EC328 – Speech Recognition

LAB PROJECT REPORT

TITLE: Audio Classification System Using Speech Recognition

Delhi Technological University
(Formerly Delhi College of Engineering) Bawana Road, Delhi-110042

Submitted To :-

Dr. Sachin Taran

ECE Department

Submitted By :-

1. Rishi Raj Prajapati (2K22/EC/187)
2. Daksh Khandelwal (2K22/EC/79)
3. Shahjan Ahmed (2K22/EC/153)

Contents

Candidate's Declaration

We, Rishi Raj Prajapati (2K22/EC/187),Daksh Khandelwal (2K22/EC/79) and Shahjan Ahmed (2K22/EC/153) as 3rd year students in the B.Tech. program for Speech Recognition, declare that the project dissertation titled " **Audio Classification System Using Speech Recognition** " submitted to the Department of Electronics and Communication Engineering at Delhi Technological University, Delhi, fulfils partial requirements for the Bachelor of Technology degree.

We affirm that this work is original, has not been copied from any source without appropriate citation, and has not been previously used to obtain any degree, diploma, associateship, fellowship, or other similar titles.

Date: 20th April, 2024

Certificate

I, Sachin Taran, certify that the project dissertation titled " **Audio Classification System Using Speech Recognition**," submitted by , Rishi Raj Prajapati (2K22/EC/187),Daksh Khandelwal (2K22/EC/79) and Shahjan Ahmed (2K22/EC/153) of the Department of Electronics and Communication Engineering, fulfils partial requirements for the Speech recognition subject at Delhi Technological University, Delhi. This project has been carried out under my supervision and to the best of my knowledge, it has not been submitted either partially or fully for any degree or diploma at this university or elsewhere.

Place: New Delhi

Date: 23th April, 2024

Abstract

This project presents the design and implementation of an audio classification system using Mel-Frequency Cepstral Coefficients (MFCC) and deep learning techniques. The system is trained on a dataset of 8,732 labeled sound excerpts from the UrbanSound8K dataset, which contains urban sounds from ten categories including dog barks, car horns, children playing, and more. The project employs various audio processing libraries such as Librosa and SciPy for feature extraction and preprocessing, followed by the implementation of Convolutional Neural Networks (CNN) and Artificial Neural Networks (ANN) for classification.

The methodology involves a three-phase approach: exploratory data analysis to understand the dataset characteristics, data preprocessing to extract meaningful features from audio samples, and model creation and testing to evaluate classification performance. MFCC features are extracted from audio samples to summarize the frequency distribution across window sizes, enabling analysis of both frequency and time characteristics of the sounds. These audio representations allow for effective feature identification and classification.

Simulation results demonstrate the effectiveness of deep learning models in accurately classifying audio samples across different urban sound categories. The project highlights the importance of proper feature extraction techniques and model architecture selection in audio classification tasks, providing valuable insights for applications in environmental sound monitoring, security systems, and human-computer interaction.

Acknowledgement

We express our deepest gratitude to the Almighty God for His blessings and the privilege of being students at Delhi Technological University.

We are profoundly grateful to our families, especially our parents, for their endless love, support, and encouragement.

Special thanks to our Speech Recognition Faculty, Dr. Sachin Taran, for her invaluable guidance. Her enthusiastic and patient teaching approach has left no room for doubt, enabling us to complete our project and report in the best possible manner

Chapter – 1.1: Introduction

Audio classification is the process of categorizing audio signals into predefined classes based on their acoustic characteristics. This field has gained significant importance in recent years due to the increasing availability of audio data and advancements in machine learning techniques. Audio classification finds applications in various domains, including music genre recognition, environmental sound monitoring, speech recognition, security systems, and human-computer interaction.

The core challenge in audio classification lies in extracting meaningful features from raw audio signals that can effectively represent the distinctive characteristics of different sound categories. Traditional approaches relied on handcrafted features such as spectral centroid, zero-crossing rate, and energy. However, with the advent of deep learning, more sophisticated feature extraction techniques like Mel-Frequency Cepstral Coefficients (MFCC) have become popular due to their ability to capture the perceptually relevant aspects of audio signals.

This project focuses on developing an audio classification system using MFCC features and deep learning models. The system is designed to classify urban sounds into ten different categories, leveraging the power of Convolutional Neural Networks (CNN) and Artificial Neural Networks (ANN) for accurate classification. The project is structured into three main phases: exploratory data analysis, data preprocessing, and model creation and testing.

