|  |
| --- |
| **BỘ GIAO THÔNG VẬN TẢI**  **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **ĐỒ ÁN CHUYÊN NGÀNH**  **CÔNG NGHỆ THÔNG TIN**  **NHẬN DIỆN KHUÔN MẶT VỚI THUẬT TOÁN  LOCAL BINARY PATTERNS HISTOGRAMS (LBPH)**  **Giảng viên hướng dẫn: ThS. Nguyễn Thái Sơn, TS. Tô Bá Lâm**  **Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 01…………………………………**  **Mã số sinh viên:…………………………………………………………………….**  **Lớp: 0101000864xx…………………………………………………………………**  **TP.Hồ Chí Minh, tháng 12/2024** |
| **BỘ GIAO THÔNG VẬN TẢI**  **HỌC VIỆN HÀNG KHÔNG VIỆT NAM**  **KHOA CÔNG NGHỆ THÔNG TIN**      **ĐỒ ÁN CHUYÊN NGÀNH**  **CÔNG NGHỆ THÔNG TIN**  **NHẬN DIỆN KHUÔN MẶT VỚI THUẬT TOÁN  LOCAL BINARY PATTERNS HISTOGRAMS (LBPH)**  **Giảng viên hướng dẫn: ThS. Nguyễn Thái Sơn, TS. Tô Bá Lâm**  **Sinh viên/ Nhóm sinh viên thực hiện: Nhóm 01…………………………………**  **Mã số sinh viên:…………………………………………………………………….**  **Lớp: 0101000864xx…………………………………………………………………**  **Thành phố Hồ Chí Minh, tháng 12/2024** |

**Danh sách Nhóm:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Ghi chú** |
| 1 | Phạm Dương Hưng | 2254810122 |  | Nhóm Trưởng |
| 2 | Nguyễn Ngọc Bảo | 2254810286 | 22ĐHTT06 |  |
| 3 | Nguyễn Hải Đăng | 2254810272 | 22ĐHTT06 |  |
| 4 | Nguyễn Văn Hải | 2254810160 |  |  |
| 5 | Trịnh Thanh Minh | 2254810061 |  |  |

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1**  *(ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(ký và ghi rõ họ tên)* |
| **Cán bộ chấm thi phúc khảo 1**  *(ký và ghi rõ họ tên)* | **Cán bộ chấm thi phúc khảo 2**  *(ký và ghi rõ họ tên)* |

# DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Ký hiệu, chữ viết tắt** | **Chữ viết đầy đủ** |
| LBPH | Local Binary Patterns Histograms |
| GUI | Graphical User Interface |
| Tkinter | Tk interface |
| SQLite | Structured Query Language Lite |
| DB | Database |

# DANH MỤC HÌNH ẢNH

[Hình 2.1 - Thuật toán Local Binary Patterns Histograms 5](#_Toc183983036)

[Hình 2.2 – Các đặc trưng Haar 6](#_Toc183983037)

[Hình 2.3 – Visual Studio Code 7](#_Toc183983038)

[Hình 2.4 – Ngôn ngữ lập trình Python 8](#_Toc183983039)

[Hình 2.5 – Thư viện OpenCV 9](#_Toc183983040)

[Hình 2.6 – Cơ sở dữ liệu SQLite 10](#_Toc183983041)

[Hình 2.7 – Giao diện đồ họa với Tkinter 11](#_Toc183983042)

[Hình 3.1 – Biểu đồ Use Case tổng quát hệ thống 16](#_Toc183983043)

[Hình 3.2 – Biểu đồ hoạt động chức năng Chụp ảnh khuôn mặt 17](#_Toc183983044)

[Hình 3.3 – Biểu đồ hoạt động chức năng Huấn luyện mô hình nhận diện 17](#_Toc183983045)

[Hình 3.4 – Biểu đồ hoạt động chức năng nhận diện khuôn mặt từ video 18](#_Toc183983046)

[Hình 3.5 – Biểu đồ hoạt động chức năng Nhận diện khuôn mặt từ ảnh 18](#_Toc183983047)

[Hình 3.6 - Biểu đồ hoạt động chức năng Hiển thị kết quả 19](#_Toc183983048)

[Hình 3.7 – Biểu đồ trạng thái chức năng Chụp ảnh khuôn mặt 19](#_Toc183983049)

[Hình 3.8 - Biểu đồ trạng thái chức năng Huấn luyện mô hình nhận diện 20](#_Toc183983050)

[Hình 3.9 - Biểu đồ trạng thái chức năng Nhận diện khuôn mặt từ video 20](#_Toc183983051)

[Hình 3.10 - Biểu đồ trạng thái chức năng nhận diện khuôn mặt từ ảnh 21](#_Toc183983052)

[Hình 3.11 – Biểu đồ trạng thái chức năng Hiển thị kết quả 21](#_Toc183983053)

[Hình 3.12 – Giao diện chính của hệ thống 22](#_Toc183983054)

[Hình 3.13 – Giao diện thêm khuôn mặt mới 23](#_Toc183983055)

[Hình 3.14 - Giao diện thông báo huấn luyện mô hình thành công 24](#_Toc183983056)

[Hình 3.15 - Giao diện nhận diện khuôn mặt từ hình ảnh 24](#_Toc183983057)

[Hình 3.16 - Giao diện nhận diện khuôn mặt từ video 25](#_Toc183983058)

[Hình 3.17 - Chức năng thu thập dữ liệu khuôn mặt 26](#_Toc183983059)

[Hình 3.18 - Chức năng huấn luyện mô hình 27](#_Toc183983060)

[Hình 3.19 - Chức năng nhận diện khuôn mặt từ ảnh 28](#_Toc183983061)

[Hình 3.20 - Chức năng nhận diện khuôn mặt từ video 29](#_Toc183983062)

# DANH MỤC BẢNG BIỂU

[Bảng 3.1 – Mô tả chức năng Quản lý cơ sở dữ liệu đối với hệ thống 12](#_Toc183957171)

[Bảng 3.2 – Mô tả chức năng Quản lý cơ sở dữ liệu đối với người dùng 12](#_Toc183957172)

[Bảng 3.3 - Mô tả chức năng Chụp ảnh khuôn mặt đối với hệ thống 12](#_Toc183957173)

[Bảng 3.4 - Mô tả chức năng Chụp ảnh khuôn mặt đối với người dùng 13](#_Toc183957174)

[Bảng 3.5 - Mô tả chức năng Huấn luyện mô hình nhận diện đối với hệ thống 13](#_Toc183957175)

[Bảng 3.6 - Mô tả chức năng Huấn luyện mô hình nhận diện đối với người dùng 13](#_Toc183957176)

[Bảng 3.7 - Mô tả chức năng Nhận diện khuôn mặt từ Video đối với hệ thống 14](#_Toc183957177)

[Bảng 3.8 - Mô tả chức năng Nhận diện khuôn mặt từ Video đối với người dùng 14](#_Toc183957178)

[Bảng 3.9 - Mô tả chức năng Nhận diện khuôn mặt từ Ảnh đối với hệ thống 14](#_Toc183957179)

[Bảng 3.10 - Mô tả chức năng Nhận diện khuôn mặt từ Ảnh đối với người dùng 15](#_Toc183957180)

[Bảng 3.11 - Mô tả chức năng Hiển thị kết quả đối với hệ thống 15](#_Toc183957181)

[Bảng 3.12 - Mô tả chức năng Hiển thị kết quả đối với người dùng 15](#_Toc183957182)

MỤC LỤC

*Trang*

[**DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT iv**](#_Toc183982996)

[**DANH MỤC HÌNH ẢNH v**](#_Toc183982997)

[**DANH MỤC BẢNG BIỂU vi**](#_Toc183982998)

[**MỞ ĐẦU ix**](#_Toc183982999)

[**CHƯƠNG 1. GIỚI THIỆU 1**](#_Toc183983000)

[1.1. Lý do chọn đề tài 1](#_Toc183983001)

[1.2. Mục tiêu đề tài 1](#_Toc183983002)

[1.3. Phạm vi đề tài 1](#_Toc183983003)

[1.4 Đối tượng nghiên cứu 2](#_Toc183983004)

[1.5. Phương pháp nghiên cứu 2](#_Toc183983005)

[1.6. Bố cục đề tài 2](#_Toc183983006)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 3**](#_Toc183983007)

[2.1. Nhận diện khuôn mặt (Face Recognition) 3](#_Toc183983008)

[2.2. Thuật toán Local Binary Patterns Histograms (LBPH) 4](#_Toc183983009)

[2.3. Haar Cascade Classifier 6](#_Toc183983010)

[2.4. Visual Studio Code 7](#_Toc183983011)

[2.5. Ngôn ngữ lập trình Python 8](#_Toc183983012)

[2.6. Thư viện OpenCV 9](#_Toc183983013)

[2.7. Cơ sở dữ liệu SQLite 10](#_Toc183983014)

[2.8. Giao diện đồ họa với Tkinter 11](#_Toc183983015)

[**CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM 12**](#_Toc183983016)

[3.1. Phân tích hệ thống 12](#_Toc183983017)

[3.1.1. Yêu cầu chức năng 12](#_Toc183983018)

[3.1.2. Yêu cầu phi chức năng 15](#_Toc183983019)

[3.1.3. Biểu đồ Use Case 16](#_Toc183983020)

[3.1.4. Biểu đồ hoạt động (Activity Diagram) 17](#_Toc183983021)

[3.1.5. Biểu đồ trạng thái (State Diagram) 19](#_Toc183983022)

[3.2. Xây dựng giao diện sản phẩm 22](#_Toc183983023)

[3.2.1. Giao diện chính của hệ thống 22](#_Toc183983024)

[3.2.3. Giao diện thêm khuôn mặt mới 23](#_Toc183983025)

[3.2.4. Giao diện thông báo huấn luyện mô hình thành công 23](#_Toc183983026)

[3.2.5. Giao diện nhận diện khuôn mặt từ hình ảnh 24](#_Toc183983027)

[3.2.6. Giao diện nhận diện khuôn mặt từ video 25](#_Toc183983028)

[3.3. Các chức năng chính của hệ thống 26](#_Toc183983029)

[3.3.1. Chức năng thu thập dữ liệu khuôn mặt 26](#_Toc183983030)

[3.3.2. Chức năng huấn luyện mô hình 27](#_Toc183983031)

[3.3.3. Chức năng nhận diện khuôn mặt từ ảnh 28](#_Toc183983032)

[3.3.4. Chức năng nhận diện khuôn mặt từ video 29](#_Toc183983033)

[**KẾT LUẬN 30**](#_Toc183983034)

[**DANH MỤC TÀI LIỆU THAM KHẢO 31**](#_Toc183983035)

# MỞ ĐẦU

Trong kỷ nguyên công nghệ số hiện đại, sự phát triển nhanh chóng của trí tuệ nhân tạo và xử lý ảnh đã mở ra những chân trời mới trong lĩnh vực nhận diện con người. Trong số các công nghệ tiên tiến, nhận diện khuôn mặt được xem là một trong những giải pháp quan trọng nhất, có khả năng ứng dụng rộng rãi và mang lại những giá trị thực tiễn to lớn.

