Voice Emotion Recognition Using MATLAB

README / Project Setup Guide

---

1. PROJECT OVERVIEW

This project implements a **Voice Emotion Recognition (VER)** system using MATLAB.
It processes speech signals, extracts MFCC-based features, trains a multiclass SVM classifier, and predicts emotions such as:

- Neutral
- Happy
- Sad
- Angry

---

2. SOFTWARE REQUIREMENTS

- MATLAB R2021b or later (tested on R2023)
- Toolboxes:
    - Signal Processing Toolbox
    - Audio Toolbox
    - Statistics and Machine Learning Toolbox

---

3. PROJECT FOLDER STRUCTURE (MANDATORY)

Create the project folder exactly like this:

first mat project 1/

```
│
├── code/
│   ├── split_ravdess.m
│   ├── step3_signal_analysis.m
```

```
│   ├── step4_preprocessing.m
│   ├── step5_framing.m
│   ├── step6_mfcc.m
│   ├── step7_delta_features.m
│   ├── step8_dataset_features.m
│   ├── step9_classifier.m
│   └── test_single_audio.m
│
├── data/
│   ├── raw/
│   ├── neutral/
│   ├── happy/
│   ├── sad/
│   └── angry/
│
├── results/
│   ├── features.mat
│   └── svm_model.mat
│
└── test_audio/
    └── test.wav
```

⚠️ **Always run MATLAB from the code folder**.

---

4. PATH SETUP (VERY IMPORTANT)

Before running any script, set the working directory:

cd 'C:\Users\sudee\OneDrive\Desktop\first mat project 1\code'

All scripts use **relative paths** like:

../data

../results

So this step is mandatory.

---

5. STEP-BY-STEP EXECUTION ORDER

STEP 1 — Dataset Splitting

Script: split_ravdess.m
Input: data/raw/*.wav
Output:

- data/neutral
- data/happy
- data/sad
- data/angry

Run:

run split_ravdess

STEP 2 — Signal Analysis (Visualization Only)

Script: step3_signal_analysis.m
Output: Time plot, FFT plot, Spectrogram

run step3_signal_analysis

STEP 3 — Preprocessing

Script: step4_preprocessing.m
Operations:

- Mono conversion

- DC removal
- Normalization
- Resampling to 16 kHz

run step4_preprocessing

STEP 4 — Framing

Script: step5_framing.m
Parameters:

- Frame length = 25 ms
- Frame shift = 10 ms

run step5_framing

STEP 5 — MFCC Extraction

Script: step6_mfcc.m
Output: MFCC matrix

run step6_mfcc

STEP 6 — Delta & Delta-Delta Features

Script: step7_delta_features.m

run step7_delta_features

STEP 7 — Dataset Feature Matrix (MOST IMPORTANT)

Script: step8_dataset_features.m

Output:

results/features.mat

Contains:

- X → Feature matrix

- Y → Label vector

run step8_dataset_features

STEP 8 — Classification

Script: step9_classifier.m

Output:

- Accuracy
- Confusion matrix
- Trained model:

results/svm_model.mat

run step9_classifier

STEP 9 — Test with New Audio

Script: test_single_audio.m

Place a new speech file here:

test_audio/test.wav

Run:

run test_single_audio

---

6. PURPOSE OF results FOLDER

results/

├── features.mat → extracted MFCC + Δ + ΔΔ features

└── svm_model.mat → trained multiclass SVM model

These files:

- Avoid re-training every time
- Are used for testing and Simulink inference

- Represent the **final output of the project**

---

7. HOW TO ADD MORE SAMPLES (OPTIONAL)

Step A — Download More Data

Download the **RAVDESS speech dataset** from the official source:

🔗 **RAVDESS Official Download Page:**
https://zenodo.org/record/1188976

Download:

Audio_Speech_Actors_01-24.zip

Step B — Extract and Copy WAV Files

From each Actor_XX folder:

- Copy **only .wav files**
- Paste them into:

data/raw

⚠️ Do NOT paste actor folders. Only WAV files.

Step C — Re-run Pipeline

After adding new samples:

run split_ravdess

run step8_dataset_features

run step9_classifier

---

8. IMPORTANT NOTES (READ THIS)

- Accuracy may be low due to:
    - Small dataset

    - Mean-based feature statistics
- This project demonstrates **pipeline correctness**, not commercial-grade accuracy.
- External random audio may be predicted as Neutral due to dataset mismatch.

---

9. PROJECT STATUS

✓ Dataset preparation
✓ Signal processing
✓ Feature extraction
✓ Multiclass classification
✓ Testing with unseen audio

Project is complete and submission-ready.

---

10. CONTACT / EXTENSIONS

Possible future work:

- Deep learning (CNN / LSTM)
- Pitch & energy features
- Real-time microphone input
- Simulink inference model