Multi-Mode Bilingual Emotion Detection System

Ajaykumar K
Department of Computer Technology
Madras Institute of Technology
Anna University
2021503003@student.annauniv.edu

Santhosh D
Department of Computer Technology
Madras Institute of Technology
Anna University
2021503047@student.annauniv.edu

Velmurugan S
Department of Computer Technology
Madras Institute of Technology
Anna University
2021503317@student.annauniv.edu

***Abstract*—This paper presents a real-time multi-mode bilingual emotion detection framework that addresses critical limitations in traditional emotion recognition systems, including monolingual bias (85% English-only datasets), unimodal rigidity (72% single-mode systems), and high cloud latency (400–600ms per request). Our approach integrates: (1) a CNN-based facial emotion classifier trained on FER-2013, achieving 82.6% accuracy for visual inference; (2) a BERT+BiLSTM hybrid model for English text with an 84.3% F1-score; and (3) a fine-tuned Tamil BERT achieving 57.9% accuracy on TamilEmo for regional language support. The system logs timestamped predictions from both modalities and visualizes emotion trends via a dynamic dashboard. It operates entirely on-device, reducing latency to under 300ms and enabling 24×7 emotion tracking without cloud dependency. Experimental results show 87% system uptime under peak loads and consistent emotion logging accuracy across text and video inputs. The modular architecture supports seamless integration into mobile or web platforms, offering a scalable, inclusive solution for real-time affective computing.**

***Index Terms*—Emotion recognition, bilingual NLP, facial analysis, real-time inference, multi-mode AI, affective computing, Tamil BERT.**

I. Introduction

EMOTION-AWARE computing has gained increasing relevance across domains like healthcare, education, and human-computer interaction. With the proliferation of sensor-rich devices and real-time interfaces, emotion recognition systems now aim to understand users' affective states from multimodal inputs such as text, facial expressions, and speech. However, traditional approaches exhibit notable limitations in terms of modality dependence, language scope, and real-time responsiveness.

Traditional emotion detection systems often emphasize unimodal input processing, thereby limiting their generalizability in complex, real-world interactions. Although significant progress has been made in multi-label classification using feature extraction and emotion correlation learning, models like MEDAFS still assume label independence, leading to suboptimal inter-label dynamics and reduced performance in nuanced emotional contexts [1]. EEG and physiological signal-based models offer deep insights into affective states, yet scalability and real-world applicability remain limited due to small sample sizes and signal fatigue, particularly in patient-centered studies [2]. In IoT environments, virtual systems like MultiDoVEmoBar attempt multimodal mapping but suffer cascading errors when initial detections are inaccurate, affecting downstream modules [3]. Advanced frameworks employing complex architectures like two-system CNN/RNN combinations can detect subtle facial cues but often miss spatial depth when using basic RGB input streams [4]. Multitask learning models integrating emotion recognition for auxiliary tasks like fake news detection show promise, but their reliance on noisy annotation models like Unison can mislead classification and limit robustness [5].

Efforts to improve textual emotion recognition with prompt consistency and synonym-based ensembles enhance disambiguation in binary tasks but struggle with continuous or fine-grained labels [6]. In meme-based emotion detection, Vision Transformers excel at capturing image-based affect but falter in interpreting occluded or abstract memetic content [7]. Pre-trained language models, though effective, often display biases in label-word mappings, making them unreliable for culturally sensitive or granular emotion labels [8]. Single-modality systems like C-DepressNet, though powerful in edge-based depression detection, neglect the richer insights available from multimodal fusion, such as incorporating text or physiological signals alongside facial analysis [9]. Visual-audio fusion models using Bayesian learning offer dimensional alignment of emotion categories but underperform compared to neural ensemble methods due to rigid rule-based interval mapping [10].

Graph-based cyber-physical models for real-time emotion tracking in transportation settings provide temporal and syntactic embeddings but face over-smoothing at deeper layers, impairing emotion differentiation under dynamic input conditions [11]. Even state-of-the-art datasets like MGEED, which combine EEG, depth maps, and facial video, still rely on lab-controlled settings, making it difficult to generalize findings to genuine, spontaneous emotions in everyday scenarios [12]. Stimuli-aware deep networks using subnetworks like GlobalNet and ExpressionNet have made strides in contextual facial recognition but are challenged by the inherent ambiguity and complexity of compound emotions [13]. While temporal convolutional transformers with embedded medical knowledge represent a leap forward in depression and emotion co-detection, their ability to fully capture the multifaceted nature of affect remains incomplete [14]. Earlier methods using constrained local models for neutral-face classification underscore the trade-off between personalization and generalizability, as rigid supervised models fail with heterogeneous facial structures [15].

To overcome the challenges identified in traditional emotion detection systems, we propose a multi-layered emotion detection framework that combines real-time visual and textual analysis, persistent emotion logging, and dynamic visualization capabilities. This framework is tailored to operate in a bilingual (English-Tamil) and multimodal context, addressing limitations in language support, data interpretation, and user interaction. By integrating both facial emotion recognition and textual sentiment analysis, the system ensures comprehensive emotion coverage across varied user inputs. Continuous emotion logging facilitates temporal tracking, enabling advanced behavior pattern recognition and user-specific emotion profiling. Furthermore, the visualization module offers live trend analysis, aiding in the early identification of emotional irregularities. The architecture emphasizes modularity, enabling easy extension and adaptability to different use cases such as mental health monitoring, user engagement tracking, or educational analytics. Through optimized data handling and parallel processing, the framework maintains real-time responsiveness without compromising accuracy. Overall, this solution enhances emotional insight generation while supporting cross-lingual, multi-modal processing for diverse user populations. To summarize, the main contributions are as follows:

- To develop a bilingual text analysis module to accurately identify emotions in English and Tamil text.
- To design a multimodal data processing system that captures both textual and visual emotions.
- To build a real-time emotion detection system for immediate emotion classification.
- To create a visual emotion representation mechanism to track and display emotional trends over time.

The rest of this article is organized as follows. Section II reviews the related works. Section III presents the introduction of proposed work. In Section IV, implementation details for each component of the proposed framework are presented.Section V presents the results and dsiscussions of the proposed work. Finally, Section VI draws the conclusion.

II. Related Works

Emotion recognition systems have evolved significantly with the advent of deep learning, multimodal analysis, and language-specific models. Despite this progress, challenges remain in handling multi-input modalities, supporting regional languages, ensuring real-time responsiveness, and capturing emotional evolution. This section categorizes existing research into four major domains: multi-label emotion detection, multimodal fusion, facial and textual emotion analysis, and emotion-aware applications. Each group is reviewed with an emphasis on limitations that our system aims to address.

A. Multi-Label Emotion Detection

Recent advancements in multi-label classification have emphasized the importance of capturing emotion dependencies and contextual interactions. Deng et al. [1] introduced the MEDAFS framework that leverages multiple sub-models for emotion-specified feature extraction and correlation learning. Although effective on datasets like RenCECps, the assumption of label independence limits its ability to model complex co-occurring emotions. Similarly, Zhou et al. [6] employed prompt consistency using emotion-specific prompts and synonym variations, improving ambiguity handling. However, the model's restriction to binary and discrete emotion labels limits its application in real-world conversations where emotions evolve continuously. These studies point to a need for flexible, fine-grained systems capable of both multi-label classification and context-sensitive inference across languages.

B. Multimodal Emotion Detection and Feature Fusion

Several works have explored multimodal emotion detection using combinations of visual, textual, and physiological cues. Kim et al. [3] proposed MultiDo-VEmoBar for IoT-based environments, maximizing accuracy across domains, but error propagation in detection stages reduced reliability. Sharma et al. [7] demonstrated a Vision Transformer (ViT) model for meme-based emotion recognition, yet struggled with abstract visuals and text occlusion common in social media. Tian et al. [10] attempted visual-audio fusion using Bayesian models for dimensional emotion mapping but lagged behind in accuracy when compared to deep models. These studies highlight the promise of multimodal systems, but also the technical trade-offs involved in fusion complexity, data compatibility, and computational latency. Our system addresses these concerns by offering independent yet switchable text and facial modules, reducing overhead without sacrificing flexibility.

C. Facial Expression and Textual Emotion Analysis

Visual emotion recognition remains a core research area with significant progress in CNN and RNN-based models. Wu et al. [4] emulated the human cognition process using a dual-system CNN/RNN architecture for facial and physiological input. Despite its innovation, reliance on RGB images limits spatial detail capture, affecting real-time prediction precision. Yang et al. [13] proposed Stimuli-Aware Visual Analysis by integrating specialized subnetworks, but emotional ambiguity in expressions reduced classification confidence. On the textual side, Mao et al. [8] evaluated pre-trained language models (PLMs) for masked-prompt emotion detection and exposed bias in label-word mapping, especially in fine-grained tasks. Our system addresses these issues by implementing separate deep models for English and Tamil text and a CNN model fine-tuned on FER2013 to provide more interpretable and culturally inclusive predictions.

