POZNAN UNIVERSITY OF TECHNOLOGY

FACULTY OF AUTOMATIC CONTROL, ROBOTICS, AND ELECTRICAL ENGINEERING

Bachelor's thesis

INTERPRETATION OF REPRESENTATIONS OF AUDIO SIGNALS IN HIDDEN LAYERS OF A NEURAL NETWORK USING NMF

Roshan Dwivedi, 146903

Supervisor
dr hab. Szymon Drgas

POZNAŃ 2024

Diploma thesis card scan will be placed here.

Abstrakt

Głębokie uczenie zrewolucjonizowało dziedzinę przetwarzania sygnałów dźwiękowych, umożliwiając bardzo dokładną klasyfikację dźwięków środowiskowych. Konwolucyjne sieci neuronowe (CNN - convolutional neural network) wykazały sukces w wydobywaniu hierarchicznych reprezentacji cech ze spektrogramów dźwięku. Niemniej jednak, interpretowalność tych wyuczonych reprezentacji pozostaje istotnym wyzwaniem, co ogranicza ich szersze zastosowanie w krytycznych aplikacjach, gdzie przejrzystość modeli jest kluczowa. Zrozumienie wewnętrznych mechanizmów działania tych sieci może prowadzić do bardziej przejrzystych i odpornych modeli, zmniejszając ryzyko związane z czarnymi skrzynkami systemów sztucznej inteligencji.

Niniejsze badania analizują wewnętrzne reprezentacje cech sieci CNN trenowanych do klasyfikacji dźwięków środowiskowych, ze szczególnym uwzględnieniem zrozumienia aktywacji ukrytych warstw przy użyciu nieujemnej factoryzacji macierzy (NMF - non-negative matrix factorization). Technika NMF została zastosowana jako metoda dekompozycji do analizy map cech generowanych wewnątrz sieci, co pozwala ujawnić ukryte struktury przyczyniające się do podejmowania decyzji klasyfikacyjnych. Badania przeprowadzono na zbiorze danych ESC-50 [17], szeroko uznawanym benchmarku obejmującym dźwięki środowiskowe z różnych kategorii, co stanowi zróżnicowane i wymagające zadanie klasyfikacyjne.

Ponadto niniejsza praca analizuje wpływ mechanizmu uwagi, który dynamicznie wzmacnia istotne cechy, jednocześnie tłumiąc te mniej informacyjne. Poprzez włączenie mechanizmu uwagi do architektury modelu, analizujemy jego rolę w poprawie ekstrakcji cech oraz wydajności klasyfikacji. Połączenie interpretowalności opartej na NMF z mechanizmami uwagi pozwala uzyskać głębszy wgląd w proces podejmowania decyzji przez sieć, co może przyczynić się do opracowania bardziej efektywnych i przejrzystych architektur sieci neuronowych.

Wykorzystując metodę NMF, praca przedstawia systematyczne podejście do zmniejszenia luki między wysokowydajnymi sieciami neuronowymi a ich interpretowalnością. Wyniki badań przyczyniają się do zwiększenia przejrzystości modeli, usprawnienia procesów selekcji cech oraz optymalizacji architektur sieci neuronowych stosowanych w aplikacjach przetwarzania dźwięku.

Słowa kluczowe: Głębokie uczenie, sieci neuronowe, klasyfikacja dźwięku, Niefaktoryzująca Macierz Rozkładu, interpretowalność cech, ESC-50, uwaga kanałowa, przejrzystość modeli

Abstract

Deep learning has revolutionized the field of audio signal processing, enabling highly accurate classification of environmental sounds. Convolutional neural networks (CNNs) have demonstrated remarkable success in extracting hierarchical feature representations from audio spectrograms. However, the interpretability of these learned representations remains a significant challenge, limiting their broader adoption in critical applications where transparency is essential. Understanding the inner workings of these networks can lead to more explainable and robust models, reducing the risks associated with black-box AI systems.

This thesis explores the internal feature representations of CNNs trained for environmental sound classification, with a particular focus on understanding hidden layer activations using non-negative matrix factorization (NMF). NMF is employed as a decomposition technique to analyze the feature maps generated within the network, revealing latent structures that contribute to classification decisions. The study is conducted using the ESC-50 dataset [17], a widely recognized benchmark comprising environmental sounds across various categories, providing a diverse and challenging classification task.

Furthermore, this study investigates the impact of the channel attention Mechanism, which dynamically enhances relevant features while suppressing less informative ones. By incorporating channel attention into the model architecture, we examine its role in improving feature extraction and classification performance. The combination of NMF-based interpretability and attention mechanisms provides deeper insights into the network's decision-making process, potentially guiding the development of more efficient and transparent neural architectures.

By leveraging NMF, this work presents a systematic approach to bridging the gap between high-performance neural networks and model explainability. The findings contribute to enhancing model transparency, improving feature selection processes, and optimizing neural network architectures for audio processing applications.

Keywords: Deep Learning, Neural Networks, Audio Classification, Non-Negative Matrix Factorization, Feature Interpretability, ESC-50, Channel Attention, Model Transparency

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The rapid advancement of deep learning techniques has significantly transformed the field of audio signal processing. With neural networks now routinely achieving high accuracy in tasks such as environmental sound classification [12, 4, 25], the focus has shifted toward understanding the internal representations that these models develop. In particular, interpreting the latent features within the hidden layers is essential to both demystify the decision-making process of neural networks and to enhance their robustness and generalization capabilities. This study addresses the challenge of interpreting representations of audio signals using non-negative matrix factorization (NMF) to decompose the complex feature maps generated in hidden layers [12, 4, 25].

The motivation for this work is twofold. First, as neural networks become deeper and more complex, it becomes increasingly difficult to pinpoint which features are critical for classification and how these features evolve across layers. Second, applying NMF offers a systematic approach to uncover hidden structures within the spectrogram representations, providing insights that could lead to improved neural architectures and more transparent model behavior. By leveraging both data-driven learning and interpretable decomposition techniques, the research aims to bridge the gap between high-performance audio classification and model explainability.

The scope of this work includes a detailed examination of the hidden layers of a convolutional neural network trained on the ESC-50 dataset [16], a well-known benchmark for environmental sound classification. The study focuses on analyzing the spectral patterns extracted by the network and validating the interpretations through both quantitative experiments and visual inspection of the feature maps. In addition, the investigation includes an evaluation of the impact of advanced mechanisms, such as channel attention [7], on feature extraction and classification performance. The underlying hypothesis is that NMF can effectively reveal meaningful sub-components of the hidden representations, thereby enhancing our understanding of the learned features and their contribution to classification decisions.

The aim of this work is to develop and apply an interpretative framework that utilizes NMF to analyze the hidden layer representations of a convolutional neural network trained for audio classification. This framework is intended to provide deeper insights into the feature extraction process and to identify potential avenues for architectural optimization and performance enhancement.

The structure of this work is as follows.

- The next chapter presents a comprehensive literature review, detailing the theoretical foundations and previous studies related to audio signal processing, neural network interpretability, and NMF.
- Chapter 3 developed methods - a neural network for sound classification with NMF-based interpretability module.
- Chapter 4 focuses on the pattern analysis and interpretation of the extracted feature maps,
- Chapter 5 concludes the thesis with a summary of the key findings and directions for future research.
- Finally, Appendix provides a detailed results section, which sheds light onto the design choices employed to obtain the spectral patterns

Chapter 2

Theoretical Foundations

In recent years, deep learning-based approaches have significantly advanced the field of audio classification, enabling accurate recognition of environmental sounds, speech, and music [11, 2]. The increasing availability of benchmark datasets, such as ESC-50 [17], has facilitated the development of robust neural network architectures that leverage spectrogram-based representations to extract meaningful audio features. This work presents a comprehensive methodology for environmental sound classification, integrating convolutional neural networks (CNNs) with advanced feature extraction and interpretability techniques, building upon the foundational work of Gaël Richard in acoustic feature analysis for environmental sound [19]

The developed workflow consists of four primary stages: data acquisition and pre-processing, feature extraction, neural network-based classification, and model analysis. The initial stage ensures standardized data processing through mel spectrogram conversion and augmentation strategies [2]. Subsequently, deep feature representations are extracted using CNNs [11], with an additional non-negative matrix factorization (NMF) technique [13] applied to enhance interpretability. The classification stage employs a carefully designed CNN-based model with dropout regularization [21] and batch normalization [9] to achieve robust multi-class predictions. Finally, visualization techniques such as Grad-CAM [18] and attention maps [24] are used to analyze model decisions, ensuring the network's outputs are based on meaningful spectral patterns.

This study aims to contribute to the field by implementing an effective deep learning pipeline for environmental sound classification, emphasizing both performance and model interpretability. The following sections provide an in-depth discussion of each stage, including data preparation, neural network architecture, feature extraction mechanisms, and performance evaluation methodologies.

2.1 Neural Networks in Audio Processing

Convolutional Neural Networks (CNNs) have emerged as a powerful tool in audio analysis due to their inherent ability to automatically learn hierarchical representations from time-frequency data, such as spectrograms. This capability allows CNNs to capture both local spectral details and global temporal patterns, which are critical for accurate audio classification. However, despite their high performance, the internal representations learned by these networks often remain opaque, posing a significant challenge in understanding how decisions are made. This thesis addresses this gap by integrating traditional CNN architectures with non-negative matrix factorization (NMF) techniques, thereby providing a transparent and interpretable framework for audio processing.

In CNN architectures designed for audio tasks, the early convolutional layers focus on extracting low-level features, such as edges and simple frequency components, directly from the spectrograms.

As the signal passes through subsequent layers, these basic features are gradually abstracted into more complex patterns—capturing harmonic structures, rhythmic elements, and other subtle spectral nuances that differentiate sound classes. This progressive abstraction is not only fundamental for effective classification but also forms the basis for our interpretability approach, as it highlights the transformation of raw audio signals into meaningful high-level representations [4].

The convolution operation, which lies at the heart of CNNs, mathematically formalizes this process. For a 2D convolution used in our audio spectrogram processing, this can be expressed as

$$(F * K)_{i,j} = \sum_{m=0}^{k_h-1} \sum_{n=0}^{k_w-1} F_{i+m,j+n} \cdot K_{m,n},$$

where F is the input spectrogram, K is the convolutional kernel of size $k_h \times k_w$, and (i, j) represents spatial coordinates in the feature map. In our enhanced architecture, these convolutions are followed by batch normalization and ReLU activation to improve training stability and introduce non-linearity [5, 9].

The Squeeze-and-Excitation (SE) blocks further refine these feature representations by explicitly modeling channel interdependencies. The SE mechanism can be formalized as:

$$\mathbf{z} = F_{ex}(F_{sq}(\mathbf{u})) = \sigma(W_2\delta(W_1\mathbf{z}_{avg,max})),$$

where $\mathbf{u}$ is the input feature map, F_{sq} represents the squeeze operation combining average and max pooling, F_{ex} is the excitation operation with fully connected layers W_1 and W_2, δ is the ReLU function, and σ is the sigmoid activation [8].

This study builds on the foundational strengths of CNNs [14] and addresses their inherent interpretability challenge by introducing an NMF-based decomposition of the input spectrogram using processed feature maps of the CNN as activations. By leveraging our five-block deep architecture with SE attention mechanisms, the work not only enhances classification performance on the ESC-50 environmental sound dataset but also provides clear insights into which spectral-temporal components contribute most to the network's decisions. In this way, our methodology bridges the gap between high-performance audio classification (achieving state-of-the-art results with 21.5M parameters) and the growing need for model transparency in complex neural systems.

2.1.1 Attention Mechanisms

Attention mechanisms, such as squeeze-and-excitation (SE) blocks [10], further enhance interpretability by recalibrating the channel-wise responses of feature maps. SE blocks operate by aggregating feature statistics and using them to modulate the importance of different channels, thereby highlighting the most informative parts of a given feature map. This selective emphasis not only improves performance but also aids in understanding which spectral regions are critical for classification.

2.2 Audio Signal Processing Techniques

Before neural networks can effectively process audio data, raw waveforms must be transformed into representations that highlight perceptually relevant features.

2.2.1 Time-Frequency Representations

For discrete (sampled) signals, the continuous Short-Time Fourier Transform (STFT) is replaced by a summation over individual samples. The discrete STFT is defined as:

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mR]\, w[n]\, e^{-j\frac{2\pi kn}{N}},$$

where:

- $x[n]$ is the discrete-time signal,
- $w[n]$ is a window function of length N,
- m is the time frame index corresponding to successive window positions,
- R is the hop size (i.e., the number of samples by which the window shifts), and
- k is the frequency bin index, with the corresponding angular frequency $\omega_k = \frac{2\pi k}{N}$.

This formulation divides the signal into potentially overlapping segments (by shifting the window by R samples) and applies the Discrete Fourier Transform (DFT) to each segment. The resulting spectrogram displays the evolution of the signal's frequency content over time [23, 1].

2.2.2 Mel Spectrograms for Discrete Signals

For sampled signals, the STFT is computed over windowed frames and its squared magnitude is mapped to the mel scale via discrete mel filter banks. These filters are sampled at FFT bin frequencies, with center frequencies defined by the inverse mel formula

$$f = 700\left(10^{\frac{m}{2595}} - 1\right).$$

Applying triangular filters to the power spectrum yields a mel spectrogram aligned with human auditory perception [22, 20].

2.2.3 Data Augmentation

To improve the robustness and generalization of the neural network, augmentation techniques such as noise addition and time shifting are employed. These methods in Section 4.2.2 simulate real-world variability by slightly altering the audio signals, thereby encouraging the model to learn invariant features that are robust to minor perturbations and can be seen in section 4.2

2.3 Non-Negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization (NMF) is a linear decomposition method that is particularly suited for non-negative data such as spectrograms. It decomposes a non-negative matrix V into the product of two non-negative matrices W and H:

$$V \approx WH,$$

where:

- V is the input spectrogram.
- W contains the basis functions that capture distinctive spectral patterns.
- H **Activation Matrix** holds the activation coefficients that indicate how these basis functions are combined over time.

The non-negativity constraints ensure that the basis and coefficients are interpretable in an additive manner, allowing each basis vector to represent a part of the overall signal. This property is especially useful when interpreting hidden layer representations from neural networks, as it provides a clear, parts-based understanding of the learned features [25].

Typically, in NMF, both W and H are optimized. However, in this approach, H is obtained from the feature maps of the Classification Network, and only W is optimized. The columns of W represent spectral patterns, which collectively compose the input spectrogram.

2.4 Integration of Methodologies

The integration of CNN-based feature extraction with classical signal processing and NMF forms a powerful framework for audio analysis[3, 12]. The process begins with the transformation of raw audio into normalized mel spectrograms, continues with the extraction of hierarchical features via a CNN, and culminates with the application of NMF to decompose these features into interpretable components.**??** This synergistic approach yields several advantages:

- **Robust Feature Representation:** The pre-processing and augmentation techniques ensure that the input data is both informative and standardized.
- **Hierarchical Abstraction:** CNNs effectively capture complex spectral-temporal relationships, enabling the discrimination of subtle differences among audio classes.
- **Interpretability:** NMF and attention mechanisms provide insights into the internal workings of the network, offering an interpretable framework for understanding the parts-based structure of audio signals.

2.5 Summary

In summary, this chapter has presented a theoretical framework for the analysis and interpretation of audio signals using neural networks. The discussion has covered:

- The role of CNNs in hierarchical feature learning and their mathematical foundations.
- The importance of signal processing techniques such as the STFT and mel spectrogram transformation, including normalization and data augmentation.
- The application of Non-Negative Matrix Factorization as a tool for decomposing and interpreting spectrogram data.
- Advanced methods for interpreting hidden layer representations, including hook-based feature extraction and attention mechanisms.

The integration of these methodologies not only enhances audio classification performance but also provides a deep understanding of the underlying spectral patterns. This theoretical foundation sets the stage for the experimental investigations and analyses discussed in subsequent chapters.

Chapter 3

Developed Methods

3.1 Introduction

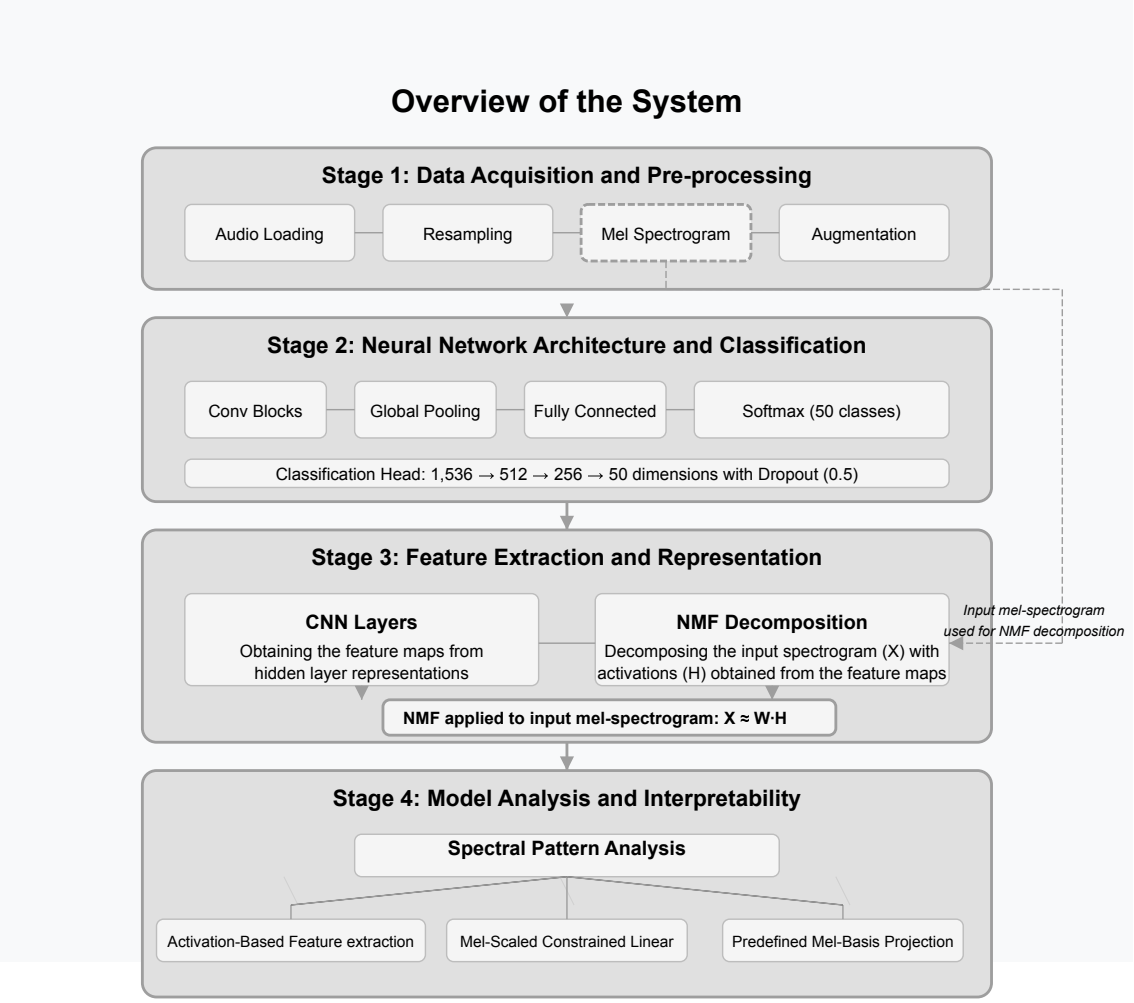


FIGURE 3.1: An overview of the developed system

This chapter outlines the comprehensive methodology developed for environmental sound classification, which integrates deep learning techniques with advanced feature extraction and interpretability tools. The system begins with the acquisition and pre-processing of raw audio data. Audio signals are first loaded and resampled to a fixed sampling rate, then transformed into normalized mel spectrograms using time-frequency analysis. To simulate real-world variability and

enhance robustness, augmentation techniques such as noise addition and time shifting are applied. This standardized pre-processing ensures that all inputs maintain consistent dimensions, which is critical for efficient batch processing in deep learning pipelines [2].

Following data pre-processing, the system employs a convolutional neural network (CNN) to extract hierarchical spectral-temporal features from the prepared audio signals. A hook-based mechanism is strategically integrated into the network, allowing the capture of intermediate layer outputs without interrupting the computational flow. These extracted feature maps are further decomposed using non-negative matrix factorization (NMF) to obtain interpretable basis functions and activation coefficients, thereby offering a parts-based interpretation of the learned representations [13]. This dual approach of using CNNs for feature extraction and NMF for decomposition enables the system to reveal underlying spectral patterns that contribute to the classification process.

3.2 Overall Workflow

The complete process comprises four primary stages:

1. **Data Acquisition and Pre-processing:** Raw audio signals are loaded, resampled, and converted into normalized mel spectrograms. Data augmentation techniques, such as noise addition and time shifting, are applied to simulate real-world variability.

2. **Feature Extraction and Representation:** A convolutional neural network (CNN) is employed to extract hierarchical spectral-temporal features. A hook-based mechanism captures intermediate layer outputs, offering insights into the learned representations. Simultaneously, NMF is used to decompose these feature maps into interpretable components .

3. **Neural Network Architecture and Classification:** The extracted features are fed into a classification head. This module, which incorporates dropout, batch normalization, and fully connected layers, is designed to achieve robust multi-class categorization. Architectural choices are justified based on efficiency and performance considerations.

4. **Model Analysis and Interpretability:** Post-training analysis now exclusively leverages NMF-based interpretability, confirming that the network's decisions are anchored in meaningful spectral patterns.

3.3 Detailed Methodology

3.3.1 Data Acquisition and Pre-processing

The initial stage of the process involves standardizing raw audio data to ensure uniformity across samples. Audio files are loaded and resampled to a fixed sampling rate, then transformed into mel spectrograms using appropriate time-frequency analysis. The spectrograms are normalized by centering and scaling, which facilitates effective training of the neural network. Moreover, to account for the variability of audio duration, each spectrogram is adjusted to a fixed size via padding or truncation. These steps ensure that every sample has consistent dimensions, crucial for efficient batch processing in deep learning pipelines.

3.4 Neural Network Architecture

The choice of neural network in the present study is convolutional neural network (CNN). The model works by processing mel spectrogram representations of audio signals and taking advantage of the powerful feature extraction property of CNNs. The choice is motivated by the capacity of CNNs for finding hierarchical structures in spectrograms, as they are applied in image classification. Since audio spectrograms have regular structure, CNNs excel at picking up useful spectral and temporal features that are necessary for good classification.

Although CNNs have demonstrated considerable proficiency at image classification, other architectures such as ResNet and MobileNet [6, 5] have been considered. ResNet, which is characterized by its deep residual connections, facilitates improved gradient flow and feature learning, thereby rendering it immensely powerful for very deep networks. Yet, its higher computational cost and memory requirements made it less convenient for our specific application. MobileNet, which is low-resource deployable, employs depthwise separable convolutions to minimize computational expense without compromising an acceptable amount of accuracy. Yet, its mobile-optimized design can result in a compromise on representational capacity relative to typical convolutional neural networks (CNNs). Our selected CNN model provides a good trade-off between computational efficiency and classification performance and therefore is an appropriate option for environmental sound classification. [4] [25]

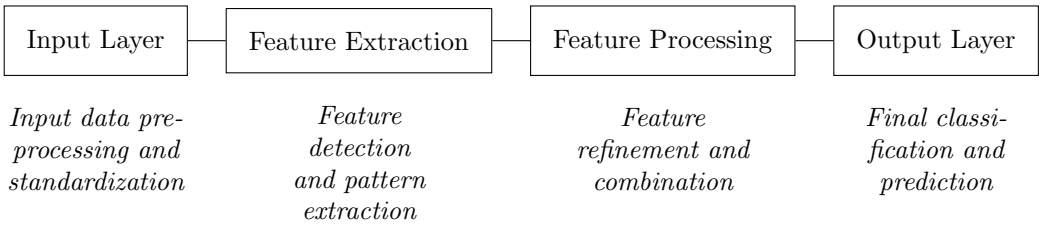


Figure 3.1: Neural Network Architecture Overview

3.5 Architecture Overview

The parts of the system is defined as follows:

- **Input Layer:** It processes 2-dimensional mel spectrograms with dimensions of the form (n_mels, max_frames), with the value of n_mels being 128 and max_frames being 431. It ensures equal representations of all the input sounds, thus enabling the extraction of similar features from different sound clips.
- **Convolutional Blocks:** A chain of convolution layers each being specifically designed to detect different levels of spectra as well as their temporal characteristics. Each block contains:
  - 2D convolutional layers with ReLU activation to extract meaningful features, leveraging spatial hierarchies in the spectrograms.
  - Batch normalization to stabilize activations, reduce internal covariate shift, and speed up convergence.
  - Max-pooling layers reduce the feature map dimensions but keep the more important details, thus reducing computational requirements.

- **Fully connected layers:** convert higher-order feature representations to a one-dimensional array, then process this information through dense layers that utilize dropout regularization to prevent overfitting and improve generalization.
- **Output Layer:** The final stage involves the use of a softmax classifier intended to yield score probabilities with 50 dimensions representing the 50 classes found in the ESC-50 set, thus enabling classification under a system-level multi-class system.

