EGCI486: Image Processing Final Project

Music Recommendation System Based on Facial Emotion Detection

Reported by

Pakapak Jungjareon	6481328
Andaman Jamprasitsakul	6481331
Thitirat Kulpornpaisarn	6580871

Submitted to

Asst. Prof. Dr. Narit Hnoohom

COMPUTER ENGINEERING INTERNATIONAL PROGRAM

MAHIDOL UNIVERSITY INTERNATIONAL COLLEGE

2025

Table of Contents

Title of the project

Music Recommendation System Based on Facial Emotion Detection

Names of the members

Pakapak Jungjareon	6481328
Andaman Jamprasitsakul	6481331
Thitirat Kulpornpaisarn	6580871

Background, Objective, and Scope

Background

This project aims to develop an intelligent song recommendation system that dynamically adapts to a user's emotional state in real-time. By analyzing facial expressions captured through a webcam, the system detects emotions such as happiness, sadness, anger, etc., and recommends songs that best match the user's mood. By integrating facial emotion recognition with Spotify's music database, the system provides a personalized and engaging listening experience that seamlessly connects image processing, emotion, and music.

Objectives

1. To create a system that detects human emotions using facial expression recognition and recommends songs that correspond to the detected mood.
2. To implement real-time face detection and emotion classification using OpenCV and You Only Look Once (YOLO).
3. Provides a real-time user interface that displays the webcam feed alongside detected faces, bounding boxes, and emotion labels. Based on these detected emotions, the system generates personalized song recommendations and presents the recommended tracks directly through the interface.
4. Songs are sourced from Spotify via Spotipy which is the Python library for the Spotify Web API.

Scopes

Real-Time Face Detection and Emotion Classification

- Use OpenCV to capture video frames from a webcam.
- Apply trained YOLO11n to detect faces and draw bounding boxes in real time.
- Classify facial emotions (angry, contempt, disgust, fear, neutral, sad, surprise, happy) using trained YOLO11n-cls.

Song Recommendation System

- Build Preference metrics from the user's selected playlist
- Map detected emotions to corresponding song moods or genres.
- Retrieve suitable tracks using the Spotify Web API and automatically queue a track for the user.

Overall

Support multiple faces, stabilize emotion signals, and keep Spotify playback smooth (no abrupt stops). The User Interface Overlay will be used to display the real-time feed from the webcam with bounding boxes and emotion labels, and the recommended songs.

Limitations

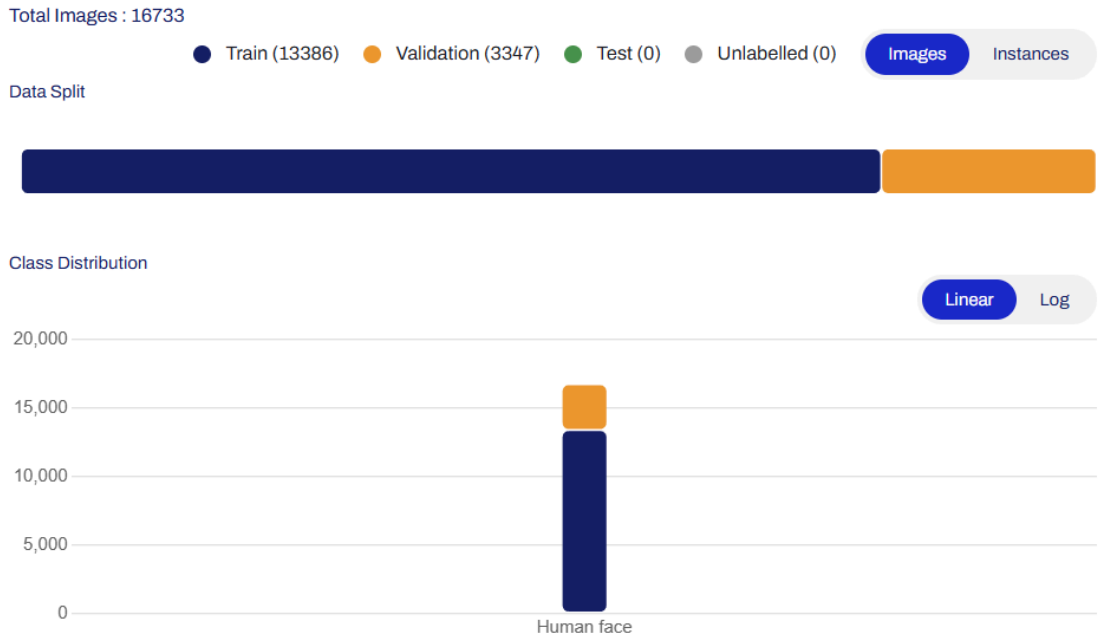
- Emotion detection accuracy depends on lighting, camera quality, and face visibility, which may cause fluctuating or incorrect emotional interpretations.
- Users must have a Premium Subscription for Spotify
- Lyrics, mood, and semantic audio features are not used, as the recommendation model relies only on numeric acoustic features (valence, energy, tempo, etc.).

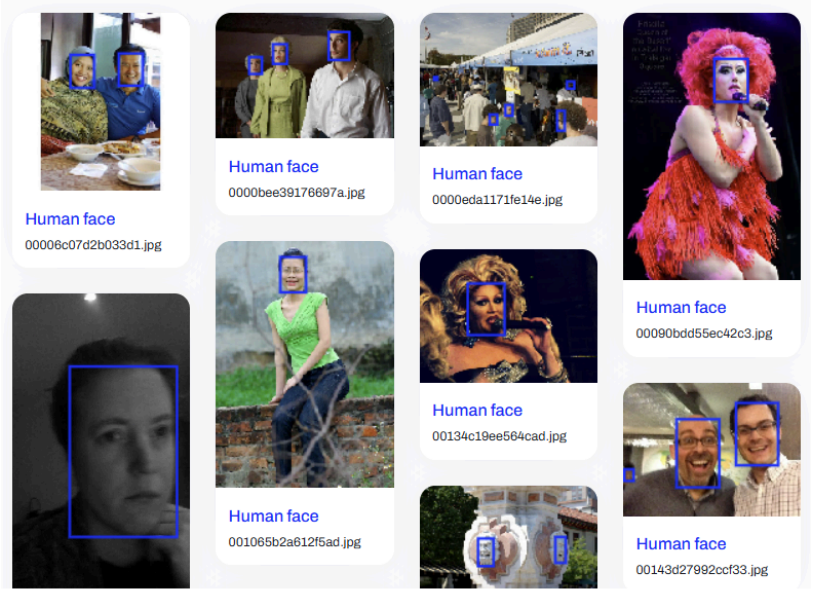
- Referenced Spotify data affects the clustering evaluation, as it has not been updated to catch up with the newer songs. The reason behind using a reference file is the API endpoint for getting audio features is deprecated by the provider.

