

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
VÀ TRUYỀN THÔNG VIỆT – HÀN  
KHOA KHOA HỌC MÁY TÍNH**

-----□□□□-----



**ĐỒ ÁN CHUYÊN NGÀNH 1  
ĐỀ TÀI:  
PHẦN MỀM CHẤM CÔNG BẰNG  
NHẬN DIỆN KHUÔN MẶT**

**Sinh viên thực hiện : Nguyễn Đức Phong – 22IT218  
: Trần Nguyên - 22IT194  
Giảng viên hướng dẫn : TS. Phạm Nguyễn Minh Nhựt**

**Đà Nẵng, tháng 5 năm 2025**

## LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến TS. Phạm Nguyễn Minh Nhựt, người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình thực hiện đồ án này. Với kiến thức chuyên môn sâu rộng và sự kiên nhẫn, cô đã không chỉ giúp chúng em khắc phục các khó khăn mà còn khuyến khích chúng em tìm tòi, nghiên cứu, phát huy tối đa khả năng của mình. Những lời động viên, nhận xét, và gợi ý của cô đã là nguồn cảm hứng quý giá để chúng em hoàn thành đề tài một cách tốt nhất.

Chúng em cũng xin gửi lời cảm ơn đến Khoa Khoa Học Máy Tính và Trường Đại Học Công Nghệ Thông Tin và Truyền Thông Việt – Hàn. Những kiến thức mà nhà trường và khoa đã trang bị trong suốt quá trình học tập là nền tảng vững chắc giúp chúng em có đủ năng lực để thực hiện và hoàn thành đồ án này. Đặc biệt, chúng em biết ơn sự hỗ trợ từ đội ngũ cán bộ, nhân viên nhà trường trong việc cung cấp cơ sở vật chất, tài liệu và điều kiện thuận lợi nhất để chúng em có thể tập trung nghiên cứu, học tập.

Chúng em hiểu rằng, để có thể hoàn thành được đồ án này, đó là nhờ sự giúp đỡ và hỗ trợ từ rất nhiều người. Một lần nữa, chúng em xin chân thành cảm ơn tất cả những ai đã trực tiếp hoặc gián tiếp giúp đỡ chúng em trong suốt quá trình thực hiện đồ án này.

Xin chân thành cảm ơn!

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Đà Nẵng, ngày.....tháng....năm 2025*

**Giảng viên hướng dẫn**

**TS. Phạm Nguyễn Minh Nhựt**

## MỤC LỤC

LỜI CẢM ƠN.....	i
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN .....	ii
MỤC LỤC .....	iii
DANH MỤC TỪ VIẾT TẮT .....	v
DANH MỤC HÌNH ẢNH.....	vi
MỞ ĐẦU .....	1
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	2
1.1. Lý do chọn đề tài.....	2
1.2. Mục tiêu nghiên cứu .....	3
1.2.1. Mục tiêu tổng quát .....	3
1.2.2. Mục tiêu cụ thể.....	3
1.3. Bài toán đặt ra .....	4
1.4. Hướng tiếp cận và giải pháp .....	5
1.5. Công nghệ sử dụng .....	6
CHƯƠNG 2. KIẾN TRÚC TỔNG QUAN.....	9
2.1. Tổng quan về xử lý hình ảnh .....	9
2.2. Tổng quan về hệ thống chấm công bằng nhận diện khuôn mặt.....	11
2.2.1. Khái niệm hệ thống chấm công nhận diện khuôn mặt.....	11
2.2.2. Vai trò của hệ thống trong quản lý chấm công .....	11
2.2.3. Các thành phần chính của hệ thống.....	11
2.2.4. Công nghệ sử dụng trong hệ thống .....	11
2.2.5. Lợi ích của hệ thống chấm công bằng nhận diện khuôn mặt.....	12
2.3. Các kỹ thuật, công nghệ, thư viện có liên quan.....	12
2.3.1. Học sâu.....	12
2.3.2. Sk-learn .....	13
CHƯƠNG 3. XÂY DỰNG HỆ THỐNG CHẤM CÔNG BẰNG NHẬN DIỆN KHUÔN MẶT.....	14
3.1. Xây dựng bộ dữ liệu ảnh khuôn mặt.....	14
3.1.1. Mục tiêu.....	14

3.1.2. Nguồn dữ liệu.....	14
3.1.3. Phương pháp thu thập .....	14
3.1.4. Quy trình cụ thể.....	14
3.2. Tiền xử lý dữ liệu.....	15
3.2.1. Phát hiện và cắt khuôn mặt .....	15
3.2.2. Chuẩn hóa ảnh khuôn mặt.....	15
3.2.3. Lưu dữ liệu đã xử lý .....	16
3.3. Xây dựng mô hình.....	16
3.3.1. Kiến trúc mô hình.....	16
3.3.2. Quá trình huấn luyện mô hình.....	17
3.3.3. Nhận diện khuôn mặt .....	17
3.3.4. Ưu điểm của mô hình LBPH.....	18
3.4. Xây dựng giao diện .....	18
3.4.1. Công nghệ sử dụng.....	18
3.4.2. Cấu trúc giao diện chính .....	20
3.4.3. Tích hợp xử lý ảnh vào giao diện.....	20
3.4.4. Trải nghiệm người dùng.....	21
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	22
4.1. Kết quả đạt được .....	22
4.2. Những điểm hạn chế .....	22
4.3. Hướng nghiên cứu.....	23
KẾT LUẬN .....	24
TÀI LIỆU THAM KHẢO .....	25

