COS30082 Applied Machine Learning

Project: Facial Recognition with Emotion & Liveness

1 Introduction

Face recognition can be classified into two categories: face classification and face verification. Face classification involves assigning a person's face to the correct face ID, while face verification involves determining whether two face images belong to the same person. Face classification is considered a closed-set problem, where the face IDs are known in advance by the model. In contrast, face verification is an open-set problem, where the model may encounter previously unseen face identities.

In this project, you will implement face verification using a convolutional neural network (CNN) to design an end-to-end face recognition attendance system for an enterprise. You must consider the following scenarios:

- Registering a new face ID when hiring new employees.
- Integrating an anti-spoofing module into your attendance system.
- Integrating an emotion detection module into your attendance system.

Although mature products for face recognition exist, this project focuses on applying machine learning techniques. Instead of using existing tools, we will build our own face recognition system. This hands-on approach allows us to better understand the core principles behind face verification.

2. Face Verification

The input to your system will be a trial — a pair of face images that may or may not belong to the same person. Your goal is to produce a numerical score that quantifies the similarity between 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 the two vectors, as shown in Fig. 1. A smaller distance indicates greater confidence that the faces in the two images belong to the same person.

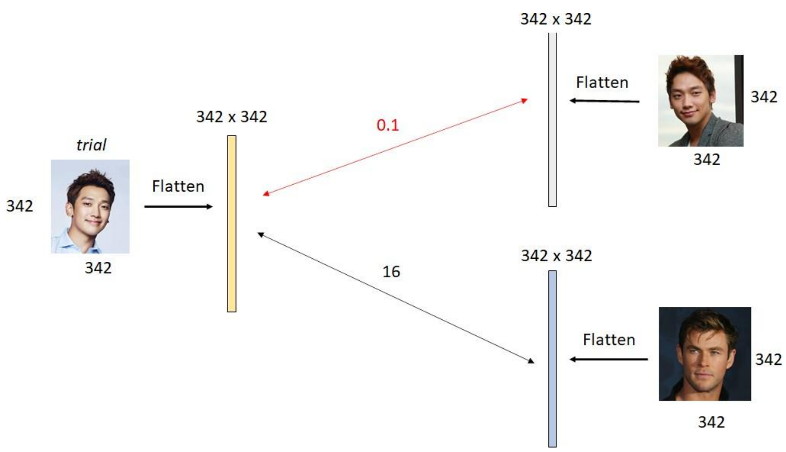

Fig. 1: The concept of face verification

3. Getting Started

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

3.1 Face Embedding

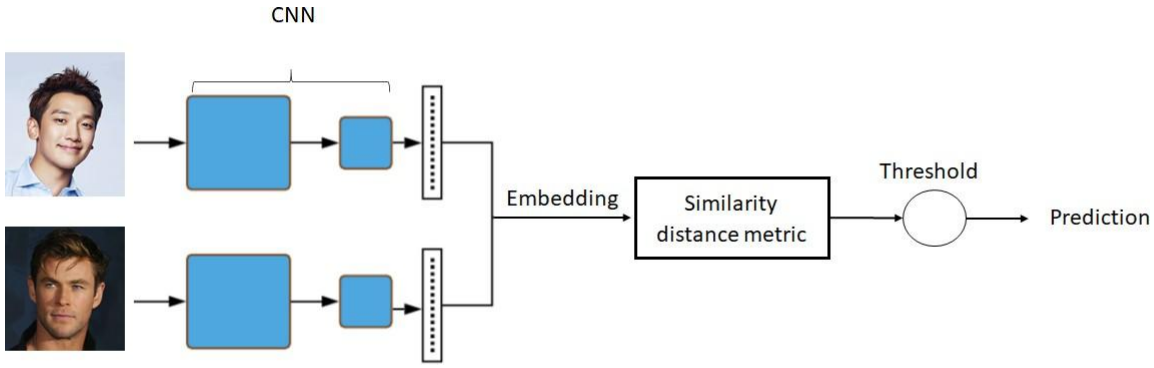

Fig 2: Face embedding similarity comparison

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

- Flattened images are usually high-dimensional, leading to increased computational costs.
- The features in the original images are not sufficiently discriminative.

Therefore, in this project, your task is to train a convolutional neural network (CNN) to extract compact, low-dimensional features that retain the most important information from the image while remaining discriminative. These compact features will be represented as fixed-length vectors, known as **face embeddings**. Your end-to-end face verification system will function as follows: given two images, each image is passed through the CNN to generate its corresponding face embedding. An appropriate metric is then applied to the embeddings to produce a similarity score. The end-to-end face verification framework is illustrated in Fig. 2.

