**"A Web-Enabled Real-Time Multimodal Emotion Detection System Integrating BERT Text Embeddings and CNN-Based Speech and Facial Models"**

Sreeja Kamera1, Dhanusha Challa1, Hannah Bwalya1, D. Ramesh Reddy2, Prof. Prakash Kodali1* 1*

Department of ECE, NIT Warangal, Warangal, Telangana, India. 2Department of ECE, VNR VJIET, Hyderabad, Telangana, India.

*Corresponding author(s). E-mail(s): kprakash@nitw.ac.in; Contributing authors: sreejackamera@gmail.com; dhanusha2394@gmail.com; bwalyahannah@gmail.com; rameshreddyd@vnrvjiet.in;

# Additional Methods

## 1.1 Facial Emotion Recognition (FER)

This module was designed to classify static images of human faces into one of seven discrete emotion categories.

### *1.1.1 Dataset Description (CK+)*

The Extended Cohn-Kanade (CK+) dataset was used for this task. It is a widely-used, high-quality benchmark for FER, containing 981 image sequences from 123 subjects. For this project, we used the peak-expression frames, which are labeled with seven emotions: anger, disgust, fear, happy, neutral, sad, and surprise. The dataset was manually split, allocating 80% of the images for training and 20% for validation.

Dataset Link: CKPLUS

### *1.1.2 Image Preprocessing and Augmentation*

A rigorous preprocessing pipeline was established to ensure model robustness and efficiency.

1. **Image Formatting:** All images were converted to grayscale (1 color channel) and uniformly resized to (48 x 48) pixels. This creates a consistent, low-dimensional input tensor (48 x 48 x 1).
2. **Normalization:** Pixel intensity values, originally in the 0-255 range, were rescaled to a floating-point range of [0, 1] using the following formula. This normalization is essential for stabilizing gradient descent during training.

$$P_{\text{ixel}_{norm}} = \frac{P_{\text{ixel}_{original}}}{255.0}$$

3. **Data Augmentation:** To prevent overfitting on the relatively small dataset, Keras's `ImageDataGenerator` was applied to the training set. This introduced random transformations, including rotations (±15°), width/height shifts (±0.15), shears (±0.15), zooms (±0.15), and horizontal flips. The validation set was not augmented.

### *1.1.3 2D-CNN Architecture and Training*

A deep 2D-Convolutional Neural Network (CNN) was designed using the Keras Sequential API to learn the spatial hierarchies of facial features.

**Architecture:** The model is composed of three main convolutional blocks, each followed by a pooling and dropout layer for regularization as shown in Fig. S1.

- **Block 1:** This block consists of a `Conv2D` layer with 512 filters and a (5x5) kernel, followed by Batch Normalization. This is connected to a second Conv2D layer with 256 filters and a (5x5) kernel, also followed by Batch Normalization. The block concludes with a MaxPooling2D layer and a Dropout layer with a rate of 0.25.
- **Block 2:** This block includes a Conv2D layer with 128 filters and a (3x3) kernel, followed by Batch Normalization. This is connected to another Conv2D layer with 128 filters and a (3x3) kernel, also followed by Batch Normalization. The block concludes with a MaxPooling2D layer and a Dropout layer with a rate of 0.25.
- **Block 3:** The final convolutional block consists of a Conv2D layer with 256 filters and a (3x3) kernel, followed by Batch Normalization. This is connected to a Conv2D layer with 512 filters and a (3x3) kernel, also followed by Batch Normalization. The block concludes with a MaxPooling2D layer and a Dropout layer with a rate of 0.25.
- **Classifier Head:** The final 6 x 6 x 512 feature maps are passed to a Flatten layer, followed by a Dense block (256 units, Relu, Batch Normalization, Dropout (0.25)).
- **Output Layer:** A Dense (7, activation='softmax') layer produces a probability distribution across the seven emotion classes.

**Training:** The model was compiled using the adam optimizer and categorical_crossentropy loss. We considered other optimizers like Nadam but selected Adam for its stable performance. Training was governed by two callbacks: EarlyStopping (to halt training when validation accuracy plateaued) and ReduceLROnPlateau (to fine-tune the learning rate). This process achieved our final validation accuracy of 98.48%.

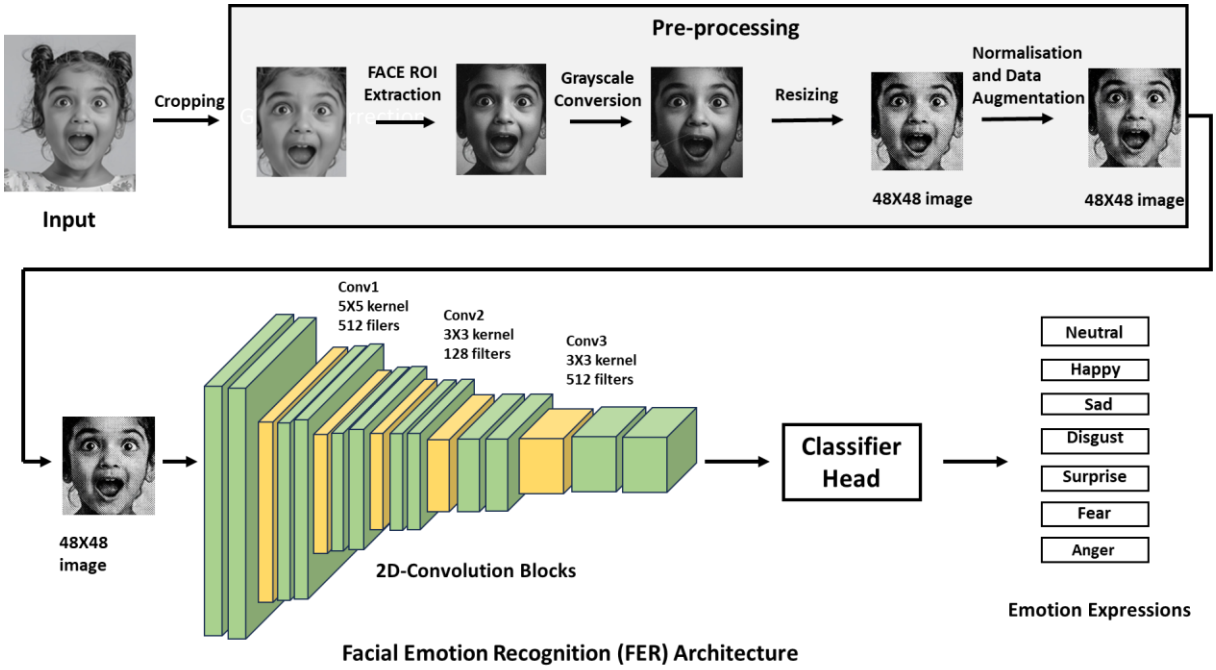


Figure S1: Architecture of 2D-CNN-based Facial Emotion Recognition System.

**Real-Time Implementation:** The final, trained model was saved as an `.h5` file. A real-time prediction script was developed using OpenCV. This script uses a Haar Cascade classifier (`haarcascade_frontalface_default.xml`) to detect a face in the webcam feed, extracts the 48 x 48 Region of Interest (ROI), performs the same normalization as the training data, and feeds the processed image into the loaded model to predict and display the emotion in real-time.

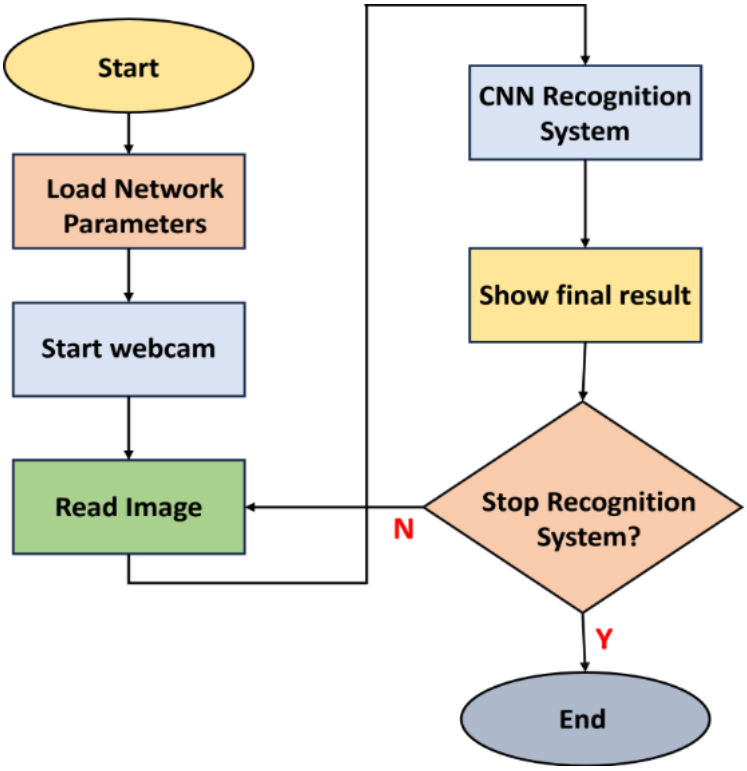


Figure S2 Flowchart of the Real-Time Facial Emotion Recognition Process

Fig. S3 illustrates the feature extraction process in the proposed 2D-CNN-based Facial Emotion Recognition model. The original facial image is first preprocessed through grayscale conversion, resizing, and normalization to ensure uniform input. The initial convolution layers extract low-level features such as edges and contours, which are then refined through pooling operations to reduce dimensionality while retaining important information. As the network deepens, higher-level layers capture more complex patterns such as facial structures and expression-specific features, enabling accurate emotion classification.