DANH MỤC TỪ VIẾT TẮT

Số thứ tự	Từ viết tắt	Tên đầy đủ
1	AI	Artificial Intelligence
2	LBPH	Local Binary Patterns Histograms
3	UI	User Interface

## **DANH MỤC HÌNH ẢNH**

Hình 3.1. Mô hình LBPH .....	17
Hình 3.2. Giao diện phần mềm chấm công. ....	18
Hình 3.3. Giao diện đăng ký khuôn mặt.....	19
Hình 3.4. Giao diện ảnh từ webcam được lấy qua OpenCV. ....	21

## MỞ ĐẦU

Trong những năm gần đây, nhu cầu tự động hóa và tối ưu hóa quy trình quản lý nhân sự ngày càng trở nên cấp thiết, đặc biệt là trong lĩnh vực chấm công và kiểm soát ra vào tại các doanh nghiệp, trường học, cơ quan và nhà máy. Các phương pháp chấm công truyền thống như quét thẻ, vân tay, hoặc ghi sổ thủ công không chỉ tốn thời gian, dễ gian lận mà còn thiếu hiệu quả trong việc quản lý số lượng lớn nhân sự, nhất là trong bối cảnh yêu cầu cao về minh bạch và chính xác.

Đặc biệt, trong thời kỳ hậu đại dịch COVID-19, việc hạn chế tiếp xúc và đảm bảo an toàn trong môi trường làm việc lại càng đặt ra yêu cầu cần có một giải pháp chấm công thông minh, không tiếp xúc và đáng tin cậy. Trong khi đó, việc triển khai các hệ thống chấm công thông minh hiện đại vẫn còn nhiều hạn chế về chi phí, tính linh hoạt, khả năng mở rộng cũng như độ chính xác trong điều kiện thực tế như ánh sáng yếu, đeo khẩu trang, thay đổi ngoại hình,...

Trước bối cảnh đó, sự phát triển mạnh mẽ của trí tuệ nhân tạo (AI), đặc biệt là các mô hình học sâu (Deep Learning) và nhận diện khuôn mặt (Face Recognition) đã mở ra những cơ hội mới trong việc xây dựng các hệ thống chấm công thông minh. Công nghệ AI ngày nay có thể nhận dạng khuôn mặt với độ chính xác cao trong thời gian thực, kể cả trong điều kiện môi trường phức tạp.

Chính vì vậy, nhóm chúng em lựa chọn đề tài **"Phần mềm chấm công bằng nhận diện khuôn mặt"** với mục tiêu xây dựng một hệ thống chấm công tự động, chính xác, an toàn và dễ triển khai. Phần mềm sử dụng công nghệ nhận diện khuôn mặt dựa trên AI để xác định danh tính nhân sự khi ra vào và ghi nhận thời gian chấm công hoàn toàn tự động, không cần tiếp xúc hay thiết bị vật lý đi kèm.

Đây là một đề tài vừa mang tính ứng dụng thực tiễn cao, vừa giúp nhóm sinh viên tiếp cận và thực hành các công nghệ hiện đại như xử lý ảnh, học sâu, nhận diện khuôn mặt và phát triển phần mềm. Đồng thời, đề tài còn góp phần thúc đẩy quá trình chuyển đổi số trong công tác quản lý nhân sự, nâng cao hiệu quả hoạt động tổ chức và giảm thiểu rủi ro gian lận trong chấm công.



## CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

### 1.1. Lý do chọn đề tài

Trong những năm gần đây, nhu cầu ứng dụng công nghệ vào quản lý nhân sự, đặc biệt là trong lĩnh vực chấm công, ngày càng trở nên phổ biến và cấp thiết. Các phương pháp chấm công truyền thống như quẹt thẻ, ký tên thủ công hoặc sử dụng vân tay đang bộc lộ nhiều hạn chế như dễ gian lận, tốn thời gian, không đảm bảo vệ sinh (đặc biệt trong thời kỳ dịch bệnh), và thiếu linh hoạt khi áp dụng cho lực lượng lao động đông đảo hoặc làm việc từ xa.

Đặc biệt, trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại các cơ quan, doanh nghiệp và tổ chức giáo dục, việc xây dựng một hệ thống chấm công tự động, thông minh, không tiếp xúc và chính xác là vô cùng cần thiết. Tuy nhiên, hiện nay các giải pháp tiên tiến thường có chi phí cao, yêu cầu hạ tầng phức tạp và khó triển khai rộng rãi tại các đơn vị vừa và nhỏ.

Trong khi đó, sự phát triển nhanh chóng của trí tuệ nhân tạo (AI), đặc biệt là các mô hình học sâu (Deep Learning) và công nghệ nhận diện khuôn mặt (Face Recognition), đã mở ra nhiều cơ hội mới trong việc xây dựng các hệ thống chấm công hiệu quả, an toàn và hiện đại. Công nghệ nhận diện khuôn mặt hiện nay có thể nhận diện người dùng một cách nhanh chóng, chính xác ngay cả trong điều kiện ánh sáng thay đổi, người đeo khẩu trang, hoặc có sự thay đổi nhỏ về ngoại hình.