Từ các hệ thống an ninh phức tạp cho đến những trải nghiệm người dùng tiện lợi, công nghệ nhận diện khuôn mặt đang dần chiếm lĩnh vai trò then chốt trong nhiều lĩnh vực như an ninh quốc gia, quản lý doanh nghiệp, giáo dục, chăm sóc sức khỏe và thậm chí là trải nghiệm số cá nhân. Sự chính xác, nhanh chóng và phi tiếp xúc của công nghệ này đã và đang trở thành một lợi thế cạnh tranh quan trọng cho các tổ chức và doanh nghiệp.

Trong bối cảnh đó, việc nghiên cứu và phát triển các thuật toán nhận diện khuôn mặt hiệu quả trở nên cực kỳ quan trọng. Mỗi thuật toán mới ra đời không chỉ là một tiến bộ công nghệ, mà còn là một bước đột phá trong việc giải quyết những thách thức về độ chính xác, tốc độ xử lý và khả năng thích ứng với môi trường thay đổi.

Đề tài “Nhận diện khuôn mặt với thuật toán Local Binary Patterns Histograms (LBPH)” được chúng em chọn với mong muốn góp phần nghiên cứu và ứng dụng một trong những phương pháp tiên tiến trong lĩnh vực xử lý ảnh và trí tuệ nhân tạo. Chúng em tin rằng, thông qua việc nghiên cứu chuyên sâu và triển khai thực tiễn, chúng em sẽ mang đến một góc nhìn mới và những đóng góp ý nghĩa cho cộng đồng nghiên cứu và phát triển công nghệ

# CHƯƠNG 1. GIỚI THIỆU

## 1.1. Lý do chọn đề tài

Trong xu thế phát triển công nghệ số hiện nay, nhu cầu về các hệ thống nhận diện an toàn, chính xác và hiệu quả đang trở nên cấp thiết hơn bao giờ hết. Các tổ chức, doanh nghiệp và cá nhân ngày càng đòi hỏi những giải pháp công nghệ không chỉ đơn thuần là nhận diện, mà còn phải đảm bảo tính bảo mật, nhanh chóng và thích ứng với nhiều môi trường khác nhau.

Xuất phát từ thực tế đó, nhóm chúng em nhận thấy tiềm năng to lớn của thuật toán Local Binary Patterns Histograms (LBPH) trong việc giải quyết những thách thức trên. Khác với các phương pháp truyền thống, LBPH mang đến ưu điểm vượt trội về khả năng xử lý ảnh trong điều kiện ánh sáng thay đổi, góc chụp đa dạng và chi phí tính toán thấp. Chính những ưu điểm này đã thuyết phục chúng em lựa chọn đây là trọng tâm nghiên cứu.

## 1.2. Mục tiêu đề tài

Mục tiêu tổng thể của đề tài nhằm xây dựng một hệ thống nhận diện khuôn mặt toàn diện, không chỉ là một công cụ công nghệ đơn thuần mà còn là một giải pháp ứng dụng thực tiễn. Chúng em hướng đến việc phát triển một hệ thống có thể tích hợp linh hoạt, dễ dàng mở rộng và đáp ứng được các yêu cầu khắt khe về độ chính xác và tốc độ xử lý.

Để hiện thực hóa mục tiêu này, chúng em đặt ra những nhiệm vụ cụ thể như: nghiên cứu chuyên sâu thuật toán LBPH, xây dựng một môi trường thử nghiệm chuyên nghiệp, phát triển giao diện người dùng trực quan và đánh giá toàn diện hiệu năng của hệ thống thông qua các bộ dữ liệu kiểm thử.

## 1.3. Phạm vi đề tài

Để đảm bảo tính khả thi và trọng tâm, việc xác định phạm vi nghiên cứu một cách rõ ràng và chi tiết là cần thiết. Về không gian nghiên cứu, đề tài tập trung tại các phòng thí nghiệm với môi trường kiểm soát, cho phép chúng em đánh giá chính xác nhất hiệu quả của thuật toán.

Số lượng khuôn mặt được giới hạn trong phạm vi 10 khuôn mặt, đủ để kiểm chứng độ tin cậy nhưng không quá phức tạp để phân tích. Các công nghệ được lựa chọn như OpenCV, Python, Tkinter và SQLite đều là những công nghệ mã nguồn mở, hiện đại và phù hợp với mục tiêu nghiên cứu.

## 1.4 Đối tượng nghiên cứu

Đối tượng nghiên không chỉ đơn thuần là một thuật toán hay một hệ thống công nghệ, mà còn là sự tích hợp đa chiều giữa các công nghệ tiên tiến. Chúng em tập trung nghiên cứu sâu về thuật toán LBPH, phương pháp phát hiện khuôn mặt Haar Cascade và khả năng vận hành tổng thể của một hệ thống nhận diện khuôn mặt hiện đại

## 1.5. Phương pháp nghiên cứu

* Phương pháp thu thập thông tin: Nghiên cứu tài liệu chuyên ngành, tham khảo các công trình khoa học liên quan
* Phương pháp phân tích: Phân tích so sánh các thuật toán nhận diện khuôn mặt
* Phương pháp thực nghiệm: Xây dựng hệ thống thử nghiệm, đánh giá độ chính xác thông qua các bộ dữ liệu kiểm thử
* Phương pháp đánh giá: Sử dụng các độ đo như độ chính xác, tốc độ xử lý, khả năng chống nhiễu

## 1.6. Bố cục đề tài

Phần còn lại của báo cáo đồ án chuyên ngành được tổ chức như sau:

Chương 2 trình bày các cơ sở lý thuyết nền tảng, bao gồm tổng quan về nhận diện khuôn mặt, chi tiết thuật toán LBPH, Haar Cascade, và các công nghệ hỗ trợ như OpenCV, SQLite và Tkinter.

Chương 3 tập trung vào phân tích hệ thống và chi tiết quá trình xây dựng sản phẩm, từ việc thiết kế giao diện đến triển khai các chức năng chính của hệ thống nhận diện khuôn mặt.

Phần kết luận sẽ tổng kết những kết quả đạt được, những hạn chế của đề tài và đề xuất hướng phát triển trong tương lai.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Trong lĩnh vực nhận diện khuôn mặt, việc lựa chọn các công nghệ và phương pháp phù hợp đóng vai trò quyết định đến hiệu quả và độ chính xác của hệ thống. Đề tài “Nhận diện khuôn mặt với thuật toán Local Binary Patterns Histograms (LBPH)” được xây dựng dựa trên nhiều cơ sở lý thuyết và công nghệ quan trọng.

Thuật toán LBPH được lựa chọn làm nền tảng chính cho hệ thống vì khả năng thích ứng tốt với các điều kiện ánh sáng khác nhau và độ chính xác cao trong việc nhận diện khuôn mặt. Để phát hiện khuôn mặt trong ảnh, chúng em sử dụng Haar Cascade Classifier – một phương pháp đã được kiểm chứng và widely adopted trong cộng đồng computer vision.

OpenCV, một thư viện mã nguồn mở về xử lý ảnh và computer vision, được chọn làm công cụ chính để triển khai các thuật toán trên. Thư viện này cung cấp các implementation hiệu quả của cả LBPH và Haar Cascade, cùng với nhiều công cụ xử lý ảnh hữu ích khác.

Để xây dựng giao diện người dùng thân thiện, chúng em sử dụng Tkinter – thư viện GUI tiêu chuẩn của Python. Tkinter được chọn vì tính đơn giản, dễ sử dụng và tích hợp sẵn trong Python. Về phần lưu trữ dữ liệu, SQLite được lựa chọn làm hệ quản trị cơ sở dữ liệu vì tính nhẹ, không cần server riêng và phù hợp với quy mô của ứng dụng.

Các phần tiếp theo sẽ trình bày chi tiết về từng công nghệ và lý thuyết nền tảng được sử dụng trong đề tài.

## 2.1. Nhận diện khuôn mặt (Face Recognition)

Nhận diện khuôn mặt là một lĩnh vực nghiên cứu quan trọng trong trí tuệ nhân tạo và xử lý ảnh, nhằm mục đích tự động xác định và xác thực danh tính của một cá nhân dựa trên các đặc điểm khuôn mặt của họ.

Nhận diện khuôn mặt là quá trình nhận dạng hoặc xác minh danh tính của một cá nhân thông qua việc so sánh các đặc trưng khuôn mặt được chụp với cơ sở dữ liệu hiện có.

