Assessment 3:

Melofy: The AI Mood-Based Music Recommendation System

36121 Artificial Intelligence Principles and Applications

Autumn 2025

Student Names and IDs:

Jaime Garcia y Garcia	13863992
Katherin Gomez Londono	24611687
Miguel Ramal	00060259
Spenser Gautama	13091323
Dipesh Shrestha	24996124

Link to Public GitHub Repository	https://github.com/kathegolon-uts/AI_a3_repo_group_5
Deployed App	https://spotimood.streamlit.app/

May 18th, 2025

Table of Contents

1. Executive Summary

1.1 Overview of the Project

This report presents Melofy, an AI-powered application that recommends music based on facial emotion. Developed for the Artificial Intelligence Principles and Applications subject at UTS, it combines facial expression recognition with a music recommendation engine to create an engaging and emotionally aware user experience. The system addresses a real-world gap in current music platforms, which often overlook users' present emotional state in favour of historical or genre-based patterns. After consent, users can take a selfie or upload one; a fine-tuned Vision Transformer (ViT) model—selected for its ability to model global visual patterns more effectively than CNNs—detects their emotion and suggests music accordingly. The app is publicly accessible via any web browser through Streamlit Community Cloud. The project demonstrates how AI can deliver emotionally responsive, real-time content by linking state-of-the-art computer vision with mood-based music recommendations.

1.2 Objectives

- Address a real-world problem by developing an AI-powered, emotion-aware web app that combines facial recognition (≥90% accuracy) with mood-based music recommendations, assessed through validation and user feedback.
- Ensure responsible AI use by protecting user privacy and promoting ethical, positive user experiences.

1.3 Key Findings

- Melofy effectively integrated AI into a real-time, user-facing app, using facial emotion recognition to deliver emotionally relevant music recommendations.
- The facial emotion recognition component, fine-tuned on the FER-2013 dataset, achieved over 91% validation accuracy for reliable mood detection, outperforming CNN-based models that reached only 60–68%.
- The app's modular and lightweight design ensures a seamless user experience while supporting future scalability and portability.
- Ethical considerations were prioritised, with user consent, secure image handling, and anonymised, short-term processing of facial data.

2. Introduction

2.1 Background and Context

Music streaming platforms have transformed how users access and engage with digital music. While many recommendation systems provide personalised content based on listening history or genre preferences, they often overlook the user’s real-time emotional context. This limits their ability to deliver recommendations that reflect a listener’s current state of mind.

Advancements in artificial intelligence, particularly in computer vision for facial recognition and emotion detection, enable systems to infer emotional states from facial expressions in real time. When combined with mood-tagged music datasets, these technologies can support more context-aware and emotionally intelligent recommendations. This project explores how AI can be used to enhance music streaming by delivering suggestions that respond to the user’s emotional state in the moment.

2.2 Problem Statement

Despite improvements in personalisation, most music recommendation systems remain reactive, relying on past behaviour, collaborative filtering, or static preferences. These models lack the ability to adjust dynamically to users’ emotional changes, which are central to how people experience and engage with music. This project addresses that gap by introducing an AI-driven solution that detects facial emotions and maps them to mood-based song recommendations. It demonstrates how AI techniques, particularly facial emotion classification and predictive modelling, can be applied to deliver real-time, emotionally aligned content, enriching everyday digital experiences in a meaningful and human-centred way.

The potential real-world significance lies in the growing demand for emotion-aware technologies that can adapt to users' psychological states. As music is closely tied to mental wellbeing, a system like Melofy could improve emotional self-awareness, support mood regulation, and enhance user satisfaction, particularly for individuals seeking emotional comfort or motivation through music. This has implications not only for the music industry but also for applications in mental health, education, and personalised digital media.

2.3 Scope of the Project

Melofy is a browser-based web application that includes:

1. A fine-tuned Vision Transformer model to classify seven facial expressions *(angry, disgust, fear, happy, sad, surprise, neutral).*
2. A mapping layer that translates these into four mood categories *(happy, calm, energetic, sad)* based on the Moodify dataset.
3. A user interface built with Streamlit that supports image upload and camera input, and embeds recommended Spotify tracks.
4. Ethical safeguards through explicit user consent, real-time processing, and anonymised session-based image handling.

The current system excludes additional input modalities (e.g., voice or text), extended emotion categories, user accounts, integrations beyond Spotify, and any songs outside the Moodify dataset.

3. Project Planning

3.1 Project Goals and Objectives

The main goal of this project is to develop an AI-based music recommendation system that responds to users' real-time emotional states. The project aims to:

1. **Implement Emotion Detection:** Use a fine-tuned Vision Transformer to classify facial expressions with high accuracy.
2. **Build a Mood-Based Recommender:** Map emotions to music moods using the Moodify dataset and suggest appropriate tracks.
3. **Develop a Responsive Web Interface:** Create a user-friendly Streamlit app supporting both image upload and camera input.
4. **Ensure Ethical and Scalable Design:** Integrate consent mechanisms, temporary image processing, and ensure deployment readiness.

These objectives reflect the core principles of applied AI, model development, system integration, and ethical deployment, and demonstrate how machine learning can enhance real-world user experiences. An effective solution has the potential to improve user engagement, emotional well-being, and satisfaction in music streaming by delivering content that aligns with users' moment-to-moment emotional needs.

3.2 Stakeholder Analysis

The project involves the following key stakeholders:

1. **End Users:** Seek music recommendations aligned with their current mood; their interaction and feedback inform usability and refinement.
2. **Developers (Project Team):** Responsible for system design, model training, integration, and deployment.
3. **Streaming Platforms:** Could benefit from improved user engagement through emotion-aware personalisation.
4. **Music Artists and Labels:** Gain indirect exposure through mood-aligned song suggestions.

3.3 Project Timeline and Milestones

Melofy followed a fast-tracked, collaborative timeline spanning just over two weeks. The project began with ideation and task allocation, followed by rapid development, integration, testing, and preparation for presentation and reporting. Tasks were divided by strengths, and regular check-ins helped maintain alignment.

A Gantt chart (Figure 1) outlines the key milestones and phases of the project:

