Drbrindhas, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Facial Emotion Recognition Based Smart Music Player

Drbrindhas,Ms. P.Abirami In, Mr. Ajay.R, Mr. Anbarasan.R,Mr.Rishihesh.M.M, Mr.Safwan.S,Mr.Sriram.V

Head of the Department,Computer Networking,PSG Polytechnic College,Coimbatore Lecturer,Computer Networking, PSG Polytechnic College, Coimbatore
3,4,5,6,7Students,Computer Networking,PSGPolytechnicCollege,Coimbatore

Abstract- This paper presents the design and implementation of a Smart Playlist Generator using Affective Computing — a real-time, AI-driven music recommendation system that personalizes playlists based on the user's emotional state. The system integrates three core components: (1) a Facial Emotion Recognition (FER) module built on OpenCV and Convolutional Neural Networks (CNNs) that classifies emotions in real time from webcam input, (2) a Natural Language Processing (NLP) module that supports Thanglish (Tamil- English transliterated) text commands for conversational interaction, and (3) a Spotify Web API integration that maps detected emotions to audio features such as valence, energy, and tempo to generate context-aware playlists. The system achieves an emotion recognition accuracy of 87– 90%, Thanglish command interpretation accuracy exceeding 90%, and a playlist-mood alignment rate of 85–90%, with an end-to-end latency of approximately 3 seconds. The architecture leverages HTML/CSS/JavaScript for the frontend, Node.js with Express for the backend, Firebase for data persistence, and Python-based AI modules for emotion and language processing. Experimental results confirm the viability of affective computing for dynamic, personalized music delivery, and the system demonstrates significant potential for next- generation human-computer interaction in multimedia platforms. Keywords: Affective Computing, Facial Emotion Recognition, Convolutional Neural Networks, Music Recommendation System, Natural Language Processing, Spotify Web API, Thanglish Processing, Human-Computer Interaction.

Keywords— Affective computing, smart playlist generator, facial emotion recognition (FER), convolutional neural networks (CNNs), OpenCV, natural language processing (NLP), Thanglish text processing, emotion-based recommendation, Spotify Web API, valence, energy, tempo, real-time AI system, personalized music recommendation, human-computer interaction, Node.js, Express, Firebase, Python AI modules, emotion detection accuracy, conversational interface, multimedia systems.

I. INTRODUCTION

The evolution of digital music consumption has progressed from passive analog formats to intelligent streaming platforms; yet the majority historical listening behavior for personalization. Traditional recommendation systems employ collaborative filtering, content-based filtering, and hybrid methods to suggest tracks, but these approaches operate in an emotional vacuum — they are incapable of perceiving or adapting to the user's immediate affective state. This gap motivates the development of an emotion-aware music player that dynamically adjusts its playlist based on real-time facial expression analysis Affective Computing, introduced by Picard [1], studies how machines can recognize, interpret, and simulate human emotions. Advances in computer vision and deep learning have demonstrated that facial muscle movements captured via Convolutional Neural Networks (CNNs) can accurately classify emotional states with high precision [2][3]. Simultaneously, the proliferation of music streaming APIs such as Spotify's Web API,

which exposes quantitative audio feature parameters including valence, energy, and tempo, provides a machine-readable bridge between emotion and music.

This paper describes the architecture, implementation, and evaluation of a Smart Playlist Generator that fuses FER, NLP-based chatbot interaction supporting Thanglish inputs, and Spotify API integration into a unified Progressive Web Application (PWA). The system operates end-to-end in approximately 3 seconds and achieves emotion classification accuracy of 87–90% under standard lighting conditions. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 describes the system architecture and design; Section 4 covers implementation details; Section 5 presents experimental results; and Section 6 concludes with future directions.

II. RELATED WORK

Emotion Recognition in Multimedia Systems

Early work on facial emotion recognition employed handcrafted features such as Local Binary Patterns (LBP) and Gabor wavelets. The seminal work of Viola and Jones [4] on Haar-cascade face detectors laid the foundation for real-time face detection in OpenCV. The introduction of the FER2013 dataset provided a standard benchmark for training deep learning models across seven emotion classes. Subsequent architectures including VGGNet [5] and lightweight models like MobileNetV2 have enabled high-accuracy, low-latency emotion classification suitable for edge and web deployment.

Music Recommendation Systems

Collaborative filtering, as implemented by platforms such as Pandora through its Music Genome Project, relies on human-annotated musical attributes but lacks real-time emotional responsiveness. Spotify employs a hybrid model combining collaborative filtering with NLP-derived metadata and audio feature analysis. Valence and energy parameters from the Spotify Web API have been shown to indirectly correspond to emotional arousal-valence dimensions described by Russell's Circumplex Model of Affect [6]. However, such systems infer mood from prior behavior rather than directly sensing the user's current state.