Công nghệ này đã phát triển mạnh mẽ trong những thập kỷ gần đây, từ các phương pháp thô sơ đến những giải thuật machine learning tiên tiến.

Các bước cơ bản trong nhận diện khuôn mặt:

1. Phát hiện khuôn mặt (Face Detection)

* Xác định vị trí khuôn mặt trong ảnh
* Tách biệt khuôn mặt khỏi nền

1. Tiền xử lý ảnh

* Điều chỉnh độ sáng, tỷ lệ, góc độ
* Chuẩn hóa khuôn mặt

1. Trích xuất đặc trưng (Feature Extraction)

* Xác định các điểm đặc trưng riêng biệt của khuôn mặt
* Chuyển đổi hình ảnh thành vector đặc trưng

1. Nhận diện/So sánh (Recognition/Matching)

* So sánh vector đặc trưng với cơ sở dữ liệu
* Xác định độ tương đồng

Nhận diện khuôn mặt có nhiều ứng dụng quan trọng như trong an ninh và giám sát, mở khóa điện tử thông minh, thanh toán không tiếp xúc, quản lý nhân sự, chăm sóc y tế, và cải thiện trải nghiệm người dùng số.

Tuy nhiên, công nghệ này cũng gặp phải một số thách thức, như sự thay đổi ánh sáng, góc độ chụp đa dạng, thay đổi biểu cảm khuôn mặt, khuôn mặt bị che khuất, và các vấn đề về bảo mật và quyền riêng tư.

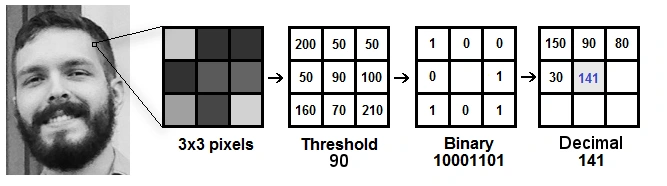
## 2.2. Thuật toán Local Binary Patterns Histograms (LBPH)

LBPH (Local Binary Patterns Histograms) được phát triển như một phương pháp thống kê để mô tả các đặc trưng khuôn mặt. Ban đầu, nghiên cứu về Local Binary Patterns (LBP) được giới thiệu vào năm 2002, với mục đích tạo ra một phương pháp đơn giản và hiệu quả để nhận dạng và phân tích các đặc điểm hình ảnh. Sau đó, LBPH được cải tiến để tăng khả năng nhận diện khuôn mặt trong nhiều điều kiện ánh sáng khác nhau, làm cho phương pháp này trở nên mạnh mẽ hơn trong các ứng dụng thực tế.

LBPH hoạt động dựa trên việc phân tích cường độ ánh sáng giữa các pixel lân cận trong một vùng nhỏ của ảnh (local region). Từng pixel được mã hóa thành một giá trị nhị phân dựa trên sự so sánh với pixel trung tâm, từ đó hình thành một mẫu nhị phân (binary pattern). Các mẫu này được tổng hợp thành biểu đồ tần suất, đại diện cho đặc trưng của khuôn mặt.

Các bước thực hiện:

* Chia ảnh khuôn mặt thành các ô nhỏ (cells).
* Tính toán LBP cho từng pixel trong mỗi ô.
* Tạo biểu đồ tần suất cho từng ô.
* Kết hợp các biểu đồ để tạo nên biểu đồ tổng hợp cho toàn bộ khuôn mặt.
* So sánh biểu đồ của ảnh đầu vào với các biểu đồ đã lưu để nhận diện.



Hình 2. - Thuật toán Local Binary Patterns Histograms

Ưu điểm:

* Khả năng chịu nhiễu cao
* Hiệu quả với các thay đổi về ánh sáng
* Chi phí tính toán thấp
* Dễ dàng triển khai

Nhược điểm:

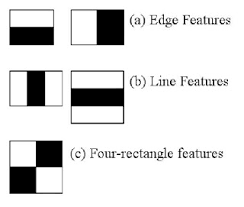
* Độ chính xác giảm với các biến dạng lớn
* Hiệu quả hạn chế với các ảnh chất lượng thấp
* Khó xử lý với các góc chụp quá khác biệt

Khi so sánh với các thuật toán nhận diện khuôn mặt khác, Eigenfaces kém hiệu quả hơn LBPH trong điều kiện ánh sáng thay đổi. Fisherfaces có độ chính xác tương đương LBPH nhưng phức tạp hơn. Trong khi đó, Deep Learning mang lại hiệu quả vượt trội nhưng đòi hỏi chi phí tính toán cao và tài nguyên phần cứng mạnh mẽ.

## 2.3. Haar Cascade Classifier

Haar Cascade Classifier là một trong những phương pháp phát hiện đối tượng tiên phong trong lĩnh vực thị giác máy tính, được Paul Viola và Michael Jones giới thiệu vào năm 2001. Đây là một kỹ thuật máy học phi sâu (shallow learning) được thiết kế để phát hiện nhanh chóng và chính xác các đối tượng trong ảnh, đặc biệt là khuôn mặt. Phương pháp này sử dụng các đặc trưng Haar - những đặc trưng hình học đơn giản được so sánh với các mẫu đã được huấn luyện trước.

Thuật toán hoạt động bằng cách quét ảnh với một cửa sổ trượt và sử dụng một loạt các bộ phân lớp (classifier) được huấn luyện từ trước. Mỗi bộ phân lớp được xây dựng dựa trên các đặc trưng Haar - các đặc trưng đơn giản được tạo ra bằng cách so sánh mức độ sáng giữa các vùng kế cận trong ảnh. Quá trình này diễn ra theo nhiều giai đoạn, mỗi giai đoạn loại bỏ dần các vùng không phải là đối tượng cần tìm.



Hình 2. – Các đặc trưng Haar

Ưu điểm:

* Tốc độ xử lý nhanh
* Độ chính xác tương đối cao với khuôn mặt thẳng
* Yêu cầu tài nguyên máy tính thấp
* Dễ dàng triển khai

Nhược điểm:

* Kém hiệu quả với các góc chụp không chính diện
* Độ chính xác giảm trong điều kiện ánh sáng phức tạp
* Khả năng phát hiện hạn chế với các biến dạng lớn

## 2.4. Visual Studio Code

Visual Studio Code (VS Code) là trình soạn thảo mã nguồn mở của Microsoft, ra mắt năm 2015 và phát hành mã nguồn dưới giấy phép MIT vào năm 2016. Nhờ tính linh hoạt, hỗ trợ nhiều ngôn ngữ lập trình và khả năng mở rộng qua extension, VS Code nhanh chóng được ưa chuộng. Đây là công cụ đa nền tảng, miễn phí, cung cấp các tính năng như gợi ý mã, gỡ lỗi và quản lý môi trường ảo. VS Code được sử dụng rộng rãi trong phát triển web, ứng dụng di động, backend, game, khoa học dữ liệu và DevOps.



**Hình 2.3 – Visual Studio Code**

Ưu điểm:

* Tính đa nền tảng và miễn phí
* Hỗ trợ mạnh mẽ cho nhiều ngôn ngữ lập trình
* Hệ thống extension phong phú
* Khả năng tích hợp cao với các công cụ phát triển

Nhược điểm:

* Tiêu thụ nhiều tài nguyên hệ thống
* Hiệu suất có thể giảm với các dự án quy mô lớn

## 2.5. Ngôn ngữ lập trình Python

Python được tạo ra bởi Guido van Rossum vào cuối những năm 1980 và ra mắt công chúng vào năm 1991. Ngôn ngữ này nổi tiếng với cú pháp đơn giản, dễ học và khả năng ứng dụng đa dạng. Phiên bản Python 2.0 ra mắt năm 2000 và Python 3.0 năm 2008 đã mang đến những cải tiến quan trọng, giúp Python phát triển mạnh mẽ như ngày nay.

Python được ứng dụng rộng rãi trong phát triển web (Django, Flask), khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa, phát triển game (Pygame), ứng dụng desktop, và phân tích tài chính. Sự dễ sử dụng và thư viện phong phú giúp Python trở thành ngôn ngữ lập trình phổ biến trong cả giáo dục lẫn công nghiệp.

* 

**Hình 2.4 – Ngôn ngữ lập trình Python**

Ưu điểm:

* Cú pháp đơn giản, dễ đọc và dễ học
* Hệ sinh thái phong phú với nhiều thư viện
* Tính linh hoạt và ứng dụng rộng rãi
* Cộng đồng lập trình viên lớn mạnh

