FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Impact of Feature Representation and Selection on the Performance of Sound Event Classification

Parinaz Binandeh Dehaghani

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

Supervisor: Professor A. Pedro Aguiar

June 30, 2025

Abstract

As Environmental Sound Classification (ESC) continues to mature, future research is expected to focus on making systems more robust, generalizable, and efficient under real-world conditions. A promising direction lies in the integration of adaptive feature selection and representation learning techniques that can dynamically respond to variations in acoustic environments, enabling models to learn the most informative patterns from complex and noisy data. In parallel, the adoption of semi-supervised and self-supervised learning paradigms can significantly reduce the dependency on large-scale annotated datasets, which are often expensive, labor-intensive, and domain-specific. These approaches hold the potential to unlock more scalable and transferable ESC systems.

From a deployment standpoint, the development of lightweight models that can operate efficiently on edge devices—such as smartphones, embedded microphones, and IoT sensors—will be essential for real-time applications in smart cities, environmental monitoring, public safety, healthcare, and autonomous systems. Lightweight architectures offer several advantages: they require fewer computational resources, consume less power, and enable low-latency inference, which is critical for applications involving continuous or real-time audio streams. Additionally, such models facilitate on-device processing, enhancing privacy and reducing the need for constant cloud communication, which is particularly beneficial in bandwidth-limited or privacy-sensitive settings.

However, cross-domain generalization remains a persistent challenge. Future ESC systems must be capable of performing consistently across diverse geographic locations, recording equipment, and acoustic conditions. Addressing this will require strategies such as domain adaptation, data augmentation, and the use of synthetic soundscapes for training. Moreover, the integration of ESC systems with multimodal sensor data—such as audio-visual streams or geolocation metadata—can further improve context-awareness and situational understanding, enabling more accurate and reliable decision-making.

In this regard, this thesis contributes to the advancement of ESC systems by systematically analyzing the role of various feature representations and selection strategies in enhancing classification performance. Through comparative experiments on standard datasets such as ESC-50 and UrbanSound8K, this work demonstrates how combining complementary audio features and applying selection methods can lead to more accurate, efficient, and generalizable models. Key contributions include: (1) evaluation of stacked audio features for improved CNN performance; (2) development of advanced pooling strategies, including Sparse Salient Region Pooling (SSRP), to enhance feature learning while maintaining low complexity; (3) adaptation of spectrogram-aware transformer architectures that model both temporal and spectral patterns; (4) thorough benchmarking using ESC-50; and (5) practical guidelines for feature engineering in ESC tasks. Together, these findings provide actionable insights for building scalable, real-time ESC systems under computational and deployment constraints.

Resumo

À medida que a Classificação de Sons Ambientais (ESC) continua a evoluir, espera-se que a investigação futura se foque em tornar os sistemas mais robustos, generalizáveis e eficientes em condições do mundo real. Uma direção promissora reside na integração de técnicas de seleção adaptativa de características e de aprendizagem de representações que possam responder dinamicamente às variações nos ambientes acústicos, permitindo que os modelos aprendam os padrões mais informativos a partir de dados complexos e ruidosos. Paralelamente, a adoção de paradigmas de aprendizagem semi-supervisionada e auto-supervisionada pode reduzir significativamente a dependência de grandes conjuntos de dados anotados, que são frequentemente dispendiosos, exigem muito trabalho manual e são específicos de determinado domínio. Estas abordagens têm o potencial de desbloquear sistemas de ESC mais escaláveis e transferíveis.

Do ponto de vista da implementação, o desenvolvimento de modelos leves que possam funcionar eficientemente em dispositivos de borda—como smartphones, microfones embutidos e sensores IoT—será essencial para aplicações em tempo real em cidades inteligentes, monitorização ambiental, segurança pública, cuidados de saúde e sistemas autónomos. Arquiteturas leves oferecem várias vantagens: requerem menos recursos computacionais, consomem menos energia e permitem inferência com baixa latência, o que é crucial para aplicações que envolvem fluxos de áudio contínuos ou em tempo real. Além disso, tais modelos facilitam o processamento local no dispositivo, melhorando a privacidade e reduzindo a necessidade de comunicação constante com a cloud, o que é particularmente benéfico em contextos com largura de banda limitada ou com requisitos sensíveis de privacidade.

No entanto, a generalização entre domínios continua a ser um desafio persistente. Os sistemas de ESC futuros devem ser capazes de ter um desempenho consistente em diferentes localizações geográficas, equipamentos de gravação e condições acústicas. Para enfrentar este desafio, serão necessárias estratégias como adaptação de domínio, aumento de dados e o uso de paisagens sonoras sintéticas para treino. Além disso, a integração de sistemas ESC com dados de sensores multimodais—como fluxos áudio-visuais ou metadados de geolocalização—pode melhorar ainda mais a perceção contextual e a compreensão situacional, permitindo uma tomada de decisão mais precisa e fiável.

Neste contexto, esta tese contribui para o avanço dos sistemas ESC através da análise sistemática do papel de várias representações e estratégias de seleção de características na melhoria do desempenho de classificação. Através de experiências comparativas em conjuntos de dados padrão como ESC-50 e UrbanSound8K, este trabalho demonstra como a combinação de características áudio complementares e a aplicação de métodos de seleção pode levar a modelos mais precisos, eficientes e generalizáveis. As principais contribuições incluem: (1) avaliação de características áudio empilhadas para melhoria do desempenho de redes neuronais convolucionais (CNN); (2) desenvolvimento de estratégias avançadas de pooling, incluindo o Sparse Salient Region Pooling (SSRP), para potenciar a aprendizagem de características mantendo baixa complexidade; (3) adaptação de arquiteturas transformer sensíveis a espectrogramas que modelam padrões

temporais e espectrais; (4) benchmarking exaustivo com o conjunto de dados ESC-50; e (5) diretrizes práticas para a engenharia de características em tarefas ESC. Em conjunto, estas descobertas oferecem orientações acionáveis para a construção de sistemas ESC escaláveis e em tempo real, considerando restrições computacionais e de implementação.

Sustainable Development Goals (SDG) of the United Nations

9 – Industry, Innovation, and Infrastructure

Goal 9.5: Enhance scientific research, upgrade the technological capabilities of industrial sectors in all countries, in particular developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers per 1 million people and public and private research and development spending.

Contribution: My research contributes to the advancement of intelligent sound monitoring technologies through the development of machine learning algorithms for environmental sound detection and classification. These systems enable the automatic interpretation of acoustic scenes using feature extraction methods and neural network-based models, including CNNs and transformers. By exploring efficient, explainable, and scalable approaches, this work supports innovation in AI-driven monitoring systems, smart infrastructure, and sustainable technology development.

Performance metrics: Increase in research publications, open-source contributions, and deployment-ready algorithms for real-world audio monitoring applications in smart infrastructure and industrial settings.

11 – Sustainable Cities and Communities

Goal 11.6: By 2030, reduce the adverse per capita environmental impact of cities, including by paying special attention to air quality and municipal and other waste management.

Contribution: Intelligent sound event detection (SED) systems support smart city initiatives by enabling real-time acoustic surveillance and automated environmental noise monitoring. These systems can help detect abnormal events (e.g., sirens, alarms, gunshots) or harmful noise levels,

thereby improving public safety and environmental quality. The application of SED in urban infrastructure fosters resilience, early detection, and faster response to environmental and emergency situations.

Performance metrics: Deployment of real-time SED systems in smart city environments; reduction in noise complaints and environmental impact assessments via acoustic data; increased integration of AI-driven audio analytics in urban monitoring platforms.

13 – Climate Action

Goal 13.1: Strengthen resilience and adaptive capacity to climate-related hazards and natural disasters in all countries.

Contribution: Sound-based monitoring technologies developed in this research can support early warning systems for natural events such as storms, landslides, or floods by detecting specific acoustic patterns in the environment. When combined with other sensor modalities, acoustic intelligence can improve disaster preparedness and response, especially in regions where traditional infrastructure is limited.

Performance metrics: Improvement in the response time and accuracy of early warning systems; integration of sound-based AI models into multi-modal environmental monitoring platforms for climate resilience.

15 – Life on Land

Goal 15.5: Take urgent and significant action to reduce the degradation of natural habitats, halt the loss of biodiversity, and, by 2020, protect and prevent the extinction of threatened species.

Contribution: The audio classification techniques developed in this work can be employed in bioacoustics monitoring systems for wildlife conservation. Machine learning models trained to detect animal calls, poaching activities, or environmental anomalies contribute to protecting biodiversity in forests, wetlands, and protected reserves. These technologies enable non-invasive, cost-effective monitoring in remote areas.

Performance metrics: Increase in biodiversity monitoring programs utilizing AI-based sound classification; accuracy and robustness of detection models in field deployment; contributions to conservation through automated analysis of environmental sound data.

Acknowledgements

I would like to express my deepest gratitude to all those who have played a significant role in my academic journey. Your unwavering support and guidance have not only shaped this work but have also been instrumental in my personal growth, molding me into the person I am today.

First and foremost, I am immensely thankful to my supervisor, Professor António Pedro Aguiar. Throughout this journey, he not only believed in my potential but also provided me with numerous opportunities to grow both academically and personally. Professor Aguiar's expertise, invaluable mentorship, and constant support played a pivotal role in shaping this dissertation, and I am profoundly grateful for his guidance.

I also extend my sincere appreciation to Dr. Danilo Pena, who patiently taught me many things and helped me progress academically. I hope we will continue to accomplish great things together.

Additionally, I would like to thank the C2SR Lab and SYSTEC for providing me with excellent working conditions.

I want to express my heartfelt gratitude to my incredible family—my parents and sisters (Azita and Nahid), particularly my sister Dr. Nahid Binandeh Dehaghani—who have been my pillars of strength throughout my academic journey. Their constant encouragement and belief in my potential have made all the difference. Without such amazing role models and their unwavering support, this dissertation would never have been possible. I cannot thank you enough for all you have done for me.

Thank you all,

Parinaz Binandeh Dehaghani

"Wherever there is air, there is sound, And where there is sound, there is information."

F. Richard Moore

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process:
During the preparation of this dissertation, the author used ChatGPT solely to improve language and enhance readability. All ideas, analysis, and original content are the work of the author. The author carefully reviewed and edited the text after using the tool and takes full responsibility for the final version of the dissertation.

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
ASC	Acoustic Scene Classification
AST	Audio Spectrogram Transformer
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DAF	Deep Audio Feature
DNN	Deep Neural Network
ESC	Environmental Sound Classification
GAP	Global Average Pooling
IoT	Internet of Things
KNN	k Nearest Neighbors
LM	Log Mel (spectrogram)
LSTM	Long Short-Term Memory
ML	Machine Learning
MFCC	Mel Frequency Cepstral Coefficient
NLP	Natural Language Processing
RF	Random Forest
RNN	Recurrent Neural Network
SAE	Stacked Auto Encoders

SED	Sound Event Detection
SGD	Stochastic Gradient Descent
SSRP	Sparse Salient Region Pooling
SVM	Support Vector Machine
TFS	Temporal-Frequency Sequential
TFP	Temporal-Frequency Parallel
TO	Temporal Only
PANN	Pretrained Audio Neural Networks
PCA	Principal Component Analysis
ViT	Vision Transformer
ZCR	Zero Crossing Rate

Chapter 1

Introduction

This chapter introduces the main topics addressed in this dissertation titled "Impact of Feature Representation and Selection on the Performance of Sound Event Classification." Section 1.1 outlines the context and motivation for the dissertation. Section 1.2 defines the key research objectives. Section 1.3 presents the major scientific contributions resulting from this study. Finally, Section 1.4 provides an outline of the dissertation structure, offering a roadmap for the chapters that follow.

1.1 Context and Motivation

Environmental Sound Classification (ESC) has become an essential area of research within audio signal processing, driven by its growing relevance in real-world applications such as smart cities, audio surveillance, healthcare monitoring, and human-computer interaction [7, 9]. Unlike speech or music, environmental sounds are inherently unstructured, diverse, and context-dependent, often exhibiting significant intra-class variability. These characteristics make their classification a complex and challenging task.

At the core of ESC lies the need to design effective feature representations that capture both spectral and temporal patterns of environmental sounds. Traditionally, this has involved handcrafted features across the time domain (e.g., zero-crossing rate, energy), frequency domain (e.g., spectral centroid, bandwidth), and time-frequency domain (e.g., Mel spectrograms, MFCCs). While such features have laid the groundwork for ESC systems, they often struggle with generalization, especially in noisy or variable acoustic settings [3, 10].

Recent research has shown that combining multiple feature types—commonly referred to as stacked features—can enhance model generalization by leveraging the complementary nature of different feature domains [11]. In parallel, deep learning methods such as Convolutional Neural Networks (CNNs) have demonstrated strong potential for learning hierarchical patterns directly from spectrogram-like representations. However, these high-capacity models are often computationally expensive, which limits their real-time or embedded deployment.

To address this trade-off, efficient architectural designs such as Sparse Salient Region Pooling (SSRP) [7] and dimensionality reduction techniques (e.g., PCA, mutual information, correlation

Figure 1.1: Acoustic sensor nodes deployed on New York City streets. Source:[1]

filtering) have been proposed. These methods aim to reduce input redundancy, accelerate training, and retain only the most informative patterns for classification [10]. Motivated by these developments, this thesis systematically investigates how feature design, selection, and integration affect the performance and efficiency of ESC systems. The aim is to systematically investigate how the design, selection, and integration of features impact the accuracy and efficiency of sound event classification systems.

Environmental Sound Classification (ESC) and Sound Event Detection (SED) technologies are increasingly used in real-world systems to enable intelligent audio-based decision-making. These methods equip machines with auditory perception, allowing them to interpret ambient sound events similarly to how humans do. ESC is particularly valuable in scenarios where visual data is insufficient or unavailable, such as in darkness, occlusion, or privacy-sensitive environments. Below are some of the key application domains where ESC and SED have demonstrated significant impact:

- **Smart Homes and Smart Cities:** Sound recognition in smart environments allows automation and context-aware services by detecting everyday sounds such as appliance beeps, water taps, footsteps, or door knocks. ESC systems enhance comfort, energy efficiency, and security in smart homes and urban settings [12, 13]. As illustrated in Figure 1.1, acoustic sensors are deployed on building facades and rooftops in urban areas to monitor and classify a wide range of environmental sounds, facilitating real-time decisions and services in smart city infrastructures.

- **Urban Surveillance and Public Safety:** ESC systems can detect critical auditory events

Figure 1.2: Example of an aggression detection interface used in urban surveillance systems. The system integrates real-time audio signal processing, including waveform monitoring and spectrogram analysis, with video feeds to detect and respond to hostile auditory events in public spaces. Source:[2]

such as gunshots, glass breaking, or human screams to support crime detection, emergency response, and public safety efforts. These systems are often integrated into smart surveillance infrastructure to trigger real-time alerts and assist in rapid decision-making [14, 15]. As illustrated in Figure 1.2, such systems typically combine audio waveform monitoring, spectrogram analysis, and video feeds to identify and respond to potentially hostile situations in public spaces.

- **Healthcare and Elderly Monitoring:** In assisted living and healthcare environments, ESC is used to monitor for audio cues like coughing, falling, snoring, or distress sounds. This supports timely medical response and improves safety for the elderly or disabled [16, 17].

- **Human-Robot Interaction (HRI):** Equipping robots with auditory perception enables them to interpret commands, navigate environments, and interact socially. ESC allows robots to recognize key sounds and adapt behaviors, improving their contextual understanding [16].

- **Wildlife Monitoring and Bioacoustics:** ESC systems are used for passive acoustic monitoring of animal behavior and biodiversity. These applications help researchers detect species presence, monitor migration, and track environmental changes using soundscapes [18, 19].

- **Transportation and Industrial Monitoring:** ESC helps detect anomalies in vehicle sounds or machinery noise, enabling predictive maintenance and enhanced operational safety. These

systems reduce downtime and prevent faults through early detection of unusual acoustic signatures [20].

The applications above demonstrate the growing utility of ESC and SED in diverse domains. Feature engineering and learning methods—core topics of this thesis—are central to improving performance in these contexts. From smart cities to ecological conservation, accurate sound classification opens the door to more responsive, intelligent systems.

1.2 Objectives

The primary objective of this thesis is to investigate and enhance the performance of Environmental Sound Classification (ESC) systems through systematic exploration of feature engineering and modern learning techniques.

The key objectives of this thesis are as follows:

- To perform a comprehensive review of feature extraction methods commonly used in audio signal analysis, including time-domain, frequency-domain, time-frequency domain, and cepstral features.
- To analyze the performance of various combinations of stacked features (e.g., MFCC + Mel Spectrogram, Log-Mel + Spectral Contrast + Chroma) for enhancing classification accuracy in ESC tasks.
- To apply and compare different algorithms (e.g., CNNs, Attention-based Transformers, Sparse Salient Region Pooling CNNs) using selected feature sets.
- To evaluate the impact of advanced feature learning approaches, such as deep representations, on the robustness and generalizability of ESC systems.
- To explore the trade-offs between computational complexity and classification performance in the context of real-world deployment of SED systems.

The outcomes of this thesis are expected to provide valuable insights into effective feature selection and learning strategies for sound-based intelligent systems, contributing to advancements in applications such as smart cities, surveillance, assistive robotics, and context-aware devices.

1.3 Contributions

A conference paper has been derived from this research and accepted for publication at the MadeAI-2025 (The 2nd International Conference of Modelling, Data Analytics and AI in Engineering), titled "Performance Comparison of CNN Models with Stacked Features for Environmental Sound Classification." Furthermore, a journal publication on the Investigation of Feature Selection Methods for Efficient Environmental Sound Classification is currently in progress and will be published in due course.

This thesis makes several significant contributions to the field of Environmental Sound Classification (ESC), with a particular focus on feature engineering and learning-based approaches. The main contributions are summarized as follows:

- **Evaluation of Stacked Feature Representations and Use of Pretrained Models:** Various combinations of audio features—such as MFCC, Log-Mel Spectrogram, Spectral Contrast, Tonnetz, and Chroma—are stacked and systematically evaluated as input to Convolutional Neural Network (CNN) models. The results show that stacked features enhance the model's ability to capture diverse audio characteristics, and that using pretrained models further improves classification performance.
- **Pooling Strategies:** Advanced architectures, including Sparse Salient Region Pooling (SSRP), are explored to enhance feature learning in CNNs. SSRP shows strong potential for improving model robustness while maintaining low computational complexity.
- **Spectrogram Transformer Architectures:** Transformer-based models are adapted and evaluated for ESC by designing spectrogram-aware attention mechanisms. Three architectures are explored—Temporal Only (TO), Temporal-Frequency Parallel (TFP), and Temporal-Frequency Sequential (TFS)—which explicitly model the temporal and spectral dimensions of input spectrograms. These variants aim to better exploit the structure of spectrograms, offering an alternative to standard Audio Spectrogram Transformer (AST) models.
- **Implementation and Benchmarking on Standard Datasets:** The proposed approaches are validated on the benchmark dataset ESC-50. Evaluation metrics such as accuracy and confusion matrices are reported, enabling fair comparison and reproducibility.
- **Practical Recommendations for Feature Engineering in ESC:** Based on experimental findings, the thesis provides practical guidelines for selecting, combining, and learning features for sound classification tasks, contributing to the design of more efficient and accurate ESC systems.

Together, these contributions provide a holistic view of the role of feature design and learning in ESC systems, bridging the gap between classical audio signal processing and modern deep learning techniques.

1.4 Dissertation Outline

This dissertation is organized into six chapters, each addressing a key component in the development of efficient and robust systems for environmental sound classification through the lens of feature engineering and deep learning.

- **Chapter 1 – Introduction:** This chapter introduces the context and motivation behind the study of Environmental Sound Classification (ESC). It outlines the real-world relevance

of ESC, states the objectives of the research, and highlights the main contributions of the dissertation.

- **Chapter 2 – State of the Art:** This chapter provides a comprehensive review of related work in ESC. It discusses various audio feature extraction methods, classification models ranging from traditional machine learning to deep learning, benchmark datasets and evaluation metrics.

- **Chapter 3 – Performance Comparison of CNN Models with Stacked Features:** This chapter investigates the impact of stacked feature representations on the performance of Convolutional Neural Networks (CNNs) in ESC tasks. It introduces the CNN and AST-based architectures used in the experiments, outlines the feature extraction pipeline, and compares different combinations of time-frequency features. The analysis includes performance trends, training efficiency, and generalization capabilities.

- **Chapter 4 – Feature Selection Methods:** This chapter focuses on feature selection techniques aimed at reducing redundancy and improving the computational efficiency of ESC models. It presents both filter-based and PCA-based selection methods, and introduces advanced pooling strategies—specifically Sparse Salient Region Pooling (SSRP)—as implicit feature selectors in CNNs. The chapter concludes with a comparative evaluation of these approaches.

- **Chapter 5 – Spectrogram Transformers for Audio Classification:** This chapter explores transformer-based architectures tailored for ESC. It presents three transformer variants (TO, TFP, TFS), describing their model configurations, input representations, and training protocols. The performance of each model is analyzed in terms of classification accuracy and computational efficiency, demonstrating the potential of spectrogram transformers in audio tasks.

- **Chapter 6 – Conclusion:** The final chapter summarizes the key findings and contributions of this dissertation. It reflects on the comparative advantages of stacked features, feature selection methods, and transformer-based models in ESC. The chapter also outlines directions for future work.

Chapter 2

State of the Art

The purpose of this chapter is to provide a comprehensive overview of the current state-of-the-art in Environmental Sound Classification (ESC). The chapter is divided into ten main sections that collectively build the foundational background for understanding recent advances in this domain. Section 2.1 introduces the concept of ESC and highlights its significance across a wide range of real-world applications. Benchmark datasets commonly used in ESC research are summarized in Section 2.2. Section 2.3 presents the evolution and taxonomy of audio feature extraction methods, tracing the development and categorization of different types of features used in ESC. Section 2.4 offers an in-depth analysis of feature extraction techniques, covering time-domain, frequency-domain, time-frequency domain, and deep features. Section 2.5 compares traditional machine learning approaches with modern deep learning models. Feature selection techniques are explored in Section 2.6, with a focus on filter methods and PCA-based approaches. Section 2.7 presents recent advancements in transformer-based models for ESC, followed by a discussion on the use of the categorical cross-entropy loss function in Section 2.8. Section 2.9 introduces a Deep Reinforcement Learning framework for audio-based applications, offering insight into decision-making in sound-aware systems. Finally, Section 2.10 details the evaluation metrics and protocols employed to assess classification performance, including accuracy, precision, recall, F1-score, confusion matrix, and standardized evaluation procedures.

2.1 Introduction to Environmental Sound Classification (ESC)

Environmental Sound Classification (ESC) is the task of automatically identifying and categorizing non-speech audio events from a wide range of acoustic environments. Unlike speech recognition or music classification, ESC deals with a diverse set of sounds such as dog barks, sirens, footsteps, rainfall, or engine noise, which vary significantly in duration, frequency content, and temporal structure [21, 22]. These sounds often occur in complex and noisy backgrounds, making their classification a challenging problem [23]. ESC has gained significant attention in recent years due to its critical role in numerous real-world applications. These include surveillance systems, smart cities, healthcare monitoring, wildlife conservation, industrial automation, and multimedia

indexing [22, 24]. The increasing availability of open datasets and advances in machine learning, particularly deep learning, have accelerated progress in this field [25].

The goal of an ESC system is to extract meaningful features from raw audio signals and use these representations to assign labels corresponding to predefined sound classes. Developing robust ESC systems requires addressing several challenges, including background noise, overlapping events, variability in sound sources, and limited labeled data. To tackle these challenges, researchers have proposed various feature extraction techniques, machine learning models, and evaluation strategies, which will be discussed in the following sections.

2.2 Benchmark Datasets

Benchmark datasets play a crucial role in the development and evaluation of environmental sound classification (ESC) and sound event detection (SED) systems. These datasets provide standardized audio recordings, annotations, and metadata that allow researchers to train, test, and compare models under consistent conditions. Table 2.1 summarizes some of the most widely used datasets in this domain.

AudioSet: One of the largest and most comprehensive datasets is AudioSet, developed by Google [13]. It contains over 2 million 10-second audio clips sourced from YouTube videos, manually annotated with 527 sound event classes. AudioSet supports general-purpose audio event detection and has become a foundational resource for deep learning models in large-scale classification tasks.

DESED: The DESED dataset focuses on sound events occurring in domestic environments, such as door knocks, speech, or music [26]. It comprises both synthetic mixtures and real-world recordings and is commonly used in the DCASE challenges for evaluating SED systems in home settings.

