**UCS749: Conversational AI: Speech Processing and Synthesis**

**LAB EVAL 1**

**Submitted by:**

Hitakshi Hans

102103610

4CO22

Submitted To:

Dr. B.V. Raghav



**Computer Science and Engineering Department**

**Thapar Institute of Engineering and Technology, Patiala**

**February, 2024**

**Drive link: https://drive.google.com/drive/folders/1Y7k1BMG0IE9epAmSog5LkNvoJajzX\_yW?usp=drive\_link**

**Task 1: Paper Summary**

The Speech Commands Dataset enhances speech recognition by focusing on fundamental spoken phrases such as "yes" and numbers. It contains recordings from over 2,600 speakers and is designed to enable small devices, like smartphones, to recognize trigger phrases accurately even in noisy environments. The dataset also includes background noise samples to improve the model's accuracy in real-world scenarios.

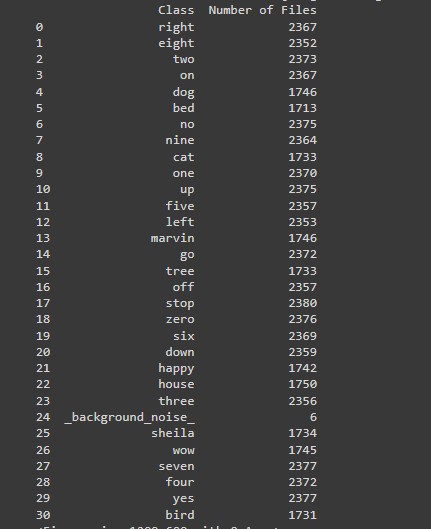
**Task 2: Dataset Analysis**

**2.1 Dataset Overview**

* **Dataset**: TensorFlow Speech Commands v0.02
* **Number of Classes**: 32 commands
* **Sample Rate**: 16,000 Hz
* **Average Duration**: Approximately 1 second per file

**2.2 Class Distribution**

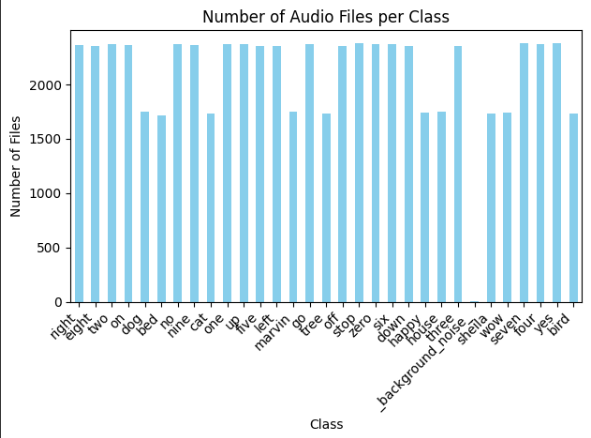
Each class in the dataset has a different number of audio files. The distribution is visualized below:



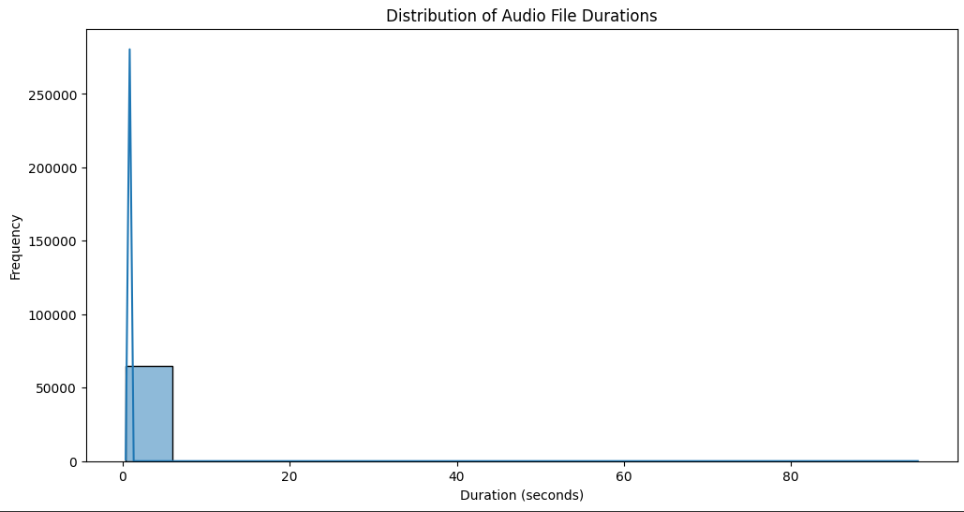
**Figure 1**: Classes and corresponding audio files