Affective Computing and NLP Integration

The integration of affective computing with conversational interfaces has been explored in healthcare, education, and customer service domains. The challenge of multilingual NLP — particularly for code-mixed or transliterated inputs such as Thanglish — remains an active research area. Models such as mBERT (Multilingual BERT) have shown promise for cross-lingual sentiment analysis [7]. The present work combines these research threads into a single, deployable system focused on music personalization.

III. SYSTEM ARCHITECTURE AND DESIGN

The proposed system follows a modular, layered architecture consisting of a frontend user interface, a Node.js backend server, Python-based AI modules, a Firebase cloud database, and the Spotify Web API. Figure 1 illustrates the high-level application flow.

Table 1. System Module Overview

Layer	Component	Technology
Frontend	User Interface, Chat Window	HTML5, CSS3, JavaScript
Backend	API Gateway, Business Logic	Node.js, Express.js
AI / ML	Emotion Detection, NLP	Python, OpenCV, TensorFlow/Keras
Database	User Data, Chat Logs	Firebase Firestore
External API	Music Catalog, Playback	Spotify Web API (OAuth 2.0)

Facial Emotion Recognition Module

The FER module captures 640×480 video frames from the user's webcam using OpenCV. Each frame is converted to grayscale and resized to 48×48 pixels for input to a CNN trained on the FER2013 dataset. The network architecture employs multiple convolutional and max-pooling layers followed by fully connected layers with a softmax output over seven emotion classes: Happy, Sad, Angry, Fear, Surprise, Disgust, and Neutral. For the deployed system, the active categories are reduced to five (Happy, Sad, Angry, Neutral, Relaxed) to improve practical usability. Each classification is accompanied by a confidence score; the system maps the highest-confidence emotion to a music genre category as described in Table 2.

Table 2. Emotion-to-Music Mapping

Detected Emotion	Music Genre		Valence (target)	Energy (target)
Happy	Pop / EDM		High (0.7–1.0)	High (0.7–1.0)
Sad	Acoustic / Lo-fi		Low (0.0–0.3)	Low (0.1–0.4)
Angry	Rock / Metal		Low (0.1–0.4)	High (0.7–1.0)
Neutral	Chill / Ambient		Medium (0.4–0.6)	Medium (0.3–0.6)
Relaxed	Jazz / Classical		Medium-High (0.5–0.8)	Low (0.1–0.4)

Thanglish NLP Processing Module

A significant portion of users in South India type Tamil words using the English alphabet — a phenomenon known as Thanglish. Conventional NLP pipelines fail to handle this transliterated input reliably. The NLP module in this system implements a custom preprocessing pipeline comprising: (1) text normalization and stop-word removal; (2) transliteration mapping using phonetic character tables and dictionary lookups; (3) language detection via the langdetect library; (4) sentiment and intent classification using a fine-tuned mBERT model. The pipeline converts inputs such as "ennaku sad songs venum" into a machine-interpretable intent: "user requests sad music."

Chatbot Interface

The chatbot provides text-based interaction with the music player, understanding commands such as play, pause, next, request mood-based playlists, and search for specific artists. It integrates with the emotion module so that responses are contextually adapted — for example, suggesting an energetic playlist upon detecting a happy expression. Chat logs are persisted in Firebase Firestore to maintain conversational continuity across sessions.

Spotify Web API Integration

Authentication is implemented using OAuth 2.0, which generates a user-specific access token upon login. The system queries the Spotify Recommendations Endpoint with seed genres derived from the detected emotion and target audio features (valence, energy, danceability, tempo). Retrieved tracks are displayed in the web player interface with album art, track names, and artist information. Playback control endpoints (play, pause, next, previous) are exposed through on-screen buttons and chatbot commands.

IV. IMPLEMENTATION

Frontend

The frontend is built using HTML5, CSS3, and vanilla JavaScript. The dashboard renders in real time using DOM manipulation and asynchronous fetch calls to

the Node.js API. The webcam feed is embedded in a <video> element, with frames extracted via <canvas> for transmission to the Python AI module. The chat interface is rendered in a scrollable panel alongside the music player controls.

Backend

The Node.js server exposes RESTful API endpoints for emotion detection, chatbot response generation, and Spotify integration. The Express.js framework manages routing and middleware. The server communicates with the Python FER service via a local HTTP call, with the emotion label and confidence score returned as a JSON payload. Spotify API tokens are refreshed automatically using the OAuth 2.0 refresh token mechanism.

AI Modules

The Python AI components are encapsulated as microservices. The FER service runs a trained CNN (based on MobileNetV2 fine-tuned on FER2013) and returns emotion predictions. The NLP service processes raw text input through the Thanglish normalization pipeline and outputs an intent label and detected sentiment. Both services communicate with the Node.js backend over local HTTP.

