Music Recommendation System through Emotion Facial Reactions

Alexandra Przysucha AJP9010@NYU.EDU
Yuhong YZ9134@NYU.EDU
Andrea Cardiel ALC9588@NYU.EDU

Milestone 3

1. Methodology

Our project makes a dual approach to emotion recognition and music recommendation. We used a deep learning-based emotion recognition model that was trained on the FER-2013 dataset and a music classification and recommendation model that used sequential audio features and a general methodology for mood-scoring logic.
For the music dataset, we had a dataset which had compiled songs from the 1960s to 2010s with each song having audio features such as valence, danceability, energy, tempo, loudness, and speechiness, along with track and artist metadata.

1. We did feature normalization where all the numerical audio features that we looked at and used Standard Scaler to make sure we had a consistent scale for the models.

2. We only chose to use the moods: *happy, sad, angry, fear, surprise*, from the FER dataset because the disgust and neutral emotions were so few in our dataset that it hardly made a difference keeping them or not, and even normally it was hard to distinguish for any well-trained model between those two.

All the songs were classified into five moods (*happy, sad, angry, fear, surprise*) as mentioned above as to why those. We mapped these to moods based on the ideas in the following papers and previous research (which is linked below) linking audio features to emotions:

1. https://kratichoudhary258.medium.com/music-mood-classification-relativity-to-music-therapy-7c44250c45dc

2. https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/

3. https://mct-master.github.io/machine-learning/2020/09/20/Music-Mood-Classifier.html

The main idea that we took from this research is that we can use audio features such as valence and tempo and energy, and more to be able to roughly estimate and map the mood of the song, and although this method is not perfect, with more time and resources such as Librosa to get the feature extractions of each song and use those to classify mood, this is sufficient (for now) for the project even if it is not perfect.

For the mood mapping, it is helpful to refer to Spotify's API definitions of the audio features (which are below from Spotify's documentation linked below) to understand how we picked the threshold and what audio features we looked at to classify songs: https://developer.spotify.com/documentation/web-api/reference/get-audio-features

- Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

- Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

- Instrumentalness predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

- Valence is a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

- Acousticness is a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

- The key the track is in. Integers map to pitches using standard Pitch Class notation. If no key was detected, the value is -1.

- Liveness detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

- Loudness is the overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.

- Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.

- Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both

music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

- Tempo is the overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

Below is how we determined the music mood mapping scoring through the audio features of each song:

- **Happy**
  High valence: The song has a positive emotional tone (`valence > 0.7`).
  High danceability: The song is rhythmic and easy to dance to (`danceability > 0.7`).
  Mode = 1: The song is in a major key, often associated with positive emotions.
  These features together indicate a joyful, uplifting mood.

- **Sad**
  Low valence: The song has a negative emotional tone (`valence < 0.3`).
  Low energy: The song is slow or subdued (`energy < 0.4`).
  Mode = 0: The song is in a minor key, often associated with somber or melancholic emotions.
  These attributes point to a sorrowful or reflective mood.

- **Angry**
  High energy: The song is intense or forceful (`energy > 0.8`).
  Low valence: The song conveys a negative emotional tone (`valence < 0.3`).
  Loudness > -5: The song is loud, emphasizing its aggressive nature.
  This combination reflects an intense, aggressive, or angry mood.

- **Fear**
  Low valence: The song conveys fear or unease (`valence < 0.3`).
  Loudness > -10: The song is somewhat loud, contributing to a sense of urgency or alarm.
  High tempo (`> 150`): A fast-paced song, which can evoke anxiety or panic.
  These features collectively indicate a fearful or tense mood.

- **Surprise**
  High energy: The song is dynamic and vibrant (`energy > 0.7`).
  High tempo (`> 140`): The song is fast-paced, creating an element of excitement.
  Moderate valence (`Between 0.4 and 0.7`): The emotional tone is balanced, neither too positive nor too negative, but unexpected or striking.
  This indicates a mood of excitement or surprise.

For our music dataset, which includes each song's track name, artist, audio features such as valence, danceability, energy, etc., AND now the moods are labeled and mapped to the songs as well, we did the following:

- To address the class imbalance in the assigned moods, we used SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic samples for minority moods, ensuring equal representation during model training.

- We split the dataset into training and validation sets (80-20). Also, the sequential audio features were reshaped to align with the input requirements of the RNN model.

For the FER dataset, which consists of grayscale facial images labeled with emotions such as *happy, sad, angry, fear, surprise*, and others, we chose to only include *happy, sad, angry, fear, surprise*, as explained earlier.

1. We made sure all the images were resized to 48×48 pixels to match the input size required by the CNN model.

2. We did normalization where the pixel values were normalized to a range of [0, 1] by dividing by 255 to reduce computational complexity and enhance convergence during training.

3. We also did data augmentation, using the `ImageDataGenerator` technique, to mitigate class imbalances and improve model generalization. The `ImageDataGenerator` does random rotations, horizontal flips, width and height shifts, and brightness adjustments for the images.

We did 2 models for each of the 2 parts (emotion and music parts).

1.1 Emotion Recognition

We did a fine-tuned CNN using a parameter we had gotten from `Hyperband` earlier in a separate part of our code. The CNN had 2 `Conv2D` layers with filters (64, 128), followed by `MaxPooling2D` layers to reduce dimensionality and extract spatial features, dropout layers (rate = 0.3) to prevent overfitting, and dense layers with ReLU activation for feature abstraction, and a final softmax layer for multi-class classification.
We also used a pre-trained model, the VGG-16 model, implementing our transfer learning knowledge from the class. The model had pre-trained weights. The base VGG16 layers were frozen initially, focusing training on custom dense layers added for emotion classification. We added dropout layers (rate = 0.5) to the dense layers for improved regularization, and a softmax output layer provided probabilities for each emotion class.

1.2 Music Classification

For the music data, we did a CNN and RNN with an attention mechanism.
The CNN processed normalized audio features using 2 `Conv1D` layers (32, 64 filters) for extracting temporal patterns, `MaxPooling1D` layers for dimensionality reduction, and dense layers with ReLU activation for abstraction and a softmax layer for mood classification. The CNN for the music model was trained with categorical cross-entropy loss and the Adam optimizer (learning rate = 0.001).

For the RNN, we used 2 stacked LSTM layers (64 and 32 units) to capture temporal dependencies in sequential audio features. For the RNN, we also had the attention mechanism implemented to prioritize significant temporal patterns, enhancing the model's ability to classify moods. The RNN attention mechanism model was trained with categorical cross-entropy loss, using the Adam optimizer with a learning rate = 0.001. For prevention of overfitting, we did `EarlyStopping` and `ReduceLROnPlateau` callbacks to adjust the learning rates as we trained.

1.3 Integration

To combine these two parts to make an emotion-music recommender, we took the emotion recognition model's output (which was a mood) and fed that into the music classification model to recommend songs based on mood.
We took this further by doing two experiments:

1. Using webcam capturing to determine the user's mood through the emotion recognition models.

2. Using an ensemble method, which combined the predictions from the CNN and VGG16 models using a weighted averaging approach. Between the two models, we calculated the average or most-likely option for the user's mood.

