Shape, square

Description automatically generated

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

----------------------------



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI**

**ỨNG DỤNG CNN TRONG BÀI TOÁN NHẬN DIỆN KHUÔN MẶT ĐỂ QUẢN LÝ NHÂN CÔNG**

**Giáo viên hướng dẫn: Mai Thanh Hồng**

**Lớp: 20251IT6094002 Khóa: K18**

**Nhóm sinh viên thực hiện: Nhóm 1**

**Mã SV: 2023607171 Họ và tên: Phan Đại Cương**

**Mã SV: 2023603478 Họ và tên: Nguyễn Thành Đạt**

**Mã SV: 2022607427 Họ và tên: Nguyễn Hữu Hải**

**Mã SV: 2023600777 Họ và tên: Nguyễn Kim Quang**

**Mã SV: 2023600656 Họ và tên: Khương Xuân Toàn**

**Hà Nội, năm 2025**

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin bày tỏ lòng biết ơn sâu sắc tới cô Mai Thanh Hồng. Trong suốt thời gian thực hiện bài tập lớn, cô đã luôn quan tâm, nhiệt tình chỉ bảo và đưa ra những định hướng quý báu giúp chúng em giải quyết các vấn đề cốt lõi của đề tài một cách hiệu quả nhất.

Chúng em cũng xin gửi lời tri ân chân thành đến Ban lãnh đạo và quý Thầy, Cô giáo Trường Công nghệ Thông tin và Truyền thông - Đại học Công nghiệp Hà Nội. Những nền tảng kiến thức vững chắc, từ đại cương đến chuyên ngành mà các thầy cô truyền đạt chính là hành trang vô giá giúp chúng em hoàn thành bài tập này cũng như vững bước trong tương lai.

Mặc dù chúng em đã nỗ lực hết sức, song do giới hạn về mặt thời gian và kinh nghiệm thực tế chưa nhiều, bài báo cáo chắc chắn khó tránh khỏi những khiếm khuyết. Chúng em rất mong nhận được sự thông cảm và những ý kiến đóng góp thẳng thắn từ quý Thầy, Cô để chúng em có thể rút kinh nghiệm và hoàn thiện bản thân hơn.

**Chúng em xin chân thành cảm ơn!**

PHIẾU HỌC TẬP CÁ NHÂN/NHÓM

**I. Thông tin chung**

1. Tên lớp: 20251IT6094002            Khóa:18

2. Họ và tên sinh viên: Nguyễn Kim Quang

3.Tên nhóm : Nhóm 1  
Họ và tên thành viên trong nhóm:

|  |  |
| --- | --- |
| Họ và tên | Mã sinh viên |
| Phan Đại Cương | 2023607171 |
| Nguyễn Thành Đạt | 2023603478 |
| Nguyễn Hữu Hải | 2022607427 |
| Nguyễn Kim Quang | 2023600777 |
| Khương Xuân Toàn | 2023600656 |

**II. Nội dung học tập**

**1. Tên chủ đề:** Ứng dụng CNN trong bài toán nhận diện khuôn mặt để quản lý nhân công.

**2. Hoạt động của sinh viên**

*Nội dung 1*: Tìm hiểu lý thuyết về mạng CNN và bài toán Face Recognition:

Mục tiêu/chuẩn đầu ra: Nắm vững kiến thức cơ bản về mạng nơ-ron tích chập (CNN): cấu trúc các lớp (Convolution, Pooling, Fully Connected).

Hiểu rõ quy trình của bài toán nhận diện khuôn mặt: Phát hiện khuôn mặt (Face Detection), Căn chỉnh (Alignment), Trích xuất đặc trưng (Feature Extraction), Định danh (Matching).

Tìm hiểu các kiến trúc mạng phổ biến ResNet để áp dụng vào bài toán.

*Nội dung 2:* Thu thập và xử lý dữ liệu khuôn mặt:

Mục tiêu/chuẩn đầu ra: Có khả năng thu thập dữ liệu ảnh khuôn mặt từ các thành viên trong nhóm hoặc bộ dữ liệu mẫu (như LFW, CelebA) để giả lập môi trường nhân sự. Thực hiện tiền xử lý dữ liệu: Sử dụng OpenCV/MTCNN để cắt vùng mặt, loại bỏ nhiễu, chuẩn hóa kích thước ảnh và chia tập dữ liệu thành Train/Validation/Test một cách hợp lý.

*Nội dung 3:* Huấn luyện, đánh giá mô hình CNN, từ đó đưa vào sử dụng thực tế:

Sử dụng thành thạo các thư viện Deep Learning (PyTorch) để xây dựng và huấn luyện mô hình CNN trích xuất đặc trưng khuôn mặt. Đóng gói mô hình thành API (sử dụng FastAPI), kết hợp với Backend (Spring Boot) và Frontend (ReactJS) để tạo thành hệ thống điểm danh hoàn chỉnh hoạt động trong thực tế.

Sử dụng docker-compose để đóng gói toàn bộ các thành phần (FastAPI, Spring Boot, ReactJS và database) thành các container độc lập, đảm bảo khả năng triển khai nhanh chóng, dễ mở rộng và đồng nhất giữa các môi trường. Hệ thống được cấu hình để các service tự động kết nối với nhau thông qua internal network của Docker, giúp quá trình triển khai và vận hành trở nên ổn định và thuận tiện hơn.

**3. Sản phẩm nghiên cứu**

Một hệ thống nhận diện khuôn mặt để quản lí nhân công bao gồm:

Mô hình CNN đã được huấn luyện và tối ưu hóa, sử dụng PyTorch để trích xuất đặc trưng khuôn mặt và đạt độ chính xác cao trong môi trường thực tế. Mô hình được triển khai dưới dạng API thông qua FastAPI, đảm bảo tốc độ xử lý nhanh và khả năng đáp ứng theo thời gian thực.

Ứng dụng Web có giao diện người dùng (Frontend – ReactJS) hỗ trợ chấm công trực tiếp qua Webcam, hiển thị thông tin nhân viên, lịch sử điểm danh và các chức năng quản trị. Hệ thống Backend gồm Spring Boot (quản lý nghiệp vụ, kết nối cơ sở dữ liệu, phân quyền…) và FastAPI (xử lý AI), được thiết kế theo kiến trúc microservice giúp tách biệt giữa xử lý AI và nghiệp vụ quản lý. Dữ liệu chấm công được lưu trữ tập trung, đảm bảo tính toàn vẹn và dễ dàng truy vấn.

Hệ thống được container hóa bằng Docker và docker-compose, bao gồm các service: ReactJS, Spring Boot, FastAPI và database (PostgreSQL/MySQL).

Tài liệu báo cáo chi tiết mô tả quy trình xây dựng mô hình, xử lý dữ liệu, thiết kế kiến trúc hệ thống, kết quả đánh giá, hiệu năng thực tế và những khó khăn gặp phải trong quá trình phát triển, kèm theo hướng cải tiến trong tương lai.

**III. Nhiệm vụ học tập**

1. Hoàn thành Tiểu luận, Bài tập lớn, Đồ án/Dự án theo đúng thời gian quy định.

2. Báo cáo sản phẩm nghiên cứu theo chủ đề được giao trước giảng viên và những sinh viên khác.

**IV. Học liệu thực hiện Tiểu luận, Bài tập lớn, Đồ án/Dự án**

**1.Tài liệu học tập:**

[1]. Trần Hùng Cường, Nguyễn Phương Nga, Giáo trình Trí tuệ nhân tạo, 2014

[2]. Joshi, Practeek, Artificial intelligence with python, 2017

[3]. AI VIET NAM – COURSE 2022 [PDF], Basic CNN - Exercise, 11/12/2022

[4]. AI VIET NAM – COURSE 2022 Basic CNN - Tutorial, 11/12/2022

[5]. Afshine Amidi, CS230 - Convolutional Neural Network, 10/2019

[6]. W3Schools. (n.d.). Machine Learning Tutorials. Truy cập tại: https://www.w3schools.com

[7]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. Truy cập tại: https://www.deeplearningbook.org/

**2.Phương tiện, nguyên liệu thực hiện Tiểu luận, Bài tập lớn, Đồ án/Dự án (nếu có)**……………………………………………………………………….

KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ ÁN/DỰ ÁN

Tên lớp : 20251IT6094002 Khóa :18

Họ và tên sinh viên:

Tên nhóm: 1

Tên chủ đề: Ứng dụng CNN trong bài toán nhận diện khuông mặt để quản lý nhân công.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần** | **Người thực hiện** | **Nội dung công việc** | **Phương pháp thực hiện** |
| 1 | Cả nhóm | Đưa ra lý do chọn đề tài | Dựa vào kiến thức tìm hiểu và hướng dẫn viết BM từ giảng viên |
| Nguyễn Kim Quang | Viết biên bản cuộc họp | Trao đổi trong nhóm và ghi lại nội dung thảo luận |
| Phan Đại Cương | - Làm Phiếu hoạt động nhóm cá nhân/nhóm  - Tìm hiểu mục đích của đề tài đề ra phương pháp nghiên cứu | Dựa vào mẫu phiếu được phát và tài liệu tham khảo từ giảng viên |
| Nguyễn Hữu Hải | -Làm phiếu hoạt động nhóm theo tuần  -Tìm hiểu về phạm vi nghiên cứu, đối tượng nghiên cứu và tiềm năng phát triển | Tham khảo đề tài tương tự và tài liệu môn học |
| Khương Xuân Toàn và Nguyễn Thành Đạt | -Tìm hiểu phạm vi nghiên cứu, đối tượng nghiên cứu và tiềm năng phát triển | Tra cứu sách, internet và tài liệu giảng viên cung cấp |
| 2 | Nguyễn Kim Quang | -Tổng hợp nội dung ra thành văn bản word  -Hỗ trợ nội dung cho các thành viên trong nhóm | Dựa vào nội dung từ các thành viên gửi và chỉnh sửa, định dạng |
| Phan Đại Cương và Nguyễn Thành Đạt | -Tìm hiểu ứng dụng thực tế | Tra cứu tài liệu, ví dụ thực tiễn trên internet |
| Khương Xuân Toàn | -Tìm hiểu nội dung kiến trúc cơ bản | Học qua slide bài giảng và tài liệu học tập |
| Nguyễn Thành Đạt | -Tìm hiểu tổng quan và khái niệm cơ bản của CNN | Tham khảo sách, internet và tài liệu giảng viên cung cấp |
| 3 | Khương Xuân Toàn | Trình bày nội dung về lớp làm phẳng và lớp kết nối hoàn toàn | Tham khảo giáo trình, bài giảng, tìm hiểu thêm từ internet và tài liệu học thuật. |
| Nguyễn Kim Quang | -Viết Phiếu học tập cá nhân/ nhóm -Tìm hiểu về lớp đầu ra và đưa ra kỹ thuật hỗ trợ | Đọc tài liệu học tập, làm việc nhóm để thảo luận và viết phiếu học tập. |
| Nguyễn Thành Đạt | -Viết hoạt động đạt được của tuần 1 vào phiếu Kế hoạch làm việc nhóm -Tìm hiểu về lớp kích hoạt và lớp pooling | Tổng hợp hoạt động nhóm, tìm hiểu tài liệu và ví dụ minh họa về lớp kích hoạt và pooling. |
| Nguyễn Hữu Hải và Phan Đại Cương | -Tìm hiểu về lớp đầu vào và lớp tích chập | Tham khảo tài liệu giảng dạy, video minh họa và đọc sách chuyên sâu về mạng nơ-ron tích chập. |
| 4 | Phan Đại Cương | -Tìm hiểu về thuật toán Avgmax | Tự nghiên cứu tài liệu, ghi chú nội dung chính |
| Nguyễn Kim Quang | - Tìm hiểu về thuật toán HOG  -Viết phiếu học tập cá nhân nhóm | Tự học qua tài liệu, trao đổi với nhóm |
| Khương Xuân Toàn và Nguyễn Hữu Hải | - Tìm hiểu về thuật toán Softmax  - Viết hoạt động đạt được của tuần 1 vào phiếu Kế hoạch làm việc nhóm | Tìm tài liệu tham khảo, thảo luận nhóm, tổng hợp vào phiếu |
| Nguyễn Thành Đạt | - Tìm hiểu thuật toán Dlib\_face\_recognition\_resnet | Tự học, chia sẻ kết quả trong nhóm |
| 5 | Phan Đại Cương | -Hỗ trợ trình bày báo cáo | Sử dụng word để định dạng báo cáo |
| Nguyễn Kim Quang | -Viết phiếu học tập cá nhân nhóm | Dựa trên nhật ký hoạt động nhóm và phản hồi của các thành viên |
| Khương Xuân Toàn | -Viết hoạt động đạt được của tuần 2,3,4 vào phiếu Kế hoạch làm việc nhóm | Dựa trên nhật ký hoạt động nhóm và phản hồi của các thành viên |
| Nguyễn Thành Đạt và Nguyễn Hữu Hải | -Hỗ trợ trình bày báo cáo | Sử dụng word để định dạng báo cáo |
| 6 | Phan Đại Cương | -Trình bày word và và đẩy các file nội dung và file code | Tổ chức folder Drive theo danh mục rõ ràng; kiểm tra định dạng và nội dung trước khi upload. |
| Nguyễn Kim Quang | -Viết Phiếu học tập cá nhân/ nhóm | Dựa trên mẫu có sẵn; tổng hợp hoạt động và tự đánh giá mức độ hoàn thành. |
| Nguyễn Hữu Hải | -Viết hoạt động đạt được của tuần 5,6  vào phiếu Kế hoạch làm việc nhóm | Dựa trên nhật ký hoạt động nhóm và phản hồi của các thành viên |
| Khươn Xuân Toàn và Nguyễn Thành Đạt | -Viết tiến độ hoạt động nhóm, trình bày đẩy file frontend và backend | Tổng hợp từ khóa từ tài liệu chính; cập nhật bảng tiến độ trong file nhóm. |
| 7 | Phan Đại Cương | -Trình bày word và và đẩy các file nội dung và file code, dữ liệu test lên drive nhóm | Tổ chức folder Drive theo danh mục rõ ràng; kiểm tra định dạng và nội dung trước khi upload. |
| Nguyễn Kim Quang | -Viết phiếu học tập cá nhân/ nhóm | Dựa trên mẫu có sẵn; tổng hợp hoạt động và tự đánh giá mức độ hoàn thành. |
| Nguyễn Thành Đạt | -Viết Phiếu học tập cá nhân/ nhóm  -Triển khai code frontend | Dựa trên mẫu có sẵn; tổng hợp hoạt động và tự đánh giá mức độ hoàn thành. |
| Nguyễn Hữu Hải | -Viết hoạt động đạt được của tuần  vào phiếu Kế hoạch làm việc nhóm  -Tìm các phần key word để cả nhóm học | Dựa trên nhật ký hoạt động nhóm và phản hồi của các thành viên. |
| Khương Xuân Toàn | -Viết tiếp tiến độ hoạt động nhóm | Tổng hợp từ khóa từ tài liệu chính; cập nhật bảng tiến độ trong file nhóm. |

*Ngày tháng năm 2025.*

**XÁC NHẬN CỦA GIẢNG VIÊN**

*(Ký, ghi rõ họ tên)*

Mai Thanh Hồng

BÁO CÁO HỌC TẬP CÁ NHÂN/NHÓM

Tên lớp: 20251IT6094002 Khóa:18

Họ và tên sinh viên:

Tên nhóm: 1

Tên chủ đề : Ứng dụng CNN trong bài toán nhận diện khuông mặt để quản lý nhân công.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tuần** | **Người thực hiện** | **Nội dung công việc** | **Kết quả đạt được** | **Kiến nghị với giảng viên hướng dẫn** |
| 1 | Phan Đại Cương | Chốt đề tài: Ứng dụng CNN trong phát hiện vết nứt trên bề mặt đá tự nhiên  Thành lập kênh giao tiếp chung cho nhóm và drive để lưu trữ tài liệu nhóm  Viết biên bản cuộc họp, phiếu hoạt động nhóm cá nhân và nhóm. | Hoàn Thành |  |
| 2 | Nguyễn Kim Quang | Tìm hiểu tổng quan về đề tài | Hoàn Thành |  |
| 3 | Khương Xuân Toàn | Trình bày tổng quan về mạng CNN  Trình bày ứng dụng thực tế và nội dung kiến trúc cơ bản của CNN | Hoàn Thành |  |
| 4 | Nguyễn Hữu Hải | Trình bày các lớp trong mô hình CNN | Hoàn Thành |  |
| 5 | Nguyễn Thành Đạt | Trình bày thuật toán Softmax và AvgMax | Hoàn Thành |  |
| 6 | Khương Xuân Toàn, Phan Đại Cương | Trình bày về thuật toán Resnet  Xây dựng mô hình CNN | Hoàn thành |  |
| 7 | Nguyễn Kim Quang, Nguyễn Hữu Hải | Trình bày thuật toán HOG và Dlib\_face\_recognition\_resnet | Hoàn thành |  |
| 8 | Nguyễn Thành Đạt | Hoàn thiện báo cáo | Hoàn thành |  |

*Ngày….tháng…..năm…...*

**XÁC NHẬN CỦA GIẢNG VIÊN**

*(Ký, ghi rõ họ tên)*

*Mai Thanh Hồng*

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc216280106)

[PHIẾU HỌC TẬP CÁ NHÂN/NHÓM 2](#_Toc216280107)

[KẾ HOẠCH THỰC HIỆN TIỂU LUẬN, BÀI TẬP LỚN, ĐỒ ÁN/DỰ ÁN 5](#_Toc216280108)

[BÁO CÁO HỌC TẬP CÁ NHÂN/NHÓM 11](#_Toc216280109)

[MỤC LỤC 14](#_Toc216280110)

[DANH SÁCH HÌNH ẢNH 19](#_Toc216280111)

[LỜI MỞ ĐẦU 20](#_Toc216280112)

[CHƯƠNG I: TỔNG QUAN VỀ DỰ ÁN 22](#_Toc216280113)

[1.1. Lý do chọn đề tài 22](#_Toc216280114)

[1.1.1. Nhu cầu ứng dụng trong quản lý nhân công 22](#_Toc216280115)

[1.1.2. Ý nghĩa của bài toán nhận diện khuôn mặt 23](#_Toc216280116)

[1.2. Mục tiêu đề tài 23](#_Toc216280117)

[1.2.1. Mục tiêu tổng quát 23](#_Toc216280118)

[1.2.2. Mục tiêu cụ thể 23](#_Toc216280119)

[1.3. Phương pháp nghiên cứu 24](#_Toc216280120)

[1.3.1. Phương pháp lý thuyết 24](#_Toc216280121)

[1.3.2. Phương pháp thực nghiệm 24](#_Toc216280122)

[1.3.3. Công cụ hỗ trợ 24](#_Toc216280123)

