Clearing the Noise: Unraveling the Secrets of Speech Recognition for Perfect Voice Commands

By

Maria Blinchevskaya,

Raphael Delouya,

Gadi G. Ezer,

Claudia Palierne

July 24, 2023

Table of Contents

Introduction

In today's tech-driven world, voice-activated systems and virtual assistants have become an integral part of our daily lives. Whether it's commanding smart devices, seeking information, or enjoying hands-free navigation, speech recognition technology plays a pivotal role in enabling seamless interactions between humans and machines. However, achieving accurate speech recognition in real-world environments, filled with diverse background noises, poses a significant challenge.

In this article, we embark on an exciting journey to explore the realm of speech recognition classification and delve into the intricacies of building a robust audio command classification model. We'll take on the TensorFlow Speech Recognition Challenge and work with the Speech Commands Dataset, comprising thousands of one-second audio clips representing various spoken commands.

Our ultimate goal is to develop an advanced model capable of accurately classifying voice commands, even in the presence of background noise. To attain this feat, we'll dive into the realms of audio signal processing and extract powerful features like Mel Frequency Cepstral Coefficients (MFCC) and Short-Time Fourier Transform (STFT). These audio representations offer valuable insights into the frequency content and temporal dynamics of speech signals, empowering our model to excel in diverse acoustic environments.

We begin our journey by dividing the dataset into distinct training, validation, and test sets. As we embark on this path, we'll examine the challenges and advantages of utilizing audio files without background noise to establish a baseline performance. Armed with this reference point, we'll venture further to tackle the complexity of background noise and witness the transformation of our model into a powerful tool for commanding the unseen acoustic landscape.

Join us as we unravel the secrets of decoding commands and embark on a mission to build a robust and reliable speech recognition model that unlocks the full potential of voice interactions in our digital age. Let's dive in and make voice commands clearer and more effortless than ever before!

The Challenge

Inspired by the TensorFlow Speech Recognition Challenge on Kaggle, we found ourselves presented with an exciting opportunity to explore the realm of speech recognition technology. Our task was to develop a robust algorithm for speech recognition classification using the provided Speech Commands Dataset. The dataset contained 6,500 audio files of approximately one-second audio clips, evenly distributed across thirteen categories. Our challenge was to accurately classify these diverse spoken commands into predefined categories, such as 'yes', 'no', 'one', 'two', 'up', 'down', and more.

The dataset's structure presented some unique challenges, as it lacked explicit separation into training, validation, and test sets. To overcome this, we employed a user-ID-based splitting strategy, ensuring that each dataset contained data from unique user-IDs without any overlap. This approach effectively minimized the risk of overfitting and reduced potential bias in our results, enabling us to create more generalized and robust models that perform exceptionally well in various scenarios.

With a clear objective in mind, our primary aim was to achieve robust audio command classification capable of handling background noise effectively. This required addressing complexities like varying linguistic patterns, diverse noise environments, and potential overlapping of voice data from different individuals across command categories.

To evaluate our models, we relied on the Multiclass Accuracy metric, measuring the average number of observations with the correct label. This provided a comprehensive assessment of the model's effectiveness in accurately classifying commands across all categories.

Excited by the potential impact of our work on voice interactions and the development of smart, user-friendly applications, we set forth on a captivating journey into the world of audio analysis and classification. With the vision of revolutionizing voice interactions for a better, more connected future, let's explore the path we undertook to decode commands and create a cutting-edge speech recognition system.

Exploring the Speech Commands Dataset

In our quest to develop a robust speech recognition classification model, we delved into the rich and diverse Speech Commands Dataset. This dataset, designed for the TensorFlow Speech Recognition Challenge, served as the foundation for our audio analysis journey. Let's take a closer look at the dataset and the valuable insights it provided us.

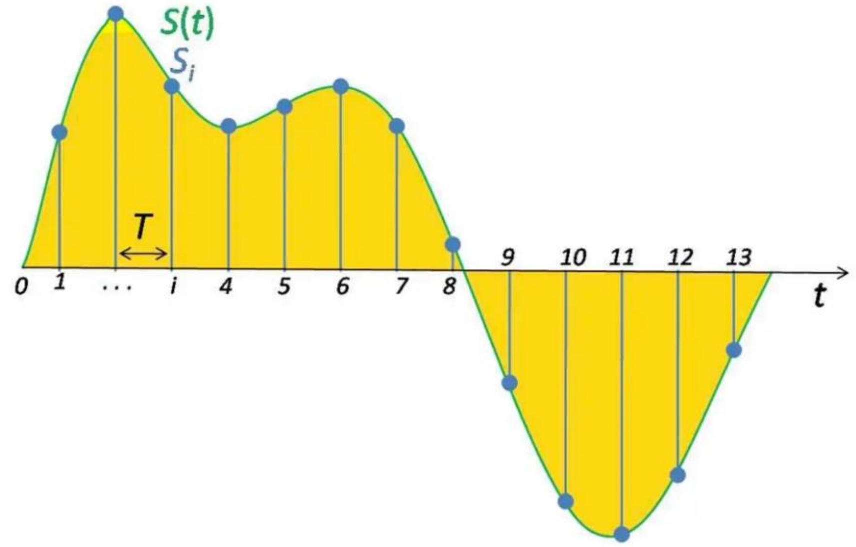


***Figure 1**: The process of digitizing a sound wave involves converting the signal into a sequence of numerical values, enabling its input into our models. This conversion is achieved by measuring the sound's amplitude at regular time intervals. These individual measurements are known as samples, and the sample rate denotes the number of samples taken per second. For example, a commonly used sample rate is approximately 44,100 samples per second, resulting in a 10-second music clip containing 441,000 samples.*

The Speech Commands Dataset consists of 6,500 audio files, each capturing one-second audio clips containing spoken English words. These audio clips are distributed evenly across thirteen categories, including commonly used commands like 'yes', 'no', 'one', 'two', 'up', 'down', and others. As we explored the dataset, we found an intriguing aspect that heightened the challenge - there was no explicit separation of the data into training, validation, and test sets.

