Data Driven Mel Filter Bank Design for Environmental Sound Analysis

Michael Moses Rothstein

A thesis submitted in partial fulfillment for the
degree of Master of Science
in
Electrical and Computer Engineering

APPROVED:

Professor Bashima Islam

Professor Patrick Schaumont

Professor Ziming Zhang

Worcester Polytechnic Institute

Abstract

Audio classification is a vital technique in environmental monitoring, facilitating the automatic categorization of audio data into predefined classes based on acoustic features. From identifying wildlife vocalizations to assessing urban noise pollution levels, its applications are diverse and pivotal in understanding and managing ecosystems and urban environments.

The conventional audio classification method often utilizes Mel Frequency Cepstral Coefficients (MFCC) extracted from audio files as input to a Deep Neural Network (DNN) classifier. However, its effectiveness is limited by a fixed filterbank structure, designed for the human audio range but lacking optimization and adaptability to diverse datasets. To address this, we propose a customized MFCC approach (Pertinant Spectral Characteristic MFCC), aligning the filterbank with dataset-specific frequency power distribution peaks, thus enhancing classification accuracy and adaptability.

Through a comparative analysis across various environmental datasets, including ESC50, UrbanSound8K, and Gunshot our study demonstrates the superiority of the Pertinant Spectral Characteristic MFCC (PSC-MFCC) approach. Specifically, we observed a notable 4.5% increase in classification accuracy and a 1.47% decrease in standard deviation compared to the traditional MFCC method, showcasing its potential to significantly enhance audio classification accuracy and precision.

These findings underscore the practical utility and efficacy of the proposed methodology in environmental audio classification tasks. By accurately capturing and distinguishing features within diverse frequency ranges across classes, the PSC-MFCC approach offers a promising avenue for advancing audio classification techniques in environmental monitoring and conservation efforts.

Acknowledgements

I would like to thank BASH LAB for their help, guidance, and assistance throughout the entirety of this project. Additionally we would like to thank Worcester Polytechnic Institute for providing us with the opportunity to complete this project. This thesis used human dataset that is publicly published online.

Contents

List of Figures

1 Introduction

Audio classification plays a significant role in analyzing sound data, categorizing it into predefined classes based on extracted features from the audio signal. Environmental audio classification is particularly important as it facilitates the monitoring and understanding of ecosystem health, human impacts on natural habitats, and the development of effective conservation strategies.

Environmental sounds encompass a diverse range of auditory signals found in natural and urban environments, such as animal calls, traffic noise, wind patterns, and water flow. Studying these sounds is crucial for understanding ecosystem health, assessing biodiversity, and monitoring human impacts on habitats. Through audio classification techniques, researchers can gain deeper insights into ecological processes, inform conservation efforts, and mitigate human-induced disturbances to ecosystems.

To classify environmental sounds, diverse recordings from natural or urban environments are collected, enabling classifiers to discern distinct features such as bird calls, traffic noise, or water flow patterns. Leveraging machine learning algorithms, these classifiers contribute to various applications including wildlife monitoring, urban soundscape analysis, and environmental impact assessments.

A common approach to audio classification involves utilizing Mel-Frequency Cepstral Coefficients (MFCC) to extract spectral features from audio signals. These features serve as inputs to machine learning models, particularly Deep Neural Networks (DNN), which excel at learning complex patterns in data. Trained on labeled datasets, DNNs can effectively classify audio signals, enabling tasks like wildlife monitoring and urban soundscape analysis with high accuracy and efficiency.

However, traditional methods for audio classification, such as the use of MFCC, have limitations in adaptability to diverse datasets such as environmental sounds. To ad-

dress this, we customize the MFCC filterbank to align with dataset-specific frequency power distribution peaks (rather than following the standard Mel-scale approach) in order to enhance classification accuracy and adaptability.

We present three significant contributions to the field. First, we introduce a novel filterbank optimization algorithm designed to enhance the extraction of spectral features from audio datasets. This algorithm generates a tailored filterbank by analyzing a uniform sample of the dataset, effectively highlighting the most pertinent spectral characteristics. Second, we introduce a customized Mel-Frequency Cepstral Coefficients (MFCC) function capable of incorporating the optimized filterbank as an input. This Pertinent Spectral Characteristic MFCC (PSC-MFCC) function facilitates more precise feature extraction, thereby improving the overall classification process. Finally, we demonstrate a notable 4.5% enhancement in classification accuracy when compared to conventional MFCC approaches, underscoring the practical utility and efficacy of the proposed methodology (Pertinent Spectral Characteristic MFCC). These contributions are summarized below.

- Filterbank optimization algorithm that enhances the extraction of key spectral features from audio.
- A customized MFCC function capable of incorporating the optimized filterbank.
- The custom MFCC (PSC-MFCC) increases classification accuracy by 4.5% compared to the traditional MFCC approach.

2 Literature Review and Background Study

This section discuses background information related to environmental sounds and audio classification.

2.1 Environmental sounds

Environmental sounds include a broad array of auditory elements found in both natural and urban settings. These sounds form an integral part of our daily surroundings, influencing our perceptions and interactions with the environment. Specialized datasets such as ESC50 and UrbanSound8k are used to facilitate the study and analysis of environmental sounds. ESC50 consists of 50 classes, each comprising 40 five-second audio clips capturing various environmental sounds, such as waves, frogs, wind, rain, thunder, insects, and helicopters[11]. Meanwhile, UrbanSound8k, with its 8,732 audio files across 10 classes, offers insights into urban auditory landscapes, featuring sounds like air conditioners, street music, dog barks, and sirens. These datasets reflect the diverse nature of environmental sounds, spanning a wide spectrum of frequencies and sources[12]. Classifiers are made to distinguish between the large range of sounds, leading to meaningful insights[4].

The analysis and classification of environmental sounds can provide a wealth of information. For instance, through the identification of bird sounds, we can better understand the avian population in a given area[2]. This knowledge can then be used to aid conservation efforts, by locating endangered bird species, and even provide information about migration patterns, enriching our understanding of these phenomena. Moreover, gunshot sound detection can help surveillance efforts[17]. By pinpointing the source of gunfire, this technology enables authorities to ascertain whether firearms are being discharged, identify the specific type of firearm involved, and assess its legality. Such information is invaluable for combating activities like poaching and ensuring public safety through vigilant monitoring. Similarly,

noise pollution classification serves to quantify and identify the sources of excessive noise[4]. This information helps in recognizing when noise levels exceed acceptable limits and devising strategies for noise pollution reduction. Furthermore, acoustic natural disaster classification utilizes audio to detect events like tsunami, volcanoes, hurricanes, and earthquakes[5]. Classifying these events before they occur can save countless lives. Finally, rainfall audio detection estimates rain intensity in real time, determining the severity of rainfall, and guiding the decisions that must be made to keep people safe. Although classifying audio signals provides many benefits it is a complicated task.

