CV Final Project

Shiwen Cao, Jinlong Zhang, Xuanming Liu, Chaorui Zhang
SID: 12113024, 12112528, 12112929, 12113001

June 15, 2024

Abstract

Facial recognition technology has become increasingly sophisticated, driven by advancements in computer vision and deep learning. In this project, we leverage the DeepFace library to implement a facial recognition system capable of verifying identities, analyzing facial attributes, and recognizing faces in images. Our system demonstrates robust performance in static image analysis but faces challenges in real-time video processing due to high computational demands. The complete implementation of our project can be found on GitHub at `https://github.com/cswccc/CV-Project/tree/main`.

1 Introduction

1.1 Face Recognition

Facial recognition is a sophisticated bio-metric technology that identifies and verifies individuals by analyzing and comparing their facial features. This technology has made significant strides in recent years, driven by advancements in computer vision, pattern recognition and deep learning.
The process of facial generally involves several critical steps:

1. **Face Detection**: The initial step in facial recognition is detecting faces within an image or video. This involves identifying the regions where faces are located. Techniques such as the Viola-Jones algorithm, Histogram of Oriented Gradients (HOG), and more recently, Convolutional Neural Networks (CNNs), are commonly employed for this purpose [6]. These methods enable the system to accurately locate faces under various conditions.

2. **Feature Extraction**: Once a face is detected, the next step is to extract distinctive features. These features include the relative position, size, and shape of the eyes, nose, mouth, and other facial landmarks. Advanced systems use deep learning models to generate high-dimensional feature vectors that encapsulate the unique characteristics of a face, allowing for robust recognition even with variations in lighting, angle, and facial expressions [1].

3. **Feature Comparison**: The extracted features are then compared against a database of known faces. This comparison involves calculating the similarity between the feature vector of the detected face and those stored in the database. Techniques such as Euclidean distance or cosine similarity are commonly used to quantify this similarity [5].

4. **Identity Verification**: Based on the similarity score, the system determines if the detected face matches any faces in the database. If a match is found, the individual's identity is verified. This process ensures accurate and efficient identification and authentication of individuals [2].

Facial recognition technology, with its high precision and efficiency, is often used in fields related to identity authentication, including facial unlocking and payment authentication of our smart devices, public place monitoring and supervision, and attendance system face scanning and clock in.

1.2 DeepFace

DeepFace is an open-source lightweight facial recognition and facial attribute analysis library that includes multiple high-performance models such as VGG-Face, FaceNet, OpenFace and DeepFace, etc [3]. This library is particularly valuable for researchers, developers, and practitioners who require advanced facial recognition capabilities in their applications.

DeepFace, as an open-source facial recognition library, offers a range of advantages that make it a popular choice. These advantages include high accuracy, ease of use, flexibility, comprehensive functionality, and community support.

1. **Ease of Use**: DeepFace is designed with user-friendliness in mind. The library provides simple and intuitive APIs, which allow users to quickly integrate facial recognition capabilities into their applications.

2. **Flexibility**: It offers flexibility by supporting multiple pre-trained models. This allows users to choose the model that best fits their specific needs in terms of accuracy, speed and computational resources. At the same time, DeepFace also supports various facial detection algorithms, including a variety of high-performance algorithms such as YuNet and Fast MtCnn.

2 Related works

Face recognition technologies have rapidly advanced in recent years, with several open-source libraries emerging to facilitate implementation. Here we will briefly introduce insight face and face-recognition, and compare their differences with DeepFace.

2.1 Face_Recognition

Face_recognition is an open-source Python library built on top of the dlib machine learning library, specifically designed to facilitate the implementation of face detection and face recognition tasks. Face_recognition provides a concise API interface, reducing the requirements for users. Meanwhile, due to its efficient implementation, face_recognition can be used for real-time face recognition tasks, suitable for applications like live video processing.

However, compared to DeepFace, face_recognition offers fewer customization options and flexibility. It is primarily designed for straightforward face recognition tasks. And this library is based on adult images, so the effect on children is not good and the robustness is low.