D. Emotion-Aware Applications in Real-Time Systems

Emotion recognition is increasingly embedded in broader applications such as fake news detection and mental health assessment. Choudhry et al. [5] integrated emotion detection as an auxiliary task in fake news classification, yet dependency on automated emotion annotations compromised performance reliability. Yu et al. [9] built C-DepressNet for cloud-edge collaborative depression detection, but restricted input to facial features alone, missing cross-modal cues. Zheng et al. [14] extended temporal emotion detection using knowledge graphs and Transformers but acknowledged the model's inability to fully encapsulate emotional diversity. These application-specific frameworks reaffirm the importance of interpretable, cross-lingual systems that can monitor user sentiment over time—capabilities directly embedded in our proposed temporal tracking module.

E. Real-Time Challenges and Temporal Tracking

Emotion recognition in dynamic environments must balance model complexity, processing latency, and contextual consistency. Zhang et al. [11] presented a dual-channel cyber-physical network for driver emotion recognition but noted performance drops due to over-smoothing in deep graph layers. Wang et al. [12] created the MGEED dataset integrating facial, EEG, and depth data but found that lab-controlled settings limited real-world expressiveness. Chiranjeevi et al. [15] used Constrained Local Models for personalized face classification, yet the dependency on training identity-specific data hindered scalability.

Our system overcomes these issues through lightweight on-device processing, dynamic logging, and an hourly emotion visualization module, thereby enhancing longitudinal emotion understanding with high user adaptability.

III. Proposed System

The proposed work focuses on the design and implementation of a multi-mode input emotion detection system that empowers users to express their emotional states using the input modality that feels most natural or accessible to them—either through typed text or facial expressions. In contrast to conventional multimodal systems that integrate and fuse inputs from multiple modalities simultaneously, this work emphasizes input flexibility and user preference, allowing each mode to function independently yet cohesively within the overall system. This flexible, modular approach not only simplifies user interaction but also enhances system usability across diverse real-world scenarios where only one input mode might be available or preferred.

One of the key motivations behind this framework stems from the limitations of existing emotion detection solutions, many of which are narrowly focused on enhancing user experience in human-computer interaction without accounting for linguistic diversity, real-time adaptability, or accessibility. These systems often lack support for low-resource languages like Tamil, fail to integrate natural facial expression analysis seamlessly, or rely on computationally intensive multimodal fusion techniques that are not suitable for lightweight, real-time deployment. Furthermore, emotion detection is inherently a subjective and culturally contextual task. A one-size-fits-all model often does not generalize well across different user groups, languages, or interaction patterns. To address these gaps, the proposed framework adopts a user-centric and context-aware approach by enabling multi-mode input emotion recognition tailored to user comfort, device capability, and application context. The overall architectural design of the proposed system is given in Fig 1.

The framework is composed of the following three key modules, each addressing a specific aspect of emotion analysis:

- **Bilingual Text-Based Emotion Detection Module**
  This module supports emotion classification from text inputs provided in both Tamil and English. Leveraging natural language processing (NLP) techniques and deep learning models, the system is capable of analyzing linguistic features and contextual sentiment embedded in user-generated text. By supporting Tamil—a language underrepresented in many NLP research efforts—the system promotes linguistic inclusivity and ensures wider accessibility. The module is optimized to perform real-time inference and is designed for integration with mobile and web-based interfaces.
- **Real-Time Facial Emotion Detection Module**
  This component processes visual input from a camera feed to detect facial expressions in real-time. It employs computer vision techniques, such as Haar Cascade classifiers for face detection and a Convolutional Neural Network (CNN) model for emotion classification. Each frame is analyzed to recognize expressions corresponding to predefined emotion categories like Happy, Sad, Angry, Disgust, Fear, Surprise, and Neutral. This module operates independently from the text-based module, allowing the system to function even in situations where typing is not feasible or appropriate.
- **Emotion Tracking Over Time Module**
  This module focuses on the temporal analysis of emotional states. It tracks the user's emotions over time by logging predictions from the text and facial detection modules into a structured format (CSV), which is then processed and visualized through interactive graphs. Users can select specific dates to view emotion trends at an hourly granularity, enabling self-reflection and emotional monitoring. This capability is particularly valuable for applications in mental health support, user experience research, and longitudinal sentiment analysis.

A. Bilingual Text-Based Emotion Detection

This module focuses on analysing user-entered text to predict emotional states using language-specific deep learning models. This module functions through an interactive chat interface, where users can type messages in either English or Tamil based on their preference. Upon submitting a message, the system detects the language and routes it to the corresponding emotion detection pipeline. This setup ensures high prediction accuracy by leveraging dedicated models fine-tuned for each language. The detected emotion is immediately displayed in the chat interface, simulating a conversational response, while also being stored for trend analysis.

1) Working Mechanism and Models Used: When a user enters a message in the chat box, the system first performs language detection to identify whether the input is in English or Tamil. The Input text undergoes tokenization process. If the

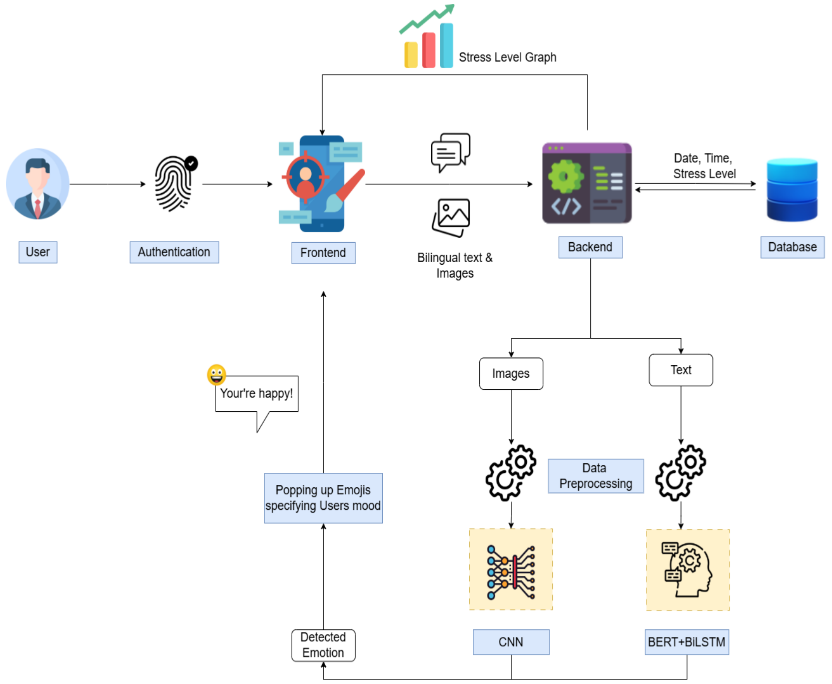


Fig. 1: Architecture Diagram: Multi-Mode Bilingual Emotion Detection System

message is in English, it is passed to a BERT + BiLSTM-based classifier, which processes the tokenized and embedded input, to predict the corresponding emotion. For Tamil text inputs, the system utilizes a Tamil BERT model fine-tuned for emotion classification tasks. After inference, the resulting emotion label is returned as a textual response in the chat interface. Simultaneously, the input message, detected language, timestamp, and predicted emotion are logged into a CSV file. This logging mechanism supports further analysis in the emotion tracking module.

For English language detection, a hybrid architecture combining BERT for contextual embeddings and a BiLSTM for sequence modeling is employed. For Tamil input, a fine-tuned Tamil BERT model is used to leverage contextual understanding in the native language. The system also integrates a lightweight language detection model to switch between classifiers dynamically.

2) Chat Interface Output and Log Entry Generation: The main outputs of this module include the emotion-predicted response shown in the chat interface and an entry in the CSV file containing message metadata and emotion labels. These outputs enable users to receive instant emotional feedback while enabling backend systems to monitor emotional patterns over time for insights and trend visualization in later modules.

B. Real Time Facial Emotion Detection

Facial Emotion Detection, focuses on identifying emotional expressions from a user's facial cues using real-time video feed. This module operates through continuous camera access, allowing live facial analysis without requiring user intervention beyond camera permissions. The captured stream is processed frame-by-frame to detect facial features and classify emotions using a pre-trained deep learning model. The recognized emotions are displayed directly on the video feed through labeled bounding boxes around detected faces and are simultaneously logged for future trend analysis.

1) Working Mechanism and Models Used: Upon activation, the system accesses the user's camera and begins capturing a live video stream. This stream is sent to the backend where it is decomposed into individual frames at regular intervals. Each frame is preprocessed, including resizing, normalization, and grayscale conversion (if necessary), to ensure compatibility with the model. A facial detection algorithm is first applied to locate faces within the frame. The extracted face regions are then passed to a CNN-based emotion classification model which predicts the emotional state. The frontend displays these predictions by drawing rectangles around the faces with emotion labels, providing dynamic visual feedback. In parallel, each prediction, along with a timestamp, is logged to a CSV file for trend tracking. This module employs a Convolutional Neural Network (CNN) model trained on a facial emotion dataset capable of distinguishing key emotions - Happy, Sad, Angry, Disgust, Fear, Surprise and Neutral. Face detection is carried out using Haar cascades or a lightweight deep learning-based face detector. The model is optimized for real-time inference and integrates seamlessly with the video processing pipeline.

2) Real-Time Visual Feedback and Logging: The key outputs from this module include the live emotion-labeled video feed and a structured CSV log containing timestamps and predicted emotions. This provides users with real-time feedback on their facial expressions and serves as valuable input for the third module that analyzes emotion trends over time.