Xuất phát từ thực tiễn đó, nhóm chúng em lựa chọn đề tài "**Phần mềm chấm công bằng nhận diện khuôn mặt**" với mục tiêu nghiên cứu và phát triển một hệ thống chấm công thông minh, sử dụng công nghệ AI để nhận diện khuôn mặt của người dùng và ghi nhận thời gian ra vào một cách tự động, không cần tiếp xúc. Hệ thống này có thể được triển khai tại văn phòng, nhà máy, trường học hoặc bất kỳ tổ chức nào có nhu cầu quản lý thời gian làm việc hiệu quả.

Đây là đề tài mang tính ứng dụng cao, thiết thực với nhu cầu thực tiễn của xã hội, đồng thời giúp nhóm sinh viên tiếp cận và vận dụng các kiến thức hiện đại về trí tuệ nhân tạo, xử lý ảnh, học máy và phát triển phần mềm. Ngoài ra, đề tài cũng hướng đến việc xây dựng các giải pháp quản lý thông minh, góp phần thúc đẩy quá trình chuyển đổi số trong công tác quản lý nhân sự, nâng cao hiệu suất làm việc và minh bạch trong các tổ chức.

## **1.2. Mục tiêu nghiên cứu**

### **1.2.1. Mục tiêu tổng quát**

Mục tiêu tổng quát của đề tài là xây dựng một ứng dụng chấm công thông minh có khả năng nhận diện khuôn mặt một cách tự động và chính xác, nhằm thay thế các phương pháp chấm công truyền thống. Hệ thống ứng dụng các kỹ thuật học sâu, đặc biệt là mạng nơ-ron nhân tạo (Neural Network), để xác định danh tính người dùng dựa trên hình ảnh khuôn mặt và ghi nhận thời gian ra/vào một cách nhanh chóng, chính xác và không tiếp xúc.

Thông qua đó, đề tài hướng đến việc khai thác tiềm năng của trí tuệ nhân tạo trong lĩnh vực quản lý nhân sự, góp phần số hóa quy trình chấm công, nâng cao hiệu quả làm việc và giảm thiểu tình trạng gian lận, sai sót trong ghi nhận thời gian làm việc. Giải pháp đặc biệt hữu ích trong bối cảnh doanh nghiệp, tổ chức cần một hệ thống giám sát hiện đại, linh hoạt và tiết kiệm chi phí.

Đề tài không chỉ nhằm giải quyết một bài toán công nghệ, mà còn hướng đến việc phát triển một giải pháp có tính ứng dụng thực tiễn cao, dễ triển khai tại các cơ quan, nhà máy, trường học hoặc tổ chức có nhu cầu quản lý nhân sự. Ứng dụng được kỳ vọng sẽ trở thành công cụ hỗ trợ đắc lực trong việc quản lý thời gian làm việc của nhân viên, đảm bảo tính minh bạch, đồng thời phù hợp với xu hướng chuyển đổi số hiện nay.

Ngoài ra, đề tài cũng tạo điều kiện để nhóm thực hiện rèn luyện và vận dụng các kiến thức đã học về trí tuệ nhân tạo, xử lý ảnh, lập trình ứng dụng và phân tích dữ liệu trong một bài toán thực tế. Qua đó, nhóm có cơ hội phát triển tư duy nghiên cứu, khả năng làm việc nhóm và kỹ năng triển khai giải pháp công nghệ trong đời sống, góp phần chuẩn bị hành trang vững chắc cho sự nghiệp sau này.

### **1.2.2. Mục tiêu cụ thể**

+ Tìm hiểu kiến thức nền tảng về công nghệ nhận diện khuôn mặt, các thuật toán học sâu (Deep Learning) thường được sử dụng như Convolutional Neural Network (CNN), Local Binary Patterns Histograms (LBPH), FaceNet, MTCNN,... nhằm phục vụ cho việc xây dựng hệ thống nhận dạng chính xác và hiệu quả..

- + Khảo sát và lựa chọn mô hình mạng Neural Network phù hợp, ưu tiên các kiến trúc có hiệu suất cao trong bài toán nhận diện khuôn mặt và nhận dạng danh tính trong thời gian thực, đảm bảo cân bằng giữa độ chính xác và tốc độ xử lý.
- + Xây dựng và xử lý bộ dữ liệu khuôn mặt, bao gồm quá trình thu thập, làm sạch, gán nhãn và tiền xử lý ảnh khuôn mặt của người dùng, nhằm đảm bảo chất lượng đầu vào cho mô hình nhận diện.
- + Huấn luyện và đánh giá mô hình nhận diện khuôn mặt, sử dụng các chỉ số đánh giá như độ chính xác, độ nhạy, tỷ lệ false positive/false negative và thời gian nhận dạng, để đảm bảo mô hình đáp ứng tốt yêu cầu thực tế.
- + Phát triển giao diện người dùng (UI) thân thiện, cho phép nhân viên thực hiện chấm công thông qua việc quét khuôn mặt bằng webcam hoặc camera từ thiết bị cố định (máy tính, máy chấm công), đồng thời hiển thị thông tin thời gian ra vào và lưu trữ dữ liệu.
- + Tích hợp chức năng quản lý dữ liệu nhân sự, bao gồm lưu trữ thông tin chấm công theo ngày/giờ, xuất báo cáo, thống kê thời gian làm việc, giúp quản trị viên dễ dàng theo dõi và phân tích hiệu suất làm việc.
- + Đánh giá toàn diện hệ thống phần mềm, từ độ chính xác của mô hình nhận diện đến mức độ tiện lợi, thân thiện với người dùng, đồng thời đề xuất các hướng mở rộng như hỗ trợ nhiều vị trí lắp đặt, nhận diện trong điều kiện ánh sáng khác nhau hoặc tích hợp thêm các công nghệ bảo mật như nhận diện kết hợp giọng nói/ký hiệu.

