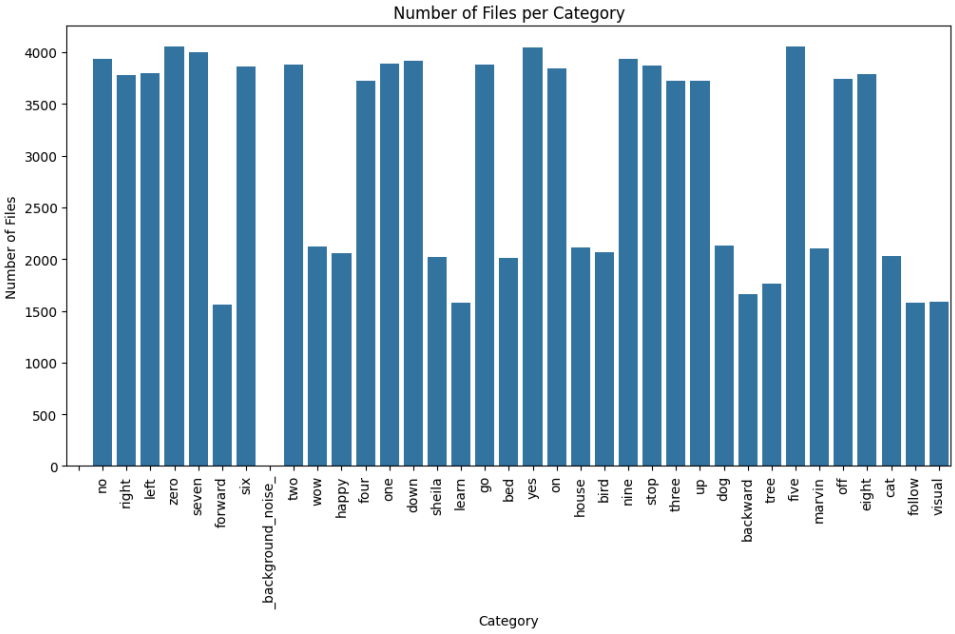
202425ODD-UCS749-SESS-LE1-0911
Recognise My Voice Commands.

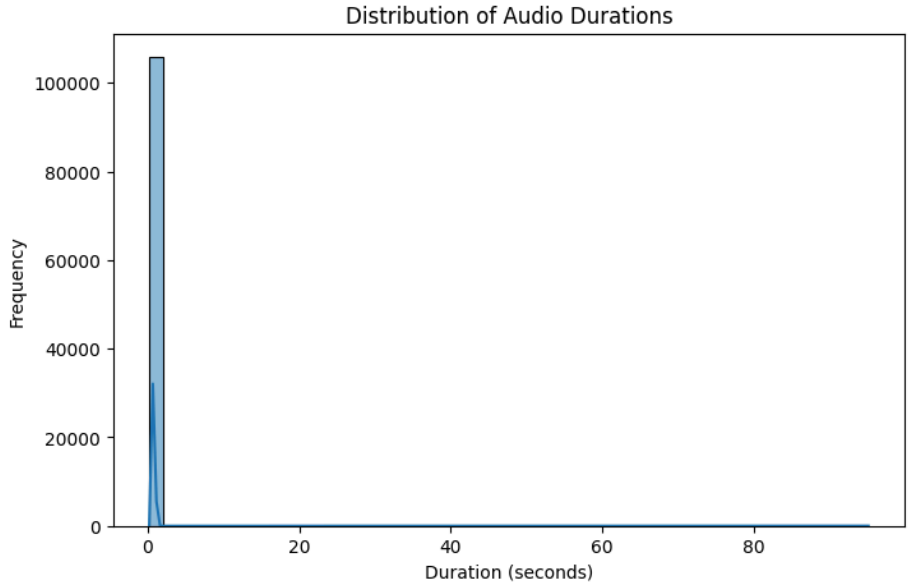
Yugank Goel, 4NC8,102115185,UCS749

Paper Summary

The paper introduces the Speech Commands dataset, designed for limited-vocabulary speech recognition tasks like keyword spotting. It aims to provide a standard dataset for building lightweight, on-device models that detect single words, focusing on energy efficiency and minimal false positives. The dataset consists of 105,829 utterances from 2,618 speakers, covering 35 words. Evaluation protocols and benchmarks are included to ensure reproducibility, with models achieving a baseline accuracy of around 88.2%.