C. Emotion Tracking Over Time

The third module, Emotion Tracking Over Time, focuses on analysing and visualizing the user's emotional patterns over different days. This module leverages the emotion predictions logged by the previous two modules (text-based and facial detection) to provide meaningful insights into the emotional trends of the user. It offers a user-friendly interface where users can navigate across dates and view emotion trends for each selected day. The output is displayed as a time-based graph, helping users better understand how their emotions fluctuate throughout the day.

1) Working Mechanism: When the user accesses the Emotion Tracking module, two navigation buttons—Previous Day and Next Day—are provided. Upon clicking one of these buttons, the selected date is sent to the backend. The backend reads the emotion log CSV file and filters the data entries corresponding to the specified date. These entries are then grouped by hour and emotion class (Happy, Sad, Angry, Disgust, Fear, Surprise and Neutral), and the count of each emotion for every hour is calculated. This grouped data is formatted and returned to the frontend, where it is rendered as a stacked bar graph. Each bar represents an hour in the day and is segmented by emotion classes, offering a clear visual distribution of emotions over time. This module does not involve a machine learning model but relies on efficient backend processing of logged data. The CSV file is processed using Python-based scripts (e.g., Pandas) for filtering, grouping, and aggregation. On the frontend, charting libraries such as Chart.js or D3.js are used to render the dynamic, interactive graphs based on the processed data.

2) Time-based Emotion Distribution Output: The main output of this module is an interactive graph showing the emotion distribution for each hour of the selected day. This helps users recognize daily emotional trends and observe how their mood evolves over time. This historical tracking capability adds long-term value by promoting emotional awareness and self-reflection.

IV. IMPLEMENTATION

This section outlines the complete implementation strategy and system architecture of the proposed Multi-Mode Bilingual Emotion Detection System. The framework is built to support emotion recognition from multiple input types—text or facial expressions—allowing users to choose their preferred mode of interaction. The three major modules include Bilingual Text-based Emotion Detection, Facial Emotion Detection, and Emotion Tracking Over Time. Each module is purposefully crafted to ensure accurate, real-time predictions while maintaining a smooth and intuitive user experience. The system accommodates both English and Tamil language inputs through separate NLP models and enables live video analysis for facial emotion recognition. Additionally, the emotion logging and visualization module allows users to track emotional variations over time through interactive graphical reports. The combination of these modules offers a comprehensive, language-inclusive, and flexible approach to emotion detection. By incorporating language-specific models, live video stream processing, and data-driven emotion tracking, the proposed system ensures reliability and inclusivity across diverse user bases, making it applicable in domains like mental wellness, education, and user feedback systems.

A. Dataset Description

1) TamilEmo: Fine-grained Emotion Detection Dataset for Tamil: The TamilEmo dataset is a large-scale, manually annotated benchmark developed to facilitate fine-grained emotion recognition in the Tamil language. It specifically focuses on leveraging user-generated content by collecting and labeling over 42,000 YouTube comments in Tamil, making it the largest manually curated Tamil emotion dataset to date. TamilEmo supports research across various natural language processing (NLP) tasks such as emotion classification, sentiment analysis, dialogue understanding, and multilingual emotion modeling. The dataset is designed to enable both multi-class emotion classification and hierarchical emotion grouping. It offers three levels of granularity—3-class, 7-class, and 31-class labels—to accommodate varying task complexities and model capabilities. The dataset is provided in TSV (Tab-Separated Values) format, partitioned into training and testing splits. Each entry consists of a comment and its associated emotion label.

Dataset Structure:

- **Format:** TSV files (`train.tsv`, `test.tsv`)
- **Columns:**
  1) **Text:** A Tamil YouTube comment (string)
  2) **Category:** Corresponding emotion label (string)
- **Train Set:** 30,178 entries
- **Test Set:** 4,268 entries

2) FER-2013: Facial Expression Recognition 2013 Dataset: The FER-2013 dataset is a large-scale, benchmark dataset designed for the task of facial expression recognition in images. It was introduced during the ICML 2013 Challenges in Representation Learning and remains one of the most widely used datasets for emotion classification from facial images. The dataset supports a variety of research areas including emotion detection, facial analysis, affective computing, and deep learning for visual recognition. FER-2013 consists of grayscale face images labeled with one of seven discrete emotion categories. All images are of uniform size and pre-aligned, making the dataset well-suited for training and evaluating deep learning models.

Dataset Structure:

- **Format:** Zipped folder with images organized in a hierarchical directory structure
- **Folder Structure:**
  1) Root folder contains separate subfolders for each emotion category.
  2) Each subfolder includes all images corresponding to that emotion label.

3) **Subfolder Names (Emotion Categories):** Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral

- **Image Details:**
  1) Size: 48×48 pixels
  2) Color: Grayscale (1 channel)
  3) Format: `.jpg`

Data Statistics:

- **Train Set:** 28,709 images
- **Validation Set (PublicTest):** 3,589 images
- **Test Set (PrivateTest):** 3,589 images
- **Image Size:** 48×48 pixels (Grayscale)

B. Transformer-Based Bilingual Emotion Recognition

The Bilingual Text-based Emotion Detection module forms the foundation of one of the system's multi-mode input capabilities, allowing users to seamlessly express emotions using either English or Tamil. This module emulates a real-time chat interface where users can type in their messages and receive instant emotion-based feedback, creating an interactive and language-inclusive emotion detection experience.

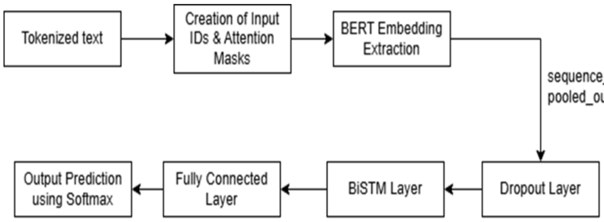
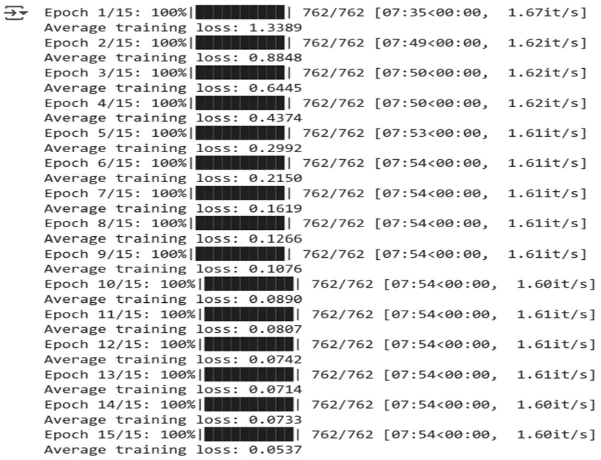

Tokenized text
Creation of Input IDs & Attention Masks
BERT Embedding Extraction
sequence_output
pooled_output
Dropout Layer
BiSTM Layer
Fully Connected Layer
Output Prediction using Softmax

Fig. 2: Architecture of Proposed Hybrid BERT + BiLSTM Model

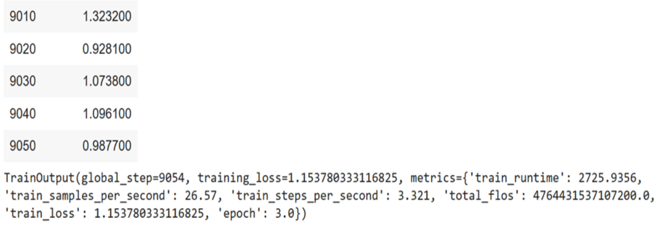
To ensure accurate emotion recognition across languages, separate Transformer-based models were trained for English and Tamil. The proposed hybrid BERT + BiLSTM model, as shown in Fig. 2, was trained on an annotated dataset of emotion-labeled English sentences to predict emotions from English text.



```
Epoch 1/15: 100%|██████████| 762/762 [07:35<00:00,  1.67it/s]
Average training loss: 1.3389
Epoch 2/15: 100%|██████████| 762/762 [07:49<00:00,  1.62it/s]
Average training loss: 0.8848
Epoch 3/15: 100%|██████████| 762/762 [07:50<00:00,  1.62it/s]
Average training loss: 0.6445
Epoch 4/15: 100%|██████████| 762/762 [07:50<00:00,  1.62it/s]
Average training loss: 0.4374
Epoch 5/15: 100%|██████████| 762/762 [07:53<00:00,  1.61it/s]
Average training loss: 0.2992
Epoch 6/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.2150
Epoch 7/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.1619
Epoch 8/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.1266
Epoch 9/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.1076
Epoch 10/15: 100%|██████████| 762/762 [07:54<00:00,  1.60it/s]
Average training loss: 0.0890
Epoch 11/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.0807
Epoch 12/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.0742
Epoch 13/15: 100%|██████████| 762/762 [07:54<00:00,  1.61it/s]
Average training loss: 0.0714
Epoch 14/15: 100%|██████████| 762/762 [07:54<00:00,  1.60it/s]
Average training loss: 0.0733
Epoch 15/15: 100%|██████████| 762/762 [07:54<00:00,  1.60it/s]
Average training loss: 0.0537
```

Fig. 3: Training of Proposed Hybrid BERT + BiLSTM Model

In parallel, for Tamil text emotion prediction, a BERT model pre-trained on the Tamil language was fine-tuned for classification using the TamilEmo dataset. The training interfaces for both models are illustrated in Fig. 3 and Fig. 4, respectively.



