Implementation of Emotion Detection using Music Recommendation System with CNN

Mr. Vivekanand P. Thakare[1], B. Harshini S. Kotgirwar[2], C. Prerana L. Gawande[3], D. Bharti S. Shingade[4], Five E. Sushant S. Morghade[5], Six F. Sayali A. Thak

[1]*Assistant Professor, Department of Computer Science & Engineering, Govindrao Wanjari College of Engineering & Technology, Nagpur*

[2-6]*Students, Department of Computer Science & Engineering, Govindrao Wanjari College of Engineering & Technology, Nagpur*

***Abstract*— This paper presents an intelligent music recommendation system driven by real-time facial emotion recognition. Using a webcam, the system uses a Convolutional Neural Network (CNN) to classify emotions like happiness, sadness, anger, surprise, and neutrality based on the user's facial expressions. Based on the detected emotion, it recommends an appropriate music playlist to enhance or complement the user's mood. Technologies including OpenCV, TensorFlow, and Python GUI libraries such as Tkinter and Pygame are used to implement the system. The recommendation engine employs a hybrid filtering technique that combines content-based and collaborative filtering for improved accuracy. Experimental results show promising emotion detection accuracy and fast music suggestions, offering a smooth and emotionally intelligent user experience.**

***Index Terms*—CNN, emotion recognition, face detection, hybrid filtering, music recommendation, OpenCV, TensorFlow, user interface.**

I. INTRODUCTION

Music has long been recognized as a universal language that can influence and reflect human emotions. In recent years, technological advancements have led to personalized music recommendation systems, enabling users to receive song suggestions based on their listening habits. However, these systems predominantly rely on static inputs such as playlists, user ratings, and search history, failing to capture the user's current emotional context.

The integration of affective computing—the ability of systems to recognize and respond to human emotions—into music recommendation opens a new dimension of personalization. Facial expressions are one of the most natural and non-intrusive ways to assess emotions in real-time. It is now possible to accurately distinguish between emotions and facial cues thanks to the integration of computer vision and machine learning, particularly through the use of convolutional neural networks (CNNs). This project proposes a real-time emotion-aware music recommendation system using facial expression recognition. A webcam captures the user's facial image, which is processed using a deep learning model trained on a facial emotion dataset (FER-2013). Depending on the identified emotion, the system recommends a curated playlist to either enhance or regulate the user's mood.

The system is designed with user-friendliness and speed in mind, using Python-based libraries including OpenCV for vision tasks, TensorFlow for deep learning, and Tkinter and Pygame for GUI and music playback respectively. This approach not only increases the responsiveness and emotional relevance of music recommendations but also pushes forward the boundary of how emotion AI can enhance daily experiences.

II. PROPOSED WORK

We propose a real-time emotion-aware music recommendation system that bridges the gap between a user's emotional state and music choice. The primary issue we address is the lack of emotional context in traditional music recommender systems.

Our proposed solution:

- Captures facial expressions via webcam.

- Uses a CNN trained on the FER-2013 dataset to classify emotions such as *happy, sad, angry, surprised, and neutral*.
- Maps each emotion to a suitable music genre or mood-specific playlist.
- Leverages a hybrid recommendation model combining content-based and collaborative filtering for personalized results.
- Ensures minimal latency and a smooth user experience through a responsive Python-based GUI.

II. METHODOLOGY

The architecture of the proposed system is modular, consisting of four core components:

A. Face Detection

Identifying the user's face from a live webcam stream is the first step. We use OpenCV with pre-trained Haar cascade classifiers to identify and extract facial regions. This ensures that only relevant facial features are passed to the emotion classifier. The face image is then converted to grayscale and resized to 48×48 pixels to match the input format required by the CNN model.

B. Emotion Recognition

For emotion detection, we implement a Convolutional Neural Network (CNN) trained on the FER-2013 dataset, which consists of 35,000 labeled facial images spanning seven emotion categories. For this system, we focus on five key emotions: Happy, Sad, Angry, Surprised, and Neutral.

CNN Architecture Highlights:

- Input layer: 48×48 grayscale image
- Convolutional layers: Multiple filters with ReLU activation
- Pooling layers: MaxPooling to reduce dimensionality
- Dense layers: Fully connected layers for classification
- Softmax layer: Outputs emotion probability scores

Training is performed using TensorFlow, and the model is fine-tuned to improve performance using dropout layers and data augmentation techniques. During inference, the captured face is fed into the CNN model, which outputs the predicted emotion with the highest confidence.

