SOUND EVENT CLASSIFICATION

Chiara Auriemma, Francesca Benesso, Anna Fusari, Filippo Marri

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano
Piazza Leonardo Da Vinci 32, 20122 Milano, Italy
[chiara.auriemma,francesca1.benesso]@mail.polimi.it
[anna.fusari,filippo.marri]@mail.polimi.it

ABSTRACT

Sound Event Classification (SEC) has become an important task in the field of audio processing, with applications ranging from environmental monitoring to human-computer interaction. Aim of this project is to develop a sound event classification system based on a Convolutional Neural Network (CNN) architecture training it on the ESC-50 dataset. At the end, the performances of the model are compared with some state-of-the-art models (QUALE?). [Da finire, deve essere una sorta di riassunto del progetto, con le tecniche utilizzate e i risultati ottenuti].

***Index Terms*—** Sound Event Classification, Convolutional Neural Network, ESC-50 dataset, performance limitations

1. INTRODUCTION

1.1. Background

Sound Event Classification (SEC) is a task that involves classification of specific sound events within an audio signal. This task has gained significant attention in recent years due to its wide range of applications, including environmental monitoring [1], human-computer interaction [2], and multimedia content analysis [3]. The goal of SEC is to accurately classify sound events in real-time or from pre-recorded audio data. The process of SEC typically involves several steps, including feature extraction, model training, and evaluation [4].

Commonly used features include Mel-frequency cepstral coefficients (MFCCs), log-spectrograms, and log-mel spectrograms. Model training involves using labeled audio data to train a machine learning model to classify sound events. Various machine learning algorithms can be used for SEC, including support vector machines (SVMs), decision trees, and deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [5]. The choice of algorithm depends on the complexity of the task and the available data.

Evaluation of the SEC system is typically done using standard metrics such as accuracy, precision, recall, and F1-score.

1.2. Literature review for project development

One of the first attempts in SEC was made by Piczak [6] in 2015, in which the sound is split in several segments and each segment is processed individually using a Convolutional Neural Network (CNN). At the end a series of Dense Layer is used to classify the segments. Final predictions for a sound are generated using either a majority-voting scheme or by taking into account the probabilities predicted for each segment.

2. METHODOLOGY

As a baseline of this project we chose a simple 2D Convolutional Neural Network (CNN) architecture based on the CONV2D model of laboratory number 4. We chose a 2D Convolutional network since we have, as input, a log mel-spectrogram. In turn, this kind of input has been chosen in order to feed the network with temporal-frequency correalated data, approach widely followed in the literature. However, the results obtained with this architecture were not satisfactory, so we decided to improve the model by adding some layers and using a more complex architecture. After several attempts with Recurrent Convolutional Neural Networks (CRNN), we opted for a multi-branch convolutional neural network (MBCNN) architecture inspired by the work of Enes Furkan Örnek [7, 8].

2.1. Feature extraction

As hinted before, the neural network is fed with a log mel-spectrogram, computed by a specific function that takes as input the signal preserving its original sample rate. The parameters are the following:

- **N fft**: 1024 samples
- **Hop size**: 512 samples
- **Number of mel bands**: 128 (predefined parameter)

The output is a 2D array of shape (128, 1723) representing how energy in different Mel frequency bands evolves over time.

2.2. Model description

The model code can be found at the following link:
`https://github.com/ChiaraAuriemma/Sound-Event-Classification`.

Our implementation utilizes a bi-dimensional Convolutional Neural Network (CNN) with four parallel branches, each extracting different levels of information from the input data using specifically calibrated kernel sizes. We highlight that our model is working in a peculiar way with the 2D layers since one dimension is, layer-by-layer, alternatively imposed equal to 1. With this method, each branch alternates between horizontal and vertical filter orientations, enabling efficient extraction of localized patterns across both time and frequency domains. The net is built following a cellular approach. Each cell is composed by a Conv2D layer followed by batch normalization and ReLU activationin in order to ensure training stability and non-linearity as its shown in figure 1. [Commento: after each Conv2D layer, we apply batch normalization and ReLU activationin order to ensure training stability and non-linearity.]

