Environmental Sound Classification With 1D CNN

Nakul Agrawal
231IT040
Dept of Information Technology
National Institute of Technology Karnataka

Aditya Bhaskar
231IT004
Dept of Information Technology
National Institute of Technology Karnataka

***Abstract*—This project develops an end-to-end environmental sound classification system using a 1D Convolutional Neural Network (CNN) model, trained on the UltraSound8K dataset, which contains over 8,000 samples spanning diverse environmental sounds. The approach includes audio signal preprocessing with techniques such as Mel-Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), and Gammatone filterbanks, alongside the application of various filters—high-pass, low-pass, Kalman, and Wiener filters—to evaluate their effects on classification performance. The primary objectives are to assess the effectiveness of these preprocessing and filtering techniques in enhancing sound classification accuracy. Evaluation metrics including accuracy, precision, recall, and F1 score are used to validate model performance, with comparisons to baseline methods to ensure robust generalization. The findings are expected to yield a high-performing model and valuable insights into the contributions of different feature extraction and filtering techniques in environmental sound classification.**

***Index Terms*—Environmental sound classification, 1D CNN, UltraSound8K, MFCC, STFT, Kalman filter, feature extraction.**

I. Introduction

Classifying environmental sounds is essential for a variety of applications, ranging from urban monitoring and smart city infrastructure to security systems. For example, detecting and classifying sounds like car honks, glass breaking, or rainfall can significantly enhance automated responses. These responses could include alerting authorities in emergency situations, such as when a glass break is detected in a security system, or adapting virtual assistant responses based on the sound of rainfall to set the appropriate environment (e.g., turning off outdoor sprinklers). Such systems could also be integrated into smart city applications, where the ability to identify and respond to environmental sounds—like traffic noise or emergency vehicle sirens—can contribute to more efficient and responsive urban management. However, distinguishing between similar noises, such as traffic sounds, human activities (e.g., talking or walking), and natural sounds (e.g., wind or birds), remains a significant challenge. This is due to the inherent complexity and variability of audio signals in dynamic, real-world environments, where background noise, overlapping sounds, and changes in the acoustics of different locations can all complicate accurate classification.

A crucial step in Environmental Sound Classification (ESC) is feature extraction, where raw audio signals are processed to highlight important sound characteristics. Common techniques include Mel-Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), and Gammatone filterbanks. MFCC captures frequency-based features that differentiate between similar sounds, such as various animal calls. STFT provides time-frequency analysis for dynamic sounds like sirens, while Gammatone filterbanks simulate human auditory perception, aiding in recognizing complex soundscapes. Filtering techniques like high-pass, low-pass, Kalman, and Wiener filters further enhance ESC performance by refining signal clarity. Each filter is applied individually to analyze its effect on classification accuracy, enabling the selection of optimal preprocessing strategies.

In this study, a 1D Convolutional Neural Network (CNN) is employed over a 2D CNN due to its efficiency and suitability for 1D time-series data like audio signals. While 2D CNNs excel in image processing, 1D CNNs capture temporal patterns directly from raw audio waveforms, requiring fewer computational resources and parameters. This makes 1D CNNs more effective for ESC, as they can efficiently learn from sequential data without the need for additional transformations.

Efficient environmental sound classification is increasingly vital for creating smarter, more secure, and responsive systems, particularly in urban environments with dynamic and unpredictable audio landscapes. These environments require models that can distinguish between overlapping sounds, like traffic and sirens or human voices and background noise, while also adapting to changes in conditions over time. As noise levels and soundscapes evolve, classification techniques must be robust and adaptable to various contexts, such as urban noise pollution, smart homes, or emergency signal detection in surveillance systems.

II. Literature Survey

A. Feature Extraction

The development of end-to-end models for environmental sound classification has seen significant advancements with the introduction of CNN architectures.

Piczak (2015) [1] laid the groundwork by demonstrating that spectrograms, when used as input for CNNs, could achieve high classification performance for environmental sounds. However, the computational demands of generating spectrograms prompted a shift toward direct processing of raw audio signals.