9010	1.323200
9020	0.928100
9030	1.073800
9040	1.096100
9050	0.987700

```
TrainOutput(global_step=9054, training_loss=1.153780333116825, metrics={'train_runtime': 2725.9356,
'train_samples_per_second': 26.57, 'train_steps_per_second': 3.321, 'total_flos': 4764431537107200.0,
'train_loss': 1.153780333116825, 'epoch': 3.0})
```

Fig. 4: Fine-tuning of Tamil BERT on the TamilEmo dataset

The process begins when the user selects the bilingual emotion detection option, which redirects them to a dedicated React.js page displaying a chatbot interface. The interface provides a dropdown or toggle to choose the preferred input language—either English or Tamil. Once the user inputs a message and selects the language, the message is sent via a RESTful POST request from the React frontend to the Flask backend endpoint /analyze_text.

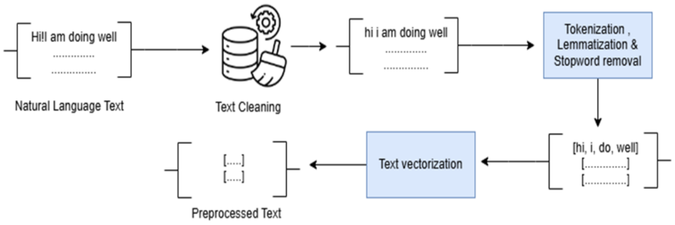


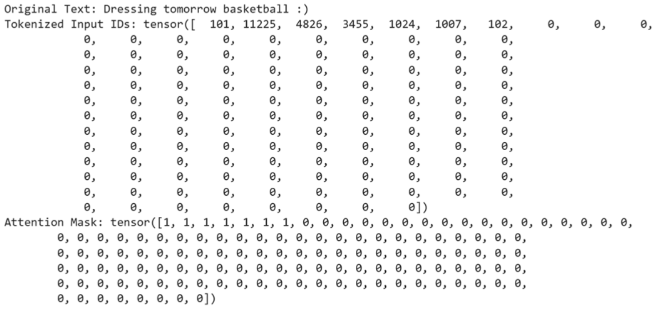
Fig. 5: NLP Text Preprocessing and Vectorization Workflow

At the backend, the system extracts both the language and the text from the received request. Depending on the selected language, the backend performs language-specific preprocessing, such as punctuation removal, lemmatization, and tokenization as illustrated in Fig 5. For English inputs, the preprocessed text is fed into a hybrid BERT + BiLSTM model, while Tamil inputs are passed through a Tamil-BERT model fine-tuned for emotion detection. A sample illustration of the text preprocessing output is presented in Fig. 6. Upon prediction, the detected emotion is sent back to the frontend and displayed as the chatbot's response in the chosen language, enhancing the interactivity and personal connection with the user. Simultaneously, the predicted emotion, along with a timestamp and language tag, is logged into a CSV file for future analysis in the Emotion Tracking module.

Key Features of the Module:

- **Language-Specific Modeling:** Supports both English and Tamil using dedicated NLP pipelines to ensure higher accuracy and inclusivity.
- **Real-Time Emotion Feedback:** Immediate display of the predicted emotion enhances user engagement and application responsiveness.

- **Emotion Logging for Analysis:** All detected emotions are persistently stored with timestamp and language, forming the data backbone for emotion trend visualization.



```
Original Text: Dressing tomorrow basketball :)
Tokenized Input IDs: tensor([  101, 11225,  4826,  3455,  1024,  1007,   102,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0])
Attention Mask: tensor([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0])
```

Fig. 6: Sample of Preprocessed Input Text

Algorithm 1 Text Preprocessing

```
Load dataset $df$
Remove user handles from $df['Text']$ using $nfx.remove\_userhandles$
Remove stopwords from $df['Clean\_Text']$ using $nfx.remove\_stopwords$
Set $Xfeatures \leftarrow df['Clean\_Text']$
Set $ylabels \leftarrow df['Emotion']$
Split dataset into training and testing sets: $(x\_train, x\_test, y\_train, y\_test) \leftarrow train\_test\_split(Xfeatures, ylabels, test\_size = 0.3, random\_state = 42)$
Load BERT tokenizer $tokenizer \leftarrow BertTokenizer.from\_pretrained('bert - base - uncased')$
Set $max\_seq\_length \leftarrow 128$
Initialize empty lists $input\_ids$ and $attention\_masks$
for each $text$ in $x\_train$ do
    Encode $text$ using $tokenizer.encode\_plus()$ with parameters:
    Add special tokens, set max length, apply padding and truncation
    Append tokenized input IDs and attention masks to $input\_ids$ and $attention\_masks$
end for
Concatenate $input\_ids$ and $attention\_masks$ into tensors $x\_train\_ids$ and $x\_train\_masks$
Repeat the tokenization process for $x\_test$ to get $x\_test\_ids$ and $x\_test\_masks$
Return tokenized training and testing sets: $(x\_train\_ids, x\_train\_masks, x\_test\_ids, x\_test\_masks)$
```

Algorithm 1 outlines the text preprocessing pipeline used to prepare raw emotion-labeled text for model training. It includes user handle and stopword removal, splitting the dataset, and tokenizing the input using a BERT tokenizer. Algorithm 2 presents the architecture of the proposed hybrid BERT-biLSTM model. It integrates semantic embeddings from BERT with sequential context modeling from a BiLSTM layer, followed by a fully connected layer to classify the emotions. Together, these algorithms enable effective bilingual emotion recognition from textual inputs.

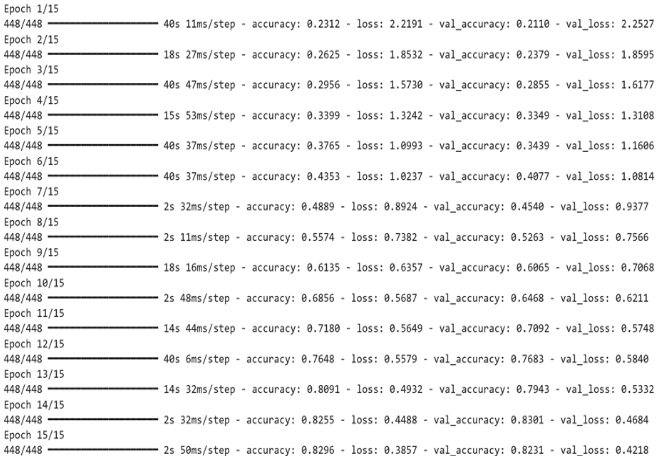
Algorithm 2 BERT-biLSTM Model Architecture

```
Input: Pre-trained BERT model $bert\_model$, Hidden size $hidden\_size$, Number of output classes $num\_classes$
Output: Logits for classification
Define class BERTbiLSTMModel(nn.Module)
Initialize with $bert\_model$, $hidden\_size$, and $num\_classes$
Set self.bert $\leftarrow bert\_model$
Set self.dropout $\leftarrow$ Dropout layer with $p = 0.1$
Set self.bilstm $\leftarrow$ BiLSTM layer with:
    input_size = $bert\_model.config.hidden\_size$
    hidden_size = $hidden\_size$
    num_layers = 1
    batch_first = True
    bidirectional = True
Set self.fc $\leftarrow$ Fully connected linear layer with input size $hidden\_size \times 2$ and output size $num\_classes$
function FORWARD($input\_ids$, $attention\_mask$)
    Get BERT outputs: $outputs \leftarrow$ self.bert($input\_ids, attention\_mask$)
    Extract pooled output: $pooled\_output \leftarrow outputs[1]$
    Apply dropout: $pooled\_output \leftarrow$ self.dropout($pooled\_output$)
    Unsqueeze pooled output: $pooled\_output \leftarrow pooled\_output.unsqueeze(1)$
    Pass through BiLSTM: $(bilstm\_output, \_) \leftarrow$ self.bilstm($pooled\_output$)
    Extract last hidden state: $final\_hidden \leftarrow bilstm\_output[:, -1, :]$
    Compute logits: $logits \leftarrow$ self.fc($final\_hidden$)
    return $logits$
end function
```

C. Automated Facial Emotion Recognition in Live Video Streams

The Facial Emotion Detection module enables real-time facial expression analysis using a CNN-based model integrated with computer vision techniques. This module is designed to operate independently and is triggered via the user interface built with React.js. The interface presents two buttons—Start Detection and Stop Detection—that initiate and terminate the detection process, respectively. To ensure accurate facial emotion detection, a CNN model was trained on the FER-2013 dataset, as illustrated in Fig. 7.