Methodology

Face Detection Model

Dataset: [Face-Detection-Dataset]





The dataset contains 16,733 images in total, including

- Train set: 13,386 images
- Validation set: 3,347 images

The dataset has only one class, *Human Face*, and a single image can have more than one Human Face labelled on it.

Training Setup

Architecture: YOLO11n

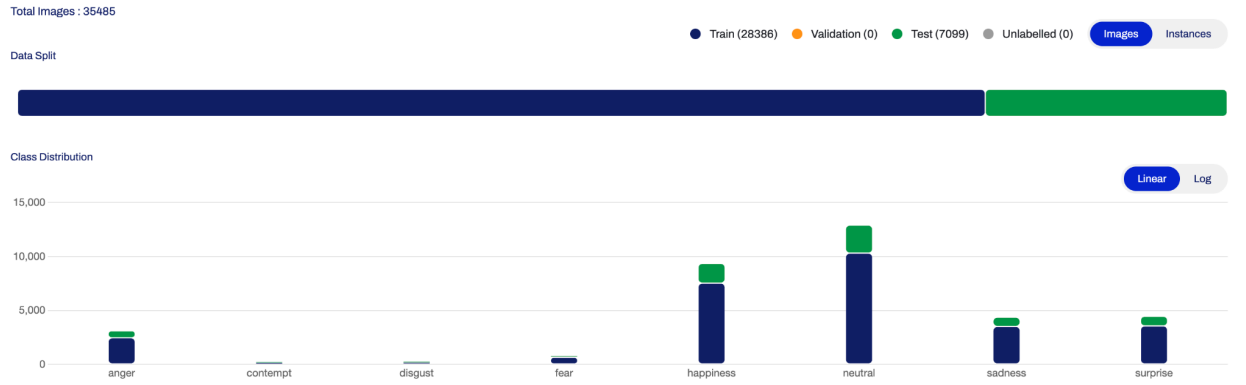
Configurations:

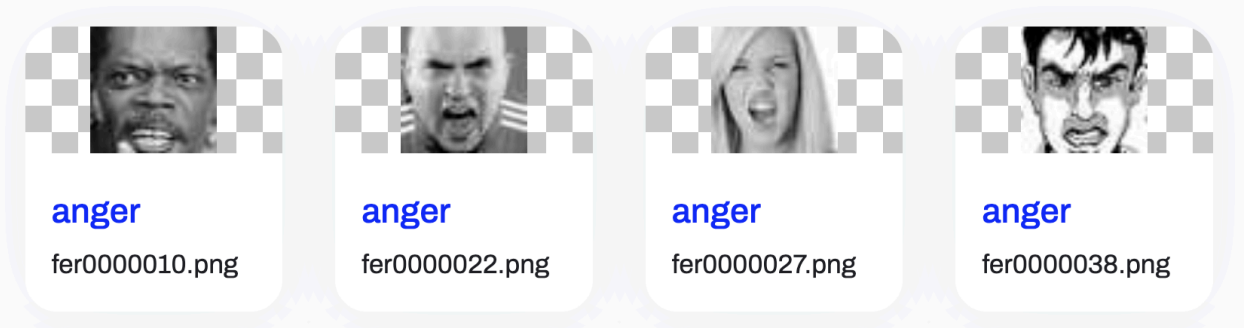
Pretrained	Yes
Epochs	20
Image Size	640
Patience	100
Cache	None
Device	GPU
Batch Size	10

Emotion Classifier Model

For the dataset for the Emotion Classifier model, we gathered two common emotion datasets that are widely used. One is FER2013, which contains 35,485 images, and another one is FERPlus (a newer version of FER2013) that provides 78,293 images.

Dataset: FER2013





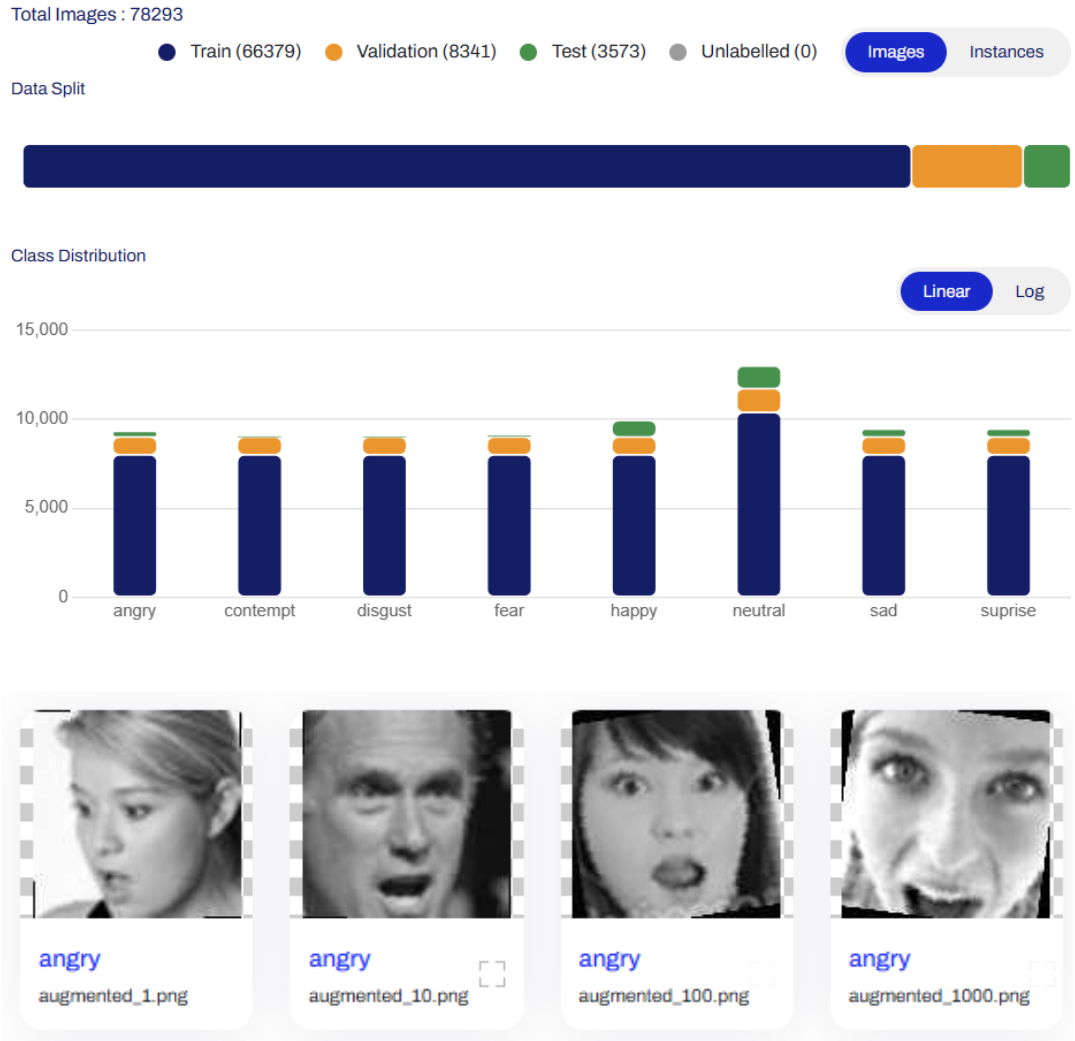
The dataset contains 35,485 images (48 x 48 pixels) in total, including:

- Train set: 28,386 images
- Test set: 7,099 images

The dataset has 8 classes:

- *Angry*
- *Contempt*
- *Disgust*
- *Fear*
- *Happy*
- *Neutral*
- *Sad*
- *Surprise*

Dataset: [FERPlus]



The dataset contains 78,293 images (112 x 112 pixels) in total, including:

- Train set: 66,379 images
- Validation set: 8,341 images
- Test set: 3,573 images

The dataset has 8 classes:

- *Angry*
- *Contempt*

- *Disgust*
- *Fear*
- *Happy*
- *Neutral*
- *Sad*
- *Surprise*

From these two datasets, we decided to choose the FERPlus to use with our application for two main reasons. One of which is the available samples for the classifier to train, validate, and test. Another reason is that FERPlus offers more balanced data between all emotion classes.

Training Setup

Architecture: YOLO11n-cls

Configurations:

Pretrained	Yes
Epochs	20
Image Size	640
Patience	100
Cache	None
Device	GPU
Batch Size	10

After detecting each face with the YOLO-based face detector, the system crops the face region, converts it to grayscale, and resizes it to the classifier's required input size.

This normalized cropped face is then fed into the emotion classification model, ensuring compatibility between training data and data used in the application.

Personalization Metrics Construction

1. User's Playlist Feature Extraction

   The system matches each song from the user's selected playlist to spotifydata_hit_ dataset, producing a feature vector (danceability, energy, valence, tempo, etc.) for every matched track. This creates a personalized feature library representing the user's listening preferences.

2. Normalize Features and Build Clusters

   All track feature vectors are standardized using *StandardScaler* and grouped using K-Means, with one cluster per emotion category. Each cluster center is compared to assumed emotion profiles that contain the target features for each emotion, allowing clusters to be labeled with the closest possible emotion based on cosine similarity.

Bridging Face Detection and Emotion Classification

To connect both model into a streamlined process, we use face detection model to locate the face on the webcam feed, crop the face, and send the focus ared to the emotion classifier. However, as the emotion classifier is trained on a grayscale dataset, the RGB input must be converted to grayscale to use with the emotion classifier. We achieve this part by using the cv2.cvtColor function to convert the cropped image into grayscale for the sake of solely emotion classification.

Recommendation Engine & Track Queueing

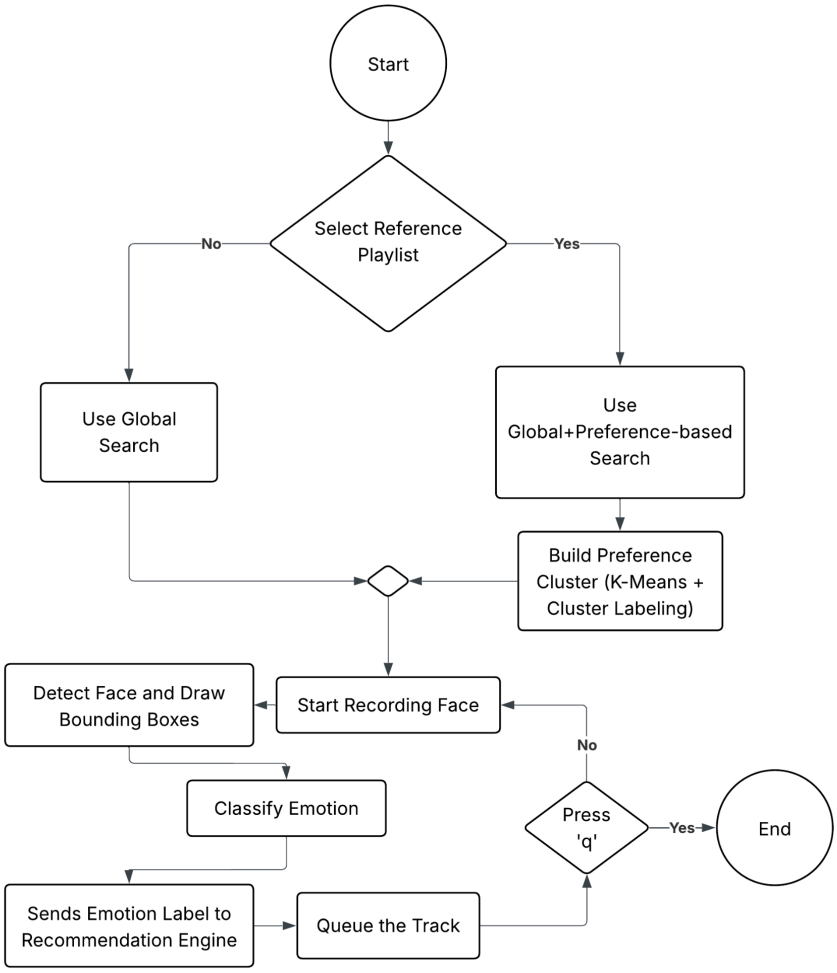
Each detected emotion is mapped to an idealized feature vector (EMO_MAP) defined by valence, energy, and tempo. Tracks from the corresponding playlist cluster are ranked by cosine

similarity to this target vector, with slight noise added for variety. These personalized results are combined with emotion-matched global Spotify tracks, filtered for recency, and delivered as final recommendations that can be queued to the user's active Spotify device.

User Interface

The user interface is generated directly within the video stream using the OpenCV library. Bounding boxes, emotion labels, and recommendation text are drawn onto each frame before display, allowing the system to present detection results and selected tracks in real time without a separate GUI framework.

Complete Application Flow



Additional Emotion Detection and Recommendation Mechanics

Instant emotion change: When the stable emotion (emotion held consistently for ≥0.5 seconds) switches to a new label, the system immediately generates new recommendations for that emotion and queues the appropriate tracks on the user's Spotify device.