2.2 Audio classification

Audio classification involves categorizing audio signals into predefined classes, typically accomplished using machine learning algorithms. Deep Neural Networks, a subset of machine learning, are commonly employed for this task. Deep Neural Networks emulate the structure of the human brain, with nodes and layers, enabling them to recognize patterns and perform classification. The general structure of Deep Neural Networks comprises three types of layers: input, hidden, and output. The input layer receives input features, while the hidden layers learn patterns in the data through weighted connections between neurons. Finally, the output layer produces the classification. Before Deep Neural Networks perform classification, they must undergo training on labeled data. During training, the network adjusts its parameters based on input-output pairs, optimizing its ability to accurately classify audio signals. After training, the Deep Neural Network's performance is evaluated through testing, where its ability to classify unseen data is accessed[6].

2.3 Acoustic feature: Mel-Frequency Cepstral Coefficients (MFCC)

The classifiers input plays a significant role in the training and testing process. Large datasets may slow down classification and require extensive computational resources,

but blindly reducing data size can remove crucial distinguishing features resulting in lower accuracy. Preprocessing is vital to balance efficiency and accuracy. Preprocessing steps are used to strike this balance. A combination of high efficiency and accuracy are an indication of successful preprocessing steps. One of the most popular audio classifier preprocessing steps is Mel-Frequency Cepstral Coefficients (MFCC). These coefficients are able to capture the spectral envelope of an audio signal and express it in a compact way. The spectral envelope ensures that key features are not lost and the compact output leads to high efficiency[9].

The Mel-Frequency Cepstral Coefficients are calculated following a six step process. The first step involves segmenting the audio signal into time-domain windows. This requires reading and analyzing the audio file, where each data point denotes an amplitude at a distinct moment. The duration between these data points is dictated by the audio file's sampling frequency, measured in samples per second (Hz). The time domain representation of the signal is segmented into different windows. The windows must be of a consistent structure. This structure has parameters such as shape, length, and overlap that describe the windowing approach. Some common windowing shapes include rectangular window, triangular window, Welch window, Hann window, and Hamming window. Once windowing is applied and the data is segmented, each segment can be individually analyzed, offering insights into the signal's characteristics across various time intervals.

After windowing the signal in the time domain, the frequency response is determined. This is achieved by computing the Discrete Fourier Transform (DFT) within each individual window. By doing so valuable information about both frequency and time is obtained. The outcome is a depiction of the frequency response across various time intervals.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \frac{kn}{N}} \tag{1}$$

This operation is computed in each frame.

X[k] represents the frequency component of the DFT output sequence.

x[n] represents the input windowed time sequence.

N represents the total number of samples in the input sequence.

e is the base of the natural logarithm.

j is the imaginary unit.

The subsequent step involves computing the power, achieved by squaring the magnitude of the windowed frequency response. This operation simplifies the data by discarding phase information while serving as a quantitative measure of signal strength. The outcome of this process is a representation of signal power across various frequencies and time intervals.

$$|X[k]|^2 = Powerof X[k] \tag{2}$$

The third step involves applying the Mel-filterbank to the spectrogram. The purpose of the Mel-filterbank is to mimic the non-linear frequency perception of the human auditory system, emphasizing important frequencies while subordinating spectral noise. The Mel-filterbank typically consists of 20 to 40 overlapping triangular filters spaced according to the Mel-scale, with denser spacing at lower frequencies gradually transitioning to sparser spacing at higher frequencies. In general, the filterbank covers the frequency range of human auditory perception, approximately 20 Hz to 20 KHz. The Mel-filterbank is illustrated in Figure 1.

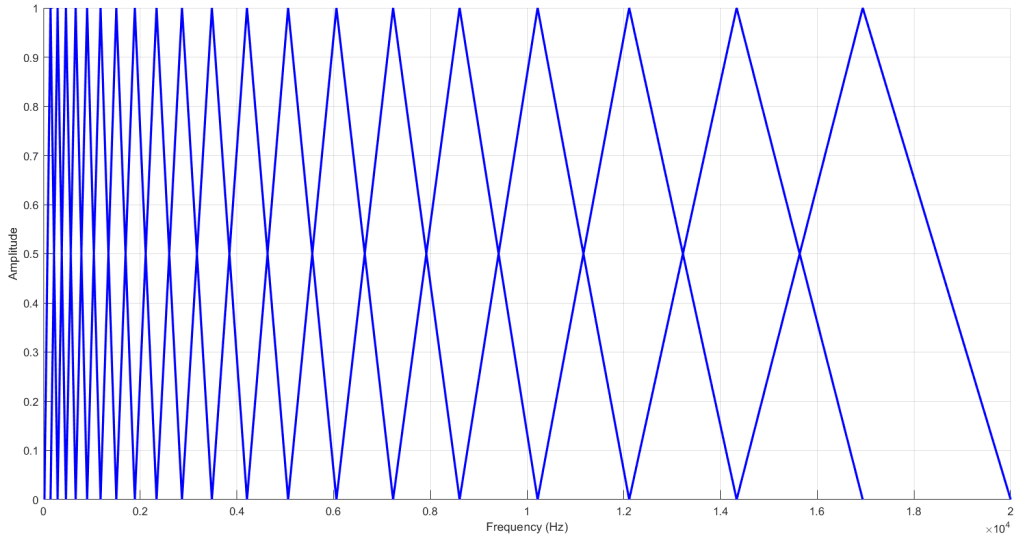

Figure 1: Mel-filterbank

The fourth step involves transforming the spectrogram onto a logarithmic scale, using a base ten logarithm. This logarithmic transformation serves to compress the dynamic range of features within the spectrogram. By doing so, the spectrogram becomes more interpretable, facilitating classification and analysis tasks.

After the logarithmic transformation, the final step involves the application of the Discrete Cosine Transform (DCT) to each row of the transformed spectrogram. This process, conducted along the frequency axis, computes the Mel-Frequency Cepstral Coefficients (MFCCs) for individual time windows. Each row of the spectrogram, corresponding to a distinct time window, undergoes an independent DCT calculation. The resulting DCT coefficients represent the MFCCs, offering a concise depiction of the spectral characteristics. Typically, 13 of these coefficients are preserved, facilitating dimensionality reduction while retaining crucial spectral features. The DCT output forms a matrix where one dimension signifies time frames, while the other represents cepstral coefficients, encompassing overall energy and spectral shape. These MFCCs function as the ultimate feature representation,

apt for tasks like speech recognition or audio classification.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \cos\left(\frac{\pi(2n+1)k}{2N}\right) \tag{3}$$

Operation computed in every frame.

X[k] represents the DCT coefficient.

x[n] represents the input log scaled power sequence.

N is the length of the input sequence.