That mood was then fed into the recommender system, which also included a YouTube integration system where songs recommended by the music model based on the user's mood were output with YouTube links. This enabled users to access the tracks directly. The recommender also provided the decade each song was from and how many songs were recommended from each decade.

2. Results and Analysis

2.1 *Emotion Model*

The dataset analysis is already presented in Milestone 2 and earlier in Milestone 3. The finding is that we face a class imbalance problem, so we use SMOTE and balancing class weights to fix this.

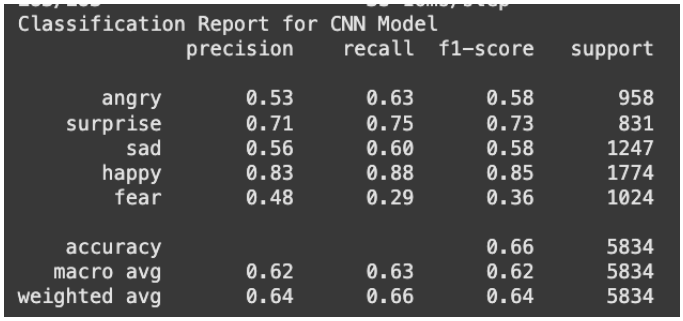
2.1.1 *Emotion Model: FineTuned CNN*

We performed the CNN hyperparameter search using *Hyperband*, and the result is as follows:

Hyperparameter	Value
initial_filters	64
n_blocks	2
filters_block_0	128
dense_units	256
learning_rate	0.0005
tuner/epochs	10
tuner/initial_epoch	3
tuner/bracket	1
tuner/round	1
filters_block_1	128
tuner/trial_id	0000

Table 1: CNN Hyperparameter Search Results

With data argumentation and hyperparameter fine-tuning, CNN Model performance is as follows.



Classification Report for CNN Model

	precision	recall	f1-score	support
angry	0.53	0.63	0.58	958
surprise	0.71	0.75	0.73	831
sad	0.56	0.60	0.58	1247
happy	0.83	0.88	0.85	1774
fear	0.48	0.29	0.36	1024
accuracy			0.66	5834
macro avg	0.62	0.63	0.62	5834
weighted avg	0.64	0.66	0.64	5834

Figure 1: Emotion Model: FineTuned CNN Classification Report

The fine-tuned CNN achieved an overall accuracy of 0.66, with strong performance for *happy* (F1=0.85) and *surprise* (F1=0.73). In contrast, *fear* showed the lowest F1 score (0.29), performing poorly despite the class balancing techniques applied. This might be as a result of the inherent difficulty in distinguishing certain emotions (e.g. *fear* vs. *angry*), which is displayed in the confusion matrix (Figure 2). Not to mention, the limited representation of the lower performing classes in the original dataset likely resulted in less effective learning; synthetic samples generated via SMOTE might not have introduced sufficient diversity for these minority classes.

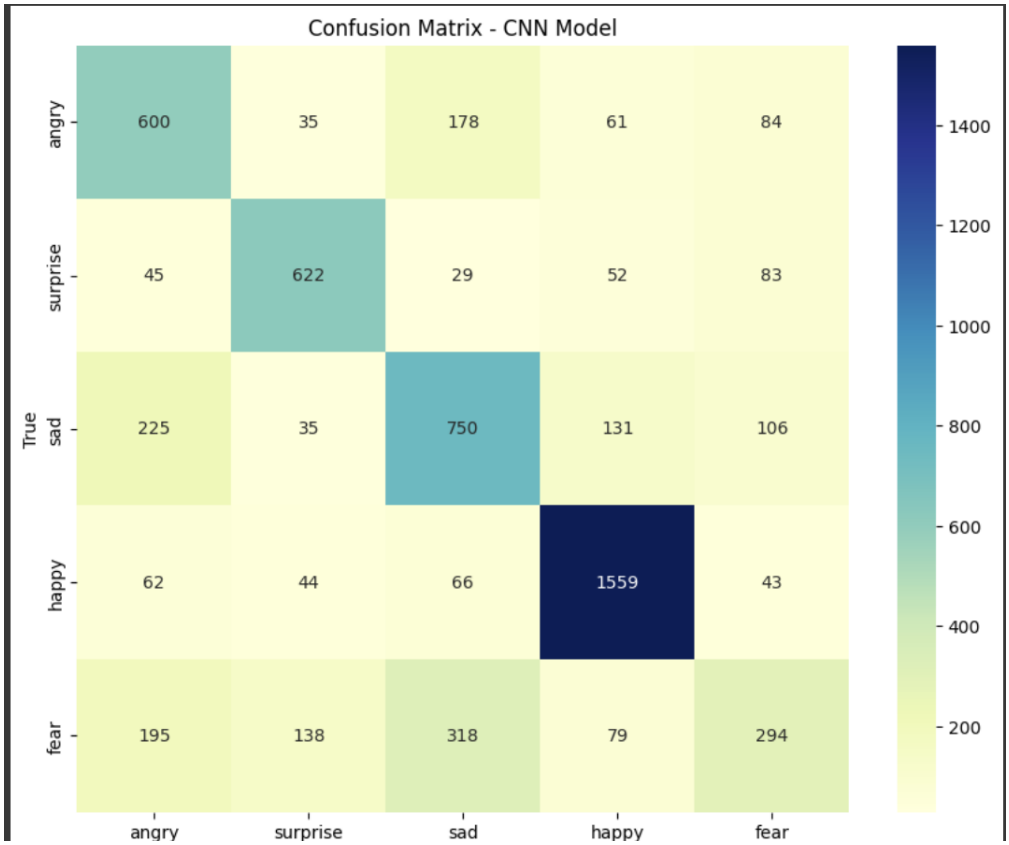


Figure 2: Emotion Model: FineTuned CNN Confusion Matrix

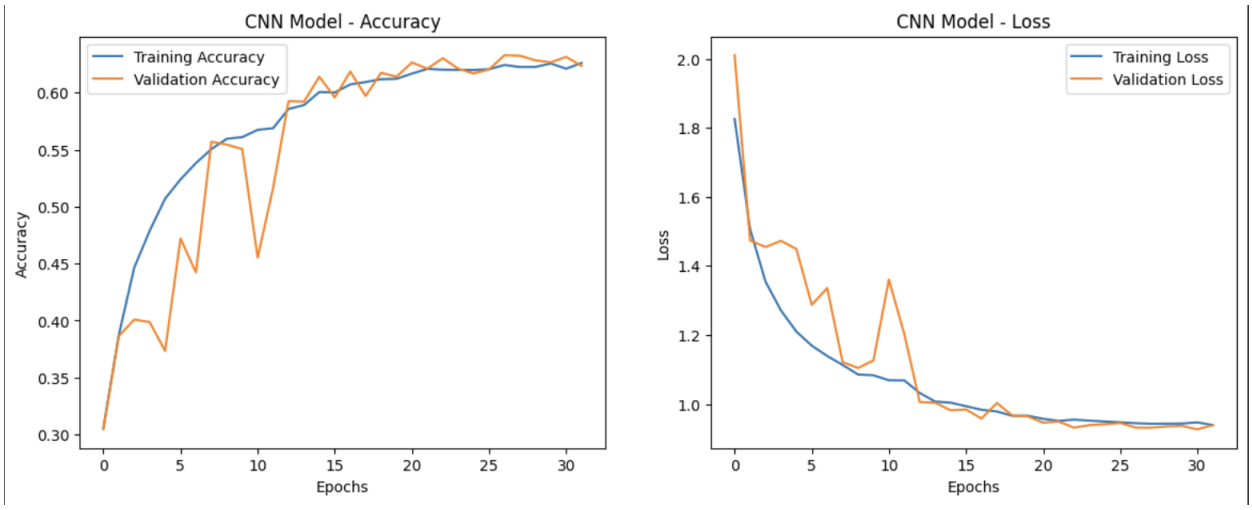


Figure 3: Emotion Model: FineTuned CNN Accuracy and Loss Over Epochs Plots

The training and validation accuracy reflect stable convergence in their stable improvement over 40 epochs. Loss decreased consistently for both datasets, indicating the model's generalization ability.

