Machine Learning Project
Face Recognition Attendance System with Liveness Detection

1 Introduction

Face recognition can be classified into two categories: face classification and face verification. Face classification involves classifying a person's face into the correct face ID, while face verification involves determining whether two face images are of the same person. We can consider face classification as the closed-set problem where the face ID is known in advance by the model, and face verification is the open-set problem where the model may encounter a new face identity.

In this project, you will deploy the concept of **face verification** using convolutional neural network to design an end-to-end system for a face recognition attendance system for an enterprise. You will have to consider the following scenario:

- You need to register a new faceID when hiring new employees.
- You need to integrate an anti-spoofing module into your attendance system.

2. Face Verification

The input to your system will be a *trial*, that is, a pair of images of faces that may or may not belong to the same person. Your goal is to produce a numerical score that quantifies the similarity of the faces in the two images. A simple approach is to flatten each image matrix into a vector, and then calculate the Euclidean distance between two vectors, as shown in Fig 1. A smaller distance indicates greater confidence that the faces in the two images are the same person.

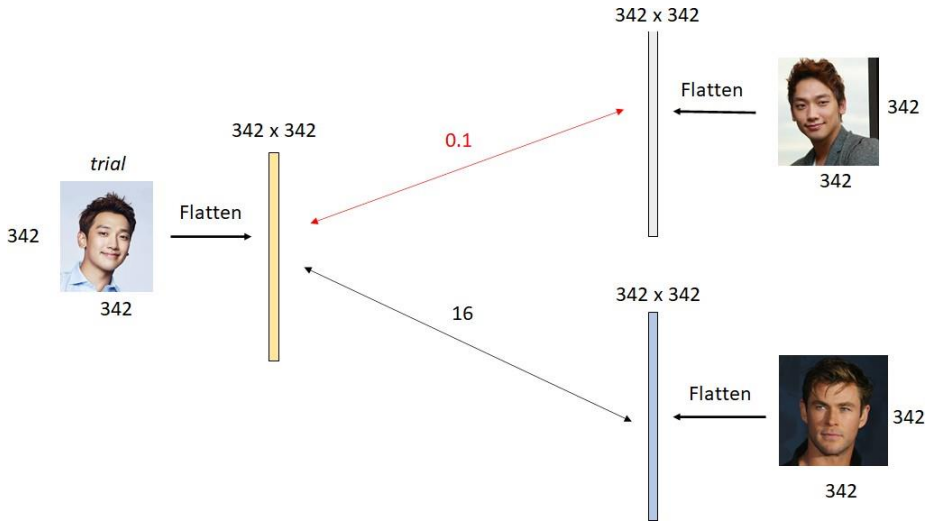

Fig. 1: The concept of face verification

3. Getting Started

In this project, you will explore the following elements to design a face recognition attendance system:

3.1 Face Embedding

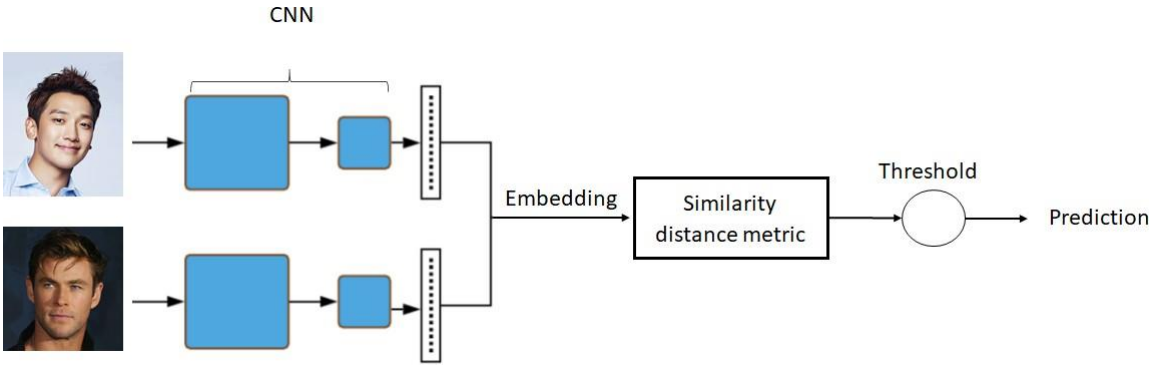

Fig 2: Face embedding similarity comparison

You are not really encouraged to directly compute the distance between two image matrices for two reasons:

- Flattened images are usually high dimensional, which leads to additional computational costs.
- The features of the original images are not discriminative enough.