2.4 Limitations of MFCC

While the conventional approach of employing Mel-Frequency Cepstral Coefficients (MFCC) and Neural Networks for classification generally yields satisfactory results, there are certain limitations worth noting, particularly in the design of the Mel-filterbank used in MFCC computation. As mentioned earlier, the Mel-filterbank comprises triangular filters, densely positioned at lower frequencies and sparsely distributed at higher frequencies following the Mel-scale. Consequently, this setup yields higher resolution at lower frequencies but diminishes resolution at higher frequencies. Although this configuration may suffice for certain datasets it lacks adaptability and robustness. This is particularly evident when working with diverse datasets, such as environmental sound datasets, consisting of varying frequency ranges and sound sources. In such cases, the MFCC often falls short in adequately capturing vital spectral information. This deficiency arises due to the uncertainty surrounding the optimal positioning of triangular filters to amplify resolution around critical spectral points while efficiently suppressing spectral noise. For instance if a dataset has a high quantity of information at both high and low frequencies the Mel-filterbank would fail to capture the information at the high frequencies. In this case a filterbank such as the one in Figure 2 would be desirable, as it provides high resolution at both high and low frequencies while

attenuating the middle frequency range.

To address this challenge, the adoption of a tailored filterbank customized to the dataset emerges as a viable solution. By tailoring the filterbank to suit the specific characteristics of the dataset this approach enhances adaptability, bolsters robustness, and ultimately augments accuracy in audio classification tasks.

Designing a custom filterbank presents a formidable challenge. The primary reason it is difficult is that the best filterbank for each dataset is unknown. A Deep Neural Network approach would rely on a dataset of optimal filterbanks for different datasets to serve as training data, which is unavailable. For this reason digital signal processing techniques are explored.

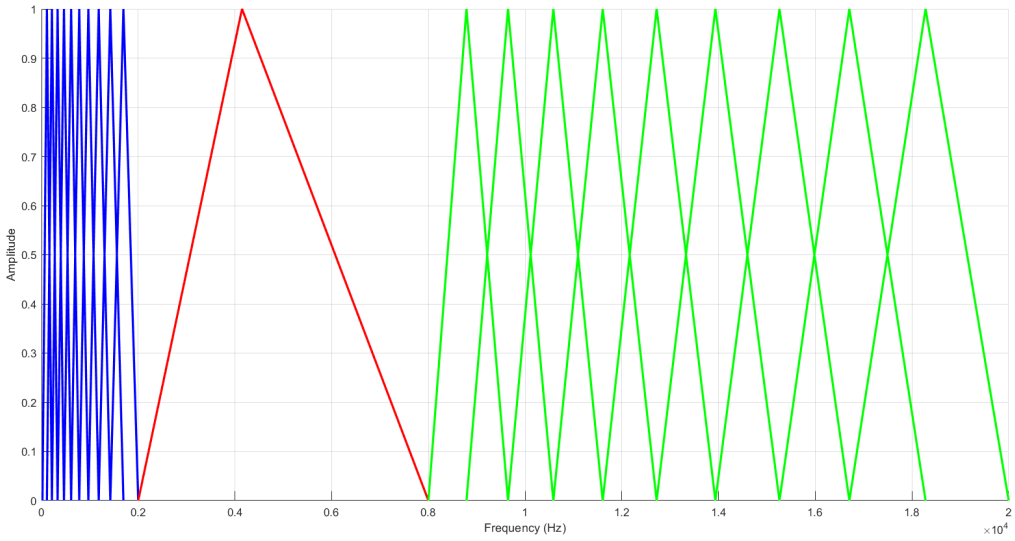

Figure 2: Custom filterbank

3 Methodology

This section discusses the methods used to create a Pertinent Spectral Characteristic (PSC-MFCC) audio classification approach. To develop and assess the PSC-MFCC function effectively, several key elements must be in place: a filterbank optimization algorithm, a PSC-MFCC function, and a Deep Neural Network audio classifier. These three components are shown in Figure 3. The optimization algorithm plays a pivotal role in determining the optimal placement of triangular filters within the custom filterbank, ensuring it aligns precisely with the spectral characteristics of the dataset. Meanwhile, the PSC-MFCC function utilizes the tailored filterbank to perform MFCC calculations. Once these components are developed, they can be used as an input to a Deep Neural Network classifier to evaluate the performance across a diverse set of environmental datasets. The performance of the classifier is evaluated by comparing the test accuracy of the traditional MFCC to the PSC-MFCC.

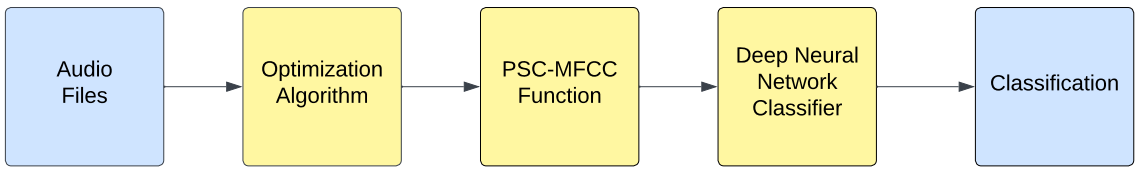

Figure 3: PSC-MFCC components

3.1 Filterbank optimization algorithm

The filterbank optimization algorithm crafts a filterbank specifically suited for the dataset's spectral characteristics. Next, the PSC-MFCC function uses this optimized filterbank to compute MFCC features. Finally, the audio classifier evaluates the performance of the PSC-MFCC function, ensuring its efficacy in diverse environmental sound audio classi-

fication tasks.

The filterbank optimization algorithm is tasked with identifying the optimal locations for the triangular filters. Due to the variability of these locations across datasets, it requires a uniform representative sample of the dataset as an input. This sample typically comprises around 30% of the dataset, with approximately 30% of the audio files from each class included. By analyzing this sample, the algorithm must determine the predominant frequencies of the dataset, which in turn guides the placement of the triangular filters within the custom filterbank.

The predominant frequencies serve as the basis for constructing bandpass triangular filters. Each set of three predominant frequencies delineates a single triangular filter. Consequently, the count of triangular filters usually falls two units short of the number of predominant frequencies. Since 20 triangular filters are commonly used in the Mel-filterbank, we require 22 predominant frequencies to form our filterbank. These filters' width mirrors the gap between the first and third predominant frequencies, while their apex aligns with the midpoint predominant frequency at an amplitude of 1. This implies that the triangular filters rise from the first predominant frequency till the second predominant frequency, then fall from the second predominant frequency till the third predominant frequency. The rising and falling frequency ranges are attenuated, by a factor between 0 and 1, while the peak center frequency is kept untouched.

There are many ways to determine the predominant frequencies used in the custom filterbank. The method used to determine the predominant frequencies has a drastic impact on the triangular filter placement and therefore the classification accuracy. We explore two separate approaches to determine the predominant frequencies.