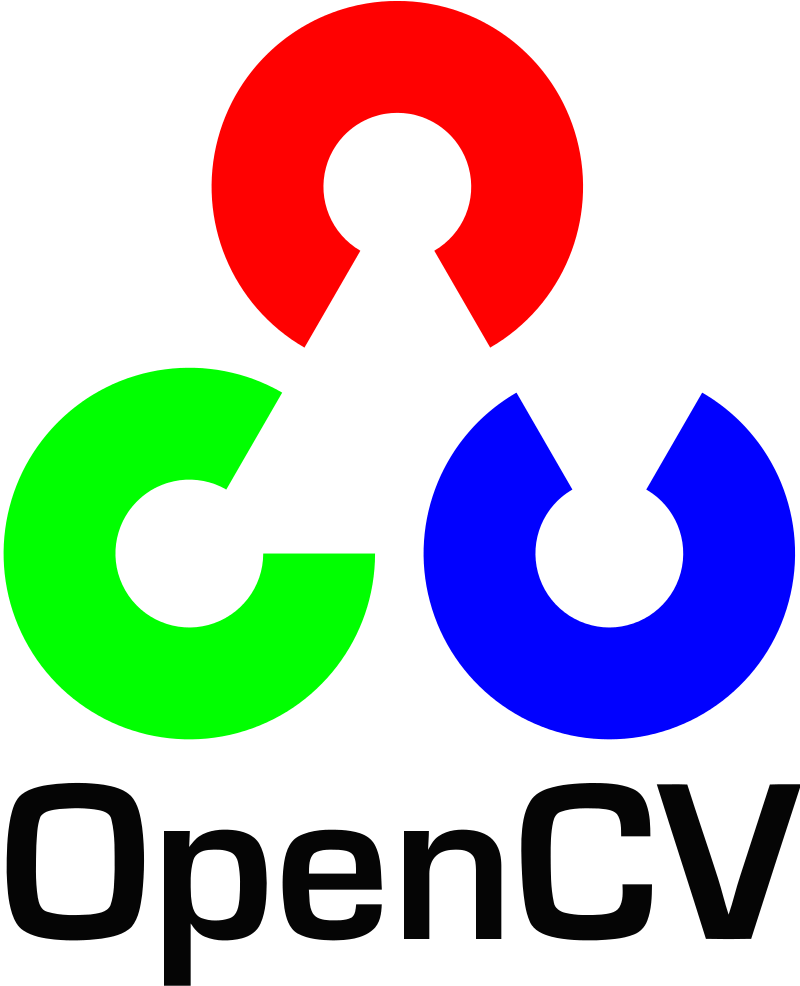
Nhược điểm:

* Tốc độ thực thi chậm hơn so với các ngôn ngữ biên dịch như C++
* Khó quản lý phiên bản và môi trường
* Hiệu năng kém trong các ứng dụng đòi hỏi xử lý song song

## 2.6. Thư viện OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được phát triển bởi Intel vào năm 1999, với mục tiêu cung cấp một nền tảng chung cho các ứng dụng thị giác máy tính và xử lý ảnh. Được viết bằng C/C++ nhưng hỗ trợ đa ngôn ngữ, OpenCV đã trở thành một trong những công cụ quan trọng nhất trong lĩnh vực xử lý ảnh và trí tuệ nhân tạo. Thư viện này cung cấp hơn 2500 thuật toán tối ưu, bao gồm các chức năng từ xử lý ảnh cơ bản đến các kỹ thuật machine learning phức tạp.

OpenCV hoạt động như một bộ công cụ toàn diện, cung cấp các hàm và lớp để thực hiện các thao tác xử lý ảnh và video. Thư viện sử dụng kiến trúc module hóa, cho phép người dùng dễ dàng tích hợp và mở rộng. Các module chính bao gồm xử lý ảnh, phát hiện đối tượng, học máy, và xử lý video.



Hình 2. – Thư viện OpenCV

Ưu điểm:

* Miễn phí và mã nguồn mở
* Hiệu suất cao
* Hỗ trợ đa nền tảng
* Cộng đồng phát triển lớn

Nhược điểm:

* Tài liệu đôi khi thiếu chi tiết
* Đòi hỏi kiến thức chuyên sâu về xử lý ảnh
* Một số chức năng phức tạp khó sử dụng

## 2.7. Cơ sở dữ liệu SQLite

SQLite ra đời năm 2000 do D. Richard Hipp phát triển, là một hệ quản trị cơ sở dữ liệu quan hệ nhúng (embedded relational database management system). Khác với các hệ quản trị cơ sở dữ liệu truyền thống, SQLite không yêu cầu máy chủ riêng biệt và toàn bộ cơ sở dữ liệu được lưu trữ trong một file duy nhất. Đây là giải pháp lý tưởng cho các ứng dụng có quy mô nhỏ và vừa, đòi hỏi tính di động và hiệu quả cao.

SQLite hoạt động trực tiếp trên file, không cần quy trình máy chủ riêng. Toàn bộ cơ sở dữ liệu được lưu trữ trong một file duy nhất trên hệ thống tệp, cho phép truy cập nhanh chóng và hiệu quả. Thư viện được viết bằng C, có thể được nhúng trực tiếp vào ứng dụng.



Hình 2. – Cơ sở dữ liệu SQLite

Ưu điểm:

* Nhẹ và không cần cài đặt máy chủ
* Tốc độ truy vấn nhanh
* Độ tin cậy cao
* Dễ dàng di chuyển và sao chép

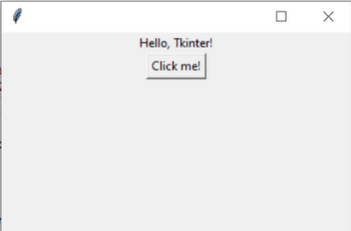
Nhược điểm:

* Không phù hợp với hệ thống lớn
* Hạn chế về đa người dùng
* Thiếu một số tính năng của hệ quản trị cơ sở dữ liệu lớn

## 2.8. Giao diện đồ họa với Tkinter

Tkinter là thư viện giao diện đồ họa (GUI) tiêu chuẩn của Python, được phát triển dựa trên giao diện Tk GUI toolkit. Ra đời từ những năm 1990, Tkinter là lựa chọn mặc định để xây dựng các ứng dụng có giao diện người dùng trong Python. Được thiết kế để đơn giản và dễ sử dụng, Tkinter cung cấp một bộ các widget (thành phần giao diện) cơ bản để xây dựng các ứng dụng đa nền tảng.

Tkinter hoạt động bằng cách tạo ra các cửa sổ và widget dựa trên các lớp và phương thức được định nghĩa sẵn. Nó sử dụng mô hình lập trình hướng đối tượng, cho phép người dùng tạo ra các giao diện phức tạp từ các thành phần đơn giản. Mỗi widget được quản lý bởi trình quản lý geometry, điều khiển việc bố trí và hiển thị các thành phần.



Hình 2. – Giao diện đồ họa với Tkinter

Ưu điểm:

* Tích hợp sẵn trong Python
* Dễ học và sử dụng
* Đa nền tảng
* Khả năng tùy chỉnh linh hoạt

Nhược điểm:

* Giao diện đơn giản, ít hiện đại
* Hiệu năng hạn chế với các ứng dụng phức tạp
* Số lượng widget có hạn
* Khả năng mở rộng kém so với các thư viện GUI hiện đại

# CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG VÀ XÂY DỰNG SẢN PHẨM

## 3.1. Phân tích hệ thống

### 3.1.1. Yêu cầu chức năng

Hệ thống nhận diện khuôn mặt sẽ bao gồm các yêu cầu chức năng chính như sau:

* **Chức năng Quản lý cơ sở dữ liệu**

**Đối với hệ thống:**

Bảng 3. – Mô tả chức năng Quản lý cơ sở dữ liệu đối với hệ thống

|  |  |
| --- | --- |
| Mô tả | Hệ thống cho phép lưu trữ và quản lý thông tin người dùng |
| Đầu vào | Nhận thông tin ID và tên người dùng. |
| Quá trình xử lý | Lưu trữ thông tin người dùng vào cơ sở dữ liệu. |
| Kết quả | Cập nhật cơ sở dữ liệu với thông tin mới. |

**Đối với người dùng**

Bảng 3. – Mô tả chức năng Quản lý cơ sở dữ liệu đối với người dùng

|  |  |
| --- | --- |
| Mô tả | Người dùng có thể thêm, cập nhật và xóa thông tin cá nhân của mình |
| Đầu vào | Nhập tên khi thêm người dùng |
| Kết quả | Nhận thông báo xác nhận khi thêm hoặc cập nhật thành công |

* **Chức năng Chụp ảnh khuôn mặt**

**Đối với hệ thống:**

Bảng 3. - Mô tả chức năng Chụp ảnh khuôn mặt đối với hệ thống

|  |  |
| --- | --- |
| Mô tả | Hệ thống tự động chụp và lưu trữ ảnh khuôn mặt của người dùng |
| Quá trình xử lý | Tự động chụp 200 ảnh khuôn mặt từ nhiều góc độ khác nhau |
| Kết quả | Lưu trữ các ảnh khuôn mặt vào thư mục riêng trong cơ sở dữ liệu |

**Đối với người dùng**

Bảng 3. - Mô tả chức năng Chụp ảnh khuôn mặt đối với người dùng

|  |  |
| --- | --- |
| Mô tả | Người dùng có thể bắt đầu quá trình chụp ảnh khuôn mặt. |
| Đầu vào | Nhấn nút "Add/Update" để bắt đầu quá trình chụp ảnh |
| Kết quả | Nhận thông báo khi quá trình chụp ảnh hoàn tất. |

* **Chức năng Huấn luyện mô hình nhận diện**

**Đối với hệ thống:**

Bảng 3. - Mô tả chức năng Huấn luyện mô hình nhận diện đối với hệ thống

|  |  |
| --- | --- |
| Mô tả | Hệ thống huấn luyện mô hình nhận diện khuôn mặt từ các bức ảnh đã lưu |
| Quá trình xử lý | Huấn luyện mô hình từ các ảnh đã lưu trong cơ sở dữ liệu |
| Kết quả | Lưu trữ mô hình đã huấn luyện để sử dụng trong tương lai |

**Đối với người dùng**

Bảng 3. - Mô tả chức năng Huấn luyện mô hình nhận diện đối với người dùng

|  |  |
| --- | --- |
| Mô tả | Người dùng có thể yêu cầu hệ thống huấn luyện mô hình nhận diện |
| Đầu vào | Nhấn nút "Train Model" để bắt đầu huấn luyện. |
| Kết quả | Nhận thông báo khi quá trình huấn luyện hoàn tất. |

* **Chức năng Nhận diện khuôn mặt từ Video**

**Đối với hệ thống:**

Bảng 3. - Mô tả chức năng Nhận diện khuôn mặt từ Video đối với hệ thống

|  |  |
| --- | --- |
| Mô tả | Hệ thống nhận diện khuôn mặt trong thời gian thực từ camera |
| Quá trình xử lý | Phát hiện và nhận diện khuôn mặt trong thời gian thực |
| Kết quả | Hiển thị kết quả nhận diện trên giao diện người dùng |

