Final Report

Yue Teng

Problem Statement

The current technology needs to provide a more efficient way to detect and classify facial expressions, such as neutral, happy, sad, and angry. It is necessary for various applications such as security, healthcare, and human-computer interaction. Thus, as a customer, I want a reliable and effective facial recognition system that meets our needs and improves our user experience.

Solution

The facial expression recognition system uses Keras, a deep learning library in Python. A convolutional neural network (CNN) model is utilized and trained on 600 facial images from 20 different people to analyze facial features. Upon initial consideration of the problem, the use of backward propagation appeared to be a viable approach, until I found Keras. According to the Internet, Keras automatically handles the backpropagation process, as long as we train with the fit method. Thus, I believe that this is a reliable and efficient solution for facial expression recognition, which helps to improve the overall user experience and customer satisfaction.

Demo

Due to the long training time, it was difficult for me to provide a recorded demo. However, I would love to show you how my code works with user input images.

The image above is an AI-generated picture portraying a smiling person. Following a wait period, the program has generated its prediction, resulting in the label '2.' The label system uses integers, where '1' represents a neutral expression, '2' stands for a happy expression, '3' for a sad expression, '4' for an angry expression, and '5' for other expressions. The program correctly identified the person in the image as displaying a happy expression.

```
38/38 [==============================] - 16s 422ms/step - loss: 0.2915 - accurac
y: 0.8955 - val_loss: 2.1371 - val_accuracy: 0.5400
Test loss: 1.7494070529937744
Test accuracy: 0.5706666707992554
Predicted label: 2
```

Assumptions, Constraints & Implications

Due to the time, I wasn't able to complete the training excluding the images with sunglasses. This clearly affects the accuracy. Fortunately, in the demo, the program still predicted the correct result. Additionally, the performance of the model may be affected by variations in lighting, angle, or other environmental factors that could affect facial recognition.

Also I am a little concerned about the model, it seems that the model assumes that the expressions are mutually exclusive and only one label can be assigned to a given face. In reality, this might not be true.

How your solution was built

The dataset to be used for this project is the one being provided, which comprises 600 images of faces from 20 individuals captured in various poses and exhibiting a range of emotions. These will allow us to train our facial recognition algorithm to detect faces properly under a variety of conditions.

I spent approximately two days enhancing the accuracy of my model, such as adjusting parameters such as the number of epochs or the addition of new layers. Unfortunately, the outcomes were not as promising as I had hoped. It appears that developing a better model in real-world scenarios is more difficult than what I imagined.

Summary

In summary, the project aims to develop a reliable facial expression recognition system that classifies neutral, happy, sad, and angry faces using a deep learning library in Python. The solution utilizes a convolutional neural network (CNN) model trained on 600 facial images from 20 individuals. Although the accuracy of the model is affected by environmental factors and incomplete training data, the program still predicts correct results in the demo.