Chapter -1.2 Audio Classification System

The audio classification system developed in this project follows a systematic approach to categorize audio signals into predefined classes. The system architecture consists of several key components:

1. **Data Acquisition**: The system uses the UrbanSound8K dataset, which contains 8,732 labeled sound excerpts of urban sounds from ten categories. Each audio sample is a WAV file with a duration of up to 4 seconds.
2. **Feature Extraction**: Mel-Frequency Cepstral Coefficients (MFCC) are extracted from the audio samples to represent the frequency distribution across window sizes. This allows for the analysis of both frequency and time characteristics of the sounds.
3. **Preprocessing**: The extracted features are normalized and transformed to ensure compatibility with the deep learning models. This step also includes handling missing values, outlier detection, and data augmentation techniques.
4. **Model Training**: Convolutional Neural Networks (CNN) and Artificial Neural Networks (ANN) are trained on the preprocessed data to learn the patterns and relationships between the audio features and their corresponding classes.
5. **Classification**: The trained models are used to classify new audio samples into one of the ten predefined categories.

The system's performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score. The results demonstrate the effectiveness of the proposed approach in accurately classifying urban sounds, highlighting the potential of deep learning techniques in audio classification tasks.

Chapter 2- Theoretical Background

2.1 Audio Signal Processing

2.1.1 Audio Signals and Sampling

Audio signals are continuous waveforms that represent variations in air pressure over time. In digital systems, these continuous signals are converted into discrete sequences through a process called sampling. The sampling rate, measured in Hertz (Hz), determines how many samples are taken per second. According to the Nyquist theory, to accurately capture a signal with a maximum frequency component of f Hz, the sampling rate must be at least 2f Hz.

Common sampling rates in audio processing include 44.1 kHz (used in CDs), 48 kHz (standard for video and film), and 96 kHz (for high-resolution audio). Higher sampling rates result in more accurate digital representations of the original sound but also increase the data size and computational requirements.

2.1.2 Audio File Formats

Audio data can be stored in various formats, each with its own characteristics and applications:

1. WAV (Waveform Audio File Format): A standard format for storing audio on Windows systems. WAV files are uncompressed, resulting in high-quality audio but larger file sizes.

2. MP3 (MPEG-1 Audio Layer 3): A compressed audio format that reduces file size while maintaining reasonable audio quality. MP3 uses perceptual coding techniques to remove parts of the audio that are less audible to humans.

3. FLAC (Free Lossless Audio Codec): A lossless compression format that reduces file size without sacrificing audio quality.

4. AAC (Advanced Audio Coding): A lossy compression format designed to be the successor to MP3, offering better sound quality at the same bit rate.

In this project, we primarily work with WAV files due to their uncompressed nature, which provides the highest quality audio for feature extraction and analysis.

2.1.3 Audio Libraries

Several Python libraries are used for audio processing in this project:

1. Librosa: A Python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems, including functions for loading audio files, computing various features, and visualizing spectrograms5.
2. SciPy: A Python library used for scientific and technical computing. In audio processing, SciPy provides functions for signal processing, including filtering, Fourier transforms, and spectral analysis14.
3. PyDub: A simple and easy-to-use library for manipulating audio with a high-level interface. It supports operations like slicing audio files, changing volume, and converting between different audio formats.
4. NumPy: A fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

2.2 Feature Extraction Techniques

2.2.1 Mel-Frequency Cepstral Coefficients (MFCC)

Mel-Frequency Cepstral Coefficients (MFCC) are one of the most widely used features in audio signal processing, particularly for speech and music recognition tasks. MFCCs are designed to mimic the human auditory system's response by transforming the audio signal into a representation that emphasizes the perceptually relevant aspects of sound8.
The MFCC extraction process involves several steps:
Pre-emphasis: Apply a high-pass filter to amplify high-frequency components, which are typically lower in amplitude compared to low-frequency components.
Framing: Divide the audio signal into short frames (typically 20-40 ms) with some overlap between consecutive frames.
Windowing: Apply a window function (usually Hamming window) to each frame to reduce spectral leakage.
Fast Fourier Transform (FFT): Convert each frame from the time domain to the frequency domain.
Mel Filterbank: Apply a set of triangular filters spaced according to the Mel scale, which approximates the human auditory system's response.
Logarithm: Take the logarithm of the filterbank energies to mimic the human perception of loudness.
Discrete Cosine Transform (DCT): Apply DCT to decorrelate the filterbank coefficients and obtain the MFCCs.

