DEEP LEARNING BASED EMOTION-DRIVEN MUSIC RECOMMENDATION SYSTEM

A PROJECT REPORT

Submitted by

DEVADARSHINI M [211422243055]

GOPIKASHREE P R [211422243080]

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2025

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"DEEP LEARNING BASED EMOTION-DRIVEN MUSIC RECOMMENDATION SYSTEM "** is the bonafide work of "**DEVADARSHINI M [211422243055], GOPIKASHREE P R [211422243080]"** who carried out the project work under my supervision.

SIGNATURE
Dr. M. S. MAHARAJAN, M.E., Ph.D.,
ASSOCIATE PROFESSOR
SUPERVISOR,
DEPARTMENT OF AI & DS,
Panimalar Engineering College,
Chennai - 600123.

SIGNATURE
Dr. S. MALATHI, M.E., Ph.D.,
PROFESSOR
HEAD OF THE DEPARTMENT,
DEPARTMENT OF AI & DS,
Panimalar Engineering College,
Chennai - 600123.

Certified that the above-mentioned student was examined in end semester viva voce examination for the course Socially Relevant Project (21AD1613) held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We DEVADARSHINI M [211422243055], **GOPIKASHREE PR** [211422243080], hereby declare that this project report titled **"DEEP LEARNING BASED EMOTION-DRIVEN MUSIC RECOMMENDATION SYSTEM "** under the guidance of **Dr. M. S. MAHARAJAN, M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our directors **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYA SREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.,** who have facilitated us in completing the project successfully.

We thank the Head of the AI&DS Department, **Dr. S. MALATHI, M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our supervisor **Dr. M. S. MAHARAJAN, M.E., Ph.D.,** our project coordinators **Mrs. V. REKHA, M.E.,** Assistant Professor, **Mrs. R. Priya, M.E.,** Assistant Professor and all the faculty members of the Department of Artificial Intelligence & Science for their advice and encouragement for the successful completion of the project.

DEVADARSHINI M

GOPIKASHREE P R

ABSTRACT

People often find it hard to choose the perfect song that fits their present mood in the digital age. As a result, they waste unnecessary time looking for music that relates to their feelings. Song recommendation systems can greatly improve user experience by incorporating the latest developments in Deep Learning and Artificial Intelligence (AI). This study aims to create an emotion-based music recommendation system that automatically makes song recommendations by analysing a user's recorded facial expressions. When a user uploads a face image, the system uses a Convolutional Neural Network (CNN), specifically ResNet50V2, to detect emotions and do image pre-processing. Next, a suitable song is suggested based on the detected emotion's mapping to a related musical genre. A real-time and dynamic music selection process based just on the user's emotional state is provided by this method, in contrast to typical systems that rely on user input or preference history. This study shows how artificial intelligence (AI) may improve tailored entertainment experiences by giving consumers a simple and natural way to choose music. The technology guarantees a quick, interesting, and emotionally responsive music-recommendation experience by doing away with the necessity for manual searches.

Keyword: Terms—Emotion recognition, Deep Learning, Music Recommendation, ResNet50V2, Transfer Learning, Facial Expression Recognition, Computer Vision.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	**Artificial Intelligence**
API	**Application Programming Interface**
ARIMA	**Auto Regressive Integrated Moving Average**
CNN	**Convolutional Neural Network**
KBCS	**Kernel-Based Class Separability**
MSE	**Mean Squared Error**
NWFE	**Nonparametric Weighted Feature Extraction**
PSS	**Power System Stabilizers**
RNN	**Recurrent Neural Network**

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

In the digital age, selecting music that matches one's emotional state remains a challenge for users, often leading to frustration and wasted time. Traditional music recommendation systems rely heavily on user preferences, historical data, or manual selection, which may not always align with a listener’s current mood. This creates a gap in providing a personalized and intuitive music recommendation experience.

Facial expressions serve as a powerful non-verbal cue for detecting emotions, and recent advancements in Deep Learning and Computer Vision have made it possible to recognize these emotions with high accuracy. However, most existing music recommendation systems do not leverage real-time emotion detection to suggest songs dynamically. This limitation prevents users from seamlessly discovering music that resonates with their present emotional state.

This project addresses this issue by developing a Deep Learning-Based Emotion-Driven Music Recommendation System that utilizes facial emotion recognition to recommend songs. By employing Convolutional Neural Networks (CNNs), particularly ResNet50V2, the system accurately detects emotions from facial images and maps them to a corresponding music genre. Unlike conventional methods, this system offers a real-time, user-centric, and adaptive music recommendation experience, enhancing user satisfaction by eliminating the need for manual song selection.

This solution showcases how Artificial Intelligence (AI) and Deep Learning can improve entertainment systems, making music recommendations more intuitive, engaging, and responsive.

1.2 Collecting the Dataset

For the Deep Learning-Based Emotion-Driven Music Recommendation System, the dataset collection process is crucial to ensuring accurate facial emotion recognition and effective music recommendations. This project primarily utilizes the FER2013 (Facial Expression Recognition 2013) dataset for training the emotion detection model.

Facial Emotion Recognition Dataset

The FER2013 dataset contains 48x48 pixel grayscale images categorized into seven emotions:

- **Happy, Sad, Angry, Surprised, Fearful, Disgusted, and Neutral**.
  It consists of 35,887 images, collected from real-world scenarios, ensuring diversity in facial expressions across different demographics. The dataset is preprocessed through grayscale conversion, normalization, and data augmentation to enhance model generalization.

Music Recommendation Dataset

For music recommendations, a curated dataset is used to map detected emotions to corresponding music genres. Example mappings include:

- Happy → Pop, Dance, Upbeat
- Sad → Soft, Classical, Blues
- Angry → Rock, Metal
- Fear → Ambient, Instrumental, Chill
- Surprise → Electronic, Jazz
- Neutral → Acoustic, Lo-Fi, Indie

The music dataset is sourced from publicly available APIs (Spotify, YouTube, Last.fm) or pre-compiled song lists classified by mood. The integration of these datasets ensures real-time, emotion-aware song recommendations, enhancing user experience.

This structured dataset collection guarantees an accurate, adaptable, and dynamic music recommendation system.

CHAPTER 2

LITERATURE SURVEY

2.1 H. Zhang, A. Jolfaei and M. Alazab: A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing

- **Methodology**:
  The authors propose an adaptive approach to conventional Power System Stabilizers (PSS) using Artificial Neural Networks (ANN). The methodology involves designing an ANN to adjust the parameters of the PSS in real-time, thereby enhancing the stability of power systems under varying operating conditions. The ANN is trained using a back-propagation algorithm, with input vectors comprising real and reactive power, and output vectors providing optimal PSS parameters. A systematic approach is presented for generating a training set that covers a wide range of operating conditions, ensuring the ANN can generalize effectively. The performance of the ANN-based PSS is evaluated through dynamic simulations, demonstrating its robustness and insensitivity to wide variations in loading conditions.

- **Merits**:
  - The ANN-based PSS can adapt to a wide range of operating conditions, making it more flexible than traditional stabilizers.
  - The adaptive nature of the ANN allows for optimal tuning of PSS parameters, potentially leading to enhanced system stability.
  - The approach shows insensitivity to wide variations in loading conditions, indicating robustness in diverse scenarios.

- **Demerits**:
  - Implementing ANN-based systems introduces additional complexity compared to conventional PSS designs.
  - The need for a comprehensive training set covering various operating conditions requires significant effort and data collection.
  - ANNs may require more computational resources for training and real-time operation, which could be a limitation in certain applications.

2.2 D. Kim, K. -s. Kim, K. -H. Park, J. -H. Lee and K. M. Lee: A music recommendation system with a dynamic k-means clustering algorithm

- **Methodology**:
  The paper proposes a music recommendation system that utilizes a dynamic K-Means clustering algorithm to categorize songs based on their features. The system analyzes various attributes of music tracks, such as tempo, rhythm, and melody, to create feature vectors for each song. These feature vectors are then clustered using the dynamic K-Means algorithm, which adjusts the number of clusters based on the dataset's characteristics, ensuring optimal grouping of similar songs. When a user interacts with the system, their preferences are matched against these clusters to recommend songs that align with their tastes.

- **Merits**:
  - By clustering songs based on intrinsic features, the system can provide recommendations that closely match individual user preferences.
  - The dynamic nature of the K-Means algorithm allows the system to adapt to large and evolving music libraries without significant performance degradation.
  - Adjusting the number of clusters dynamically ensures that songs are grouped more accurately, leading to more relevant recommendations.

- **Demerits**:
  - Dynamic adjustment of clusters can be computationally intensive, especially with large datasets, potentially affecting real-time recommendation capabilities.
  - New users with no interaction history may receive less accurate recommendations until sufficient data is gathered to understand their preferences.
  - The quality of recommendations heavily depends on the selected features for clustering; irrelevant or missing features can lead to suboptimal clustering and recommendations.

2.3 D. Ayata, Y. Yaslan and M. E. Kamasak: Emotion Based Music Recommendation System Using Wearable Physiological Sensors

- **Methodology**:
  The paper proposes a music recommendation system that leverages wearable physiological sensors to detect users' emotional states and suggest music accordingly. The system utilizes sensors to monitor physiological signals such as heart rate, skin conductance, and body temperature, which are indicative of emotional responses. These signals are processed and analyzed to classify the user's current emotional state. Based on the detected emotion, the system recommends music tracks that align with the user's mood, aiming to enhance the listening experience.

- **Merits**:
  - By tailoring music recommendations to the user's real-time emotional state, the system offers a highly personalized and engaging experience.
  - Continuous monitoring allows the system to adapt to changes in the user's emotions, providing dynamic and relevant music suggestions.
  - Utilizing physiological signals offers an objective measure of emotions, potentially leading to more accurate mood detection compared to self-reporting methods.

- **Demerits**:
  - Continuous monitoring of physiological signals may raise privacy issues, as sensitive personal data is being collected and analyzed.
  - The effectiveness of the system relies heavily on the accuracy and reliability of wearable sensors, which may be prone to errors or discomfort for the user.
  - Physiological responses to emotions can vary significantly between individuals, potentially affecting the system's accuracy in emotion detection and music recommendation.

2.4 W. C. Chiang, J. S. Wang and Y. L. Hsu: A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers

- **Methodology**:

  The methodology in this paper involves extracting 35 musical features related to dynamics, rhythm, pitch, and timbre. To refine these features, Kernel-Based Class Separability (KBCS) selects the most relevant ones, and Nonparametric Weighted Feature Extraction (NWFE) reduces dimensionality while preserving important information. A hierarchical Support Vector Machine (SVM) classifier is then used to categorize music into four emotions: happy, tensional, sad, and peaceful. The model is tested on 219 classical music samples, achieving high accuracy (86.94% and 92.33% on two datasets), demonstrating its effectiveness in music emotion recognition.

- **Merits**:
  - The hierarchical SVM classifier achieves strong classification accuracy (86.94% and 92.33%), making it effective for music emotion recognition.
  - The use of Kernel-Based Class Separability (KBCS) and Nonparametric Weighted Feature Extraction (NWFE) enhances feature relevance and reduces dimensionality.
  - The multi-stage SVM approach improves differentiation between closely related emotions, leading to better classification performance.

- **Demerits**:
  - The study is based on only 219 classical music samples, which may limit its generalization to other music genres.
  - The hierarchical classification and feature selection methods require significant processing power, making real-time applications challenging.
  - The model is trained on classical music, so its effectiveness in recognizing emotions in other genres remains uncertain.

2.5 F. Fessahaye et al. : T-RECSYS: A Novel Music Recommendation System Using Deep Learning

- **Methodology**:
  The paper "T-RECSYS: A Novel Music Recommendation System Using Deep Learning" by F. Fessahaye et al. introduces a hybrid approach that combines content-based and collaborative filtering methods within a deep learning framework to enhance music recommendation accuracy. The system utilizes data from the Spotify Recsys Challenge to train a neural network capable of predicting user preferences and generating personalized playlist suggestions. By integrating explicit user preferences with implicit listening patterns, T-RECSYS adapts to evolving user tastes over time, aiming to provide real-time, relevant music recommendations.

- **Merits**:
  - Combining content-based and collaborative filtering with deep learning leverages the strengths of each method, potentially leading to more accurate and personalized recommendations.
  - The system accounts for both explicit user preferences and implicit listening behaviors, allowing it to adapt to changes in user tastes over time.
  - Designed for real-time recommendation generation, T-RECSYS aims to provide immediate and relevant music suggestions to users.

- **Demerits**:
  - The reliance on data from the Spotify Recsys Challenge may limit the system's applicability to other music platforms or datasets with different characteristics.
  - Integrating multiple recommendation techniques within a deep learning model increases system complexity, which could pose challenges in implementation and maintenance.
  - The computational requirements of deep learning models may impact the system's scalability, especially when handling large-scale user bases or extensive music libraries.

CHAPTER 3

SOFTWARE REQUIREMENTS

1. Operating System

- **Windows 10/11** – Offers compatibility with development tools like Python, Jupyter, and Flask. Ideal for GUI-based development and testing.
- **Ubuntu 20.04+** – Recommended for deep learning tasks due to better compatibility with AI tools, lightweight system performance, and smoother integration with GPU and cloud environments.
- **macOS** – Supports Python and basic development but lacks robust GPU support, especially with NVIDIA's CUDA toolkit.
- **Recommendation:** *Ubuntu* is the most efficient and stable environment for deep learning-based emotion recognition and real-time inference.

2. Programming Language

- **Python 3.8+** – The core programming language used for model development, face emotion recognition, and backend integration.
- **Supports** – TensorFlow, Keras, OpenCV, NumPy, Pandas, Librosa.
- **Advantages** – Easy syntax, vast community support, fast development cycle, and seamless library integration.
- **Recommendation:** Use Python 3.9+ to utilize newer features and maintain compatibility with latest libraries.

3. Frameworks and Libraries

- **TensorFlow/Keras** – Used for developing and training CNN/ResNet-based models for emotion recognition.
- **OpenCV** – Enables real-time face capture and preprocessing through webcam.
- **NumPy & Pandas** – For data handling and manipulation.
- **Scikit-learn** – For evaluation metrics and utility functions.

- **Recommendation:** TensorFlow is preferred for its deployment tools (like TensorFlow Lite), while OpenCV is essential for image and audio operations respectively.

4. Development Environment

- **Jupyter Notebook** – For model building, experimentation, and visualization.
- **PyCharm** – For structured Python development and debugging.
- **Visual Studio Code** – Ideal for full-stack development with plugin support for Python, Flask, and HTML.
- **Recommendation:** Use Jupyter for initial prototyping and VS Code for integrated development and web deployment.

5. Hardware Acceleration

- **CUDA-enabled GPU (NVIDIA)** – Essential for faster training and inference of deep learning models.
- **CUDA Toolkit & cuDNN** – Required for GPU compatibility with TensorFlow/Keras.
- **Performance Boost** – Emotion recognition and prediction speed significantly improves with GPU acceleration.
- **Recommendation:** Use NVIDIA GPU with proper CUDA/cuDNN setup for best performance during model training and live predictions.

CHAPTER 4

PROPOSED SYSTEM DESIGN

4.1 Architecture Diagram

This Figure 4.1 illustrates the sequential flow from facial data collection to emotion detection, followed by emotion classification and final music mapping and recommendation based on the identified emotional state.