UrbanSound8K: This dataset, introduced by Salamon et al. [27], is a curated dataset of 8,732 short audio clips belonging to 10 urban sound classes such as sirens, drilling, or car horns. It has been widely used for ESC tasks due to its manageable size and clear class definitions.

ESC-50: This is another popular dataset designed for environmental sound classification [28]. It consists of 2,000 audio clips categorized into 50 balanced classes, making it ideal for benchmarking deep learning models and feature extraction techniques.

TUT Acoustic Scenes 2017: The TUT Acoustic Scenes 2017 dataset, released as part of the DCASE challenges, contains recordings from 15 different acoustic scenes such as cafés, streets, and public transport [15]. It is often used in acoustic scene classification tasks and includes both stereo audio and high-quality annotations.

These datasets differ in terms of sample rate, number of channels, and class diversity, which provide flexibility and coverage for various audio classification tasks. Their public availability and standardization have been instrumental in driving advancements in machine learning-based audio analysis.

Table 2.1: Overview of commonly used benchmark datasets in Environmental Sound Classification.

Dataset	Description	Publisher	Classes	Audio Files
AudioSet	Large-scale collection of over 2 million 10-second audio clips from YouTube, labeled with 527 sound event categories.	Google	527	2,084,320
DESED	Dataset of sound events in domestic settings, including both synthetic mixtures and real audio recordings, used in DCASE challenges.	DCASE Community	10	15,000
UrbanSound8K	Contains 8,732 labeled audio clips from 10 urban sound classes such as siren, engine idling, and street music.	NYU	10	8,732
ESC-50	A curated dataset of 2,000 labeled environmental audio recordings across 50 balanced classes.	Karol Piczak	50	2,000
TUT 2017	Recordings of real-world acoustic scenes (e.g., café, metro, street) with 15 class labels, used for acoustic scene classification.	DCASE Community	15	15,000

2.3 Evolution and Taxonomy of Audio Feature Extraction Methods

In simple terms, feature extraction is a process of highlighting the most dominating and discriminating characteristics of a signal. A suitable feature mimics the properties of a signal in a much more compact way. The evolution of audio signal features is illustrated in Figure 2.1. These features can be broadly categorized into four groups: time-domain, frequency-domain, time-frequency (joint) domain, and deep (learned) features. The oldest and simplest features are extracted from the time domain. The time domain features evolved up to the late 1950s [29, 30, 31]. To date, the time domain features play an important role in audio analysis and classification. To analyze the spectrum of audio signals, several features like pitch, formants, etc. were evolved from the frequency domain and employed in various applications to date. The evolution of frequency domain features was around the 1950s to 1960s [32, 33]. In the later 1960s, the joint time-frequency [34, 35] feature extraction algorithms were developed. Since then, these features

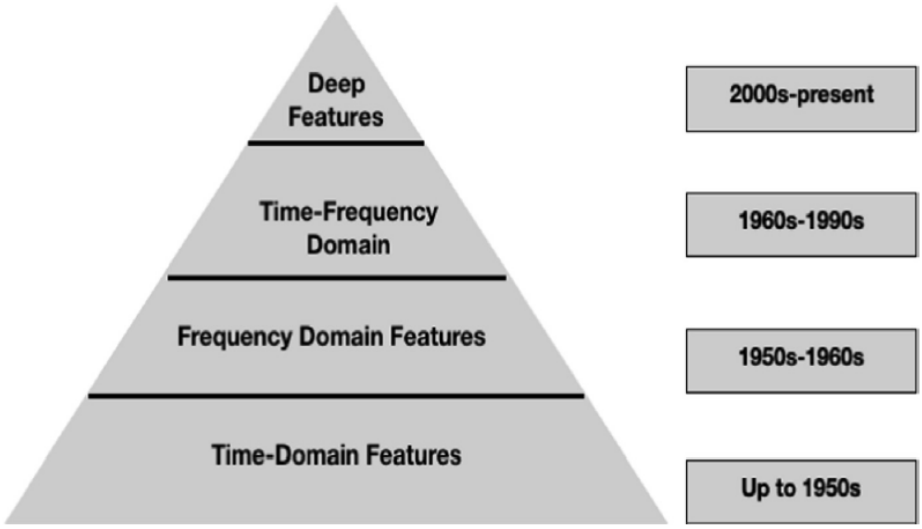


Figure 2.1: Evolution of audio feature extraction methods over time. Feature types are categorized into four stages: time-domain, frequency-domain, time-frequency domain, and deep features, corresponding to their respective periods of development. Source: [3].

are used in audio signal processing algorithms. Since the evolution of deep learning, the deep features are extensively used in various applications; in audio signal processing, deep features have been used since 2010 in the area of acoustic scene classification [36, 37], speaker recognition [38] and audio video analysis [9].

To illustrate the general process underlying most traditional and modern audio features, Figure 2.2 presents the typical feature extraction pipeline. This includes frame blocking, windowing, spectral analysis, and the transformation of frequency-based information into compact feature vectors for subsequent analysis.

2.4 Feature Extraction

Feature extraction involves emphasizing the most prominent and distinctive characteristics of a signal. An effective feature provides a compact representation that captures the essential properties of the signal. Figure 2.1 illustrates the progression of audio signal features, which can be broadly categorized into time-domain features, frequency-domain features, time-frequency domain features, and deep features [3]. These categories are explained in detail below. To represent all these features, we used the audio file ”1-97392-A-0.wav” from the ESC-50 dataset as a sample. The duration is 5 seconds.

2.4.1 Time Domain Features

Time-domain features are directly extracted from the raw audio waveform without transforming the signal into the frequency domain. These features are computationally efficient and provide

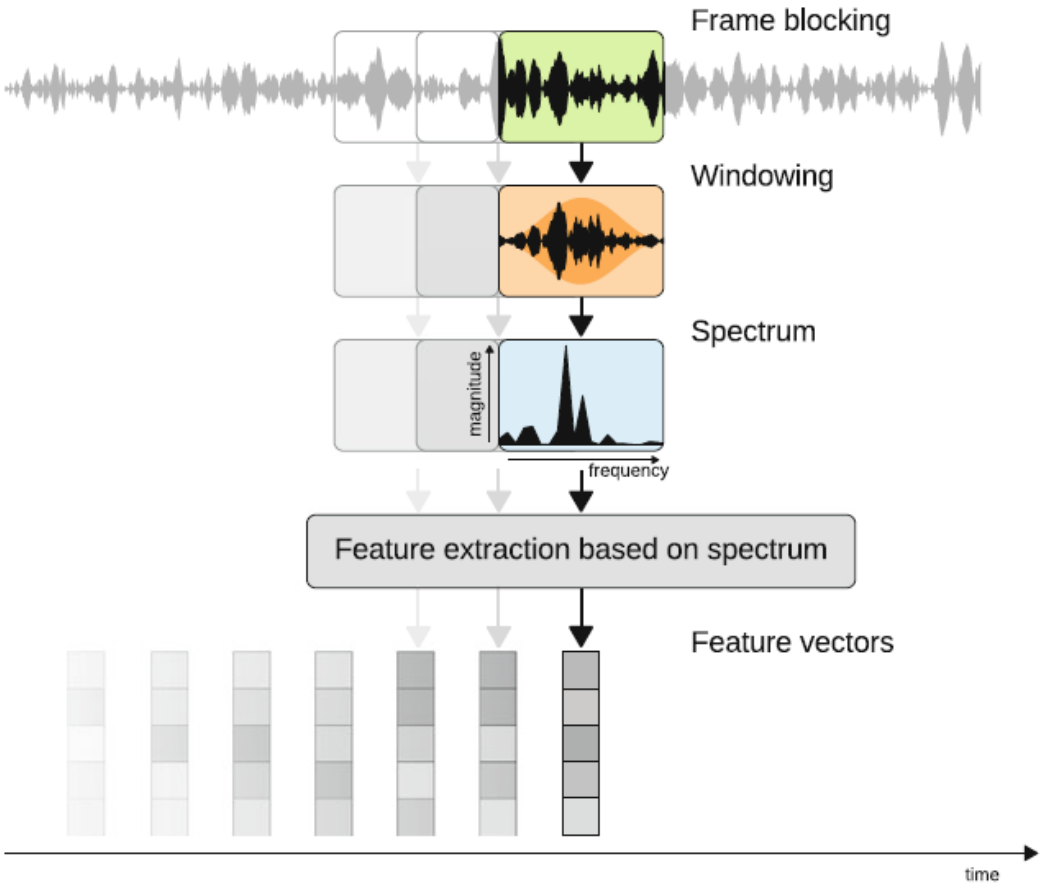


Figure 2.2: The processing pipeline of feature extraction. Source: [4].

useful information about the signal's amplitude and temporal structure. Two of the most widely used time-domain features are described below [39, 40].

Zero-Crossing Rate (ZCR): The Zero-Crossing Rate (ZCR) measures how frequently the audio signal changes sign—i.e., crosses the zero amplitude level—within a given analysis frame. It is particularly effective for detecting percussive or transient sounds such as clicks, claps, or fricatives [39]. The ZCR is computed as:

$$\text{ZCR} = \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbf{1}[x_t \cdot x_{t+1} < 0] \tag{2.1}$$

where:

- T is the total number of time samples in the analysis frame.
- x_t is the amplitude of the signal at time t.
- $\mathbf{1}[\cdot]$ is the indicator function, returning 1 if the condition is true and 0 otherwise.

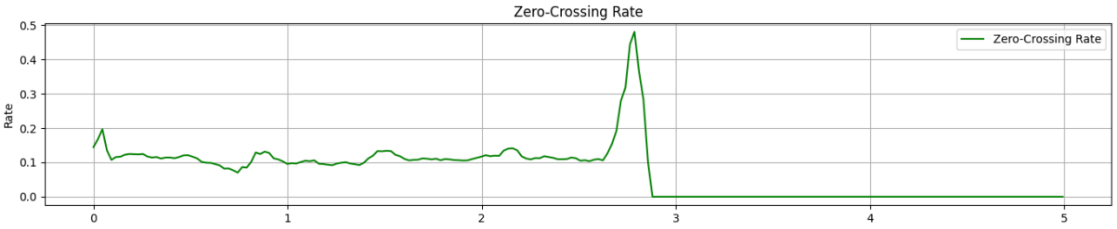


Figure 2.3: Time evolution of the Zero-Crossing Rate (ZCR) for the audio signal. Peaks indicate regions with higher frequency sign changes, typically associated with noisy or transient sounds. The flat region after 3 seconds suggests silence or a sustained tone with minimal zero crossings.

The term $x_t \cdot x_{t+1} < 0$ in Equation 2.1 checks for a sign change between consecutive samples, indicating a zero-crossing.

Figure 2.3 illustrates the time evolution of the ZCR for the input audio signal. In this example, the signal becomes silent after 2.8 seconds. Peaks in the curve correspond to segments with frequent sign changes, indicating higher energy transients or noisy events, while flat regions suggest silence or tonal stability.

Root Mean Square (RMS) Energy: The Root Mean Square (RMS) energy quantifies the average power of the signal over an analysis frame, as illustrated in Figure 2.4. It serves as a reliable indicator of signal loudness and is useful for distinguishing between high-energy and low-energy events, such as differentiating speech from silence or detecting loud sounds like explosions or sirens [41, 40]. RMS energy is computed as:

$$\text{RMS} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} x_t^2} \tag{2.2}$$

where:

- T is the total number of time samples in the frame.
- x_t is the amplitude value at time t.

As shown in Equation 2.2, RMS is the square root of the mean squared amplitude, providing a measure of the signal's intensity.

2.4.2 Frequency Domain Features

Frequency-domain features are extracted by transforming the audio signal from the time domain to the frequency domain, typically using methods such as the Discrete Fourier Transform (DFT). These features are essential for capturing the spectral characteristics of audio signals and are widely used in ESC as well as in general audio analysis tasks [39, 4, 40]. Below, some of the most commonly used frequency domain features are described.

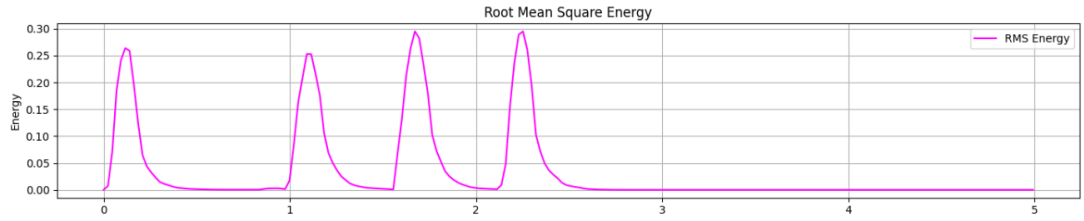


Figure 2.4: Time evolution of Root Mean Square (RMS) energy for the audio signal. The peaks indicate frames with high energy (loud or active segments), while near-zero values indicate silence or low-energy intervals.

Spectral Centroid (SC): The Spectral Centroid represents the "center of mass" of the spectrum and is correlated with the perceived brightness of a sound. Bright sounds such as whistles tend to have a higher spectral centroid, whereas dull sounds have a lower one. The spectral centroid is computed as:

$$\mathrm{SC} = \frac{\sum_k f_k \cdot |X(f_k)|}{\sum_k |X(f_k)|} \tag{2.3}$$

where:

- f_k is the frequency of the k^{th} bin,
- $|X(f_k)|$ is the magnitude of the spectrum at bin k.

Equation 2.3 provides a weighted average of the frequency bins based on their amplitude contributions.

Figure 2.5 shows the time evolution of the spectral centroid for a given audio signal. High peaks in the plot indicate segments with strong high-frequency components (e.g., transients or fricatives), while lower values represent tonal or low-frequency content.

Spectral Bandwidth (SB): The Spectral Bandwidth measures the dispersion or spread of the spectral energy around the spectral centroid. It gives an indication of how concentrated or wide the energy is distributed across frequencies. It is defined as:

$$\mathrm{SB} = \sqrt{\frac{\sum_k (f_k - C)^2 \cdot |X(f_k)|}{\sum_k |X(f_k)|}} \tag{2.4}$$

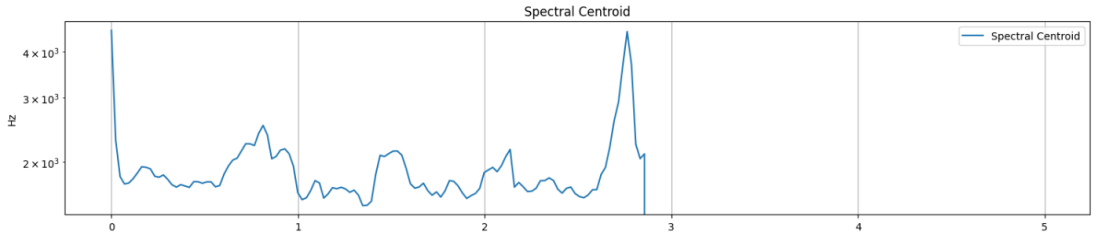


Figure 2.5: Time evolution of the Spectral Centroid for the input audio signal.

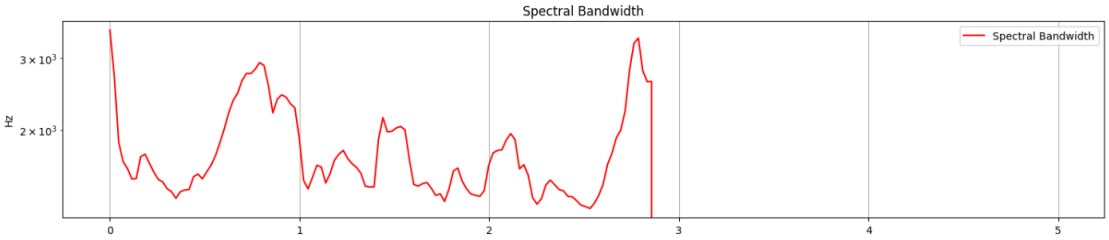


Figure 2.6: Time evolution of the Spectral Bandwidth.

where:

- f_k and $|X(f_k)|$ are the same as in Equation 2.3,
- C is the spectral centroid.

Equation 2.4 reflects how tightly or loosely the energy is packed around the centroid. Figure 2.6 shows the time evolution of the spectral bandwidth for the input audio signal. Spectral bandwidth measures the spread of the spectrum around the spectral centroid, indicating the distribution and complexity of frequency content. Peaks in the plot represent frames with wide spectral dispersion, often corresponding to noisy or transient-rich sounds, while lower values indicate narrowband or more tonal components.

Spectral Contrast (SCon): The Spectral Contrast describes the difference in energy between spectral peaks (harmonic content) and valleys (non-harmonic or noise content) in sub-bands of the spectrum. It is especially useful for distinguishing between tonal and noisy sounds [42]. Spectral contrast in sub-band f is calculated as:

$$\mathrm{SCon}(f) = 10 \cdot \log_{10}\left(\frac{P_{\max}(f)}{P_{\min}(f) + \varepsilon}\right) \tag{2.5}$$

where:

- $P_{\max}(f)$ is the maximum energy in the sub-band,
- $P_{\min}(f)$ is the minimum energy in the sub-band,
- ε is a small constant to avoid division by zero.

Equation 2.5 is applied to multiple frequency sub-bands, capturing the overall spectral shape and structure. Figure 2.7 shows a Time-frequency representation of Spectral Contrast. The vertical axis denotes the frequency sub-bands, while the horizontal axis represents time. Higher intensity (brighter areas) corresponds to greater spectral contrast—i.e., a larger difference between peaks and valleys—indicating tonal or structured components, whereas darker areas suggest flatter, noise-like spectral content.

While these features are based on frequency content, many are computed frame-by-frame — so they also show temporal evolution if applied in a sliding window, making them time-frequency features in practical usage.

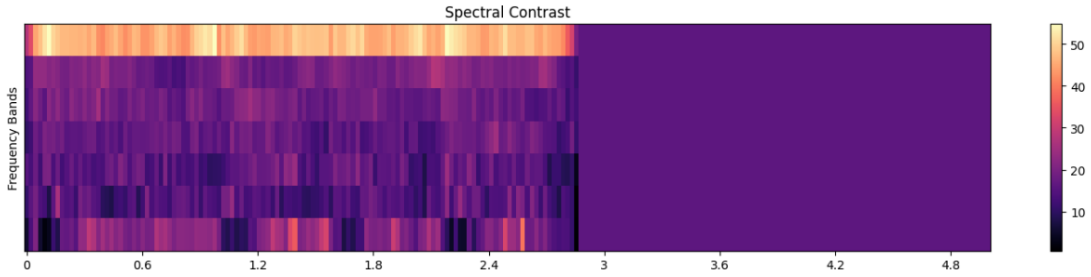


Figure 2.7: Representation of Spectral Contrast.

2.4.3 Time-Frequency Domain Features

Time-frequency representations are essential in ESC as they simultaneously capture how the frequency content of an audio signal evolves over time. Unlike purely time-domain or frequency-domain features, time-frequency representations provide a two-dimensional view, which is crucial for identifying transient, overlapping, or dynamic sound events [4, 41]. Below, some of the most commonly used time-frequency domain features are described.

2.4.3.1 Short-Time Fourier Transform (STFT)

The Short-Time Fourier Transform is a widely used method that divides the signal into overlapping frames and computes the Fourier Transform for each. This allows observation of how spectral content changes over time.

$$\mathrm{STFT}\{x(t)\}(f,t) = \int_{-\infty}^{\infty} x(\tau) w(t-\tau) e^{-j2\pi f \tau} d\tau \tag{2.6}$$

where:

- $x(t)$: Original time-domain signal.
- $w(t-\tau)$: Window function centered at t (e.g., Hamming).
- f: Frequency.
- τ: Time-shift variable.

Equation 2.6 outputs a 2D representation where the x-axis is time, the y-axis is frequency, and magnitude is represented as color or intensity. Figure 2.8 shows the Short-Time Fourier Transform (STFT) spectrogram of the input audio signal using a Hamming window and a hop size of 512 samples. The horizontal axis represents time in seconds, while the vertical axis represents frequency in Hertz. Color intensity encodes the magnitude of frequency components in decibels (dB). Brighter areas indicate higher energy at specific frequencies and times, revealing the time-frequency structure of periodic and transient events.

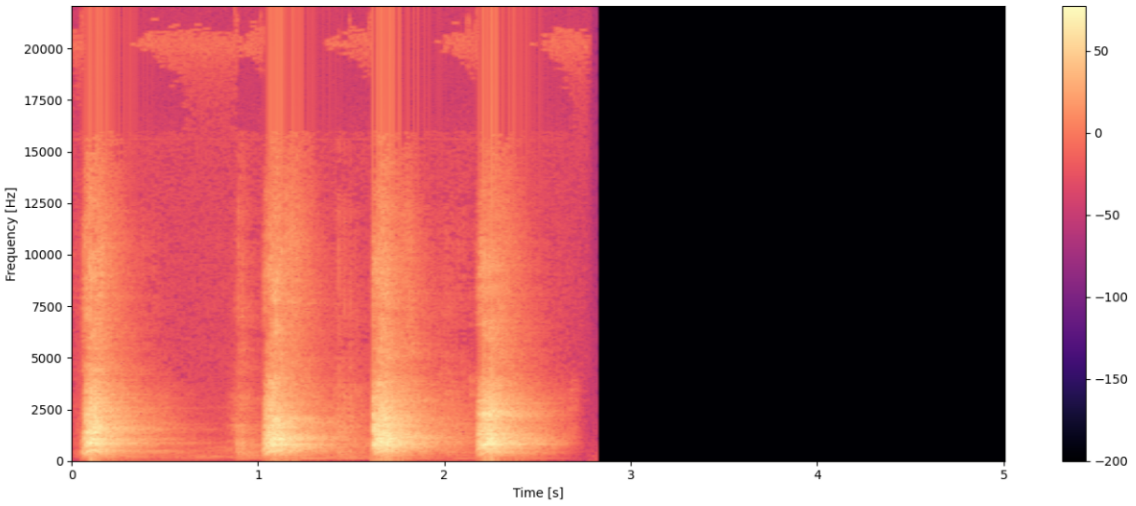


Figure 2.8: STFT spectrogram showing the time-frequency energy distribution of the audio signal

2.4.3.2 Mel Spectrogram

The Mel Spectrogram is a time-frequency representation that maps the frequency axis of a spectrogram onto the Mel scale—a perceptual scale where pitches are spaced according to human auditory perception [43]. This nonlinear scale offers finer resolution at lower frequencies and coarser resolution at higher ones, reflecting how we naturally perceive sound. The computation involves applying a Mel filter bank to the power spectrum, as shown in equation 2.7. To enhance interpretability and dynamic range, the Log-Mel Spectrogram (LM) applies a logarithmic transformation to the Mel-scaled power spectrum, compressing the large range of spectral energies into a more manageable scale. Figure 2.9 illustrates the Mel spectrogram of the input audio signal, where brighter regions indicate higher energy in perceptually significant frequency bands, particularly emphasizing lower-frequency content.

$$M(f,t) = \sum_{k} |X(f,t)|^2 \cdot m_k(f) \tag{2.7}$$

where:

- $|X(f,t)|^2$: Power spectrum.
- $m_k(f)$: Mel filter bank.

Computation steps include STFT, power spectrum calculation, Mel filter bank application, and logarithmic compression [44].

2.4.3.3 Mel-Frequency Cepstral Coefficients (MFCCs)

Mel-Frequency Cepstral Coefficients (MFCCs) are among the most widely used features in audio analysis, especially in tasks such as speech recognition and Environmental Sound Classification

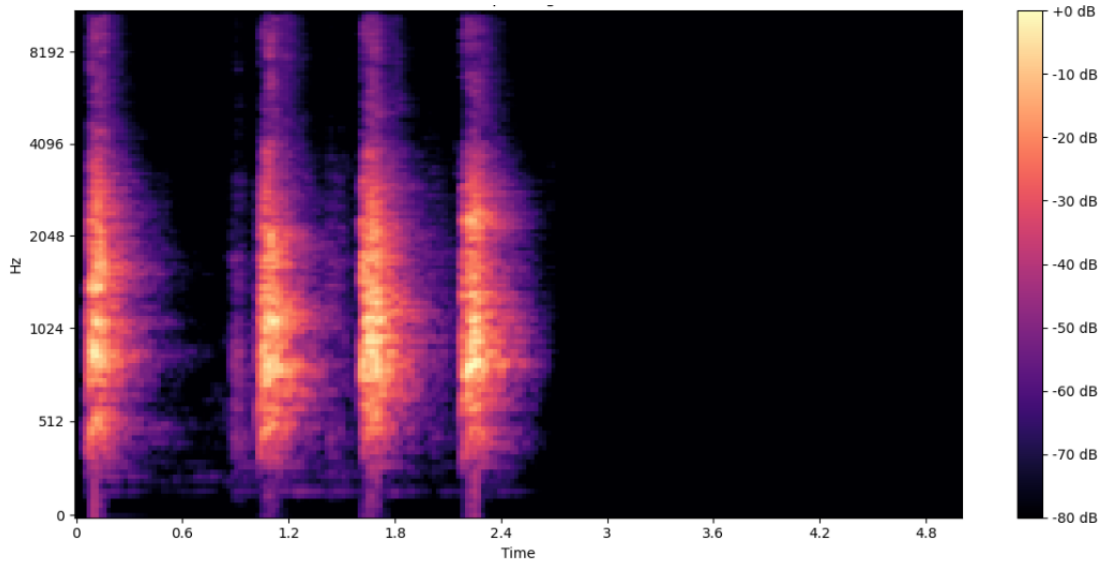


Figure 2.9: Mel spectrogram of the input audio signal showing the energy distribution across Mel-scaled frequency bands over time.

