Spoken Digits Recognition Using Machine Learning

CE362 – Digital Signal Processing Term Project

Group Details: Hamza A. Siddiqui (2020147)

***Problem Statement*: Voice recognition is an important technology that has taken a rise due to the artificial intelligence revolution with hundreds of applications. In this project, we aim to develop a voice recognition system to accurately recognize spoken digits. The task involves training a neural network model on the spoken digits dataset and evaluating its performance in terms of accuracy. By achieving high accuracy, the system can be used in various applications, such as automated phone systems, voice-controlled devices, and other IoT systems.**

I. Introduction

In this project, we have utilized the spoken digits dataset to train a neural network model for recognizing spoken digits through cepstrum analysis. Cepstrum method makes it possible to extract information about the signal energy, the pulse response of the speaker's throat, and the frequency of vibration of the vocal cords. We can compare two signals through the correlation coefficient extracted through cepstrum analysis.

The dataset used, consists of audio recordings of different individuals speaking the digits from 0 to 9. Each audio sample is preprocessed to extract relevant features, and a lowpass filter is applied to enhance the signal quality. The extracted features are then used to train a neural network model using python's 'Keras' library.

By accurately identifying the spoken digits, the system can be integrated into various applications that require voice input, simplifying user interaction and enabling efficient processing of voice-based commands.

II. Methods

A. *Data Selection and Preprocessing*

The data for this project was taken from the spoken digits dataset that can be found at Kaggle. The dataset consists of .wav recordings of different individuals speaking digits 0 to 9.

In order to prepare the audio data for relevant feature extraction the data was preprocessed. The preprocessing included applying a lowpass filter; the filter helped remove high-frequency noise from the signal and hence improved the signal quality and reduced the impact of irrelevant noise on the recognition process. The function 'apply_lowpass_filter()' was made to complete this task.

```python
# Function to apply lowpass filter to audio signal
def apply_lowpass_filter(signal, cutoff_freq, sr):
    b, a = scipy.signal.butter(4, cutoff_freq / (sr / 2), 'low')
    filtered_signal = scipy.signal.lfilter(b, a, signal)
    return filtered_signal
```

Python

B. *Feature Extraction Through Cepstrum Analysis*

Cepstrum is a frequency-domain representation of the power spectrum of a signal and can capture important characteristics of the digital signal.

For the cepstral feature extraction, we took the Short-time Fourier Transform (STFT) of the preprocessed signal using **librosa.stft()**. The absolute value of the complex valued STFT coefficients was taken to extract the magnitude spectrum using **np.abs()**. To enhance the perceptual properties of the spectrum, a logarithmic compression was applied to the magnitude spectrum. The resultant logarithmic spectrum is transformed into cepstral domain using the Inverse Fourier Transform using **librosa.feature.mfcc()**. The resulting cepstrum holds the information related to the vocal tract and speech.

For the execution of the above process, a function was made and Python's Librosa library was used:

```python
# Applying Low-Pass Filter
filtered_signal = apply_lowpass_filter(signal, cutoff_freq, sr)  #
stft = np.abs(librosa.stft(filtered_signal))
cepstrum = librosa.feature.mfcc(S=librosa.amplitude_to_db(stft), n
return cepstrum.T
```

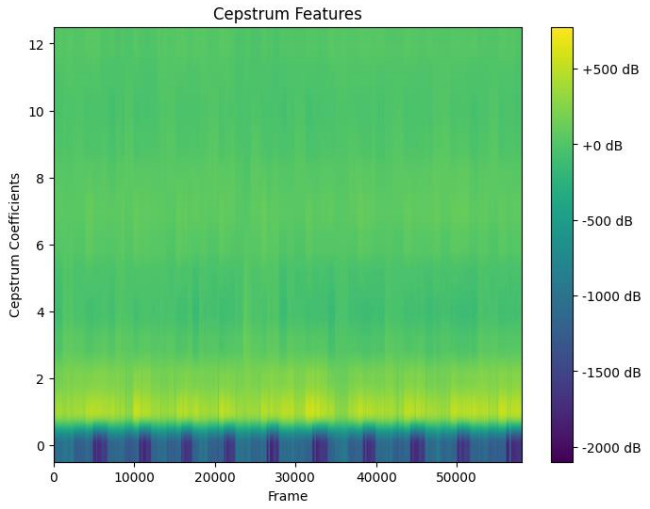
C. *Machine Learning Model*

The extracted cepstral features used as inputs to the artificial neural network that was built using

the Keras library. The machine learning model was able to learn and generalize the patterns in the audio signals to predict the corresponding spoken digits.