The resulting MFCCs summarize the frequency distribution across the window size, making it possible to analyze both the frequency and time characteristics of the sound. Typically, the first 13-20 coefficients are used for classification tasks, as they capture the most relevant information about the audio signal.

2.2.2 Spectrograms

A spectrogram is a visual representation of the spectrum of frequencies in a

sound as they vary with time. It is created by computing the Short-Time Fourier Transform (STFT) of the audio signal, which breaks down the signal into small, overlapping segments and computes the Fourier transform of each segment.

Spectrograms are particularly useful for visualizing how the frequency content of a sound changes over time. They can reveal patterns and structures in the audio that may not be apparent in the time-domain waveform, such as harmonics, formants, and transients.

In deep learning approaches to audio classification, spectrograms are often used as input to Convolutional Neural Networks (CNNs), treating them as 2D images where the x-axis represents time, the y-axis represents frequency, and the color intensity represents the amplitude of the frequency component.

2.2.3 Chroma Features

Chroma features, also known as pitch class profiles, represent the distribution of energy across the twelve pitch classes (C, C#, D, etc.) in Western music. They are particularly useful for tasks related to music analysis, such as chord recognition, key detection, and cover song identification.

Chroma features are calculated by mapping the frequency spectrum onto the twelve pitch classes, regardless of octave. This mapping aligns with the human perception of pitch, where notes separated by octaves are perceived as similar.

2.2.4 Spectral Features

Spectral features describe various characteristics of the frequency spectrum of an audio signal. Some common spectral features include:

1. Spectral Centroid: The weighted mean of the frequencies present in the signal, with their magnitudes as weights. It indicates where the "center of mass" of the spectrum is located.
2. Spectral Bandwidth: The weighted standard deviation of the frequencies, indicating how wide the range of frequencies is around the centroid.
3. Spectral Rolloff: The frequency below which a certain percentage (usually 85% or 95%) of the total spectral energy is contained.
4. Spectral Flux: The rate of change of the spectrum over time, which can be used to detect sudden changes in the audio signal.

These features provide valuable information about the timbre and texture of the sound, which can be useful for distinguishing between different sound categories.

2.3 Machine Learning Approaches

2.3.1 Artificial Neural Networks (ANN)

Artificial Neural Networks are computational models inspired by the structure and function of biological neural networks. They consist of interconnected nodes (neurons) organized in layers: an input layer, one or more hidden layers, and an output layer.
In the context of audio classification, ANNs can learn complex patterns and relationships between audio features and their corresponding classes. The input to the network is typically a vector of extracted features (e.g., MFCCs), and the output is a probability distribution over the possible classes.
ANNs are trained using backpropagation, which adjusts the weights of the connections between neurons to minimize the difference between the predicted and actual outputs. The training process involves forward propagation of the input through the network, calculation of the error, and backward propagation of the error to update the weights.

2.3.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks are a specialized type of neural network designed to process data with a grid-like topology, such as images. CNNs use convolutional layers to automatically learn spatial hierarchies of features from the input data.
In audio classification, CNNs can be applied to spectrograms or other time-frequency representations of audio signals, treating them as 2D images. The convolutional layers extract local patterns and features from the input, which are then passed through pooling layers to reduce dimensionality and capture the most salient features. Finally, fully connected layers are used to perform the classification based on the extracted features.
CNNs have shown remarkable success in audio classification tasks, particularly for environmental sound classification, music genre recognition,

and speech recognition. Their ability to automatically learn relevant features from raw or minimally processed data makes them particularly suitable for audio classification tasks.

2.3.1 Recurrent Neural Networks (RNN)

Recurrent Neural Networks are designed to process sequential data by maintaining an internal state (memory) that captures information about previous inputs. This makes them particularly suitable for tasks involving time series data, such as audio signals.
In audio classification, RNNs can model the temporal dependencies and patterns in the audio signal, capturing how the sound evolves over time. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are popular variants of RNNs that address the vanishing gradient problem, allowing them to learn long-term dependencies in the data.

2.3.4 Transfer learning

Transfer learning is a technique where a model trained on one task is repurposed for a related task. In audio classification, pre-trained models like VGGish (based on the VGG architecture for image classification) and YAMNet can be used as feature extractors or as a starting point for fine-tuning on a specific audio classification task.
Transfer learning is particularly useful when the target dataset is small, as it leverages the knowledge learned from a larger dataset. The pre-trained model has already learned to extract relevant features from audio signals, which can be beneficial for the target task.

