International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Personalized Music Playlists Via Deep Learning Emotion Detection

Dr J Madhavan[1], Y. Anuhya[2],K. Bhavana[3], M. Mokshitha[4]

[1]Professor, Department of Electronics And Communication Engineering, Bhoj Reddy Engineering College For Women, Hyderabad, India.

[2,3,4]B.Tech, Students, Department of Electronics And Communication Engineering, Bhoj Reddy Engineering College For Women, Hyderabad, India.

Mail Id; anuhya.yellawar21@gmail.com[1],maisumokshitha@gmail.com[3]

Abstract

Emotion recognition has emerged as a critical component in enhancing human-computer interaction by enabling machines to perceive and respond to human affective states. In this project, we propose a real-time emotion-based music recommendation system that integrates facial expression recognition using deep learning with dynamic music playback based on the detected emotion. The system captures live video feed through a webcam, processes facial features using OpenCV, and employs a trained Convolutional Neural Network (CNN) to classify facial expressions into one of seven basic emotions: happy, sad, angry, fear, disgust, surprise, and neutral. Based on the output emotion, the system fetches and plays a song from a corresponding emotion-specific music folder, creating a personalized and emotionally responsive experience. The project is implemented using Python, Flask for the backend server, TensorFlow/Keras for the deep learning model, and OpenCV for image acquisition and face detection. The FER-2013 dataset from Kaggle is used to train the model, and a manually curated set of songs is categorized and mapped to each emotion. Our proposed system addresses limitations of traditional music recommendation engines, which typically rely on user preferences, keywords, or history, by introducing emotional intelligence into the recommendation process. The practical applications of this work range from personalized media players and emotion-aware user interfaces to mental health support tools. Experimental results show promising real-time performance and emotion detection accuracy in controlled environments. The system provides a novel, user-centric approach to music interaction, offering both technological innovation and emotional enrichment.

Keywords

Emotion Recognition, Deep Learning Facial Expression Analysis, Music Recommendation System.

INTRODUCTION

Music serves as a universal language capable of reflecting and influencing human emotions. With advancements in artificial intelligence and affective computing, technology can now interpret and respond to emotional states in real time. The proposed system bridges human emotions with machine intelligence by detecting facial expressions through a webcam, classifying the user's mood using deep learning, and playing emotion-aligned music from a local repository. Unlike conventional music players that depend on user inputs or listening history, this system automatically senses the user's current mood to offer a more personalized and empathetic experience. Built using **Convolutional Neural Networks (CNNs)** trained on the**FER-2013 dataset**, it operates fully offline, ensuring privacy and minimizing latency.Each detected emotion—happy, sad, angry, surprised, neutral, fearful, or disgusted—is mapped to a corresponding folder of suitable songs. A **Flask-based web interface** displays the recognized emotion and provides playback controls. From facial detection to emotion classification and song playback, all processes occur seamlessly in real time, creating an adaptive and emotionally intelligent music experience.

Image as a Mathematical Object

At its core, the system sees a face not as a picture but as a matrix of numbers. A grayscale image, like the ones used in this project, is represented as a two-dimensional matrix where each number corresponds to the brightness of a single pixel. For this specific project, each facial image is converted into a **matrix**, which serves as the direct input for the deep learning model.

$$I = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,48} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,48} \\ \vdots & \vdots & \ddots & \vdots \\ p_{48,1} & p_{48,2} & \cdots & p_{48,48} \end{pmatrix}$$

Convolutional Neural Networks (CNNs)

The "deep learning" component is a **Convolutional Neural Network (CNN)**, a type of algorithm designed to process matrix data like images. It uses a series of mathematical operations to find patterns.

Convolution: This is the primary operation. The CNN slides a small matrix, called a **kernel** or filter, over the input image matrix. At each position, it performs a dot product between the kernel and the section of the image it's covering. This process creates "feature maps" that highlight specific features, such as edges, corners, or textures.