Song ending with stable emotion: If the stable emotion remains unchanged and the current song has less than approximately 20 seconds remaining, the system preemptively queues the next track that matches the same emotion, maintaining a smooth, mood-aligned listening experience.

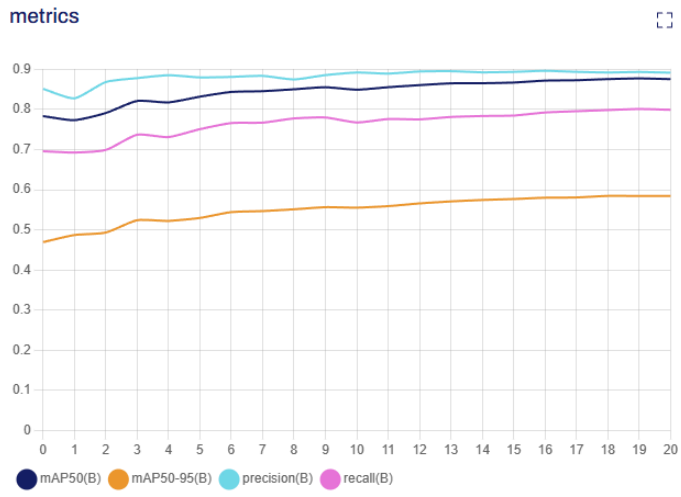
Emotion voting (multi-face): the system assigns an emotion to each face and applies a majority vote to produce a single scene emotion. Ties are resolved using temporal smoothing (preferring the previous frame) or defaulting to neutral, ensuring stable and consistent scene-level emotion detection.

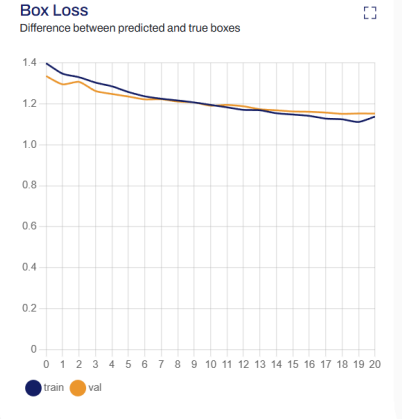
Experiment results

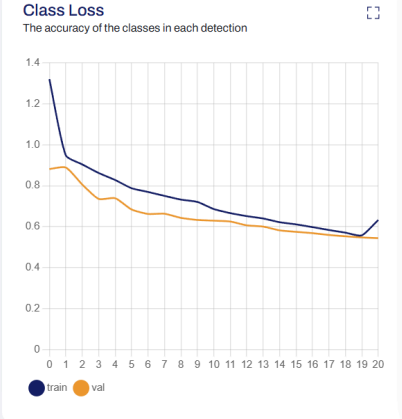
Face Detection Model

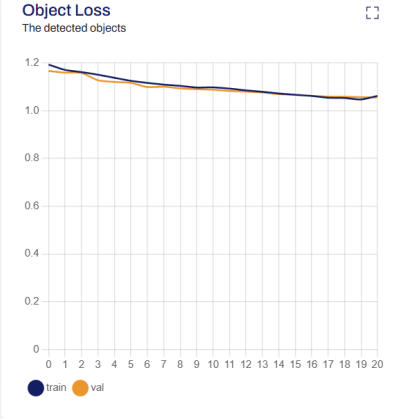
Path to Ultralytics HUB: https://hub.ultralytics.com/models/YaCSN3TYdCmXVLnAFtZW

```
Validating /content/runs/detect/train/weights/best.pt...
Ultralytics 8.3.232 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (NVIDIA L4, 22693MiB)
YOLO11n summary (fused): 100 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 168/168 11.6it/s 14.5s
                   all       3347      10299      0.891      0.799      0.875      0.584
Speed: 0.1ms preprocess, 0.8ms inference, 0.0ms loss, 0.8ms postprocess per image
Results saved to /content/runs/detect/train
Ultralytics HUB: Syncing final model...
: 100% ━━━━━━━━━━━━ 5.2MB 4.0MB/s 1.3s
Ultralytics HUB: Done ✅
Ultralytics HUB: View model at https://hub.ultralytics.com/models/YaCSN3TYdCmXVLnAFtZW 🚀
```







