Web-based Toolkit for Face Detection & Recognition

Team 07

Faculty of Engineering - Cairo University

Dr. Ahmed Badawy

Eng. Omar Dawah & Eng. Laila Abbas

Collaborators

Name	Section	B.N
Amir Hesham Khalil	1	10
Omar Mohamed Ahmed	1	46
Ahmed Emad	1	3
Hazem Zakaria	1	16
Omar Tarek	1	42

Introduction

In today's digital landscape, the demand for advanced image processing technologies continues to surge, with applications ranging from security systems to personalized user experiences. Among these, face detection and recognition stand out as fundamental components powering a myriad of innovative solutions.

Our project delves into the realm of computer vision, focusing on the implementation of robust face detection and recognition capabilities. Leveraging state-of-the-art technologies and methodologies, we aim to develop a versatile and accurate system capable of detecting faces within images and recognizing individuals based on their facial features.

Face Detection

Face detection serves as the initial step in our endeavor. By employing sophisticated algorithms and deep learning models, our system can effectively identify and localize faces within images, regardless of variations in lighting conditions, facial expressions, or poses. This capability finds applications in various domains, including surveillance, human-computer interaction, and digital photography.

Face Recognition

Building upon the foundation of face detection, our project extends its capabilities to encompass face recognition. This process is built upon our ORL dataset which contains 36 different personal identities, each having 10 different image poses. Through the integration of machine learning

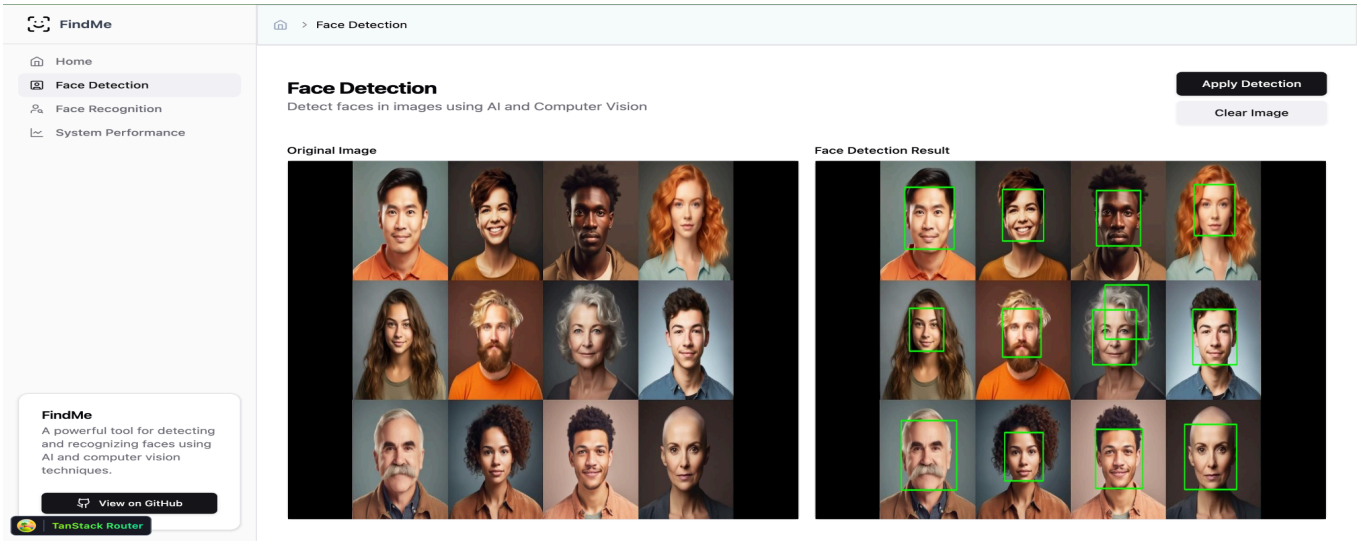
algorithms and facial feature extraction techniques, our system can discern unique facial characteristics and match them against our pre-existing dataset of known individuals.

