Multimodal Emotion Recognition System (Face + Speech) Project Report

Abstract

This project implements two practical applications built around emotion recognition: (1) a **multimodal emotion detector** that predicts a user's emotion from **facial expressions (webcam)** and **speech (microphone)** in a real-time workflow, first detecting and stabilizing the face emotion, then detecting short audio to classify speech emotion, and finally displaying both results so the user can compare how emotion appears across modalities; and (2) a **gamified emotion system** that turns facial emotion recognition into a two-player game where the program sets a target emotion and scores each player based on how confidently their live facial expression matches it, creating an interactive way to demonstrate and test the model's behavior.

1. Project Objectives

- Real-time emotion recognition from face and speech.
- Shared 6-class label space across modalities for consistency.
- Runnable demos (multimodal and game mode) using saved trained weights.
- Clear separation between training notebooks and inference scripts.

2. Problem Definition and System Requirements

The work can be framed as two applied AI tasks built on the same core capability. Task A multimodal inference: given a live stream of video frames from a webcam and a short segment of microphone audio, the system must output two emotion predictions, one from the face stream and one from the speech stream, each as a 6-class label with a confidence/probability, and present the result in a way that is usable during a live demo with clear UI states, stable outputs, and predictable interaction flow. Task B gamification: given two faces visible simultaneously and a system-selected target emotion, the system must continuously estimate each player's emotion confidence and compute a fair round outcome by comparing how well each player matches the target, then update scores and declare a winner at the end of the game. Across both tasks, the project enforces a shared 6-emotion label space for consistency and comparability, relies on real-time performance constraints frame skipping / lightweight face detection;

short audio capture rather than long continuous recording, and requires specific local model artifacts to run (FER weights file and SER model folder/weights), plus hardware configuration (camera/microphone IDs) to ensure the system starts correctly on the target machine.

3. Model Deployment and Runtime Workflow

3.1 Emotion classes and label alignment

The system uses a shared 6-class emotion space across both face and speech so outputs are comparable: Angry, Fear, Happy, Neutral, Sad, Surprise. The project intentionally excludes Disgust to avoid mismatched label sets between modalities and to keep the multimodal output consistent.

3.2 Training data used

Two independent training pipelines are used (one per modality).

- **Face Emotion Recognition (FER):** trained on **RAF-DB** and exported as a ResNet-based classifier checkpoint.
- **Speech Emotion Recognition (SER):** trained using multiple emotion-speech datasets (**RAVDESS, CREMA-D, MELD**) and exported as a WavLM-based classifier with its saved model assets.

3.3 Runtime deployment artifacts

- **FER weights:** rafdb_resnet50_6classes_weighted.pth
- **SER assets:** wavlm_ser_model/ (Hugging Face-style model assets) and wavlm_ser_model.pth (PyTorch weights)

3.4 Multimodal inference workflow (Face + Speech)

FER (Face Emotion Recognition)

Face emotion inference is deployed as a real-time webcam pipeline that performs **face detection → face crop → preprocessing → ResNet-based classification → stable label output**. ResNet-50 was selected as the FER backbone because it is a strong, well-tested CNN architecture and works reliably as a **pretrained feature extractor** for facial-expression classification when fine-tuned on a task-specific dataset. Fine-tuning was done by **replacing the final fully-connected layer** to match the **6 aligned emotion classes**, then training in **two phases**: first training the classification head, and then fine-tuning deeper layers to adapt higher-level visual features to RAF-DB facial expressions. To handle class imbalance, the training used **weighted cross-entropy** and **oversampling** for underrepresented classes (Fear/Angry/Sad), with standard augmentation to improve robustness. The trained checkpoint is saved as rafdb_resnet50_6classes_weighted.pth and deployed in the realtime scripts (run_webcam.py, emotion_game.py, multimodal_system.py) using the same inference preprocessing: face crop → resize to 224×224 → ImageNet normalization → softmax probabilities. Runtime stability is improved using (a) **frame skipping + cached predictions** in the webcam-only demo and (b) a **stability buffer** in the multimodal demo that locks the face emotion only when the same label persists for a fixed window, reducing flicker in live use.

The same FER deployment is reused for gamification. The game requires **two faces** and assigns players based on left-to-right ordering (leftmost = Player 1, rightmost = Player 2) to prevent identity swapping across frames. (emotion_game.py) During each round, instead of using only the top predicted label, the implementation extracts the **probability of the target emotion** for each player (get_confidence) and accumulates these scores over the round; the round winner is decided using the **average confidence** across frames for stability and fairness.