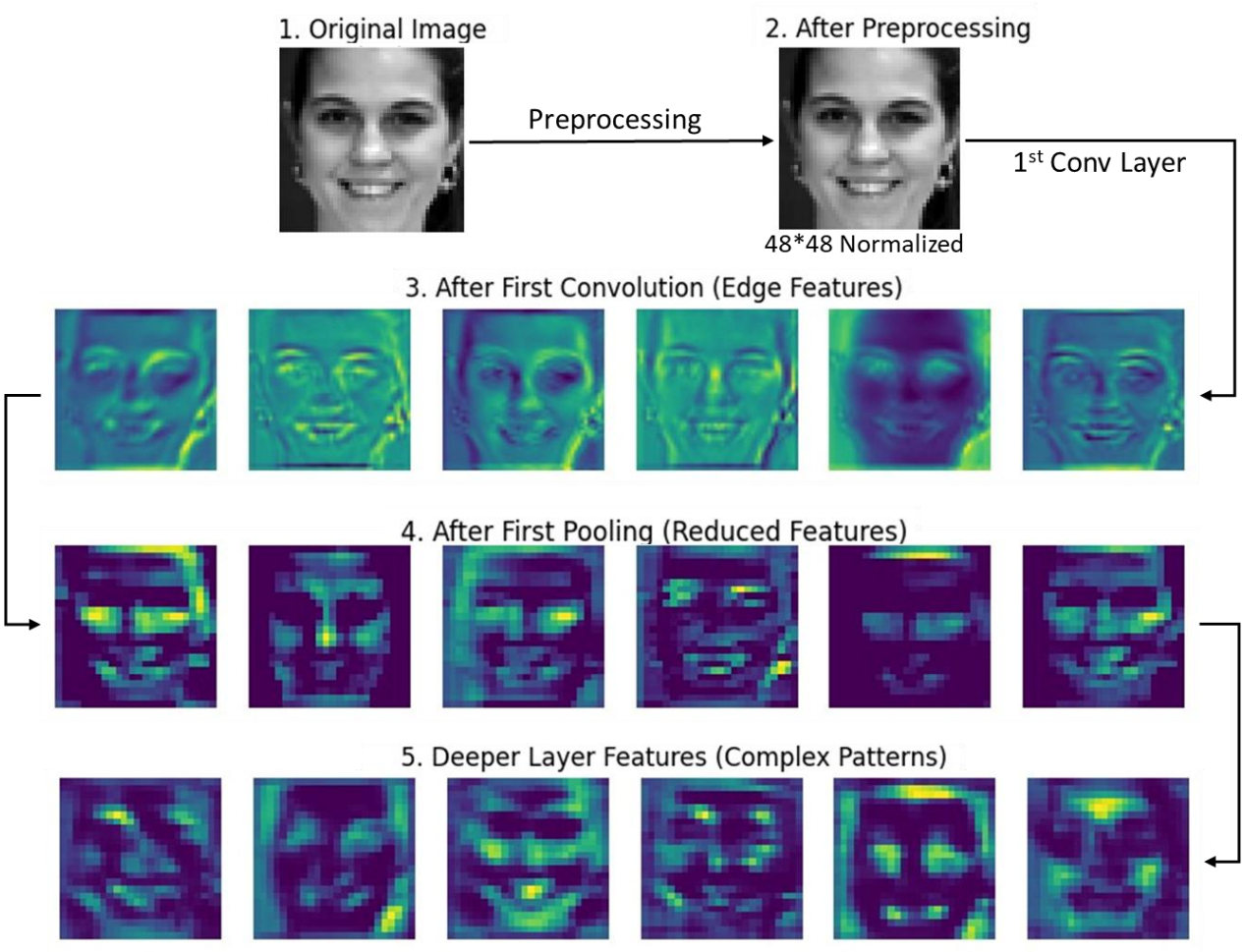


Figure S3 Feature Extraction Process in 2D-CNN for Facial Emotion Recognition

## 1.2 Speech Emotion Recognition (SER)

The Speech Emotion Recognition (SER) module aims to automatically classify emotions from human speech using a deep learning-based Convolutional Neural Network (CNN) model. The overall workflow consists of four major stages: data preprocessing, feature extraction, model architecture and training, and evaluation.

### *1.2.1 Dataset Description (Ravdess, TESS, SAVEE)*

To build a robust, speaker-independent model, a composite dataset was created by combining three publicly available audio-visual datasets:

- Ravdess (Ryerson Audio-Visual Database of Emotional Speech and Song)(1440)
- TESS (Toronto Emotional Speech Set)(5600)
- SAVEE (Surrey Audio-Visual Expressed Emotion)(480)

Dataset Links: Ravdess , SAVEE , TESS

This combined dataset provided a diverse set of speakers and recording conditions, labeled with eight emotions: angry, calm, disgust, fear, happy, neutral, sad, and surprise.

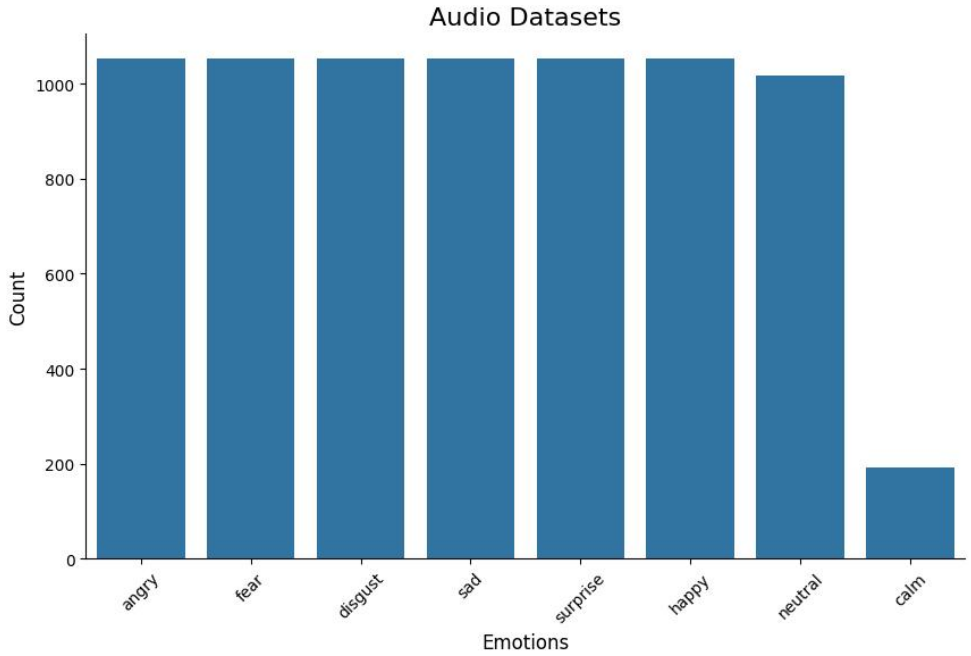


Figure S4: Emotion Class Distribution of the Combined Audio Datasets (Ravdess, TESS, SAVEE)

### *1.2.2 Audio Feature Extraction*

Instead of using 2D spectrogram images, our methodology relies on an efficient 1D feature vector that summarizes the acoustic properties of the entire audio clip. This vector is created using the `librosa` library.

For each audio file, a set of features is extracted across all its frames. The np.mean() function is then used to find the average of these features over time, collapsing them into a single 1D vector. This process is represented by the formula:

$$f_{\text{vector}} = \frac{1}{T}\sum_{t=1}^{T} f(t)$$

where:

- $f(t)$represents the feature vector at time frame $t$,
- $T$is the total number of frames in the audio segment,
- $f_{\text{vector}}$is the final aggregated feature vector used as input to the 1D-CNN model.

The final vector contains 162 concatenated features:

- **Mel-Frequency Cepstral Coefficients (MFCCs):** 20 features

- **Mel Spectrogram:** 128 features
- **Chroma STFT:** 12 features
- **RMS Energy:** 1 feature
- **Zero-Crossing Rate (ZCR):** 1 feature

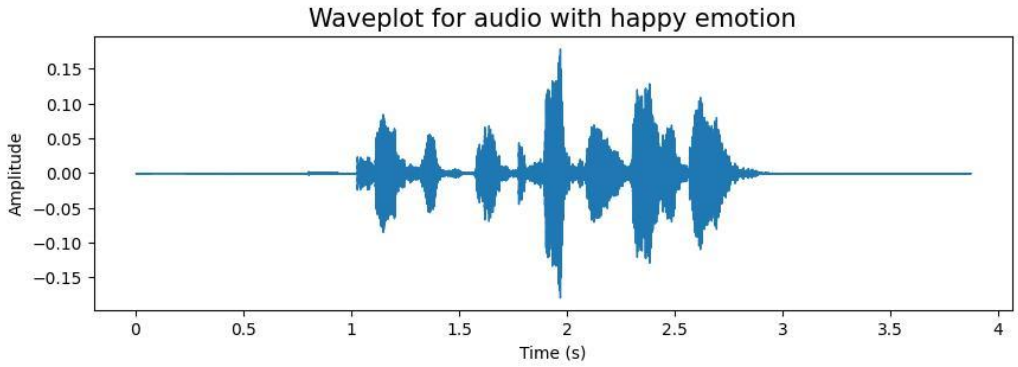


Figure S5: Waveform Visualization of a "Happy" Emotion Audio Sample

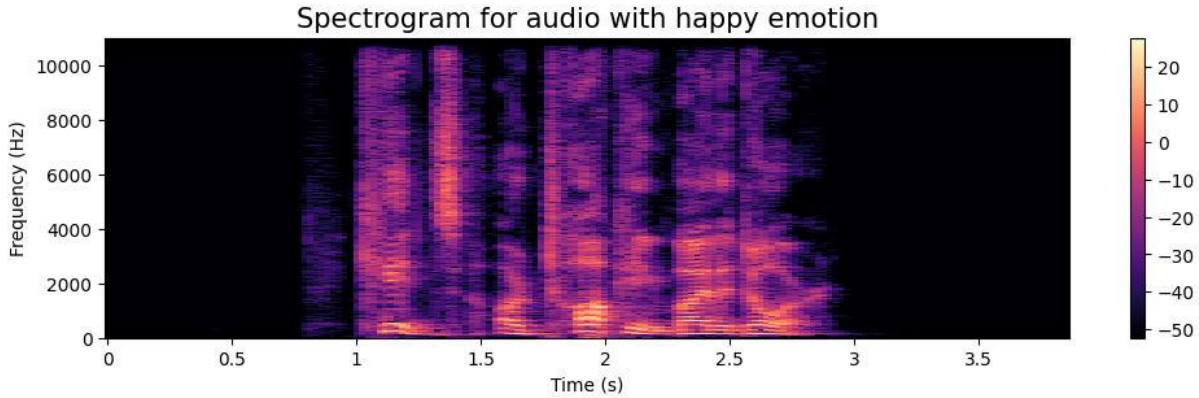


Figure S6: Spectrogram Visualization of a "Happy" Emotion Audio Sample

### *3.2.3 Data Augmentation*

To build a robust model, we tripled the size of the training dataset by applying two augmentation techniques to the raw audio *before* the 162 features were extracted:

1. **Noise:** Random Gaussian noise was added to a copy of the audio file.
2. **Stretch & Pitch:** A time-stretch and a pitch-shift were applied to another copy.

This resulted in three feature vectors (original, noise, stretch/pitch) for every audio file, making the model more resilient to background noise and variations in speech tempo.

### *1.2.4 1D-CNN Architecture and Training*

A 1D-Convolutional Neural Network (1D-CNN) was chosen, as it is highly effective at finding temporal patterns in sequential data like our 162-feature vector.

- **Preprocessing:** Before training, all feature vectors were scaled using StandardScaler (fit on the training set). The vectors were then expanded from (162,) to (162, 1) to be compatible with the Conv1D layers as shown in Fig S7.
- **Architecture:** The Keras Sequential model consists of four convolutional blocks:
  - **Input Layer:** Accepts the (162, 1) shaped tensor.
  - **Block 1:** Conv1D (256 filters, 5-kernel, relu) → MaxPooling1D
  - **Block 2:** Conv1D (256 filters, 5-kernel, relu) → MaxPooling1D
  - **Block 3:** Conv1D (128 filters, 5-kernel, relu) → MaxPooling1D → Dropout(0.2)
  - **Block 4:** Conv1D (64 filters, 5-kernel, relu) → MaxPooling1D
  - **Classifier Head:** Flatten → Dense(32, 'relu') → Dropout(0.3)
  - **Output Layer:** A Dense(8, 'softmax') layer produces probabilities for the eight emotion classes.
- **Training:** The model was compiled with the adam optimizer and categorical_crossentropy loss. Training was guided by EarlyStopping and ReduceLROnPlateau callbacks, achieving a final validation accuracy of 90.18%.

**Real-Time Implementation:** The final .h5 model and the StandardScaler object were saved. The real-time script captures live microphone audio, applies the *exact same* 162-feature extraction and scaling pipeline, and uses the loaded model to predict the emotion.

