Faculty of Engineering

Cairo University

Computer Vision (SBE3230)

Task 5

Name	BN
Madonna Mosaad	49
Nariman Ahmed	71
Nancy Mahmoud	72
Yassien Tawfik	81

Under Supervision of :

Dr /Ahmed Badawy

Eng /Omar Hesham

Eng /Yara Wael

Table of Contents

Project Overview

This project implements a comprehensive face detection and recognition system with performance evaluation capabilities. The system is built using Python and leverages OpenCV (cv2) for image processing and face detection, along with custom implementations for face recognition and performance analysis. The project is structured in a modular way, with separate components for face detection, face recognition, and performance evaluation.

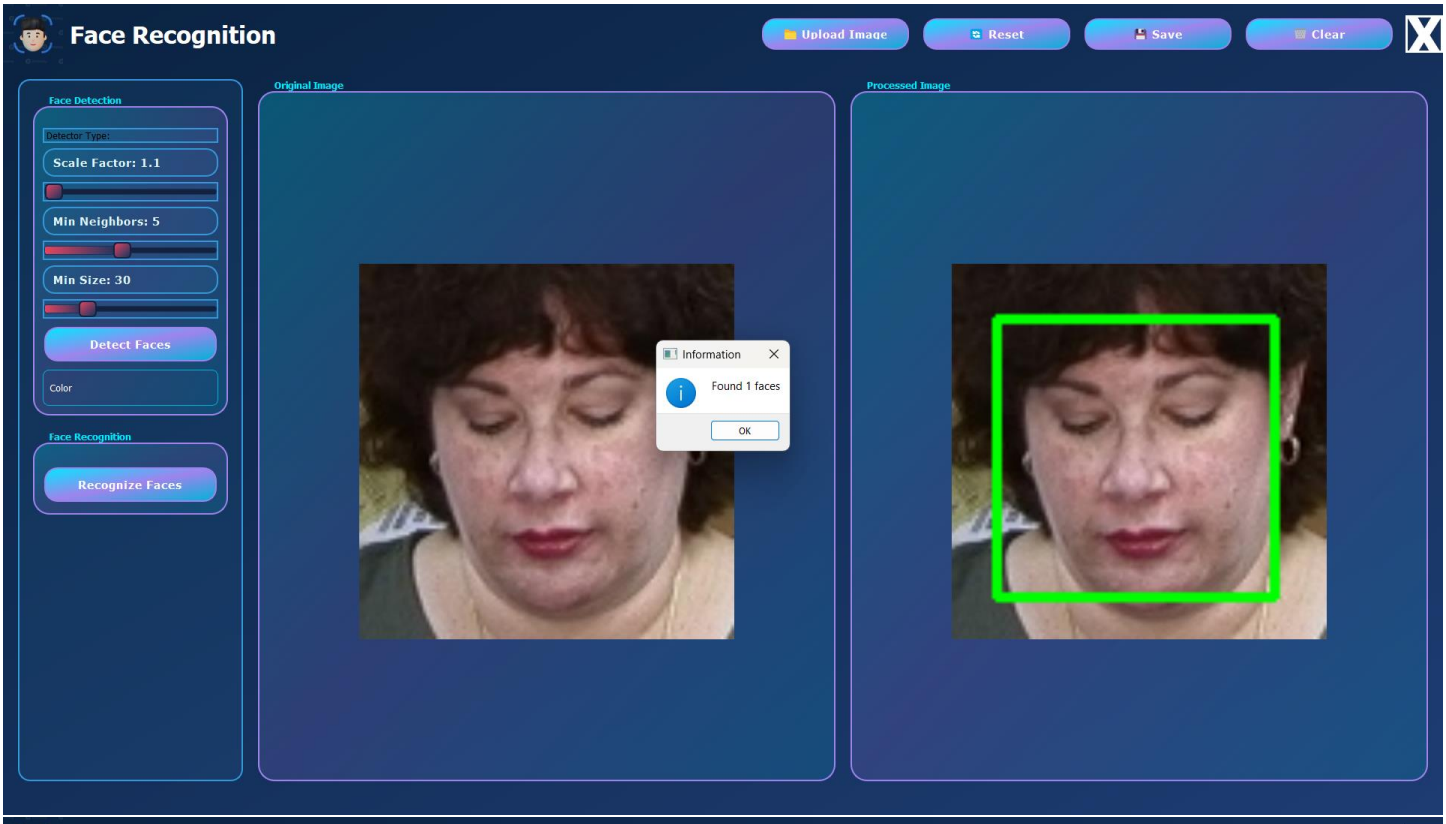