2.2 Insightface

InsightFace is a high-performance open-source face recognition library that leverages advanced deep learning models to achieve state-of-the-art accuracy in face recognition tasked. Developed by the DeepInsight team, InsightFace provides tools for training and deploying face recognition models and has gained significant traction in both academic research and industrial applications due to its robustness and efficiency.

InsightFace incorporates cutting-edge models like ArcFace, CosFace and SphereFace, which are known for their high accuracy and robustness in face recognition. This library is optimized for both speed and accuracy, making it suitable for large-scale deployments. It also supports efficient

model inference on various hardware platforms including GPUs, which accelerates the processing time significantly.

The advanced features and capabilities of InsightFace come with a more complex setup and configuration compared to simpler libraries. And it requires significant computational resources.

2.3 Summary of Comparision

Feature	DeepFace	Face_recognition	InsightFace
Models Supported	VGG-Face, FaceNet, OpenFace, DeepID, ArcFace	Built on Dlib, uses HOG and CNN	ArcFace, CosFace, SphereFace
Ease of Use	High	Very High	Moderate
Accuracy	Very High	Moderate	Very High
Flexibility	High	Low	High
Computational Requirements	High	Low to Moderate	High
Additional Features	Facial attribute analysis (age, gender, etc.)	Basic face recognition	Training and inference tools
Ideal Use Cases	General-purpose applications, research	Educational purposes, simple tasks, real-time	Large-scale systems, commercial use, research

Table 1: Comparison of DeepFace, Face_recognition, and InsightFace

Table 1 demonstrates the differences between DeepFace, Face_recognition, and InsightFace.

3 Experiments

3.1 Datasets

In our experiments, we utilize the VGGFace2 dataset, which is a large-scale face image dataset containing approximately 3.3 million images from 9,131 different identities [4]. The dataset comprises images collected from the internet, exhibiting a wide range of variations in terms of environment, illumination, pose, and expression. With its diverse and extensive collection of images, VGGFace2 serves as a valuable resource for training and evaluating face recognition models.

3.2 Implementation Details

To achieve facial recognition functionality, we first encapsulated a FaceRec class to implement calls to DeepFace. At the sametime, the results returned by the DeepFace call were processed to meet our needs.

We mainly used the three functions of DeepFace: Deepface.verify for the facial verification, Deepface.analyze for facial attribute analysis and DeepFace.find for facial recognition. For these three functions, we have designed a UI interfaces to operate them separately.

3.2.1 Facial Verification

We designed verifyTwoFaces() method in FaceRec to achieve this goal. It input two images and output a list, which contains verified, face1-box, face2-box, timecost. Verified stands for whether two images are the same person, face1-box and face2-box stand for the position of recognized face in two images respectively, timecost is the running time of the method. Additionally, if the method doesn't recognize the face in the image, the list will be null. We also designed some UI by Python, we show the result in the bottom line: if two images are the same person's face, the line will be "They are the Same person."; if two images are not the same person's face, the line will be "They are not the Same person."; if no face recognized, the line will be if two images are the same person's face, the line will be "They are the Same person."; if images are not uploaded, the line will be "Please upload two images first."; Otherwise, the line will be the error message. We also provide drawing rectangle box on images if they are recognized as face. Some of UI examples are as follows:

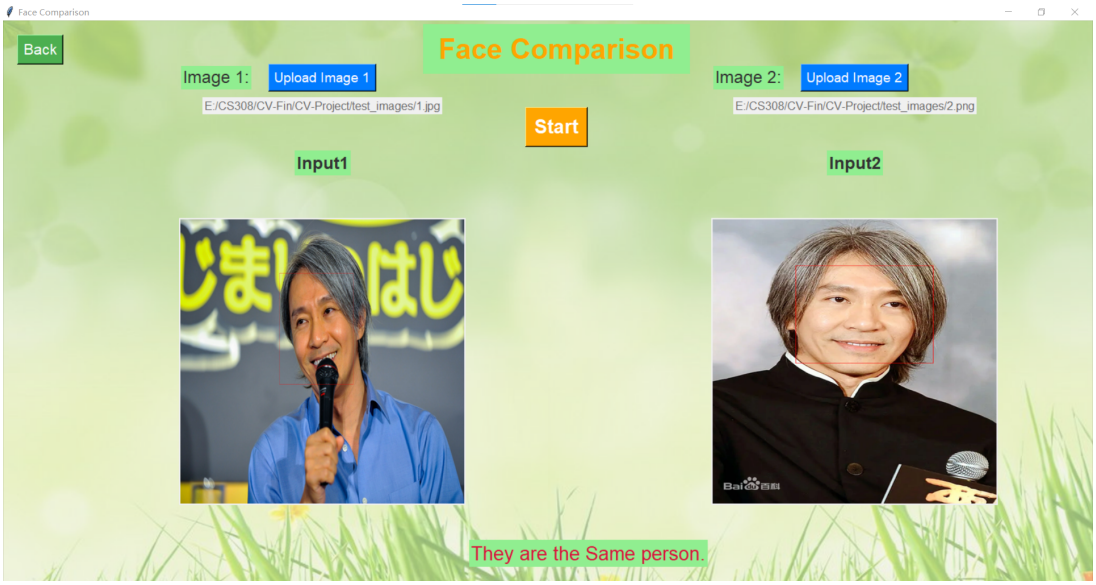

Figure 1: Facial Verification Case1

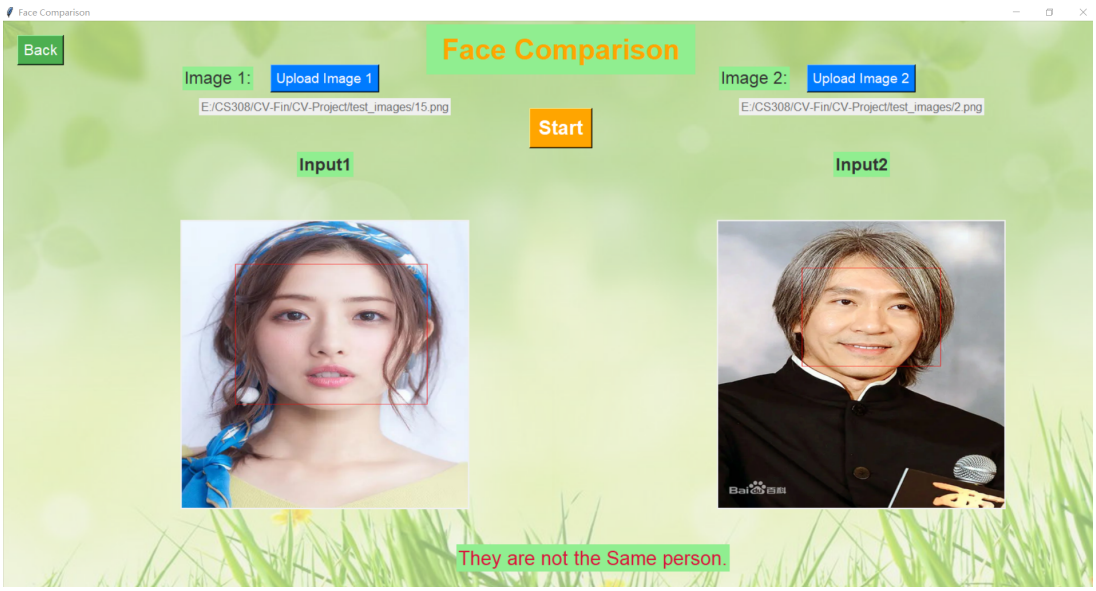

Figure 2: Facial Verification Case2

3.2.2 Facial Attribute Analysis

In the facial attribute analysis section, we have implemented attributes display for multiple faces, and for each face, it has a unique ID. After uploading the image, we call the FaceRec's method to analyze the image, and then display the analyzed information and photos in real-time. Fig.3 shows the analysis results of multiple faces.

Figure 3: Attribute analysis for different images

3.2.3 Facial Recognition

In this part, we take a photo that contains a face and look for the same face in the document database. A query is made from all the photos in the database, and the three images with the best matching faces are selected and listed. At the same time, list the identity of the image next to the selected image. You can see that the image of the corresponding person was successfully found. Besides, it also support more than one faces in one figure.