The model used is **Support Vector Machine (SVM)** and is trained on 7 images out of 10 for each person, while leaving 3 images for evaluation and testing. This functionality opens doors to a multitude of applications, from access control systems to personalized user experiences in social media platforms and e-commerce.

Experiments and Results

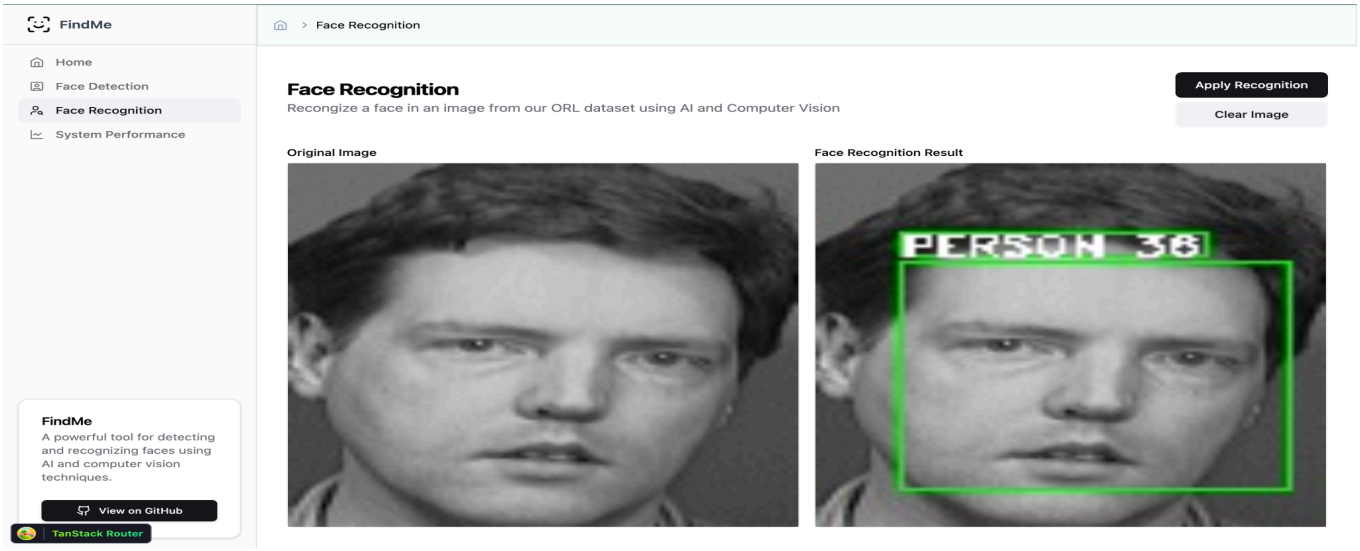
1. Face Detection

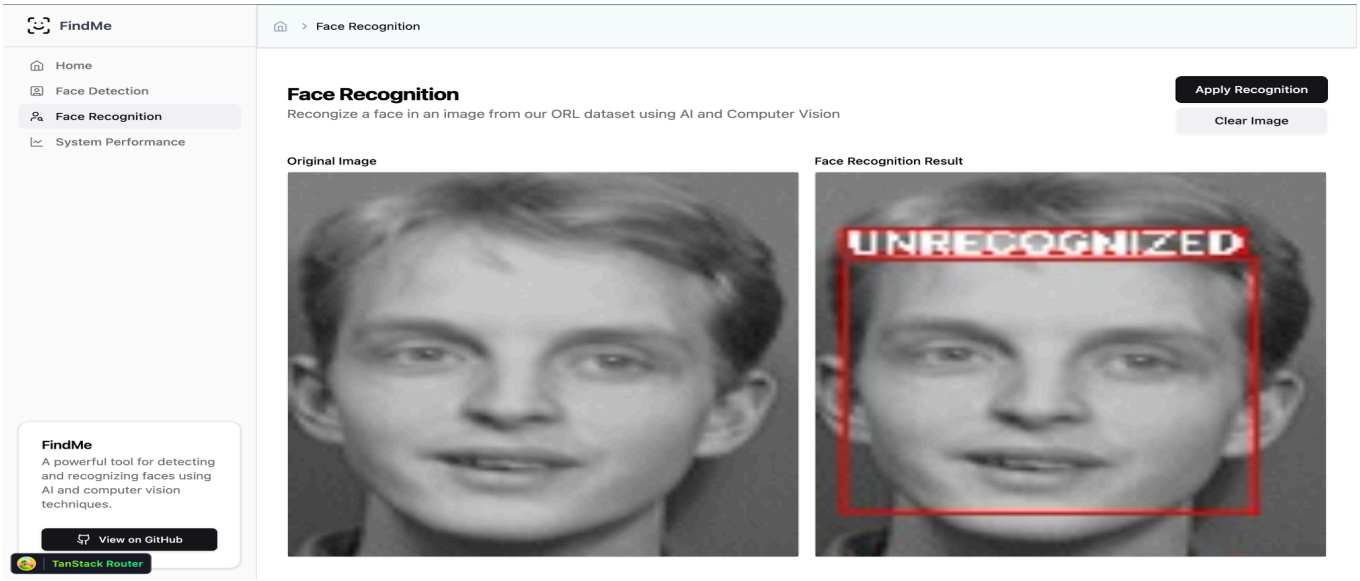
- Experiment: We conducted experiments to evaluate the face detection capability of our system. Various images with different compositions, lighting conditions, and angles were supplied to the system.
