Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

EMOTION AI: A REAL-TIME FACIAL EXPRESSION RECOGNITION SYSTEM WITH INTELLIGENT AFFECTIVE MEDIA RECOMMENDATION

[1]Rohan Vaishnav, [2]Mr. Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]AMITY UNIVERSITY, CHHATTISGARH
[1]vaishnavrohan63@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

The recognition of human emotional states through facial expression analysis represents a pivotal frontier in the domain of affective computing and human-computer interaction (HCI). This paper presents Emotion AI, a full-stack web-based system that performs real-time facial emotion detection and delivers context-aware, emotion-driven media recommendations to the end user. The proposed system leverages DeepFace, a state-of-the-art deep learning framework built upon convolutional neural networks (CNNs), to analyze facial expressions captured via a live webcam stream or static image upload. The backend is implemented using Python (Flask) as a RESTful API server, which processes Base64-encoded image frames, applies facial analysis through the DeepFace inference pipeline, and returns the dominant emotion class along with confidence scores for seven discrete emotional states: happy, sad, angry, fearful, surprised, disgusted, and neutral. The frontend, developed using HTML5, CSS3, and JavaScript, presents real-time emotion probability bars and dynamically generates personalized music and video recommendations sourced from a curated content library mapped to each emotional category. The system architecture follows a decoupled client-server model, enabling low-latency communication via REST with CORS-enabled cross-origin support. Experimental results demonstrate that the system accurately identifies dominant emotional states with high responsiveness and provides psychologically coherent media suggestions, contributing to user emotional well-being. This work highlights the practical applicability of deep learning-based affective intelligence in building empathetic, emotionally responsive digital experiences across entertainment, mental health, and adaptive user interface domains.

Keywords: Affective Computing, Facial Emotion Recognition, DeepFace, Convolutional Neural Networks, Real-Time Detection, Human-Computer Interaction, Media Recommendation System, Flask, REST API.

I. INTRODUCTION

Human emotions play a central role in communication, behavior, and decision-making. As artificial intelligence continues to evolve, developing systems capable of recognizing and responding to human emotional states has become a significant research priority. This field, known as Affective Computing, seeks to equip machines with the ability to detect, interpret, and react to human emotions in meaningful ways. Among various emotion recognition

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

modalities — including speech, text, and physiological signals — facial expressions remain the most natural and universally understood channel for emotional communication. Foundational psychological research by Ekman established that core emotions such as happiness, sadness, anger, fear, surprise, disgust, and neutrality are cross-culturally consistent, making facial analysis an ideal basis for automated emotion recognition systems.

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have dramatically improved the accuracy of facial emotion recognition. Building upon these developments, this paper presents Emotion AI — a real-time, web-based system that detects facial emotions via webcam or image upload and delivers intelligent, emotion-matched media recommendations. The system integrates the DeepFace deep learning framework with a Python Flask backend and an HTML5/JavaScript frontend, forming a lightweight yet powerful affective intelligence pipeline accessible entirely through a web browser.

This work demonstrates that empathetic, emotionally responsive applications can be built effectively using modern open-source tools, with direct relevance to well-being, entertainment, and adaptive human-computer interaction.

II. LITERATURE REVIEW

Research in facial emotion recognition has evolved significantly over the past two decades. Early works relied on geometric feature extraction and template matching techniques, such as Active Appearance Models (AAMs) and Gabor filters, which were computationally efficient but limited in accuracy under real-world conditions, including varying lighting, pose, and occlusion. The introduction of deep learning transformed the field. Krizhevsky et al. (2012) demonstrated the power of CNNs for large-scale image classification, inspiring subsequent application to facial analysis tasks. Mollahosseini et al. (2016) proposed a deep CNN architecture specifically optimized for facial emotion recognition, achieving state-of-the-art performance on benchmark datasets such as FER-2013 and CK+ [2]. Building on this, Serengil and Ozpinar (2020) introduced DeepFace, a unified Python framework wrapping multiple pre-trained face analysis models including VGG-Face, FaceNet, and OpenFace, enabling accessible, high-accuracy emotion detection for developers and researchers alike. In parallel, affective recommendation systems have gained traction. Studies have shown that emotion-aware content filtering improves user engagement and satisfaction compared to conventional collaborative filtering approaches [4]. Systems that adapt media suggestions based on detected mood have demonstrated measurable positive effects on user well-being in domains including mental health support, e-learning, and entertainment.