### 1.3. Bài toán đặt ra

Trong thực tế, việc chấm công tại nhiều cơ quan, doanh nghiệp và cơ sở sản xuất vẫn chủ yếu dựa vào các phương pháp truyền thống như ký tên, quét thẻ hoặc sử dụng máy chấm công vân tay. Tuy nhiên, các phương pháp này tồn tại nhiều hạn chế như dễ bị gian lận (chấm công hộ), mất thời gian, hao mòn thiết bị, và đặc biệt là thiếu an toàn vệ sinh trong bối cảnh dịch bệnh. Ngoài ra, việc quản lý và tổng hợp dữ liệu chấm công từ các phương pháp thủ công thường tốn nhiều công sức, dễ xảy ra sai sót và khó tích hợp với các hệ thống quản lý nhân sự hiện đại.

Mặt khác, tại các doanh nghiệp nhỏ, cơ sở sản xuất ở vùng nông thôn hoặc những nơi có điều kiện công nghệ hạn chế, việc đầu tư vào các hệ thống chấm công hiện đại thường gặp khó khăn do chi phí cao hoặc thiếu nhân lực kỹ thuật. Điều này đặt ra nhu

cầu cấp thiết về một hệ thống chấm công tự động – vừa dễ triển khai, vừa có độ chính xác cao – để giúp các đơn vị quản lý nhân sự hiệu quả hơn, đồng thời nâng cao tính minh bạch và chuyên nghiệp trong môi trường làm việc.

Chính vì vậy, việc ứng dụng công nghệ nhận diện khuôn mặt vào chấm công đang được xem là một giải pháp tiên tiến, an toàn và tiết kiệm, phù hợp với xu hướng chuyên đổi số hiện nay.

#### **Bài toán đặt ra là:**

+ Làm thế nào để xây dựng một hệ thống có khả năng nhận diện khuôn mặt một cách tự động, chính xác và đáng tin cậy, phục vụ cho mục tiêu chấm công, bằng cách ứng dụng các kỹ thuật xử lý ảnh kết hợp với thuật toán nhận diện khuôn mặt LBPH (Local Binary Patterns Histograms) trong môi trường thời gian thực.?

+ Làm thế nào để ứng dụng có thể giao tiếp hiệu quả với người dùng, cung cấp thông báo về thời gian chấm công, hiển thị lịch sử ra vào, cảnh báo khi nhận diện không thành công hoặc phát hiện bất thường trong thời gian làm việc?

+ Làm thế nào để đảm bảo giao diện người dùng đơn giản, dễ sử dụng, hoạt động ổn định trên nhiều nền tảng (máy tính, máy chấm công, thiết bị di động), đồng thời có khả năng mở rộng trong tương lai như: hỗ trợ thêm điểm chấm công, nâng cao độ chính xác mô hình nhận diện trong điều kiện ánh sáng khác nhau, tích hợp chức năng báo cáo và phân tích tự động?

Từ những câu hỏi này, đề tài định hướng xây dựng một giải pháp tích hợp giữa mô hình nhận diện khuôn mặt sử dụng Deep Learning và hệ thống giao diện thân thiện với người dùng, kết hợp các chức năng quản lý chấm công thông minh, nhằm đáp ứng nhu cầu hiện đại hóa và tự động hóa quy trình quản lý nhân sự trong doanh nghiệp, trường học và tổ chức.

#### **1.4. Hướng tiếp cận và giải pháp**

Hướng tiếp cận được chia thành hai phần chính:

##### **1. Xây dựng mô hình nhận diện khuôn mặt phục vụ chấm công:**

+ Thu thập và xử lý dữ liệu: Hệ thống cho phép người dùng đăng ký khuôn mặt bằng cách chụp trực tiếp qua webcam. Mỗi người dùng được gán một mã định danh (ID) duy nhất. Trong quá trình đăng ký, hệ thống thu thập nhiều ảnh khuôn mặt từ các góc độ, biểu cảm khác nhau nhằm tăng độ đa dạng của dữ liệu. Sau đó, mỗi ảnh được

chuyển sang ảnh xám (grayscale), phát hiện và cắt vùng chứa khuôn mặt bằng thuật toán Haar Cascade để chuẩn hóa kích thước trước khi đưa vào huấn luyện.

- + Lựa chọn thuật toán nhận diện khuôn mặt: Hệ thống sử dụng thuật toán Local Binary Patterns Histograms (LBPH) – một phương pháp hiệu quả trong nhận diện khuôn mặt với độ chính xác cao trong điều kiện ánh sáng ổn định. LBPH phân tích kết cấu của ảnh thông qua các mẫu nhị phân cục bộ, từ đó trích xuất đặc trưng khuôn mặt.

- + Huấn luyện mô hình: Sau khi thu thập ảnh khuôn mặt, mô hình LBPH được huấn luyện với dữ liệu hình ảnh đã xử lý. Quá trình huấn luyện được thực hiện bằng thư viện `cv2.face.LBPHFaceRecognizer_create()` trong OpenCV, lưu lại mô hình đã huấn luyện vào file (ví dụ: `trainer.yml`) để sử dụng cho quá trình nhận diện sau này.