Abdoli et al. (2019) [2] introduced a 1D CNN model capable of bypassing the spectrogram generation, processing raw audio directly, and reducing the overall complexity

and computational load. This model was highly efficient for environmental sound classification, offering a significant advantage in terms of resource usage. Additionally, the exploration of filtering techniques such as Mel-Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), and Gammatone filterbanks has been integral to improving feature extraction. These filtering methods help capture distinct aspects of sound: MFCC focuses on capturing timbre and spectral features, STFT analyzes the dynamic frequency content over time, and Gammatone filterbanks, inspired by the human auditory system, decompose sound in a way that is more aligned with perceptual auditory processing. The combination of raw audio processing with these advanced filtering techniques allows CNN models to better classify complex sounds with overlapping characteristics, making the approach more robust in real-world environmental sound classification challenges.

B. Works on Filtering Techniques

Filtering techniques are essential in environmental sound classification as they improve the quality of input signals by removing unwanted noise and emphasizing relevant features. High-pass and low-pass filters are commonly used to isolate specific frequency ranges. High-pass filters remove low-frequency background noise, such as rumbling, while low-pass filters suppress high-frequency noise, such as electrical interference. These filters ensure that the model focuses on the most important frequency components of the sound (Zhao & Li, 2017; Fang & Kim, 2018) [3] [4].

Additionally, advanced statistical filters like Kalman and Wiener filters are effective in noisy environments. The Kalman filter is an adaptive filtering technique that estimates the state of a signal in real time, continuously adjusting to noise fluctuations (Hassan & Lee, 2017) [5]. Similarly, the Wiener filter minimizes mean square error to reduce noise and enhance the signal (Zhang & Wang, 2018) [6]. These filters are particularly useful for improving signal quality when the signal-to-noise ratio is low (Lee & Cho, 2016) [7].

By integrating these filtering techniques with Convolutional Neural Networks (CNNs), researchers have enhanced classification accuracy in environmental sound models. Using filters like Kalman and Wiener before feeding data into a CNN allows the model to concentrate on the relevant sound features, improving overall performance, especially in complex, overlapping sound scenarios (Salamon & Bello, 2017; Jiang & Yang, 2019) [8] [9].

C. Problem Statement and Objectives

Problem Statement: Despite the significant progress in environmental sound classification using Convolutional Neural Networks (CNNs), there remains a lack of comprehensive exploration into the impact of advanced filtering techniques, such as high-pass, low-pass, Kalman, and Wiener filters, on model robustness and accuracy in real-world noisy environments. While various feature extraction methods, including MFCC, STFT, and Gammatone filterbanks, have been employed to improve model performance, their integration with these filtering techniques for enhancing signal quality, especially under low signal-to-noise ratios, has not been adequately addressed. This gap in research highlights the need for a detailed study on how combining different filtering methods can enhance environmental sound classification models, particularly for handling overlapping sounds and dynamic acoustic conditions in urban or noisy environments.

Objectives:

- To use MFCC, STFT, and Gammatone filtering techniques for feature extraction in an end-to-end environmental sound classification model.
- To evaluate and compare the effectiveness of different filtering techniques in improving classification performance and robustness using a 1D CNN model.

