**MID SEMESTER LAB EVALUATION**

**CONVERSATIONAL AI: SPEECH PROCESSING AND SYNTHESIS(UCS749)**

**Submitted by:**

**Pratham Agarwal**

**102103607**

**BE Fourth Year, COE**

Submitted to:

**Dr. BRAHMADESAM VENKATARAMAIYER R**



**Computer Science and Engineering Department**

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

**September 2024**

**Summary of Research Paper : Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition**

The paper "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition" introduces a dataset of short audio clips of spoken words designed for training and evaluating keyword spotting models. The dataset focuses on recognizing specific words efficiently, even in noisy environments, to support applications like voice-activated devices. It includes collection methods, quality control, and baseline model results.

**Comprehensive Report on Audio Classification Using CNNs with Mel-Spectrograms**

**1. Introduction**

This project focuses on building an audio classification model using a Convolutional Neural Network (CNN) with Mel-spectrograms as input features. The initial model is trained on the Speech Commands Dataset (speech\_commands\_v0.02) provided by TensorFlow, which consists of one-second-long audio clips of various spoken commands. The trained model is later used as a pretrained model for another dataset, demonstrating the effectiveness of transfer learning in audio classification tasks.

**2. Dataset Description**

**2.1 Initial Dataset - Speech Commands Dataset**

The dataset is automatically downloaded and extracted from the official TensorFlow repository. It consists of multiple folders, each representing a specific spoken command. Each folder contains audio files in .wav format.

**2.1.1 Data Preprocessing**

1. **Audio Loading**: Audio files are loaded using the librosa library at a sample rate of 16kHz.
2. **Duration Standardization**: All audio clips are trimmed or padded to ensure a uniform duration of 1 second (16,000 samples).
3. **Mel-Spectrogram Conversion**: Audio clips are converted to Mel-spectrograms with 32 Mel bands and a maximum frequency of 8kHz.
4. **Normalization**: Mel-spectrogram values are normalized to a range between 0 and 1 by dividing by 255.

**2.1.2 Data Augmentation**

To enhance the model's robustness and prevent overfitting, various data augmentation techniques are applied:

1. **Noise Addition**: Random noise is added to audio samples.
2. **Pitch Shifting**: The pitch of audio samples is randomly shifted by a few semitones.

**2.2. New Dataset - Secondary Audio Dataset**

1. **Data Collection:** The new dataset contains audio samples of spoken words or phrases, different from the initial Speech Commands Dataset, but formatted similarly with .wav files.
2. **Data Preprocessing**: The preprocessing steps are the same as for the initial dataset to ensure compatibility with the pretrained model. This includes:
   1. **Loading Audio:** Using librosa to load audio at 16kHz.
   2. **Standardizing Duration:** Ensuring all clips are 1 second long.
   3. **Converting to Mel-Spectrograms:** Using 32 Mel bands and 8kHz max frequency.
   4. **Normalization:** Normalizing Mel-spectrogram values to [0, 1].