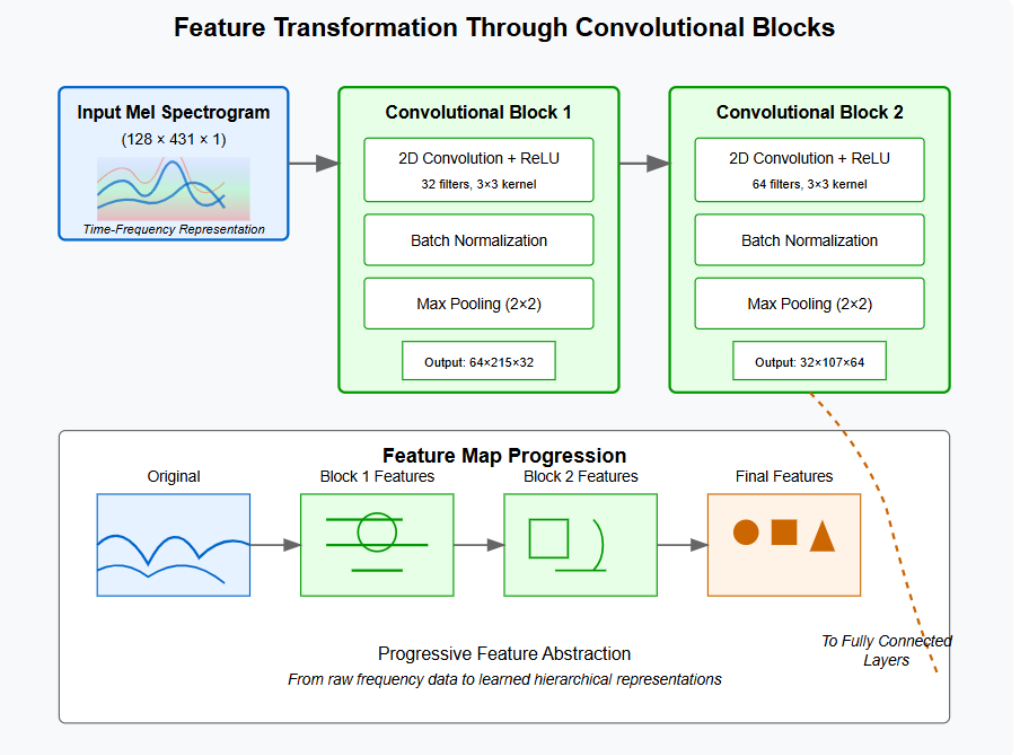


FIGURE 3.2: Feature Transformation in Convolutional Neural Network

The detailed layer-wise representation is presented in the table below:

Layer Type	Parameters	Activation	Purpose
Conv2D (32)	3x3 kernel, stride=1, padding=same	ReLU	Extract spectral-temporal features
MaxPool2D	Pool size 2x2	-	Reduce spatial size, retain key features
Conv2D (64)	3x3 kernel, stride=1, padding=same	ReLU	Learn deeper spectral patterns
MaxPool2D	Pool size 2x2	-	Further downsampling
Conv2D (128) → Conv2D (256)	3x3 kernel, stride=1, padding=same	ReLU	Capture complex hierarchical representations
GlobalAvgPool2D	-	-	Reduce feature maps to a single vector
Fully Connected (256)	-	ReLU	Learn high-level feature relationships
Dropout (0.5)	-	-	Prevent overfitting
Fully Connected (128)	-	ReLU	Further feature abstraction
Fully Connected (50)	-	Softmax	Output class probabilities

TABLE 3.1: Neural Network Architecture

3.6 Architectural Choices

Convolutional Neural Networks (CNNs) are particularly effective for analyzing mel spectrograms due to their ability to capture structured spatial patterns. Unlike traditional machine learning methods that rely on handcrafted features, CNNs automatically learn and extract relevant spectral

and temporal characteristics. This makes them well-suited for audio classification tasks, as they can efficiently recognize complex dependencies within spectrogram data.

To enhance training stability, *Batch Normalization* is applied after each convolutional layer. This technique normalizes activations, reducing internal covariate shifts and ensuring the network trains efficiently even with deeper architectures. Additionally, *Dropout Regularization* is incorporated into fully connected layers to mitigate overfitting. By setting a dropout rate of 50%, the model achieves an optimal balance between learning and generalization, preventing reliance on specific neurons.

Another key component is *Global Average Pooling*, which reduces the dimensionality of feature maps while preserving essential information. This not only prevents overfitting but also helps maintain strong feature representations. Finally, a *Softmax Classifier* at the output layer ensures probabilistic predictions, making it ideal for multi-class classification. By assigning confidence scores to each class, the model can effectively interpret and differentiate between multiple categories.

3.7 Classification Head Architecture

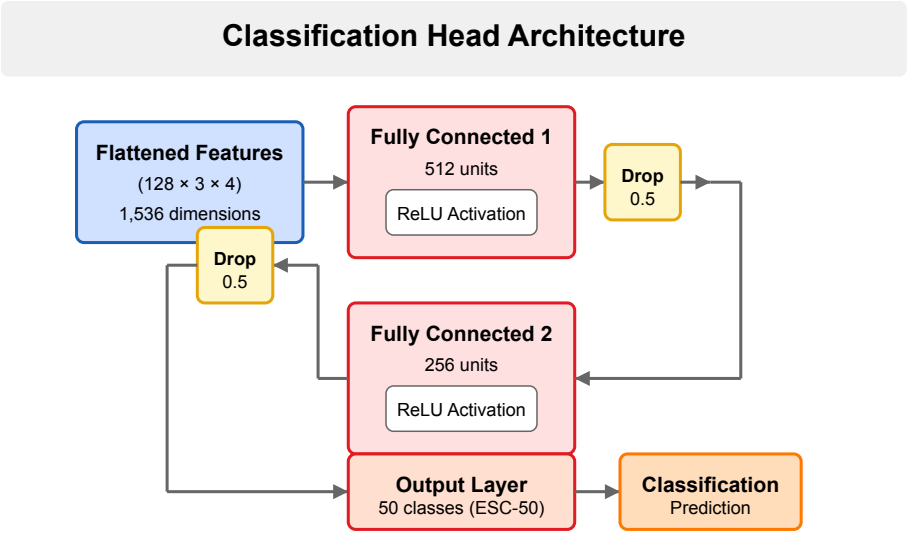


FIGURE 3.3: Classification Architecture visualisation

The classification head depicted in the diagram represents a specialized architecture for audio event classification, particularly tailored for the ESC-50 (Environmental Sound Classification) dataset, which comprises 50 distinct sound categories.

3.7.1 Architectural Components and Flow

The architecture begins with a flattened feature representation of dimensionality 1,536 (organized as $128 \times 3 \times 4$ tensor), likely extracted from convolutional or transformer-based layers that process spectral or time-domain audio features. This high-dimensional representation undergoes several transformations through a network of fully-connected layers and regularization components.

- **First Fully-Connected Layer** A dense layer with 512 units implementing dimensionality reduction while preserving discriminative features. This layer employs ReLU activation functions to introduce non-linearity, enabling the model to learn complex decision boundaries.

- **Dropout Regularization (0.5)** A critical regularization mechanism that stochastically deactivates 50% of neurons during training, thus preventing co-adaptation of feature detectors and mitigating overfitting—particularly important given the relatively limited size of ESC-50 (2,000 samples).
- **Second Fully-Connected Layer** A subsequent dense layer with 256 units, further reducing dimensionality while refining the feature representation. This layer also utilizes ReLU activation, maintaining the non-linear mapping capability.

3.7.2 Output Layer

The final fully-connected layer consists of 50 units, corresponding to the sound categories in ESC-50. Although not explicitly stated, this layer presumably employs softmax activation to produce normalized probability distributions across classes.

3.7.3 Theoretical Implications

This classification head architecture exemplifies several important principles in audio classification:

- **Progressive Dimensionality Reduction** The gradual reduction from 1,536 $\rightarrow$ 512 $\rightarrow$ 256 $\rightarrow$ 50 dimensions reflects the principle of hierarchical feature abstraction, where increasingly compact representations capture higher-level semantics.
- **Aggressive Regularization**The implementation of multiple dropout layers with relatively high rates (0.5) indicates a deliberate emphasis on generalization capability, particularly important for environmental sound classification where variability within classes can be substantial.
- **Multi-path Information Flow**The inclusion of skip connections acknowledges that different sound classes may benefit from different levels of feature abstraction—some sounds may be recognized from fine-grained spectro-temporal patterns, while others might require higher-level semantic processing.

This architecture represents a balanced approach to the audio classification task, incorporating modern deep learning practices while maintaining computational efficiency appropriate for the ESC-50 dataset's scale and complexity.

3.8 NMF based method for Interpretability of the Neural Network

In this work, the employed neural network architectures that promote reconstruction and interpretability of audio features by incorporating transformation methods along with perceptually motivated constraints. The core idea is to employ the a neural network in such a way that inputs the extracted feature from a layer of the classifier model and outputs an interpretable spectral patterns as discussed section 2.3 in that are meaningful and interpretable bridging the gap between raw feature extraction and human audition.

The method is founded on three key models:

1. Linear Activation-Based Feature Transformation: This model uses a linear transformation with constraints, where the weights are guaranteed to be non-negative through the application of the Softplus function. The lack of a bias term also aids in preserving the stable and linear mapping of input features to output mel-scaled spectrograms. This architecture is especially ideal for spectrogram-like data since it maintains the non-negative and interpretable nature of frequency representations.

2. Mel-Scaled Constrained Linear Transformation: This model extends standard linear transformations with dynamically generated filter weights that are frequency-dependent. By adding perceptual principles—adaptive filter center frequencies, dynamic bandwidth controls, and triangular spectral approximations—it is able to accurately simulate the human auditory system. This allows the model not only to transform features but also to comprehend and highlight the spectral context, creating a representation that preserves subtle auditory details.
3. Predefined Mel-Basis Projection: In contrast to the adaptive strategy of the former models, this model is based on a more restricted methodology. It is initialized with its weights close to approximating a mel filter bank with a Gaussian-shaped basis. This enables an interpretable and structured frequency transformation right from the start while nevertheless enabling small adaptations throughout the training procedure. This type of approach is particularly beneficial for small datasets, as it restricts the amount of trainable parameters without a loss of interpretability.

The proposed methods integrate robust pre-processing, advanced feature extraction, and interpretability techniques into a cohesive workflow. This methodology provides a framework for analyzing audio signal representations within deep learning models. By combining CNN-based feature extraction with NMF decomposition and interpretability analyses, the approach not only enhances classification performance but also offers valuable insights into the underlying spectral patterns. This comprehensive framework sets the stage for subsequent experimental validation and further architectural optimization.

Chapter 4

Experiments

The Dataset used in this study is the **ESC-50-dataset** [17], a well known benchmark for environmental sound classification. The dataset contains environmental audio recordings of a wide range of everyday sounds. It is designed to facilitate research in machine learning-based classification and feature learning.

4.1 Dataset

The ESC-50 dataset contains **2,000 labeled audio recordings**, each lasting **5 seconds** and sampled at **44.1 kHz**. The recordings fall into **50 classes**, with **40 examples** per class. The classes have been grouped into five broad categories:

- **Animals** (i.e., dog bark, cat meow, birds chirping)
- **Natural soundscapes & water sounds** (i.e., rain, thunderstorm, waves)
- **Human sounds** (i.e., coughing, sneezing, laughter)
- **Household internal sounds** (e.g., typing, door knock, vacuum cleaner)
- **Urban sounds** (e.g., engine noise, siren, car horn)

Each sample is a monophonic waveform, so the dataset is appropriate for a variety of audio-based machine learning tasks, such as classification and feature analysis, in a broad sense.

4.1.1 Data Split

In order to train and test the neural network, the dataset is split into three distinct subsets: **training, validation, and testing**. According to the default ESC-50 protocol, the dataset is already split into **five folds** for cross-validation. In this work, four folds (1,600 samples) are utilized for training and validation, and one fold (400 samples) for testing. In particular:

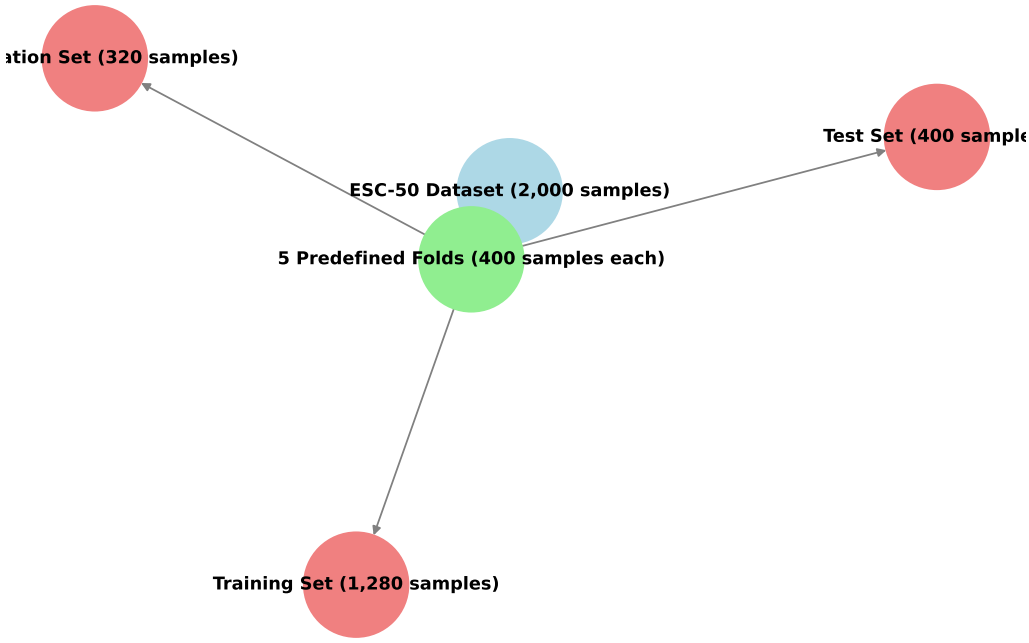


FIGURE 4.1: Visualisation of the ESC-50 Dataset

- **Training Set:** 1,280 samples (80% of the training data)
- **Validation Set:** Consists of 320 samples, corresponding to 20% of the training set.
- **Test Set:** Consists of 400 samples, corresponding to a single predefined fold.

The split above exposes the model to a wide variety of auditory signals, while keeping an independent test set for an unbiased assessment of performance. The folds established in ESC-50 offer a standardized basis for comparison with previous research.

4.1.2 Data Characteristics

- **Audio Format:** WAV files, 16-bit, 44.1 kHz
- **Duration:** Each clip is exactly **5 seconds** long
- **Mono-channel:** Single-channel audio
- **Class Distribution:** Each class is **balanced**, with **40 samples per class**
- **Predefined Folds:** The dataset is pre-divided into **five non-overlapping folds** for cross-validation

TABLE 4.1: ESC-50 Audio Characteristics

Characteristic	Description
Audio Format	WAV files, 16-bit, 44.1 kHz
Duration	5 seconds
Channels	Mono
Class Distribution	40 samples per class
Folds	5 non-overlapping folds for cross-validation

This dataset is a challenging yet nicely formatted benchmark for the evaluation of neural networks for audio classification. Its class balance ensures no particular sound class dominates the learning process, so the model trains fairly and unbiasedly.

4.2 Pre-processing

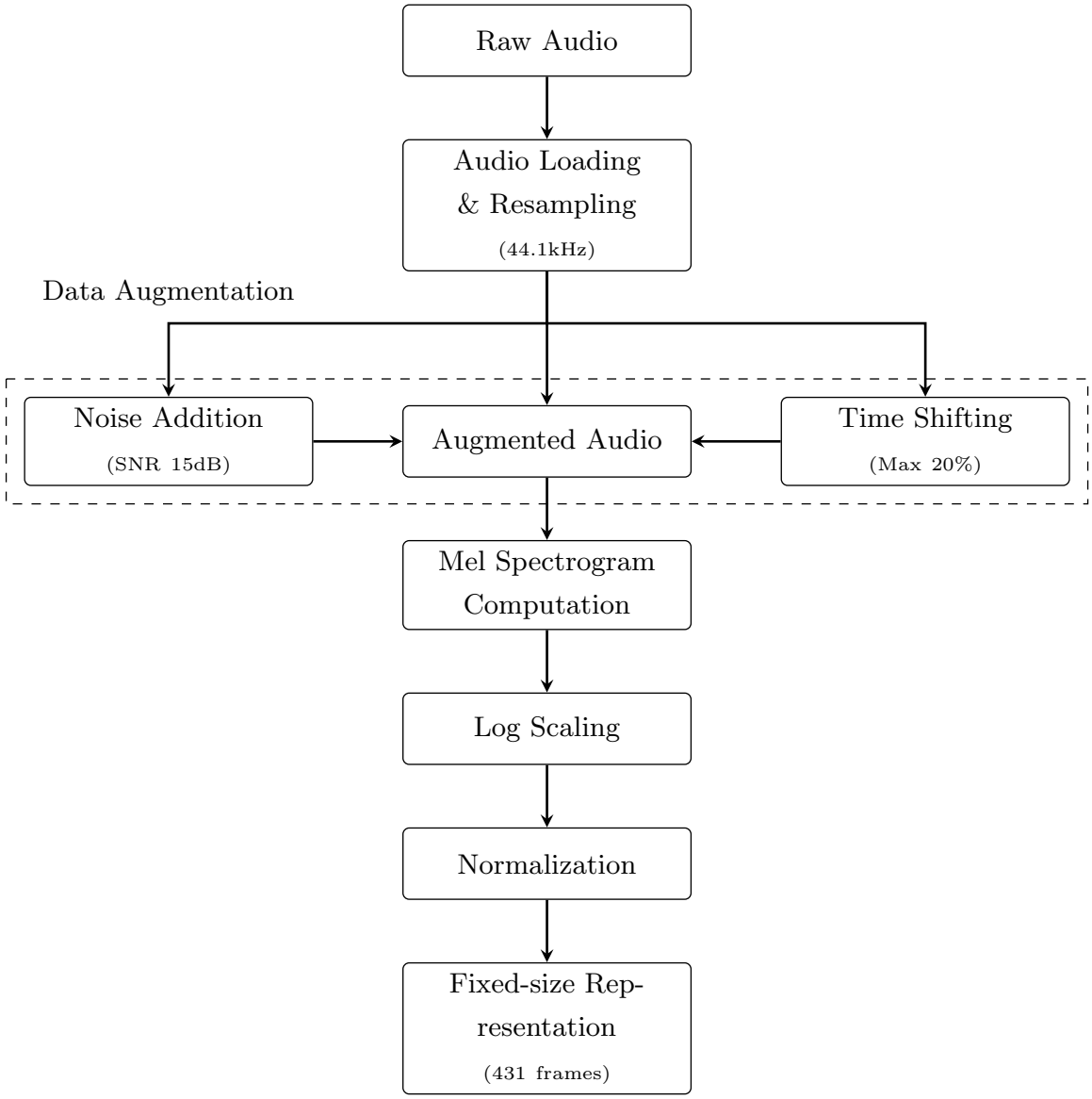


FIGURE 4.2: Audio preprocessing pipeline for feature extraction and augmentation

To prepare the dataset for neural network training, a pre-processing pipeline was implemented Fig 4.2, aimed at enhancing the diversity of the dataset using data augmentation techniques.

4.2.1 Audio-Loading and Resampling

The process begins with the loading of audio files and resampling it to a uniform rate of 44.1[kHz] using the librosa library, which ensures all audio sample are processed at the same temporal resolution, eliminating any chances of discrepancies.

4.2.2 Augmentation

In order to address the limited size of the dataset and improve the generalization capabilities of the neural net, two augmentation methods were implemented to introduce variability and to mimic real-world conditions.

Noise Addition Random noise is added to the audio signal to simulate the environmental interference.

The type of noise considered are:

- **White Noise**, characterized by a flat spectral density.
- **Pink Noise**, which exhibits a higher energy concentration at lower frequencies.
- **Brown Noise** with an steeper emphasis on lower frequencies.

The signal to Noise ration (SNR) is set to 15[dB], to ensure the noise is perceptible but does not dominate over the added noise.

Time Shifting To ensure temporal variability the audio signals are shifted along the time axis randomly, the shift is constrained to a maximum of 20% of the total audio length to ensure the temporal structure of the audio remains intact while providing a meaningful variation.

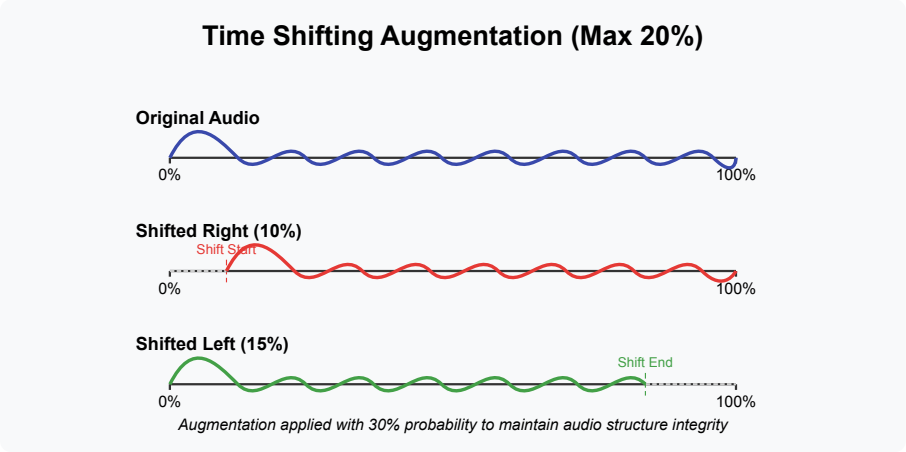


FIGURE 4.3: Time shifting Comparison

Additionally, these augmentations are applied independently,with a 30% probability, the audio was shifted in time by a random amount. The maximum shift amount was constrained to 20% of the total length of the audio signal. This simulates slight misalignments or shifts in the temporal characteristics of the sound, introducing variability without altering the nature of the sound.

Feature Extraction After augmentation and Time Shifting, the audio signals are converted into mel spectrograms, which is commonly used in audio signal processing. The mel-spectrogram effectively captures the perceptual characteristics of sound by mapping frequencies to the mel scale, which aligns more closely with human auditory perception. The spectrograms are computed using the short-time fourier transform (STFT) with the following parameters:

- Number of Mel Bands (n_mel) = 128,
- Hop Length = 512 samples,
- FFT Window size (n_fft) = 2048 samples,
- Frequency Range = 20[Hz] to half the sampling rate (22.05 [kHz])

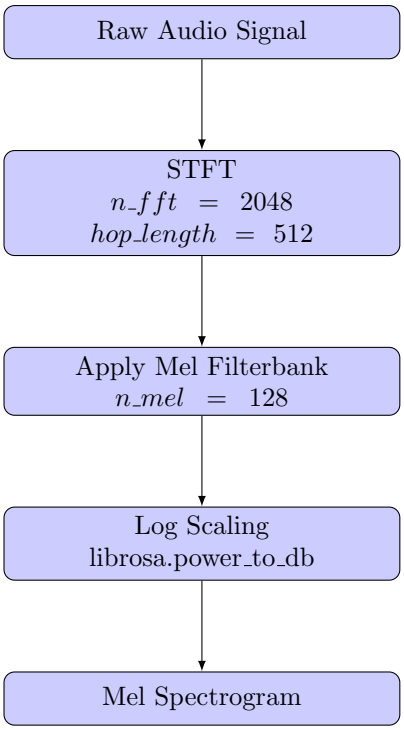


FIGURE 4.4: Audio Feature Extraction Pipeline

The resulting mel spectrograms are then converted to a logarithmic scale using the function `librosa.power_to_db`, which emphasizes the lower amplitude components and are often critical for recognizing subtle patterns in audio signals.

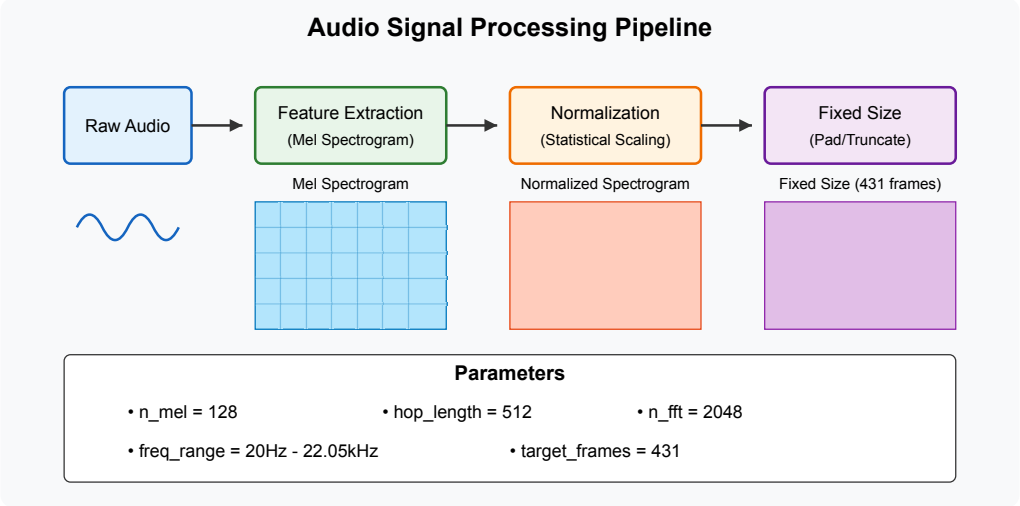


FIGURE 4.5: Spectrogram Padding and Truncation Process (Target: 431 Frames)

Normalization In order to reduce the effect of amplitude differences between between samples, the mel spectrograms were normalized by statistical scaling, each spectrogram was centered by subtracting the mean and scaled by dividing the standard deviation, with a small constant `1e-6` added to the denominator to avoid division by zero. That means all features are on the same scale, which makes the learining process more effective.

