International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Neural network-based spectrogram feature extraction with KNN ensemble for environmental sound classification

Shashi Verma [1,*], Garima Srivastava [1] and Lalita Kumari [2]

[1] *Department of Computer Science and Engineering, Amity University, Lucknow, Uttar Pradesh, India.*
[2] *Department of Computer Science and Engineering, Amity University, Patna, India.*

International Journal of Science and Research Archive, 2026, 19(02), 790-797

Publication history: Received on 10 April 2026; revised on 10 May 2026; accepted on 12 May 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.2.1053

Abstract

Predicting real-world events is a central objective of machine learning, where pattern recognition in complex signals plays a critical role. Accurate pattern extraction facilitates effective feature representation, which in turn enhances classification performance. In this study, we focus on extracting discrimi-native features using a neural network for environmental sound classification. The network is trained on image repre-sentations of audio signals, primarily spectrograms, which enable the model to capture both temporal and frequency characteristics of sound effectively. The features learned by the neural network are subsequently utilized within a k-nearest neighbors (KNN) ensemble framework for classifica-tion. This ensemble approach allows us to evaluate the ro-bustness and generalization capability of the extracted feature representations. Experimental results demonstrate strong per-formance. The proposed system was evaluated on the DCASE-2017 Acoustic Scene Classification dataset, achiev-ing a classification accuracy of 96.23%. Additionally, testing on the UrbanSound8K dataset yielded an accuracy of 86.7%. These findings indicate that neural network–derived features are highly effective for reliable acoustic scene identi-fication

Keywords: Environmental Sound Classification; Spectrograms; Acous-Tics Scene Analysis

1. Introduction

Sound is around us and helps us understand the world. Environmental Sound Classifcation is about identifying the sounds we hear every day. Think about the sounds of cars, doors and machines. Of just guessing what they are we want to detect them more accurately. Lately people have become more interested in recognizing sounds. Devices are now better at listening to us. Machines are not just learning from text and images. Also from sound patterns. Real-life audio is not always easy to understand. There is a lot of noise. The signals can be weak. Earlier attempts at classification used predefined labels or handcrafted rules. Modern methods are different. They look at scenes more directly and try to find patterns. Even if the signals seem messy they can still carry information. One important thing is the quality of the sound features and the accuracy of the labels. Environmental audio has layers of information. Sometimes these layers are subtle. Sometimes they are distorted by background noise. So analyzing these signals is a interesting task.

Classic machine learning approaches are still used. Methods like Support Vector Machine and Linear Discriminant Analysis were used earlier. They helped separate categories with moderate success. Deep learning architectures are now taking over. They handle feature hierarchies better. Natural sounds do not follow fixed patterns. Traffic noise changes every second. Animals produce unpredictable signals. Sometimes audio signals weaken during transmission. Of strong and clean signals systems often deal with faint fragments. A common starting step is transforming waves into visual representations. Spectrograms are widely used for this purpose. Once converted into images audio signals become easier to analyze through convolution-based networks. In words sound becomes something that models can see.

* Corresponding author: Shashi Verma

Transfer learning is also an idea. A neural model trained for one task can adapt to another task without starting from scratch. Sometimes all layers are. In other cases only the final classification layers change. Inside a neural network feature formation begins in convolution layers. Each layer builds more abstract information from earlier ones. Features accumulate gradually of appearing suddenly. Along with this process audio signals can also be rearranged into feature forms.

One common representation is Mel-Frequency Cepstral Coefficients. This feature has been widely used in recognition tasks. Different experiments compare features to see which ones contribute more to classification performance. Model structure also plays a role. Not just the features themselves. The way they move through the network changes the final outcome. Layer after layer the system gradually improves its understanding of patterns. Comparing results between feature combinations becomes important here. Sometimes stacked features outperform ones that rely on a single representation.

By grouping signals rather than analyzing them individually models often respond differently even when the surrounding environment stays unchanged. Another question appears during experimentation. Does transfer learning actually improve Environmental Sound Classification performance? Previous knowledge might reduce training time. Still keep classification accuracy stable. Ultimately success depends on two things working together. The system must provide answers but also do so efficiently. Accuracy matters,. Speed matters too.