**3. Exploratory Data Analysis (EDA)**

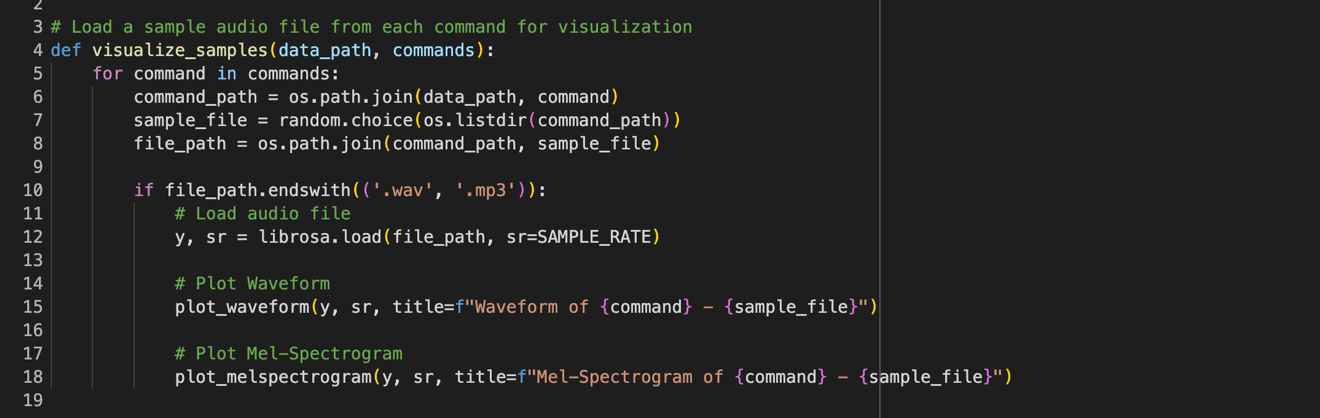
**3.1. Class Distribution**

The dataset contains several command classes, and their distribution is analyzed using a bar plot. The plot reveals any class imbalances that could affect the model's performance.



**3.2. Visualizing Audio Samples**

For each command category, a random audio sample is selected, and both its waveform and Mel-spectrogram are visualized.



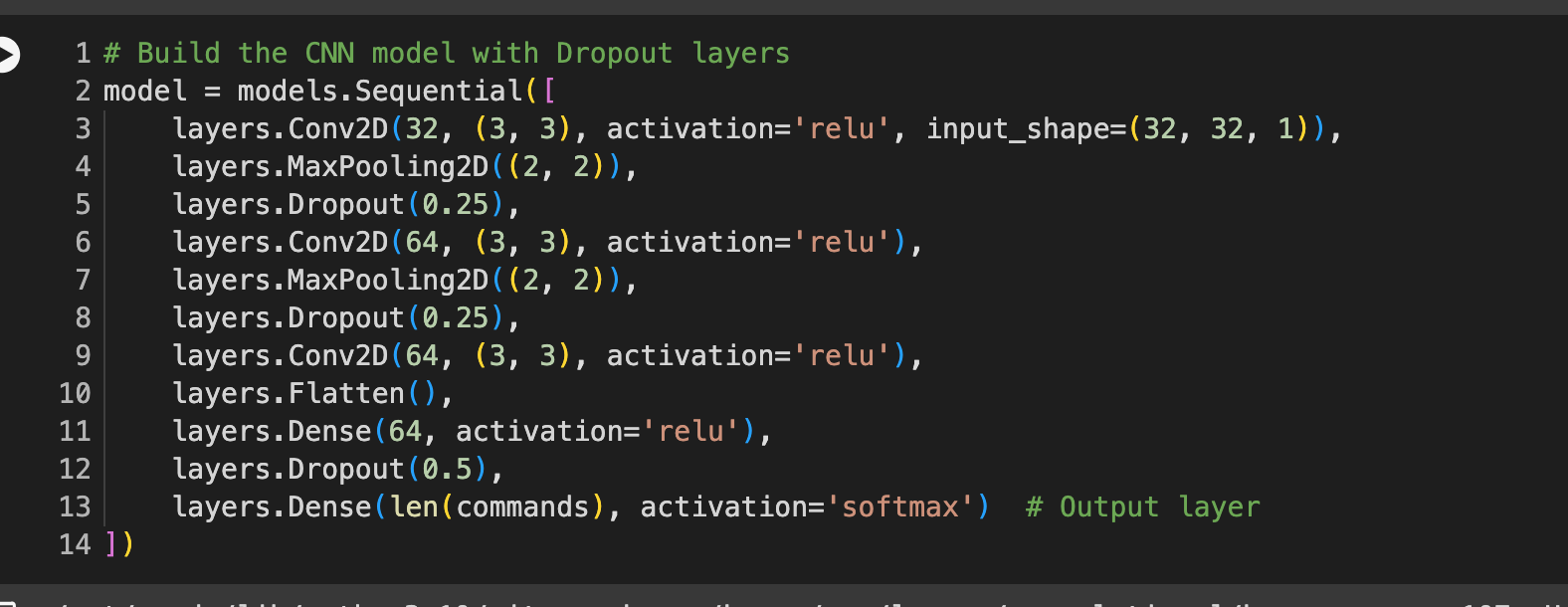
**3.3. Findings from EDA**

1. The dataset contains a reasonable distribution of commands with some minor class imbalances.
2. Mel-spectrograms provide a visual representation that captures the essential frequency patterns needed for classification.