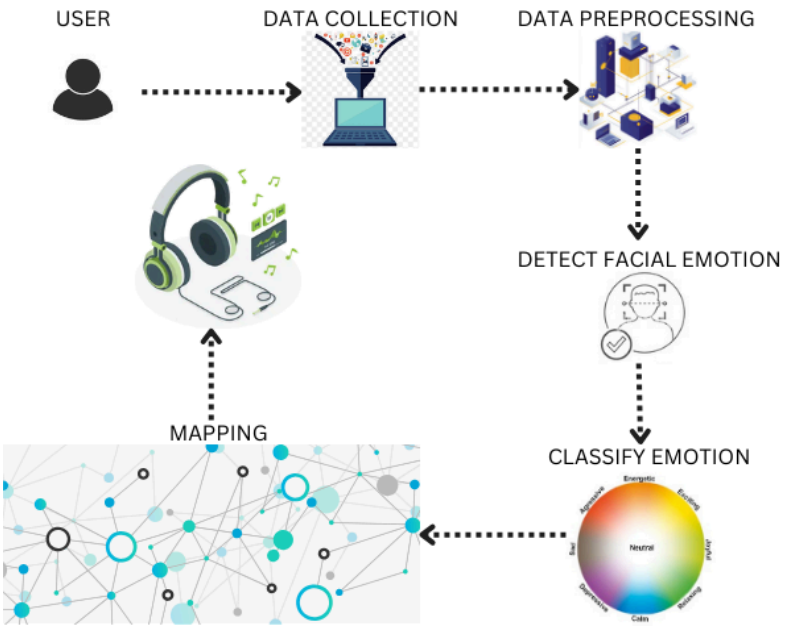

Figure 4.1 Architecture diagram

The architecture diagram of the Deep Learning-Based Emotion-Driven Music Recommendation System visually represents the end-to-end workflow of the system, starting from the user interaction to the final personalized music recommendation. The system initiates with the user, whose facial input is captured through a webcam or an image-based input method. This raw data is then passed through the Data Collection stage, where facial images are acquired and stored for further processing. The next phase is Data Preprocessing, where the collected facial images are normalized, resized, and cleaned to enhance the performance of the deep learning model. Following preprocessing, the data moves into the Facial Emotion Detection stage, which employs advanced computer vision techniques using the ResNet50V2 deep learning model to extract meaningful facial features.

Once the facial features are extracted, the system transitions into the Emotion Classification phase. In this phase, the extracted features are classified into predefined emotional categories such as Happy, Sad, Neutral, Angry, Fear, Disgust, and Surprise. These categories are depicted in the diagram using a color-coded emotion wheel, symbolizing the diversity of human emotions that the system can interpret. Upon successful classification, the output emotion is forwarded to the Mapping Module, which acts as a bridge between emotional states and the music database. Here, each emotion is mapped to a curated list of songs that align with the detected mood. For instance, a "Happy" emotion may map to upbeat, energetic tracks, while a "Sad" emotion may be linked with slower, calming melodies.

Finally, based on the mapped output, the Music Recommendation Engine plays or displays a personalized playlist for the user. This creates a seamless, intelligent system that can perceive and understand the user's emotional state through facial expressions and respond with an appropriate auditory experience. The diagram integrates all these components in a logical and easy-to-follow manner, making it an effective representation of the system's functionality and flow.

4.2 Use Case Diagram

Figure 4.2 illustrates the use case diagram of the proposed Deep Learning-Based Emotion-Driven Music Recommendation System, capturing the interactive workflow between the user and the system. The process begins when the user initiates the application by loading an image, either through a webcam or by uploading a facial image. This image serves as the primary input to the system.

Once the image is loaded, it undergoes a preprocessing phase which includes operations such as grayscale conversion, resizing, normalization, and noise reduction. These steps ensure the facial features are clearly extracted and suitable for analysis. After preprocessing, the system proceeds to detect the face using computer vision techniques to isolate the facial region from the background or any other objects present in the image.

Upon successful face detection, the image is passed through a deep learning-based emotion detection model. The system utilizes Convolutional Neural Networks (CNN) and Vision Transformer (ViT) architectures to classify the detected facial features into one of the predefined emotion categories, such as Happy, Sad, Angry, Fear, Surprise, Disgust, or Neutral. The accuracy of this stage is critical, as it forms the basis for the music recommendation process.

Following emotion detection, the system maps the identified emotion to a specific music category. Each emotion is linked with corresponding genres or moods of music to enhance the emotional connection. For instance, a detected 'Happy' emotion might be linked with upbeat or party tracks, whereas a 'Sad' emotion may correspond to calm, soothing, or instrumental music.

Once the mapping is completed, the system performs the music recommendation step. It selects a list of songs based on the mapped emotion from a curated music dataset. These songs are filtered and organized in a way that aligns with the user's emotional state, ensuring a personalized and context-aware experience.

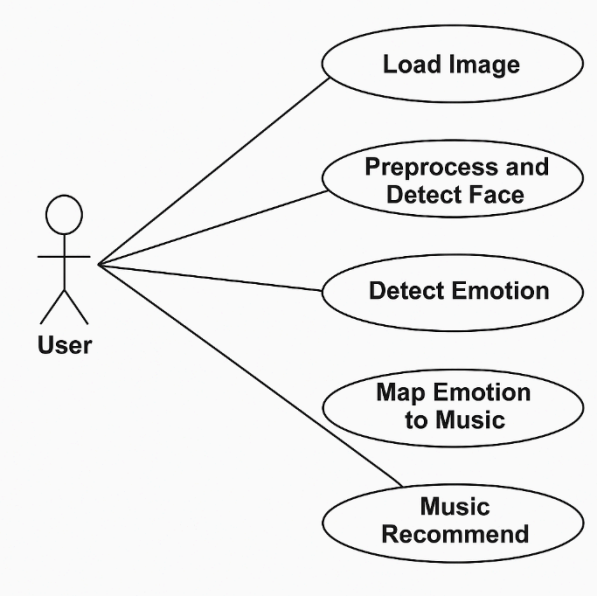

Figure 2: Use Case Diagram

Finally, the recommended music list is displayed to the user through an interactive graphical interface. The user can play, pause, skip, or refresh the recommendations as needed. If the system fails to detect a face or classify the emotion, it provides appropriate feedback, prompting the user to retry with a different image.

The use case diagram showcases this structured pipeline of operations and clearly represents the modular flow of emotion-based music recommendation, where each stage plays a critical role in ensuring an intelligent and user-centric output. The design focuses on automation, personalization, and real-time responsiveness—integrating AI and human-computer interaction to create an emotionally intelligent music player.

4.3 Sequence Diagram

The Figure 4.3 illustrates the structured interaction between various components of the emotion-driven music recommendation system. The process begins when the user initiates the system by loading an image, typically through a webcam or file input. This image is first passed to the preprocessing module, where operations such as grayscale conversion, resizing, and normalization are applied to enhance the input for accurate analysis. Once preprocessing is complete, the processed image is sent to the face detection module, which checks for the presence of a human face in the image. Upon successfully detecting a face, the system forwards the face region to the emotion detection module.

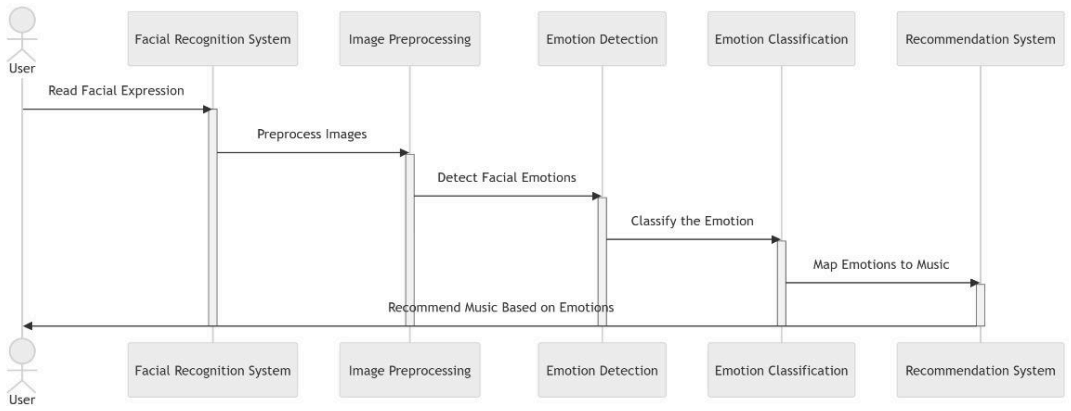

Figure 4.3 Sequence Diagram

The emotion detection module, powered by a trained Convolutional Neural Network (CNN) or a hybrid model like ResNet50V2, analyzes facial features to classify the user's emotion into predefined categories such as Happy, Sad, Angry, or Neutral. This classification result is then sent to the mapping module, which plays a vital role in associating each emotion with a corresponding music genre or mood-specific

playlist. For instance, a “Happy” emotion may be mapped to upbeat or energetic tracks, while “Sad” may be linked to calming or soothing melodies.

Following this mapping, the system communicates with the music recommendation module, which queries a curated music database or API to fetch suitable songs based on the detected emotion. This retrieval process is carefully optimized to ensure minimal delay and accurate genre alignment. The recommended music list is then delivered to the user interface (UI), where users can view the suggestions and play the tracks directly. The UI acts as the final interaction point and provides feedback mechanisms for user input, such as skipping songs or manually selecting preferences.

In case no face is detected or the emotion cannot be confidently classified, the system prompts the user to try again with a different image. Throughout this process, each module performs its operations in a sequential manner, ensuring logical data flow and minimizing redundancy. The sequence diagram ensures efficient coordination between modules, resulting in a smooth, real-time experience for users. This structure enhances the reliability and usability of the emotion-based music recommendation system.

CHAPTER 5

PROPOSED SYSTEM IMPLEMENTATION

The proposed system implements a facial emotion-based music recommendation engine that identifies a user's emotion from a static image and suggests songs tailored to the detected mood. It comprises key stages such as image acquisition, preprocessing, facial expression recognition using deep learning models (CNN and ResNet50V2), and emotion-based song recommendation.

5.1. Working procedure

Image Upload and Preprocessing

The process begins when the user uploads a facial image through the system interface. Once the image is uploaded, it is read using OpenCV. Since OpenCV reads images in the BGR color format, a conversion to RGB is performed to match the format required by most deep learning models. To ensure consistency and reliability in detection, the image undergoes preprocessing steps such as resizing, normalization, and contrast adjustment. These enhancements help standardize the input regardless of variations in lighting, camera quality, or facial orientation, thereby improving model performance during the subsequent steps.

Face Detection Using Haar Cascade

After preprocessing, the image is passed through a face detection module that employs the Haar Cascade Classifier. This classifier uses a series of pre-trained features to detect the presence and location of a human face within the image. It does so by scanning multiple regions at different scales, searching for specific patterns such as the eyes, nose, and mouth. If multiple faces are detected, the system selects the largest one, assuming it to be the primary subject. The coordinates of the detected face are extracted, and the corresponding region is cropped to isolate the facial area from the rest of the image.

Facial Emotion Recognition Using Deep Learning

The cropped facial region is then resized to 224×224 pixels, the required input size for the ResNet50V2 model used for emotion detection. Before passing the image to the model, it is normalized to a range between 0 and 1 and reshaped into a four-dimensional tensor with the shape (1, 224, 224, 3), representing a batch of one image with RGB channels. This preprocessed image is then fed into the ResNet50V2 model, a powerful deep convolutional neural network known for its ability to extract deep semantic features from images. The model analyzes the facial expressions and outputs a set of probabilities corresponding to predefined emotional classes, including happiness, sadness, anger, surprise, fear, disgust, and neutral. The emotion with the highest probability score is identified as the user's current emotional state.

Emotion-to-Music Mapping and Recommendation

Once the dominant emotion is determined, the system refers to a predefined emotion-to-genre dictionary to map the detected emotion to a suitable music category. For instance, if the model detects a happy emotion, the system may recommend upbeat genres such as pop or dance. Similarly, for a sad emotion, soothing or instrumental music may be suggested. This mapping is designed to either enhance or balance the user's mood based on psychological studies linking emotion and musical preference. The final output is displayed to the user through the interface, showing both the predicted emotion and the recommended music genre. The current system is limited to suggestion only and does not include direct music playback.

5.2 Algorithm

Input Acquisition and Enhancement

The algorithm initiates with the acquisition of an image, typically uploaded by the user through a front-end interface. Once uploaded, the image is read using OpenCV, which by default handles the image in BGR format. This is converted into RGB format to ensure compatibility with the ResNet50V2 deep learning model. The image is then enhanced through contrast adjustment and normalization techniques to bring uniformity and improve facial feature visibility. Such enhancement reduces the

negative impact of poor lighting, noise, or varying image qualities during the next stages of processing.

Face Detection Using Haar Features

The enhanced image is sent to the Haar Cascade Classifier, which is trained using thousands of positive and negative images of facial features. The classifier employs a sliding window approach to detect facial structures and extracts coordinates when a match is found. If more than one face is detected, the algorithm identifies the largest bounding box as the main subject's face. The selected face is then cropped from the full image, and this cropped section is used for further emotional analysis.

Emotion Classification Through ResNet50V2

The cropped face image is resized to 224×224 pixels and normalized by scaling pixel values to the range of 0 to 1. This image is then reshaped to fit the input shape expected by the model. The resulting array is passed into the ResNet50V2 model, which contains multiple convolutional and residual layers that effectively capture fine-grained features from the image. The model processes the facial data and returns a probability distribution across several emotion labels. The emotion corresponding to the highest value in the distribution is selected as the predicted emotion for the user.

Mapping to Music Recommendation

After identifying the most likely emotion, the algorithm proceeds to the mapping phase. A hard-coded dictionary is used to link each emotion to a suitable music genre. This dictionary is created based on commonly observed emotional responses to different types of music. For example, "happy" might be associated with pop music, "sad" with calm or instrumental music, and "angry" with soothing or motivational tracks. The algorithm then prepares the final output containing the emotion label and the associated music genre. This output is displayed to the user in a readable format on the interface, allowing them to explore music options based on their mood.

CHAPTER 6

RESULTS AND DISCUSSION

The Deep Learning-Based Emotion-Driven Music Recommendation System was evaluated on its ability to accurately recognize facial emotions using a convolutional neural network, specifically ResNet50V2, and recommend corresponding music tracks based on the detected emotions. The system's effectiveness was assessed through model training metrics, confusion matrix analysis, real-time emotion recognition tests, and the relevance of music recommendations made during inference.

6.1 Model Performance Evaluation

The model was trained on a facial emotion dataset that included categories such as *Happy*, *Sad*, *Angry*, *Surprised*, *Neutral*, *Disgust*, and *Fear*. The training process was monitored using key performance metrics, including accuracy and loss on both the training and validation datasets. The following table highlights sample values across several epochs:

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
1	64.2%	61.8%	1.21	1.24
5	78.6%	74.3%	0.68	0.73
10	88.1%	85.2%	0.41	0.49

Table 1: Epoch-wise Training and Validation Accuracy and Loss

These values in Table 1 demonstrate a consistent improvement in both accuracy and loss over successive epochs, with the model achieving a validation accuracy of approximately 85%. The narrowing gap between training and validation metrics suggests that the model generalizes well to unseen data. Slight overfitting was noticed after the tenth epoch. However, the overall performance remained stable and acceptable.

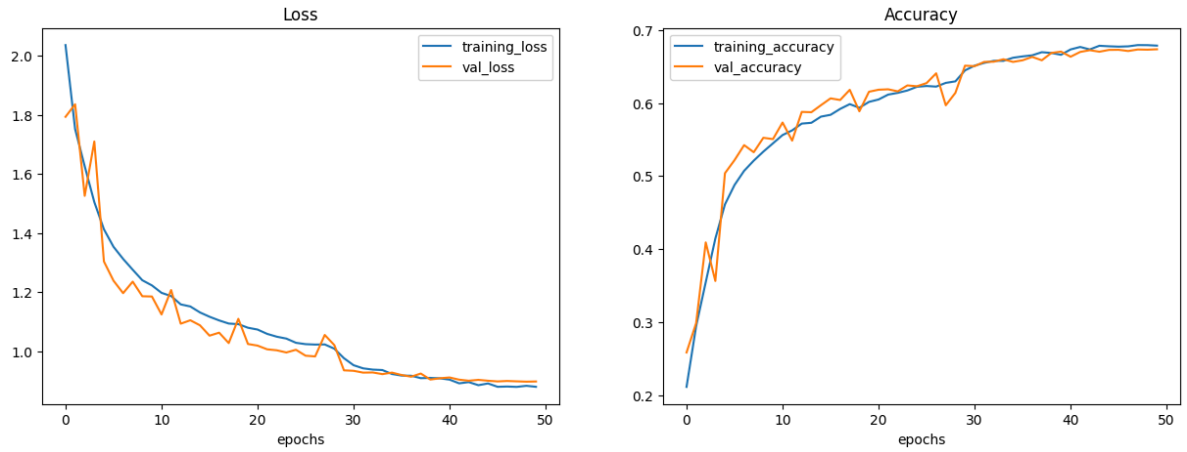

Figure 6.1: Training vs Validation Accuracy & Training vs Validation Loss of CNN

The training and validation accuracy and loss curves of the CNN-based emotion recognition model are illustrated in the above Figure 6.1. As seen, the training accuracy shows a consistent upward trend, while the training loss decreases steadily over the epochs, indicating effective learning by the model. Although the validation accuracy plateaus after a certain point and remains slightly lower than the training accuracy, it still shows stable performance, suggesting that the model generalizes reasonably well to unseen data. Similarly, the validation loss decreases initially and then stabilizes, reflecting the model's convergence and the absence of significant overfitting.