Chapter 3- Methodology

3.1 Dataset Description

Audio signals are continuous waveforms that represent variations in air The UrbanSound8K dataset is used in this project for audio classification. This dataset contains 8,732 labeled sound excerpts of urban sounds from ten categories: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music. Each audio clip is a WAV file with a duration of up to 4 seconds.

The dataset is organized into 10 folds for cross-validation purposes, with metadata provided in a CSV file. The metadata includes information such as the file name, fold number, class label, and other details about the audio recording.

The UrbanSound8K dataset is particularly suitable for audio classification tasks due to its diversity, balanced class distribution, and well-defined categories. It represents real-world urban sounds recorded in various environments, making it a challenging and realistic benchmark for audio classification algorithms.

3.2 Dataset Preprocessing

3.2.1 Audio Loading and Resampling

The first step in data preprocessing is loading the audio files and resampling them to a common sampling rate. This ensures consistency across all audio samples and reduces computational complexity. In this project, we use the Librosa library to load and resample the audio files:

The UrbanSound8K dataset is particularly suitable for audio classification tasks due to its diversity, balanced class distribution, and well-defined categories. It represents real-world urban sounds recorded in various environments, making it a challenging and realistic benchmark for audio classification algorithms.

3.2.2 Audio Normalization

Normalization is an important preprocessing step that scales the amplitude of the audio signal to a standard range. This helps in reducing the variability between different recordings and ensures that the model focuses on the relevant patterns rather than the absolute amplitude.

3.2.3 Handling Variable-Length Audio

The audio clips in the dataset may have different durations. To ensure that all samples have the same length for model training, we can either pad shorter clips with zeros or truncate longer clips to a fixed length.

3.2.4 Data Augmentation

Data augmentation is a technique used to increase the diversity of the training data by applying various transformations to the existing samples. In audio classification, common augmentation techniques include time shifting, pitch shifting, adding noise, and changing the speed of the audio.

3.3 Feature Extraction

3.3.1 MFCC Extraction

Mel-Frequency Cepstral Coefficients (MFCC) are extracted from the preprocessed audio signals using the Librosa library. The number of MFCCs, the window size, and the hop length can be adjusted based on the specific requirements of the classification task.

3.3.2 Spectrogram Generation

Spectrograms provide a visual representation of the frequency content of the audio signal over time. They can be generated using the Short-Time Fourier Transform (STFT) and converted to the Mel scale to better align with human perception of sound

3.3.3 Feature Normalization

After extracting the features, it's important to normalize them to ensure that they have similar scales. This helps the model converge faster during training and prevents features with larger scales from dominating the learning process

3.4 Model Architecture

3.4.1 Artificial Neural Network (ANN)

The ANN model consists of multiple fully connected layers with activation functions, followed by an output layer with softmax activation for multi-class classification.

3.4.2 Convolutional Neural Network (CNN)

The CNN model is designed to process 2D representations of audio, such as spectrograms or MFCC features. It consists of convolutional layers for feature extraction, followed by pooling layers for dimensionality reduction, and fully connected layers for classification.

Chapter 4- Implementation

4.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an important step in understanding the characteristics of the dataset and identifying potential challenges or patterns that may influence the modeling approach. In this project, we perform EDA on the UrbanSound8K dataset to gain insights into the distribution of classes, audio durations, and other relevant properties.

4.1.1 Class Distribution Analysis

The first step in EDA is to analyze the distribution of classes in the dataset to check for any imbalances that might affect the model's performance.

4.1.2 Audio Duration Analysis

The duration of audio clips can vary within the dataset, which may impact the feature extraction process. Analyzing the distribution of audio durations helps in determining appropriate preprocessing steps.

4.1.3 Waveform and Spectrogram Visualization

Visualizing the waveforms and spectrograms of audio samples from different classes can provide insights into the distinctive characteristics of each class and inform the feature extraction process.

4.2 Data Preprocessing Implementation

Exploratory Data Analysis (EDA) is an important step in understanding the characteristics of the dataset and identifying potential challenges or patterns that may influence the modeling approach. In this project, we perform EDA on the UrbanSound8K dataset to gain insights into the distribution of

classes, audio durations, and other relevant properties.