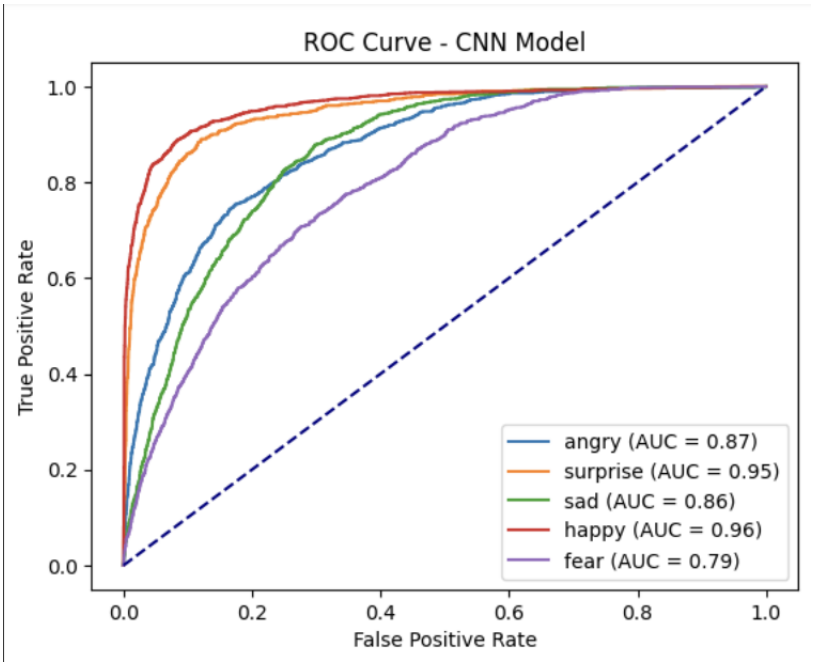


Figure 4: Emotion Model: FineTuned CNN ROC Curve

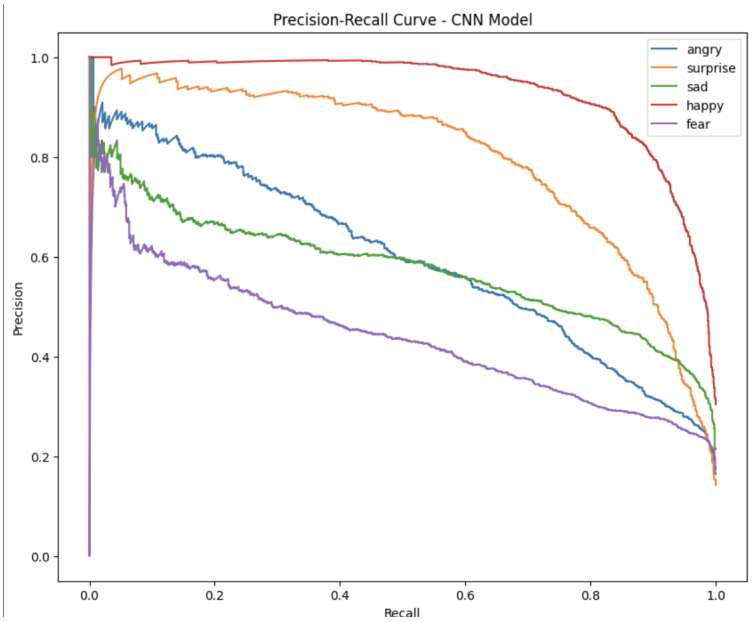


Figure 5: Emotion Model: FineTuned CNN PR Curve

The ROC curve (Figure 4) shows high AUC scores across all emotions, with *happy* (0.96) achieving the highest AUC and *fear* (0.79) the lowest. This reflects a good separation of positive and negative classes. The PR curve (Figure 5) highlights the model's performance, showing strong precision and recall for *happy* and *surprise*. However, *fear* demonstrates lower precision, signaling challenges in maintaining accuracy for positive predictions.

2.1.2 ***Emotion Model: VGG-16***

Classification Report for VGG16 Model

	precision	recall	f1-score	support
angry	0.39	0.27	0.32	958
surprise	0.58	0.59	0.59	831
sad	0.46	0.35	0.39	1247
happy	0.50	0.80	0.61	1774
fear	0.39	0.21	0.27	1024
accuracy			0.48	5834
macro avg	0.46	0.44	0.44	5834
weighted avg	0.47	0.48	0.45	5834

Figure 6: Emotion Model: VGG-16 Classification Report

The VGG-16 model achieved an overall accuracy of 0.48, with strong performance for *happy* (F1=0.61) and poor performance for fear (F1=0.27). This might be explained by the frequent misclassification as seen in the confusion matrix (Figure 2). While *happy* is recognized reasonably well, *sad* and *fear* are both often confused with other emotions like *angry*.