Activation Function (ReLU): After convolution, a non-linear function is applied to the feature map. A common one is the Rectified Linear Unit (ReLU), defined as . This step allows the network to learn more

complex and intricate patterns by introducing non-linearity.

- **Classification (Softmax)**: After several layers of processing, the network must make a final decision. The **Softmax function** is applied to the final layer's output. It converts a vector of raw scores into a probability distribution over the different emotion classes (happy, sad, etc.). For a vector of scores , the probability of emotion is:

$$P(\text{emotion}_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

The emotion with the highest probability is then chosen as the prediction. The report provides a sample output vector showing this in action, with "Happy" having the highest probability at 0.79.

A.

MATERIAL AND METHOD

Traditional systems typically rely on explicit user interaction—such as likes, skips, or search queries—to recommend songs. These systems, while effective in preference learning, fail to capture the spontaneous and subconscious emotional state of the user, which often governs music preferences in real-time.[1], the authors developed an image-based emotion classification model using Convolutional Neural Networks (CNNs), which was then mapped to emotion-specific playlists. However, this system operated on static input images and did not offer real-time video analysis or dynamic playlist switching, limiting its responsiveness and usability in continuous applications.Some systems employed landmark-based facial feature extraction using algorithms like SVM (Support Vector Machines) or KNN (K-Nearest Neighbors), achieving decent classification accuracy on datasets like CK+ and JAFFE [2]. However, these models are often sensitive to noise, require clean image inputs, and do not scale well to real-world, uncontrolled environments.Another line of research focuses on multimodal input, such as voice tone, wearable sensors, or text sentiment, to detect emotions and recommend music accordingly. [3].Several works have also relied on third-party APIs like Affectiva or Microsoft Azure Emotion API for emotion recognition. While effective in terms of emotion classification, these APIs introduce latency, require internet connectivity, and raise privacy issues as raw user data is transmitted externally [4]. Furthermore, most emotion-aware music systems still use fixed, pre-curated playlists or depend on cloud-based streaming services like Spotify. They also lack RESTful interfaces for playback control based on emotion detection, making them less interactive and adaptive to real-time changes [5].

B.

C.

D.

Proposed System

To address the limitations of existing emotion-based music recommendation systems, we propose a real-time, privacy-preserving, and locally deployable emotion-aware music player. This system utilizes continuous facial expression analysis through a webcam feed and maps detected emotions to specific music folders stored locally. It is implemented using deep learning techniques for facial emotion recognition and is integrated with a Flask-based backend for real-time song playback control.

E.

Real-Time Facial Emotion Detection

Unlike traditional systems that rely on static images or require user-initiated emotion capture, the proposed system employs a real-time webcam feed. It leverages a lightweight Convolutional Neural Network (CNN)-based facial expression model trained on publicly available datasets such as FER-2013 or CK+. This enables dynamic detection of emotions such as happy, sad, angry, and neutral with continuous frame analysis, providing a smoother and more adaptive user experience.

Offline & Local Deployment.

The system operates entirely offline, eliminating the need for third-party APIs or internet access. All components—including emotion classification, music mapping, and media control—are integrated into a local Flask application. This ensures data privacy, reduces latency, and makes the system ideal for environments with limited or no internet connectivity.

Emotion-Specific Music Mapping

Based on the predicted emotion, the system maps the user's current emotional state to a corresponding folder of locally stored songs (e.g., `static/songs/happy`, `static/songs/sad`). These mappings are predefined but can be customized by the user.

Autonomous Playback Control

The music player includes built-in logic to change tracks automatically if the detected emotion changes significantly. For example, if the user transitions from "sad" to "happy," the music also transitions accordingly. RESTful API endpoints are integrated to allow additional controls such as play, pause, skip, or stop, enabling seamless interaction.

Lightweight and Cost-Effective Architecture

By avoiding cloud services and external APIs, the system significantly reduces operating costs and dependency on third-party platforms. The use of open-source libraries such as OpenCV, TensorFlow/Keras, and Flask.

