EECS442 Group 23 Final Project Report
Group member: Haowen Shi, Yijun Yan, Haojie Ye, Jiahao Zhang
Project Name: Emoji Mask
Project GitHub link: https://github.com/yanyiju/EECS442_project

Executive Overview

This project aims to cover emoji (or any specific mask) to people's faces in a set of photo albums. Emoji Mask will identify people's emotions in a photo and cover their faces by matching emojis with proper sizes and orientations. This project will not only serve as a fun method of photo processing but also provide a potential use in protecting personal privacy in social media (mask/unmask target's face by learning the same person's faces implemented).

Background and Impact

Emoji Mask is useful in classifying photos according to their emotions and add emoji masks (or any other specified masks) over it. It has the potential for privacy protection. Consider replacing emojis with Mosaics, when any photos or videos with unauthorized faces are posted on the Internet, Emoji Mask can detect the faces and add Mosaics to it, being robust in face size, orientation and face alignment. For recreational usage, it is an efficient way of counting faces, classifying emotion and determining the event background of the photo.

Method

Face Detection using Haar Cascades[1]
Eigen-face classification of emotion[2]
More layers on the eigenface classification
Image Transformation on emojis attached to original images
LSQ method to calculate the similarity between the test face image and sample face[3]

Prototype

There are four major tasks in this project.

(1): Locate all the faces in given images of different sizes

(2): Classify each face into seven emotions: happy, surprised, neutral, sad, etc.

(3): Attach faces with proper sizes, face center, emotion label, and orientations

(4): Extract the features of the sample faces and add emojis on faces of the same person.

We used four prototypes with nuances to tackle these tasks:

Face Detection and Face Landmark Extraction

The face detection is the basis of this project, and it needs high accuracy and efficiency. Since the related knowledge is not fully covered in classes, this part is finally realized through package dlib based on the referenced cv2 face training model of haarcascade_frontalface.xml.

To extract the major facial landmarks, we also referenced cv2 and dlib to extract the location of face locations of left eyes, right eyes, left eyebrows, right eyebrows, nose, mouth, and jaw[4]. For each face box in the image, these locations form a list, and it is saved for the usage of the model of emotion classification and emoji grafting.

Emotion classification

We implemented and improved the traditional eigenface algorithm discussed in the class as the base model[2]. We used the Extended Cohn-Kanade Dataset[5][6] as the base data to generate the emotion classification eigenfaces. Then we considered the improvement on the potential downside of eigenface, which is the ignorance of background noise (as the street number recognition in homework 3 has indicated). Our solution is to first add the weight of the location of extracted facial landmark features. Then add a pooling layer to filter out background images. A normalization layer is then added to the layer to make the training robust to different light conditions.

Emoji Grafting

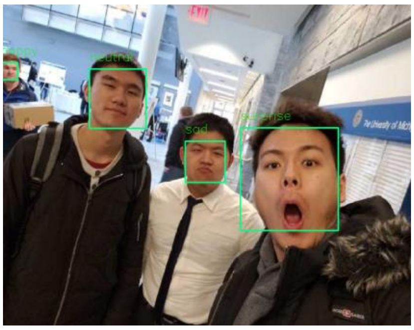
During Face Detection, we can get the key points' information of landmarks for each face. Based on the left eye and right eye information, we can roughly get the center of a person's face at the middle point between left and right eye and the angle of inclination of the face from the slope of the line from left eye to the right one.

Recognize one specific subject

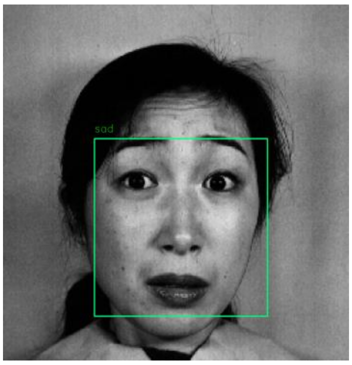
This part is aimed to recognize one same person's faces across all given pictures given in the .\album. Assuming that each face is a 2D plane and has a relatively small angle with the camera projective plane (or the face is very possibly nor detected), which means we can ignore the projective issues, we respectively calculate the distances between nose (face center) and mouth/right_eyebrow/left_eyebrow/right_eye/left_eye/jaw[4] and establish a normalized matrix for each face. The classifying based on the distances between any two faces is an LSQ problem[3]. If the distance between the sample face and the unknown face is lower than a threshold, then these two faces can be recognized as one person.