To overcome this challenge, we adopted a user-ID-based splitting strategy. This approach ensured that audio data from each unique user-ID was placed in one specific dataset, preventing any overlap and potential bias in our results. This careful partitioning allowed us to create more generalized and reliable models, ready to tackle the complexities of real-world acoustic environments.

As we continued our exploration, we discovered that the dataset offered a unique opportunity to incorporate background noise into our audio samples. We sourced six diverse background noise audio files, including sounds of doing dishes, a meowing cat, an exercise bike, pink noise, a running tap, and white noise. Adding background noise to our data allowed us to create a more realistic and challenging environment for our models, ensuring their adaptability to handle real-world scenarios.

Furthermore, the dataset presented us with 1,814 different user-IDs, each uniquely assigned to individual voices across various categories. This diversity introduced additional complexity, as different categories may share the same user-ID, and each category may have multiple records from the same user-ID. Our model's ability to accurately classify commands despite these variations was a crucial aspect of our exploration.

To tackle the speech recognition classification problem, we extracted key features from the audio clips to form the basis of our model input. We explored two widely used techniques: Mel Frequency Cepstral Coefficients (MFCC) and Short-Time Fourier Transform (STFT). MFCC captures the essential frequency components of the audio, while STFT represents a time-frequency spectrogram. Evaluating the performance of these techniques allowed us to determine the most suitable feature representation for our model.

With a deeper understanding of the Speech Commands Dataset, its intricacies, and the potential challenges it presented, we were well-equipped to embark on the next phase of our audio analysis journey. Armed with the knowledge gained from exploring the dataset, we ventured into the realm of model development, seeking to create a cutting-edge speech recognition system capable of decoding commands accurately in diverse acoustic environments. Let's dive into the world of model building and decoding audio commands enriched with background noise.

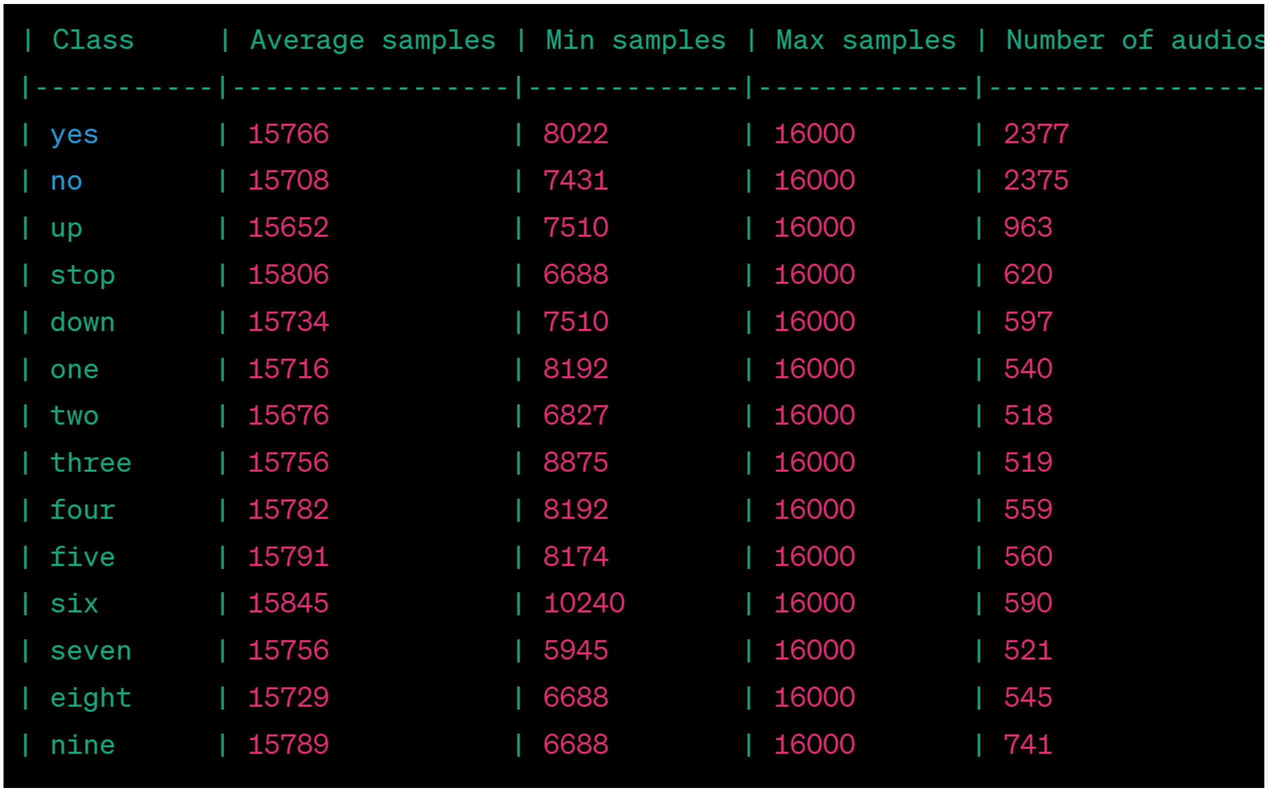
Preprocessing and Data Splitting

As we ventured into building a robust speech recognition classification system, the significance of preprocessing and data splitting became evident. Effective data preparation and partitioning formed the bedrock for the success of our model. Let's delve into the preprocessing steps and data splitting techniques that paved the way for our speech recognition journey.