[1.3.4. Đối tượng nghiên cứu 24](#_Toc216280124)

[1.3.5. Phạm vi nghiên cứu 25](#_Toc216280125)

[1.4. Tiềm năng phát triển 25](#_Toc216280126)

[1.5. Tầm nhìn dài hạn 25](#_Toc216280127)

[1.6. Ngôn ngữ và công nghệ áp dụng 26](#_Toc216280128)

[CHƯƠNG II: TỔNG QUAN VỀ BÀI TOÁN FACE RECOGNITION 28](#_Toc216280129)

[2.1. Mô tả bài toán 28](#_Toc216280130)

[2.2. Mô hình áp dụng: Resnet 28](#_Toc216280131)

[2.3. Thách thức đặc thù của dữ liệu khuôn mặt 29](#_Toc216280132)

[2.4. Mục tiêu của mô hình CNN 29](#_Toc216280133)

[CHƯƠNG III: CƠ SỞ LÝ THUYẾT VÀ MÔ HÌNH CNN 30](#_Toc216280134)

[3.1. Convoulutional Neural Network (CNN) 30](#_Toc216280135)

[3.1.1. Tổng quan Convolutional Neural Network (CNN) 30](#_Toc216280136)

[3.1.1.1. Normal Neural Network trên images 30](#_Toc216280137)

[3.1.1.2. CNN hoạt động ra sao? 30](#_Toc216280138)

[3.1.2. Kiến trúc của một mạng CNN 32](#_Toc216280139)

[3.1.2.1.Các kiểu tầng 32](#_Toc216280140)

[3.1.2.1.1. Tầng tích chập (CONV) 32](#_Toc216280141)

[3.1.2.1.2. Pooling (POOL) 33](#_Toc216280142)

[3.1.2.1.3. Fully Connected (FC) 33](#_Toc216280143)

[3.1.3. Các siêu tham số của bộ lọc 34](#_Toc216280144)

[3.1.3.1. Các chiều của một bộ lọc 34](#_Toc216280145)

[3.1.3.2. Stride 34](#_Toc216280146)

[3.1.3.3. Zero-padding 34](#_Toc216280147)

[3.1.4. Điều chỉnh siêu tham số 35](#_Toc216280148)

[3.1.4.1. Tính tương thích của tham số trong mô hình tích chập 35](#_Toc216280149)

[3.1.4.2. Hiểu về độ phức tạp của mô hình 36](#_Toc216280150)

[3.1.4.3. Trường Thụ Cảm (receptive field) 37](#_Toc216280151)

[3.1.5. Các hàm kích hoạt thường gặp (activation funcion) 38](#_Toc216280152)

[3.1.5.1. Rectified Linear Unit (ReLU) 38](#_Toc216280153)

[3.1.5.2. Softmax 38](#_Toc216280154)

[3.2. Mô hình Resnet 39](#_Toc216280155)

[3.2.1 Giới thiệu tổng quan 39](#_Toc216280156)

[3.2.2. Cấu trúc và nguyên lý hoạt động của mô hình resnet 39](#_Toc216280157)

[3.2.2.1. Cấu trúc 39](#_Toc216280158)

[3.2.2.2. Khối residual cơ bản: 40](#_Toc216280159)

[3.2.3. Ưu điểm và hạn chế 42](#_Toc216280160)

[3.2.3.1. Ưu điểm: 42](#_Toc216280161)

[3.2.3.2. Hạn chế: 43](#_Toc216280162)

[3.2.4. Ứng dụng trong dự án nhận diện khuôn mặt quản lý nhân công 43](#_Toc216280163)

[3.3. Thuật toán dlib\_face\_recognition\_resnet\_model 46](#_Toc216280164)

[3.3.1. Tổng quan 46](#_Toc216280165)

[3.3.2. Kiến trúc và nguyên tắc huấn luyện 46](#_Toc216280166)

[3.3.3. Cách hoạt động (Inference pipeline) 47](#_Toc216280167)

[3.3.4. Ưu điểm 47](#_Toc216280168)

[3.3.5. Hạn chế và lưu ý 47](#_Toc216280169)

[3.4. Thuật toán HOG 48](#_Toc216280170)

[3.4.1. Thuật toán HOG (Histogram of Oriented Gradients) 48](#_Toc216280171)

[3.4.1.1 Tiền xử lý 49](#_Toc216280172)

[3.4.1.2 Tính Gradient 49](#_Toc216280173)

[3.4.1.3. Xây dựng biểu đồ Histogram trong các Cell 50](#_Toc216280174)

[3.4.1.4. Chuẩn hóa khối (Block Normalization) 50](#_Toc216280175)

[3.4.1.5. Tổng hợp Vector đặc trưng: 51](#_Toc216280176)

[3.4.2. Ưu điểm và nhược điểm của thuật toán 51](#_Toc216280177)

[3.5. Dữ liệu và tiền xử lí: 52](#_Toc216280178)

[3.5.1 Giới thiệu dữ liệu 52](#_Toc216280179)

[3.5.1.1. Nguồn dữ liệu 52](#_Toc216280180)

[3.5.1.2 Mô tả đặc điệm dữ liệu 53](#_Toc216280181)

[3.5.1.3. Các bước tiền xử lí 53](#_Toc216280182)

[CHƯƠNG IV: SO SÁNH THUẬT TOÁN HOG VÀ THUẬT TOÁN 55](#_Toc216280183)

[4.1. Tổng quan mục tiêu so sánh 55](#_Toc216280184)

[4.2. Mục đích sử dụng của hai thuật toán 55](#_Toc216280185)

[4.3. So sánh chi tiết 56](#_Toc216280186)

[CHƯƠNG V: SERVING MODEL 59](#_Toc216280187)

[5.1. Giới thiệu về Serving Model 59](#_Toc216280188)

[5.2. Kiến trúc API Service 59](#_Toc216280189)

[5.2.1. Khởi tạo FastAPI App 59](#_Toc216280190)

[5.2.2. Endpoint nhận diện chính 60](#_Toc216280191)

[5.2.3. . Endpoint tìm kiếm khuôn mặt 61](#_Toc216280192)

[5.2.4. Endpoint thêm người dùng 62](#_Toc216280193)

[5.2.5. Endpoint reload cache 63](#_Toc216280194)

[5.3. Kiến trúc Search Index 64](#_Toc216280195)

[5.3.1. Lớp SearchIndex với FAISS/Annoy 64](#_Toc216280196)

[5.3.2. Query method 66](#_Toc216280197)

[5.4. Database Layer 66](#_Toc216280198)

[5.4.1. Kết nối database 66](#_Toc216280199)

[5.4.2. Các hàm chính trong db.py 68](#_Toc216280200)

[5.5. Công cụ hỗ trợ 71](#_Toc216280201)

[5.5.1. Encode faces từ dataset 71](#_Toc216280202)

[5.5.2. Script nhận diện từ video 73](#_Toc216280203)

[5.5.3. Script nhận diện từ ảnh 76](#_Toc216280204)

[5.5.4. Thu thập dữ liệu từ webcam 78](#_Toc216280205)

[5.6. Deployment và Scaling 78](#_Toc216280206)

[5.6.1. Chạy API sever 78](#_Toc216280207)

[5.6.2. Docker deployment 78](#_Toc216280208)

[CHƯƠNG VI: THIẾT KẾ HỆ THỐNG GIAO DIỆN TRỰC QUAN 80](#_Toc216280209)

[6.1. Công cụ và kiến trúc hệ thống được lựa chọn 80](#_Toc216280210)

[6.2. Các tính năng chính của công nghệ sử dụng 80](#_Toc216280211)

[6.3. Triển khai ứng dụng 81](#_Toc216280212)

[KẾT LUẬN 83](#_Toc216280213)

[TÀI LIỆU THAM KHẢO 85](#_Toc216280214)

DANH SÁCH HÌNH ẢNH

[Hình 1: Local connected layer 31](#_Toc216280230)

[Hình 2: Shared parameters 31](#_Toc216280231)

[Hình 3: Cấu trúc một mạng CNN 32](#_Toc216280232)

[Hình 4: CONV step 32](#_Toc216280233)

[Hình 5: avg pooling 33](#_Toc216280234)

[Hình 7: Fully Connected (FC) 33](#_Toc216280235)

[Hình 8: Bộ lọc 34](#_Toc216280236)

[Hình 9: Stride 34](#_Toc216280237)

[Hình 10: Padding 36](#_Toc216280238)

[Hình 11: Công thức tính kích thước receptive field tại layer k 37](#_Toc216280239)

[Hình 12: receptive field 37](#_Toc216280240)

[Hình 14: Mô hình Resnet 40](#_Toc216280241)

[Hình 15: Khối residual cơ bản 40](#_Toc216280242)

[Hình 16: Ví dụ sử dụng ReactJS để gửi ảnh đi nhận diện 82](#_Toc216280243)

LỜI MỞ ĐẦU

Đặt vấn đề: Việc quản lý nhân sự và chấm công (timekeeping) đóng vai trò cốt lõi trong việc vận hành và tối ưu hóa chi phí của mọi doanh nghiệp. Các phương pháp truyền thống như thẻ từ, vân tay hay chấm công giấy tuy phổ biến nhưng dần bộc lộ nhiều hạn chế: dễ xảy ra tình trạng quên thẻ, chấm công hộ (gian lận), nguy cơ lây nhiễm bệnh qua tiếp xúc bề mặt hoặc tốn kém thời gian tổng hợp dữ liệu. Nếu không có giải pháp quản lý chặt chẽ, các vấn đề này sẽ dẫn đến sai sót trong tính lương, giảm kỷ luật lao động và gây thất thoát tài chính cho tổ chức.

Do đó, trong bài viết này, mục đích chính của chúng em là tìm hiểu và Xây dựng hệ thống CNN nhằm tự động hóa việc điểm danh, nhận diện danh tính nhân viên và hỗ trợ quản lý ra vào dựa trên dữ liệu hình ảnh khuôn mặt.

**Mục tiêu nghiên cứu:**

Xây dựng và huấn luyện mô hình mạng nơ-ron tích chập (CNN) để trích xuất đặc trưng và định danh chính xác nhân viên trong cơ sở dữ liệu.

Đánh giá định lượng về độ chính xác (accuracy), tốc độ phản hồi (latency) và khả năng chống giả mạo (anti-spoofing) của mô hình khi triển khai trên bộ dữ liệu thực tế tại môi trường doanh nghiệp.

**Đối tượng và phạm vi:**

Đối tượng: Ảnh khuôn mặt kỹ thuật số (RGB) hoặc video stream từ camera giám sát, đảm bảo độ phân giải tối thiểu để trích xuất đặc trưng (ví dụ: ≥ 112x112px cho vùng mặt).

Điều kiện ghi hình: Môi trường ánh sáng văn phòng hoặc ánh sáng tự nhiên, góc nghiêng khuôn mặt (pose) dao động trong khoảng Yaw/Pitch/Roll từ -30° đến +30°, khoảng cách từ camera đến người chụp từ 0.5m – 2m.

Giới hạn: Hệ thống tập trung vào nhận diện khuôn mặt người lớn, không xét các trường hợp che khuất quá 50% khuôn mặt (khẩu trang kín, mũ bảo hiểm fullface) hoặc điều kiện ánh sáng quá yếu (ban đêm không có hồng ngoại).

**Cấu trúc báo cáo:**

* Chương 1: Tổng quan về dự án.
* Chương 2: Tổng quan về bài toán Face Recognition.
* Chương 3: Cơ sở lý thuyết và mô hình CNN.
* Chương 4: So sánh thuật toán
* Chương 5: Servicing Model
* Chương 6: Thiết kế hệ thống giao diện trực quan

CHƯƠNG I: TỔNG QUAN VỀ DỰ ÁN

## Lý do chọn đề tài

Trí tuệ nhân tạo (AI), đặc biệt là học sâu (Deep Learning), đang trở thành nền tảng công nghệ cốt lõi trong nhiều lĩnh vực như tài chính, y tế, sản xuất, giao thông và quản trị doanh nghiệp. Trong đó, thị giác máy tính (Computer Vision) – nhánh AI cho phép máy móc phân tích và hiểu hình ảnh – đã tạo ra nhiều ứng dụng thực tế mang tính tự động hóa cao.

Một trong những ứng dụng phổ biến và quan trọng nhất của Computer Vision chính là nhận diện khuôn mặt (Face Recognition). Công nghệ này đang được triển khai rộng rãi trong hệ thống giám sát an ninh, mở khóa thiết bị, giao dịch tài chính, kiểm soát truy cập và quản lý nhân sự.

Do đó, phát triển một hệ thống nhận diện khuôn mặt tự động giúp tối ưu quy trình quản lý nhân công, tăng tính bảo mật, giảm phụ thuộc vào giấy tờ và nâng cao hiệu quả vận hành.

### Nhu cầu ứng dụng trong quản lý nhân công

Trong doanh nghiệp, đặc biệt tại nhà máy, văn phòng, công trường, hoạt động quản lý nhân sự — bao gồm chấm công, kiểm soát ra vào và theo dõi số lượng lao động — vẫn chủ yếu dựa vào thẻ từ, ký tên, hoặc ghi chép thủ công. Những phương pháp này tồn tại nhiều hạn chế:

* Dễ xảy ra gian lận chấm công hộ.
* Tốn thời gian, nhân lực quản lý.
* Khó lưu trữ và tổng hợp dữ liệu dài hạn.
* Độ chính xác và minh bạch chưa cao.

Do đó, việc xây dựng hệ thống nhận diện khuôn mặt tự động dựa trên CNN sẽ giúp tăng tính bảo mật, giảm chi phí vận hành và thúc đẩy chuyển đổi số trong quản lý nhân sự.

### 1.1.2. Ý nghĩa của bài toán nhận diện khuôn mặt

* Tăng hiệu quả và độ chính xác trong chấm công.
* Tự động hóa quy trình quản lý doanh nghiệp.
* Hạn chế lỗi chủ quan của con người.
* Nâng cao an ninh nội bộ và kiểm soát truy cập.

Vì vậy, nhóm lựa chọn đề tài này nhằm nghiên cứu tính khả thi, hiệu quả và tiềm năng ứng dụng thực tế của công nghệ AI trong môi trường doanh nghiệp.

## Mục tiêu đề tài

### Mục tiêu tổng quát

Xây dựng mô hình học sâu dựa trên CNN(Resnet) để nhận diện khuôn mặt nhân công, góp phần hỗ trợ hệ thống chấm công – kiểm soát ra/vào tự động.

### Mục tiêu cụ thể

* Thu thập, chuẩn hóa và tiền xử lý dữ liệu khuôn mặt.
* Huấn luyện mô hình CNN để phân loại/nhận diện danh tính.
* Tối ưu tham số mô hình nhằm cải thiện accuracy.
* Đánh giá mô hình bằng accuracy, loss, confusion matrix.
* Xây dựng pipeline ứng dụng thử nghiệm trong thực tế.

## Phương pháp nghiên cứu

### Phương pháp lý thuyết

* Tìm hiểu nền tảng về mạng nơ-ron tích chập (CNN).
* Nghiên cứu kiến trúc Resnet và các mô hình nhận diện khuôn mặt phổ biến.
* Khảo sát ứng dụng AI trong quản lý nhân sự và thị giác máy tính.
* Đọc các bài báo khoa học, tài liệu chuyên ngành liên quan.

### Phương pháp thực nghiệm

* Triển khai mô hình CNN bằng thư viện PyTorch(dùng thư viện face\_reconigition).
* Huấn luyện mô hình trên tập dữ liệu ảnh crack/non-crack.
* Đánh giá và điều chỉnh siêu tham số dựa trên kết quả thực nghiệm.

### Công cụ hỗ trợ

* Ngôn ngữ lập trình: Python, Html, Css, Java, Javascript.
* Thư viện: PyTorch, Torchvision, Matplotlib, SpringBoot, ReactJS.
* Công cụ xử lý ảnh: OpenCV.
* Môi trường phát triển: Google Colab hoặc Kaggle, Intelliij, Visual Studio Code.
* Đóng gói các thành phần của hệ thống: Docker.

### Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm ảnh khuôn mặt của nhân viên, được thu thập từ camera trong môi trường thực tế hoặc từ các bộ dữ liệu chuẩn phục vụ mục đích mô phỏng và huấn luyện. Các ảnh này có thể ở nhiều điều kiện khác nhau như:

* Điều kiện ánh sáng thay đổi (môi trường trong nhà, ngoài trời, ánh sáng yếu).
* Nhiều góc mặt khác nhau (chính diện, nghiêng nhẹ trái/phải).
* Chất lượng camera khác nhau (Webcam độ phân giải thấp đến cao).
* Biểu cảm khuôn mặt khác nhau của nhân viên.

Dữ liệu khuôn mặt được dùng cho các mục đích chính:

* Huấn luyện và tối ưu mô hình CNN trích xuất đặc trưng.
* Xây dựng tập embedding để nhận diện từng nhân viên.
* Kiểm tra và đánh giá độ chính xác, độ ổn định của hệ thống.
* Mô phỏng quá trình chấm công tự động qua Webcam.

### Phạm vi nghiên cứu

* Giới hạn trong bài toán nhận diện/kiểm tra danh tính.
* Không nghiên cứu theo dõi chuyển động, phát hiện cảm xúc hay video tracking.
* Không tập trung vào bảo mật sinh trắc học nâng cao.
* Quy mô dữ liệu ở mức vừa, phục vụ thí nghiệm mô phỏng.

## Tiềm năng phát triển

* Mở rộng sang mô hình mạnh hơn như FaceNet, MobileFaceNet.
* Tích hợp hệ thống chấm công tự động theo thời gian thực.
* Kết nối với phần mềm quản lý nhân sự (HRM).
* Triển khai trên thiết bị edge như Raspberry Pi, camera AI, Jetson Nano.
* Phát triển giao diện ứng dụng web/mobile hỗ trợ doanh nghiệp.

## Tầm nhìn dài hạn

Hệ thống nhận diện khuôn mặt có thể trở thành một phần trong nền tảng quản trị thông minh, giúp doanh nghiệp:

* Tự động hóa 100% quy trình chấm công và kiểm soát ra/vào
* Giảm chi phí quản lý nhân sự
* Nâng cao an toàn và bảo mật nội bộ
* Tạo dữ liệu lớn (HR analytics) phục vụ tối ưu vận hành

Trong tương lai, công nghệ có thể kết hợp với IoT, camera AI, blockchain hoặc robot giám sát để hình thành hệ sinh thái quản lý doanh nghiệp thông minh.

## Ngôn ngữ và công nghệ áp dụng

Trong khuôn khổ đề tài, nhóm lựa chọn Python làm ngôn ngữ lập trình trung tâm phục vụ xử lý dữ liệu, xây dựng mô hình học sâu và thực nghiệm nhờ cú pháp rõ ràng, cộng đồng lớn và hệ sinh thái thư viện mạnh. Bên cạnh đó, Java, HTML, CSS và JavaScript được sử dụng trong giai đoạn phát triển giao diện và tích hợp hệ thống, tạo nền tảng cho ứng dụng quản lý nhân công trực quan và dễ triển khai.