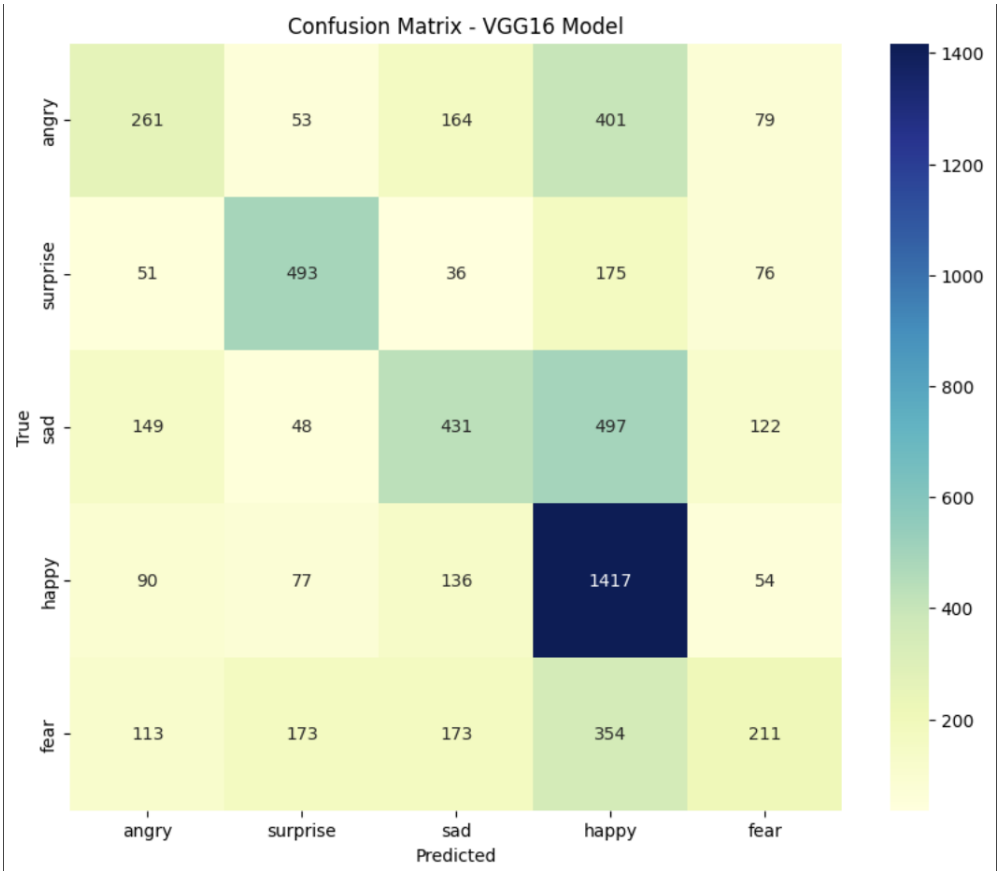


Figure 7: Emotion Model: VGG-16 Confusion Matrix

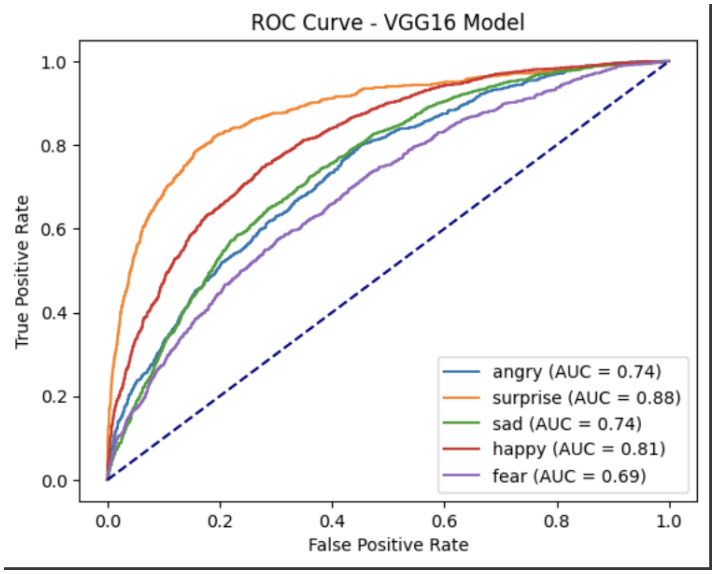


Figure 8: Emotion Model: VGG-16 ROC Curve

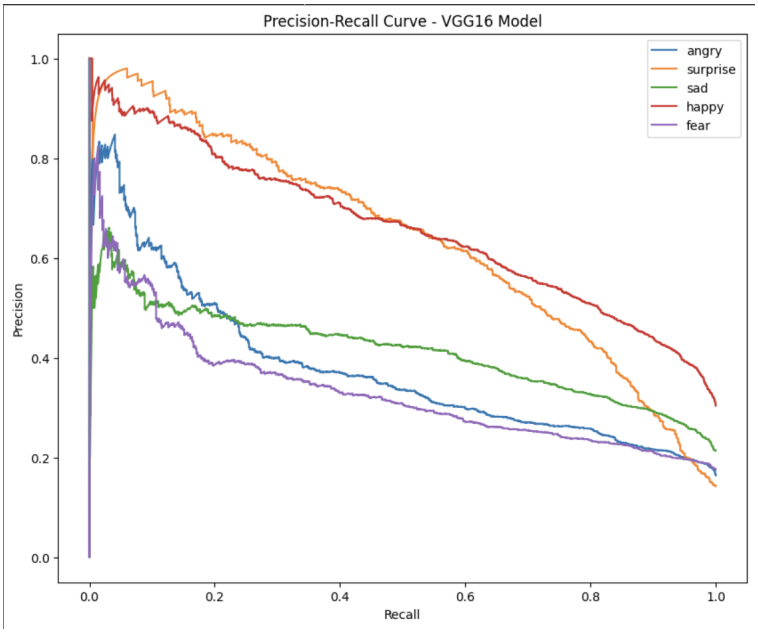


Figure 9: Emotion Model: VGG-16 PR Curve

The PR curve and ROC curve of VGG-16 have similar shape and characteristic compared with that of CNN, but less good performance.

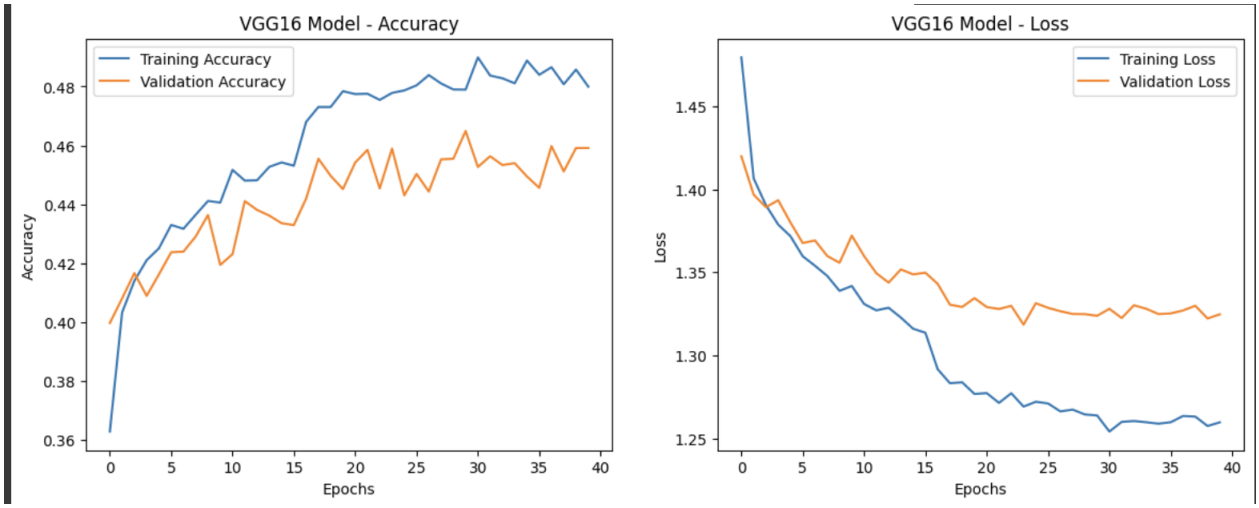


Figure 10: Emotion Model: VGG-16 Accuracy and Loss Over Epochs Plots

Analysis

The CNN model achieved an accuracy of approximately 66%, while the VGG-16 model only reached around 48%. Despite applying class balancing techniques, both models exhibited better recognition performance for major classes like happy compared to minor classes such as fear. This result highlights the persistent challenge of class imbalance, where underrepresented classes are more difficult to learn.