2. Literature Review

Earlier work in recognition relied heavily on manually designed features. Researchers extracted traits like Mel-Frequency Cepstral Coefficients then used classifiers such as Support Vector Machine or Gaussian Mixture Model to categorize sounds. These techniques worked to some extent. Capturing fine-grained changes in frequency or timing was still difficult gradually deep learning models started taking over this task. Convolutional neural networks began detecting patterns that older rule-based systems often missed. Of raw signals sound spectrograms served as the input turning audio into image-like data. Early research exploring neural networks for Environmental Sound Classification appeared in study [1]. There 2D spectrogram images were passed into the network like photographs. Results showed improvement over traditional methods. However the architecture itself remained relatively shallow.

Later work in [2] experimented with augmentation techniques. For example pitch shifting or time stretching modified the signals. These changes increased training diversity. As a result models became more robust while predicting unseen sounds. Researchers then began designing convolutional neural network structures. In study [3] the architecture arranged convolution layers in a ladder- configuration. This design combined features from levels simultaneously. The approach performed well on datasets including the ESC-10 dataset, ESC-50 dataset along with the K dataset.

Another design appeared in [4]. Of a single path the model used parallel convolution filters. Each branch focused on different frequency ranges at the time. Accuracy improved further though the system required computational resources. Some studies attempted to skip spectrogram preprocessing. Research in [5] trained networks on raw audio signals using one-dimensional convolutions. This method sometimes outperformed spectrogram-based systems. Still end-to-end learning of this kind typically demands labeled datasets, which are not always available.

Earlier attempts, such as those in [6] and [7] also used neural networks for environmental sound recognition. However their models stayed relatively simple. Were trained on broader audio collections like the AudioSet dataset. Because of this focus they did not always capture smaller acoustic scene details needed for Environmental Sound Classification tasks.

Temporal dynamics in signals became another area of interest. Study [9] introduced an architecture known as Convolutional Recurrent Neural Network. This model combines convolution layers with processing. It performs well when sounds change over time though computational cost increases noticeably. Another direction focused on efficiency[10]. Research in [11] proposed a dual-path convolutional neural network architecture designed for lightweight deployment. The model achieved performance while keeping the parameter count small. Without mechanisms for tracking sequential changes its ability to handle overlapping audio events remained limited.

Attention mechanisms also appeared in work. Studies [12]. [13] Added attention layers to convolutional neural network frameworks. These mechanisms allow networks to focus on time-frequency regions in spectrograms. Such methods improved robustness against background noise though they also increased memory consumption. A different idea explored in [14] involved using receptive fields simultaneously. This approach attempted to capture variations across scales. While useful in theory it also increased model complexity significantly.

Looking across these studies, a few patterns appear. Convolutional neural networks generally outperform earlier machine learning approaches for Environmental Sound Classification. Also, Mel-spectrograms continue to serve as the common input representation. Data augmentation tends to improve performance on samples. Multi-scale or attention-based architectures often show resilience in complex environments. They require more powerful hardware. Meanwhile lightweight models such as those in [15] achieve processing speeds but sometimes lose classification precision especially when temporal dynamics are ignored.

Because of this gap balancing accuracy with efficiency remains challenging. Some models achieve accuracy but become too large, for real-time use. Others run faster yet struggle to distinguish sound differences. This gap motivated the design used in the work. The goal is a convolutional neural network architecture that runs efficiently on practical devices while still capturing important acoustic patterns. In words reducing complexity should not mean sacrificing essential recognition ability.

3. Methodology

The process starts with clips from a dataset. We take five traits from those clips. First we. Prepare the raw audio. Then we start learning from trained models. Of building everything from scratch we use what already exists. Training helps us sound categories better. Each step helps the model get better at labeling sound clips.

3.1. Dataset

We use the K dataset. It has 8,732 short sound clips, each about four seconds long. These clips are labeled into ten sound classes. Examples include car horns, drilling sounds and police sirens. This dataset is often used to test systems that recognize sounds. It is used in studies on traffic monitoring, city noise measurement and urban acoustic mapping. The dataset is open and easy to handle making it a good starting point for audio recognition projects those focused on urban environments.

