense layers as input with a dropout of 0.2 to be used. As a result, it was determined that ReLU would be used as the activation function in both layers with input sizes of 64 and 128 for the first layer since its input shape is (21,8), which indicates 20 iterations and will inform LSTM how many numbers of instances it should go through the process once the input has been applied. Afterward, the outcome from the time dispersed dense layer is utilized as the input in the flatten layer, and the process repeats until the desired result is achieved. When we were finished with the flattening process, we were left with a vector of input data, which we then passed through the dense layer. We were able to transform the information provided to a discrete probability distribution and use that distribution as an input in the dense layer by utilizing the Softmax activation function in the dense layer of the network.

We have utilized the Adam, an optimization technique, which measures the rate of development at which a parameter adapts to changes in its environment. The Adam optimizer outperforms the previous ones in terms of performance and provides a gradient descent that is tuned. For individual parameters, the adaptive learning rates are used to estimate the appropriate level of learning. In many circumstances, it has been shown that Adam favors error surfaces with flat minima, which is a good optimizer. The parameters β_1 and β_2 only specify the periods over which the learning rates degrade, and not necessarily the learning

rate itself. The acquisition rates will be all over the place if they degrade quickly. It will take a long time to learn the learning rates if they degrade slowly. The learning rates are automatically calculated based on a moving estimate of the parameter gradient, and the parameter squared gradient in all circumstances.

3 Results

Our research strategy involved identifying attributes that are both effective and accurate for the DCNN-LSTM model. In this section, we assess our model in terms of experiments conducted. We also evaluate the effectiveness of our proposed ensemble method, pre-trained weights, and finally compare to some of the previous state-of-the-art models.

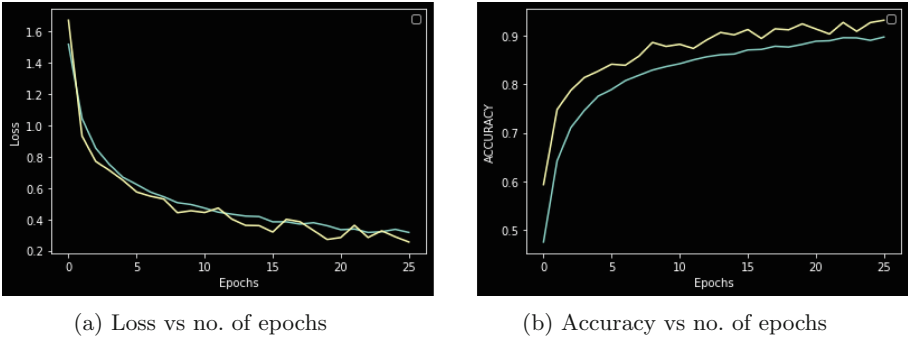


Fig. 4. Validation loss & accuracy of our proposed model

In the data preprocessing module, we stacked three data augmentation techniques, background noise injection, time-stretching, and pitch shifting, to reduce overfitting & evaluate the performance of our model. To extract spectral features from the spectrograms, MFCCs, Mel Spectrogram, Chroma STFT, Tonnetz were stacked with one another in addition to computing zero-crossing rate (ZRC) and Root Mean Square (RMS) value for each frame of the audio data and obtained 169 features in total to work with. Stacking the techniques was effective in enhancing our model’s performance considerably. We then fed the data into our proposed DCNN-LSTM model illustrated in Sect. 2.3, evaluated performance metrics, and validated the model for the dataset. Stacking those techniques helped us to reach a better validation accuracy of 93.19% with an epoch size of 26 and used Stratified 10-fold cross-validation to ensure the robustness of the result in terms of modeling CNN (layer 3 Conv2D, epochs of 50) with 86.1% and LSTM (layer 2, epochs of 200) with 87.75% respectively for the training process. Figure 4 illustrates the validation loss and accuracy of our proposed DCNN-LSTM model in the y-axis and the number of epochs in the x-axis as we can see that with the increase in the number of epochs, the validation error of our model decreases exponentially for

both the training and testing data. The epoch count was set at 50. Still, the error stopped improving after 26 epochs and returned the accuracy and loss result due to the early Callback function without further increasing the computational time for the model. Table 1 shows the comparison accuracy of our proposed model with the previous state-of-the-art models.