Both models were trained for 40 epochs, the CNN model demonstrated strong generalization ability. Its training and validation accuracy curves align well, showing a steady improvement over epochs.

In contrast, the VGG-16 model encountered a performance plateau for the validation set. A likely reason for this is that only the final layers of the pre-trained VGG-16 model were unfrozen for fine-tuning, which constrained its ability to adapt to the FER dataset's specific characteristics.

2.2 *Music Model*

2.2.1 *Music Model: CNN*

Classification Report for CNN Music Model

	precision	recall	f1-score	support
angry	0.90	0.99	0.94	4331
surprise	0.89	0.99	0.94	4371
sad	0.78	0.78	0.78	4375
happy	0.90	0.66	0.76	4320
fear	0.91	0.96	0.93	4264
accuracy			0.87	21661
macro avg	0.87	0.87	0.87	21661
weighted avg	0.87	0.87	0.87	21661

Figure 11: Music Model: CNN Classification Report

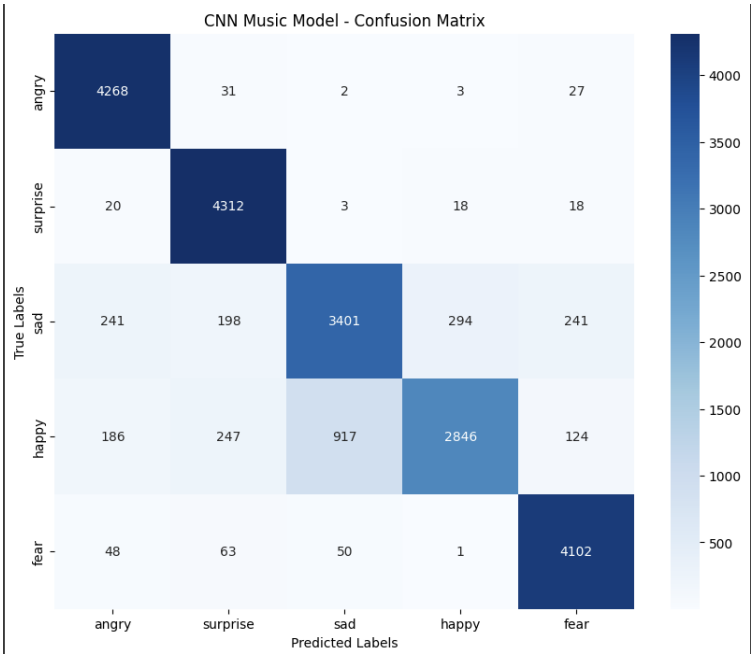
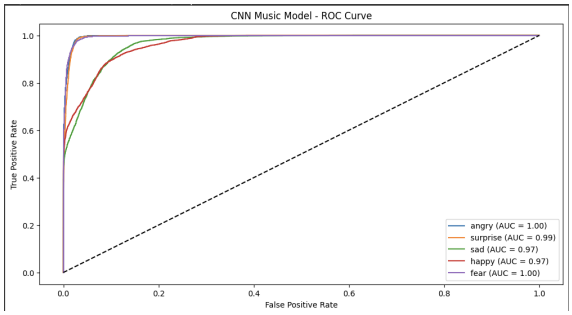
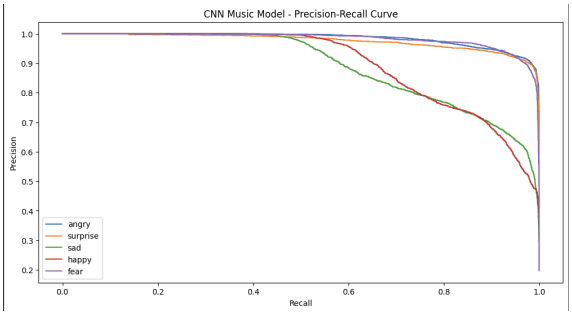


Figure 12: Music Model: CNN Confusion Matrix

This fine-tuned CNN achieved an overall accuracy of 0.87, with strong performance for *angry* (F1=0.94), *fear* (F1=0.93) and *surprise* (F1=0.94), while *happy* and *sad* showed lower F1 scores (F1=0.76, F1=0.79 , respectively). Nonetheless, misclassifications are infrequent as displayed in the confusion matrix (Figure 14).



(a) Music Model: CNN ROC Curve



(b) Music Model: CNN PR Curve

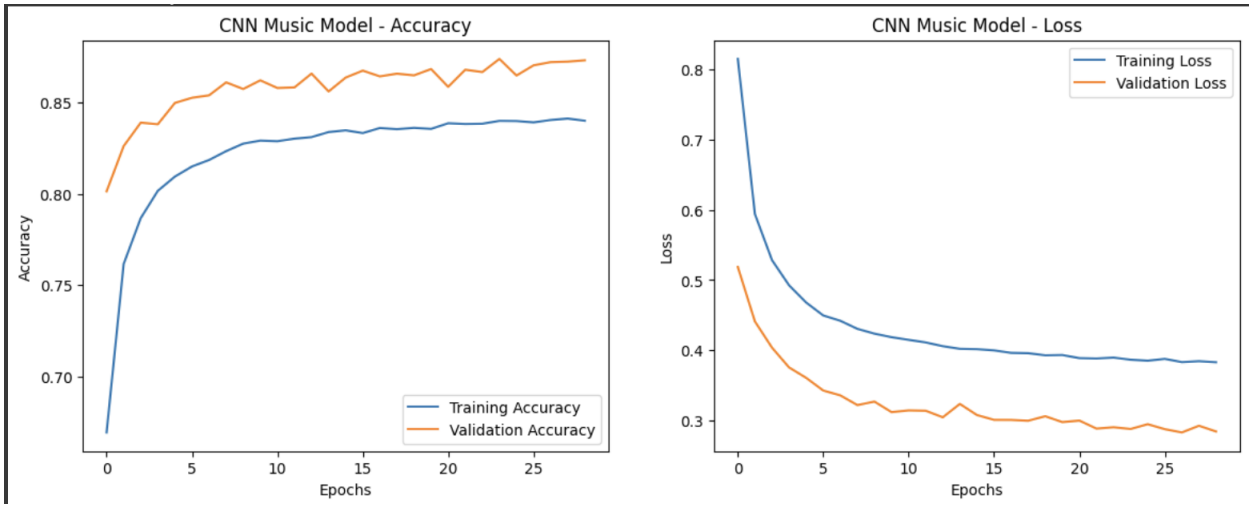


Figure 14: Music Model: CNN Accuracy and Loss Over Epochs Plots

Given 40 epochs, the CNN achieved 87% accuracy, but saw a gap between training and validation accuracy, indicating that the CNN model has a tendency towards overfitting after the initial epochs. The CNN may rely heavily on training patterns rather than generalizing well to unseen data.