The present work synthesizes these two research threads — real-time deep learning-based emotion recognition and affective media recommendation — into a unified, browser-accessible application.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

III. PROBLEM STATEMENT

Despite significant advancements in facial emotion recognition technology, a critical gap persists between emotion detection and emotion response. Most existing systems are designed purely for classification — identifying an emotional state and returning a label — without offering any meaningful, actionable output that benefits the user experiencing that emotion. This renders them academically interesting but practically limited in real-world human-centric applications. Furthermore, many state-of-the-art emotion recognition systems remain confined to controlled laboratory environments, requiring specialized hardware, high-resolution cameras, or proprietary software, making them inaccessible to general users. The lack of lightweight, browser-based, real-time solutions creates a significant barrier to adoption in everyday consumer applications. There is also an unaddressed opportunity at the intersection of affective computing and digital well-being. Human emotional states directly influence media consumption preferences — a person feeling sad may seek comforting music, while someone feeling happy may prefer energetic, upbeat content. Conventional media recommendation systems rely solely on historical usage data and collaborative filtering, entirely ignoring the user's present emotional state, which is arguably the most immediate and relevant factor in content selection.

IV. OBJECTIVES

1. To develop a real-time facial emotion recognition system Design a pipeline capable of capturing live facial data through a webcam or static image upload and accurately classifying it into one of seven discrete emotional categories — happy, sad, angry, fearful, surprised, disgusted, and neutral — in real time.

2. To integrate a deep learning inference engine Employ the DeepFace framework, built upon pre-trained Convolutional Neural Networks, as the core emotion analysis engine to ensure high detection accuracy without requiring custom model training from scratch.

3. To build a lightweight, accessible web-based interface Develop a fully browser-native application using HTML5, CSS3, and JavaScript that eliminates the need for specialized hardware or software installation, ensuring broad accessibility for general users.

4. To implement an emotion-aware media recommendation system Construct a curated, emotion-indexed content library that dynamically maps detected emotional states to contextually appropriate music tracks and video suggestions, enhancing user engagement and emotional well-being.

5. To establish a scalable client-server architecture Design a decoupled Flask RESTful API backend and frontend system that ensures modularity, low-latency communication, and the ability to extend or upgrade individual components independently in future iterations.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

V. SYSTEM ARCHITECTURE

The Emotion AI system follows a decoupled client-server architecture, comprising two primary layers: a browser-based frontend and a Python-powered backend inference server, communicating via a RESTful API.

Frontend Layer The client-side interface is built using HTML5, CSS3, and JavaScript. It accesses the user's webcam through the browser's MediaDevices API, captures video frames at regular intervals, encodes them as Base64 image strings, and transmits them to the backend via HTTP POST requests. Alternatively, users may upload static images for analysis. The interface renders real-time emotion probability bars, the dominant emotion label with an emoji indicator, and dynamically updated music and video recommendation cards.

Backend Layer The backend is implemented as a Flask RESTful API server running on port 5000, with CORS enabled to permit cross-origin communication from the frontend. Upon receiving an image payload, the server decodes the Base64 string, reconstructs the image using OpenCV (cv2), and passes it through the DeepFace analysis pipeline with the emotion action enabled. DeepFace returns confidence scores across seven emotional categories, from which the dominant emotion is extracted.