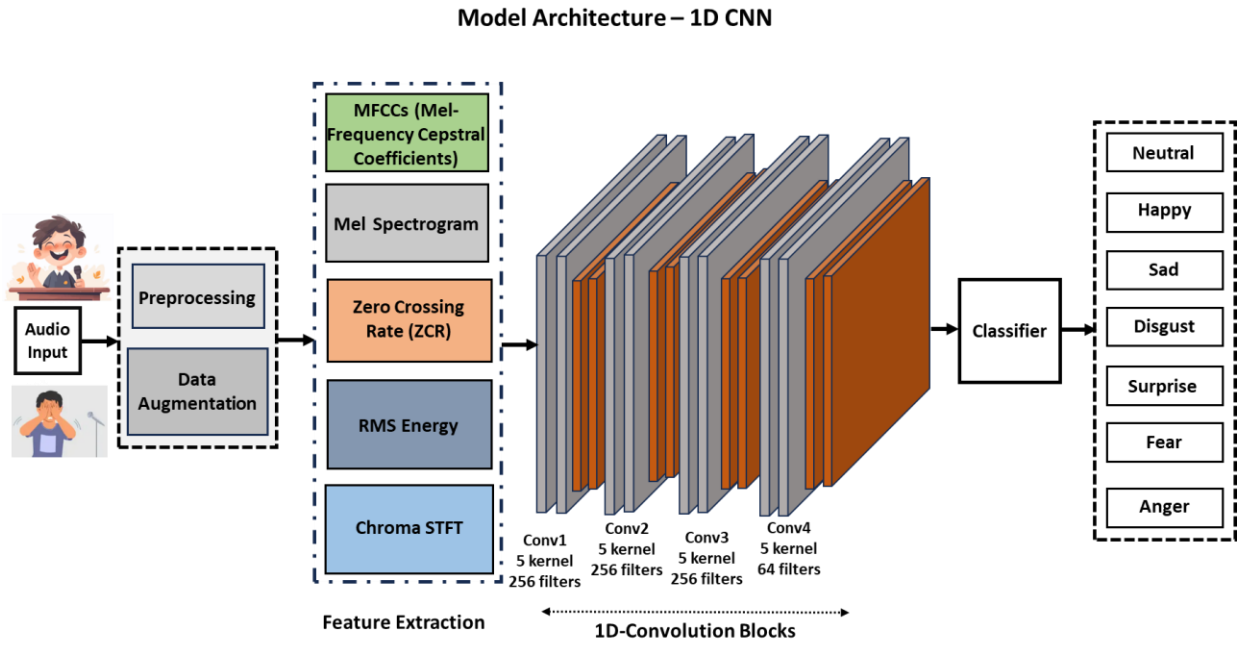


Figure S7: Architecture of the 1D-CNN for SER

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv1d (Conv1D) | (None, 162, 256) | 1,536 |
| max_pooling1d (MaxPooling1D) | (None, 81, 256) | 0 |
| conv1d_1 (Conv1D) | (None, 81, 256) | 327,936 |
| max_pooling1d_1 (MaxPooling1D) | (None, 41, 256) | 0 |
| conv1d_2 (Conv1D) | (None, 41, 128) | 163,968 |
| max_pooling1d_2 (MaxPooling1D) | (None, 21, 128) | 0 |
| dropout (Dropout) | (None, 21, 128) | 0 |
| conv1d_3 (Conv1D) | (None, 21, 64) | 41,024 |
| max_pooling1d_3 (MaxPooling1D) | (None, 11, 64) | 0 |
| flatten (Flatten) | (None, 704) | 0 |
| dense (Dense) | (None, 32) | 22,560 |
| dropout_1 (Dropout) | (None, 32) | 0 |
| dense_1 (Dense) | (None, 8) | 264 |

Figure S8: 1D-CNN Model Architecture Summary for SER

## 1.3 Text Emotion Recognition (TER)

This module was developed to classify the emotional content of written text, based on semantic content and context, into one of six discrete categories.

### *1.3.1 Dataset: GoEmotions*

The GoEmotions dataset was used for training and evaluating the text emotion recognition model. This dataset consists of English-language Reddit comments labeled with 28 fine-grained emotion categories, along with a neutral class. The dataset is widely used in affective computing research due to its real-world conversational nature and diversity of emotional expressions.

Dataset Link: GoEmotions, Yaaho_Answers_Topics

For practical deployment in a mental health monitoring system, it was necessary to reduce the label space to a smaller, interpretable set of core emotions. Therefore, the following six emotions were selected:

$$\text{Selected Emotions} = \{\text{anger, fear, joy, sadness, neutral, surprise}\}$$

The dataset was filtered to retain only samples containing at least one of these selected emotions. Since some comments had multiple emotion labels, the first matching selected emotion was chosen as the final class label for each sample.

This filtering process ensured that the model focused on emotions most relevant to mental health monitoring while maintaining sufficient training data for robust learning.

### *1.3.2 Text Processing and Tokenization*

Unlike traditional NLP pipelines that require extensive manual preprocessing (such as stop word removal, stemming, or lemmatization), BERT internally handles linguistic variations through its pre-trained embeddings. Therefore, minimal preprocessing was applied.

The text was tokenized using the BERT tokenizer (BERT-base-uncased), which converts raw text into a format suitable for Transformer models. Tokenization involved:

- Splitting text into subword tokens
- Mapping tokens to integer IDs (input_ids)
- Generating an attention_mask to differentiate real tokens from padding

The following tokenization parameters were used:

- Maximum sequence length: 128 tokens
- Padding: Applied to all sequences to maintain uniform input size
- Truncation: Enabled for texts exceeding 128 tokens

The tokenizer thus transformed each text input into two tensors:

- input_ids: Numerical representation of tokens
- attention_mask: Binary mask indicating valid tokens

### *1.3.3 BERT-Based Model Architecture*

A pre-trained BertForSequenceClassification model was fine-tuned on the filtered GoEmotions dataset. This model consists of:

- 3 Embedding Layers, Token, Position and Segment as shown in Fig. S9.
- The embedding layer in BERT converts input tokens into dense vector representations by combining token embeddings, positional embeddings, and segment embeddings, enabling the model to understand word meaning, word order, and sentence boundaries.
- 12 Transformer encoder layers, each with multi-head self-attention mechanisms
- A classification head on top of the final hidden representation

- An output layer with 6 neurons, corresponding to the selected emotion classes

The model architecture enables bidirectional context understanding, meaning each word is interpreted based on both preceding and following words in a sentence. This is particularly beneficial for emotion detection, where meaning often depends on context.

The output of the model consists of raw scores (logits) for each emotion class, which are converted into probabilities using the softmax function.

### *1.3.4 Training Procedure*

The model was fine-tuned using supervised learning with labeled emotion data.

Training details:

- Optimizer: **AdamW**
- Learning rate: $\mathbf{2 \times 10^{-5}}$
- Batch size: **16**
- Number of epochs: **3**
- Loss function: **Cross-Entropy Loss**, suitable for multi-class classification

During training, the model learned to minimize the difference between predicted emotion probabilities and true labels. After training, the model and tokenizer were saved for real-time deployment using:

model.save_pretrained("emotion_model_manual")

tokenizer.save_pretrained("emotion_model_manual")

### *1.3.5 Model Evaluation*

The trained BERT model was evaluated on a held-out validation set using accuracy as the primary metric. Predictions were obtained by selecting the class with the highest probability:

$$\hat{y} = \arg\ \max\ (\text{logits})$$

The model achieved a validation accuracy in the range of 84.77%, demonstrating strong performance in recognizing emotional states from text. Compared to traditional LSTM-based approaches, BERT exhibited superior generalization and contextual understanding, making it more suitable for real-world chatbot and mental health applications.

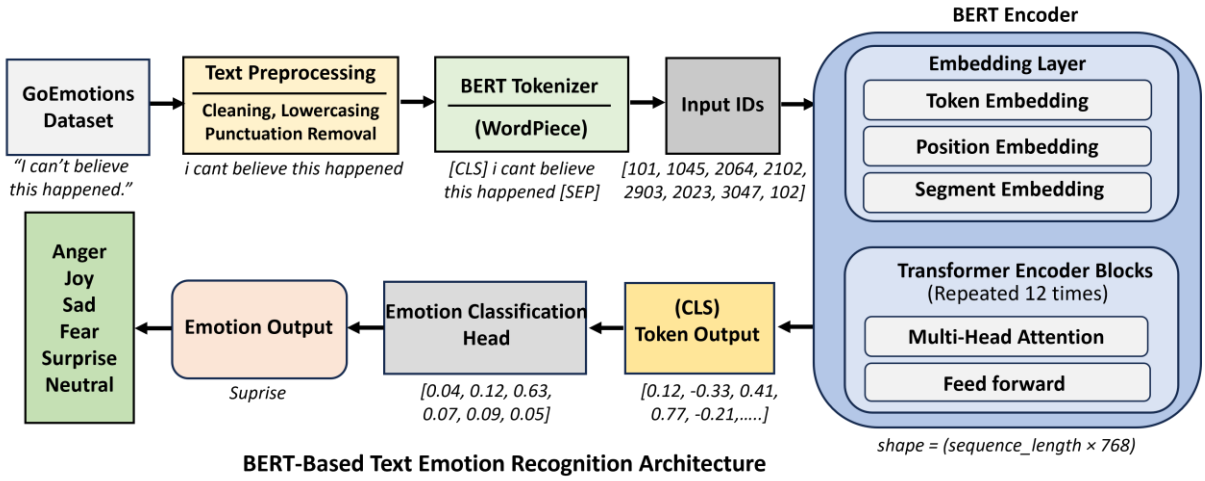


Figure S9: BERT Architecture for TER

### *1.3.6 Real Time Text Processing Pipeline*

For deployment, a real-time text inference function was implemented to perform both emotion recognition and topic classification simultaneously. This function operates as follows:

- Takes raw user text as input.
- Tokenizes the text using the saved BERT tokenizer (BERT-base-uncased).
- Feeds the processed input to the trained BERT-based Emotion Classification Model, which outputs the predicted emotion along with confidence scores.
- In parallel, the same tokenized input is passed to the trained BERT-based Topic Classification Model, which predicts the corresponding topic category from the predefined set of Yahoo Answers topics.

Thus, for each user input, the system produces two outputs:

1. The detected emotional state (e.g., anger, joy, sadness, etc.) with confidence.
2. The predicted topic category (e.g., Science, Sports, Health, etc.).

This dual prediction module is subsequently integrated into the multimodal fusion pipeline, where the text-based emotion prediction contributes to the final fused emotional state, while the topic prediction provides contextual understanding of the user's discourse, enabling better interpretability and more meaningful downstream analysis (such as chatbot responses or trend analysis in the backend).

## 1.4 Multimodal Fusion Framework

The multimodal emotion recognition system integrates facial, speech, and text-based emotional cues to produce a single, holistic emotional prediction. The system follows a Decision-Level (Late) Fusion Strategy, where each unimodal model independently analyzes its respective input modality, and their outputs are combined at the decision stage using a weighted aggregation approach.

This strategy ensures modularity, interpretability, and robustness, while allowing each specialized model to operate optimally before fusion.

### *1.4.1 Common Emotion Space and Label Alignment*

Since the three unimodal models (FER, SER, and TER) are trained on different datasets and produce emotion labels in different formats, a unified representation is necessary for fusion.

A common emotion space is defined as:

COMMON_EMOTIONS = {anger, disgust, fear, happy, neutral, sad, surprise}

To align the outputs of each modality with this common set, modality-specific mapping functions are applied:

- **FACE_MAP** aligns facial emotion labels to the common set (e.g., *sadness* → *sad*).
- **SPEECH_MAP** aligns speech emotion labels (e.g., *calm* → *neutral, angry* → *anger*).
- **TEXT_MAP** aligns text-based emotion labels (e.g., *joy* → *happy, sadness* → *sad*).

This mapping step ensures that all three modalities contribute to a consistent emotional representation before fusion.

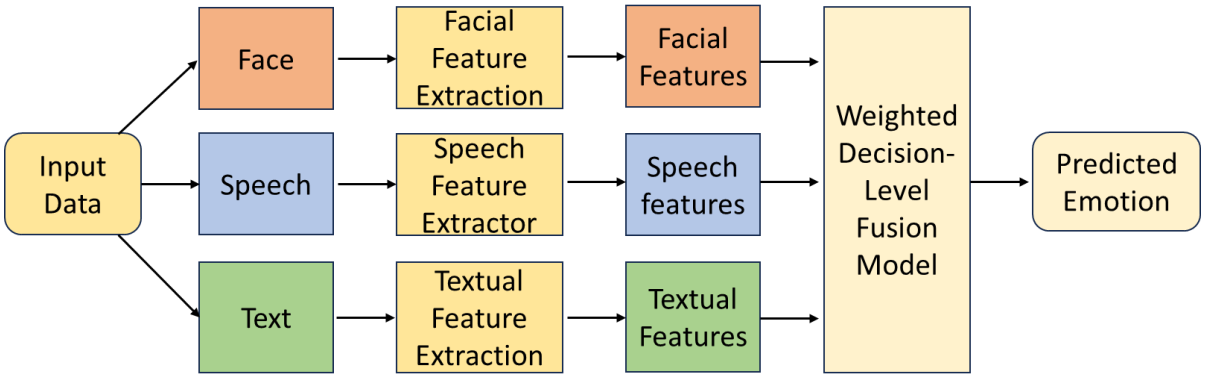


Figure S10: Multimodal Fusion Architecture

### *1.4.2 Late-Fusion Strategy*

The proposed integration model is a Decision-Level Late Fusion strategy. This approach is chosen for its modularity, robustness, and effectiveness in handling complex, real-world scenarios.