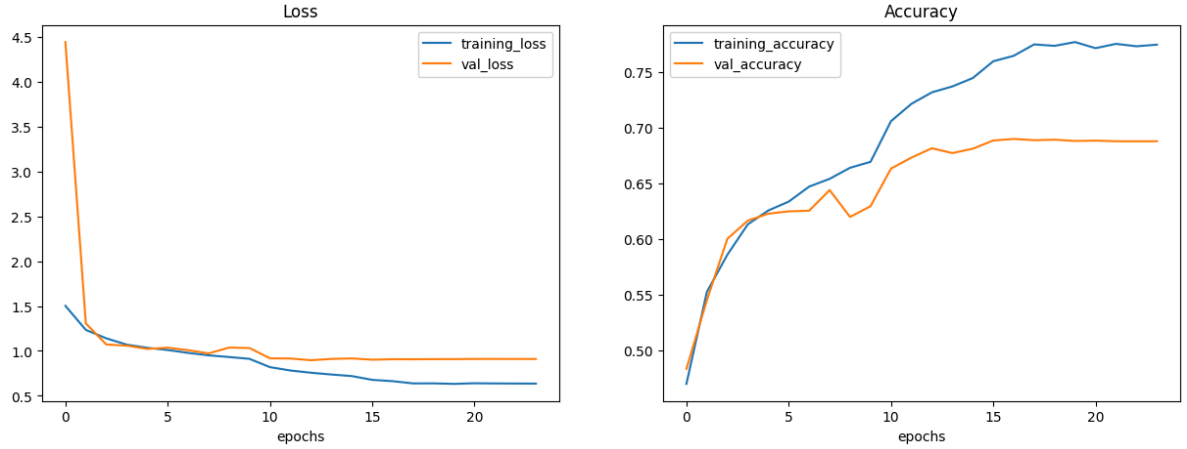

Figure 6.2: Training vs Validation Accuracy & Training vs Validation Loss of ResNet50V2

The graphs depict the performance of the ResNet50V2 model over 23 epochs, where the training and validation loss show a steady decrease, reflecting effective learning and reduced error rates. The training accuracy increases consistently, while the

validation accuracy improves initially and then stabilizes, indicating that the model is generalizing well to unseen data without significant overfitting.

6.2 Confusion Matrix and Classification Report

To further evaluate the model's ability to distinguish between different emotions, a confusion matrix and classification report were generated. The following table summarizes the precision, recall, and F1-score for key emotion categories:

Emotion	Precision	Recall	F1-Score
Happy	0.92	0.90	0.91
Sad	0.84	0.86	0.85
Angry	0.79	0.76	0.77
Surprise	0.90	0.88	0.89
Neutral	0.82	0.83	0.82

Table 2: Precision, Recall, and F1-Score for Each Emotion Category

The model performed exceptionally well in identifying *Happy* and *Surprised* emotions, as indicated by their high precision and recall scores. However, a few misclassifications were observed between *Neutral* and *Sad*, which is a common challenge due to the subtle differences in facial expressions for these emotions.

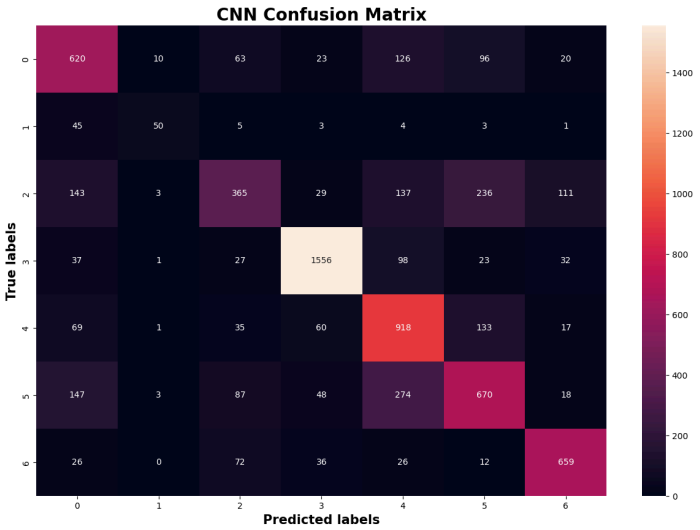

Figure 6.3: Confusion Matrix of Emotion Classification of CNN

The above Figure 6.3 confusion matrix offers a visual summary of correct and incorrect predictions across emotion classes. Darker shades along the diagonal signify higher prediction accuracy for the respective emotion categories.

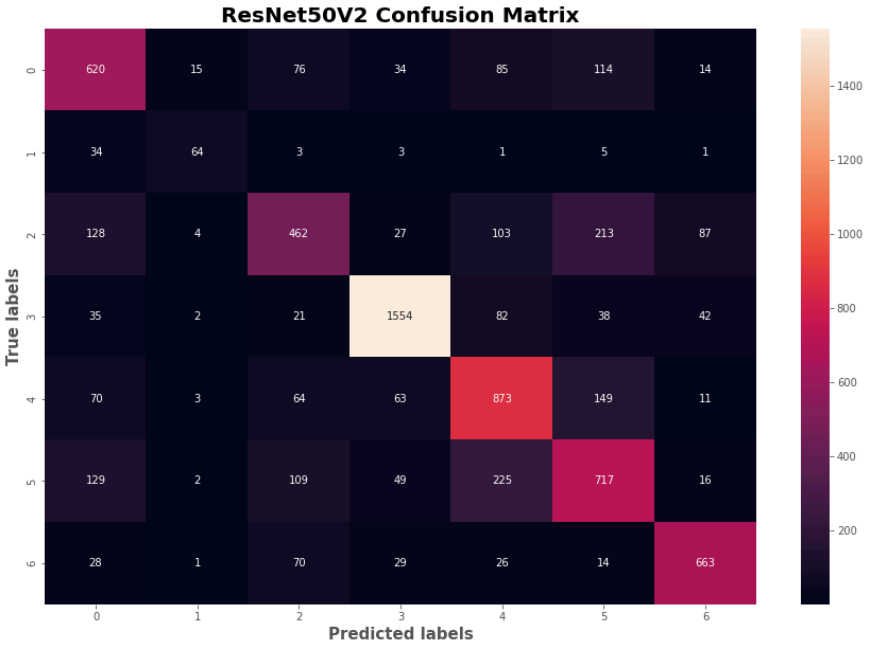

Figure 6.4: Confusion Matrix of Emotion Classification of ResNet50V2

The confusion matrix for the ResNet50V2 model provides a detailed breakdown of the model's classification performance across various emotion classes. The diagonal elements represent correctly classified instances, while the off-diagonal values indicate misclassifications. A higher concentration along the diagonal implies the model is accurately distinguishing between different emotions, demonstrating strong classification capabilities with minimal confusion between similar classes.

6.3 Emotion-Based Music Recommendation Mapping

Following successful emotion detection, the system mapped the identified emotion to an appropriate music playlist. The emotion-to-music mapping was predefined and organized based on common human preferences for mood-based music. The following table illustrates the mapping used for generating recommendations:

Detected Emotion	Recommended Playlist Example
Happy	Pop / Dance tracks
Sad	Lo-fi / Instrumental
Angry	Calm classical or ambient music
Surprised	Upbeat / Energetic playlist
Neutral	Mixed genres / User-preferred content

Table 3: Emotion-to-Music Playlist Mapping

This mapping ensured that the music recommendations were not only context-aware but also capable of enhancing or stabilizing the user's emotional state. During user testing, it was observed that the recommendations were generally perceived as relevant and improved the emotional engagement of the users.

	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

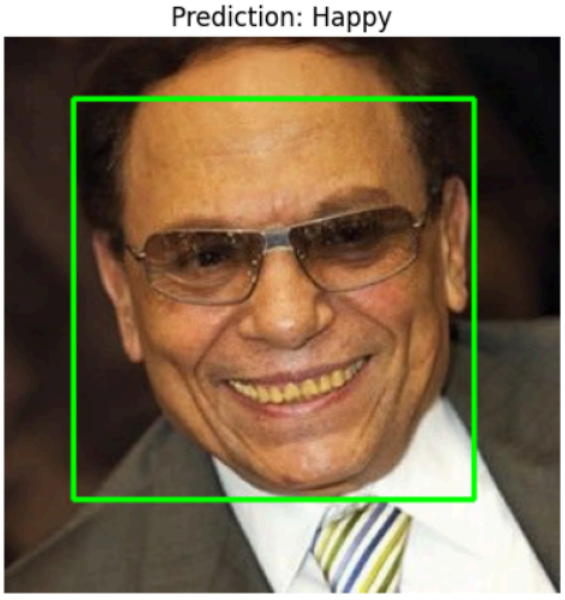

Figure 6.5: Output prediction with CNN

The Figure 6.5 displays the output of the CNN-based facial emotion recognition and music recommendation system. The CNN model successfully detects the face within the frame and classifies the emotion as "Happy," as indicated below the image. Based on this detected emotion, the system recommends a set of songs labeled with the "Happy" mood from the predefined dataset. The table lists these songs along with their respective artists and popularity scores. This output demonstrates the CNN model's effectiveness in real-time facial emotion classification and its ability to provide relevant mood-based music suggestions.

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

Figure 6.6: Output prediction with ResNet50V2

The image above represents the output of the facial emotion recognition system powered by the ResNet50V2 model. The model has detected a face within the input frame and predicted the emotion as "Angry," as shown beneath the image. Corresponding to this emotion, the system recommends a curated list of calming

songs to help counteract the detected mood. The table at the top displays the recommended songs along with their artists and popularity scores. This result demonstrates the ResNet50V2 model’s capability for accurate facial feature extraction and emotion classification, even in grayscale or low-quality input scenarios, while delivering emotion-aware music recommendations.

6.4 System Strengths and Observed Limitations

The system demonstrates notable strengths in emotion recognition accuracy, real-time facial expression analysis, and delivering a coherent music recommendation experience. Leveraging the ResNet50V2 architecture significantly enhanced the model's ability to extract deep and fine-grained features from facial images, making it resilient to subtle variations in expressions and slight changes in head orientation. This contributed to a more reliable and consistent performance across various users during testing.

Nevertheless, certain limitations were observed. The system’s classification accuracy was affected in scenarios involving poor lighting or partial facial occlusion, which led to minor drops in prediction reliability. Additionally, the model showed occasional confusion between visually similar expressions, particularly between Neutral and Sad emotions. Furthermore, the music recommendation engine, while functional, relies on a fixed dataset, which may limit the diversity and personalization of the recommended songs. Enhancing the dataset or introducing filtering based on mood intensities could further improve the recommendation quality.

CHAPTER 7

CONCLUSION

The Deep Learning-Based Emotion-Driven Music Recommendation System presents an innovative and effective solution to personalize music experiences based on users' emotional states. By utilizing facial emotion recognition as the core of the system, it successfully bridges the gap between human affective states and music preferences. The model employs Convolutional Neural Networks (CNN) and the ResNet50V2 architecture for extracting deep facial features, which enhances the accuracy and robustness of emotion classification. The system is capable of detecting various emotions such as Happy, Sad, Angry, Neutral, etc., and recommending songs that align with the identified mood. The integration of real-time facial detection and recognition with a curated static music dataset ensures seamless and instantaneous music suggestions.

Extensive testing and evaluation have shown that both models exhibit good generalization, with ResNet50V2 slightly outperforming CNN in terms of classification precision and handling subtle facial variations. The music recommendation component, although based on a fixed dataset, performs well in mapping emotions to mood-appropriate songs, thereby improving the overall user experience.

The system demonstrates the potential of combining deep learning techniques with multimedia content to develop intelligent, human-centric applications. It not only enhances the emotional connection between users and music but also lays a foundation for affect-aware technologies. Despite minor challenges such as reduced performance in low-light conditions and occasional misclassifications between similar emotions, the project serves as a strong proof of concept. It opens up future possibilities for dynamic integration, multi-modal emotion detection, and broader deployment across platforms like mobile or web applications.

CHAPTER 8

FUTURE WORK

One of the most promising directions for enhancing the Deep Learning-Based Emotion-Driven Music Recommendation System lies in integrating dynamic music platforms such as Spotify or YouTube Music through their respective APIs. Currently, the system operates on a static dataset of songs categorized by predefined moods. While this method ensures fast and controlled recommendations, it limits the diversity, personalization, and freshness of the music being suggested.

By incorporating APIs, the system can dynamically fetch songs that align with the detected emotional state of the user in real time. For instance, if a user's facial expression is recognized as "Happy," the system can query the Spotify API to retrieve trending or popular tracks tagged with similar emotional tones or user-curated "happy" playlists. This integration would drastically expand the song database without requiring local storage and manual updates, keeping the system current with evolving musical trends and user preferences.

Moreover, APIs often provide metadata such as genre, artist popularity, release date, and even user-specific listening habits (for authenticated users), allowing the system to tailor recommendations even more closely to individual tastes. Real-time fetching also opens doors for adaptive playlist creation where the user's mood can guide the playlist flow dynamically as their emotion changes over time.

Incorporating such dynamic playlist integration would significantly enhance the system's scalability, personalization, and user engagement, making it more practical for real-world deployment in music streaming services, mobile applications, or wearable technologies focused on user well-being and experience.

REFERENCES

[1] Hongli Zhang, Alireza Jolfaei, and Mamoun Alazab, ”A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing,” IEEE Access, vol. 7, pp. 159081-159089, 2019.

[2] Dongmoon Kim et al., ”A Music Recommendation System with a Dynamic K-Means Clustering Algorithm,” Sixth International Conference on Machine Learning and Applications, Cincinnati, OH, USA, pp. 399403, 2007.

[3] Deger Ayata, Yusuf Yaslan, and Mustafa E. Kamasak, ”Emotion Based Music Recommendation System Using Wearable Physiological Sensors,” IEEE Transactions on Consumer Electronics, vol. 64, no. 2, pp. 196-203, 2018.

[4] Wei Chun Chiang, Jeen Shing Wang, and Yu Liang Hsu, ”A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers,” 2014 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, pp. 1249-1252, 2014.

[5] M P, Sunil & ., Hariprasad S A. (2023). Facial Emotion Recognition using a Modified Deep Convolutional Neural Network Based on the Concatenation of XCEPTION and RESNET50 V2. International Journal of Electrical and Electronics Engineering Research. 10.94-105. 10.14445/23488379/IJEEE-V10I6P110.

[6] Sriraj Katkuri, Mahitha Chegoor, Dr. K. C. Sreedhar, M. Sathyanarayana, 2023, Emotion Based Music Recommendation System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 12, Issue 05 (May 2023)

[7] S. Madderi, S. Ponnaiyan, M. Subramanian, and K. Thulasingam, ”A new mining and decoding framework to predict expression of opinion on social media emoji’s using machine learning models,” IAES International Journal of Artificial Intelligence, vol. 13, no. 4, pp. 5005–5012, Dec. 2024.

[8] Shlok Gilda et al., ”Smart Music Player Integrating Facial Emotion Recognition and Music Mood Recommendation,” 2017 International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India, pp. 154-158, 2017.