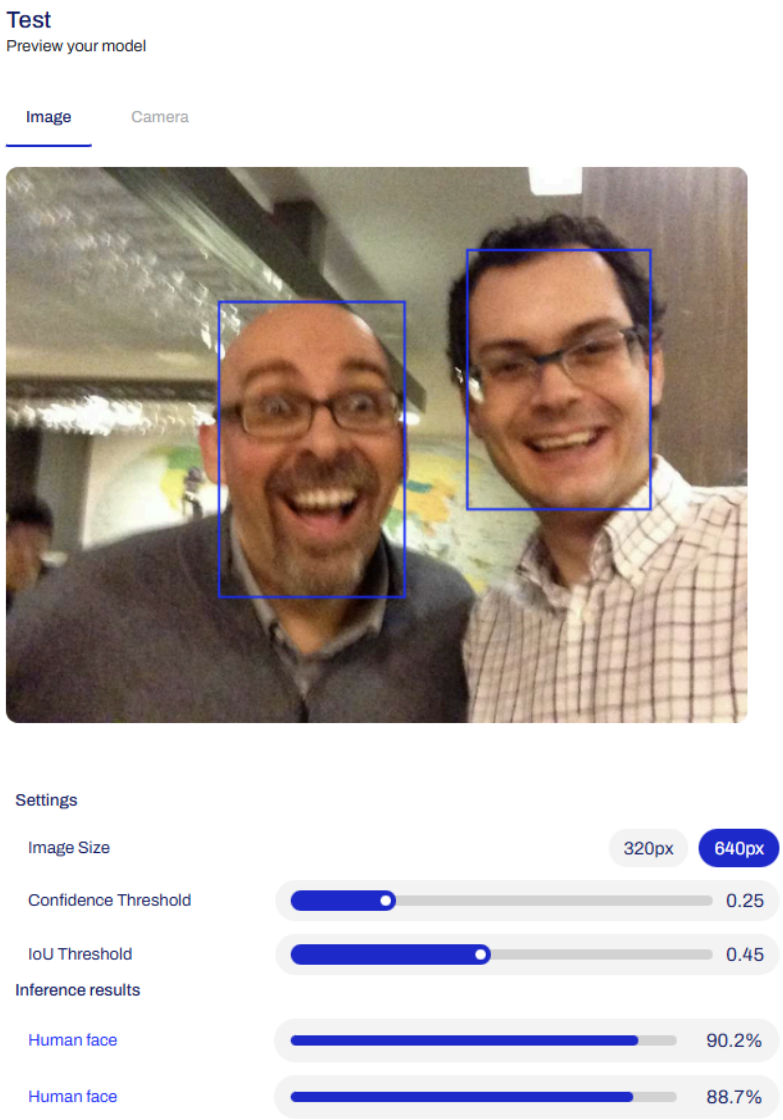


Mean Average Precision 50 (mAP50(B)): 0.875

Mean Average Precision 50-95 (mAP50-95(B)): 0.584

Precision (B): 0.891

Recall (B): 0.799

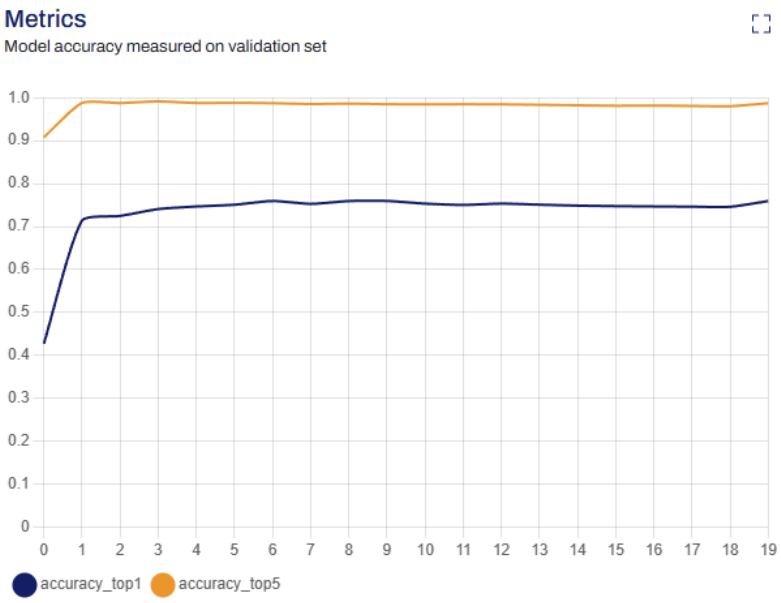


Emotion Classifier Model

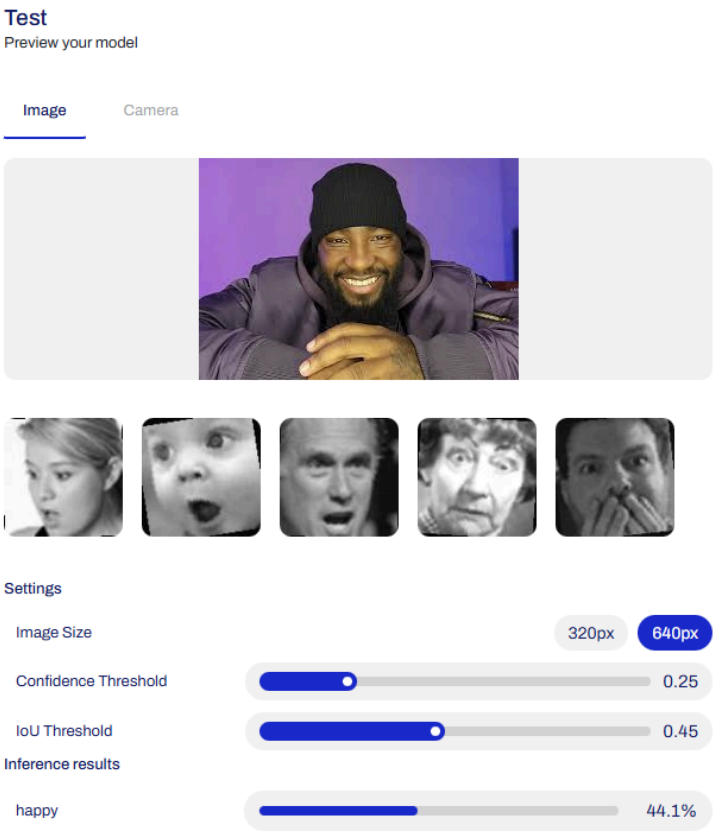
Path to Ultralytics HUB: https://hub.ultralytics.com/models/6kxLxRsMOYfr0EPtU0bP

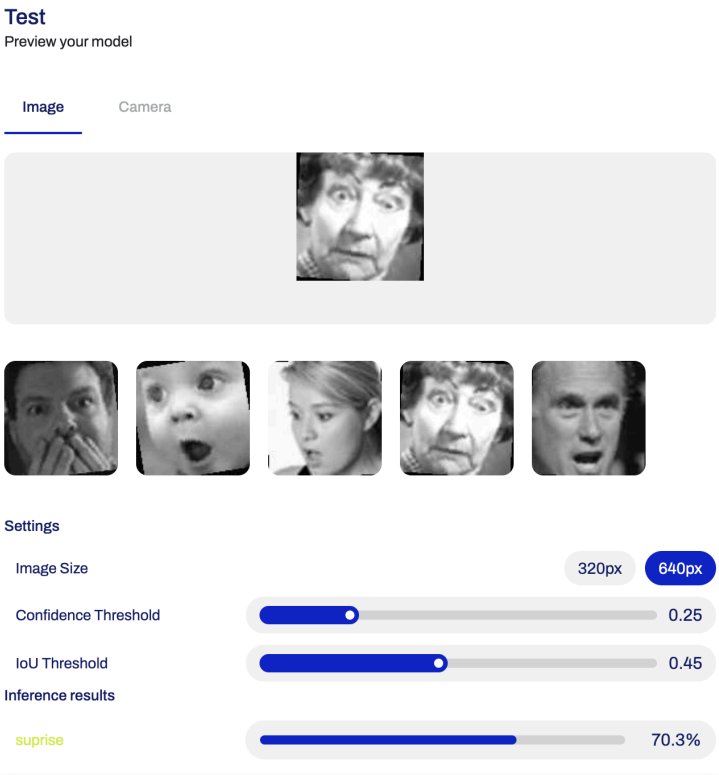
```
20 epochs completed in 2.017 hours.
Optimizer stripped from /content/runs/classify/train/weights/last.pt, 3.2MB
Optimizer stripped from /content/runs/classify/train/weights/best.pt, 3.2MB

Validating /content/runs/classify/train/weights/best.pt...
Ultralytics 8.3.233 🚀 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (NVIDIA L4, 22693MiB)
YOLO11n-cls summary (fused): 47 layers, 1,536,272 parameters, 0 gradients, 3.2 GFLOPs
WARNING ⚠ Skipping /content/datasets/ferplus.zip unzip as destination directory /content/datasets/ferplus is not empty.
train: /content/datasets/ferplus/train... found 66379 images in 8 classes ✅
val: /content/datasets/ferplus/validation... found 8341 images in 8 classes ✅
test: /content/datasets/ferplus/test... found 3573 images in 8 classes ✅
               classes   top1_acc   top5_acc: 100% ━━━━━━━━━━━━ 418/418 20.2it/s 20.7s
                   all       0.76      0.987
Speed: 0.5ms preprocess, 0.6ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to /content/runs/classify/train
Ultralytics HUB: Syncing final model...
: 100% ━━━━━━━━━━━━ 3.1MB 2.7MB/s 1.1s
Ultralytics HUB: Done ✅
Ultralytics HUB: View model at https://hub.ultralytics.com/models/6kxLxRsMOYfr0EPtU0bP 🚀
```