(ESC). They are designed to approximate the human auditory system's response by mapping the frequency axis to the perceptually motivated Mel scale [45, 44].

MFCCs capture the short-term power spectrum of an audio signal and emphasize frequencies that are more relevant to human perception. Figure 2.10 represents a sample MFCC representation. This makes them robust against noise and variations, and particularly effective in representing timbral properties of sounds [46]. The process of computing MFCCs involves the following key steps:

1. **Pre-emphasis**: Apply a high-pass filter to the raw signal to boost high-frequency components.

2. **Framing**: Segment the audio signal into short, overlapping frames (typically 20–40 ms).

3. **Windowing**: Apply a window function (e.g., Hamming window) to reduce spectral leakage.

4. **Fast Fourier Transform (FFT)**: Transform the windowed signal into the frequency domain.

5. **Mel Filter Bank**: Apply triangular filter banks spaced on the Mel scale to model auditory resolution as illustred in Figure 2.11.

6. **Logarithmic Compression**: Convert the filter bank energies into logarithmic scale.

7. **Discrete Cosine Transform (DCT)**: Apply DCT to decorrelate the coefficients and compress information into a few dimensions.

The n^{th} MFCC coefficient is computed as follows:

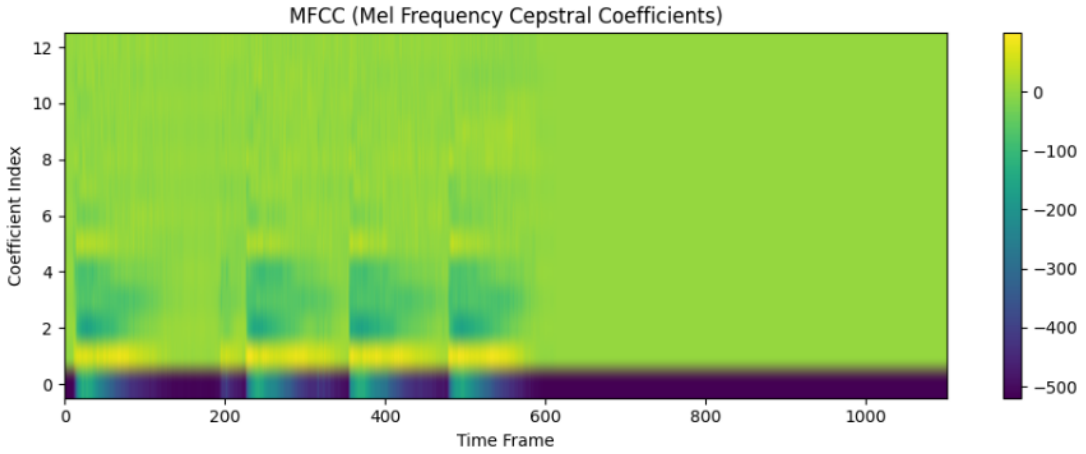


Figure 2.10: Visualization of MFCCs extracted from an audio signal. The x-axis represents time frames, the y-axis corresponds to cepstral coefficient indices, and the color intensity indicates the magnitude of each coefficient over time.

$$\text{MFCC}_n = \sum_{k=1}^{K} \log(S(k)) \cdot \cos\left[n \cdot \frac{(k-0.5)\pi}{K}\right] \tag{2.8}$$

where

- K: Total number of Mel filter banks.
- $S(k)$: Log energy of the k^{th} Mel-filtered spectral bin.
- n: Index of the cepstral coefficient.

Equation (2.8) represents the DCT step, which compresses the log-energy spectrum into a small number of decorrelated coefficients, typically the first 12 or 13. These coefficients form a compact representation of the spectral envelope, which is critical for recognizing both speech and environmental sounds. MFCCs have become a de facto standard in many audio classification tasks due to their efficiency and perceptual relevance [44].

2.4.3.4 Gammatone Frequency Cepstral Coefficients (GFCCs)

Gammatone Frequency Cepstral Coefficients (GFCCs) are a robust alternative to Mel-Frequency Cepstral Coefficients (MFCCs), particularly effective in noisy environments. Unlike MFCCs, which rely on Mel-scale filter banks, GFCCs are derived using a gammatone filter bank that closely mimics the filtering characteristics of the human cochlea [47]. This biologically inspired design enables GFCCs to better capture perceptually relevant features, especially in non-stationary or background-noise-rich conditions [48]. The process of computing GFCCs involves the following key steps:

1. **Framing and Windowing**: Divide the audio signal into short, overlapping frames, typically 20–40 ms, with a window function applied to minimize edge effects.

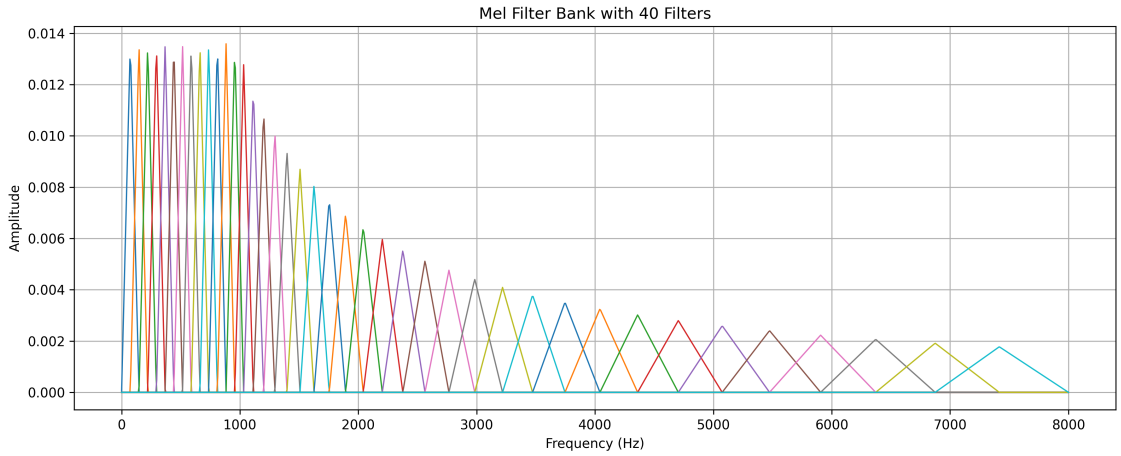


Figure 2.11: Mel-scale filter bank with 40 triangular filters, commonly used for computing Mel-frequency cepstral coefficients (MFCCs).

2. **Gammatone Filtering**: Pass each frame through a filter bank consisting of gammatone filters, each tuned to a specific center frequency.

3. **Energy Calculation**: Compute the energy of the output of each filter to form the auditory spectrum.

4. **Logarithmic Compression**: Apply a logarithmic function to mimic the non-linear response of human loudness perception.

5. **DCT Transformation**: Apply the Discrete Cosine Transform to decorrelate the features and compress them into a lower-dimensional representation.

The impulse response of a gammatone filter is given by:

$$G(t) = t^{n-1} e^{-2\pi bt} \cos(2\pi f_c t + \phi) \tag{2.9}$$

where :

- t: Time variable.
- n: Order of the filter (typically set to 4).
- b: Bandwidth of the filter.
- f_c: Center frequency.
- ϕ: Phase offset.

Equation (2.9) models the auditory filter shape as observed in human auditory experiments, making it particularly suitable for tasks like speech and environmental sound classification under variable acoustic conditions [49].

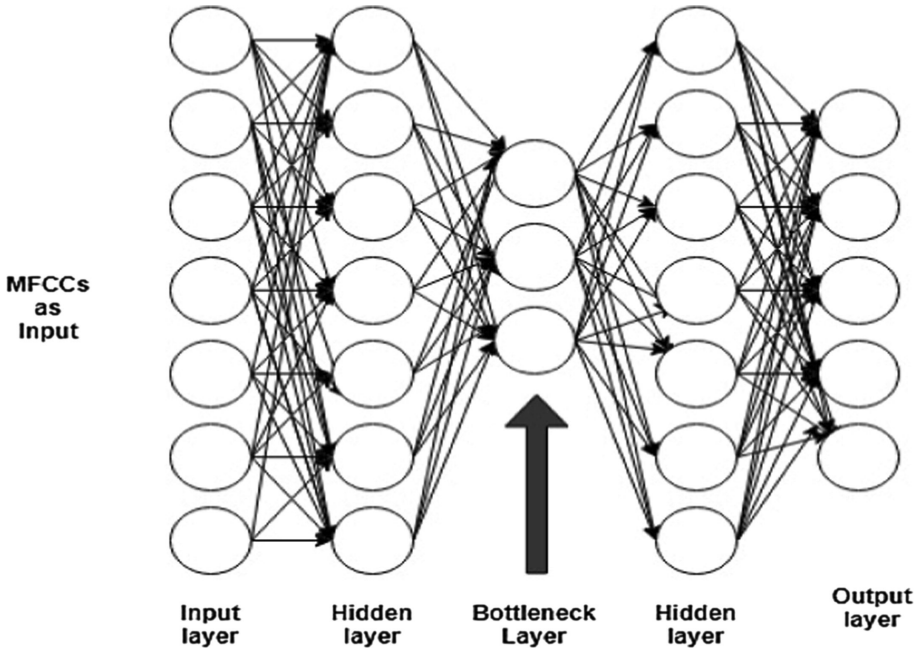


Figure 2.12: Deep feature from bottleneck layer. Source:[3].

In addition to commonly used features, several other perceptually and harmonically meaningful representations have been explored in environmental sound and music classification. One such feature is Tonnetz, a tonal representation derived from music theory that captures the harmonic relationships between pitches, such as perfect fifths and minor thirds. It provides a compact embedding of chroma vectors into a space that reflects tonal proximity and has been effectively used for harmonic change detection and music information retrieval [50, 51]. Another widely used feature is the Chroma vector, which represents the energy distribution across the 12 pitch classes (C, C♯, D, ..., B), irrespective of the octave in which they occur. Chroma features are particularly effective in capturing harmonic and melodic content and are commonly used in chord recognition, cover song identification, and key estimation [52].

2.4.4 Deep Features

Deep learning has emerged as a powerful tool for extracting high-level representations from low-level input data. In the context of audio analysis, the features obtained from the intermediate or hidden layers of deep learning models are commonly referred to as deep features. These features can be extracted from various neural network architectures, including Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), Stacked Autoencoders (SAEs), and Long Short-Term Memory networks (LSTMs), both unidirectional and bidirectional.

Typically, conventional audio features such as MFCCs or Mel-spectrograms are provided as input to a deep neural network. The quality and characteristics of the resulting deep features depend on the depth and architecture of the model. In shallower networks, the earlier layers often

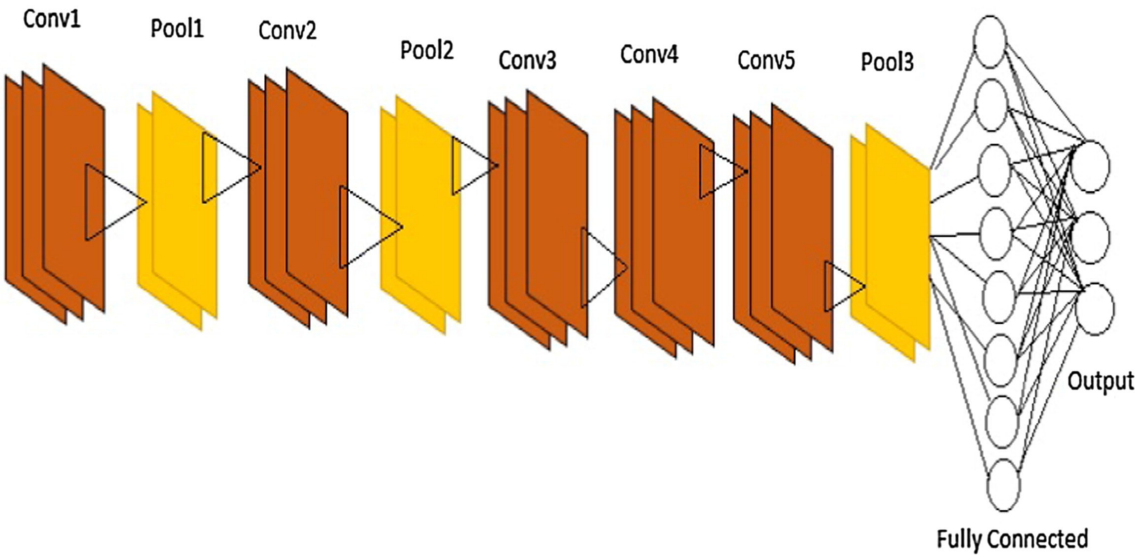


Figure 2.13: A typical Convolutional Neural Network (CNN) architecture used for feature extraction in audio analysis. Source:[3]

capture speaker-dependent or low-level characteristics, whereas deeper layers tend to extract class-specific, highly discriminative features. In some cases, bottleneck layers—narrow hidden layers within a deep architecture—are specifically designed to produce compact and informative deep features. Figure 2.12 illustrates how deep audio features (DAFs) are extracted from a bottleneck layer within a DNN.

In the case of CNNs, deep features can be extracted from any layer depending on the intended application. As illustrated in Figure 2.13, a typical CNN consists of three main components: convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply multiple filters to the input spectrogram, generating feature maps that highlight local patterns. Pooling layers reduce the dimensionality of these maps, lowering computational cost while retaining essential information. Fully connected layers synthesize these localized patterns into global representations. Deep features extracted from earlier convolutional layers typically capture low-level details such as edges or textures, while those from deeper convolutional or fully connected layers contain more abstract, semantic information.

Deep features have demonstrated strong performance across various audio signal processing tasks, including acoustic scene classification [36, 37], speaker recognition [38], audio-visual analysis [9], gender recognition [53], emotion recognition [54], and spoofing detection [55]. Their ability to capture rich, abstract representations makes them well-suited for modern machine learning pipelines in audio-based systems.

2.5 Traditional Models vs. Deep Learning Models

The task of Environmental Sound Classification (ESC) has traditionally been approached using classical machine learning algorithms such as Support Vector Machines (SVM), K-Nearest

Neighbors (KNN), and Random Forests (RF), combined with hand-engineered features like Mel-Frequency Cepstral Coefficients (MFCCs), Zero-Crossing Rate (ZCR), and Spectral Centroid [39, 40]. These traditional models are computationally efficient and relatively easy to implement; however, they rely heavily on the quality and discriminative power of the manually extracted features.

With the advent of deep learning, feature extraction and classification have become more tightly integrated. Deep neural networks, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and more recently Transformer-based architectures, are capable of learning complex representations directly from raw waveforms or time-frequency representations such as spectrograms and Mel spectrograms [21, 25, 12]. These models have shown superior performance compared to traditional approaches, particularly on large and diverse datasets.

Recent studies have demonstrated the performance gap between traditional and deep learning models. For example, in the ESC-50 benchmark dataset, shallow models such as SVM or RF typically achieve accuracies in the range of 40–60% depending on the features used, while CNN-based architectures have pushed performance above 75% in some configurations [56]. Additionally, large-scale pretraining (e.g., on AudioSet) and transfer learning have enabled even higher accuracies using pre-trained audio models like PANNs (Pretrained Audio Neural Networks) [12] and AST (Audio Spectrogram Transformer) [57].

While deep learning models require significantly more computational resources and training data, they offer several advantages: automatic feature learning, better generalization, and the ability to capture spatial and temporal patterns in audio signals. On the other hand, traditional models may still be preferable in resource-constrained environments or when working with small datasets, especially when combined with robust handcrafted features.

Overall, the trend in recent ESC research clearly favors deep learning models for their scalability and superior performance, particularly in complex and noisy acoustic environments.

Table 2.2: Test Accuracy of ML Models [6]

Model	**Feature**	**Test Accuracy**
RF	MFCC + Melspectrogram	62.50%
RF	Melspectrogram	58.00%
RF	MFCC	55.25%
SVM	MFCC	42.00%
KNN	MFCC	38.00%

Table 2.2 presents the test accuracies of several traditional machine learning models applied to environmental sound classification (ESC) using different combinations of acoustic features. Among the evaluated classifiers, Random Forest (RF) achieved the highest accuracy of 62.50% when using a fusion of MFCC and Mel Spectrogram features. This finding supports previous studies suggesting that feature-level fusion can improve performance by combining both temporal and spectral representations of audio [58]. When RF was trained with only Mel Spectrogram or

MFCC features, the accuracies dropped slightly to 58.00% and 55.25%, respectively, indicating that while each feature type is informative, their combination yields a richer representation.

In comparison, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models, both using MFCC features, achieved test accuracies of 42.00% and 38.00%, respectively. These results reflect the limitations of traditional shallow models in handling the complex, overlapping class distributions typically found in environmental audio datasets [40, 39]. Although MFCCs remain a widely used baseline due to their compact and perceptually meaningful representation of sound, their discriminative capacity may be insufficient for highly diverse soundscapes without further augmentation or feature transformation.

There has been growing research interest in Environmental Sound Classification (ESC) focused on combining multiple auditory features to enhance model performance. [11] proposed a CNN-based approach that leverages stacked features—specifically Log-Mel Spectrogram, Spectral Contrast, Chroma features, MFCC, and Tonnetz—visualized using the Librosa library. By stacking these features as separate channels and applying transfer learning to a convolutional architecture, the model effectively captured both spectral and temporal characteristics of environmental sounds. When evaluated on the UrbanSound8K dataset, the combination of Log-Mel Spectrogram, Spectral Contrast, and Chroma features achieved the highest validation accuracy of 95.13%, outperforming models based on single-feature inputs. These results demonstrate the effectiveness of feature fusion and transfer learning in improving ESC performance, particularly in noisy or complex urban settings.

2.6 Feature Selection Methods

Traditional feature selection methods have long been employed in statistical learning and data mining tasks to improve model interpretability and efficiency. These approaches typically operate independently of any learning algorithm and rely on intrinsic properties of the data—such as correlation, redundancy, or statistical dependency—to rank or filter features prior to model training. Although they are often overlooked in deep learning pipelines, recent studies suggest that their integration can enhance performance by reducing irrelevant feature dimensions and improving the input space quality for convolutional architectures in ESC and related domains [59].

2.6.1 Filter Methods

Filter methods are commonly used feature selection techniques that evaluate the relevance of features independently from the learning algorithm. These methods measure statistical relationships between individual features and the target variable, thus allowing the removal of irrelevant or redundant features before model training. Two widely used statistical criteria in filter methods are cross-correlation and mutual information.

2.6.1.1 Cross-correlation

Cross-correlation assesses linear relationships between pairs of features and targets, quantifying the degree of similarity in their temporal or spatial behavior. High absolute values of cross-correlation indicate strong linear dependencies, enabling the identification of highly redundant or irrelevant features that can be excluded from further analysis [60]. The Pearson correlation coefficient between two feature vectors x and y is computed as:

$$\rho_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.10}$$

where $\bar{x}$ and $\bar{y}$ denote the mean values of the respective vectors.

2.6.1.2 Mutual Information

Mutual Information (MI) captures both linear and nonlinear dependencies between variables. It quantifies how much knowing one variable reduces uncertainty about another, thus identifying features that contain significant information regarding the class label. MI between variables X and Y is defined as:

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{2.11}$$

where $p(x,y)$ is the joint probability distribution of X and Y, and $p(x)$, $p(y)$ are the marginal probabilities. Mutual information has been widely used for feature selection in audio classification tasks [61, 10].

2.6.2 PCA-based Method

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms high-dimensional data into a set of orthogonal components, each capturing a portion of the variance present in the original dataset. The transformation is performed by solving the eigenvalue decomposition of the covariance matrix:

$$\Sigma \mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{2.12}$$

where Σ is the covariance matrix of the input data, $\mathbf{v}_i$ are the eigenvectors (principal components), and λ_i are the corresponding eigenvalues. The projection of data onto a lower-dimensional subspace is then computed by:

$$Z = XW_k \tag{2.13}$$

where X is the centered input data matrix and W_k is the matrix formed by the top-k eigenvectors.

In the context of ESC, PCA can significantly improve computational efficiency and classification accuracy by removing redundant and correlated features [62]. By projecting audio features onto a lower-dimensional space, PCA facilitates more robust and generalized classifier performance, making it particularly beneficial for applications with limited computational resources.

2.6.3 Sparse Salient Region Pooling

Sparse Salient Region Pooling (SSRP) is a global pooling strategy to improve the efficiency and performance of lightweight CNNs for environmental sound classification. Unlike traditional pooling methods such as Global Average Pooling (GAP), which treat all time-frequency regions equally, SSRP focuses on a sparse set of the most salient regions in the feature map. This selective pooling mechanism enhances the model's ability to capture class-specific acoustic patterns while maintaining low complexity and no additional learnable parameters. SSRP has been shown to outperform conventional pooling methods in both accuracy and efficiency across multiple ESC benchmarks [7]. Table 2.3 shows a comparison between GAP and SSRP.

Table 2.3: Comparison of CNN models using GAP and SSRP pooling on ESC-10 and ESC-50 datasets [7].

Model	ESC-10 Accuracy	ESC-50 Accuracy	Feature
CNN + GAP	90.60%	70.30%	Log-Mel Spectrogram
CNN + SSRP_B	94.20%	84.10%	Log-Mel Spectrogram
CNN + SSRP_T	94.80%	84.70%	Log-Mel Spectrogram

2.7 Transformer-Based Models for ESC

In recent years, transformer-based architectures have emerged as promising alternatives to convolutional neural networks (CNNs) for audio classification tasks, thanks to their ability to model long-range dependencies and capture global contextual information. While CNNs have been successfully applied to spectrograms and Mel-frequency cepstral coefficients (MFCCs), they often face challenges in effectively modeling temporal dynamics across extended time scales. To address this, researchers proposed the Spectrogram Transformer framework, introducing a family of models that integrate domain-specific sampling techniques to better utilize the structural properties of spectrograms [8]. Specifically, they introduced time-dimension sampling and frequency-dimension sampling to separately capture temporal and spectral characteristics. These embeddings were then processed through a series of novel transformer architectures, including the Temporal Only (TO) Transformer, which applies self-attention exclusively in the temporal domain. They further designed other variants—Temporal-Frequency Sequential (TFS), Temporal-Frequency Parallel (TFP), and Two-stream Temporal-Frequency (TSTF)—each combining temporal and frequency attention in different ways. Among these, the TO Transformer achieved a top-1 accuracy of 57.17% on the ESC-50 dataset without any pre-training, significantly outperforming prior transformer-based models such as the AST (Audio Spectrogram Transformer), which achieved

45.35% accuracy under the same conditions [57]. AST, originally based on the Vision Transformer (ViT) architecture, treats spectrogram patches as visual tokens and relies heavily on large-scale pre-training (e.g., ImageNet) to perform well. In contrast, Zhang et al.'s TO Transformer achieves higher accuracy with a simpler 6-layer architecture and nearly half the parameters, demonstrating both effectiveness and computational efficiency. These results highlight the potential of integrating task-specific spectrogram representations with transformer architectures for improving environmental sound classification. Table 2.4 presents the Top-1 accuracy results of various transformer-based models evaluated on the ESC-50 dataset,

Table 2.4: Performance comparison of transformer models on ESC-50 dataset (Top-1 Accuracy %) [8]

Model	**fold 1**	**fold 2**	**fold 3**	**fold 4**	**fold 5**	**Average**
AST	45.00	44.50	44.25	48.25	44.75	45.35
TO	56.00	51.25	62.36	61.75	54.50	57.17
TFS	45.75	45.75	52.68	50.25	46.00	48.09
TFP	49.50	47.00	59.13	56.50	52.25	52.88
TSTF	56.25	52.50	61.72	60.25	55.50	57.24

2.8 Categorical Cross-Entropy Loss Function

The categorical cross-entropy loss function is a widely used objective function in multi-class classification problems, where each input belongs to one of several mutually exclusive classes. It measures the difference between the true label distribution, typically represented as a one-hot encoded vector, and the predicted probability distribution produced by the model using a softmax activation layer. A one-hot encoded vector is a binary vector of length equal to the number of classes, where all elements are 0 except for the position corresponding to the true class, which is set to 1. For example, if there are 5 classes and the correct class is class 3, the one-hot vector would be $[0,0,1,0,0]$. Mathematically, for a classification task with C classes, the categorical cross-entropy loss for a single sample is defined as:

$$\mathscr{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

where y_i is the true label (1 for the correct class, 0 otherwise), and $\hat{y}_i$ is the predicted probability for class i. Given that y_i is one-hot encoded, the loss simplifies to the negative log of the predicted probability assigned to the correct class:

$$\mathscr{L} = -\log(\hat{y}_{\text{true class}})$$

This formulation penalizes incorrect or low-confidence predictions, encouraging the model to assign a higher probability to the correct class. Categorical cross-entropy is convex with respect

to the predicted probabilities, which makes it suitable for optimization via gradient-based methods.The function is widely adopted in deep learning for domains such as image classification, audio event detection, and natural language processing. Its effectiveness lies in its connection to maximum likelihood estimation, where minimizing cross-entropy is equivalent to maximizing the likelihood of the correct label under the predicted distribution [63]. In practice, this loss is combined with a softmax output layer, which ensures that the model's outputs form a valid probability distribution. It is especially useful in applications like environmental sound classification, where accurate prediction of mutually exclusive labels is essential [64].