Data Preprocessing:

a) **Background Noise Augmentation:** To enhance our model's adaptability to real-world scenarios, we introduced background noise to our audio samples. For each audio file, we randomly selected one of the six background noise audio files and adjusted its intensity to be slightly lower than the average peak of the original audio. This augmentation empowered our models to learn and tackle challenging acoustic environments with confidence.

b) **Padding:** To handle varying audio lengths in our dataset, we applied padding to ensure uniform input dimensions. By padding the audio clips with zeros to match the maximum length, we created consistent representations for our models, enabling seamless training and classification.



```
| Class      | Average samples | Min samples | Max samples | Number of audios
|-----------|----------------|------------|------------|-----------------
| yes        | 15766           | 8022        | 16000       | 2377
| no         | 15708           | 7431        | 16000       | 2375
| up         | 15652           | 7510        | 16000       | 963
| stop       | 15806           | 6688        | 16000       | 620
| down       | 15734           | 7510        | 16000       | 597
| one        | 15716           | 8192        | 16000       | 540
| two        | 15676           | 6827        | 16000       | 518
| three      | 15756           | 8875        | 16000       | 519
| four       | 15782           | 8192        | 16000       | 559
| five       | 15791           | 8174        | 16000       | 560
| six        | 15845           | 10240       | 16000       | 590
| seven      | 15756           | 5945        | 16000       | 521
| eight      | 15729           | 6688        | 16000       | 545
| nine       | 15789           | 6688        | 16000       | 741
```

***Figure 2**: The table presents statistical information for different audio file classes. Each audio file in the dataset has a fixed sampling rate of 16 kHz, resulting in a duration determined by dividing the number of samples by 16,000. As observed, some audio files possess fewer than 16,000 samples, resulting in a duration of less than one second. For audio files with durations less than one second, padding was applied to standardize the length to 16,000 samples. The table displays the average, minimum, and maximum number of samples for each class, along with the total number of audio files in each category.*

c) **Feature Extraction:** For effective audio representation, we harnessed the power of two techniques - Mel Frequency Cepstral Coefficients (MFCC) and Short-Time Fourier Transform (STFT). While MFCC captured the spectral content of the audio, STFT produced a time-frequency spectrogram. Both techniques offered unique insights into the frequency components and temporal characteristics of the audio signals, facilitating accurate pattern recognition and classification.

Data Splitting

a) **Maintaining Balanced Datasets:** Ensuring the datasets remain balanced is vital for robust model training. To achieve this, we employed a balanced sample domain approach, evenly distributing audio clips across the thirteen categories. Subsequently, we allocated these audio clips to the training, validation, and test sets. By adopting this strategy, we ensured that our model received exposure to a diverse range of voice commands, facilitating reliable performance across all categories.

b) **User-ID-Based Split**: To ensure generalization and avoid overfitting, we adopted a strategic user-ID-based splitting approach. By grouping audio files based on their corresponding user-IDs, we ensured that each dataset contained audio data from unique individuals, minimizing bias and enhancing model performance.

Our meticulous data preprocessing and the adoption of a user-ID-based splitting strategy paved the way for reliable and generalized datasets, ready for model training. As we progressed with our speech recognition classification system, we knew that these critical steps would lay the foundation for innovative voice interactions and the development of user-friendly applications across various industries.

Understanding MFCC and STFT

MFCC (Mel Frequency Cepstral Coefficients) is a powerful feature extraction technique widely used in speech and audio signal processing. Its primary purpose is to capture essential spectral characteristics of an audio signal in a manner that closely resembles human auditory perception. Particularly suited for speech recognition and classification tasks, MFCC transforms complex audio waveforms into a concise representation, making it easier to identify phonetic and acoustic features. In our case, we extracted 20 coefficients for each audio frame, with a total of 30 audio frames sampled at a rate of 16000 Hz. This allows us to calculate the duration of each audio frame, which is approximately 533.33 milliseconds. The effectiveness of MFCC for speech recognition classification stems from its ability to provide a compact representation, focus on phonetic and acoustic features, exhibit robustness to noise and variability, and align with human auditory perception.