- + Đánh giá mô hình: Mô hình được kiểm tra qua quá trình nhận diện thời gian thực bằng webcam. Hệ thống so sánh khuôn mặt được phát hiện với mô hình đã huấn luyện và trả về ID tương ứng nếu khớp, đồng thời ghi nhận thời gian chấm công. Hệ thống đạt độ chính xác ổn định trong điều kiện môi trường kiểm soát (ánh sáng rõ, góc nhìn chính diện).

## 2. Phát triển ứng dụng chấm công và giao diện người dùng:

- + Thiết kế giao diện người dùng (UI): Xây dựng giao diện thân thiện, dễ sử dụng trên máy tính hoặc thiết bị di động. Giao diện cho phép người dùng đăng nhập, quản lý hồ sơ nhân sự, xem lịch sử chấm công và hiển thị kết quả nhận diện theo thời gian thực.\

- + Tích hợp mô hình AI vào ứng dụng: Sử dụng các framework như TensorFlow, PyTorch, hoặc OpenCV để tích hợp mô hình nhận diện vào hệ thống. Ứng dụng có thể hoạt động trực tiếp trên thiết bị (edge computing) hoặc trên server tùy theo nhu cầu triển khai.

Thông qua hướng tiếp cận này, hệ thống phần mềm sẽ mang lại một giải pháp chấm công hiện đại, chính xác và dễ triển khai, góp phần giảm thiểu sai sót, tiết kiệm thời gian và nâng cao hiệu quả quản lý nhân sự trong các tổ chức và doanh nghiệp.

### 1.5. Công nghệ sử dụng

Để xây dựng hệ thống chấm công bằng nhận diện khuôn mặt một cách hiệu quả, nhóm đã kết hợp nhiều công nghệ thuộc các lĩnh vực xử lý ảnh, lập trình giao diện, và quản lý dữ liệu. Cụ thể:

### 1. Ngôn ngữ lập trình

+ Python: Ngôn ngữ chính được sử dụng cho quá trình xử lý dữ liệu, xây dựng và huấn luyện mô hình AI nhờ vào hệ sinh thái phong phú, dễ học và có cộng đồng lớn hỗ trợ.

### 2. Thư viện và framework AI

+ OpenCV (cv2): Thư viện chính dùng cho các tác vụ xử lý ảnh như truy xuất webcam, phát hiện khuôn mặt (sử dụng Haar Cascade), huấn luyện và nhận diện khuôn mặt (sử dụng LBPH – Local Binary Patterns Histograms), và xử lý dữ liệu ảnh.

+ Tkinter: Thư viện GUI tích hợp trong Python, dùng để xây dựng giao diện desktop cho phần mềm. Cho phép tạo các thành phần như nút bấm, khung hiển thị ảnh, hộp nhập thông tin người dùng, giúp người dùng tương tác dễ dàng với hệ thống.

+ NumPy / PIL: Hỗ trợ xử lý dữ liệu ảnh và ma trận ảnh trong quá trình tiền xử lý và huấn luyện mô hình.

### 3. Cơ sở dữ liệu

+ CSV: Sử dụng để lưu trữ thông tin người dùng, dữ liệu chấm công và các dữ liệu liên quan dưới dạng bảng tính đơn giản, thuận tiện cho việc đọc, ghi và quản lý dữ liệu nhỏ đến vừa.

### 4. Giao diện người dùng

#### **Giao diện đồ họa (Tkinter):**

- Cho phép **đăng ký khuôn mặt**: Người dùng nhập ID và tên, sau đó hệ thống tự động chụp nhiều ảnh khuôn mặt và lưu vào thư mục dữ liệu.

- Cho phép **chấm công**: Hệ thống nhận diện khuôn mặt từ webcam, xác định danh tính và ghi thời gian chấm công vào file CSV.

- Giao diện hiển thị ảnh webcam trực tiếp, nút chức năng rõ ràng, thân thiện và dễ sử dụng.

### 5. Công cụ hỗ trợ khác

- IDLE / VS Code / PyCharm: Các môi trường phát triển tích hợp được sử dụng trong quá trình viết và kiểm thử mã nguồn.

- GitHub (tùy chọn): Hỗ trợ quản lý phiên bản mã nguồn nếu hệ thống phát triển theo nhóm.

Việc kết hợp các công nghệ trên đã giúp hệ thống chấm công hoạt động hiệu quả, đảm bảo tính chính xác trong nhận diện, giao diện dễ sử dụng, đồng thời dễ mở rộng và tích hợp vào các hệ thống quản lý nhân sự trong tương lai.

## CHƯƠNG 2. KIẾN TRÚC TỔNG QUAN

### 2.1. Tổng quan về xử lý hình ảnh

Xử lý hình ảnh là bước tiền đề quan trọng trong hệ thống nhận diện khuôn mặt, đặc biệt trong phần mềm chấm công tự động. Quá trình xử lý ảnh giúp chuẩn hóa dữ liệu đầu vào, loại bỏ nhiễu và trích xuất những đặc trưng quan trọng từ khuôn mặt, từ đó tăng hiệu quả và độ chính xác của mô hình nhận diện.