Conceptual Overview: In this framework, information is combined at the very end of the process, after each specialized model has made its own independent prediction as shown in Fig S10. This is the opposite of "early fusion," which would involve combining all raw data at the beginning.

An effective analogy is to consider our three trained models as a panel of independent experts:

- A Facial Expert (our 2D-CNN) who only looks at the user's face.
- A Vocal Expert (our 1D-CNN) who only listens to the user's tone of voice.
- A Language Expert (our BERT) who only reads the user's text.

**Illustrative Example: Resolving Sarcasm**

To demonstrate the power of this fusion strategy, consider a user speaking the sentence, "That's just great," in a sarcastic manner.

A unimodal system would likely fail:

- **TER Model (Text):** Would analyze the positive words and incorrectly predict **'Joy'** (e.g., P_ter (Joy) = 0.80).
- **FER Model (Face):** Would see an angry or disgusted expression and predict **'Anger'** (e.g., P_fer(Anger) = 0.90).
- **SER Model (Speech):** Would hear a sharp, negative tone and also predict **'Anger'** (e.g., P_ser(Anger) = 0.70).

By applying our proposed late-fusion formula with weights tuned to prioritize non-verbal cues (e.g., w_fer = 0.5, w_ser = 0.4, w_ter = 0.1), the system correctly resolves the ambiguity:

- $P_{final}(\text{Anger}) = (0.5 \cdot 0.90) + (0.4 \cdot 0.70) + (0.1 \cdot 0.0) = \mathbf{0.73}$
- $P_{final}(\text{Joy}) = (0.5 \cdot 0.0) + (0.4 \cdot 0.0) + (0.1 \cdot 0.80) = \mathbf{0.08}$

The final, fused probability for 'Anger' (73%) is overwhelmingly higher than for 'Joy' (8%), leading to the correct classification. This demonstrates the model's robustness and ability to understand complex, context-dependent emotions.

### *1.4.3 Weighted Decision-Level (Late) Fusion Strategy*

The system employs a weighted scoring mechanism to combine the confidence scores from the three modalities.

For each emotion $e \in$ COMMON_EMOTIONS, a cumulative score is computed as:

$$S(e) = 0.35 \cdot C_{face}(e) + 0.25 \cdot C_{speech}(e) + 0.40 \cdot C_{text}(e)$$

where:

- $C_{face}(e)$, $C_{speech}(e)$, and $C_{text}(e)$ represent the confidence scores output by the respective unimodal models for emotion $e$.

The final predicted emotion is determined by:

$$\text{Final Emotion} = \arg\max_{e} S(e)$$

**Weight Justification**

The fusion weights are assigned based on the relative reliability and informational content of each modality:

- **Text (0.40):** Assigned the highest weight because it provides explicit semantic information about the user's emotional state.
- **Face (0.35):** Given substantial weight as facial expressions are strong indicators of affect.
- **Speech (0.25):** Given a slightly lower weight due to susceptibility to environmental noise and variability in vocal expression.

This weighting scheme allows the system to prioritize more reliable modalities while still incorporating all available emotional cues.

### *1.4.4 Real-Time Parallel Processing Architecture and Fusion Model*

To achieve real-time performance, the system processes video and audio/text inputs in parallel using multithreading:

- **Video Processing Thread:**
  - Captures live webcam frames.
  - Runs the facial emotion recognition model.
  - Outputs detected facial emotion and confidence score.
- **Audio and Text Processing Thread:**
  - Records live audio input.
  - Performs speech emotion recognition.
  - Transcribes speech to text.
  - Runs text emotion recognition on the transcribed text.
  - Outputs both speech and text emotion predictions with confidence scores.

Both threads execute simultaneously, reducing latency and ensuring synchronized multimodal inference. Once both threads complete execution, their outputs are passed to the fusion module.

**Fusion Model Implementation:**

The fusion function integrates the modality-specific outputs as follows:

1. Initializes a score dictionary for all common emotions.
2. Maps each modality's predicted emotion to the common emotion space.
3. Adds the weighted confidence score to the corresponding emotion.
4. Selects the emotion with the highest cumulative score.
5. Returns a structured multimodal output containing:
    - Final fused emotion
    - Overall confidence score
    - Detailed breakdown of each modality's prediction

This structured output enhances interpretability and allows analysis of individual modality contributions.

## 1.5 WellNest Multimodal Mental Health System

### *1.5.1 System Overview*

**WellNest** is a production-grade multimodal mental health platform designed to monitor and support user well-being through advanced Artificial Intelligence. The system integrates real-time emotion recognition from facial expressions, vocal tonality, and textual sentiment to provide a holistic view of a user's emotional state as shown in Fig. S11. Components include a web-based management dashboard, an emotion-aware AI chatbot, and a cross-device Progressive Web App (PWA) for mobile accessibility.

### *1.5.2 Technical Architecture*

The application follows a Client-Server Architecture designed for high responsiveness and privacy:

- **Backend Framework**: Powered by Python Flask, providing a robust RESTful API for data processing, authentication, and AI model orchestration.
- **Frontend Technologies**: Built using Vanilla JavaScript, HTML5, and CSS3. The design utilizes Glassmorphism and Responsive Web Design (RWD) principles to ensure a premium user experience across desktops and mobile devices.
- **Database Level**: Utilizes SQLite for secure, structured storage of user profiles, emotional trends, and session logs.

- **Infrastructure & Deployment**: Employs Cloudflare Tunneling technology to expose the local server safely to the internet, enabling the application to function as a mobile PWA without complex server hosting. Refer Table S1 for Technical Stack of WellNest.

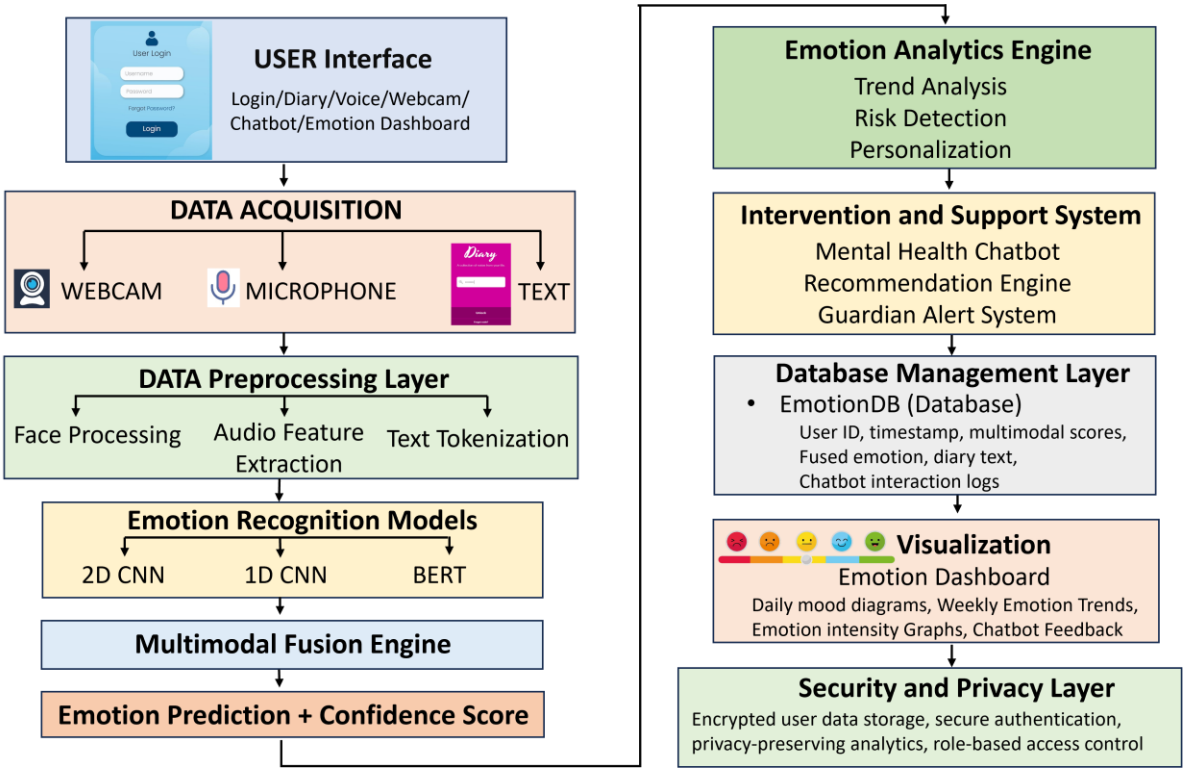


Figure S11: System Architecture Diagram

### *1.5.3 Core Features and Functionality*

**1. Multimodal Emotion Recognition**

This is the system's "Digital Pulse." It captures data through a 5-second sampling window:

- **Visual Analysis**: Analyzes facial landmarks and micro-expressions via the webcam using Deep Learning models (DeiT-ViT).
- **Audio Analysis**: Processes vocal features (pitch, tone, tempo) to detect emotional nuances in speech.
- **Text Analysis**: Performs sentiment analysis on transcribed speech or typed input.
- **Late Fusion Engine**: A specialized algorithm "fuses" these three streams, weighting them to produce a final, high-confidence "Fused Emotion" result.

**2. Emotion-Aware AI Chatbot**

Unlike standard bots, the WellNest chatbot is contextually aware of the user's current mood.

- **Empathetic Responding**: If the system detects 'Sadness' or 'Anxiety', the bot automatically adjusts its tone to be more supportive and offers coping mechanisms.
- **Proactive Popups**: The system triggers "fallback popups" or supportive messages immediately after an emotion detection session to provide instant verbal care.

**3. Interactive Mental Health Diary**

The digital diary allows users to record their thoughts while the system performs background analysis:

- **Topic Modeling**: Automatically categorizes diary entries into topics (e.g., Work, Relationships, Health).
- **Sentiment Sync**: Each diary entry is linked to an emotional state, which is reflected in the user's longitudinal mood calendar.

**4. Mental Health Analytics Dashboard**

A data-driven interface that visualizes a user's progress over time:

- **Weekly Trends**: Line charts showing the dominant mood over the last 7 days.
- **Daily Distribution**: Pie charts or histograms showing the breakdown of emotions throughout a single day.
- **Consecutive Mood Monitoring**: Tracks persistence of negative emotions to identify potential risks.

**5. Proactive Safety Alert System**

The system acts as a safety net:

- **Warning Level**: Triggered if the user records negative emotions (e.g., 'Sad') for 5 consecutive days.
- **Critical Level**: Triggered after 7 consecutive days of distress, suggesting professional intervention or notifying a guardian.

| **Component** | **Technology** |
|---|---|
| Backend | Flask (Python 3.x) |
| Frontend | Vanilla JavaScript, CSS (Flexbox/Grid), HTML5 |
| Database | SQLite3 |
| AI Models | DEI-ViT (Visual), Gemini/Transformers (Text/Speech) |
| Networking | Cloudflare Tunnel (for PWA) |
| Visualizations | Chart.js / Custom CSS Graphics |

Table S1: Technology Stack

### *1.5.4 Data Handling and Privacy*

Data integrity and user privacy are central to WellNest's design.

**1. Data Storage Schema**

The system manages data across four primary tables:

1. **Users Table**: Stores encrypted credentials, profile details, and Guardian Contact Information (for emergency alerts).
2. **Emotion Results Table**: Stores metadata for every detection session, including individual scores for video, speech, and text, as well as the final fused emotion and confidence levels.
3. **Chat History**: Records user-bot interactions to maintain conversational context and improve empathy.
4. **Diary Entries**: Stores long-form text content, associated dates, detected topics, and sentiment scores.