Database

Firebase Firestore is used as a NoSQL, document-oriented database. Collections are defined for: Users (authentication details, Spotify tokens), Emotion Logs (timestamp, detected emotion, confidence), Playlists (emotion label, track IDs, Spotify URIs), Chat Logs (user message, chatbot response, session ID), and Activity Logs (playback events, skips, likes).
Table 3. Per-Emotion Classification Accuracy

V. EXPERIMENTAL RESULTS AND EVALUATION

Emotion Detection Accuracy

The FER module was evaluated on 100 real- time webcam captures from diverse users under varied lighting conditions. Emotion categories tested included Happy, Sad, Neutral, Angry, and Relaxed. Under standard indoor lighting, the system achieved a classification accuracy of 87–90%. Performance degraded to approximately 72% in low-light conditions and 68% for faces with partial occlusion (e.g., glasses or hand-covering). Table 3 summarizes the per-emotion accuracy.

Table 3. Per-Emotion Classification Accuracy

Emotion	Accuracy (%)	Precision	Recall
Happy	93	0.94	0.92
Sad	88	0.87	0.89
Angry	85	0.84	0.86
Neutral	90	0.91	0.89
Relaxed	86	0.85	0.87
Average	**88.4**	**0.88**	**0.89**

NLP and Chatbot Performance

The NLP module was evaluated on a test set of 100 mixed-language (English and Thanglish) text inputs. Correct intent classification was achieved in over 90% of cases. The chatbot achieved a command success rate of 92% across tasks including play/pause commands, mood-based playlist requests, and track search queries. Minor errors were observed for highly ambiguous inputs with unconventional spelling or heavy code-mixing.

Music Recommendation Alignment

Playlist-mood alignment was assessed by comparing recommended track audio features (valence and energy retrieved from the Spotify Audio Features endpoint) against the target ranges defined in Table 2. Across 50 test sessions, 85–90%of recommended

tracks fell within the intended feature ranges. User satisfaction feedback from informal user testing further confirmed the perceived emotional appropriateness of the generated playlists.

System Performance

End-to-end latency — from webcam frame capture to playlist update on screen — averaged 2.8 seconds on a mid-range laptop (Intel Core i5, 8 GB RAM). User interface responsiveness for chatbot commands and playback controls averaged below 500 milliseconds. CPU utilization remained under 60% during continuous monitoring, with GPU acceleration (when available) reducing CNN inference time by approximately 40%.

Table 4. System Performance Summary

Metric	Result
Emotion Detection Accuracy	87–90% (standard conditions)
Thanglish NLP Accuracy	> 90%
Chatbot Command Success Rate	> 92%
Playlist-Mood Alignment Rate	85–90%
End-to-End Latency	~2.8 seconds
UI Response Time	< 500 ms

VI. DISCUSSION

The results demonstrate that combining facial emotion recognition with API-driven music recommendation produces a demonstrably responsive and personalized listening experience. The 87–90% emotion classification accuracy is consistent with state-of-the-art CNN-based FER systems evaluated under similar conditions. The Thanglish NLP module's 90%+ accuracy addresses a significant real-world gap in conversational AI systems designed for South Indian users.

Several limitations were identified. First, the FER module is sensitive to environmental lighting and facial occlusion — challenges common to all vision-based affective systems. Second, the system's emotion taxonomy is limited to five categories; finer-grained emotional states such as nostalgia or melancholy are not yet captured. Third, the system depends on the Spotify Web API, introducing a cloud dependency and rate- limiting constraints. Fourth, the current architecture is designed for single-user deployment; multi-user simultaneous access requires further backend scaling.

VII. CONCLUSION AND FUTURE WORK

This paper presented a Smart Playlist Generator that integrates real-time facial emotion recognition, Thanglish NLP processing, and Spotify Web API connectivity to deliver emotion- adaptive, personalized music recommendations. The system achieved strong accuracy and low- latency performance on standard hardware, validating affective computing as a practical foundation for next-generation music applications.

Future enhancements include: (1) multimodal emotion fusion combining facial, speech, and textual cues for improved robustness; (2) reinforcement learning for individualized playlist optimization over time; (3) offline edge deployment to eliminate cloud API dependency; (4) expanded language support encompassing additional Indian languages; and (5) collaborative playlist features supporting multi-user emotional contexts. This work lays a foundation for broader research in emotion-aware human-computer interaction and affective multimedia systems.

REFERENCES

1. R. W. Picard, Affective Computing. Cambridge, MA: MIT Press, 1997.
2. P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Proc. IEEE CVPR, pp. 511–518, 2001.
3. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proc. ICLR, 2015.
4. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.
5. J. A. Russell, "A Circumplex Model of Affect," Journal of Personality and Social Psychology, vol. 39, no. 6, pp. 1161–1178, 1980.
6. D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson, 2023.
7. Spotify Developer Documentation, "Web API Reference," https://developer.spotify.com/documentation/web-api/, 2025.
8. OpenCV Documentation, "Open Source Computer Vision Library," https://opencv.org/, 2025.