III. Proposed Methodology

A. Different Feature Extraction Methods

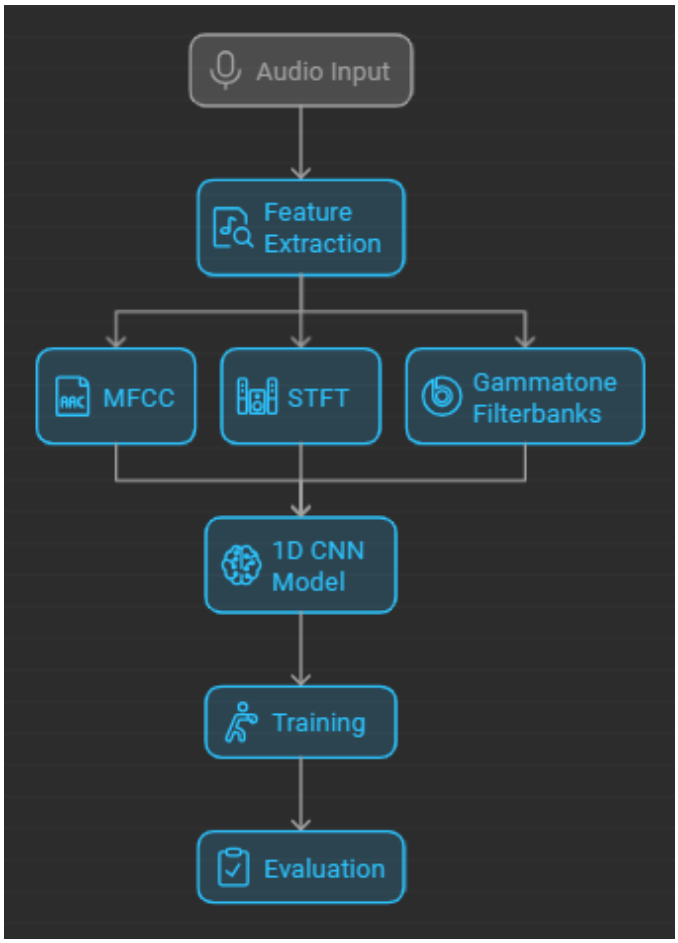

Fig. 1. Methodology For Objective 1.

Environmental sound classification involves distinguishing between various types of sounds like car horns, rain, or dog barks using machine learning models. In this objective, we explore the performance of a 1D Convolutional Neural

Network (CNN) when trained on three different audio features: MFCC, STFT, and Gammatone filterbanks.

To process audio signals for classification with a 1D CNN, begin by collecting and preparing the dataset, ensuring all samples have a consistent sampling rate (e.g., 16kHz or 44.1kHz) to maintain uniformity across inputs. Normalize the audio signals to control for variations in amplitude, which helps the model focus on the sound features rather than signal intensity differences. Then, extract features from the audio data using techniques like Mel-frequency cepstral coefficients (MFCC), Short-Time Fourier Transform (STFT), and Gammatone filterbanks to capture distinct aspects of the sound.We extracted three types of features from the audio files—MFCC, STFT, and Gammatone filterbanks—to capture different aspects of the sound signals:

1) MFCC (Mel-Frequency Cepstral Coefficients): MFCCs are widely used in speech and audio processing due to their ability to represent the short-term power spectrum. We computed 50 MFCC coefficients per frame with a frame size of 25 ms and an overlap of 10 ms. Figure 2 illustrates MFCCs for one of the categories.

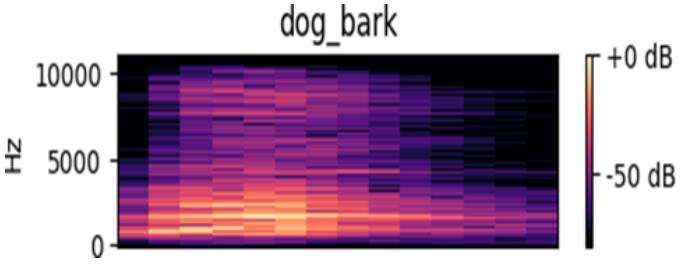

Fig. 2. Example MFCC for a category.

2) STFT (Short-Time Fourier Transform): STFT provides a time-frequency representation of the sound signal. By applying STFT, we computed a spectrogram for each audio sample using parameters consistent with MFCC, such as window length and overlap. The magnitude of the spectrogram was then used as input to the model. Figure 3 shows an example spectrogram for one of the categories.

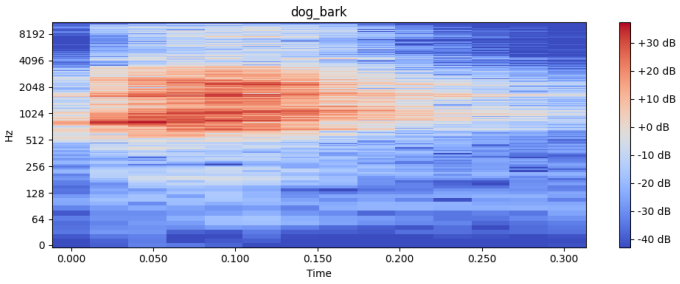

Fig. 3. Example STFT for a category.

3) Gammatone Filterbank: Gammatone filterbank features were extracted to simulate the frequency analysis performed by the human auditory system. A filterbank consisting of 64 gammatone filters, with a window time of 0.02 s and hop time of 0.01 s, was used to decompose the audio signals into frequency bands. The energy of each frequency band over time was computed, followed by optional log compression to match the scale of the other features. Figure 4 presents Gammatone filterbank features for one of the categories.