Statistical Normalization

$$X_{norm} = \frac{X-\mu}{\sigma+\varepsilon}$$

Where:

μ	= mean of spectrogram	X	= original spectrogram
σ	= standard deviation	ε	= 1e-6 (to avoid division by zero)

FIGURE 4.6: Normalization Formula for Mel Spectrograms

Fixed size representation Since audio in the chosen dataset has different durations in the range of 1-5 seconds, the corresponding mel spectrograms differ in a number of time frames. To make the input size of the neural network constant, the spectrograms are padded/truncated to have a target number of time frames (431). If a spectrogram is longer than the target length, it is truncated. Where necessary, shorter spectrograms are zero-padded with constant values that are equal to the minimum value of the spectrogram. The padding ensures no information is artificially introduced, so the integrity of the original signal is preserved.

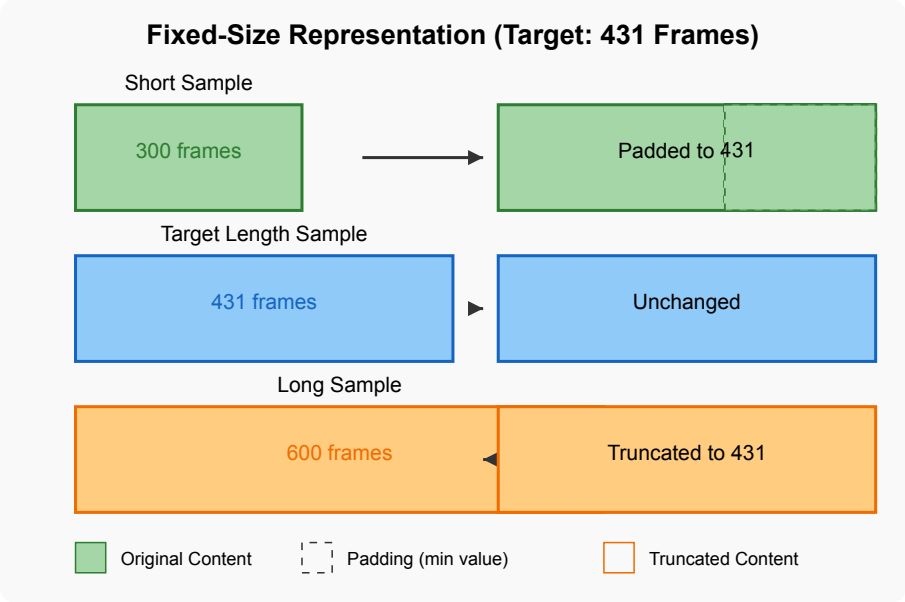


FIGURE 4.7: Spectrogram Padding and Truncation Process (Target: 431 Frames)

The above-mentioned preprocessing was selected to have maximum representational power in the dataset while maintaining compatibility with the neural network architecture. While techniques for data augmentation add robustness, simulating real variations in environmental conditions, the steps of feature extraction and normalization transform raw audio into compact and standardized representations. In this way, data preparation places the neural network at a better position to pick up meaningful patterns in the signals and generalize more appropriately to audio samples that it may never have seen before.

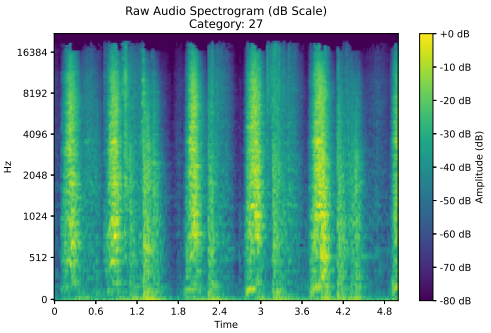


FIGURE 4.8: Mel Spectrogram of Raw Audio

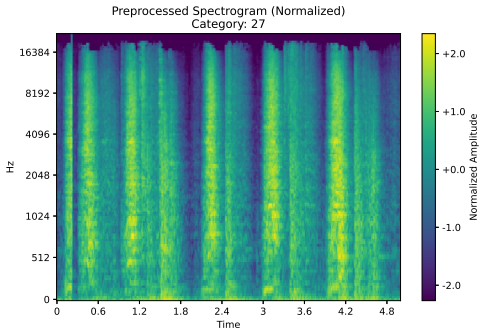


FIGURE 4.9: Preprocessed Mel Spectrogram

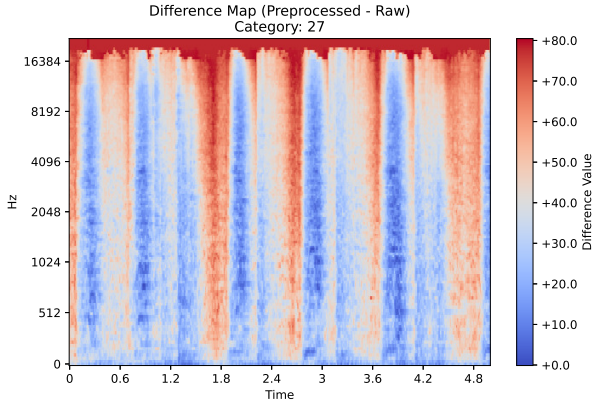


FIGURE 4.10: Difference Highlighted Between Raw and Preprocessed Spectrogram

Before & after pre-processing The transformation of an audio sample through preprocessing is illustrated in Figures 4.8, 4.9, and 4.10.

Figure 4.8 represents the **raw audio** converted into a **Mel spectrogram**, where frequency components are displayed over time using a logarithmic scale. Figure 4.9 shows the **preprocessed spectrogram**, which has undergone normalization and augmentation techniques such as noise addition and time shifting. Subtle differences in intensity and structure can be observed, reflecting the transformations applied to enhance robustness for machine learning applications.

Finally, Figure 4.10 visualizes the **difference map** between the raw and preprocessed spectrograms. The red regions indicate increased energy, while the blue regions show attenuation, highlighting spectral changes introduced by preprocessing.

4.3 Obtained Patterns

While the detailed results are shared in section **??**, In this section, we analyze the activation patterns of different models when processing the same input audio signals and examine how these models utilize discriminative filters to classify different sound categories. Understanding these patterns is essential to evaluating the role of each model's learned features in distinguishing between speech, music, noise, animal sounds, and machinery. Unlike the performance metrics discussed in a later section, this analysis focuses on the internal representations formed by each model, offering insights into how specific filters contribute to classification decisions.

To achieve this, we first present a visual comparison of activation patterns for the same set of inputs across three models: LinearNoBiasSoftplus, MelConstrainedLinear, and MelBasisTransform. This comparison allows us to assess whether different models focus on similar or distinct spectral features. Following this, we conduct a class-wise analysis of discriminative filters, illus-

trating which filters play a significant role in distinguishing between different audio classes in each model. By comparing these filter activation patterns, we aim to highlight differences in feature prioritization and classification strategies across models.

4.3.1 Model Activations for the Same Inputs

Each row in Figure 4.11 corresponds to a different audio sample, with:

- The leftmost column showing the raw input spectrogram.
- The three right columns displaying activations for each model:
  - **LinearNoBiasSoftplus**
  - **MelConstrainedLinear**
  - **MelBasisTransform**

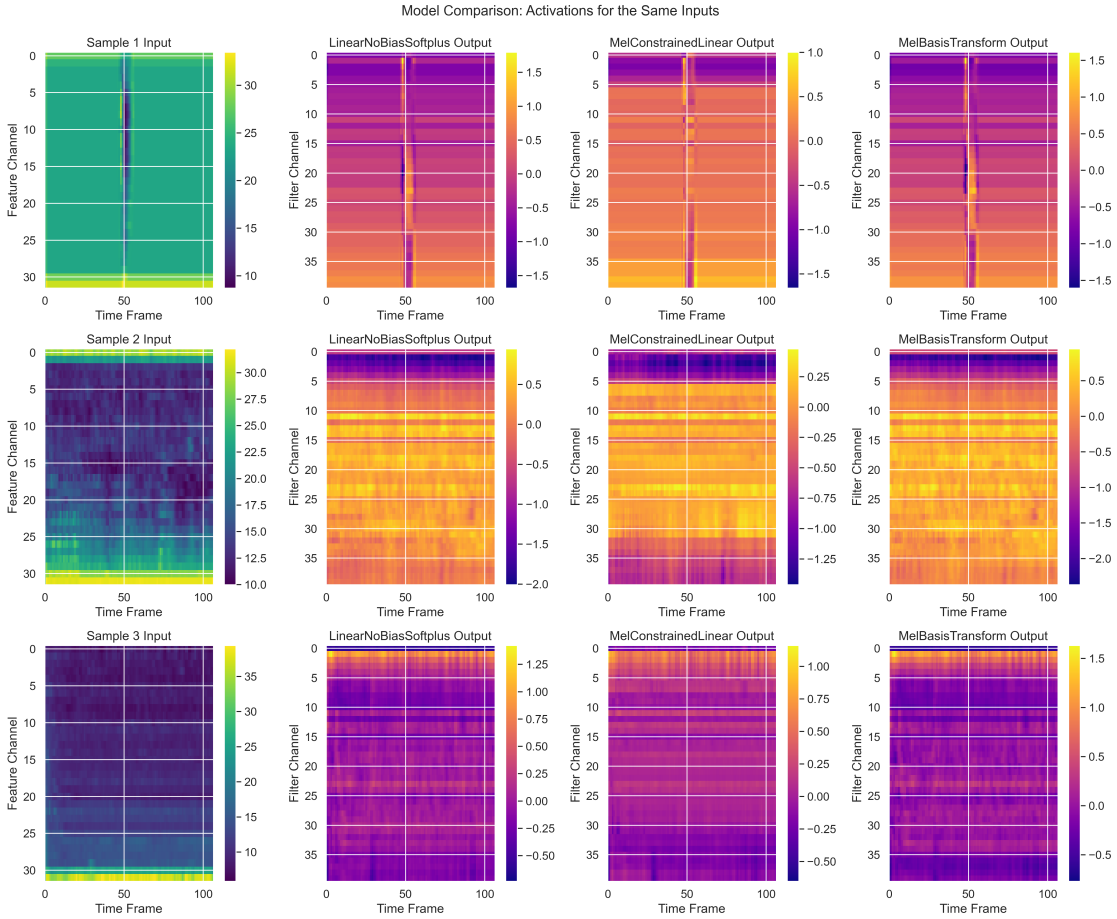


FIGURE 4.11: Model activations for different inputs.The first column shows the raw spectrogram, while the remaining three columns display activations from different models.

4.3.2 Observations on Learned Transformation Functions

To further examine how each model transforms the frequency domain, we visualize the learned mapping functions for three distinct approaches. In Figure 4.12, the linear model's transformation functions highlight how a simpler parameterization can still capture essential spectral cues. By

contrast, the melBasis approach, depicted in Figure 4.13, demonstrates a tighter alignment with the conventional mel scale, thereby emphasizing perceptually relevant frequency bands. Finally, Figure 4.14 illustrates the MelConstrained model, which introduces explicit constraints to preserve key aspects of the mel distribution while allowing some degree of learning flexibility. These visualizations serve as a comparative lens through which we can identify how each method prioritizes different portions of the frequency spectrum, ultimately shaping the downstream classification performance.

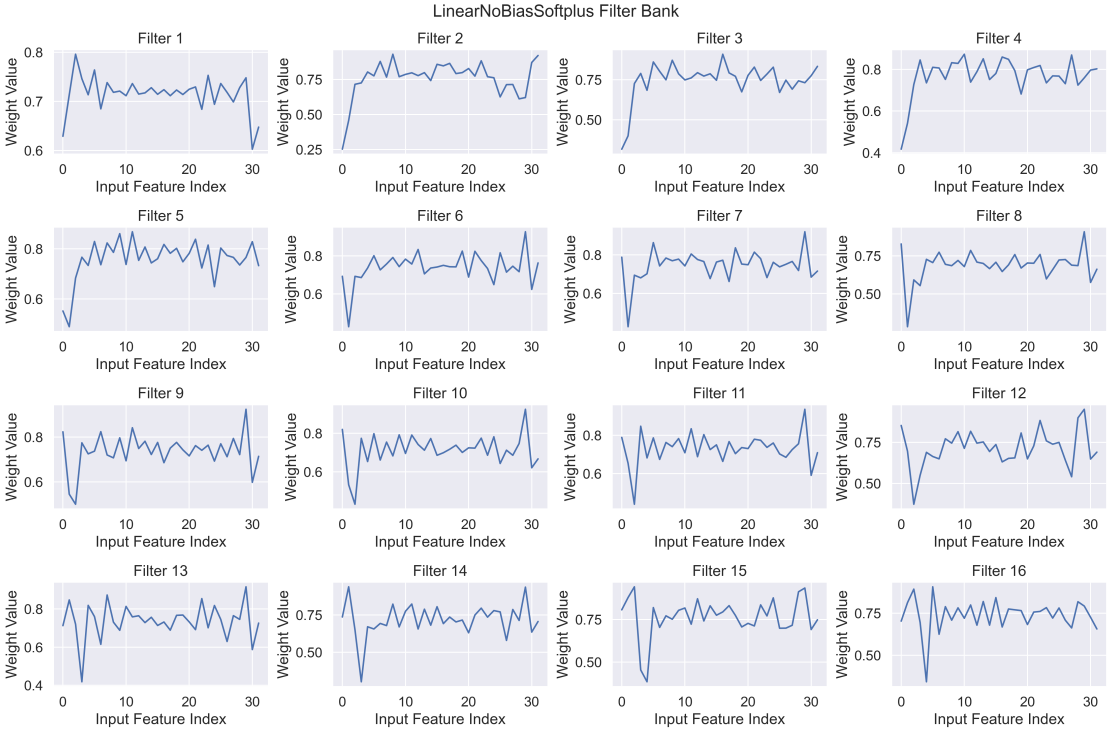


FIGURE 4.12: Visualization of learned transformation functions for linear approach

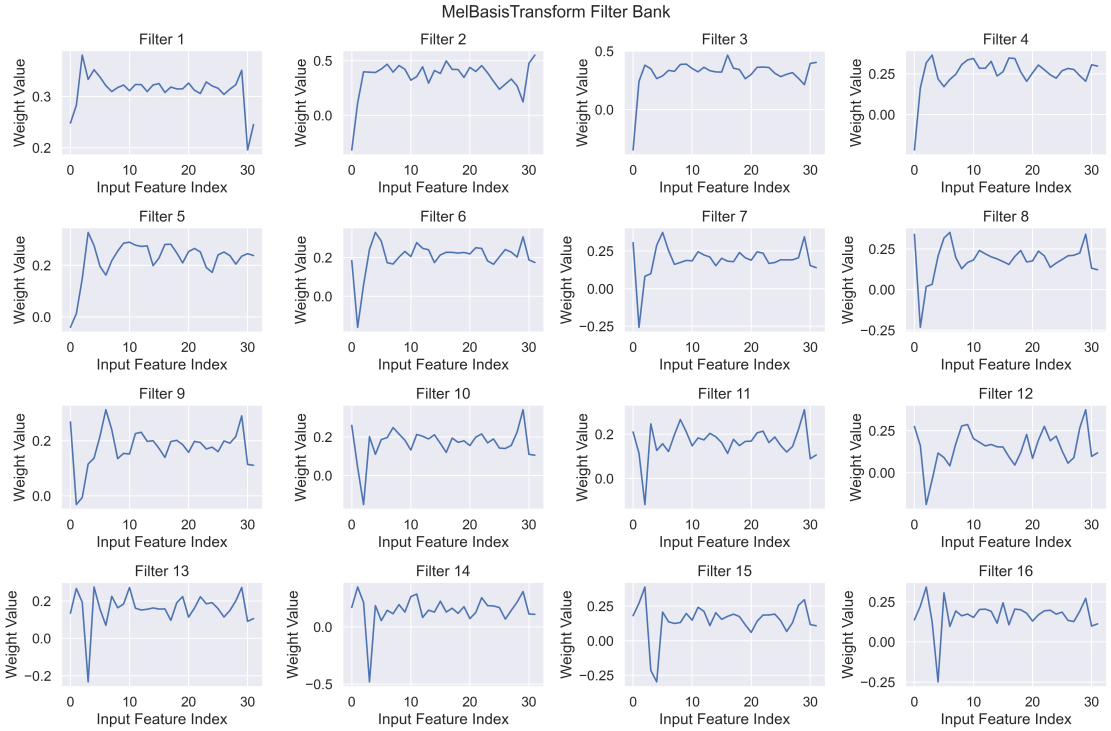


FIGURE 4.13: Visualization of learned transformation functions for melBasis approach

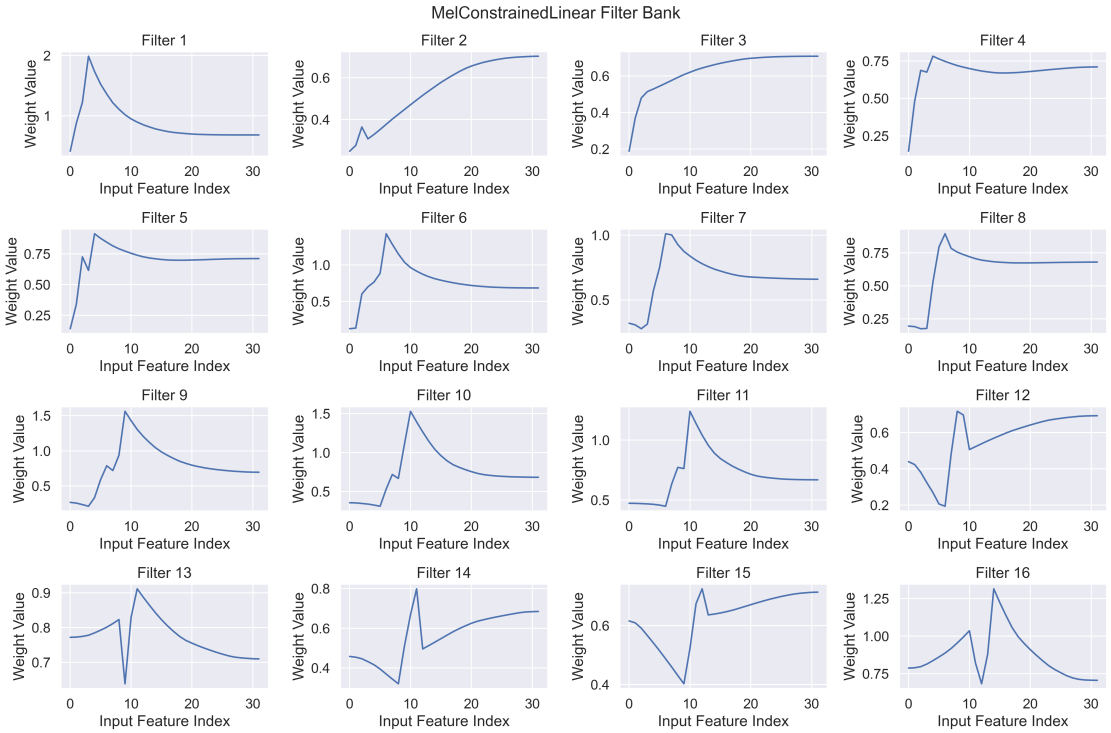


FIGURE 4.14: Visualization of learned transformation functions for MelConstrained approach

Each row corresponds to a single model, with subplots illustrating how specific frequency bins are mapped to output magnitudes. From the plots, several key observations emerge. First, the *MelBasisTransform* fig 4.13 model tends to allocate more energy to mid-to-high frequency ranges, potentially capturing salient harmonic structures. In contrast, the *LinearNoBiasSoftplus*

fig 4.12 approach appears to maintain a smoother, gradually increasing response, suggesting that its monotonic non-negative constraint encourages a more uniformly distributed emphasis across the frequency spectrum. Meanwhile, the *MelConstrainedLinear* fig 4.14 model exhibits behavior that closely aligns with the standard mel scale, focusing on perceptually relevant frequency bands.

These differences underscore how architectural design choices and activation constraints can significantly influence the learned representations. By examining these transformation curves, we can gain insight into why certain models might perform better in tasks requiring discrimination of subtle spectral nuances. Moreover, this visualization helps confirm that the learned transformations remain interpretable, revealing which parts of the frequency spectrum each model deems most important for downstream classification tasks.

4.3.3 Class analysis of the obtained patterns

In order to evaluate the patterns, several plots were considered in order to illustrate the discriminative filters for different audio classes across the models. Each model identifies specific filters that play a key role in distinguishing between the classes.

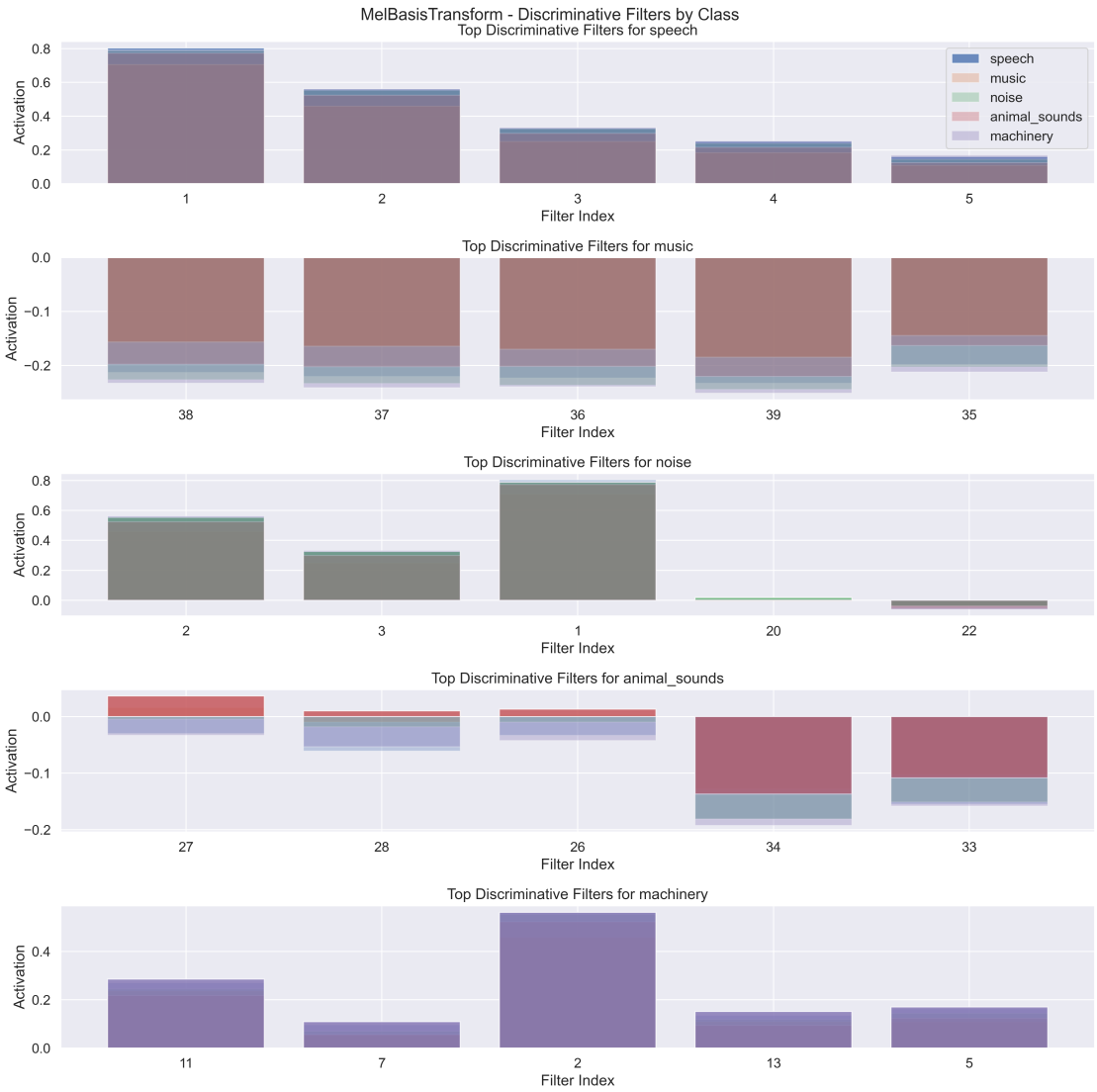


Figure 4.15: Discriminative Filters in the MelBasisTransform Model.

This plot displays the most important filters for classifying speech, music, noise, animal sounds, and machinery. Speech relies on filters 1, 2, and 3, while music is associated with higher-index filters with negative activation values.