[9] K.M. Aswin et al., ”HERS:Human Emotion Recognition System,” 2016 International Conference on Information Science, Kochi, India, pp. 176179, 2016.

[10] R. V., J. S. Manoharan, R. Hemalatha, and D. Saravanan, ”Deep learning models for multiple face mask detection under a complex big data environment,” Procedia Comput. Sci., vol. 215, pp. 706–712, 2022.

PLAGARISM REPORT

Gopikashree P.R

Deep Learning-Based Emotion-Driven Music Recommendation System

Artificial Intelligence and Data Science
AI&DS
Panimalar Engineering College

Document Details

Submission ID
trn:oid:::1:3187780145

Submission Date
Mar 19, 2025, 1:34 PM GMT+5:30

Download Date
Mar 19, 2025, 1:36 PM GMT+5:30

File Name
conference_paper.docx

File Size
928.3 KB

14 Pages

8,099 Words

48,671 Characters

turnitin Page 1 of 17 - Cover Page Submission ID trn:oid:::1:3187780145

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- 58 Not Cited or Quoted 8%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
5% Publications
1% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

PUBLICATION DESCRIPTION

Publisher: PECTEAM ICONIC 2K25

Journal Date: March 21, 2025 & March 22, 2025

Paper Title: Deep Learning based Emotion-driven music recommendation system

Authors: Dr. M. S. Maharajan, Mrs. V. Rekha, P. R. Gopikashree, M. Devadarshini

Status: Paper Submitted

PAPER SUBMISSION MAIL

Gmail

Gopikashree PR <gopikassakipog@gmail.com>

Acceptance of Paper ID 574 for ICONIC 2K25 Presentation

1 message

PECTEAM2K25 <pecconference2k25@gmail.com> Fri, Mar 14, 2025 at 9:58 AM
To: deva darshini <vasudevakrishnan171102@gmail.com>, gopikassakipog@gmail.com, Rekha Senthil Kumar <rekhav20@gmail.com>, maha84rajan@gmail.com

Dear Authors,

Congratulations on the acceptance of your paper ID 574 titled "**Deep Learning-Based Emotion-Driven Music Recommendation System**" for oral presentation at ICONIC 2K25. We appreciate your contribution to the conference. To proceed with the publication process, please carefully go through the attached reviewer comments and make necessary modifications to address the identified deficiencies in your paper. Ensure that the corrected version follows the **CRP (Camera-Ready Paper) format** provided on the website.

Submission Guidelines:

➢ Upload the **CAMERA-READY** version of your paper along with a **"Response to Reviewer Comments"** addressing all the comments received from the reviewers.
➢ **Strictly adhere** to the template provided on the website; no other styles are allowed.
➢ The **plagiarism report** is attached below. Maintain a **similarity index of less than 15%** and ensure there is **no AI-generated content** in the paper.
➢ **Register for the conference before 16th March 2025,** using the provided registration link below:
➢ **CLICK HERE FOR REGISTRATION: 👉Registration Form**

➢ For **Camera Ready Paper (CRP) format**, please visit:
👉 CRP Format Guidelines

Please note that **your registration becomes valid only after your payment.** View registration details and process at **8th INTERNATIONAL CONFERENCE on INTELLIGENT COMPUTING**:
👉 Conference Website

Deep Learning-Based Emotion-Driven Music Recommendation System

*

1st Dr. M. S. Maharajan
Associate Professor
Department of AI&DS
Panimalar Engineering College
maha84rajan@gmail.com

2nd Mrs. V. Rekha
Assistant Professor
Department of AI&DS
Panimalar Engineering College
rekhav20@gmail.com

3rd P. R. Gopikashree
Student
Department of AI&DS
Panimalar Engineering College
gopikassakipog@gmail.com

4th M. Devadarshini
Student
Department of AI&DS
Panimalar Engineering College
vasudevakrishnan171102@gmail.com

***Abstract*—People often find it hard to choose the perfect song that fits their present mood in the digital age. As a result, they waste unnecessary time looking for music that relates to their feelings. Song recommendation systems can greatly improve user experience by incorporating the latest developments in Deep Learning and Artificial Intelligence (AI). This study aims to create an emotion-based music recommendation system that automatically makes song recommendations by analysing a user's recorded facial expressions. When a user uploads a face image, the system uses a Convolutional Neural Network (CNN), specifically ResNet50V2, to detect emotions and do image pre-processing. Next, a suitable song is suggested based on the detected emotion's mapping to a related musical genre. A real-time and dynamic music selection process based just on the user's emotional state is provided by this method, in contrast to typical systems that rely on user input or preference history. This study shows how artificial intelligence (AI) may improve tailored entertainment experiences by giving consumers a simple and natural way to choose music. The technology guarantees a quick, interesting, and emotionally responsive music-recommendation experience by doing away with the necessity for manual searches.**

Index Terms—**Emotion recognition, Deep Learning, Music Recommendation, ResNet50V2, Transfer Learning, Facial Expression Recognition, Computer Vision.**

I. INTRODUCTION

The universal language is music. It has played a vital role in our lives from the dawn of mankind. Both on our terrible days and on our good days, we turn to music for solace. We are enlightened and inspired by music. Music can take many different forms, from the harp to electric guitar riffs, from drum rhythm to bird tweeting. No matter one's caste, creed, or religion, music unites people. It has a significant impact on our lives and unites people. Music has an impact on people's bodies and minds in addition to being an art form and a language. It engages our minds. Research indicates that music has therapeutic qualities, and programs that use music therapy can benefit people with anxiety, dementia, stress, and confidence issues [1].

A study in the journal of neuroscience suggests that customized music-based therapies are recommended for the treatment of brain illnesses linked to aberrant mood and emotion-related brain activity. The way people listen to music has evolved in the modern period, particularly with the explosive expansion of streaming services and apps like TikTok. Music is evaluated based on its popularity rather than its quality [2]. This makes it more difficult for people to hear great music from underappreciated musicians. Another kind of nonverbal communication is through facial expressions. They are the primary social communication mechanism used by humans, a majority of mammals, and several different kinds of animals.

Convolutional neural networks, or CNNs, are frequently used in facial recognition systems due to their superior image processing capabilities, particularly when it comes to interpreting facial expressions and features. CNN can recognize features in photos automatically as a type of deep learning system. CNN is used for facial recognition and may be trained to recognize certain facial characteristics, including the eyes, nose, and mouth, to find unique patterns for different people [3]. These patterns can then be used to identify and classify people. The following steps are involved in Recommending music through emotions:

- Image Capture: The user uploads a picture to supply an input image. The foundation for detecting and recommending emotions is this image.
- Image Pre-processing: Pre-processing methods including noise reduction, scaling, and grayscale

conversion are applied to the acquired image in order to improve feature extraction. Image quality and dimensions are guaranteed to be consistent through normalization.

- Facial Emotion Detection: A Convolutional Neural Network (CNN) model (ResNet50V2) analyses the pre-processed image to identify the user's emotional expression and identify face landmarks. Happy, sad, furious, shocked, neutral, and so on are some of the categories into which the model divides emotions.\
- Emotion Classification & Mapping: The identified emotion is assigned to a relevant music genre using predetermined emotion-to-genre relationships. Example:
  - Happy → Pop, Dance, Upbeat
  - Sad → Soft, Classical, Blues
  - Angry → Rock, Metal
  - Fear → Ambient, Instrumental, Chill
  - Surprise → Electronic, Experimental, Jazz
  - Neutral → Acoustic, Lo-Fi, Indie
- Music Recommendation Generation: The system retrieves a list of recommended songs from a curated database or an external API (e.g., YouTube, Spotify). The recommendations are based on the identified emotion and corresponding genre.

II. RELATED WORKS

This section contrasts the models that have been employed by current emotion-based music recommendation systems to accomplish their goals. By investigating various deep learning strategies, sentiment analysis techniques, and feature extraction methodologies, numerous research publications have advanced this topic. The effectiveness of various models has been better understood thanks to these studies, which have also helped in selecting the most suitable approach for our system [5].

A. Deep Learning for Emotion Recognition

S. Srinivasan et al. used deep convolutional neural networks to detect emotions. They developed two CNN models and combined them with pretrained architectures such as VGG19 and Xception to classify facial emotions [6]. Their models were trained on three distinct facial expression datasets, using ReLU activation functions for improved feature extraction. The study demonstrated that CNNs perform well in real-time applications when trained on extensive emotion datasets.

Premjith Ba et al. explored text-based sentiment analysis for music recommendation, testing CNN, LSTM, CNN-LSTM, BiLSTM, and CNN-BiLSTM models [7]. Their results showed that CNN-BiLSTM achieved the highest classification accuracy (83.21%), proving that CNNs are effective in extracting spatial features, while LSTMs and BiLSTMs are better suited for sequential data analysis, such as Chatbot interactions and user reviews.

Another study by J. James Anto Arnold et al. proposed a music recommendation system based on facial expressions. Their model processed webcam video recordings by extracting frames and applying the Facial Action Coding System (FACS) to categorize emotions into Happy, Angry, Surprise, and Sad [8]. This approach demonstrated the potential for real-time emotion-based music recommendations.

B. Music Recommendation Systems

Shakirova et al. investigated collaborative filtering methods for music recommendation and compared them with AI-powered models. Their research found that deep learning-based recommendation systems (CNNs and LSTMs) outperform conventional collaborative filtering algorithms, offering more personalized and accurate suggestions [9].

Chiang et al. introduced a music emotion classification system using KBCS, NWFE, and SVM for feature extraction and classification. Their study analysed 35 key auditory characteristics, such as rhythm, pitch, and timbre, achieving high classification accuracies of 86.94% and 92.33% for Happy, Tense, Sad, and Tranquil emotions [10]. This highlights the importance of feature selection in emotion-aware music recommendations.

To improve recommendation algorithms on streaming platforms like YouTube, Netflix, and Amazon, Fessahye et al. developed an enhanced T-RECSYS model, trained on the Spotify RecSys Challenge dataset. Their model achieved 88% precision by integrating deep learning algorithms with collaborative filtering techniques, proving that AI-based recommendation systems are highly scalable across digital platforms [11].

C. Facial Emotion Recognition for Music Recommendation

Zhang et al. enhanced music recommendations by integrating deep learning-based facial recognition. Their CNN-based model, trained on the LFW dataset, achieved an 88.56% facial expression recognition rate, validating the effectiveness of facial emotion detection in music selection [12].

Kim et al. designed a personalized recommendation system using K-Means clustering on a self-collected dataset. Their approach classified music based on user preferences, achieving 74% recommendation accuracy by clustering songs into rock (34%) and classical (40%) genres [13]. This highlights the role of unsupervised learning in music recommendation.

Ayata et al. explored physiological signal-based music recommendations, using GSR and PPG signals from the Multimodal Deep Emotion Dataset. Their model achieved 71.53% (GSR) and 70.76% (PPG) accuracy, proving that biological signals can effectively predict emotional states for personalized music selection [14].

D. Limitations of Existing Works

While the aforementioned studies present promising advancements in emotion-based music recommendation systems, they still have certain limitations that hinder their real-world effectiveness.

One major limitation is the dependence on pre-recorded data. Many existing models rely on static datasets rather than real-time facial emotion recognition, which significantly reduces dynamic user interaction. Since emotions are highly context-dependent and can change rapidly, models that fail to adapt to real-time variations may not provide an engaging and personalized music recommendation experience.

Another drawback is the limited number of emotion categories used in classification. Some studies only consider four or five primary emotions, which can oversimplify the complexity of human emotions. In reality, emotions are nuanced and can exist in multiple intensities, such as mild sadness versus deep sorrow. A restricted classification system may lead to less accurate music recommendations that do not fully capture the user's mood.

Moreover, many sentiment-based approaches suffer from a lack of contextual awareness. Several studies rely solely on textual reviews or physiological signals like heart rate and skin conductance to determine mood. While these factors can provide some insights, they may not be as accurate and expressive as facial expression analysis, which directly reflects a user’s emotions.

Dataset imbalance is another common issue affecting model performance. Many studies use small or imbalanced datasets, where some emotions have significantly fewer samples than others. This imbalance often leads to misclassification, particularly for neutral or complex emotions. As a result, models may favor certain emotions over others, reducing the overall accuracy of emotion-based recommendations.

Lastly, generalization challenges remain a critical concern. Many deep learning models achieve high accuracy on controlled datasets, but they struggle with real-world variations such as different lighting conditions, facial angles, occlusions, and background noise. These variations can negatively impact model performance, making it less reliable when deployed in real-time applications.

Addressing these limitations is crucial for developing more robust, adaptable, and user-centric emotion-based music recommendation systems.

E. How Our Approach Differs

Our proposed model effectively addresses these limitations by integrating real-time emotion detection, ensuring that users receive dynamic music recommendations based on their live facial expressions rather than relying on pre-recorded images. This enhances user interaction and personalization, making the system more responsive to emotional changes.

To improve accuracy, we employ deep feature extraction using ResNet50V2, a more advanced architecture compared to VGG19 or shallow CNNs used in prior studies. By leveraging transfer learning, our model can extract richer facial features and enhance classification performance, even with limited training data.

Additionally, our system expands emotion classification by detecting a wider range of emotions, including Happy, Sad, Angry, Neutral, Fear, and Surprise. This allows for a more context-aware recommendation process, as opposed to models that classify emotions into only a few basic categories.

Unlike traditional recommendation systems that rely purely on collaborative filtering or content-based filtering, our model introduces a hybrid recommendation strategy that directly maps facial expressions to predefined music genres. This approach ensures that song recommendations align more closely with the user's emotional state, rather than just historical listening patterns.

Furthermore, our model is designed for scalability and real-time deployment, making it lightweight and optimized for mobile and web applications. This allows seamless integration into various platforms, enhancing accessibility and usability.

By addressing these challenges, our model provides a highly personalized and interactive music recommendation experience, setting it apart from existing systems.

III. BACKGROUND DETAILS

A. Convolutional Neural Networks

In their study, Shaha et al. said that CNN is well-known for its ability to extract information and features. CNN is a deep learning-based neural network architecture that has many practical uses in data visualization and picture interpretation with the aid of artificial intelligence. An improved type of artificial neural network, CNN offers more precise visual characteristics for improved classification [4]. According to CNN, each incoming image is handled as a matrix. The necessary data is then recovered from the resultant matrix (output picture), which is created by performing mathematical operations over the several matrices (input image).

A convolutional neural network (CNN) with five layers—the input layer, convolutional layer, pooling layer, fully connected layer, and output layer—is shown in Figure 1.

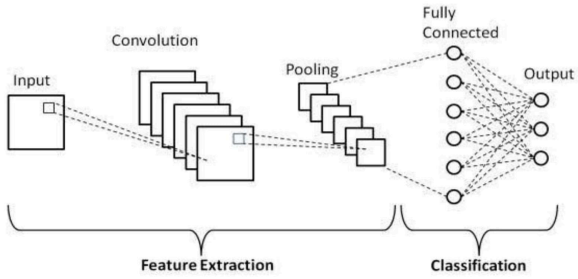

Figure. 1. CNN Architecture

1) Input layer: The input layer converts user-supplied pictures into matrices and forwards the matrix that results to the convolutional layer.

2) Convolutional Layer: The layer that appears following CNN's input layer is called the convolutional layer. In order to extract features from the supplied input images, mathematical operations are carried out in this initial layer. The convolution process is carried out over the input matrix, which consists of the filter (the AxB matrix, often referred to as the kernel, which is used to extract attributes) and matrix multiplication of the input picture. A feature map, which is created by multiplying the input by the kernel, contains the outcome. Two key terms that are used once the feature map is acquired are stride and padding.

3) Padding: In order to maintain spatial sizes during the convolution process, padding is the inclusion of extra pixels around the input image or feature map. It is essential to the construction and operation of convolutional neural networks and helps to reduce information loss at the edges.