Mô hình nhận diện khuôn mặt được xây dựng dựa trên PyTorch, kết hợp Torchvision để quản lý tập dữ liệu ảnh, tiền xử lý và hỗ trợ các kiến trúc CNN. Các biểu đồ theo dõi quá trình huấn luyện, độ chính xác và hàm mất mát được trực quan hóa bằng Matplotlib, giúp đánh giá và phân tích mô hình một cách rõ ràng.

Phần ứng dụng thực tế được hiện thực hóa bằng Spring Boot cho backend đảm nhiệm xử lý nghiệp vụ và xây dựng API, kết hợp ReactJS để xây dựng giao diện web tương tác, hỗ trợ triển khai hệ thống nhận diện khuôn mặt trong quản lý nhân công.

Quá trình phát triển, huấn luyện và thử nghiệm mô hình được thực hiện trên Google Colab hoặc Kaggle nhằm tận dụng GPU miễn phí cho tính toán tăng tốc. Ngoài ra, nhóm sử dụng IntelliJ IDEA cho phát triển Java backend và Visual Studio Code cho lập trình Python, JavaScript và quản lý cấu trúc dự án, đảm bảo quy trình làm việc linh hoạt và nhất quán.

Để đảm bảo khả năng triển khai đồng nhất trên nhiều môi trường khác nhau, nhóm sử dụng Docker để đóng gói từng thành phần của hệ thống—bao gồm mô hình AI (FastAPI), backend Spring Boot, frontend ReactJS và cơ sở dữ liệu—thành các container độc lập. Việc sử dụng Docker giúp giảm thiểu lỗi phát sinh do chênh lệch môi trường, đồng thời đơn giản hóa quá trình phân phối và triển khai.

Bên cạnh đó, nhóm áp dụng Docker Compose để quản lý và điều phối nhiều container trong hệ thống. Với một file cấu hình duy nhất, các dịch vụ có thể tự động khởi chạy, kết nối mạng nội bộ, chia sẻ tài nguyên và hoạt động đồng bộ. Điều này giúp quá trình phát triển, kiểm thử và triển khai hệ thống trở nên thuận tiện, nhất quán và dễ dàng mở rộng.

CHƯƠNG II: TỔNG QUAN VỀ BÀI TOÁN FACE RECOGNITION

## 2.1. Mô tả bài toán

Nhận diện khuôn mặt (face recognition) là một bài toán thị giác máy tính quan trọng trong nhiều ứng dụng thực tế như an ninh, giám sát, quản lý ra/vào hay chấm công tự động. Mục tiêu là xác định xem khuôn mặt trong ảnh thuộc về người nào dựa trên đặc trưng hình ảnh.

Trong bài tập lớn này, chúng em tập trung vào nhiệm vụ phân loại danh tính nhân sự, tức là:

* Input: ảnh khuôn mặt nhân viên được thu thập từ camera
* Output: dự đoán nhân viên tương ứng hoặc xác thực có/không thuộc hệ thống

Vấn đề khó nằm ở chỗ khuôn mặt người thay đổi rất nhiều theo góc chụp, ánh sáng, biểu cảm, phụ kiện (kính, khẩu trang), làm mô hình dễ nhầm lẫn giữa những khuôn mặt có đặc điểm tương đồng. Bên cạnh đó, môi trường làm việc thực tế (nhà máy, văn phòng, công trường) thường có điều kiện ánh sáng không ổn định, khiến hình ảnh bị nhiễu, mờ, thiếu nét.

Do đó, mô hình cần “thông minh” đủ để trích xuất đặc trưng khuôn mặt ổn định và nhận diện chính xác trong nhiều bối cảnh khác nhau. Trong bài, chúng em sử dụng mạng CNN (kiến trúc LeNet) kết hợp tiền xử lý ảnh và chuẩn hóa dữ liệu nhằm giúp mô hình học được các đặc trưng phân biệt giữa các khuôn mặt nhân công.

## 2.2. Mô hình áp dụng: Resnet

Kiến trúc CNN được áp dụng trong bài toán này là Resnet-34. ResNet (Residual Network) là một trong những kiến trúc mạng nơ-ron tích chập (CNN) quan trọng và có ảnh hưởng lớn trong lĩnh vực thị giác máy tính, được đề xuất bởi Kaiming He và cộng sự vào năm 2015. ResNet ra đời nhằm giải quyết degradation problem — hiện tượng khi mô hình càng sâu thì độ chính xác lại giảm, mặc dù về lý thuyết mô hình sâu hơn có khả năng học đặc trưng tốt hơn.

## 2.3. Thách thức đặc thù của dữ liệu khuôn mặt

Các thách thức đối với dữ liệu khuôn mặt:

* Biến thiên ánh sáng: camera trong công ty có thể sáng quá, tối quá, ngược sáng.
* Phụ kiện che khuất: khẩu trang, kính, mũ bảo hộ.
* Nhiễu hình ảnh: camera thấp chất lượng, chuyển động gây mờ.
* Các khuôn mặt có nét tương đồng, dễ gây nhầm lẫn.
* Dữ liệu không cân bằng: người nhiều ảnh – người ít ảnh

Vì vậy, cần tiền xử lý ảnh (crop mặt, chuẩn hóa, cân bằng dữ liệu) và lựa chọn mô hình CNN phù hợp để mô hình học được đặc trưng thực sự đại diện cho từng nhân viên.

## 2.4. Mục tiêu của mô hình CNN

Huấn luyện và sử dụng mô hình Resnet để nhận diện và phân loại khuôn mặt nhân công**.** Tối ưu độ chính xác thông qua các siêu tham số (learning rate, batch size…)**.** Đánh giá hiệu quả mô hình trên tập kiểm thử bằng accuracy và loss**.** Tạo nền tảng cho hệ thống chấm công – quản lý nhân sự tự động trong tương lai

CHƯƠNG III: CƠ SỞ LÝ THUYẾT VÀ MÔ HÌNH CNN

## 3.1. Convoulutional Neural Network (CNN)

### 3.1.1. Tổng quan Convolutional Neural Network (CNN)

#### 3.1.1.1. Normal Neural Network trên images

Mỗi image có hàng ngàn pixels, mỗi pixel được xem như 1 feature, vậy nếu như một bức ảnh có kích thước 1000 \* 1000, thì sẽ có 1.000.000 features. Với normal feed-forward neural networks, mỗi layer là full-connected với previous input layer.

Trong noral feed-forward neural network, với mỗi layer, có 1.000.000 pixels, mỗi pixel lại kết nối full-connected với 1.000.000 pixels ở layer trước, tức sẽ có 1012 tham số. Đây là còn số quá lớn để có thể tính được vào thời điểm đó, bởi vì với mô hình có nhiều tham số, sẽ dễ bị overfitted và cần lượng lớn data cho việc training, ngoài ra còn cần nhiều memory và năng lực tính toán cho việc training và prediction.

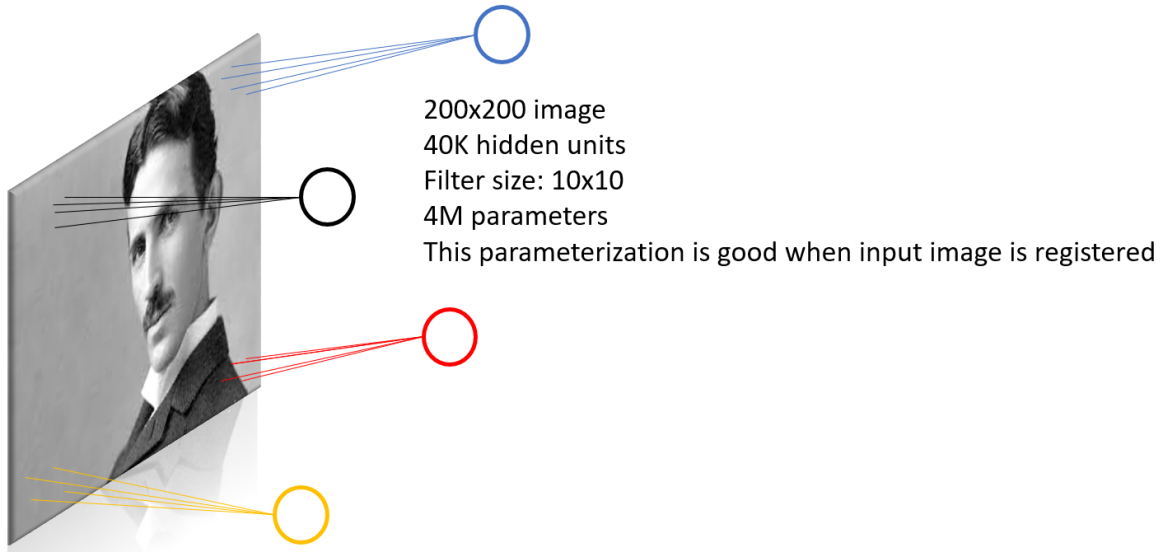
#### 3.1.1.2. CNN hoạt động ra sao?

Có 2 đặc tính của image hình thành nên cách hoạt động của CNN trên image, đó là feature localization và feature independence of location.

Feature localization: mỗi pixel hoặc feature có liên quan với các pixel quanh nó.

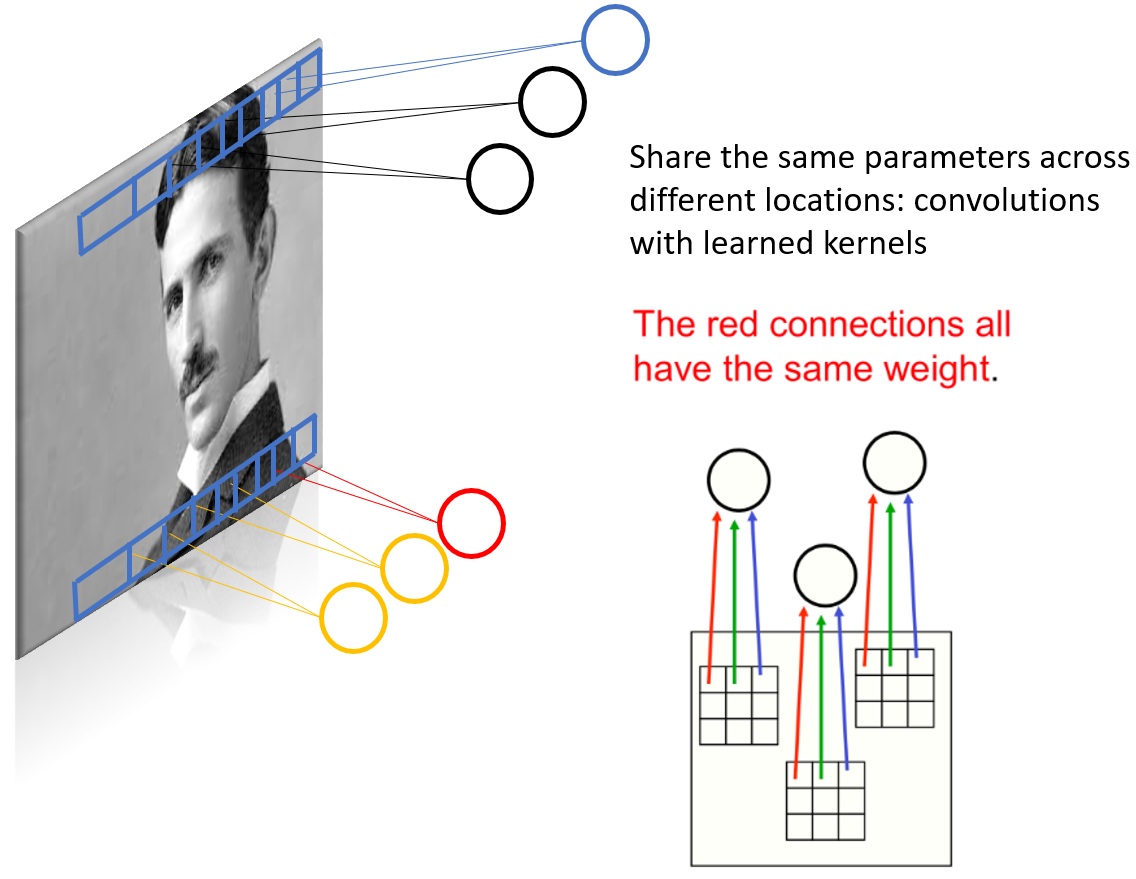
**Feature Independence of location:** mỗi feature dù nó có nằm ở đâu trong bức ảnh, thì nó vẫn mang giá trị của feature đó. CNN xử lý vấn đề có quá nhiều tham số với Shared parameters (feature independence of location) của Locally connected networks (feature loclization), được gọi là Convolution Net.

**Locally connected layer:** Trong hidden layer đầu tiên, mỗi node sẽ kết nối tới một cụm nhỏ pixels của input image chứ không phải toàn bộ image, gọi là small portion. Theo cách này, ta sẽ có ít kết nối hơn, vì thế ít tham số hơn giữa input và hidden layer đầu tiên.



Hình 1: Local connected layer

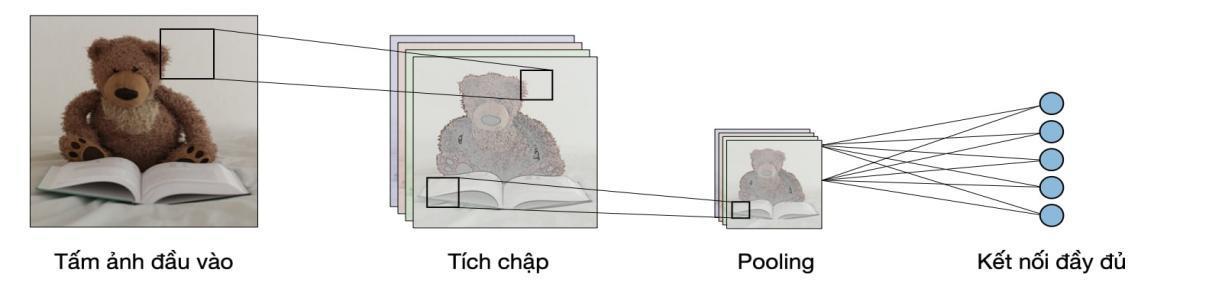
**Shared parameters**: Có nhưng khu vực, mà việc tìm ra feature là giống nhau về cách làm, vì vậy ta có thể dùng chung bộ parameter, trong hình trên là phía phải bên trên và phía trái bên dưới. Tức ta chia sẻ bộ parameter giữa những vị trí khác nhau trong bức ảnh.



Hình 2: Shared parameters

### 3.1.2. Kiến trúc của một mạng CNN

Kiến trúc truyền thống của một mạng CNN - Mạng neural tích chập (Convolutional neural network), còn được biết đến với cái tên CNNs, là một dạng mạng neural được cấu thành bởi các tầng sau:



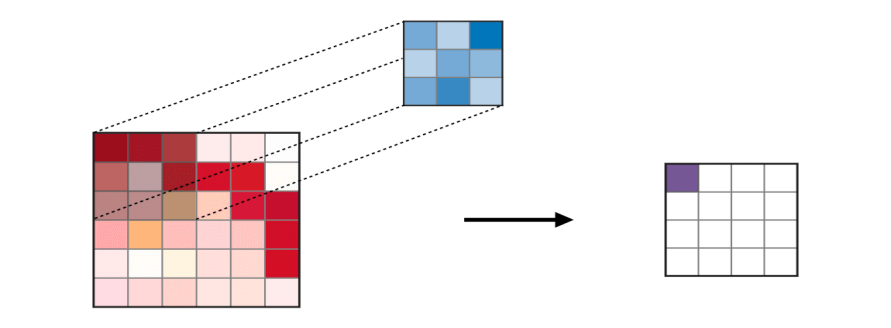
Hình 3: Cấu trúc một mạng CNN

Tầng tích chập và tầng pooling có thể được hiệu chỉnh theo các siêu tham số (hyperparameters) được mô tả ở những phần tiếp theo.

#### 3.1.2.1.Các kiểu tầng

##### 3.1.2.1.1. Tầng tích chập (CONV)

Tầng tích chập(CONV) sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào I theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước và bộ lọc F và độ trượt (stride) S. Kết quả đầu ra O được gọi là feature map hay activation map.

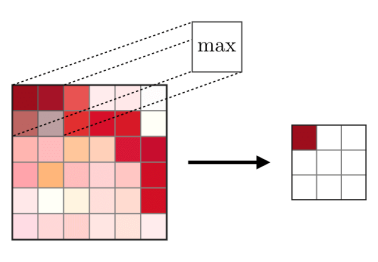
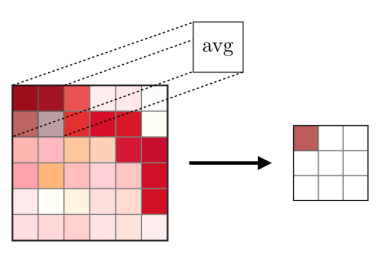


Hình 4: CONV step

*Lưu ý: Bước tích chập cũng có thể được khái quát hóa cả với trường hợp một chiều (1D) và ba chiều (3D).*

##### 3.1.2.1.2. Pooling (POOL)

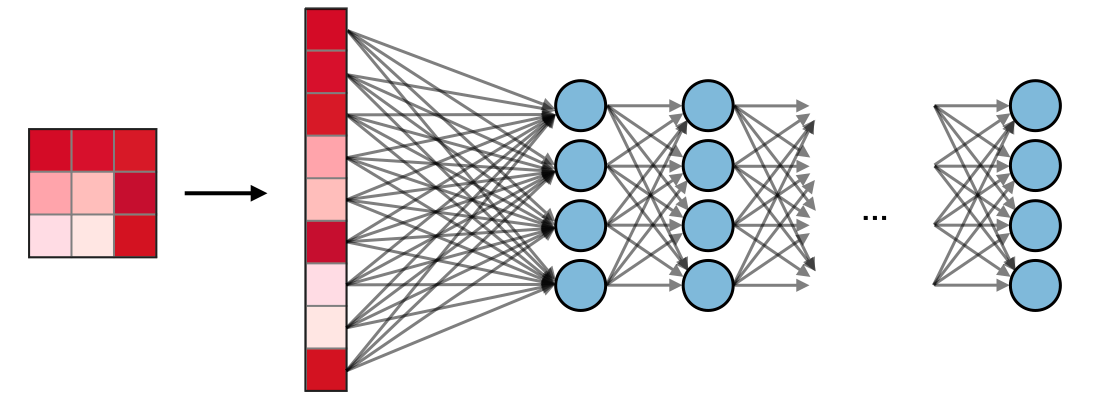
Tầng pooling (POOL) là một phép downsampling, thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Cụ thể, max pooling và average pooling là những dạng pooling đặc biệt, mà tương ứng là trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.