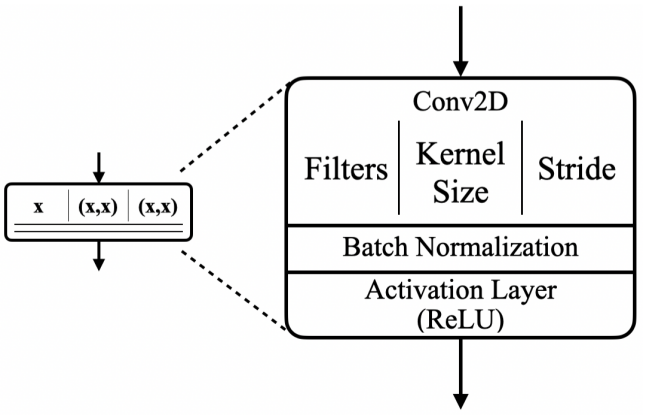

Figure 1: Structure of a cell of the architecture.

After processing through these branches, the outputs are merged via element-wise addition, followed by additional convolutional layers with larger filter depths to further abstract the combined feature maps. A global average pooling layer condenses the spatial information, and the network concludes with a dense softmax layer that outputs class probabilities over 50 categories. The entire architecture is depicted in Figure 2.

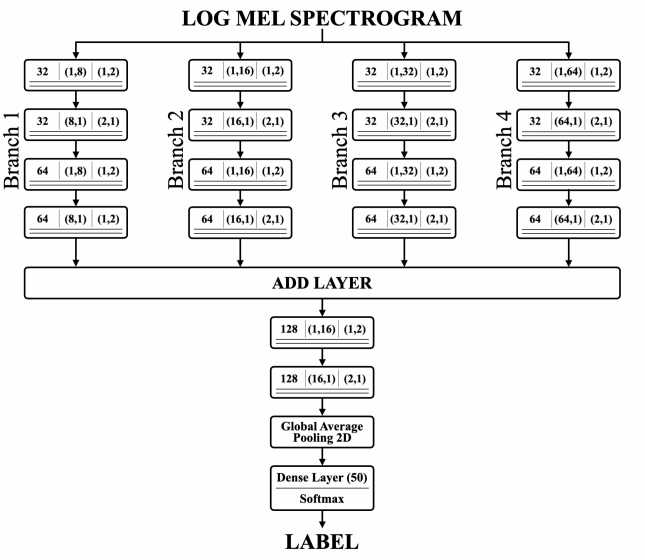

Figure 2: Architecture of the multi-branch convolutional neural network.

The model is optimized using the Adam optimizer with AMSGrad and learning rate equal to 0.00005, and trained using sparse categorical cross-entropy loss.

The theoretical justification for our filter size selection derives from signal processing principles in audio analysis. Smaller filters (1×8) effectively capture localized patterns and high-frequency components, while larger filters (1×64) extract broader patterns and low-frequency information.

Not only mathematical reasons justify our approach, but also biological ones. In facts, the human auditory system employs a wave bank filter mechanism to capture specific frequency components in audio signals, with different regions of the basilar membrane responding to distinct frequency ranges. By incorporating parallel branches with carefully selected filter sizes, the network similarly detects patterns within specific frequency ranges, enabling multi-scale feature representation.

3. EVALUATION

3.1. Dataset analysis and preprocessing

The dataset used for this project is the well-known ESC-50 dataset [9], which contains 2000 labeled isolated eviromental sound events from 50 different classes, with each class containing 40 samples. Each sound of the dataset is a mono recording available in WAV format (Ogg Vorbis compress at 192 kbit/s) with a sample rate of 44.1 kHz and a bit depth of 16 bits. Clips in this dataset have been manually extracted from public field recordings gathered by the Freesound project [10]. The resulting dataset is available under a Creative Com- mons non-commercial license through the Harvard Dataverse project [9].