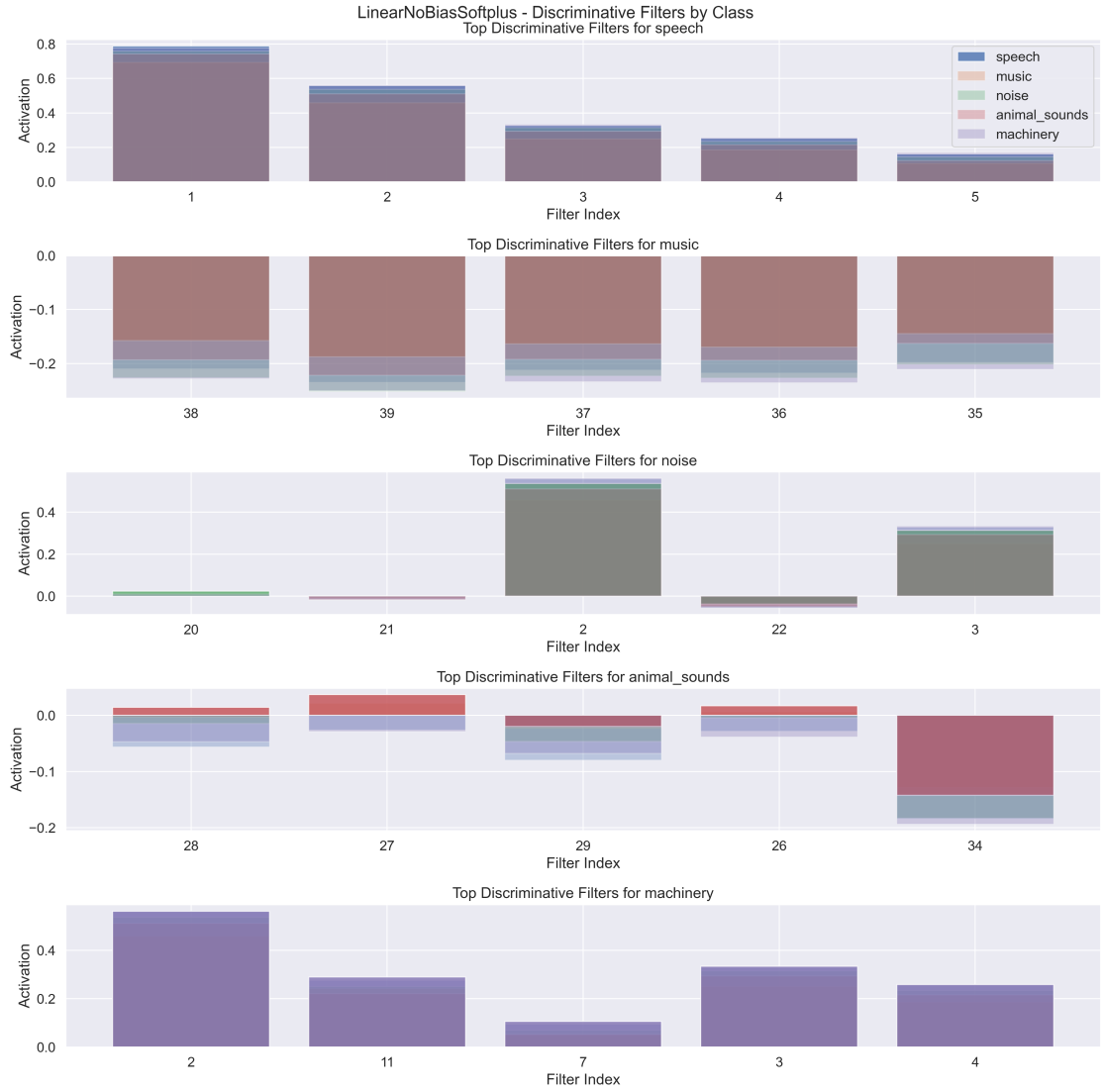


FIGURE 4.16: Discriminative Filters in the LinearNoBiasSoftplus Model.

This model follows a similar trend as MelBasisTransform but introduces subtle differences in noise and machinery classification. Noise is primarily identified using filters 20, 21, and 2, while machinery detection relies on filters 2, 11, and 7.

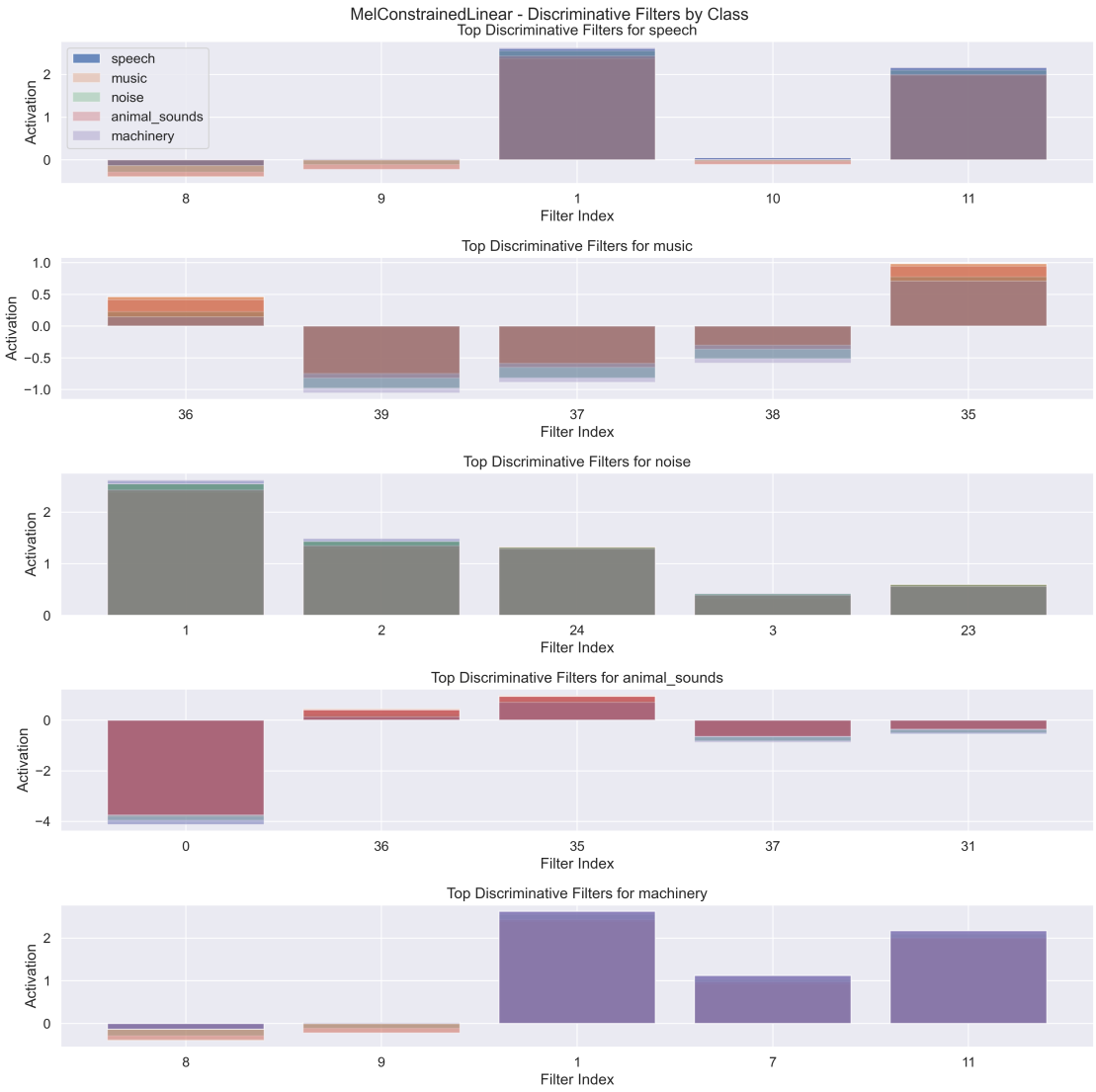


FIGURE 4.17: Discriminative Filters in the MelConstrainedLinear Model.

The MelConstrainedLinear model demonstrates the most significant shifts in filter prioritization. Speech is now associated with filters 8, 9, and 1, while noise classification exhibits the strongest activation levels. The model also introduces extreme negative activations for certain filters in animal sounds.

These comparison highlight the approach of different models to prioritize the the varying sets of filters to classify the respected audio signals effectively, maintaining some consistencies such as the low-index filters for speech and the reliance on high-index filters for music. The first two model in fig 4.15 & 4.16 show similarities. whereas the 4.17 shows the most deviation due to its nature.

4.3.4 Model Predictions vs. Target Spectrogram

The second set of plots (Figure 4.18) shows how each model reconstructs the target spectrogram:

- The top left plot displays the target spectrogram (ground truth).
- The rest three plots compare predictions from the three models.

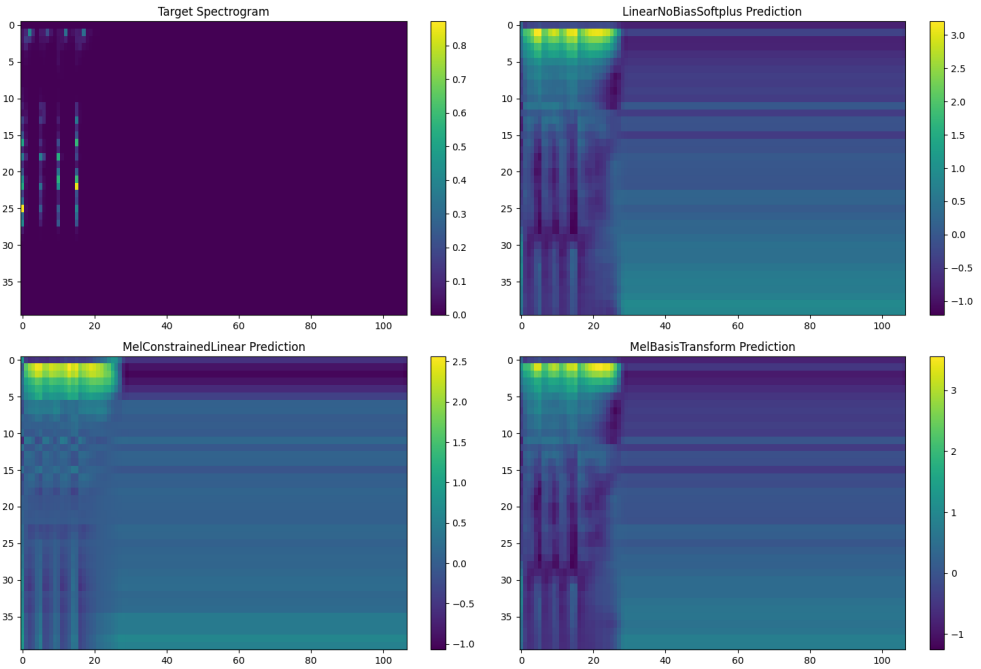


FIGURE 4.18: Model predictions compared to the target spectrogram. Each row corresponds to a different model's prediction.

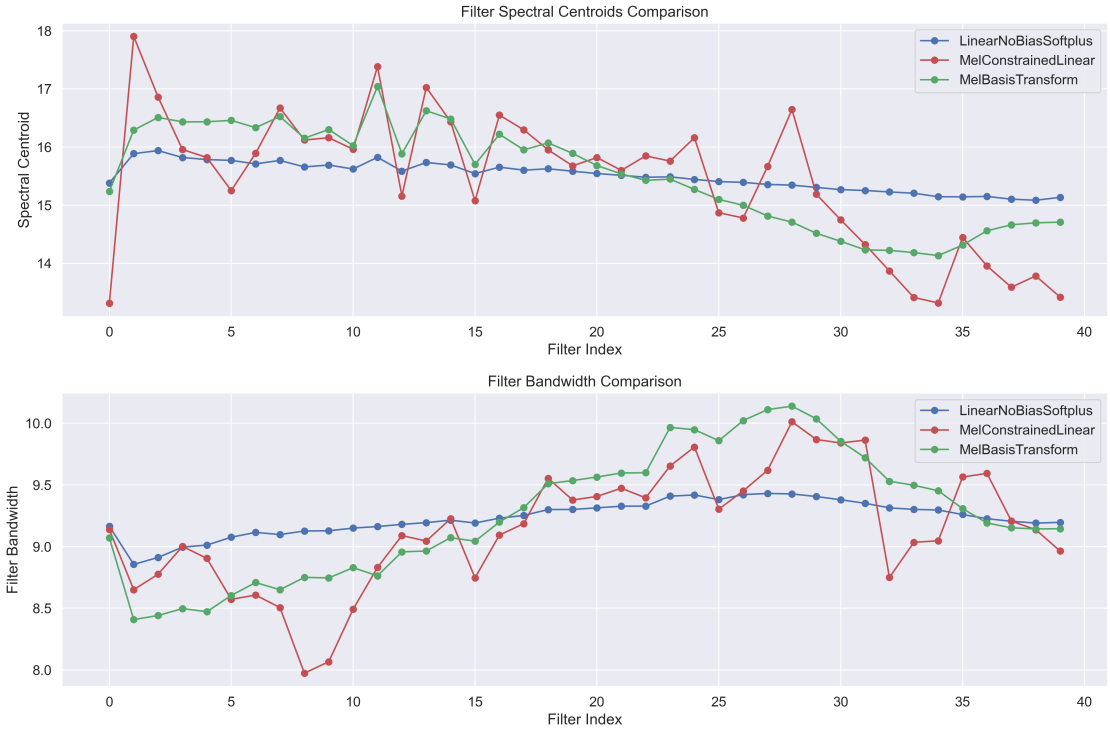


FIGURE 4.19: Spectral pattern Comparison of all the models

The analysis of the prediction outputs from the different models reveals distinct characteristics for each approach. The **LinearNoBiasSoftplus** model demonstrates a notable ability to capture high-energy regions within the audio signal, effectively highlighting the dominant spectral features. However, this flexibility comes at a cost. The lack of strong constraints in the model leads to the introduction of artifacts, resulting in outputs that are noticeably noisier. This observation

suggests that while the model can adapt to varying input conditions, the unconstrained nature of its transformation may inadvertently amplify irrelevant or spurious details.

In contrast, the **MelConstrainedLinear** model produces a much more structured transformation. Its predictions are characterized by smooth and coherent frequency representations that are closely aligned with the mel scale, reflecting the model's strict adherence to auditory perceptual principles. This rigorous constraint ensures that the output remains stable and well-organized, which is beneficial for tasks that require a clear and consistent spectral structure. However, the imposition of these strict mel constraints also means that some finer details of the signal can be lost in the transformation process, potentially omitting subtle yet significant auditory cues.

The **MelBasisTransform** model, on the other hand, appears to offer a middle ground between the two approaches. Its predictions exhibit qualities similar to those of the MelConstrainedLinear model, such as the maintenance of a structured spectral representation. At the same time, the MelBasisTransform model introduces a slight degree of adaptability, allowing it to fine-tune the output for improved accuracy. This balance between structured learning and flexibility suggests that the model is capable of retaining the benefits of the mel structure while also accommodating variations in the data that may be critical for precise audio interpretation.

Overall, the key takeaway from these observations is that each model exhibits unique strengths and trade-offs. The **LinearNoBiasSoftplus** model offers flexibility in capturing dynamic high-energy regions but suffers from the drawback of noise due to its less constrained nature. In comparison, the **MelConstrainedLinear** model provides highly structured outputs that align well with human auditory perception, although it may sacrifice some detail in the process. Meanwhile, the **MelBasisTransform** model strikes an effective balance between the two, offering a blend of structure and adaptability that makes it a compelling choice when fine-tuning for improved accuracy is necessary. Ultimately, the observations indicate that while the MelConstrainedLinear model aligns best with perceptual criteria, the MelBasisTransform model offers a valuable trade-off by incorporating adaptive features into a well-structured framework.

4.3.5 Visualization and Interpretation

While the current implementation focuses on feature extraction and storage, we recommend developing a complementary visualization strategy. An ideal approach would involve:

- Creating heatmap representations of the feature maps
- Generating dimensionality reduction visualizations (e.g., t-SNE or UMAP)
- Developing interactive exploration tools for the extracted features

4.4 Model Analysis and Architecture Efficiency of the Classifier Model

In this section, we present a comprehensive analysis of our proposed audio classification model architecture. The model comprises a sophisticated arrangement of convolutional layers augmented with squeeze-and-excitation (SE) attention mechanisms, demonstrating both high capacity and efficient feature extraction capabilities.

4.4.1 Parameter Distribution and Computational Complexity

The model contains 21.57M trainable parameters, distributed across multiple specialized components. Figure 4.20 illustrates this distribution, with the later convolutional blocks accounting for

the majority of parameters. Notably, ConvBlock5 alone comprises 65.6% (14.16M) of the total parameter count, while the SE attention mechanisms add only 2.4% overhead despite their significant contribution to model performance.

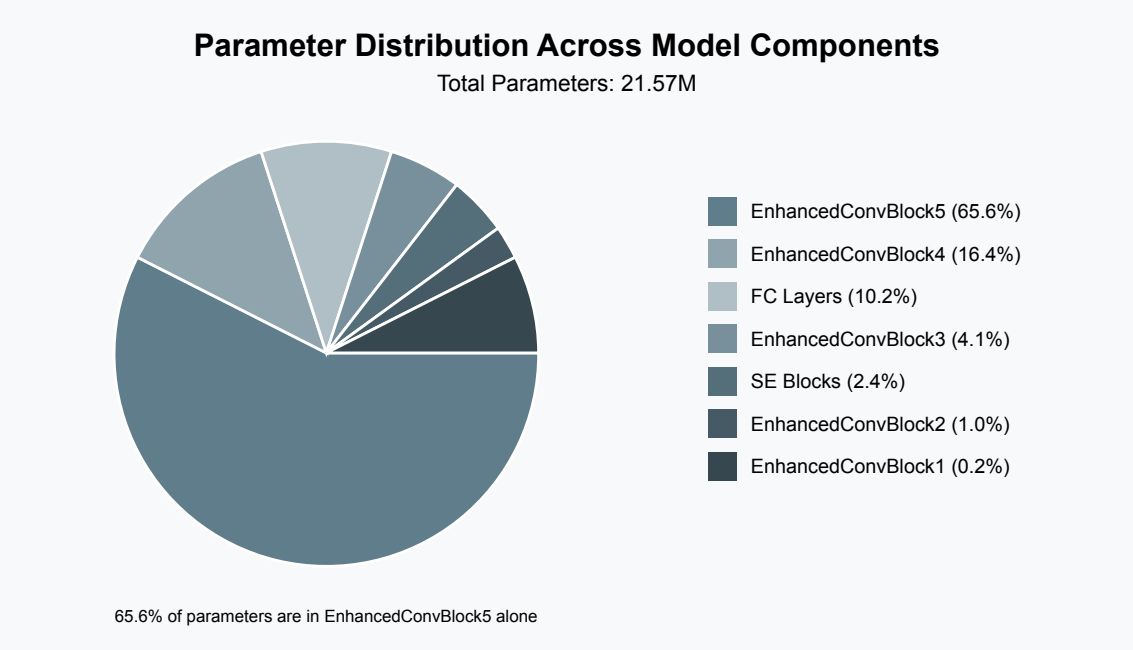


FIGURE 4.20: Parameter distribution showing the concentration of parameters in later convolutional blocks.

The computational complexity of the network is approximately 14.09 GFLOPs, which positions it between lightweight mobile architectures and heavy computational models. This computational profile achieves a favorable balance between expressiveness and efficiency, with an operations density of ∼0.65 GFLOPs per million parameters. When compared to established architectures like ResNet-50 (25.6M parameters, ∼4.1 GFLOPs) and MobileNetV2 (3.5M parameters, ∼0.3 GFLOPs), our model demonstrates a parameter efficiency of 431,398 parameters per class for the 50-class ESC-50 dataset classification task.

TABLE 4.2: Parameter and computational complexity across architectural components

Component	**Parameters**	**Percentage (%)**	**Cumulative (%)**
ConvBlock5	14,157,824	65.6	65.6
ConvBlock4	3,540,992	16.4	82.0
FC Layers (combined)	2,201,650	10.2	92.2
ConvBlock3	885,760	4.1	96.3
SE Blocks (all)	524,544	2.4	98.7
ConvBlock2	221,824	1.0	99.7
ConvBlock1	37,696	0.2	99.9
Total	**21,569,906**	**100.0**	

4.4.2 Information Flow Analysis

Our network implements a gradual dimensionality transformation strategy typical of convolutional neural networks, but with carefully tuned proportions for audio spectrogram analysis. Starting with input spectrograms of shape $[1, 128, 431]$ (representing a single-channel frequency-time representation), the network progressively reduces spatial dimensions while expanding channel capacity.

The experiments follows an approximate halving pattern through five convolutional blocks, resulting in a 99.9% reduction in spatial elements—from 55,168 elements in the input to just 52 elements after the final convolutional layer. Concurrently, the channel dimension expands exponentially from 1 to 1024, creating a compact but information-dense representation. This dual transformation exemplifies the network's capacity to distill essential acoustic features from high-dimensional inputs.

The theoretical receptive field of the network expands geometrically through the layers, reaching 127×127 at the final convolutional layer. This extensive receptive field allows the model to capture both local spectral patterns and global temporal relationships in the audio spectrograms, which is crucial for distinguishing between acoustically similar environmental sounds.

4.4.3 Attention Mechanism Implementation

A distinctive feature of our architecture is the incorporation of Squeeze and Ease (SE) attention blocks after convolutional stages 2 through 5. These blocks implement a dual-pooling strategy, combining both average and maximum pooling operations to capture complementary statistical information about channel activations. This approach differs from the original SE implementation [10], which relied solely on average pooling.

Each SE block employs a consistent reduction ratio of 8:1 for the bottleneck representation, balancing parameter efficiency with representational capacity. The progressive increase in parameter count across SE blocks (from 6,144 in SE1 to 393,216 in SE4) reflects the increasing channel dimensions they process. Despite their relatively modest parameter contribution (2.4% of total parameters), these attention mechanisms significantly enhance the network's discriminative capability by recalibrating channel-wise feature responses.

4.4.4 Results from the Audio Classifier Network

The proposed hook-based feature extraction methodology transcends traditional spectral analysis techniques. By leveraging the learned representations of a deep neural network, we move beyond linear decomposition methods, capturing complex, non-linear relationships inherent in audio signals. This approach provides a more nuanced and adaptive method of understanding spectral patterns, with potential applications ranging from environmental sound classification to advanced audio signal processing.

Which gives us the following feature maps for different layers with respect to the input mel-spectrogram:

- X-Axis (Spatial Dimensions of the Feature Map): Generally represents the width (or time dimension in the context of audio spectrograms).
- Y-Axis (Spatial or Frequency Dimension): Represents the height (or frequency dimension) of the feature maps.
- Color Scale/Intensity: Visualizes the activation intensities, where different colors or shades indicate the level of activation. These figures help demonstrate how the network progressively extracts and refines features through various layers.