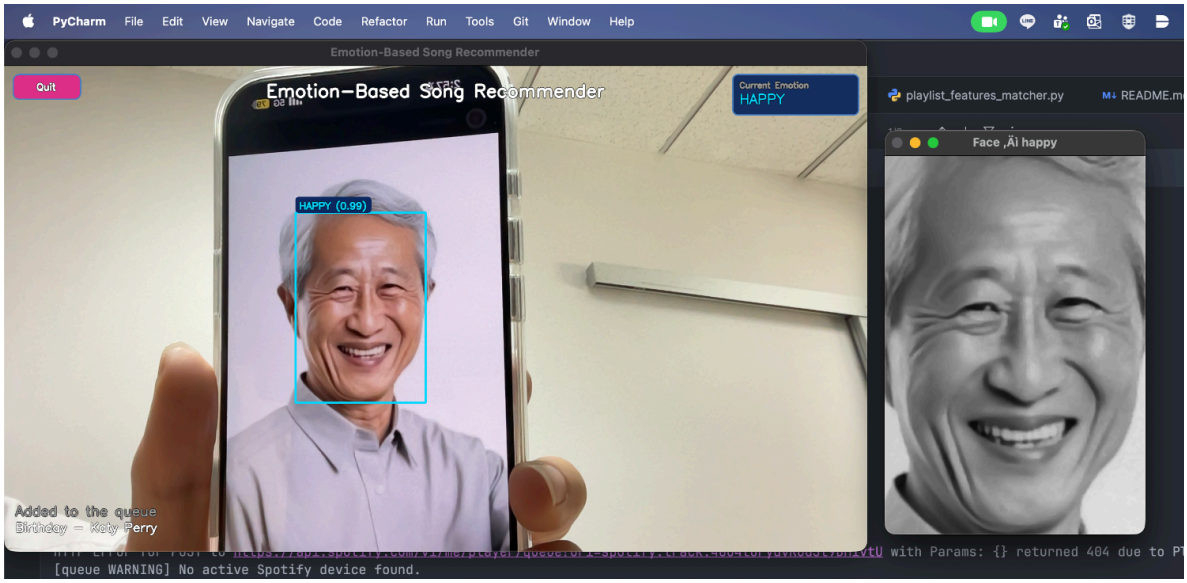


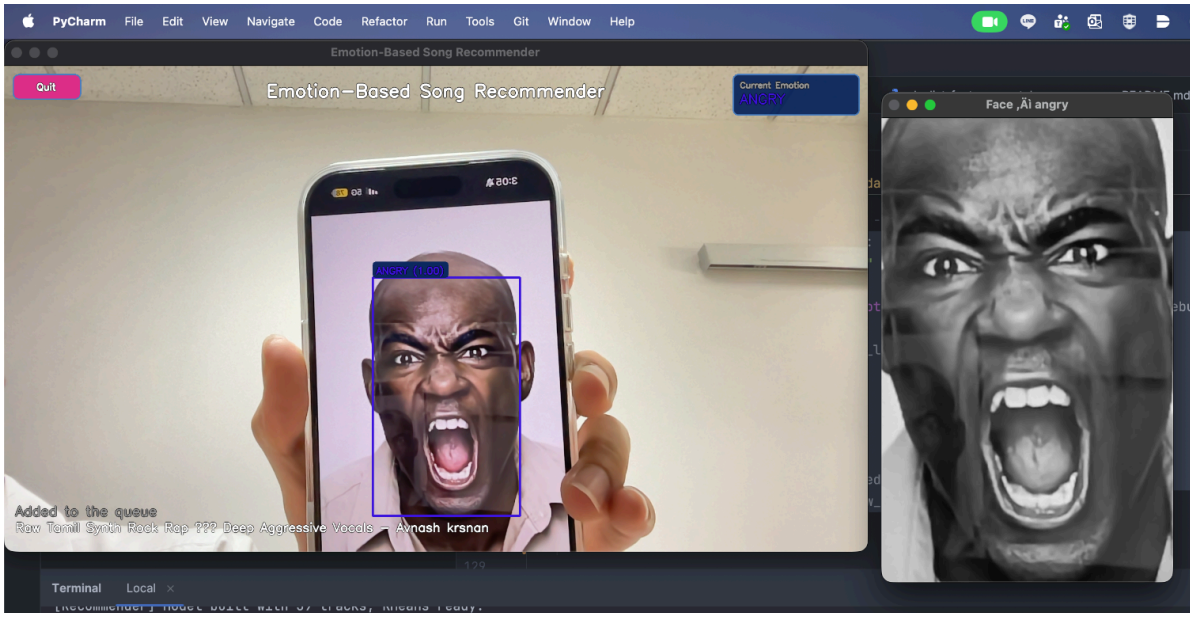
Overall Model Accuracy: 0.76


Test
Preview your model
Image
Camera
Settings
Image Size
320px
640px
Confidence Threshold
0.25
IoU Threshold
0.45
Inference results
happy
44.1%


Test
Preview your model
Image
Camera
Settings
Image Size
320px
640px
Confidence Threshold
0.25
IoU Threshold
0.45
Inference results
suprise
70.3%

Real-time Testing (face detection and emotion classification)





These are the screenshots that illustrate how face detection and emotion classifier models work together. First, on the webcam feed, the face from the input is detected by the face detection model and cropped. Then, the cropped area (shown on the right-hand side) is converted into a grayscale image and fed into the emotion classifier, returning the predicted emotion and confidence score.

Personalization Metrics Construction

```
[Matcher] 39 / 175 tracks matched with features.
[Recommender] Model built with 39 tracks; KMeans ready.
```

From a playlist, certain tracks were successfully matched with full audio features and used to build the user-specific preference profile. These matched tracks were standardized and clustered using K-Means, enabling the system to associate each cluster with an emotion target. This cluster-to-emotion mapping forms the basis of the personalized recommendation metric used during inference.