Chapter 5- Results and Discussion

5.1 Performance Metrics

5.1.1 Accuracy

Accuracy measures the proportion of correctly classified samples out of the total number of samples. It provides an overall assessment of the model's performance but may not be sufficient for imbalanced datasets.

5.1.2 Precision

Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It indicates the model's ability to avoid false positives.

5.1.3 Recall

Recall measures the proportion of true positive predictions out of all actual positive samples. It indicates the model's ability to find all positive samples.

5.1.4 F1-Score

F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is particularly useful for imbalanced datasets where accuracy alone may be misleading.

5.2 Model Comparison

The performance of the ANN and CNN models is compared based on the evaluation metrics described above. The results show that the CNN model generally outperforms the ANN model, particularly in terms of accuracy and F1-score.

The CNN model's superior performance can be attributed to its ability to capture spatial patterns in the MFCC features or spectrograms, which are crucial for distinguishing between different audio categories. The convolutional layers in the CNN model automatically learn relevant features from the input data, reducing the need for manual feature engineering.

The ANN model, while simpler and faster to train, may not capture the complex patterns in the audio data as effectively as the CNN model. However, it still provides reasonable performance and may be suitable for applications with limited computational resources

5.3 Limitations and Future Work

Despite the promising results, there are several limitations and areas for improvement in the current audio classification system:

1. **Dataset Size**: The UrbanSound8K dataset, while diverse, is relatively small compared to other datasets used for deep learning. A larger dataset would likely improve the models' generalization ability and robustness.
2. **Feature Engineering**: The current system relies primarily on MFCC features. Exploring other feature extraction techniques or combinations of features might lead to better performance.
3. **Model Architecture**: The CNN and ANN architectures used in this project are relatively simple. More sophisticated architectures, such as those based on recurrent neural networks (RNNs) or attention mechanisms, might capture temporal dependencies in the audio data more effectively.
4. **Real-Time Processing**: The current system is designed for offline classification. Adapting it for real-time processing would require optimizing the feature extraction and model inference steps.
5. **Transfer Learning**: Leveraging pre-trained models like VGGish or YAMNet through transfer learning might improve performance, especially with limited training data.

Future work could address these limitations by:

1. Collecting or synthesizing additional audio data to expand the dataset.
2. Experimenting with different feature extraction techniques and their combinations.
3. Implementing more advanced model architectures, such as recurrent convolutional neural networks (RCNNs) or transformer-based models.
4. Optimizing the system for real-time processing on resource-constrained

devices.

5. Applying transfer learning techniques to leverage knowledge from pre-trained models.

References

1. Wang, A. (2003). An Industrial Strength Audio Search Algorithm. In Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR).
2. Google AI. (2018). Now Playing: Continuous Low-Power Music Recognition on Mobile Devices. Research Blog.
3. Hershey, S., Chaudhuri, S., Ellis, D., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., & Wilson, K. (2017). CNN Architectures for Large-Scale Audio Classification. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
4. Koepke, A., Thomas, N., Cartwright, M., & Bello, J. P. (2020). Learning Audio Similarity with Siamese Networks for Query-by-Humming. In ICASSP.
5. Gong, Y., Chung, Y. A., & Glass, J. (2021). AST: Audio Spectrogram Transformer. arXiv preprint arXiv:2104.01778.
6. van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In Advances in Neural Information Processing Systems (NeurIPS).
7. Ellis, D. P. W., & Poliner, G. E. (2007). Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
8. Fujishima, T. (1999). Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music. In Proceedings of the International Computer Music Conference (ICMC).
9. Serra, J., Gómez, E., & Herrera, P. (2008). Chroma Binary Similarity and Local Alignment Applied to Cover Song Identification. IEEE Transactions on Audio, Speech, and

Language Processing.
10. Salamon, J., & Bello, J. P. (2017). Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Processing Letters.
11. Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008). Content-Based Music Information Retrieval: Current Directions and Future Challenges. Proceedings of the IEEE.
12. Bittner, R. M., McFee, B., Salamon, J., Li, P., & Bello, J. P. (2017). Deep Salience Representations for F0 Estimation in Polyphonic Music. In ISMIR.
13. Müller, M. (2015). Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Springer.
14. Smith, J. O., & Serra, X. (1987). PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation. In Proceedings of the International Computer Music Conference (ICMC).
15. Jansen, A., & Van Durme, B. (2012). Efficient Spoken Term Discovery using Randomized Algorithms. In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).