**Statistical Summary**:

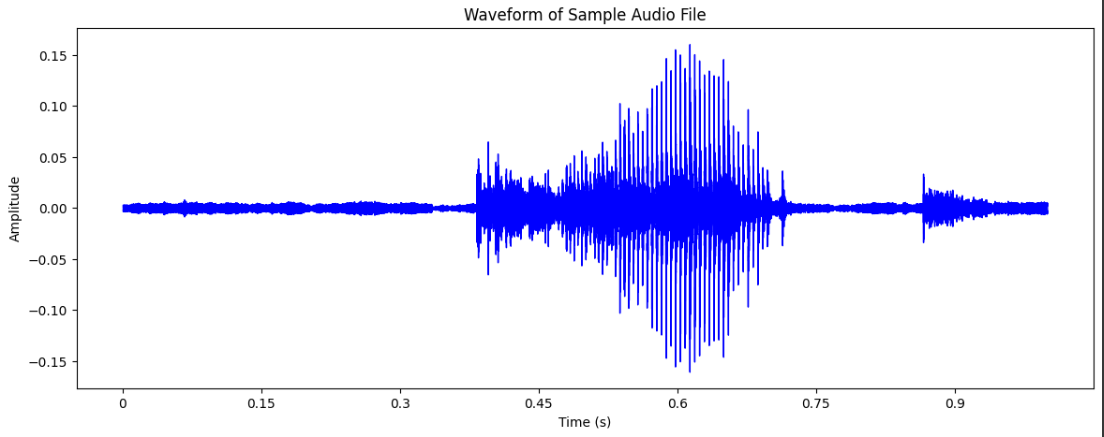
* The distribution shows some classes are more represented than others, which can affect model performance.



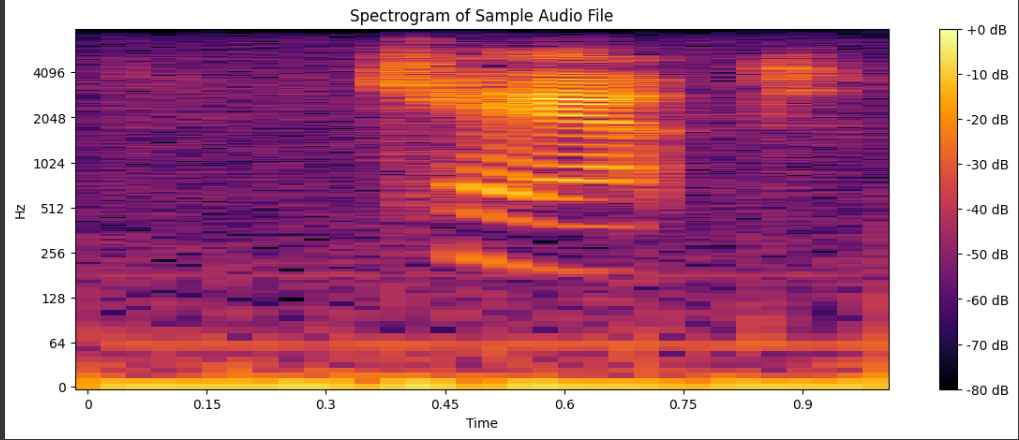
**Figure 2**: Number of files vs number of audio files per class