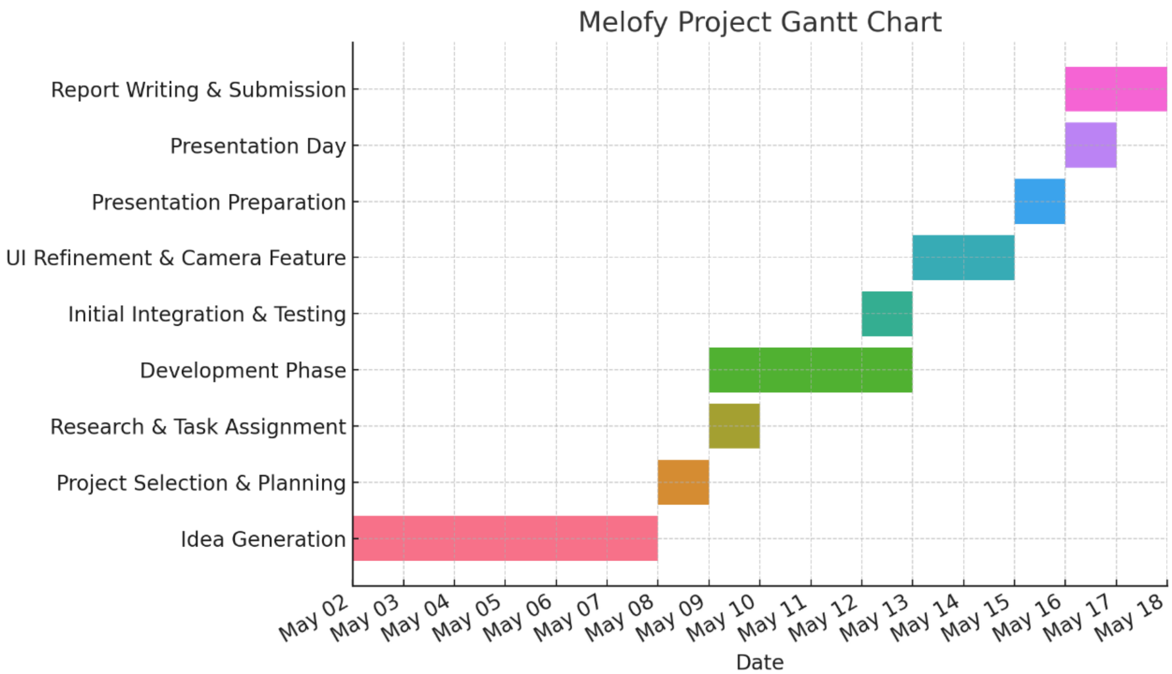


Figure 1 Melofy Project Gantt Chart

3.4 Risk Assessment and Mitigation

Risk	Description	Mitigation
Time Constraints	Short development window	Early task allocation, clear milestones, and regular progress check-ins
Model Performance	Risk of misclassification or overfitting	Used pre-trained ViT and evaluated with F1-score and precision/recall
Dataset Limitations	Class imbalance in FER-2013 and Moodify datasets	Applied F1-score and manual review to reduce bias
Mood Labels Mismatch	Model outputs 7 emotions; Moodify supports 4 mood labels	Created a consistent mapping layer and validated integration
Ethical Concerns	Use of facial image data	Added consent checkbox and ensured temporary, in-session image processing

Table 1 Risks and their Mitigation of the Project

4. Methodology/System Overview

4.1 System Architecture

Melofy follows a modular architecture (Figure 2) that connects a Streamlit frontend with a backend powered by computer vision and a music recommendation engine. Upon user consent, a webcam or uploaded image is processed by a fine-tuned Vision Transformer (ViT) model to detect one of seven emotions. These are mapped to four target moods (happy, calm, energetic, sad), which guide the retrieval of songs from the Moodify dataset. If sad is detected, a secondary uplifting track is added. Recommended songs are embedded via Spotify, and users provide feedback with a thumbs-up/down.

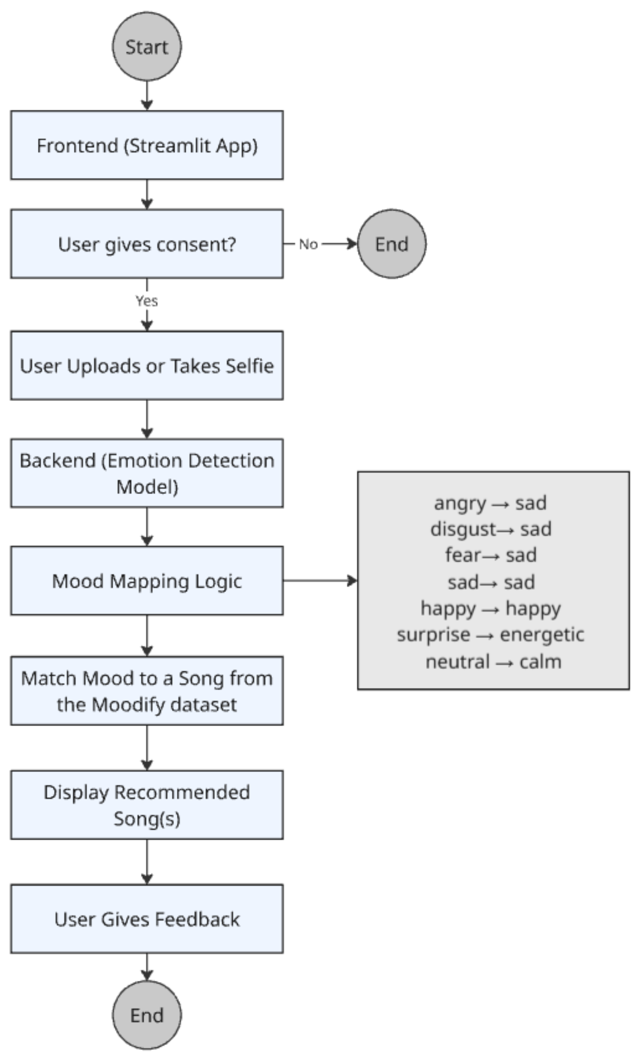


Figure 2 Melofy System Architecture Diagram

4.2 Tools and Technologies Used

Tool / Technology	Purpose
Streamlit	Real-time user interaction via web app interface
ViT (Vision Transformer)	Emotion detection via deep learning (Hugging Face)
Pandas	Dataset filtering and in-memory processing
Spotify Embed	Displaying music directly within the app
Poetry	Dependency and environment management
GitHub	Version control and collaborative development
OneDrive	Shared storage for reports and datasets
VS Code	Main development environment

Table 2 Tools and Technologies used in the Project and their Purpose

4.3 Database Design and Management

Melofy uses two static datasets loaded into memory at runtime:

Dataset	Usage	Format & Size	Labels	Source
FER-2013	Emotion model training/testing	48×48 RGB images (~32,000 samples)	Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral	Kaggle
Moodify	Song recommendation	CSV with ~278,000 entries	Sad, Happy, Energetic, Calm	Kaggle

Table 3 Characteristics of the Datasets used in the Project

Preprocessing: Both datasets were pretty cleaned so only in the FER-2013, images were resized to 224×224 and augmented (rotation, flip, sharpness) for training. During validation, only resizing and normalisation were applied.

As illustrated in Figures 3 and 4, both datasets exhibited class imbalances, for instance, a higher number of "Happy" samples compared to the underrepresented "Disgust" class. As a result, accuracy alone was not a reliable performance metric. Greater emphasis was placed on F1-score, precision, and recall. This imbalance was also taken into account during model integration to minimize bias in the recommendation outcomes.

