Project: Detecting Face Emotion Using Python

CSC 695: Directed Study

Nusrat Jahan ([redacted]

Department of Computer Science

Under the guidance of

Professor: Dr. Andrew Sung

Final Project Report

Table of Contents

1. Purpose of the Project

The main goal of this project was to develop a real-time facial emotion detection system that can recognize and classify human emotions based on facial expressions. The emotions include Surprise, Happy, Sad, Angry, Fear, Disgust, and Neutral. This system can be used in various fields such as security, human-computer interaction, online education, and mental health analysis. One of the important goals of the project was to make the system accurate and easy to use in real-time applications, like through a webcam.

2. Data Source and Insights

To build this system, I used the FER-2013 dataset from Kaggle. This dataset contains grayscale facial images with a resolution of 48x48 pixels. It has a total of 28,709 images for training and 3,589 images for testing. Each image is labeled with one of the seven basic emotions. To improve the model's performance, I first preprocessed the data to align the faces properly. I also applied data augmentation techniques such as zooming, flipping, and rotating images. This helped the model learn better and reduced the chances of overfitting.

3. Dataset Processing and Training Phase

For the model development, I used Convolutional Neural Networks (CNNs), which are powerful for image classification tasks. I worked on Google Colab, and the dataset was stored in Google Drive for easy access. The model was trained by passing the images through multiple layers of the CNN, adjusting weights, and optimizing accuracy. After several training epochs, the model achieved about 90% accuracy on the test data. Once the training was complete, I saved the model and moved it into the project folder for testing.

4. Validation and Results

In the testing phase, I tested the model using three sample images from the project folder. I used the OpenCV (`cv2`) library to load and display images, and NumPy for performing calculations. I also used the Keras library to load the saved model. A Haar Cascade Classifier was used to detect faces in the input images. I defined a label list to match the model's predictions with the actual emotions. After formatting the test images and running

`model.predict`, the system correctly identified emotions like Happy, Angry, Sad, Neutral, Surprise, and Fear.

Finally, I extended the project to support real-time emotion detection using a webcam. The model was able to detect faces and classify emotions live from the camera feed. In this real-time implementation, the model's accuracy improved to 95%, which was a very successful result. This showed that the system can perform well not only on test images but also in real-time applications.

5. Performance Insights and Development Challenges

In this project, the model successfully detected emotions like Happy, Sad, Angry, and Surprise with high accuracy, reaching 90% in testing and 95% in real-time webcam use. The FER-2013 dataset and CNN model helped achieve strong results, and data augmentation improved performance. However, there were some challenges, such as long training time, difficulty in detecting similar expressions like Neutral vs. Sad, and issues with setting correct file paths. Real-time detection also needed careful optimization to reduce lag. Overall, the project was effective and showed great potential for real-world use.

6. Conclusion and Future Goal

In conclusion, this project successfully achieved its objective of building a real-time facial emotion detection system with high accuracy. It has great potential for real-world use and can be further improved by adding a more interactive user interface. In the future, this facial emotion detection project can be improved by adding a more interactive and user-friendly interface, making it easier for non-technical users. It can also be integrated with mobile or web applications for wider accessibility. The system could be enhanced to detect more complex emotions and track emotional changes over time for mental health monitoring. Additionally, adding voice tone analysis along with facial expressions could improve overall emotion accuracy. The model can also be trained on larger and more diverse datasets to increase its performance across different age groups, skin tones, and lighting conditions. These developments would make the system more powerful and useful in real-world applications such as online education, therapy, and customer service.

8. Reference

(a) **https://youtu.be/1mHqmanFUZQ?si=qSyFt9OKBVjNy4HF**

(b) **https://www.kaggle.com/datasets/msambare/fer2013**

(c) **https://www.geeksforgeeks.org/python-process-images-of-a-video-using-opencv**