Face Detection

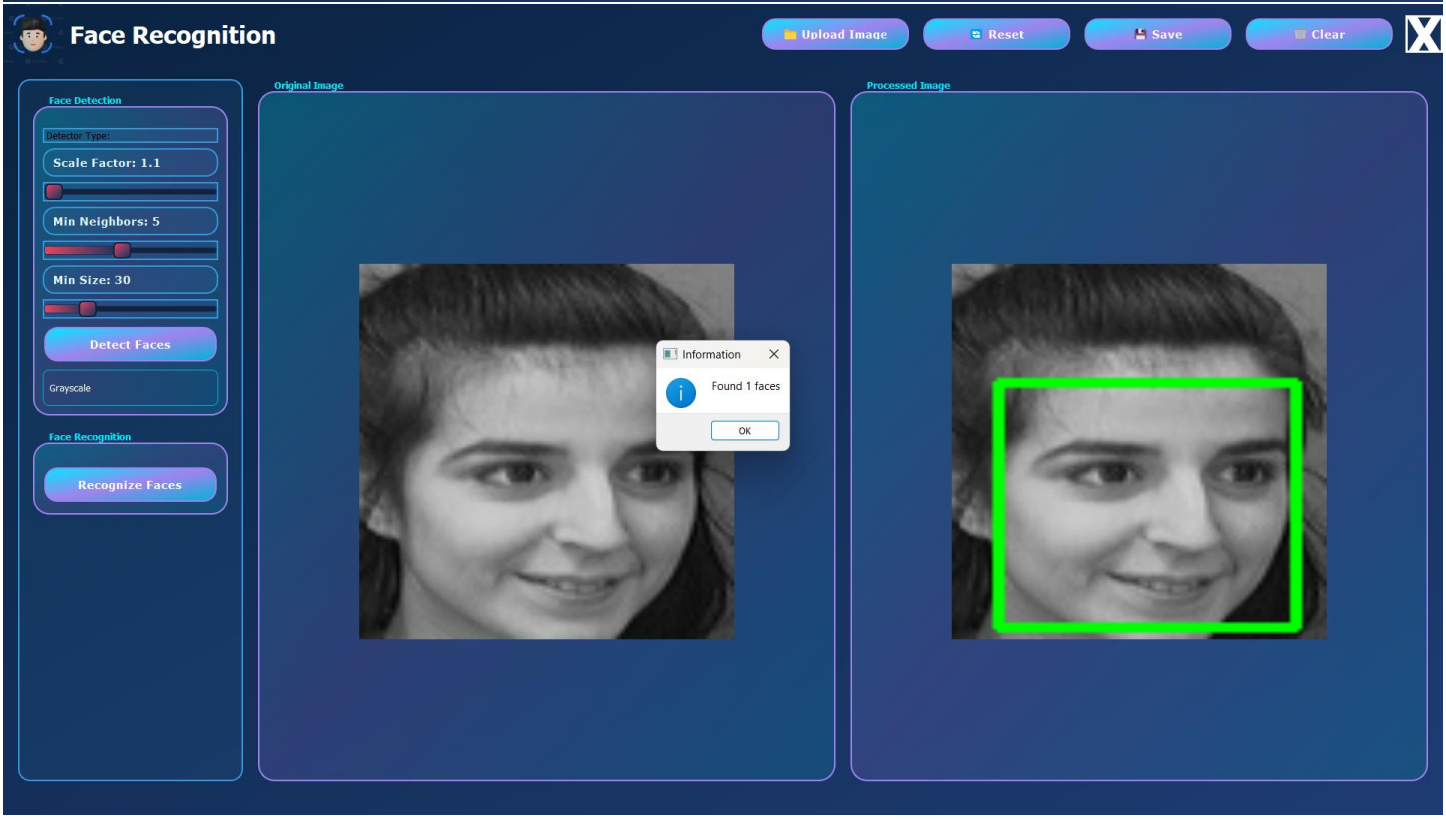
Overview

Face detection is the process of identifying and locating human faces in digital images or video frames. In this project, it involves determining the presence of faces and their spatial locations (bounding boxes) within an image.

Methods Used

1. Haar Cascade Classifier
   - The project implements face detection using OpenCV's Haar Cascade Classifier
   - Uses a pre-trained cascade classifier loaded from an XML file
   - Key parameters:
     - scale_factor: how much the image size is reduced at each scale
     - min_neighbors: minimum neighbors required for detection
     - min_size: minimum face size to detect
2. Detection Process
   - Converts input images to grayscale
   - Applies multi-scale detection using detectMultiScale
   - Returns bounding boxes (x, y, width, height) for detected faces
   - Includes visualization capabilities through draw_faces method


Face Recognition
Upload Image
Reset
Save
Clear
Face Detection
Detector Type:
Scale Factor: 1.1
Min Neighbors: 5
Min Size: 30
Detect Faces
Color
Face Recognition
Recognize Faces
Original Image
Processed Image
Information
Found 1 faces
OK


Face Recognition
Upload Image
Reset
Save
Clear
Face Detection
Detector Type:
Scale Factor: 1.1
Min Neighbors: 5
Min Size: 30
Detect Faces
Grayscale
Face Recognition
Recognize Faces
Original Image
Processed Image
Information
Found 1 faces
OK

Observations:

Face Recognition

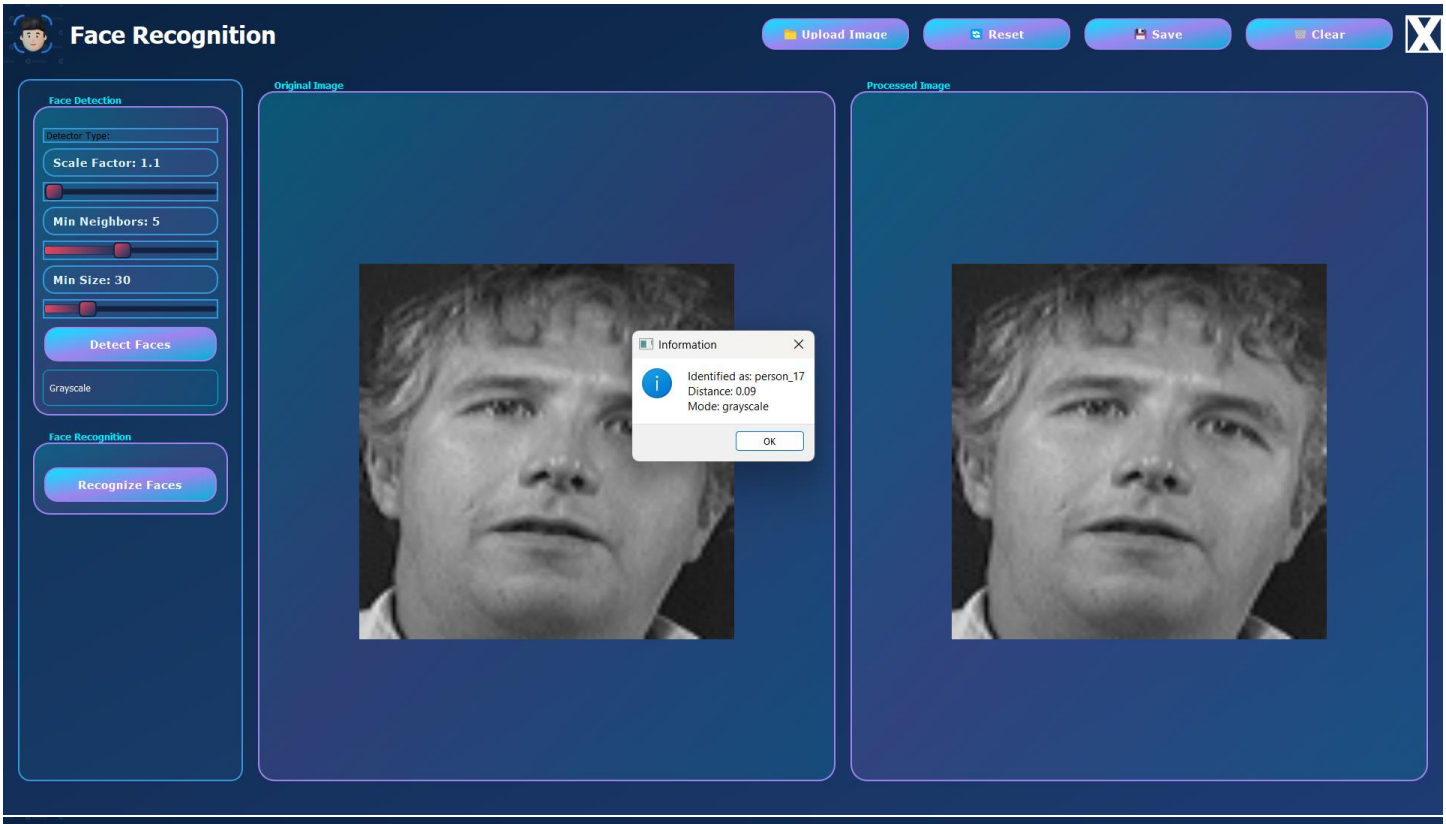
Overview

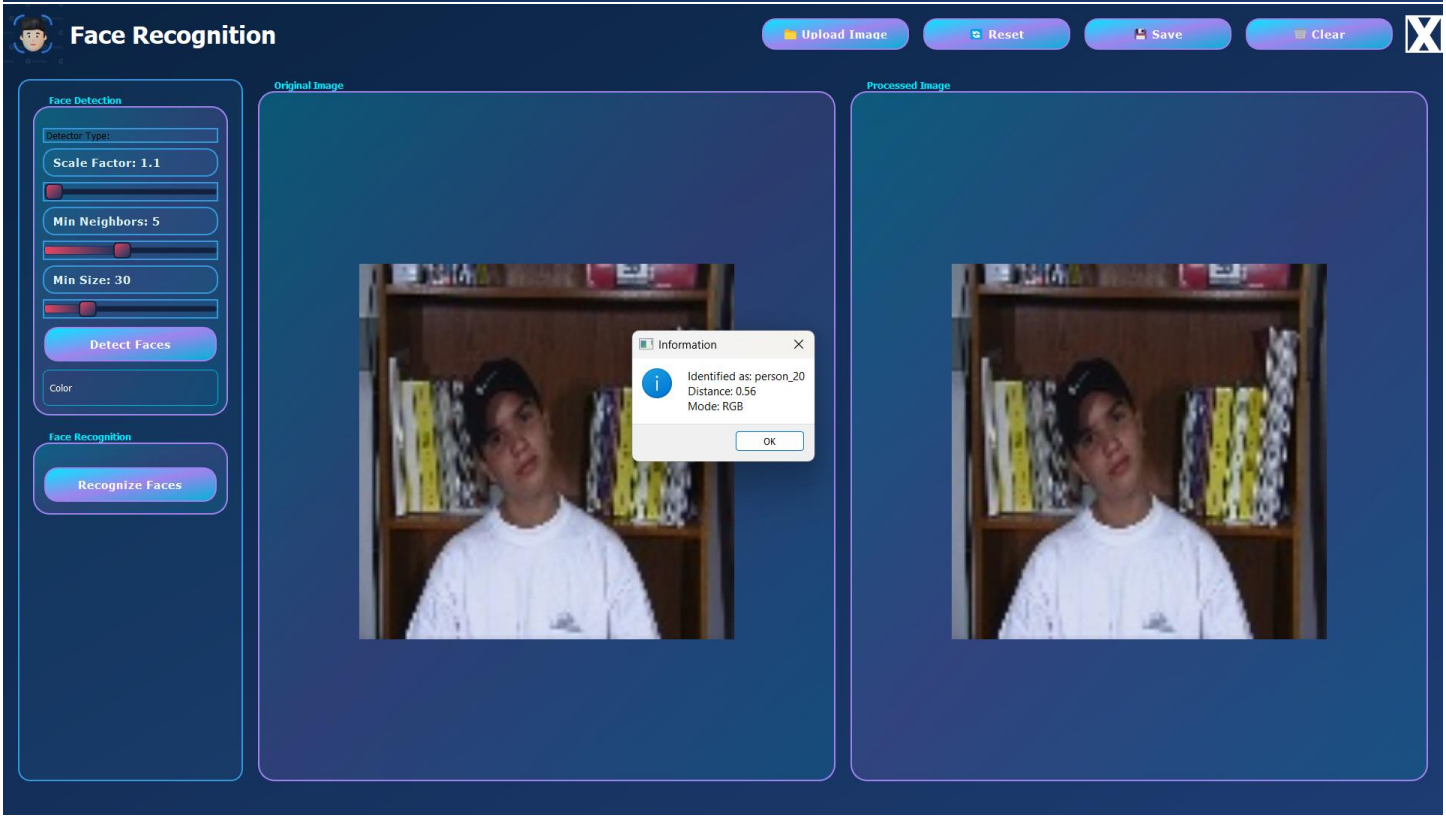
Face recognition is the process of identifying or verifying a person's identity using their facial features. It goes beyond detection by matching detected faces against a known database of faces.