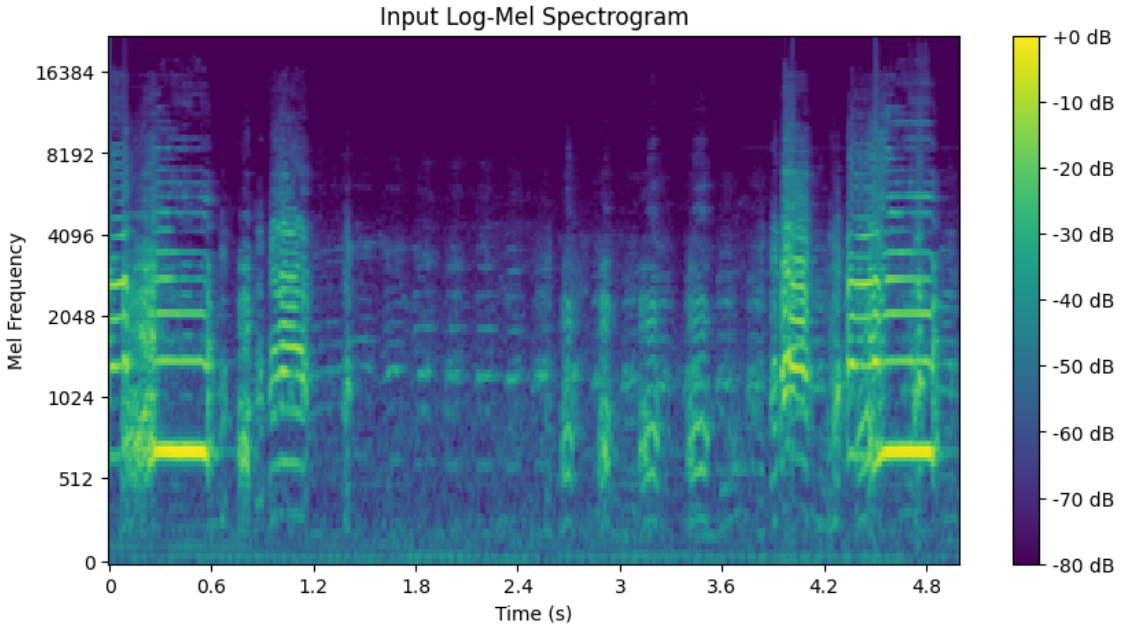


FIGURE 4.21: Input mel- spectrogram

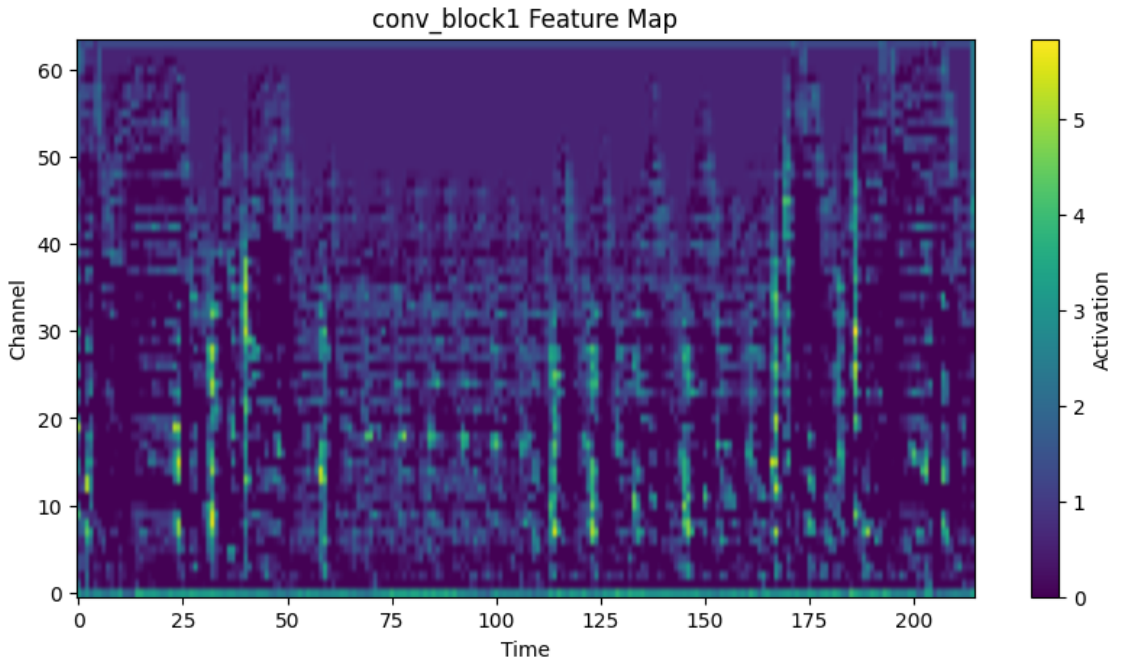


FIGURE 4.22: Extracted feature maps from Convolution Block 1

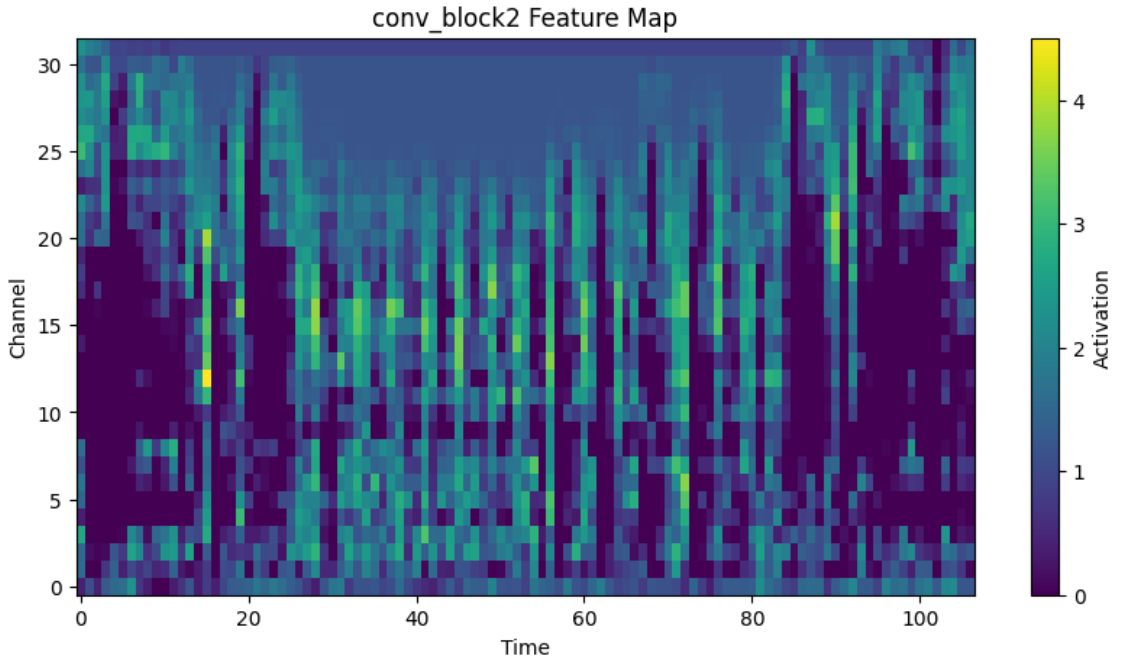


FIGURE 4.23: Extracted feature maps from Convolution Block 2

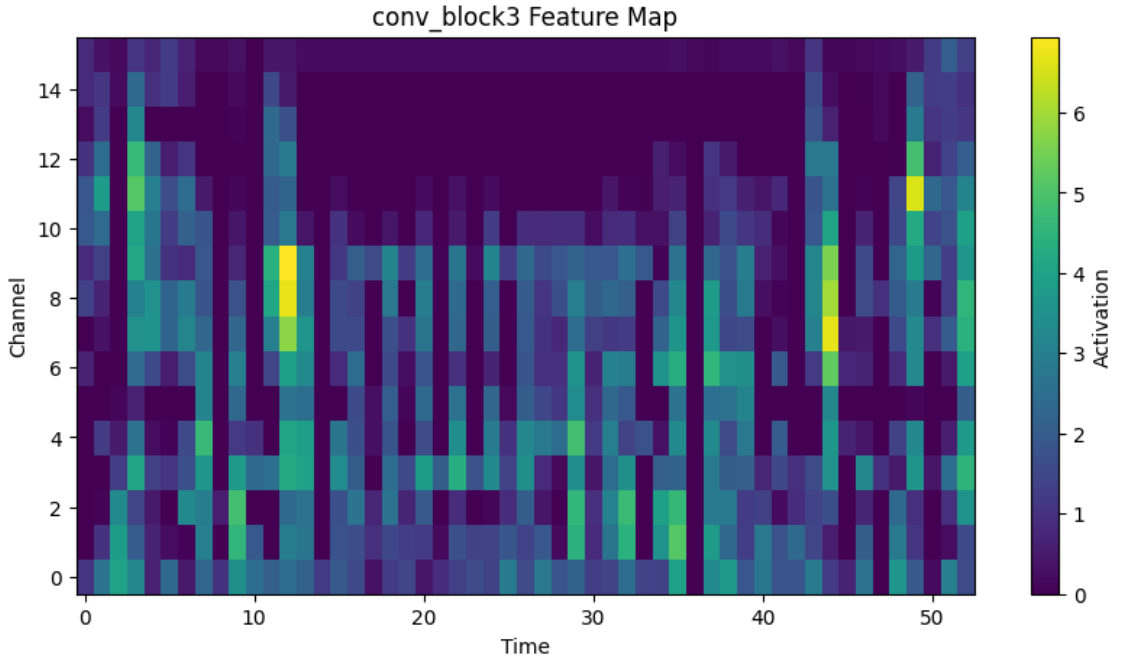


FIGURE 4.24: Extracted feature maps from Convolution Block 3

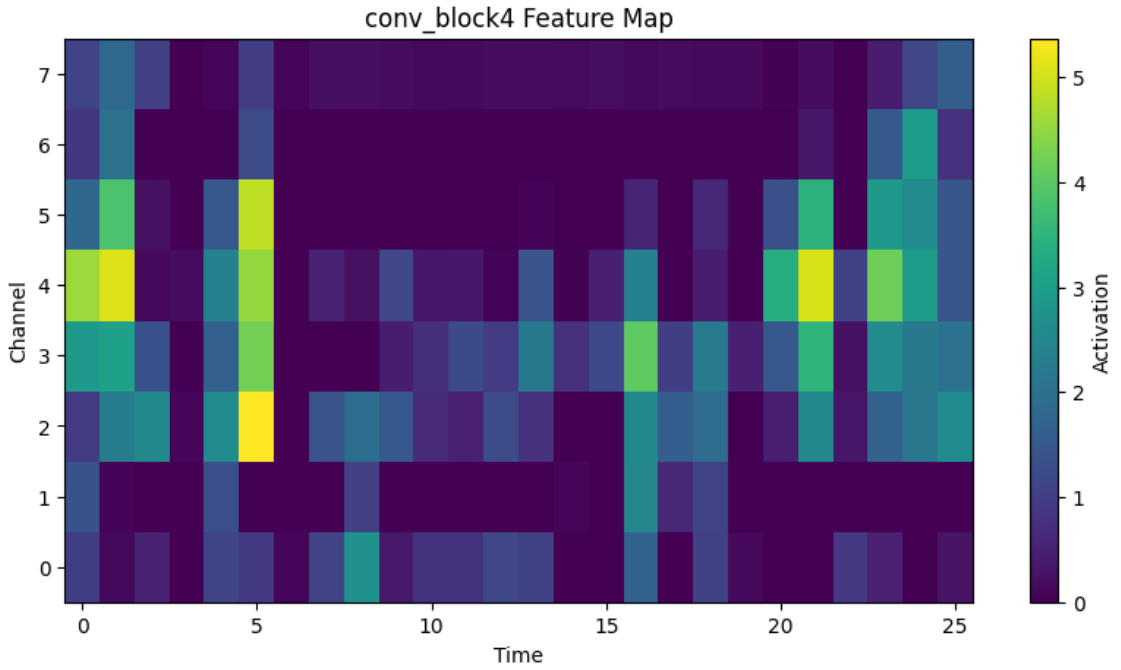


FIGURE 4.25: Extracted feature maps from Convolution Block 4

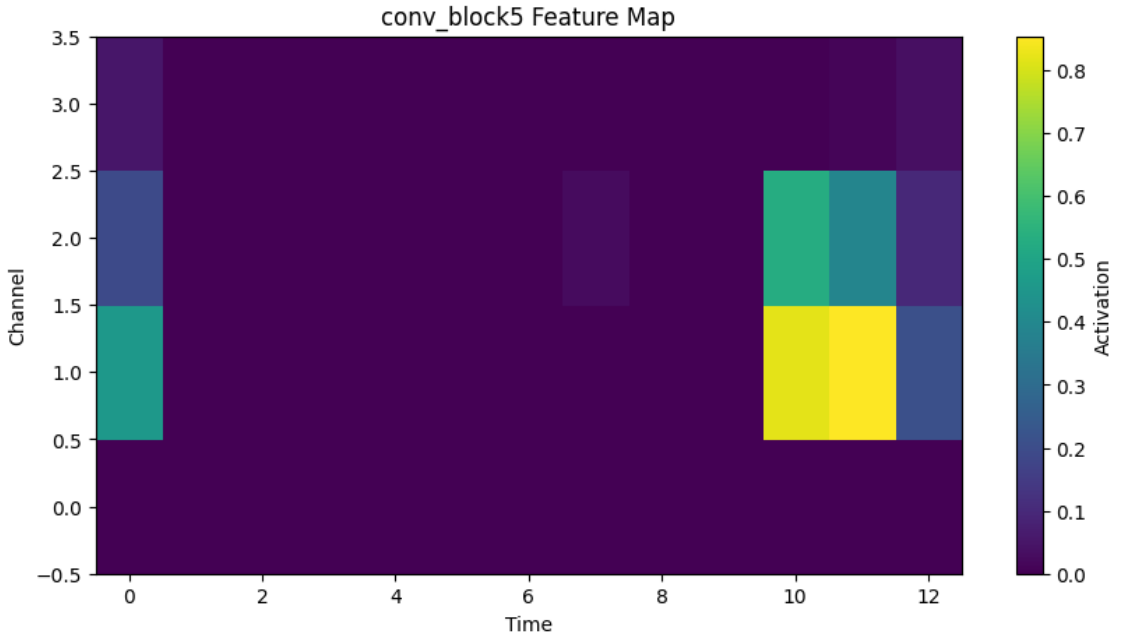


FIGURE 4.26: Extracted feature maps from Convolution Block 5

4.4.5 Significance of the SE blocks

For the SE (Squeeze-and-Excitation) blocks, the visualizations focuses on how the attention mechanism recalibrates channel-wise feature responses, often shown by distinct patterns in the activation maps.

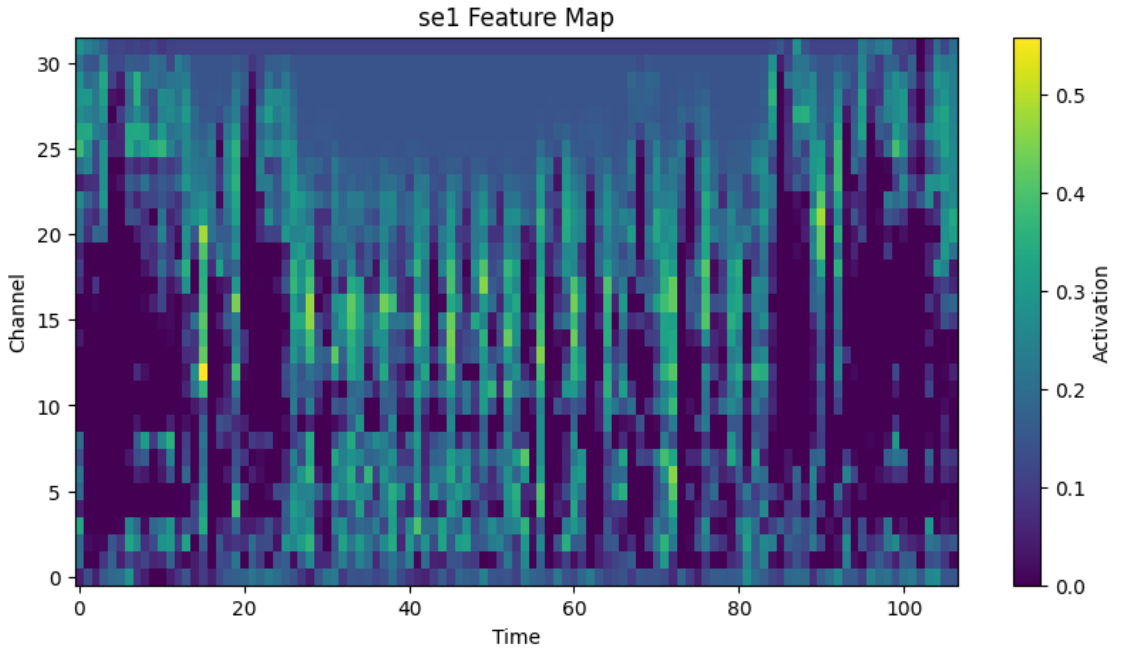


FIGURE 4.27: Extracted feature maps from SE 1 blocks of the neural network

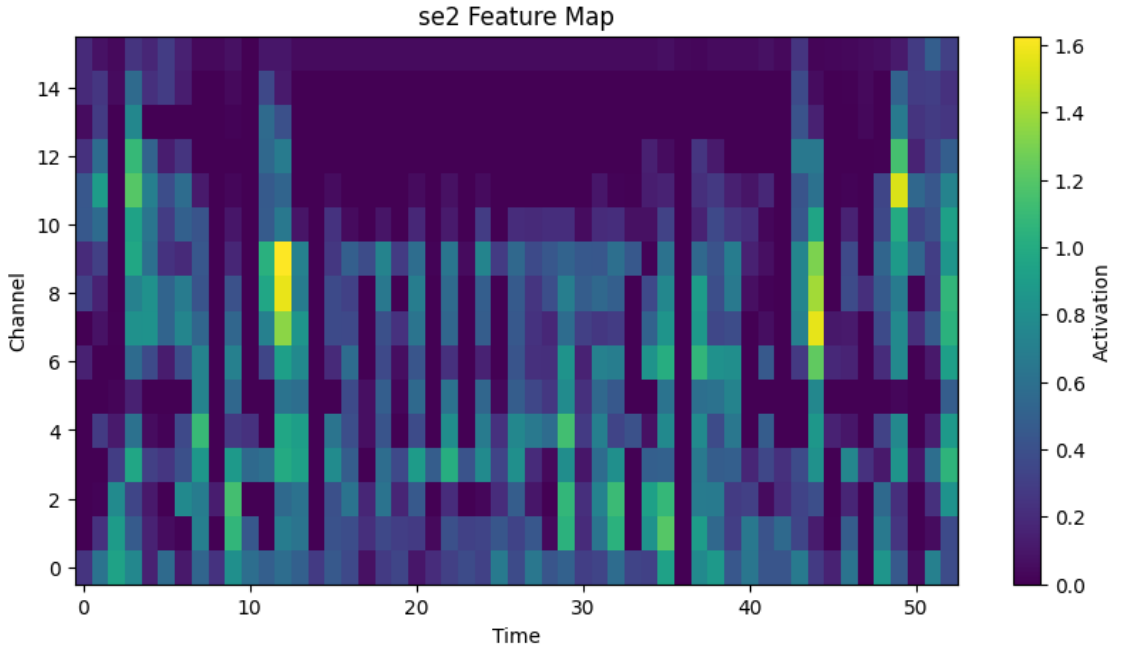


FIGURE 4.28: Extracted feature maps from SE 2 blocks of the neural network

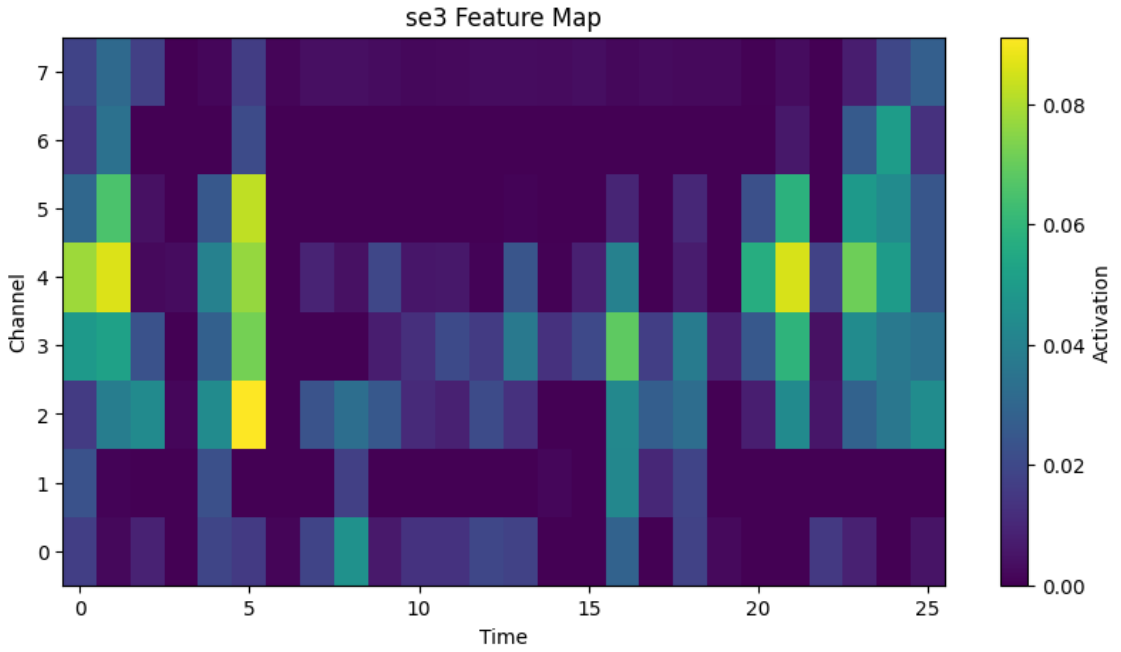


FIGURE 4.29: Extracted feature maps from SE 3 blocks of the neural network

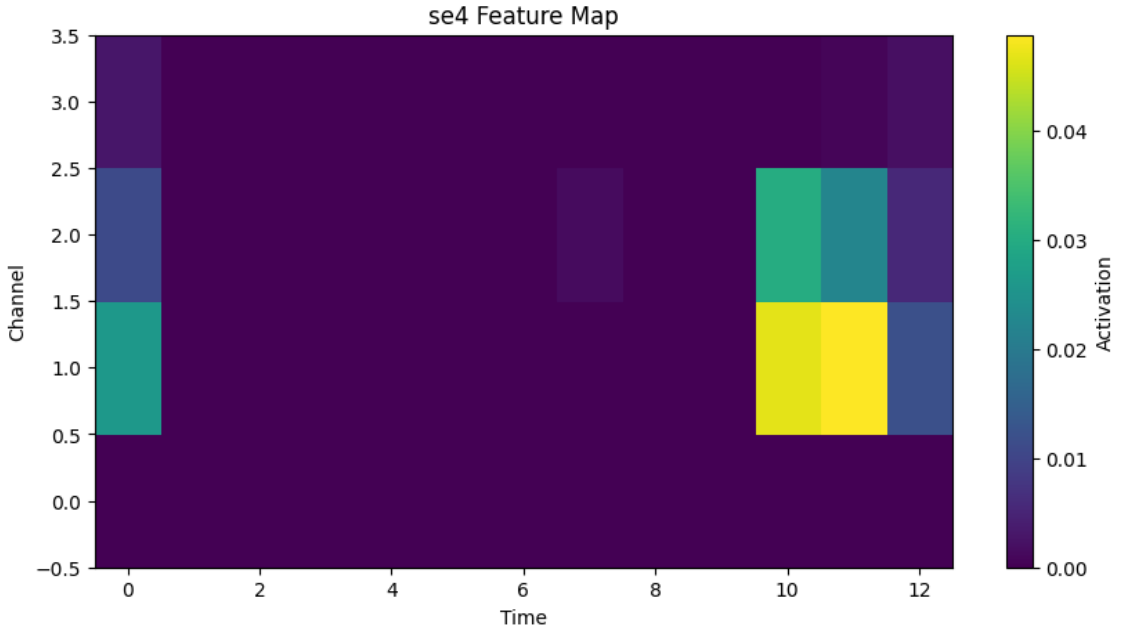


FIGURE 4.30: Extracted feature maps from SE 4 blocks of the neural network

4.5 Memory Footprint and Computational Graph Analysis

The memory footprint and computational graph analysis were conducted using the PyTorch framework, which provides tools to inspect the memory usage and performance of a model. In this section, we describe how the reported memory usage and the memory-to-computation ratio were calculated.

4.5.1 Memory Footprint Breakdown

The memory footprint of the model was measured during training, with the results presented as follows:

- **Total memory usage:** 304.64 MB
- **Memory allocated to the forward/backward computational graph:** 218.14 MB (71.6% of the total memory)
- **Memory allocated to parameters:** 86.28 MB (28.3% of the total memory)

The memory allocated to the computational graph is used for storing activations, gradients, and intermediate results during the forward and backward passes. This represents the majority of the memory usage, as the network has many intermediate layers that require storage for their output. On the other hand, the memory allocated to parameters represents the weights and biases of the model, which are learned during training.

4.5.2 Memory-to-Computational Efficiency

To assess the memory-to-computation ratio, the following metric was calculated:

$$\text{Memory-to-computation ratio} = \frac{\text{Memory allocated to the computational graph}}{\text{GFLOPs}}$$

The total memory allocated to the forward/backward computational graph is 218.14 MB, and the corresponding computational workload (measured in GFLOPs) was estimated to be 10.09 GFLOPs. This leads to the following calculation for the memory-to-computation ratio:

$$\text{Memory-to-computation ratio} = \frac{218.14\,\text{MB}}{10.09\,\text{GFLOPs}} = 21.62\,\text{MB per GFLOP}$$

This ratio indicates that the architecture is moderately memory-intensive, as it requires 21.62 MB of memory for each GFLOP of computation. This is expected given the high number of parameters and large feature map dimensions present in the model.

4.5.3 Summary of Results

The following conclusions can be drawn from the analysis:

- The model exhibits a relatively high memory-to-computation ratio of 21.62 MB per GFLOP, reflecting the balance between computational demands and memory usage in the model.
- The majority of memory (71.6%) is allocated to the forward/backward computational graph, highlighting the need for efficient memory management during training.
- A smaller portion of memory (28.3%) is dedicated to storing model parameters, which is typical for deep neural networks with a large number of weights and biases.

The peak memory usage occurs during the forward pass through the earlier convolutional blocks, where feature maps retain larger spatial dimensions despite channel expansion. This memory profile suggests that training efficiency could be improved through gradient checkpointing techniques or mixed precision training, particularly when scaling to larger batch sizes.

4.5.4 Architectural Insights and Optimization Potential

Several observations emerge from our analysis that inform potential optimization strategies:

- The parameter distribution reveals a significant concentration in later convolutional layers, suggesting these as primary targets for pruning or factorization techniques if model compression is desired.
- The relatively high parameter-to-class ratio (431,398 parameters per class) indicates potential overfitting risks, which could be mitigated through stronger regularization strategies or knowledge distillation.
- The SE attention mechanisms provide substantial benefit with minimal parameter overhead, suggesting that similar efficiency-focused attention mechanisms could be valuable additions to other parts of the network.
- The memory profile during training is dominated by activations rather than parameters, indicating that memory-efficient training techniques would be more effective than parameter quantization for improving training efficiency.

In conclusion, our architectural analysis reveals a model with significant capacity and a thoughtful balance between feature extraction depth and computational efficiency. The progressive spatial reduction coupled with channel expansion effectively distills complex audio spectrograms into discriminative representations, while the selective application of attention mechanisms enhances feature quality at critical stages in the network.

4.6 Training Details

- **Loss Function:** Categorical Cross-Entropy, suitable for multi-class classification problems. It ensures the model learns to differentiate between distinct sound categories.
- **Optimizer:** Adam optimizer with an initial learning rate of 0.001, chosen for its adaptive learning rate mechanism that enhances convergence. Adam dynamically adjusts the learning rate based on first-order and second-order moment estimates, making it effective for deep networks.
- **Batch Size:** Set to 32, balancing efficient training while maintaining stable gradient updates. A larger batch size would require more memory, while a smaller batch size might result in unstable gradients.
- **Epochs:** 300 epochs to allow sufficient learning cycles for model convergence. The number of epochs was chosen based on empirical analysis, ensuring the network had ample time to learn complex patterns.
- **Learning Rate Decay:** The learning rate was reduced on validation loss plateaus to fine-tune performance. This helped refine the model by gradually decreasing updates as training progressed.
- **Early Stopping:** Training was monitored using validation accuracy, with early stopping implemented to prevent unnecessary overfitting. If validation loss stopped improving, training was halted to avoid excessive parameter updates that could lead to overfitting.