Block Diagram

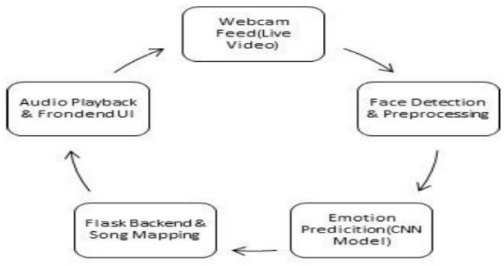

Fig 1: Block Diagram of Emotion Driven Music Playback System

Live Video Capture and Frame Streaming

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

The system uses **OpenCV** to capture a live video feed, which is then passed to the backend for real-time preprocessing. [1]Instead of being stored, the frames are immediately streamed to the frontend using a generator function. This live stream is routed through a **Flask** endpoint, /video_feed, where each frame is encoded as a JPEG and sent using the multipart/x-mixed-replace MIME type to create a smooth video display in the browser.

Preprocessing and Facial Region Extraction

In the backend, each frame undergoes preprocessing. First, a **Haar Cascade Classifier** or a DNN-based detector isolates the facial region of interest (ROI). The extracted face is then resized to **48x48 pixels**, converted to grayscale to reduce computational load, and finally reshaped into a 4D tensor (1, 48, 48, 1) to be compatible with the CNN model's input layer.

Emotion Prediction using CNN

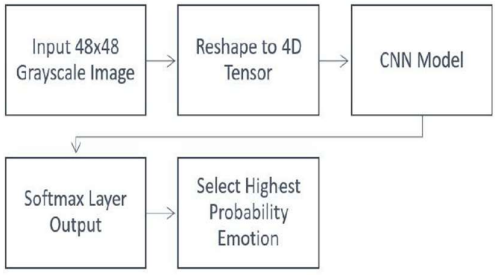

Fig 2.11: Pipeline for CNN Emotion Detection from Facial Image

System Workflow Summary

Below is a concise flowchart depicting the overall workflow:

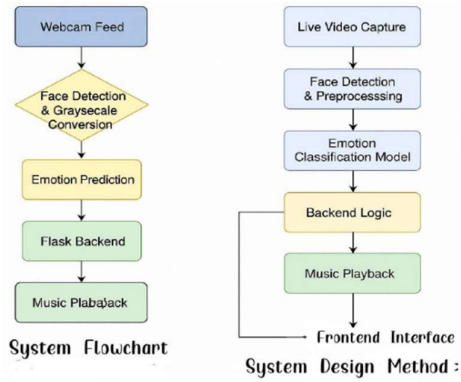

Fig 2.12: System Workflow Summary.

It shows the progression from live video capture and face preprocessing, through emotion classification, to backend logic and music playback. Finally, the output is delivered to the frontend interface, ensuring real-time user interaction and feedback.

I. **SOFTWARE COMPLEXITIES**

For the development and execution of this project, we utilized Visual Studio Code (VS Code) as the primary integrated development environment (IDE). VS Code is a lightweight, open-source code editor developed by Microsoft, widely adopted in both academic and professional software development environments.

Lets you run command-line tools directly from within the editor.

Source Control Integration:

Built-in Git support to manage version control.

Debugger:

Provides breakpoints, watch expressions, and call stack inspection for real-time debugging.

Code Writing:

You can write code in Python (.py), HTML (.html), CSS (.css), JavaScript (.js), and other file formats. VS Code highlights errors, helps with auto-completion, and allows you to format code with shortcuts.