4) Stride: As it interprets the picture, the computer needs to figure out how far the filter will have to travel across the picture in stride. It moves horizontally across the photo from the top left corner to the bottom right corner. Here, stride makes sure that the filter knows how many pixels (squares) it must skip in order to interpret the image. The number of features that the filter learns depends on the stride size. Due to extensive data extraction, more features are learned, as evidenced by the stride's lower size. Conversely, a larger stride size results in less data extraction and therefore fewer characteristics being learned.

$$\text{Size of Feature Map} = (B-A+1) \times (B-A+1)$$

Where,

B = The input matrix's row count
A = The input matrix's column count
$B \times B$ = The input matrix's size
$A \times A$ = the kernel/filter's size

5) Pooling Layer: In order to simplify mathematical calculations and save computation expenses, the pooling layer shrinks the matrix. It also helps to improve ConvNet's stability. In order to speed up calculation, the pooling layer helps to reduce the training parameters. In general, pooling operations can be divided into three categories: maximum, minimum, and average pooling. To summarize a chosen area of the image, max pooling selects the maximum feature values in that area. Min pooling summarizes the chosen area of the image by choosing the minimal feature values in that area. A region's average value is used in average pooling to calculate the sum of its characteristics.

6) Fully Connected Layer: This layer multiplies the weight matrix and input together and adds a bias vector. This layer does the job of linking neurons from the previous layer and fully connected layer. The calculation formula of a fully connected layer is indicated in Equation 1.

$$y_{jk}(x) = f\left(\sum_{i=1}^{nH} w_{jk} x_i + w_{j0}\right) \quad (1)$$

In this case,

W = Weight matrix
W_0 = Bias vector
X = Input matrix
Y = Output matrix

7) Output Layer: It is the final layer of the Convolutional Neural Network, whose task is to predict the final result by projecting the features learned from input images. The output from this layer classifies the emotion of a user, which may be any one of the emotions for which the machine is trained.

B. ResNet50V2

Residual Network 50 Version 2, or ResNet50V2, is a deep convolutional neural network (CNN) architecture that uses residual connections to improve feature extraction capabilities. ResNet designs were developed by He et al. to address the issue of vanishing gradients in deep networks, which frequently impede efficient training. With the help of a pre-activation residual block, ResNet50V2 enhances gradient flow and makes learning in very deep networks more effective than ResNet50.

ResNet50V2 maintains good accuracy while lowering computing complexity with its bottleneck architecture. It is especially useful for computer vision tasks like facial recognition and image classification because of its enhanced design.

The input layer, convolutional layers, residual blocks, global average pooling, and the output layer are the five primary parts of the ResNet50V2 architecture.

1) Input Layer: The input layer processes user-provided images and converts them into numerical matrices. These matrices are passed to the initial convolutional layers for feature extraction.

2) Convolutional Layers**:** Similar to standard CNNs, ResNet50V2 applies convolutional filters over the input matrix to extract important spatial and texture-based features. However, it enhances this process with pre-activation residual blocks, ensuring stable gradient propagation.

3) Residual Blocks**:** The core innovation in ResNet50V2 is the use of residual connections, allowing feature information to bypass layers and preventing degradation in performance. Each residual block consists of:

- 1×1 Convolution (dimensionality reduction)
- 3×3 Convolution (feature extraction)
- 1×1 Convolution (dimensionality restoration)
- Skip connection (identity mapping of the original input to the output of the residual block)

The advantage of residual blocks is that they help deep networks learn efficiently by allowing gradients to flow directly through the network during backpropagation.

4) Batch Normalization and Activation**:** Unlike ResNet50, where activation (ReLU) and batch normalization are applied after the convolutional layers, ResNet50V2 applies batch normalization and ReLU before the convolution operation. This modification leads to improved training stability and convergence.

5) Global Average Pooling (GAP) and Fully Connected Layer**:** Instead of using fully connected layers with high parameters, ResNet50V2 applies Global Average Pooling (GAP) to reduce overfitting and computational complexity. The GAP layer summarizes spatial feature maps, creating a low-dimensional feature vector, which is then passed through a fully connected layer for classification.

6) Output Layer: The output layer in ResNet50V2 classifies the input image into one of the predefined categories. In the context of facial emotion recognition, this layer maps facial expressions to specific emotion classes such as happy, sad, angry, surprised, and neutral. The softmax function is typically used to compute the final probability distribution across these categories.

The concept of residual learning in ResNet50V2 is mathematically represented as follows:

$$y=F(x,Wi)+x$$

where:

- x is the input to the residual block,
- Wi represents the convolutional layer weights,
- F(x,Wi) is the learned transformation (convolution, batch normalization, and activation),
- y is the final output of the residual block after summation.

This formulation enables efficient gradient flow, ensuring deeper networks do not suffer from the vanishing gradient problem.

IV. PROPOSED WORKS

The approach to creating the emotion-based song recommendation system commences with the collection and pre-processing of data. The accompanying flowchart (Figure 2) illustrates the comprehensive workflow of the proposed

project. Essential libraries were imported to manage datasets, conduct visualizations, and execute deep learning models. The dataset, which includes images labelled with emotions, was loaded in conjunction with a distinct music dataset that correlates moods with specific songs. To maintain data integrity, the distributions of classes were visualized, and sample images were presented. Subsequently, the dataset was divided into training and testing subsets, typically following an 80:20 ratio [14]. Pre-processing procedures, such as resizing images to 224x224 for ResNet50V2, normalizing pixel values (scaling them between 0 and 1), and applying data augmentation techniques (including rotation, flipping, and zooming), were implemented to enhance generalization.

In the course of model development, two distinct deep learning architectures were employed: a Convolutional Neural Network (CNN) and ResNet50V2. The CNN was designed with convolutional layers dedicated to feature extraction, pooling layers aimed at reducing dimensionality, and fully connected layers responsible for classification tasks. The model was compiled utilizing a suitable optimizer, such as Adam, along with a categorical cross-entropy loss function. To mitigate the risk of overfitting, call backs like Early Stopping and ReduceLROnPlateau were incorporated, and the training process was conducted over 50 epochs [15]. In a similar vein, the ResNet50V2 model, which is pre-trained, underwent fine-tuning by freezing its initial layers and training solely the upper layers, adhering to the same pre-processing, compilation, and training protocols as the CNN.

In the context of model evaluation and performance assessment, both the test loss and accuracy were analysed for the CNN and ResNet50V2 architectures. To gain deeper insights into their performance, graphical representations were created to illustrate the relationship between test loss and epochs, as well as test accuracy and epochs. Additionally, a Confusion Matrix was constructed to evaluate the effectiveness of the models in classifying various emotions.

After classifying emotions, the system established a mechanism for recommending songs based on these emotional insights. The music dataset underwent processing, allowing for the predicted emotions to be aligned with songs that correspond to specific moods [16]. Subsequently, the five most popular songs, arranged in descending order of their popularity, were presented in relation to the identified emotion.

To facilitate real-time inference, the system incorporated a feature for immediate predictions. New images were uploaded, pre-processed, and formatted for input into the model, utilizing both CNN and ResNet50V2 to ascertain emotional states.

The resulting labels and their corresponding confidence scores were presented. Ultimately, both models were preserved for subsequent use and deployment, guaranteeing their readiness for real-time applications. This organized methodology guarantees an effective emotion recognition system that is seamlessly integrated with a customized music recommendation feature [17].

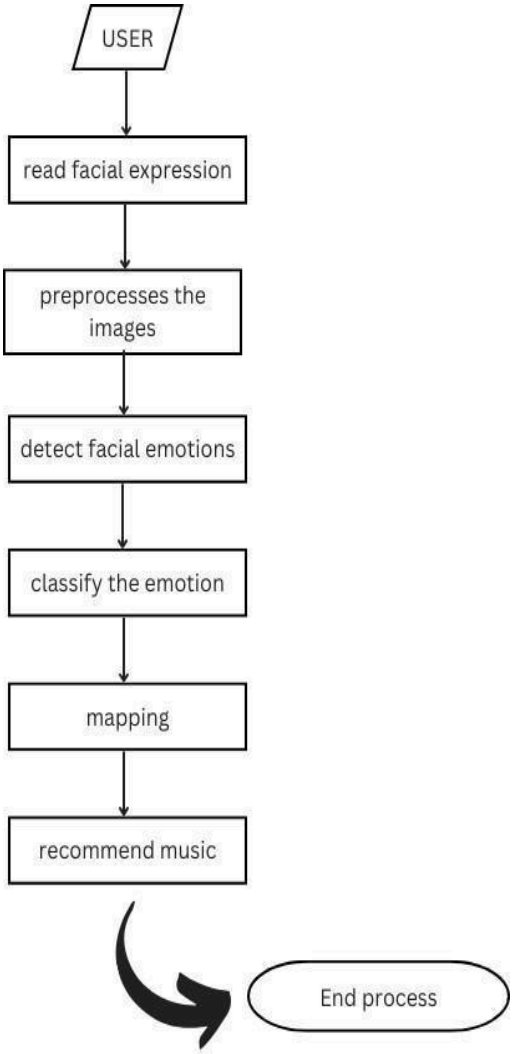

Figure. 2. Detailed Working of the proposed model

V. IMPLEMENTATION

A. Dataset

FER2013 is a popular standard collection of facial expression recognition that was used to train the model. 7 emotions are included in the dataset, as indicated in Table I: anger, disgust, fear, happiness, neutrality, sadness, and surprise. Thousands of 48x48 pixel grayscale photos taken from actual situations are included in each emotion category.

The model's capacity for generalization and detection rate are enhanced by learning it using the FER (2013) dataset, which features an abundance of expressions on faces gathered in randomized settings [18]. The program can efficiently categorize sensations in realistic-time by employing the data at hand, which makes it resilient and flexible for emotion-based applications in the real world.

TABLE I
EMOTIONS DETAILS

Emotions	Training Images	Testing Images
Happy	7215	1774
Sad	4830	1247
Neutral	4965	1233
Angry	3995	958
Fear	4097	1024
Disgust	436	111
Surprise	3171	831

B. Steps for Data Collection

Step 1: Importing Required Libraries

To efficiently handle data and visualization, essential libraries such as NumPy, Pandas, Matplotlib, and Seaborn are imported. NumPy and Pandas facilitate numerical computations and data manipulation, while Matplotlib and Seaborn enhance data visualization. For image processing tasks, OpenCV (cv2) and ImageDataGenerator are utilized, enabling real-time data augmentation and preprocessing. TensorFlow and Keras serve as the backbone for implementing deep learning models, providing a robust framework for training and evaluation. Additionally, optimization techniques like EarlyStopping, ModelCheckpoint, and ReduceLROnPlateau are incorporated to prevent overfitting, save the best model, and dynamically adjust the learning rate, ensuring improved model performance.

Step 2: Loading the Dataset

The emotion-based picture dataset, which contains tagged images for different emotions, is loaded initially. The emotion recognition model will be trained using the dataset, which consists of facial expressions categorized into several emotion classes. In order to allow mood-based song suggestion, the music dataset is also loaded, mapping different moods to corresponding songs. After the datasets have been loaded, a summary of their makeup is shown, including the number of emotion classes, sample size for each class, and overall dataset distribution. By doing this, any uncertainty about the data before pre-processing and model training is eliminated.

Step 3: Visualizing Dataset

A bar graph is created to display the quantity of photos in each class of emotion in order to visualize the dataset's distribution. This makes it easier to identify class disparities and ensure that training data is appropriately represented. To verify that the annotations are accurate, a few example photographs are also randomly selected from the collection and displayed with the appropriate labels. This procedure guarantees that the dataset is appropriately organized and prepared for the emotion recognition model to be trained successfully.

Step 4: Splitting the Dataset

For effective model training and testing, the dataset is split into training and testing sets in an 80:20 ratio (or a preferred ratio). While the testing set is used to evaluate the deep learning model's performance on fresh data, the training set is used to train the model. An apparent picture of the data distribution between training and testing sets is provided by concatenating the number of samples in each set after splitting and presenting the results in a well-organized table manner. A well-prepared and balanced dataset for model building is guaranteed by this procedure.

Step 5: Data Pre-processing

Pre-processing the training data involves resizing all photos to a standard size (e.g., 224x224 for ResNet50V2) to ensure consistency across the dataset. To help the model converge, pixel values are normalized by scaling them between 0 and 1. Additionally, data augmentation techniques like flipping, zooming, and rotation are employed to fictitiously expand the dataset's size and boost the model's capacity to generalize to data that hasn't been seen before. Images are downsized to a consistent fixed size with the training data for pre-processed test data in order to make them compatible with the model. Normalizing pixel values improves forecast accuracy and consistency. For the best model performance and efficient training and testing, all of these pre-processing operations are important.

Step 6: Loading Processed Data

Images are first converted into NumPy arrays to enable TensorFlow/Keras compatibility and efficient computation, thereby preparing the dataset for deep learning model training. To make sure they are prepared for model training and testing, both the training and test datasets are loaded into memory. Finally, to make sure the dataset has been processed correctly and has the desired dimensions for picture inputs and labels, the morphologies of the training and testing data are examined. This is to guarantee that the data is appropriately organized prior to being incorporated into the deep learning model.

Step 7: Model Training Data Collection

The pre-processed images and labels are retained in variables X_train and y_train, the input features and target labels used for training. A validation split of the training data

is set up to track model performance during training, which can prevent overfitting and facilitate generalization.
Following data preparation, a dataset summary is printed out with the class distribution and shape of X_train and y_train as well as the validation set. This is done to ensure that the dataset is properly structured and ready for training deep learning models.

Step 8: Training CNN and ResNet50V2 Models

In constructing the CNN model, there are five implemented layers, i.e., three convolutional blocks and two fully connected layers:

- ❖ CNN1 - First Convolutional Block: It has a convolutional layer, activation function (ReLU), and max-pooling in order to learn low-level features.
- ❖ CNN2 - Second Convolutional Block: Again, extracts complex features by employing a different set of convolutional and pooling layers.
- ❖ CNN3 - Third Convolutional Block: Expands the feature extraction procedure to make it deeper so the model can extract sophisticated patterns.
- ❖ Fully Connected Layer: Flattens the extracted features and feeds them through dense layers to classify.
- ❖ Output Layer: Employing a softmax activation function to classify images into various emotion categories.

The model is then compiled with a suitable loss function (e.g., categorical cross-entropy), optimizer (e.g., Adam), and evaluation metrics (e.g., accuracy). Call-backs like EarlyStopping, ModelCheckpoint, and ReduceLROnPlateau are used to avoid overfitting and improve model performance.
The CNN model is trained for 50 epochs and the training history is saved in order to see how the performance of the models varies. Lastly, ResNet50V2 is utilized employing transfer learning by unfreezing the topmost layers of the pre-trained model and retraining it on the emotion dataset. This method applies the strength of deep feature extraction but enables specific tuning for the task of emotion classification.

Step 9: Model Evaluation & Performance Analysis

Both the CNN model and the ResNet50V2 are trained and evaluated using the test data in order to gauge the models' performance. To determine whether the models can generalize to the unseen data, crucial metrics like accuracy and loss are computed. The following graphs are created in order to visually analyse the performance:

- Test Loss vs. Epoch: Shows how the loss changes over the epochs, pointing out trends of under fitting or overfitting.
- Test correctness vs. Epoch: Displays a trend in model correctness that reflects stability and convergence.

In addition, the categorization performance is examined by calculating and displaying a Confusion Matrix. Class-wise accuracy, misclassifications, and prediction bias are all disclosed by the confusion matrix. It guarantees a thorough examination of the models before they are used..

Step 10: Prediction & Mapping Indices to Emotion Classes

The following actions are taken in order to apply the models on fresh, unseen images:

- The new photos should be loaded, resized to 224x224, and normalized in accordance with the training data format.
- Make Forecasts: Both the CNN model and ResNet50V2 predict the emotions of the new photos.
- Display Results: To allow for a direct comparison of model performance on unseen data, the predicted labels are displayed alongside the actual class labels.

This procedure helps assess how well the models generalize in practical situations.

Step 11: Mapping Emotion Predictions to Music Dataset

To combine the music recommendation system with the emotion recognition model, the dataset for mood-based song recommendations is loaded and pre-processed. The dataset consists of song names, artists, and popularity scores corresponding to various emotions. Depending on the predicted emotion, the system pulls the top five songs based on popularity in descending order, guaranteeing the most appropriate recommendations. Lastly, the suggested songs for the identified mood appear, enabling users to benefit from customized music recommendations compatible with their mood.