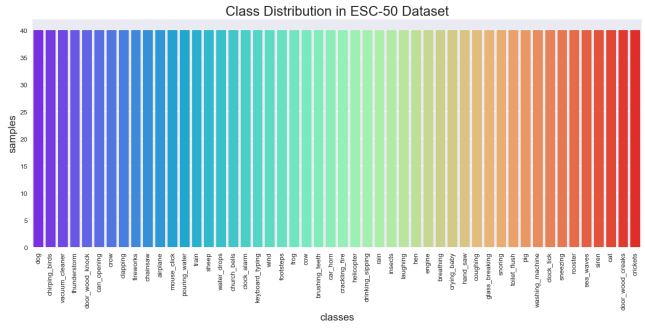

Figure 3: Graphical demontration of the balance of the dataset used.

According to the analysis that will be done on the results inspired by *Attention based Convolutional Recurrent Neural Network for Environmental Sound Classification* [11], the type of soound events in the dataset can be divided into three main categories:

- **Transient sounds**: this category includes sounds that have a short duration and are characterized by a sudden onset, such as the one of a glass breaking, a thunderstorm, or a firework.
- **Continous sounds**: this category includes sounds that have a longer duration and are characterized by a continuous or sustained sound, such as the one a vacuum cleaner, a car engine running, or pouring water.
- **Intermittent sounds**: this category includes sounds that have a periodic or irregular pattern, such as the one a dog barking, a bird chirping, or a man coughing.

Some examples are reported in Fig. 4, where we can see the power spectrogram of a transient sound (glass breaking), a continuous sound (vacuum cleaner), and an intermittent sound (dog barking).

According to what is reported on the paper in which the ESC-50 dataset is presented [9], we higlight how some sounds are more difficult to classify than others, such as the sounds of a washing machine, an helicopter, or an engine due to their similar spectrograms. If we look to Fig. 5, we can see how the three spectrograms are extremely similar.

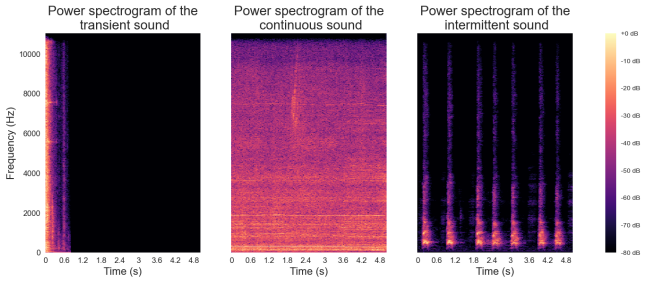

Figure 4: Power spectrogram of a transient, a continous and an intermittent sound.

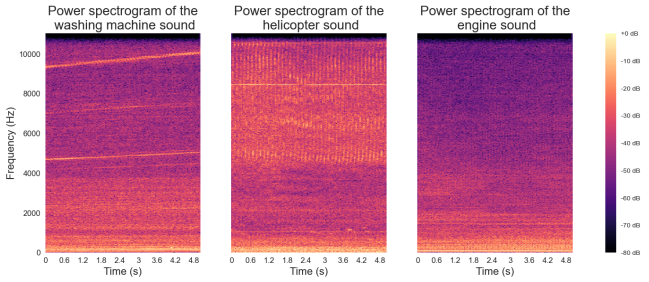

Figure 5: Power spectrogram of the sound produced by a washing machine, an helicopter and an engine.

This misclassification happens not only for machines, but for humans too. This will be taken into account in the results section, where we will see how the model performs on different classes of sounds.

It is also important to note that, even though we consider the same class, the variability of the sounds is very high, as we can see by comparing the spectrograms of three different samples of the *dog barking* class. As we can see in Fig. 6, the three spectrograms are very different from each other. According to the classification defined at the beginngin, the first one can be considered an impulsive sound, the second one a continuous sound, and the third one an intermittent sound. This means that, given the nature of the sound in this class, the onset information is not so useful.

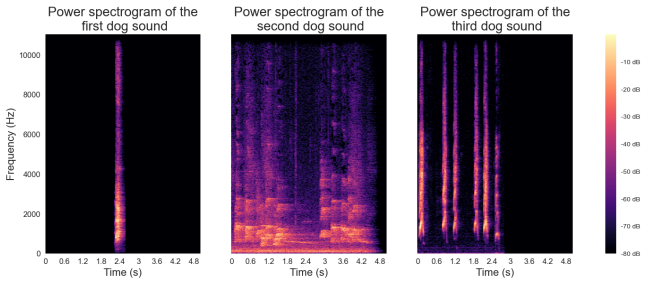

Figure 6: Power spectrogram of three different sounds belonging to the same *dog barking* class.