Hình 5: avg pooling Hình 6: max pooling

##### 3.1.2.1.3. Fully Connected (FC)

Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.



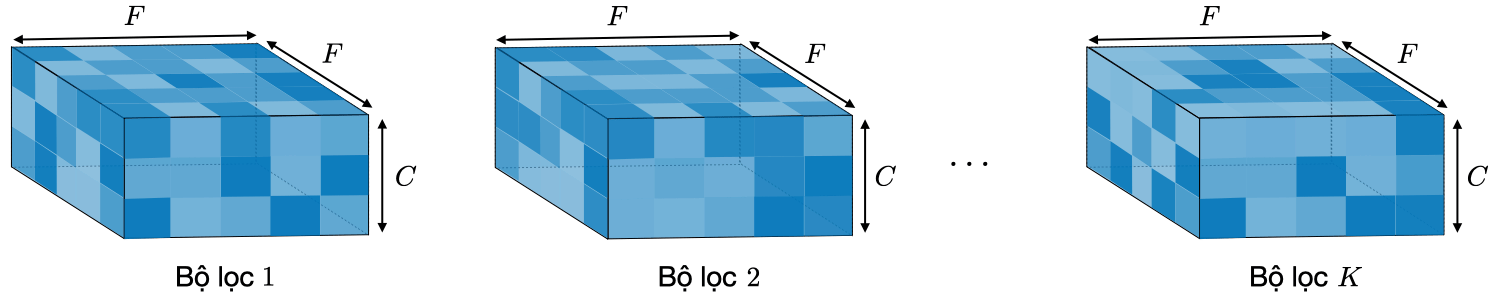
Hình 7: Fully Connected (FC)

### 3.1.3. Các siêu tham số của bộ lọc

Tầng tích chập chứa các bộ lọc mà rất quan trọng cho ta khi biết ý nghĩa đằng sau các siêu tham số của chúng.

#### 3.1.3.1. Các chiều của một bộ lọc

Một bộ lọc kích thước *F* x F áp dụng lên đầu vào chứa C kênh thì có kích thước tổng kể là F x F x C thực hiện phép tích chập trên đầu vào kích thước I x I x C và cho ra một feature map (hay còn gọi là activation map) có kích thước O x O x 1.



Hình 8: Bộ lọc

*Lưu ý: Việc áp dụng K bộ lọc có kích thước F x F cho ra một feature map có kích thước O x O x K.*

#### 3.1.3.2. Stride

Đối với phép tích chập hoặc phép pooling, độ trượt SS ký hiệu số pixel mà cửa sổ sẽ di chuyển sau mỗi lần thực hiện phép tính.



Hình 9: Stride

#### 3.1.3.3. Zero-padding

Zero-padding là tên gọi của quá trình thêm PP số không vào các biên của đầu vào. Giá trị này có thể được lựa chọn thủ công hoặc một cách tự động bằng một trong ba những phương pháp như Valid, Same, Full:

**Valid:**

* Không sử dụng padding
* Bỏ phép tích chập cuối nếu số chiều không khớp

**Same:**

* Sử dụng padding để làm cho feature map có kích thước **wps**
* Kích thước đầu ra thuận lợi về mặt toán học
* Còn được gọi là ‘half’ padding

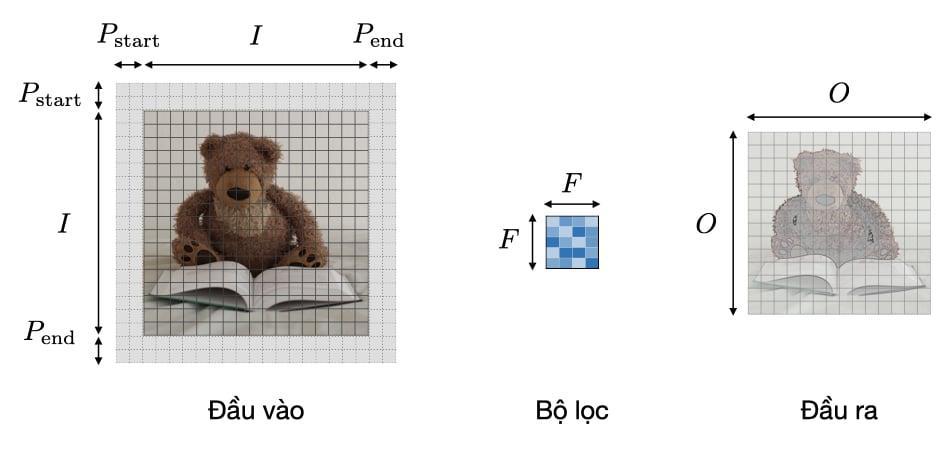
**Full**

* Padding tối đa sao cho các phép tích chập có thể được sử dụng tại các rìa của đầu vào
* Bộ lọc ‘thấy’ được đầu vào từ đầu đến cuối

### 3.1.4. Điều chỉnh siêu tham số

#### 3.1.4.1. Tính tương thích của tham số trong mô hình tích chập

Bằng cách ký hiệu II là độ dài kích thước đầu vào, FF là độ dài của bộ lọc, PP là số lượng zero padding, SS là độ trượt, ta có thể tính được độ dài OO của feature map theo một chiều bằng công thức:



Hình 10: Padding

***Lưu ý:*** *Trong một số trường hợp, Pstart=Pend ≜P, ta có thể thay thế Pstart+Pend bằng $2P trong công thức trên.*

#### 3.1.4.2. Hiểu về độ phức tạp của mô hình

Để đánh giá độ phức tạp của một mô hình, cách hữu hiệu là xác định số tham số mà mô hình đó sẽ có. Trong một tầng của mạng neural tích chập, nó sẽ được tính toán như sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CONV** | **POOL** | **FC** |
| Minh Họa | table-conv | table-pool | table-fc |
| **Kích thước đầu vào** | *I×I×C* | *I×I×C* | *Nin​* |
| **Kích thước đầu ra** | *O×O×K* | *O×O×C* | *Nout​* |
| **Số lượng tham số** | *(F×F×C+1) . K* | *0* | *(Nin​+1) × Nout​* |
| **Lưu ý** | - Một tham số bias với mỗi bộ lọc  - Trong đa số trường hợp, *S < F*  - Một lựa chọn phổ biến cho *K*là *2C* | - Phép pooling được áp dụng lên từng kênh (channel-wise)  - Trong đa số trường hợp, *S = F* | - Đầu vào được làm  phẳng  - Mỗi neuron có một tham số bias  - Số neuron trong một tầng FC phụ thuộc vào ràng buộc kết cấu |

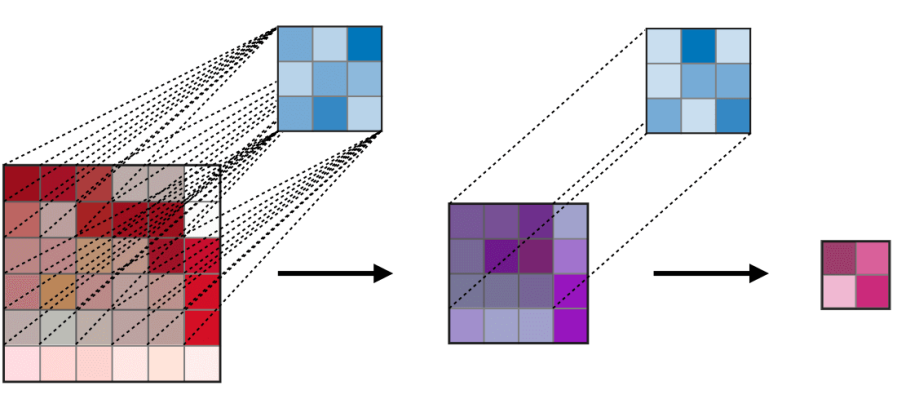
#### 3.1.4.3. Trường Thụ Cảm (receptive field)

Trường thụ cảm (receptive field) tại tầng k là vùng được ký hiệu Rk×Rk của đầu vào mà những pixel của activation map thứ có thể "nhìn thấy". Bằng cách gọi Fj​ là kích thước bộ lọc của tầng j và Si là giá trị độ trượt của tầng i và để thuận tiện, ta mặc định S0=1, trường thụ cảm của tầng được tính toán bằng công thức:



Hình 11: Công thức tính kích thước receptive field tại layer k

*Trong ví dụ bên dưới, ta có F1 = F2 =3 và S1 = S2 = 1, nên cho ra được R2 = 5*



Hình 12: receptive field

### 3.1.5. Các hàm kích hoạt thường gặp (activation funcion)

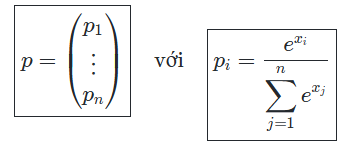
#### 3.1.5.1. Rectified Linear Unit (ReLU)

Tầng rectified linear unit (ReLU) là một hàm kích hoạt g được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở bảng dưới:

|  |  |  |
| --- | --- | --- |
| **ReLU** | **Leaky ReLU** | **ELU** |
| *g*(*z*) = max(0, *z*) | *g*(*z*) = max(*ϵz*, *z*) với ϵ ≪ 1 | *g*(*z*) = max(*α*(*ez* − 1), *z*) với α≪ |
| **relu** | **leaky-relu** | **elu** |
| Độ phức tạp phi tuyến tính có thể thông dịch được về mặt sinh học | Gán vấn đề ReLU chết cho những giá trị âm | Khả vi tại mọi nơi |

#### 3.1.5.2. Softmax

Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các b giá trị*x∈ Rn* và cho ra là một vector gồm các xác suất p ∈Rn thông qua một hàm softmax ở cuối kiến trúc. Nó được định nghĩa như sau:



*Hình 13: Softmax*

## 3.2. Mô hình Resnet

### 3.2.1 Giới thiệu tổng quan

ResNet (Residual Network) là một trong những kiến trúc mạng nơ-ron tích chập (CNN) quan trọng và có ảnh hưởng lớn trong lĩnh vực thị giác máy tính, được đề xuất bởi Kaiming He và cộng sự vào năm 2015. ResNet ra đời nhằm giải quyết degradation problem — hiện tượng khi mô hình càng sâu thì độ chính xác lại giảm, mặc dù về lý thuyết mô hình sâu hơn có khả năng học đặc trưng tốt hơn.

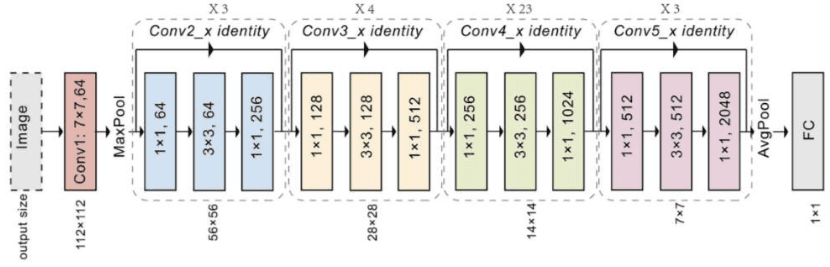
Ý tưởng cốt lõi của ResNet là sử dụng residual learning thông qua các skip connection (kết nối tắt), cho phép tín hiệu đầu vào được truyền trực tiếp sang các tầng sâu hơn mà không bị biến đổi hoàn toàn. Cơ chế này giúp mô hình hạn chế hiện tượng vanishing gradient, ổn định quá trình huấn luyện và cho phép xây dựng mạng rất sâu (lên đến hàng trăm tầng) mà vẫn đạt hiệu quả cao.

Trong nhiều benchmark như ImageNet, ResNet đã đạt kết quả vượt trội và hiện vẫn được xem là kiến trúc nền tảng trong nhiều mô hình hiện đại (FaceNet, ArcFace, YOLO, Mask R-CNN…).

### 3.2.2. Cấu trúc và nguyên lý hoạt động của mô hình resnet

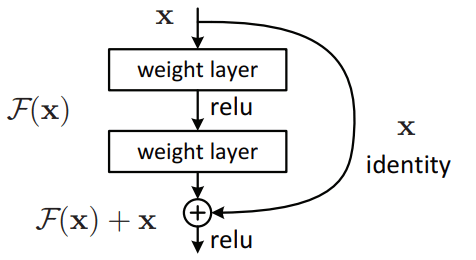
#### 3.2.2.1. Cấu trúc

Cấu trúc của ResNet được xây dựng dựa trên việc chia nhỏ mạng thành các khối residual (Residual Blocks), trong đó mỗi khối bao gồm một số lớp tích chập (convolutional layers) cùng với một kết nối tắt trực tiếp từ đầu vào đến đầu ra của khối. Ý tưởng chính là việc học một hàm residual F(x) thay vì học trực tiếp một hàm mục tiêu H(x) . Điều này giúp duy trì tính ổn định và độ chính xác khi mạng có nhiều tầng.



Hình 14: Mô hình Resnet

#### 3.2.2.2. Khối residual cơ bản:



Hình 15: Khối residual cơ bản

Một khối residual cơ bản trong ResNet có cấu trúc như sau:

*y = F(x, {W\_i}) + x*

Trong đó:

* x là đầu vào của khối.
* y là đầu ra của khối.
* F(x, {W\_i}) là hàm residual, thường được biểu diễn bằng một chuỗi các lớp tích chập với trọng số W\_i.

**3.2.2.2. Nguyên lý hoạt động**

Nguyên lý hoạt động của ResNet dựa trên việc thay vì học trực tiếp hàm H(x) , mô hình sẽ học một hàm residual F(x) = H(x) – x. Sau đó, đầu ra của mô hình sẽ là: H(x) = F(x) + x

Cách tiếp cận này dựa trên giả định rằng việc học một hàm sai khác F(x) sẽ dễ dàng hơn so với việc học trực tiếp hàm H(x). Khi mô hình học đượcF(x), nó thực chất đang học cách “điều chỉnh” đầu vào x để đạt được đầu ra mong muốn H(x). Nhờ vào các kết nối tắt (skip connections), các tín hiệu thông tin có thể dễ dàng được truyền qua mạng mà không bị suy giảm hoặc mất mát, đặc biệt khi mạng trở nên rất sâu.

Các kết nối tắt trong ResNet mang lại nhiều lợi ích, bao gồm:

* Giảm thiểu vanishing gradient: Khi mạng trở nên rất sâu, các giá trị gradient có thể bị giảm mạnh (vanishing) hoặc tăng mạnh (exploding), gây khó khăn trong quá trình huấn luyện. Kết nối tắt giúp giữ lại các giá trị gradient, giảm thiểu hiện tượng này.
* Dễ dàng học các đặc trưng: Việc học một hàm residual F(x) thường đơn giản hơn so với việc học một hàm ánh xạ hoàn toàn mới H(x), do đó mô hình có thể nhanh chóng đạt được độ chính xác cao hơn.
* Tăng độ sâu mà không giảm hiệu suất: Với các kết nối tắt, ResNet có thể được mở rộng đến hàng trăm hoặc hàng ngàn tầng mà không gặp phải vấn đề suy giảm hiệu suất.

**3.2.3. Kiến trúc chi tiết của mô hình ResNet-34**

ResNet-34 là phiên bản mở rộng của ResNet-18, với 34 tầng. Cấu trúc này cung cấp nhiều khả năng biểu diễn hơn so với ResNet-18 nhưng vẫn duy trì độ phức tạp tính toán ở mức hợp lý. Các lớp trong ResNet-34 bao gồm:

* Conv1: Tầng tích chập đầu tiên với kernel size 7×7, stride 2, và 64 filters, theo sau là Batch Normalization và ReLU activation.
* MaxPool: Lớp MaxPooling với kernel size 3×3 và stride 2.
* Conv2\_x: Gồm 3 khối residual, mỗi khối có 2 tầng tích chập với 64 filters, stride 1.
* Conv3\_x: Gồm 4 khối residual, mỗi khối có 2 tầng tích chập với 128 filters, stride 2.
* Conv4\_x: Gồm 6 khối residual, mỗi khối có 2 tầng tích chập với 256 filters, stride 2.
* Conv5\_x: Gồm 3 khối residual, mỗi khối có 2 tầng tích chập với 512 filters, stride 2.
* Average Pooling: Lớp pooling trung bình, giảm chiều dữ liệu xuống 1×1.
* Fully Connected Layer: Lớp kết nối đầy đủ với số lượng neurons bằng số lớp đầu ra (output classes).

ResNet-34 là một lựa chọn phổ biến cho các bài toán yêu cầu khả năng phân loại tốt hơn so với ResNet-18 mà vẫn giữ được thời gian huấn luyện ở mức chấp nhận được.

### 3.2.3. Ưu điểm và hạn chế

#### 3.2.3.1. Ưu điểm:

* Giảm vanishing gradient hiệu quả: nhờ skip connection, gradient có thể quay ngược về lớp đầu mà không bị mất mát.
* Huấn luyện nhanh và ổn định: đặc biệt trên dataset vừa và nhỏ.
* Khả năng trích xuất đặc trưng khuôn mặt tốt: nhờ chiều sâu 34 tầng.
* Số lượng tham số vừa phải: dễ triển khai trên máy tính doanh nghiệp, edge device, camera AI.
* Tổng quát hóa tốt: phù hợp nhiều môi trường ánh sáng và góc chụp khác nhau.

#### 3.2.3.2. Hạn chế:

* Chính xác thấp hơn ResNet-50/101 trên dataset cực lớn.
* Basic block không tối ưu cho embedding độ phân biệt cao như ArcFace.
* Khi input nhiễu mạnh (che mặt, khẩu trang), cần thêm augmentation hoặc fine-tuning.
* Vẫn yêu cầu GPU để huấn luyện hiệu quả.

### 3.2.4. Ứng dụng trong dự án nhận diện khuôn mặt quản lý nhân công

Trong đề tài, nhóm lựa chọn ResNet-34 làm backbone trích xuất đặc trưng khuôn mặt nhờ các lý do:

* Hiệu năng – tốc độ cân bằng phù hợp hệ thống điểm danh thời gian thực.
* Mô hình không quá nặng có thể chạy trên GPU tầm trung hoặc server nội bộ doanh nghiệp.
* Khả năng phân biệt khuôn mặt tốt nhờ biểu diễn embedding ổn định, giúp tăng độ chính xác xác thực nhân viên.
* Dễ fine-tuning trên dataset khuôn mặt của doanh nghiệp để cải thiện độ nhận diện.
* Thuận lợi triển khai thực tế khi tích hợp vào camera an ninh, cổng ra vào, phần mềm quản lý nhân sự.

Quy trình sử dụng trong dự án:

1. Tiền xử lý ảnh khuôn mặt (phát hiện – cắt – căn chỉnh).
2. Chuẩn hóa và đưa vào ResNet-34.
3. Mô hình tạo vector embedding 128D/512D.
4. So sánh embedding bằng cosine similarity → xác định danh tính nhân công.
5. Ghi nhận điểm danh / thời gian vào-ra.