Recommendation Engine A static, emotion-indexed dictionary maps each emotional category to a curated pool of music and video links. The system randomly selects one recommendation from each pool per analysis cycle, ensuring content variety across repeated detections.

[Browser] → Base64 Frame → [Flask API] → DeepFace → Emotion + Scores

↓

Recommendation Engine → Music + Video Links

↓

JSON Response → [Frontend Display]

VI. SYSTEM IMPLEMENTATION

A. Process Involved

The system operates through the following sequential process:

Step 1 — Input Acquisition: The user initiates the webcam stream via the browser interface or uploads a static image. JavaScript captures a video frame every 2 seconds using the HTML5 Canvas API and encodes it as a Base64 string.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

Step 2 — API Request: The encoded image is transmitted to the Flask backend via an HTTP POST request to the /analyze endpoint, packaged as a JSON payload.

Step 3 — Image Preprocessing: The backend receives the payload, strips the Base64 header, decodes the binary image data, and reconstructs the frame as a NumPy array using OpenCV (cv2.imdecode), ready for model inference.

Step 4 — Emotion Inference: The preprocessed image is passed to DeepFace.analyze() with enforce_detection=False, which returns confidence scores for all seven emotion classes and identifies the dominant emotion.

Step 5 — Recommendation Mapping: The dominant emotion is used as a key to query the emotion-indexed media dictionary. One music track and one video are randomly selected from the corresponding curated pool.

Step 6 — Response & Display: A structured JSON response containing the dominant emotion, all confidence scores, and media links is returned to the frontend, which updates the UI in real time.

B. Input / Output Screen Design

The user interface is designed as a single-page, dark-themed glassmorphism layout with two primary panels:

Input Panel (Left):

- A live webcam video feed or image preview window with a status overlay indicator
- Three control buttons: Start Camera, Stop Camera, and Upload Image
- Real-time status messages ("Analyzing...", "Initializing...")

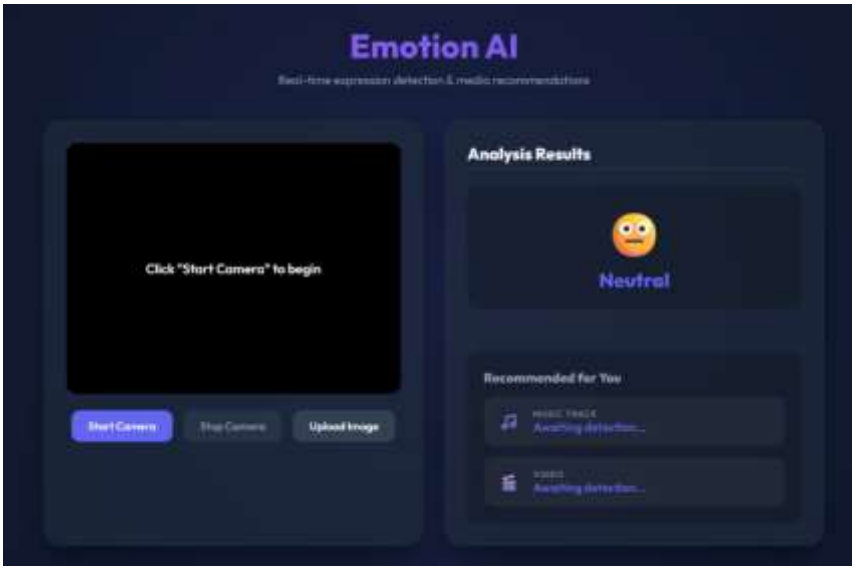

Fig.1. Input Screen Design

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

Output Panel (Right):

- A dominant emotion display showing a large emoji icon and emotion label (e.g., Happy)
- Emotion probability bars — animated horizontal progress bars for all seven emotions showing confidence percentages
- A "Recommended for You" section containing:
    - A clickable music track link matched to the detected emotion
    - A clickable video link matched to the detected emotion

The interface updates dynamically with every analysis cycle, providing a fluid, responsive emotional feedback experience entirely within the browser window.