2.9 Deep Reinforcement Learning Framework for Audio-Based Applications

Figure 2.14 illustrates a schematic overview of Deep Reinforcement Learning (DRL) agents applied to audio-based applications [5]. In this framework, the environment consists of audio signals—such as raw waveforms or their corresponding time-frequency representations (e.g., spectrograms). These inputs are first processed through deep learning models such as Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), or Recurrent Neural Networks (RNNs), which serve as feature extractors. The resulting feature representations define the agent's perception of the environment state. Based on these states, the RL agent generates outputs known as actions, which are application-specific and may include selecting an audio label (in classification tasks), triggering an alert (in surveillance systems), modifying a signal processing parameter, or choosing a sequence of transformations or decision rules in an adaptive pipeline. These actions are then evaluated by the environment in the form of rewards. These rewards are used to update the agent's policy. Depending on the learning strategy, DRL agents can be value-based, policy-based, or model-based. The interaction loop continues iteratively, enabling the agent to learn an optimal policy that maximizes cumulative rewards by improving its audio-driven decision-making capabilities.

2.10 Evaluation Metrics

Evaluation metrics are critical for assessing the performance and generalization capabilities of sound classification systems. In the domain of Environmental Sound Classification (ESC), selecting appropriate evaluation criteria ensures that models are robust, accurate, and comparable across different datasets and tasks [65, 28, 13].

2.10.1 Accuracy

Accuracy is the most straightforward and commonly used metric. It measures the proportion of correctly classified audio samples out of the total number of samples:

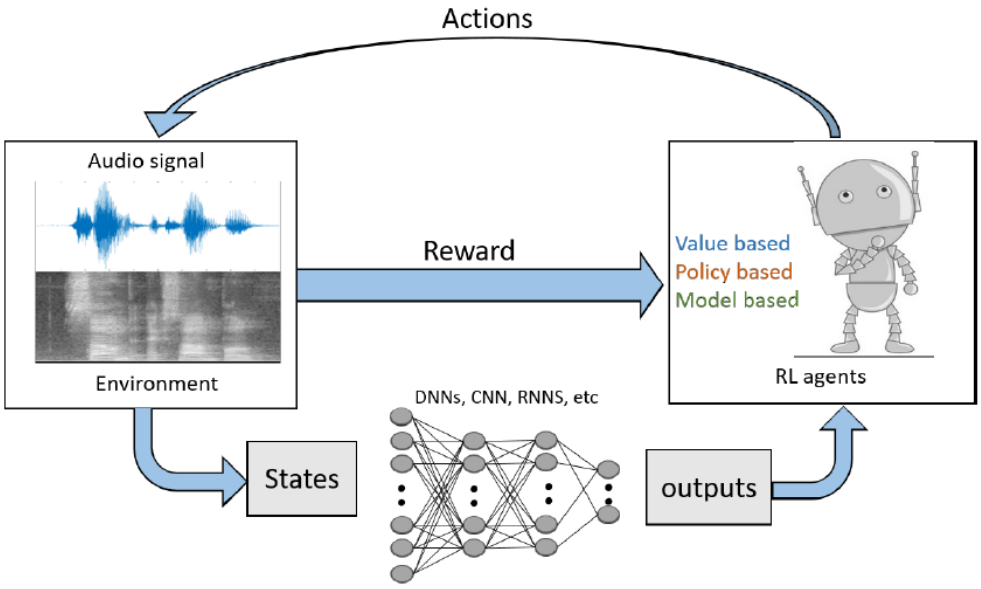


Figure 2.14: Schematic diagram of DRL agents for audio-based applications, where the DL model (via DNNs, CNN, RNNs, etc.) generates audio features from raw waveforms or other audio representations for taking actions that change the environment from state s_t to a next state s_{t+1} [5].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.14}$$

where TP, TN, FP, and FN refer to true positives, true negatives, false positives, and false negatives, respectively. Although useful, accuracy may not provide a complete picture in imbalanced datasets [15].

2.10.2 Precision, Recall, and F1-Score

These metrics provide deeper insight, especially in multi-class and imbalanced settings.

Precision is the proportion of correctly predicted positive observations to the total predicted positives:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.15}$$

Recall (or sensitivity) is the proportion of correctly predicted positive observations to all actual positives:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.16}$$

F1-Score is the harmonic mean of precision and recall:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.17}$$

These metrics are particularly useful for ESC systems where some sound classes are under-represented [66].

2.10.3 Confusion Matrix

The confusion matrix is a visualization tool that shows the performance of a classification model in terms of actual versus predicted labels. It is particularly helpful for identifying which sound classes are often confused with one another, offering insight into model weaknesses [67].

2.10.4 Evaluation Protocols

To ensure fairness and reproducibility, standard evaluation protocols such as cross-validation (e.g., 5-fold or 10-fold) or fixed train-test splits (as in ESC-50 and UrbanSound8K) are often employed. These protocols help assess generalization performance and reduce the risk of overfitting [27, 28]. In summary, choosing the right evaluation metric depends on the dataset, the balance between classes, and the specific goals of the ESC system. For comprehensive assessment, multiple metrics are often used in conjunction to provide both global and class-wise performance insights [67, 13].

Chapter 3

Impact of Stacked Features and Pretrained Models on Classification Performance

This chapter investigates the effectiveness of using stacked acoustic features to enhance environmental sound classification (ESC). By combining complementary features, the approach aims to improve the robustness and generalization of CNN models. This chapter begins in Section 3.1 by outlining the feature extraction and preprocessing process, with a special focus on the stacked feature approach. Section 3.2 presents the deep learning architectures used in this study. Section 3.3 details the experimental setup, covering dataset organization, model parameters, and training protocols. Section 3.4 provides a comprehensive results and analysis section. Finally, Section 3.5 concludes the chapter with key findings and insights drawn from the comparative experiments.

3.1 Feature Extraction and Preprocessing

To extract meaningful features from audio signals, we utilized the Librosa library [68], which provides multiple methods for feature extraction in environmental sound classification. Extracted features include cepstral-based features, spectral features, tonal features, which are essential for capturing various acoustic properties of environmental sounds. We investigated stacked feature inputs to leverage complementary spectral, cepstral, tonal, and auditory characteristics to enhance CNN-based ESC performance. Inspired by the effectiveness of feature aggregation shown in prior research [11], we selected several combinations to provide diverse acoustic perspectives. They are presented in Table 3.1.

After extracting individual features, we aggregate them into a unified representation to serve as input to the CNN models. This process, known as feature stacking, involves combining multiple spectrogram-based features into a multi-channel image-like structure. Each extracted feature—such as MFCC, GTCC, Log-Mel Spectrogram, Chroma, Tonnetz, or Spectral Contrast—is initially represented as a two-dimensional array corresponding to the time-frequency domain.

Table 3.1: Feature Configurations and their Descriptions.

Feature Configuration	**Description**
LM	Serves as a baseline, capturing spectral energy distribution.
LM + TZ	Combines spectral and tonal features, improving discrimination between harmonic and non-harmonic sounds.
LM + SPC + CH	Emphasizes distinctions between harmonic-rich, percussive, and transient sounds through combined spectral contrast and pitch class energy.
MFCC + LM	Integrates cepstral and spectral information, enhancing classification across environmental sounds.
MFCC + TZ	Enhances the classification of structured (tonal) and unstructured (non-tonal) sounds, such as distinguishing musical notes from random noise.
MFCC + GTCC + CH + LM	Provides the most comprehensive representation, integrating cepstral, spectral, tonal, and auditory system characteristics, thus significantly improving robustness to noise and enhancing overall classification accuracy.

However, the shape of these arrays can differ across features. To make them compatible for stacking, we first resize all feature maps to a common dimension of 128×128. This resizing is achieved either through zero-padding or interpolation, depending on the original shape of the feature. Once all features share the same spatial dimensions, they are concatenated along a new axis representing channels, similar to how RGB channels are arranged in color images.

For instance, when combining MFCC, GTCC, and Log-Mel Spectrogram, the resulting input tensor has a shape of $128 \times 128 \times 3$, with each "channel" corresponding to one of the three features. In the case of MFCC + GTCC + CH + LM, the resulting stacked input has a shape of $128 \times 128 \times 4$. This approach enables the model to learn from multiple acoustic perspectives simultaneously, capturing spectral, cepstral, and tonal characteristics of the sound. Additionally, the image-like structure of the input makes it naturally suited for convolutional neural networks, which excel at processing spatially correlated data. As shown in our experiments, this method of aggregation is both computationally efficient and effective for improving classification performance.

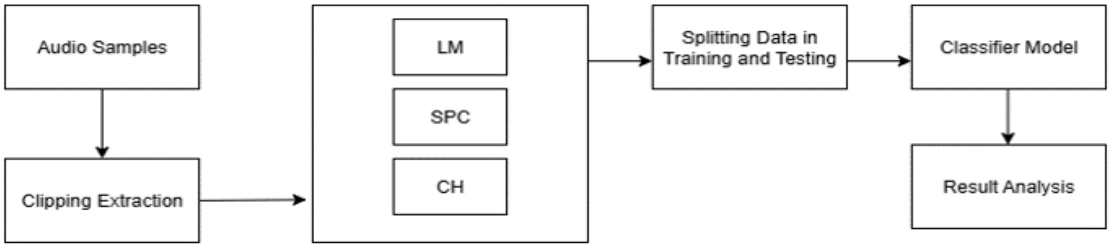


Figure 3.1: Workflow of the environmental sound classification system.

3.2 Deep Learning Architectures

We employed various architectural models, which are explained in detail below. As shown in Figure 3.1, the process begins with audio samples, followed by clipping extraction. Features are then extracted using log-Mel spectrograms (LM), spectral contrast (SPC), and chroma features (CH). The data is split into training and testing sets before being fed into a classifier model, and the results are finally analyzed.

3.2.1 Convolutional Neural Network (CNN) architectures

We explore two CNN architectures designed for environmental sound classification. Both architectures are tailored to process multi-channel feature representations derived from stacked spectral, cepstral, and tonal features, enabling them to learn discriminative patterns in environmental sounds.

3.2.1.1 CNN Model 1 (Baseline CNN)

The first model (CNN1) as we have shown in Figure 3.2-a , is a straightforward yet effective CNN architecture consisting of four convolutional layers, followed by global average pooling and dense layers. This design is intended to efficiently extract hierarchical features while maintaining a balance between model complexity and computational efficiency. The architecture comprises four convolutional layers with small 2×2 kernels, utilizing the ReLU activation function. Feature extraction starts with 32 filters in the initial layers and progresses to 64 filters, enabling the network to learn increasingly complex representations. Each convolutional layer is followed by max pooling, which reduces the spatial dimensions of the feature maps, minimizing computational cost while preserving essential patterns. To further refine the extracted features, a global average pooling layer aggregates spatial features into a single feature vector, reducing dimensionality before being fed into the dense layers. The final classification is performed using a 1024-unit dense layer with ReLU activation, followed by a softmax layer for multi-class sound recognition. The model is compiled using the Adam optimizer with a learning rate of 0.001 and is trained using the categorical cross-entropy loss function to optimize performance in multi-class classification.

3.2.1.2 CNN Model 2 (Enhanced CNN with Batch Normalization and Dropout)

The second model (CNN2) as we have shown in Figure 3.2-b, is an enhanced CNN architecture that incorporates batch normalization and dropout to improve training stability and generalization. While it follows a similar structure to CNN1, it introduces additional regularization techniques to mitigate overfitting and optimize learning efficiency. In this architecture, Batch Normalization is applied after each convolutional and dense layer to normalize activations, improve gradient flow, and accelerate training convergence. Dropout (0.25 probability) incorporated after the second and fourth convolutional layers to reduce overfitting by preventing co-adaptation of neurons. Global

Average Pooling is utilized to reduce feature dimensionality while retaining essential spatial information before the dense layers. A 1024-unit dense layer with batch normalization and ReLU activation is followed by a softmax classifier for multi-class sound recognition.

The selection of architectural parameters for both CNN1 and CNN2 was guided by a combination of empirical tuning and insights from prior research. Initially, we experimented with various CNN configurations using different numbers of convolutional layers, kernel sizes, and filter counts. Through systematic testing on the ESC-50 dataset, we observed that four convolutional layers with small kernel sizes (2×2) provided a good balance between feature learning capacity and overfitting prevention. Similarly, the use of 32 and 64 filters in early and deeper layers, respectively, was found to yield strong performance while keeping the model computationally efficient. The choice of global average pooling and dense layers was inspired by common best practices in audio classification tasks and further validated through experimentation. The dropout rate (0.25) and use of batch normalization in CNN2 were adopted from standard regularization strategies as explored in [69], [25], and similar audio classification works. We also tested different dropout rates (0.3, 0.5) and optimizer learning rates (e.g., 0.0001, 0.0005, 0.001), ultimately selecting those that led to stable convergence and the best validation performance. Thus, the final CNN configurations were the result of a combination of design inspiration from prior works and hyperparameter tuning on our development set. This process ensured that the models were both effective and generalizable to the target classification tasks.

3.2.2 Audio Spectrogram Transformer (AST)

The AST is a deep learning model designed for audio classification and analysis. It adapts the transformer architecture from Natural Language Processing (NLP) to spectrogram-based audio representations, offering an alternative to CNN-based methods. AST processes audio by dividing spectrograms into patches and applying a self-attention mechanism to capture long-range dependencies, enabling more effective feature learning. AST leverages pretrained models on large-scale datasets (e.g., AudioSet) and can be fine-tuned for domain-specific applications. Our AST architecture processes audio data represented as Mel-spectrograms. First, each spectrogram is partitioned into patches using a convolutional layer with a patch size of 16×18, embedding each patch into a 192-dimensional feature vector. These embedded patches are then concatenated with a learnable classification token (CLS token) and augmented by positional embeddings to preserve spatial context. Subsequently, the embeddings pass through a transformer encoder composed of 4 layers, each containing multi-head self-attention with 6 heads and a Feed-Forward Network (FFN) with a hidden dimension of 384. Each encoder layer incorporates dropout (rate=0.3), layer normalization, and GELU activations, enhancing generalization and convergence stability during training.

The classification task leverages the CLS token from the final encoder output, further processed through layer normalization and dense layers. Specifically, a fully-connected layer of 512 units followed by dropout (rate=0.4) is used, culminating in a softmax classifier for multi-class sound recognition across 50 classes. To optimize training, the AST model utilizes the AdamW

optimizer with weight decay (1e-4) and employs cosine learning rate decay from an initial rate of 5e-5. Additional regularization techniques include global gradient clipping to improve training stability. Training efficiency and convergence are further enhanced by enabling mixed-precision and XLA acceleration.

3.2.3 Retraining the models using UrbanSound8K dataset

To assess the generalization capability of the pre-trained model, we fine-tune it using the UrbanSound8K dataset, a widely used benchmark for environmental sound classification. This retraining process is designed to adapt the ESC-50 pre-trained model to a new dataset while preserving its previously learned feature representations, thus improving its performance on different types of audio data. Instead of training the model from scratch, we utilize transfer learning. This involves leveraging the knowledge embedded in the ESC-50 model and modifying its classification head to align with the specific label structure of UrbanSound8K. By doing so, we efficiently adapt the model to a new domain while retaining valuable learned features from the original training process.

The input structure of the model remains unchanged, with retraining applied exclusively to the final classification layer. This selective fine-tuning approach ensures that the lower-level feature extraction layers, which have already learned meaningful audio representations, remain intact, allowing the model to generalize well across different datasets. To achieve optimal adaptation, the model is trained for 50 epochs with a batch size of 32, striking a balance between stability and convergence speed. This training strategy enables the model to refine its decision boundaries for the new dataset while preventing overfitting and ensuring that previously learned knowledge is effectively transferred.

3.3 Experimental Setup

To evaluate the performance of our proposed approach for environmental sound classification, we conducted a series of experiments using two CNN architectures (CNN-1 and CNN-2) and the UrbanSound8K and ESC-50 datasets. Our experimental setup involves four main stages: data preprocessing, feature extraction, model training (including pre-training and fine-tuning), and performance evaluation. For data preprocessing, all audio files were resampled to 22,050 Hz for consistency. Each clip was trimmed or zero-padded to a fixed length of 3 seconds. Feature values were normalized to have zero mean and unit variance.

The dataset was then divided into 80% training and 20% testing, using stratified sampling [1] to preserve class distribution. During feature extraction, we converted each audio clip into multiple time-frequency representations, including MFCC, GTCC, Chroma, Tonnetz, Spectral Contrast, and Log-Mel spectrogram. These features, originally varying in shape, were resized to a uniform

[1] Stratified sampling is a technique used to ensure that each class is represented proportionally in both the training and testing sets. This helps maintain the original class distribution and results in more reliable and balanced model evaluation.

dimension of 128 × 128 using zero-padding or interpolation. The processed feature maps were then stacked to create a multi-channel input representation analogous to RGB images.

For model training and fine-tuning, each CNN model was first pre-trained on the ESC-50 dataset and subsequently fine-tuned on the UrbanSound8K dataset using transfer learning. During this phase, the pre-trained convolutional layers were frozen to preserve learned representations, and only the final classification layers were retrained. The models were trained using the categorical cross-entropy loss function and the Adam optimization algorithm. Accuracy was used as the main evaluation metric. Training was performed over 150 epochs on ESC-50 and 50 epochs for fine-tuning on UrbanSound8K, with a batch size of 32. Model validation was performed using the held-out test set [2] to assess generalization. The model is implemented using Google Colab [3] as the development environment. The deep learning models are built and trained using the Keras API with TensorFlow as the backend. Librosa is used for audio signal processing and feature extraction.

3.4 Results and Analysis

Figure 3.3 to Figure 3.8 illustrates the training and validation accuracy and loss of the CNN-1 and CNN-2 models during fine-tuning on the UrbanSound8K dataset, following pre-training on ESC-50, when different input features are used. The overall performance comparison of these models, including precision, recall, F1-score, and training setup details, is summarized in Table 3.2.

3.4.1 Performance Trends and Observations

In this subsection, we provide a comprehensive analysis of the experimental findings, focusing on the performance behavior of the proposed models under various conditions.

3.4.1.1 Impact of Feature Stacking on Accuracy

Single-feature models (e.g., LM alone) achieved reasonable accuracy, with CNN-2 reaching 85.45 percent validation accuracy and CNN-1 reaching 86.83%. Feature fusion significantly improved classification accuracy. CNN-1 fine-tuned on MFCC+GTCC+CH+ML achieved 92.46% validation accuracy, making it the best-performing model. CNN-2 with the same features reached 86.71%, confirming that multi-feature representations enhance classification performance. MFCC+TZ alone performed well, with CNN-1 reaching 91.29% accuracy, highlighting the importance of cepstral features.

[2] A held-out test set is a portion of the data that is excluded from training and used only once after training to evaluate the model's generalization performance on unseen data.

[3] https://github.com/ParinazBinandeh1986/Environmental-Sound-Classification-ESC-

Table 3.2: Performance comparison of CNN models

Model	Features	Training Setup	B.N	Val.Acc	Train.Acc	Precision	Recall	F1-score	Epochs
1	LM	ESC, All.L	No	0.68	1.00	0.68	0.68	0.66	150
1	LM+TZ	ESC, All.L	No	0.65	1.00	0.66	0.66	0.64	150
1	LM+MFCC	ESC, All.L	No	0.64	1.00	0.68	0.64	0.63	150
1	MFCC+TZ	ESC, All.L	No	0.62	1.00	0.65	0.62	0.61	150
1	LM+SPC+CH	ESC, All.L	No	0.62	1.00	0.65	0.62	0.62	150
1	MFCC+GTCC+CH+LM	ESC, All.L	No	0.67	1.00	0.70	0.67	0.67	150
2	LM	ESC, All.L	Yes	0.45	0.68	0.59	0.45	0.44	150
2	LM+TZ	ESC, All.L	Yes	0.66	0.98	0.71	0.67	0.65	150
2	LM+MFCC	ESC, All.L	Yes	0.59	0.99	0.64	0.59	0.59	150
2	MFCC+TZ	ESC, All.L	Yes	0.62	0.97	0.69	0.63	0.61	150
2	LM+SPC+CH	ESC, All.L	Yes	0.53	0.81	0.67	0.54	0.54	150
2	MFCC+GTCC+CH+LM	ESC, All.L	Yes	0.58	0.97	0.67	0.59	0.56	150
2	LM	ESC+US8K, Last.L	Yes	0.85	0.91	0.86	0.85	0.85	50
2	LM+TZ	ESC+US8K, Last.L	Yes	0.85	0.89	0.85	0.85	0.85	50
2	LM+MFCC	ESC+US8K, Last.L	Yes	0.86	0.90	0.87	0.86	0.86	50
2	MFCC+TZ	ESC+US8K, Last.L	Yes	0.87	0.92	0.87	0.87	0.87	50
2	LM+SPC+CH	ESC+US8K, Last.L	Yes	0.85	0.89	0.86	0.85	0.85	50
2	MFCC+GTCC+CH+LM	ESC+US8K, Last.L	Yes	0.87	0.90	0.88	0.87	0.87	50
1	LM	ESC+US8K, Last.L	No	0.87	0.95	0.88	0.88	0.87	50
1	LM+TZ	ESC+US8K, Last.L	No	0.88	0.96	0.89	0.88	0.88	50
1	LM+MFCC	ESC+US8K, Last.L	No	0.91	0.98	0.92	0.92	0.92	50
1	MFCC+TZ	ESC+US8K, Last.L	No	0.91	0.99	0.92	0.91	0.92	50
1	LM+SPC+CH	ESC+US8K, Last.L	No	0.85	0.92	0.86	0.85	0.85	50
1	MFCC+GTCC+CH+LM	ESC+US8K, Last.L	No	0.92	1.00	0.92	0.92	0.92	50

Note: Model 1 refers to the baseline CNN and Model 2 to the enhanced CNN with batch normalization and dropout (Section 3.3.1). `ESC+US8K, Last.L` denotes initial training on ESC-50, followed by fine-tuning only the last layers on the UrbanSound8K dataset. **B.N** = Batch Normalization.

3.4.1.2 Comparison Between CNN-1 and CNN-2

CNN-1 consistently outperformed CNN-2 across all feature sets. CNN-1 fine-tuned on MFCC + GTCC + CH + ML achieved 92.46% validation accuracy, while CNN-2 reached 86.71% with the same features. CNN-1 had a better balance between precision and recall, with an F1-score of 92.12%, compared to CNN-2's 86.66%. CNN-2 models showed slightly lower recall, indicating higher misclassification rates for certain classes. Batch normalization was used in CNN-2 models but not in CNN-1. Despite this, CNN-1 still outperformed CNN-2, suggesting that its architecture learns robust feature representations even without normalization. CNN-2 models benefited from batch normalization, showing better convergence and slightly lower variance in validation accuracy.

3.4.1.3 Precision, Recall, and F1-Score Analysis

The best-performing model (CNN-1 with MFCC+GTCC+CH+ML) reached 100% training accuracy, 92.34% precision, and 92.46% recall, demonstrating strong generalization to unseen samples.

CNN-1 trained on LM+MFCC achieved 91.30% validation accuracy, with 91.84% precision and 91.44% recall, confirming the importance of MFCC features for environmental sound classification. CNN-2 models had slightly lower recall, indicating some difficulty in classifying minority sound classes, but their precision remained high.