Furthermore, we underline how some of the ambiental sounds, like the one of the wind, have no univoque structure: by breaking down their spectrograms in their harmonic and percussive components as it is done in Fig. 6, it is evident that the difference it is not so clear since the two plots are almost equal.

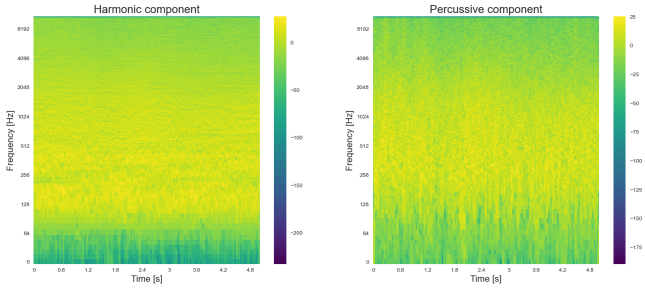

Figure 7: Harmonic and percussive decomposition of the wind blowing sound.

This fast analysis of the dataset has been done to understand the limitations of the model and the difficulties that it could encounter during the training phase.

A part of the dataset, 200 elements (10% of the total), has been separated for testing. The remaining samples have been splitted according to the stratifiedkfold module of sklearn [12] in a training and a validation set.

Drawing inspiration by the Salamon and Bello paper [13], five different techniques have been implemented to process the training set:

- **Time Stretching (TS)**: the audio signal is stretched or compressed in time by a random factor within a specified range. A last boolean parameter crop the processed audio to the original length, so that the model can be trained on the same length of the original audio.
- **Pitch Shifting (PS)**: the audio signal is shifted in pitch by a random factor within a specified range.
- **Background Noise (BN)**: a Gaussian noise is added to the audio signal with a specified SNR range.
- **Dynamic Range Compression (DRC)**: the dynamic range of the audio signal is compressed from a certain threshold with a specified ratio, attack time, and release time. [Commento: aggiungere che è fatto con la funzione di spotify]
- **Convolution with Impulse Responses (CIR)**: the audio signal is convolved with the *MIT Acoustical Reverberation Scene Statistics Survey* dataset of impulse response [14] to simulate different acoustic environments.

For all the results presented in this paper, the training dataset has been preprocessed using the following parameters:

- TS: factor between 0.8 and 1.25
- PS: factor between -5 and 5 semitones
- BN: SNR between 5 and 40 dB
- DRC: threshold of -20 dB, ratio of 4:1, attack time of 10 ms, release time of 100 ms
- CIR: active

3.2. Evaluation metrics

The evaluation of the sound event classification system is performed using several metrics to assess its performance. Firstly, the test ac-

curacy, the classification reports and the confusion matrix are computed to evaluate the overall performance of the model.

To evaluate our model, we use 5 folds cross-validation, which allows us to obtain a more robust estimate of the model's performance. Furthermore, we ensured that there is no data leakage between training and validation sets by doing data augmentation only on the training set after the splitting.

The training of the models have been performed on the A100 GPU by Google Colab.

4. RESULTS

4.1. Main model results

The results of the training of the model for each fold are reported in Figures 8, 9, 10, 11 and 12.

Figure 8: Result of the training for the zero fold.

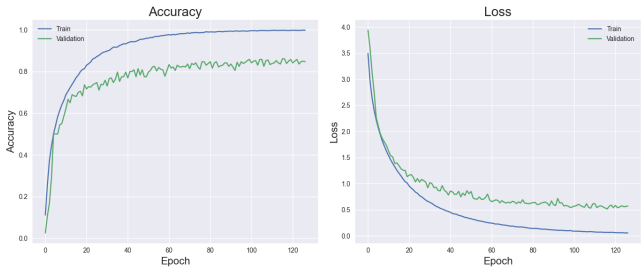

Figure 9: Result of the training for the first fold.

