**A Face Recognition Application with Pre-trained Models**

Face Recognition (FR) has been becoming one of the mostly debated topics in the machine learning, or more specifically, deep learning. It is well known, though maybe little out of date, convolutional neural network is perfectly suitable for the image-related problem. In this small project, we explored a range of ways to realize the single and multiple FR.

Our project is to develop an FR application that can detect the faces in the test photos and return the possibility of each label in the training photos. We have 5 persons in the training data, each 20 photos.

FR is usually done in two ways. One is to train a neural-network based model, preferably a ConvNet model, to classify the images. However, in terms of human faces, the vast compute and time resources with a great number of images is required to achieve a high enough accuracy. We do not have such a gigantic dataset. Thus, we opted the second approach in the end, transfer learning. Namely, we borrowed the results from some well-trained FR model.

We explored

simple logistic regression,

CNN,

part of VGG (pre-trained ~10h by labled faces in the wild (LFW)),

mobileNet (pre-trained ~12h by LFW),

and, [pre-trained vgg16](https://github.com/rcmalli/keras-vggface).

We finally achieved the training accuracy 0.99 and test accuracy 1.00 using the pre-trained vgg16 in keras.

**Detect single face**

1. The face in the target image would be detected. Haar Cascade would return the rotated and cropped face with the highest score.
2. Vggface embeddings would calculate for the target face (after step 1) and the faces in the training data, with the labeled predicted by vggface.
3. Distances between the embeddings of the target face and training date with the predicted label would be caculated. And in this sense, we display the most similar 3 images. (Note that they are the photos of the same person.)

**Detect multiple faces**

1. The faces in the target image would be detected. It is done by the function '.detecMuliScale3'.
2. Vggface embeddings would calculate for the target faces (after step 1) and the faces in the training data, with the labeled predicted by vggface.
3. Distances between the embeddings of each of the target faces, and training data with the predicted label would be caculated. And in this sense, we display the most similar 3 images of each of the target faces.