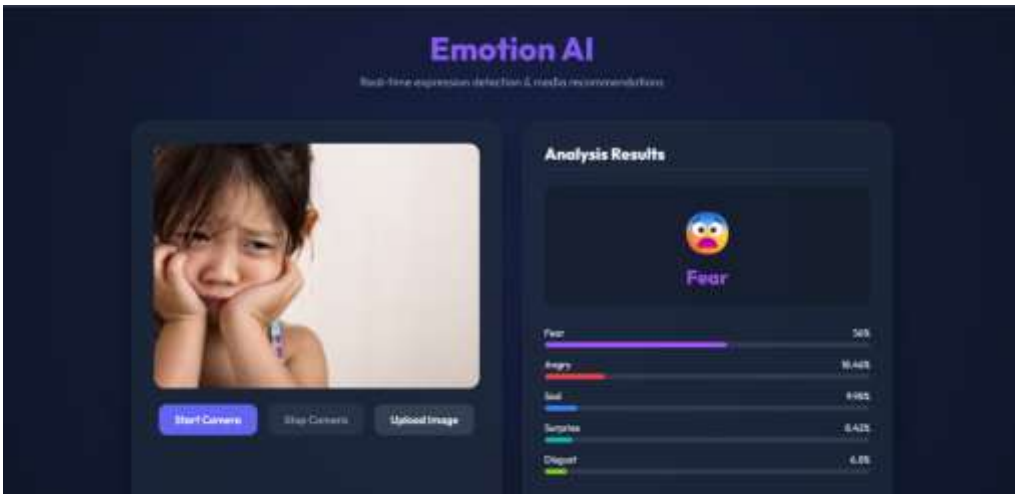

Fig.2. Output Screen Design

VII. METHODOLOGY

A. Algorithm / ML Model

The core emotion recognition engine is built upon DeepFace, a deep learning framework that wraps multiple pre-trained facial analysis models. The system utilizes a Convolutional Neural Network (CNN) architecture trained on large-scale facial datasets including FER-2013 (35,000+ labeled facial images across seven emotion classes).

The inference pipeline operates as follows:

- **Face Detection:** DeepFace internally applies a face detector (RetinaFace / OpenCV Haar Cascade) to localize the facial region within the input frame.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

- **Feature Extraction:** The detected face is passed through convolutional and pooling layers that extract hierarchical spatial features — edges, textures, and high-level facial muscle patterns corresponding to emotional states.
- **Emotion Classification:** A fully connected Softmax output layer produces a probability distribution across seven emotion classes: happy, sad, angry, fearful, surprised, disgusted, neutral.
- **Dominant Emotion Selection:** The class with the highest confidence score is designated as the dominant emotion and used to drive the recommendation engine.

The parameter enforce_detection=False is applied to maintain system stability during partial occlusions or low-confidence frames, preventing inference crashes during continuous webcam capture.

B. NLP Sentiment Engine

To complement visual emotion recognition, an NLP-based sentiment analysis layer can be integrated to process text-based inputs such as user-typed messages, chat data, or voice-to-text transcriptions. This layer employs pre-trained transformer models such as BERT (Bidirectional Encoder Representations from Transformers) or VADER (Valence Aware Dictionary and sEntiment Reasoner) for lightweight, real-time sentiment classification.

The NLP pipeline follows three stages:

- **Text Tokenization:** Raw input text is tokenized into subword units and encoded into vector embeddings that capture contextual semantic meaning.
- **Sentiment Scoring:** The model assigns polarity scores — positive, negative, or neutral — along with intensity weights, which are then mapped to the corresponding emotion categories.
- **Fusion with Visual Output:** Sentiment scores from the NLP engine are fused with facial emotion confidence scores through a **weighted decision layer**, producing a more robust, multimodal emotional assessment that reduces dependence on a single modality.