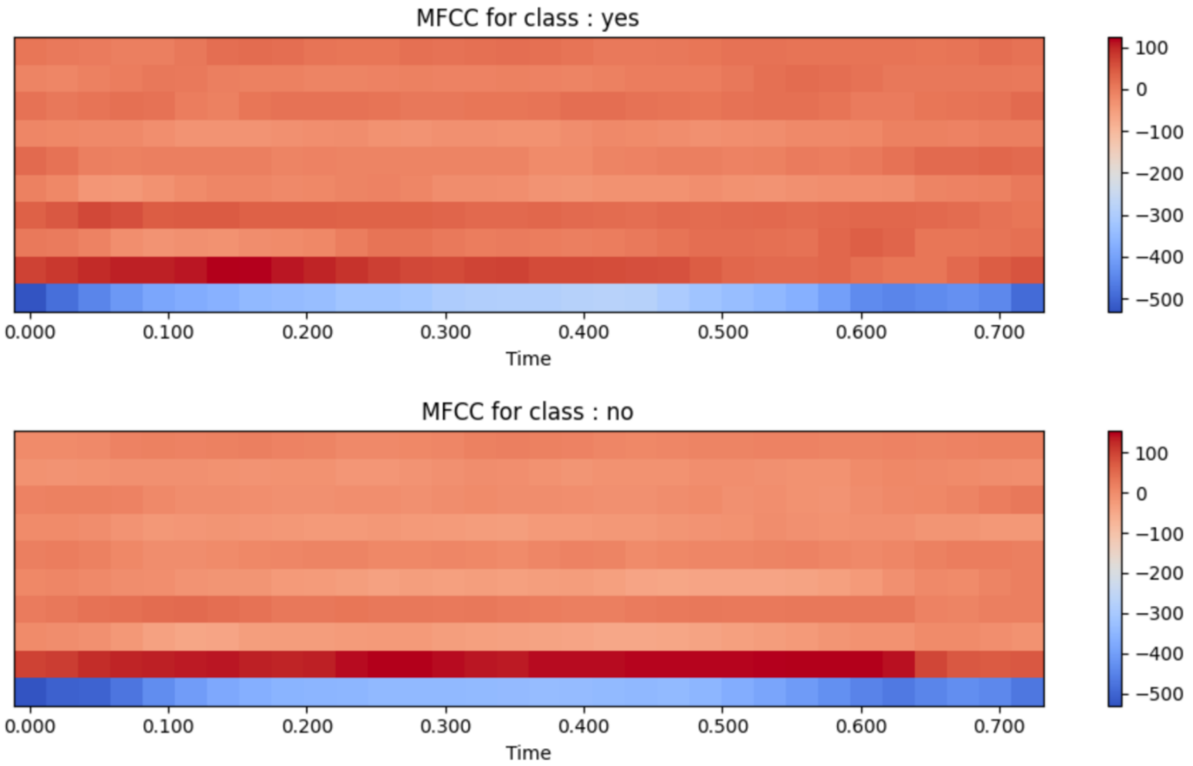


Figure 3: *The title of the plot displays the class name and the name of the specific audio sample, providing a clear reference to which class and audio file the visualization corresponds to. The plot provides a temporal representation of the Mel Frequency Cepstral Coefficients (MFCCs) with time on the x-axis. The color scale, indicated by the color bar on the right, corresponds to the magnitude of the MFCCs. Higher color intensity indicates stronger frequency components at specific moments in time, while lower intensity signifies weaker components. This visualization helps us comprehend how the MFCCs evolve over time, giving us valuable information about the audio signal's spectral characteristics and highlighting prominent frequency components at different time instances.*

STFT (Short-Time Fourier Transform) is a time-frequency analysis technique used to examine the spectral content of a signal over short, overlapping time intervals. Unlike the standard Fourier Transform, which gives a single frequency representation for the entire signal, STFT enables us to observe how the signal's frequency content changes over time. This temporal analysis is crucial for understanding time-varying signals, such as speech and audio, where the frequency components may vary rapidly. The output of STFT is a spectrogram, a two-dimensional representation where time is represented on the x-axis, frequency on the y-axis, and color intensity or magnitude represents the amplitude of frequency components at each time-frequency point. By computing the STFT over successive frames, we can track how the frequency content of the signal evolves over time.

STFT plays a pivotal role in speech recognition classification for several reasons. First, it allows time-frequency analysis, enabling the capture of time-varying frequency components in speech signals. Second, the spectrogram obtained from STFT serves as a feature representation for training speech recognition models, providing discriminative information for classification. Third, STFT can be used as a preprocessing step before applying other techniques like MFCC extraction. It offers a time-frequency representation of the audio signal, which can be further processed to extract relevant features for speech recognition tasks. Finally, STFT contributes to noise reduction algorithms, identifying and reducing noise components in the time-frequency domain, leading to enhanced accuracy in speech recognition systems, even in noisy environments.

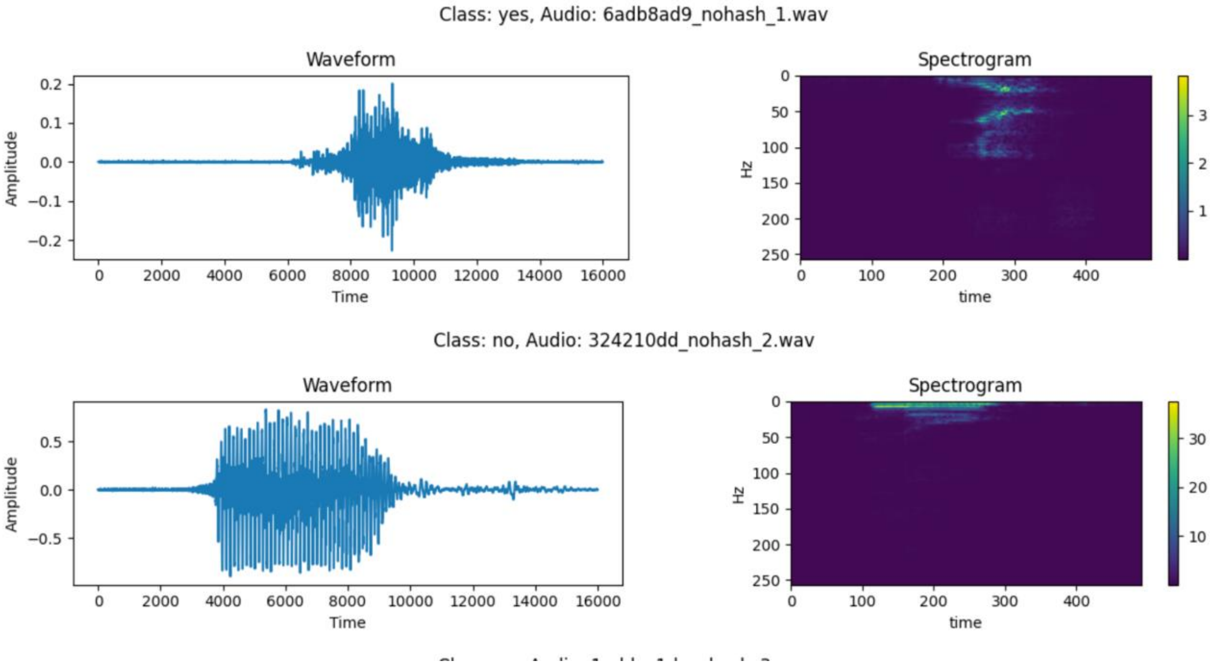


Figure 4: *The plot's title shows both the class name and the specific audio sample, offering a clear reference to the corresponding class and audio file. The left side of the plot presents the waveform of the selected audio sample, illustrating the audio signal's amplitude variation over time. This waveform visualization showcases how the audio intensity changes as the signal progresses. On the right side of the plot, the spectrogram of the same audio sample is depicted. The spectrogram illustrates the audio signal's frequency content over time, revealing how different frequency components evolve throughout the audio duration. The color intensity or magnitude in the spectrogram indicates the amplitude of frequency components at various time-frequency points.* This visualization provides valuable insights into the unique characteristics of the audio samples in each class, offering a comprehensive understanding of the waveform and frequency distribution for each category.