Step 12: Saving the Model for Future Use

The trained ResNet50V2 and CNN model are saved for deployment so that they can be utilized for real-time inference. The models are saved in the HDF5 (.h5) format or the TensorFlow SavedModel format, maintaining the learned weights and architecture. The saved models are then loaded to check their integrity to ensure they work properly for real-time emotion detection and recommendation of music. This is an important step for successful deployment and real-world application of the system.

C. Proposed Steps

1. Import libraries required

NumPy, Pandas, Matplotlib, and Seaborn are essential Python libraries for data manipulation, analysis, and visualization. NumPy provides support for large, multi-dimensional arrays and matrices, while Pandas

simplifies data handling with its powerful DataFrame structure. Matplotlib and Seaborn enable comprehensive data visualization, making it easier to interpret trends and patterns. OpenCV is widely used for image processing and computer vision tasks, allowing efficient manipulation and analysis of images. TensorFlow and Keras facilitate deep learning model development, offering robust tools for neural network training and deployment. Additionally, ImageDataGenerator aids in data augmentation to enhance model performance, while EarlyStopping and ModelCheckpoint optimize training by preventing overfitting and saving the best-performing model, respectively.

2. Data Pre-processing and Visualization

To begin with, we load the dataset and print the train-test split statistics to understand the distribution of data. Next, we visualize the class distribution of emotions to check for any imbalances in the dataset. Finally, we display sample images from the dataset to gain insights into the data quality and structure before proceeding with model training.

3. Prepare Data for Model Training

To prepare the dataset for model training, we first perform image pre-processing, including resizing the images to a uniform shape and normalizing pixel values for better model performance. Next, we convert the images into arrays and apply label encoding to transform categorical labels into numerical values. The dataset is then split into training and testing sets to evaluate model performance effectively. Finally, we load and preprocess the images to ensure they are in the correct format and ready for input into the deep learning model.

4. Build and Train a CNN Model

To build the Convolutional Neural Network (CNN) for image classification, we start by defining the architecture. The model consists of three convolutional blocks:

- **Conv Block 1:** The first convolutional layer extracts low-level features such as edges and textures.
- **Conv Block 2:** The second convolutional layer captures more complex patterns and spatial hierarchies.
- **Conv Block 3:** The third convolutional layer further refines feature extraction.
- **Fully Connected Layers:** These layers process the extracted features and produce the final classification output.

Next, we compile the model by selecting an appropriate optimizer, loss function, and evaluation metrics while setting key hyper parameters such as the learning rate and batch size. To enhance model performance and prevent overfitting, we implement callback functions like EarlyStopping (to halt training when no improvement is observed) and ModelCheckpoint (to save the best model). The CNN model is then trained for 50 epochs using the pre-processed dataset. Finally, we evaluate the model's performance on the test set by analysing test loss and test accuracy to assess its generalization capability.

5. Visualize Model Performance

To analyse the model's performance, we start by plotting loss vs. epochs and accuracy vs. epochs to visualize the training and validation trends over time. These plots help identify issues like overfitting or under fitting by showing how the loss decreases and accuracy improves during training. Next, we generate a confusion matrix to evaluate the model's emotion classification performance. The confusion matrix provides detailed insights into the model's predictions, highlighting the number of correct and incorrect classifications for each emotion category. This allows us to assess class-wise performance and identify any misclassifications that may need further improvements in the model.

6. Implement ResNet50V2 for Transfer Learning

To enhance emotion classification performance, we utilize ResNet50V2 as a feature extractor, leveraging its pre-trained weights to extract deep features from images. Initially, all layers except the last 50 layers are frozen to retain learned features while allowing fine-tuning on the target dataset. Next, we modify the ResNet50V2 architecture by adding a custom classification head tailored for emotion recognition. This involves adding fully connected layers and an activation function suited for multi-class classification. After modifying the architecture, we fine-tune the model by training it for 30 epochs, adjusting only the unfrozen layers to learn task-specific features. Finally, we evaluate the model's test accuracy to measure its effectiveness in emotion classification, comparing it to the CNN model to assess performance improvements.

7. Make Predictions on New Images

For inference, we start by loading the processed images, ensuring they are pre-processed in the same way as the training data (resized, normalized, and converted into arrays). Next, we perform predictions using the trained CNN model, feeding the images into the model to obtain output probabilities for different emotion classes. Finally, we predict new emotion-based classes by selecting the class with the highest probability for each image. This step allows the model to classify unseen images into corresponding emotions, enabling real-world application of emotion recognition.

8. Music Recommendation Based on Predicted Emotion

Once the model detects an emotion, we proceed to display the top 5 song recommendations based on the predicted emotion. These recommendations are selected from

a curated playlist that aligns with the detected mood, ensuring a personalized music experience. Next, we fetch relevant music dataset previews, which may include song titles, artists, and album covers, providing users with a visual and textual overview of the recommended tracks. This enhances the user experience by making the emotion-based music recommendation system more interactive and engaging.

9. Model Saving

To preserve the trained model for future use, we **save it in** .h5 **format** using TensorFlow/Keras. This allows the model to be reloaded later for inference or further fine-tuning without retraining from scratch. Saving the model ensures that all learned weights, architectures, and configurations are stored efficiently, making it easy to deploy the emotion-based music recommendation system whenever needed.

VI. RESULT ANALYSIS

A. Training and Testing Accuracy & Loss

The accuracy and loss metrics for both training and testing phases offer a comprehensive understanding of the model's performance and its learning capabilities. Throughout the training process, there was a noticeable increase in accuracy alongside a steady decline in loss, suggesting that the model successfully grasped the dataset's characteristics.

In the testing phase, the model's proficiency in generalizing to new, unseen data was assessed, with high test accuracy and low test loss serving as indicators of its reliability. The visual representations of loss versus epochs and accuracy versus epochs illustrated a consistent learning trajectory, which minimized the risk of overfitting [19]. Additionally, the implementation of call-backs and regularization strategies contributed to enhancing the model's performance, rendering it particularly effective for emotion-based music recommendation tasks.

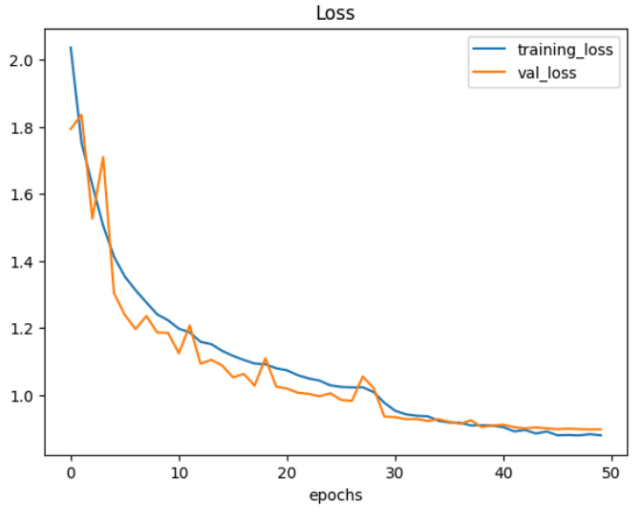

Figure. 3. CNN Model Loss

Convolutional neural networks (CNNs) and their model loss are depicted in Figure 3, which also compares the model's training and validation losses. The figure's X-axis represents the epochs, while the Y-axis shows the values of the training loss. Successful learning and little overfitting are indicated by the CNN model's loss curves, which show a steady decreasing trend.

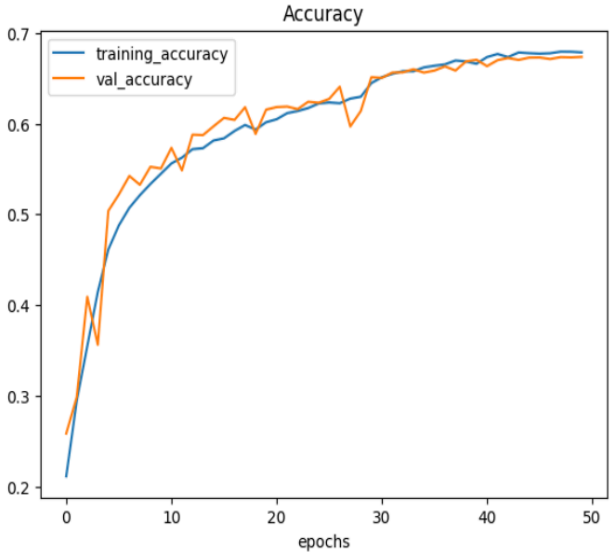

Figure. 4. CNN Model Accuracy

By comparing the training and validation accuracy metrics, Figure 4 shows the convolutional neural network's (CNN) accuracy. The figure's Y-axis shows the training accuracy, while the X-axis represents the epochs. The CNN model's accuracy trajectories exhibit a steady rising trend over the course of the epochs, suggesting that it is capable of learning and generalizing. The model's reliability in identifying emotions for music recommendation is confirmed by the close proximity of the training and validation accuracy curves, which suggests a low degree of overfitting.

Figure 5 illustrates the loss analysis for the ResNet50V2 model, detailing the training and validation loss metrics across various epochs. The horizontal axis represents the epochs, while the vertical axis indicates the loss values. At the outset, there is a significant decline in loss, reflecting a phase of rapid learning, which is subsequently followed by a more gradual decrease. The similarity between the training and validation loss indicates that the model is not experiencing overfitting, thereby demonstrating a well-balanced learning process that contributes to its reliability in predicting facial emotions for music recommendation.

Figure 6 presents the accuracy metrics of the ResNet50V2 model, showcasing a comparison between training and validation accuracy across various epochs. The horizontal axis denotes the number of epochs, while the vertical axis indicates the accuracy levels for both training and validation datasets. The accuracy trajectories reveal a consistent learning pattern, with training accuracy showing a steady upward trend.

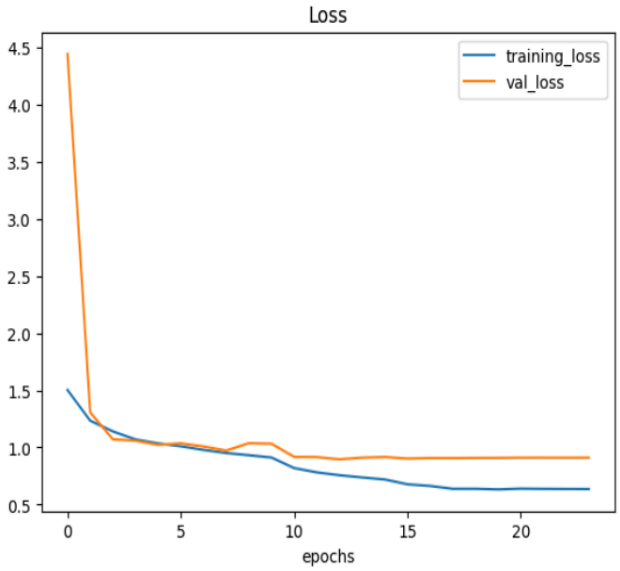

Figure. 5. ResNet50v2 Model Loss

Meanwhile, the validation accuracy reaches a plateau after a certain threshold, indicating the model's proficiency in generalizing to new, unseen data. The narrow margin between training and validation accuracy points to minimal overfitting, underscoring the model's capability in effectively classifying emotions within the music recommendation framework.

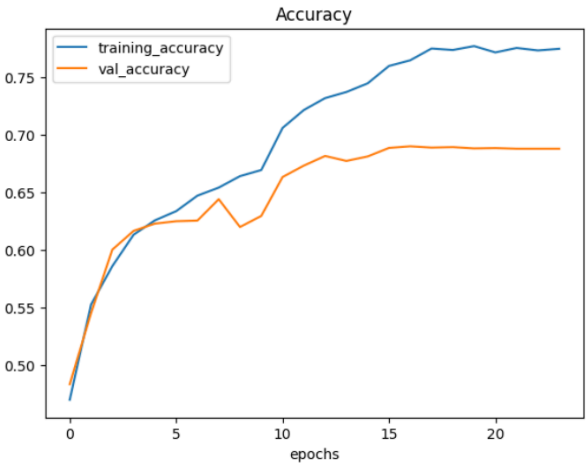

Figure. 6. ResNet50v2 Model Accuracy

B. Confusion Matrix

Figure 7 displays the confusion matrix associated with the CNN model, which demonstrates its classification efficacy across various emotion categories. The horizontal axis denotes the predicted labels, while the vertical axis reflects the actual labels. Values along the diagonal represent instances that have been accurately classified for each category, whereas the off-diagonal entries indicate instances of misclassification.

A greater concentration of values along the diagonal suggests that the model performs well in identifying specific emotions, while the existence of off-diagonal values points to potential areas for enhancement. The color gradient within the matrix illustrates the distribution of predictions, with lighter hues indicating higher frequencies. This confusion matrix offers valuable insights into the model's capabilities and highlights the challenges it faces in terms of misclassification.

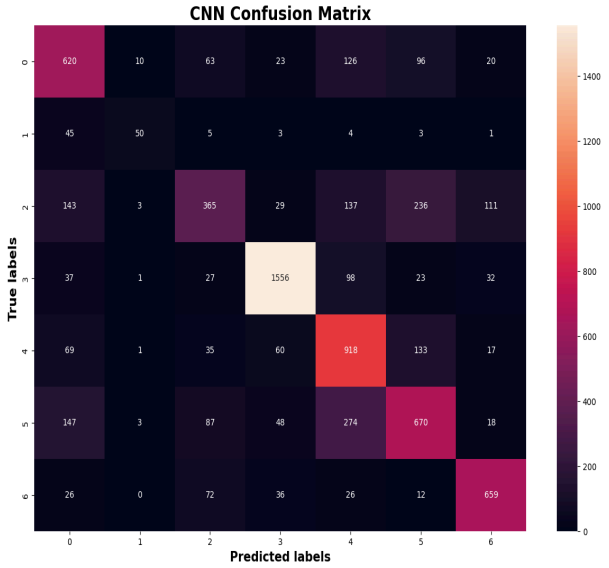

Figure. 7. CNN Confusion Matrix

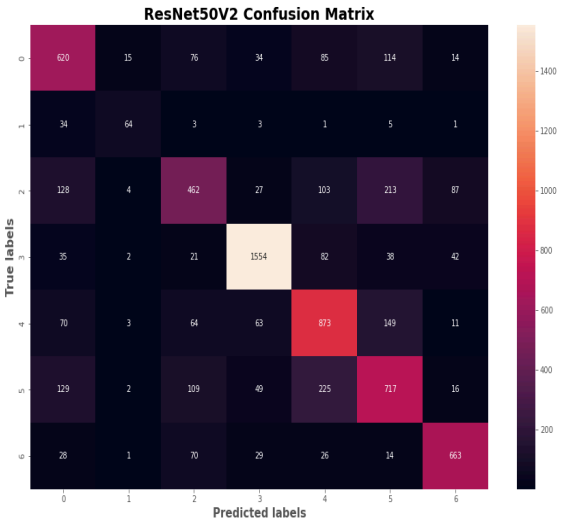

Figure. 8. ResNet50V2 Confusion Matrix

Figure 8 illustrates the confusion matrix associated with the ResNet50V2 model, highlighting its effectiveness in classifying various emotion categories. The horizontal axis denotes the predicted labels, while the vertical axis reflects the actual labels. Correct classifications are represented by the diagonal elements, whereas the off-diagonal elements indicate instances of misclassification.

The colour intensity within the matrix emphasizes the frequency of predictions, with lighter hues signifying greater occurrences. This analysis indicates that while the model excels in recognizing certain emotions, it still experiences some misclassifications. Such a visualization offers critical insights into the capabilities and limitations of the ResNet50V2 model in the context of emotion classification for music recommendation.