3.2. Preprocessing the Dataset

The sound clips in the UrbanSound8K dataset have durations. Some are longer. Some are shorter. For training a network we need fixed-length inputs. So we adjust every clip to a duration. We choose a length of 3 seconds, which's the average duration of the dataset. If a recording is longer we trim it. If it is shorter we repeat it until it fills the window.

This adjustment helps a lot. When all samples have the time span the model avoids confusion caused by uneven inputs. The learning stage becomes more stable. Uniform timing also helps the classifier judge sounds under conditions.

3.3. Feature Extraction

We extract audio features using the Librosa library. This library offers ways to analyze sound signals. From each recording we collect measurable properties, including pitch, intensity and rhythm patterns. After extraction the dataset turns into vectors representing sound behavior. These values then move forward into the learning pipeline.

3.3.1. MFCC

One way to represent sound is Mel-Frequency Cepstral Coefficients or MFCC. This method breaks the audio signal into frequency components. It uses the mel scale, which reflects how humans perceive pitch. The process starts by converting energy into the frequency domain. Then we apply scaling. Next we do a cosine transform that compresses the information into numbers called coefficients. MFCC features often perform better in speech or voice analysis. Their design focuses on how human ears interpret sound.

3.3.2. Log-Mel Spectrogram

Another representation we use is the Log-Mel spectrogram. In this method the frequency axis is transformed into the mel scale. This transformation makes the spacing between frequencies to human hearing perception. After the mel conversion we apply an operation to the amplitude values. The final output forms a spectrogram where both time and frequency information remain visible.

3.3.3. Chroma Feature

The chroma feature groups audio energy according to pitch classes. Of looking at every frequency separately similar pitches are combined into one category. The chroma representation usually contains twelve bins, each matching a semitone within an octave. These correspond to notes such as A, A#, B and C.

3.3.4. Spectral Texture

Texture describes how energy spreads across different frequency bands. It focuses on changes between frequency regions. In terms the feature tracks how energy shifts across the spectrum during the audio clip. These variations provide information about the structure or "texture" of the sound.

3.3.5. Stacked Features

After extraction we combine features generated through Librosa into groups. Of feeding only one feature type into the model we stack several together. During experimentation we create six feature combinations. Each combination tests how well a certain mix of features can represent sounds.

3.3.6. Convolutional Neural Network

The learning model we use is a Convolutional Neural Network or CNN. A CNN contains layers arranged in sequence. Typically the structure begins with an input layer, followed by convolution layers pooling operations and dense layers toward the end. What makes CNNs from older neural networks is the convolution operation itself. Of connecting every neuron to the entire input each neuron observes only a small region.

3.4. Architecture

The architecture of the CNN begins with a layer containing 32 filters with a 2 × 2 kernel size. A stride of 1 is used, followed by a ReLU activation function. After that max pooling reduces the dimensions of the feature map. A second convolution layer repeats a structure. Again 32 filters with 2 × 2 kernels are applied. ReLU activation follows, along with another max pooling step to compress the data further. Next comes a convolution layer containing 64 filters. These filters capture detailed patterns in the spectrogram representation. ReLU activation appears again along with pooling to shrink the feature maps. A fourth convolution layer continues the extraction process using 64 filters. Once this stage finishes the output feature maps are flattened into a one- vector. This flattened vector then passes into a connected layer, with 1024 neurons. Finally the network ends with a Softmax output layer, which generates probability scores for the 10 environmental sound classes. The predicted class corresponds to the category with the highest probability value.

4. Experimental Work

4.1. Datasets

To check how well the Environmental Sound Classification task works two datasets were used for testing. The first dataset is the K dataset. This dataset puts city noises into ten categories. These categories include air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren and street music. The UrbanSound8K dataset has 8,732 clips. Each clip is 4 seconds long. The audio sampling rate is 22.05 kHz. One issue with this dataset is that the number of clips in each category is not the same. Some categories have clips while others have fewer clips. The lengths of the clips also vary slightly. The second dataset used in this work is the DCASE-2017 Acoustic Scene Classification dataset. This dataset is organized differently. It has 4,680 files for development and 1,620 audio files for evaluation. Each clip in this dataset is 10 seconds long. Unlike the K dataset the categories in the DCASE-2017 Acoustic Scene Classification dataset are balanced. This means that each category has the same number of audio clips. The sampling rate for these recordings is 44.1 kHz. The DCASE-2017 Acoustic Scene Classification dataset has 15 categories.