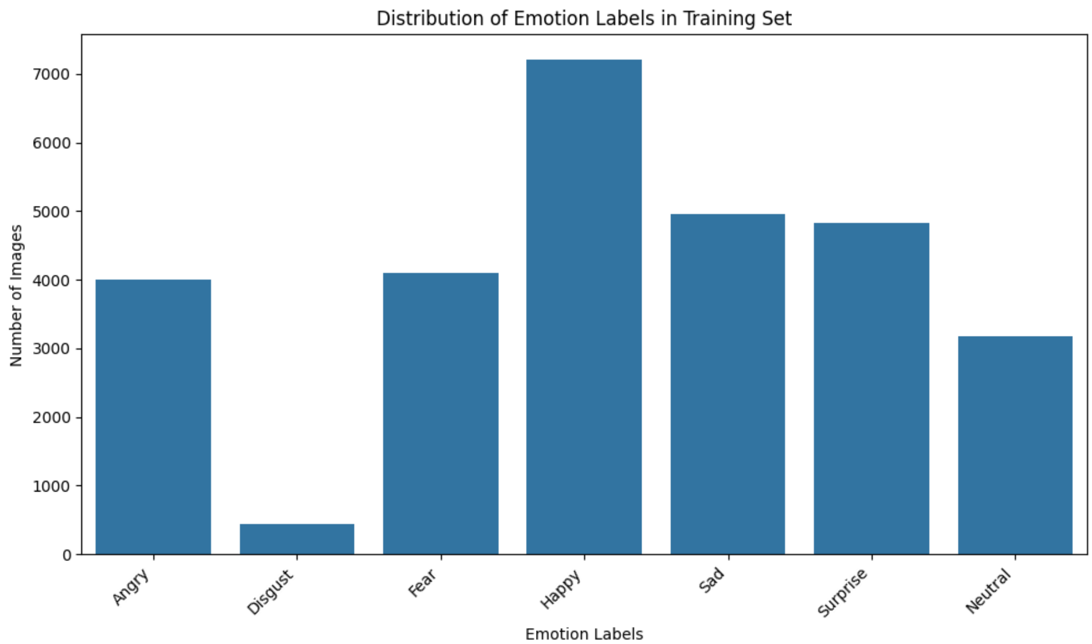


Figure 3 Distribution of Emotion Labels in Training Set

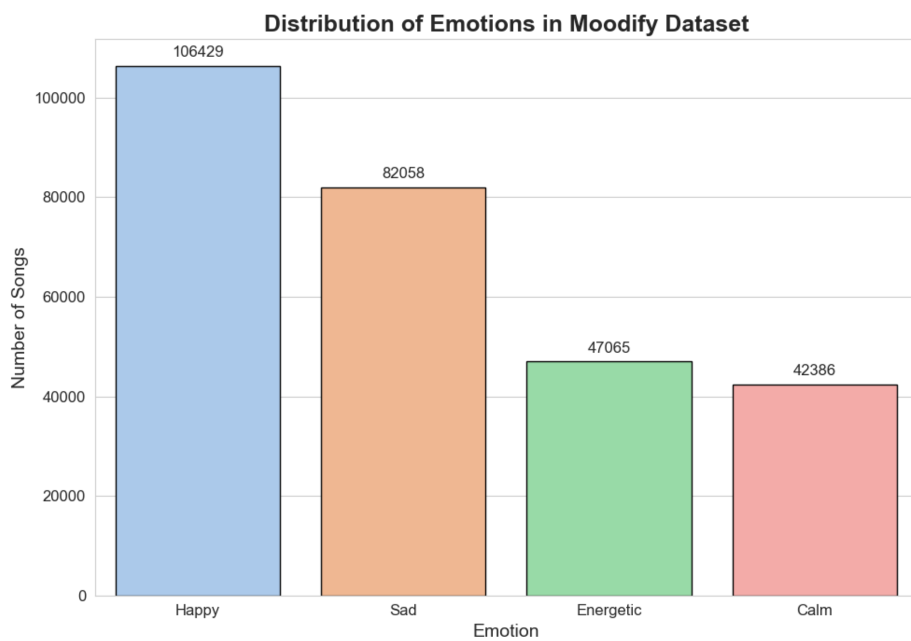


Figure 4 Distribution of Emotions in Moodify Dataset

Mapping Layer: The seven model-predicted emotions were mapped to four moods using the following logic:

Emotion (from Model Output)	Mood (from Moodify Dataset)
Angry	Sad
Disgust	Sad
Fear	Sad
Sad	Sad
Happy	Happy
Surprise	Energetic
Neutral	Calm

Table 4 Mapping Layer (Model Moods - Moodify Moods)

EDA confirmed Moodify label quality:

- *Energetic* songs had higher loudness and danceability.
- *Calm* and *Sad* were lower on both.

These patterns validate the Moodify labels and reinforce the system's mapping between facial emotions and musical moods.

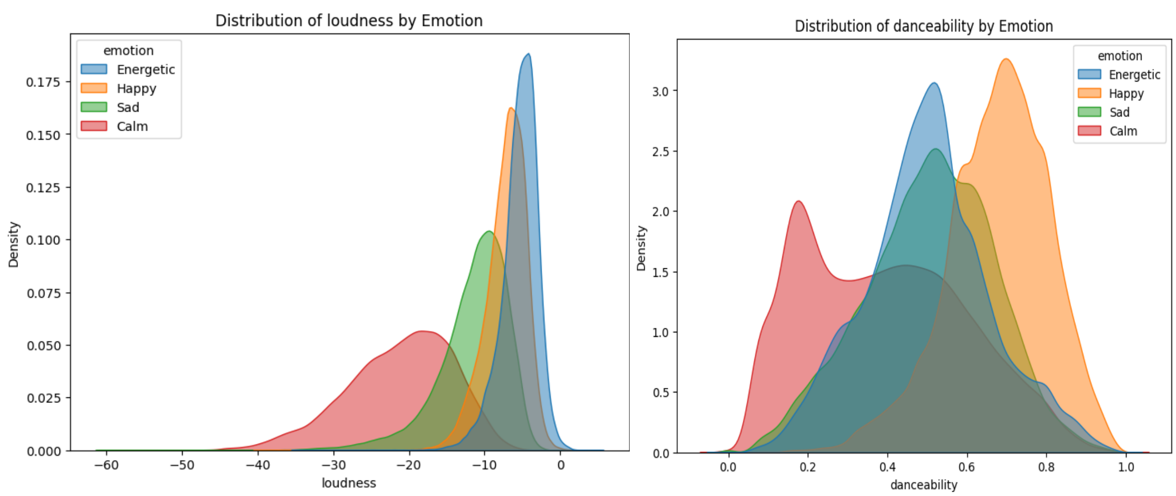


Figure 5 Distribution of loudness and danceability by Emotion

4.4 Security Considerations

The system operates without storing user data or requiring authentication, reducing exposure to common security threats. All data processing occurs locally in the browser session, eliminating the risk of external breaches. Dependencies are managed via Poetry to ensure consistent, secure environments, and the app is deployed using Streamlit Community Cloud, which provides HTTPS encryption by default.

5. Design and Implementation

5.1 System/Software Design

Melofy was designed for simplicity, modularity, and rapid deployment. Rather than relying on complex backend infrastructure, the system uses lightweight, open-source tools that integrate smoothly. Each module (interface, emotion detection model, mood mapping logic, and music recommendation), functions independently, which makes development and testing more manageable.

Streamlit was selected for its ease of use and minimal setup requirements, enabling fast frontend development. The backend model is treated as a plug-in that returns predicted emotion labels from input images. Static CSV datasets (FER-2013 and Moodify) and in-memory filtering via Pandas replaced the need for a database or persistent storage.

This modular design supports testing, error handling, and future feature extensions such as user profiles or adaptive learning.

5.2 Development Methodology

An agile, iterative workflow guided development. Team members took responsibility for core areas (model training, data integration, interface design) and collaborated using GitHub for version control.

Early development focused on environment setup and evaluating emotion models. Once the image input flow was functioning in Streamlit, integration with the Moodify dataset and Spotify embedding followed. Issues such as Spotify URI handling and label mismatches were resolved through group debugging and testing. This approach allowed the system to evolve rapidly while maintaining consistency.

5.3 Coding and Development Process

Development was done in Python using Visual Studio Code. Hugging Face's transformers library was used to load and fine-tune the Vision Transformer (ViT) model on the FER-2013 dataset.

Model Development

We used a Vision Transformer (ViT) model for facial emotion recognition, fine-tuned on the FER-2013 dataset. ViT offers a transformer-based alternative to CNNs, which have long dominated image classification tasks. Unlike CNNs, which focus on local patterns using filters and pooling layers, ViT treats an image as a sequence of patches. The image is divided into fixed-size patches (e.g. 16×16), each of which is flattened and linearly projected into an embedding vector. These patch embeddings are then enriched with **positional encodings** to retain spatial information.

A special learnable **[CLS] token** is prepended to the sequence. During training and inference, this token accumulates global information through **multi-head self-attention** layers in the transformer encoder. The final representation of the [CLS] token is then passed to a **multi-layer perceptron (MLP)** for classification (see Figure 6).

