Document Summary: Building Face Detection and Recognition System

1. Setup Environment

Required libraries were installed using pip, including NumPy, OpenCV, scikit-learn, pandas, and matplotlib.

2. Import Libraries

Necessary libraries such as tkinter, OpenCV, NumPy, scikit-learn, pandas, and os were imported for building the system.

3. GUI for User Interaction

1. **File Paths Setup:** Defined paths for user images, users CSV, model, PCA, and label dictionary.
2. **Capture Images Function:** Implemented a function to capture images for a new user using the webcam and OpenCV. Images were saved to the appropriate directory and user information was updated in the users CSV.
3. **Train Model Function:** Developed a function to train the face recognition model using the SVM classifier with PCA for dimensionality reduction. The trained model, PCA, and label dictionary were saved to files, and evaluation metrics such as classification report and confusion matrix were printed.
4. **Load Images and Labels Function:** Created a function to load images and labels from the users CSV and respective directories.
5. **Login User Function:** Implemented a function for users to log in using Face ID. The system detects the user's face using the webcam and recognizes them using the trained model. If successful, a login message is displayed; otherwise, the user is prompted to try again.

4. Start GUI

1. **Create User Function:** Defined a function to prompt users to input a new username and capture images for registration.
2. **App GUI Function:** Created the main GUI window using tkinter with buttons for creating a new user, logging in using Face ID, and training the model. The GUI event loop was started to handle user interactions.