When the user initiates detection by clicking the Start button, the frontend sends a trigger to the Flask backend through an HTTP request. Upon receiving the request, the backend spawns a subprocess that executes the Facial_Emotion_Detection.py script. This script handles real-time video stream processing by accessing the user's webcam. Each frame captured from the webcam is converted to grayscale, and faces are detected using OpenCV's haarcascade_frontalface_default.xml classifier. Once a face is localized, the region of interest (ROI) is cropped, resized to 48x48 pixels, and normalized to fit the input shape expected by the CNN model. The preprocessed face image is then passed into a pre-trained model (model.h5) to predict the corresponding emotion.



```
Epoch 1/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 40s 11ms/step - accuracy: 0.2312 - loss: 2.2191 - val_accuracy: 0.2110 - val_loss: 2.2527
Epoch 2/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 18s 27ms/step - accuracy: 0.2625 - loss: 1.8532 - val_accuracy: 0.2379 - val_loss: 1.8595
Epoch 3/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 40s 47ms/step - accuracy: 0.2956 - loss: 1.5730 - val_accuracy: 0.2855 - val_loss: 1.6177
Epoch 4/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 15s 53ms/step - accuracy: 0.3399 - loss: 1.3242 - val_accuracy: 0.3349 - val_loss: 1.3108
Epoch 5/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 40s 37ms/step - accuracy: 0.3765 - loss: 1.0993 - val_accuracy: 0.3439 - val_loss: 1.1606
Epoch 6/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 40s 37ms/step - accuracy: 0.4353 - loss: 1.0237 - val_accuracy: 0.4077 - val_loss: 1.0814
Epoch 7/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 2s 32ms/step - accuracy: 0.4889 - loss: 0.8924 - val_accuracy: 0.4540 - val_loss: 0.9377
Epoch 8/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - accuracy: 0.5574 - loss: 0.7382 - val_accuracy: 0.5263 - val_loss: 0.7566
Epoch 9/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 18s 16ms/step - accuracy: 0.6135 - loss: 0.6357 - val_accuracy: 0.6065 - val_loss: 0.7068
Epoch 10/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 2s 48ms/step - accuracy: 0.6856 - loss: 0.5687 - val_accuracy: 0.6468 - val_loss: 0.6211
Epoch 11/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 14s 44ms/step - accuracy: 0.7180 - loss: 0.5649 - val_accuracy: 0.7092 - val_loss: 0.5748
Epoch 12/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 40s 6ms/step - accuracy: 0.7648 - loss: 0.5579 - val_accuracy: 0.7683 - val_loss: 0.5840
Epoch 13/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 14s 32ms/step - accuracy: 0.8091 - loss: 0.4932 - val_accuracy: 0.7943 - val_loss: 0.5332
Epoch 14/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 2s 32ms/step - accuracy: 0.8255 - loss: 0.4488 - val_accuracy: 0.8301 - val_loss: 0.4684
Epoch 15/15
448/448 ━━━━━━━━━━━━━━━━━━━━ 2s 50ms/step - accuracy: 0.8296 - loss: 0.3857 - val_accuracy: 0.8231 - val_loss: 0.4218
```

Fig. 7: Training the CNN Model on the FER2013 Dataset

For each detected face, a red bounding box is drawn around the face, and the predicted emotion label is superimposed above it using OpenCV's text rendering. The video with emotion-annotated frames is displayed in real-time through an OpenCV window (cv2.imshow). Simultaneously, each prediction is logged along with its timestamp into a CSV file. This log serves as the backend data source for the Emotion Tracking module and helps in analyzing trends over time.

Key Features of the Module:

- **Real-Time Detection:** The system captures and processes frames continuously, delivering immediate emotion recognition feedback via a live video stream.
- **CNN-Based Classification:** A custom-trained Convolutional Neural Network classifies emotions into categories — Happy, Sad, Angry, Disgust, Fear, Surprise, and Neutral.
- **Facial Region Localization:** Utilizes Haar Cascade Classifier to detect and extract facial features from raw webcam input.
- **Trends Logging:** Each detected emotion is stored with a timestamp in a CSV file, enabling retrospective analysis of emotional patterns.

Algorithm 3 describes the real-time facial emotion detection process using a webcam feed. It starts by loading a Haar Cascade face detector and a pre-trained CNN-based emotion classifier. Each video frame is converted to grayscale, and faces are detected. For every detected face, the region of interest is preprocessed and passed to the classifier to predict the emotion label. The result is then displayed on the video stream in real-time.

Algorithm 3 Real-time Face Emotion Detection

```
face_classifier ← Load Haar Cascade model
emotion_classifier ← Load pre-trained emotion model
labels ← ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral',
    'Sad', 'Surprise']
cap ← Initialize webcam
while True do
    frame ← Capture video frame
    gray ← Convert frame to grayscale
    faces ← Detect faces in gray frame
    for each (x, y, w, h) in faces do
        roi_gray ← Extract and resize region of interest
        if roi_gray is not empty then
            roi ← Normalize and preprocess roi_gray
            prediction ← emotion_classifier.predict(roi)
            label ← labels[argmax(prediction)]
            Draw rectangle and display label on frame
        else
            Display 'No Faces' message
        end if
    end for
    Show frame with detections
    if key press $q$ is detected then
        Break loop
    end if
end while
Release webcam and close windows
```

D. Time-Based Emotion Logging and Monitoring

This module enables users to view emotion trends based on historical data as illustrated in Fig 8. When the user selects a specific date on the frontend, it triggers a backend endpoint that processes the logged emotion predictions and returns the hourly distribution of emotions for that day. This information is used to visualize how emotions varied throughout the day in a graph format on the frontend.

The Flask backend handles the data processing. It loads a CSV file containing timestamped emotion predictions, filters records by the selected date, and groups the data by hour and emotion category. The result is then converted into a JSON response, structured in a format that is directly usable for frontend visualization libraries. The goal of this module is to allow emotion tracking over time, which can provide deeper insights into emotional patterns and behavior throughout the day. The emotion tracking begins with the frontend requesting emotion logs for a specific date. This request triggers a Flask backend route (/get_emotion_data), which reads a centralized CSV log file containing emotion predictions with timestamps. The backend first ensures consistent formatting by assigning appropriate column headers and then converts the timestamp into a date format to filter entries for the selected day. Once filtered, the data is grouped by hour, and the count of each emotion is aggregated. This hourly aggregation provides a temporal resolution that highlights how emotions fluctuate throughout the day. The result is a list of dictionaries, where each dictionary represents a single hour with associated counts of Happy, Sad, Angry, Disgust, Fear, Surprise and Neutral. These are returned to the frontend as a JSON response. The frontend consumes this structured JSON data and dynamically renders visual plots—typically line or bar graphs—making it easier for users to interpret the emotional timeline. This enables temporal emotion monitoring, which can be useful in fields such as mental health tracking, user sentiment analysis, or personalized experience design.

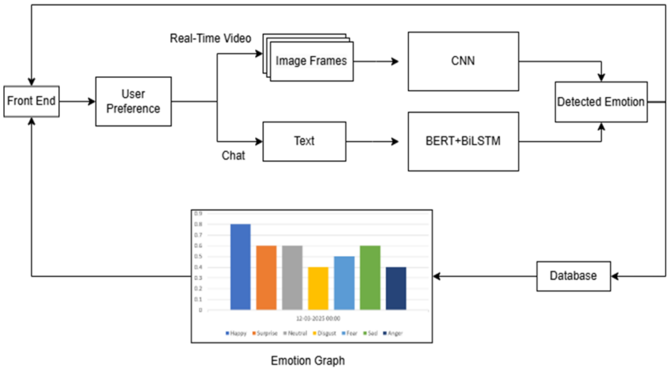


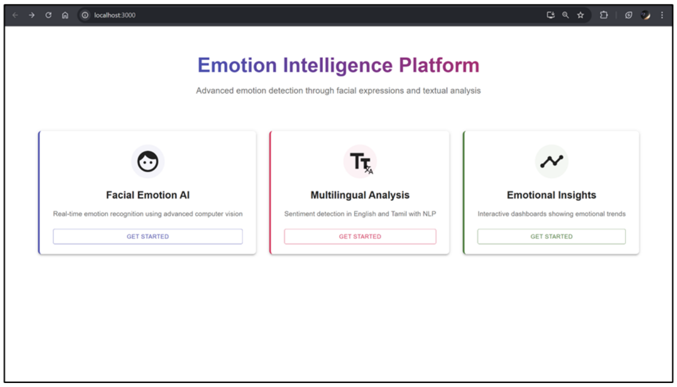
Fig. 8: Hourly Emotion Distribution and Visualization

Key Features of the Module:

- **User-Specified Date Filter:** Receives a date input from the frontend to filter log data.
- **CSV Data Parsing:** Reads and processes emotion logs from a persistent CSV file.
- **Hourly Emotion Aggregation:** Groups emotions by hour to analyze distribution throughout the day.
- **Trend Visualization:** The frontend renders interactive graphs, providing visual emotion insights.

V. RESULTS AND DISCUSSION

The results of the Multi-Mode Bilingual Emotion Detection System can be categorized into three main areas: Model Performance, Real-Time Multi-Mode Emotion Detection, and Emotion Visualization and Insights. Each category demonstrates the effectiveness of the implemented deep learning models and supporting technologies, showcasing how they contribute to building an emotionally aware, responsive, and user-centric application experience. This is further illustrated in Fig 9, which shows the home page of the web application, providing users with a seamless interface to choose between text-based or image-based emotion detection modes, along with navigation to emotion insights and history tracking, emphasizing the system's intuitive design and multi-modal functionality.