**2. Data Processing Workflow**

1. **Ingestion**: Raw media (video/audio) is buffered temporarily in volatile memory.
2. **Extraction**: The AI models extract only the feature vectors and emotion labels.
3. **Persistence**: Only the resulting metadata (e.g., "Emotion: Happy", "Confidence: 0.92") is saved to the SQLite database. Raw video and audio files are not stored long-term, ensuring user privacy.
4. **Retrieval**: The Analytics engine queries this metadata using standard SQL to generate real-time charts and reports.

**3. Security Measures**

- **Session Management**: Uses HTTP-only cookies and user-specific IDs to prevent unauthorized data access.
- **Password Hashing**: User passwords are never stored in plain text; they are hashed using secure cryptographic algorithms.
- **Tunnel Encryption**: The Cloudflare Tunnel provides end-to-end encryption for data traveling between the mobile device and the local server.

# Additional Results

This chapter presents the experimental setup, evaluation of metrics, and performance results of the three independent unimodal classification models. Each model's performance is analyzed, followed by a comparative discussion and a summary of the project's limitations and conclusions.

## 2.1 Experimental Setup

All models were developed in the Python 3.x programming environment. The deep learning models were built, trained, and evaluated using the TensorFlow and Keras high-level APIs.

The following key libraries were essential to the project:

- **Facial Emotion Recognition (FER):** OpenCV (for real-time face detection and image processing), scikit-learn (for data splitting), and pandas.
- **Speech Emotion Recognition (SER):** librosa (for audio preprocessing and feature extraction), scikit-learn (for StandardScaler and metrics), and pandas.
- **Text Emotion Recognition (TER):** Hugging Face *Transformers* (for BERT-based sequence classification), *Datasets* library (for loading and processing GoEmotions and Yahoo Answers Topics), *PyTorch* (for model training and inference), *BertTokenizer* (for tokenization, padding, and truncation), and *scikit-learn* (for evaluation metrics such as accuracy).All models were trained and executed in a standard laptop environment, utilizing the CPU for computation.

## 2.2 Model Performance: Facial Emotion Recognition (FER)

The 2D-CNN model for facial emotion recognition was the highest-performing model developed in this project.

- **Result:** The model achieved a validation accuracy of 98.48%.
- **Analysis:** This outstanding result was achieved on the CK+ dataset. The training and validation loss/accuracy plots as shown in Figure (S12 – S13) showed smooth convergence. The high accuracy is attributed to:
    - The deep 6-layer 2D-CNN architecture, which was able to learn a rich hierarchy of facial features.
    - The heavy use of BatchNormalization after each convolutional layer, which stabilized and accelerated training.
    - The use of Dropout and Data Augmentation, which successfully prevented the deep model from overfitting on the relatively small CK+ dataset.
- **Real-Time Performance:** The model was successfully deployed in a real-time OpenCV script. It proved to be computationally efficient, running smoothly on a live webcam feed and accurately classifying the 7 posed emotions. The real time testing outputs are shown in Figure S14.

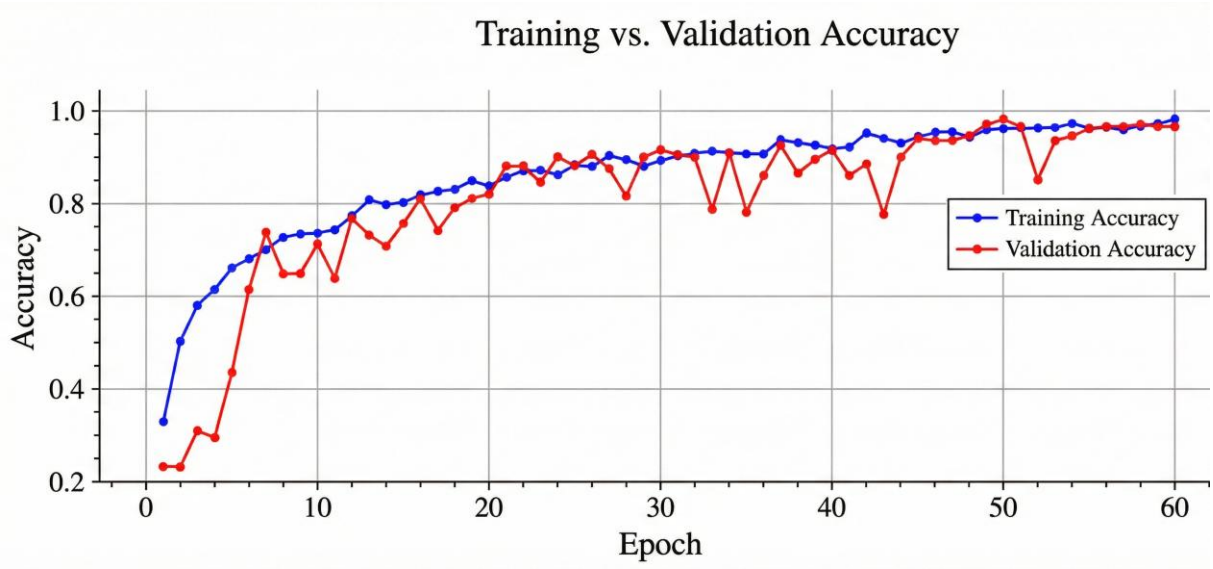


Figure S12: FER Model Training and Validation Accuracy

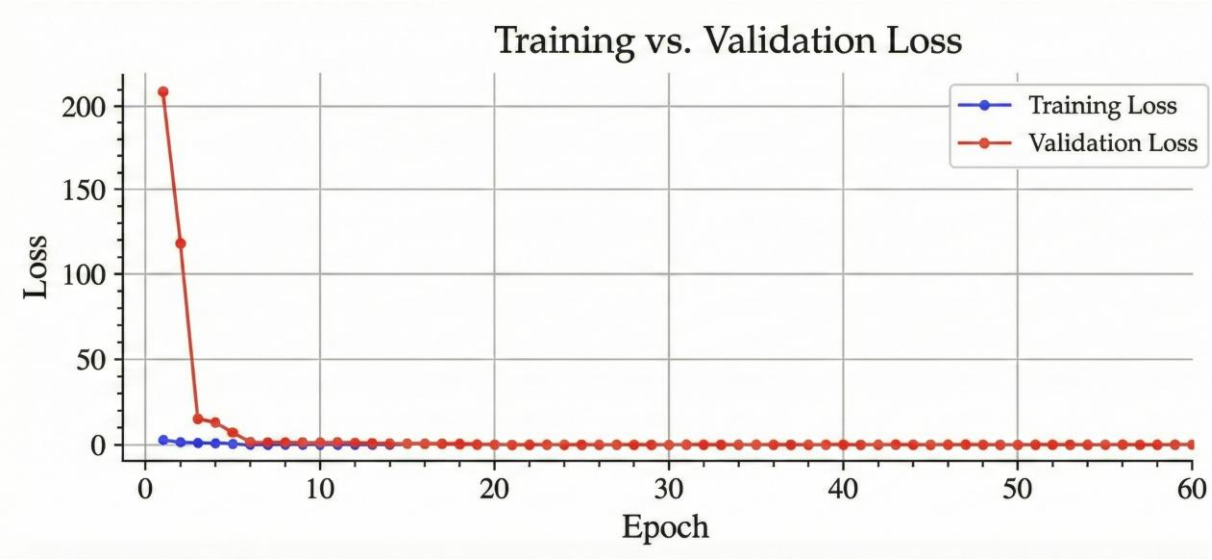


Figure S13: FER Model Training and Validation Loss

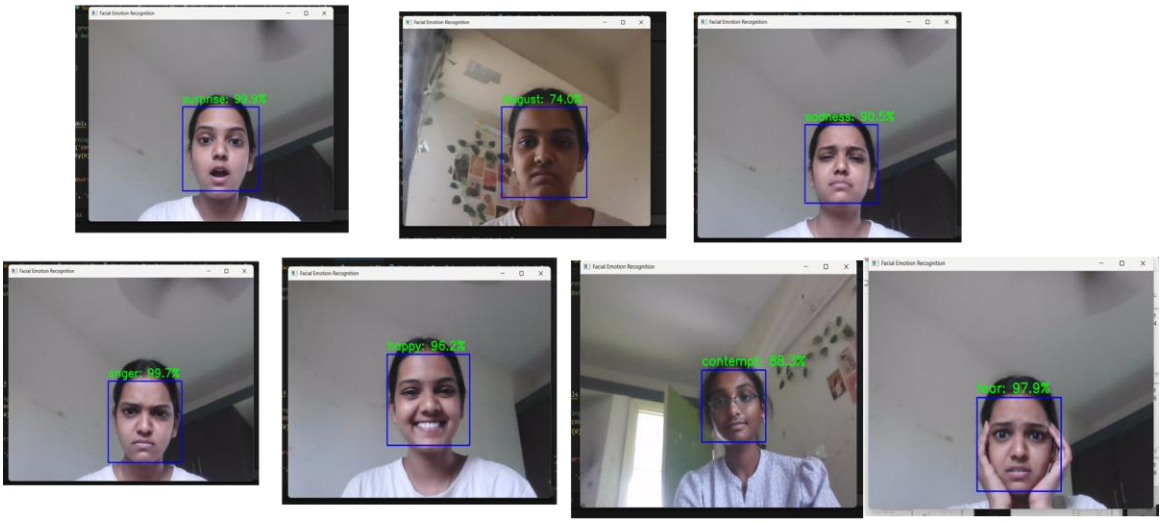


Figure S14: Demonstration of the Real-Time Facial Emotion Recognition (FER) System

## 2.3 Model Performance: Speech Emotion Recognition (SER)

The 1D-CNN model for speech emotion recognition also achieved a high-performance benchmark.

- **Result:** The model achieved a final validation accuracy of 90.18%. The detailed classification report on the test set is shown below.
- **Analysis:** The classification report shows a very strong macro-average F1-Score of 86%, indicating the model is robust across all classes as shown in Figure S15.
    - **Strengths:** The model is exceptionally good at identifying 'angry' (92% F1) and 'disgust' (86% F1).
    - **Weaknesses:** As shown in the confusion matrix plot (generated by the code), the primary source of error was the confusion between acoustically similar classes, particularly 'calm' (75% F1), which was often mistaken for 'neutral'. This is an expected challenge in SER.
    - The use of a diverse, composite dataset (Ravdess, TESS, SAVEE) and our audio augmentation pipeline (noise, stretch) were critical to achieving this high, generalizable result.