SER (Speech Emotion Recognition)

WavLM (microsoft/wavlm-base) was selected as the SER backbone because it provides pretrained speech representations that can be adapted efficiently to emotion classification with limited labeled emotion data, avoiding training an audio model from scratch. Fine-tuning was performed as a supervised 6-class audio classification task by combining multiple datasets (RAVDESS, CREMA-D, MELD) and mapping their original labels into the shared emotion space; “Disgust” is removed to keep the final label set consistent with FER. To reduce speaker leakage, the dataset preparation enforces **speaker-disjoint train/val/test splits**, and training uses **weighted cross-entropy** to address class imbalance; the best checkpoint is selected using **macro-F1** to reflect balanced performance across classes. An optional final adaptation step is implemented as **head-only fine-tuning** (freezing the WavLM encoder) on a small custom dataset, with early stopping to reduce overfitting. For deployment, the multimodal demo records a short mic segment in a background thread, audio is recorded using sounddevice at the microphone’s **native sampling rate** (configured as 48 kHz in the code), then flattened and passed into the SER inference function. Then applies peak normalization, resamples to **16 kHz** (WavLM requirement), extracts features using the saved local model assets, and outputs the predicted speech emotion with confidence; the deployed artifacts are wavlm_ser_model/ (feature extractor/config) and wavlm_ser_model.pth (weights).

UI and interaction flow (Multimodal + Game)

The deployment includes a UI-driven interaction model so the system behaves predictably during demonstration and does not mix partial results. In the multimodal demo (multimodal_system.py), the OpenCV window (“Multimodal System”) runs as a **state machine**: **WAITING_FACE** continuously detects a face and displays the instruction “HOLD AN EXPRESSION TO LOCK IN”; once the face emotion stabilizes and locks (via the 10-frame rule), the system enters **LISTENING**, dims the screen, shows the locked face label, and prompts “SPEAK NOW” while the audio thread records; after speech prediction, **SHOW_RESULT** renders a split-screen output displaying **FACE** and **VOICE** predictions with confidence, then automatically resets back to face detection after a fixed display time.

The gamification module (emotion_game.py) reuses the same FER pipeline but wraps it in a structured, round-based game loop: the system waits until **two faces** are detected, assigns **Player 1/Player 2** consistently using **left-to-right ordering** to avoid identity swapping, then runs a **countdown** before starting each round with a **target emotion prompt** displayed on screen. During the active round, the game does not rely only on the top predicted label; it extracts and accumulates each player’s **confidence for the target emotion** (get_confidence) across frames, and determines the round winner using the **average confidence** to make scoring stable and fair. The OpenCV window (“Facial Emotion Battle”) continuously overlays the target emotion, timers, live scores, and winner/final-result messages so the full interaction is understandable in real time.

4. Experiments (Offline)

FER (Face Emotion Recognition)

Face emotion detection was trained and tested on the RAF-DB dataset. RAF-DB has 7 emotions, but the **Disgust** class was removed so the final setup uses **6 classes**: Surprise, Fear, Happy, Sad, Angry, Neutral. The class mapping used was **0: Surprise, 1: Fear, 2: Happy, 3: Sad, 4: Angry, 5: Neutral**. RAF-DB has fewer samples for **Fear, Angry, and Sad**, so oversampling was applied to reduce imbalance: **Fear repeated 5×, Angry 3×, Sad 2×**. A pretrained **ResNet-50** was fine-tuned by replacing the final layer for 6-class output and training in two phases: **10 epochs** for the classifier head and **35 epochs** for full fine-tuning. Training used **Adam** optimizer (LR **1e-3** then **1e-4**), **batch size 32**, **weighted cross-entropy loss**, and augmentation with **Albumentations**. Final offline accuracy was around **95% (train), 90% (validation), 88% (test)**, and the final saved model is rafdb_resnet50_6classes_weighted.pth.

SER (Speech Emotion Recognition)

Speech emotion detection was trained using a combined dataset made from **RAVDESS, CREMA-D, and MELD**. These datasets are different in nature (acted speech vs more conversational speech), so the task becomes harder and accuracy is expected to be lower than training on one single clean dataset. All labels were mapped to the same 6 emotion classes used in FER. Examples of mapping: **MELD “joy” → happy** and **CREMA-D “calm” → neutral**. Training and evaluation used **speaker-disjoint splits** so the same speaker does not appear in both training and testing. Because of dataset diversity, offline accuracy was not very high after training on the merged data. After that, the classifier head was fine-tuned using a small personal recorded dataset, and the accuracy on that testing was around **67%**. The saved SER model artifacts are wavlm_ser_model/ and wavlm_ser_model.pth.