- Approach 1: Incremental Power Distribution Peaks
- Approach 2: Power Distribution Peaks

3.1.1 Approach 1: Incremental Power Distribution Peaks

Approach one begins with a uniform sample of the dataset. The uniform sample is found by randomly taking 30% of the audio files in each class of the dataset. This ensures that each class is fairly represented. Next, each audio file is analyzed individually.

Initially, each audio file is read in the time domain. Audio files comprise a predefined number of samples, where each sample corresponds to a specific amplitude. Additionally, the audio file maintains a fixed sampling rate, denoted in samples per second. To determine the time associated with each sample, the sample number is multiplied by the inverse of the sampling rate. This calculation yields the temporal information for each sample. The combination of time data and corresponding amplitudes constitutes the fundamental time domain representation of the audio file.

Upon obtaining the time domain representation of each audio file, they undergo a transformation into the frequency domain. Analyzing audio files in the frequency domain reveals information about the amplitude and phase of frequencies inherent within the signal. Such analysis facilitates the extraction of spectral characteristics, thereby highlighting the components comprising the signal. This transformation is accomplished using the discrete Fourier transform (DFT). The discrete Fourier transform is calculated using a summation. The summation spans the number of samples in the audio file. The result of the summation is a function in terms of k, where k denotes the frequencies present in the signal. The frequencies range from 0 to half of the sample rate due to the Nyquist theorem. After computing the Fourier transform of each audio file information about frequency, amplitude, and phase is obtained. The next step is to find the power of each audio signal.

The computation of power relies on the output of the discrete Fourier transform (DFT), where power represents the signal's strength across various frequencies. This computation combines both amplitude and phase information, emphasizing the power of the frequency response while disregarding individual phase and amplitude details. As a result,

this process condenses the information, offering a concise metric of strength across different frequencies. The power is found by squaring the magnitude of the frequency response. In some instances, this computation undergoes scaling by factors such as frequency or total power, a technique known as power normalization. This normalization practice ensures a fair comparison among different audio signals by facilitating the evaluation of their relative power distributions, rather than fixating on specific power amplitudes. After the power is found in each audio file information about frequency and power is obtained. The subsequent step is to apply a summation across the power of each audio file.

The summation is computed by adding the power of each audio file together. As a result of the audio files sharing the same sampling rate and length, their periodograms exhibit identical structures. Consequently, the power representation (periodogram) of each audio file will possess the same length and comprise powers at identical frequencies. This alignment of periodograms allows for straightforward point-by-point addition. Thus, addition is completed by adding the corresponding elements in the power array of each audio file. After the summation is calculated the result is the accumulated power at each frequency. The accumulated periodogram represents the frequency and power of the entirety of the uniform dataset sample. The next step is to segment the periodogram.

The periodogram is split into two intervals of equal length, thereby providing two segments, each spanning a different frequency range. The length of each segment is determined by dividing the maximum frequency present in the signal (derived from half the sampling rate) by two. This is done to ensure an adequate representation in each of the two frequency ranges. The next step is to use the segmented periodogram to determine the highest peaks.

Each segment of the periodogram is analyzed independently. Within each segment, the frequencies associated with the highest powers are identified and placed in ascending order. As these frequencies carry the most power, they encapsulate important information

about the distinctive characteristics of an audio signal. These frequencies are designated as predominant frequencies. As previously mentioned, 20 triangular bandpass filters require 22 predominant frequencies. Therefore to establish 20 triangular bandpass filters in the filterbank, 11 periodogram peaks are detected within each segmented periodogram. This approach provides comprehensive resolution of the important frequencies regardless of if they are in high or low frequency ranges.

3.1.1.1 Limitations

While this approach effectively aligns the filterbank with the dataset, it comes with certain limitations. The primary concern lies in its imposition of equal resolution at both high and low frequencies. Although the triangular filters are strategically positioned to capture significant frequencies within each periodogram segment, it's important to note that one segment might contain more information than another. In such cases, the distribution of filters across segmented periodograms should not be uniform. The solution to this is to not limit the distribution of triangular filters to certain frequency ranges. This can be implemented by not setting a requirement for how many periodogram peaks must be present in a certain range of frequencies. The second approach aims to fix this limitation.

3.1.2 Approach 2: Power Distribution Peaks

The second approach begins in a similar manner to the first approach. First, a uniform sample of the dataset is taken. The uniform sample of the dataset captures around 30% of each class in the dataset. Next, each audio file in the uniform sample is read in the time domain. The time domain representation of each signal is transformed into the frequency domain. In doing so information about frequency, amplitude, and phase is gained. After this, the power of each of the frequency responses is found and summed together. This results in the summed periodogram, similar to that of approach one.

Once the summed periodogram is computed, a divergence between approach one and approach two emerges. In approach one, the summed response is segmented, resulting in a sub optimal distribution of triangular filters. Conversely, in approach two, the periodogram remains unsegmented, with peaks directly identified from the accumulated periodogram. Upon completion, frequencies corresponding to these peaks are stored in an array, representing the predominant frequencies.

However, interpreting this array at face value is not viable due to certain frequency intervals having an excessive number of predominant frequencies. This surplus of predominant frequencies within narrow frequency bands yields redundant information. This issue is exacerbated by the constraint of limiting the total number of predominant frequencies to 22. Consequently, an overabundance of predominant frequencies within specific frequency ranges not only increases redundancy but also compromises resolution (the number of predominant frequencies) in other critical frequency domains. The solution to this issue is to implement a filter.

The array of predominant frequencies undergoes a filtering algorithm aimed at eliminating redundancies, thereby optimizing the distribution of bandpass filters. This algorithm is designed to maintain adequate spacing between predominant frequencies, needed for achieving proper spectral resolution. By preventing predominant frequencies from clustering too closely, the filtering algorithm ensures that the corrected array maintains sufficient spacing. Specifically, it enforces a constraint where no more than two predominant frequencies are allowed within a 50Hz interval. This approach facilitates the proper distribution of triangular filters throughout the frequency spectrum. Ultimately, the filtering algorithm prevents redundancy and contributes to enhanced spectral resolution. The resulting filtered predominant frequency array guides the placement of triangular filters in the filterbank, determining the frequencies with the highest resolution.

3.2 PSC-MFCC function

After computing the custom filterbank, the following step involves integrating it into the Mel-Frequency Cepstral Coefficients (MFCC) calculations. This integration is accomplished by developing a PSC-MFCC function. The PSC-MFCC function is designed to accept the custom filter bank and apply it in a manner similar to the traditional MFCC's utilization of the Mel-filterbank. Implemented like the traditional MFCC, the PSC-MFCC function sticks to a consistent 6-step process. Throughout these steps, specific parameters are employed to maintain alignment with the traditional Librosa MFCC methodology. The only difference lies in the filterbank utilized. By following this approach, the PSC-MFCC function guarantees a seamless transition from traditional to PSC-MFCC calculations while leveraging the improved capabilities of the custom filterbank.