These categories include beach, bus, cafe or restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train and tram. During the DCASE 2017 Challenge the rankings were based on how accurate the classifications were.

4.2. Evaluation Method and Criteria

The CNN model was. Evaluated using different setups for the two datasets. For the DCASE-2017 Acoustic Scene Classification dataset the development dataset was used for training. Then the evaluation dataset was used to measure performance. For the K dataset the situation was different. The dataset was divided randomly with 90% of the data used

for training and the rest used for evaluation. To further check the reliability of the classifications 10-fold cross-validation was used. This means that the dataset was split into ten parts. One part was used for testing and the other nine parts were used for training. This process was repeated until every part had been used for testing.

Several measures were used to evaluate the systems performance. These measures include accuracy, specificity, sensitivity, precision and the F-score. Each measure looks at an aspect of the classifier's behavior. The values for these measures were obtained from the confusion matrix. The evaluation metrics were then calculated using equations.

4.3. Experimental Setup and Results

As mentioned earlier a spectrogram representation was used to convert signals into time-frequency images. Of using the raw sound, the audio was first converted into spectrogram images. These images show how the frequency energy changes over time. Several parameters were chosen for the spectrogram process. The window size was set to 1024. The window type was Hamming. The overlap size was fixed at 256. The FFT size was set to 3000. These values were chosen because they gave stable results in early trials.

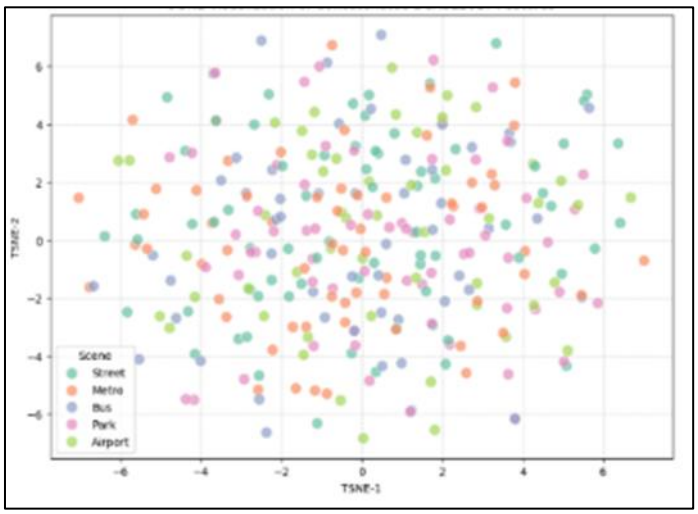

Figure 1 The scatter plot of the concatenated features for the DCASE2017-ASC dataset

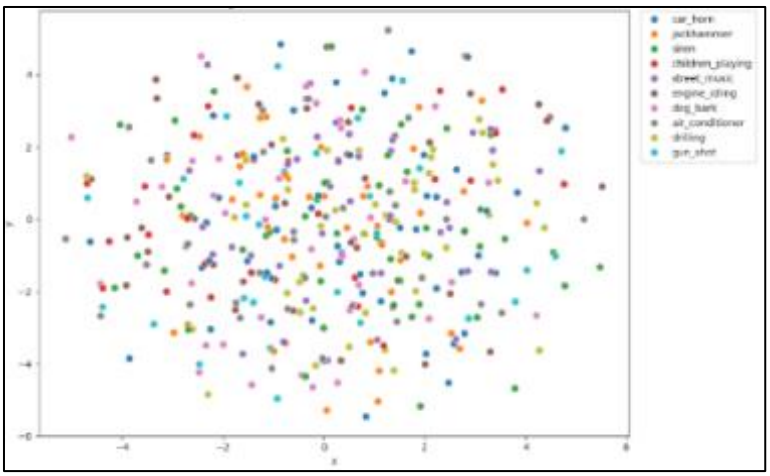

Figure 2 The scatter plot of the concatenated features for the UrbanSound8K dataset

Table 1 The effect of input size on the performance of the proposed method

The input size	UrbanSound8k (Acc%)	DCASE-2017 ASC (Acc%)
20×20	72.10%	79.60%
50×50	79.20%	88.10%
100×100	86.70%	96.23%
200×200	83.10%	92.10%

The spectrogram images were initially 875 × 656 × 3. However, the CNN model required an input. So, the images were resized to 100 × 100 × 3 before being fed into the network. Once resized the spectrogram images were fed directly into the CNN model for learning. The CNN layers had parameters. For example, the first convolution layer used a 3 × 3 filter size with 8 filters. The max-pooling layers had a pixel block size of 2 × 2 and the stride value was also 2 × 2. This step reduced the dimensions while keeping the important feature patterns.