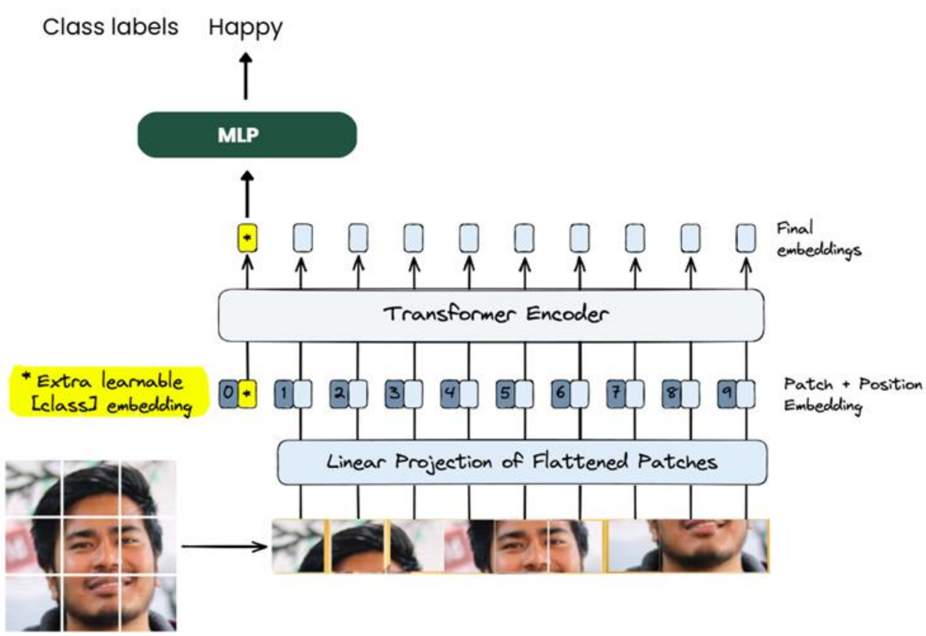


Figure 6 ViT Classification Architecture for Emotion Detection

The table below shows the vision transformer training settings used:

Setting	Value
Model Architecture	ViT (Vision Transformer)
Pretrained Base	Hugging Face (ViT-Base)
Dataset	FER-2013
Optimiser	AdamW
Learning Rate	3e-5
Weight Decay	0.1
Scheduler	Cosine with 1000 warm-up steps
Loss Function	Cross Entropy
Epochs	3
Evaluation Metrics	F1-score, Precision, Recall

Table 5 Vision Transformer Training Configuration

This architecture is particularly effective for facial emotion detection because:

- It models **global relationships** across the entire face, which is helpful in distinguishing subtle differences (e.g. fear vs. surprise).
- It reduces inductive biases compared to CNNs, allowing more flexibility to learn task-specific patterns.
- The attention mechanism allows the model to focus on relevant facial regions regardless of position).

By selecting ViT, we demonstrated the application of a state-of-the-art transformer-based architecture in a real-world classification problem, integrating modern deep learning techniques within an ethical and interactive user system.

Recommendation Logic

After emotion classification, the label is mapped to one of four mood categories. The system then filters the Moodify dataset using Pandas and selects a track at random. For the "sad" mood, an additional uplifting song is displayed.

Interface and Integration

Streamlit was used to build the web interface, allowing users to upload or capture images. Spotify tracks are embedded using dynamically generated HTML, and simple thumbs-up/down feedback buttons were added for basic interactivity. No user accounts or storage are required.

Code Structure and Deployment

The codebase was divided into clear modules for emotion detection, mood mapping, and recommendation logic. Continuous testing was performed locally and in the deployed environment via Streamlit Community Cloud. Poetry was used to manage dependencies and ensure reproducibility across systems.

5.4 Testing Strategy and Implementation

- **Emotion Detection:** Sample images with varied lighting and expressions were used to evaluate model robustness. Predictions were manually verified to ensure alignment with visual input. Tests included edge cases such as low-light images and partial occlusion.
- **Mood Mapping and Recommendation:** Unit tests confirmed that emotions were consistently mapped to the correct mood category. The dual-track logic for sad detections was verified through repeated tests with controlled inputs.
- **User Interface:** Streamlit components (camera, uploader, Spotify embed) were tested both locally and in deployment. The interface was checked for responsiveness and error handling across devices.
- **User Feedback:** Five test users provided informal feedback. Most found the interface clear, and the recommendations relevant. Based on their input, minor UI tweaks and text improvements were made.

6. Results and Analysis

6.1 Performance Evaluation

The facial emotion detection model, based on Vision Transformer (ViT), was evaluated using F1-score, precision, and recall, more appropriate than raw accuracy given the imbalanced nature of the FER-2013 dataset.

The model achieved:

- **F1-score:** 91.39%
- **Accuracy:** 91.36%
- **Per-class precision and recall:** >85%

As shown in Figure 7, the confusion matrix demonstrates strong classification performance, particularly for neutral, fear, and surprise. Some misclassifications were observed between angry, fear, and surprise, emotions that are frequently confused even in human perception.

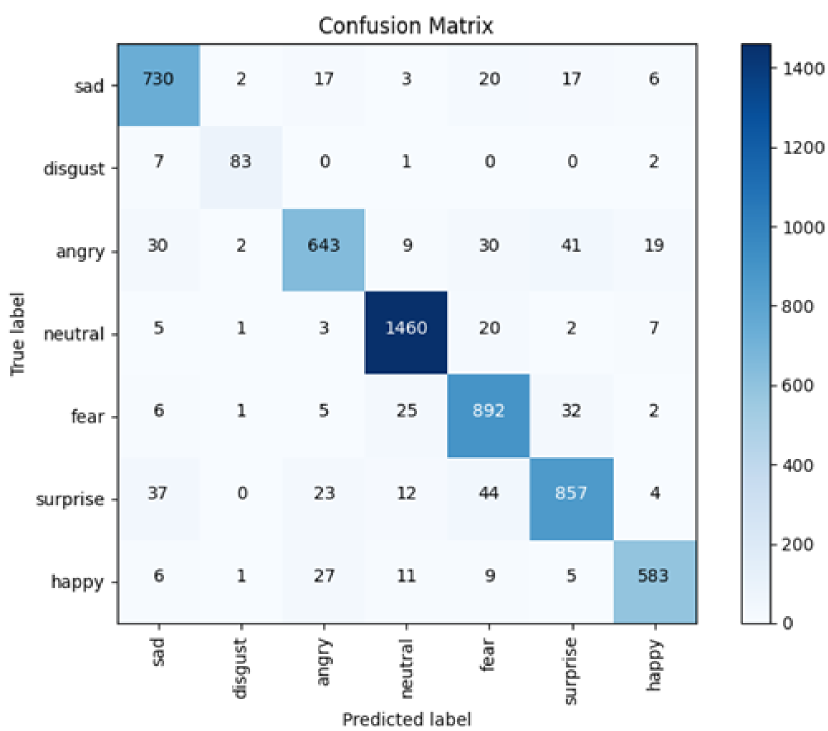


Figure 7 ViT Confusion Matrix

To further explore these errors, Figure 8 displays sample FER-2013 images. For instance, a surprised face was mislabelled as neutral, and a surprised expression was confused with sadness, highlighting the challenge of interpreting facial ambiguity.

Despite these limitations, the model consistently returned emotionally relevant outputs. The simplified mood mapping helped preserve stability in the recommendation system and ensured each detection produced a valid and actionable result.