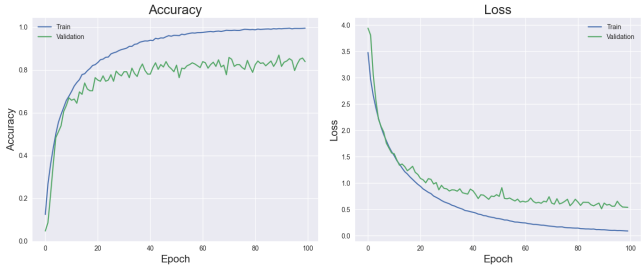

Figure 10: Result of the training for the second fold.

The average test accuracy over the 5 folds reached by the model is 86% with an average loss of 0.63. The performnce for each fold is reported in Table 1.

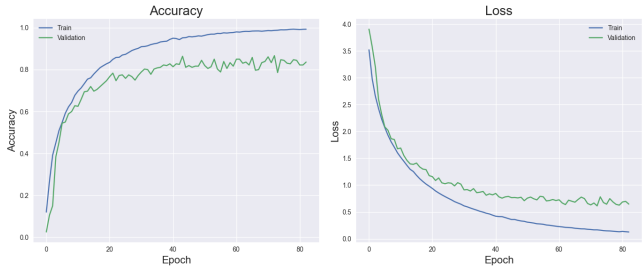

Figure 11: Result of the training for the third fold.

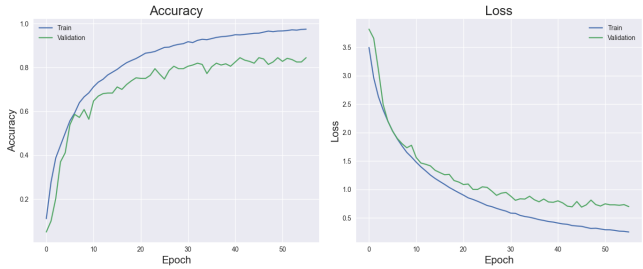

Figure 12: Result of the training for the fourth fold.

In Fig 13 and Fig 14 the validation and the test confusion matrixes of the model with the best accuracy (88%) are reported. The labeling of the classes is reported in the appendix.

The reoprts are reported in Table 2 for the validation and in Table 3 for the test.

We highlight here how precision and recall are well balanced meaning that the model is not biased towards a specific class.
By looking at the test confusion matrix, we can see that the model is able to classify correctly most of the sounds, with some exceptions. One of them is the *wind* class that is misclassified 3 times out of 4: one time classified as *train*, one as *airplane* and as *sea waves*. As we can notice, all the three sounds are continuous and noisy sounds, so it is not surprising that the model has some difficulties in classifying them. This match with what we assumed by looking at Fig. 5: sounds with similar spectrograms are more difficult to classify. It is interesting to notice that the recall of these three classes is not high even in the results of the experiment conducted by Piczak [9] in which a group oh humnas listened to the sounds and classified them. This leads us to think that since the model is inspired by the human auditory system, it is not surprising that it has the same difficulties in classifying the sounds.

Another class that is misclassified is the *coughing* class, which is confused with the *sneezing* and *water drops* class. We underline how all these sounds are intermittent sounds and that they have similar spectrograms.

For what it concerns false positive, these are the errors made by the model: the sound of the *airplane* and *helicopter* has been classified as *wind*, the sound of *sneezing* and *laughing* as *caughing* and for two times the *glass breaking* sound has been misclassified as *can opening*. We can justify these results by claiming the same motivation we exposed before: similar spectrograms lead to misclassification.

[Commento: manca il confronto con la baseline]

Table 1: Results of the main model for each fold.