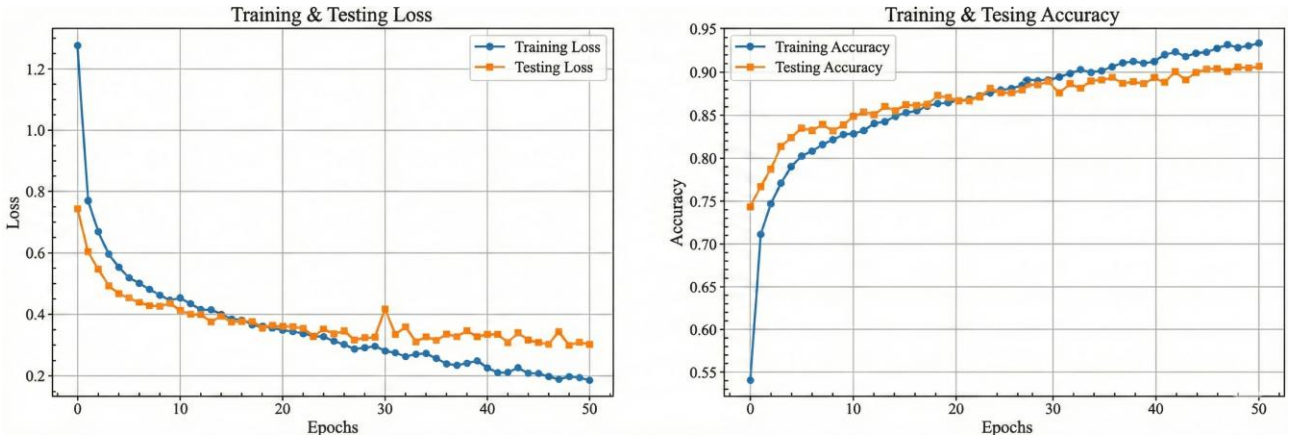


Figure S15: SER Model Training and Validation Loss/Accuracy

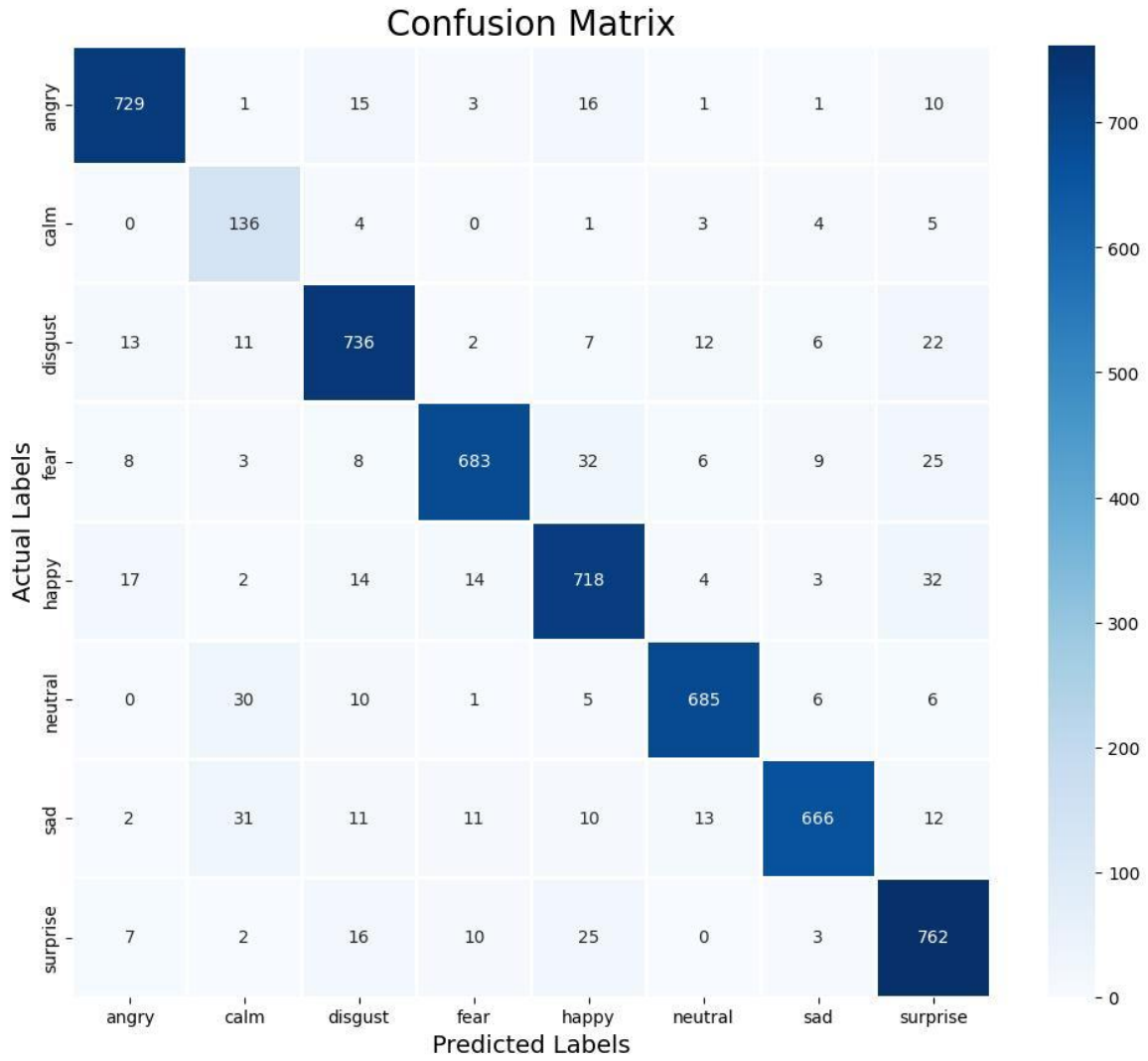


Figure S16: Confusion Matrix of SER Model

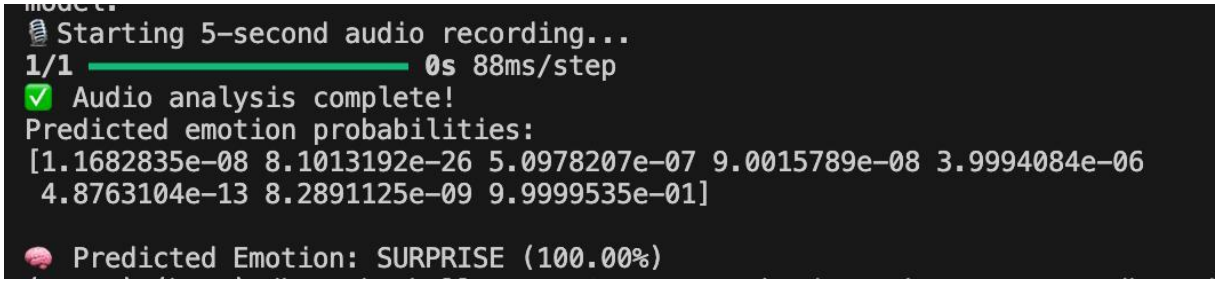


Figure S17: Real-Time SER System in Operation

## 2.4 Model Performance: Text Emotion Recognition (TER)

The BERT-based Text Emotion Recognition (TER) model was evaluated on the filtered GoEmotions validation set using standard classification metrics. Unlike traditional recurrent models, BERT leverages bidirectional contextual embeddings, allowing it to capture nuanced semantic and emotional cues from text.

- **Result:** The fine-tuned BERT model achieved a validation accuracy of approximately 84.77%, demonstrating strong performance in recognizing emotional states from textual data.
- **Analysis:** The BERT-based text emotion model achieved strong performance on the filtered GoEmotions dataset, with smooth training and validation convergence.The high accuracy is attributed to:
  - The bidirectional Transformer architecture of BERT, which effectively captures contextual meaning in text.
  - Use of a pre-trained *bert-base-uncased* model, enabling transfer of knowledge from large-scale corpora.
  - An effective fine-tuning setup using AdamW optimizer (lr = 2e-5) and batch size 16, ensuring stable training.
  - Subword tokenization, which improved robustness to informal language, slang, and spelling variations.

- **Real-Time Applicability:** The trained BERT model was successfully integrated into the real-time pipeline. During inference:

  - User text is tokenized using the saved BERT tokenizer
  - The model outputs predicted emotion and confidence score
  - This output is passed to the multimodal fusion module

We have tested different models for TER, few of the models gave high accuracy but the dataset used for them are not as robust as the dataset used for the proposed model. Even thought the accuracy is high yet they cannot understand the emotion in a sentence like it understands in BERT as shown in Table S2.

| Model Name | Accuracy Score | F1 Score | Recall Score | Precision Score |
|---|---|---|---|---|
| Bidirectional LSTM | 0.8800 | 0.8270 | 0.8127 | 0.8512 |
| LSTM | 0.6245 | 0.4133 | 0.4431 | 0.3915 |
| Stack LSTM | 0.3475 | 0.0859 | 0.1666 | 0.0579 |
| Bidirectional GRU | 0.8795 | 0.8326 | 0.8438 | 0.8265 |
| GRU | 0.3475 | 0.0859 | 0.1666 | 0.0579 |
| Stack GRU | 0.8890 | 0.8405 | 0.8473 | 0.8374 |
| **Proposed BERT** | **0.8477** | **0.85** | **0.85** | **0.85** |

Table S2: Comparative Analysis of Sequential Models for TER

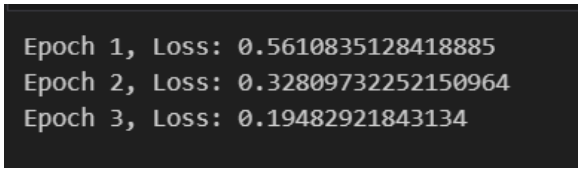


Figure S18: Training for TER

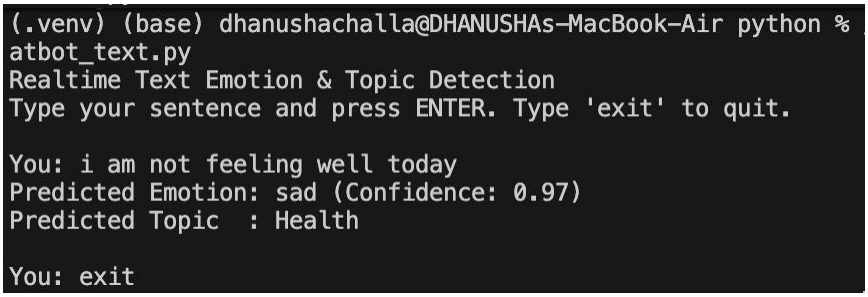


Figure S19: Demonstration of the Real-Time TER System in Operation

## 2.5 Model Performance: Multimodal Emotion Recognition and Backend

**Result:**

The multimodal system successfully performed real-time decision-level (late) fusion of facial, speech, and text emotions using a weighted confidence-based approach and reliably stored outputs in EmotionDB.

**Analysis:**

- **Strengths:**
    - The weighted fusion strategy (Face: 0.35, Speech: 0.25, Text: 0.40) effectively integrated semantic and non-verbal cues to produce a coherent final emotion.
    - Parallel multithreading enabled low-latency, real-time inference across all three modalities.
    - The system remained functional even when one modality was unreliable (e.g., poor lighting or noisy audio).

- **Backend Performance (EmotionDB):**
    - Successfully logged each prediction with fused emotion, confidence score, modality-wise outputs, and timestamp.
    - Enabled real-time visualization of recent emotional history in the backend.