2.2.2 *Music Model: RNN with Attention Mechanism*

Classification Report for Attention RNN Model

	precision	recall	f1-score	support
angry	0.95	0.99	0.97	4331
surprise	0.95	0.99	0.97	4371
sad	0.84	0.79	0.81	4375
happy	0.84	0.78	0.81	4320
fear	0.95	0.98	0.97	4264
accuracy			0.91	21661
macro avg	0.90	0.91	0.90	21661
weighted avg	0.90	0.91	0.90	21661

Figure 15: Music Model: RNN with Attention Mechanism Classification Report

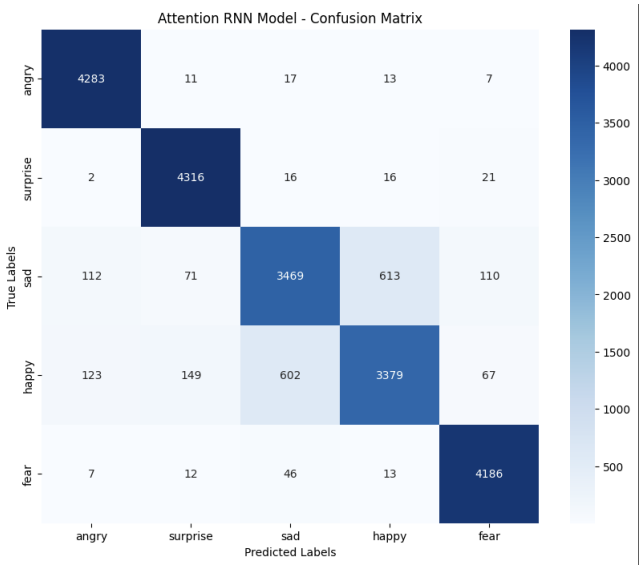


Figure 16: Music Model: RNN with Attention Mechanism Confusion Matrix

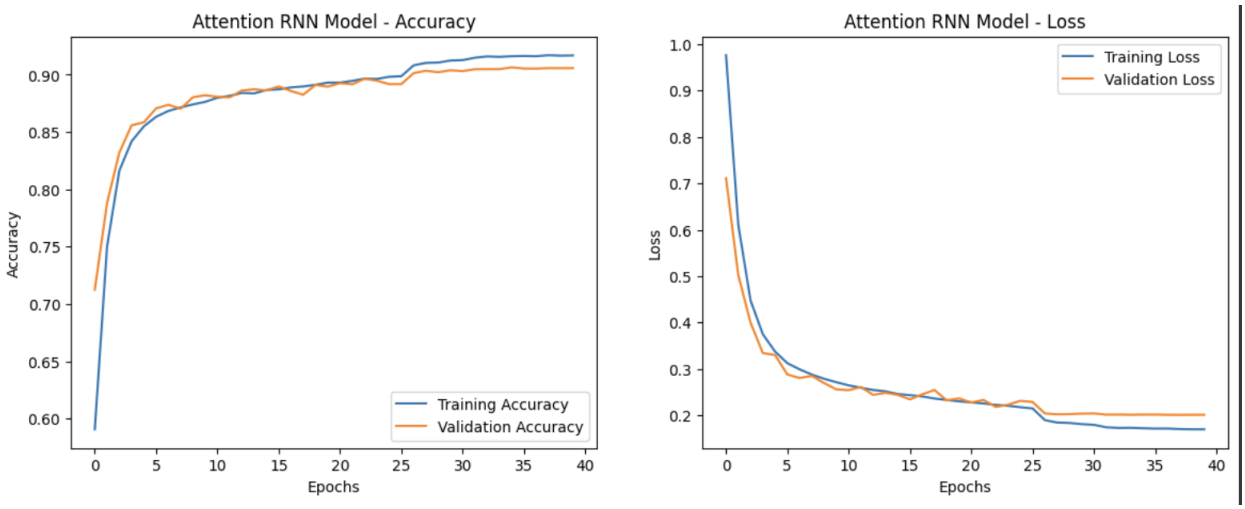
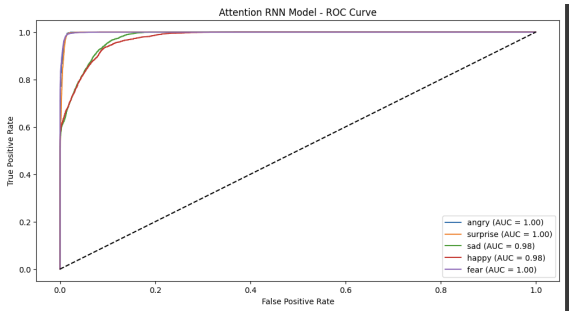
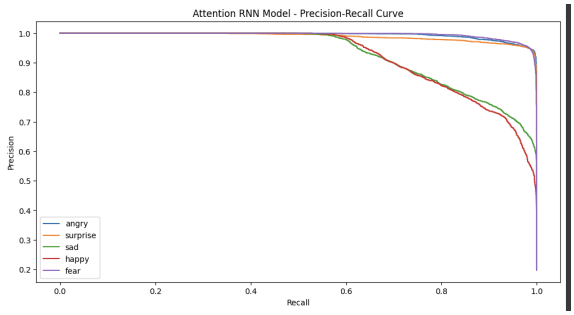


Figure 17: Music Model: RNN with Attention Mechanism Accuracy and Loss Over Epochs Plots



(a) Music Model: RNN with Attention Mechanism ROC Curve



(b) Music Model: RNN with Attention Mechanism PR Curve

The RNN-attention model achieved a validation accuracy of approximately 90% after 40 epochs. The training and validation accuracy closely align, indicating minimal overfitting. The convergence is smooth and stable after epoch 15, suggesting good learning generalization.

Both models show strong performance for all classes but slightly reduced AUC for sad and happy, suggesting challenges in capturing nuanced emotional features. Overall, the Attention RNN Model outperforms the CNN Music Model due to better generalization, stronger handling of class imbalance, and superior precision-recall performance across all emotion classes.

One reason for the strong performance of these models is that the mood labels were designed to be relatively easy to learn, using Music Mood Scoring as the criteria. However, the actual performance may decline when evaluated in real-world user testing, where the complexity of music perception and individual differences come into play.

2.3 *Image Emotion Detection: Using Pascal*

(a) Image Emotion Detection: Using Pascal + CNN

(b) Image Emotion Detection: Using Pascal + VGG

Figure 19: Comparison of Image Emotion Detection Models: CNN vs VGG

- In this image, we show when giving the program an input image, which in this case we used a smiling photo Alexandra took of our previous data science professor and Alexandra's research lab P.I. at the NYU's Fox Lab, and we used the CNN and VGG-16 model to annotate the mention it classified and how confident it is in the emotion prediction.

- The CNN and VGG-16 have high confidence percentages in classifying Professor Pascal Wallisch as happy in this image, and thus the recommender outputted 10 "happy" songs from our music dataset. Not seen in the image but implemented in our code, there is also our ensemble method implemented so it combines the weights of the predictions from both the CNN and VGG-16 model and weighs them accordingly with more of the weight on the CNN predictions as it performs better than VGG-16 model in this case.

- Additionally, we also use the CNN music model and RNN model to give its best prediction on what songs it should be recommended given the image as well and given what the emotion models classify the input image's emotion to be.