There are different approaches you can use to train a convolutional neural network (CNN) to extract discriminative face embeddings. The two common methods are:

- **Metric Learning (Self-Supervised):** Metric learning trains a CNN to map face images into an embedding space where similar faces are close together and different faces are far apart. For example, in the triplet loss method, an image called the anchor is compared to a positive sample (same identity) and a negative sample (different identity). The model learns to minimize the distance between the anchor and the positive, while maximizing the distance between the anchor and the negative.

- **Supervised Learning (Classification-Based):** Supervised learning treats face recognition as a classification problem, training the CNN to assign each image to a known identity class. A softmax layer is used during training, and once the model learns to classify correctly, the representation from the layer before the softmax is used as the face embedding.

In this project, you are expected to implement **both** approaches and compare their performance.

3.2 System Evaluation for Face Verification

Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) will be used as evaluation metrics to assess model performance.

A threshold is used to decide whether to accept or reject a pair — that is, a pair is accepted if the similarity score is above the threshold and rejected if it is below. The ROC curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

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

3.3 Similarity Distance Metric

You need to conduct some research to select an appropriate distance metric for the face verification task. The two most commonly used distance metrics are **cosine similarity** and **Euclidean distance**. Both metrics are capable of achieving state-of-the-art performance. You are encouraged to test and compare the performance of both.

4. Anti-Spoofing Module

Facial recognition systems are vulnerable to being deceived by "spoofed" or "non-real" faces, such as printed photos or images displayed on mobile phones. To enhance the security of these systems, you are required to develop effective methods for detecting such counterfeit faces. This process is commonly referred to as liveness detection.

The anti-spoofing process must be seamlessly integrated with the existing facial recognition system. You may use existing tools to support your implementation.

5. Emotion Detection Module

To enhance functionality, you are required to implement an emotion detection module. This module should be capable of identifying a user's emotional state from their facial expression in real time. The detected emotion can be displayed as icons accompanied by emotion labels. This will enrich user interaction and provide additional insights for enterprise applications, such as workplace sentiment analysis or mental health monitoring.

You may use pre-trained models or train your own model using publicly available datasets. The emotion detection process must be seamlessly integrated with the existing facial recognition system.

6. Dataset

The dataset used in this project is based on a public dataset from Kaggle. The folder structure is as follows:

- classification_data/ This folder contains three subfolders: train_data, val_data, and test_data. Each subfolder includes images of one person. The folder name is their ID.

    - train_data: Used for training both classification and metric learning models.
    - val_data: Used to validate classification accuracy. Not needed for metric learning.
    - test_data: Used to test classification accuracy. Also not used for metric learning.
- verification_data/   This folder contains images used for face verification tasks.
- verification_pairs_val.txt   This file contains verification trials.
    - The first two columns are image paths.
    - The third column is the label: 1 for the same person, 0 for different people. Your AUC score for face verification should be computed based on this file.

7. User Interface

The expected input to the system is an image of a person's face. The expected output is to determine whether the person has already been registered in the database. If so, the system should return their identity; if not, it must register a new identity. The system should also detect spoofed or non-real faces and display the corresponding emotion. You may choose to implement a graphical user interface (GUI) to allow users to easily access and interact with the available functions.

8. Project Report

The report should include at least two sections: **Methodology** and **Results and Discussion**. It must provide a detailed description of the overall system framework, including components for face verification, anti-spoofing, and emotion detection. Each model used should be described comprehensively, covering aspects such as the training scheme, loss function, hyperparameters, and any other relevant details. The report should also discuss performance differences between the models and justify the selection of the best-performing approach based on evaluation results.

9. Marking Scheme

Marks are based on the implementation of key components and your understanding of the required knowledge and techniques, as outlined in the criteria below.

Assessment Criteria	**Mark**
1. Face Recognition Attendance System Framework Assessment of the implementation of the two baseline approaches: classification and metric learning for face verification, as well as any additional approaches used.	30
2. Anti-Spoofing Assessment of the design and effectiveness of the liveness detection module for identifying spoofed faces (e.g., printed images or screen photos).	15
3. Emotion Detection Assessment of the integration and performance of the emotion recognition module in identifying emotional states (e.g., happy, sad, angry) and displaying them appropriately in the user interface.	10
4. User Interface Evaluation of the usability, functionality, and completeness of the graphical user interface (GUI), including features for registration, verification, liveness detection, and emotion display.	10

5. Project Report Evaluation of the clarity and completeness of the project report, focusing on how well the Methodology and Results and Discussion sections are presented. Marks will be awarded based on the quality of explanation, organization, and justification of key design decisions.	20
6. Project Demonstration and Innovation Assessment of the quality and clarity of the project demonstration. Innovation and any features that extend beyond the basic requirements will be awarded higher marks.	15
Total	**100**

10. Submission

- Project report.
- Source code.