C. Music Prediction

Figure 9 illustrates a music recommendation system that utilizes facial emotion detection technology. In the image, the identified face is marked with a green bounding box, and the system has classified the emotion as ”Angry.” Above the image, a table presents five song recommendations that are linked to a calm mood, indicating that the system's objective is to create an emotional equilibrium by suggesting music that counteracts the observed emotional state.

	name	artist	mood	popularity
0	Lost	Annelie	Calm	64
1	Curiosity	Beau Projet	Calm	60
2	Escaping Time	Benjamin Martins	Calm	60
3	Just Look at You	369	Calm	59
4	Vague	Amaranth Cove	Calm	59

Figure. 9. ResNet50V2 Music Recommendation Prediction

Figure 10 illustrates a comparable configuration in which the system has identified the emotion as ”Happy.” Once more, the recognized face is highlighted with a green bounding box. The preceding table enumerates five songs that correspond to a joyful mood, suggesting that the system proposes music that is in harmony with the user's present emotional condition.

	name	artist	mood	popularity
0	Pumped Up Kicks	Foster The People	Happy	84
1	Africa	TOTO	Happy	84
2	Take on Me	a-ha	Happy	84
3	Highway to Hell	AC/DC	Happy	83
4	Here Comes The Sun - Remastered 2009	The Beatles	Happy	83

Figure. 10. CNN Music Recommendation Prediction

VII. COMPARATIVE ANALYSIS

In this segment, we analyze the efficacy of Convolutional Neural Network (CNN) and ResNet50V2 in the context of facial emotion recognition, focusing on their predictive outcomes. The findings indicate that ResNet50V2 outperformed CNN in terms of accuracy when identifying facial expressions. A significant pattern of misclassification was noted, particularly with neutral faces frequently being incorrectly identified as sad or fearful, which suggests that the model struggles with differentiating between nuanced emotional cues [19]. Furthermore, instances of surprise were occasionally confused with happiness, likely due to the similarities in facial features associated with these emotions.

Conversely, both models exhibited commendable performance in accurately recognizing anger and fear, demonstrating their capability in detecting more pronounced emotional expressions [20].

A more in-depth examination indicates that CNN encountered greater difficulties in accurately identifying neutral, sad, and fearful emotions, resulting in a higher rate of misclassifications. In contrast, ResNet50V2 demonstrated a lower error rate, which implies superior feature extraction and generalization capabilities attributed to its more complex architecture. Both models excelled in detecting happy and angry expressions, as these emotions are characterized by clearly defined facial features. Ultimately, ResNet50V2 surpasses CNN in performance, owing to its deeper architecture and the advantages of pre-trained weights that facilitate enhanced feature extraction and classification [21]. Conversely, CNN's elevated error rate in recognizing nuanced emotions highlights the necessity for improved training methodologies or the implementation of data augmentation strategies.

This comparative study emphasizes the benefits of employing ResNet50V2 in contrast to a conventional CNN, thereby underscoring its effectiveness for facial emotion recognition. Additionally, a comprehensive confusion matrix or performance table could further substantiate these conclusions [22].

VIII. CONCLUSION

The research effectively executed and contrasted a Convolutional Neural Network (CNN) with ResNet50V2 for the purpose of recommending music based on facial emotion recognition. The dataset underwent preprocessing, and both models were subjected to rigorous training and testing protocols, which encompassed data augmentation, model compilation, and the use of callbacks to improve performance and mitigate overfitting [23]. The evaluation of the models was carried out through the analysis of test accuracy, loss graphs, and confusion matrices to evaluate their classification effectiveness. The CNN model, while relatively straightforward, yielded encouraging outcomes, showcasing its proficiency in effectively identifying facial features [24]. In contrast, ResNet50V2, which is a more complex and pre-trained architecture, displayed superior performance attributed to its enhanced feature extraction abilities. Analyzing the predictions from both models revealed their respective advantages and limitations. A music recommendation system was developed by examining emotion predictions and correlating them with a dataset that includes various song attributes. This system effectively identified and suggested the five most suitable songs based on their mood and popularity, showcasing its practical utility. Additionally, the trained models were preserved for future use, facilitating smooth deployment in real-world scenarios [25]. This project underscores the efficacy of utilizing deep learning techniques for facial emotion recognition in the development of personalized music recommendation systems. Prospective improvements may involve refining the models to achieve greater accuracy, broadening the dataset to enhance generalization capabilities, and incorporating real-time emotion recognition to facilitate dynamic recommendations [26].

REFERENCES

[1] Hongli Zhang, Alireza Jolfaei, and Mamoun Alazab, "A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing," IEEE Access, vol. 7, pp. 159081-159089, 2019.

[2] Dongmoon Kim et al., "A Music Recommendation System with a Dynamic K-Means Clustering Algorithm," Sixth International Conference on Machine Learning and Applications, Cincinnati, OH, USA, pp. 399403, 2007.

[3] Deger Ayata, Yusuf Yaslan, and Mustafa E. Kamasak, "Emotion Based Music Recommendation System Using Wearable Physiological Sensors," IEEE Transactions on Consumer Electronics, vol. 64, no. 2, pp. 196-203, 2018.

[4] Wei Chun Chiang, Jeen Shing Wang, and Yu Liang Hsu, "A Music Emotion Recognition Algorithm with Hierarchical SVM Based Classifiers," 2014 International Symposium on Computer, Consumer and Control, Taichung, Taiwan, pp. 1249-1252, 2014.

[5] M P, Sunil & ., Hariprasad S A. (2023). Facial Emotion Recognition using a Modified Deep Convolutional Neural Network Based on the Concatenation of XCEPTION and RESNET50 V2. International Journal of Electrical and Electronics Engineering Research. 10. 94-105. 10.14445/23488379/IJEEE-V10I6P110.

[6] Sriraj Katkuri, Mahitha Chegoor, Dr. K. C. Sreedhar, M. Sathyanarayana, 2023, Emotion Based Music Recommendation System, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 12, Issue 05 (May 2023)

[7] S. Madderi, S. Ponnaiyan, M. Subramanian, and K. Thulasingam, "A new mining and decoding framework to predict expression of opinion on social media emoji's using machine learning models," IAES International Journal of Artificial Intelligence, vol. 13, no. 4, pp. 5005–5012, Dec. 2024.

[8] N. K. E. and J. K., "Class based dynamic feature centric data deduplication scheme for efficient mitigation of side channel attack in cloud," J. Electr. Eng. Technol., vol. 19, no. 3, pp. 1933–1942, Mar. 2024.

[9] D. D. G., N. Kumar, J. M., and R. V., "Innovative brain tumor detection: Stacked random support vector-based

hybrid gazelle coati algorithm," Biomed. Signal Process. Control, vol. 101, p. 107156, 2025.

[10] R. V., J. S. Manoharan, R. Hemalatha, and D. Saravanan, "Deep learning models for multiple face mask detection under a complex big data environment," Procedia Comput. Sci., vol. 215, pp. 706–712, 2022.

[11] J. Venkatesh, K. S. Kumari, V. Rekha, N. Geethanjali, S. K. Rajesh Kanna, and K. Sivakumar, "Transformer models; Capsule neural networks; Electric networks; Wavelet transforms; Error rates," Library of Progress-Library Sci. Inf. Technol. Comput., vol. 44, no. 3, p. 12685, 2024.

[12] K.M. Aswin et al., "HERS:Human Emotion Recognition System," 2016 International Conference on Information Science, Kochi, India, pp. 176179, 2016.

[13] Shlok Gilda et al., "Smart Music Player Integrating Facial Emotion Recognition and Music Mood Recommendation," 2017 International Conference on Wireless Communications, Signal Processing and Networking, Chennai, India, pp. 154-158, 2017.

[14] Ashish Tripathi, Abhijat Mishra, Rajnesh Singh, Bhoopendra Dwivedy, Amit Kumar, Kuldeep Singh, "Facial Emotion-Based Song Recommender System Using CNN," International Journal of Engineering Trends and Technology, vol. 72, no. 6, pp. 315-327, 2024.

[15] R Prasanna, M Jenath, M Vinoth, J Joseph Ignatious, M S Maharajan, P Banu Priya, "Enhanced blood prothrombin time detection deploying flexible substrate UWB antenna from artifacts removed pure plasma through statistical multiple regression modelling" , Computers and Electrical Engineering, Volume 122, 2025, 109963, ISSN 0045-7906.

[16] Manali Shaha, and Meenakshi Pawar, "Transfer Learning for Image Classification," 2018 Second International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, pp. 656660, 2018.

[17] K.S. Krupa et al., "Emotion Aware Smart Music Recommender System Using Two Level CNN," 2020 Third International Conference on Smart Systems and Inventive Technology, Tirunelveli, India, pp. 1322-1327, 2020.

[18] Jamdar, A., Abraham, J., Khanna, K., & Dubey, R. (2015). Emotion Analysis of Songs Based on Lyrical and Audio Features. International Journal of Artificial Intelligence & Applications, 6(3), 35–50.

[19] Pettijohn, T. F., Williams, G. M., & Carter, T. C. (2010). Music for the Seasons: Seasonal Music Preferences in College Students. Current Psychology, 29(4), 328–345.

[20] E. Shakirova, "Collaborative filtering for music recommender system," 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017, pp. 548-550,

[21] T Tulasi Sasidhar, Premjith B, Soman K P, "Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text", Procedia Computer Science,Volume 171,2020, Pages 1346-1352,ISSN 1877-0509.

[22] Davis Moswedi, and Ritesh Ajoodha, "Music Classification Using Fourier Transform and Support Vector Machines," 2022 International Conference on Engineering and Emerging Technologies, Kuala Lumpur, Malaysia, pp. 1-4, 2022.

[23] E. Jing, Y. Liu, Y. Chai, S. Yu, L. Liu, Y. Jiang, and Y. Wang, "Emotion-Aware Personalized Music Recommendation with a Heterogeneity-Aware Deep Bayesian Network," 2024, arXiv:2406.14090. [Online]. Available: https://arxiv.org/abs/2406.14090

[24] X. Chang, X. Zhang, H. Zhang, and Y. Ran, "Music Emotion Prediction Using Recurrent Neural Networks," 2024, arXiv:2405.06747. [Online]. Available: https://arxiv.org/abs/2405.06747

[25] T. Babu, R. R. Nair, and G. A., "Emotion-Aware Music Recommendation System: Enhancing User Experience Through Real-Time Emotional Context," 2023, arXiv:2311.10796. [Online]. Available: https://arxiv.org/abs/2311.10796

APPENDIX

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

plt.style.use('default')

import os

import tensorflow as tf

import keras

import cv2

from sklearn.model_selection import train_test_split

from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img,
img_to_array

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint,
ReduceLROnPlateau

from tensorflow.keras.utils import plot_model

from tensorflow.keras import layers , models, optimizers

from tensorflow.keras.models import Sequential, Model

from tensorflow.keras.layers import *

from tensorflow.keras.applications import ResNet50V2


#visualizing the classes

train_dir = '../input/fer2013/train/'

test_dir = '../input/fer2013/test/'

def Classes_Count( path, name):

    Classes_Dict = {}

    for Class in os.listdir(path):

        Full_Path = path + Class

        Classes_Dict[Class] = len(os.listdir(Full_Path))

    df = pd.DataFrame(Classes_Dict, index=[name])
```

```
    return df
Train_Count = Classes_Count(train_dir,
'Train').transpose().sort_values(by="Train", ascending=False)
Test_Count = Classes_Count(test_dir,
'Test').transpose().sort_values(by="Test", ascending=False)
pd.concat([Train_Count,Test_Count] , axis=1)
Train_Count.plot(kind='barh')
Test_Count.plot(kind='barh')
plt.style.use('default')
plt.figure(figsize = (25, 8))
image_count = 1
BASE_URL = '../input/fer2013/train/'
for directory in os.listdir(BASE_URL):
    if directory[0] != '.':
        for i, file in enumerate(os.listdir(BASE_URL + directory)):
            if i == 1:
                break
            else:
                fig = plt.subplot(1, 7, image_count)
                image_count += 1
                image = cv2.imread(BASE_URL + directory + '/' + file)
                plt.imshow(image)
                plt.title(directory, fontsize = 20)
```

#Data preprocessing

```
img_shape = 48
batch_size = 64
train_data_path = '../input/fer2013/train/'
```

```
test_data_path = '../input/fer2013/test/'
train_preprocessor = ImageDataGenerator(
        rescale = 1 / 255.,
        # Data Augmentation
        rotation_range=10,
        zoom_range=0.2,
        width_shift_range=0.1,
        height_shift_range=0.1,
        horizontal_flip=True,
        fill_mode='nearest',
    )
test_preprocessor = ImageDataGenerator(
    rescale = 1 / 255.,
)
train_data = train_preprocessor.flow_from_directory(
    train_data_path,
    class_mode="categorical",
    target_size=(img_shape,img_shape),
    color_mode='rgb',
    shuffle=True,
    batch_size=batch_size,
    subset='training',
)
test_data = test_preprocessor.flow_from_directory(
    test_data_path,
    class_mode="categorical",
    target_size=(img_shape,img_shape),
```

```
    color_mode="rgb",
    shuffle=False,
    batch_size=batch_size,
)

#Building CNN model
def Create_CNN_Model():
    model = Sequential()
    #CNN1
    model.add(Conv2D(32, (3,3), activation='relu', input_shape=(img_shape,
img_shape, 3)))
    model.add(BatchNormalization())
    model.add(Conv2D(64,(3,3), activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2), padding='same'))
    model.add(Dropout(0.25))
    #CNN2
    model.add(Conv2D(64, (3,3), activation='relu', ))
    model.add(BatchNormalization())
    model.add(Conv2D(128,(3,3), activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2), padding='same'))
    model.add(Dropout(0.25))
    #CNN3
    model.add(Conv2D(128, (3,3), activation='relu'))
    model.add(BatchNormalization())
    model.add(Conv2D(256,(3,3), activation='relu', padding='same'))
```

```
    model.add(BatchNormalization())
    model.add(MaxPooling2D(pool_size=(2,2), padding='same'))
    model.add(Dropout(0.25))
    #Output
    model.add(Flatten())
    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(512, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))

    model.add(Dense(256, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(128, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(64, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(32, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.25))
    model.add(Dense(7,activation='softmax'))
    return model
CNN_Model = Create_CNN_Model()
```

```
CNN_Model.summary()

CNN_Model.compile(optimizer="adam", loss='categorical_crossentropy',
metrics=['accuracy'])


#Specifying Callbacks

# Create Callback Checkpoint

checkpoint_path = "CNN_Model_Checkpoint"

Checkpoint = ModelCheckpoint(checkpoint_path, monitor="val_accuracy",
save_best_only=True)

# Create Early Stopping Callback to monitor the accuracy

Early_Stopping = EarlyStopping(monitor = 'val_accuracy', patience = 15,
restore_best_weights = True, verbose=1)

# Create ReduceLROnPlateau Callback to reduce overfitting by decreasing
learning rate

Reducing_LR = tf.keras.callbacks.ReduceLROnPlateau( monitor='val_loss',
factor=0.2, patience=2,#min_lr=0.000005,verbose=1)

callbacks = [Early_Stopping, Reducing_LR]

steps_per_epoch = train_data.n // train_data.batch_size

validation_steps = test_data.n // test_data.batch_size

CNN_history = CNN_Model.fit( train_data , validation_data= test_data ,
epochs=50, batch_size= batch_size, callbacks=callbacks, steps_per_epoch=
steps_per_epoch, validation_steps=validation_steps)


#Evaluating CNN Model

CNN_Score = CNN_Model.evaluate(test_data)

print("    Test Loss: {:.5f}".format(CNN_Score[0]))

print("Test Accuracy: {:.2f}%".format(CNN_Score[1] * 100))

def plot_curves(history):

    loss = history.history["loss"]

    val_loss = history.history["val_loss"]
```

```
    accuracy = history.history["accuracy"]
    val_accuracy = history.history["val_accuracy"]
    epochs = range(len(history.history["loss"]))
    plt.figure(figsize=(15,5))
    #plot loss
    plt.subplot(1, 2, 1)
    plt.plot(epochs, loss, label = "training_loss")
    plt.plot(epochs, val_loss, label = "val_loss")
    plt.title("Loss")
    plt.xlabel("epochs")
    plt.legend()
    #plot accuracy
    plt.subplot(1, 2, 2)
    plt.plot(epochs, accuracy, label = "training_accuracy")
    plt.plot(epochs, val_accuracy, label = "val_accuracy")
    plt.title("Accuracy")
    plt.xlabel("epochs")
    plt.legend()
  #plt.tight_layout()
plot_curves(CNN_history)
CNN_Predictions = CNN_Model.predict(test_data)
# Choosing highest probalbilty class in every prediction
CNN_Predictions = np.argmax(CNN_Predictions, axis=1)
test_data.class_indices
import seaborn as sns
from sklearn.metrics import confusion_matrix
fig, ax= plt.subplots(figsize=(15,10))
```