3.4.2 Comparison with AST Model

Compared to AST, the CNN models with stacked features achieved superior performance when trained solely on the ESC-50 dataset. As shown in Table 3.3, the AST model without any pretraining yielded a significantly lower validation accuracy of 0.44, underscoring its reliance on large-scale pretraining to reach its full potential [57]. In contrast, our CNN architectures achieved validation accuracies of up to 0.68, demonstrating their strong ability to capture local time-frequency patterns from limited data. However, when the AST model was initialized with weights pretrained on the large-scale AudioSet dataset and subsequently fine-tuned on ESC-50, its performance improved drastically, reaching a validation accuracy of 0.99. Meanwhile, the CNN models with pretraining achieved a maximum accuracy of 0.92. This substantial improvement highlights AST's strength in leveraging global context and rich audio representations through transfer learning. Although AST is computationally more intensive than our CNN models, its high accuracy when pretrained makes it a powerful benchmark for environmental sound classification.

Table 3.3: Validation accuracy for CNN architectures and the AST model.

Model	**Features**	**Training Setup**	**Val. Accuracy**
CNN-1	LM	ESC, All.L	0.68
CNN-2	LM + TZ	ESC, All.L	0.66
AST	Mel Spectrogram	ESC, All.L	0.44
CNN-1	MFCC + GTCC + CH + LM	ESC + US8K, Last.L	0.92
CNN-2	MFCC + GTCC + CH + LM	ESC + US8K, Last.L	0.87
AST	Mel Spectrogram	Audioset + ESC, Last.L	0.99

3.4.3 Training Time and Computational Efficiency of CNN Models

The computational efficiency of a deep learning model is a crucial factor in real-world applications, especially when deploying models on resource-constrained devices. In our experiments, we analyzed the training time for CNN-1 and CNN-2 when fine-tuned on the UrbanSound8K dataset using a last-layer retraining strategy. CNN-2 required more training time than CNN-1. The results indicate that CNN-2 had a longer training time compared to CNN-1 across all feature combinations. This discrepancy can be attributed to CNN-2's more complex architecture, which includes additional batch normalization layers that increase computational overhead. Batch normalization layers, while beneficial for stable convergence, introduce extra computations per batch, leading to longer training times. Training time also varied based on feature selection. Single-feature models (e.g., LM alone) trained faster compared to multi-feature models. Models trained

on MFCC+GTCC+CH+ML required more time, as they involved processing multiple spectral representations, increasing data dimensionality. Despite the increased training time, multi-feature models provided superior classification performance, suggesting a trade-off between accuracy and computational cost.

CNN-1 consistently converged faster than CNN-2, requiring fewer iterations to reach optimal performance. CNN-2 models trained with batch normalization showed slower convergence in the initial training phases but achieved more stable accuracy in later epochs. While CNN-2 demonstrated a higher computational cost, its use of batch normalization and deeper layers improved generalization for some feature sets. CNN-1, being computationally efficient, is better suited for applications where inference speed and resource constraints are important (e.g., real-time sound classification on embedded devices).

The best-performing model (CNN-1 fine-tuned on MFCC+GTCC+CH+ML) achieved high accuracy while maintaining a reasonable training time, making it a strong candidate for deployment. The findings suggest that CNN-1 is a more computationally efficient model, achieving strong classification performance with reduced training time. CNN-2, while more computationally demanding, benefits from batch normalization, leading to more stable training. Depending on the application, a trade-off between speed and accuracy should be considered, with CNN-1 preferred for real-time applications and CNN-2 for scenarios where stability and slightly higher recall are prioritized.

3.4.4 Training Time and Computational Efficiency of AST Model

The AST model required significantly more time to train compared to the CNN-based architectures, primarily due to its greater complexity and the inclusion of multi-head self-attention mechanisms within the Transformer encoder. This increased computational demand stems from the AST's ability to capture long-range dependencies and global contextual patterns in audio data. While this complexity enhances the model's representational power, it also introduces a trade-off between performance and efficiency—particularly relevant when computational resources are limited. Nevertheless, as illustrated in Figure 3.9, the confusion matrix of the AST model pretrained on AudioSet and fine-tuned on ESC-50 demonstrates consistently accurate classification across most sound categories. These results reaffirm the value of large-scale pretraining, which enables the AST to generalize well even on smaller datasets such as ESC-50.

3.5 Discussion

CNN models provide competitive accuracy with lower training costs, especially when enhanced with rich audio features. On the other hand, the AST model, when supported by large-scale pretraining, achieves improved results but at a higher computational expense. This presents a clear trade-off: CNNs are suitable for resource-constrained scenarios, whereas AST models are better suited for applications where accuracy is the top priority and computational resources are not a limiting factor.

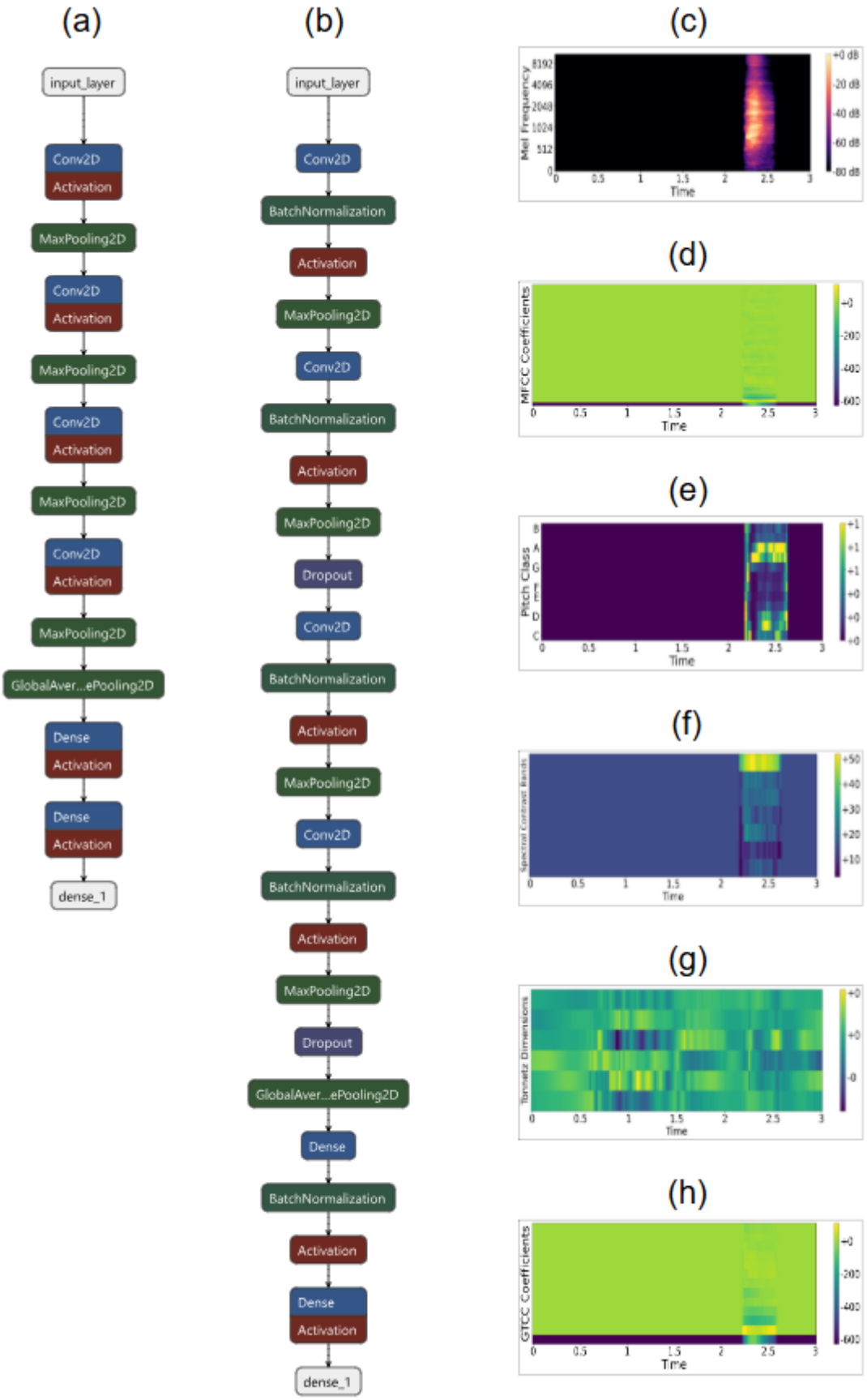


Figure 3.2: CNN Architectures and examples of feature applied to a sample ("dog" sample from ESC-50): (a) CNN-1 architecture. (b) CNN-2 architecture. (c) Log-Mel spectrogram. (d) MFCC. (e) Chroma. (f) Spectral contrast. (g) Tonnetz. (h) GTCC.

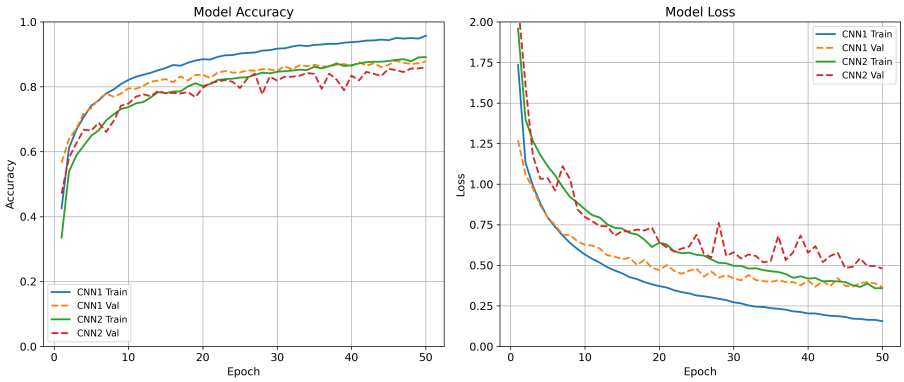


Figure 3.3: Comparison of Training and Validation Accuracy and Loss for CNN1 and CNN2 using Log-Mel (LM) Features Only.

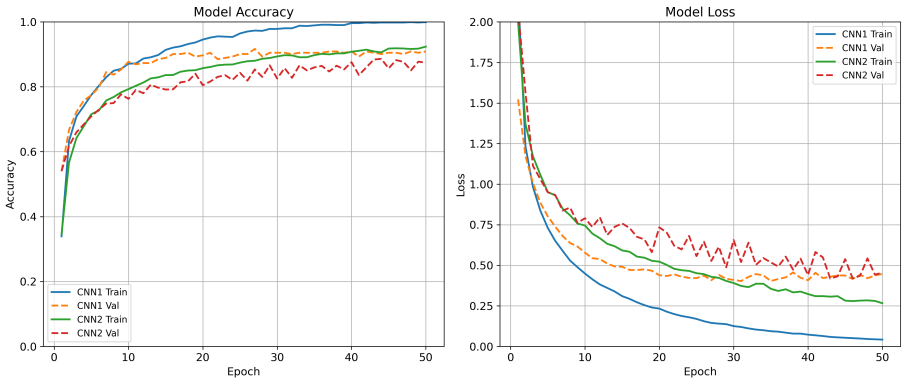


Figure 3.4: Comparison of Training and Validation Accuracy and Loss for CNN1 and CNN2 using stacked Log-Mel (LM) and MFCC features.

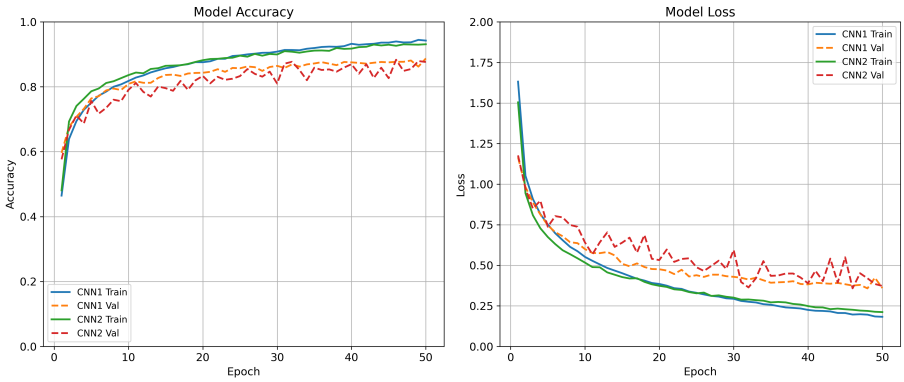


Figure 3.5: Comparison of Training and Validation Accuracy and Loss for CNN1 and CNN2 using stacked Log-Mel (LM) and Tonnetz (TZ) features.

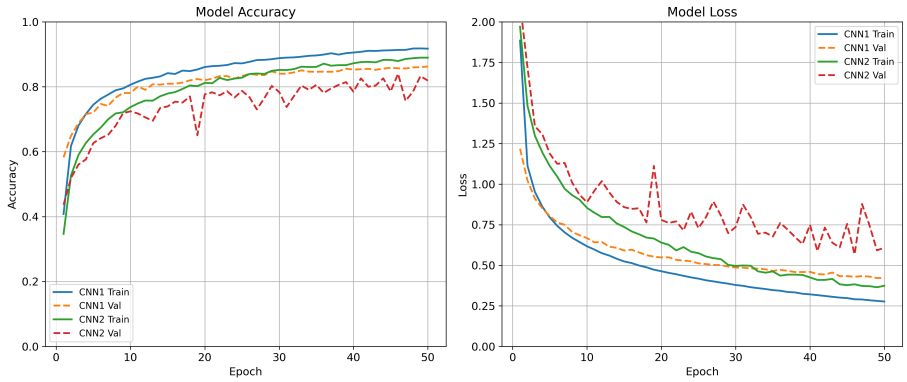


Figure 3.6: Comparison of Training and Validation Accuracy and Loss for CNN1 and CNN2 using stacked MFCC and TZ features.

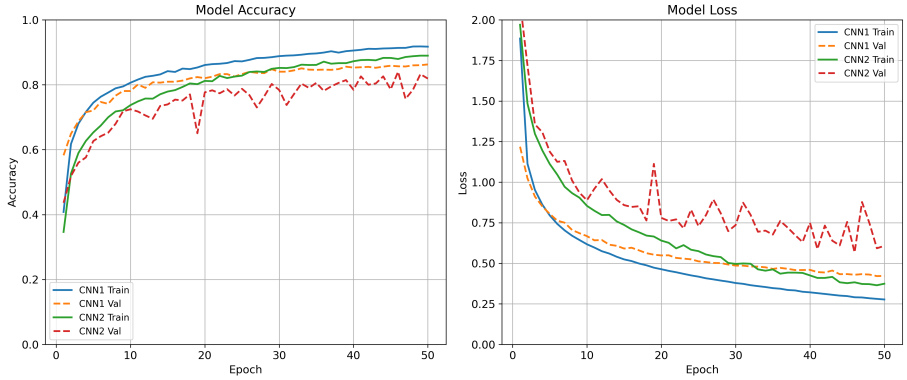


Figure 3.7: Comparison of Training and Validation Accuracy and Loss for CNN1 and CNN2 using stacked Log-Mel (LM), Spectral Contrast (SPC), and Chroma (CH) features.

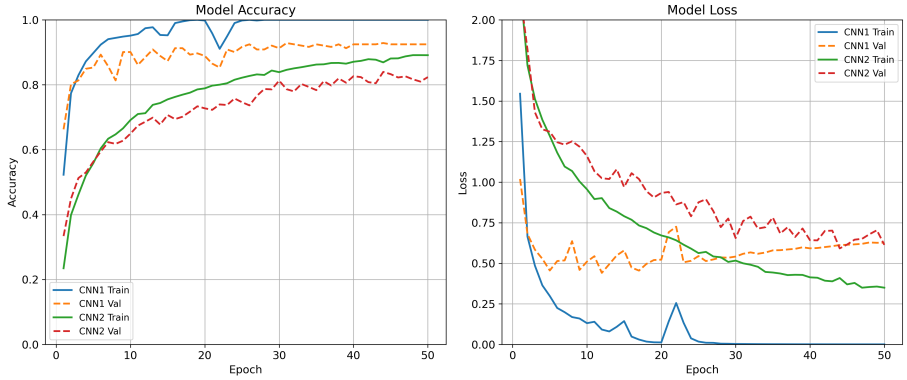


Figure 3.8: Comparison of training and validation accuracy and loss for CNN1 and CNN2 using stacked MFCC, Chroma, Mel-spectrogram, and GTCC features.

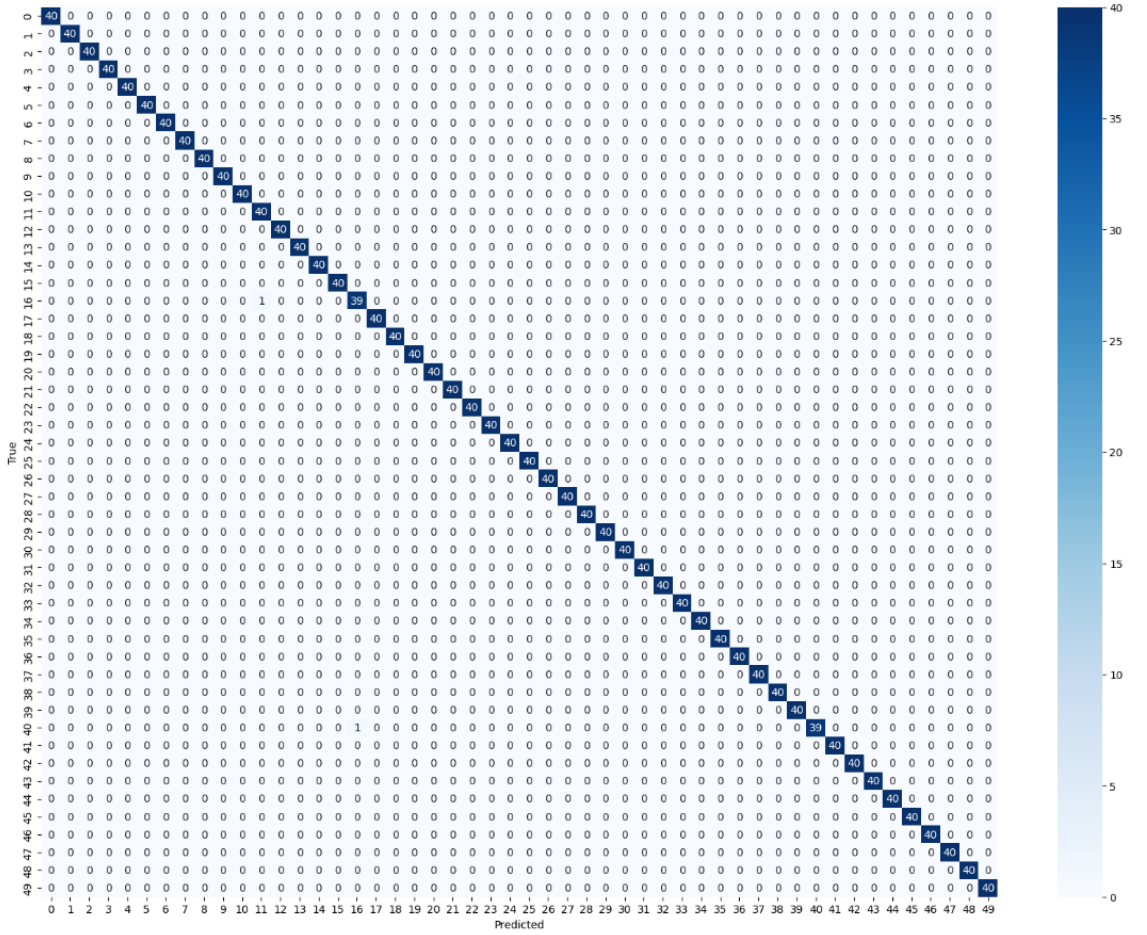


Figure 3.9: Confusion matrix for the AST model pretrained on AudioSet and fine-tuned on the ESC-50 dataset. The matrix illustrates high classification accuracy across most sound categories, indicating the model's effectiveness in leveraging large-scale pretraining.

Chapter 4

Feature Selection Methods

Feature selection is a vital stage in ESC, where high-dimensional and potentially redundant input features can hinder model performance and increase computational cost. This chapter presents a systematic exploration of both traditional and advanced feature selection and pooling strategies within ESC pipelines. Section 4.1 provides an overview of the motivation and objectives behind feature selection. Section 4.2 details the implementation of Principal Component Analysis (PCA) as a dimensionality reduction technique. Section 4.3 introduces an energy-based filter method that selects the most informative Mel frequency bands. Section 4.4 investigates the role of pooling mechanisms in implicit feature selection. Section 4.5 concludes the chapter by summarizing the comparative performance of these methods.

4.1 Introduction

Recent advances in Deep Learning, particularly CNNs, have significantly improved ESC systems' accuracy. However, these improvements typically come at the cost of increased model complexity and computational demands, limiting their applicability to resource-constrained environments such as IoT devices and embedded systems. Consequently, efficient and robust feature selection methods are vital to addressing this issue, as they can substantially reduce network complexity, accelerate training and inference processes, and enhance the CNN's capacity to learn meaningful representations.

Traditional feature selection methods, such as filter approaches leveraging cross-correlation and mutual information, as well as dimensionality reduction techniques like PCA, have been widely utilized across diverse fields within machine learning and data mining [61]. While these traditional methods historically played a pivotal role in various applications, their exploration in the context of CNNs has been relatively limited due to the inherent feature extraction and representation learning capabilities of CNN architectures. Nonetheless, recent research within audio classification domains, particularly Environmental Sound Classification (ESC), Acoustic Scene Classification (ASC), and Sound Event Detection (SED), has demonstrated that these conventional techniques can significantly enhance performance by identifying and retaining highly informative

features, thereby eliminating redundancy and irrelevance. Consequently, such methods contribute effectively towards reducing model complexity, enhancing computational efficiency, and improving overall classification capabilities [59].

4.2 Implementation of PCA

To evaluate the effectiveness of feature selection via PCA in environmental sound classification, a complete deep learning pipeline was implemented using PyTorch. The system is designed around a simple yet effective CNN architecture trained on PCA-reduced log-Mel spectrograms extracted from the ESC-50 dataset.

4.2.1 Feature Selection via PCA

Each audio file in the ESC-50 dataset was first transformed into a log-Mel spectrogram using Librosa. The spectrograms consisted of 40 Mel bands and were computed using a window size of 1024 and a hop length of 256. To facilitate dimensionality reduction, the 2D spectrograms were flattened into 1D feature vectors. These flattened features were then standardized and passed through PCA, retaining a fixed number of principal components to reduce redundancy while preserving the most informative aspects of the audio signal. PCA was employed primarily for dimensionality reduction, transforming high-dimensional spectrogram features into a lower-dimensional subspace while retaining as much variance as possible. Although unsupervised, this transformation acts as a form of feature selection by preserving the most informative components. The resulting PCA features were reshaped into a 2D format compatible with the input layers of the CNN, maintaining architectural consistency and improving computational efficiency.

4.2.2 CNN Architecture

The model architecture consisted of three convolutional blocks, each followed by Batch Normalization, ReLU activation, and Average Pooling (2×2). The final feature map was aggregated using global average pooling and passed through a fully connected layer with 128 units and a dropout of 0.5 to mitigate overfitting. The final output layer consisted of 50 units (one per ESC-50 class) with a softmax activation function.

To enhance generalization and improve robustness to overfitting, Mixup augmentation was applied during training. This technique creates synthetic training samples by linearly interpolating pairs of audio features and their corresponding labels. The model was trained using a custom loss function that interpolates between two target labels weighted by the mixup ratio. This not only improved decision boundaries but also reduced sensitivity to noise and label variance. The training was conducted using 5-fold stratified cross-validation to ensure a fair and robust evaluation across all 50 classes. Each fold involved training on 80% of the data and validating on the remaining 20%. The model was optimized using Stochastic Gradient Descent (SGD) with a learning rate of 0.1 and a momentum of 0.9.

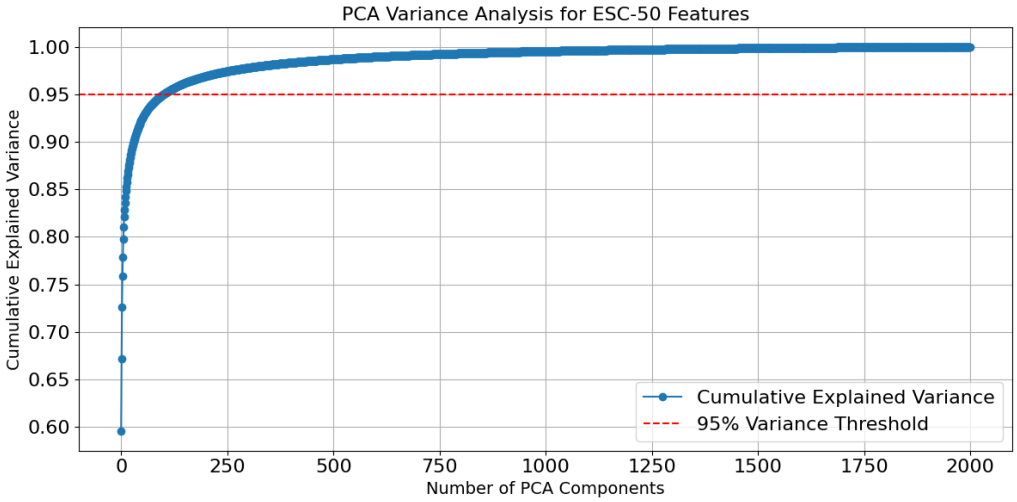


Figure 4.1: Cumulative explained variance curve for PCA applied to the ESC-50 log-Mel spectrogram features. The red dashed line indicates the 95% variance threshold, achieved with 101 components.