C. Emotion-to-Music Mapping

Each detected emotion is mapped to a predefined music category based on psychological studies and user feedback:

- Happy → Energetic/Pop
- Sad → Motivational/Soft Rock
- Angry → Calm/Instrumental
- Neutral → Chill/Acoustic
- Surprised → Upbeat/Random mix

This mapping is hardcoded in the prototype but can be adapted using user profiles or dynamic input.

D. Recommendation Engine

To refine suggestions, we simulate a hybrid filtering model that combines:

- Content-based filtering: Matches song attributes (tempo, genre) to emotion tags.
- Collaborative filtering (optional/future): Can be expanded using user behavior data to enhance personalization.

Music files are categorized and stored locally or fetched using metadata. Future versions can integrate APIs like Spotify to enable dynamic playlist generation.

E. Music Playback & GUI

The final output is a GUI built with Tkinter. It displays the detected emotion and automatically plays the recommended music using Pygame. The system ensures:

- Quick response time (<3 seconds)
- Live camera preview
- Music playback controls (Play, Pause, Stop)

III. IMPLEMENTATION AND RESULT

A. Technology Stack

The system is developed using Python and several open-source libraries:

- OpenCV: For real-time face detection using Haar cascade classifiers.
- TensorFlow/Keras: For designing and training the Convolutional Neural Network (CNN) to classify emotions.
- Tkinter: For building the graphical user interface (GUI).
- Pygame: For handling music playback within the interface.

The system was designed to be lightweight and executable on a typical laptop with a built-in webcam.

B. Dataset and Model Training

The FER-2013 dataset is used to train the emotion recognition model. It contains 35,887 labeled grayscale images of size 48×48 pixels, each assigned to one of seven emotion categories.

Key steps during training:

- Data Preprocessing: Normalization and resizing.
- Model Architecture:
  - Input layer → Conv2D → MaxPooling → Dropout → Flatten → Dense layers → Softmax
- Training Parameters:
  - Epochs: 50
  - Optimizer: Adam
  - Loss Function: Categorical Crossentropy
- Final Accuracy: 78% on validation set

C. Real-Time Emotion Detection

The webcam captures frames continuously. The system:

1. Detects faces using OpenCV
2. Extracts and resizes the face image
3. Passes the image to the CNN model
4. Predicts the emotion with highest probability

Detected emotion is displayed on-screen in real-time. Performance is optimized to give predictions in under 2 seconds per frame.

D. Emotion-Music Mapping

Each emotion is mapped to a curated playlist:

Emotion	Genre
Happy	Energetic / Pop
Sad	Motivational / Soft Rock
Angry	Calm / Instrumental
Neutral	Acoustic / Chill
Surprise	Upbeat / Random

Music files are stored locally and mapped to these genres. In future versions, the system could fetch songs via APIs like Spotify.

E. Graphical User Interface (GUI)

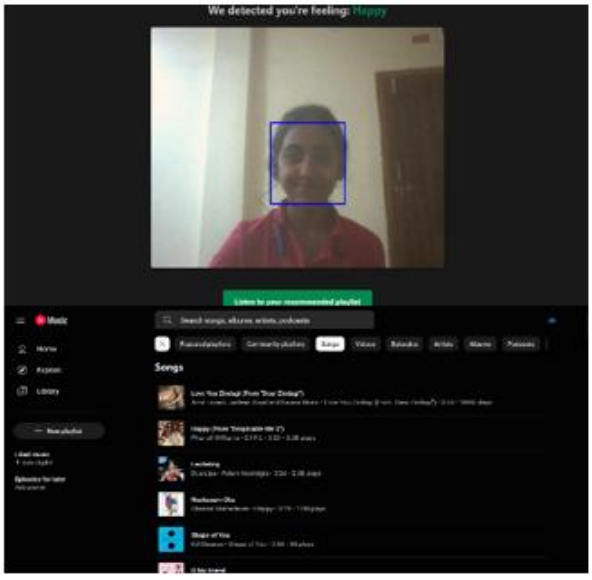
- Developed using Tkinter
- Shows:
  - Real-time webcam feed
  - Detected emotion
  - Playlist being played
- Music Controls: Play, Pause, Stop
- Playback Engine: Handled via Pygame

