

Face-Based Emoji Recommendation Engine

Gabriel MERCIER, Brahim TOUAYOUC

December 14, 2024

Abstract

In the domain of image analysis, we developed a pipeline integrating deep learning models and computer vision techniques to analyze and recognize human faces in order to suggest the most appropriate emoji. We constructed a convolutional neural network (CNN) trained on a large dataset to classify emotions into eight categories. The model achieved an accuracy exceeding that of DeepFace (a facial recognition system developed by researchers at **Facebook AI**), establishing its efficacy. Additionally, we incorporated computer vision algorithms to extract facial landmarks, enabling the synthesis of emotions with user-specific features to recommend one of 15 emojis. This system bridges emotion detection with practical applications, creating an approach to emoji recommendation based on real-time facial analysis.

1 Introduction

The rise of deep learning and computer vision has significantly advanced emotion recognition systems, enabling diverse applications in human-computer interaction, entertainment, and social media. This project explores the integration of emotion detection and personalized emoji recommendations, using advanced convolutional neural networks (CNNs) and computer vision techniques. By analyzing facial expressions, we aim to identify user emotions in real-time and map them to suitable emojis, creating a seamless and interactive experience.

To achieve this goal, we utilized datasets like AffectNet and FER2013, which offer rich diversity and comprehensive annotations, as the foundation for training our emotion detection model and evaluating it. This paper outlines the methodologies employed, including data preprocessing, model architecture, and integration of facial landmarks for feature extraction. The results highlight the potential of combining neural networks and geometric facial analysis to bridge the gap between emotion recognition and practical, user-oriented applications.

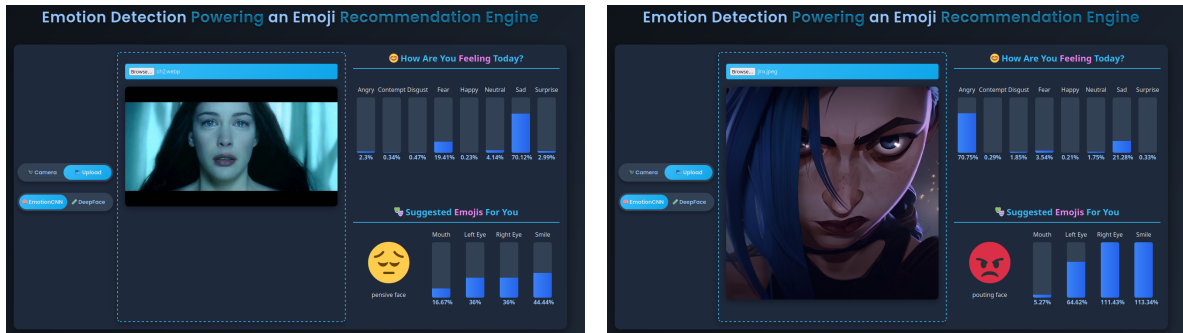


Figure 1: Emotion detection interface powered by DeepFace and EmotionCNN, displaying predicted emotions and emoji suggestions based on input facial images.

2 Training Datasets

For our project, we used two well-known emotion recognition datasets: **AffectNet** [MHM19] and **FER2013** [GEC⁺13]. These datasets were selected for their wide range of facial expressions, diversity in image sources, and comprehensive annotations, which together provide a solid foundation for training and evaluating our emotion detection model.

2.1 AffectNet

AffectNet is a large-scale dataset designed for facial expression recognition, containing over 1 million facial images. These images are labeled across 8 different emotions: *anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise*, *neutral*, and *contempt*. The dataset includes both posed and spontaneous facial expressions, collected from a wide range of subjects with varying demographics, which ensures diversity. This extensive dataset makes it highly suitable for training emotion recognition models in various settings.