#### Quy trình xử lý ảnh trong hệ thống:

Trong phần mềm chấm công này, quá trình xử lý ảnh sử dụng các kỹ thuật xử lý ảnh truyền thống với thư viện OpenCV, các bước chính bao gồm:

##### 1. Chụp ảnh và chuyển đổi sang ảnh xám (Grayscale):

Hình ảnh khuôn mặt được thu nhận từ camera hoặc nguồn ảnh, sau đó chuyển sang ảnh xám để giảm thiểu dữ liệu không cần thiết và tập trung vào đặc trưng khuôn mặt. Việc chuyển ảnh sang grayscale giúp giảm tải xử lý và cải thiện hiệu suất nhận diện.

##### 2. Phát hiện khuôn mặt (Face Detection):

Sử dụng bộ phân loại Haar Cascade để phát hiện vị trí khuôn mặt trong ảnh. Bộ phát hiện này nhận diện các vùng khuôn mặt trong ảnh xám dựa trên đặc trưng Haar, đảm bảo chỉ những vùng có khuôn mặt mới được xử lý tiếp.

##### 3. Trích xuất vùng khuôn mặt:

Sau khi phát hiện được vùng khuôn mặt, phần mềm cắt vùng này và lưu vào thư mục theo mã sinh viên và tên, chuẩn bị cho việc huấn luyện mô hình:

for (x, y, w, h) in faces:

```
cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
```

```
sampleNum += 1
```

```
cv2.imwrite(
```

```
    f"{path}\\\"
```

```
    + Name
```

```
    + "_"
```

```
    + Enrollment
```

```
    + "_"
```



```

+ str(sampleNum)
+ ".jpg",
gray[y:y+h, x:x+w]
)

```

Việc này giúp đảm bảo chỉ những vùng khuôn mặt được tập trung xử lý và lưu trữ, loại bỏ các chi tiết thừa không liên quan.

#### 4. Huấn luyện mô hình nhận diện khuôn mặt (LBPH):

- Sau khi thu thập đủ ảnh mẫu, mô hình LBPH (Local Binary Patterns Histograms) được huấn luyện để tạo ra một bộ nhận diện khuôn mặt:

```

recognizer = cv2.face.LBPHFaceRecognizer_create()
faces, Id = getImagesAndLables(trainimage_path)
recognizer.train(faces, np.array(Id))
recognizer.save(trainimagelabel_path)

```

- Hàm `getImagesAndLables` đọc ảnh từ thư mục lưu trữ, chuyển ảnh về dạng mảng numpy và lấy ID từ tên file để gán nhãn:

```

pilImage = Image.open(imagePath).convert("L")
imageNp = np.array(pilImage, "uint8")
Id = int(os.path.split(imagePath)[-1].split("_")[1])
faces.append(imageNp)
Ids.append(Id)

```

- LBPH là phương pháp nhận diện khuôn mặt mạnh mẽ trong điều kiện ánh sáng khác nhau và biến đổi biểu cảm khuôn mặt, phù hợp với môi trường thực tế của phần mềm chấm công.

#### Ý nghĩa thực tiễn:

+ Chuyển đổi ảnh sang grayscale giúp giảm kích thước dữ liệu đầu vào, tăng tốc độ xử lý và tránh bị ảnh hưởng bởi màu sắc không đồng nhất trong môi trường làm việc.

+ Phát hiện khuôn mặt chính xác bằng Haar Cascade giúp hệ thống xác định đúng vị trí khuôn mặt, tránh nhầm lẫn với các đối tượng khác.

+ Lưu ảnh khuôn mặt riêng biệt theo từng cá nhân hỗ trợ việc tạo dữ liệu huấn luyện chuẩn và quản lý hiệu quả.

Huấn luyện mô hình LBPH cung cấp độ chính xác cao trong nhận diện khuôn mặt, đồng thời có khả năng chịu được biến đổi ánh sáng, góc nhìn và biểu cảm khuôn mặt.

## **2.2. Tổng quan về hệ thống chấm công bằng nhận diện khuôn mặt**

### **2.2.1. Khái niệm hệ thống chấm công nhận diện khuôn mặt**

Hệ thống chấm công nhận diện khuôn mặt là một ứng dụng sử dụng công nghệ xử lý ảnh và trí tuệ nhân tạo để tự động nhận diện và ghi nhận thời gian làm việc của nhân viên thông qua việc xác định khuôn mặt cá nhân. Thay vì sử dụng phương pháp truyền thống như thẻ từ hay điểm danh bằng tay, hệ thống này giúp nâng cao tính chính xác, nhanh chóng và tiện lợi trong quản lý nhân sự.

### **2.2.2. Vai trò của hệ thống trong quản lý chấm công**

Trong môi trường doanh nghiệp hoặc các tổ chức, hệ thống chấm công nhận diện khuôn mặt giúp:

- Tự động ghi nhận thời gian ra vào của nhân viên mà không cần sự can thiệp thủ công.
- Giảm thiểu gian lận hoặc nhầm lẫn so với việc sử dụng thẻ hoặc ký tên.
- Tối ưu hóa quá trình quản lý, báo cáo chấm công, từ đó nâng cao hiệu suất làm việc.
- Tích hợp lưu trữ dữ liệu ảnh, danh sách nhân viên, và tạo báo cáo chi tiết về tình hình đi làm.