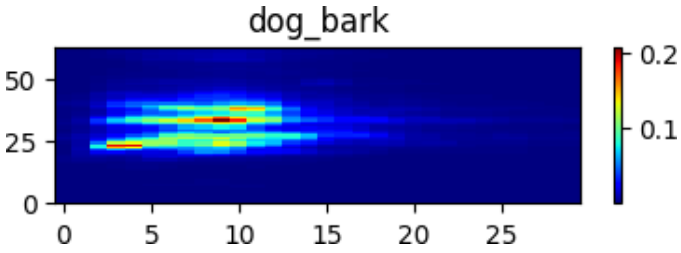

Fig. 4. Example Gammatone for a category.

Each feature extraction method emphasizes different qualities of the audio signal: MFCC focuses on perceived loudness and pitch, STFT captures time-frequency information, and Gammatone filterbanks offer an auditory-filtering approach. Once extracted, these feature representations are fed into a 1D CNN for training. After training the model on each feature set, evaluate and compare classification performance across methods

B. Different Filtering Methods

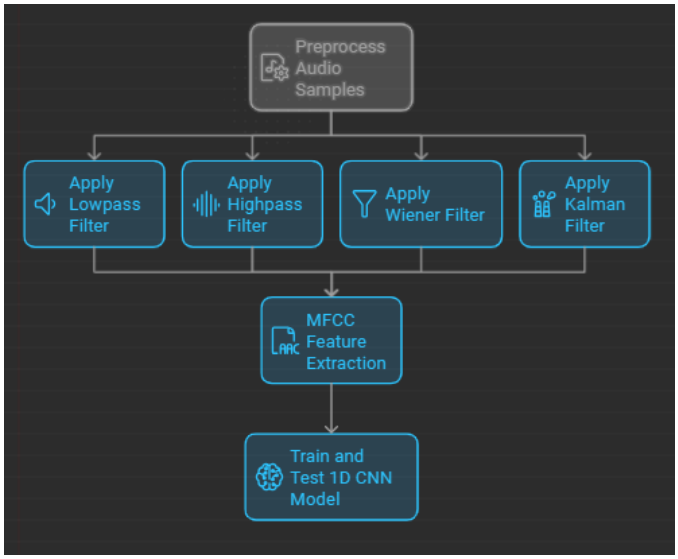

Fig. 5. Methodology For Objective 2.

Each audio sample is preprocessed to ensure consistency in sample rate, amplitude normalization, and length, maintaining uniformity across the dataset. Resampling aligns the temporal resolution, and amplitude normalization (e.g., z-score scaling) minimizes the impact of volume variations, letting the model focus on sound characteristics. Samples are trimmed or padded to a fixed length for consistent input size, ensuring compatibility with the model architecture. Filters such as high-pass and Wiener are applied to enhance signal clarity by isolating

relevant frequencies and reducing background noise, allowing critical sound features to stand out. MFCCs are then extracted due to their efficiency and ability to capture key frequency characteristics, making them ideal for distinguishing between subtle variations in environmental sounds.

1) Lowpass Filter: The lowpass filter allows sounds with frequencies below a specific cutoff point to pass through while blocking higher frequencies. It is particularly useful for reducing high-frequency noise that can interfere with identifying low-frequency sounds, such as traffic noise or machinery. By filtering out unnecessary high-pitched sounds, the lowpass filter helps the model focus on deeper sounds, enhancing the clarity of the signals used for classification.

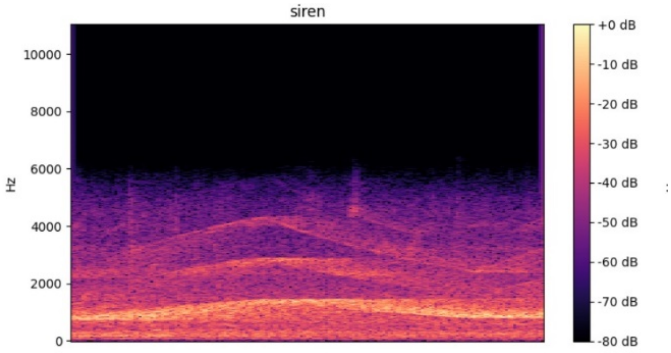

Fig. 6. Lowpass Filter.