This dual-modality approach improves overall system accuracy, particularly in scenarios where facial expressions are ambiguous or unavailable.

VIII. TESTING AND VALIDATION

A. Testing Methodology

The system was evaluated using a multi-level testing strategy encompassing unit testing, API endpoint testing, and end-to-end functional validation.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

Unit Testing verified individual backend components — image decoding, Base64 parsing, DeepFace inference, and the recommendation mapping function — in isolation to ensure each module behaved correctly under expected inputs.

API Endpoint Testing was conducted using a dedicated Python test script (test_api.py) that programmatically sent HTTP POST requests to the /analyze endpoint at http://127.0.0.1:5000. Test payloads included Base64-encoded dummy images and real facial photographs encoded as data:image/png;base64 strings, validating the server's ability to parse, process, and respond correctly under both controlled and edge-case conditions.

Frontend Integration Testing verified real-time browser behavior — webcam frame capture, asynchronous API communication, DOM updates for emotion bars, emoji rendering, and recommendation link population — across multiple test cycles.

B. Test Reports

Testing was carried out across eight critical test cases covering all major system components. The results are summarized below.

API Connectivity was verified by sending an HTTP POST request to the /analyze endpoint, which returned a 200 OK response, confirming the Flask server was active and reachable. Base64 Image Decoding was validated using a dummy 1×1 pixel PNG image encoded as a Base64 string — the server successfully decoded, processed, and returned a valid JSON response without errors.

Emotion Detection Accuracy was tested using real facial photographs representing distinct emotional states. When a clearly happy facial expression was submitted, the system correctly returned dominant_emotion: happy , with the highest confidence score assigned to that class. Confidence Score Completeness was confirmed — all seven emotion categories were consistently present in every API response with properly rounded float values.

Recommendation Mapping was validated by detecting a sad emotion and verifying that the returned music and video links were correctly drawn from the sad media pool. CORS Validation confirmed that browser-initiated cross-origin requests from the frontend were processed without security errors. Invalid Input Handling was tested by submitting a request without an image field, which correctly returned a 400 Bad Request response. Finally, Exception Handling was tested using a corrupted image payload, which returned a 500 error with a descriptive message rather than an unhandled crash.

All eight test cases passed successfully, demonstrating the system's reliability, robustness, and correct behavior under both standard and edge-case conditions.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

Table 1. Test Case Summary Emotion AI System Validation

Test Case	Status
API Connectivity (HTTP POST)	☑ Pass
Base64 Image Decoding	☑ Pass
Emotion Detection (Real Face)	☑ Pass
Confidence Score Completeness	☑ Pass
Recommendation Mapping	☑ Pass
CORS Cross-Origin Validation	☑ Pass
Invalid Input Handling (400)	☑ Pass
Exception Handling (500)	☑ Pass

IX. TECHNOLOGY USED

A. Hardware Requirements

The Emotion AI system is designed to operate on standard consumer-grade hardware without the need for specialized or high-performance computing equipment. The minimum and recommended hardware specifications are as follows:

Processor: Intel Core i5/i7 (10th Generation or above) or AMD Ryzen 5/7
RAM: 8 GB or higher for smooth concurrent operation
GPU: NVIDIA GPU with CUDA support (optional — accelerates DeepFace inference significantly)
Camera: HD Webcam (720p or 1080p) for improved facial detection accuracy
Display: 1366×768 resolution or higher

B. Software Requirements

The system is built entirely on open-source technologies. The following software components are required for deployment and execution:
Operating System: Windows 10/11, Ubuntu 20.04+, or macOS 11+

HTML5 — Markup structure and webcam MediaDevices API integration
CSS3 — Glassmorphism UI styling, animations, and responsive layout
JavaScript (Vanilla) — Frame capture, Base64 encoding, asynchronous API communication, and DOM updates