Recommendation Engine & Track Queueing

```
=== Recommendations for emotion (stable): neutral | reason: emotion changed ===
1. Lofi — LOFI BEATS
   https://open.spotify.com/track/23Vd1pgi6KTPRAusuMEtT9
Queued 1 track(s) to your active Spotify device.
```

```
=== Recommendations for emotion (stable): contempt | reason: emotion changed ===
1. Unwind — KIKI
   https://open.spotify.com/track/4voqanmgMdgpRdtH6NoiC4
Queued 1 track(s) to your active Spotify device.
```

```
=== Recommendations for emotion (stable): surprise | reason: emotion changed ===
1. experimental music volume 01 — Django Beaudoin
   https://open.spotify.com/track/3SveDgYiRhEXm2TTolSnf5
Queued 1 track(s) to your active Spotify device.
```

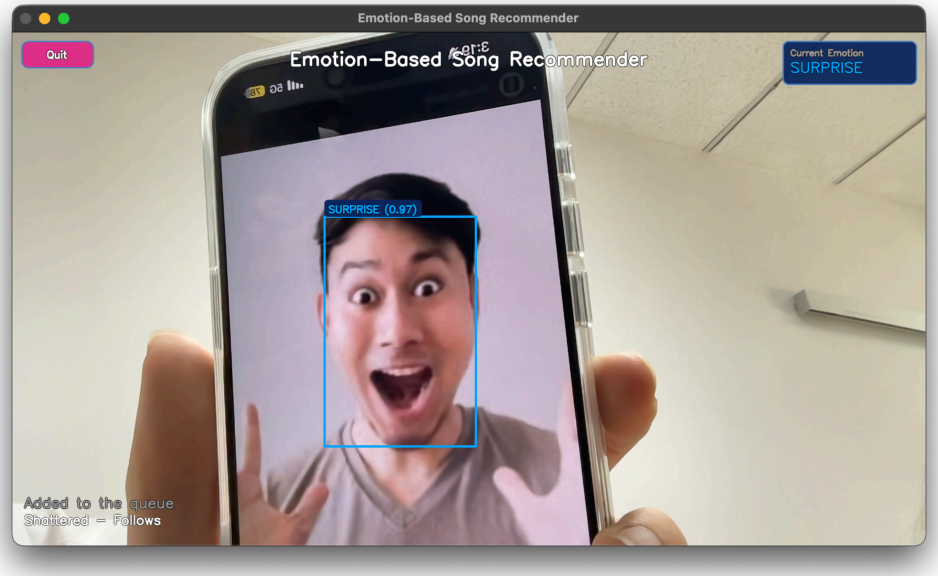
```
=== Recommendations for emotion (stable): sad | reason: emotion changed ===
1. Sad Acoustic Ballad Backing Track In A Minor — Tom Bailey Backing Tracks
   https://open.spotify.com/track/6J7xGtBO4GdmxvNMFAzYBp
Queued 1 track(s) to your active Spotify device.
```

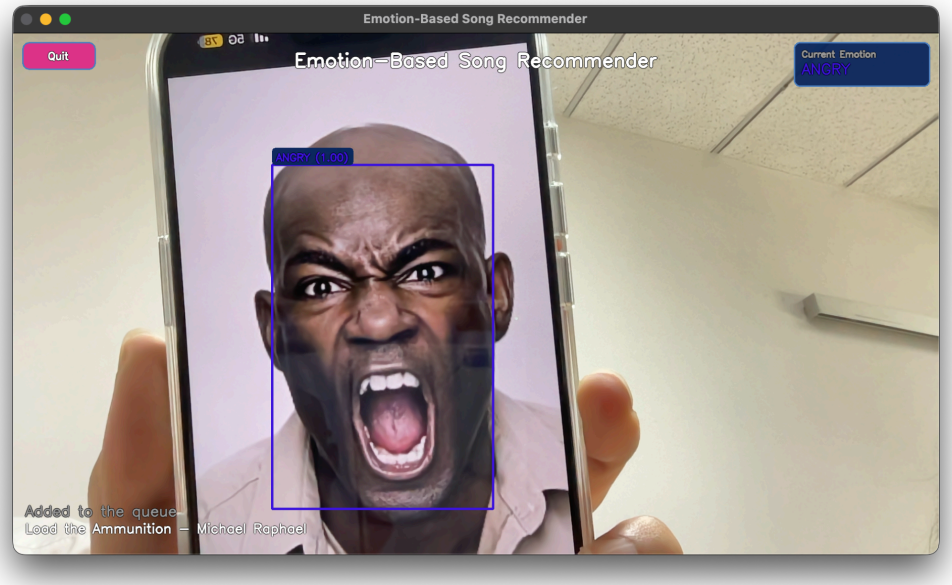
```
=== Recommendations for emotion (stable): happy | reason: emotion changed ===
1. Happy Pop Feel Good Vibes Morning Coffee Upbeat Acoustic Instrumental Pop for Vlogs and Lifestyle Videos — Inspirational Vibes, AI-Assisted
   https://open.spotify.com/track/007GCUDb4xHjZtxWjwMbCp
Queued 1 track(s) to your active Spotify device.
```

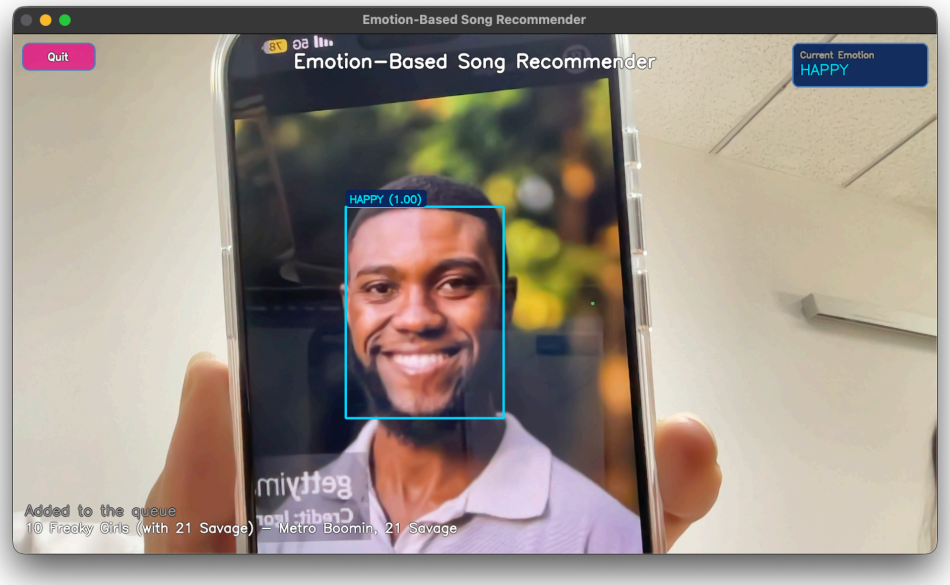
```
=== Recommendations for emotion (stable): angry | reason: emotion changed ===
1. The Middle — Zedd, Maren Morris, Grey
   https://open.spotify.com/track/09IStsImFySgyp0pIQdqAc
Queued 1 track(s) to your active Spotify device.
```

The system successfully generated real-time song recommendations based on emotion label inputs (neutral, contempt, surprise, sad, happy, and angry). For each detected emotion, the recommender returned a matched track, combining personalized playlist-based preferences with emotion-aligned global Spotify results (or solely global results if the user does not select the playlist to be preference metrics), and automatically queued it to the user's active Spotify device. The outputs confirm that the queueing mechanism operates smoothly and that the engine consistently adapts recommendations as the user's emotional state changes.