Fold	Test Accuracy	Test Loss
0	0.85	0.65
1	0.87	0.60
2	0.88	0.58
3	0.86	0.62
4	0.85	0.64
Average	0.86	0.63
Standard Deviation	0.024	

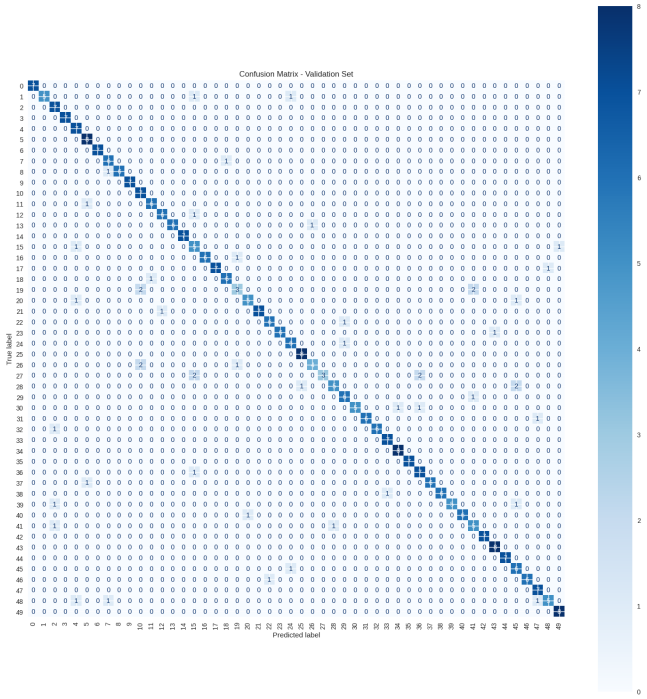

Figure 13: Confusion Matrix of the validation set.

4.2. Comparison: main model with different inputs

During the training of the main model, we tried to use different inputs to see how they affect the performance of the model. At the end, we notice that all the results are similar, with a slight improvement in the accuracy when using the log mel-spectrogram as input. They are reported in Table 4.

We notice how MFCCs are the worst input. This could seems strange thinking that MFFCs are the most processed input we can use among the ones tested. MFCCs could be effective with classical machine learning algorithms, where the features were extracted from the raw data and then used to train the model. However, with the improvements made by neural networks, is no longer necessary perform classical feature engineering since the networks are able to extract better features by themselves. For what we just said, we can undesrtand why even the simple STFT works better than MFCCs.

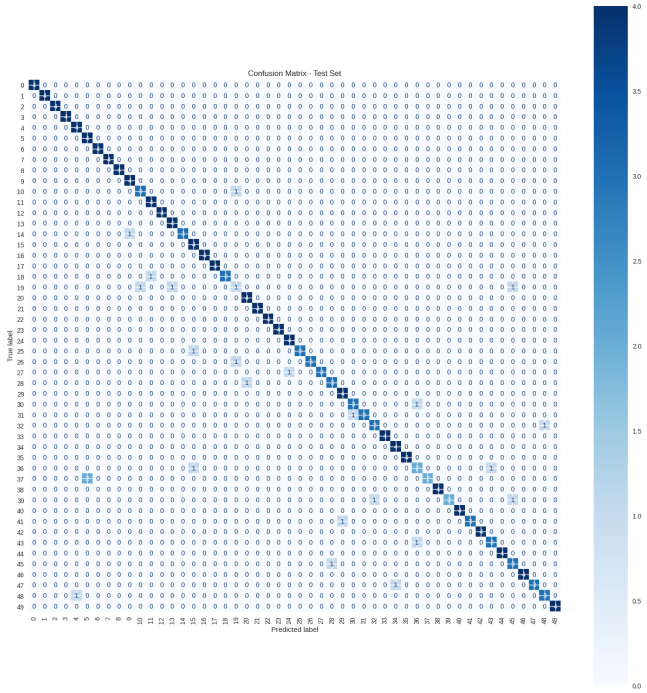

Figure 14: Confusion Matrix of the test set.

4.3. Comparison: main model with

5. ACKNOWLEDGMENTS

This work was supported by the Politecnico di Milano, within the framework of the Selected Topics in Music and Acoustic Engineering Course 2025. A special thanks goes to the course instructor, Prof. JULIO JOSÉ CARABIAS ORTI, for being one of the brightest stars in the sky of artificial intelligence. Last but not least, we would like to thank our wallet: without those 24 euros, we would not have been able to run anything. Thank you for your support, we are grateful for your generosity. Grazie a tutt coloro che ci hanno supportato e sopportato.