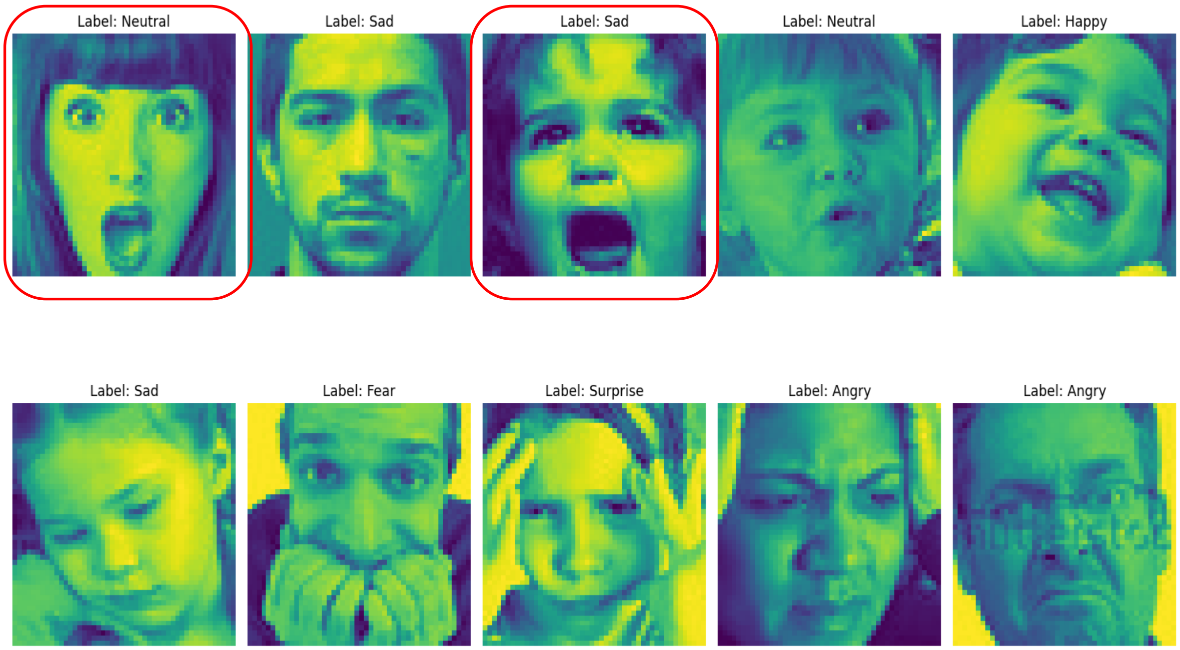


Figure 8 Samples of the FER-2013 Dataset

Recommendation Functionality

The mood-to-music mapping logic operated reliably. For each detected emotion, the system selected a song from the corresponding mood category in the Moodify dataset. For sad classifications, the system correctly delivered both a mood-aligned and uplifting track, as intended.

Spotify embeds rendered properly in both local and deployed environments. Average system response time, measured from image input to song recommendation, was 2–3 seconds, enabling real-time use without perceptible lag.

6.2 User Testing and Feedback

Five participants tested the app in real-world conditions using both the webcam and image upload features. Informal feedback was gathered to assess usability, perceived accuracy, and content relevance.

Key feedback themes:

- The interface was intuitive and user-friendly.
- Detected emotions generally aligned with participants’ actual expressions.
- Music recommendations, especially for *happy* and *calm*, were described as appropriate.
- The additional uplifting track for sad moods was a well-received feature.

Suggestions included:

- Clearer UI instructions (e.g., camera vs. upload)
- Adding variety in recommendations (e.g., multiple tracks per mood)

Minor UI adjustments and default messaging improvements were made based on this feedback.

6.3 Challenges Faced and Solutions

Several issues arose during development, requiring practical workarounds to maintain functionality:

Challenge	Solution
Spotify URI formatting errors	Standardised all URIs to match embed format
FER-2013 data issues	Cleaned duplicate and mislabelled entries using Pandas
Emotion misclassification	Observed neutral-to-sad or sad-to-surprise errors (see Figure 8); documented as model limitation
Unsupported file formats	Restricted inputs to JPEG/PNG with validation and error messages
Label mismatch with Moodify	Created a mapping from 7 emotion classes to 4 mood categories

Table 6 Challenges Faced and Solutions

Despite these challenges, the system achieved its intended goals, accurately classifying user emotions, returning relevant recommendations, and maintaining a seamless user experience within a lightweight, ethical, and scalable design.

7. Deployment and Maintenance

7.1 Deployment Plan

Melofy was deployed via **Streamlit Community Cloud**, selected for its ease of integration with Python-based ML apps and no-cost hosting model. This platform allowed rapid deployment and iteration without manual server configuration.

Deployment steps:

1. Code was pushed to a public GitHub repository.
2. Streamlit Cloud linked to the repo and automatically identified the entry point.
3. Dependencies were installed using pyproject.toml (managed by Poetry).

4. A public URL was generated, offering immediate access without local installation.
5. Any commit to the main branch triggers automatic redeployment.

Streamlit also supports version rollback and error logging, making it ideal for proof-of-concept development. If needed, the app can later be migrated to platforms like **AWS, GCP,** or **Heroku** to support more advanced backend services, user authentication, or high-traffic loads.

7.2 Maintenance and Support Strategies

Melofy's modular architecture reduces the need for complex maintenance. Hosting through Streamlit Cloud manages uptime, builds, and redeployment automatically.

- **Dependency Management:** Poetry ensures reproducible environments and clean updates.
- **Version Control:** GitHub is used for collaboration, issue tracking, and version history.
- **Monitoring:** Informal testing and local logging currently suffice; more robust monitoring can be added if scaled.
- **Support:** While no formal support system exists, user feedback has already shaped iterative improvements. In a future production context, structured channels (e.g. user guides, changelogs, or a bug reporting form) could be introduced.

7.3 Future Enhancements

As a functional prototype, Melofy lays the groundwork for several AI-driven improvements:

Enhancement Area	Description
Expanded Emotion Classes	Include more nuanced emotional states (e.g. anxiety, boredom, excitement).
Model Refinement	Test hybrid or fine-tuned models (e.g. CNN-ViT ensembles) to improve precision.
Playlist Generation	Use Spotify APIs to suggest full playlists instead of single songs.
User Profiles	Enable account login, mood tracking, and personalised music history.
Multi-Modal Input	Add voice tone or sentiment-from-text detection to broaden accessibility.
Mobile App	Develop a dedicated mobile application for on-the-go use.
Cloud Migration	Move to AWS, GCP, or Azure to support growth in users or data volume.

Table 7 Future Enhancements

These enhancements would enable Melofy to evolve into a production-ready, context-aware music recommender system grounded in modern AI practices.

8. Individual contributions

Each team member contributed to both individual components and collaborative project development. Roles were assigned based on technical strengths and interests, and weekly meetings ensured alignment and timely progress.

- **Jaime Garcia:** Developed core frontend features in the Streamlit app, including camera input handling, feedback functionality, and layout design. Led deployment on Streamlit Community Cloud, resolved runtime errors during testing, and contributed to the report (technical sections and introduction). Also assisted with preparing the project demo and slides.
- **Miguel Ramal:** Proposed the initial concept, explored relevant datasets and research to support project feasibility, and helped refine the technical direction. Led project planning discussions, conducted local and deployed testing, reviewed GitHub commits, and contributed to the Executive Summary, Introduction, and Project Planning sections. Also assisted with final proofreading and presentation delivery.
- **Dipesh Shrestha:** Selected, fine-tuned, and evaluated the Vision Transformer (ViT) model using FER-2013, and deployed it via Hugging Face for integration. Provided support in aligning model outputs with the app pipeline. Designed the presentation slides and contributed to writing model development and evaluation sections in the report.
- **Katherin Gomez:** Set up the complete development environment, including GitHub, OneDrive, Poetry, and Pyenv. Built the backend of the Streamlit app, implemented the frontend structure, and fully integrated the emotion detection model to the app. Developed the mood mapping and recommendation logic, handled user consent flow, and conducted EDA on the Moodify dataset. Authored major sections of the report (Methodology, Architecture, and Design), edited and refined the full report and slides, and created the User Guide and supporting documentation.
- **Spenser Gautama:** Supported testing across modules, and contributed to the report by refining formatting, improving layout clarity, and creating visual tables. Assisted with the presentation script.

