Dysarthria detection project.

Developing a system to detect dysarthric speech using machine learning techniques, specifically a Convolutional Neural Network (CNN) applied to Mel-Frequency Cepstral Coefficients (MFCCs) extracted from speech audio.

Let's break down the project and discuss its components:

1. Data Loading:

- The `load\_dysarthric\_data.m` function is responsible for loading the speech data. Currently, it contains placeholder code that generates random data. You'll need to replace this with actual data loading from your dataset.

- Your dataset is located at "D:\Amrita\OneDrive - Amrita university\Amrita\S6\Lab\19EAC386 Speech Processing\Project\Code\F\_Con". It contains audio samples from female control speakers (without dysarthria).

2. Feature Extraction:

- The `extract\_features.m` function extracts MFCC features from the audio data. It uses MATLAB's `mfcc` function to compute the coefficients along with their first and second derivatives (delta and delta-delta).

3. CNN Model:

- The `cnn\_dysarthria\_detection.m` and `dysarthria\_detection.m` scripts contain similar code for training a CNN on the MFCC features.

- The CNN architecture consists of several convolutional layers followed by batch normalization, ReLU activation, and max pooling. It ends with fully connected layers and a softmax classifier.

4. Model Training and Evaluation:

- The data is split into training and testing sets.

- The CNN is trained using the Adam optimizer with a specified learning rate and number of epochs.

- After training, the model's performance is evaluated on the test set, and metrics like accuracy and confusion matrix are computed.

5. Additional Analysis:

- Per-class accuracy is calculated and visualized.

- An error analysis is performed, showing misclassified samples.

- A learning curve is generated by training the model on increasing subsets of the training data.

6. Alternative Acoustic Model:

- There's also code for training and testing a Gaussian Mixture Model - Hidden Markov Model (GMM-HMM) acoustic model (`train\_acoustic\_model.m`, `test\_acoustic\_model.m`, `trainGMM.m`).

- This provides an alternative approach to the CNN for comparison.

To proceed with your project, here are some steps and suggestions:

1. Data Preparation:

- Load your actual audio files from the "F\_Con" directory.

- You'll need to create a balanced dataset by also including dysarthric speech samples. The current dataset only contains control (non-dysarthric) samples.

- Organize your data into a structure or table with columns for file paths and labels (dysarthric vs. non-dysarthric).

2. Feature Extraction:

- Use the `extract\_features.m` function to compute MFCCs for all your audio samples.

- Make sure to normalize or standardize your features if needed.

3. Model Training:

- Split your data into training and testing sets.

- Train the CNN model using `cnn\_dysarthria\_detection.m` or `dysarthria\_detection.m`.

- Experiment with different hyperparameters (learning rate, batch size, number of epochs) and model architectures to see what works best.

4. Evaluation and Analysis:

- Evaluate your model's performance on the test set.

- Analyze the confusion matrix, per-class accuracies, and learning curve.

- Perform error analysis to understand where your model is making mistakes.

5. Comparison:

- Train and evaluate the GMM-HMM acoustic model on the same data.

- Compare its performance with the CNN model.

6. Validation:

- Consider using k-fold cross-validation to get a more robust estimate of your model's performance.

7. Reporting:

- Document your methodology, experiments, and results.

- Discuss the strengths and limitations of your approach.

- Suggest potential improvements or future work.