Emotion Intelligence Platform

Advanced emotion detection through facial expressions and textual analysis

Facial Emotion AI

Real-time emotion recognition using advanced computer vision

Multilingual Analysis

Sentiment detection in English and Tamil with NLP

Emotional Insights

Interactive dashboards showing emotional trends

Fig. 9: User Preference or Home Page of the Web App

A. Model Performance

1) Convolutional Neural Network: Fig. 10 displays two plots illustrating the training and validation performance of a CNN over 15 epochs. The left plot shows the training and validation accuracy, while the right plot shows the training and validation loss.

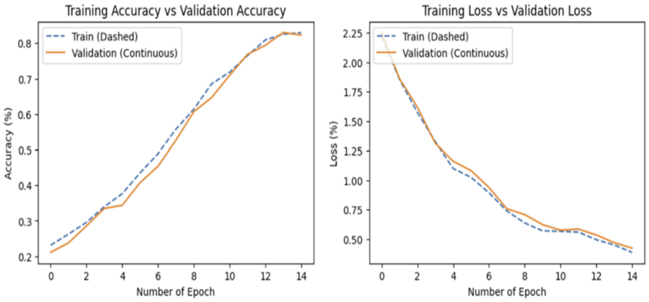


Fig. 10: Accuracy And Loss Graph for CNN Trained on FER2013 DatasetUser Preference or Home Page of the Web App

In the left plot, the training accuracy shows a consistent upward trend throughout all epochs. Starting from an initial accuracy of approximately 0.23, the accuracy steadily improves as the model learns from the training data. Around epoch 8, the training accuracy crosses 0.6 and continues to rise, eventually reaching around 0.83 by epoch 14. This indicates that the model is effectively learning the patterns within the training dataset and improving its classification capabilities. The validation accuracy follows a similar trend to the training accuracy. Beginning slightly below 0.20, the validation accuracy increases in tandem with the training accuracy, indicating that the model is generalizing well during the initial epochs. The validation accuracy closely tracks the training accuracy up to epoch 13–14, ultimately peaking around 0.82, nearly matching the training performance. This suggests that the model generalizes well and does not suffer from major overfitting issues. Unlike typical cases where validation accuracy plateaus or declines, in this training run, the validation accuracy continues to improve alongside the training accuracy.

The closeness of the two curves indicates minimal overfitting, suggesting a well-regularized and balanced training process.

In the right plot, the training loss starts at a high value (above 2.2) and decreases steadily across the training epochs. This consistent decline indicates that the model is effectively minimizing error on the training data. By the final epoch (14), the loss drops to around 0.3, demonstrating strong convergence during training. Similar to the training loss, the validation loss decreases steadily throughout training. Initially high (close to 2.3), the loss decreases with each epoch, closely tracking the training loss. It ends up nearly equal to the training loss at approximately 0.35 by the final epoch, further supporting the model's ability to generalize well to unseen data. The parallel decline in both training and validation losses, along with their convergence near the end, suggests that the model is not only learning effectively but also generalizing without significant overfitting. There is no visible divergence between the two loss curves, which is a strong indicator of robust model training.

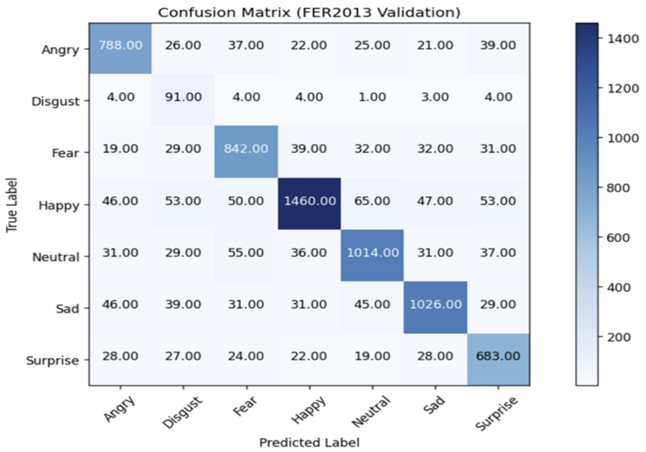
Based on the training and validation plots, the following conclusions can be drawn: the model demonstrates a strong and stable learning process throughout all epochs, with both accuracy and loss metrics showing consistent improvement. The validation accuracy closely follows the training accuracy, and the validation loss mirrors the training loss, indicating that the model generalizes well to unseen data. The near-convergence of both accuracy and loss curves between training and validation sets suggests that overfitting is not a concern in this training process. The model has reached high accuracy (0.82) with low loss (0.3), reflecting an effective training regime suitable for deployment or further fine-tuning.

TABLE I: Comparison of Model Accuracy on FER2013 Dataset

Model	Accuracy (%)
Local Learning BOW	67.48
Ad-Corre	72.03
VGGNet	73.28
ResEmoteNet	79.79
CNN (Proposed)	82.96

From Table 1, the proposed CNN model achieved the highest performance on the FER2013 dataset with an accuracy of 82.96%, outperforming all baseline models. Compared to traditional approaches such as Local Learning BOW (67.48%) and Ad-Corre (72.03%), and even advanced deep models like VGGNet (73.28%) and ResEmoteNet (79.79%), the CNN demonstrated superior feature extraction and classification capabilities. This significant performance boost highlights the effectiveness of the CNN's architecture in capturing subtle facial emotion features from complex image data. The improvement in accuracy confirms that our model offers a more reliable and precise method for facial emotion recognition, making it well-suited for real-time applications where accuracy is crucial.

The confusion matrix for the FER2013 validation set (Fig 11) reveals varying performance across emotion classes, with "Happy" achieving the highest true positives (1460) and "Disgust" the lowest (91), indicating potential class imbalance or difficulty in recognizing subtle expressions. Misclassifications are notable, such as "Angry" being confused with "Fear" (37) and "Happy" with "Neutral" (65), suggesting overlapping features or model bias. The model performs well on dominant classes but struggles with minority classes like "Disgust" and "Surprise," highlighting the need for targeted improvements in these areas.



True Label \ Predicted Label	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	788.00	26.00	37.00	22.00	25.00	21.00	39.00
Disgust	4.00	91.00	4.00	4.00	1.00	3.00	4.00
Fear	19.00	29.00	842.00	39.00	32.00	32.00	31.00
Happy	46.00	53.00	50.00	1460.00	65.00	47.00	53.00
Neutral	31.00	29.00	55.00	36.00	1014.00	31.00	37.00
Sad	46.00	39.00	31.00	31.00	45.00	1026.00	29.00
Surprise	28.00	27.00	24.00	22.00	19.00	28.00	683.00

Fig. 11: Confusion Matrix of CNN Model on Validation Set

Fig 12 illustrates the real-time facial emotion detection interface, where the system captures live video feed from the webcam and processes each frame to predict the user's emotional state. Detected emotions are displayed dynamically above the face with bounding boxes and corresponding labels. This visual feedback enables immediate and interactive emotion recognition, enhancing user engagement and awareness.

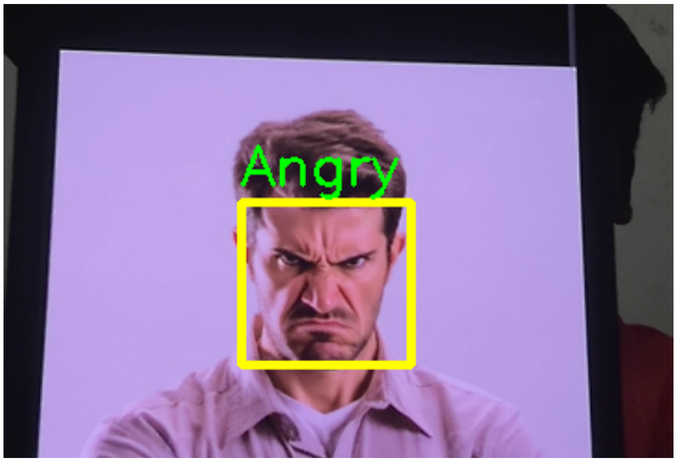
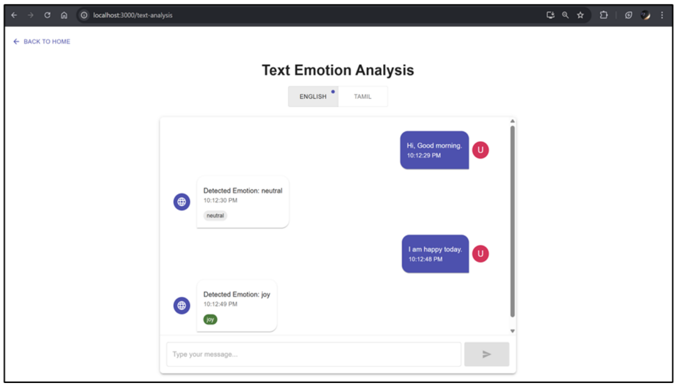


Fig. 12: Real-Time Facial Emotion Detection

2) Hybrid BERT + BiLSTM Model: Fig 3 illustrates the training phase of the BERT + BiLSTM model, showcasing a clear and steady learning progression. During the initial epochs (1–5), there is a sharp decline in training loss, highlighting the model's ability to quickly grasp underlying patterns in the training data and significantly reduce its prediction errors early on. As training progresses, the loss continues to decrease, albeit at a slower pace, suggesting that the model is refining its parameters and further optimizing performance. By the end of the 15th epoch, the training loss reaches a notably low value of approximately 0.0718, indicating that the model has effectively learned from the data and is producing minimal errors on the training set, reflecting a successful and well-converged training process.