4.2.3 Visual Analysis of Feature Importance Using PCA

To assess the potential for dimensionality reduction in the environmental sound classification task, PCA was applied to the log-Mel spectrogram features extracted from the ESC-50 dataset. Each audio sample was converted into a 2D log-Mel spectrogram using 40 Mel frequency bands and approximately 430 time frames, resulting in a high-dimensional representation of roughly 17,200 features per instance. These spectrograms were flattened into one-dimensional vectors and standardized using normalization to ensure all features contributed equally to the PCA transformation. PCA was performed without restricting the number of components, and the cumulative explained variance ratio was analyzed to determine how much of the total variance could be retained with fewer dimensions. As shown in Figure 4.1, the curve rises sharply, indicating that most of the data's variance is captured by the first few principal components. Notably, the first 101 principal components preserved approximately 95% of the total variance, marking a substantial dimensionality reduction of over 99% from the original feature space. This analysis confirmed the effectiveness of PCA in compressing the spectrogram data while retaining most of its informative structure. The resulting reduced-dimensional representation was later reshaped and fed into the CNN, enabling more efficient training with lower memory and computational costs.

4.2.4 Dimensionality Reduction via PCA

To develop an efficient Environmental Sound Classification (ESC) system, a complete deep learning pipeline was implemented using PyTorch. The ESC-50 dataset was used, where each 5-second audio sample was converted into a log-Mel spectrogram with 40 Mel bands. While this representation effectively captures perceptually relevant features, it results in high-dimensional input data, which can increase computational cost and overfitting risk.

Each log-Mel spectrogram was computed using a sample rate of 22,050 Hz, a window size of 1024, and a hop length of 256. For a 5-second audio sample, the number of audio samples is:

$$\text{samples} = 22050 \times 5 = 110250$$

Using Librosa's default frame calculation:

$$\text{time frames} = 1 + \left\lfloor \frac{110250 - 1024}{256} \right\rfloor \approx 428$$

Thus, the resulting Mel spectrogram has a shape of 40×428, which is flattened to a vector of:

$$40 \times 428 = 17,120 \text{ features per sample.}$$

To reduce this dimensionality, Principal Component Analysis (PCA) was applied as a feature selection and dimensionality reduction technique. The flattened features were first standardized using z-score normalization. PCA was then performed to retain 95% of the total variance, which resulted in selecting only 101 principal components. This reduced the input dimension from 17,120 to 101, which represents a dimensionality reduction of approximately:

$$\frac{17120 - 101}{17120} \times 100 \approx 99.41\%.$$

This drastic reduction offers several benefits:

- Computational Efficiency: Reduces training and inference time.
- Memory Optimization: Lowers memory usage, making the model more deployable.
- Noise Reduction: Removes irrelevant or redundant components in the data.
- Better Generalization: Fewer parameters reduce the risk of overfitting.

The PCA-reduced features were reshaped to $(\text{samples}, 1, 101, 1)$ to match the input format of a lightweight Convolutional Neural Network (CNN) designed for this task. The network includes three convolutional layers with batch normalization and pooling, followed by a global average pooling layer and a fully connected output layer. Training was performed using 5-fold stratified cross-validation. To enhance model robustness, Mixup data augmentation was used during training, which creates interpolations between input samples and their corresponding labels. The effectiveness of PCA in retaining relevant information is demonstrated in Figure 4.1, which shows that the first 101 components preserve 95% of the dataset's variance.

4.3 Feature Selection via Mel Band Energy

In this implementation, log-Mel spectrograms were extracted from the ESC-50 dataset, which consists of 50 sound classes sampled at 22.05 kHz. Each spectrogram was computed using 40

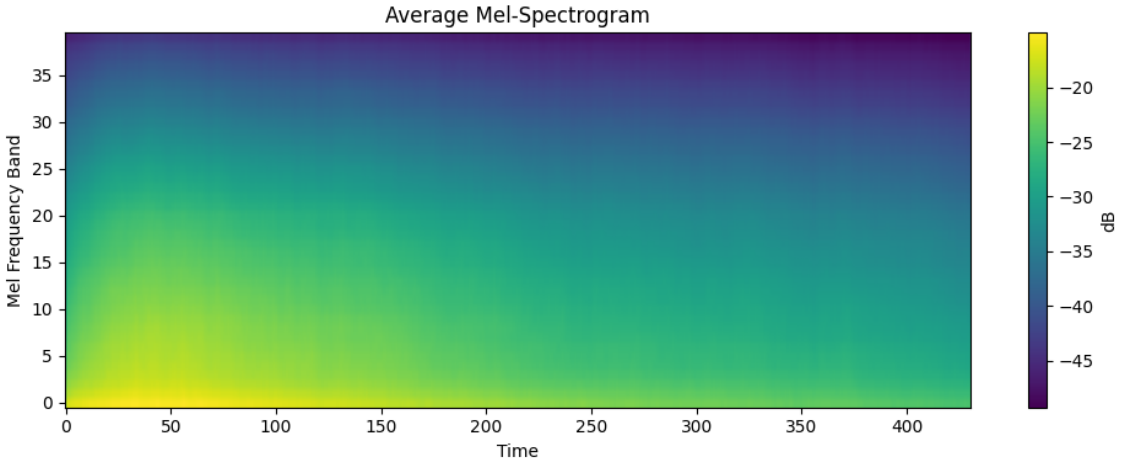


Figure 4.2: Average Mel-spectrogram computed over all samples in the ESC-50 dataset.

Mel frequency bands over a 5-second audio duration. After computing the Mel spectrograms for all samples, we calculated the average energy of each Mel band across the dataset (as shown in Figure 4.2). Based on this analysis, we selected the top 20 Mel bands with the highest average energy (as illustrated in Figure 4.3). This approach serves as a simple yet effective filter-based feature selection method, aimed at reducing dimensionality while retaining the most informative frequency components.

The rationale for selecting Mel bands with the highest average energy is grounded in the assumption that bands with consistently high energy across samples tend to capture dominant and discriminative characteristics of environmental sounds. By focusing on these prominent bands, we improve the signal-to-noise ratio and reduce the influence of less relevant or noisy frequency bands. Advantages of Selecting Top 20 Mel Bands:

- Dimensionality Reduction: Reduces the input feature size by 50%, decreasing the model's memory footprint and computational load.
- Improved Efficiency: With fewer input dimensions, the CNN trains faster and requires fewer parameters, which is particularly beneficial for resource-constrained environments.
- Noise Suppression: By eliminating low-energy bands that may carry background noise or irrelevant information, the model focuses on the most salient acoustic features.
- Better Generalization: Simplified input features reduce the risk of overfitting, especially when training with a small dataset like ESC-50.
- Interpretability: Identifying and analyzing high-energy bands gives insights into which frequency regions are most important for classifying specific sound classes.

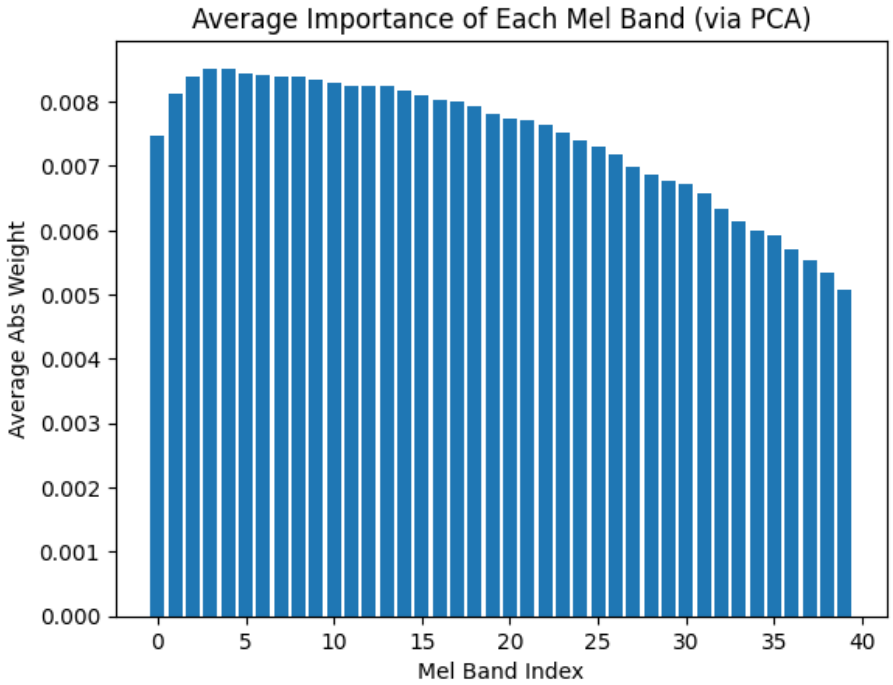


Figure 4.3: Average absolute weights of Mel-frequency bands derived from PCA Component 1.

The selected Mel band features were reshaped to match the CNN input format and fed into a lightweight CNN consisting of three convolutional layers with batch normalization, ReLU activation, pooling, and a final fully connected layer. To enhance generalization, Mixup data augmentation was employed during training. Mixup creates synthetic training examples by linearly combining pairs of inputs and labels, promoting smoother decision boundaries and robustness to noise. For evaluation, a 5-fold stratified cross-validation was used to ensure balanced class representation in each fold. Training was performed for 100 epochs per fold using a batch size of 64 and SGD optimization with momentum.This method of selecting top Mel bands based on energy combines domain knowledge with empirical statistics, yielding a compact, efficient, and interpretable feature representation for environmental sound classification.

4.3.1 Results and Analysis

To evaluate the effect of dimensionality reduction on classification performance, three CNN models were trained using different feature representations derived from log-Mel spectrograms. The baseline model, which used the full log-Mel spectrogram without any reduction, achieved the highest accuracy of 66.75%. However, this comes at the cost of increased computational complexity and potential overfitting due to high input dimensionality and possible redundancy in the feature space. When PCA was applied to reduce the input size to 101 principal components—capturing 95% of the original variance and resulting in a 99.41% dimensionality reduction—the classification accuracy dropped significantly to 37.60%. This reduction suggests that while PCA effectively

compresses data by eliminating redundancy, it may also discard subtle but crucial spectral patterns that are important for CNN-based learning in sound classification tasks. This highlights the trade-off: while aggressive dimensionality reduction can improve computational efficiency, it may degrade classification performance if it eliminates discriminative features.

In contrast, selecting the top 20 Mel frequency bands based on average energy offered a better compromise. This filter-based method reduced dimensionality by 50% while still preserving the most informative frequency components. It achieved an accuracy of 62.75%, significantly outperforming the PCA-based model and closely approaching the performance of the baseline. This approach helps to reduce irrelevant or low-energy features that might otherwise introduce noise into the learning process. By focusing the CNN on the most salient spectral regions, the model is not only more efficient to train but also more robust, as it operates over a more relevant and compact feature space. Table 4.1 summarizes the performance of the three models, highlighting the trade-offs between different dimensionality reduction strategies. It demonstrates that while PCA achieves the highest reduction in input size, it compromises model performance. In contrast, energy-based band selection maintains a better balance between efficiency and accuracy. Overall, these results demonstrate the importance of carefully balancing dimensionality reduction and classification performance. While PCA emphasizes variance preservation, energy-based Mel band selection aligns better with the spectral characteristics of environmental sounds, resulting in more meaningful and efficient feature representations for CNN training.

Table 4.1: Comparison of Dimensionality Reduction Methods on CNN Performance

Model	**Input Type**	**Reduction Method**	**Dimensionality Reduction (%)**	**Accuracy (%)**
CNN	Log-Mel Spectrogram	None	0.00	66.75
CNN	Log-Mel Spectrogram	PCA (101 components)	99.41	37.60
CNN	Log-Mel Spectrogram	Top 20 Mel Bands (by energy)	50.00	62.75

4.4 Feature Selection and Pooling Mechanisms

Feature selection and pooling are two critical mechanisms in CNN-based models that jointly influence the model's ability to learn discriminative representations from audio data. In the context of ESC, feature selection techniques help in identifying the most informative acoustic attributes, thereby reducing redundancy and improving model generalization. Simultaneously, pooling operations play a central role in summarizing local features across spatial or time-frequency domains, enabling the network to focus on high-level patterns while controlling model complexity. While traditional pooling techniques such as max and average pooling are widely used, their uniform treatment of all feature regions can limit performance in tasks requiring fine-grained temporal or

spectral attention. To address this, advanced pooling methods like Sparse Salient Region Pooling (SSRP) have been proposed, offering enhanced discriminative focus by selectively aggregating only the most salient regions of the feature map. The combination of feature selection and advanced pooling mechanisms is, therefore, crucial for developing efficient and accurate CNN models for ESC, especially under noisy and resource-constrained conditions [70, 7].

4.4.1 Traditional Pooling Methods

Traditional pooling methods, such as Max Pooling and Average Pooling, are fundamental techniques in CNNs for dimensionality reduction and feature aggregation. Max Pooling operates by selecting the maximum activation value from each local region of the feature map, effectively capturing the most prominent features while discarding weaker activations. This strategy enhances feature robustness and reduces spatial variance, making it well-suited for capturing distinct patterns. Conversely, Average Pooling computes the mean value of each local region, preserving more generalized information across spatial dimensions. While both methods contribute to reducing computational complexity and mitigating overfitting, they often fail to emphasize the most discriminative regions, particularly in complex auditory scenes like environmental sound classification. In these scenarios, the ability to selectively prioritize salient audio patterns is crucial, motivating the exploration of advanced pooling strategies such as SSRP.

4.4.2 Advanced Pooling Methods

Pooling methods like SSRP, initially developed for image processing tasks in computer vision, have been adapted to enhance CNN-based environmental sound classification systems by focusing on sparse and salient regions within time-frequency audio representations [70]. SSRP applied to ESC focuses on selectively capturing sparse, salient regions within the time-frequency audio representations, which contain the most informative and discriminative characteristics relevant for accurate sound classification.

Unlike traditional pooling methods, such as Global Average Pooling (GAP), SSRP does not treat all regions equally but rather prioritizes sparse, high-relevance regions [7]. This selective approach significantly enhances the CNN's ability to learn meaningful patterns from the data, improving classification performance while simultaneously reducing computational requirements. SSRP thus addresses key limitations associated with conventional pooling strategies, making it particularly suitable for ESC applications in resource-constrained environments.

4.4.2.1 Pooling Methods based on SSRP

This section outlines the methodologies used for evaluating environmental sound classification using sparse salient region pooling techniques. We focus on two pooling strategies, which are discussed in detail below.

Sparse Salient Region Pooling - Basic (SSRP-B) The *Basic variant of Sparse Salient Region Pooling (SSRP-B)* is the simplest form of the SSRP mechanism, designed to enhance feature aggregation in time-frequency representations. In SSRP-B, a fixed-size temporal window W is used to pool activation values over time. Unlike traditional pooling methods that uniformly aggregate all regions, SSRP-B focuses on salient temporal regions—specifically, those with the highest average activation—thereby filtering out less informative or noisy segments. Within each temporal window, the region exhibiting the highest mean activation is identified and selected. The pooled representation is then obtained by averaging the activation values within this salient segment. This selective pooling enhances the model's ability to focus on discriminative patterns, improving robustness and classification performance.

The pooled output for each channel c and frequency bin f is computed as:

$$z_c(f) = \frac{1}{W} \sum_{i=1}^{W} m_c^D(t+i, f)$$

where:

- $z_c(f)$ is the resulting pooled feature for channel c at frequency f,
- $m_c^D(t+i, f)$ denotes the activation value at time step $t+i$ and frequency f in channel c,
- W is the temporal window size.

Here, the activation value refers to the output of a neural unit after convolution and non-linear transformation (e.g., ReLU), representing the intensity or importance of detected features. By emphasizing only the most informative segments within each window, SSRP-B effectively reduces noise and enhances the quality of the learned representations.

Top-K Salient Region Pooling (SSRP-T) SSRP-T extends the basic pooling mechanism by allowing the model to consider multiple high-activation regions instead of just the single most active window. Here, the top K activations (sorted by their mean values) are averaged to form the final pooled representation, capturing more salient regions distributed over time. The pooled representation is computed as:

$$z_c(f) = \frac{1}{K} \sum_{k=1}^{K} s_c^{[k]}(f) \tag{4.1}$$

where K is the number of top activations selected, $s_c^{[k]}(f)$ denotes the k^{th} highest mean activation for channel c and frequency f.

4.4.3 Implementation of SSRP

In order to evaluate the effectiveness of SSRP mechanisms in CNN architectures for environmental sound classification, two distinct variants—SSRP-B (block-based) and SSRP-T (top-K temporal)—were implemented and tested on the ESC-50 dataset using TensorFlow and Keras.

4.4.3.1 SSRP-T (Top-K Temporal Pooling)

The SSRP-T method focuses on retaining the most salient temporal activations within each frequency band and channel. The implementation proceeds as follows:

- After the final convolutional layer, the 4D feature map of shape (B,T,F,C)—representing batch size, time, frequency, and channels—is reshaped and transposed so that the temporal axis is last.
- For each feature channel and frequency bin, the top-K highest activation values are selected across the time dimension.
- The mean of these top-K values is computed and concatenated into a fixed-length feature vector for classification.

This technique ensures that the model emphasizes temporally sparse yet highly informative regions, which are typical of short, high-energy sound events in environmental recordings.

4.4.3.2 SSRP-B (Fixed Window Pooling)

The SSRP-B mechanism utilizes a fixed-size sliding window (e.g., w=4 or w=6) across the temporal axis:

- Patches of temporal segments are extracted.
- The average activation is computed for each window, and the window with the highest mean is selected.
- This results in a compact representation by preserving only the most salient block of activations for each frequency-channel pair.

Unlike global pooling methods such as GAP (Global Average Pooling), SSRP-B prioritizes localized energy bursts—thereby aligning better with the transient and intermittent nature of many environmental sound events. The effectiveness of this strategy is illustrated in Figure 4.4, which shows the confusion matrix for SSRP-B with w=4.

4.4.3.3 Shared CNN Architecture and Training Setup

Both SSRP variants were integrated into a lightweight CNN consisting of:

- Three convolutional layers with filter sizes 3×3 and increasing channel depths (32, 64, 128), each followed by Batch Normalization, ReLU activation, and Average Pooling (applied in the first two layers).
- A dropout layer (p=0.5) to mitigate overfitting.
- A final dense layer with 128 units followed by a softmax output for 50-class classification.

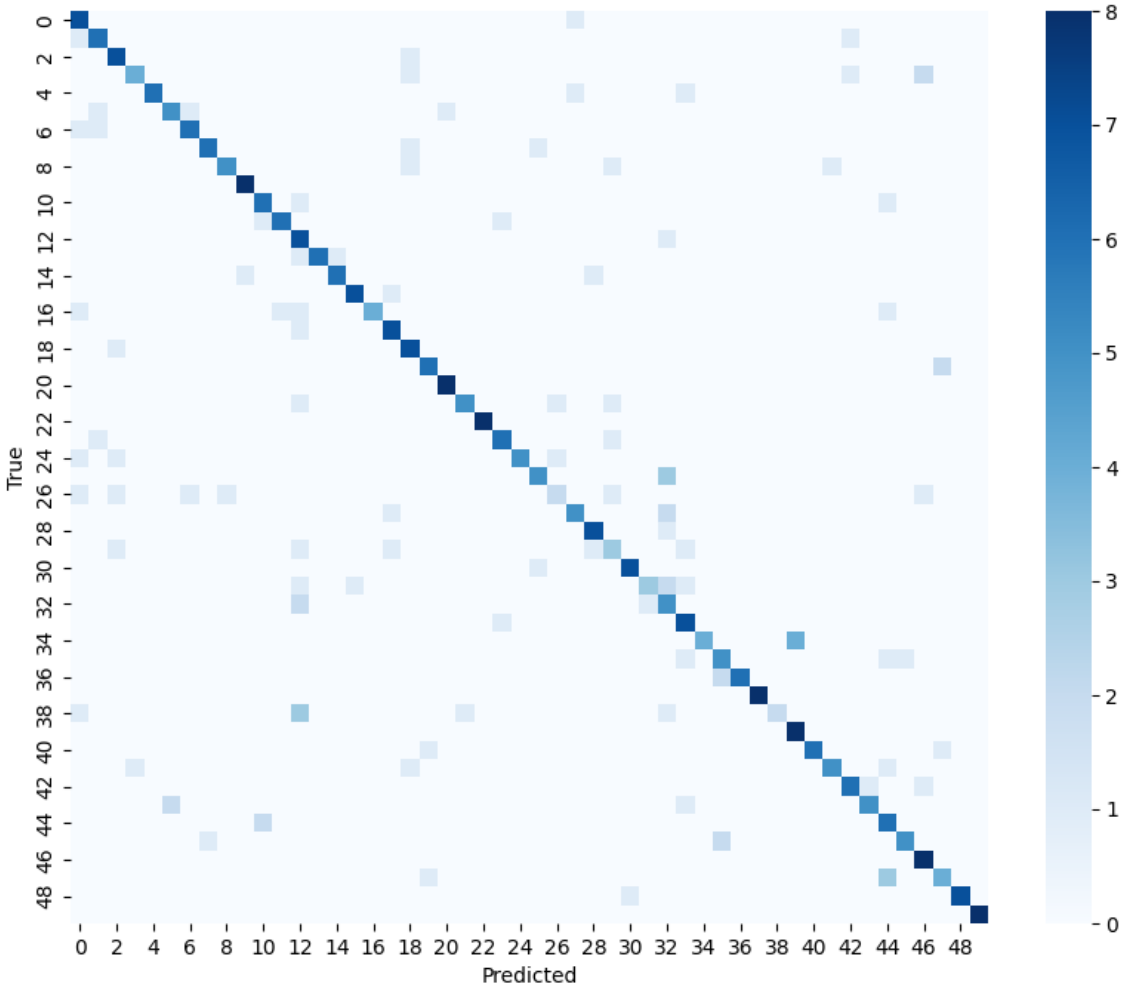


Figure 4.4: Confusion matrix for SSRP-B with $W = 4$.

The log-mel spectrograms (with shape 431×40) were extracted using librosa and normalized before feeding them into the models. For training, 5-fold Stratified Cross-Validation was employed, combined with Mixup data augmentation in the early training epochs to further enhance generalization. The models were trained using the SGD optimizer with momentum, and learning rate scheduling and early stopping callbacks were used to prevent overfitting.

4.4.3.4 Evaluation

For each fold, performance was assessed using accuracy and confusion matrices. Validation accuracy per epoch was logged and visualized, and final results were averaged across folds. The results showed that SSRP-based pooling strategies improved the model's ability to focus on meaningful regions in the spectrograms, thereby increasing classification robustness and efficiency.

4.4.4 Results and Analysis

Table 4.2 presents the classification accuracy of CNN models integrated with two different SSRP pooling variants—SSRP-B (block-based) and SSRP-T (top-K temporal)—using both Log-Mel and MFCC inputs. For the SSRP-B variant with Log-Mel features, the model achieved its highest accuracy of 72.85% when the window size w=4, outperforming other tested window configurations (w=2,6,8). Notably, increasing the window size beyond 4 resulted in performance degradation, suggesting that excessively large pooling windows may dilute salient temporal structures crucial for environmental sound classification. This trend is further illustrated in Figure 4.5, which shows the validation accuracy curves across 5 folds for SSRP-B with w=4 and w=6. The figure confirms that w=4 consistently yields more stable and higher accuracy across folds compared to w=6, emphasizing its suitability for preserving local temporal cues. In contrast, SSRP-B using MFCC input showed lower overall accuracy, with a maximum of 64.60% at w=4, reinforcing that Log-Mel representations provide richer time-frequency information than MFCCs for the SSRP mechanism.