**Đối với người dùng**

Bảng 3. - Mô tả chức năng Nhận diện khuôn mặt từ Video đối với người dùng

|  |  |
| --- | --- |
| Mô tả | Người dùng có thể yêu cầu nhận diện khuôn mặt từ video. |
| Đầu vào | Nhấn nút "Recognize Face" để bắt đầu nhận diện. |
| Kết quả | Nhận tên người dùng hoặc thông báo "Unknown" nếu không nhận diện được |

* **Chức năng Nhận diện khuôn mặt từ Ảnh**

**Đối với hệ thống:**

Bảng 3. - Mô tả chức năng Nhận diện khuôn mặt từ Ảnh đối với hệ thống

|  |  |
| --- | --- |
| Mô tả | Hệ thống nhận diện khuôn mặt từ hình ảnh tải lên |
| Đầu vào | Nhận hình ảnh tải lên từ người dùng |
| Quá trình xử lý | Thực hiện nhận diện khuôn mặt từ hình ảnh đã tải lên |
| Kết quả | Hiển thị kết quả nhận diện trên giao diện |

**Đối với người dùng**

Bảng 3. - Mô tả chức năng Nhận diện khuôn mặt từ Ảnh đối với người dùng

|  |  |
| --- | --- |
| Mô tả | Người dùng có thể tải lên hình ảnh để nhận diện khuôn mặt |
| Đầu vào | Tải lên hình ảnh cần nhận diện. |
| Kết quả | Nhận kết quả nhận diện từ hệ thống |

* **Chức năng Hiển thị kết quả**

**Đối với hệ thống:**

Bảng 3. - Mô tả chức năng Hiển thị kết quả đối với hệ thống

|  |  |
| --- | --- |
| Mô tả | Hệ thống hiển thị kết quả nhận diện cho người dùng |
| Quá trình xử lý | Thực hiện nhận diện khuôn mặt từ mô hình huấn luyện |
| Kết quả | Hiển thị kết quả nhận diện trên giao diện |

**Đối với người dùng**

Bảng 3. - Mô tả chức năng Hiển thị kết quả đối với người dùng

|  |  |
| --- | --- |
| Mô tả | Người dùng có thể tải lên hình ảnh để nhận diện khuôn mặt |
| Kết quả | Nhận thông tin về tên người dùng hoặc thông báo "Unknown" |

### 3.1.2. Yêu cầu phi chức năng

Các yêu cầu phi chức năng của hệ thống nhận diện khuôn mặt đóng vai trò quan trọng trong việc đảm bảo hiệu suất và trải nghiệm người dùng. Đầu tiên, về hiệu năng, hệ thống cần đảm bảo thời gian chụp ảnh và xử lý nhận diện không quá 2 giây cho mỗi lần nhận diện. Điều này giúp đáp ứng nhu cầu sử dụng trong thực tế, đặc biệt trong các tình huống cần xác thực nhanh chóng.

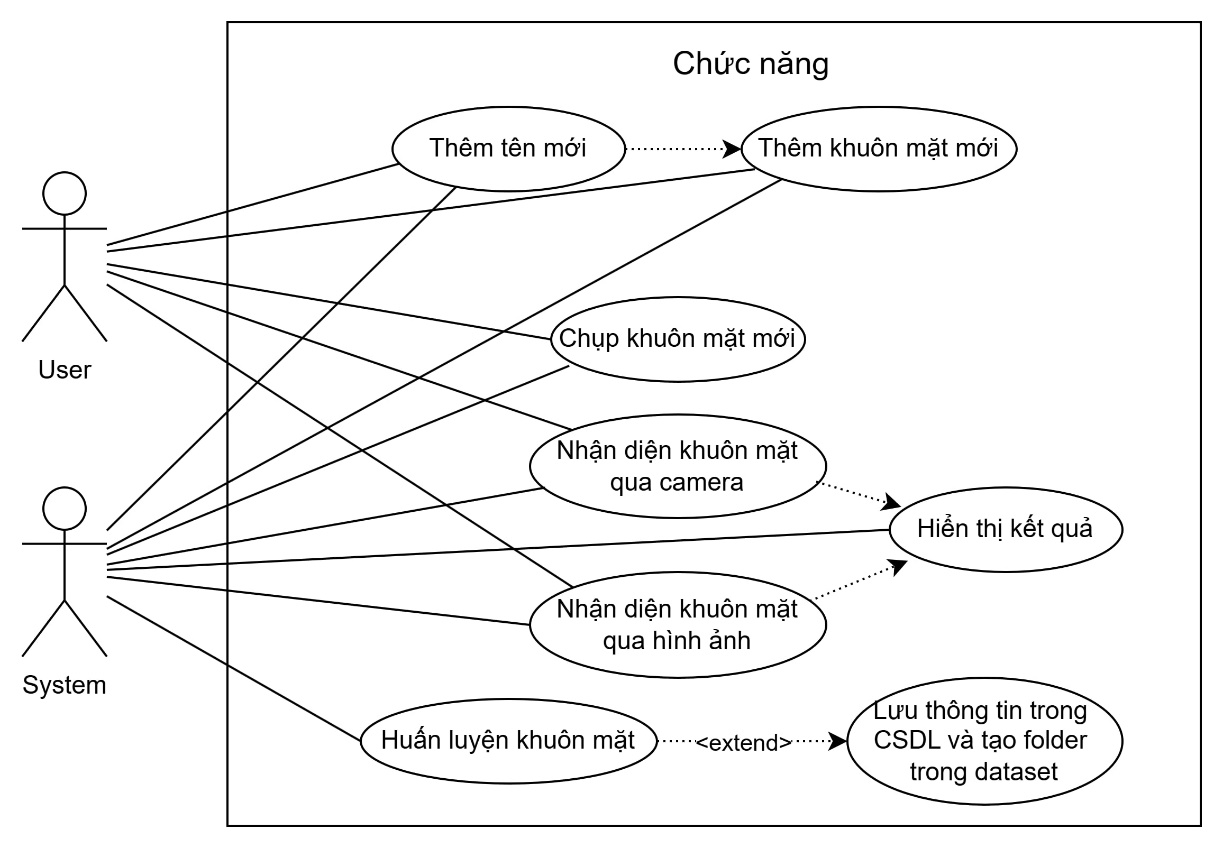
Độ tin cậy cũng là một yếu tố quan trọng. Hệ thống phải đạt được độ chính xác cao trong việc nhận diện khuôn mặt, với tỷ lệ sai sót thấp hơn 5%. Khả năng xử lý các điều kiện ánh sáng khác nhau và các góc độ khuôn mặt cũng cần được xem xét để tăng cường tính hiệu quả.

Về bảo mật, dữ liệu người dùng và thông tin khuôn mặt cần được mã hóa và bảo vệ an toàn. Hệ thống cần có cơ chế xác thực người dùng trước khi cho phép truy cập vào các chức năng quản lý cơ sở dữ liệu, nhằm đảm bảo thông tin nhạy cảm không bị rò rỉ.

Tính khả dụng của hệ thống cũng rất quan trọng. Giao diện người dùng cần thân thiện và dễ sử dụng, với hướng dẫn rõ ràng cho từng chức năng. Hệ thống nên có khả năng hoạt động liên tục mà không bị ngắt quãng, đảm bảo trải nghiệm người dùng mượt mà.

Cuối cùng, khả năng mở rộng của hệ thống là một yếu tố cần lưu ý. Hệ thống nên được thiết kế để dễ dàng bổ sung các tính năng mới trong tương lai, như nhận diện cảm xúc hay phân tích hành vi, giúp không ngừng nâng cao hiệu quả và đáp ứng nhu cầu ngày càng đa dạng của người dùng.