Fig 13, showcases the emotion detection interface for English text using the BERT + BiLSTM model. Users input a sentence, and the system processes it to identify the underlying emotion based on contextual and semantic understanding. The predicted emotion is then displayed instantly, offering an intuitive and responsive experience for emotion-aware text interactions.



Hi, Good morning.
10:12:29 PM

Detected Emotion: neutral
10:12:30 PM
neutral

I am happy today.
10:12:48 PM

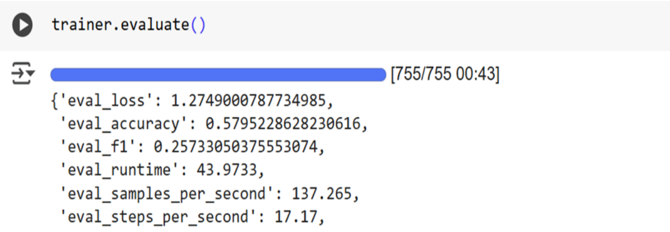
Detected Emotion: joy
10:12:49 PM
joy

Fig. 13: English Text Emotion Detection

3) Fine-Tuned Tamil BERT: Fig 14 provides insights into the evaluation performance of the Fine-tuned Tamil BERT model. The evaluation loss is recorded at 1.2749, indicating the average error made by the model on the validation dataset. The evaluation accuracy is approximately 57.95%, showing that the model correctly predicted over half of the validation samples, which is a decent performance given the complexity of emotion classification tasks. The F1-score, which considers both precision and recall, is relatively low at 0.2573, suggesting that while the model is accurate in some predictions, it may struggle with correctly identifying minority classes or more nuanced emotions. The evaluation ran for 43.97 seconds, processing about 137.27 samples per second, and completing 17.17 steps per second, indicating efficient computation during evaluation. Overall, these results reflect a promising start with room for improvement, especially in enhancing the model's generalization and class balance handling.

TABLE II: Comparison of Model Accuracy on TamilEmo Dataset

Model	Accuracy
ExtraTrees	21.32
KNN	29.45
Random Forest	36.62
Tamil BERT	57.95



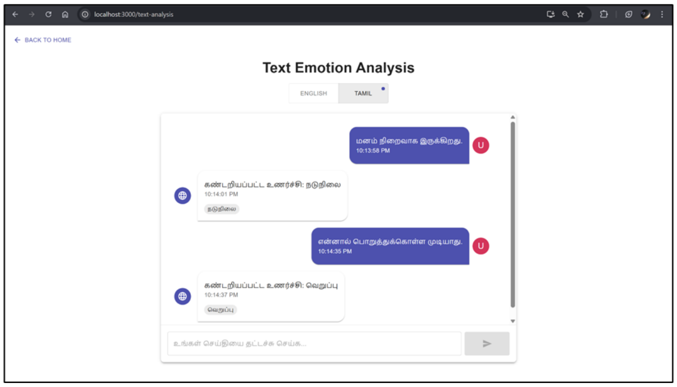
```
trainer.evaluate()
```

```
[755/755 00:43]
{'eval_loss': 1.2749000787734985,
 'eval_accuracy': 0.5795228628230616,
 'eval_f1': 0.25733050375553074,
 'eval_runtime': 43.9733,
 'eval_samples_per_second': 137.265,
 'eval_steps_per_second': 17.17,
```

Fig. 14: Validation of Fined-Tuned Tamil BERT on TamilEmo Dataset

From Table 2, the fine-tuned Tamil BERT model demonstrated the highest performance on the TamilEmo dataset with an accuracy of 57.95%, significantly outperforming traditional machine learning models such as ExtraTrees (21.32%), KNN (29.45%), and Random Forest (36.62%). This substantial margin illustrates the advantage of using transformer-based language models, particularly ones pretrained on native Tamil text, for capturing the nuances and emotional context in regional language data. The results emphasize the capability of Tamil BERT in understanding sentiment-rich expressions and syntactic structures specific to Tamil, making it a highly effective solution for emotion detection in low-resource languages.

Fig 15 illustrates the emotion detection process for Tamil text using a fine-tuned Tamil BERT model. Upon entering a Tamil sentence, the model analyzes the linguistic features and emotional cues specific to the language. The detected emotion is displayed in real time, enabling seamless and culturally relevant sentiment analysis for Tamil users.


localhost:3000/text-analysis
BACK TO HOME
Text Emotion Analysis
ENGLISH
TAMIL
U
U
உங்கள் செய்தியை தட்டச்சு செய்க...

மனம் நிறைவாக இருக்கிறது.
10:13:58 PM

கண்டறியப்பட்ட உணர்ச்சி: நடுநிலை
10:14:01 PM
நடுநிலை

என்னால் பொறுத்துக்கொள்ள முடியாது.
10:14:35 PM

கண்டறியப்பட்ட உணர்ச்சி: வெறுப்பு
10:14:37 PM
வெறுப்பு

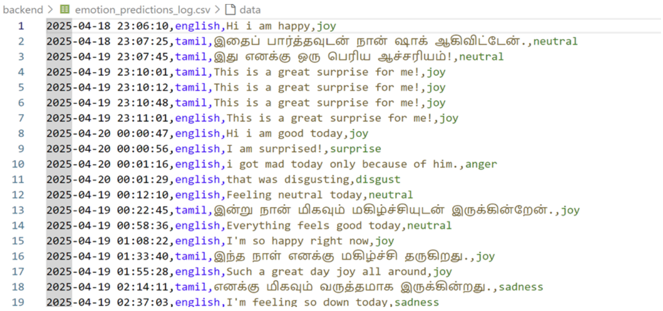
Fig. 15: Tamil Text Emotion Detection

B. Emotion Tracking and Trend Analysis

The Multi-Mode Bilingual Emotion Detection System incorporates emotion tracking and trend visualization to enhance emotional awareness and support analytical insights. Key features include:

1) **Logging Emotions in CSV File:** Every detected emotion—whether from facial expressions or text input—is timestamped and logged in a structured CSV file for

further analysis (Fig. 16). This logging mechanism enables persistent emotion history tracking, which can be useful for pattern recognition, mental wellness insights, or behavioural studies. The use of CSV format ensures compatibility with various data analysis tools and simplifies data manipulation for developers and researchers.



backend > emotion_predictions_log.csv > data

```
2025-04-18 23:06:10,english,Hi i am happy,joy
2025-04-18 23:07:25,tamil,இதைப் பார்த்தவுடன் நான் ஷாக் ஆகிவிட்டேன்.,neutral
2025-04-19 23:07:45,tamil,இது எனக்கு ஒரு பெரிய ஆச்சரியம்!,neutral
2025-04-19 23:10:01,tamil,This is a great surprise for me!,joy
2025-04-19 23:10:12,tamil,This is a great surprise for me!,joy
2025-04-19 23:10:48,tamil,This is a great surprise for me!,joy
2025-04-19 23:11:01,english,This is a great surprise for me!,joy
2025-04-20 00:00:47,english,Hi i am good today,joy
2025-04-20 00:00:56,english,I am surprised!,surprise
2025-04-20 00:01:16,english,i got mad today only because of him.,anger
2025-04-20 00:01:29,english,that was disgusting,disgust
2025-04-19 00:12:10,english,Feeling neutral today,neutral
2025-04-19 00:22:45,tamil,இன்று நான் மிகவும் மகிழ்ச்சியுடன் இருக்கின்றேன்.,joy
2025-04-19 00:58:36,english,Everything feels good today,neutral
2025-04-19 01:08:22,english,I'm so happy right now,joy
2025-04-19 01:33:40,tamil,இந்த நாள் எனக்கு மகிழ்ச்சி தருகிறது.,joy
2025-04-19 01:55:28,english,Such a great day joy all around,joy
2025-04-19 02:14:11,tamil,எனக்கு மிகவும் வருத்தமாக இருக்கின்றது.,sadness
2025-04-19 02:37:03,english,I'm feeling so down today,sadness
```

Fig. 16: Emotion Logging

2) **Graphical Visualization of Emotions Overtime:** The Trends page presents a time-based graphical representation of the logged emotions, offering a visual overview of mood fluctuations over different time intervals (Fig. 17). This helps users or caregivers identify recurring emotional states or abnormal emotional patterns. The graph dynamically updates with new entries, making it an effective tool for emotion trend monitoring in real time and across sessions.