The PSC-MFCC function begins by analyzing the audio files in the time domain. This is done using the Librosa load audio file function. While using this function the sampling rate parameter is properly set to the sampling rate in the audio files description.The output of the load function is a floating point time series representation of the signal, where each value represents the audio signals amplitude at a different time index.

The initial step in computing MFCC involves applying a windowing process to the time domain representation of the signal. This windowing operation serves two primary purposes: tracking time intervals and mitigating spectral leakage. Time tracking involves segmenting the signal into discrete time frames, while spectral leakage reduction aims to minimize unwanted abruptness in the time domain. To complete this goal a Hann window is employed. A Hann window is used because it is the default window type utilized by the MFCC function in the Librosa library, thus maintaining maximum similarity between our PSC-MFCC function and the traditional Librosa MFCC function. Hann windows comprise a certain length and overlap. The length describes how many samples a window lasts and the overlap describes how many samples one window intersects with another. Following

the default Librosa MFCC parameters, a 2048 sample length and a 512 sample hop length (length of non overlapping samples) are used in the PSC-MFCC function. This implies that each window lasts for 2048 samples and has an overlap of 1536 samples.

Upon applying windowing to the signal, the frequency response within each window is determined through the Fourier transform. Utilizing a Fourier transform with 2048 terms, matching the length of the Hamming window, facilitates this computation. In each window, the Short Time Fourier Transform (STFT) function from Librosa conducts both the windowing and Fourier transform operations, yielding essential information on time, frequency, amplitude, and phase. Furthermore, using this function exactly matches the computations done in the traditional MFCC Librosa function.

The third step is to find the power of the frequency response in each window. This is done to combine the phase and amplitude into a metric of strength at each frequency. In each window, the magnitude squared of the frequency response is taken, resulting in power. After the power computations are performed, information about time, frequency, and power remain. The power computations are done using the numpy abs function and the square operator.

After obtaining the power, the custom filterbank is applied to it. The goal of this step is to emphasize the power at key frequencies while attenuation the others. This involves multiplying each triangular filter in the filterbank with the frequency domain power response and summing the results within each time frame (window). The numpy dot function aids this process. As a result of this step, we gain insights into time (derived from windowing), frequency (from the Fourier transform), and scaled power (obtained through the magnitude squared and the custom filterbank).

The fifth step involves applying a logarithmic scale to the power spectrum. This process is executed similarly to the default behavior in the Librosa MFCC function, utilizing the power to dB function from Librosa. Consistency is preserved between the PSC-MFCC

and the traditional Librosa MFCC by keeping the default parameters when applying the power to dB function.

In the final step, step 6, the Discrete Cosine Transform (DCT) is applied to the data. This transformation is computed separately for each time frame. Within each time frame, the scaled power (frequency domain) is weighted by various terms defined by the DCT equation, and these weighted terms are then summed together. The outcome of this process is a summation within each time frame, dictated by a variable k. In the summation, k denotes the coefficient in the MFCC. Thirteen coefficients are computed for each time frame, ensuring that there is a distinct value corresponding to each coefficient at every time frame. This computation is carried out using the Scipy fft pack DCT function in both the PSC-MFCC and traditional Librosa MFCC. The Scipy DCT function has certain parameters. These parameters include axis, type, and norm. In both the traditional Librosa MFCC and the PSC-MFCC function these parameters are kept consistent. In doing so, axis is set to -2, type is set to 2, norm is set to norm, and only the first 13 coefficients are kept.

After completing the sixth step, the MFCC function is finished. Since the parameters in the PSC-MFCC function are set identically to the ones used in the traditional Librosa MFCC, the only difference between the functions is the application of the custom filterbank. This is essential as it permits a fair comparison between the traditional Librosa MFCC and the PSC-MFCC. The comparison between the traditional Librosa MFCC and the PSC-MFCC is done through the results of a DNN classifier.

3.3 PSC-MFCC Deep Neural Network

The classifier is built upon a hybrid architecture known as Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM), consisting of a total of ten layers. The network is structured with five layers dedicated to the Convolutional Neural Network (CNN) segment and the remaining five to the Long Short-Term Memory (LSTM) segment. This de-

sign combines the spatial feature extraction capabilities of CNNs with the temporal modeling prowess of LSTMs, resulting in robust analysis of sequential data.

The CNN consists of two convolutional layers responsible for extracting intricate spatial features from the input data (MFCC). These convolutional layers are augmented by two batch normalization layers, positioned after each convolutional stage. Batch normalization is used to stabilize and accelerate the training process by normalizing activations, thereby mitigating issues such as internal covariate shift and ensuring smoother gradient propagation. After each convolutional layer, a max-pooling layer is introduced to downsample the feature maps, effectively reducing their spatial dimensions while retaining key information. This operation helps in alleviating computational expense and preventing overfitting by focusing on the most critical features.

The network adeptly captures temporal relationships within input sequences through its utilization of LSTM layers. Consisting of five LSTM layers, this segment specializes in modeling long-range dependencies and sequential patterns within the data. Each LSTM unit within this set operates as a memory cell, adept at selectively retaining and updating information over time. By assembling multiple LSTM layers, the network acquires the capability to learn temporal representations, facilitating the extraction of temporal dynamics embedded within the input sequences.

The combination of CNN and LSTM within this classifier represents a fusion of spatial and temporal modeling capabilities. By leveraging the feature extraction proficiency of CNNs alongside the sequential modeling abilities of LSTMs, the network can excel across a spectrum of audio classification tasks. The accuracy of this audio classifier architecture is used to determine the performance of the PSC-MFCC and traditional Librosa MFCC across a variety of diverse environmental sound datasets. These datasets include unique environmental sound sources that are collected in distinct settings.

4 Experimental setup

This section discusses the datasets used to evaluate the Pertinent Spectral Characteristic (PSC-MFCC) approach.

4.1 Dataset overview

Three different environmental sound datasets are explored. While using these datasets, 80% of the data is used for training the NN and 20% of the data is used for testing the NN. These datasets are ESC50, UrbanSound8k, and Gunshot. The three datasets have distinct sound sources and a different number of classes. Finding the classification accuracy of each of the three datasets using the PSC-MFCC method and traditional Librosa MFCC method yields perspective on the advantages of the PSC-MFCC over the traditional Librosa MFCC.

4.1.1 ESC50 dataset

The first dataset that is implemented is ESC50. ESC50 is an environmental sound dataset with 50 classes. Each of these classes refers to a different sound source. Thus, there are 50 different sound sources in the dataset. Having 50 different classes contributes to the diversity of the dataset, making it a tough challenge for classification. Each of the 50 classes has 40 audio files. These audio files are WAV files and have a duration of 5 seconds. Therefore there are a total of 2,000 audio files holding 10,000s of data. The 50 different classes in the dataset include animal sounds, vehicle sounds, and weather sounds[11].