6. REFERENCES

[1] R. Narasimhan, X. Z. Fern, and R. Raich, "Simultaneous segmentation and classification of bird song using cnn," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 146–150.

[2] S. Cunningham, H. Ridley, J. Weinel, and R. Picking, "Supervised machine learning for audio emotion recognition," *Personal and Ubiquitous Computing*, vol. 25, no. 4, pp. 637–650, 2021. [Online]. Available: https://doi.org/10.1007/s00779-020-01389-0

[3] A. Kumar and B. Raj, "Weakly supervised scalable audio content analysis," 2016. [Online]. Available: https://arxiv.org/abs/1606.03664

[4] S. Padmaja and N. Sharmila Banu, "A systematic literature review on sound event detection and classification," in *2025 5th International Conference on Trends in Material Science and Inventive Materials (ICTMIM)*, 2025, pp. 1580–1587.

[5] A. Bansal and N. K. Garg, "Environmental sound classification: A descriptive review of the literature," *Intelligent Systems with Applications*, vol. 16, p. 200115, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667305322000539

[6] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6.

[7] E. F. Örnek, "Audio classification using cnn on esc-50 dataset," https://github.com/sweat0198/audio_classification_CNN_ESC-50, 2023, accessed: 2025-06-09.

[8] S. A. Latifi, H. Ghassemian, and M. Imani, "Classification of heart sounds using multi-branch deep convolutional network and lstm-cnn," 2025. [Online]. Available: https://arxiv.org/abs/2407.10689

[9] K. J. Piczak, "Esc: Dataset for environmental sound classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM, 2015, pp. 1015–1018.

[10] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: An open dataset of human-labeled sound events," *arXiv preprint arXiv:2010.00475*, 2020, accessed: 2025-06-09. [Online]. Available: https://arxiv.org/abs/2010.00475

[11] Z. Zhang, S. Xu, T. Qiao, S. Zhang, and S. Cao, "Attention based convolutional recurrent neural network for environmental sound classification," 2019. [Online]. Available: https://arxiv.org/abs/1907.02230

[12] S. learn developers, "Stratifiedkfold — scikit-learn 1.5.2 documentation," 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

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

[14] J. Traer and J. H. McDermott, "Statistics of natural reverberation enable perceptual separation of sound and space," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. E7856–E7865, 2016. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.1612524113

7. APPENDIX - SOUND CLASSES

- 0: dog
- 1: chirping birds
- 2: vacuum cleaner
- 3: thunderstorm
- 4: door wood knock
- 5: can opening
- 6: crow
- 7: clapping
- 8: fireworks
- 9: chainsaw
- 10: airplane
- 11: mouse click
- 12: pouring water
- 13: train
- 14: sheep
- 15: water drops
- 16: church bells
- 17: clock alarm
- 18: keyboard typing
- 19: wind
- 20: footsteps
- 21: frog
- 22: cow
- 23: brushing teeth
- 24: car horn
- 25: crackling fire
- 26: helicopter
- 27: drinking sipping
- 28: rain
- 29: insects
- 30: laughing
- 31: hen
- 32: engine
- 33: breathing
- 34: crying baby
- 35: hand saw
- 36: coughing
- 37: glass breaking
- 38: snoring
- 39: toilet flush
- 40: pig
- 41: washing machine
- 42: clock tick
- 43: sneezing
- 44: rooster
- 45: sea waves
- 46: siren
- 47: cat
- 48: door wood creaks
- 49: crickets

Table 2: Classification report on the validation set related to the model with best test accuracy (fold 4).