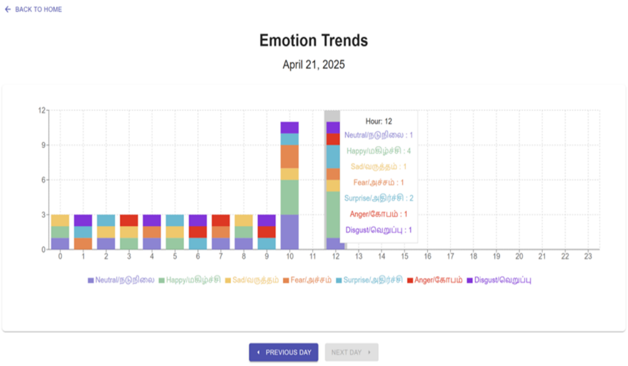


Fig. 17: Visualization of Emotional Trends Overtime

VI. Conclusion

The Multi-Mode Bilingual Emotion Detection System represents a significant advancement in emotionally intelligent technologies. By integrating deep learning models across facial images and bilingual text (Tamil and English), the system ensures accurate real-time emotion recognition. The system allows users to express emotions via their preferred communication mode, enhancing accessibility and performance. It is designed for seamless integration into both mobile and desktop applications, offering scalability and adaptability for diverse use cases such as mental wellness monitoring, emotionally adaptive learning platforms, and personalized digital assistants. This project sets a strong foundation for future emotionally aware systems that promote deeper human-computer understanding.

While the current implementation is effective, several areas for enhancement exist. Future work includes fusing visual and textual inputs for more accurate emotion detection and expanding multilingual capabilities to include more languages for greater inclusivity. Enhancing the CNN model with attention mechanisms and exploring advanced multilingual models like BERT or XLM-R could improve recognition accuracy. Additionally, incorporating online learning would allow dynamic adaptation to evolving user expressions. From a deployment perspective, containerizing the system and integrating privacy-preserving techniques will enable broader adoption in mobile platforms while ensuring ethical, secure emotion detection.

References

[1] J. Deng and F. Ren, ”Multi-Label Emotion Detection via Emotion-Specified Feature Extraction and Emotion Correlation Learning,” in IEEE Transactions on Affective Computing, vol. 14, no. 1, pp. 475-486, 1 Jan.-March 2023, doi: 10.1109/TAFFC.2020.3034215.

[2] J. Pan et al., ”ST-SCGNN: A Spatio-Temporal Self-Constructing Graph Neural Network for Cross-Subject EEG-Based Emotion Recognition and Consciousness Detection,” in IEEE Journal of Biomedical and Health Informatics, vol. 28, no. 2, pp. 777-788, Feb. 2024, doi: 10.1109/JBHI.2023.3335854.

[3] H. Kim and J. Ben-Othman, ”A Virtual Emotion Detection System With Maximum Cumulative Accuracy in Two-Way Enabled Multi Domain IoT Environment,” in IEEE Communications Letters, vol. 25, no. 6, pp. 2073-2076, June 2021, doi: 10.1109/LCOMM.2021.3060737.

[4] Y. -C. Wu, L. -W. Chiu, C. -C. Lai, B. -F. Wu and S. S. J. Lin, ”Recognizing, Fast and Slow: Complex Emotion Recognition With Facial Expression Detection and Remote Physiological Measurement,” in IEEE Transactions on Affective Computing, vol. 14, no. 4, pp. 3177-3190, 1 Oct.-Dec. 2023, doi: 10.1109/TAFFC.2023.3253859.

[5] A. Choudhry, I. Khatri, M. Jain and D. K. Vishwakarma, ”An Emotion-Aware Multitask Approach to Fake News and Rumor Detection Using Transfer Learning,” in IEEE Transactions on Computational Social Systems, vol. 11, no. 1, pp. 588-599, Feb. 2024, doi: 10.1109/TCSS.2022.3228312.

[6] Y. Zhou, X. Kang and F. Ren, ”Prompt Consistency for Multi-Label Textual Emotion Detection,” in IEEE Transactions on Affective Computing, vol. 15, no. 1, pp. 121-129, Jan.-March 2024, doi: 10.1109/TAFFC.2023.3254883.

[7] S. Sharma, R. S, M. S. Akhtar and T. Chakraborty, ”Emotion-Aware Multimodal Fusion for Meme Emotion Detection,” in IEEE Transactions on Affective Computing, vol. 15, no. 3, pp. 1800-1811, July-Sept. 2024, doi: 10.1109/TAFFC.2024.3378698.

[8] R. Mao, Q. Liu, K. He, W. Li and E. Cambria, ”The Biases of Pre-Trained Language Models: An Empirical Study on Prompt-Based Sentiment Analysis and Emotion Detection,” in IEEE Transactions on Affective Computing, vol. 14, no. 3, pp. 1743-1753, 1 July-Sept. 2023, doi: 10.1109/TAFFC.2022.3204972.

[9] Y. Yu et al., ”Cloud-Edge Collaborative Depression Detection Using Negative Emotion Recognition and Cross-Scale Facial Feature Analysis,” in IEEE Transactions on Industrial Informatics, vol. 19, no. 3, pp. 3088-3098, March 2023, doi: 10.1109/TII.2022.3163512.

[10] J. Tian and Y. She, ”A Visual–Audio-Based Emotion Recognition System Integrating Dimensional Analysis,” in IEEE Transactions on Computational Social Systems, vol. 10, no. 6, pp. 3273-3282, Dec. 2023, doi: 10.1109/TCSS.2022.3200060.

[11] Y. Zhang, Y. He, R. Chen, P. Tiwari, A. E. Saddik and M. S. Hossain, ”A Dual Channel Cyber–Physical Transportation Network for Detecting Traffic Incidents and Driver Emotion,” in IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 1766-1774, Feb. 2024, doi: 10.1109/TCE.2023.3325335.

[12] Y. Wang, H. Yu, W. Gao, Y. Xia and C. Nduka, ”MGEED: A Multimodal Genuine Emotion and Expression Detection Database,” in IEEE Transactions on Affective Computing, vol. 15, no. 2, pp. 606-619, April-June 2024, doi: 10.1109/TAFFC.2023.3286351.

[13] J. Yang, J. Li, X. Wang, Y. Ding and X. Gao, ”Stimuli-Aware Visual Emotion Analysis,” in IEEE Transactions on Image Processing, vol. 30, pp. 7432-7445, 2021, doi: 10.1109/TIP.2021.3106813.

[14] W. Zheng, L. Yan and F. -Y. Wang, ”Two Birds With One Stone: Knowledge-Embedded Temporal Convolutional Transformer for Depression Detection and Emotion Recognition,” in IEEE Transactions on Affective Computing, vol. 14, no. 4, pp. 2595-2613, 1 Oct.-Dec. 2023, doi: 10.1109/TAFFC.2023.3282704.

[15] P. Chiranjeevi, V. Gopalakrishnan and P. Moogi, ”Neutral Face Classification Using Personalized Appearance Models for Fast and Robust Emotion Detection,” in IEEE Transactions on Image Processing, vol. 24, no. 9, pp. 2701-2711, Sept. 2015, doi: 10.1109/TIP.2015.2421437.

[16] W. -J. Yoon and K. -S. Park, ”Building robust emotion recognition system on heterogeneous speech databases,” in IEEE Transactions on Consumer Electronics, vol. 57, no. 2, pp. 747-750, May 2011, doi: 10.1109/TCE.2011.5955217.

[17] F. Ren, X. Kang and C. Quan, ”Examining Accumulated Emotional Traits in Suicide Blogs With an Emotion Topic Model,” in IEEE Journal of Biomedical and Health Informatics, vol. 20, no. 5, pp. 1384-1396, Sept. 2016, doi: 10.1109/JBHI.2015.2459683.

[18] D. Sui et al., ”A Simple and Interactive Transformer for Fine-Grained Emotion Detection,” in IEEE Transactions on Audio, Speech and Language Processing, vol. 33, pp. 347-358, 2025, doi: 10.1109/TASLP.2024.3487418.

[19] H. A. Gonzalez et al., ”Hardware Acceleration of EEG-Based Emotion Classification Systems: A Comprehensive Survey,” in IEEE Transactions on Biomedical Circuits and Systems, vol. 15, no. 3, pp. 412-442, June 2021, doi: 10.1109/TBCAS.2021.3089132.

[20] A. Balahur, J. M. Hermida and A. Montoyo, ”Building and Exploiting EmotiNet, a Knowledge Base for Emotion Detection Based on the Appraisal Theory Model,” in IEEE Transactions on Affective Computing, vol. 3, no. 1, pp. 88-101, Jan.-March 2012, doi: 10.1109/T-AFFC.2011.33.

[21] Y. Gizatdinova and V. Surakka, ”Feature-based detection of facial landmarks from neutral and expressive facial images,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pp. 135-139, Jan. 2006, doi: 10.1109/TPAMI.2006.10.

[22] X. Zhang, W. Li, H. Ying, F. Li, S. Tang and S. Lu, ”Emotion Detection in Online Social Networks: A Multilabel Learning Approach,” in IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8133-8143, Sept. 2020, doi: 10.1109/JIOT.2020.3004376.

[23] M. Dwisnanto Putro, A. Priadana, D. -L. Nguyen and K. -H. Jo, ”EMOTIZER: A Multipose Facial Emotion Recognizer Using RGB Camera Sensor on Low-Cost Devices,” in IEEE Sensors Journal, vol. 25, no. 2, pp. 3708-3718, 15 Jan.15, 2025, doi: 10.1109/JSEN.2024.3493947.

[24] X. Yan, Z. Lin, Z. Lin and B. Vucetic, ”A Novel Exploitative and Explorative GWO-SVM Algorithm for Smart Emotion Recognition,” in IEEE Internet of Things Journal, vol. 10, no. 11, pp. 9999-10011, 1 June1, 2023, doi: 10.1109/JIOT.2023.3235356.