- Result: Our system successfully detected faces in all supplied images, demonstrating robustness across diverse scenarios. The detection results below are visually presented, highlighting the regions containing detected faces:



2. Face Recognition

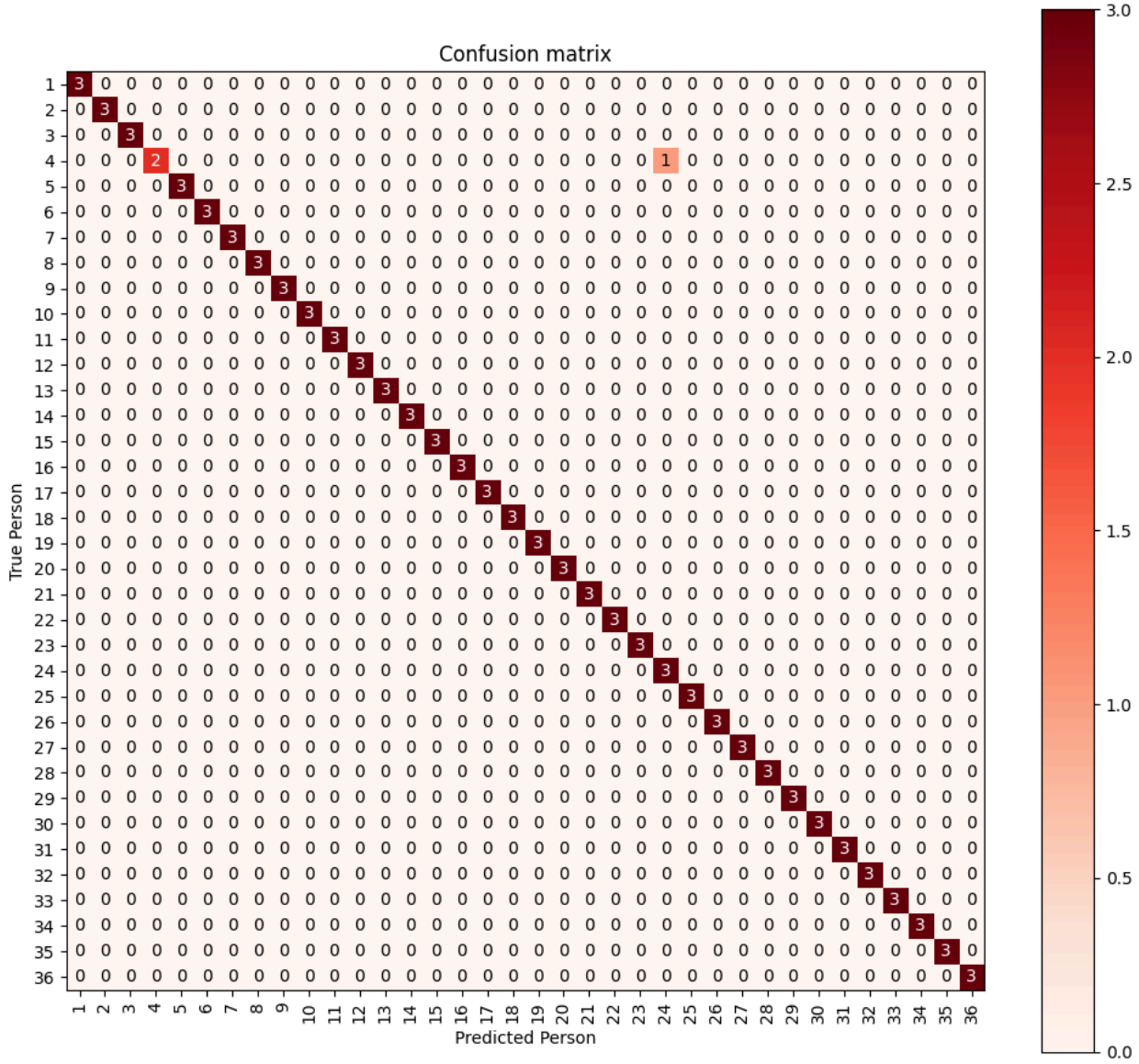
- Experiment: To assess the face recognition functionality, we supplied images of individuals from our dataset and individuals not present in the dataset.
- Result: The system accurately recognized individuals present in the dataset, providing a match with high confidence. For individuals not present in the dataset, the system appropriately indicated their absence, ensuring reliability in recognition tasks as below:



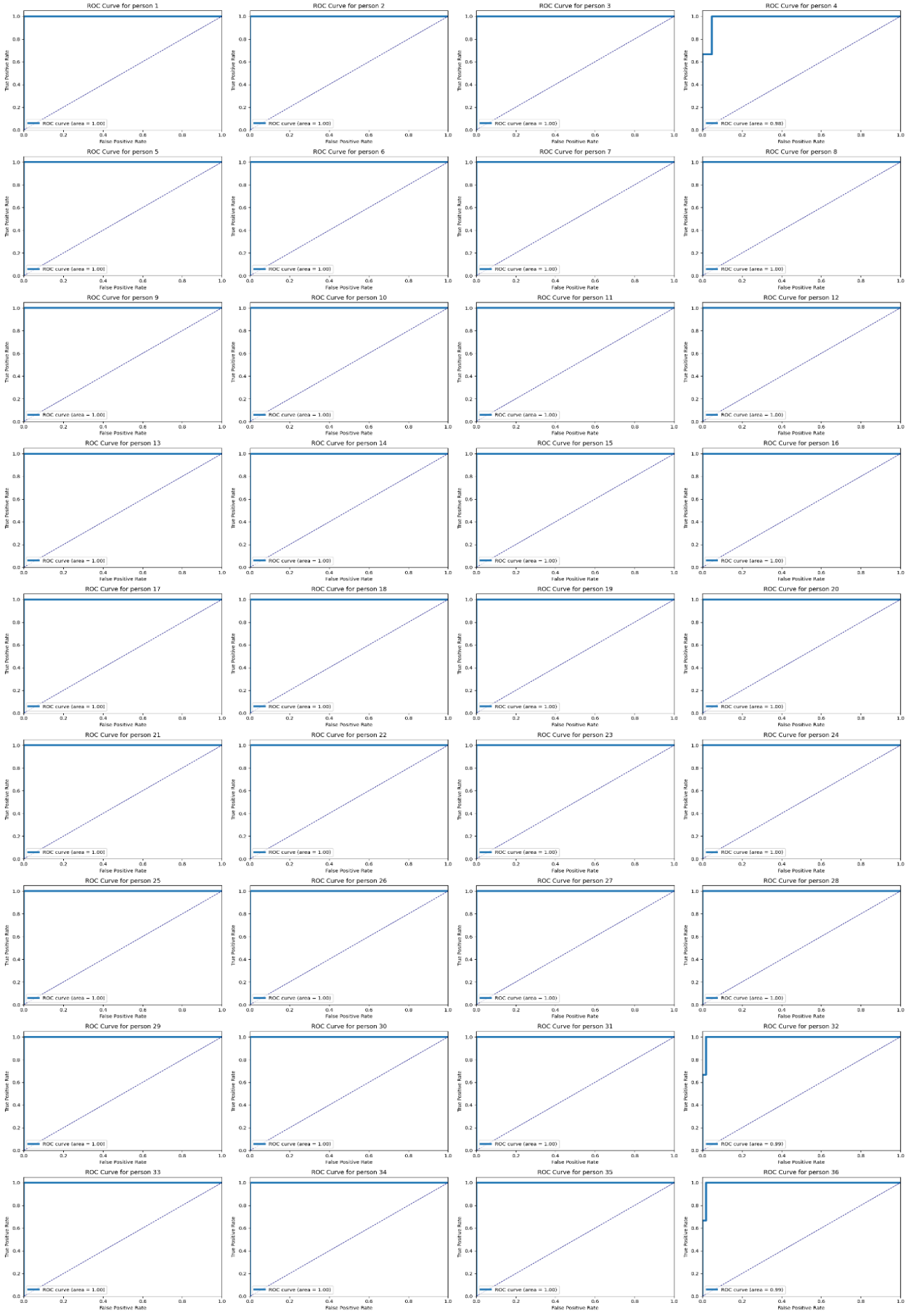


3. Performance and ROC Curve Analysis

- Experiment: We conducted receiver operating characteristic (ROC) curve analysis to evaluate the performance of our face recognition model across different threshold settings.
- Result: The ROC curve demonstrated the trade-off between true positive rate (sensitivity) and false positive rate (1 - specificity) at various threshold values. The area under the ROC curve (AUC) indicated the overall performance of the model, with higher AUC values signifying superior discriminative ability.



ROC curves for each person (class):



A. Software Design and Architecture

The application is structured around a robust software architecture, employing FastAPI for the backend, React for the frontend, and Tailwind CSS for styling. This architecture facilitates efficient organization and separation of concerns, akin to the Model-View-Controller (MVC) pattern, ensuring clear distinction between backend logic, frontend presentation, and styling concerns.

B. Used Tools and Technologies

- **FastAPI:** FastAPI serves as the backbone of our application, providing a high-performance web framework for building APIs with Python. Its asynchronous capabilities and automatic interactive documentation make it ideal for rapidly developing robust backend services.
- **React:** React is utilized for developing the frontend of our application, offering a declarative and component-based approach to building user interfaces. Its efficient rendering and state management capabilities enable us to create dynamic and interactive UIs for our users.
- **Tailwind CSS:** Tailwind CSS is employed for styling the frontend components of our application. Its utility-first approach and customizable design system allow for rapid UI development and easy maintenance.

C. Goals

- **Modularity**: The chosen architecture facilitates modular design, allowing for the independent development and testing of backend and frontend components.
- **Separation of Concerns**: By separating the backend logic (FastAPI) from the frontend presentation (React), we enhance maintainability and scalability while ensuring clear separation of concerns.
- **Ease of Development**: FastAPI, React, and Tailwind CSS collectively provide a conducive environment for rapid application development. This enables developers to focus on building and iterating on features rather than getting bogged down in low-level implementation details.
- Additionally, employing a modern frontend-backend separation enables us to leverage the strengths of each technology stack, resulting in a streamlined development process and a user-friendly application that meets the needs of our users.