4.1.2 UrbanSound8k dataset

The second dataset used is UrbanSound8k. This dataset has 8,732 audio files and ten different classes. The sounds in this dataset stem from sources that are often found in urban environments. The ten classes present in the dataset are air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. Together these WAV files encompass the diverse audio heard in urban settings[12].

4.1.3 Gunshot dataset

The concluding dataset under analysis is the Gunshot dataset, comprising 2s long sounds of gunfire encountered both in wilderness and urban settings. This dataset encompasses 9 distinct classes, each representing the unique sound produced by a different model of firearm upon discharge. The sound sources in the dataset are AK-12, AK-47, IMI Desert Eagle, M16, M249, M4, MG-42, MP5, and Zastava M92. Between all of the classes their are 851 audio files. These classes offer a comprehensive exploration of firearm sounds across diverse environments[13].

4.2 Programming languages and Libraries

Python is used to develop the components and algorithms used in this project. Python is a high-level, general-purpose programming language. There are many Python libraries that are utilized in this project. Some of the key libraries used in this project are NumPy, PyTorch, Librosa, Matplotlib, SciPy, and spafe. In addition to Python, MATLAB is used to calculate metrics such as accuracy and standard deviation which are listed in the results section.

4.3 Evaluation paramaters

In order to evaluate the performance of the traditional Librosa MFCC function and the PSC-MFCC function on different datasets an audio classifier is required. The audio classifier takes in the MFCC of an audio signal and uses them to determine which predefined class best suits the signal. The accuracy of the classifier serves as a metric of performance. Comparing the accuracy of the audio classifier using the PSC-MFCC to the accuracy of the classifier using the traditional Librosa MFCC provides insights into the benefits of the custom filterbank. This classification is performed through a Deep Neural Network (DNN). When using the DNN certain parameters are used. The ESC50 and Gunshot dataset are run for 300 epochs, while the UrbanSound8k is only run for 40 epochs. For each of the datasets, the batch size is set to 64. The results found using these paramaters are outlined in the next section.

5 Results

This section discusses the results of the Pertinent Spectral Characteristic (PSC-MFCC) approach.

5.1 Results overview

Once the methods are completed, they yield a specialized MFCC audio classifier. This classifier is crafted using a dataset-specific filterbank tailored by the filterbank optimization algorithm. Employing this custom filterbank, the PSC-MFCC function computes the features. Subsequently, the audio classifier leverages these custom MFCC features to classify a dataset, facilitating an evaluation of the customized MFCC approach's performance.The traditional Librosa MFCC undergoes assessment using the same DNN and datasets as the PSC-MFCC approach. This comparative analysis identifies any performance disparities between the PSC-MFCC and the traditional Librosa MFCC approaches.

For each dataset, both the PSC-MFCC approach and the traditional Librosa MFCC approach undergo training and testing iteratively for 5 runs. The reported overall accuracy for each dataset is derived from the average test accuracy across these five runs. This iterative process of testing and training five times serves to mitigate outliers and enhance the precision of the results. The precision of the results is represented by the standard deviation of the 5 runs.

5.2 ESC50

In the initial dataset, ESC50, the PSC-MFCC approach achieved an accuracy of 46%, surpassing the traditional Librosa MFCC approach, which attained 44% accuracy (Figure 4). Additionally, the PSC-MFCC method exhibited a lower standard deviation of 2.7%

compared to 6.9% for the traditional Librosa MFCC approach. The enhanced efficacy of the PSC-MFCC approach over the traditional Librosa MFCC approach can be attributed to the dataset's diversity. With ESC50 encompassing 50 distinct sound sources, each with its unique frequency range, employing a filterbank tailored specifically to this dataset enables comprehensive coverage of important frequency ranges. This tailored approach significantly boosts the classifier's performance by ensuring the capture of all pertinent acoustic features.

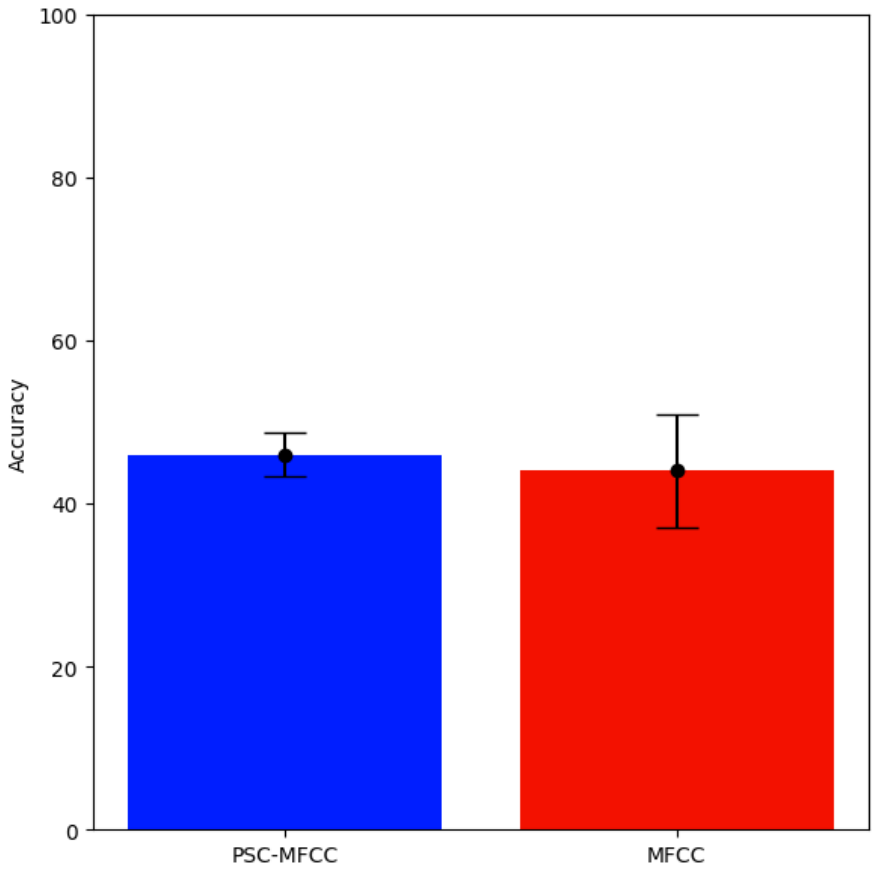

Figure 4: ESC50 accuracy

5.3 UrbanSound8k

Moving on to the UrbanSound8k dataset, the PSC-MFCC technique demonstrated a significant performance improvement with an accuracy of 78.4%, in contrast to the 69.2%

accuracy of the traditional Librosa MFCC method (Figure 5). The custom approach also showcased a slightly higher standard deviating of 4.8% compared to 4.0% for the traditional Librosa MFCC approach. While the precision of the PSC-MFCC approach may exhibit a slight decline compared to the traditional Librosa MFCC approach, the overall accuracy significantly surpasses it. This discrepancy arises from the substantial variations in frequency ranges across different classes. Despite the minor reduction in precision, the notable enhancement in accuracy underscores the effectiveness of the PSC-MFCC method in accurately capturing and distinguishing features within diverse frequency ranges across classes.