Audio Classification Without Background Noise

In our pursuit of achieving robust audio command classification, our journey began by addressing a simpler classification problem: recognizing clean audio commands without any background noise. This critical phase allowed us to establish a solid foundation before venturing into the more complex task of handling audio commands with background noise effectively.

Baseline Model and Noise-Free Audio Command Classification

To start, we constructed a baseline model tailored to handle the classification of clean audio commands. This initial step served as a performance benchmark, enabling us to evaluate the model's accuracy under ideal, noise-free conditions. By training the model on pristine audio commands, we gained valuable insights into its core features and patterns without being influenced by background noise.

There are several compelling reasons why we opted to begin with clean audio commands:

a) **Simplicity and Clarity:** By focusing on clean audio commands, we gained a clear understanding of the fundamental characteristics present in the data. This approach simplified the training process and allowed us to develop a solid foundation for the subsequent stages.

b) **Establishing a Performance Benchmark:** Training the baseline model on clean audio commands provided us with a performance benchmark for our classification task. This reference point allowed us to gauge the model's progress as we introduced more realistic and challenging scenarios involving background noise.

c) **Identifying Noise Impact:** Evaluating the baseline model's performance in the presence of background noise revealed how noise affected its accuracy. Understanding these impacts helped us devise strategies to mitigate noise-related challenges.

d) **Data Augmentation:** Using the clean audio data as a baseline, we augmented the dataset artificially by introducing various types of background noise to the audio commands. This data augmentation technique bolstered the model's robustness and adaptability to real-world conditions.

Exploring MFCC and STFT

With the baseline model in place, we delved into feature extraction techniques essential for speech recognition tasks: MFCC (Mel Frequency Cepstral Coefficients) and STFT (Short-Time Fourier Transform).

MFCC proved to be an invaluable feature extraction technique. By capturing essential spectral characteristics of the audio signal in a manner akin to human auditory perception, MFCC efficiently reduced the dimensionality of the feature space. Its compact representation simplified the classification process and reduced computational complexity.

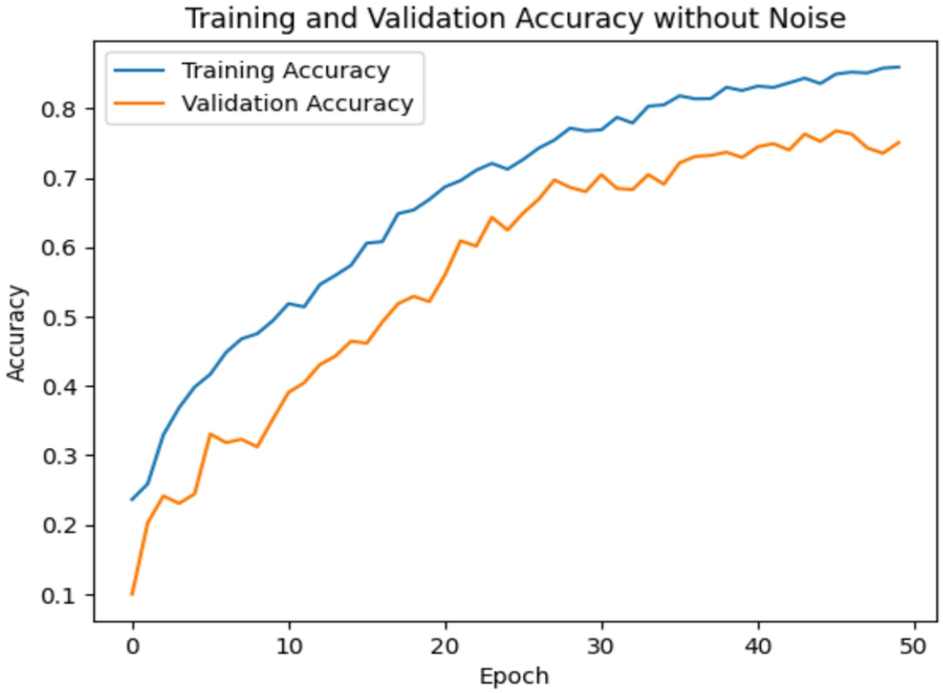


Figure 5: *An illustration of the training and validation accuracy of the baseline model using STFT without background noise.*