Nhờ ResNet-34, hệ thống đạt độ chính xác cao, giảm lỗi nhận diện nhầm và hoạt động ổn định trong điều kiện ánh sáng và góc chụp đa dạng tại môi trường doanh nghiệp.

**3.3. Thuật toán dlib\_face\_recognition\_resnet\_model**

**3.3.1. Tổng quan**

Dlib\_face\_recognition\_resnet\_model\_v1 là mô hình nhận diện khuôn mặt được cung cấp dưới dạng mô hình tiền huấn luyện trong thư viện dlib. Mô hình ánh xạ mỗi ảnh khuôn mặt đã được căn chỉnh (aligned face) sang một vector đặc trưng 128 chiều (embedding), sao cho các ảnh cùng một người có embeddings gần nhau trong không gian Euclid/cosine, còn ảnh của người khác thì cách xa. Mô hình này thường dùng để thực hiện so khớp (verification/recognition) bằng cách so sánh khoảng cách giữa các vector.

**3.3.2. Kiến trúc và nguyên tắc huấn luyện**

Kiến trúc: Mô hình dựa trên một biến thể ResNet (ResNet-style deep convolutional network) thiết kế để xuất ra vector 128-D cho mỗi khuôn mặt. Dlib cung cấp file mô hình (dlib\_face\_recognition\_resnet\_model\_v1.dat) qua kho lưu trữ dlib-models.

Nguyên tắc huấn luyện: Mô hình được huấn luyện để tối thiểu hóa khoảng cách giữa embeddings của cùng một người và tối đa hóa khoảng cách giữa embeddings của khác người. Trong thực tế, các kỹ thuật huấn luyện như triplet loss / tương tự margin-based losses được sử dụng để học embedding khuôn mặt. Tài liệu và hướng dẫn tham khảo cũng nêu rõ mô hình học ra embedding 128-D sử dụng dữ liệu mặt lớn và các chiến lược đào tạo để đạt tính phân biệt.

**3.3.3. Cách hoạt động (Inference pipeline)**

1. Phát hiện khuôn mặt (Face detection): phát hiện vùng chứa mặt trên ảnh (ví dụ HOG, MMOD của dlib, hoặc MTCNN/SSD).
2. Căn chỉnh khuôn mặt (Face alignment): dùng bộ dự đoán landmark để xoay/crop khuôn mặt theo chuẩn, tăng độ ổn định của embedding.
3. Trích xuất embedding: feed ảnh khuôn mặt đã căn chỉnh vào mô hình ResNet của dlib để lấy vector 128-D.
4. So sánh & quyết định: so sánh khoảng cách Euclidean (hoặc cosine) giữa embeddings. Nếu khoảng cách nhỏ hơn ngưỡng (ví dụ phổ biến ~0.6 cho Euclidean trong nhiều hướng dẫn), hai ảnh được coi là cùng một người; ngược lại là khác người.

**3.3.4. Ưu điểm**

* Sử dụng mô hình tiền huấn luyện mạnh: có thể đưa vào ứng dụng ngay, không cần huấn luyện lại từ đầu.
* Embeddings 128-D gọn và hiệu quả: dễ lưu trữ, so sánh nhanh, phù hợp cho hệ thống quản lý nhân sự cần tra cứu nhanh.
* Khả năng khái quát tốt: khi được huấn luyện trên tập lớn, mô hình nhận diện tốt trong nhiều điều kiện ánh sáng/biểu cảm (tất nhiên còn tùy dữ liệu huấn luyện và chất lượng ảnh đầu vào).

**3.3.5. Hạn chế và lưu ý**

* Không hoàn toàn bất biến với ảnh kém chất lượng: mô hình có thể suy giảm với ảnh mờ, quá nhiễu hoặc độ phân giải rất thấp; một số so sánh cho thấy FaceNet/ArcFace có thể xử lý tốt hơn trong điều kiện cụ thể. Vì vậy cần tiền xử lý tốt hoặc cân nhắc mô hình khác khi ảnh thực tế quá xấu.
* Ngưỡng quyết định cần hiệu chỉnh: giá trị ngưỡng (threshold) cho khoảng cách Euclidean nên được hiệu chỉnh dựa trên tập dữ liệu thực tế của hệ thống (trade-off giữa false accept và false reject).
* Việc huấn luyện lại mô hình đòi hỏi tài nguyên lớn: dlib cung cấp file .dat tiền huấn luyện; nếu cần huấn luyện từ đầu hoặc fine-tune, người thực hiện cần tập dữ liệu lớn và chi phí tính toán. Trên GitHub/dlib có nhiều thảo luận liên quan tới cách huấn luyện lại mô hình.

## 3.3. Thuật toán dlib\_face\_recognition\_resnet\_model

### 3.3.1. Tổng quan

Dlib\_face\_recognition\_resnet\_model\_v1 là mô hình nhận diện khuôn mặt được cung cấp dưới dạng mô hình tiền huấn luyện trong thư viện dlib. Mô hình ánh xạ mỗi ảnh khuôn mặt đã được căn chỉnh (aligned face) sang một vector đặc trưng 128 chiều (embedding), sao cho các ảnh cùng một người có embeddings gần nhau trong không gian Euclid/cosine, còn ảnh của người khác thì cách xa. Mô hình này thường dùng để thực hiện so khớp (verification/recognition) bằng cách so sánh khoảng cách giữa các vector.

### 3.3.2. Kiến trúc và nguyên tắc huấn luyện

Kiến trúc: Mô hình dựa trên một biến thể ResNet (ResNet-style deep convolutional network) thiết kế để xuất ra vector 128-D cho mỗi khuôn mặt. Dlib cung cấp file mô hình (dlib\_face\_recognition\_resnet\_model\_v1.dat) qua kho lưu trữ dlib-models.

Nguyên tắc huấn luyện: Mô hình được huấn luyện để tối thiểu hóa khoảng cách giữa embeddings của cùng một người và tối đa hóa khoảng cách giữa embeddings của khác người. Trong thực tế, các kỹ thuật huấn luyện như triplet loss / tương tự margin-based losses được sử dụng để học embedding khuôn mặt. Tài liệu và hướng dẫn tham khảo cũng nêu rõ mô hình học ra embedding 128-D sử dụng dữ liệu mặt lớn và các chiến lược đào tạo để đạt tính phân biệt.

### 3.3.3. Cách hoạt động (Inference pipeline)

1. Phát hiện khuôn mặt (Face detection): phát hiện vùng chứa mặt trên ảnh (ví dụ HOG, MMOD của dlib, hoặc MTCNN/SSD).
2. Căn chỉnh khuôn mặt (Face alignment): dùng bộ dự đoán landmark để xoay/crop khuôn mặt theo chuẩn, tăng độ ổn định của embedding.
3. Trích xuất embedding: feed ảnh khuôn mặt đã căn chỉnh vào mô hình ResNet của dlib để lấy vector 128-D.
4. So sánh & quyết định: so sánh khoảng cách Euclidean (hoặc cosine) giữa embeddings. Nếu khoảng cách nhỏ hơn ngưỡng (ví dụ phổ biến ~0.6 cho Euclidean trong nhiều hướng dẫn), hai ảnh được coi là cùng một người; ngược lại là khác người.

### 3.3.4. Ưu điểm

* Sử dụng mô hình tiền huấn luyện mạnh: có thể đưa vào ứng dụng ngay, không cần huấn luyện lại từ đầu.
* Embeddings 128-D gọn và hiệu quả: dễ lưu trữ, so sánh nhanh, phù hợp cho hệ thống quản lý nhân sự cần tra cứu nhanh.
* Khả năng khái quát tốt: khi được huấn luyện trên tập lớn, mô hình nhận diện tốt trong nhiều điều kiện ánh sáng/biểu cảm (tất nhiên còn tùy dữ liệu huấn luyện và chất lượng ảnh đầu vào).

### 3.3.5. Hạn chế và lưu ý

* Không hoàn toàn bất biến với ảnh kém chất lượng: mô hình có thể suy giảm với ảnh mờ, quá nhiễu hoặc độ phân giải rất thấp; một số so sánh cho thấy FaceNet/ArcFace có thể xử lý tốt hơn trong điều kiện cụ thể. Vì vậy cần tiền xử lý tốt hoặc cân nhắc mô hình khác khi ảnh thực tế quá xấu.
* Ngưỡng quyết định cần hiệu chỉnh: giá trị ngưỡng (threshold) cho khoảng cách Euclidean nên được hiệu chỉnh dựa trên tập dữ liệu thực tế của hệ thống (trade-off giữa false accept và false reject).
* Việc huấn luyện lại mô hình đòi hỏi tài nguyên lớn: dlib cung cấp file .dat tiền huấn luyện; nếu cần huấn luyện từ đầu hoặc fine-tune, người thực hiện cần tập dữ liệu lớn và chi phí tính toán. Trên GitHub/dlib có nhiều thảo luận liên quan tới cách huấn luyện lại mô hình.

## 3.4. Thuật toán HOG

### 3.4.1. Thuật toán HOG (Histogram of Oriented Gradients)

HOG là viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.

Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directins) để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là “tế bào” (cells) và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn.

Có 5 bước cơ bản để xây dựng một vector HOG cho hình ảnh, bao gồm:

* Tiền xử lý
* Tính gradient
* Tính vector đặc trưng cho từng ô (cells)
* Chuẩn hóa khối (blocks)
* Tính toán vector HOG

#### 3.4.1.1 Tiền xử lý

Để thuật toán hoạt động hiệu quả và giảm thiểu nhiễu:Resize: Ảnh đầu vào thường được đưa về một kích thước chuẩn để đảm bảo tính nhất quán của vector đặc trưng.

Grayscale: Chuyển đổi ảnh sang không gian màu xám. Việc loại bỏ thông tin màu sắc giúp giảm khối lượng tính toán xuống 1/3 (từ 3 kênh RGB xuống 1 kênh), đồng thời giúp thuật toán tập trung hoàn toàn vào cấu trúc hình học thay vì bị phân tán bởi màu sắc.

#### 3.4.1.2 Tính Gradient

Đây là bước nền tảng để xác định các cạnh (edges) và đường nét khuôn mặt. Tại mỗi điểm ảnh (x, y), thuật toán tính toán đạo hàm (sự thay đổi độ sáng) theo phương ngang (x) và phương dọc (y).

Sử dụng toán tử Sobel hoặc đơn giản là hiệu số giữa các pixel lân cận :

Gradient theo phương x: 

Gradient theo phương y:

Từ hai giá trị này, ta xác định được Cường độ (Magnitude) và Góc hướng (Orientation) của gradient tại mỗi pixel:

Cường độ gradient (m):



Góc hướng gradient (): 

Ý nghĩa: Những vùng phẳng (như da trán mịn) sẽ có cường độ gradient nhỏ, trong khi những vùng biên (như viền mắt, môi) sẽ có cường độ gradient lớn.

#### 3.4.1.3. Xây dựng biểu đồ Histogram trong các Cell

Việc tính gradient cho từng pixel là quá chi tiết và nhạy cảm với nhiễu. Để khắc phục, ảnh được chia thành các ô nhỏ gọi là Cell (kích thước tiêu chuẩn 8 x 8 pixels).

Trong mỗi Cell, một biểu đồ Histogram hướng gradient (Histogram of Oriented Gradients) được xây dựng:

* Histogram gồm 9 bins, tương ứng với các góc từ 0∘ đến 180∘ (mỗi bin rộng 20∘).
* Mỗi pixel trong Cell sẽ "bỏ phiếu" vào bin tương ứng với góc hướng () của nó.
* Trọng số phiếu bầu: Chính là giá trị cường độ gradient (m). Pixel nào có gradient càng mạnh (cạnh càng sắc nét) thì đóng góp vào biểu đồ càng lớn.

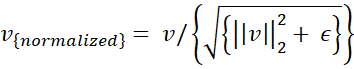
Kết quả: Mỗi Cell 8 x 8 (64 pixels) được tóm tắt lại thành một vector 9 chiều (9 bins), đại diện cho hình dạng cục bộ của vùng đó.

#### 3.4.1.4. Chuẩn hóa khối (Block Normalization)

Độ sáng của ảnh phụ thuộc nhiều vào điều kiện môi trường (chụp ngoài trời, trong bóng râm). Cùng một khuôn mặt nhưng độ tương phản khác nhau sẽ làm thay đổi giá trị độ lớn gradient.

Để giải quyết vấn đề này, thuật toán thực hiện chuẩn hóa theo các Block:

* Mỗi Block bao gồm 2 x 2 Cells (tức 16 x 16 pixels).
* Các Block này được trượt khắp ảnh (có sự chồng lấn - overlapping, thường là 50%).
* Trong mỗi Block, ta ghép 4 vector của 4 Cells lại thành một vector dài 36 chiều (4 cells x 9 bins ).Áp dụng chuẩn hóa L2-Norm để đưa vector về đơn vị chuẩn:



(với ε là hằng số nhỏ để tránh chia cho 0)

#### 3.4.1.5. Tổng hợp Vector đặc trưng:

Sau khi trượt Block qua toàn bộ bức ảnh, tất cả các vector đã chuẩn hóa được nối lại với nhau (concatenate) để tạo thành một Vector đặc trưng HOG (HOG Descriptor) duy nhất.

Vector này chứa đựng thông tin toàn cục về cấu trúc hình học của khuôn mặt, loại bỏ các yếu tố nhiễu về ánh sáng và màu sắc. Vector này sau đó được đưa vào bộ phân lớp Linear SVM (Support Vector Machine) để quyết định xem vùng ảnh đó có chứa khuôn mặt hay không. SVM thực hiện phép nhân vô hướng vector này với vector trọng số đã được huấn luyện (w) và so sánh với ngưỡng (b). Nếu wᵀ ⋅ x + b > 0 → Có khuôn mặt. Nếu wᵀ ⋅ x + b < 0 → Không phải khuôn mặt (Nền).

### 3.4.2. Ưu điểm và nhược điểm của thuật toán

Ưu điểm:

* Mạnh mẽ về hình dáng: Rất giỏi trong việc bắt các cạnh và hình dáng đặc trưng (như dáng người đứng).
* Chống nhiễu ánh sáng: Nhờ bước chuẩn hóa block, nó hoạt động tốt trong các điều kiện ánh sáng khác nhau.
* Bất biến với biến đổi hình học nhỏ: Do sử dụng histogram trên các cell nhỏ, nếu vật thể dịch chuyển nhẹ vài pixel, feature vector không thay đổi quá nhiều.

Nhược điểm:

* Tốc độ: Tính toán khá chậm so với các thuật toán đơn giản hơn (như Haar Cascade) do phải tính gradient và histogram cho từng pixel.
* Không xử lý tốt việc xoay: Nếu vật thể bị xoay nghiêng đáng kể, HOG sẽ hoạt động kém (trừ khi ta train dữ liệu với các ảnh xoay).
* Cổ điển: Hiện nay Deep Learning (CNN) thường vượt trội hơn HOG về độ chính xác, dù HOG vẫn hữu ích trong các hệ thống tài nguyên thấp hoặc dữ liệu ít.

## 3.5. Dữ liệu và tiền xử lí:

### 3.5.1 Giới thiệu dữ liệu

#### 3.5.1.1. Nguồn dữ liệu

Dữ liệu ảnh khuôn mặt được thu thập từ các nguồn sau (liệt kê theo thứ tự ưu tiên):

* Ảnh chụp trực tiếp bằng thiết bị của nhóm nghiên cứu trong các buổi thu thập dữ liệu (điều kiện ánh sáng tự nhiên và nhân tạo).
* Bộ dữ liệu công khai được phép sử dụng (nếu có), đã qua kiểm định về bản quyền và quyền riêng tư.
* Ảnh do người tham gia cung cấp thông qua biểu mẫu đồng ý tham gia nghiên cứu

Định dạng dữ liệu:

* Định dạng tập tin: .jpg, .png
* Kích thước ảnh nguyên gốc: (640 x 480)

Mục tiêu của dữ liệu:

* Huấn luyện mô hình nhận diện danh tính (face recognition) — biến phân loại hoặc embedding học để so khớp khuôn mặt.
* Đánh giá độ chính xác, độ nhạy và tính ổn định của thuật toán khi gặp điều kiện ánh sáng và background khác nhau.
* Nghiên cứu sự ảnh hưởng của tiền xử lý (chuẩn hóa, căn chỉnh) và augmentations lên hiệu năng mô hình.

#### 3.5.1.2 Mô tả đặc điệm dữ liệu

Đa dạng về điều kiện ánh sáng: ảnh được chụp trong nhiều điều kiện (ánh sáng mạnh, ánh sáng yếu, một phần bị chói lóa, nguồn sáng phía trước, phía sau hoặc lệch góc), điều này tạo ra thách thức cho thuật toán do thay đổi biểu hiện vùng sáng-tối trên khuôn mặt.

Góc chụp khác nhau: bao gồm ảnh chính diện, nghiêng trái/phải, một vài ảnh cận cảnh hoặc chụp từ xa; góc lệch có thể làm thay đổi tỷ lệ biểu diễn đặc trưng.

Background phức tạp: một số ảnh có nền đông người, vật thể gây nhiễu, họa tiết phong phú hoặc ánh sáng nền không đồng nhất — làm khó bước phát hiện và tách nền.

### 3.5.1.3. Các bước tiền xử lí

Loại bỏ ảnh hỏng: ảnh corrupt, ảnh có lỗi nén, hoặc kích thước quá nhỏ bị loại bỏ.

Lọc theo ánh sáng: giữ lại các ảnh có ánh sáng phù hợp — loại bỏ hoặc đánh nhãn ảnh bị lóa, quá tối, hoặc vùng khuôn mặt bị over-exposed. (Trong báo cáo này, bước lọc ban đầu đã thực hiện: “Lọc ra các bức ảnh có ánh sáng phù hợp không bị lóa”.

Lọc theo độ nét: loại bỏ ảnh quá mờ bằng chỉ số laplacian variance hoặc phương pháp đo độ sắc nét.