Class	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	7
1	1.00	0.71	0.83	7
2	0.70	1.00	0.82	7
3	1.00	1.00	1.00	7
4	0.70	1.00	0.82	7
5	0.80	1.00	0.89	8
6	1.00	1.00	1.00	7
7	0.75	0.86	0.80	7
8	1.00	0.86	0.92	7
9	1.00	1.00	1.00	7
10	0.64	1.00	0.78	7
11	0.86	0.86	0.86	7
12	0.86	0.86	0.86	7
13	1.00	0.86	0.92	7
14	1.00	1.00	1.00	7
15	0.50	0.71	0.59	7
16	1.00	0.86	0.92	7
17	1.00	0.88	0.93	8
18	0.86	0.86	0.86	7
19	0.60	0.43	0.50	7
20	0.83	0.71	0.77	7
21	1.00	0.88	0.93	8
22	0.86	0.86	0.86	7
23	1.00	0.86	0.92	7
24	0.75	0.86	0.80	7
25	0.89	1.00	0.94	8
26	0.80	0.57	0.67	7
27	1.00	0.43	0.60	7
28	0.83	0.62	0.71	8
29	0.75	0.86	0.80	7
30	1.00	0.71	0.83	7
31	1.00	0.86	0.92	7
32	1.00	0.86	0.92	7
33	0.88	1.00	0.93	7
34	0.89	1.00	0.94	8
35	1.00	1.00	1.00	7
36	0.70	0.88	0.78	8
37	1.00	0.86	0.92	7
38	1.00	0.86	0.92	7
39	1.00	0.71	0.83	7
40	1.00	0.86	0.92	7
41	0.62	0.71	0.67	7
42	1.00	1.00	1.00	7
43	0.89	1.00	0.94	8
44	1.00	1.00	1.00	7
45	0.60	0.86	0.71	7
46	1.00	0.86	0.92	7
47	0.78	1.00	0.88	7
48	0.83	0.62	0.71	8
49	0.89	1.00	0.94	8
Accuracy			0.86	360
Marco avg	0.88	0.86	0.86	360
Weighted avg	0.88	0.86	0.86	360

Table 3: Classification report on the test set related to the model with best test accuracy (fold 4).

Class	Precision	Recall	f1-score	Support
0	1.00	1.00	1.00	4
1	1.00	1.00	1.00	4
2	1.00	1.00	1.00	4
3	1.00	1.00	1.00	4
4	0.80	1.00	0.89	4
5	0.67	1.00	0.80	4
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	4
8	1.00	1.00	1.00	4
9	0.80	1.00	0.89	4
10	0.75	0.75	0.75	4
11	0.80	1.00	0.89	4
12	1.00	1.00	1.00	4
13	0.80	1.00	0.89	4
14	1.00	0.75	0.86	4
15	0.67	1.00	0.80	4
16	1.00	1.00	1.00	4
17	1.00	1.00	1.00	4
18	1.00	0.75	0.86	4
19	0.33	0.25	0.29	4
20	0.80	1.00	0.89	4
21	1.00	1.00	1.00	4
22	1.00	1.00	1.00	4
23	1.00	1.00	1.00	4
24	0.80	1.00	0.89	4
25	1.00	0.75	0.86	4
26	1.00	0.75	0.86	4
27	1.00	0.75	0.86	4
28	0.75	0.75	0.75	4
29	0.80	1.00	0.89	4
30	0.75	0.75	0.75	4
31	1.00	0.75	0.86	4
32	0.75	0.75	0.75	4
33	1.00	1.00	1.00	4
34	0.80	1.00	0.89	4
35	1.00	1.00	1.00	4
36	0.50	0.50	0.50	4
37	1.00	0.50	0.67	4
38	1.00	1.00	1.00	4
39	1.00	0.50	0.67	4
40	1.00	1.00	1.00	4
41	1.00	0.75	0.86	4
42	1.00	1.00	1.00	4
43	0.75	0.75	0.75	4
44	1.00	1.00	1.00	4
45	0.60	0.75	0.67	4
46	1.00	1.00	1.00	4
47	1.00	0.75	0.86	4
48	0.75	0.75	0.75	4
49	1.00	1.00	1.00	4
Accuracy			0.88	200
Macro avg	0.89	0.88	0.88	200
Weighted avg	0.89	0.88	0.88	200

Table 4: Performances for different inputs.

Input	Test Accuracy	Test Loss
log mel-spectrogram	0.86	0.63
log STFT	0.83	0.71
MFCCs	0.78	0.78