Table 2 Comparison for the DCASE-2017 ASC to the proposed method with the other CNN models and classifiers

Classifiers	**AlexNet**	**VGG16**	**VGG19**	**ResNet-50**	**Proposed CNN**
Fine Tree	61.56%	62.32%	61.44%	60.10%	67.20%
KNN	69.18%	70.78%	67.16%	67.54%	79%
SVM	80.35%	81.20%	79.34%	79.10%	85.40%
Boosted Trees Ensembles	5 5.92%	58.56%	57.30%	56.10%	68.40%
Bagged Trees Ensembles	55.30%	56.18%	55.68%	55.45%	75.40%
Subspace Discriminant Ensembles	68.20%	69.26%	67.96%	66.12%	8.20%
Subspace KNN Ensembles	76.15%	79.34%	75.82%	74.53%	6.23%

During training the CNN used a -batch size of 128 and the initial learning rate was set to 0.005. The Adam optimizer handled parameter updates during learning. The deep features extracted from the connected layers were also studied. The first connected layer produced features with a dimension of 500 and the second fully connected layer produced features with a dimension of 450. These features were visualized using scatter plots. Most of the hyperparameters were not chosen randomly. They were adjusted times during experiments and the final configuration was selected because it produced the highest classification accuracy.

Table 3 Comparison for the Urbansound8k to the proposed method with the other CNN models and classifiers

Classifiers	**AlexNet**	**VGG16**	**VGG19**	**ResNet-50**	**Proposed CNN**
Fine Tree	35.7%	38.10%	34.50%	35.20%	44.10%
KNN	70.35%	71.35%	69.80%	70.80%	84.20%
SVM	76.20%	77.10%	75.00%	75.95%	78.00%
Boosted Trees Ensembles	44.60%	45.80%	42.90%	43.85%	50.00%
Bagged Trees Ensembles	63.10%	64.55%	62.10%	63.20%	67.80%
Subspace Discriminant Ensembles	60.50%	61.60%	59.20%	60.65%	65.40%
Subspace KNN Ensembles	71.35%	72.35%	70.65%	70.90%	86.70%

Table 4 The scores of the other performance criteria for the DCASE-2017 ASC dataset

Classes	Sensitivity	Specificity	Precision	F-score
Beach	0.9722	0.9980	0.9722	0.9722
Bus	0.9722	0.9987	0.9813	0.9767
Cafe/Restaurnt	0.9630	0.9947	0.9286	0.9455
Car	0.9630	0.9987	0.9811	0.9720
City center	0.9630	0.9954	0.9369	0.9498
Forest Path	0.9630	0.9967	0.9541	0.9585
Grocery store	0.9722	0.9967	0.9545	0.9633
Home	0.9259	0.994	0.9615	0.9434
Library	0.9352	0.9947	0.9266	0.9309

Table 5 The scores of the other performance criteria for the Urbansound8k dataset

Classes	Sensitivity	Specificity	Precision	F-score
Air conditioner	0.8810	0.9899	0.9024	0.8916
Car horn	0.8506	0.9771	0.8043	0.8268
Children playing	0.8090	0.9847	0.8571	0.8324
Dog bark	0.9022	0.9885	0.9022	0.9022
Drilling	0.8889	0.9819	0.8627	0.8756
Engine idling	0.8617	0.9846	0.8710	0.8663
Gun shot	0.8902	0.9848	0.8588	0.8743
Jackhammer	0.8095	0.9797	0.8095	0.8095
Siren	0.9136	0.9950	0.9487	0.9308
Street music	0.8659	0.9861	0.8659	0.8659

The CNN input size was also tested with values. The model worked best when the input dimension was 100 × 100 which became the configuration used in the system. The proposed CNN model performed better than well-known deep architectures, including AlexNet, VGG network and ResNet-50. For the DCASE-2017 Acoustic Scene Classification dataset the model achieved a classification accuracy of 96.23% when paired with Subspace KNN Ensemble classifiers. This result is higher than traditional classifiers. A similar trend was seen for the K dataset. The proposed system reached 86.70% accuracy, which's higher than several existing approaches. Besides accuracy class-wise metrics were also examined. The results suggest that the model maintains precision across several acoustic scenes.