2) Highpass Filter: In contrast, a highpass filter allows only higher frequencies to pass while blocking lower frequencies. This filter is effective for emphasizing sounds that contain significant high-frequency components, such as alarms, sirens, or bird calls. By removing low-frequency background noise, like the hum of distant traffic, the highpass filter sharpens the clarity of the high-pitched sounds, making it easier for the model to classify them accurately.

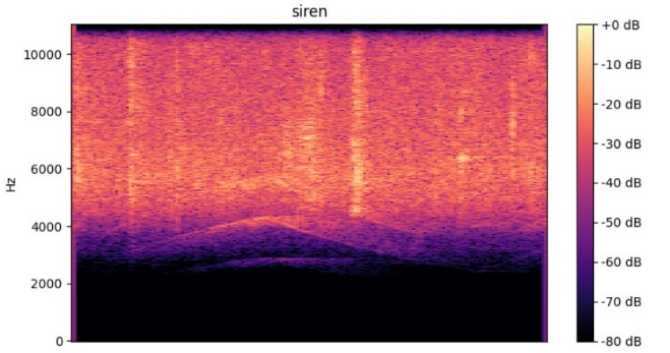

Fig. 7. Highpass Filter.

3) Wiener Filter: The Wiener filter is an adaptive filter that aims to reduce noise based on the characteristics of the signal. It works by estimating the signal and noise levels, allowing it to minimize overall noise without distorting the main sound. This filter is particularly useful in real-world scenarios where environmental sounds may vary in noise levels, as it helps maintain the integrity of the sound while improving clarity. Applying the Wiener filter can enhance the quality of diverse urban sounds in the dataset.

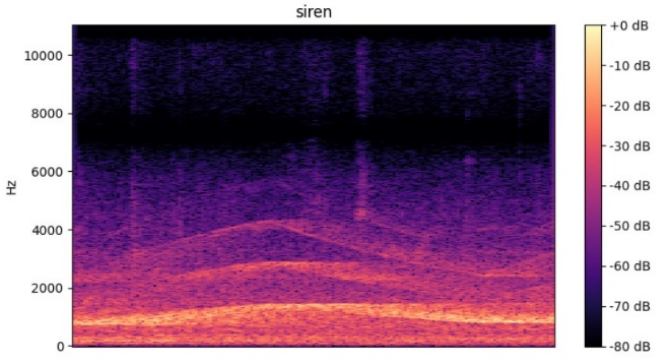

Fig. 8. Wiener Filter.

4) Kalman Filter: The Kalman filter is a statistical approach that uses a predictive model to reduce noise in time-series data. It operates by making predictions about the signal and then updating those predictions based on actual measurements. This method is effective in smoothing audio signals by removing random fluctuations or noise, allowing the model to focus on the underlying structure of the sounds. By applying the Kalman filter, we can achieve a clearer representation of environmental sounds, improving feature extraction for classification.

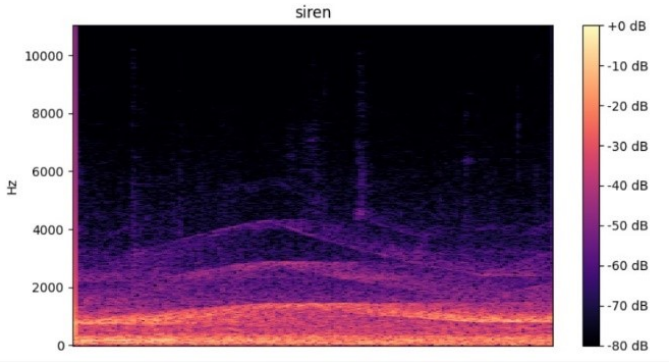

Fig. 9. Kalman Filter.