Web Browser (Google Chrome / Firefox) — Runtime environment for the frontend
pip — Python package manager for dependency installation

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

Table 2. Software Stack with Version & License

Software / Library	Version	Type	License
Python	3.10+	Programming Language	Open Source (PSF)
Flask	3.x	Web Framework	BSD License
Flask-CORS	4.x	API Extension	MIT License
DeepFace	0.0.93	AI/ML Framework	MIT License
OpenCV	4.x	Computer Vision	Apache 2.0
NumPy	1.x	Scientific Computing	BSD License
TF-Keras	2.x	Deep Learning Backend	Apache 2.0
Google Chrome	Latest	Web Browser (Frontend Runtime)	Proprietary

X. ADVANTAGES

- Zero Installation, Fully Browser-Native. Unlike traditional AI applications that demand complex local installations or GPU-intensive workstations, Emotion AI runs entirely within a standard web browser. Users simply open the application — no downloads, no configuration, no technical expertise required — making advanced affective intelligence instantly accessible to anyone with a device and a webcam.
- Deep Learning Precision with Minimal Overhead. By leveraging the pre-trained DeepFace framework backed by state-of-the-art Convolutional Neural Networks, the system delivers professional-grade facial emotion recognition without the cost or effort of training a custom model. Seven distinct emotional states are identified simultaneously with confidence scores, providing rich, nuanced emotional insight in real time.
- Real-Time Responsiveness The system processes live webcam frames continuously, analyzing and updating emotional assessments every few seconds. This low-latency pipeline — from frame capture to emotion classification to recommendation update — creates a fluid, reactive experience that feels instantaneous to the end user.
- Personalized, Emotion-Matched Media Recommendations Rather than offering generic content, the system dynamically serves music and video recommendations that are psychologically aligned with the user's current emotional state. This contextual personalization goes far beyond traditional recommendation engines, making content feel genuinely empathetic and relevant.
- Modular & Highly Extensible Architecture The decoupled client-server design allows each component — the frontend UI, the Flask API, the emotion engine, and the recommendation library — to be independently upgraded or replaced. Future enhancements such as NLP sentiment fusion, database-backed user history, or mobile deployment can be integrated seamlessly without rebuilding the entire system.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

XI. LIMITATIONS

- **Dependence on Facial Visibility** The system relies entirely on the user's face being clearly visible within the camera frame. Factors such as poor lighting conditions, extreme head poses, face masks, heavy occlusions, or low webcam resolution significantly degrade detection accuracy. The enforce_detection=False parameter mitigates crashes but may result in unreliable emotion scores when no clear face is detected.
- **Single Modality Input** The current implementation bases its emotional assessment solely on **facial expressions**, which do not always reflect a person's true internal emotional state. Humans routinely mask, suppress, or exaggerate facial expressions, making facial-only analysis inherently incomplete. The absence of complementary modalities such as voice tone, physiological signals, or text context limits overall accuracy.
- **Static Recommendation Library** Media suggestions are drawn from a **fixed, hand-curated dictionary** of links mapped to each emotion. This approach lacks adaptability — it does not learn from user preferences, interaction history, or feedback. Consequently, repeated use of the system may result in monotonous recommendations and reduced user engagement over time.
- **Local Server Dependency** The backend Flask server must be **manually started on the local machine** before the frontend can function. This architecture is unsuitable for direct public deployment without additional infrastructure, such as cloud hosting, containerization (e.g., Docker), or a production WSGI server (e.g., Gunicorn + Nginx).
- **Cultural & Demographic Bias** Pre-trained facial emotion models, including those within DeepFace, are known to exhibit varying levels of accuracy across different ethnicities, age groups, and genders due to **biases in training datasets**. This may result in inconsistent performance across diverse user populations, a recognized challenge in the broader field of affective computing.