Overall


Emotion-Based Song Recommender
Quit
Emotion–Based Song Recommender
Current Emotion
SURPRISE
SURPRISE (0.97)
Added to the queue
Shattered – Follows

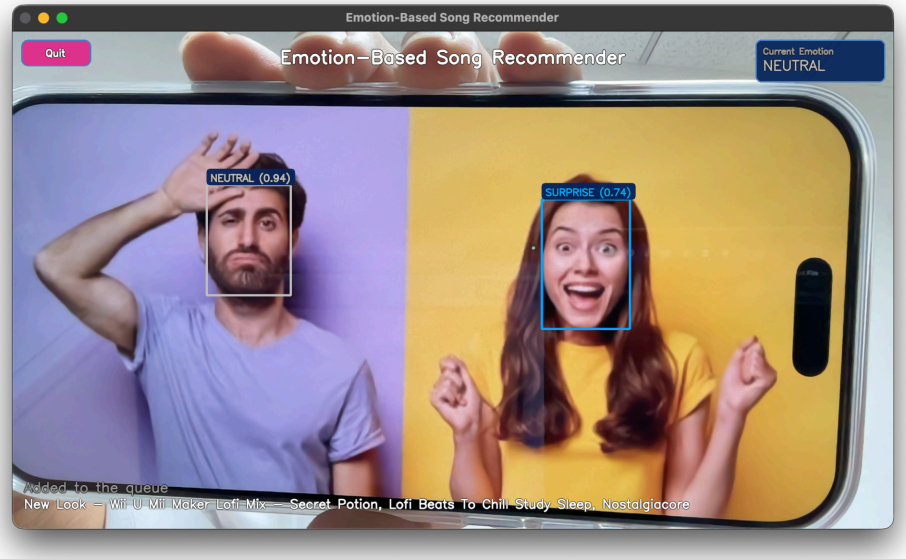

Emotion-Based Song Recommender
Quit
Emotion–Based Song Recommender
Current Emotion
ANGRY
ANGRY (1.00)
Added to the queue
Load the Ammunition – Michael Raphael

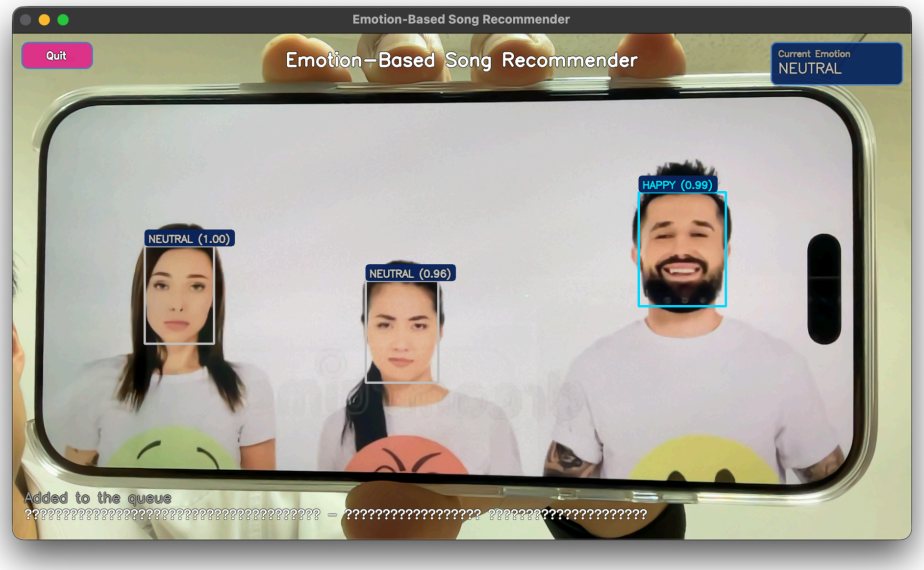


The system integrates real-time facial detection, emotion classification, and music recommendation into a unified pipeline. Live webcam frames are processed to locate faces and assign emotion labels with confidence score, which are then used to select and queue appropriate tracks via the Spotify Web API. The interface displays both the detected emotion and the

corresponding recommended song, demonstrating the system's complete end-to-end functionality.

Multiple Face/Emotion





For multiple face processing, there will be emotion voting to find the final emotion label for the recommendation system. From the screenshot, it is easy if all faces are the same emotion. Nonetheless, if the emotions are different, the final emotion will stick to the majority of emotional appearance, and if it is a tie, the final emotion will be **neutral**.

Conclusion

The implemented system demonstrates that real-time affect-aware recommendation is viable with commodity hardware and APIs. Modular components (emotion detector, recommender, UI overlay) keep the codebase maintainable, while the stabilization logic prevents microexpression noise from polluting the recommendation trigger. Combining personalized embeddings with deduplicated global fallbacks yields recommendations that feel both timely and contextually appropriate, and Spotify queue automation keeps the listening experience smooth.

Discussion and Future Work

The system demonstrates that real-time face detection, emotion classification, and audio-feature–based music recommendation can be integrated into a single functional pipeline. However, several constraints remain. Emotion detection performance is sensitive to lighting conditions, camera quality, and partial occlusions, which directly impact downstream recommendations. The current majority-vote stabilization may also produce inconsistent results when multiple faces express different emotions simultaneously. On the recommendation side, personalization is limited by the availability and coverage of audio features in the referenced Spotify dataset, which may exclude newer tracks and reduce the effectiveness of cluster-based preference modeling. Furthermore, the Spotify playback workflow relies on the presence of an active device session, which can cause interruptions when no session is available, and user must have a premium subscription.

Future work to pursue

- **Enhance robustness:** Expand datasets and test the model under varied lighting and occlusion to improve reliability.
- **Upgrade affect models**: Explore transformer-based or multimodal emotion recognition incorporating cues such as speech or pose.
- **Improve recommendation context**: Include temporal factors, listening history, or user interaction patterns instead of relying solely on playlist features.
- **Refresh feature datasets**: Update the referenced Spotify audio-feature dataset regularly to improve clustering reliability and support newer tracks.