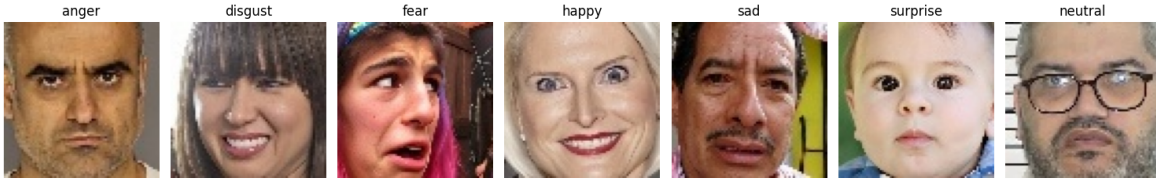


Figure 2: Examples of facial expressions from different emotions (AffectNet dataset)

2.2 FER2013

FER2013 is another widely used dataset in the field of facial emotion recognition, consisting of 35,887 labeled images. The dataset is categorized into 7 emotions: *anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise*, and *neutral*. The images in FER2013 are sourced from the internet and include both posed and spontaneous expressions. Although the dataset is smaller than AffectNet, it is well-balanced and widely used for training and benchmarking emotion detection systems.



Figure 3: Examples of facial expressions from different emotions (FER2013 dataset)

2.3 Data Union

To improve the performance and generalization of our emotion detection model, a random **union** was performed between the AffectNet and FER2013 datasets. This union combined the strengths of both datasets, expanding the overall dataset size and incorporating a greater variety of emotional expressions. The joint dataset thus benefits from the large scale and the real-world diversity provided by both independent datasets. The combined data was preprocessed to standardize image resolutions, perform data augmentation, and ensure consistency in emotion label categories, optimizing it for effective model training.

2.4 Data Preprocessing

Before training, we applied a series of preprocessing steps to optimize the dataset for training. The preprocessing pipeline included three main stages: **face detection and cropping**, **data augmentation**, and **handling class imbalance**.

2.4.1 Face Detection and Cropping

We used the **Haar Cascade** [SK07] face detection algorithm to locate and crop the faces from the images. In cases where no face was detected, the entire image was retained. This ensured that the model focused only on the facial regions, eliminating irrelevant background information.

2.4.2 Data Augmentation and Normalization

We applied the following transformations to enhance the dataset:

- **Grayscale Conversion:** Converted images to grayscale to reduce complexity while retaining facial features.
- **Random Horizontal Flip:** Augmented the data by flipping images horizontally, improving model robustness to face orientation.
- **Random Rotation and Color Jitter:** Introduced minor rotations and variations in brightness, contrast, and saturation to account for real-world variability in lighting conditions.
- **Random Cropping and Resizing:** Ensured that all images were standardized to 48x48 pixels while introducing minor random cropping to enhance robustness to localization noise.
- **Tensor Conversion and Normalization:** Converted images to tensors and normalized them to a mean of 0.5 and a standard deviation of 0.5 to stabilize training.

These preprocessing steps ensured that the model focused on the most relevant features and was robust to variations in input data.

2.4.3 Handling Class Imbalance

To address the issue of imbalance in the emotion dataset, we implemented a **Weighted Random Sampler** to ensure that all emotion classes contributed equally to the training process. This was achieved by:

- Computing the frequency of each emotion class in the training dataset.
- Assigning higher sampling weights to underrepresented classes and lower weights to overrepresented classes.
- Using a `WeightedRandomSampler` in the `DataLoader` to sample images with probabilities inversely proportional to their class frequencies.

This approach prevented the model from being biased toward dominant classes and allowed for better generalization across all emotion categories. By combining these preprocessing steps with balanced sampling, we ensured a fair and effective training process.

Transformation



- Face Detection and Cropping
- Grayscale Conversion
- Random Horizontal Flip
- Random Rotation
- Color Jitter
- Resize and Random Crop
- Normalize



Figure 4: Original image (left) and preprocessed image (right) after applying transformations. The transformations include face detection, grayscale conversion, augmentation steps (flip, rotation, and color jitter), and normalization for training.

3 Model Architecture

The model architecture consists of several layers including convolutional layers, batch normalization, max pooling, dropout, and fully connected layers. 5 shows a detailed description of the architecture:

- **Input:** The input image is a grayscale image with dimensions 48×48 , i.e., $1 \times 48 \times 48$.
- **Output:** The final output layer consists of 8 neurons, corresponding to the 8 emotion classes, with softmax activation for classification.

4 Evaluations

In this section, we present the evaluation of our trained model and compare it to DeepFace, both evaluated on an unseen test dataset of 11,840 images. We report key metrics including accuracy, confusion matrix, and classification report.