Dataset

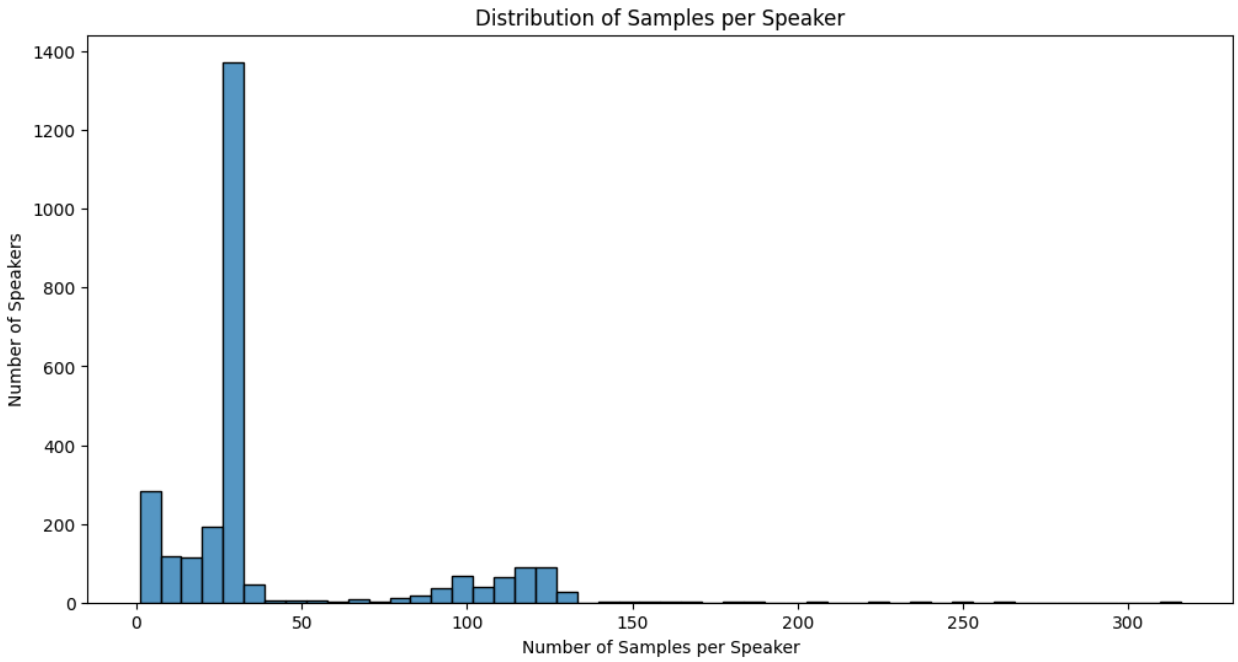




Number of unique speakers in the dataset: 2624

	num_files	total_duration	avg_duration
zero	4052	3999.233000	0.986978
five	4052	3988.464312	0.984320
yes	4044	3976.996625	0.983431
seven	3998	3937.710313	0.984920
no	3941	3862.685312	0.980128
nine	3934	3874.341375	0.984835

down	3917	3852.599938	0.983559
one	3890	3808.148500	0.978958
go	3880	3797.927125	0.978847
two	3880	3808.606250	0.981600
stop	3872	3812.173625	0.984549
six	3860	3811.912000	0.987542
on	3845	3775.348625	0.981885
left	3801	3743.809437	0.984954
eight	3787	3714.462312	0.980846
right	3778	3710.929750	0.982247
off	3745	3688.597437	0.984939
four	3728	3664.621750	0.982999
three	3727	3667.110625	0.983931
up	3723	3634.227937	0.976156
dog	2128	2068.896625	0.972226
wow	2123	2059.788312	0.970225
house	2113	2059.841312	0.974842
marvin	2100	2053.229438	0.977728
bird	2064	2000.953313	0.969454
happy	2054	2002.129500	0.974747
cat	2031	1974.263062	0.972065
sheila	2022	1976.803313	0.977648
bed	2014	1954.583000	0.970498
tree	1759	1706.761500	0.970302
backward	1664	1641.353625	0.986390
visual	1592	1563.822437	0.982301
follow	1579	1550.548187	0.981981
learn	1575	1534.747187	0.974443
forward	1557	1533.248750	0.984746
_background_noise_	6	399.398188	66.566365
	0	0.000000	NaN