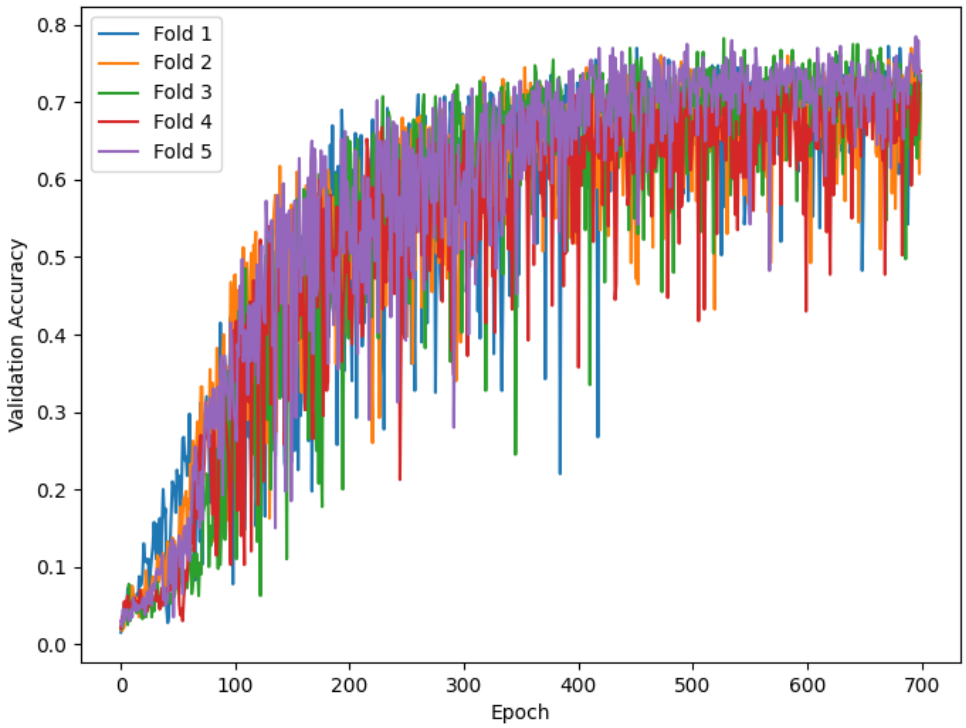
The SSRP-T variant, which selects the top-K most salient temporal activations, consistently outperformed SSRP-B across all tested K values. The model achieved peak accuracy of 80.69% at k=12, demonstrating the effectiveness of prioritizing sparse high-energy activations. Furthermore, results indicate a stable high-performance range between k=10 and k=14, with only a slight decline at k=16. This highlights the advantage of SSRP-T in capturing key temporal cues across frequency bins, which is critical for ESC tasks characterized by transient and diverse sound patterns. Overall, the SSRP-T model with Log-Mel input and k=12 demonstrated the best performance, affirming the value of sparse temporal pooling and feature selection in enhancing CNN-based environmental sound classification.

Table 4.2: Accuracy comparison of CNN with different SSRP variants using Log Mel and MFCC inputs.

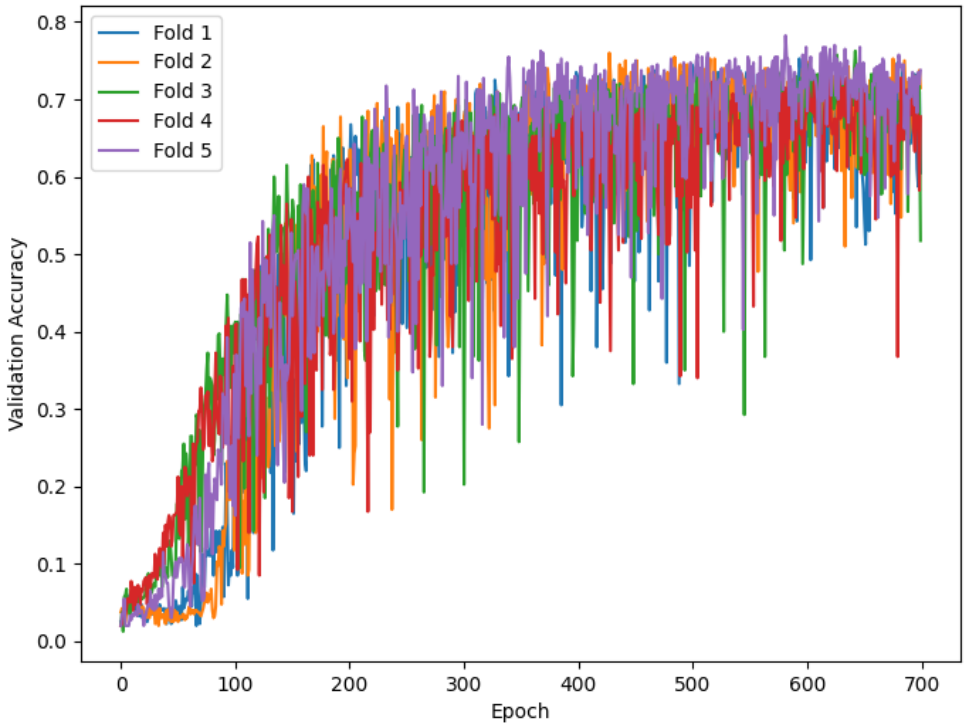
Model	**Input**	**Hyperparameter**	**Accuracy (%)**
CNN + SSRP-B	Log Mel	$w = 2$	71.15
CNN + SSRP-B	Log Mel	$w = 4$	72.85
CNN + SSRP-B	Log Mel	$w = 6$	65.05
CNN + SSRP-B	Log Mel	$w = 8$	66.09
CNN + SSRP-B	MFCC	$w = 4$	64.60
CNN + SSRP-B	MFCC	$w = 6$	62.95
CNN + SSRP-T	Log Mel	$k = 4$	75.20
CNN + SSRP-T	Log Mel	$k = 8$	77.60
CNN + SSRP-T	Log Mel	$k = 10$	80.60
CNN + SSRP-T	Log Mel	$k = 12$	80.69
CNN + SSRP-T	Log Mel	$k = 14$	78.65
CNN + SSRP-T	Log Mel	$k = 16$	70.59

4.5 Discussion

These findings highlight several important insights. First, PCA—despite being a well-established feature selection technique—may not preserve localized temporal-frequency saliency essential for environmental sound discrimination. In contrast, SSRP pooling explicitly emphasizes such salient patterns by prioritizing high-activation regions, resulting in richer feature representations and stronger model generalization. Second, SSRP's ability to retain task-relevant spatial cues without incurring significant computational overhead makes it well-suited for resource-constrained applications. This makes SSRP-based models a compelling alternative to traditional dimensionality reduction techniques in audio classification tasks. In conclusion, while PCA provides a basic mechanism for reducing dimensionality, the superior accuracy and robustness of SSRP-based CNNs, particularly SSRP-T, demonstrate the value of sparsity-aware pooling strategies for enhancing model performance in environmental sound classification.



(a) Validation accuracy (SSRP-B, $W = 4$)



(b) Validation accuracy (SSRP-B, $W = 6$)

Figure 4.5: Comparison of validation accuracy across 5 folds for different window sizes in SSRP-B pooling. (a) Validation accuracy for SSRP-B with $W = 4$. (b) Validation accuracy for SSRP-B with $W = 6$.

Chapter 5

Spectrogram Transformers for Audio Classification

This chapter introduces a focused investigation into the use of Spectrogram Transformer architectures for audio classification. Section 5.1 outlines the motivations and objectives behind adopting spectrogram-based transformer models. Section 5.2 delves into the architectural details and design principles of the proposed models, describing how each one processes time and frequency components to enhance classification performance. In Section 5.3, the experimental setup is described, including preprocessing and feature extraction techniques, model configuration, and training procedures. Finally, Section 5.4 provides a discussion of the results, highlighting key observations and comparative insights across model variants.

5.1 Motivations and Objectives

Transformer architectures have achieved notable success in various domains, including natural language processing and computer vision. More recently, their application to audio classification has gained momentum—particularly with models such as the Audio Spectrogram Transformer (AST), which adopts Vision Transformer (ViT) mechanisms to classify audio spectrograms. However, despite their strong performance, these models often treat spectrograms as images, applying spatially uniform patching and self-attention, which may overlook the inherently structured and asymmetric nature of time-frequency representations in audio.

Environmental sounds typically exhibit distinct temporal dynamics and frequency-dependent characteristics. For instance, events such as dog barks and sirens are temporally localized, while sounds like rain or engine noise span broader frequency bands. Existing transformer models like AST do not explicitly account for this temporal-spectral decomposition, potentially limiting their ability to capture fine-grained acoustic patterns efficiently. This chapter is motivated by the need to address these limitations by exploring transformer designs that explicitly attend to the temporal and spectral dimensions of spectrograms. Specifically, we present and evaluate three architectures:

Temporal Only (TO), Temporal-Frequency Parallel (TFP), and Temporal-Frequency Sequential (TFS).

These models aim to better exploit the structured nature of spectrograms by introducing customized attention mechanisms that decouple or recombine temporal and frequency contexts. The objective of this chapter is to analyze how attention mechanisms that process the time and frequency axes, either independently or jointly, affect model performance on Environmental Sound Classification (ESC) tasks.

5.2 Architectural Details and Design Principles

The Spectrogram Transformer architectures explored in this study are designed to explicitly model the structural characteristics of audio spectrograms by attending to their temporal and frequency dimensions in a more informed and modular way. Rather than treating the spectrogram as a uniform visual grid—as is common in vision transformer (ViT) based approaches like AST—these models introduce time-frequency decomposed attention mechanisms. This section outlines the architectural foundations and design rationales behind the three proposed variants: Temporal Only (TO), Temporal-Frequency Parallel (TFP), and Temporal-Frequency Sequential (TFS).

5.2.1 Temporal Only (TO)

The TO model simplifies the attention mechanism by applying self-attention exclusively along the time axis. Each frequency bin is treated as an independent stream, and attention is computed across time frames. This design assumes that temporal progression carries dominant discriminative information, while frequency relationships are either stable or less critical. It benefits from reduced complexity and faster computation, making it a lightweight option for time-dominant audio tasks.

5.2.2 Temporal-Frequency Parallel (TFP)

The TFP model decomposes attention into two separate branches: one dedicated to temporal attention and the other to frequency attention. These branches operate in parallel and independently capture dependencies along their respective axes. The outputs from both branches are later fused (e.g., via summation or concatenation) before being passed to the feedforward layers. This structure allows the model to simultaneously learn time and frequency patterns while preserving axis-specific attention behaviors.

5.2.3 Temporal-Frequency Sequential (TFS)

In contrast to TFP, the TFS model applies attention operations in a sequential manner. It first computes self-attention along one axis (typically time), and then processes the resulting representation with attention along the other axis (frequency). This order-aware processing captures conditional dependencies—i.e., how frequency patterns evolve over time or vice versa. While it introduces some directional bias, it provides a structured and staged refinement of the input representation.

5.3 Experimental Setup

To evaluate the performance of the proposed spectrogram transformer architectures, a series of experiments were conducted using the widely adopted ESC-50 dataset. The dataset is pre-partitioned into five folds to support standardized cross-validation, and the experiments in this study follow this protocol to ensure a fair comparison and reproducibility.

5.3.1 Preprocessing and Feature Extraction

Each audio clip is first loaded using torchaudio, resampled to a consistent sample rate of 44.1 kHz if needed, and transformed into a log-mel spectrogram. This is achieved using a MelSpectrogram transform with 128 mel bins, followed by a decibel scaling operation via AmplitudeToDB. To ensure uniform input shapes, the resulting spectrograms are standardized to a size of 100 time steps × 128 frequency bins. If the number of time frames is less than 100, zero-padding is applied; if it exceeds 100, the spectrogram is truncated. The spectrograms are then normalized on a per-feature basis to improve training stability and consistency across the dataset.

5.3.2 Model Configuration

5.3.2.1 Temporal Only (TO)

The implemented model corresponds to the Temporal-Only (TO) spectrogram transformer, specifically designed to model sequential relationships across the time axis of an audio spectrogram while ignoring dependencies across the frequency axis. This design aligns with the hypothesis that in many environmental sound classification (ESC) tasks, the temporal evolution of sound patterns often carries more discriminative information than frequency-local correlations.

The TO architecture processes a log-mel spectrogram input of shape 100×128, where 100 corresponds to the number of time frames and 128 to the number of mel-frequency bins. The model begins with a TemporalEmbedding layer that projects each 128-dimensional frequency vector (i.e., each time step) into a 256-dimensional embedding space using a linear transformation. A learnable [CLS] token is prepended to the sequence to serve as a global aggregator of the temporal context. Additionally, positional encodings are learned and added to each time-step embedding, allowing the model to retain information about the order of the sequence. The core of the architecture consists of a Transformer Encoder built from 4 stacked layers, each containing: mmulti-head self-attention with 4 heads, a feedforward sub-network with ReLU activation, and layer normalization and dropout for regularization.

Importantly, all self-attention is applied only along the temporal dimension, treating each time frame independently of the others in the frequency axis. After encoding, the final representation of the [CLS] token is passed through a fully connected classifier layer, which maps the 256-dimensional vector to 50 output logits—corresponding to the 50 classes in the ESC-50 dataset. This architecture is lightweight and computationally efficient while retaining the ability to capture global temporal dependencies critical for ESC tasks.

5.3.2.2 Temporal-Frequency Parallel (TFP)

The Temporal-Frequency Parallel (TFP) transformer architecture is designed to process the time and frequency dimensions of spectrograms in separate but concurrent branches. This architecture leverages the hypothesis that temporal and spectral patterns in environmental sounds contain distinct and complementary information, and that learning them in parallel can enhance model performance. The model begins by linearly projecting the input log-mel spectrogram of shape (batch,100,128), where 100 represents time steps and 128 represents mel-frequency bins. Two separate embedding paths are constructed:

- Temporal Branch: Each time step is treated as a token consisting of 128 frequency bins. These vectors are projected to a 256-dimensional embedding space using a linear layer. A learnable classification token ([CLS]) is prepended to the sequence, and learnable temporal positional embeddings are added to preserve the temporal order. The sequence is then passed through a Transformer encoder composed of 4 layers, each with 4 attention heads and a hidden dimension of 256.

- Frequency Branch: In parallel, the spectrogram is transposed to treat each frequency bin as a token, each consisting of 100 time steps. This sequence is likewise projected into a 256-dimensional space and prepended with a [CLS] token. Frequency positional embeddings are added to retain spatial consistency, and the sequence is passed through another identical Transformer encoder.

After attention processing, the output [CLS] tokens from both branches are extracted and concatenated to form a combined representation of size 512. This representation is then passed through a final linear classifier, which outputs a prediction over the 50 classes defined in the ESC-50 dataset.

5.3.2.3 Temporal-Frequency Sequential (TFS)

Temporal-Frequency Sequential (TFS) transformer architecture aims to capture both temporal dynamics and frequency-dependent patterns in environmental sounds through a two-stage attention mechanism. Unlike models that apply attention jointly or in parallel across time and frequency axes, TFS processes them sequentially—first applying self-attention along the temporal axis, followed by attention along the frequency axis. The model consists of two specialized embedding modules:

- Temporal Embedding Layer: Each 128-bin frequency vector at a given time step is linearly projected into a 256-dimensional embedding space. A learnable [CLS] token is prepended to the sequence, and positional embeddings are added to preserve temporal order.

- Frequency Embedding Layer: After temporal processing, the output is transposed, treating each frequency bin's time evolution as a sequence. The same embedding strategy is applied to encode frequency-specific temporal patterns.

Each embedding stream is followed by a dedicated Transformer Encoder:

- The temporal encoder consists of 4 transformer blocks, each with 4 attention heads and a model dimension of 256.
- The frequency encoder also includes 4 transformer blocks with the same configuration.

After processing, the [CLS] token output from both encoders is extracted and averaged to form a unified audio representation. This is then passed to a final classification head, which is a linear layer mapping the 256-dimensional embedding to 50 class logits, corresponding to the 50 categories in the ESC-50 dataset. This dual-stage transformer configuration enables the model to first extract time-dependent features, then refine them with frequency-aware attention, thereby modeling spectrograms in a sequential and semantically structured way.

5.3.3 Training Details

The model is trained from scratch using the Adam optimizer with a learning rate of 1e-4. Cross-entropy loss is used as the training criterion. Training is conducted for 20 epochs, with each batch containing 64 spectrograms. To improve generalization and avoid overfitting, SpecAugment-style augmentations (including time masking and frequency masking) are optionally applied. Additionally, label smoothing and dropout are utilized within the Transformer layers to regularize training. Model checkpoints and best validation performance metrics are saved throughout training. A training history is logged, capturing training and validation accuracy and loss per epoch.

Figures 5.1, 5.2, and 5.3 represent the training and validation performance of the Temporal-Only (TO), Temporal-Frequency Parallel (TFP), and Temporal-Frequency Sequential (TFS) transformer models, respectively, on the ESC-50 dataset.

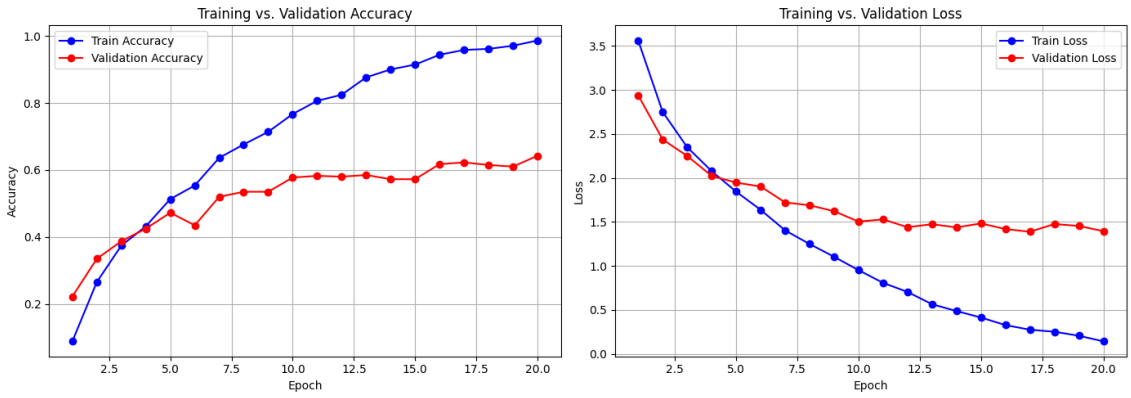


Figure 5.1: Training and validation performance of the Temporal-Only (TO) transformer model on the ESC-50 dataset. The left plot shows the progression of training and validation accuracy across 20 epochs, while the right plot illustrates the corresponding loss curves.

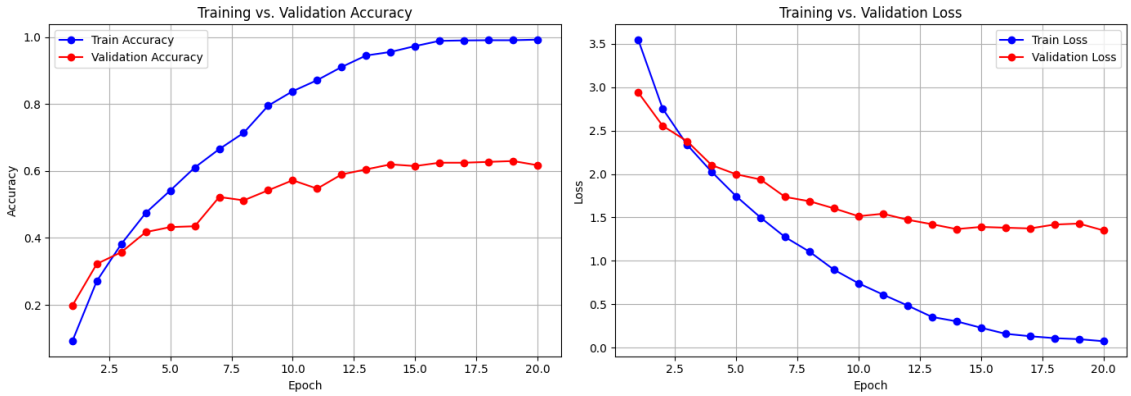


Figure 5.2: Training and validation performance of the Temporal-Frequency Parallel (TFP) transformer model on the ESC-50 dataset. The left plot shows the progression of training and validation accuracy across 20 epochs, while the right plot illustrates the corresponding loss curves.

5.4 Discussion

The experimental results highlight important insights into the role of temporal and spectral modeling in transformer-based architectures for environmental sound classification (ESC). As presented in Table 5.1, the Temporal Only (TO) variant achieved the highest validation accuracy of 64.25%, outperforming both the Temporal-Frequency Sequential (TFS) and Temporal-Frequency Parallel (TFP) variants, which achieved 61% and 61.75%, respectively. These findings suggest that temporal attention alone can capture a substantial portion of the discriminative patterns required for ESC tasks. This could be attributed to the inherently sequential nature of environmental sounds, where temporal cues often play a more dominant role than spectral relationships. In contrast, models that attempt to simultaneously or sequentially attend to both temporal and frequency dimensions (TFP and TFS) showed slightly lower performance, possibly due to increased model complexity or insufficient optimization under limited training data. Figures 5.4, 5.5, and 5.6 represent the confusion matrices for the Temporal-Only (TO), Temporal-Frequency Parallel (TFP), and Temporal-Frequency Sequential (TFS) transformer models, respectively, on the ESC-50 dataset.

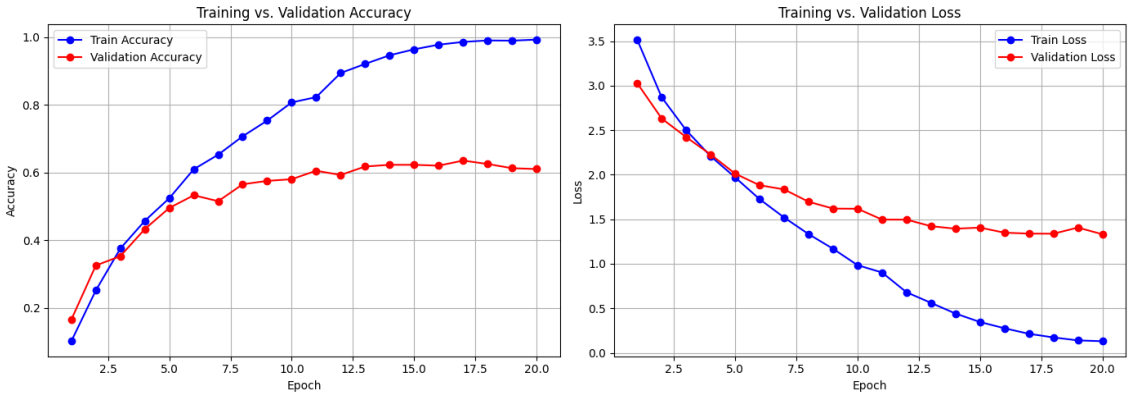


Figure 5.3: Training and validation performance of the Temporal-Frequency Parallel (TFS) transformer model on the ESC-50 dataset. The left plot shows the progression of training and validation accuracy across 20 epochs, while the right plot illustrates the corresponding loss curves.