Results

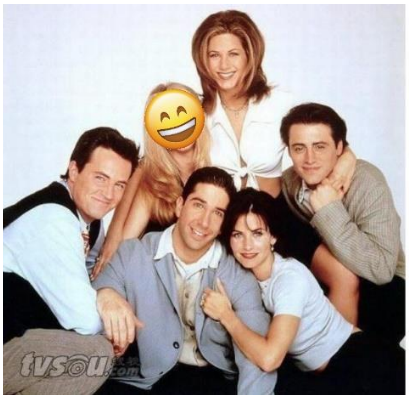
The classification of human emotion turns out to have a relatively median accuracy of 25.1% (Baseline 14.3%). We used a different dataset with ground truth: *The Japanese Female Facial Expression (JAFFE) Database* [7] to verify the overall accuracy. We also used daily photos for verification.
A success classification of emotion for all people appearing in the image is shown in the following:



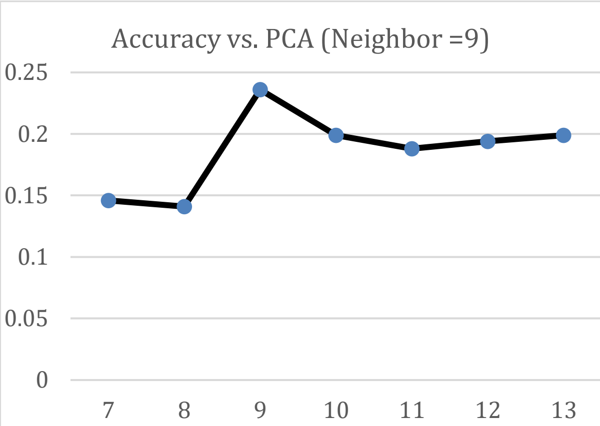

A failed classification of emotion for all people appearing in the image is shown in the following (Output label: Sad. Ground truth label: Fear) [7]:

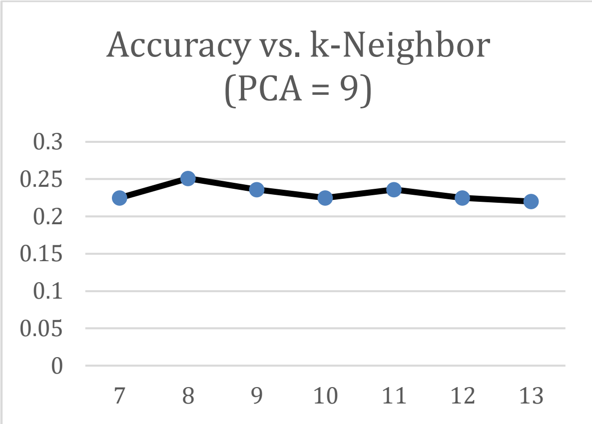



In the recognizing mode [8][9] (Target is Phoebe), Phoebe is recognized in both images and face mask is only applied to her):




The accuracy vs. PCA choosing relation is plotted in the following:





Improvement

Compared with Atul Balaji's classification of emotions[10] with TensorFlow of an average accuracy of around 60%, the eigenface method performs poorly in prediction accuracy. One reason is that our classification dataset uses less training images (<1,000) than Balaji's training set (>10,000).

Also, in classifying the emotions with eigenface method, we did not make the training itself being robust to rotation and projection, and even after normalization, faces still vary a great amount, which makes each emotion model less generic. The future improvement lies in applying image transformation into the pre-processing of the image set to make it robust to rotation and projection.

In recognizing one target object, we extract the distance between each face's nose to the other five facial landmarks (left eyebrow, right eyebrow, left eye, right eye, and mouth). We then normalize the distance and form a distance vector as the descriptive vector. This description is not robust to projection, where the distance between nose and left/right eyes can vary significantly. If the photo album has a person with very different angles, it will fail to recognize the same person.

Works Cited

[1] OpenCV/ data/ haarcascades (2013), GitHub repository: opencv https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_alt.xml
[2] *Eigenfaces for Recognition*, Matthew Turk and Alex Pentland, Journal of Cognitive Neuroscience 1991 Vol. 3, 71-86
[3] *Finding your Lookalike: Measuring Face Similarity Rather than Face Identity*, Amir Sadovnik, Wassim Gharbi, Thanh Vu, Lafayette College, Easton, PA, 2018
[4] jrosebr1/imutils (2015), GitHub repository: imultils https://github.com/jrosebr1/imutils
[5] Kanade, T., Cohn, J. F., & Tian, Y. (2000). Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), Grenoble, France, 46-53.
[6] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101.
[7] Michael J. Lyons, Shigeru Akemastu, Miyuki Kamachi, Jiro Gyoba. Coding Facial Expressions with Gabor Wavelets, 3rd IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205 (1998).
[8] *15 Punny Best Friend Instagram Captions For National Best Friends Day 2018*, Bustle, https://www.bustle.com/p/15-punny-best-friend-instagram-captions-for-national-best-friends-day-2018-9260558
[9] Friends, Wikipedia, https://en.wikipedia.org/wiki/Friends
[10] atulapra/Emotion-detection (2016), GitHub repository: Emotion-detection https://github.com/atulapra/Emotion-detection