For example, categories like "Metro station" and "Park" produced perfect classification scores. This shows that the CNN architecture, when combined with ensemble-based classifiers can handle sound environments effectively. Overall, the results suggest that the proposed approach adapts well to environmental sound recognition tasks.

5. Conclusion

This study presented a CNN-based model designed for sound classifica-tion. The architecture contains three convolution layers, three max-pooling layers and normalization stages. The network also includes three connected layers, with Softmax and classification layers at the end. The CNN structure shows ability in identifying sound categories. Another advantage is the speed of the model. Training and prediction are relative-ly efficient. During testing the connected layers

were used in a different way. Of relying on the Softmax classifier deep features extracted from these layers were used as inputs for the K-Nearest Neighbors classifier. This classifier was chosen because it works reliably with datasets of this type.

The experiments were carried out using the DCASE-2017 Acoustic Scene Classification dataset and the UrbanSound8K dataset. The main focus was to measure classification accuracy. The results show that the proposed CNN architecture, with deep feature extraction performs well in recognizing environmental sounds. The comparison with existing state-of-the-art methods also suggests improvement, over previous ap-proaches.

In terms the CNN model and the deep features produced by it work effectively as a combined system. Environmental sound patterns can be recognized reliably using this setup. The Environmental Sound Classifi-cation task is important. The proposed approach can be used to improve the accuracy of sound recognition systems. The Environmental Sound Classification task is a problem and the proposed approach can be used to develop more accurate sound recognition systems.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] K. J. Piczak, "Environmental sound classification with convolutional neural networks," Proc. IEEE MLSP, pp. 1–6, 2015.

[2] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," IEEE Signal Process. Lett., vol. 24, no. 3, pp. 279–283, 2017.

[3] F. Demir, M. Turkoglu, M. Aslan, and A. Sengur, "A new pyramidal concatenated CNN approach for environmental sound classification," Applied Acoustics, vol. 170, pp. 107520, 2020.

[4] Y. Su, J. Liang, Q. Guo, and Y. Qian, "Multi-branch CNN for environmental sound classification," Applied Acoustics, vol. 156, pp. 135–145, 2019.

[5] J. Dai, Y. Zhang, and H. Li, "Raw waveform-based 1D CNN for environmental sound classification," Proc. IEEE ICASSP, pp. 126–130, 2020.

[6] S. Hershey et al., "CNN architectures for large-scale audio classification," Proc. IEEE ICASSP, pp. 131–135, 2017.

[7] S. Kumar, S. Sahu, and R. Rajan, "Environmental sound recognition using deep CNNs trained on AudioSet," IEEE Access, vol. 8, pp. 116–128, 2020.

[8] B. McFee et al., "Librosa: Audio and music signal analysis in Pyth on," Proc. SciPy, pp. 18–25, 2015.

[9] E. Çakir, T. Heittola, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," IEEE/ACM Trans. Audio Speech Lang. Process., vol. 25, no. 6, pp. 1291–1303, 2017.

[10] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and Mel-spectrograms for music and audio classification," IEEE ICASSP, pp. 115–119, 2015.

[11] Z. Chen and Y. Zhang, "Lightweight dual-path CNN for real-time environmental sound classification," IEEE Access, vol. 9, pp. 112548–112560, 2021.

[12] A. Mesaros, T. Heittola, and T. Virtanen, "Acoustic scene classification boosted with attention mechanisms," DCASE Workshop, pp. 10–14, 2020.

[13] Y. Gong, C. Chung, and J. Glass, "AST: Audio spectrogram transformer," Proc. Interspeech, pp. 571–575, 2021.

[14] S. Zhang, Y. Wang, and X. Li, "Multi-receptive-field CNN for robust environmental sound classification," Sensors, vol. 20, no. 11, pp. 3172, 2020.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017.