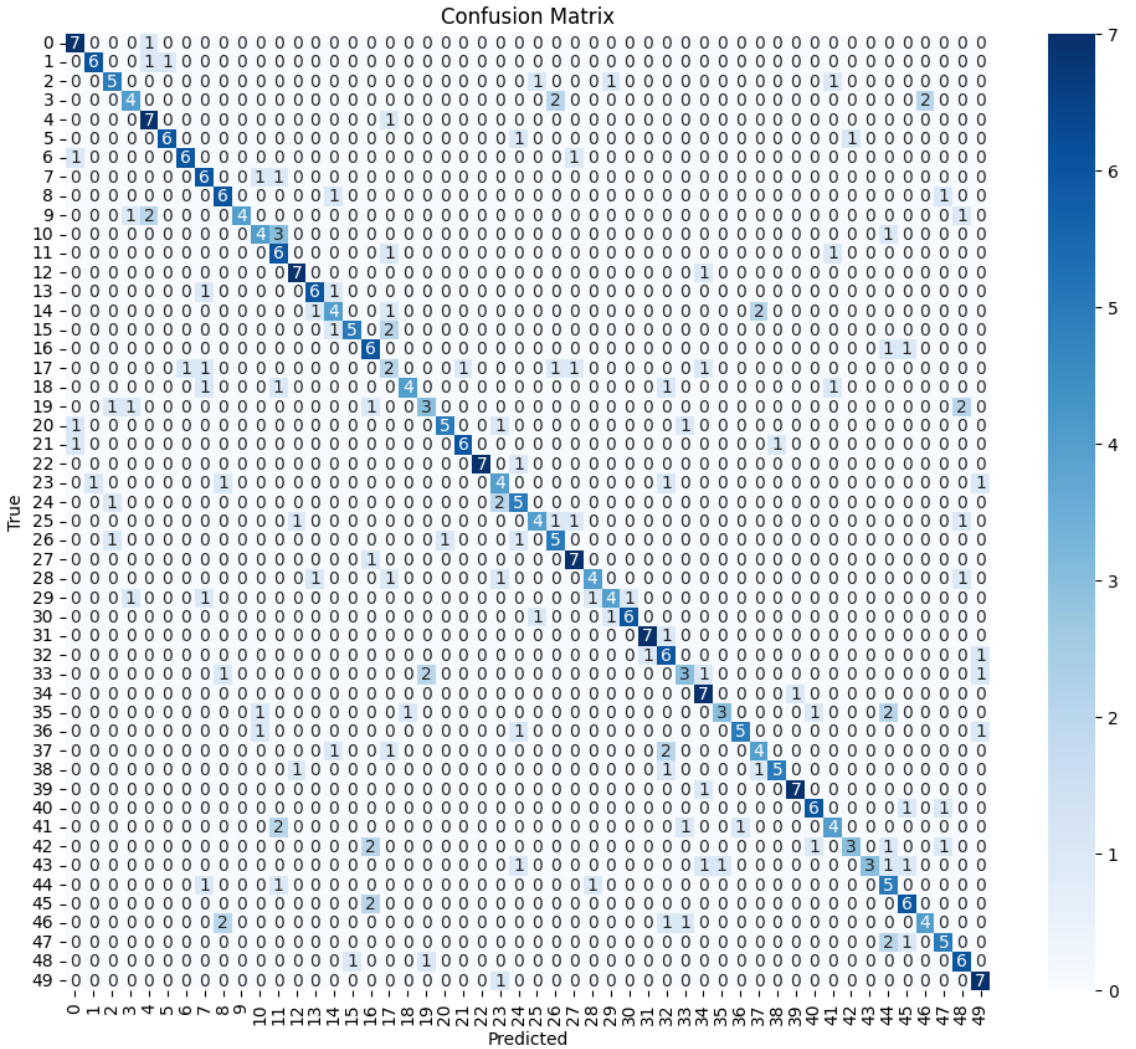


Figure 5.4: Confusion matrix for the Temporal-Only (TO) transformer model on the ESC-50 dataset.

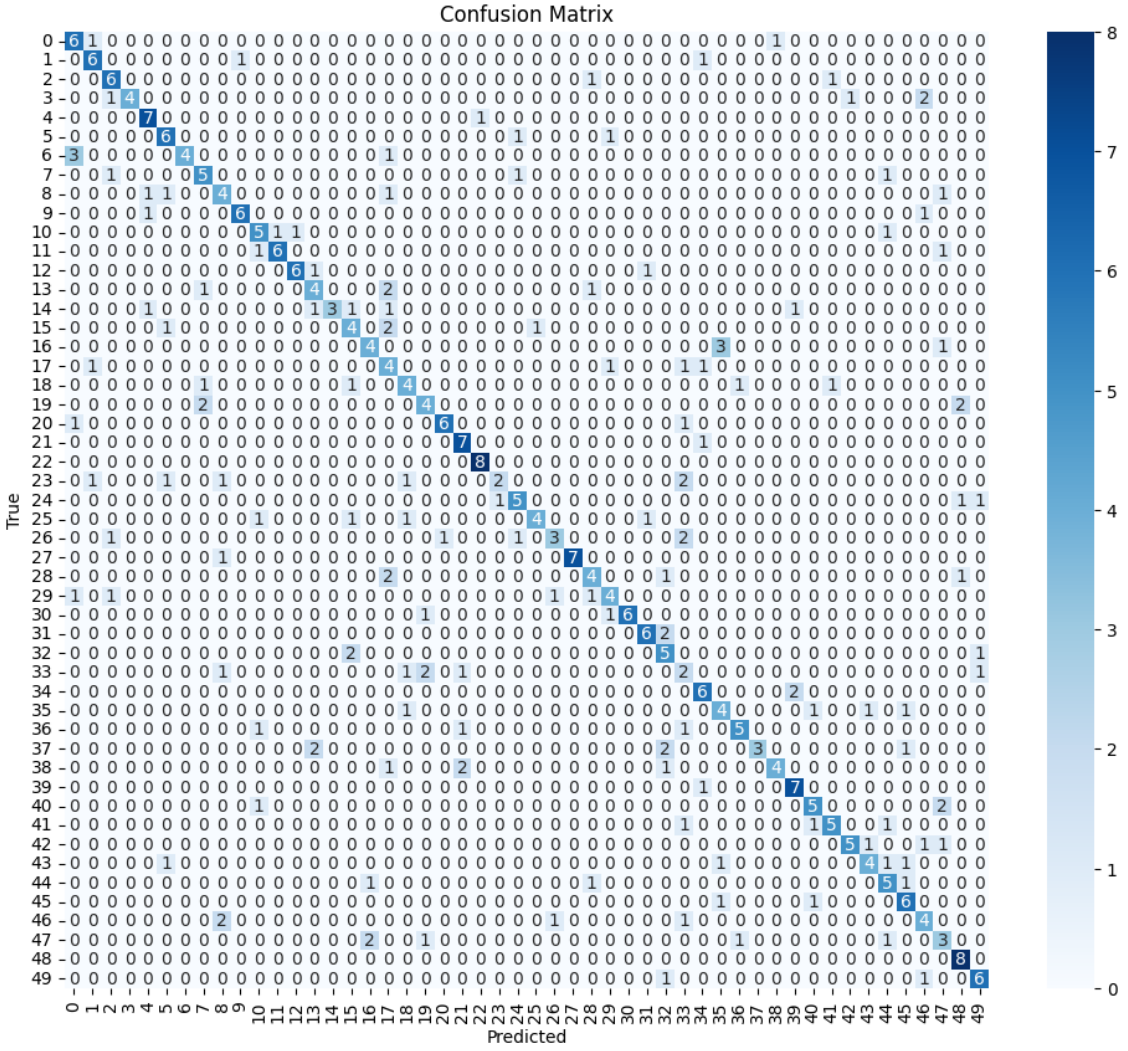


Figure 5.5: Confusion matrix for the Temporal-Frequency Parallel (TFP) transformer model on the ESC-50 dataset.

Table 5.1: Validation accuracy of different transformer-based architectures

Model	**Validation Accuracy (%)**
Temporal Only (TO)	64.25
Temporal-Frequency Sequential (TFS)	61.00
Temporal-Frequency Parallel (TFP)	61.75

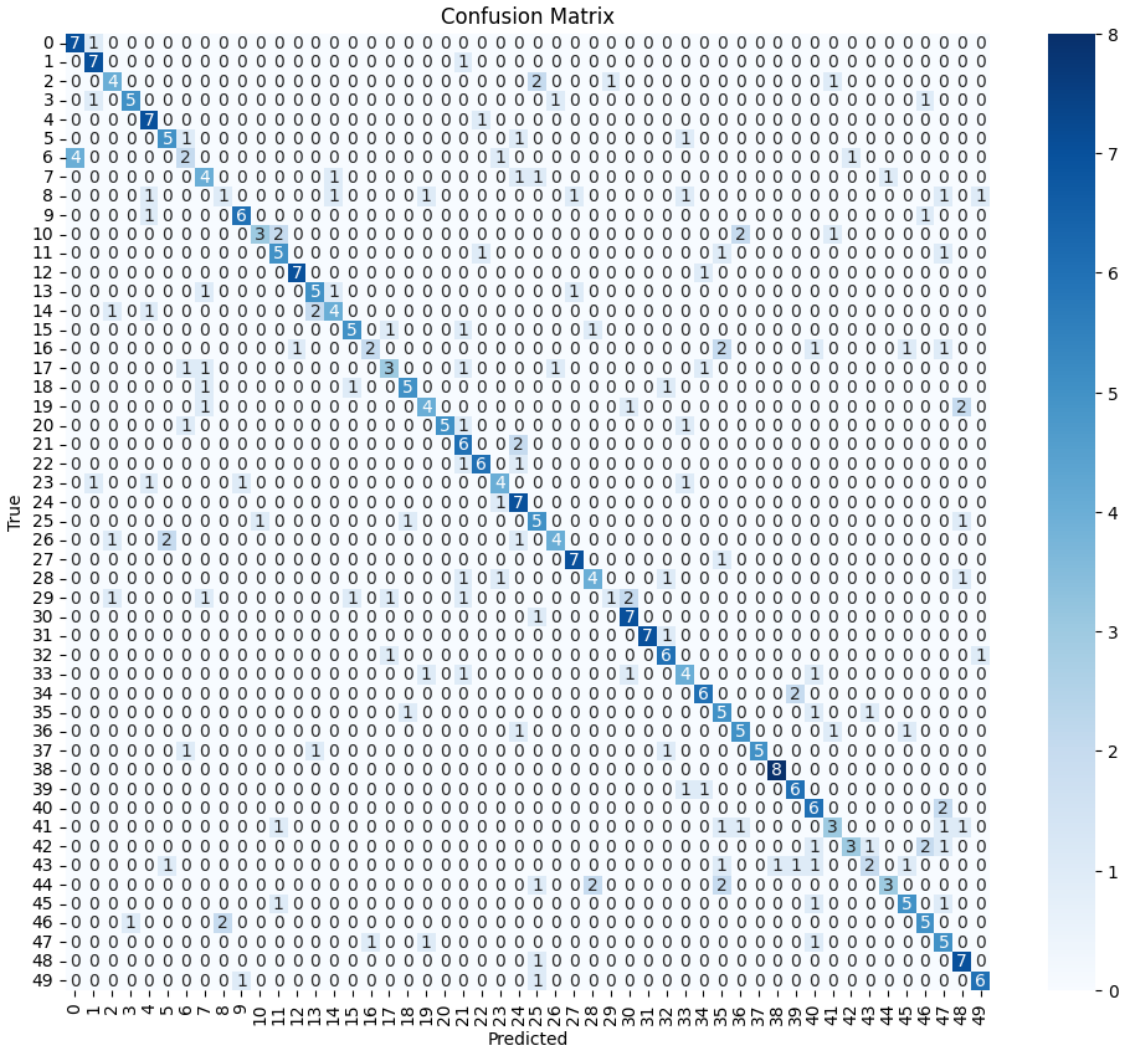


Figure 5.6: Confusion matrix for the Temporal-Frequency Sequential (TFS) transformer model on the ESC-50 dataset.

Chapter 6

Conclusions

To investigate the most efficient approach for achieving high accuracy and robust performance across varying conditions, this thesis explored multiple directions in the domain of environmental sound classification, focusing on three key aspects: (i) the impact of stacked feature representations and pretrained models on improving CNN performance, (ii) feature selection and pooling strategies in CNN-based architectures, and (iii) the implementation of Spectrogram Transformer architectures for audio classification.

Stacked Features for Enhanced CNN Performance

In the first major component of this research, we evaluated the impact of stacking multiple feature types—such as MFCCs, log-Mel spectrograms, Chroma, Spectral Contrast, and Tonnetz—as input to CNN models. The hypothesis was that aggregating complementary features would lead to more robust and generalizable representations. Extensive experiments were conducted on ESC-50 and UrbanSound8K datasets. Results confirmed that stacked features improved classification performance across all CNN variants. Combinations like MFCC + log-Mel or Log-Mel + Chroma + Spectral Contrast provided richer input representations that boosted overall classification accuracy. The improvement was particularly notable in the generalization phase when the models were fine-tuned on UrbanSound8K after being pre-trained on ESC-50, demonstrating the transferability of multi-feature representations.

The comparative evaluation also highlighted the trade-offs between performance and complexity. While stacked features increased input dimensionality, they improved feature diversity, allowing even lightweight CNNs to perform competitively in ESC tasks. We also observed that the AST model exhibits a dual nature in performance. When trained solely on the ESC-50 dataset without pretraining, the AST model shows limited effectiveness, achieving a validation accuracy of only 0.44. This underperformance highlights the AST model's reliance on large-scale pretraining to extract meaningful audio representations. However, when the AST model is pretrained on the large-scale AudioSet dataset and subsequently fine-tuned on ESC-50, its validation accuracy dramatically increases to 0.99, outperforming all CNN variants. Despite its superior performance,

the AST model requires significantly more training time compared to CNNs. This is due to its architectural complexity, particularly the Transformer-based encoder with multi-head self-attention layers. While this complexity enables the AST to effectively model global temporal and spectral dependencies, it also introduces a higher computational burden.

Feature Selection and Pooling Strategies

The second part of the thesis investigated the effectiveness of traditional feature selection techniques and advanced pooling methods for CNN-based ESC. Dimensionality reduction using Principal Component Analysis (PCA) was applied to flattened log-Mel spectrograms, revealing that while PCA can reduce feature redundancy, it may also remove subtle but important spectral features, leading to a significant drop in classification accuracy. This trade-off highlights the challenge of balancing compact representations with the retention of discriminative information in ESC tasks.

To overcome the limitations of uniform pooling strategies, we implemented two variants of Sparse Salient Region Pooling (SSRP): SSRP-B (basic fixed-window pooling) and SSRP-T (top-k based pooling). These models demonstrated substantial improvements over baseline architectures, with SSRP-T achieving a peak accuracy of 80.69% on the ESC-50 dataset using log-Mel input. The SSRP variants effectively emphasized salient time-frequency regions, enhancing model focus on discriminative features without significantly increasing complexity. The analysis demonstrated that SSRP can be a strong substitute for handcrafted feature selection by enabling CNNs to learn sparse and informative activations.

Transformer-Based Spectrogram Models

The final chapter focused on Spectrogram Transformers—an emerging deep learning architecture for audio classification. Specifically, we implemented and evaluated three variants: Temporal-Only (TO), Temporal-Frequency Parallel (TFP), Temporal-Frequency Sequential (TFS). These models were applied to log-Mel spectrogram inputs and trained on the ESC-50 dataset. Experimental results showed that, when trained from scratch, transformer-based models generally underperformed compared to CNN architectures. Their performance improves significantly only when pretrained on large-scale audio datasets. Due to their complexity and resource requirements, these models are most suitable in scenarios where computational constraints are not a limiting factor. Despite the higher computational cost, transformer models offer promising potential through their ability to capture global contextual information. Their scalability and interpretability—enabled by attention mechanisms—make them strong candidates for future applications in sound recognition and explainable environmental sound classification systems.

Overall, these models were applied to log-Mel spectrogram inputs and trained on the ESC-50 dataset. Experimental results revealed that, when trained from scratch, transformer-based architectures generally underperformed compared to Convolutional Neural Networks (CNNs). Their

performance improved significantly only when pretrained on large-scale audio datasets, highlighting their reliance on extensive training data. Due to their greater complexity and computational demands, transformers are better suited for scenarios where resources are not a limiting factor. In contrast, CNN-based models—particularly those enhanced with SSRP—consistently achieved superior performance while maintaining lower computational overhead. The CNN+SSRP architecture proved especially effective, striking an optimal balance between accuracy, efficiency, and robustness. This makes it highly suitable for real-time ESC applications on resource-constrained devices. While transformer models offer potential advantages in capturing global contextual information and supporting interpretability through attention mechanisms, our findings suggest that CNN+SSRP currently provides a more practical and high-performing solution for scalable environmental sound classification.

Future Work

Future research can extend this work in several directions:

- Investigate hybrid architectures that combine CNNs with Transformer blocks for efficient and scalable ESC models.
- Explore self-supervised pretraining and data augmentation strategies to improve model generalization in low-data settings.
- Deploy lightweight variants of SSRP and Transformer models on embedded hardware for real-time inference.
- Extend the evaluation to include larger, more diverse datasets such as AudioSet or real-world sound events.

In conclusion, this thesis advances the field of environmental sound classification by demonstrating how feature selection, feature stacking, pretrained models with fine-tuning, and attention-based learning can be strategically integrated to develop ESC systems that are both accurate and computationally efficient.

References

[1] Charlie Mydlarz, Mohit Sharma, Yitzchak Lockerman, Ben Steers, Claudio Silva, and Juan Pablo Bello. The life of a new york city noise sensor network. *Sensors*, 19(6):1415, 2019.

[2] Shannon Mattern. Urban auscultation; or, perceiving the action of the heart. *Places Journal*, 2020.

[3] Garima Sharma, Kartikeyan Umapathy, and Sridhar Krishnan. Trends in audio signal feature extraction methods. *Applied Acoustics*, 158:107020, 2020.

[4] Tuomas Virtanen, Mark D Plumbley, and Dan Ellis. *Computational analysis of sound scenes and events*. Springer, 2018.

[5] Siddique Latif, Heriberto Cuayáhuitl, Farrukh Pervez, Fahad Shamshad, Hafiz Shehbaz Ali, and Erik Cambria. A survey on deep reinforcement learning for audio-based applications. *Artificial Intelligence Review*, 56(3):2193–2240, 2023.

[6] Tong Lin, Yuchen Zhang, and Shiqi Liu. Environmental sound classification. http://noiselab.ucsd.edu, 2023. Accessed: 2025-06-30.

[7] Hamed Riazati Seresht and Karim Mohammadi. Environmental sound classification with low-complexity convolutional neural network empowered by sparse salient region pooling. *IEEE Access*, 11:849–862, 2022.

[8] Yixiao Zhang, Baihua Li, Hui Fang, and Qinggang Meng. Spectrogram transformers for audio classification. In *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE, 2022.

[9] Naoya Takahashi, Michael Gygli, and Luc Van Gool. Aenet: Learning deep audio features for video analysis. *IEEE Transactions on Multimedia*, 20(3):513–524, 2017.

[10] Xueyi Cheng. A comprehensive study of feature selection techniques in machine learning models. *Insights in Computer, Signals and Systems*, 1(1):10–70088, 2024.

[11] Shilpa Gupta, Varun Srivastava, and Deepika Kumar. Environment sound classification using stacked features and convolutional neural network. In *Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing*, pages 42–50, 2024.

[12] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

[13] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[14] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. A multi-device dataset for urban acoustic scene classification. *arXiv preprint arXiv:1807.09840*, 2018.

[15] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. Dcase 2017 challenge setup: Tasks, datasets and baseline system. In *DCASE 2017-workshop on detection and classification of acoustic scenes and events*, 2017.

[16] Aimee Allen, Tom Drummond, and Dana Kulić. Sound judgment: Properties of consequential sounds affecting human-perception of robots. *arXiv preprint arXiv:2502.02051*, 2025.

[17] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen. Context-dependent sound event detection. *EURASIP Journal on audio, speech, and music processing*, 2013:1–13, 2013.

[18] Miguel A Acevedo, Carlos J Corrada-Bravo, Héctor Corrada-Bravo, Luis J Villanueva-Rivera, and T Mitchell Aide. Automated classification of bird and amphibian calls using machine learning: A comparison of methods. *Ecological Informatics*, 4(4):206–214, 2009.

[19] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 17(10):1733–1746, 2015.

[20] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE, 2020.

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

[22] Dan Barchiesi, Dimitrios Giannoulis, Dan Stowell, and Mark D Plumbley. Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34, 2015.

[23] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, 2017.

[24] Yajuan Zhang and Li Wu. Attention-based cnn-bilstm for environmental sound classification. *Applied Acoustics*, 172:107619, 2021.

[25] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[26] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin Salamon. Sound event detection in domestic environments with weakly labeled data and soundscape synthesis. In *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[27] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, 2014.

[28] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015.

[29] Homer Dudley and S Balashek. Automatic recognition of phonetic patterns in speech. *The Journal of the Acoustical Society of America*, 30(8):721–732, 1958.

[30] Frieda Goldman-Eisler. Speech analysis and mental processes. *Language and speech*, 1(1):59–75, 1958.

[31] KN Stevens. Autocorrelation analysis of speech sounds. *The Journal of the Acoustical Society of America*, 22(5_Supplement):677–677, 1950.

[32] Calvin G Howard. Speech analysis-synthesis scheme using continuous parameters. *The Journal of the Acoustical Society of America*, 28(6):1091–1098, 1956.

[33] Ralph K Potter and John C Steinberg. Toward the specification of speech. *The Journal of the Acoustical Society of America*, 22(6):807–820, 1950.

[34] A Rihaczek. Signal energy distribution in time and frequency. *IEEE Transactions on information Theory*, 14(3):369–374, 1968.

[35] G Gambardella. Time scaling and short-time spectral analysis. *The Journal of the Acoustical Society of America*, 44(6):1745–1747, 1968.

[36] Yanxiong Li, Xue Zhang, Hai Jin, Xianku Li, Qin Wang, Qianhua He, and Qian Huang. Using multi-stream hierarchical deep neural network to extract deep audio feature for acoustic event detection. *Multimedia Tools and Applications*, 77:897–916, 2018.

[37] Yanxiong Li, Xianku Li, Yuhan Zhang, Wucheng Wang, Mingle Liu, and Xiaohui Feng. Acoustic scene classification using deep audio feature and blstm network. In *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 371–374. IEEE, 2018.

[38] Mohammad Hasan Rahmani, Farshad Almasganj, and Seyyed Ali Seyyedsalehi. Audio-visual feature fusion via deep neural networks for automatic speech recognition. *Digital Signal Processing*, 82:54–63, 2018.

[39] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[40] Dimitrios Giannoulis, Emmanouil Benetos, Dan Stowell, Mathias Rossignol, Mathieu Lagrange, and Mark D Plumbley. Detection and classification of acoustic scenes and events: An ieee aasp challenge. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013.

[41] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. Acoustic event detection in real life recordings. In *2010 18th European signal processing conference*, pages 1267–1271. IEEE, 2010.

[42] Daxin Jiang, Lie Lu, and Hong-Jiang Zhang. Music type classification by spectral contrast feature. *IEEE International Conference on Multimedia and Expo*, 1:113–116, 2002.

[43] Robert V. Shannon, Fan-Gang Zeng, Vijay Kamath, Joel Wygonski, and Michael Ekelid. Speech recognition with primarily temporal cues. *Science*, 270(5234):303–304, 1995.

[44] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[45] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[46] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.

[47] Andreas G Katsiamis, Emmanuel M Drakakis, and Richard F Lyon. Practical gammatone-like filters for auditory processing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2007:1–15, 2007.

[48] Jun Qi, Dong Wang, Yi Jiang, and Runsheng Liu. Auditory features based on gammatone filters for robust speech recognition. In *2013 IEEE international symposium on circuits and systems (ISCAS)*, pages 305–308. IEEE, 2013.

[49] Daniel PW Ellis. Gammatone-like spectrograms. *web resource: http://www. ee. columbia. edu/dpwe/resources/matlab/gammatonegram*, 2009.

[50] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 21–26, 2006.

[51] Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*, volume 5. Springer, 2015.

[52] Daniel PW Ellis and Graham E Poliner. Identifyingcover songs' with chroma features and dynamic programming beat tracking. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1429. IEEE, 2007.

[53] Fatih Ertam. An effective gender recognition approach using voice data via deeper lstm networks. *Applied Acoustics*, 156:351–358, 2019.

[54] Abdul Malik Badshah, Nasir Rahim, Noor Ullah, Jamil Ahmad, Khan Muhammad, Mi Young Lee, Soonil Kwon, and Sung Wook Baik. Deep features-based speech emotion recognition for smart affective services. *Multimedia Tools and Applications*, 78:5571–5589, 2019.

[55] Yanmin Qian, Nanxin Chen, and Kai Yu. Deep features for automatic spoofing detection. *Speech Communication*, 85:43–52, 2016.

[56] Yuji Tokozume and Tatsuya Harada. Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2721–2725. IEEE, 2017.

[57] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[58] Sachin Chachada and C-C Jay Kuo. Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processing*, 3:e14, 2014.

[59] Jivitesh Sharma, Ole-Christoffer Granmo, and Morten Goodwin. Environment sound classification using multiple feature channels and attention based deep convolutional neural network. In *Proceedings of Interspeech 2020*, pages 1186–1190, 2020.

[60] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[61] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[62] Jayita Mitra and Diganta Saha. An efficient feature selection in classification of audio files. *arXiv preprint arXiv:1404.1491*, 2014.

[63] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[64] Yaoshiang Ho and Samuel Wookey. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE access*, 8:4806–4813, 2019.

[65] Eduardo Fonseca, Xavier Favory, Jordi Pons, and Xavier Serra. Learning sound event classifiers from web audio with noisy labels. *ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25, 2019.

[66] Jordi Pons, Joan Serrà, and Xavier Serra. Training neural audio classifiers with few data. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20. IEEE, 2019.

[67] Emre Cakır, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1291–1303, 2017.

[68] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. *SciPy*, 2015:18–24, 2015.

[69] Zohaib Mushtaq and Shun-Feng Su. Environmental sound classification using a regularized deep convolutional neural network with data augmentation. *Applied Acoustics*, 167:107389, 2020.

[70] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):194–201, 2011.