CHƯƠNG IV: SO SÁNH THUẬT TOÁN HOG VÀ THUẬT TOÁN

## 4.1. Tổng quan mục tiêu so sánh

Trong bài toán quản lý nhân công bằng nhận diện khuôn mặt, hệ thống cần:

Độ chính xác cao trong điều kiện thực tế (ánh sáng, góc chụp, mệt mỏi, di chuyển…). Nhận diện được nhiều người khác nhau trong tập lớn. Ổn định theo thời gian (đeo khẩu trang, thay kiểu tóc, già hóa…). Tốc độ xử lý đáp ứng thời gian thực. Dễ triển khai, mở rộng và tích hợp vào hệ thống doanh nghiệp. Do đó, cần phân tích ưu – nhược điểm của hai thuật toán phổ biến trong dlib:

(1) HOG + SVM (đặc trưng thủ công truyền thống)

(2) dlib\_face\_recognition\_resnet (mạng sâu CNN — ResNet)

## 4.2. Mục đích sử dụng của hai thuật toán

|  |  |
| --- | --- |
| Thuật toán | Chức năng chính |
| HOG | Phát hiện khuôn mặt dựa trên đặc trưng cạnh & hình dạng |
| dlib\_face\_recognition\_resnet | Nhận diện/so khớp danh tính dựa trên embedding khuôn mặt |

HOG chủ yếu dùng cho face detection, trong khi ResNet dùng cho face recognition. Trong hệ thống thực tế, HOG thường là bước đầu, còn ResNet đảm nhiệm nhận diện.

## 4.3. So sánh chi tiết

|  |  |  |
| --- | --- | --- |
| **Tiêu chí so sánh** | **HOG** | **Dlib\_face\_recognition\_resnet** |
| **Loại thuật toán** | **Truyền thống (feature descriptor thủ công)** | **Deep Learning, mô hình CNN (ResNet)** |
| **Mục đích chính** | **Phát hiện khuôn mặt (Face Detection)** | **Nhận diện/so khớp khuôn mặt (Face Recognition)** |
| **Cơ chế hoạt động** | **Trích gradient, histogram hướng cạnh, chuẩn hóa block** | **Học tự động đặc trưng bằng mạng sâu → sinh vector embedding 128-D** |
| **Đặc trưng trích xuất** | **Thủ công, chỉ dựa vào cạnh và biên** | **Tự động học các đặc trưng cấp cao, trừu tượng, bền vững** |
| **Khả năng phân biệt người** | **Thấp – không phải mô hình nhận diện** | **Rất cao, embedding 128D tối ưu để phân biệt** |
| **Độ chính xác trong môi trường thực tế** | **Trung bình, dễ sai khi ánh sáng đổi hoặc mặt nghiêng** | **Cao, ổn định dưới nhiều điều kiện (ánh sáng, biểu cảm, góc chụp)** |
| **Độ nhạy với góc nghiêng** | **Nhạy, dễ lỗi nếu mặt nghiêng > 15–20°** | **Tốt nhờ bước alignment trước khi encode** |
| **Tốc độ xử lý** | **Nhanh trên CPU, phù hợp máy yếu** | **Chậm hơn, cần GPU để đạt realtime** |
| **Ứng dụng phù hợp** | **Camera rẻ, hệ thống chỉ cần phát hiện** | **Hệ thống chấm công, kiểm soát nhân công, yêu cầu chính xác cao** |
| **Ưu điểm nổi bật** | **Nhanh, nhẹ, không cần GPU** | **Độ chính xác cao, dùng được mô hình pre-trained, dễ tích hợp** |
| **Nhược điểm nổi bật** | **Không dùng để nhận diện, yếu trong môi trường thật** | **Chậm hơn, yêu cầu phần cứng mạnh hơn** |

Với bài toán nhận diện khuôn mặt để quản lý nhân công, yêu cầu độ chính xác cao, khả năng chịu biến thiên ánh sáng – góc chụp – biểu cảm và tính mở rộng tốt theo số lượng nhân viên, mô hình dlib\_face\_recognition\_resnet vượt trội hoàn toàn và là lựa chọn phù hợp. HOG vẫn có thể sử dụng ở bước phát hiện khuôn mặt, nhưng không đủ mạnh để đảm nhiệm vai trò nhận diện.

Vì vậy, hệ thống trong đề tài lựa chọn dlib\_resnet 128D làm thuật toán cốt lõi để đảm bảo tính chính xác, ổn định và hiệu quả trong môi trường ứng dụng thực tế.

CHƯƠNG V: SERVING MODEL

## 5.1. Giới thiệu về Serving Model

Trong quy trình phát triển các hệ thống Trí tuệ nhân tạo (AI), giai đoạn "Model Serving" đóng vai trò chuyển đổi các thuật toán nhận diện từ môi trường thử nghiệm sang môi trường thực tế. Mục tiêu của chương này là xây dựng một hệ thống Backend Service (API) có khả năng tiếp nhận yêu cầu từ các thiết bị đầu cuối (camera giám sát, máy chấm công), xử lý hình ảnh thông qua mô hình CNN và trả về kết quả định danh nhân viên.

Hệ thống được thiết kế theo kiến trúc Microservice, tách biệt lớp xử lý logic nhận diện và lớp lưu trữ dữ liệu, đảm bảo tính mở rộng và dễ dàng bảo trì. Cấu trúc mã nguồn của module serving được tổ chức trong thư mục src/, bao gồm các thành phần chính: api (giao tiếp), db (lưu trữ), và search (tìm kiếm vector).

## 5.2. Kiến trúc API Service

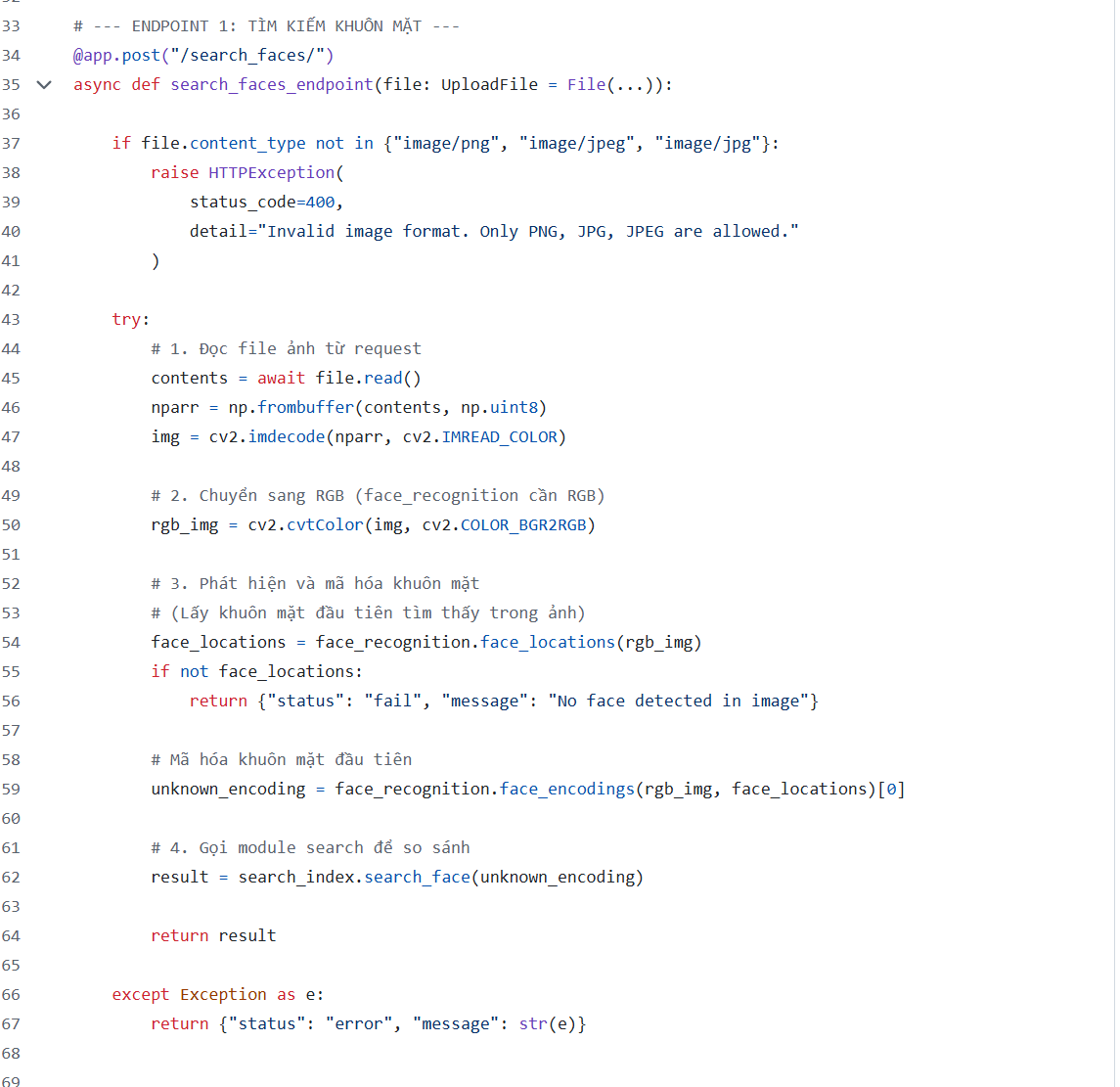
### 5.2.1. Khởi tạo FastAPI App

**File:**api.py



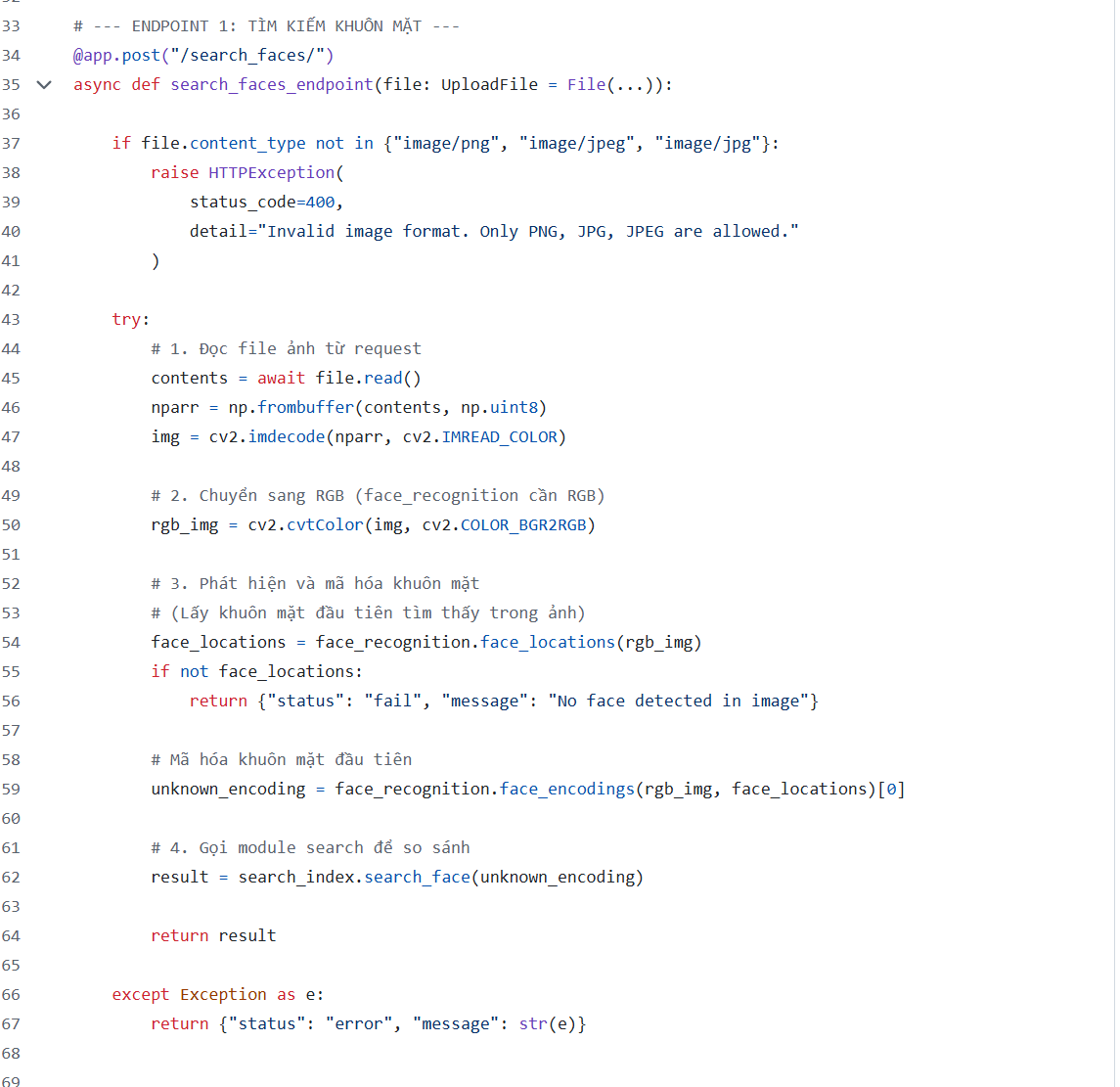
### 5.2.2. Endpoint nhận diện chính

**File:**api.py**- Endpoint**/recognize



### 5.2.3. . Endpoint tìm kiếm khuôn mặt

**File:**api.py**- Endpoint**/ /search\_faces



### 5.2.4. Endpoint thêm người dùng

**File:**api.py**- Endpoint**/persons



### 5.2.5. Endpoint reload cache

**File:**api.py**- Endpoint**/reload\_db



## 5.3. Kiến trúc Search Index

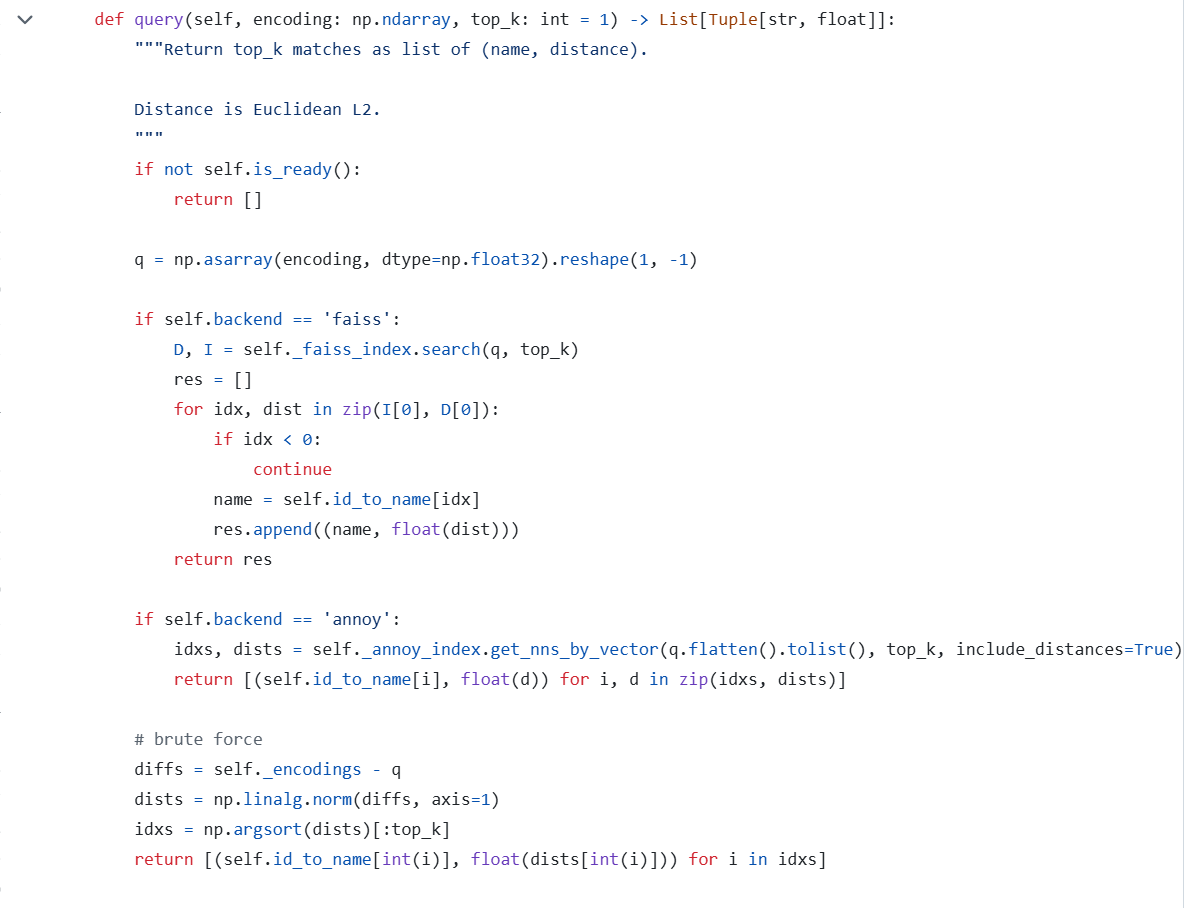
### 5.3.1. Lớp SearchIndex với FAISS/Annoy