- **Data Mixup:** To improve generalization and prevent overfitting, mixup augmentation was applied during training. Mixup involves combining two different audio samples by linearly interpolating both their mel spectrogram representations and their labels. This technique helps the model learn smoother decision boundaries, making it more robust to variations in environmental sounds. By training on blended audio representations, the network is encouraged to focus on essential features rather than noise or dataset-specific artifacts, ultimately leading to improved classification performance.

4.7 Hardware and Performance

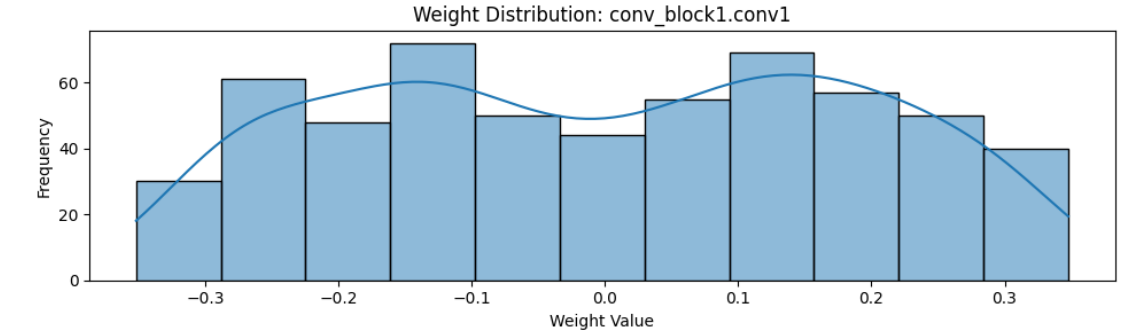
- **Training Environment:** The model was trained on Google Colab using a Tesla T4 GPU, leveraging accelerated hardware for efficient processing. Training on the free-tier runtime ensured accessibility while still achieving robust performance. Despite hardware limitations, the model was optimized to make the most of the available computational resources.
- **Final Model Performance:** The best validation accuracy achieved was **88.5%**, demonstrating strong classification capability on the ESC-50 dataset. This result suggests that the model is capable of distinguishing between different environmental sound classes with high reliability.
- **Training Time:** Each epoch took approximately 21 seconds, making the training process manageable within the available computational resources. Over 300 epochs, this resulted in a total training duration of around **105 minutes**.
- **Generalization Ability:** The model's performance on validation data indicates strong generalization. The use of dropout, batch normalization, and data augmentation contributed to reducing overfitting, ensuring robust classification on unseen test data.

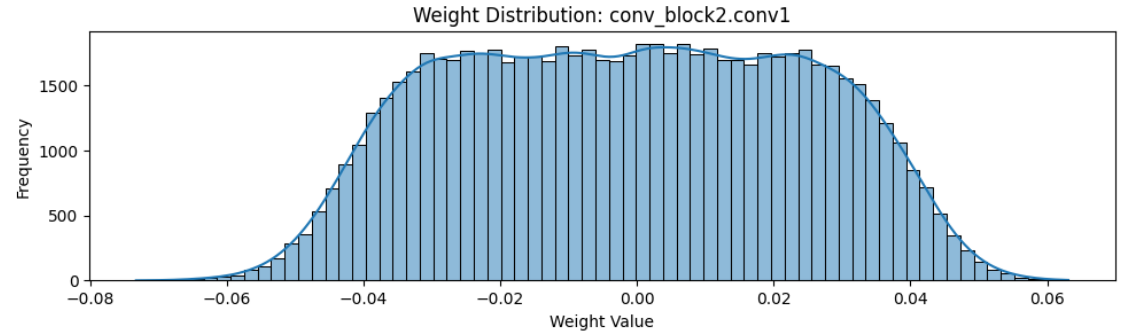
4.8 Performance Analysis

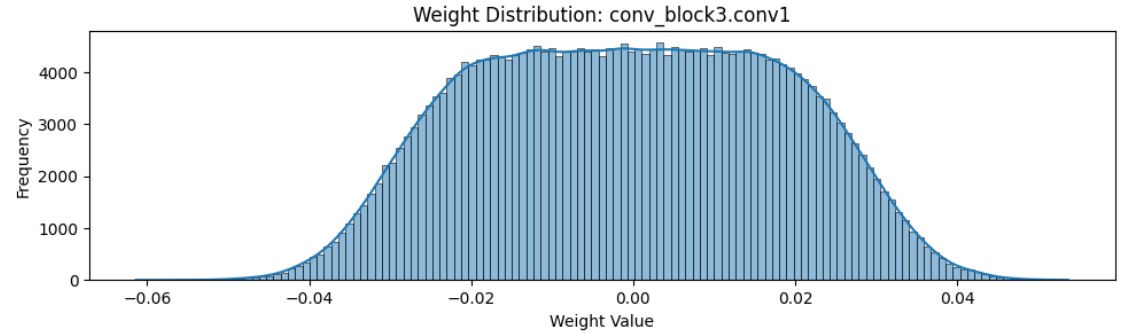
4.8.1 Weight Distribution

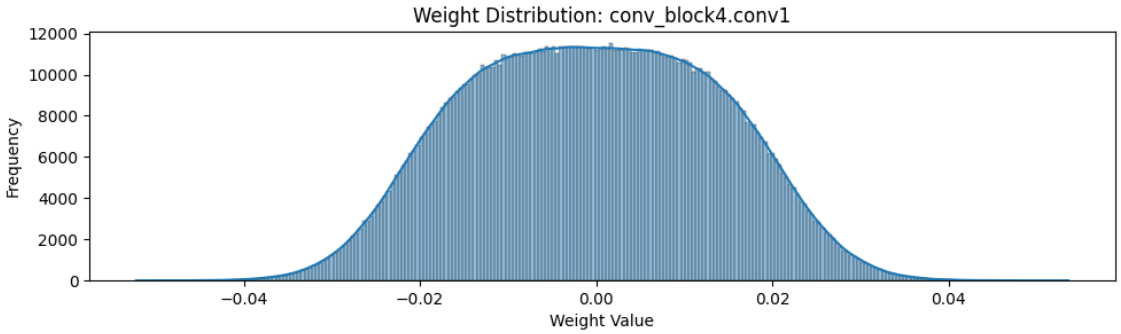
The weight distribution histograms reveal critical insights into the model's weight initialization and learning dynamics across different convolutional layers. Notably, the distributions exhibit a Gaussian-like profile, characteristic of well-normalized neural network weights.

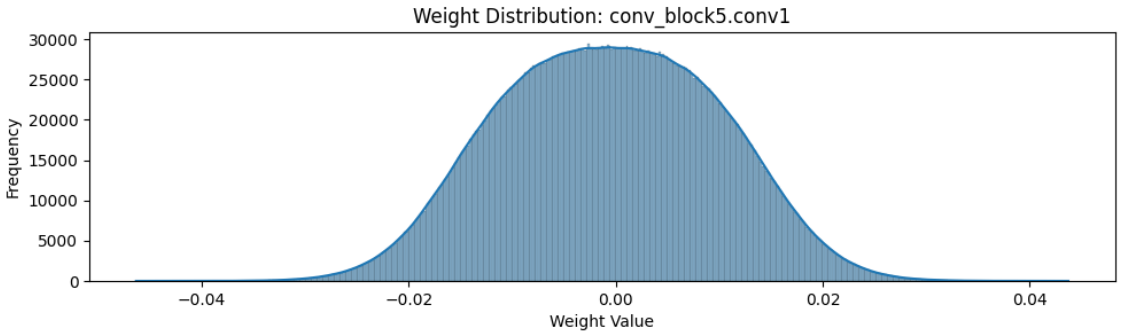
The first convolutional layer (conv_block1.conv1) in fig 4.31 demonstrates a broader, more irregular distribution compared to deeper layers, spanning approximately -0.3 to 0.3. In contrast, subsequent layers (conv_block2 through conv_block5) display progressively more concentrated, symmetrical distributions centered near zero, with reduced variance. This progression suggests effective weight normalization techniques and potentially indicates the model's ability to learn increasingly refined feature representations through deeper network stages. The fully connected layers (fc1 and fc2) maintain similar Gaussian distributions, further corroborating the model's robust weight initialization and training strategy.

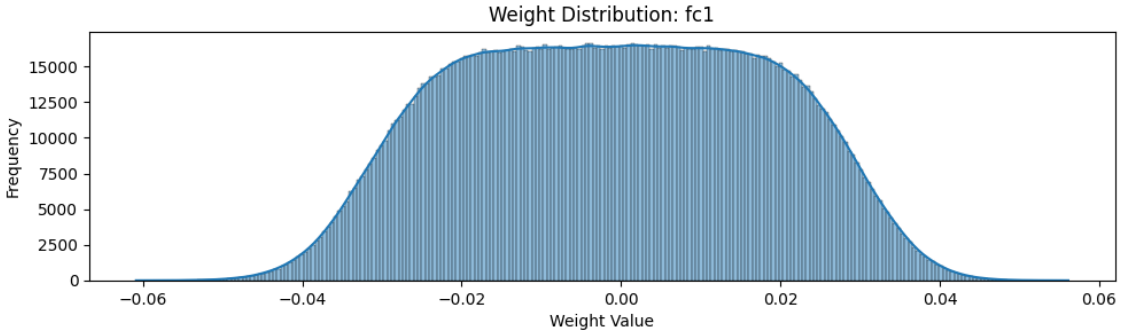

Weight Distribution: conv_block1.conv1
Frequency
Weight Value


Weight Distribution: conv_block2.conv1
Frequency
Weight Value


Weight Distribution: conv_block3.conv1
Frequency
Weight Value


Weight Distribution: conv_block4.conv1
Frequency
Weight Value


Weight Distribution: conv_block5.conv1
Frequency
Weight Value


Weight Distribution: fc1
Frequency
Weight Value

4.8.2 Confusion Matrix

The confusion matrix in the fig 4.32 provides a comprehensive visualization of the model's classification performance across 50 distinct audio categories. The predominant dark blue diagonal elements indicate high classification accuracy for most classes, suggesting robust feature extraction and discriminative capabilities.

However, subtle off-diagonal variations reveal nuanced inter-class confusion patterns. Some adjacent or semantically similar sound categories exhibit marginally higher misclassification rates, which could be attributed to acoustic similarities or overlapping spectral characteristics.

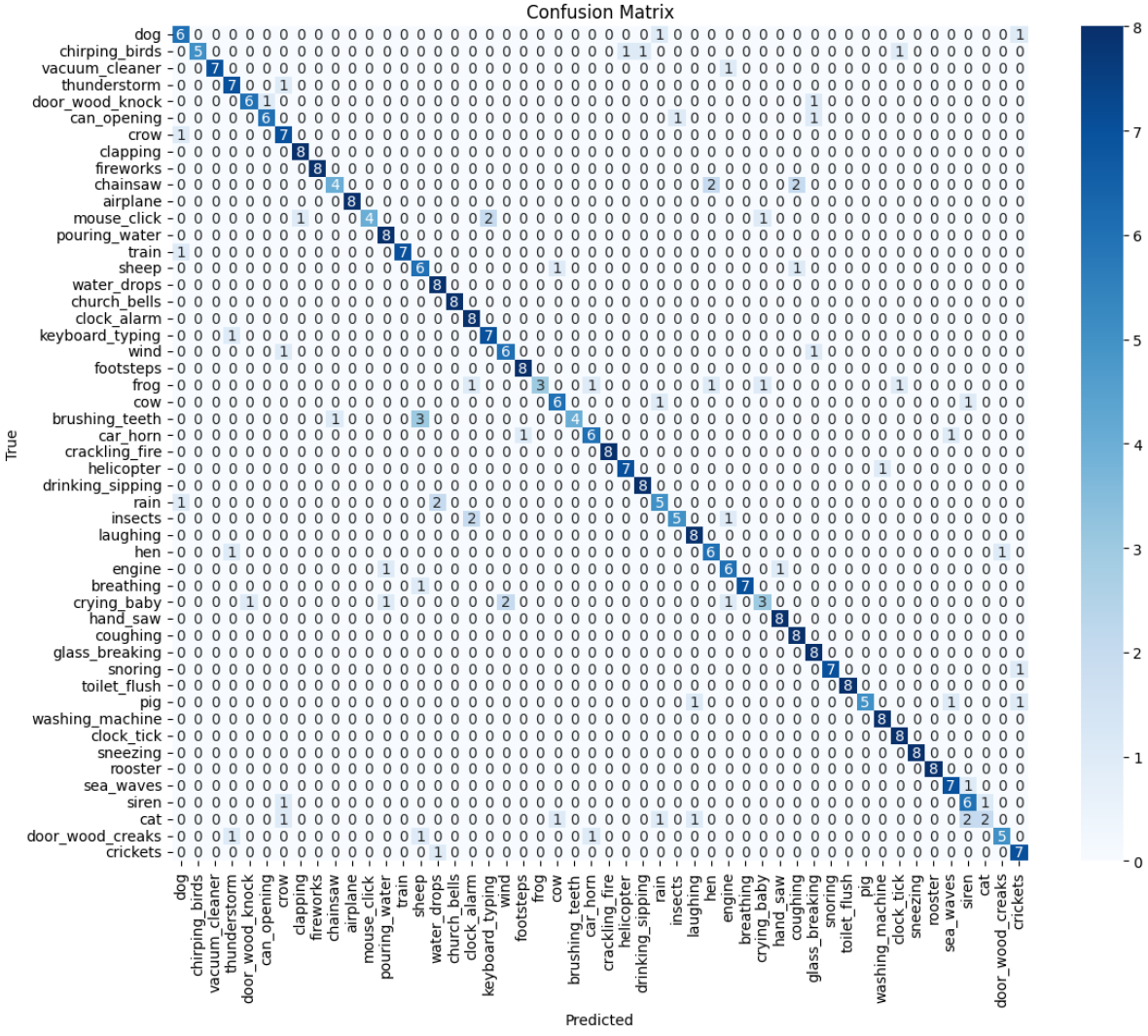


FIGURE 4.32: Confusion Matrix of the trained Model

Noteworthy observations Include:

- Consistent high-accuracy regions for specific sound classes
- Minimal cross-category misclassifications
- Potential challenges in distinguishing between acoustically proximate sound types.

4.8.3 Receiver Operating Characteristic

The ROC curve analysis in fig 4.33 demonstrates exceptional model performance, as evidenced by the near-perfect micro-average ROC curve with an Area Under the Curve (AUC) of 0.99. This metric indicates the model's outstanding binary classification capabilities across all sound categories.

The selected class-specific ROC curves (Classes 13, 39, 30, 45, and 17) uniformly exhibit AUC values approaching 1.00, signifying the discriminative power, Minimal false positive and false negative rates, Consistent performance across diverse audio categories

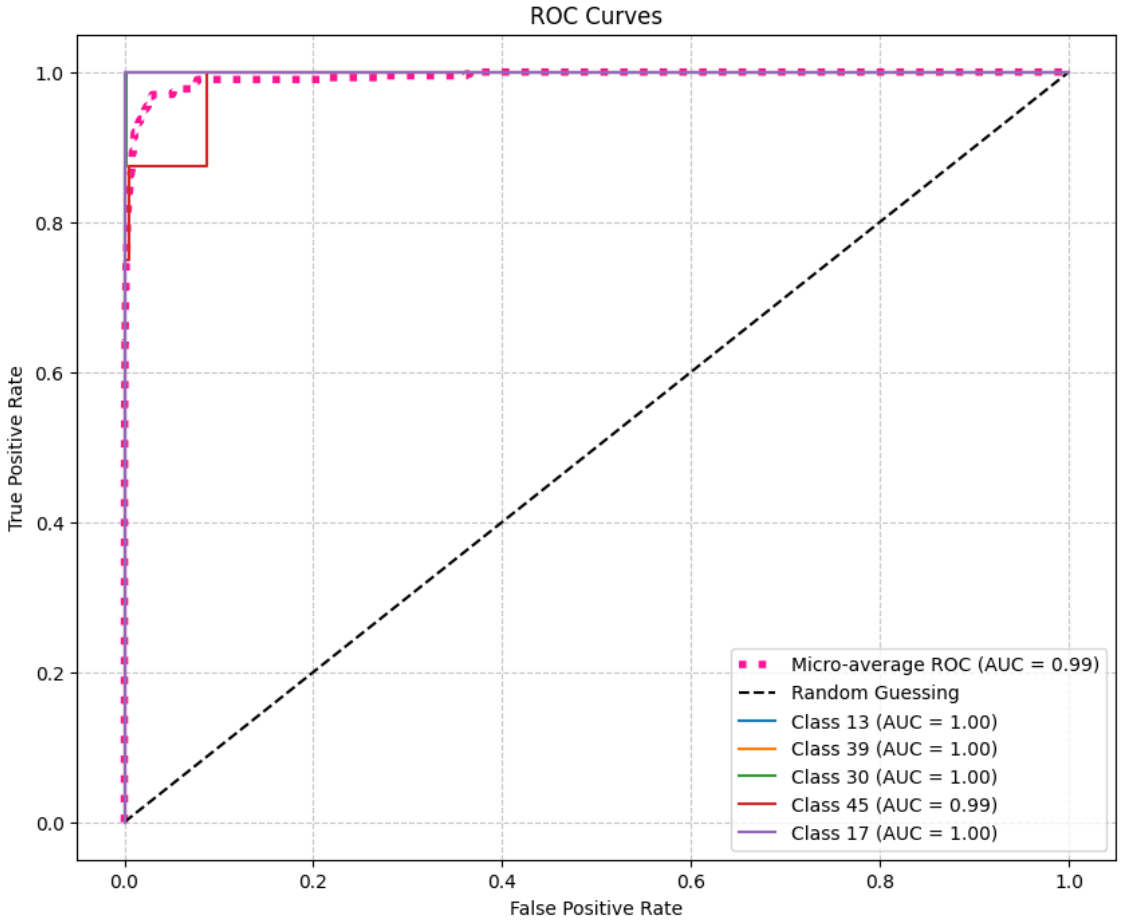


Figure 4.33: ROC Curve

The proximity of individual class curves to the top-left corner of the ROC space further underscores the model's high-precision classification capabilities.

4.8.4 Per-Class Performance Metrics

Figure 4.34 presents a comprehensive visualization of per-class performance metrics for a multi-class audio classification model. The horizontal bar graph displays precision (blue), recall (orange), and F1-score (green) for each audio class, revealing varying model performance across different sound categories.

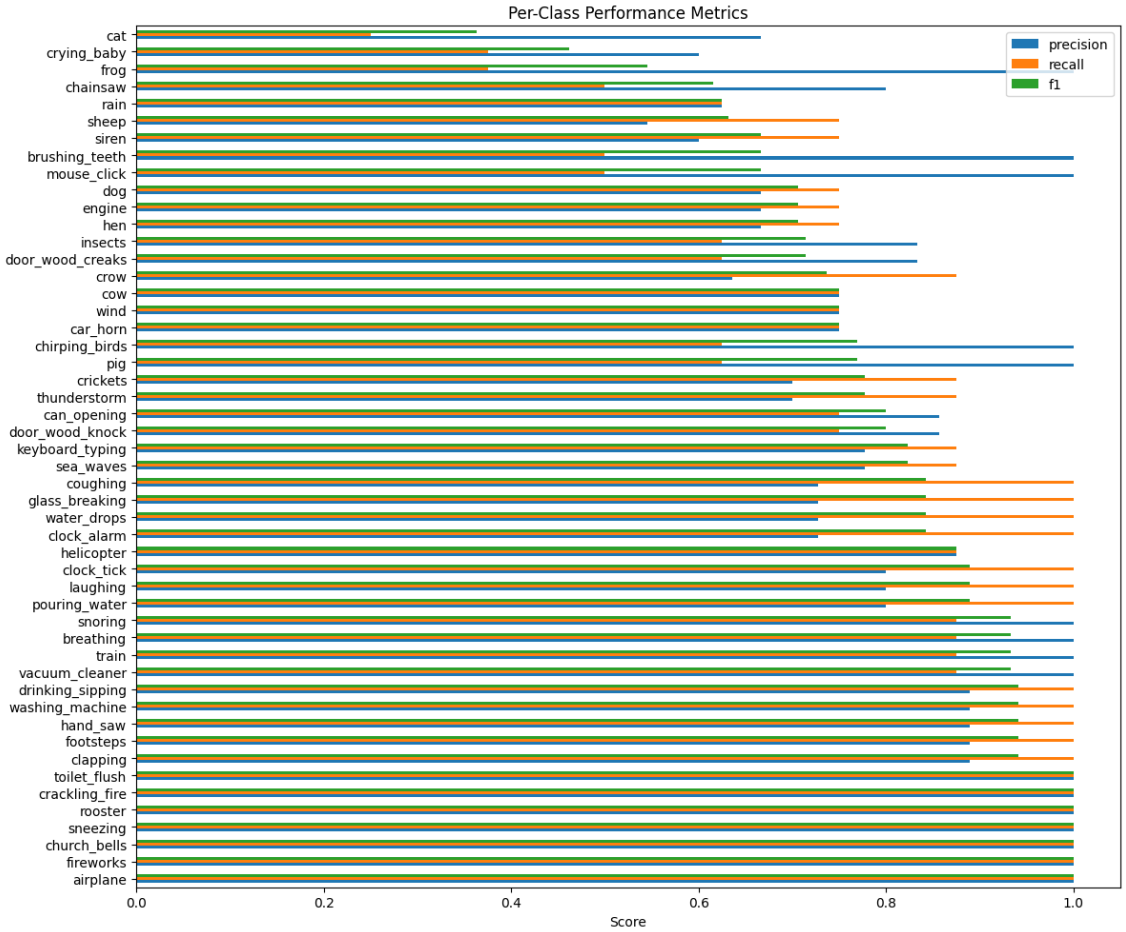


FIGURE 4.34: Per-class performance metrics for audio classification model

Notable observations include:

- Classes such as *brushing_teeth*, *mouse_click*, and *pig* demonstrate high precision.
- Some classes exhibit balanced performance with nearly equivalent precision, recall, and F1-scores.
- Certain classes show significant disparities between precision and recall, indicating potential classification challenges.

4.8.5 Spectral Visualization and Misclassification Analysis

Figure 4.35 illustrates input mel spectrograms and Grad-CAM overlay visualizations, providing insights into the model's feature attention and misclassification patterns[15].

The misclassification examples reveal intriguing characteristics:

- Class 21 (True) was misclassified as Class 17 with a low confidence of 0.11
- Class 6 (True) was predicted as Class 0 with a confidence of 0.12
- Spectral patterns suggest complex feature interactions leading to misclassifications

The Grad-CAM overlays highlight the model's attention regions, demonstrating how different frequency and temporal features contribute to classification decisions.

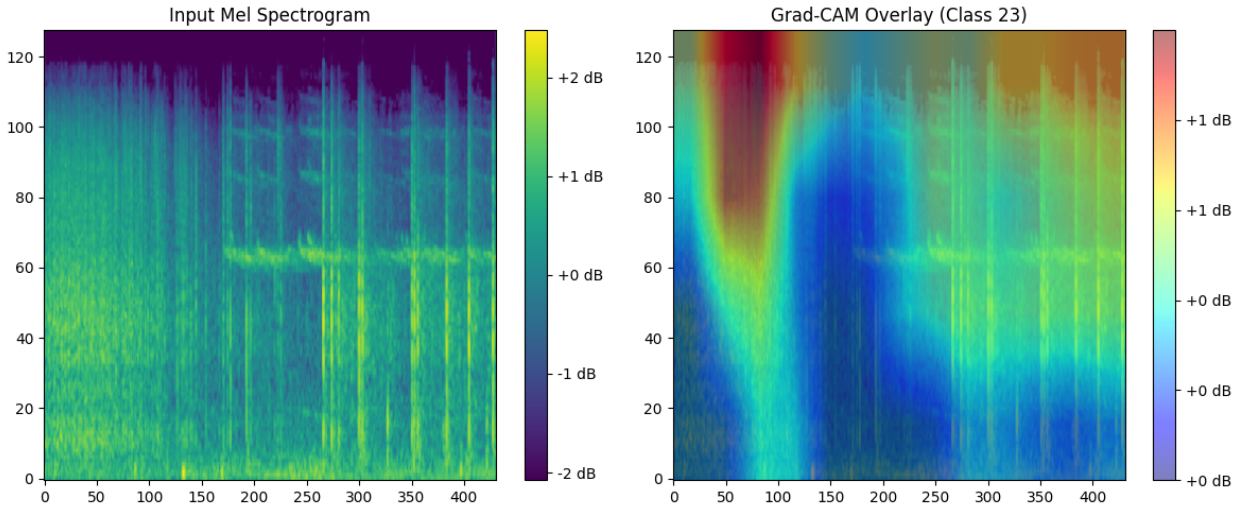


FIGURE 4.35: Mel spectrogram and Grad-CAM visualization of model misclassifications

Chapter 5

Conclusion

This thesis has demonstrated the feasibility of interpreting the hidden representations of convolutional neural networks using a NMF-based method. The key contribution of this work is the demonstration that Non-negative Matrix Factorization (NMF) can effectively decompose the input spectrogram using activations from hidden layers of the classification neural netowrk. As detailed in Chapter 2, this approach diverges from traditional end-to-end CNN models by uncovering the underlying spectral structures in a clear and interpretable way. Chapter 3 further shows that combining classical signal processing techniques with modern deep learning not only enhances efficiency but also improves the model's ability to generalize across diverse audio scenarios. Moreover, the integration of channel attention mechanisms, as explored in Chapter 4, significantly boosts the network's discriminative performance by dynamically adjusting feature responses. Together, these contributions establish a novel framework for creating interpretable audio classification models and offer a promising direction for future research.