**4. Model Building and Training**

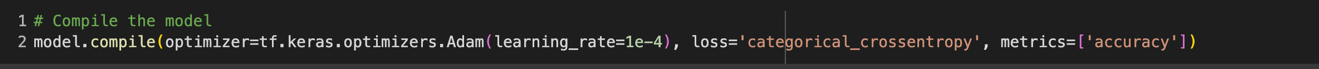
#### 4.1. Initial Model Architecture - CNN for Audio Classification

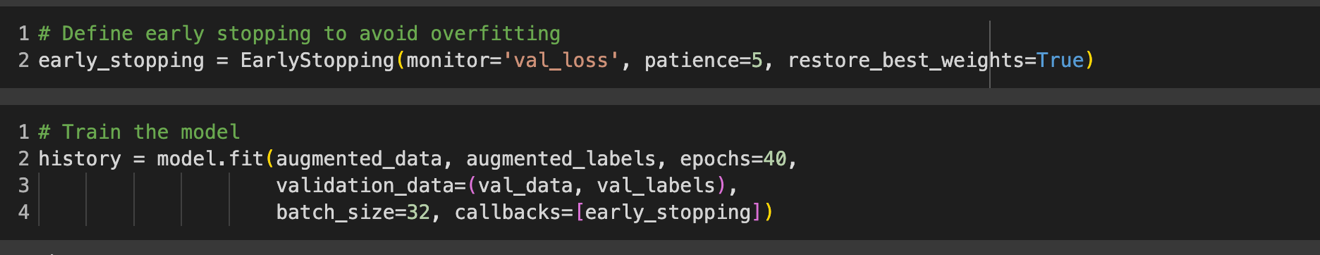
1. **Architecture**: The model consists of three convolutional layers with dropout and max pooling, followed by a dense layer and a softmax output layer.
2. **Training**: The model is trained using the Adam optimizer and categorical cross-entropy loss. An early stopping callback is used to prevent overfitting by monitoring the validation loss.
3. **Performance**: The model achieves an accuracy of 86% on the test dataset, indicating good generalization on unseen data.



**4.2. Training the Model**

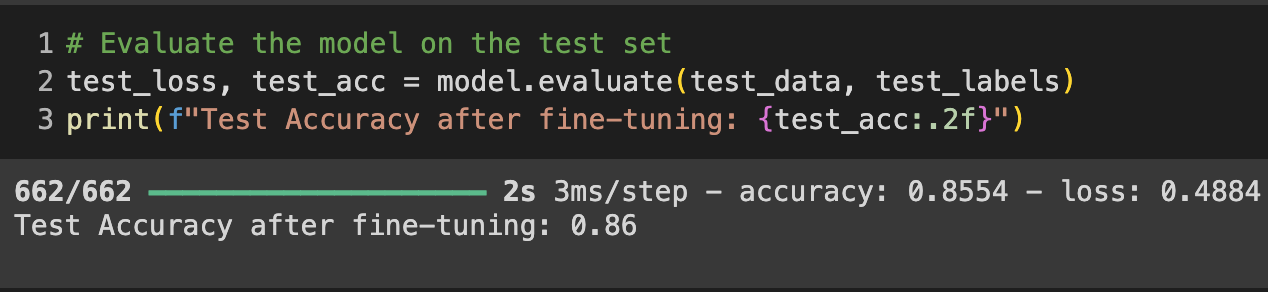
The model is trained using the Adam optimizer and categorical cross-entropy loss. An early stopping **callback** is used to prevent overfitting by monitoring the validation loss.





**4.3. Model Performance**

The model achieves an accuracy of **86%** on the test dataset, indicating good generalization on unseen data.