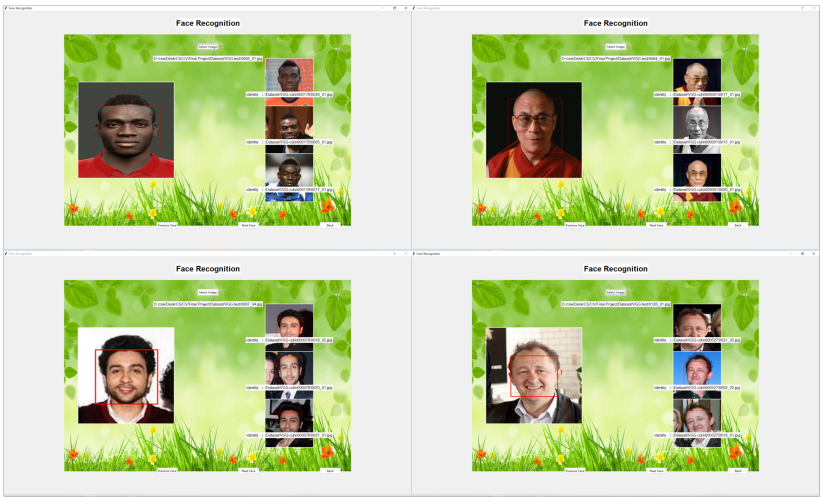

Figure 4: face recognition

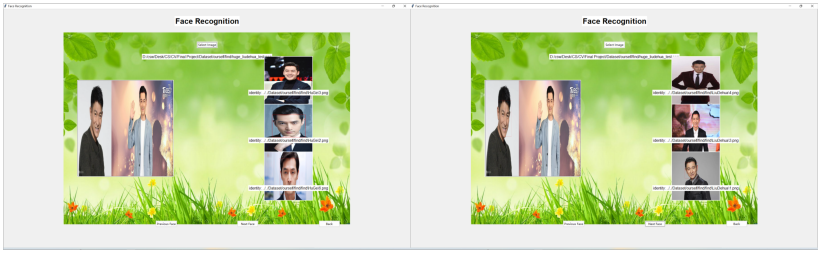

Figure 5: Caption

4 Conclusion

DeepFace, an open-source facial recognition and analysis library, offers a range of advanced models for tasks such as face detection, recognition, verification, emotion analysis, and demographic prediction. In our project, we utilized the basic features of DeepFace and implemented a simple system for facial recognition, which achieved good performance. However, due to the high reliance on computing resources, DeepFace does not perform well in real-time task. We attempted to use video as input for face recognition, but the inference speed is too slow, resulting in a very sluggish display. In the future, we are preparing to add different facial recognition libraries to provide users with more choices and better performance.

5 Contributions

Cao Shiwen (12113024): 25%
Zhang Jinlong (12112519): 25%
Liu Xuanming (12112929): 25%
Zhang Chaorui (12113001): 25%

References

[1] Guo, G., Zhang, N., & Zhu, Y. (2019). Deep learning for face recognition: a critical analysis. *ACM Computing Surveys (CSUR)*, 52(3), 1-35.

[2] Liu, W., Wen, Y., Yu, Z., & Yang, M. (2018). Large-Margin Softmax Loss for Convolutional Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning* (Vol. 70, pp. 507-516).

[3] Serengil, S. I., & Ozpinar, A. (2021). HyperExtended LightFace: A Facial Attribute Analysis Framework. *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, 1-4. doi:10.1109/ICEET53442.2021.9659697. Retrieved from `https://ieeexplore.ieee.org/document/9659697`

[4] Qiong Cao, Li Shen, Wei Xie, Omkar M. Parkhi, and Andrew Zisserman. "VGGFace2: A dataset for recognising faces across pose and age." *International Conference on Automatic Face & Gesture Recognition*, 2018.

[5] Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429, 215-244.

[6] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2020). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10), 1499-1503.