### 3.1.3. Biểu đồ Use Case



Hình 3. – Biểu đồ Use Case tổng quát hệ thống

### 3.1.4. Biểu đồ hoạt động (Activity Diagram)

* Chức năng Chụp ảnh khuôn mặt

A screenshot of a chat

Description automatically generated

Hình 3. – Biểu đồ hoạt động chức năng Chụp ảnh khuôn mặt

* Chức năng Huấn luyện mô hình nhận diện

A screenshot of a computer

Description automatically generated

Hình 3. – Biểu đồ hoạt động chức năng Huấn luyện mô hình nhận diện

* Chức năng Nhận diện khuôn mặt từ Video

A screenshot of a computer

Description automatically generated

Hình 3. – Biểu đồ hoạt động chức năng nhận diện khuôn mặt từ video

* Chức năng Nhận diện khuôn mặt từ Ảnh

A screenshot of a computer

Description automatically generated

Hình 3. – Biểu đồ hoạt động chức năng Nhận diện khuôn mặt từ ảnh

* Chức năng Hiển thị kết quả

A screenshot of a computer

Description automatically generated

Hình 3. - Biểu đồ hoạt động chức năng Hiển thị kết quả

### 3.1.5. Biểu đồ trạng thái (State Diagram)

* Chức năng Chụp ảnh khuôn mặt

A screenshot of a phone

Description automatically generated

Hình 3. – Biểu đồ trạng thái chức năng Chụp ảnh khuôn mặt

* Chức năng Huấn luyện mô hình nhận diện

A screenshot of a phone

Description automatically generated

Hình 3. - Biểu đồ trạng thái chức năng Huấn luyện mô hình nhận diện

* Chức năng Nhận diện khuôn mặt từ Video

A screenshot of a black screen

Description automatically generated

Hình 3. - Biểu đồ trạng thái chức năng Nhận diện khuôn mặt từ video

* Chức năng Nhận diện khuôn mặt từ Ảnh

A screenshot of a phone

Description automatically generated

Hình 3. - Biểu đồ trạng thái chức năng nhận diện khuôn mặt từ ảnh

* Chức năng Hiển thị kết quả

A screenshot of a cell phone

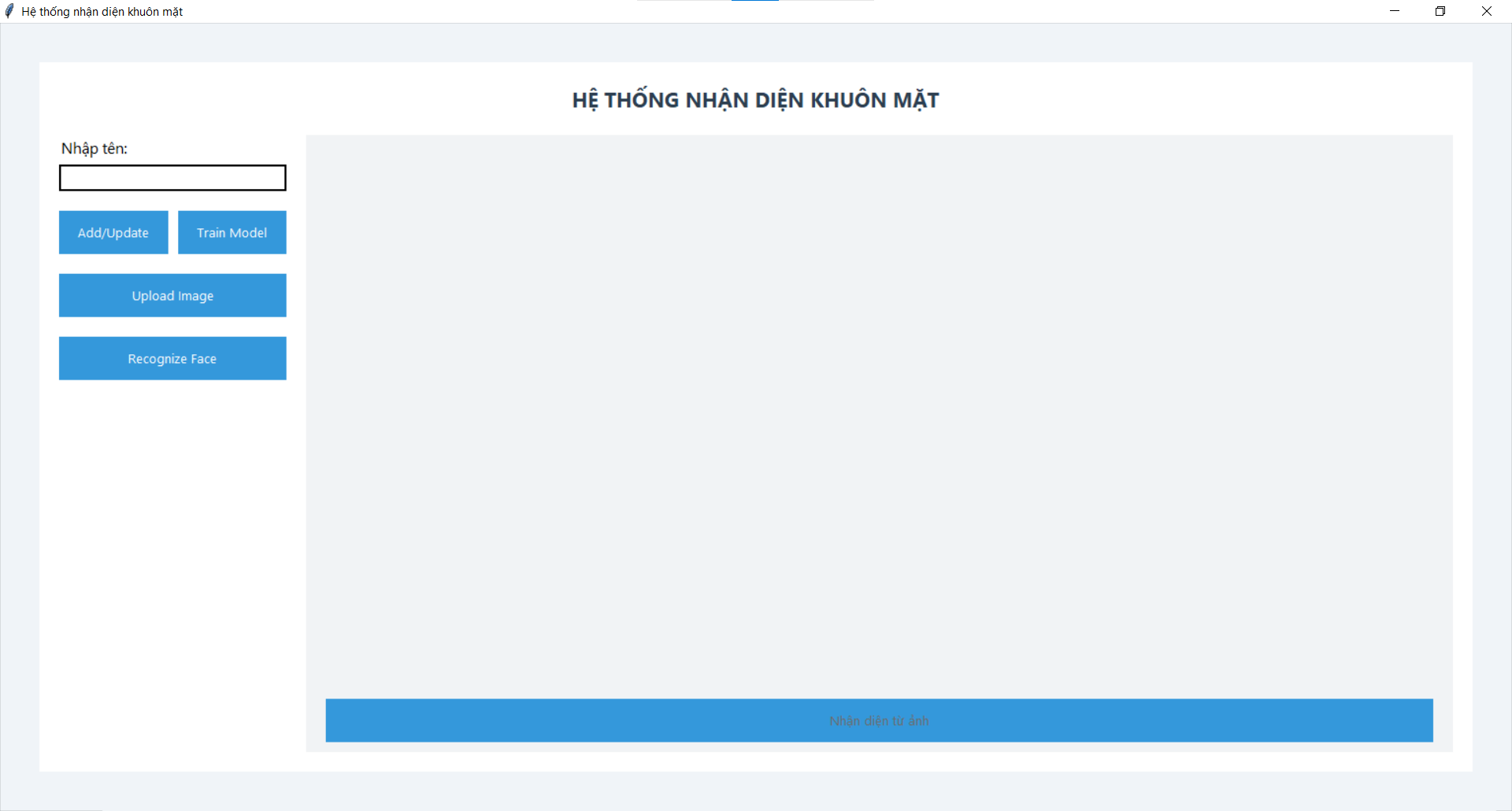
Description automatically generated

Hình 3. – Biểu đồ trạng thái chức năng Hiển thị kết quả

## 3.2. Xây dựng giao diện sản phẩm

### 3.2.1. Giao diện chính của hệ thống

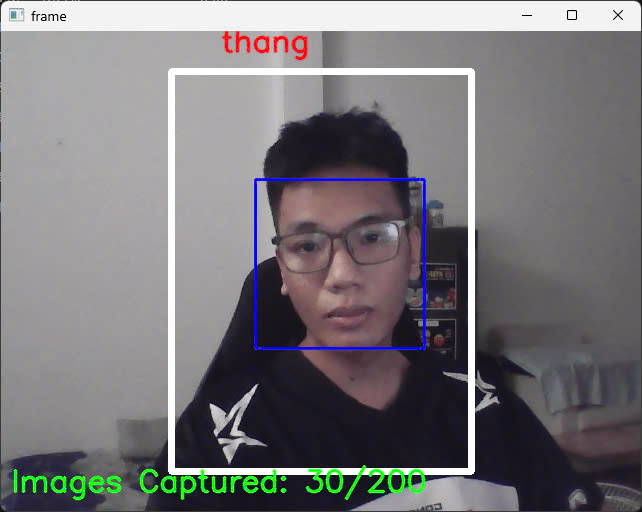
Giao diện chính là nơi tập trung các chức năng chính của hệ thống nhận diện khuôn mặt. Người dùng có thể nhập tên, thêm/cập nhật dữ liệu khuôn mặt mới, huấn luyện mô hình, tải ảnh lên, nhận diện khuôn mặt từ video và thực hiện nhận diện. Giao diện được thiết kế đơn giản, trực quan với các nút chức năng và khu vực hiển thị được sắp xếp theo thứ tự logic, giúp người dùng dễ dàng thao tác.



Hình 3. – Giao diện chính của hệ thống

### 3.2.3. Giao diện thêm khuôn mặt mới

Giao diện này cho phép người dùng thêm dữ liệu khuôn mặt mới vào hệ thống. Người dùng nhập tên ở giao diện chính sau đó nhấn Add/Update hệ thống sẽ khởi tạo cửa sổ mới cho phép người dùng chụp 200 ảnh ở mọi góc mặt để chuẩn bị dữ liệu cho việc huấn luyện. Giao diện chứa các thông tin như hướng dẫn người dùng, khung nhận diện, số lượng ảnh đã chụp.



Hình 3. – Giao diện thêm khuôn mặt mới

### 3.2.4. Giao diện thông báo huấn luyện mô hình thành công

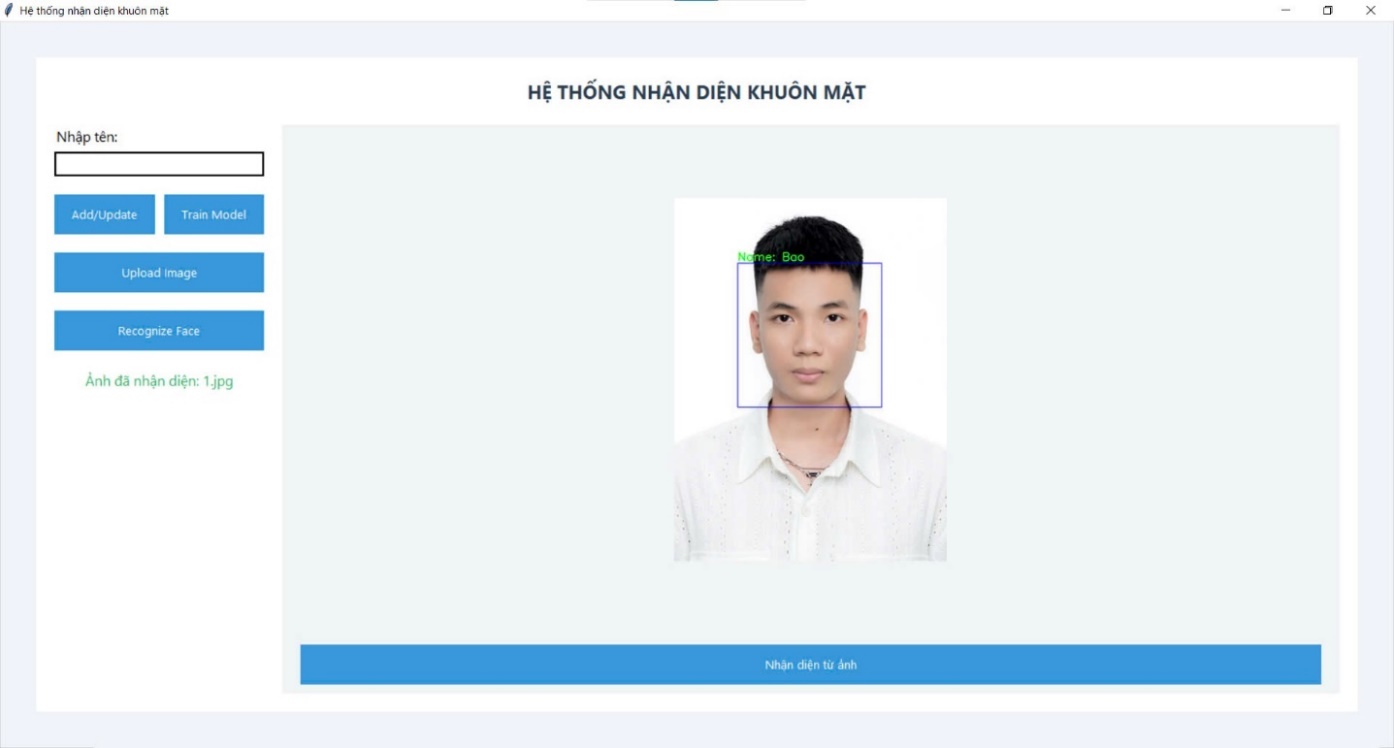
Sau khi có đủ dữ liệu khuôn mặt, khi người dùng chọn “Train Model”, hệ thống sẽ tiến hành huấn luyện mô hình nhận diện. Giao diện này hiển thị thông báo về trạng thái huấn luyện, giúp người dùng biết được quá trình huấn luyện đã hoàn tất và mô hình đã sẵn sàng để sử dụng.



