README

Speaker Recognition System

This project implements a speaker recognition system using MFCC feature extraction, Gaussian Mixture Models (GMM) for training, and inference for speaker identification.

Directory Structure

```
project/
│
├── input_audio/              # Directory for input audio files (e.g., .wav 
files)
├── features/                 # Directory where extracted features are saved
├── gmm_files/                # Directory where trained GMM models are stored
├── trained_models/           # Directory for trained models (link provided 
below)
├── feature_extraction.py     # Code for feature extraction (Code 1)
├── model_training.py         # Code for training GMM models (Code 2)
├── inference_code.py         # Code for speaker inference (Code 3)
└── README.md                 # Documentation
```

Prerequisites

Ensure you have the following installed:

- **Python**: Version 3.7 or later
- **Required Libraries**:
  - numpy
  - scikit-learn
  - python_speech_features
  - scipy

Install the dependencies using the following command:

```
pip install numpy scikit-learn python-speech-features scipy
```

Steps to Run

1. Feature Extraction (needs to be executed first)

1. Place the input `.wav` files in the `input_audio/` directory and give the directory to the feature extraction code
2. Run the feature extraction script to extract MFCC features:
3. Extracted features will be saved in the `features/` directory as `.npy` file.

2. Model Training (execute this after making .npy files)

1. Ensure the extracted feature files (`.npy` and `.pkl`) are available in the `features/` directory, and give the features directory to the code as input.
2. Run the model training script
3. The trained GMM models will be saved in the `gmm_files/` directory as `.gmm` files.

3. Inference (Speaker Identification)

1. Make sure the trained `.gmm` files are available in the `gmm_files/` directory.
2. Place the test audio file (e.g., `test.wav`) in the `input_audio/` directory.
3. Run the inference script to identify the speaker
4. When prompted, enter the filename of the test audio (e.g., `test.wav`).
5. The script will display the predicted speaker's name based on the GMM models.

Techniques and Methods:

The mfcc features are extracted from the audio and are normalized and scaled in the first code. Later the values stored in a .npy file are used to train the model and the output of model is stored as .gmm file.
Now the gmm files are used test the given input audio file to identify the speaker.

Trained Models

You can download the pre-trained models (`.gmm` files) from the following link:

https://drive.google.com/drive/folders/1FdTnM0OuFV0h579jv70oRaFoURF0tPP4

To start the model from scratch the following dataset can be used(optional):

https://drive.google.com/drive/folders/1L1NESNH3eqYvbOZeygmd37L6phi6elaL?usp=sharing

Place the downloaded models in the `gmm_files/` directory.

Troubleshooting

- Ensure all directories (`input_audio/`, `features/`, `gmm_files/`) are correctly set up as per the directory structure.
- Verify that the required libraries are installed, and file paths are accurate.
- Ensure the input files are in `.wav` format and the sampling rate matches the required specifications.

DONE BY(from HITAM COLLEGE):
B.V. DATHATREYA - 22E51A6604
M. PAVAN RAJ - 22E51A6634
S. MIHIR MUDIRAJ - 22E51A6749