### **2.2.3. Các thành phần chính của hệ thống**

- Thu nhận hình ảnh: Sử dụng camera để chụp ảnh khuôn mặt nhân viên.
- Xử lý hình ảnh: Áp dụng thuật toán nhận diện khuôn mặt (ví dụ như LBPH Face Recognizer) để trích xuất đặc trưng và so sánh với dữ liệu đã lưu.
- Quản lý dữ liệu: Lưu trữ thông tin nhân viên và lịch sử điểm danh vào file CSV hoặc cơ sở dữ liệu.
- Giao diện người dùng: Cung cấp các cửa sổ nhập liệu, thông báo trạng thái, và quản lý dữ liệu qua Tkinter.

### **2.2.4. Công nghệ sử dụng trong hệ thống**

- OpenCV: Thư viện xử lý ảnh và nhận diện khuôn mặt.

- Tkinter: Thư viện giao diện người dùng GUI cho Python, giúp tạo cửa sổ nhập liệu và hiển thị thông báo.

- Pandas, CSV: Quản lý và lưu trữ dữ liệu nhân viên và lịch sử chấm công.

- Python: Ngôn ngữ lập trình chính để phát triển hệ thống.

### **2.2.5. Lợi ích của hệ thống chấm công bằng nhận diện khuôn mặt**

- Tiết kiệm thời gian và chi phí: Giảm thiểu thời gian điểm danh thủ công và tránh sai sót.

- Tăng độ chính xác: Nhận diện chính xác từng cá nhân dựa trên khuôn mặt, tránh gian lận.

- Dễ dàng quản lý: Lưu trữ và xử lý dữ liệu điện tử thuận tiện cho việc báo cáo.

- Tính tiện lợi và thân thiện với người dùng: Giao diện đơn giản, dễ sử dụng cho nhân viên và quản lý.

## **2.3. Các kĩ thuật, công nghệ, thư viện có liên quan**

### **2.3.1. Học sâu**

Học sâu là một nhánh của học máy (Machine Learning), sử dụng các mạng nơ-ron nhiều lớp (deep neural networks) để tự động trích xuất đặc trưng và học mô hình phức tạp từ dữ liệu lớn. Trong bài toán nhận diện bệnh ngoài da từ ảnh, học sâu đặc biệt hiệu quả vì có khả năng xử lý ảnh đầu vào trực tiếp, từ đó nhận diện các đặc trưng hình học và màu sắc đặc trưng của từng loại bệnh.

Mô hình nhận diện khuôn mặt trong hệ thống chấm công được xây dựng dựa trên thuật toán LBPH (Local Binary Patterns Histograms) – một phương pháp đơn giản nhưng hiệu quả trong bài toán nhận diện khuôn mặt. LBPH hoạt động bằng cách trích xuất các đặc trưng cục bộ từ hình ảnh khuôn mặt thông qua biểu diễn nhị phân của các pixel lân cận, sau đó xây dựng histogram đặc trưng để so sánh với cơ sở dữ liệu đã huấn luyện.

So với các mô hình học sâu như CNN, LBPH có ưu điểm là nhẹ, dễ huấn luyện, không yêu cầu GPU, và hoạt động ổn định trong các ứng dụng thời gian thực như chấm công. Trong đồ án này, mô hình LBPH được tích hợp trực tiếp trong thư viện OpenCV, và đã được huấn luyện bằng ảnh khuôn mặt thu thập từ webcam trong quá trình đăng ký người dùng.

Ưu điểm của học sâu là khả năng hoạt động hiệu quả trên dữ liệu phi cấu trúc như ảnh, âm thanh, văn bản và không yêu cầu phải thiết kế thủ công đặc trưng. Tuy nhiên, nhược điểm là cần lượng dữ liệu lớn, tài nguyên tính toán cao, và dễ bị quá khớp nếu không được huấn luyện cẩn thận.

### **2.3.2. Sk-learn**

Scikit-learn (sklearn) là một thư viện mã nguồn mở trong Python, chuyên dùng cho các tác vụ học máy như phân loại, hồi quy, gom nhóm, chọn đặc trưng và đánh giá mô hình. Mặc dù không được sử dụng để xây dựng mô hình CNN (vốn dùng PyTorch), sklearn đóng vai trò quan trọng trong đánh giá mô hình.

Trong đồ án này, các hàm từ sklearn.metrics được dùng để:

- Tính độ chính xác (accuracy) của mô hình.
- Sinh ma trận nhầm lẫn (confusion matrix) để đánh giá chi tiết kết quả phân loại.
- Tạo báo cáo phân loại (classification report) cho từng lớp bệnh.

Scikit-learn có cú pháp dễ hiểu, tài liệu đầy đủ và rất hiệu quả cho các bước xử lý sau huấn luyện mô hình học sâu.

## CHƯƠNG 3. XÂY DỰNG HỆ THỐNG CHẤM CÔNG BẰNG NHẬN DIỆN KHUÔN MẶT

### 3.1. Xây dựng bộ dữ liệu ảnh khuôn mặt

Dữ liệu đóng vai trò then chốt trong việc huấn luyện và đánh giá hiệu quả của các mô hình học sâu. Với bài toán nhận diện khuôn mặt phục vụ cho chấm công, bộ dữ liệu hình ảnh cần đảm bảo sự đa dạng về biểu cảm khuôn mặt, góc chụp, điều kiện ánh sáng và độ phân giải để mô hình có thể học được các đặc trưng phân biệt giữa các cá nhân khác nhau.