Hình 3. - Giao diện thông báo huấn luyện mô hình thành công

### 3.2.5. Giao diện nhận diện khuôn mặt từ hình ảnh

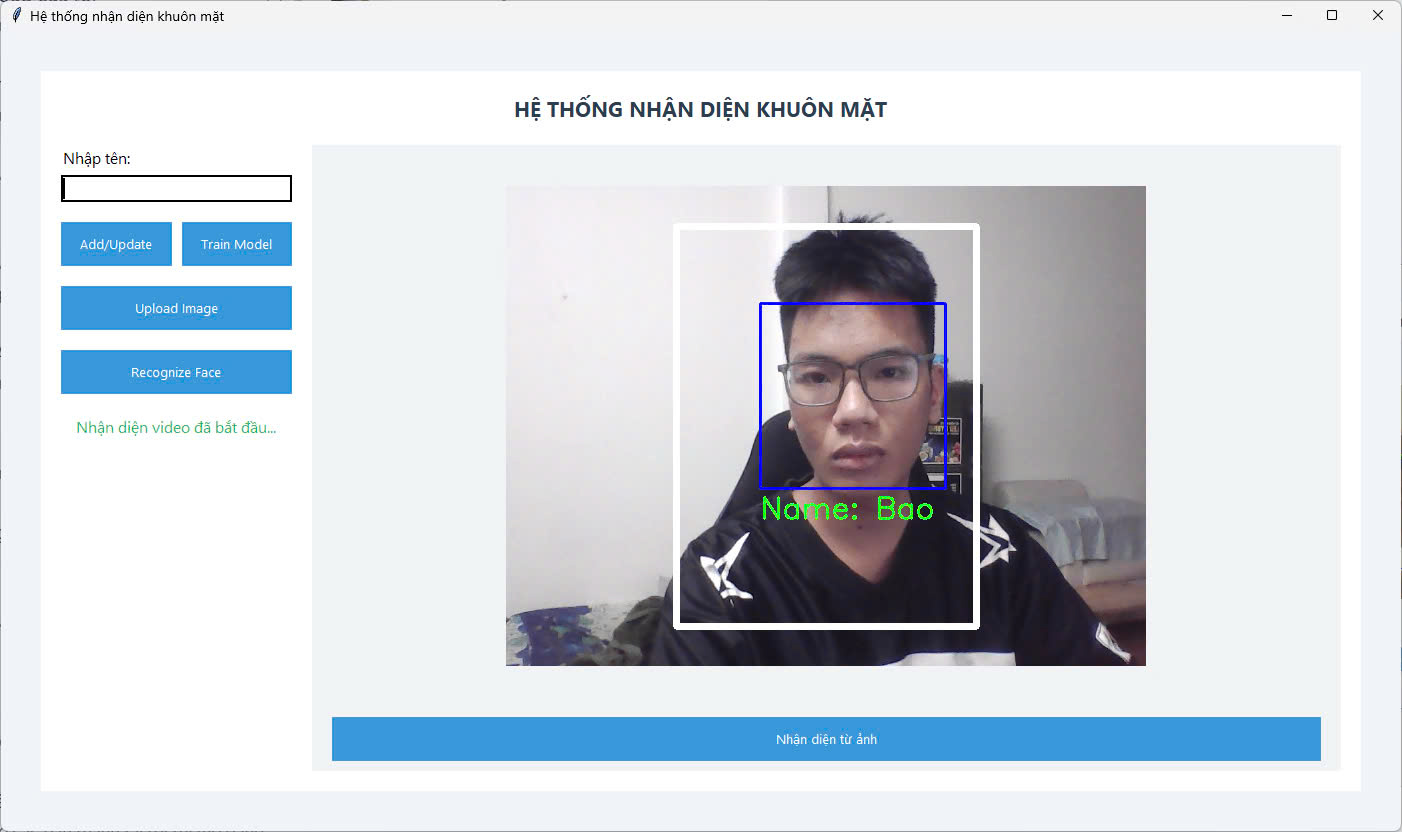
Giao diện này cho phép người dùng tải lên một hình ảnh khi chọn “Upload” và thực hiện nhận diện khuôn mặt khi chọn “Nhận diện từ ảnh”. Hệ thống sẽ phân tích hình ảnh, phát hiện khuôn mặt và hiển thị kết quả nhận diện kèm theo tên của người được nhận diện. Phần này hữu ích cho việc nhận diện từ ảnh chụp hoặc ảnh có sẵn.



Hình 3. - Giao diện nhận diện khuôn mặt từ hình ảnh

### 3.2.6. Giao diện nhận diện khuôn mặt từ video

Chức năng này cho phép nhận diện khuôn mặt theo thời gian thực thông qua video stream. Hệ thống liên tục phát hiện và nhận diện khuôn mặt xuất hiện trong khung hình, hiển thị kết quả nhận diện ngay lập tức. Đây là tính năng quan trọng cho các ứng dụng giám sát hoặc kiểm soát truy cập thời gian thực.



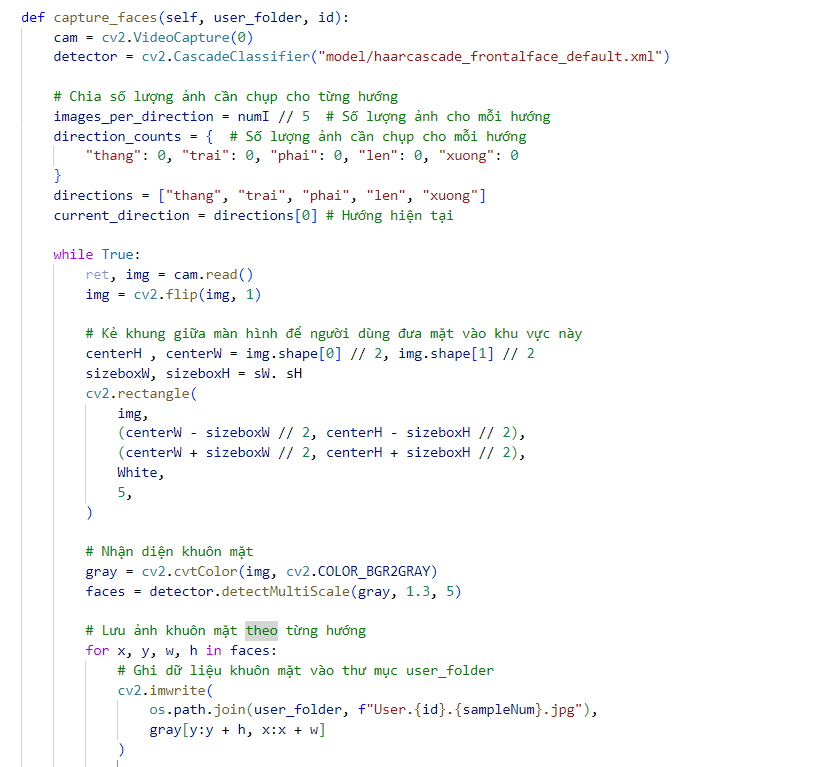
Hình 3. - Giao diện nhận diện khuôn mặt từ video

## 3.3. Các chức năng chính của hệ thống

### 3.3.1. Chức năng thu thập dữ liệu khuôn mặt

Chức năng thu thập dữ liệu khuôn mặt được thực hiện trong phương thức capture\_faces() của lớp DatabaseManager. Thuật toán hoạt động như sau:

* Sử dụng camera để chụp ảnh khuôn mặt
* Chia số lượng ảnh cho 5 hướng khác nhau: thẳng, trái, phải, lên, xuống
* Sử dụng Haar Cascade Classifier để phát hiện khuôn mặt
* Hiển thị hướng dẫn và vùng chụp ảnh trên màn hình
* Lưu ảnh khuôn mặt vào thư mục dataset



Hình 3. - Chức năng thu thập dữ liệu khuôn mặt

### 3.3.2. Chức năng huấn luyện mô hình

Quá trình huấn luyện diễn ra trong lớp TrainModel với các bước chính:

* Quét thư mục dataset để lấy ảnh khuôn mặt
* Chuyển đổi ảnh sang ảnh xám
* Sử dụng thuật toán LBPH (Local Binary Patterns Histograms) để huấn luyện
* Lưu mô hình đã huấn luyện