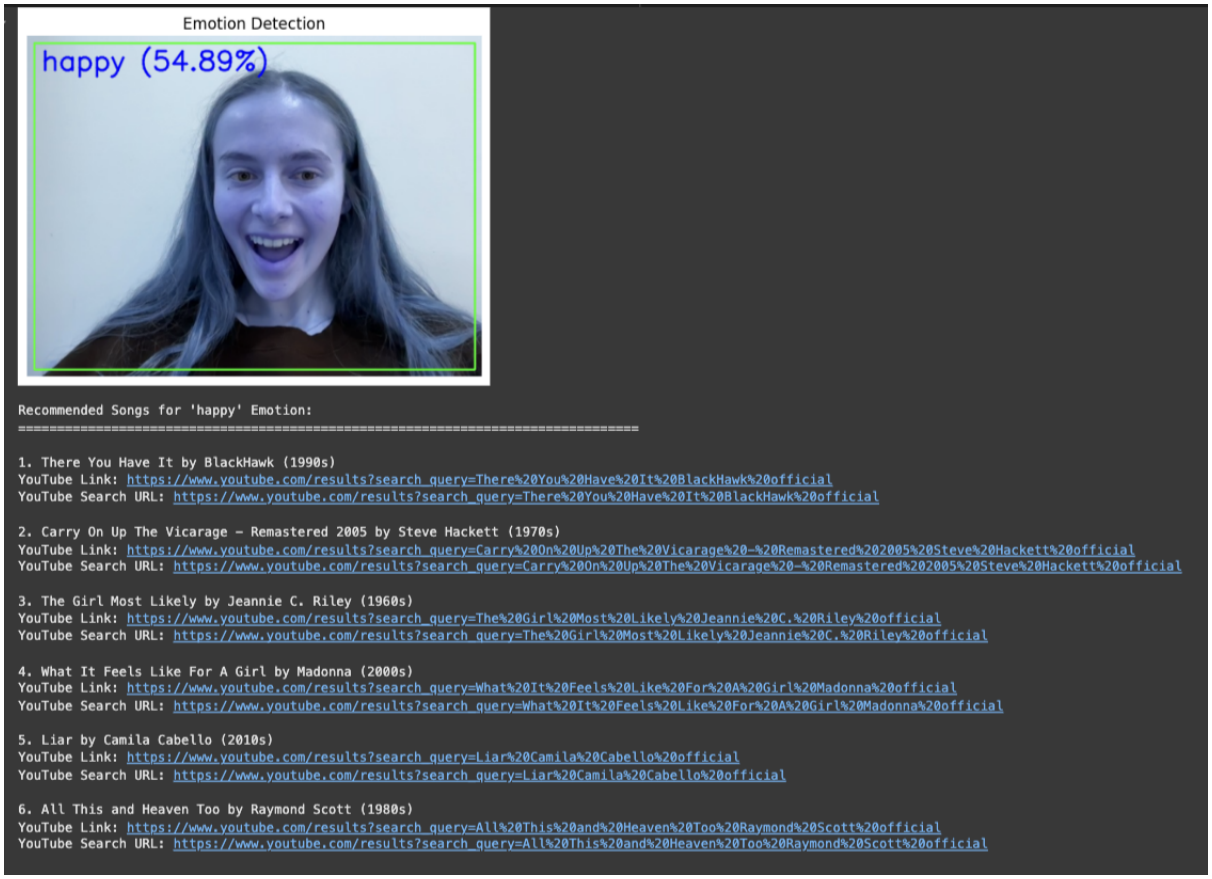


Figure 20: Image Emotion Detection: Using Webcam + YouTube Song Recommendations

- And here is our experimentation work where we used the captured image from the device's webcam that the program runs on, to use the same technique as with the Pascal image emotion detection, but doing it in real-time.
- In the code, we show the emotion predictions of both the VGG-16 model and the fine-tuned CNN emotion model on the webcam-captured image, but we also show in the following code block the ensemble method for those 2 model's predictions on the webcam-captured image, similarly done in the Pascal inputted image.
- It is important to note that there are a few caveats such as the model has issues predicting emotions correctly when other things are in the background, as well as sometimes the ensemble method does more harm than good because as we discussed earlier the VGG-16 model does a very good job predicting happy but some other emotions not as well, whilst the CNN model might do a better job at happy detecting and classification and also possibly slightly better at the other emotions as well than VGG-16.
- Lastly, we were also, interestingly enough, able to integrate the song recommendation with YouTube where the user given they capture their face on the

webcam, will get the YouTube links to songs that the model recommends based on the emotion detected, making it extremely accessible for users to immediately listen to their recommended music.

3. Addtional Exploration

1. We actually trained ResNet model too, with only 15% average accuracy and only learn the feature of happiness. The model was abandoned due to poor performance.

2. Base on the analysis, we find one major limitation hindering the improvement of emotion model accuracy lies in the inherent shortcomings of the FER dataset. The dataset's unbalanced class, low resolution and single-label annotations restrict the capacity of large pre-trained models to fully learn and generalize effectively.
To further investigate this assumption, we utilized the FER+ dataset, which offers cleaned and re-labeled images. Unlike FER, FER+ incorporates multi-label annotations derived through voting, enabling the output to represent a probabilistic distribution across emotions.

Classification Report:

	precision	recall	f1-score	support
anger	0.69	0.85	0.76	331
surprise	0.84	0.78	0.81	446
sadness	0.85	0.74	0.79	450
happiness	0.96	0.85	0.90	929
fear	0.36	0.78	0.49	95
accuracy			0.81	2251
macro avg	0.74	0.80	0.75	2251
weighted avg	0.85	0.81	0.82	2251

Figure 21: CNN on FER+ dataset

VGG16 Classification Report

	precision	recall	f1-score	support
angry	0.42	0.32	0.36	958
surprise	0.57	0.60	0.59	831
sad	0.43	0.42	0.43	1247
happy	0.52	0.75	0.62	1774
fear	0.40	0.19	0.25	1024
accuracy			0.49	5834
macro avg	0.47	0.45	0.45	5834
weighted avg	0.47	0.49	0.47	5834

Figure 22: VGG-16 on FER+ dataset

On FER+, the accuracy of the CNN model raises around 10% with 30 epochs, the VGG-16 model raises around 5% with only 20 epochs. The imbalance of accuracy among classes still exists as we are using the same image set, but the accuracy for all classes has

been improved and the gap between major class and minor class is shortened with the multi-labeled annotation.
Our results demonstrate that, with identical data augmentation, models trained on the FER+ dataset achieve significantly better performance. This finding highlights the potential of our models when applied to more advanced and higher-quality datasets like FER+.

