Face Detection and Recognition

Introduction:

The Face Detection and Recognition project is a Python-based application that uses computer vision and machine learning techniques to detect and recognize faces in images. It utilises the face_recognition library for face recognition and OpenCV (cv2) for face detection. This report provides an in-depth overview of the project, discussing its important features and their functions.

Key Features and Functions:

Library Imports:

OpenCV (cv2): OpenCV is a popular open-source computer vision library. In this project, it is used for face detection using a pre-trained Haar Cascade classifier.

face_recognition: This library is a user-friendly wrapper built on top of dlib's deep learning facial recognition toolkit. It provides face recognition capabilities, including face encoding and matching.

os: The os module is used for file operations, enabling the script to read image files from a specified directory.

numpy: The numpy library is employed for numerical operations and efficient data handling.

Loading a Pre-trained Face Detection Model (face_cascade):

A Haar Cascade classifier for frontal face detection is loaded using OpenCV. This model is utilised to identify faces within an image.

Loading Known Faces (known_face_encodings and known_face_names):

The script loads a dataset of known faces from a specified directory (known_faces_dir).

It iterates through image files in the directory, extracts names from filenames, and encodes each face using the face_recognition library.

Encoded face data is stored in two lists: known_face_encodings (for the face encodings) and known_face_names (for corresponding names).

Face Detection Function (detect_faces(image)):

This function takes an image as input and performs face detection.

It converts the image to grayscale, enhancing face detection accuracy.

The Haar Cascade classifier is applied to detect faces in the grayscale image.

The function returns a list of detected face rectangles.

Face Recognition Function (recognize_faces(image)):

This function takes an image as input and performs face recognition.

It converts the image from BGR (OpenCV format) to RGB (required by face_recognition).

face_recognition is used to find face locations and encodings in the RGB image.

Detected face encodings are compared with known face encodings to recognize faces.

The function returns a list of recognized names for the detected faces.

Main Function (main(image_path)):

The main function serves as the entry point of the script.

It loads the input image specified by image_path.

Face detection and recognition are performed using the recognize_faces function.

Recognized names for each detected face are printed to the console.

Entry Point of the Script:

The script's main execution block checks if it is being executed directly (if __name__ == "__main__":).

If so, it calls the main function with the path to the input image.

Conclusion:

The Face Detection and Recognition project provides a practical implementation of face detection and recognition using Python. It leverages the face_recognition library for its robust face recognition capabilities and OpenCV for efficient face detection. The project's modular structure allows for easy integration into other applications and provides a foundation for building more advanced facial recognition systems.