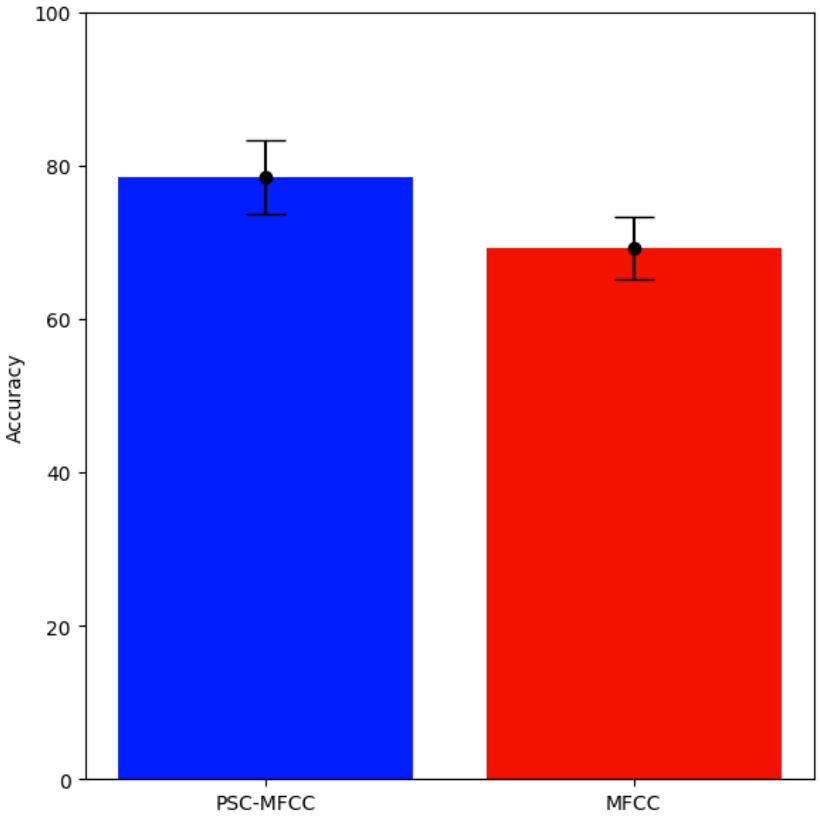

Figure 5: UrbanSound8k accuracy

5.4 Gunshot

Lastly, in the Gunshot dataset, the PSC-MFCC approach achieved a respectable accuracy of 63.8%, slightly trailing behind the 65% accuracy of the traditional Librosa MFCC

approach (Figure 6). However, the PSC-MFCC approach had a lower standard deviation of 0.4% compared to 1.4% for the traditional Librosa MFCC method. There is a minor decrease in accuracy but a marked improvement in precision when comparing the PSC-MFCC approach to the traditional Librosa MFCC approach. This distinction can be attributed to the fact that the Gunshot dataset has lower frequencies that are adequately captured by the standard Mel-filterbank used in the traditional method.

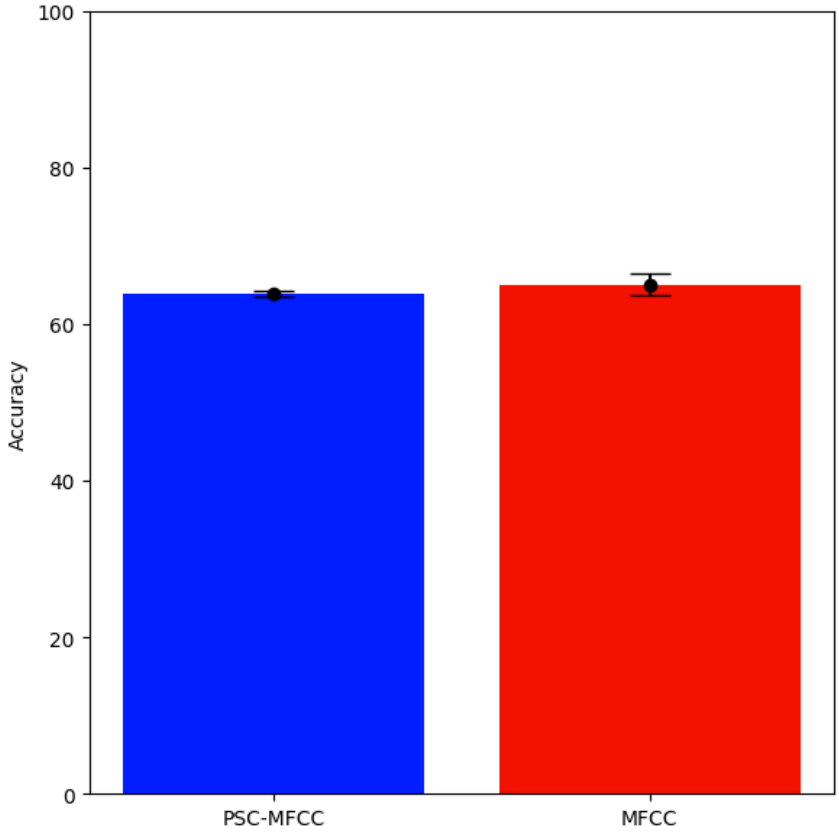

Figure 6: Gunshot accuracy

5.5 Simplified classifier

In addition to running the results on the classifier described earlier, a simplified model is also implemented. The simplified model follows the same structure as the original model, however the LSTM portion is removed. This is done to understand the impact of the PSC-MFCC on a less complicated classifier. The results of the simplified model are displayed

in Figure 7. When using a simplified model, the PSC-MFCC demonstrates a significant improvement in performance over the traditional Librosa MFCC. Across the three datasets, the PSC-MFCC offers an average accuracy increase of 5.67% compared to the traditional Librosa MFCC.