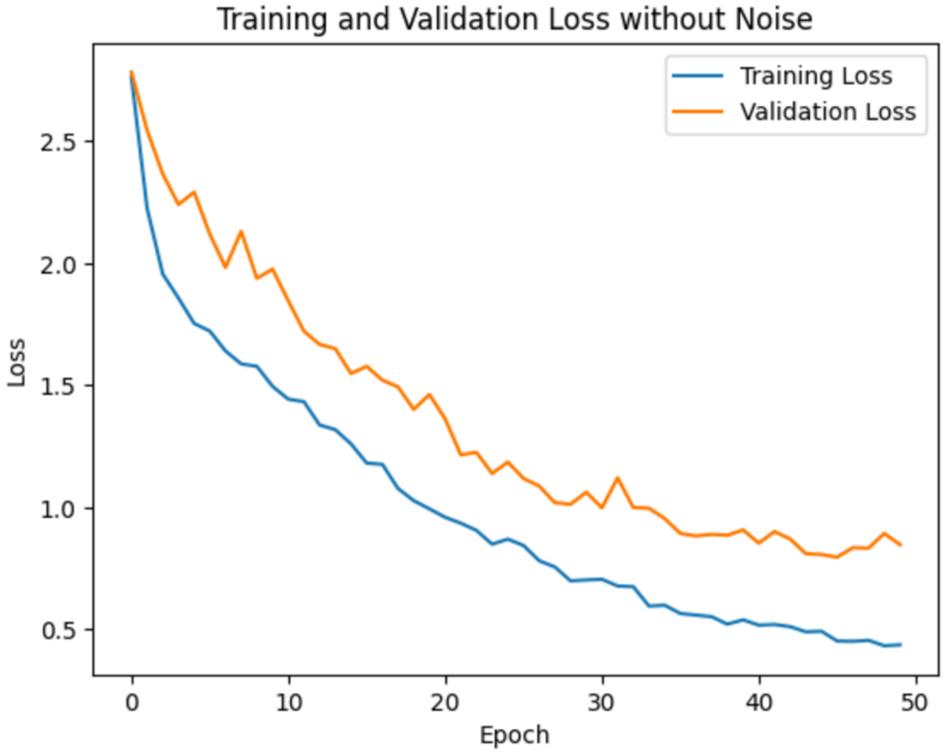


Figure 6: *An illustration of the training and validation loss of the baseline model using STFT without background noise.*

STFT allowed us to analyze the spectral content of the audio signal over short, overlapping time intervals. Unlike traditional Fourier Transform, STFT provided a time-varying representation of frequency content, which was crucial for understanding time-varying signals like speech and audio.

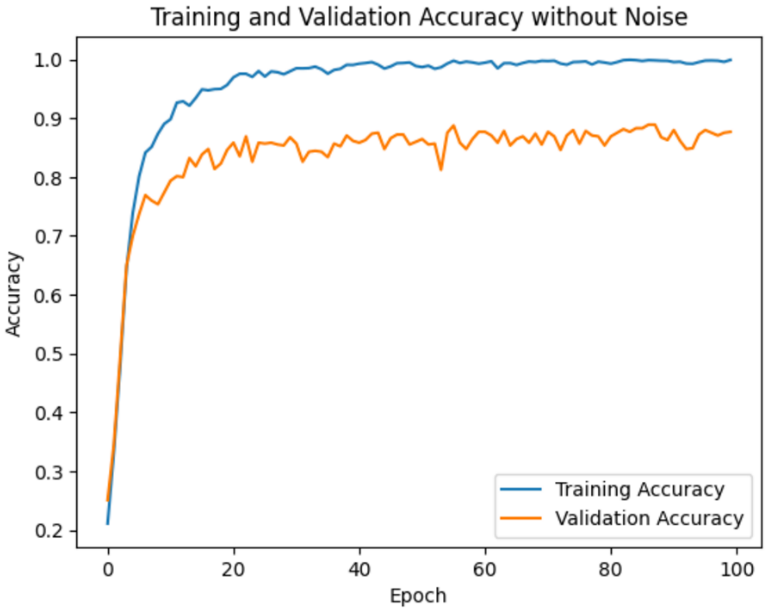


***Figure** 7: An illustration of the training and validation accuracy of the baseline model using MFCC's without background noise.*

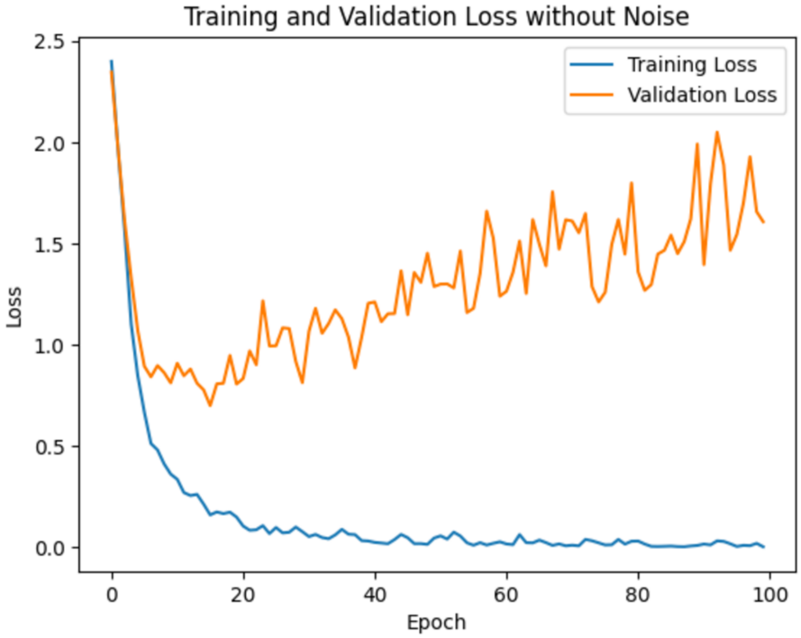


***Figure 8**: An illustration of the training and validation accuracy of the baseline model using MFCC's without background noise.*