**File:**search\_index.py



### 5.3.2. Query method

**File:**search\_index.py



## 5.4. Database Layer

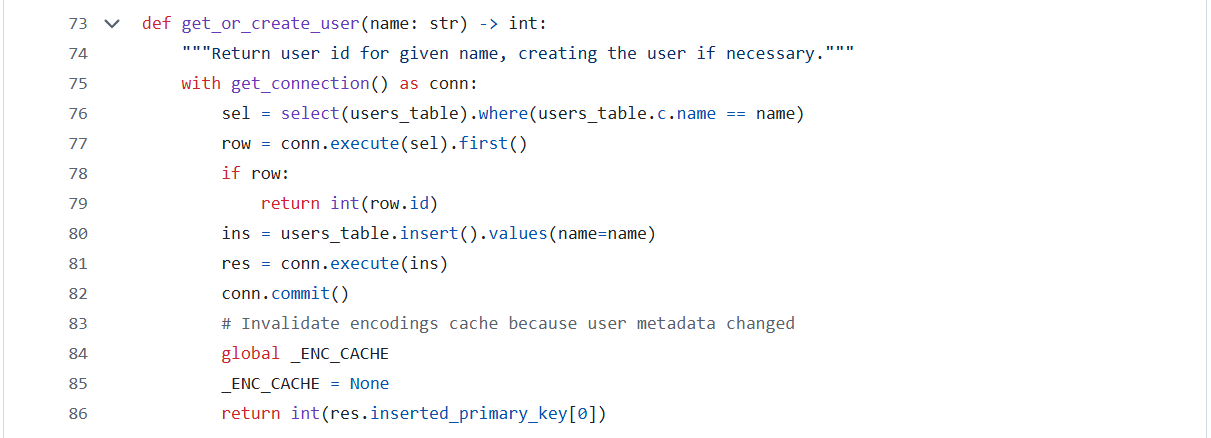
### 5.4.1. Kết nối database

**File:**db.py

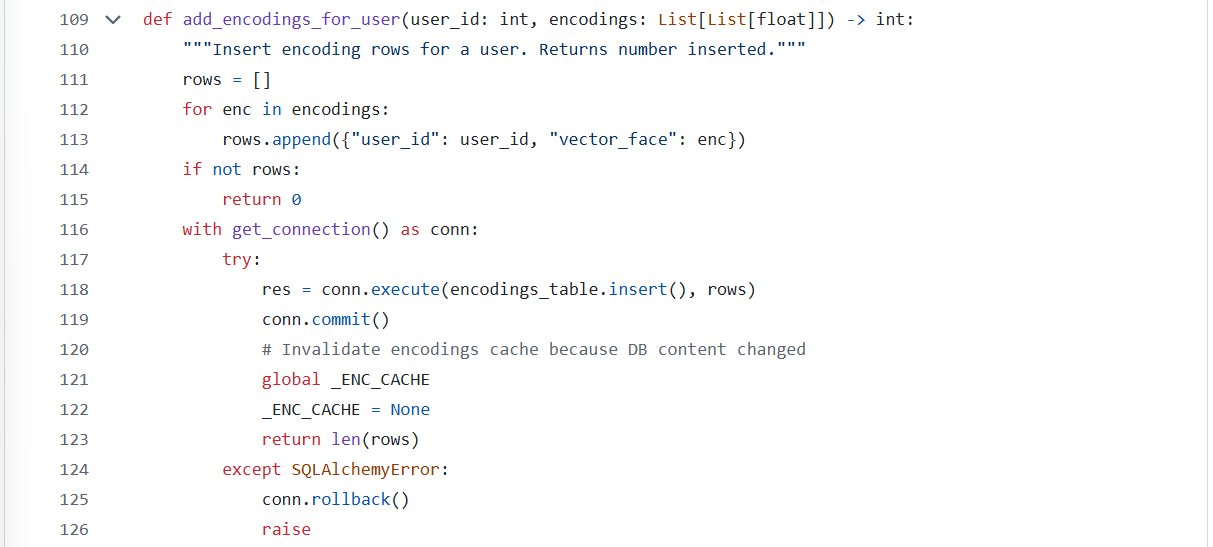


### 5.4.2. Các hàm chính trong db.py

Tạo user mới nếu chưa tồn tại



Thêm encodings cho user



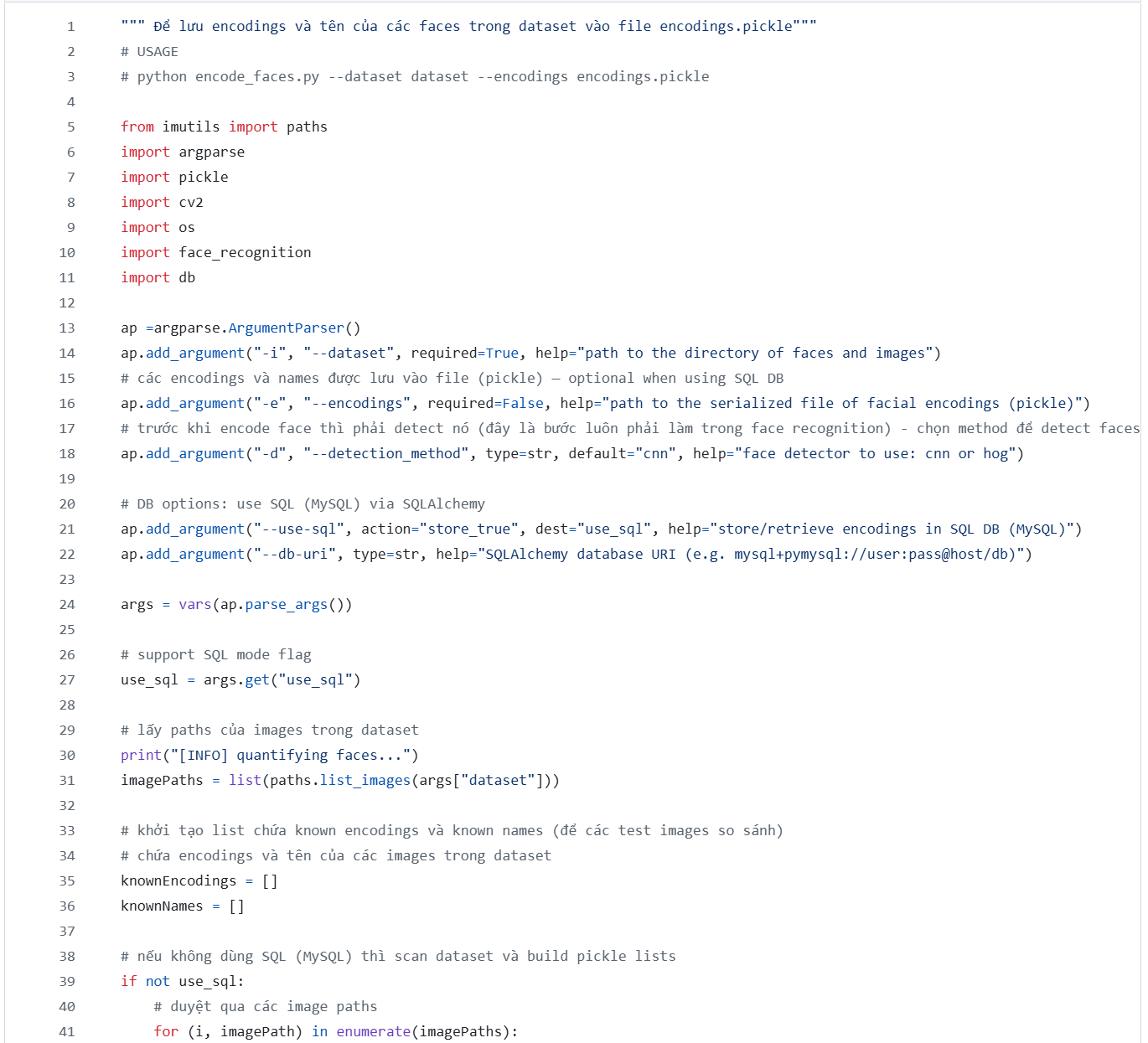
Tìm match tốt nhất cho encoding



## 5.5. Công cụ hỗ trợ

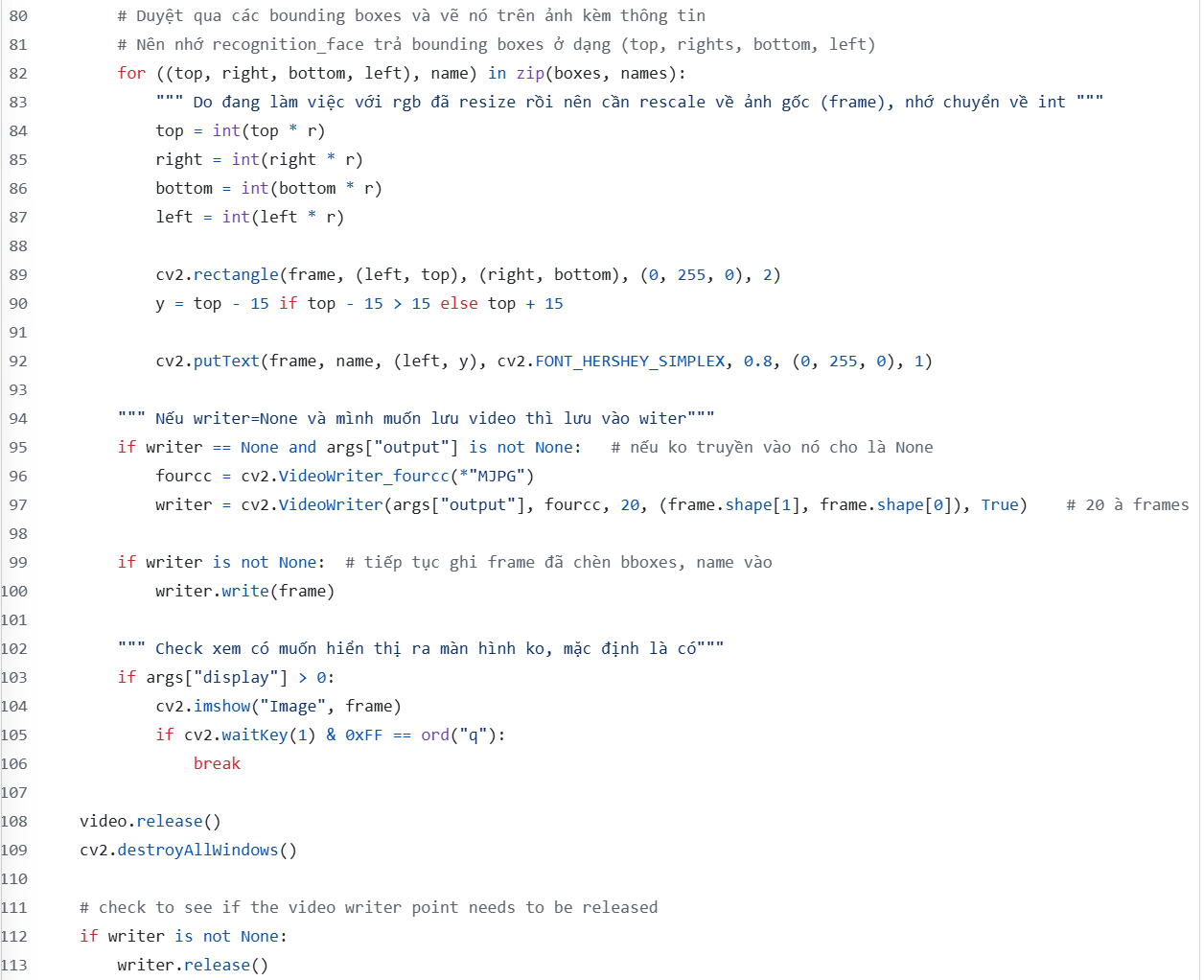
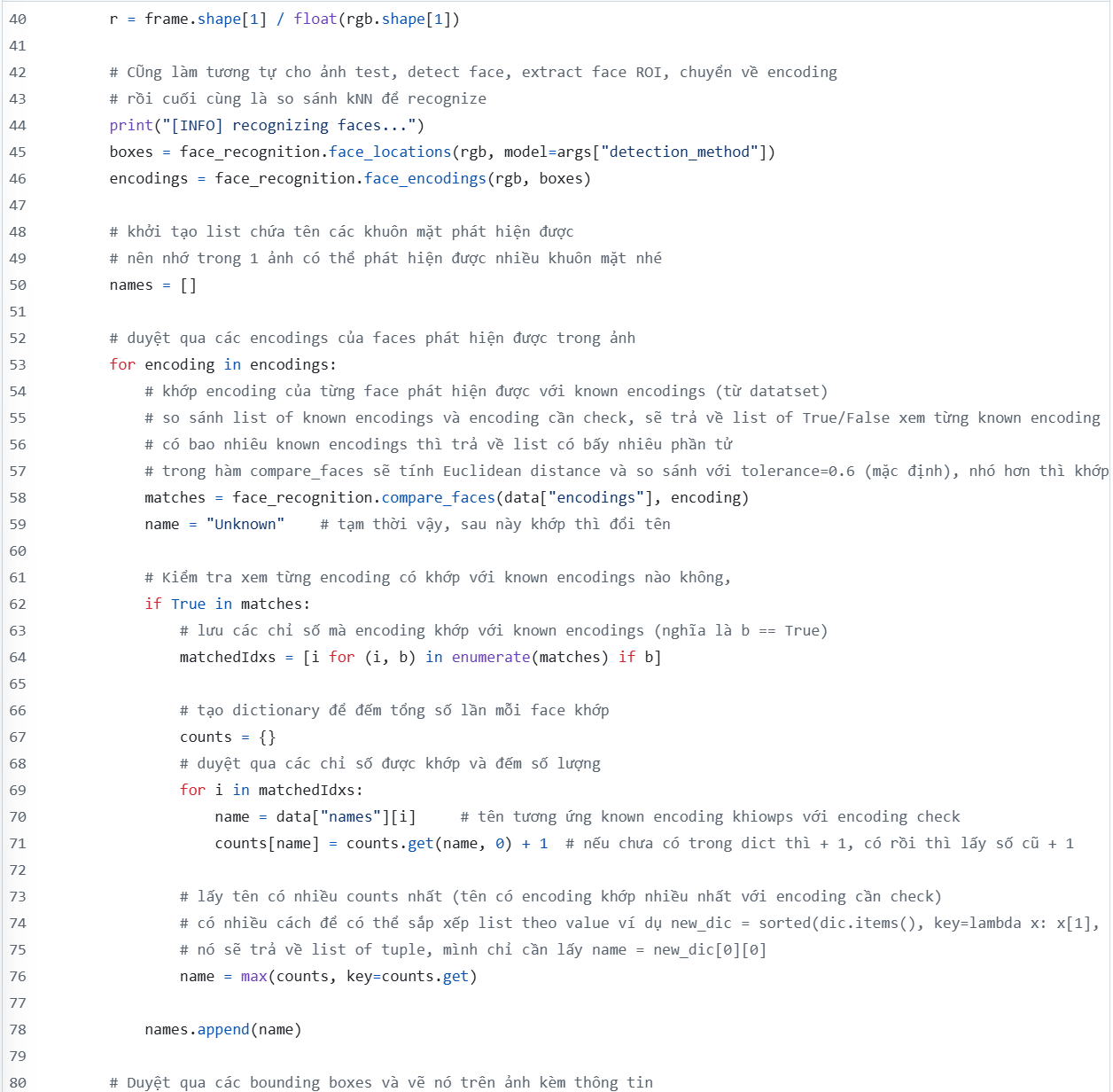
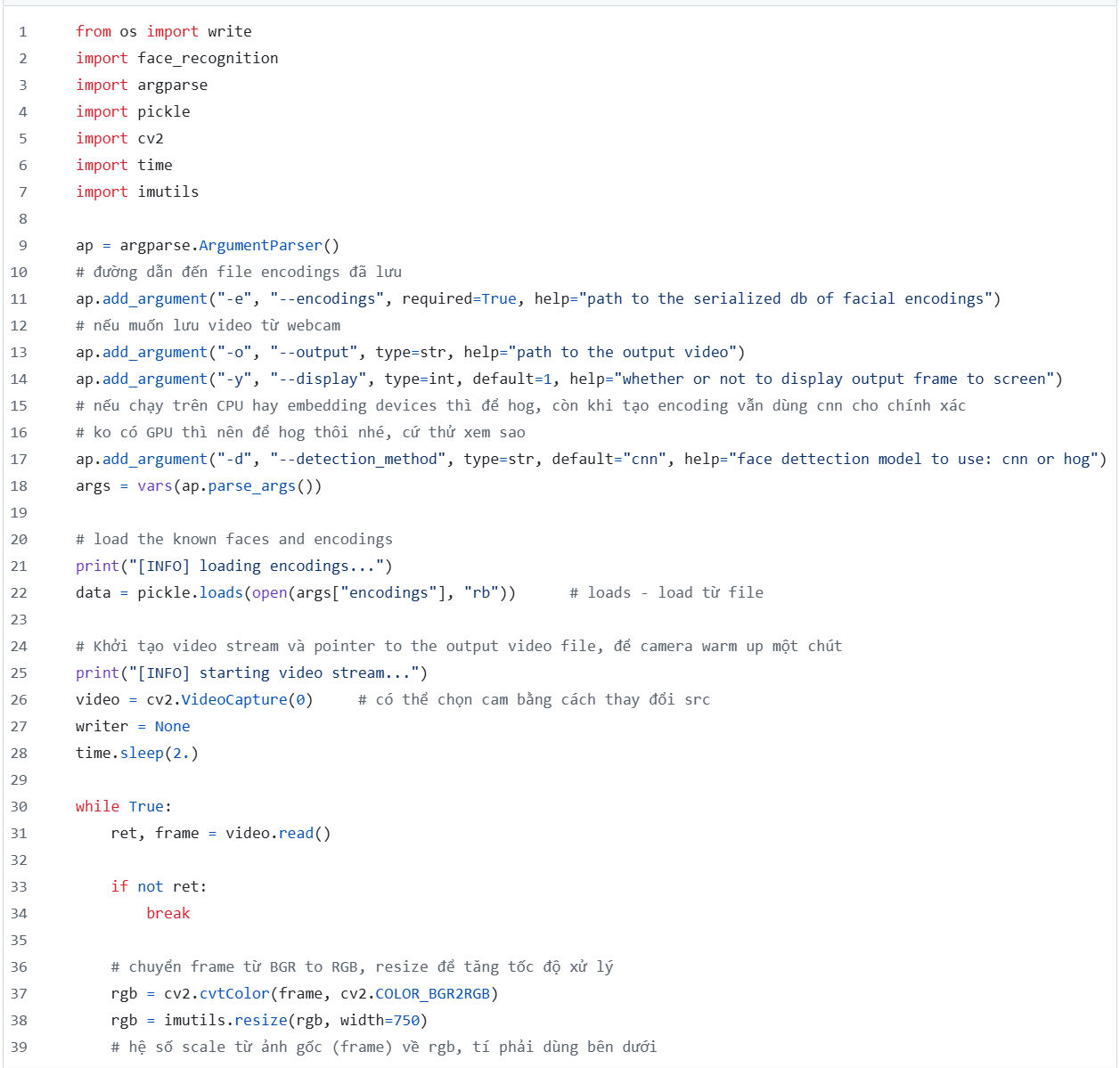
### 5.5.1. Encode faces từ dataset