Therefore, in this project, your task is to *train* a convolutional neural network (CNN) model to extract compact, low-dimensional features that retain the most important information in the image and are also discriminative. These compact features will be represented in a fixed length vector, known as the *face embedding*. Your end-to-end face verification system will function in the following way: given two images, each image will be passed through the CNN to generate the corresponding face embeddings, you will use an appropriate metric between the embeddings to produce the similarity scores. The end-to-end face verification framework is shown in Fig. 2.

There are different approaches you can use to train the CNN to extract discriminative face embedding. These approaches can be based on **self-supervised learning**, such as *metric learning*, which directly models the similarity between two images, or **supervised learning**, which is *classification* based on identity classes. ***In this project, at the minimum, you should implement these two different approaches and compare their performance.***

3.2 System Evaluation for Face Verification

To ensure a consistent comparison, each group should use similar evaluation metrics to assess the performance of your model for the face verification task, namely the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC).

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. Given the similarity scores for many trials, some threshold is set to accept or reject pairs, i.e., accept when the similarity score is above the threshold or reject when the similarity score is below the threshold.

The AUC of the ROC curve represents the probability that a classifier will rank a randomly chosen positive pair (same person) higher than a negative pair (different people) based on similarity scores.

3.3 Similarity Distance Metric

You need to do some research to select an appropriate distance metric for the face verification task. The two most popular distance metrics used in verification are cosine similarity and Euclidean distance. Both of these metrics are capable of achieving state-of-the-art scores, you can test and compare both performances.

4. Anti-Spoofing Module

Facial recognition systems are susceptible to being deceived by "spoofed" or "non-real" faces, such as printed papers or images displayed on mobile phones. In order to enhance the security of these systems, you are required to develop effective methods for detecting these counterfeit faces, a process commonly referred to as liveness detection. Various approaches, including texture analysis (utilizing learned or hand-crafted features) and movement analysis, can be employed to address this challenge.

During the demo, the evaluation of the anti-spoofing module will be tested by its capability of countering the spoofing attempts of the panel during the demo.

5. Dataset

The dataset used is based on the dataset published in Kaggle. You can download it in Canvas. The structure of the dataset folder is as follows:

- classification_data: There are three sub-folders in train_data, val_data and test_data. Each contain images of one person and the name of that subfolder represents their ID.
  - The train_data is used as the training set for both the *classification* and the *metric learning task*.
  - val_data is the validation set used to validate the classification accuracy. You only use it for classification task, so you can skip this for metric learning.
  - test_data is the testing set used to test the classification accuracy. You only use it for classification task, so you can skip this for metric learning.

- verification_data: This is the directory that contains the images for the Verification Validation
- verification_pairs_val.txt: This file contains the trials for Verification Validation. The first two column are the images path of the trial. The third column contains the true label for the pair (1 is same, 0 is not same). This verification set is also your test set as well. You are supposed to use the data in this file to validate (test) your AUC score.

6. User Interface for Face Recognition Attendance System

The expected input to the system is the image of a person's face and the expected output is to check if the person has been registered in the database, if so, what is his/her identity? If not, the system must register a new identity. You may choose to implement a GUI to allows user to easily access and manipulate available functions. Note that the anti-spoofing system must be able to detect non-real faces during the facial recognition process.

Marking Scheme

Requirements	***Mark***
1. Face Recognition Attendance System Framework (15%): Implement the two baseline approaches: classification and metric learning forface verification, and the anti-spoofing system.	30
2. Project Report (10%): This report comprises two sections: *Methodology* and *Results and Discussion*. It should provide a detailed description of the overall framework, including face recognition and the spoofing algorithm and approaches. The report presents comprehensive details of each model, encompassing information such as the training scheme, loss function, hyperparameters, and other relevant aspects. Furthermore, the report discusses the performance differences observed between the models and justifies the selection of the model that yielded the bestresults.	20
3. User Interface (10%) The minimum requirement for the user interface of your face recognition attendance system is that it must be able to meet the criteria listed in Section 6.	20
4. Project Presentation (10%): You are required to present the Face Recognition Attendance System to the lecturer / tutor. You will be asked to do a live demonstration. The date and time of the presentation will be announced later.	20
5. Innovation (5%) Any innovation that extends the current basic system.	10
Total	**100**

SUBMISSION:

- Project report.
- Source code.