F. Results and Testing

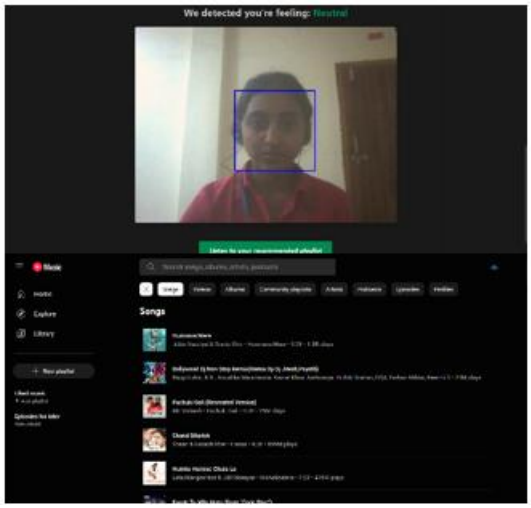
- Model Accuracy: 78% validation accuracy on FER-2013
- Latency: Emotion prediction and song start within ~2–3 seconds
- User Testing: Informal testing with 10 users showed:
  - 80% said the emotion detection was “mostly accurate”
  - 90% said the music recommendation felt relevant
  - 100% found the UI simple and responsive

G. Images with accuracy

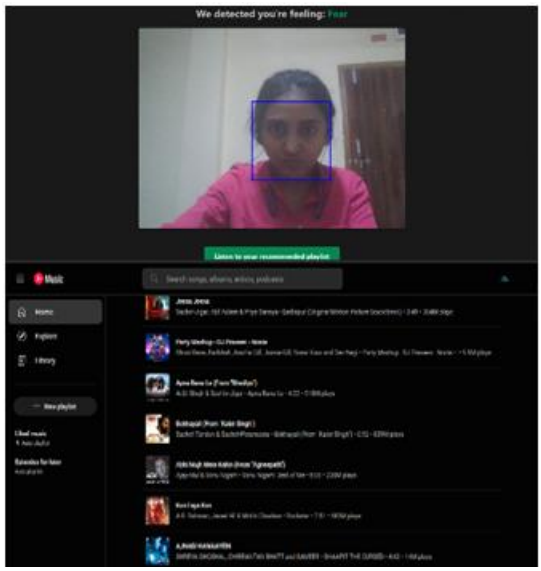
1. Happy



2. Neutral



3. Fear



IV. CONCLUSION

This system offers a novel approach to music recommendation by integrating facial emotion recognition with a hybrid recommendation engine. It enhances the user experience by reducing manual interaction and providing emotionally relevant music. Future improvements may include incorporating voice emotion analysis and expanding the music database through APIs like Spotify. The system represents a step toward more empathetic, human-centric computing. The system's accuracy and personalization could also be enhanced by incorporating feedback mechanisms and learning from user behavior over time. Emotion-aware digital assistants, mental health support, and other applications that can adapt to human feelings could all be developed using this combination of emotional intelligence and artificial intelligence.

What We Are Solving:

- Issue: Static music recommendation systems that ignore real-time emotional context.
- Our Solution: An intelligent, real-time, facial emotion-based music recommender system that adapts to the user's feelings, making music consumption more personalized and empathetic.
- Future Scope: Integration with APIs like Spotify, voice-based emotion detection, and dynamic playlist generation based on live user feedback and mood history.

REFERENCES

[1] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," Journal of Personality and Social Psychology, 1971.
[2] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.
[3] S. Zhang, X. Zhao, and X. Lei, "Facial Emotion Recognition with Deep Learning," IEEE Transactions on Affective Computing, 2017.
[4] OpenCV Documentation. https://docs.opencv.org/
[5] FER-2013 Dataset. https://www.kaggle.com/datasets/msambare/fer2013
[6] TensorFlow Documentation. https://www.tensorflow.org/
[7] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 emotion challenge," Proceedings of Interspeech, pp. 312–315, 2009.
[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.
[9] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text, and images using deep features," in Proc. 18th Int. Soc. Music Inform. Retrieval Conf. (ISMIR), 2017, pp. 23–30.
[10] A. Kapoor, W. Burleson, and R. W. Picard, "Automatic prediction of frustration," International Journal of Human-Computer Studies, vol. 65, no. 8, pp. 724–736, 2007.