While the findings of this thesis are promising, several limitations must be acknowledged. The study predominantly relies on the ESC-50 dataset, as described in Chapter 4, which, despite being a well-recognized benchmark, may not fully represent the diversity found in real-world audio environments. Additionally, the computational and memory requirements of the proposed model, discussed in detail in Chapter 4, indicate that further optimization is necessary to achieve real-time performance, particularly for large-scale applications. Although the integration of NMF has improved interpretability, the current analysis would benefit from more advanced visualization techniques and interactive tools, which could provide deeper insights into the non-linear interactions within the network. Future research should focus on expanding the dataset to include a wider range of audio sources, exploring alternative regularization and model compression strategies, and developing comprehensive visualization frameworks to facilitate a more thorough understanding of the internal processes of deep audio models.

Bibliography

[1] Jont B. Allen and Lawrence R. Rabiner. A unified approach to short-time fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, 1977.

[2] Keunwoo Choi, Ginos Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. *arXiv preprint arXiv:1609.04243*, 2017.

[3] John R. Hershey et al. Cnn architectures for large-scale audio classification. In *ICASSP*, 2017.

[4] Heqing He, Ke Wu, Wei Zheng, and Zhi Li. Sound source localization using a convolutional neural network. *Sensors*, 21(23):8031, 2021.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[7] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, 2015.

[10] Gang Sun Jie Hu, Li Shen. Squeeze-and-excitation networks. In *CVPR*, 2018.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[13] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[14] Y. Liu, L. Xie, and W. Wang. A survey of sound source localization with deep learning methods. *The Journal of the Acoustical Society of America*, 152(1):107–123, 2022.

[15] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[16] K. J. Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pages 1015–1018. ACM, 2015.

[17] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press.

[18] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[19] Romain Serizel, Victor Bisot, Slim Essid, and Gaël Richard. Acoustic features for environmental sound analysis. In *Computational Analysis of Sound Scenes and Events*, pages 71–101. Springer, 2018.

[20] M. Slaney. Auditory toolbox (version 2.2). Technical report, Interval Research Corporation, 1998.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[22] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[23] Carnegie Mellon University. Short-time fourier transforms. `https://course.ece.cmu.edu/~ece491/lectures/L25/STFT_Notes_ADSP.pdf`. Accessed: 2025-03-29.

[24] Fei Wang, Meng Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[25] H. Wang, Y. Zhang, X. Li, and J. Chen. Sound source localization based on residual network and channel attention mechanism. *Scientific Reports*, 13(1):1234, 2023.

Appendix A

Patterns and Interpretation

In this chapter, we present a comprehensive analysis of the employed architectures to obtain and compare the different approaches mentioned in chapter 3

A.1 Feature Extraction

The extraction of meaningful representations from spectrograms is crucial for a wide array of applications—from environmental sound classification to music information retrieval. This chapter presents a detailed investigation into spectral pattern decomposition using a deep learning framework. In contrast to traditional linear decomposition techniques like Non-Negative Matrix Factorization (NMF), our method leverages hook-based feature extraction to obtain intermediate representations from a neural network. These feature maps are then interpreted and analyzed, revealing insights about underlying spectral patterns.

To explore different methods chosen to reconstruct the audio features in order to show meaningful spectral patterns, we must first take a look at the results we have extracted from the trained neural network as Input vs. Target spectrogram A.1

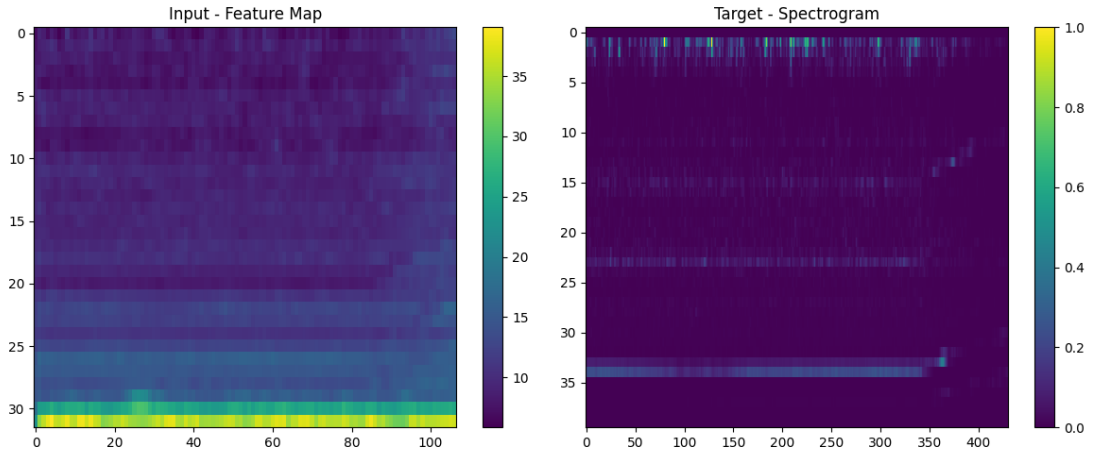


FIGURE A.1: Mel Filterbank Visualization

A.2 Results

We introduced neural network architectures that employ transformation strategies to enhance audio feature reconstruction and interpretation. These architectures integrate constraint-driven oper-

ations and perceptually informed spectral mappings, thereby extending traditional transformation methods. The discussion will detail three primary models—LinearNoBiasSoftplus, MelConstrained Linear, and MelBasisTransform—each designed to optimize the conversion of audio signals into interpretable spectral patterns. Through theoretical analysis and experimental validation, this chapter elucidates how these novel transformation strategies improve model transparency, efficiency, and overall classification performance, setting the stage for further exploration of training methodologies and visualization techniques in subsequent sections.

A.2.1 LinearNoBiasSoftplus: Constraint-Driven Linear Transformation

The model is a simple yet effective neural network layer that applies a linear transformation to the input while ensuring non-negative weights using the Softplus function. Unlike a standard fully connected layer, this model does not include a bias term, which can help maintain stability in audio feature transformations. The softplus function ensures that the learned weights remain positive, preventing the issue of negative filter weights, which could distort spectral features.

This model is particularly useful for processing spectrogram-like data, where the frequency representation should remain meaningful and non-negative. The weight initialization is performed using Xavier uniform initialization, optimizing the initial distribution for better convergence. By leveraging this approach, the model attempts to learn a direct mapping from input feature maps to mel-scaled spectrogram outputs, making it a good baseline for spectral transformation tasks. The softplus activation function plays a crucial role, providing several computational advantages:

- Smooth, non-linear weight constraints,
- Prevention of vanishing gradient problems,
- Controlled weight evolution during training

Below, are the results from this approach:

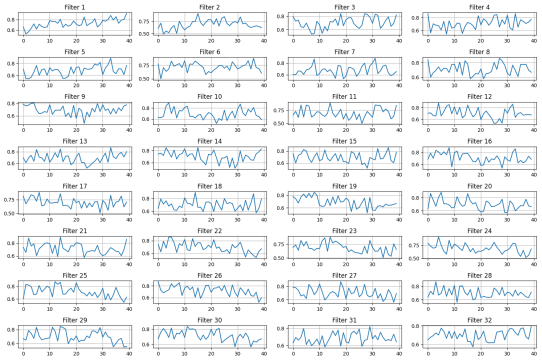


FIGURE A.2: Filters at Epoch 20

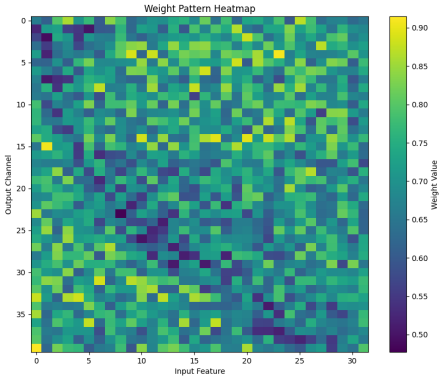


FIGURE A.3: Heatmap at Epoch 20

FIGURE A.4: Comparison of Filters and Heatmap at Epoch 20

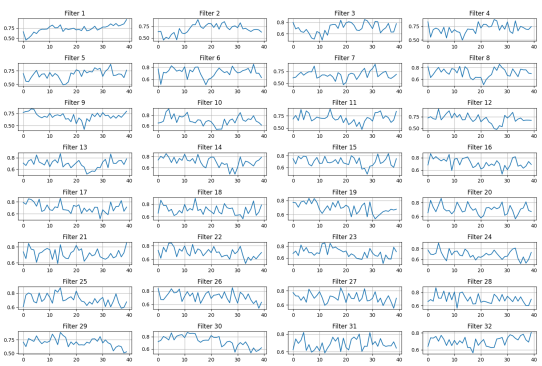


FIGURE A.5: Filters at Epoch 40

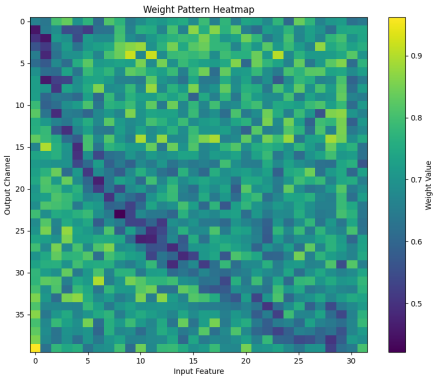


FIGURE A.6: Heatmap at Epoch 40

FIGURE A.7: Comparison of Filters and Heatmap at Epoch 40

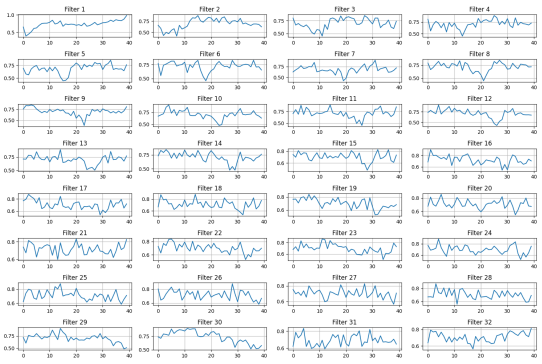


FIGURE A.8: Filters at Epoch 60

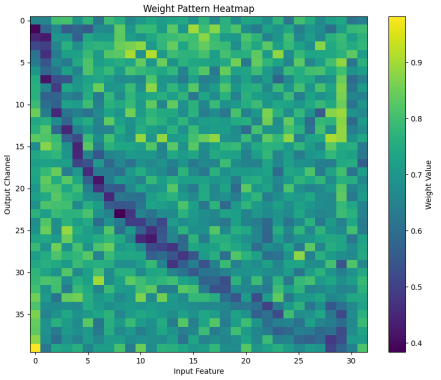


FIGURE A.9: Heatmap at Epoch 60

FIGURE A.10: Comparison of Filters and Heatmap at Epoch 60

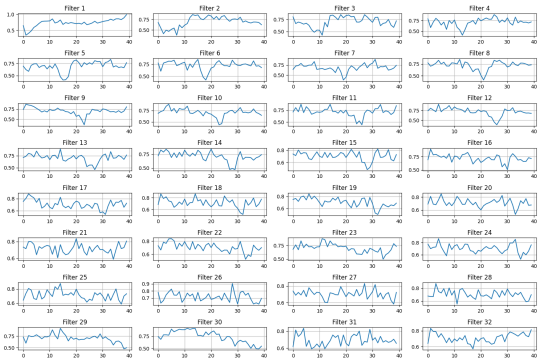


FIGURE A.11: Filters at Epoch 80

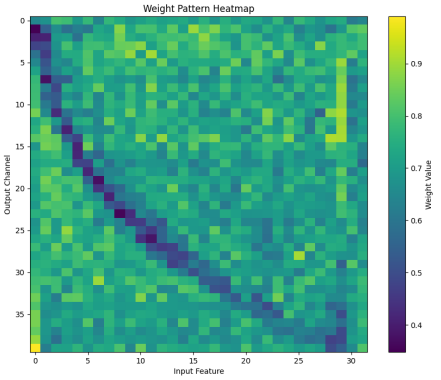


FIGURE A.12: Heatmap at Epoch 80

FIGURE A.13: Comparison of Filters and Heatmap at Epoch 80

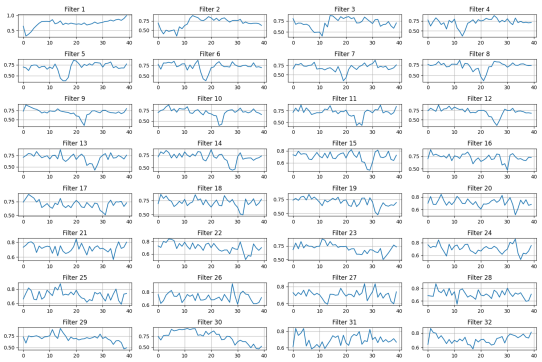


FIGURE A.14: Filters at Epoch 100

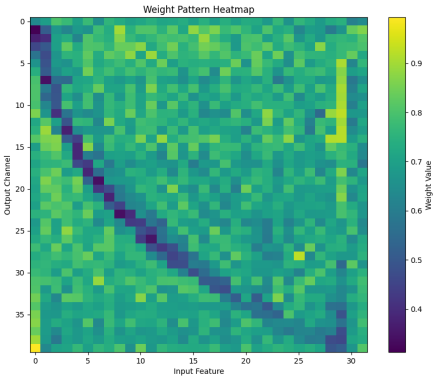


FIGURE A.15: Heatmap at Epoch 100

FIGURE A.16: Comparison of Filters and Heatmap at Epoch 100

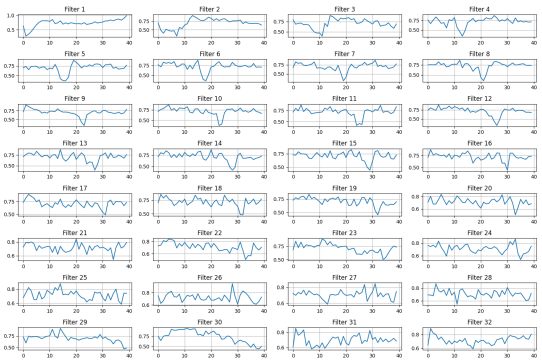


FIGURE A.17: Filters at Epoch 120

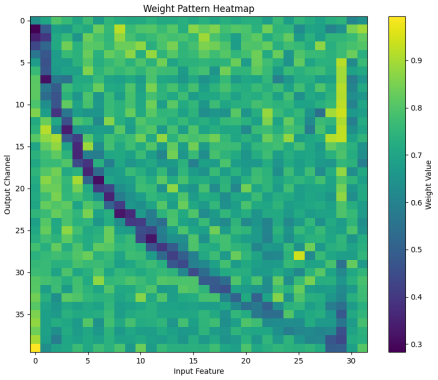


FIGURE A.18: Heatmap at Epoch 120

FIGURE A.19: Comparison of Filters and Heatmap at Epoch 120

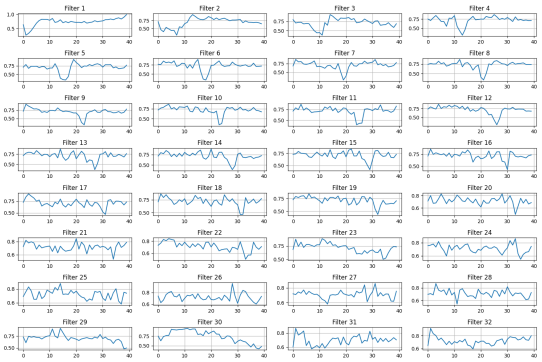


FIGURE A.20: Filters at Epoch 140

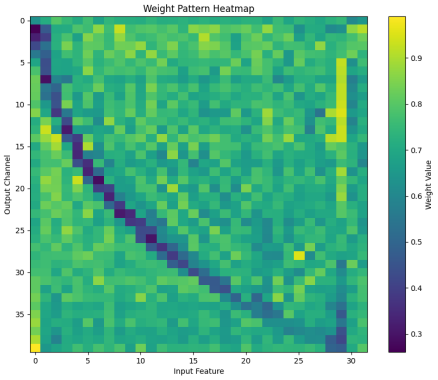


FIGURE A.21: Heatmap at Epoch 140

FIGURE A.22: Comparison of Filters and Heatmap at Epoch 140

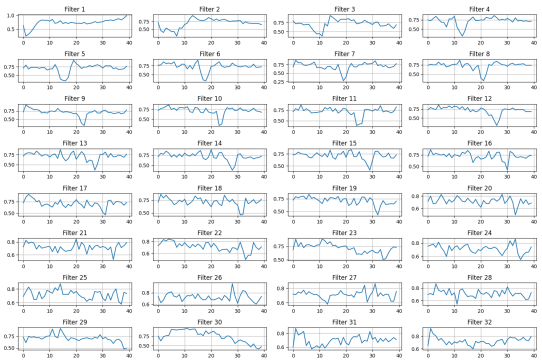


FIGURE A.23: Filters at Epoch 160

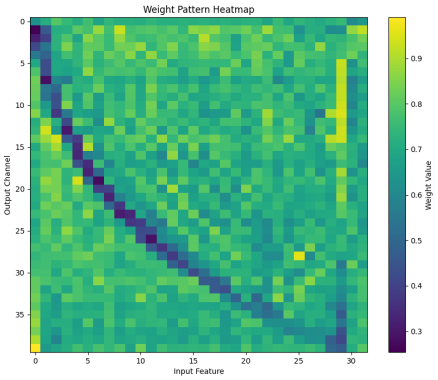


FIGURE A.24: Heatmap at Epoch 160

FIGURE A.25: Comparison of Filters and Heatmap at Epoch 160

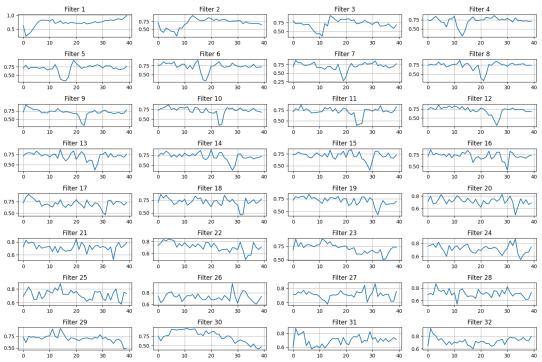


FIGURE A.26: Filters at Epoch 180

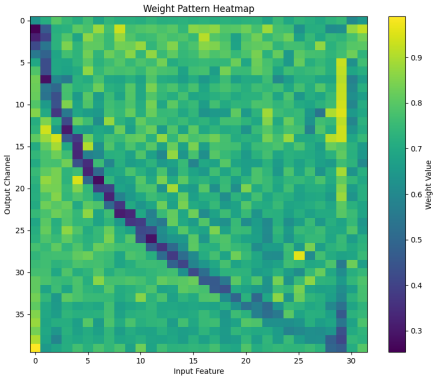


FIGURE A.27: Heatmap at Epoch 180

FIGURE A.28: Comparison of Filters and Heatmap at Epoch 180

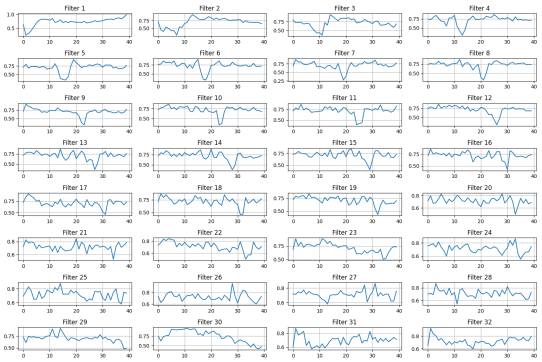


FIGURE A.29: Filters at Epoch 200

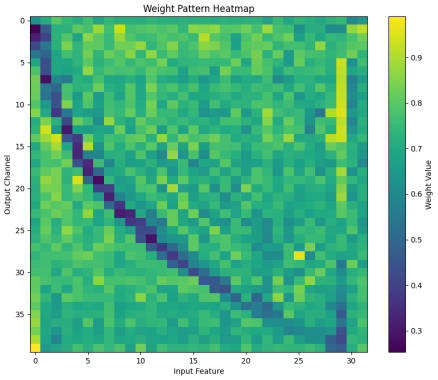


FIGURE A.30: Final heatmap at Epoch 200

FIGURE A.31: Comparison of Filters and Heatmap at Epoch 200

The model's tensor transposition strategy allows for efficient computational flow, treating frequency dimensions as learnable, transformable spaces rather than static representations.

A.2.2 MelConstrained Linear: Perceptually-Informed Spectral Mapping

The MelConstrained Linear model represents the pinnacle of perceptually-informed neural network design. Unlike traditional linear transformations, this architecture dynamically generates frequency-dependent filter weights that mimic the human auditory system's frequency perception. Key innovation lies in its ability to learn:

- Adaptive filter center frequencies,
- Dynamic bandwidth controls.
- Triangular spectral shape approximations

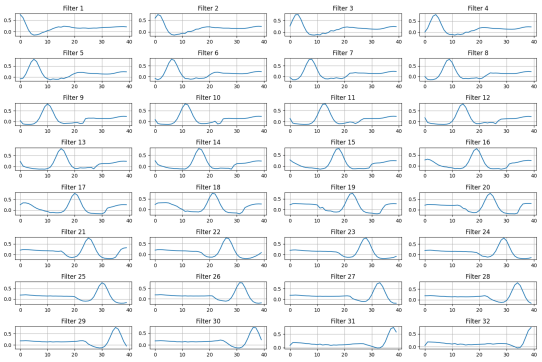


FIGURE A.32: Filters at Epoch 20

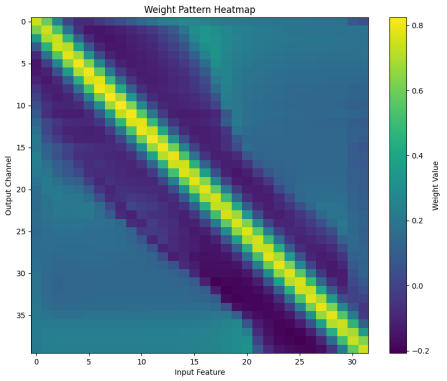


FIGURE A.33: Heatmap at Epoch 20

FIGURE A.34: Comparison of Filters and Heatmap at Epoch 20 (Mel Basis)

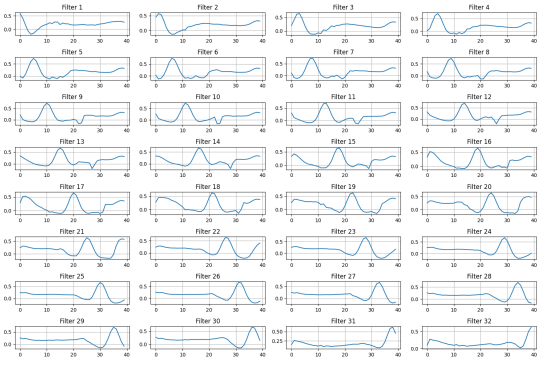


FIGURE A.35: Filters at Epoch 40

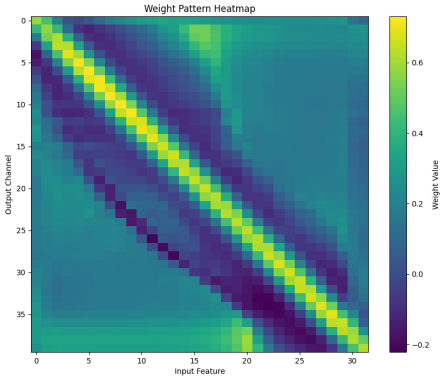


FIGURE A.36: Heatmap at Epoch 40

FIGURE A.37: Comparison of Filters and Heatmap at Epoch 40 (Mel Basis)

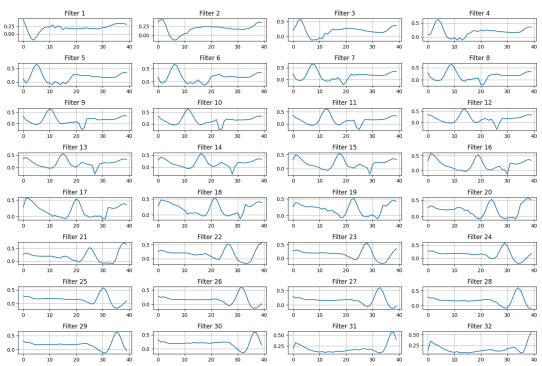


FIGURE A.38: Filters at Epoch 60

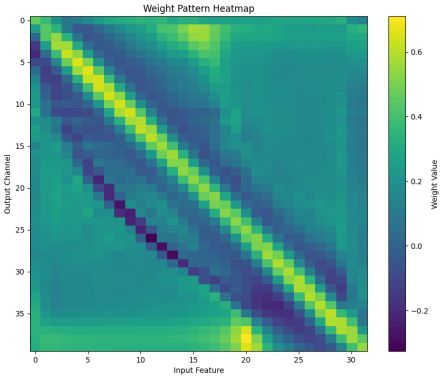


FIGURE A.39: Heatmap at Epoch 60

FIGURE A.40: Comparison of Filters and Heatmap at Epoch 60 (Mel Basis)