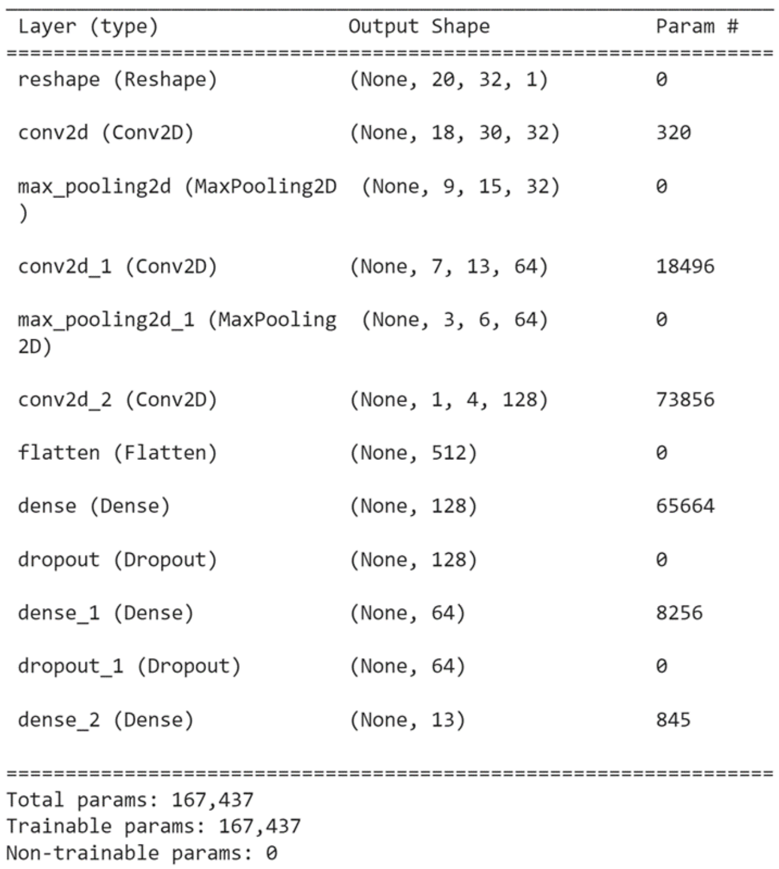
Our journey in exploring clean audio commands and their feature extraction paved the way for the subsequent phase of our project: conquering audio commands with background noise. Armed with the knowledge and insights gained from this foundation, we are now poised to build a robust and efficient audio command classification system capable of excelling in real-world environments, revolutionizing voice interactions across various applications.

Enhancing the Model: Audio Classification with Background Noise

With our baseline model established and modifications made, we set out to enhance its capabilities further by training it to handle audio commands with background noise. Our objective was to develop a robust and versatile model that could accurately classify commands even in challenging real-world acoustic environments.

Choosing the Optimal Base Model:

After thorough evaluation, we identified the noise-free audio command classification model that utilized MFCC as the input, achieving superior performance. This model served as the base upon which we would enhance its capacity to handle background noise effectively.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 reshape (Reshape)           (None, 20, 32, 1)         0

 conv2d (Conv2D)             (None, 18, 30, 32)        320

 max_pooling2d (MaxPooling2D  (None, 9, 15, 32)        0
 )

 conv2d_1 (Conv2D)           (None, 7, 13, 64)         18496

 max_pooling2d_1 (MaxPooling  (None, 3, 6, 64)         0
 2D)