Table 1. Proposed model vs previous state-of-the-art models

Model	Dataset	Accuracy (%)
logmel-CNN [16]	ESC-50	78.3
DCNN + Mix-up [17]	UrbanSound8K	83.7
DenseNet (Pretrained Ensemble) [10]	UrbanSound8K	87.42
Conv1D + Gammatone [4]	UrbanSound8K	89
DCNN with Multiple Features + Mix-up [14]	ESC-50	88.5
GoogleNet [5]	UrbanSound8K	93
TSCNN-DS [15]	UrbanSound8K	97.2
Proposed DCNN-LSTM + Stacked Features & Augmentation	UrbanSound8K	93.19

4 Conclusion

This paper proposes an approach to urban sound classification, which comprises a deep neural network of two different neural network models, CNN and LSTM. Also, in combination with two separate stacks of various multiple data augmentation and feature extraction techniques. UrbanSound8K has been used to train and test our models, one of the finest datasets of this domain. With the aforementioned feature engineering, training, validating, and testing the model on this dataset assists us to acquire a decent result of 93.19% accuracy, which is pretty much close to state-of-the-art result and better than other previous works.

Though we have emphasized data augmentation on a single dataset, the comparison would be more relevant if we could also work with other prominent datasets. Our model's such accuracy comes without any usage of pre-trained models and transfer learning. So, there remains a scope of future work of using these two, possibly improving our existing accuracy. Moreover, a simple stack of DCNN-LSTM has been effectively used for urban sound classification and has achieved a high score, and it is a matter of future research that whether various combinations of more sophisticated models of recurrent neural networks or convolutional neural networks can bring much better score.

References

1. Vasant, P., Zelinka, I., Weber, G.-W.: Intelligent Computing and Optimization. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-00979-3>. ISBN 978-3-030-00978-6

2. Vasant, P., Zelinka, I., Weber, G.-W.: *Intelligent Computing and Optimization*. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-33585-4>. ISBN 978-3-030-33585-4
3. Vasant, P., Zelinka, I., Weber, G.-W.: *Intelligent Computing and Optimization*. Springer, Cham (2020). <https://doi.org/10.1007/978-3-030-68154-8>
4. Abdoli, S., Cardinal, P., Koerich, A.L.: End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Syst. Appl.* **136**, 252–263 (2019)
5. Boddapati, V., Petef, A., Rasmusson, J., Lundberg, L.: Classifying environmental sounds using image recognition networks. *Procedia Comput. Sci.* **112**, 2048–2056 (2017). *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France*
6. Costa, Y.M., Oliveira, L.S., Silla, C.N.: An evaluation of convolutional neural networks for music classification using spectrograms. *Appl. Soft Comput.* **52**, 28–38 (2017)
7. Giannakopoulos, T., Pikrakis, A.: Audio features. In: Giannakopoulos, T., Pikrakis, A. (eds.) *Introduction to Audio Analysis*, pp. 59–103. Academic Press, Oxford (2014)
8. Hershey, S., et al.: CNN architectures for large-scale audio classification (2017)
9. Li, J., Dai, W., Metze, F., Qu, S., Das, S.: A comparison of deep learning methods for environmental sound detection. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 126–130. IEEE (2017)
10. Palanisamy, K., Singhania, D., Yao, A.: Rethinking CNN models for audio classification (2020)
11. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE (2015)
12. Salamon, J., Bello, J.P.: Unsupervised feature learning for urban sound classification. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 171–175. IEEE (2015)
13. 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*, MM 2014, pp. 1041–1044. Association for Computing Machinery, New York (2014)
14. Sharma, J., Granmo, O.C., Goodwin, M.: Environment sound classification using multiple feature channels and attention based deep convolutional neural network. In: *INTERSPEECH*, pp. 1186–1190 (2020)
15. Su, Y., Zhang, K., Wang, J., Madani, K.: Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors* **19**(7), 1733 (2019)
16. 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)
17. Zhang, Z., Xu, S., Cao, S., Zhang, S.: Deep convolutional neural network with mixup for environmental sound classification. In: Lai, J.-H., et al. (eds.) *PRCV 2018. LNCS*, vol. 11257, pp. 356–367. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03335-4_31