**Project Report**

**MoodMusic AI — Emotion-Based Music Recommendation System**

Name – Vaibhav Pant(S24CSEU1765)

Piyush Punera(S24CSEU1772)

A report submitted in part fulfilment of the degree of

BTech in Computer Science

Supervisor: Dr. Susmita Das



School of Computer Science Engineering and Technology

# Abstract

**Introduction**

In today’s digital era, personalisation is key. Music platforms often suggest content based on listening history, but **MoodMusic AI** goes a step further — it recommends playlists based on the user's current emotional state captured through their facial expressions.

Using computer vision and deep learning, MoodMusic AI analyzes emotions and links them with curated playlists. The system uses:

* **Computer Vision (OpenCV + Haar Cascade)**
* **Deep Learning (PyTorch + ResNet-18)**
* **Flask Web Framework**
* **Matplotlib-based Analytics Dashboard**
* **SQLite Database for User Data & Emotion History**

The entire project revolves around integrating ML with a real-world application — a seamless experience where users can visualize their emotional trends and enjoy AI-curated music.

**Proposed Works/Methodology**

### ****1. Dataset Preparation****

* Images resized to 224×224
* Normalized using ImageNet standard values
* Augmentation through PyTorch transforms

### ****2. Model Architecture — ResNet-18****

* Pretrained on ImageNet
* Last layer replaced with custom classifier of 3 outputs
* Feature layers frozen to speed up training
* Optimizer: **Adam**
* Loss: **CrossEntropyLoss**

### ****3. Web Backend – Flask****

* User authentication
* Image upload API /predict
* Emotion logging
* Analytics rendering
* REST endpoints for playlist liking

### ****4. Database – SQLite****

Tables created:

* users
* emotions
* playlists

### ****5. Visualization – Matplotlib****

* Line chart: emotion trend over time
* Pie chart: emotion distribution
* Rendered as Base64 images inside HTML

**Data structure and algorithms used.**

## ****Data Structures****

### ****1. Tensors (PyTorch)****

* Used to store images, model weights, and inputs.
* Enables fast GPU-based computation.

### ****2. Lists****

* Store emotions, timestamps, chart data, query results.

### ****3. Dictionaries****

* Store key-value pairs like JSON responses:  
  { "emotion": "HAPPY" }
* Label mappings: {"SAD":0, "NEUTRAL":1, "HAPPY":2}

### ****4. SQLite Tables****

* users → username, email, password
* emotions → emotion + timestamp
* playlists → liked playlist links
* Internally uses **B-Trees** for efficient searching.

### ****5. Graph Structures (Matplotlib)****

* Line chart data
* Pie chart data

## ****Algorithms****

### ****1. Convolutional Neural Network (ResNet18)****

* Used for emotion classification into HAPPY, SAD, NEUTRAL.
* Uses **convolutions, pooling, residual blocks, backpropagation**.

### ****2. Haar Cascade (OpenCV Face Detection)****

* Detects face from webcam image.
* Uses **Haar features + AdaBoost + cascade classifier**.

### ****3. Image Preprocessing****

* Resize → 224×224
* Normalize
* Convert to tensor

### ****4. SQL Retrieval Algorithms****

* SQLite uses **B-Tree indexing** for storing and retrieving user mood history.

### ****5. JavaScript Algorithms****

* Webcam capture
* Canvas image processing
* Sending image to backend
* Updating UI and playlist

**Result Analysis**

### ****Features Working****

✔ Real-time emotion recognition  
✔ Automatic Spotify playlist recommendation  
✔ Emotion logging per user  
✔ Analytics Dashboard:

* **Emotion Trends (Line chart)**
* **Emotion Distribution (Pie chart)**

### ****User Features****

✔ Login/Signup  
✔ Save favorite playlists  
✔ Delete saved playlists  
✔ View mood history

**Conclusion**

MoodMusic AI successfully integrates deep learning–based emotion analysis with music recommendation.  
The project shows how AI can automate mood-driven personalization.

Future enhancements include:

* More emotion classes
* Larger dataset
* Real Spotify API integration
* Mobile App version