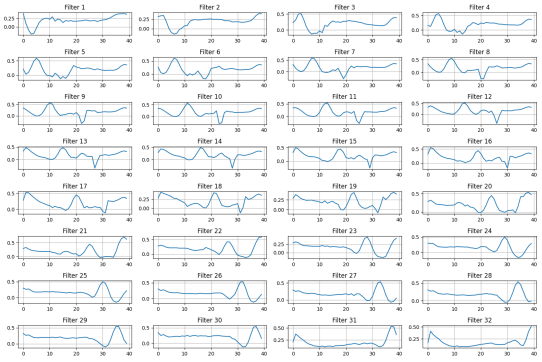


FIGURE A.41: Filters at Epoch 80

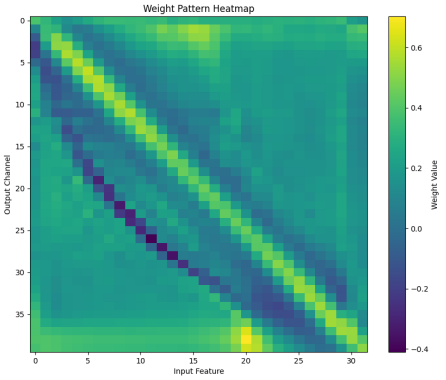


FIGURE A.42: Heatmap at Epoch 80

FIGURE A.43: Comparison of Filters and Heatmap at Epoch 80 (Mel Basis)

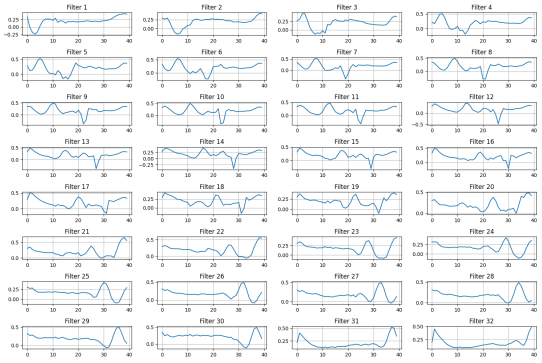


FIGURE A.44: Filters at Epoch 100

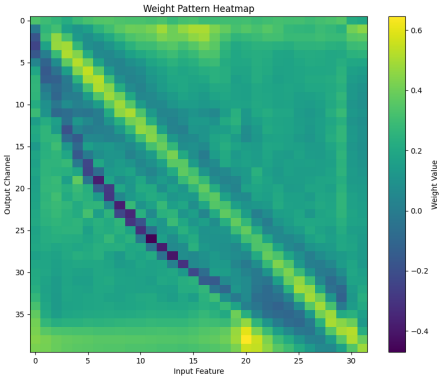


FIGURE A.45: Heatmap at Epoch 100

FIGURE A.46: Comparison of Filters and Heatmap at Epoch 100 (Mel Basis)

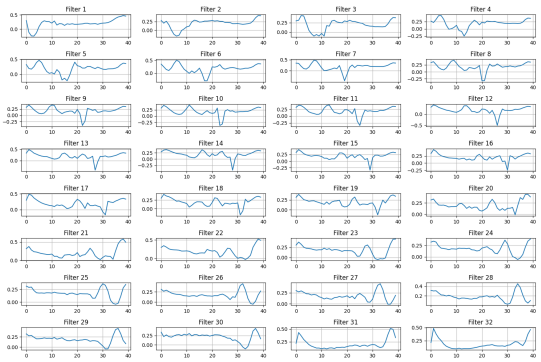


FIGURE A.47: Filters at Epoch 120

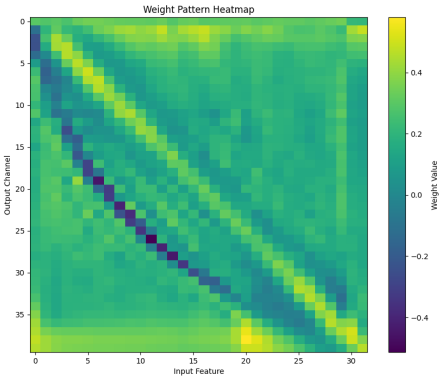


FIGURE A.48: Heatmap at Epoch 120

FIGURE A.49: Comparison of Filters and Heatmap at Epoch 120 (Mel Basis)

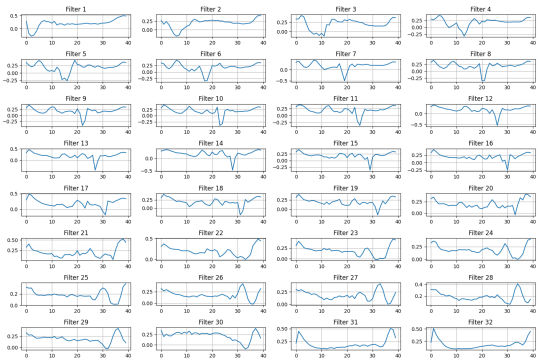


FIGURE A.50: Filters at Epoch 140

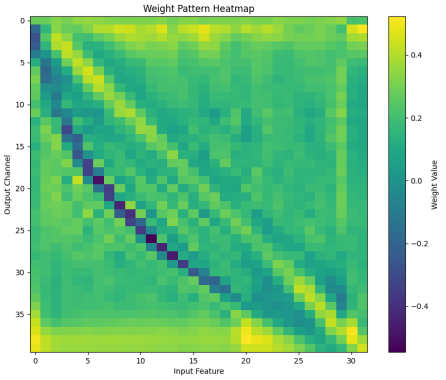


FIGURE A.51: Heatmap at Epoch 140

FIGURE A.52: Comparison of Filters and Heatmap at Epoch 140 (Mel Basis)

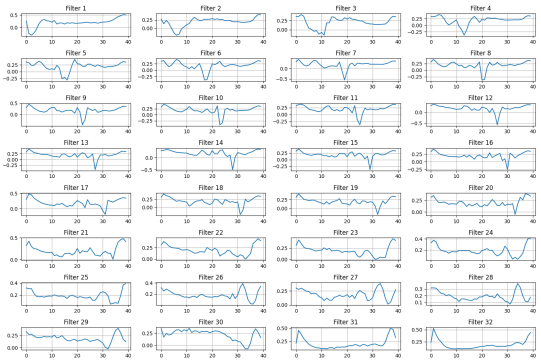


FIGURE A.53: Filters at Epoch 160

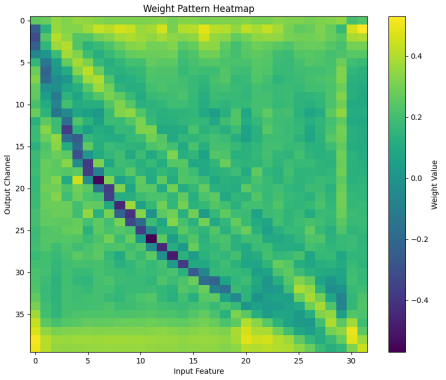


FIGURE A.54: Heatmap at Epoch 160

FIGURE A.55: Comparison of Filters and Heatmap at Epoch 160 (Mel Basis)

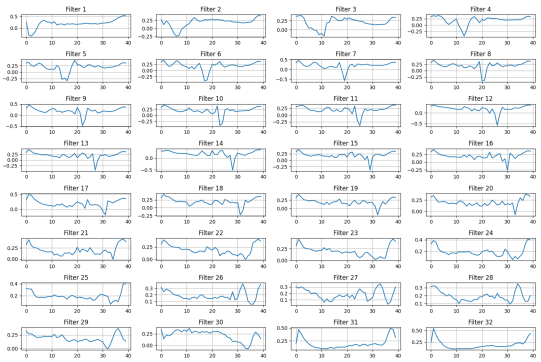


Figure A.56: Filters at Epoch 180

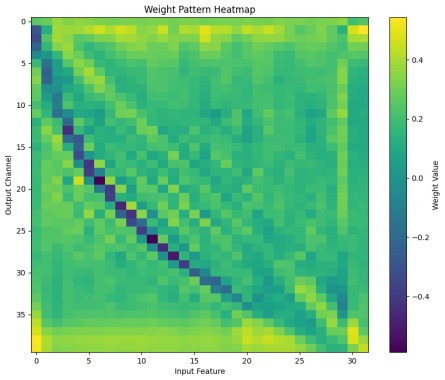


Figure A.57: Heatmap at Epoch 180

Figure A.58: Comparison of Filters and Heatmap at Epoch 180 (Mel Basis)

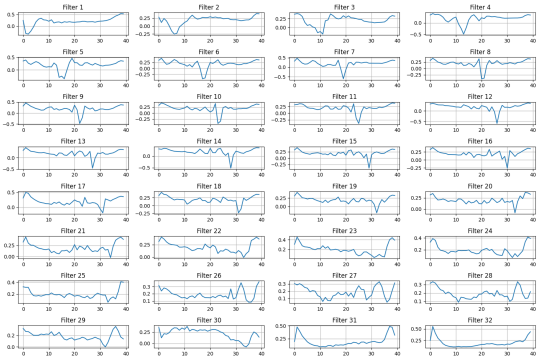


Figure A.59: Filters at Epoch 200

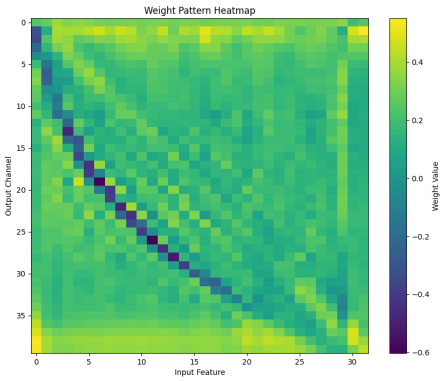


Figure A.60: Heatmap at Epoch 200

Figure A.61: Comparison of Filters and Heatmap at Epoch 200 (Mel Basis)

Think of this model as a neural network that doesn't just transform features but understands their spectral context. By generating frequency-dependent filter weights, it creates a transformation that respects the nuanced nature of audio signals. The model's computational core involves dynamically generating triangular filter shapes that adapt to the input's spectral characteristics. This approach goes beyond static transformations, creating a learning mechanism that can capture subtle spectral variations.

A.2.3 MelBasisTransform Model

The MelBasisTransform model takes a more rigid approach by directly initializing the weights to approximate a mel filter bank. Unlike the previous models, which learn the transformation from data, this model starts with predefined mel scaling and only fine-tunes the scaling factors through training. This approach ensures that the learned representation remains interpretable and structured while still allowing minor adaptations based on the dataset. The model initializes its weights using a Gaussian-shaped approximation of mel filters, which helps capture the logarithmic nature of human frequency perception. This approach is particularly beneficial when working with small datasets, as it reduces the number of trainable parameters and enforces a well-structured frequency transformation. However, this model is less flexible than the other two, as it relies heavily on the predefined filter structure rather than learning from raw data.

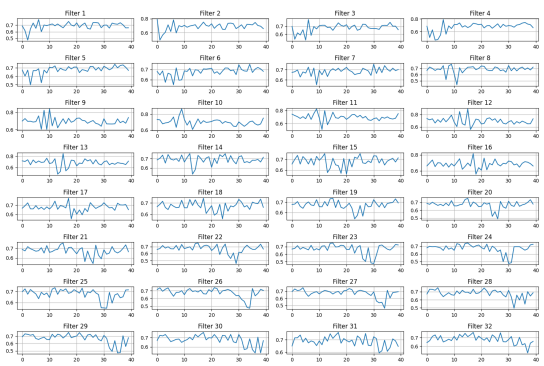


FIGURE A.62: Filters at Epoch 20

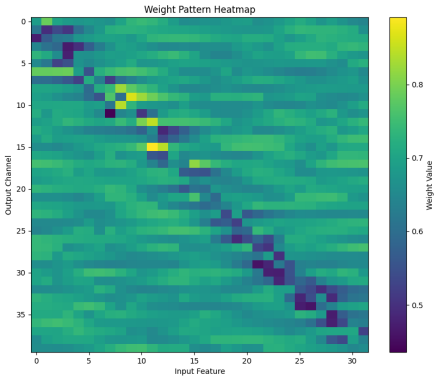


FIGURE A.63: Heatmap at Epoch 20

FIGURE A.64: Comparison of Filters and Heatmap at Epoch 20 (Mel Constrained)

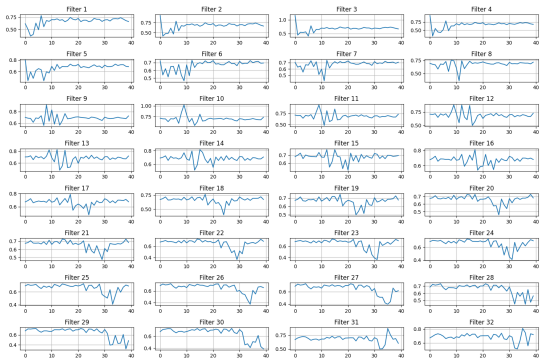


FIGURE A.65: Filters at Epoch 40

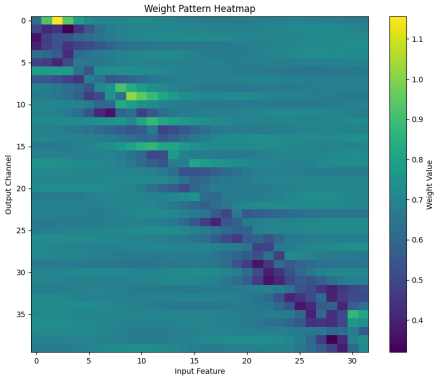


FIGURE A.66: Heatmap at Epoch 40

FIGURE A.67: Comparison of Filters and Heatmap at Epoch 40 (Mel Constrained)

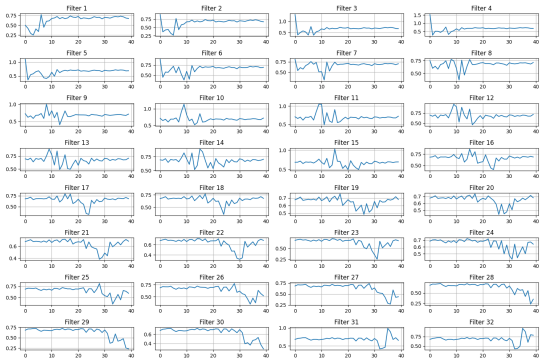


FIGURE A.68: Filters at Epoch 60

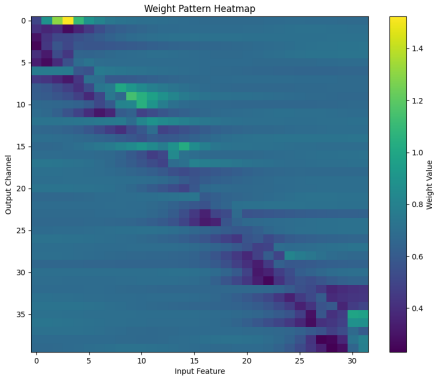


FIGURE A.69: Heatmap at Epoch 60

FIGURE A.70: Comparison of Filters and Heatmap at Epoch 60 (Mel Constrained)

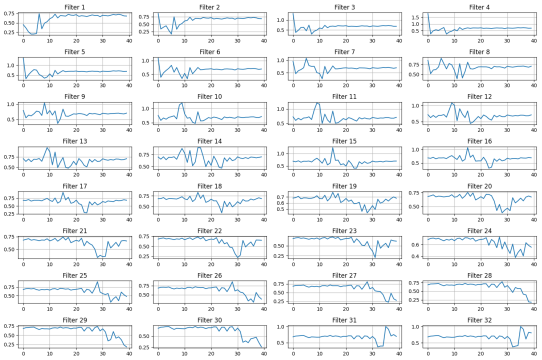


FIGURE A.71: Filters at Epoch 80

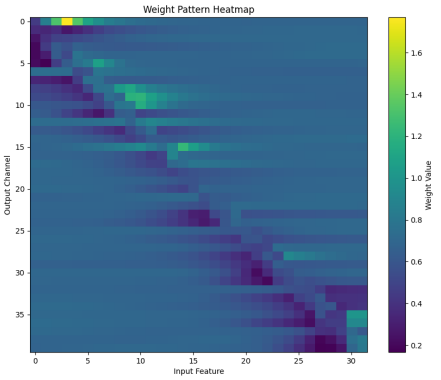


FIGURE A.72: Heatmap at Epoch 80

FIGURE A.73: Comparison of Filters and Heatmap at Epoch 80 (Mel Constrained)

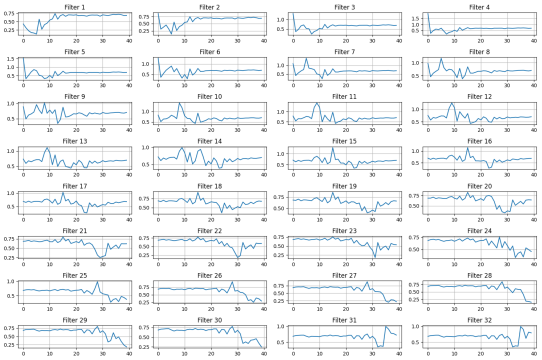


FIGURE A.74: Filters at Epoch 100

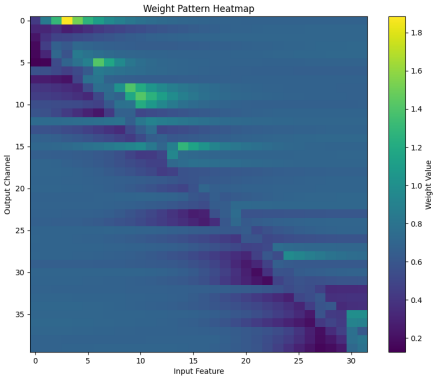


FIGURE A.75: Heatmap at Epoch 100

FIGURE A.76: Comparison of Filters and Heatmap at Epoch 100 (Mel Constrained)

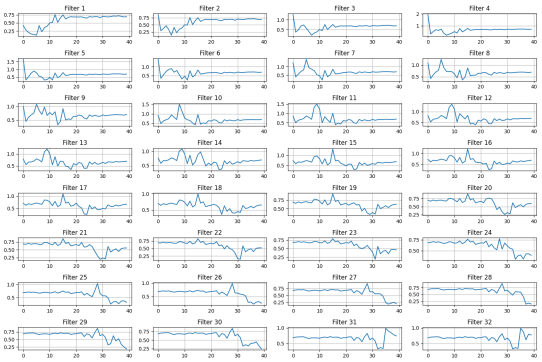


FIGURE A.77: Filters at Epoch 120

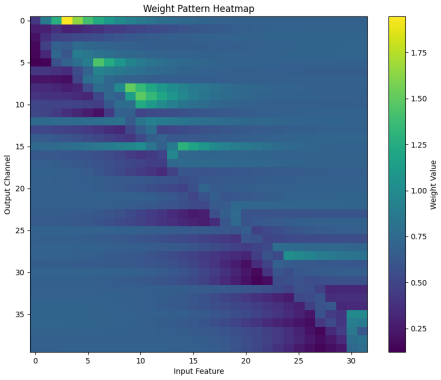


FIGURE A.78: Heatmap at Epoch 120

FIGURE A.79: Comparison of Filters and Heatmap at Epoch 120 (Mel Constrained)

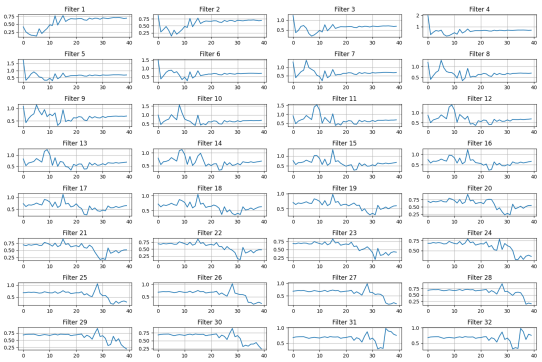


FIGURE A.80: Filters at Epoch 140

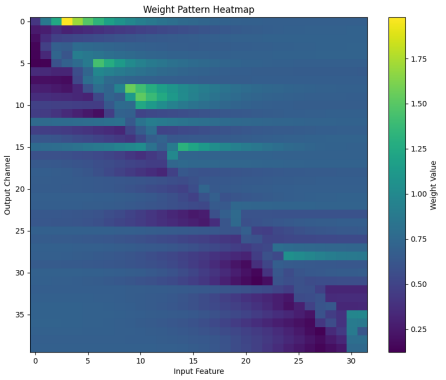


FIGURE A.81: Heatmap at Epoch 140

FIGURE A.82: Comparison of Filters and Heatmap at Epoch 140 (Mel Constrained)

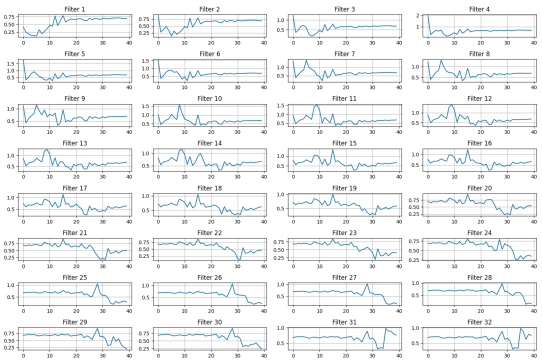


FIGURE A.83: Filters at Epoch 160

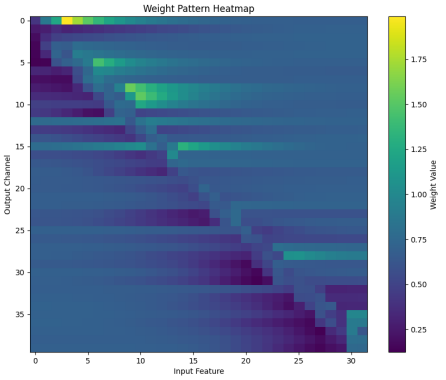


FIGURE A.84: Heatmap at Epoch 160

FIGURE A.85: Comparison of Filters and Heatmap at Epoch 160 (Mel Constrained)

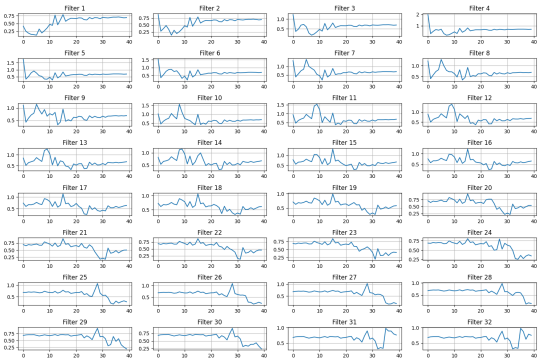


FIGURE A.86: Filters at Epoch 180

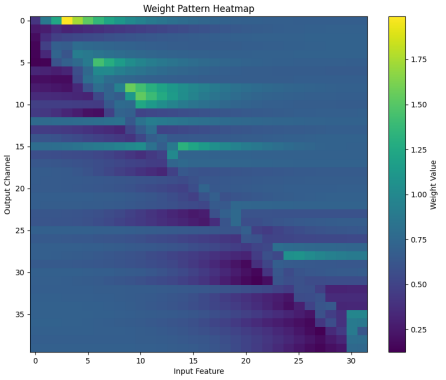


FIGURE A.87: Heatmap at Epoch 180

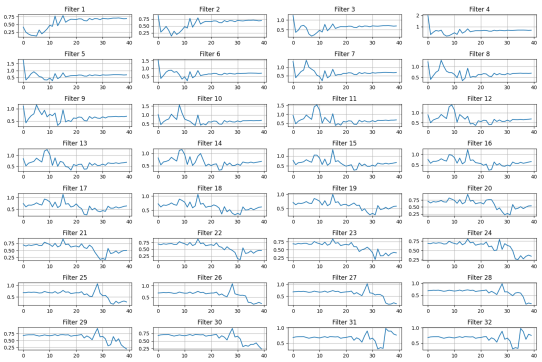


FIGURE A.88: Filters at Epoch 200

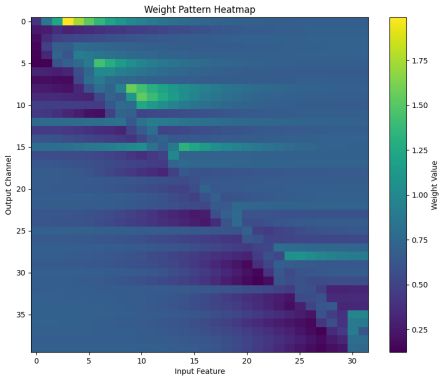


FIGURE A.89: Heatmap at Epoch 200

A.2.4 Training Methodology

The training process introduces a novel spectral smoothness regularization technique that fundamentally differs from traditional loss functions. Instead of merely minimizing prediction errors, this approach encourages coherent, smooth weight transitions across frequency bands. Spectral smoothness loss can be conceptualized as a gentle constraint that prevents abrupt changes in learned representations. It's analogous to creating a smooth landscape of learned features rather than a jagged, discontinuous terrain. By penalizing rapid weight changes between adjacent frequency bands, the training methodology ensures:

- More stable learned representations
- Reduced overfitting
- Enhanced generalization capabilities

A.2.5 Visualization and Interpretability

The visualizations help us understand how each model processes input spectrograms and transforms them into different learned representations. The Fig 4.11 plots illustrate activations for multiple input samples, while the fig 4.18 plots show predicted outputs compared to the target spectrogram.