**File:**encode\_faces.py



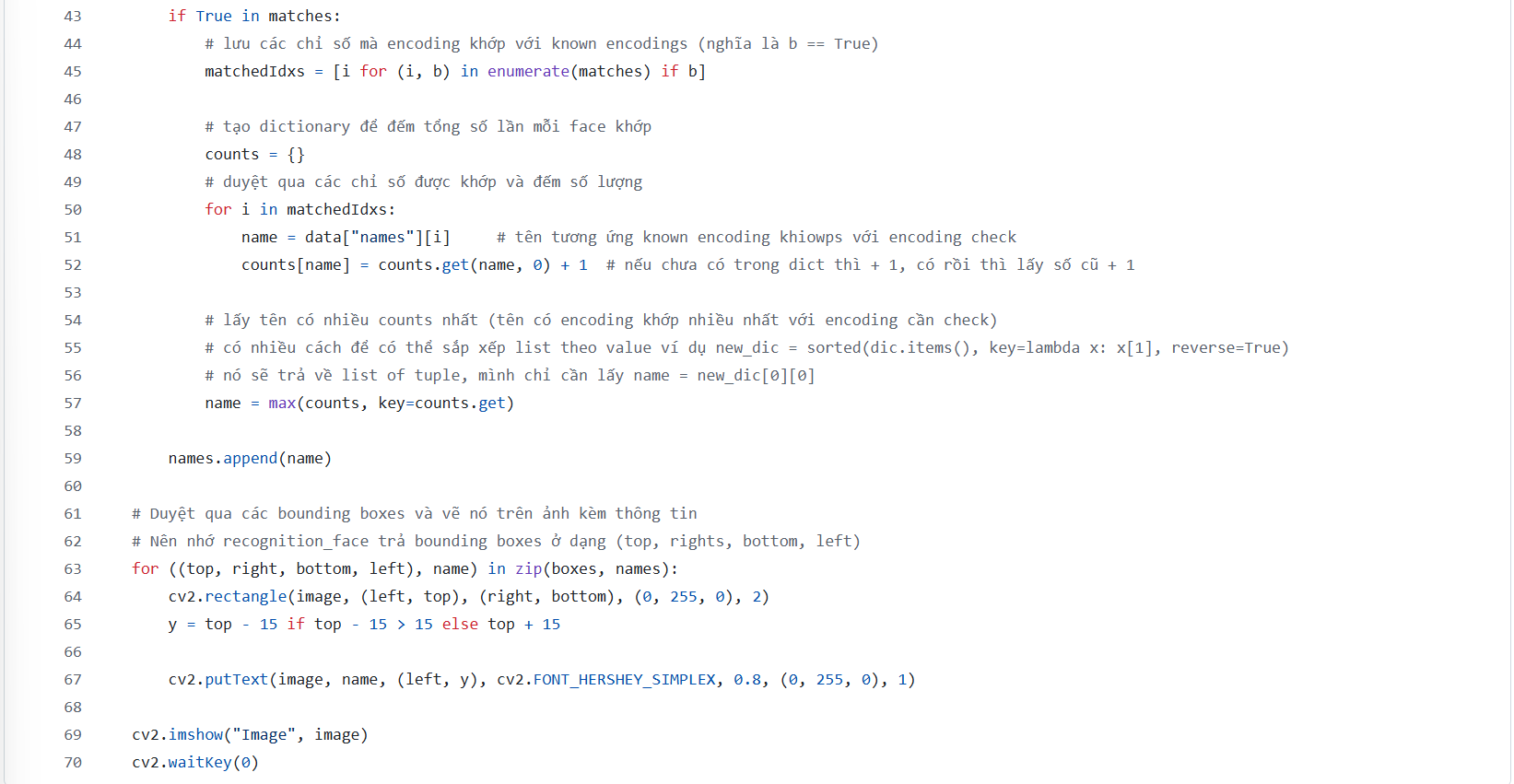
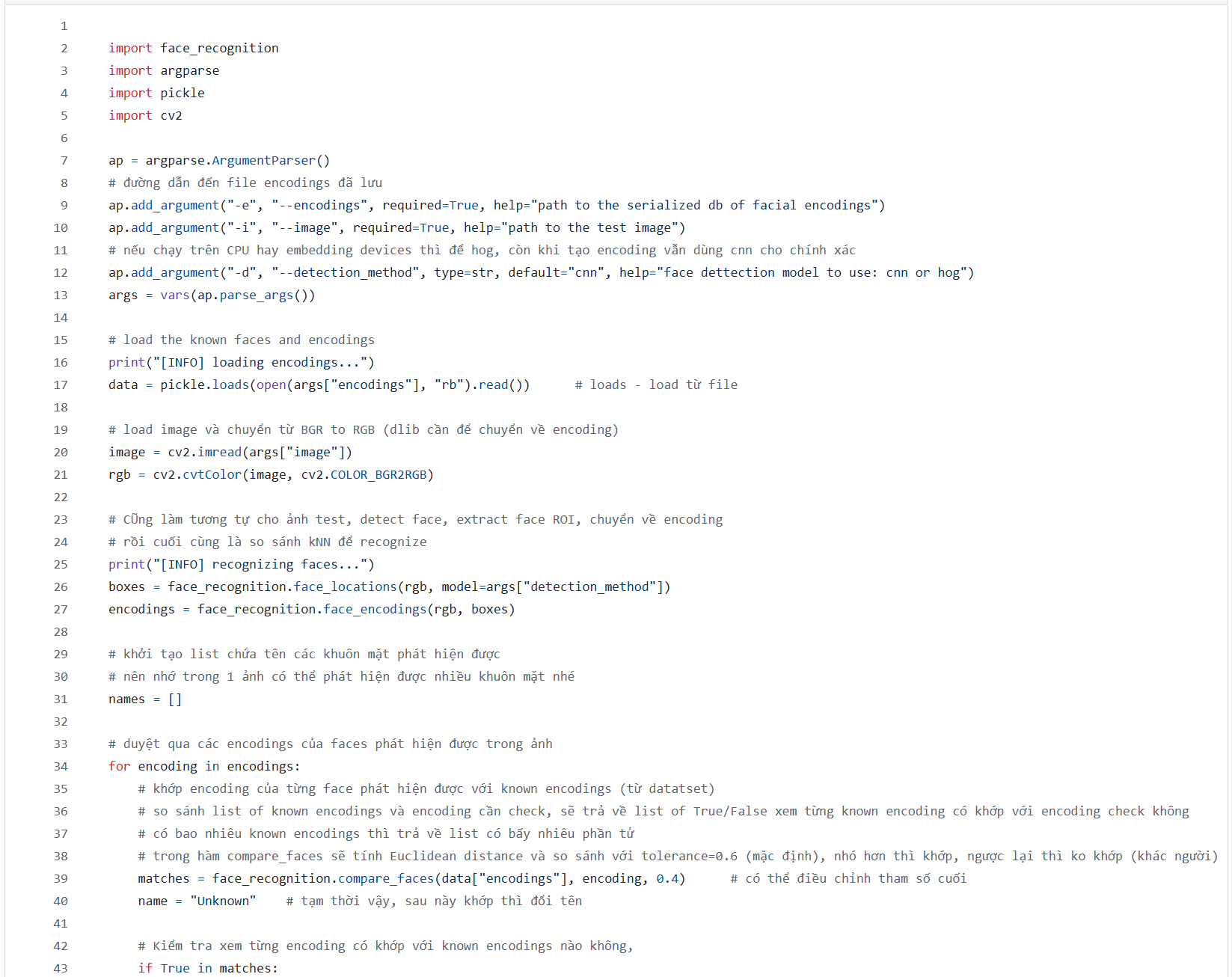
### 5.5.2. Script nhận diện từ video

**File:**recognize\_faces\_video.py



### 5.5.3. Script nhận diện từ ảnh

**File:**recognize\_faces\_image.py

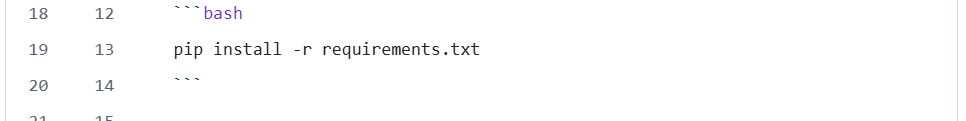


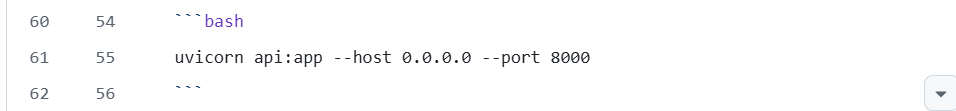
### 5.5.4. Thu thập dữ liệu từ webcam

**File:**build\_dataset.py****

## 5.6. Deployment và Scaling

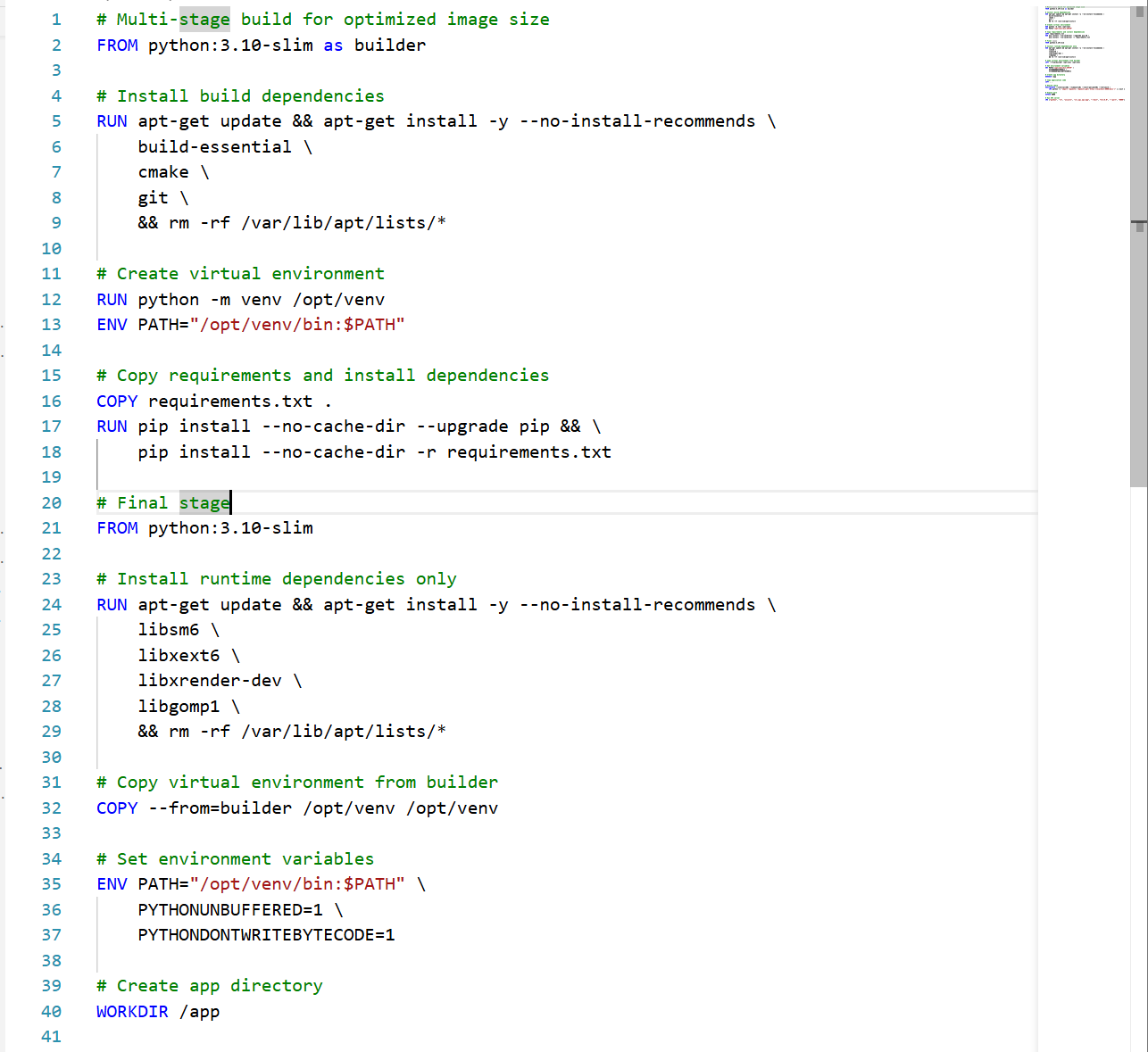
### 5.6.1. Chạy API sever





### 5.6.2. Docker deployment

**File:**Dockerfile





CHƯƠNG VI: THIẾT KẾ HỆ THỐNG GIAO DIỆN TRỰC QUAN

## 6.1. Công cụ và kiến trúc hệ thống được lựa chọn

Để đảm bảo hệ thống hoạt động ổn định, có khả năng mở rộng và giao diện thân thiện với người dùng, nhóm chúng em quyết định xây dựng hệ thống theo mô hình Client-Server với sự kết hợp của các công nghệ hiện đại:

* Frontend (Giao diện người dùng): Sử dụng ReactJS. Đây là thư viện JavaScript mã nguồn mở do Facebook phát triển, chuyên dùng để xây dựng giao diện người dùng (UI) dạng Single Page Application (SPA). ReactJS giúp trải nghiệm người dùng mượt mà, không cần tải lại trang khi chuyển đổi tác vụ.
* Backend (Xử lý nghiệp vụ): Sử dụng Spring Boot (Java) làm nền tảng chính để quản lý người dùng, chấm công và kết nối cơ sở dữ liệu.
* AI Service (Xử lý ảnh): Sử dụng FastAPI (Python) để đóng gói mô hình CNN nhận diện khuôn mặt. FastAPI được chọn vì tốc độ xử lý cực nhanh và hỗ trợ tốt các thư viện AI như PyTorch.

## 6.2. Các tính năng chính của công nghệ sử dụng

ReactJS (Frontend):

* Kiến trúc hướng thành phần (Component-based): Giúp chia nhỏ giao diện thành các phần độc lập (Header, Sidebar, CameraView...) để dễ dàng quản lý và tái sử dụng.
* Virtual DOM: Tối ưu hóa hiệu năng, giúp việc hiển thị luồng video (camera stream) và cập nhật kết quả nhận diện diễn ra mượt mà theo thời gian thực.
* Hệ sinh thái phong phú: Sử dụng các thư viện hỗ trợ mạnh mẽ như react-webcam (xử lý camera), axios (gọi API), và Material-UI hoặc Ant Design (thiết kế giao diện).

Spring Boot & FastAPI (Backend):

* Spring Boot: Cung cấp khả năng bảo mật cao (Spring Security), quản lý giao dịch chặt chẽ và dễ dàng tích hợp với các hệ quản trị cơ sở dữ liệu (MySQL/PostgreSQL) để lưu trữ lịch sử chấm công.
* FastAPI: Hỗ trợ xử lý bất đồng bộ (Asynchronous), giúp việc nhận ảnh từ Frontend và trả về kết quả vector đặc trưng khuôn mặt đạt độ trễ thấp nhất (Low latency).

## 6.3. Triển khai ứng dụng

Quy trình hoạt động của ứng dụng được thiết kế như sau:

* Client (ReactJS): Mở camera, chụp ảnh khuôn mặt người dùng và gửi dữ liệu ảnh (Base64 hoặc Multipart) về Server.
* AI Service (FastAPI): Nhận ảnh, tiền xử lý, đưa qua mô hình CNN để trích xuất đặc trưng và xác định danh tính.
* Core Backend (Spring Boot): Nhận kết quả định danh, kiểm tra thông tin nhân viên trong Database và ghi nhận thời gian chấm công.

Dưới đây là đoạn mã giả lập (pseudo-code) phần xử lý quan trọng tại Frontend sử dụng ReactJS để gửi ảnh đi nhận diện:



Hình 16: Ví dụ sử dụng ReactJS để gửi ảnh đi nhận diện

KẾT LUẬN

Trong bài báo cáo này, nhóm đã hoàn thành việc xây dựng và triển khai hệ thống học sâu để giải quyết bài toán nhận diện khuôn mặt phục vụ công tác quản lý nhân sự. Mô hình được chọn là Resnet, một kiến trúc mạng nơ-ron tích chập (CNN) hiện đại, đã được huấn luyện và kiểm tra trên bộ dữ liệu khuôn mặt nhân viên thực tế. Mô hình đã cho kết quả định danh chính xác với độ chính xác lên đến [9x.x]% trên tập validation và [9x.x]% trên tập test, chứng minh tính khả thi của việc ứng dụng trí tuệ nhân tạo vào quy trình chấm công tự động.

Trong suốt quá trình nghiên cứu, nhóm đã gặp phải một số thách thức đặc thù như sự thay đổi góc nghiêng khuôn mặt (pose), phụ kiện che khuất (kính, khẩu trang) và điều kiện ánh sáng chênh lệch. Tuy nhiên, các kỹ thuật tiền xử lý như Căn chỉnh khuôn mặt (Face Alignment) và tăng cường dữ liệu (Augmentation) đã giúp mô hình cải thiện khả năng tổng quát hóa. Việc áp dụng các phương pháp như thay đổi độ sáng, xoay ảnh và cắt ngẫu nhiên đã giúp mô hình trích xuất được các đặc trưng bất biến tốt hơn từ dữ liệu đầu vào.

Sau khi huấn luyện thành công, nhóm đã đóng gói và triển khai mô hình thành một hệ thống sử dụng thư viện FastAPI, SpringBoot cho backend và ReactJS cho frontend. API này cho phép hệ thống gửi ảnh chụp từ camera và nhận về kết quả định danh (ID nhân viên/Người lạ) với độ trễ thấp.Tạo ra một ứng dụng web trực quan cho phép người quản lý tải ảnh hoặc bật webcam để điểm danh thử nghiệm mà không cần cấu hình phức tạp.

Mô hình này không chỉ đạt độ chính xác khả quan mà còn sẵn sàng cho việc thử nghiệm thực tế, phục vụ cho hệ thống chấm công tại các văn phòng hoặc khu vực sản xuất. Việc triển khai thành công API và Web App giúp việc tích hợp vào các hệ thống ERP hoặc HR sẵn có trở nên dễ dàng và thuận tiện hơn.

Tuy nhiên, hệ thống vẫn còn những hạn chế cần khắc phục. Vấn đề nhận diện sai (False Positive) trong điều kiện ánh sáng quá yếu hoặc khi có nhiều người trong cùng khung hình vẫn còn tồn tại. Trong tương lai, nhóm dự định sẽ nghiên cứu các kiến trúc Loss function tiên tiến hơn như ArcFace hoặc CosFace để tối ưu hóa khoảng cách đặc trưng. Bên cạnh đó, tính năng chống giả mạo (Anti-spoofing) cần được phát triển để ngăn chặn việc chấm công hộ bằng hình ảnh hoặc video.

Hướng phát triển tiếp theo:

Cải tiến mô hình: Áp dụng các kiến trúc mạng chuyên sâu hoặc hàm mất mát tối ưu (như ArcFace) để cải thiện độ chính xác trong môi trường giám sát diện rộng.

Chống giả mạo (Liveness Detection): Tích hợp module phát hiện thực thể sống để phân biệt khuôn mặt thật với ảnh in hoặc màn hình điện thoại, đảm bảo tính minh bạch trong chấm công.

Ứng dụng di động: Phát triển ứng dụng Mobile (Android/iOS) tích hợp GPS để nhân viên có thể chấm công từ xa khi đi công tác.

TÀI LIỆU THAM KHẢO

[1] Trần Hùng Cường, Nguyễn Phương Nga, Giáo trình Trí tuệ nhân tạo, 2014

[2]. Joshi, Practeek, Artificial intelligence with python, 2017

[3]. AI VIET NAM – COURSE 2022 [PDF], Basic CNN - Exercise, 11/12/2022

[4]. AI VIET NAM – COURSE 2022 [PDF]Basic CNN - Tutorial, 11/12/2022

[5]. Afshine Amidi, CS230 - Convolutional Neural Network, 10/2019

[6]. [Python Tutorial: Streamlit | DataCamp](https://www.datacamp.com/tutorial/streamlit)

[7]. [FastAPI class - FastAPI](https://fastapi.tiangolo.com/reference/fastapi/)

[8]. [What is LeNet? - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-lenet/)

[9]. [Python Tutorial](https://www.w3schools.com/python/)

[10]. [Mô hình ResNet](https://aicandy.vn/mo-hinh-resnet-dot-pha-trong-nhan-dien-hinh-anh/)