 conv2d_2 (Conv2D)           (None, 1, 4, 128)         73856

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 128)               65664

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 13)                845

=================================================================
Total params: 167,437
Trainable params: 167,437
Non-trainable params: 0
```

***Figure 9**: An overview of our primary model utilized with MFCCs of audio commands containing background noise.*

Training with Audio Commands Containing Background Noise

To equip our base model for handling background noise, we employed a training process using audio commands augmented with various types of noise. The model was compiled using the Categorical Crossentropy loss function and the accuracy metric. We incorporated the Adam optimizer with an initial learning rate of 0.001 and applied an exponential decay schedule with a decay rate of 0.99 every 200 steps (staircase mode) to adjust the learning rate during training.

Model Checkpoint Callback for Optimal Model Preservation

During training, we utilized the Model Checkpoint callback to save the best-performing model based on validation accuracy. This ensured that we retained the most optimal version of the model, contributing to its robustness and effectiveness.

Outstanding Performance and Robustness

Our enhanced model showcased exceptional performance during training, achieving a training set loss of 0.0215 and an accuracy of 0.9956. On the validation set, it attained a loss of 2.3977 and an accuracy of 0.8431 (see Figure 6 and Figure 7). The model's ability to generalize well was evident in its performance on the test set, achieving an accuracy of 0.86. This strong generalization indicated the model's robustness and suitability for audio command classification tasks, even in the presence of background noise.

By enhancing our base model to adeptly handle background noise, we have crafted a reliable and efficient system for recognizing voice commands in real-world scenarios. This model's adaptability and effectiveness open up a world of possibilities for various applications and industries, ushering in a new era of seamless and accurate voice interactions across the globe.

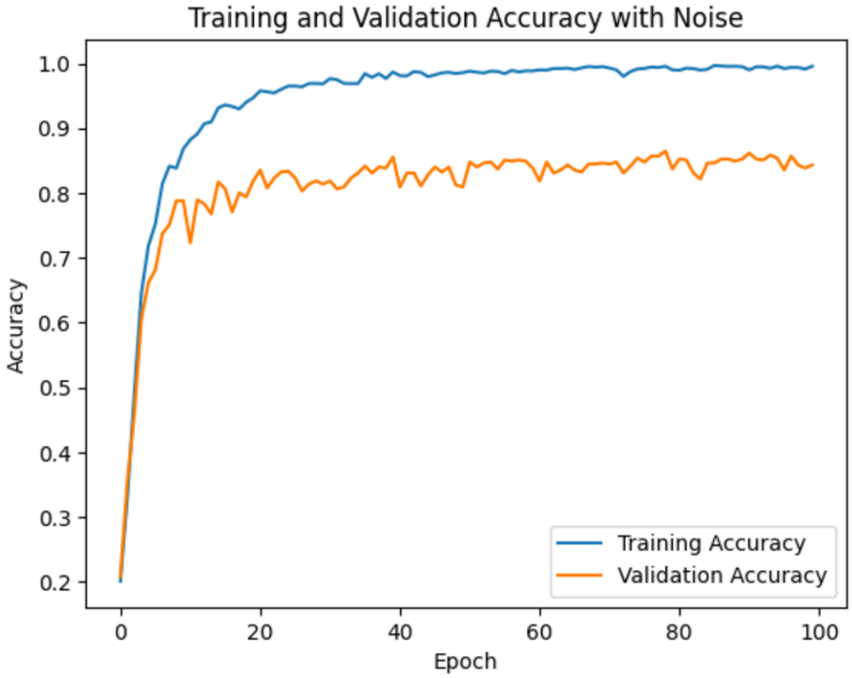


Figure 10: *An illustration of the training and validation set accuracy per epoch of our primary model, which utilizes MFCCs of audio commands containing background noise.*

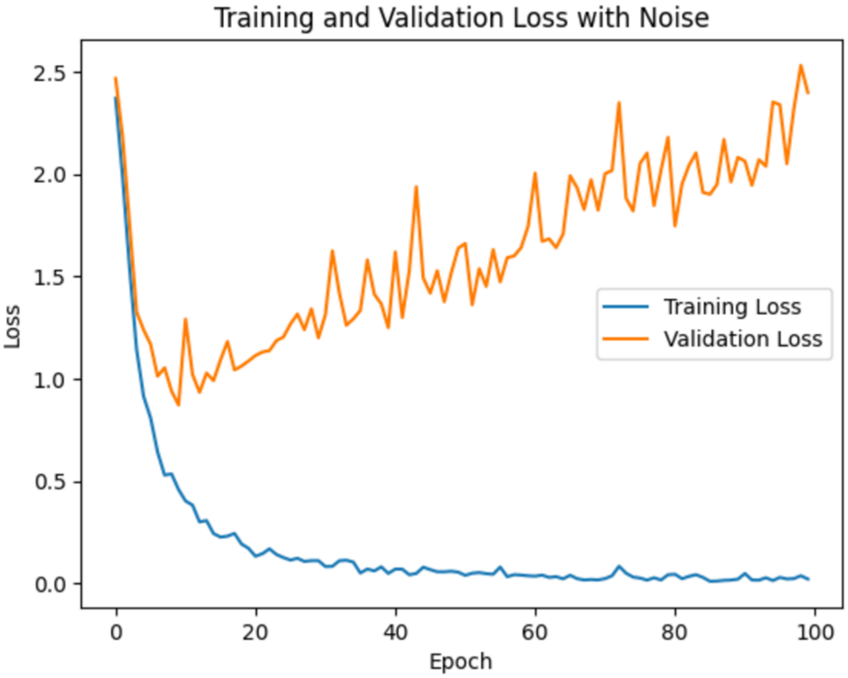


Figure 11: *An illustration of the training and validation set loss per epoch of our primary model, which utilizes MFCCs of audio commands containing background noise.*

Applications of the Robust Audio Command Classification Model

The development of our robust audio command classification model with background noise opens up a wide array of exciting and practical applications across various industries and domains. Let's explore some of the potential applications that can benefit from this innovative technology:

a) **Voice-Controlled Assistants:** Our model can be seamlessly integrated into voice-activated assistants and virtual personal assistants, empowering them to accurately understand and respond to user commands, even in noisy environments. This advancement enhances user experience and improves the overall efficiency of voice-controlled systems.

b) **Smart Home Devices:** The model's speech recognition capabilities can power smart home devices, enabling users to effortlessly control and interact with their smart appliances, lighting, security systems, and entertainment devices using voice commands. This integration simplifies daily tasks and enhances home automation.

c) **Automotive Voice Control:** In automotive settings, our model can be utilized in voice recognition systems, allowing drivers to control various in-car functions, such as navigation, entertainment, and climate control, hands-free. This contributes to safer driving experiences and improved driver convenience.

d) **Call Center Automation:** By integrating the model into call center systems, voice-based interactions can be automated, streamlining customer service processes, reducing call handling times, and enhancing overall efficiency.

e) **Industrial Automation:** In industrial environments, the model can be applied to voice-controlled machinery and equipment, simplifying operation and reducing the need for manual controls. This can lead to improved productivity and enhanced safety in industrial settings.

f) **Accessibility Solutions:** Our model can serve as a part of speech recognition solutions for individuals with disabilities, allowing them to interact with technology and devices using voice commands. This inclusion fosters accessibility and independence for users with diverse needs.

g) **Security and Surveillance:** In security and surveillance systems, the model can be employed to detect and respond to specific voice commands for access control and authentication, bolstering security measures.

h) **Gaming and Entertainment:** Our model can elevate voice recognition in gaming consoles and entertainment devices, offering a more immersive and interactive gaming and entertainment experience.

i) **Language Learning and Education:** By integrating the model into language learning apps and educational platforms, it can provide accurate feedback and assessment based on spoken commands, enhancing language learning experiences.

Conclusion and Future Directions

Our journey in developing the robust audio command classification model has been a gratifying one, yielding significant advancements in the field of speech recognition. Through various iterations and enhancements, we successfully created a highly reliable system capable of accurately classifying voice commands, even in challenging acoustic environments with background noise.

As we conclude this endeavor, we look towards the future with a passion for further exploration and innovation. The applications of speech recognition technology are vast, and we envision our model becoming an integral part of everyday life, simplifying tasks, enabling greater accessibility, and revolutionizing voice interactions.

In the coming years, we aspire to continue our research, exploring cutting-edge techniques, and staying abreast of the latest developments in speech recognition and machine learning. Our commitment to advancing the field is driven by a vision of a more connected world, where voice-enabled applications enhance productivity, efficiency, and user experiences on a global scale.

We extend our gratitude to the research community, industry pioneers, and AI enthusiasts for their valuable contributions to the field, as we collectively shape the future of voice technology. With collaboration and innovation, we are confident that speech recognition will continue to evolve, transforming how we interact with technology and redefining the boundaries of possibility.

As we embark on this journey of discovery, we invite others to join us in exploring the uncharted realms of speech recognition, as we create a future where voice becomes the key to unlocking a new era of human-computer interaction. Together, we can build a world where technology adapts to us, understands us, and empowers us, forging a path towards a more intelligent and inclusive future.