4. Conclusion

- **Strengths**

  1. New way to integrate in-the-moment data, such as selfies to adapt music recommendations dynamically based on current emotional states and environmental contexts.
  2. The project delivers a fully functional pipeline, integrating multiple stages seamlessly: capturing photos, analyzing emotions, recommending songs, and jumping to corresponding music resources. This end-to-end workflow offers a smooth user experience, making it highly practical and usable.
  3. The project extensively leverages machine learning techniques learned in class, such as data augmentation: SMOTE and class balancing for better handling of imbalanced datasets. Additionally, model fine-tuning was performed with hyperparameter optimization on CNN and VGG16, also implement RNN-attention for music-mood feature learning, improving performance and ensuring a robust classification system.
  4. High Accuracy and Reliable Predictions: By combining PR Curves, ROC curves, and confusion matrices, the evaluation framework provides a comprehensive understanding of model performance. The ensemble method, which combines predictions from CNN and VGG16, further enhances overall accuracy, ensuring reliable emotion recognition and song recommendations
  5. The modular notebook design allows for easy customization and scalability, making it possible to expand the system to include additional features such as environmental context or historical user preferences in the future or using more advanced emotion recognition dataset.

- **Limitations**

  Dataset limitations

  (a) Relative to more recent datasets, such as AffectNet or the DEAP dataset, FER might be considered outdated as it lacks richer emotional dimensions such as arousal. This can result in a more extreme classification of emotions. It is important to note, however, that other datasets are less accessible and have significant computational needs as compared to FER, making it the optimal choice for our purposes.

  (b) FER has a low pixel resolution (48x48), which likely causes the loss of subtle facial features, such as details around the mouth and eyes, further constraining the accuracy of emotion recognition.

(c) With respect to our custom music dataset with mood labels, the generation of such labels was derived from several papers: Nuzzolo (2015); Choudhary (2015); Luna (2020) : Given that this indicates that the labeling processing is largely rooted in psychological principles, it may still suffer from mislabeling due to cultural variability and individual differences in music perception, as emotions elicited by music are often subjective and context-dependent.

Model Complexity Tradeoffs

The implementation of ensemble methods, while enhancing prediction accuracy through aggregated voting mechanisms, introduces significant computational costs. This increased processing overhead could potentially hinder real-time mood detection applications, particularly in resource-constrained environments. The additional complexity in deployment and maintenance must be carefully weighed against the marginal improvements in classification performance.

Recommendation System Constraints

The current song recommendation mechanism employs a simplified mood-to-playlist mapping, where each detected mood triggers the random selection of 10 songs from different decades sharing that emotional label. Although this one-to-many approach provides some variety, it still operates within a rigid framework that fails to account for the complexity of human emotional responses to music, which often span multiple mood categories simultaneously. Furthermore, the system's inability to incorporate user listening history and preferences limits its capacity to provide personalized recommendations, potentially reducing its practical utility.

Evaluation Framework Scope

We used a comprehensive suite of traditional machine learning metrics, including Precision-Recall curves, ROC curves, and confusion matrices, all of which provided robust validation of the model's technical performance. However, these metrics are not an ideal assessment of real-world effectiveness. The absence of qualitative measures, particularly those capturing user experience and satisfaction, presents an incomplete picture of the system's practical value.

- **Future Work**

  Our project suggests several promising directions for expanding the system's capabilities and enhancing its practical utility:

  Enhanced Emotion Recognition

  The facial emotion recognition could be improved through multi-dimensional emotion modeling. This might include incorporating arousal or valence indices to enable more nuanced detection or using more advanced datasets that already contain such dimensions. Datasets with higher resolution or pre-trained models finetuned on microexpressions might also be worthy of further consideration.

  Music-Mood Classification

  The music-mood labeling system might be enhanced through the implementation of audio feature extraction via criteria from Librosa, a widely used audio feature

extraction tool that maps audio features to emotional states. This would allow for more precise matching between a song's musical characteristics and its emotional qualities.

Context-Aware Recommendations

The recommendation system could expand its practical implementation by considering environmental context in addition to emotional states through image processing, identifying settings like coffee shops, outdoor spaces, or clubs. This environmental awareness could be mapped to relevant musical attributes–for example, suggesting calmer tracks in a coffee shop. Future consideration might explore the interplay between a user's current emotional state, environment, and user preferences. This is to say that if a user is detected smiling while it's raining, and their history shows they enjoy rain-related music, the system could specifically recommend upbeat songs about embracing the rain. This multi-dimensional approach would analyze environmental factors from images (like lighting and weather conditions) alongside detected emotions and historical preferences to create more nuanced, contextually aware recommendations.

5. Work Flow

-*1. Dataset Preparation and Analysis* (Done by Everyone) Curated a custom `music_moods_dataset` by combining data from multiple decades and classifying moods using audio features from existing music moods dataset.
Analyze the FER dataset and the custom dataset using visualization techniques to explore data distributions and correlations.
-*2. Data Loading and Preprocessing* (Done by Alexandra + Andrea) Load and preprocess data from the FER dataset, custom dataset, and auxiliary datasets. Perform data augmentation and class balancing to improve model performance.
-*3. Model Development* (Done by Andrea + Alexandra) Build a CNN model for facial emotion detection. Build a separate music-mood prediction model to map moods to songs.
-*4. Logic Integration and Prototyping* (Done by Alexandra + Yuhong) Develop the song recommendation logic, combining outputs from the emotion detection model with the music-mood prediction model to recommend relevant songs. Connect the two processes into a unified pipeline for end-to-end functionality.
-*5. Exploration of Alternative Models* (Done by Everyone) Experiment with pre-trained models for emotion detection, such as ResNet and VGG16. (The ResNet was abandoned later because of poor performance) Try alternative architectures for music-mood prediction, such as Attention-RNN or Transformer-based models.
-*6. Model Tuning and Ensemble Methods* (Done by Alexandra + Yuhong) Finetune all models by optimizing hyperparameters and architecture configurations. Implement ensemble methods to combine multiple models, improving accuracy and robustness. Evaluate performance using metrics such as confusion matrices, PR curves, ROC curves, and classification reports.
-*7. Feature Enhancements* (Done by Andrea) Add a web-camera selfie feature for real-time emotion detection. Integrate a YouTube scraping function to dynamically pull up recommended songs.

-*8. Dataset Refinement and Future Direction Exploration* (Done by Yuhong) Switch to the FER+ dataset to validate assumptions about dataset-based accuracy bottlenecks, exploring its enhanced emotional dimensions. Analyze potential improvements and lay the groundwork for future development directions.
-*9. Final Wrap Up*
Milestone 3 was done by Everyone
Slides were done by Everyone
Github was prepared by Yuhong

We all collaboratively worked on idea generation, dataset creation and loading, model building and fine-tuning, and paperwork writing(a.k.a the Milestones, the Slides) through effective communication, thanks to everyone's contribution. Additionally, we stuck to the timeline we had kept in Milestone 2.

References

Krati Choudhary. Music mood classification: Relativity to music therapy. 2015. URL https://kratichoudhary258.medium.com/music-mood-classification-relativity-to-music-therapy-7c44250c45dc.

Rayam Luna. Music mood classification, September 20 2020. URL https://mct-master.github.io/machine-learning/2020/09/20/Music-Mood-Classifier.html#:~:text=%5B%20Features%20%5D-,The%20perception%20of%20emotion%20on%20music%20is%20typically%20related%20to,extracted%20from%20the%20audio%20files. Online article.

Michael Nuzzolo. Music mood classification. *Big Data, Consumer Technologies, Emerging Technologies*, March 25 2015. URL https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/.