9. Conclusion and Recommendations

9.1 Summary of Findings

This project successfully fulfilled the objectives of the *Artificial Intelligence Principles and Applications* assignment by applying AI techniques, specifically deep learning for computer vision, in a functional, interactive system. Melofy demonstrates how facial emotion recognition can be used to deliver context-aware, personalised music recommendations in real time.

By fine-tuning a **Vision Transformer (ViT)** on the FER-2013 dataset, we implemented an advanced emotion classification model capable of recognising nuanced facial expressions. This model was embedded in a modular recommendation pipeline, where outputs were mapped to mood-labelled songs from the Moodify dataset. The system was deployed as a lightweight web app using Streamlit, enabling seamless integration between AI model inference and user interaction.

Crucially, the project exemplified **responsible AI development**, combining accurate predictions with ethical safeguards such as consent-based data use, in-session image processing, and privacy-by-design. Performance was validated using metrics tailored for class-imbalanced data (F1-score, precision, recall), and misclassification patterns were analysed to highlight model limitations and future areas for improvement.

Through user testing, the system was shown to be both functional and engaging. Compared to traditional recommender systems that rely on behavioural data, Melofy responds dynamically to a user's current emotional state, demonstrating the value of AI systems that adapt to real-time human context.

More broadly, this project illustrated a foundational AI concept: the use of structured representations (image → emotion → mood) and model-driven reasoning to produce intelligent, real-world outputs. It serves as a working example of how modern AI architectures can be ethically and effectively applied to augment everyday digital experiences.

9.2 Recommendations for Future Work

To extend Melofy's capabilities and further explore applied AI techniques, we recommend:

- **Model Enhancement:** Investigate more advanced or ensemble approaches (e.g. hybrid CNN–ViT models) to improve recognition of subtle or overlapping emotional states.
- **Expanded Emotion and Mood Mapping:** Add support for a wider emotional spectrum (e.g. anxiety, boredom, excitement) and more granular mood labels to increase recommendation relevance.
- **Playlist Recommendations:** Move from single-song outputs to full playlist generation via Spotify's API, enabling deeper engagement.
- **Multi-Modal Emotion Input:** Integrate complementary AI techniques such as sentiment analysis from text or prosodic features from voice to expand input flexibility.
- **User Profiles and Adaptive Learning:** Introduce authentication and feedback loops to support long-term personalisation, emotion tracking, and dynamic model improvement.
- **Cloud-Based Scalability:** If user demand grows, migrate to scalable infrastructure (AWS, GCP, Azure) to support persistent storage, analytics, and concurrent usage.

Overall, Melofy stands as a proof of concept for human-aware AI, one that demonstrates how emotion recognition, when combined with responsible design, can improve personalisation in a meaningful, real-time way.

9.3 Limitations

The following table outlines the current limitations of Melofy, as observed during development and testing.

Limitation	Description
Emotion Granularity	The system recognises 7 facial emotions but maps them to only 4 music moods, limiting nuance.
Random Song Selection	Recommendations are randomly sampled from mood-matched songs, without adapting to user preferences.
Model Confusion	Similar expressions such as *fear*, *surprise*, and *angry* are occasionally misclassified.
Web-only Design	The app is optimised for desktop and may not function fully on mobile devices or small screens.
Single Modality Input	Only facial expression is used; emotional cues from voice or text are not yet supported.
No User Personalisation	There is no login or tracking of individual user preferences or mood history.

Table 8 Limitations of the Project

10. References

ElenaRyumina. (n.d.). *face_emotion_recognition* [Machine learning model]. Hugging Face. Retrieved May 8, 2025, from https://huggingface.co/ElenaRyumina/face_emotion_recognition

Kulkarni, A., Mahajan, P., Nimbokar, G., Raut, N., & Tupe, U. (2023). *AI based song recommendations system*.

Mahadik, A., Milgir, S., Patel, J., Kavathekar, V., & Jagan, V. B. (2021). *Mood based music recommendation system*.

Orzanai. (n.d.). *Moodify* [Dataset]. GitHub. Retrieved May 9, 2025, from https://github.com/orzanai/Moodify

Python Poetry. (n.d.). *Managing dependencies*. Poetry Documentation. Retrieved May 8, 2025, from https://python-poetry.org/docs/managing-dependencies/

Sambare, M. (n.d.). FER-2013: Facial Expression Recognition Dataset [Dataset]. Kaggle. Retrieved May 18, 2025, from https://www.kaggle.com/datasets/msambare/fer2013

Spotify. (n.d.). *Web API*. Spotify for Developers. Retrieved May 9, 2025, from https://developer.spotify.com/documentation/web-api

Streamlit. (n.d.). *Deploy your app on Community Cloud*. Streamlit Docs. Retrieved May 12, 2025, from https://docs.streamlit.io/deploy/streamlit-community-cloud/deploy-your-app/deploy

11. Appendices

A. Project Documentation

This appendix documents Melofy, an AI-powered web app that recommends music based on real-time facial emotion detection. It integrates deep learning, computer vision, and public music datasets to showcase AI in a human-centred context.

The codebase, data, and deployment pipeline were designed for modularity, reproducibility, and usability, with all key components version-controlled and documented on GitHub.

A.1 GitHub Repository

- **Repository Name**: AI_a3_repo_group_5
- **Access**: GitHub Link – Public Repository
- **Deployment URL**: Streamlit App – Melofy

A.2 Project Structure

```
AI_a3_repo_group_5/
│
├── .venv/                   # Local virtual environment (excluded from repo)
├── data/
│   ├── 278k_labelled_uri.csv  # Moodify dataset (Spotify URIs and mood labels)
│   └── FER-2013/            # FER-2013 facial emotion dataset (local use only)
│
├── notebooks/
│   ├── moodify_eda.ipynb      # EDA on Moodify dataset
│   └── facial-expression-recognition.ipynb  # Model testing on FER-2013
│
├── reports/
│   └── AI_a3_report_24611687.pdf  # Final report (PDF version)
│
├── src/
│   ├── __init__.py
│   ├── emotion_detection.py   # Emotion detection using Hugging Face ViT
│   └── music_recommender.py   # Mood mapping and song recommendation logic
│
├── app.py                   # Streamlit frontend application
├── requirements.txt         # Dependency list (used by Streamlit Cloud)
├── poetry.lock              # Poetry lock file for package versions
├── pyproject.toml           # Poetry configuration (project + dependencies)
├── README.md                # Project overview and setup instructions
├── .gitignore               # Ignore virtual environments and cache
└── .python-version          # Python version config for Pyenv
```

A.5 Setup Instructions (for local use)

Prerequisites

- Python 3.10+
- Poetry
- Git

1. **Clone the repository**
   git clone https://github.com/kathegolon-uts/AI_a3_repo_group_5
   cd AI_a3_repo_group_5
2. **Set up virtual environment**
   poetry install
3. **Run the app locally**
   poetry run streamlit run app.py

A.6 Deployment Notes

- The app is deployed on **Streamlit Community Cloud**.
- No authentication or database is used.
- All image processing occurs in-session and is discarded after inference.
- GitHub commits to the main branch trigger auto-deployment.

A.7 Documentation and References

- **Hugging Face Model:** https://huggingface.co/shrestha1/vit-Facial-Expression-Recognition
- **Dataset sources:**
  - FER-2013 Dataset
  - Moodify Dataset

B. User Guide

Melofy is a web-based application that uses AI to detect your facial expression and recommend a Spotify song that matches, or uplifts, your mood. It is free to use and requires no login or downloads.

B.1 How to Access the App

- **URL**: https://spotimood.streamlit.app/
- Compatible with Chrome, Firefox, Safari, and Edge.
- Best experience on desktop (camera input may vary on mobile).