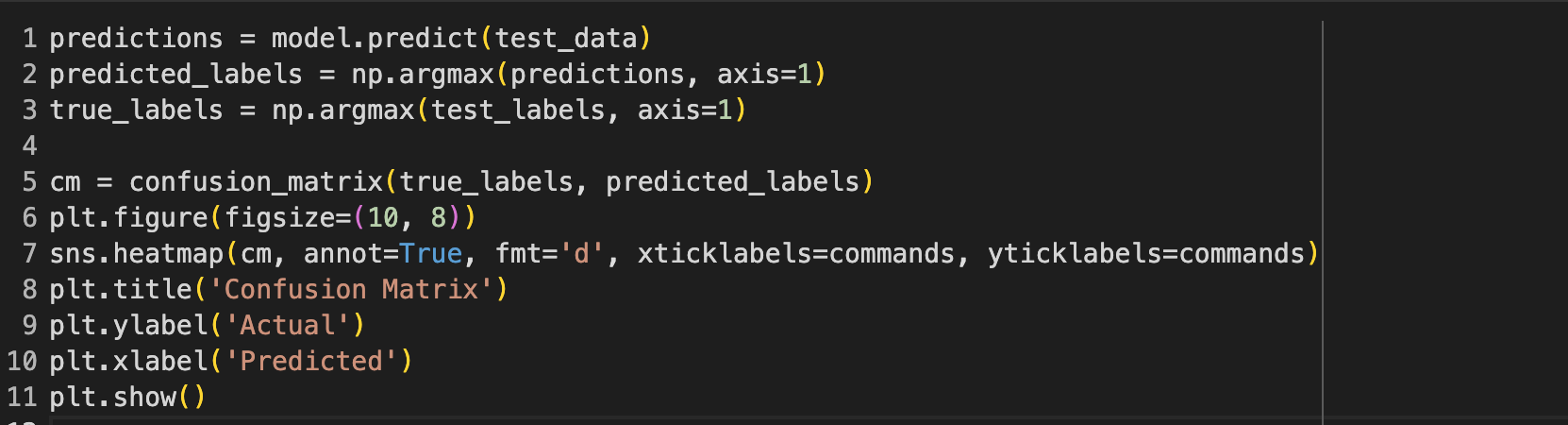
**4.4. Additional Models and Comparison**

Several models with slight variations in architecture are tested to find the best-performing model. The chosen model outperforms others in terms of both validation and test accuracy.

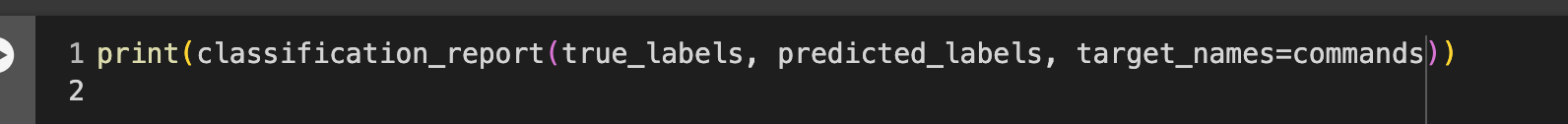
**5. Evaluation and Results**

**5.1. Confusion Matrix**

The confusion matrix provides a detailed breakdown of model performance across different classes, highlighting any misclassifications.



**5.2. Classification Report**

A classification report is generated to provide precision, recall, and F1-score for each command category.

**6. Conclusion**

1. **Model Performance**: The final model achieves an accuracy of **86%** on the test data, which is a strong result for audio classification tasks.
2. **EDA Insights**: Mel-spectrograms proved to be effective features for the classification of audio commands.
3. **Data Augmentation**: Augmentation techniques like adding noise and shifting pitch contributed to improved model generalization.
4. **Future Work**: Future improvements could involve experimenting with more complex architectures like recurrent neural networks (RNNs) or transformers, fine-tuning hyperparameters, and further data augmentation strategies.

**7. References**

1. TensorFlow Speech Commands Dataset: [Link](http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz)
2. Librosa Documentation: [Link](https://librosa.org/doc/latest/index.html)
3. TensorFlow Documentation: [Link](https://www.tensorflow.org/guide)