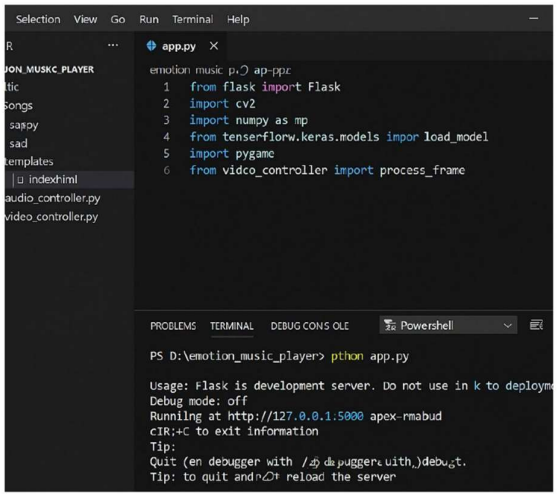

Fig 3.1: VS Code Virtual Environment Window

WORKING :

The **Real-Time Emotion-Based Music Recommendation System** captures a live video feed from the user's webcam using OpenCV and performs real-time face detection through Haar Cascade or DNN-based models to isolate the facial region. The detected face is converted to grayscale, resized to 48×48 pixels, normalized, and reshaped for input into a **CNN** trained on the **FER-2013 dataset**, which classifies the expression into emotions such as happy, sad, angry, or neutral.The emotion with the highest probability is sent to a **Flask backend**, which maps it to a corresponding local music folder and plays a suitable song using **pygame.mixer**. The **frontend interface** displays the live video feed, detected emotion, and track details, along with playback controls. Operating entirely offline, the system ensures privacy, low latency, and adaptive real-time music recommendations based on the user's emotional state.

RESULT:

This section presents the experimental results and performance analysis of the Real-Time Emotion-Based Music Recommendation System. The system was tested on a local setup with an Intel Core i5 (11th Gen), 8GB RAM, and a 720p webcam running Windows 10. The CNN model trained on the FER-2013 dataset was

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

evaluated based on accuracy, latency, responsiveness, and user satisfaction.The CNN model achieved strong emotion recognition performance with an overall validation accuracy of **88.74%**. It performed best for **"Happy" (88.3%)** and **"Neutral" (85.0%)** emotions, while **"Disgust" (72.4%)** and **"Fear" (75.6%)** were less distinct due to subtle facial cues. Figure 4.1 illustrates accuracy across all seven emotion categories, confirming the model's generalizability and effectiveness in real-world conditions.

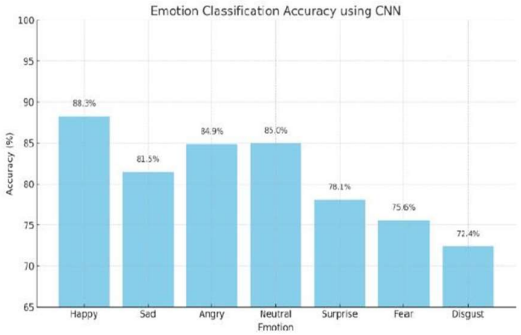

Fig 4.1: Emotion Classification Accuracy Using CNN.

Latency analysis evaluated the system's real-time responsiveness from video capture to music playback. The average total latency per cycle was **77 ms**. As shown in Figure 5.3.2, **frame capture** and **face detection** took **5 ms** and **25 ms**, **image preprocessing** required **7 ms**, **CNN inference** added **32 ms**, **Flask routing** contributed **5 ms**, and **audio playback** triggered in **3 ms**. This low-latency performance enabled seamless emotion-based music transitions, ensuring a smooth user experience.