B.2 System Requirements

Requirement	Description
Internet	Stable internet connection recommended
Browser	Modern browser (Chrome preferred)
Camera Access	For real-time selfie capture (optional)
File Upload Option	JPEG or PNG images only

B.3 How to Use Melofy

Step 1: Read and Accept Consent

Before using the app, you must tick the **consent checkbox** to agree that your image will be processed temporarily and anonymously during the session.

Step 2: Upload or Take a Photo

Choose between two options:

- **Take a selfie** using your webcam
- **Upload an image** (JPG or PNG)

Tip: Use a well-lit image that clearly shows your face.

Step 3: Let the AI Detect Your Mood

The AI model analyses your facial expression and classifies it into one of the following emotions:

- **Happy**
- **Calm**
- **Energetic**

- **Sad**

Step 4: Receive a Song

Melofy shows a **Spotify-embedded track** matching your mood.

- If you're *sad*, it gives you two tracks: one that reflects your mood and one to lift it.

Step 5: Give Feedback (Optional)

Click 👍 or 👎 to tell us whether you liked the recommendation. This helps us understand user experience (no data is stored).

B.4 Example Use Case

1. A student feeling sad uploads a selfie.
2. The AI detects "sad" and recommends a chill acoustic track and an energetic song to uplift the mood.
3. The student listens, enjoys it, and clicks 👍.

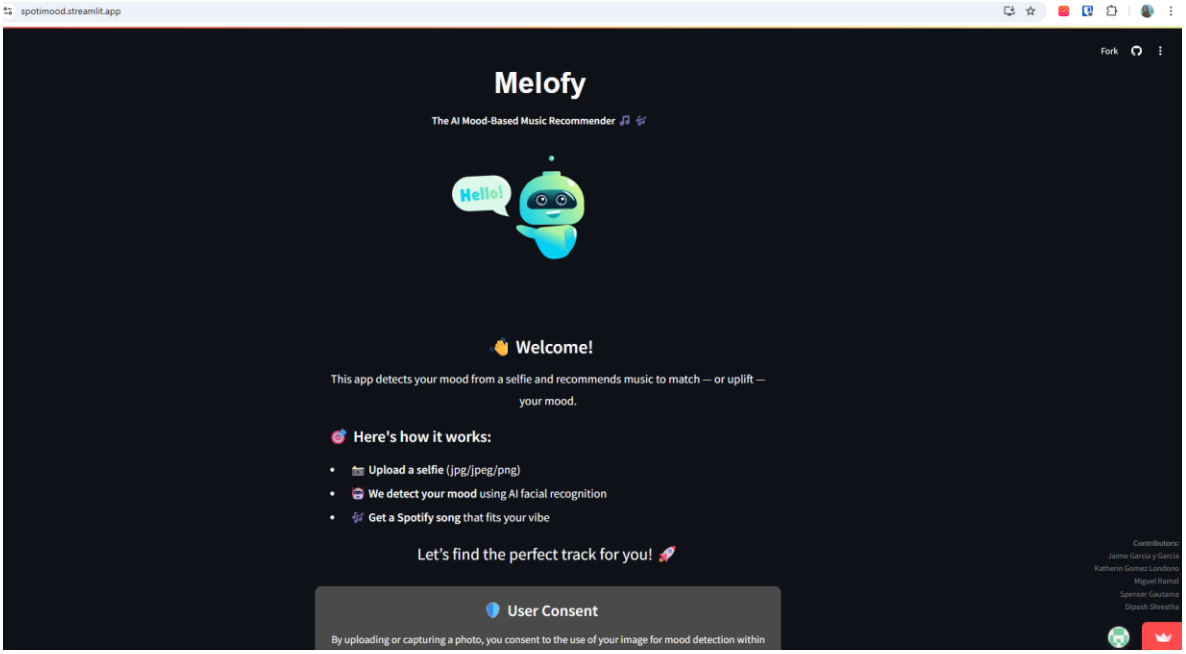
B.5 Troubleshooting Tips

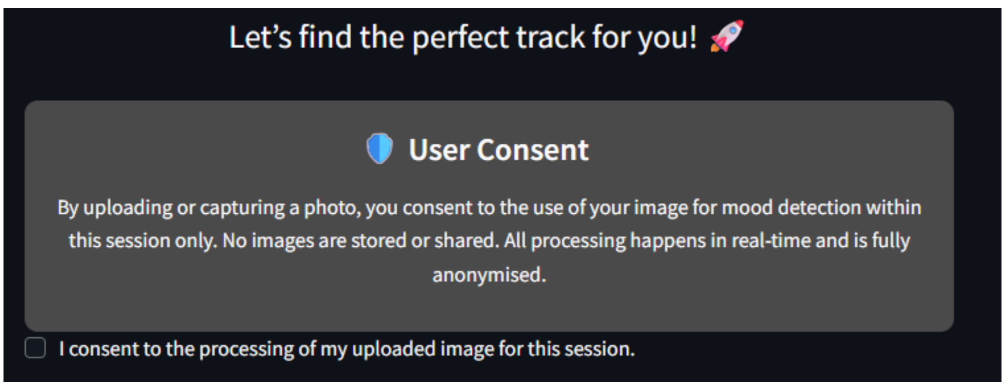
Issue	**Solution**
Image not uploading	Make sure it's in JPG or PNG format
Camera not working	Ensure browser has permission to use the webcam
Song not loading	Refresh the page or check your network connection
No emotion detected	Try a clearer or more front-facing image