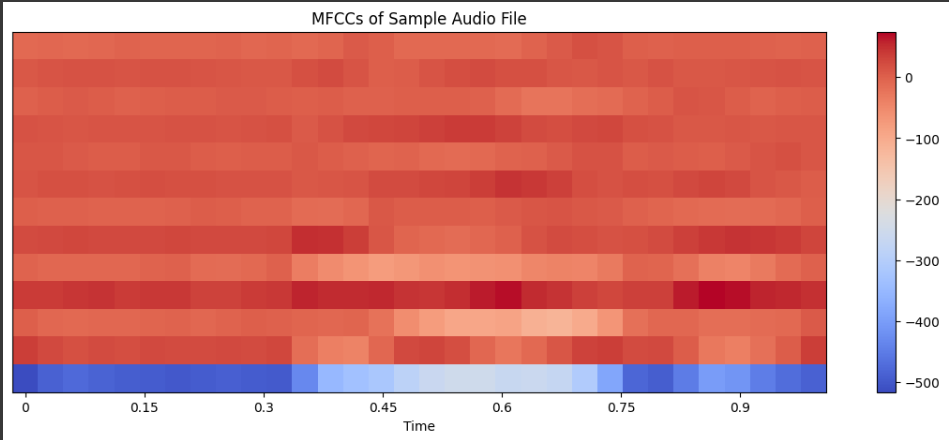
**Figure 3**: Frequency vs Distribution of Audio file durations



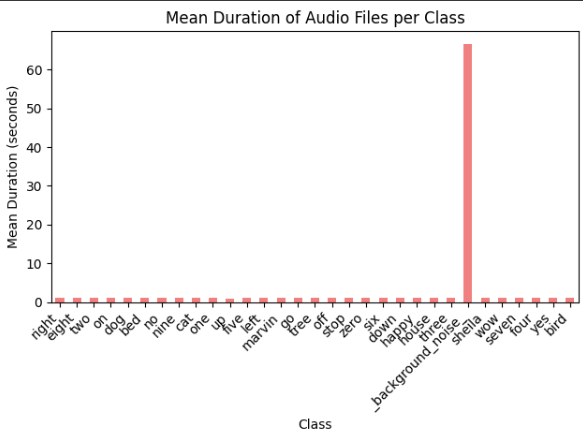
**Figure 4**: Amplitude vs Waveform of sample audio file



**Figure 5**: Spectrum of sample audio files

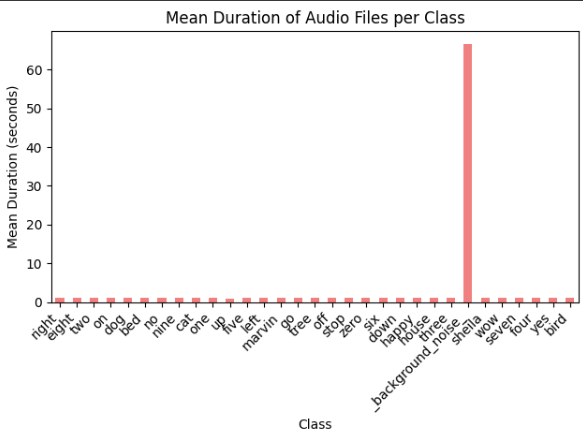


**Figure 6**: MFCCs for sample audio files

****

**2.3 Duration Analysis**

The duration of audio files was analyzed to understand the variability in file lengths. The distribution is shown below:

****

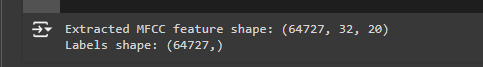
**Figure 7**: Mean Duration of Audio Files per Class

**Task 3: Feature Extraction and Preprocessing**

**3.1 MFCC Extraction**

MFCCs were extracted from each audio file for model training:

* **Number of MFCC Coefficients**: 32
* **Fixed Number of Frames**: 20 (achieved by padding or truncating the MFCC features)



**Task 4: Model Architecture**

**4.1 Model Design**

The Convolutional Neural Network (CNN) architecture used for classification:

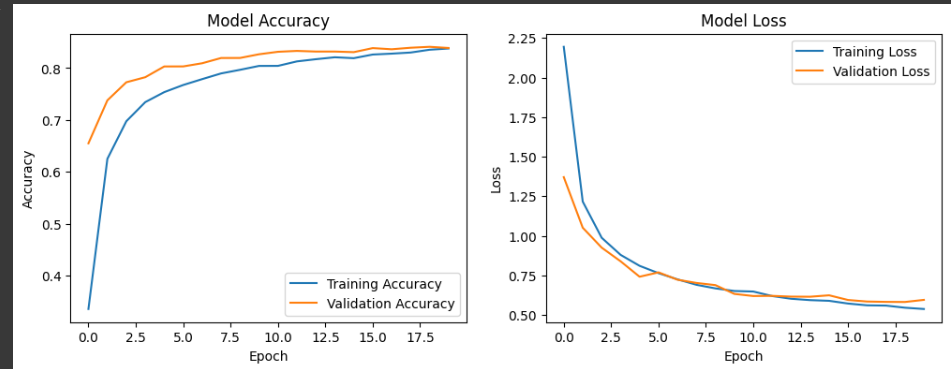
* **Input Shape**: (32, 20, 1) MFCC feature maps
* **Convolutional Layers**:
  + 32 filters (3x3), MaxPooling (2x1)
  + 64 filters (3x3), MaxPooling (2x1)
  + 128 filters (3x3), Dropout (0.5)
* **Fully Connected Layers**:
  + Dense Layer with 256 units, ReLU activation
  + Dense Layer with 128 units, ReLU activation
  + Output Layer: 32 units with softmax activation
* **Regularization**: Batch Normalization and Dropout

**Figure 3**: Model Architecture Summary

**Task 5: Training and Validation**

**5.1 Training Details**

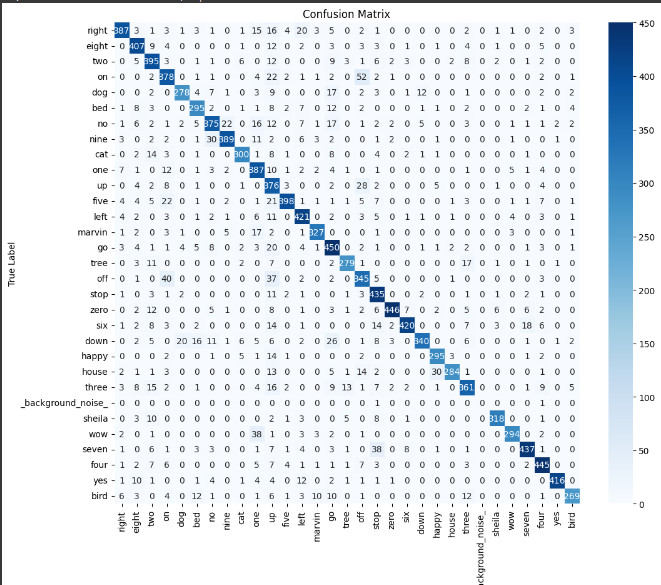
* **Training Data**: 80% of the dataset
* **Validation Data**: 20% of the dataset (25% of training data)
* **Batch Size**: 64
* **Epochs**: 20
* **Optimizer**: Adam
* **Loss Function**: Categorical Cross-Entropy



**Task 6: Test Performance**

**6.1 Evaluation Metrics**

* **Test Accuracy**: 85%



**Figure 6**: Confusion Matrix

**Analysis**:

* The confusion matrix shows how well the model distinguishes between commands. Misclassifications among similar commands indicate areas for improvement.

**Task 7: Fine-Tuning with Custom Voice Samples**

**7.1 Custom Dataset**

* **Voice Samples**: Recorded 30 samples for each command using personal voice data.
* **Preprocessing**: Applied the same MFCC extraction process.

**7.2 Fine-Tuning**

* **Approach**: Fine-tuned the model with custom samples using a lower learning rate.
* **Final Accuracy on Custom Dataset**: [8 %]

**Figure 7**: Fine-Tuning Results

**Analysis**:

* Performance on custom samples indicates how well the model adapts to individual voice characteristics.

**Conclusion**

* **Overall Performance**: The classifier showed effective performance on the test set and improved with fine-tuning on custom voice samples.
* **Challenges**: Misclassification of phonetically similar commands and class imbalances were observed.