4.1 Our Model Results

Test Accuracy: 61.66%

Emotion	Precision	Recall	F1-Score
Anger	0.59	0.53	0.56
Contempt	0.54	0.64	0.59
Disgust	0.52	0.43	0.47
Fear	0.48	0.44	0.46
Happy	0.90	0.78	0.84
Neutral	0.62	0.62	0.62
Sad	0.47	0.57	0.52
Surprise	0.60	0.72	0.65

Table 1: Classification Report (Our Trained Model)

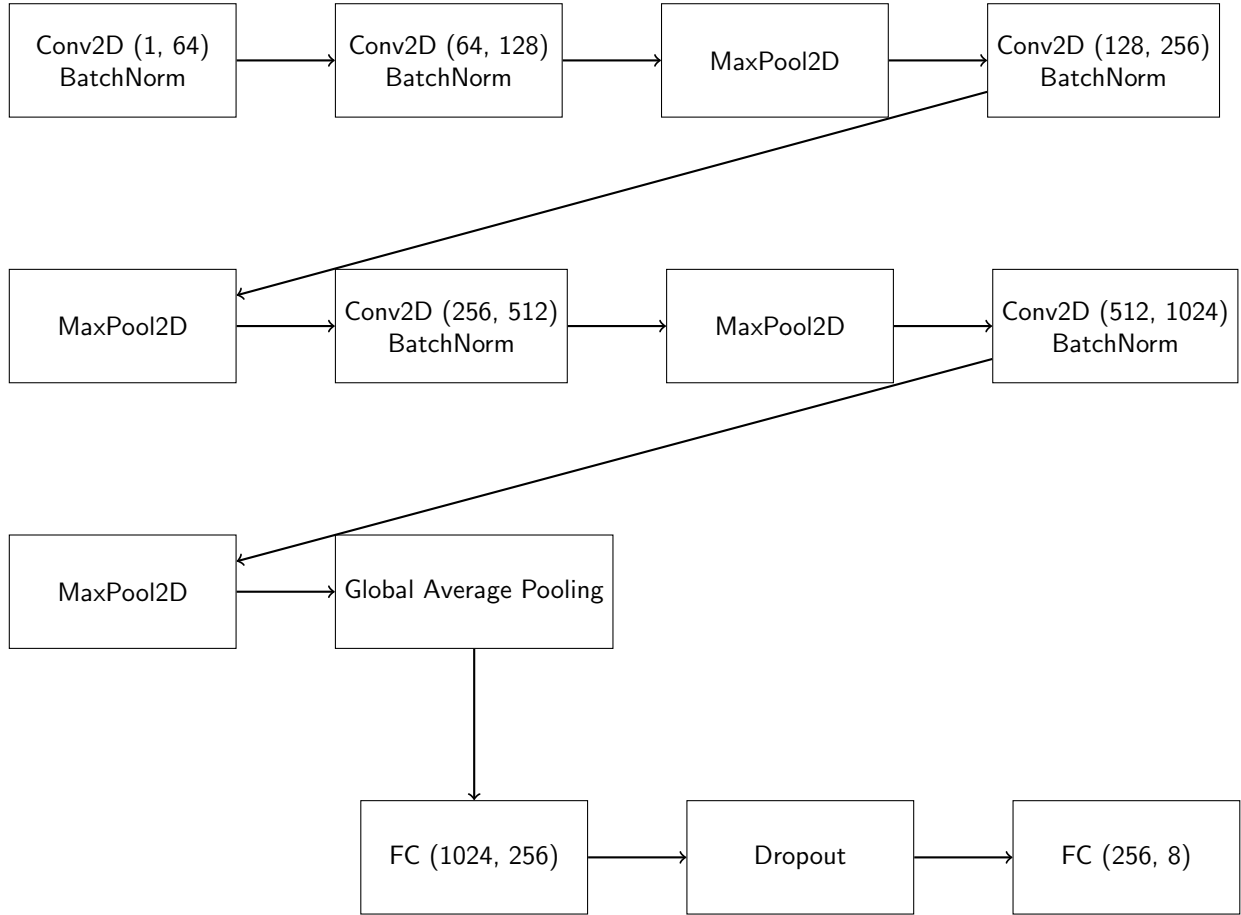


Figure 5: The architecture of the EmotionCNN model, consisting of multiple convolutional layers with batch normalization, max-pooling, a global average pooling layer, and fully connected layers with dropout regularization.