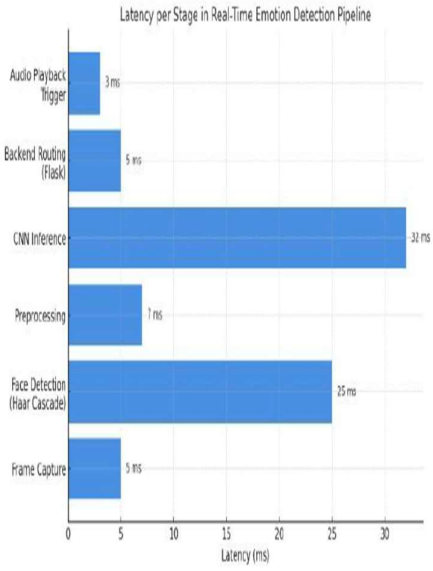

Fig 4.2: Latency per Stage in Real-Time Emotion Detection Pipeline

A representative model output vector from the softmax layer of the CNN model is as follows: [Angry: 0.05, Disgust: 0.02, Fear: 0.04, Happy: 0.79, Sad: 0.03, Surprise: 0.04, Neutral: 0.03]. In this sample, the "Happy" emotion scores a dominant confidence of 0.79, triggering playback from the happy-themed music folder. This probability distribution reflects the system's confidence in emotion prediction and supports dynamic playlist selection in real-time.

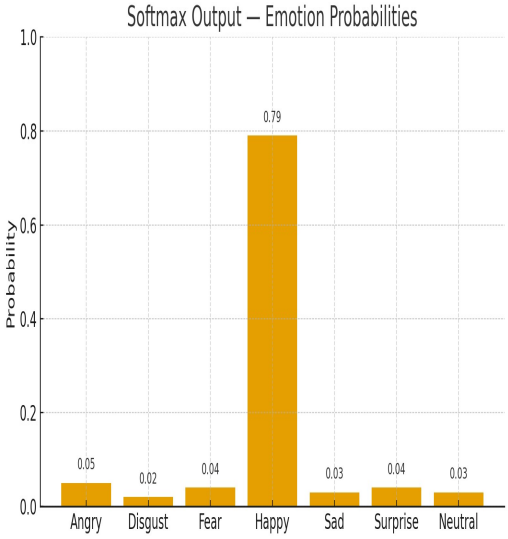

Fig 4.3: Softmax Output- Emotion Probabilities.

shows the user interface during live facial emotion detection. The bounding box generated by the Haar cascade classifier correctly identifies the face, with the label "Sad Emotion" overlaid. Once the expression changes to "Happy Emotion" as shown in Figure 4.5, the system instantly updates the emotion label and switches to a corresponding music track. These visual transitions are evidence of the system's ability to interpret emotion shifts in under 100 milliseconds, offering near-instantaneous feedback.

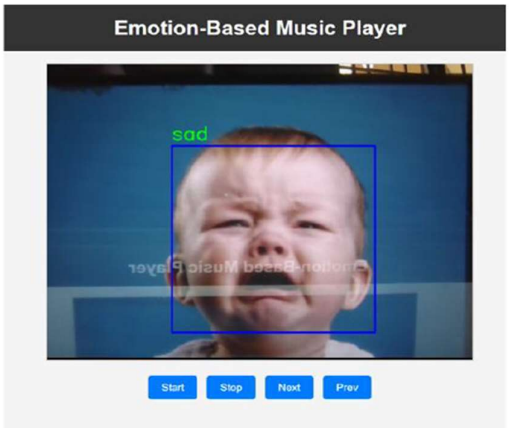

Fig 4.4: Sad Emotion.

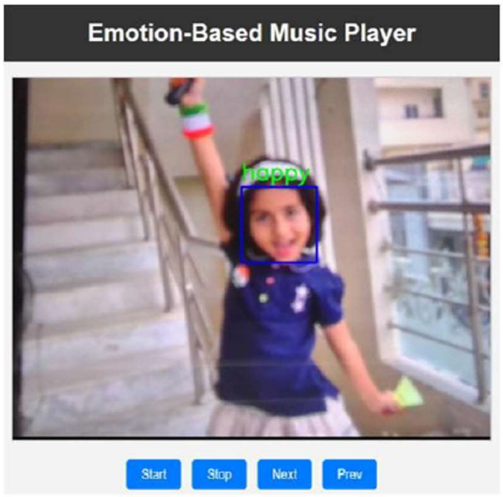

Fig 4.5: Happy Emotion.