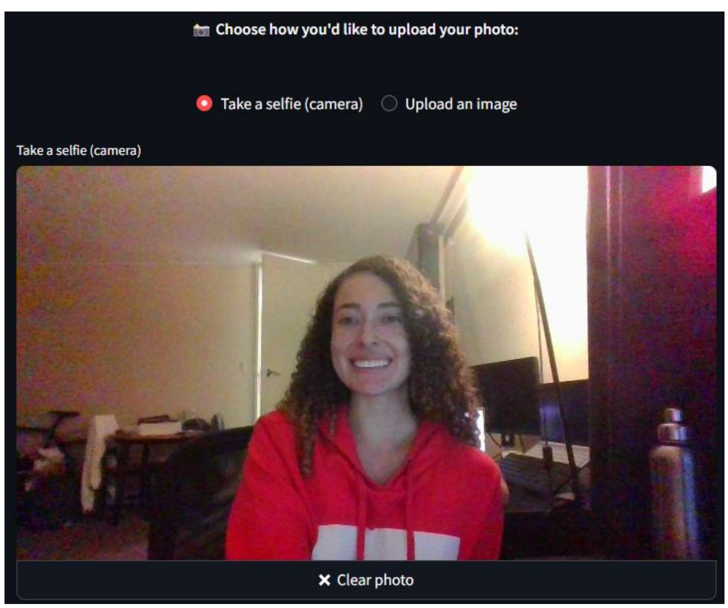
B.6 Privacy and Data Use

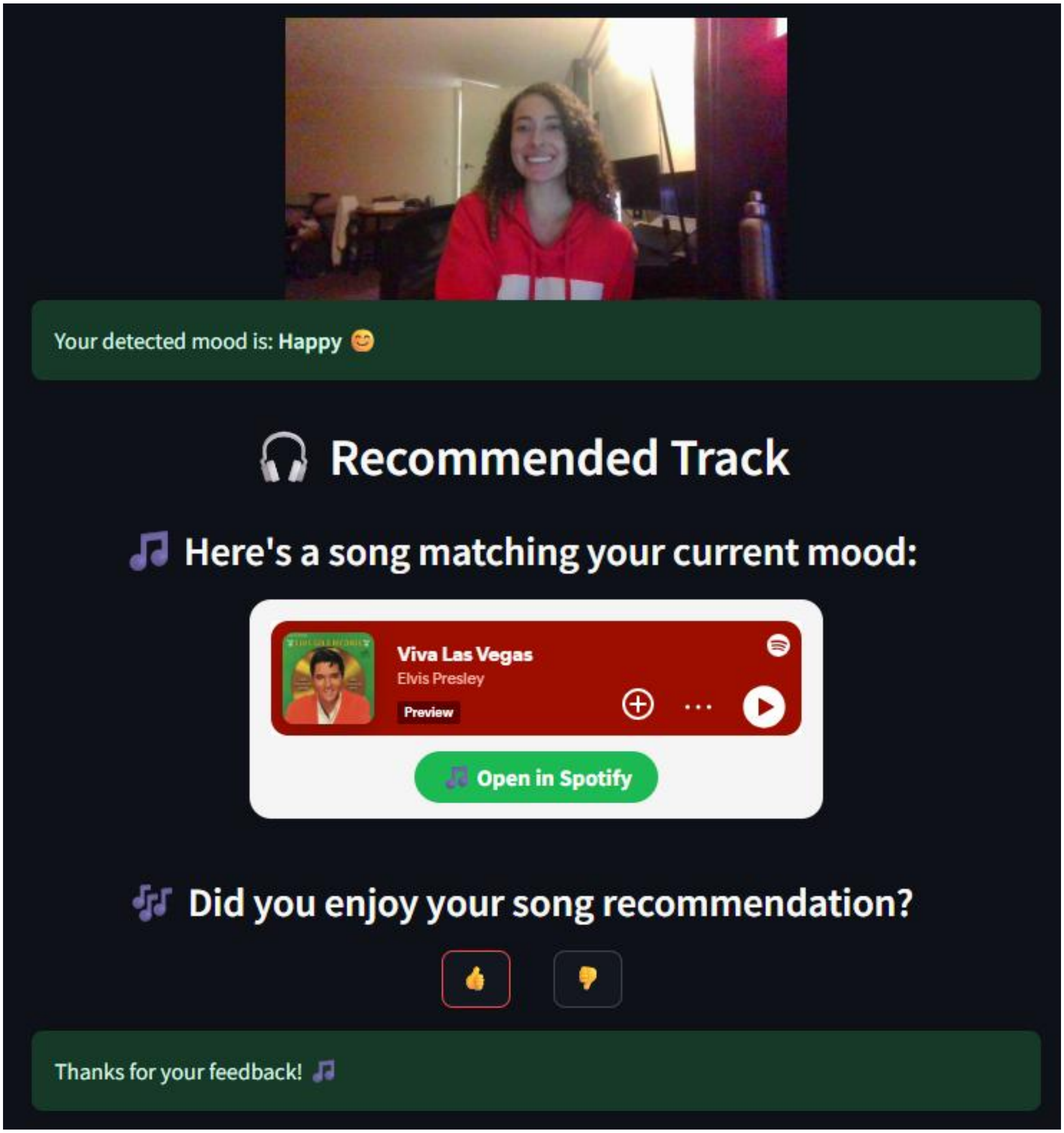
- Your image is **not saved** and is discarded immediately after processing.
- No personal information is collected.
- The system runs entirely within your session.

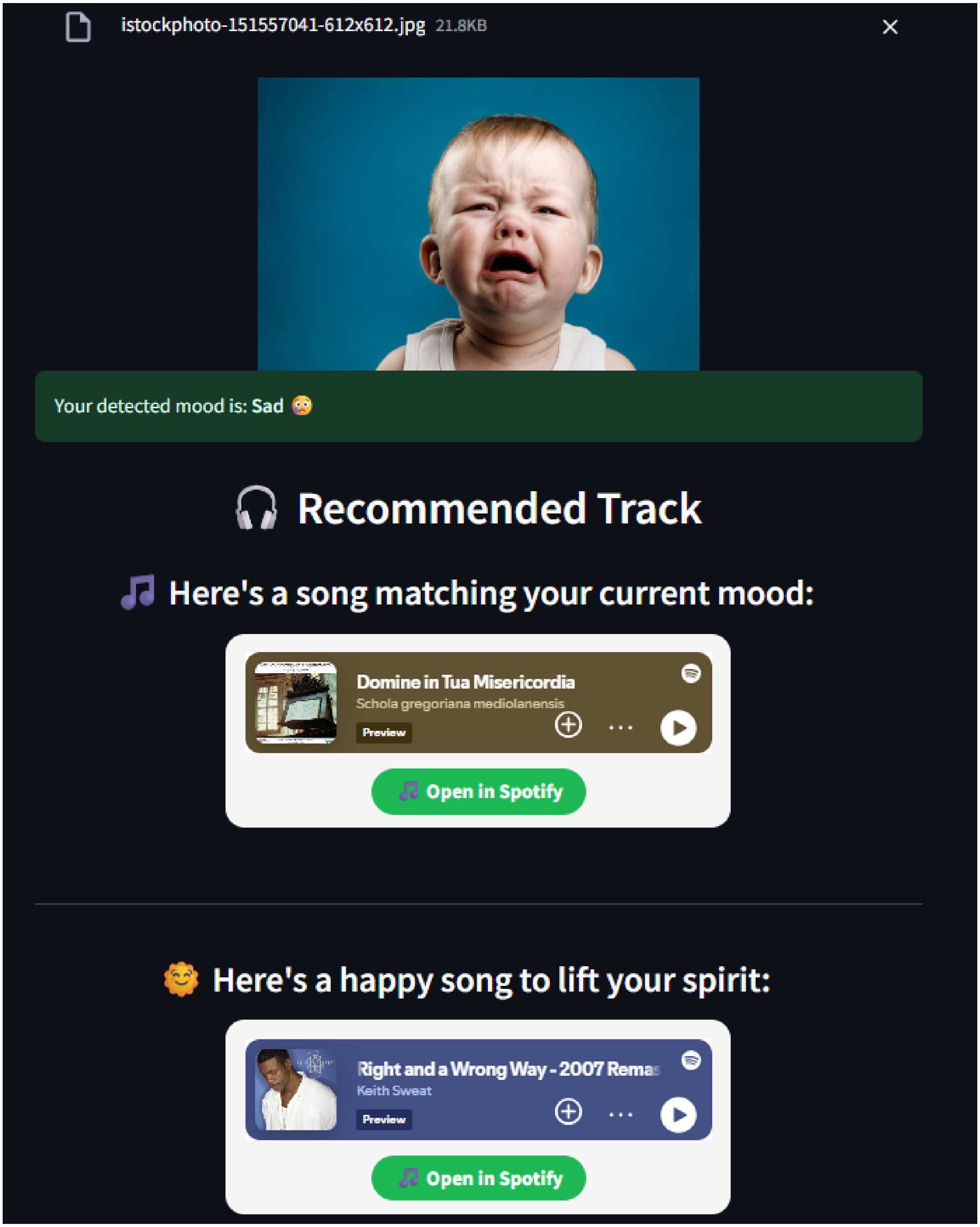
C. Additional Data








Your detected mood is: Happy 😊
🎧 Recommended Track
🎵 Here's a song matching your current mood:
Viva Las Vegas
Elvis Presley
Preview
🎵 Open in Spotify
🎶 Did you enjoy your song recommendation?
👍
👎
Thanks for your feedback! 🎵


istockphoto-151557041-612x612.jpg 21.8KB
Your detected mood is: Sad
Recommended Track
Here's a song matching your current mood:
Domine in Tua Misericordia
Schola gregoriana mediolanensis
Preview
Open in Spotify
Here's a happy song to lift your spirit:
Right and a Wrong Way - 2007 Remas
Keith Sweat
Preview
Open in Spotify