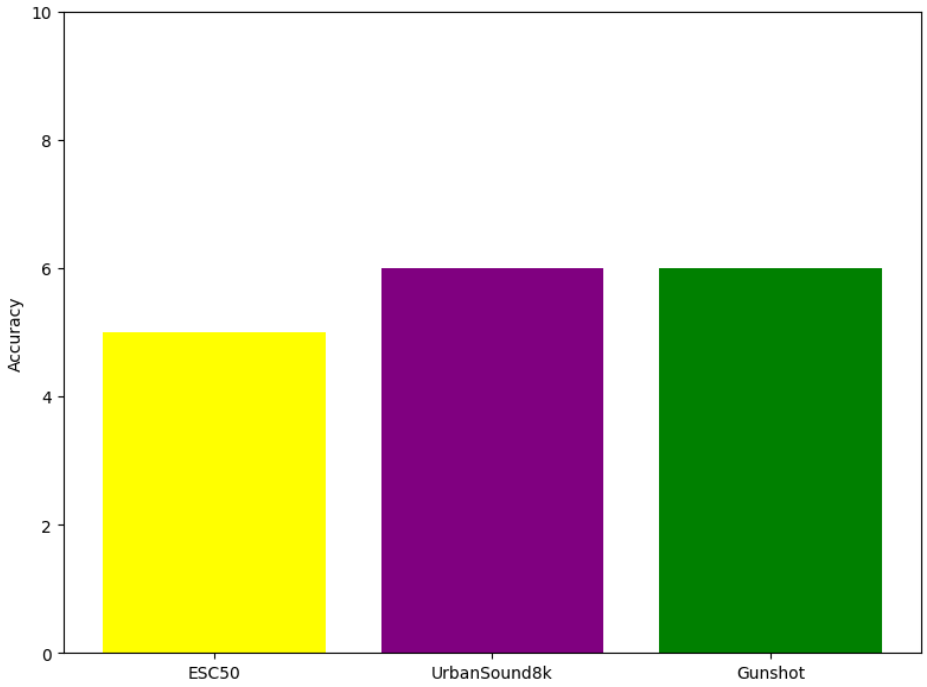

Figure 7: Accuracy difference between PSC-MFCC and MFCC on less complex classifier

5.6 Results summary

In general, the PSC-MFCC approach outperformed the traditional Librosa MFCC approach, showcasing an average increase of 4.5% in accuracy and a smaller average standard deviation of 1.47%. This indicates a significant enhancement in both accuracy and precision of classification tasks when employing the PSC-MFCC approach compared to the traditional Librosa MFCC approach on environmental datasets.

6 Limitations and Future Work

This section discusses the limitations of the PSC-MFCC approach as well as future areas of research. While the PSC-MFCC approach enhances classification accuracy and precision, it is not without its limitations. These drawbacks encompass susceptibility to low Signal Noise Ratio (SNR), balancing performance and cost, consistency in performance, additional computational demands, and extended processing times.

6.1 Low SNR

One constraint associated with the PSC-MFCC method is its susceptibility to the Signal Noise Ratio (SNR). In scenarios where an audio signal exhibits a low SNR, the noise power may overshadow the signal power. Given that the PSC-MFCC approach positions bandpass filters at frequencies with high power, in low SNR conditions, these filters may inadvertently capture spectral noise while overlooking the actual signal. To address this challenge, it becomes imperative to establish predominant frequencies based on criteria beyond merely identifying peaks in the frequency power response. One potential strategy involves comparing the periodograms of distinct classes within the dataset to pinpoint frequencies where the power response exhibits the greatest disparity. Subsequently, these identified frequencies can serve as the predominant frequencies guiding the placement of triangular band pass filters within the custom filterbank.

6.2 Balancing performance and cost

Another limitation concerns striking a balance between performance and cost. While the PSC-MFCC approach has superior accuracy and precision compared to the traditional Librosa MFCC method, it necessitates additional computations and longer processing

times. Therefore, while the PSC-MFCC approach offers enhanced performance, it also demands a greater allocation of resources, presenting a trade-off between performance gains and resource consumption.

6.3 Performance consistency

A third limitation pertains to performance consistency. While, on average, the PSC-MFCC approach outperformed the traditional Librosa MFCC method, there are instances where its performance deviates. For instance, in the Gunshot dataset, the PSC-MFCC approach exhibited slightly lower accuracy compared to the traditional method. Additionally, when assessed on the Urbansound8k dataset, the PSC-MFCC approach displayed a higher standard deviation than its traditional counterpart. Despite generally yielding superior results, the PSC-MFCC approach does not consistently outperform the traditional method across all datasets and scenarios.

6.4 Additional resources

When employing the PSC-MFCC approach, additional computations are necessary to derive the custom filterbank. In contrast, traditional Librosa MFCC utilizes a predefined filter bank following the Mel-scale, independent of the dataset. The heightened computational demand of the PSC-MFCC approach translates to longer processing times compared to the traditional method. Consequently, the PSC-MFCC approach necessitates greater resource allocation compared to the traditional Librosa MFCC approach.

7 Conclusion

Audio classification plays a crucial role in environmental monitoring, enabling automated categorization of audio data for various applications such as wildlife monitoring and urban noise assessment. The conventional audio classification method relies on a fixed filterbank structure which leads to limitations in adaptability. We propose a PSC-MFCC approach that aligns the filerbank with dataset-specific characteristics. In doing so we achieve an 4.5% increase in accuracy across various environmental sound datasets compared to the traditional method. This highlights its potential to enhance audio classification.

References

[1] K. J. Piczak, "Esc," *Proceedings of the 23rd ACM international conference on Multimedia*, 10 2015.

[2] A. E. Mehyadin, A. M. Abdulazeez, D. A. Hasan, and J. N. Saeed, "Birds sound classification based on machine learning algorithms," *Asian Journal of Research in Computer Science*, pp. 1–11, 06 2021.

[3] J. Jun and D. P. Agrawal, "Optimum detection probability with partially controlled random deployment of wireless sensors with mobile base stations," *Lecture notes in electrical engineering*, pp. 1–10, 01 2014.

[4] P. Maijala, Z. Shuyang, T. Heittola, and T. Virtanen, "Environmental noise monitoring using source classification in sensors," *Applied Acoustics*, vol. 129, pp. 258–267, 01 2018.

[5] A. O. Ekpezu, I. Wiafe, F. Katsriku, and W. Yaokumah, "Using deep learning for acoustic event classification: The case of natural disasters," *The Journal of the Acoustical Society of America*, vol. 149, pp. 2926–2935, 04 2021.

[6] M. Russo, V. F. Puglisi, R. Avanzato, and F. Beritelli, "A cnn-based audio sensor for rainfall estimation: Implementation on embedded board," *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 09 2021.

[7] "librosa.feature.spectral — librosa 0.10.1 documentation."

[8] "librosa.stft — librosa 0.8.0 documentation."

[9] "librosa.feature.melspectrogram — librosa 0.8.0 documentation."

[10] "scipy.fftpack.dct — scipy v1.7.1 manual."

[11] "Esc50-dataset."

[12] C. Gorgolewski, "Urbansound8k," 2014.

[13] "Gunshot audio dataset."

[14] Mathuranathan, "Power and energy of a signal : Demystified," 12 2013.

[15] "Short-time fourier transform - an overview | sciencedirect topics."

[16] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, p. 90–93, 01 1974.

[17] P. Suman, S. Karan, V. Singh, and R. Maringanti, "Algorithm for gunshot detection using mel-frequency cepstrum coefficients (mfcc)," *Lecture Notes in Electrical Engineering*, pp. 155–166, 2014.