5. Results (Online / Real-Time)

FER real-time results (webcam + gamification):

After fine-tuning, the FER model was tested in real time using an external webcam and it was able to detect facial emotions correctly in live conditions. The same FER model was then used to build the gamification feature. In the game, two faces are detected and players are assigned using **left-to-right order** (left person = Player 1, right person = Player 2) to keep player identity stable. Each round has a target emotion, and scoring is done using the **confidence value of the target emotion** over multiple frames, then averaging it to decide the winner. This made the game stable and fair, and it worked successfully in real-time testing.

SER real-time results (microphone):

After training and saving the SER model, it was tested using an external microphone for real-time speech emotion prediction. The system produced accurate results when emotions were spoken clearly and also returned meaningful probability distributions (for example, speaking in a sad tone might give **~80% sad**, **~15% neutral**, **~5% fear**). Audio inference stayed consistent because audio was **peak-normalized** and **resampled to 16 kHz**, which matches the WavLM requirement and reduces microphone device differences.

Multimodal real-time results (final feature):

Finally, FER and SER were combined into the main multimodal system where emotion is detected from both **face and speech** in one run. The webcam detects and stabilizes the face emotion first, then the microphone records speech and predicts the speech emotion, and both results are shown together. This confirmed that both models work together properly in a real-time setup and demonstrate the main goal of multimodal emotion detection.

6. Limitations

- **Subjective labels:** Emotion labels are not fully objective, so dataset annotations can be noisy or inconsistent.
- **Acted speech mismatch:** Acted datasets like **RAVDESS** and **CREMA-D** do not perfectly represent natural real-life emotions and speaking style.
- **Camera sensitivity:** FER accuracy is affected by lighting, face angle, distance, motion blur, and occlusions (hands, glasses, hair, masks).

- **Audio sensitivity:** SER accuracy depends on background noise, room echo, microphone quality, and speaking volume/clarity.
- **Simple VAD trigger:** The current RMS-based voice activity detection can trigger incorrectly in loud environments or miss speech when the voice is soft.

7. Future Work

- Replace Haar-cascade face detection with a more robust detector (e.g. RetinaFace) to improve performance under different lighting, angles, and partial occlusion.
- Add stronger temporal smoothing for FER (e.g., weighted moving average or majority voting over a sliding window) to further reduce prediction flicker in real time.
- Improve speech triggering by replacing simple RMS-based VAD with a proper voice activity detector (e.g., WebRTC VAD) to reduce false triggers in noisy environments and reduce missed speech.
- Add noise handling for SER using basic denoising / noise-reduction or training with noise augmentation so the model generalizes better in real environments.
- Implement an actual multimodal fusion strategy to produce a single final emotion output instead of only showing face and speech results separately.
- Collect more real-world speech samples for the target setup (same mic/environment) and fine-tune further to improve SER accuracy beyond the current head-only tuning.

8. Conclusion

This project successfully implemented a complete emotion recognition system with two main features: multimodal emotion detection and gamification. A ResNet-50 model was fine-tuned on RAF-DB for face emotion recognition and worked well both offline and in real-time webcam testing. A WavLM-based model was trained for speech emotion recognition using multiple datasets and then improved using a small personal recorded dataset, and it produced meaningful predictions with confidence scores during real-time microphone inference. Finally, both models were integrated into a single multimodal pipeline where face and speech emotions are detected in one workflow, and the same face model was reused to build a stable two-player emotion game. Overall, the project demonstrates that the trained models and real-time deployment work properly and meet the main goal of detecting emotions from both face and speech with an additional interactive gamification feature.

References

- **RAF-DB (FER)** — *Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild* (CVPR 2017) https://openaccess.thecvf.com/content_cvpr_2017/papers/Li_Reliable_Crowdsourcing_and_CVPR_2017_paper.pdf
- **RAVDESS (SER)** — *The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)* (PLOS ONE 2018) https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0196391
- **CREMA-D (SER)** — *CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset* (2014) https://pmc.ncbi.nlm.nih.gov/articles/PMC4313618/
- **MELD (SER)** — *MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations* (2019) https://aclanthology.org/P19-1050/

- **ResNet-50 backbone (FER)** — *Deep Residual Learning for Image Recognition* https://arxiv.org/abs/1512.03385
- **WavLM backbone (SER)** — *WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing* https://arxiv.org/abs/2110.13900
- **Multimodal emotion recognition overview / fusion survey** — Wu et al. (2014) *Survey on audiovisual emotion recognition: databases, features, and data fusion strategies* https://www.cambridge.org/core/journals/apsipa-transactions-on-signal-and-information-processing/article/survey-on-audiovisual-emotion-recognition-databases-features-and-data-fusion-strategies/5BA206CFFEC3BAE321842B8EB820E179
- **Fine-tuning SSL speech models for SER (optional but relevant)** — Diatlova et al. (2024) *Adapting WavLM for Speech Emotion Recognition* https://arxiv.org/abs/2405.04485