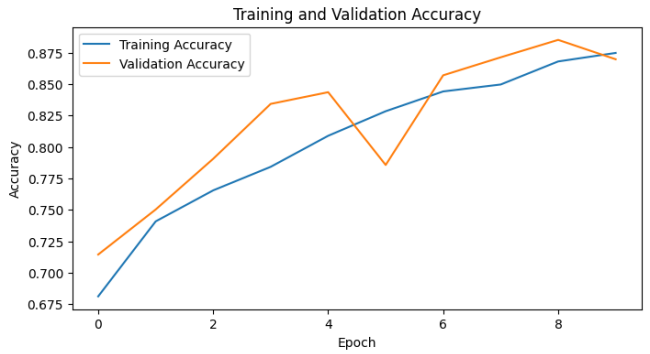
The model architecture included three layers and made use of the ReLU and Softmax activation functions. This resulted in a high accuracy of the model that enabled us to accurately classify the spoken digits.

The cepstrum features extracted are shown in the following graph:

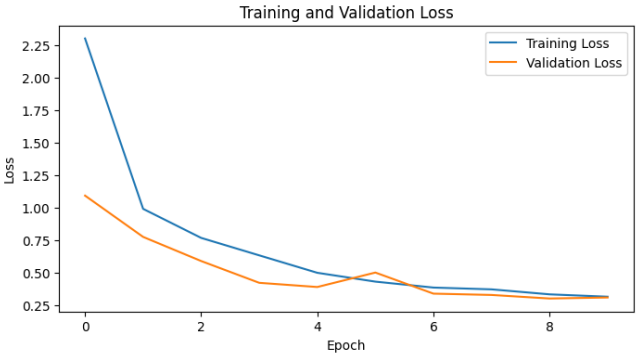


III. Results

Our data was split into training and testing data. 80% of the data was used as training data and the rest of the 20% was used as the test data. After the compilation of the artificial neural network, the model was tested with the testing data. The following graph was plotted which shows how our training accuracy increased with the increase of number of epochs and similarly our validation accuracy also increased with the increase in number of epochs. The graphs helped in understanding the model's performance trends and determining the optimal number of training epochs for achieving the highest accuracy on the validation data.



Similarly, the loss of the ANN model during training and validation was also measured and visualized through the following graph which shows how the loss improved during the training of the model.



The average test loss and test accuracy was also calculated. Our values show that the model achieved good performance on the test data and was able to classify 87% of the spoken digits.

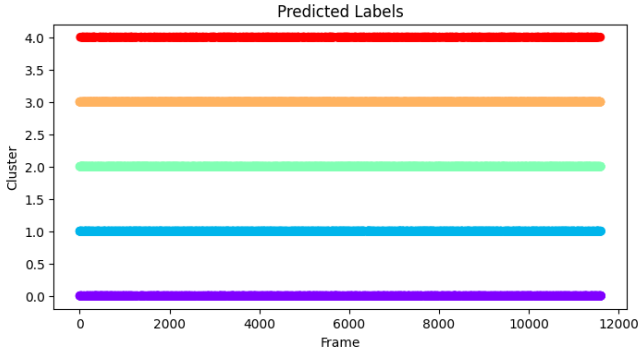
```
Test Loss: 0.3106944262981415
Test Accuracy: 0.8697450160980225
```

IV. Discussion

Our goal to accurately recognize spoken digits by implementing a voice recognition system using cepstrum analysis and an Artificial Neural Network (ANN) model. This project has been successful in achieving its goal.

The use of cepstrum analysis in this project was successful in capturing the relevant features in the frequency domain that were necessary for distinguishing between different spoken digits. The resulting cepstrum features provided a compact representation of the spectral information, enabling the model to learn discriminative patterns for digit recognition.

The model achieved a high-test accuracy of 0.8697, showing that it was successful in recognizing the spoken digits with a relatively high level of precision as indicated below.



The project demonstrates the potential of using cepstrum analysis in voice recognition applications and highlights the effectiveness of ANN models in handling complex classification tasks.

However, it's worth noting that the project's success is subject to certain limitations. The evaluation was performed on a specific dataset, and the performance may vary when applied to different datasets or real-world scenarios. In conclusion, the project successfully implemented a voice recognition system for spoken digit recognition using cepstrum analysis and an ANN model.

REFERENCES

Following are the books, articles, and documentation that was used for the completion of this project:

[1] Proakis, J. G., & Manolakis, D. G. (2006). Digital Signal Processing: Principles, Algorithms, and Applications (3rd ed.). Prentice Hall.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Librosa: Audio and Music Signal Analysis in Python. (n.d.). Retrieved from https://librosa.org/

[3] K. Elissa, "Title of paper if known," unpublished.

[4] TensorFlow. (n.d.). Retrieved from https://www.tensorflow.org/

[5] Rejwan, S. (2021). Speech Recognition Using Deep Learning: A Review. Journal of Electrical and Computer Engineering, 2021, Article ID 5517012. doi: 10.1155/2021/5517012

[6] Chen, Y., Deng, L., & Liu, X. (2014). Deep learning and its applications to signal and information processing [PDF]. IEEE Access, 2, 1548-1554. doi: 10.1109/ACCESS.2014.2352453