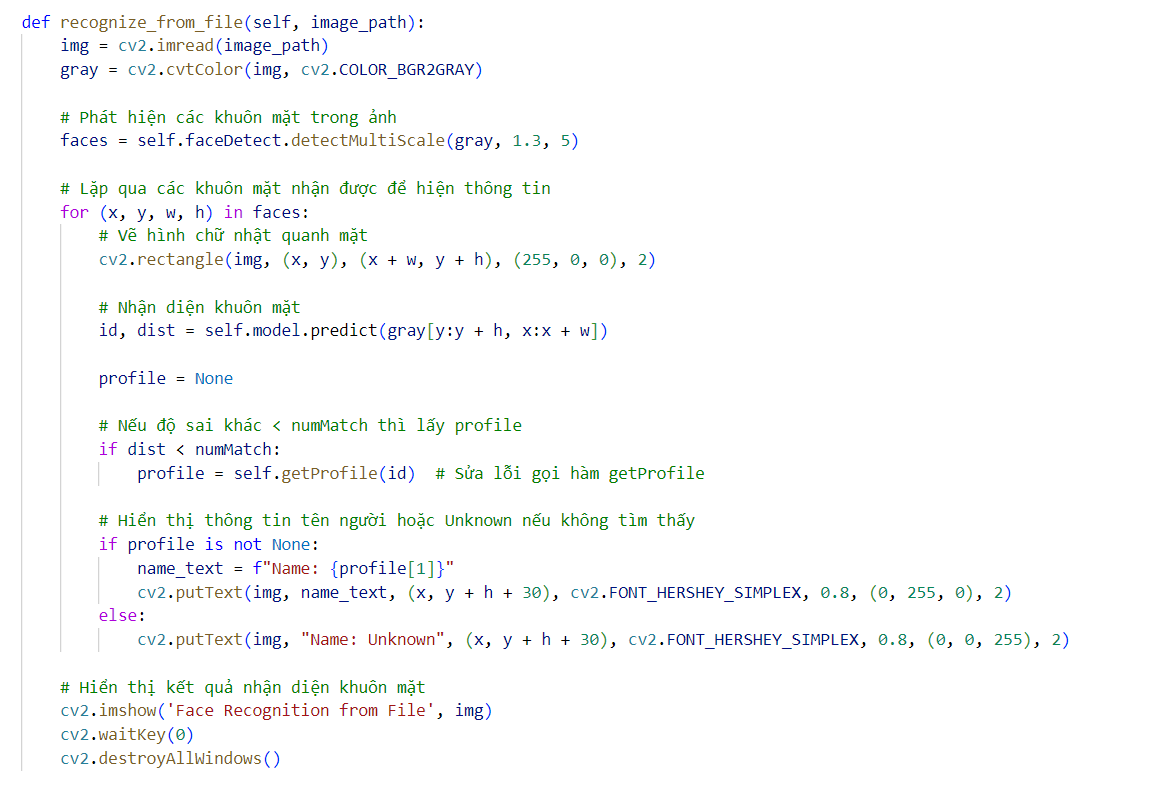


Hình 3. - Chức năng huấn luyện mô hình

### 3.3.3. Chức năng nhận diện khuôn mặt từ ảnh

Thuật toán xử lý ảnh:

* Đọc ảnh và chuyển sang ảnh xám
* Sử dụng Haar Cascade để phát hiện khuôn mặt
* Sử dụng mô hình LBPH đã huấn luyện để nhận diện
* So sánh độ chính xác với ngưỡng numMatch

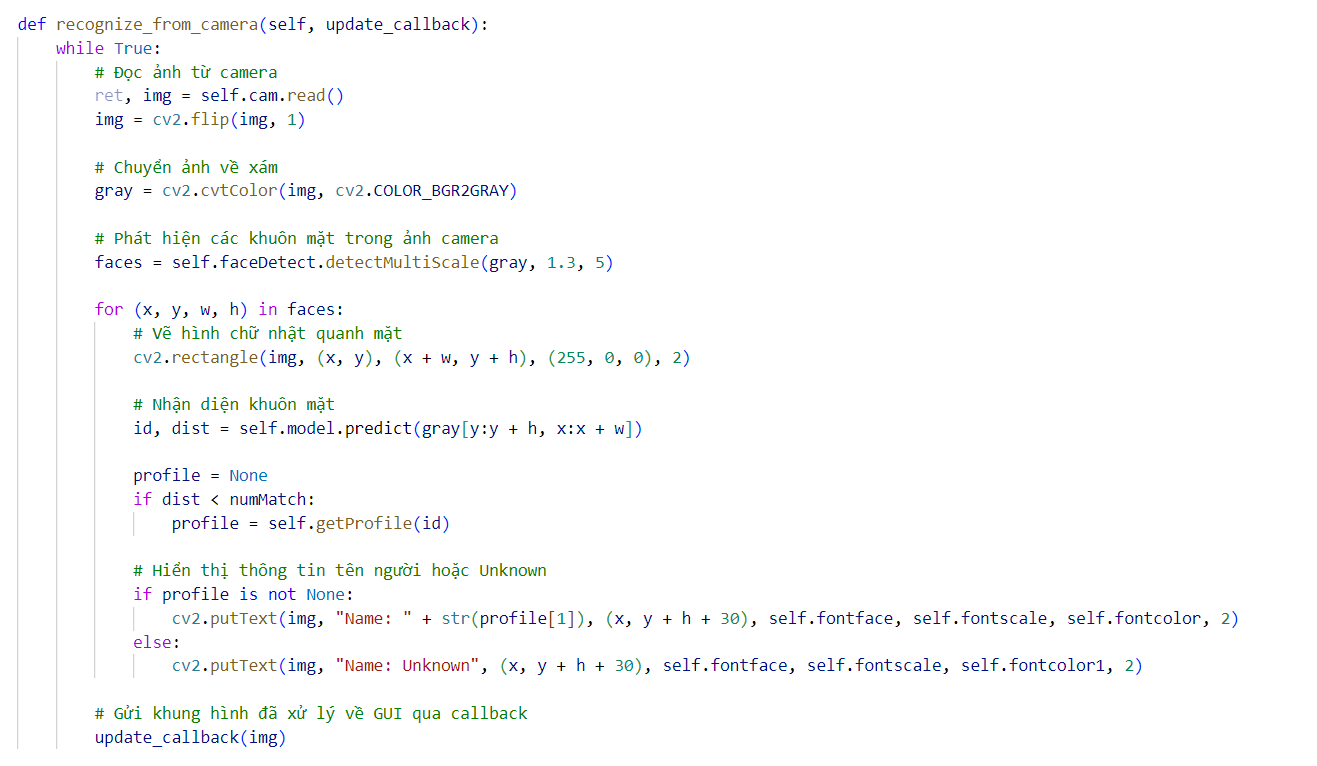


Hình 3. - Chức năng nhận diện khuôn mặt từ ảnh

### 3.3.4. Chức năng nhận diện khuôn mặt từ video

Thuật toán xử lý video stream:

* Liên tục đọc khung hình từ camera
* Lật ngược khung hình
* Phát hiện và nhận diện khuôn mặt
* Hiển thị thông tin nhận diện trên từng khung hình



Hình 3. - Chức năng nhận diện khuôn mặt từ video

# KẾT LUẬN

Trong quá trình thực hiện đồ án chuyên ngành, nhóm chúng em đã xây dựng thành công một hệ thống nhận diện khuôn mặt dựa trên thuật toán Local Binary Patterns Histograms (LBPH). Hệ thống được phát triển với các chức năng chính như thu thập dữ liệu khuôn mặt, huấn luyện mô hình, nhận diện khuôn mặt từ hình ảnh và video, cũng như hiển thị kết quả trực quan. Qua đó, sản phẩm đáp ứng được các yêu cầu cơ bản về tính chính xác, tốc độ xử lý và tính khả dụng như đã đề ra trong mục tiêu ban đầu. Những kết quả đạt được không chỉ minh chứng cho tính ứng dụng thực tiễn của thuật toán LBPH mà còn khẳng định tính khả thi của các công nghệ hỗ trợ như OpenCV, SQLite và Tkinter trong lĩnh vực xử lý ảnh.

Bên cạnh các thành công, đồ án cũng gặp phải một số hạn chế như hiệu quả nhận diện còn giảm khi điều kiện ánh sáng thay đổi lớn, hoặc khi khuôn mặt có các góc chụp quá khác biệt. Thêm vào đó, hệ thống chưa tích hợp các tính năng nâng cao như nhận diện cảm xúc hay phân tích hành vi. Những thách thức này đã mang lại nhiều bài học quý báu về cách tối ưu thuật toán, cách xử lý dữ liệu hiệu quả, cũng như sự cần thiết của việc kiểm thử hệ thống trong nhiều điều kiện thực tế.

Trong tương lai, hệ thống sẽ được cải tiến theo các hướng sau:

* Nâng cao độ chính xác: Tích hợp thêm các kỹ thuật deep learning để tăng khả năng nhận diện trong các điều kiện phức tạp.
* Tối ưu hóa hiệu năng: Sử dụng GPU để cải thiện tốc độ xử lý và khả năng hoạt động thời gian thực.
* Mở rộng chức năng: Thêm các tính năng nhận diện cảm xúc, phân tích hành vi, và tích hợp các công nghệ như IoT hoặc AI để ứng dụng trong giám sát an ninh hoặc quản lý doanh nghiệp.
* Cải thiện giao diện người dùng: Nâng cấp giao diện thân thiện và hiện đại hơn, đồng thời tích hợp đa ngôn ngữ để đáp ứng nhu cầu sử dụng rộng rãi.

# DANH MỤC TÀI LIỆU THAM KHẢO

[1] Nguyễn Công Phú, Trí tuệ nhân tạo và ứng dụng, NXB Khoa học và Kỹ thuật, 2020.

[2] Phạm Văn Tiến, Xử lý ảnh số, NXB Giáo dục Việt Nam, 2021.

[3] Nguyễn Thanh Tùng, Nhập môn lập trình Python, NXB Thông tin và Truyền thông, 2022.

[4] Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2022.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.

[6] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Pearson, 2018.

[7] Paul Viola and Michael Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2001.

[8] SQLite Documentation. Available at: https://www.sqlite.org/docs.html

[9] Python Tkinter Documentation. Available at: https://docs.python.org/3/library/tkinter.html

[10] OpenCV Documentation. Available at: https://docs.opencv.org/

[11] LBPH Algorithm Overview. Available at: https://towardsdatascience.com/

[12] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. I-511–I-518.

[13] A. Ahonen, A. Hadid, and M. Pietikäinen, “Face Description with Local Binary Patterns: Application to Face Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037–2041, 2006.

[14] G. Bradski, “The OpenCV Library,” Dr. Dobb's Journal of Software Tools, 2000.