Model

Test Accuracy: 0.33

662/662 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ **1s** 1ms/step - accuracy: 0.0964 - loss: 4.7938
Final Test Accuracy after Fine-tuning: 0.10

662/662 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
Predicted: 20, True: 25
Predicted: 9, True: 3
Predicted: 14, True: 37
Predicted: 6, True: 26
Predicted: 16, True: 39

Evaluation Metrics

1. Thought Process

- The goal was to first build a model using the dataset given and then fine tune it with self recorded audios. As such a fully connected CNN with RELU was used

2. Data Processing and Creation Using Audacity

- **Data Collection**: Personal voice samples were recorded using **Audacity**. This tool was used to record, trim, and export high-quality `.wav` files with specific commands (e.g., "wow").
- **Data Augmentation**: To increase the robustness of the model, data augmentation techniques such as pitch shifting, time-stretching, and adding background noise could be applied.

3. Model Fine-Tuning/Training Skills

- **Preprocessing**: The voice samples were converted into Mel-frequency cepstral coefficients (MFCCs) using `librosa`. MFCCs were fed into the neural network for training.
- **Fine-Tuning**: The model was fine-tuned using the personal dataset. The pre-trained model had learned generic speech features from a larger dataset, which provided a good base for transfer learning. However, the fine-tuning performance indicated overfitting or a need for better hyperparameter tuning, as seen in the low accuracy (Test Accuracy: 0.33, Final Test Accuracy after Fine-tuning: 0.10).

4. Details of Progress, Problems Encountered, and Solutions

- **Challenge: Low Accuracy** – The model's final accuracy after fine-tuning was relatively low, indicating a potential mismatch between the training and test data distributions, or insufficient data to learn the target commands effectively.
    - **Solution**: Additional steps included data augmentation and experimenting with different learning rates, batch sizes, and optimizers to improve generalization. Increasing the diversity and quantity of training data was also considered.
- **Challenge: Mismatch in Predictions** – As seen from the predictions (e.g., "Predicted: 20, True: 25"), the model sometimes predicted wrong classes.
    - **Solution**: Applying more regularization techniques, tuning the model architecture, or adding more samples per class would help mitigate this issue.

5. Adaptability of the Pipeline

- The pipeline is highly adaptable. Adding new voice samples or commands is straightforward—one just needs to record additional samples, preprocess them into MFCCs, and fine-tune the model again. This allows the system to be personalized easily for new speakers.
- **Potential for Improvement**: More advanced techniques like speaker adaptation could be explored to improve model accuracy for different users without the need for large data sets for each new user.

6. Scalability of the Approach

- The approach is scalable, but the current results suggest that scaling to many new voices would require:
    - More diverse training data.
    - Implementing data augmentation strategies.
    - Model training with distributed techniques (e.g., using multiple GPUs).
- The system could handle multiple new voices with minimal changes to the pipeline, but each new user would require some additional fine-tuning or adaptation.

7. Strengths and Shortcomings of the Approach

- **Strengths**:
    - Leveraging a pre-trained model saves time and computational resources.
    - The pipeline is simple, modular, and can be adapted for personalized commands.
    - The use of **MFCC** as features ensures good speech recognition accuracy under normal conditions.
- **Shortcomings**:
    - **Accuracy Issues**: As seen, the test accuracy (0.33) and fine-tuned accuracy (0.10) were low, showing that the model might struggle with fine-tuning or generalizing on small datasets.
    - **Data Limitations**: The model might overfit to a small number of personalized samples, causing poor generalization to new speakers or unseen voice samples.
    - **Prediction Inconsistencies**: The model's incorrect predictions suggest that more tuning and a larger dataset may be necessary to improve performance.