IV. Experimental Setup

A. Dataset

The UrbanSound8K dataset is a widely used collection of labeled audio clips designed for environmental sound classification. It consists of 8,732 sound recordings, each lasting up to 4 seconds, which have been categorized into 10 distinct sound classes, representing various urban sound events. The dataset is structured to provide a diverse range of acoustic environments, making it ideal for training and evaluating models aimed at identifying and classifying sounds in real-world, urban contexts. Here's the breakdown of the categories and the total number of samples:

TABLE I
DATASET CATEGORIES AND NUMBER OF SAMPLES

Category	Number of Samples
Air Conditioner	1,000
Car Horn	429
Children Playing	1,000
Dog Bark	1,000
Drilling	1,000
Engine Idling	1,000
Gunshot	374
Jackhammer	1,000
Siren	929
Street Music	1,000
Total	**8,732**

B. Libraries Used

The project utilizes several Python libraries for different tasks in audio processing and machine learning. *Librosa* is used for audio and music analysis, enabling the extraction of various audio features, *NumPy* is employed to handle arrays and perform operations on the extracted features. *Pandas* is used for displaying and manipulating the data in tabular form, making it easier to work with audio features. To visualize the processed data, *Matplotlib* is used for plotting various graphs, such as spectrograms, that help in analyzing the structure of the audio signals. Finally, *Keras* is leveraged to build and train a 1D Convolutional Neural Network (CNN) model for classifying the audio data based on the extracted features.In addition, *scikit-learn* (*sklearn*) is used for various machine learning tasks, including model evaluation, feature scaling, and splitting the data into training and testing sets. Its rich set of algorithms and utilities, such as cross-validation and performance metrics, helps optimize and assess the model's performance. *SciPy* is another crucial library that provides scientific and technical computing functionality.

C. Model Structure

The core of the classification system is a 1D Convolutional Neural Network (CNN). The CNN model consisted of several convolutional layers followed by max-pooling operations. Specifically, we used 3 convolutional layers with increasing numbers of filters (64, 128, 256) and kernel size of 3. Each convolutional layer used ReLU activation, followed by a max-pooling operation to reduce the dimensionality of the feature maps. After the convolutional layers, two fully connected (dense) layers with 256 and 128 units were added, using dropout (rate 0.5) to prevent overfitting. The output layer consisted of a softmax function, producing probabilities for each class in the dataset.

The models were trained using the Adam optimizer with a learning rate of 0.001 and categorical cross entropy as the loss function. We used a batch size of 32 and trained for 50–250 epochs based on the features and based on the validation accuracy to prevent overfitting. The evaluation metrics included classification accuracy, precision, recall, and F1-score, which were calculated on the held-out test set. The same model was used for both the objectives

The code for proposed methodology is available at: https://github.com/Nakul155/Environmental-Sound-Classification

V. RESULT AND ANALYSIS

In this section, we will present and discuss the results of applying different feature extraction methods and filtering methods. The main goal is to compare the accuracy by using different extraction and filtering methods.

The results of the environmental sound classification experiments are analyzed based on the performance metrics obtained from the 1D CNN model trained on different filtered audio datasets. The primary metric evaluated is the F1-score, which balances precision and recall to provide a comprehensive measure of the model's performance in classifying various environmental sounds from the UrbanSound8K dataset.

A. Objective 1: Feature Extraction

In this study, STFT emerged as the best-performing feature extraction set for environmental sound classification using a 1D CNN, achieving the highest test accuracy of 92.1% and an F1-score of 0.92. The time- frequency representation provided by STFT allowed the model to capture detailed temporal and spectral characteristics of the audio signals, which contributed to its superior performance. Although STFT had a larger dimensionality compared to the other feature sets, the model was able to effectively utilize this information to classify environmental sounds more accurately.

MFCC, a widely-used feature in audio analysis, followed closely with a test accuracy of 91% and an F1-score of 0.91. MFCC provided a compact representation of the short-term power spectrum, which helped in balancing classification performance and computational efficiency. Although it didn't outperform STFT, MFCC offered faster training convergence due to its lower dimensionality and was still highly effective for environmental sound classification.

Surprisingly, Gammatone filterbank features performed the least well, with a test accuracy of 90.1% and an F1-score of 0.90. Despite their auditory-inspired design, which mimics human hearing, Gammatone features did not capture the relevant sound patterns as effectively as STFT or MFCC. The model trained on Gammatone features showed slower convergence and struggled more with differentiating between certain classes of environmental sounds, which may indicate that this representation was less suitable for this particular task or dataset. Gammatone took the highest amount of time for feature extraction among all of the filters.