4.2 DeepFace Results

Test Accuracy: 51.00%

Emotion	Precision	Recall	F1-Score
Happy	0.81	0.82	0.82
Sad	0.47	0.56	0.51
Anger	0.00	0.00	0.00
Disgust	0.59	0.12	0.20
Neutral	0.47	0.60	0.52
Surprise	0.73	0.51	0.60
Fear	0.45	0.50	0.47

Table 2: Classification Report (DeepFace)

4.3 Comparison

We summarize the key differences between our model and DeepFace as follows:

- *Test Accuracy:* Our model achieved a significantly higher accuracy of 61.66%, compared to DeepFace’s 51.00%.
- *F1-Score:* Our model consistently outperforms DeepFace in terms of F1-score across multiple emotions, with notably higher values for emotions like "Happy" (0.84 vs 0.82) and "Surprise" (0.65 vs 0.60).
- *Precision and Recall:* Our model demonstrates a more balanced performance with better precision and recall, particularly for "Happy" (0.90 vs 0.81) and "Neutral" (0.62 vs 0.47).

Our model performs consistently better across various emotions, demonstrating stronger overall generalization on the test dataset. However, it is important to note that comparing these models directly is not entirely fair. DeepFace was trained on a dataset with potentially different characteristics from ours. For example, our dataset consists of images that are already cropped and resized to (1, 48, 48), whereas DeepFace was trained on a dataset of images with different preprocessing steps. Specifically, DeepFace was trained on a private dataset, which uses faces of varying sizes and resolutions. Thus, any comparison should take into account these differences in the dataset distributions and preprocessing steps, as it could affect the model performance significantly.[\[TYRW14\]](#)

4.4 Training and Validation Metrics

To evaluate the performance of our model during training, we monitored two key metrics: loss and accuracy, for both the training and validation datasets. These metrics provide insights into how well the model is learning and generalizing to unseen data. Below, we present the visualizations of training and validation loss, as well as training and validation accuracy, across all epochs.

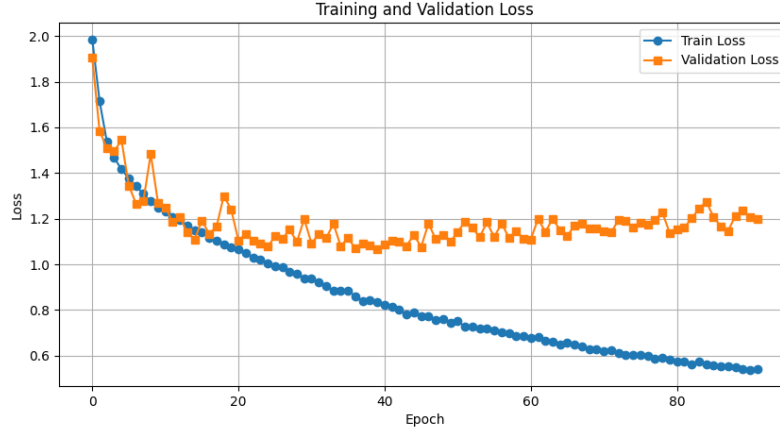


Figure 6: Training and Validation Loss over epochs.

The loss plot (Figure 6) shows a significant reduction in training loss, which is expected as the model adjusts its parameters to better predict the training dataset. The validation loss stabilizes, which reflects the model’s generalization to unseen data, although occasional fluctuations indicate potential overfitting.

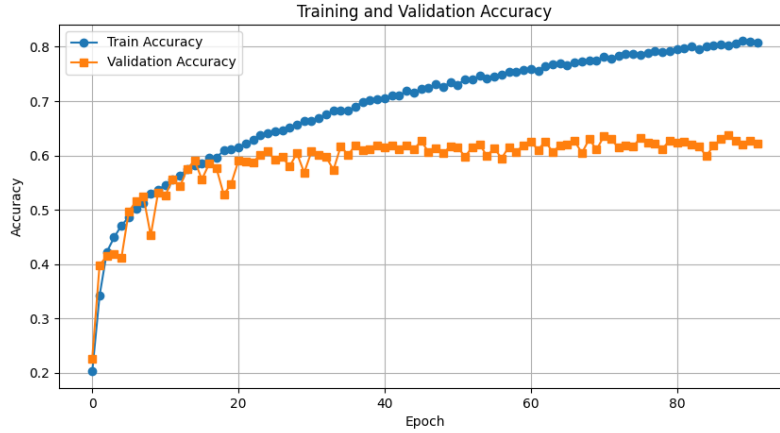


Figure 7: Training and Validation Accuracy over epochs.