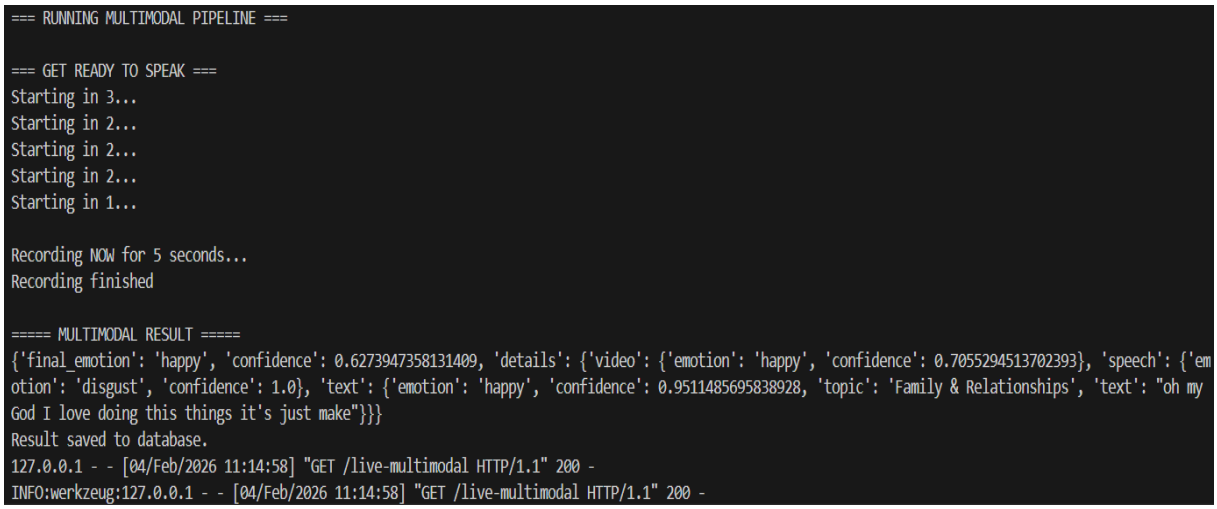


Figure S20: Multimodal testing in real time

Pretty-print


```
{
  "confidence": 0.5476085305213928,
  "details": {
    "speech": {
      "confidence": 0.7118760347366333,
      "emotion": "sad"
    },
    "text": {
      "confidence": 0.9895957112312317,
      "emotion": "happy",
      "text": "I love doing that it's so",
      "topic": "Family & Relationships"
    },
    "video": {
      "confidence": 0.43362927436828613,
      "emotion": "happy"
    }
  },
  "final_emotion": "happy"
}
```


Figure S21: Backend Live Multimodal Testing Output

Pretty-print


```
{
  "history": [
    {
      "confidence": 0.5476085305213928,
      "fused_emotion": "happy",
      "timestamp": "2026-02-03T21:21:56.831249"
    },
    {
      "confidence": 0.3756227731704712,
      "fused_emotion": "happy",
      "timestamp": "2026-02-03T21:21:31.908768"
    },
    {
      "confidence": 0.40211115479469295,
      "fused_emotion": "disgust",
      "timestamp": "2026-02-03T21:20:59.210254"
    },
    {
      "confidence": 0.7455726593732834,
      "fused_emotion": "happy",
      "timestamp": "2026-02-03T15:42:47.977495"
    },
```


Figure S22: Emotion History of Individual from Emotion_db

## 2.6 WellNest: A Mental Wellness Monitoring and Support Platform

The developed system, WellNest, was successfully implemented as Progressive Web Application (PWA)a fully functional web and mobile-compatible . The application integrates multimodal emotion recognition with interactive user interfaces, real-time analytics, and AI-driven support features. The results demonstrate the system's capability to provide continuous emotional monitoring and personalized mental health assistance.

### *2.6.1 User Interaction and Support Features*

The WellNest system integrates multiple user-centric features, including an emotion-aware chatbot, digital diary, and real-time mood detection interface as shown in Figure S23, to provide continuous mental health support. The chatbot dynamically adapts its responses based on the detected emotional state, offering empathetic and context-aware conversations that enhance user engagement as shown in Figure S24. The digital diary module allows users to record daily thoughts, which are automatically analyzed for sentiment and linked to corresponding emotional states, enabling structured emotional journaling. Additionally, the system provides an intuitive dashboard with a "Start Mood Scan" feature that captures multimodal inputs (facial, speech, and text) and displays the predicted emotion with a confidence score as shown in Figure S24. Together, these features create a cohesive and interactive environment for monitoring and improving emotional well-being.

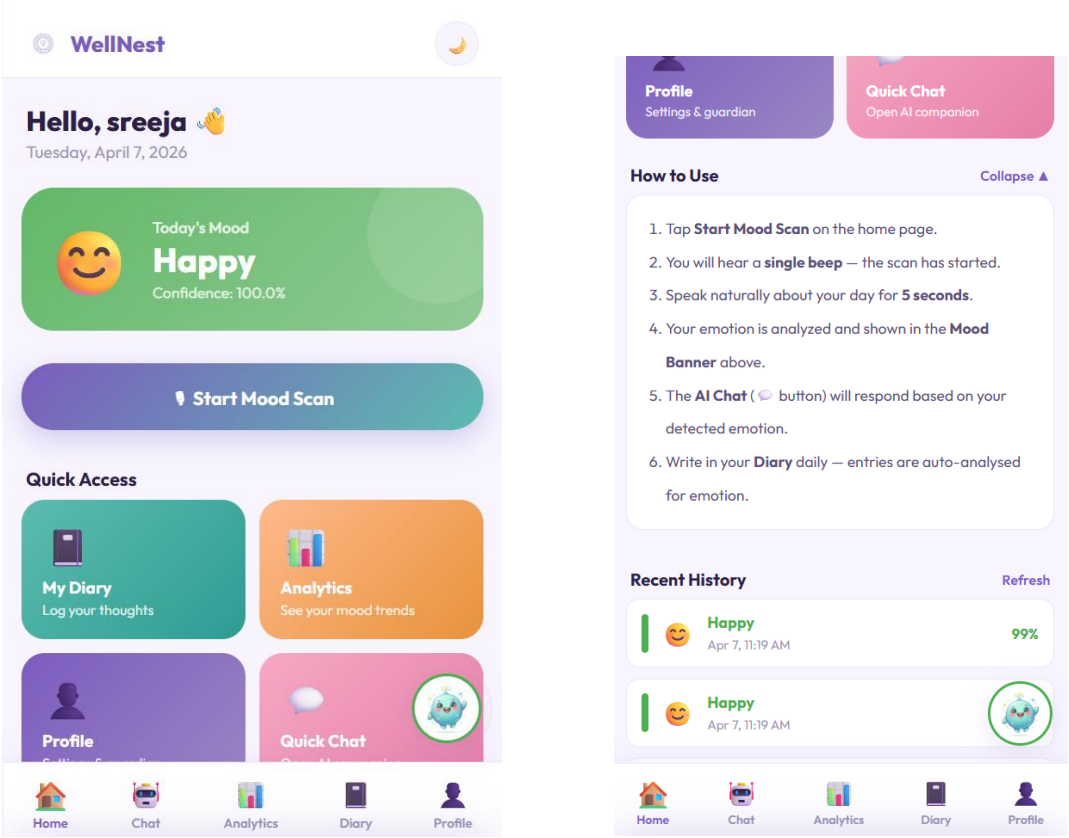


Figure S23: User Interface of WellNest

### *2.6.2 Data Handling and Safety Mechanisms*

WellNest ensures secure and efficient data management while prioritizing user privacy. The system stores only processed emotion metadata (such as predicted emotion labels and confidence scores) instead of raw audio or video inputs, thereby minimizing privacy risks. All user data, including diary entries and chat interactions, is securely managed using structured database storage with proper authentication mechanisms. Furthermore, the system incorporates a guardian alert mechanism designed for proactive intervention as shown in Figure S24. In cases where prolonged negative emotional patterns are detected, the system can trigger alerts or recommendations, enabling timely support. This combination of privacy-preserving data handling and safety-oriented design enhances the reliability and real-world applicability of the platform.

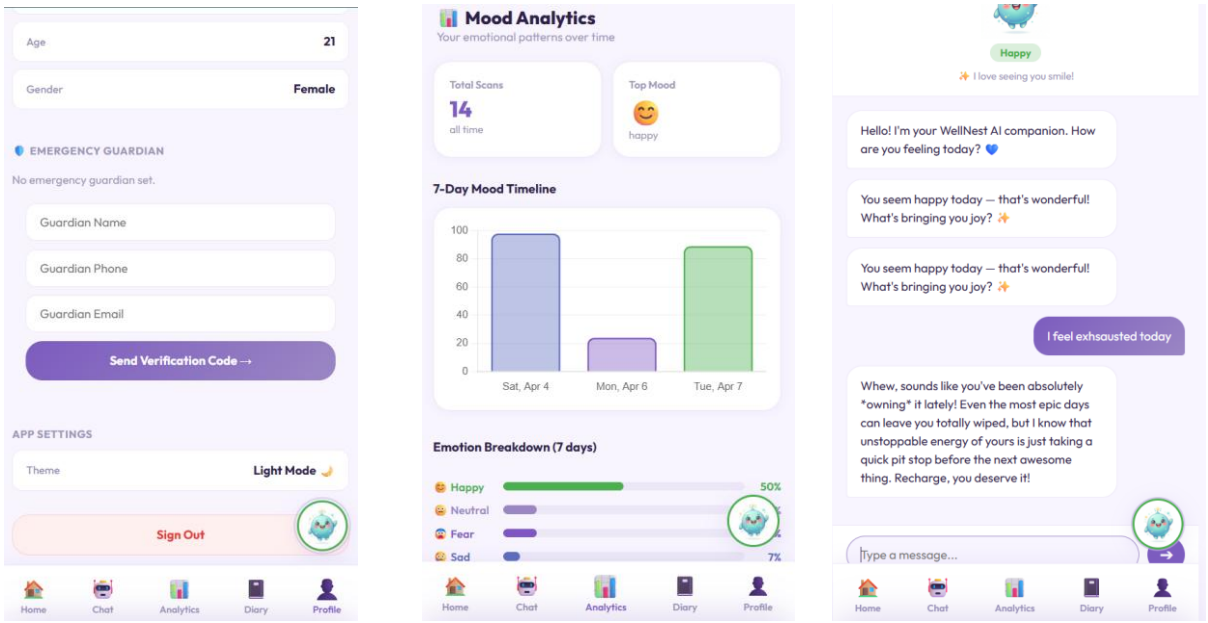


Figure S24: WellNest Features

Furthermore, the study confirmed the effectiveness of the multimodal fusion approach, enabling more reliable emotion prediction by combining visual, audio, and textual inputs. These validated components form the foundation of the WellNest platform, supporting real-time emotion monitoring, intelligent interaction, and scalable mental wellness analysis. The Web application of WellNest is shown in figure [S25- S29]