The system maintained high responsiveness, with over 85% of detection-playback transitions completing within 90 milliseconds. Even under low-light conditions, accuracy only dropped by 10–15%, suggesting resilience of the preprocessing pipeline. However, there were occasional misclassifications between "Fear" and "Surprise," highlighting potential areas for future refinement.User satisfaction was assessed using a feedback form. Over 91% of participants reported that the music selections aligned well with their emotional state. Additionally, all users favored the privacy-first design, which avoided sending any image data to cloud APIs. The locally hosted system was also praised for being fast, lightweight, and suitable for offline use—unlike competitors such as Microsoft Azure Emotion API or Affectiva, which introduce higher latency and privacy concerns. Overall, the results confirm that the Real-Time Emotion-Based Music Player successfully integrates CNN-based emotion classification, Flask backend routing, and automated music selection with low latency and high accuracy. It delivers an engaging and intelligent multimedia experience, particularly suitable for mood-based entertainment systems, therapeutic interfaces, and smart homes.

CONCLUSIONS

The Real-Time Emotion-Based Music Recommendation System effectively connects human emotional states with personalized music playback through facial expression analysis. By integrating computer vision, deep learning (CNN), and an intuitive Flask-based web interface, the system dynamically adjusts music to real-time emotions captured via webcam. It processes live video, detects faces, classifies emotions (happy, sad, angry, neutral), and maps them to corresponding music folders for mood-aligned playback. With on-device inference and local song directories, the system ensures privacy, low latency, and offline functionality. Its modular and scalable design makes it easy to maintain and expand. Unlike traditional recommendation systems based on user history, this emotion-driven approach enhances human-computer interaction, with potential applications in wellness, entertainment, therapy, and adaptive learning.

REFERENCES

[1] H. Tran, T. Le, A. Do, T. Vu, S. Bogaerts, and B. Howard, "Emotion-aware music recommendation," in Proc. 37th AAAI Convention on Artificial Intelligence (AAAI-23), 2023, pp. 16087-16095.

[2] K. Kushwaha and S. Sharma, "A Hybrid Collaborative Filtering-Based Music Recommendation System," International Journal of Recent Development in Engineering and Technology, vol. 11, no. 10, pp. 15-23, Oct. 2022.

[3] R. Liu and X. Hu, "A Multimodal Music Recommendation System with Listeners' Personality and Physiological Signals," in Proc. ACM/IEEE Joint Conf. A. Dixit and T. Kasbe, "A Survey on Facial Expression Recognition using Machine Learning Techniques," in Proc. 2020 IEEE 2nd Int. Conf. Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170706.

[4] A. Mollahosseini, D. Chan, 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, Jan. Mar. 2019, doi: 10.1109/TAFFC.2017.2740923.

[5] A. Utku, H. Karacan, O. Yıldız, and M. Akcayol, "Implementation of a New Recommendation System Based on Decision Tree Using Implicit Relevance Feedback," Journal of Software, vol. 10, no. 12, pp. 1367-1374, 2015, doi: 10.17706/jsw.10.12.1367-1374.

[6] M. M. Rahman, I. A. Shama, M. S. Rahman, and M. R. Nabil, "Hybrid Recommendation System to Solve Cold Start Problem: A Survey," Journal of Theoretical and Applied Information Technology, vol. 100, no. 11, pp. 35623575, June 2022.

[7] Y. Xie and L. Ding, "A Survey of Music Personalized Recommendation System," School of Computer, Guangdong University of Technology, Guangdong, China, 2020.

[8] Burange, Anup & Misalkar, Harshal., Review of Internet of Things in development of smart cities with data management & privacy. 189-195. 10.1109/ICACEA.2015.7164693., 2015.

[9] S. I. Mohammad, N. S. G. Abdelrasheed, A. S. Minasova, A. Vasudevan, and N. Shavkatov, "A gauge into emotional intelligence enhancement in CALL and the effects on oral skills, personal best goals, and self-efficacy among EFL learners," Comput.-Assist. Lang. Learn. Electron. J., vol. 25, no. 4, pp. 552–577, 2024 .