The accuracy plot (Figure 7) reveals a steady improvement in training accuracy, which aligns with the decreasing training loss. The validation accuracy, however, stabilizes after an initial rise, demonstrating the model’s ability to generalize to the validation dataset. This stabilization suggests that further improvements in training may not necessarily lead to better validation performance.

5 Introduction of the complementary method

The objective of this part is to leverage the previous trained emotion recognition model to recommend an emoji, selected from a predefined list of 15 emojis, based on the real-time facial expression of a person. We analyze key facial landmarks to extract meaningful features and match them to the emoji that most closely aligns with the current facial configuration. These meaningful features are :

- **Mouth openness** (d_{mouth}): Vertical distance between the upper and lower lips.
- **Smile width** (d_{smile}): Horizontal distance between the corners of the mouth.

- **Left and Right Eye openness** (d_{eye}): The distance for each eye is calculated as the vertical distance between the upper and lower eyelids.

5.1 Dataset

To achieve our goal, we manually created a dataset that associates specific values for the features with each of the 15 emojis. This dataset was **constructed through an empirical process** where we determined the minimum and maximum values for each feature (*mouth openness*, *smile width*, and *eye openness*) by analyzing various facial expressions. Using these empirical boundaries, we assigned feature values to each emoji based on our judgment, ensuring that the chosen values accurately reflect the corresponding facial expression for each emoji. This handcrafted approach allowed us to design a dataset tailored to our specific application, despite its inherent subjectivity.

5.2 Distances Extraction

To achieve this, the project utilizes the **Mediapipe** library [LTM⁺19] for real-time facial landmark detection. **Mediapipe** identifies 478 landmarks on the face, which are used to compute the 4 key features. Through the documentation, we managed to find the landmarks associated with our points of interest, enabling us to compute these features effectively.

All these distances were calculated using the L2 distance, for example for the mouth openness :

$$d_{\text{mouth}} = \sqrt{(x_{\text{upper_lips}} - x_{\text{lower_lips}})^2 + (y_{\text{upper_lips}} - y_{\text{lower_lips}})^2}$$

5.3 Feature Matching and Emoji Classification

As mention earlier, each emoji in the dataset is characterized by predefined values for the three extracted features. To determine the best-matching emoji for a given face, the following steps are performed:

1. Calculate the L2 distance L_2 for each emoji:

$$L_2^j = \sqrt{\sum_{i=1}^4 (f_i^{\text{real}} - f_i^j)^2}, \quad \text{for all } j \in [0, 15] \quad (1)$$

where f_i^{real} represents the feature extracted from the real-time image (e.g., d_{mouth}), and f_i^j is the corresponding reference value for the emoji. Here, 4 represents the number of characteristics, and 15 corresponds to the total number of emojis.

2. Pass the computed distances through a **normalization function** to obtain the **opposite probability vector** for each emoji:

$$P(\text{emoji}_j) = \frac{L_2^j}{\sum_{k=1}^{15} L_2^k} \quad (2)$$

where $P(\text{emoji}_j)$ is the probability of emoji j , and L_2^j is the distance for emoji j .

6 Combinaison between the two methods

6.1 Dataset

Similar to the approach used in the complementary method, we constructed a dataset associating each emoji with a probability distribution over the 8 emotions detected by our CNN. To achieve this, we utilized the dataset presented in the work by Shoeb and de Melo [SdM20], which explores the relationship between emojis and emotions based on an analysis of tweets. Building on this foundation, we empirically refined the dataset to align with the 8 emotions recognized by our CNN model (*anger*, *disgust*, *fear*, *happiness*, *neutral*, *sadness*, *contempt*, and *surprise*). Notably, the original dataset did not include explicit mappings for *contempt* and *surprise*, so we added these emotions by analyzing similar emotional contexts and assigning probability distributions based on our judgment.