Methods Used

1. PCA-based Face Recognition
   - Implements Principal Component Analysis (PCA) for face recognition
   - Uses a pre-trained model with the following components:
     - Mean vector (pca_mean.npy)
     - Standard deviation vector (pca_std.npy)
     - Principal Components (pca_components.npy)
     - Training embeddings (pca_embeddings.npy)
     - Label mapping (label_map.json)

2. Recognition Process
   - Image Preprocessing:
     - Resizes images to 100x100 pixels
     - Converts to grayscale
     - Applies Contrast Limited Adaptive Histogram Equalization (CLAHE)
     - Performs z-score normalization
     - Projects onto principal components
     - L2 normalization of embeddings
   - Recognition:
     - Uses cosine similarity for face matching
     - Returns predicted label, name, and distance score (ranging from 0 (perfect match) to 2 (completely different)).


Face Recognition
Upload Image
Reset
Save
Clear
Face Detection
Detector Type:
Scale Factor: 1.1
Min Neighbors: 5
Min Size: 30
Detect Faces
Grayscale
Face Recognition
Recognize Faces
Original Image
Processed Image
Information
Identified as: person_17
Distance: 0.09
Mode: grayscale
OK

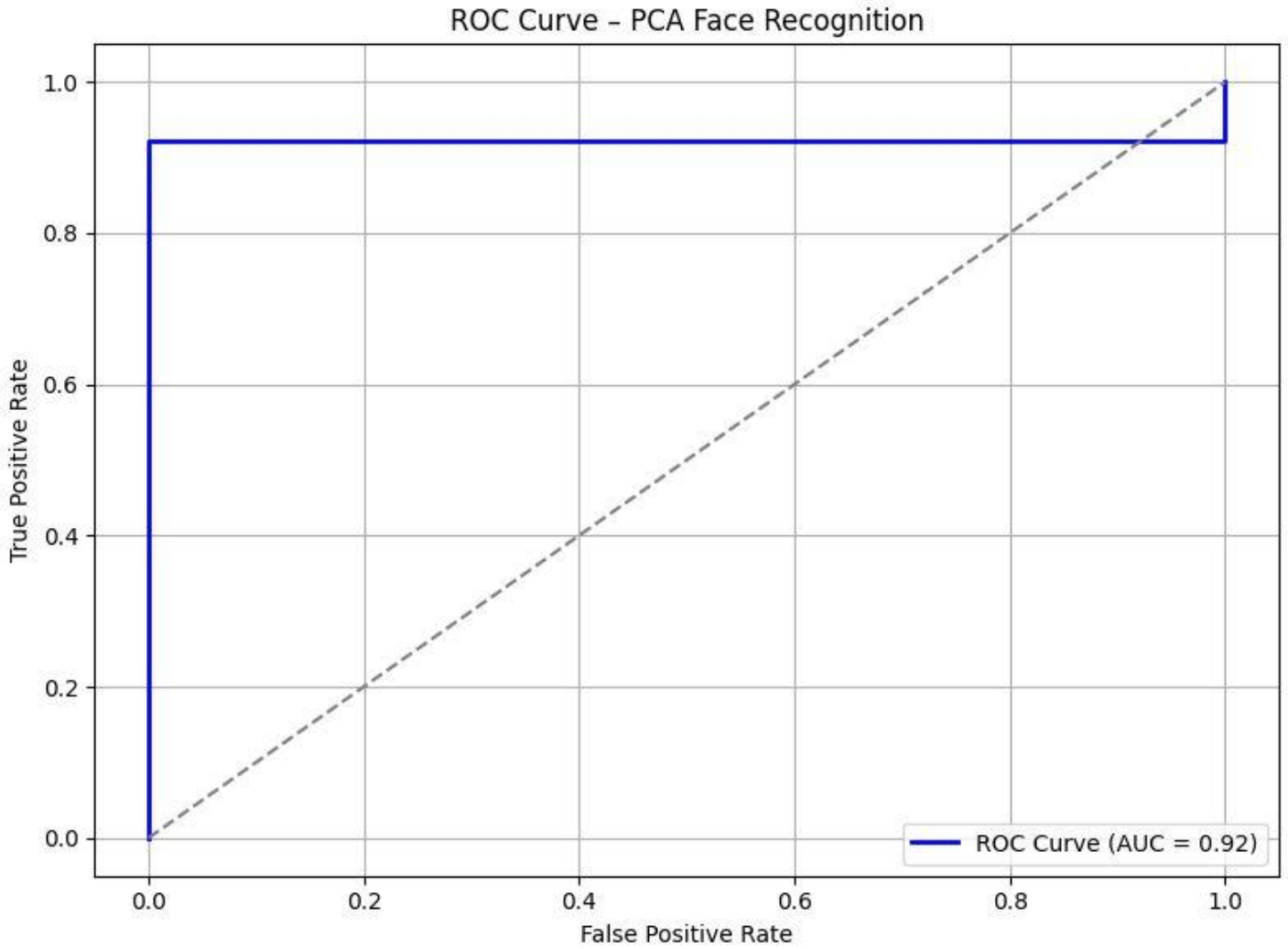

Face Recognition
Upload Image
Reset
Save
Clear
Face Detection
Detector Type:
Scale Factor: 1.1
Min Neighbors: 5
Min Size: 30
Detect Faces
Color
Face Recognition
Recognize Faces
Original Image
Processed Image
Information
Identified as: person_20
Distance: 0.56
Mode: RGB
OK

Observations:

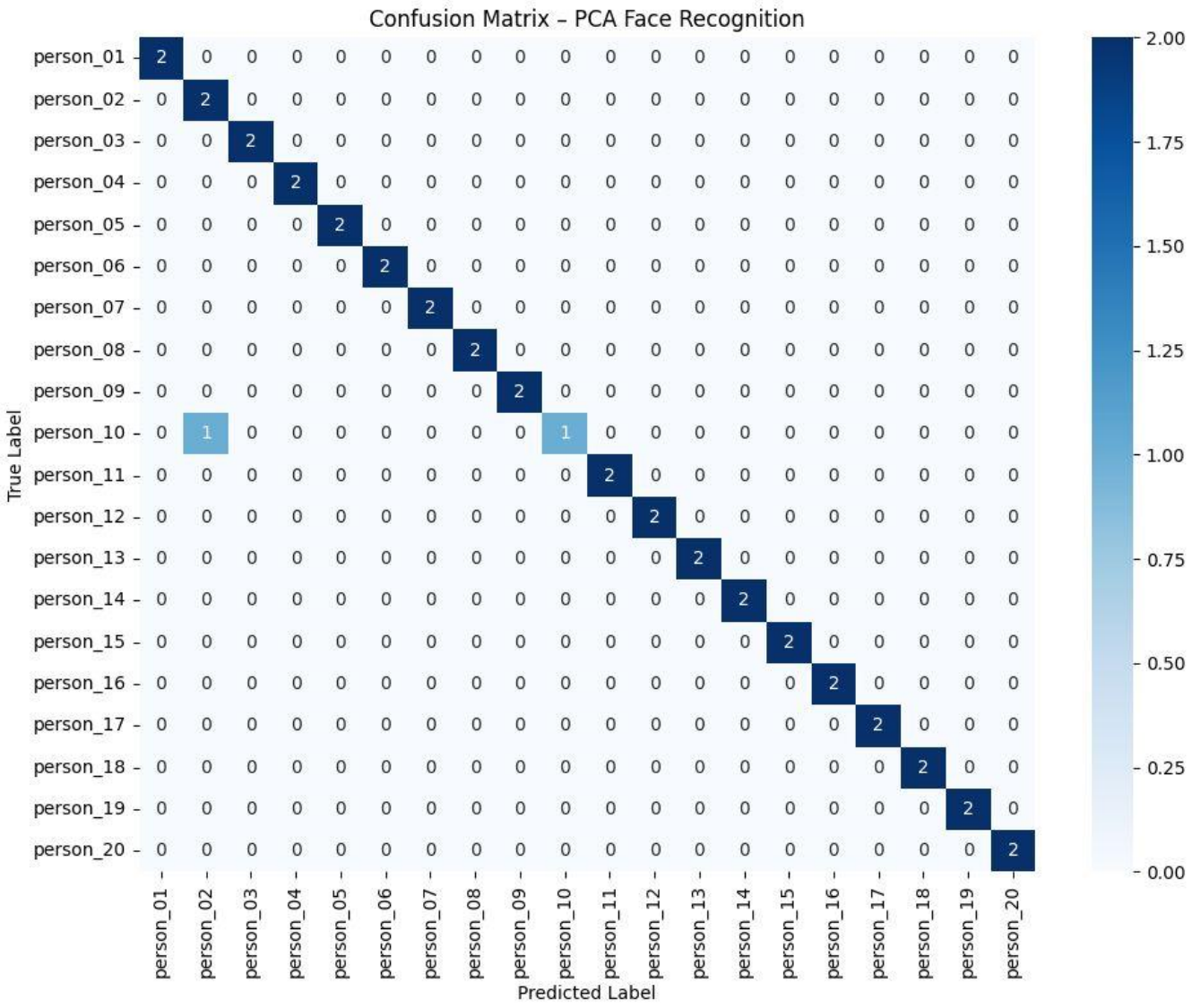
Performance Analysis

To evaluate the effectiveness of the face recognition model, we conducted a comprehensive performance analysis using several statistical metrics. These include accuracy, precision, recall, and the F1-score, which provide insight into the model's classification capabilities. In addition, we plotted the Receiver Operating Characteristic (ROC) curve and calculated the Area Under the Curve (AUC) to assess the model's ability to distinguish between genuine and impostor faces across various thresholds. These evaluation metrics collectively offer a robust understanding of the model's strengths and potential areas for improvement.

ROC Curve:



Confusion Matrix:



Performance Metrics

- Accuracy: 0.9750
- Precision: 0.9833
- Recall: 0.9750
- F1 Score: 0.9733
- Specificity: 0.9987
- AUC Score: 0.9231

Observation:

Our model might seem too good to be true, showing almost perfect performance, but that's because we are working on a small and finite data that consists of 20 subjects so we were able to train the model to this level of accuracy.