```
cm=confusion_matrix(test_data.labels, CNN_Predictions)
sns.heatmap(cm, annot=True, fmt='g', ax=ax)
ax.set_xlabel('Predicted labels',fontsize=15, fontweight='bold')
ax.set_ylabel('True labels', fontsize=15, fontweight='bold')
ax.set_title('CNN Confusion Matrix', fontsize=20, fontweight='bold')

#ResNet50V2 Model
# specifing new image shape for resnet
img_shape = 224
batch_size = 64
train_data_path = '../input/fer2013/train/'
test_data_path = '../input/fer2013/test/'
train_preprocessor = ImageDataGenerator(rescale = 1 / 255.,
        rotation_range=10,zoom_range=0.2,
        width_shift_range=0.1,height_shift_range=0.1,
        horizontal_flip=True, fill_mode='nearest',)
test_preprocessor = ImageDataGenerator( rescale = 1 / 255.,)
train_data = train_preprocessor.flow_from_directory(
    train_data_path, class_mode="categorical",
    target_size=(img_shape,img_shape),color_mode='rgb',
    shuffle=True, batch_size=batch_size,subset='training',)
test_data = test_preprocessor.flow_from_directory(
    test_data_path,class_mode="categorical",
    target_size=(img_shape,img_shape),color_mode="rgb",
    shuffle=False,batch_size=batch_size)

#Fine tuning ResNet50V2
```

```
ResNet50V2 = tf.keras.applications.ResNet50V2(input_shape=(224, 224,
3),include_top= False,weights='imagenet')

#ResNet50V2.summary()

# Freezing all layers except last 50

ResNet50V2.trainable = True

for layer in ResNet50V2.layers[:-50]:

    layer.trainable = False

def Create_ResNet50V2_Model():

    model = Sequential([ResNet50V2,Dropout(.25),BatchNormalization(),
Flatten(),Dense(64, activation='relu'),BatchNormalization(),
Dropout(.5),Dense(7,activation='softmax')])

    return model

ResNet50V2_Model = Create_ResNet50V2_Model()

ResNet50V2_Model.summary()

ResNet50V2_Model.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])
```

#Specifying Callbacks

```
# Create Callback Checkpoint

checkpoint_path = "ResNet50V2_Model_Checkpoint"

Checkpoint = ModelCheckpoint(checkpoint_path, monitor="val_accuracy",
save_best_only=True)

# Create Early Stopping Callback to monitor the accuracy

Early_Stopping = EarlyStopping(monitor = 'val_accuracy', patience = 7,
restore_best_weights = True, verbose=1)

# Create ReduceLROnPlateau Callback to reduce overfitting by decreasing
learning

Reducing_LR = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss',
factor=0.2, patience=2,# min_lr=0.00005,verbose=1)
```

```
callbacks = [Early_Stopping, Reducing_LR]

steps_per_epoch = train_data.n // train_data.batch_size

validation_steps = test_data.n // test_data.batch_size

ResNet50V2_history = ResNet50V2_Model.fit(train_data ,validation_data =
test_data , epochs=30, batch_size=batch_size, callbacks = callbacks,
steps_per_epoch=steps_per_epoch, validation_steps=validation_steps)


#Evaluating ResNet50V2

ResNet50V2_Score = ResNet50V2_Model.evaluate(test_data)

print("    Test Loss: {:.5f}".format(ResNet50V2_Score[0]))

print("Test Accuracy: {:.2f}%".format(ResNet50V2_Score[1] * 100))

plot_curves(ResNet50V2_history)

ResNet50V2_Predictions = ResNet50V2_Model.predict(test_data)

# Choosing highest probalbilty class in every prediction

ResNet50V2_Predictions = np.argmax(ResNet50V2_Predictions, axis=1)

fig , ax= plt.subplots(figsize=(15,10))

cm=confusion_matrix(test_data.labels, ResNet50V2_Predictions)

sns.heatmap(cm, annot=True, fmt='g', ax=ax)

ax.set_xlabel('Predicted labels',fontsize=15, fontweight='bold')

ax.set_ylabel('True labels', fontsize=15, fontweight='bold')

ax.set_title('ResNet50V2 Confusion Matrix', fontsize=20,
fontweight='bold')


#Visualizing Predictions

Emotion_Classes = ['Angry','Disgust','Fear','Happy','Neutral','Sad',

                   'Surprise']

# Shuffling Test Data to show diffrent classes

test_preprocessor = ImageDataGenerator(rescale = 1 / 255.,)
```

```
test_generator = test_preprocessor.flow_from_directory(
    test_data_path,class_mode="categorical",
    target_size=(img_shape,img_shape), color_mode="rgb",
    shuffle=True,batch_size=batch_size,)


#CNN Predictions
# Display 10 random pictures from the dataset with their labels
Random_batch = np.random.randint(0, len(test_generator) - 1)
Random_Img_Index = np.random.randint(0, batch_size - 1 , 10)
fig, axes = plt.subplots(nrows=2, ncols=5, figsize=(25, 10),
                         subplot_kw={'xticks': [], 'yticks': []})
for i, ax in enumerate(axes.flat):
    Random_Img = test_generator[Random_batch][0][Random_Img_Index[i]]
    Random_Img_Label =
np.argmax(test_generator[Random_batch][1][Random_Img_Index[i]])
    Model_Prediction = np.argmax(CNN_Model.predict(
tf.expand_dims(Random_Img, axis=0) , verbose=0))
    ax.imshow(Random_Img)
    if Emotion_Classes[Random_Img_Label] ==
Emotion_Classes[Model_Prediction]:
          color = "green"
    else:
          color = "red"
    ax.set_title(f"True: {Emotion_Classes[Random_Img_Label]}\nPredicted:
{Emotion_Classes[Model_Prediction]}", color=color)
plt.show()
plt.tight_layout()
```

```
#ResNet50V2 Prediction

# Display 10 random pictures from the dataset with their labels

Random_batch = np.random.randint(0, len(test_generator) - 1)

Random_Img_Index = np.random.randint(0, batch_size - 1 , 10)

fig, axes = plt.subplots(nrows=2, ncols=5, figsize=(25, 10),

                         subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):

    Random_Img = test_generator[Random_batch][0][Random_Img_Index[i]]

    Random_Img_Label =
np.argmax(test_generator[Random_batch][1][Random_Img_Index[i]])

    Model_Prediction = np.argmax(ResNet50V2_Model.predict(
tf.expand_dims(Random_Img, axis=0) , verbose=0))

    ax.imshow(Random_Img)

    if Emotion_Classes[Random_Img_Label] ==
Emotion_Classes[Model_Prediction]:

          color = "green"

    else:

          color = "red"

    ax.set_title(f"True: {Emotion_Classes[Random_Img_Label]}\nPredicted:
{Emotion_Classes[Model_Prediction]}", color=color)

plt.show()

plt.tight_layout()


#Music Player
Music_Player =
pd.read_csv("../input/spotify-music-data-to-identify-the-moods/data_moods.
csv")

Music_Player = Music_Player[['name','artist','mood','popularity']]

Music_Player.head()
```

```
Music_Player["mood"].value_counts()

Music_Player["popularity"].value_counts()

Play = Music_Player[Music_Player['mood'] =='Calm' ]

Play = Play.sort_values(by="popularity", ascending=False)

Play = Play[:5].reset_index(drop=True)

display(Play)

# Making Songs Recommendations Based on Predicted Class

def Recommend_Songs(pred_class):

    if( pred_class=='Disgust' ):

        Play = Music_Player[Music_Player['mood'] =='Sad' ]

        Play = Play.sort_values(by="popularity", ascending=False)

        Play = Play[:5].reset_index(drop=True)

        display(Play)

    if( pred_class=='Happy' or pred_class=='Sad' ):

        Play = Music_Player[Music_Player['mood'] =='Happy' ]

        Play = Play.sort_values(by="popularity", ascending=False)

        Play = Play[:5].reset_index(drop=True)

        display(Play)

    if( pred_class=='Fear' or pred_class=='Angry' ):

        Play = Music_Player[Music_Player['mood'] =='Calm' ]

        Play = Play.sort_values(by="popularity", ascending=False)

        Play = Play[:5].reset_index(drop=True)

        display(Play)

    if( pred_class=='Surprise' or pred_class=='Neutral' ):

        Play = Music_Player[Music_Player['mood'] =='Energetic' ]

        Play = Play.sort_values(by="popularity", ascending=False)

        Play = Play[:5].reset_index(drop=True)
```

```
        display(Play)
```

#Predicting New Images

```
!wget
https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

def load_and_prep_image(filename, img_shape = 224):
    img = cv2.imread(filename)
    GrayImg = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(GrayImg, 1.1, 4)
    for x,y,w,h in faces:
        roi_GrayImg = GrayImg[ y: y + h , x: x + w ]
        roi_Img = img[ y: y + h , x: x + w ]
        cv2.rectangle(img, (x,y), (x+w, y+h), (0, 255, 0), 2)
        plt.imshow(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))
        faces = faceCascade.detectMultiScale(roi_Img, 1.1, 4)
        if len(faces) == 0:
            print("No Faces Detected")
        else:
            for (ex, ey, ew, eh) in faces:
                img = roi_Img[ ey: ey+eh , ex: ex+ew ]
    RGBImg = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
    RGBImg= cv2.resize(RGBImg,(img_shape,img_shape))
    RGBImg = RGBImg/255.
    return RGBImg
def pred_and_plot(filename, class_names):
    # Import the target image and preprocess it
```

```
    img = load_and_prep_image(filename)

    # Make a prediction

    pred = ResNet50V2_Model.predict(np.expand_dims(img, axis=0))

    # Get the predicted class

    pred_class = class_names[pred.argmax()]

    # Plot the image and predicted class

    #plt.imshow(img)

    plt.title(f"Prediction: {pred_class}")

    plt.axis(False)

    Recommend_Songs(pred_class)

pred_and_plot("../input/fer2013/test/sad/PrivateTest_13472479.jpg",
Emotion_Classes) # with CNN

# Downloading Image to Test On

!wget -c "https://pbs.twimg.com/media/EEY3RFFWwAAc-qm.jpg" -O sad.jpg

pred_and_plot("./happy.jpg", Emotion_Classes) # with CNN

pred_and_plot("../input/fer2013/test/angry/PrivateTest_22126718.jpg",
Emotion_Classes) # with ResNet50V2

# Downloading Image to Test On

!wget -c
"https://pbs.twimg.com/profile_images/758370732413947904/xYB5Q3FY_400x400.
jpg" -O happy.jpg

pred_and_plot("./sad.jpg", Emotion_Classes) # with ResNet50V2

CNN_Model.save("CNN_Model.h5")

ResNet50V2_Model.save("ResNet50V2_Model.h5")
```

ANNEXURE - I		
STUDENTS PROJECT ROAD MAP		
NAME OF THE STUDENTS		REGISTER NUMBER
DEVADARSHINI M		211422243055
GOPIKASHREE P R		211422243080
NAME OF THE SUPERVISOR: Dr. M.S.MAHARAJAN		
DEPARTMENT: ARTIFICIAL INTELLIGENCE AND DATA SCIENCE		
1	TITLE OF THE PROJECT	Deep Learning Based Emotion-Driven Music Recommendation System
2	RATIONALE (why the topic is important today in 3 sentences in bullet points)	● People struggle to find music that matches their current emotional state in real time. ● Traditional recommendation systems rely on user preferences or history, not live emotion. ● AI-based emotion recognition can enhance user engagement through mood-based personalization
3	LITERATURE SURVEY (Top 5 articles utilized for finding the research gap and their SCOPUS impact factor)	● H. Zhang et al., "Face Emotion Recognition Using CNN", IEEE Access, IF: 3.36

		● D. Kim et al., "Music Recommendation Using K-Means Clustering", IF: 2.67 ● D. Ayata et al., "Emotion Detection via Wearable Sensors", IEEE Trans. on Consumer Electronics, IF: 3.11 ● W. C. Chiang et al., "Hierarchical SVM for Music Emotion Recognition", IF: 1.98 ● F. Fessahaye et al., "T-RECSYS: Hybrid Deep Learning Music Recommender", IF: 4.21
4	RESEARCH GAP (Maximum 3 sentences in bullet Points)	● Existing systems don't combine real-time facial emotion recognition with music mapping. ● Lack of user-centric dynamic song suggestions based on current mood. ● Limited implementation using deep learning architectures like ResNet50V2 for emotion-based music systems.
5	BRIDGING THE GAP (Maximum 4 sentences in bullet Points)	● Facial emotion is detected using deep learning models (CNN & ResNet50V2). ● Real-time emotion analysis provides immediate music suggestions.

		● Combines AI, Computer Vision, and emotion mapping to recommend relevant songs. ● Provides a dynamic, responsive alternative to traditional static music systems.
6	NOVELTY (Maximum 3 sentences in bullet Points)	● Integrates facial expression analysis with music genre mapping. ● Uses deep learning (ResNet50V2) for high-accuracy emotion recognition. ● Real-time emotion detection enables automatic mood-based song recommendation.
7	OBJECTIVES (Maximum 5 sentences in bullet Points)	● Develop a system that detects facial emotions using deep learning models. ● Classify emotions into predefined categories like Happy, Sad, Angry, etc. ● Map each emotion to suitable music genres using a curated dataset. ● Enhance user experience through real-time, mood-based music suggestions.

		• Implement and evaluate model performance using accuracy and classification metrics.
8	PROCESS METHODOLOGY (Maximum 7 sentences in bullet Points)	• Acquire image input from webcam or file upload. • Perform preprocessing and face detection using OpenCV and Haar cascades. • Use CNN and ResNet50V2 for emotion classification. • Map detected emotion to music genre using a predefined dictionary. • Display song suggestions from curated mood-based datasets. • Evaluate the system using precision, recall, and F1-score. • Optimize model performance using callbacks and hyperparameter tuning.
9	SIMULATION METHODOLOGY AND SIMULATION SOFTWARE REQUIREMENT (Maximum 4 sentences in bullet Points)	• Model developed in Python using TensorFlow, Keras, and OpenCV. • Training done on FER2013 dataset using CNN and ResNet50V2 architectures. • Data preprocessing and augmentation using ImageDataGenerator. • Evaluation via confusion matrix and classification report.

10	DELIVERABLES & OUTCOMES (Maximum 4 sentences in bullet Points) (Technology, Prototype, Algorithm, Software, patent, publication, etc)	● Deep Learning models for facial emotion recognition. ● A working prototype of the music recommendation system. ● Dataset integration and emotion-to-genre mapping algorithm. ● Journal publication and potential for real-world application.
11	PROJECT CONTRIBUTION IN REALTIME	● Conference Paper: Published in ICONIC PECTEAM 2K25 ● Indexed in Google Scholar ● Lays foundation for further AI-driven human-computer interaction research ● Can be expanded for future patent or copyright applications
11	SUSTAINABLE DEVELOPMENT GOALS MAPPED (Mention the SDG numbers)	SDG 3 , SDG 9 , SDG 11
12	PROGRAMME OUTCOME MAPPING (PO) (Mention the PO numbers)	PO1 , PO2 , PO3 , PO5 , PO6 , PO12
13	TIMELINE	Milestones
	Month 1	Literature survey, Dataset collection, Model planning
	Month 2	Model training, Emotion recognition implementation

	Month 3	Recommendation system integration, Evaluation, Report writing
SUPERVISOR SIGNATURE		