6.2 Methodology

Similar to the complementary method, we first use the refined dataset to compute a 15-dimensional vector representing the distance between the live image and each emoji. For each emoji, the distance is calculated using its probability distribution from the dataset and the probability distribution from our neural network. These distances are then transformed into probabilities using a Softmax function.

In summary, we now have two probability distributions for the 15 emojis, derived from the two methods presented: one from the CNN-based emotion detection and the other from the complementary feature-based approach.

These **distributions are merged using a weighted average**, where the weight is determined by a floating-point value $\alpha \in [0, 1]$. The final probability distribution for each emoji is computed as:

$$P_{\text{final}}(\text{emoji}_j) = \alpha \cdot P_{\text{CNN}}(\text{emoji}_j) + (1 - \alpha) \cdot P_{\text{measures}}(\text{emoji}_j)$$

Here, $P_{\text{CNN}}(\text{emoji}_j)$ and $P_{\text{measures}}(\text{emoji}_j)$ represent the probabilities for emoji j obtained from the CNN and complementary methods, respectively. The parameter α allows us to balance the contributions of the two methods, enabling fine-tuning based on performance.

7 Quantitative Results

7.1 Dataset

To quantitatively evaluate the effectiveness of our emoji recommendation system, we constructed a **handcrafted dataset** that maps facial images to our selected 15 emojis. These emojis were chosen to represent the emotional spectrum identified in the **FER2013** dataset, with two emojis selected for each emotion (except for *contempt*, for which only one emoji was available due to its distinctiveness and rarity). The dataset contains 10 images per emotion for a total of 150 images. This dataset served as a benchmark to provide a concrete, measurable basis for comparing different methods and configurations.







Emotion	Facial Image	Emoji
Happiness		
Sadness		
Fear		

Table 3: Examples of facial images and their corresponding emojis in the dataset.

7.2 Fine Tuning

To identify the optimal balance between the CNN-based and complementary feature-based methods, we applied our pipeline with **different values of α** (the weighting factor that determines the contribution of the emotion). For each $\alpha \in [0, 1]$ (with a step of 0.05), we calculated the accuracy of the combined approach by comparing the predicted emojis to the ground truth.

The figure below shows the accuracy of the system across different values of α . The results indicate that the best performance is achieved **with $\alpha=0.55$ with an accuracy of 52.6%**, suggesting that

slightly more weight should be given to the CNN-based method while still incorporating the complementary features for optimal results.

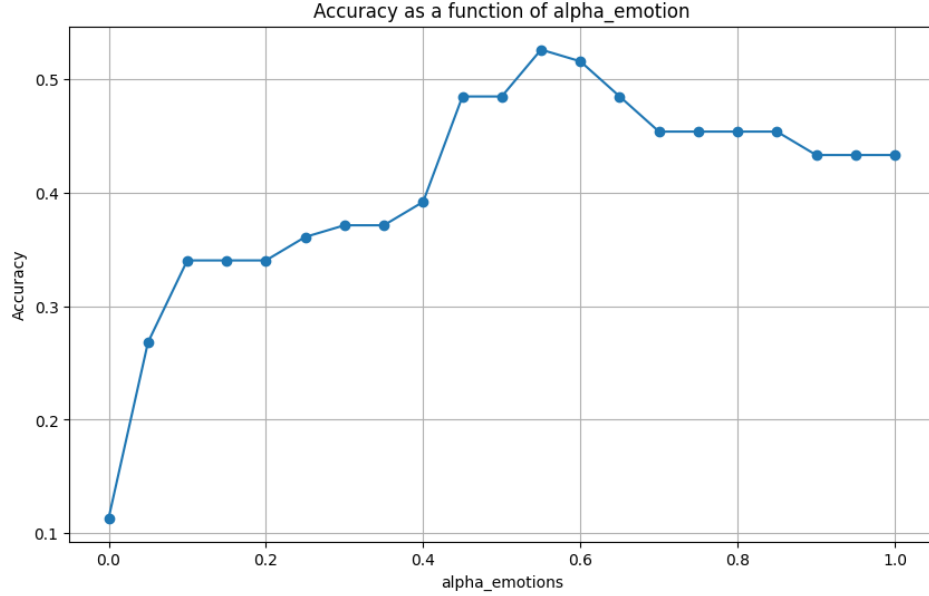


Figure 8: Accuracy of the system across different values of α .