Category	F1 Score	Accuracy
MFCC	0.91	91.01%
Gammatone	0.89	90.04%
STFT	0.92	92.10%

TABLE II
COMPARISON OF F1 SCORES AND ACCURACY ACROSS FEATURE EXTRACTION METHODS

B. Objective 2: Filtering Methods

The lowpass filter achieved an F1-score of 0.89. This high score suggests that it effectively enhanced the model's ability to classify sounds with significant low-frequency components while allowing for fast feature extraction.

The highpass filter yielded an F1-score of 0.87. Although slightly lower than the lowpass filter, this score indicates that the filter successfully emphasized high-frequency sounds, and it also allowed for fast feature extraction.

The Wiener filter produced the best F1-score of 0.91. This result demonstrates its effectiveness in reducing noise and preserving essential sound features, leading to improved classification accuracy while still allowing for fast extraction.

The Kalman filter achieved an F1-score of 0.88, which reflects good performance in smoothing the audio signal. However, it is important to note that this filter has a slower extraction rate compared to the others.

Overall, the Wiener filter provided the highest F1-score, indicating superior performance in environmental sound classification. The lowpass and highpass filters also performed well, particularly in their respective frequency domains, while maintaining fast extraction times. In contrast, the Kalman filter, although effective, demonstrated slower extraction speeds, which may be a consideration in applications requiring real-time processing.

Category	Precision	Recall	F1-Score
Lowpass	0.89	0.89	0.89
Highpass	0.88	0.87	0.87
Wiener	0.92	0.91	0.91
Kalman	0.89	0.88	0.88

TABLE III
COMPARISON OF F1 SCORES AND ACCURACY ACROSS FILTERING METHODS

VI. CONCLUSION AND FUTUREWORK

This study presented a comprehensive approach to Environmental Sound Classification (ESC) using a 1D Convolutional Neural Network (CNN) model. Our ESC system was developed by applying key feature extraction techniques—MFCC, STFT, and Gammatone filterbanks—to capture essential audio characteristics from diverse environmental sounds. Filtering techniques, including high-pass, low-pass, Kalman, and Wiener filters, were evaluated to optimize classification performance by refining signal clarity and focusing on relevant sound patterns. The 1D CNN architecture was chosen for its efficiency and effectiveness in processing sequential data directly from 1D audio waveforms, achieving promising results with lower computational demands than 2D CNN models.

For future work, several techniques could further improve ESC performance. Exploring hybrid models that combine 1D and 2D CNN layers might enhance the ability to capture both temporal and spatial features in sound data, benefiting sounds with complex frequency variations. Additionally, experimenting with advanced feature extraction methods, such as wavelet transforms, could offer richer time-frequency representations. Transfer learning, using pre-trained models on large audio datasets, may also improve performance by leveraging learned audio representations. Finally, implementing attention mechanisms could enhance model focus on significant portions of the audio signal, potentially improving accuracy in noisy environments.

These future directions offer valuable avenues for advancing ESC systems, enabling greater accuracy and resilience across varied soundscapes, and further enhancing applications in urban planning, wildlife conservation, and public safety.

REFERENCES

[1] 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.

[2] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-end environmental sound classification using a 1d convolutional neural network," 2019. [Online]. Available: https://arxiv.org/abs/1904.08990

[3] Y. Zhao and Z. Li, "Environmental sound classification with convolutional neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 421–425.

[4] H. Fang and C. Kim, "Environmental sound classification using feature selection and cnn," *IEEE Access*, vol. 6, pp. 40 162–40 171, 2018.

[5] M. M. Hassan and S. Lee, "Noise reduction for environmental sound classification using kalman filter," in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 2341–2345.

[6] S. Zhang and D. Wang, "Speech enhancement using wiener filter in environmental sound classification systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1390–1400, 2018.

[7] K. Lee and Y. Cho, "Wiener filter-based preprocessing for noise-robust sound classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 657–661.

[8] 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.

[9] X. Jiang and J. Yang, "Environmental sound classification using deep convolutional neural networks with denoising preprocessing," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 348–354.