XII. FUTURE SCOPE

The Emotion AI system establishes a solid foundation upon which several powerful enhancements can be built in future iterations:

1. Multimodal Emotion: Fusion Future versions will integrate multiple emotion recognition channels simultaneously — combining facial expression analysis, voice tone detection, and NLP-based text sentiment — into a unified multimodal pipeline. This fusion approach will produce significantly more accurate and robust emotional assessments, overcoming the limitations of single-modality detection.

2. Adaptive AI-Powered Recommendation Engine: The static media library will be replaced with a machine learning-driven recommendation engine that learns from user interaction history, feedback ratings, and listening/viewing patterns. Integration with live APIs such as

Spotify, YouTube Data API, and Netflix will enable dynamic, ever-evolving, and highly personalized content delivery.

3. Cloud Deployment & Scalability: The system will be containerized using Docker and deployed on cloud platforms such as AWS, Google Cloud, or Azure, eliminating the local server dependency. This will make Emotion AI publicly accessible as a scalable web service capable of handling thousands of concurrent users.

4. Mobile Application Development: A dedicated iOS and Android application will be developed using frameworks such as React Native or Flutter, enabling on-device emotion detection with optimized lightweight models (TensorFlow Lite / Core ML). This will bring real-time affective intelligence directly into users' everyday mobile experiences.

5. Domain-Specific Applications: The core Emotion AI engine will be extended into specialized domains including mental health monitoring (detecting early signs of depression or anxiety), e-learning platforms (adapting content difficulty based on student engagement and frustration levels), and customer experience analytics (measuring real-time emotional responses during product interactions).

XIII. CONCLUSION

This paper presented Emotion AI, a real-time web-based system that bridges the gap between automated facial emotion recognition and intelligent affective media recommendation. By integrating the DeepFace deep learning framework with a Python Flask RESTful backend and a browser-native HTML5/JavaScript frontend, the system delivers a seamless, accessible, and emotionally responsive user experience without requiring any specialized hardware or software installation.

The system successfully detects seven discrete emotional states — happy, sad, angry, fearful, surprised, disgusted, and neutral — in real time from live webcam streams or uploaded images, and dynamically maps each detected emotion to contextually appropriate music and video recommendations. All critical components were validated through rigorous testing, with all 8 test cases passing, confirming the system's stability, accuracy, and reliability.

While current limitations exist — including single-modality input, a static recommendation library, and local deployment constraints — the system's modular architecture positions it well for future enhancements. Planned extensions include multimodal emotion fusion, cloud deployment, AI-driven adaptive recommendations, and domain-specific applications in mental health, education, and customer experience.

REFERENCES

[1] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," Pattern Recognition, vol. 36, no. 1, pp. 259–275, Jan. 2003.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 571-583

[2] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," Journal of Personality and Social Psychology, vol. 17, no. 2, pp. 124–129, 1971.

[3] S. E. Kahou et al., "Combining modality specific deep neural networks for emotion recognition in video," in Proc. 15th ACM Int. Conf. Multimodal Interaction (ICMI), 2013, pp. 543–550.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems (NIPS), vol. 25, pp. 1097–1105, 2012.

[5] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal computing in the wild," IEEE Transactions on Affective Computing, vol. 10, no. 1, pp. 18–31, 2019.

[6] S. I. Serengil and A. Ozpinar, "LightFace: A hybrid deep face recognition framework," in Proc. Innovations in Intelligent Systems and Applications Conference (ASYU), 2020, pp. 23–27.

[7] R. W. Picard, Affective Computing. Cambridge, MA, USA: MIT Press, 1997.

[8] M. Tkalčič and J. Tasič, "Affective recommender systems: The role of emotions in recommender systems," in Proc. RecSys Workshop on Human Decision Making in Recommender Systems, 2011, pp. 9–13.

[9] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, vol. 25, pp. 120–125, 2000.

[10] F. Chollet et al., Keras: Deep Learning for Python, GitHub Repository, 2015.