Dizon, Rohann Gabriel D.
Dulatre, Rainier A.
Perez, Patrick Hans A.
Pineda, Juan Roberto G.

Exploring Multimodal Emotion Recognition

The prototype we built combines three modalities for emotion recognition: facial expressions using DeepFace, hand gestures with MediaPipe, and voice recognition using HuggingFace.

Data for Testing

Testing was done using live webcams and audio samples through our devices' cameras and microphones. No external datasets or recordings were used for testing since we built the program to essentially utilize our facial expressions, hand movements, and voices in real-time.

Features Extracted

The system extracts different types of features depending on the modality. DeepFace processes the face image through a convolutional neural network (CNN) to generate embeddings, or compact numerical representations that capture subtle patterns like eye curvature, lip shape, and brow tension, and compares these against learned patterns from large datasets to predict a dominant emotion.

MediaPipe Hands detects different hand landmarks by using lightweight neural networks that generate heatmaps for joint positions and refine them into coordinates. These landmark features are geometric, essentially x and y coordinate positions and visibility scores, which can be used to infer emotions if any of the gestures detected satisfy a custom rule that is set for the coordinates.

Audio features were extracted using HuBERT. Key features include MFCC, prosodic features, and contextual speech representations. MFCC (Mel-frequency cepstral coefficients) is used to represent the audio in a form that the computer can understand. Prosodic features involve the analysis of pitch, stress, rhythm, and volume to predict emotion. Contextual speech representations provide a deeper understanding of the projected emotion by capturing the entire sequence of sound.

Models and Tools Used

For facial emotion recognition, DeepFace was used, which leverages CNNs trained on large-scale datasets to extract embeddings from facial images and match them to emotion categories. To capture gestures, MediaPipe provided efficient real-time landmark detection for Hands, offering hand keypoints. HuggingFace's HuBERT was employed as a pre-trained model for vocal emotion recognition, which converts raw audio into meaningful embeddings through self-supervised learning, which are then fine-tuned to classify vocal emotions. Finally, OpenCV served as the backbone for handling video input, performing preprocessing tasks like color space conversion, and rendering results with visualization overlays.

Processing Pipeline

The processing pipeline begins with data capture, where both video and audio streams are recorded in real time. The video frames are handled by OpenCV, which reads from the webcam and performs basic preprocessing such as converting frames from BGR to RGB for compatibility with the models. Every few frames, DeepFace analyzes the cropped facial region using a CNN to generate embeddings and predict the dominant facial emotion. At the same time, MediaPipe Hands extracts hand landmarks from each frame, which are passed into rule-based functions to detect gestures like thumbs up and the rock and roll hand sign. Parallel to the visual stream, the audio feed from the microphone is processed by HuBERT, which transforms the raw waveform into embeddings that capture the acoustic and prosodic features of speech. These embeddings are then mapped to vocal emotion categories. Finally, the outputs of these facial, gestures, and vocal emotions are integrated and displayed on screen through OpenCV overlays, creating a multimodal, real-time system for emotion recognition.

Experience

One of the main challenges faced was dealing with inaccuracies and occasional misclassifications across the different modalities. Facial and gesture recognition in particular proved to be quite sensitive to lighting conditions and camera angles, which meant that even small changes in the environment could significantly affect detection accuracy. The quality of the hardware also plays a role, wherein lower-resolution cameras reduced the precision of landmark tracing, and low-quality microphones also introduced inaccuracies in vocal emotion recognition. On the software side, setting up the environment sometimes led to dependency conflicts and compatibility issues across different libraries. Another limitation was that we were not fully well-versed in machine learning and affective computing, so understanding model behavior and making adjustments was quite challenging.

Despite these issues, the system ultimately worked, even if not perfectly. This hands-on experience also led to a clear understanding of how multimodal emotion recognition and affective computing can be applied in practice. This practical exposure was valuable not only for testing the feasibility of different models and frameworks but also for gaining insights into how such systems can be built and improved over time.