As α approaches 1 (i.e., relying entirely on the CNN-based method), the performance declines slightly, suggesting that the geometric features captured by the complementary approach add valuable context. Conversely, when α approaches 0, the complementary method alone lacks the depth needed for robust emoji recommendations.

These results demonstrate the **efficacy of our combined approach**.

8 Failed Attempt

In this approach, the goal was to utilize the facial landmarks detected by MediaPipe to generate a simplified representation of the face. This involved creating a binary image with white contours on a black background to emphasize key facial features: the eyes, the mouth, and the overall shape of the face. The face was then cropped into a standardized frame. A similar preprocessing step was applied to emojis, reducing them to white contours on a black background, capturing their defining features.

To **measure the similarity between the preprocessed face image and each emoji**, several distance metrics were tested, including the L1 norm, the L2 norm, Hu moments, and a shape similarity function.

Despite these efforts, the approach was **unsuccessful** probably due to significant discrepancies in the shapes and proportions of the faces. Emojis typically feature exaggerated and rounded facial structures, with greater amplitude in eye and mouth movements compared to real-world faces. These differences resulted in poor similarity scores across all tested metrics, making the method ineffective and unstable for matching real facial features to emojis.

9 Conclusion

In this project, we successfully designed and implemented a robust pipeline for emotion-based emoji recommendation, bridging deep learning with geometric facial feature analysis. Our system combines

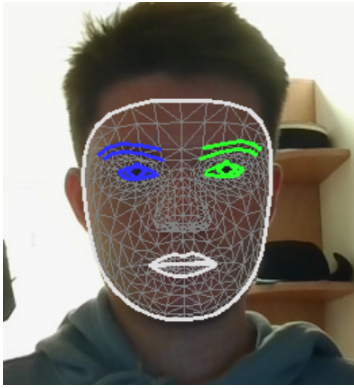


Figure 9: Face with MediaPipe Landmarks



Figure 10: Face after the described processing

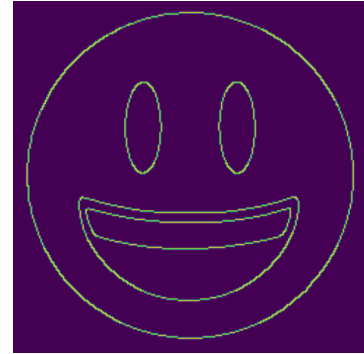


Figure 11: Same processing for the emoji

the strengths of convolutional neural networks (CNNs) for emotion recognition and complementary methods for facial feature extraction, enabling it to recommend emojis based on real-time facial analysis. By evaluating and fine-tuning the balance between these approaches, we demonstrated the importance of integration and adaptation in creating effective, user-oriented applications.

One of the most rewarding aspects of this endeavor was the hands-on experience of working collaboratively on a challenging, interdisciplinary project. From preprocessing large datasets to refining model architectures and validating results, every stage required effective communication and problem-solving. This project not only strengthened our technical skills in computer vision and deep learning but also taught us lessons in teamwork and time management.

Future work could involve expanding the dataset to include more emojis, incorporating temporal consistency checks for smoother predictions, and optimizing the feature extraction pipeline for improved real-time performance.

References

- [GEC⁺13] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests, 2013.
- [LTM⁺19] Camillo Lugaresi, Fan Tang, Qian Matteti, Jax Hernandez, Michael Kim, and et al. Mediapipe: A framework for building perception pipelines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 123–132. IEEE, 2019.
- [MHM19] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, January 2019.
- [SdM20] Abu Awal Md Shoeb and Gerard de Melo. Emotag1200 : Understanding the association between emojis and emotions . In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4635–4645, Online, November 2020. Association for Computational Linguistics.
- [SK07] Adam Schmidt and Andrzej Kasiński. *The Performance of the Haar Cascade Classifiers Applied to the Face and Eyes Detection*, volume 45, pages 816–823. 10 2007.

- [TYRW14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.