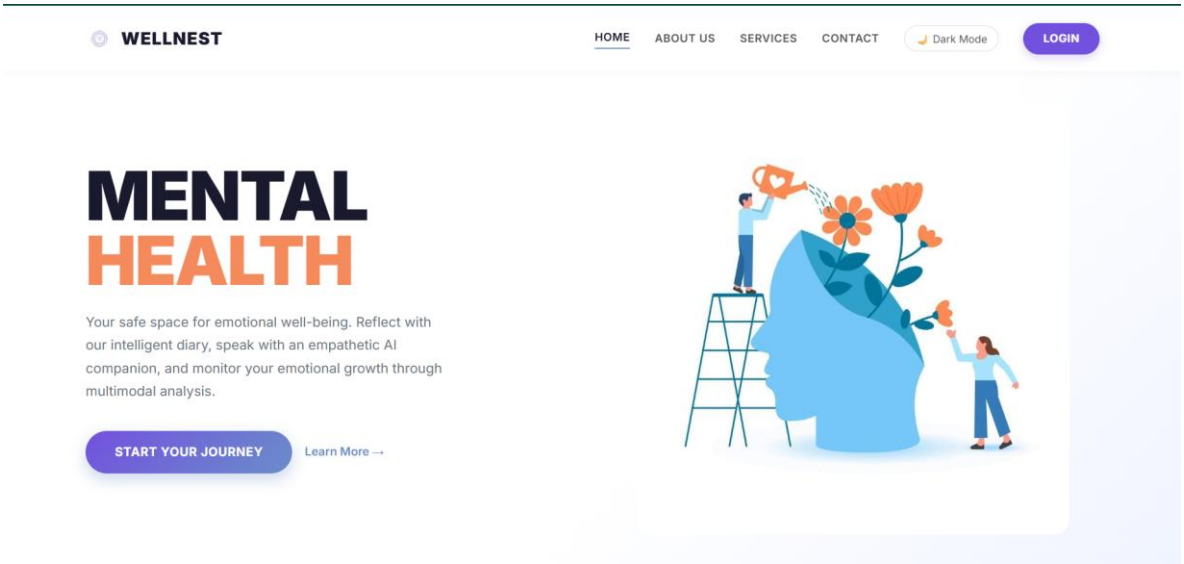


Figure S25: Web Application Interface User Page

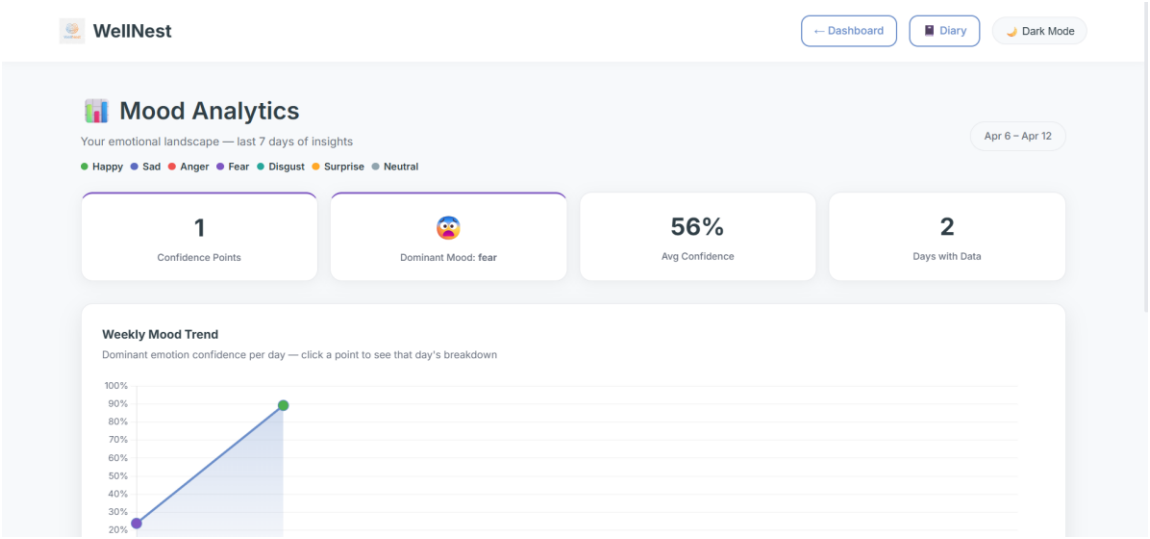


Figure S26(a): Data Analytics Page

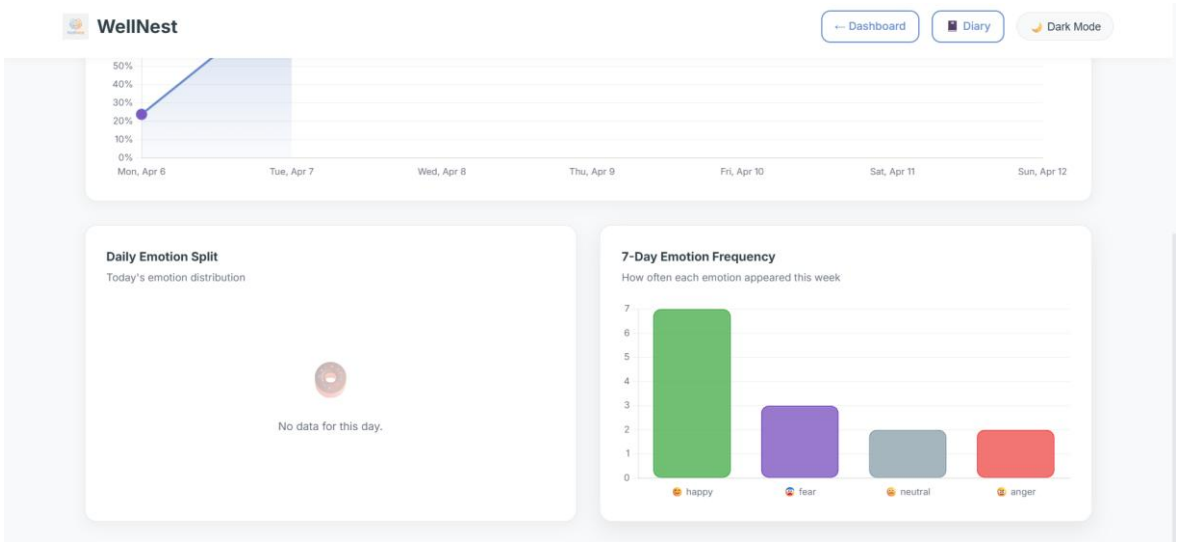


Figure S26(b): Data Analytics Page

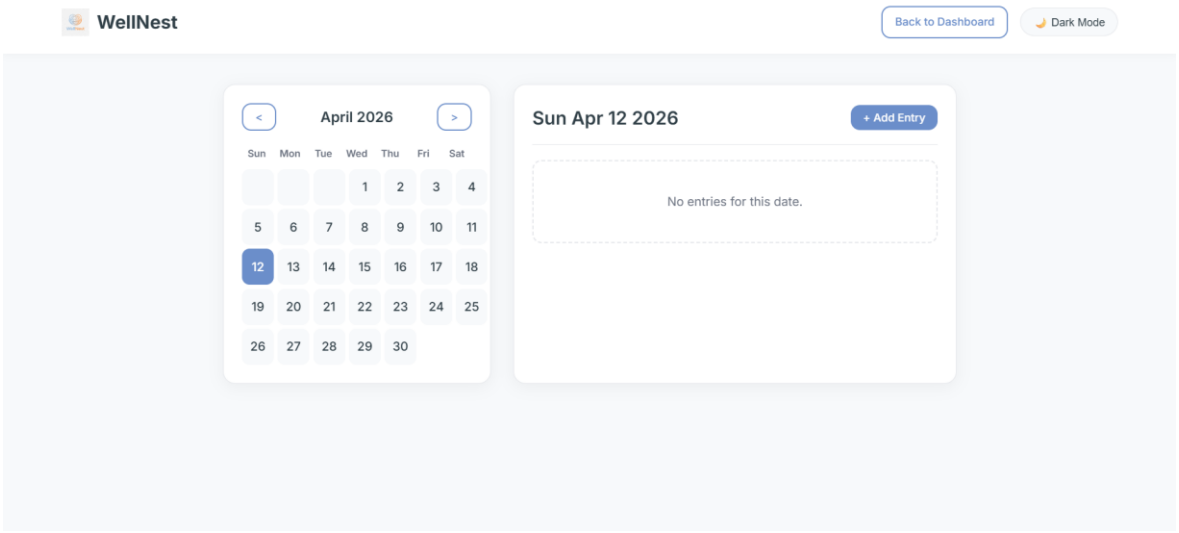


Figure S27: Dairy Entry Feature in Web Application

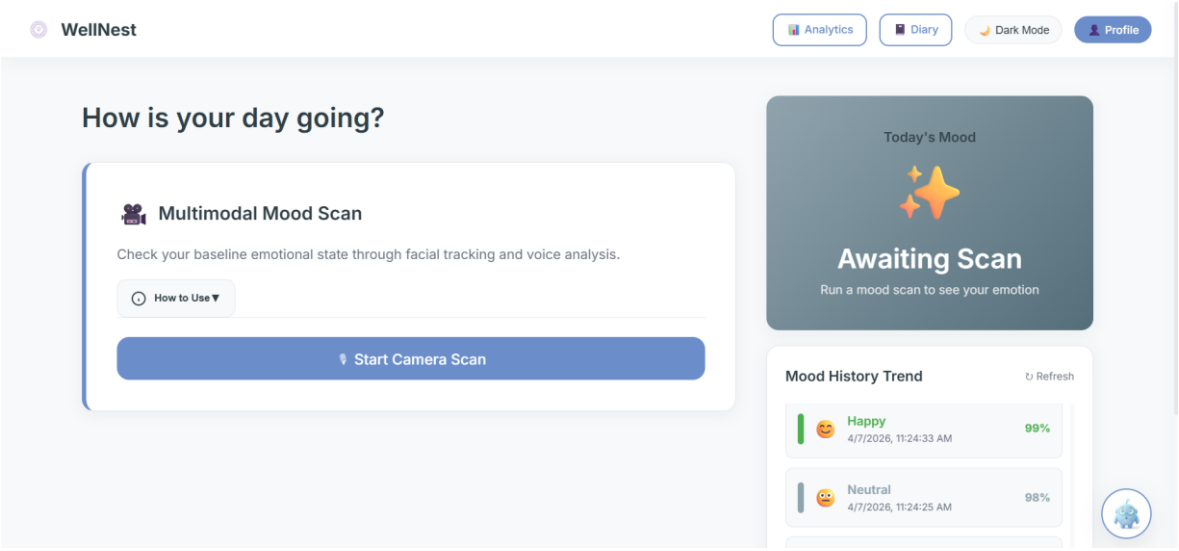


Figure S28: Multimodal Mood Scan Page

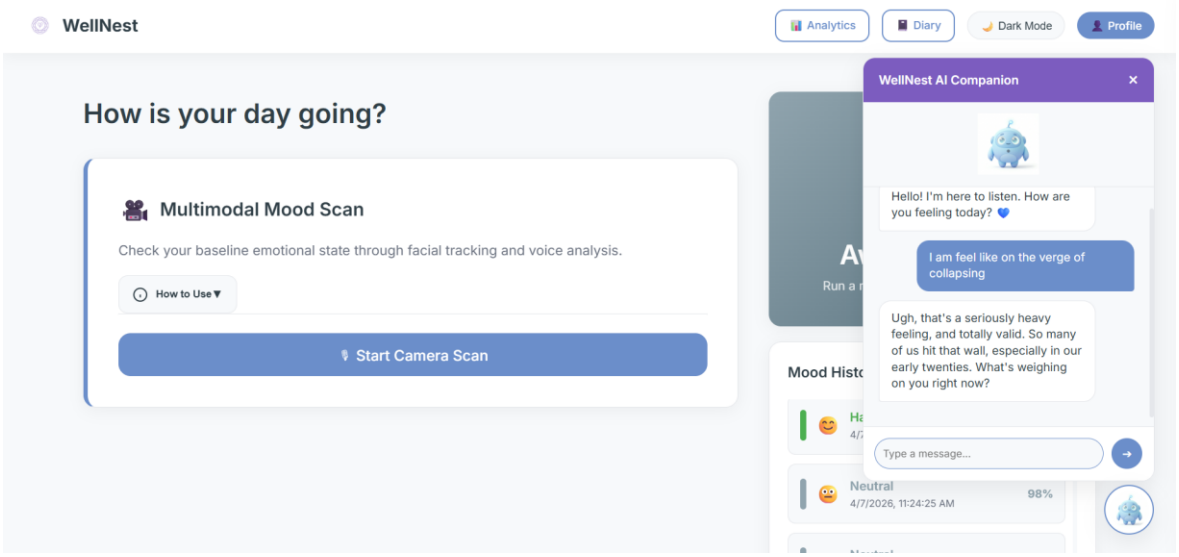


Figure S29: Chatbot Feature

## Summary of Work

This project successfully addressed a key challenge in affective computing by developing high-performance unimodal models and integrating them into a fully functional real-time multimodal system within the WellNest platform. The primary objective of building a robust multimodal emotion recognition framework was achieved through the design, validation, and deployment of three complementary modalities—facial, speech, and text—combined into a unified decision-making system.

1. Facial Emotion Recognition (FER): A deep 2D-CNN model was developed and trained on the CK+ dataset, achieving a validation accuracy of 98.48%.

2. Speech Emotion Recognition (SER): A 1D-CNN model trained on a composite dataset (RAVDESS, TESS, SAVEE) achieved 90.18% validation accuracy.

3. Text Emotion Recognition (TER): A fine-tuned BERT-based model achieved 84.77% accuracy for emotion classification, along with a topic classification accuracy of approximately 74%.

Beyond unimodal performance, the project successfully implemented a Decision-Level (Late) Multimodal Fusion framework, where predictions from all three modalities are integrated in real time using a weighted confidence-based aggregation strategy. The system employs parallel processing to ensure efficient and synchronized execution, producing a single, reliable fused emotional output.

Furthermore, the complete system was deployed as part of the WellNest mental wellness platform, integrating backend storage (EmotionDB) to manage multimodal results, including fused emotions, confidence scores, modality-wise predictions, and timestamps. This enables real-time emotion tracking, analytics, and personalized user interaction. Unlike traditional offline approaches, the entire pipeline—including AI models, fusion logic, and backend infrastructure—was successfully implemented and tested in real time, demonstrating its scalability, efficiency, and applicability for practical mental health monitoring solutions.