#### 3.1.1. Mục tiêu

Xây dựng một bộ dữ liệu ảnh khuôn mặt của nhân viên để phục vụ quá trình huấn luyện mô hình nhận diện khuôn mặt. Mỗi nhân viên được chụp nhiều ảnh ở các tư thế khác nhau để tăng tính đa dạng của tập dữ liệu.

#### 3.1.2. Nguồn dữ liệu

Dữ liệu được thu thập trực tiếp từ camera máy tính hoặc camera điện thoại (qua IP stream), thông qua phần mềm chấm công được phát triển. Người dùng sẽ đối mặt với camera để hệ thống tự động phát hiện và ghi nhận khuôn mặt.

#### 3.1.3. Phương pháp thu thập

- Sử dụng thư viện OpenCV (cv2.VideoCapture) để mở luồng video từ webcam.
- Sử dụng bộ phát hiện khuôn mặt Haar Cascade (haarcascade\_frontalface\_default.xml) để xác định vùng khuôn mặt.
- Cắt và lưu ảnh khuôn mặt về thư mục tương ứng với từng ID người dùng.

```
dataset_path = "dataset"
```

```
face_classifier = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

```
...
```

```
face = face_classifier.detectMultiScale(gray, ...)
```

```
cv2.imwrite(f"{dataset_path}/user_{id}_{img_id}.jpg", face_crop)
```

#### 3.1.4. Quy trình cụ thể

- Người dùng nhập ID và tên qua giao diện.
- Hệ thống mở webcam và hiển thị khung ảnh liên tục.
- Khi phát hiện khuôn mặt, hệ thống tự động:

- Cắt vùng mặt.
- Ghi ảnh thành file .jpg theo định dạng user\_<id>\_<số thứ tự>.jpg.
- Thu thập khoảng 60 ảnh cho mỗi người để đảm bảo chất lượng huấn luyện.
- Tất cả ảnh được lưu trữ trong thư mục dataset/.

### 3.2. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là bước quan trọng giúp mô hình học sâu hoạt động hiệu quả, đảm bảo chất lượng đầu vào đồng nhất, loại bỏ nhiễu và tăng cường khả năng nhận diện chính xác. Với hệ thống chấm công bằng nhận diện khuôn mặt, quá trình tiền xử lý ảnh được thực hiện ngay sau khi thu thập ảnh từ webcam, trước khi đưa vào mô hình nhận diện.

#### 3.2.1. Phát hiện và cắt khuôn mặt

Sau khi thu thập dữ liệu, hệ thống sử dụng thư viện OpenCV để phát hiện khuôn mặt trong khung hình và cắt phần khuôn mặt ra khỏi ảnh gốc. Cụ thể, thuật toán Haar Cascade Classifier được sử dụng để phát hiện khuôn mặt trong thời gian thực:

```
face_classifier = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

Hàm face\_extractor được định nghĩa để trích xuất khuôn mặt từ khung hình:

```
def face_extractor(img):  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    faces = face_classifier.detectMultiScale(gray, 1.3, 5)  
    if faces is ():  
        return None  
    for (x, y, w, h) in faces:  
        cropped_face = img[y:y+h, x:x+w]  
    return cropped_face
```

- Ảnh được chuyển sang thang độ xám (cv2.cvtColor) để tăng hiệu quả phát hiện.
- Các khuôn mặt được phát hiện dựa trên mô hình Haar cascade.
- Phần khuôn mặt được cắt ra và lưu lại để sử dụng trong quá trình huấn luyện hoặc nhận diện.

#### 3.2.2. Chuẩn hóa ảnh khuôn mặt

Sau khi khuôn mặt được trích xuất, ảnh sẽ trải qua các bước chuẩn hóa để đảm bảo kích thước và định dạng đồng nhất. Cụ thể:

- Resize ảnh về kích thước cố định 200x200 pixel để phù hợp với đầu vào của mô hình:

```
face = cv2.resize(face, (200, 200))
```

- Chuyển ảnh sang thang độ xám để giảm số chiều dữ liệu và tập trung vào đặc trưng hình dáng khuôn mặt:

```
face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
```

### 3.2.3. Lưu dữ liệu đã xử lý

- Các ảnh sau khi xử lý sẽ được lưu vào thư mục faces/user/ với định dạng .jpg. Mỗi ảnh có tên tương ứng với số thứ tự để thuận tiện cho việc huấn luyện mô hình:

```
file_name_path = "./faces/user/" + str(count) + ".jpg"
```

```
cv2.imwrite(file_name_path, face)
```

- Việc lưu ảnh theo định dạng chuẩn sẽ giúp dễ dàng tạo tập dữ liệu đầu vào cho quá trình huấn luyện mô hình nhận diện khuôn mặt với thư viện cv2.face.LBPHFaceRecognizer\_create().

## 3.3. Xây dựng mô hình

Để xây dựng mô hình nhận diện khuôn mặt trong phần mềm chấm công, nhóm sử dụng thuật toán LBPH (Local Binary Patterns Histograms) – một phương pháp truyền thống nhưng hiệu quả trong việc nhận dạng khuôn mặt nhờ khả năng trích xuất đặc trưng cục bộ và phân biệt được các khuôn mặt khác nhau trong điều kiện ánh sáng và góc nhìn thay đổi.

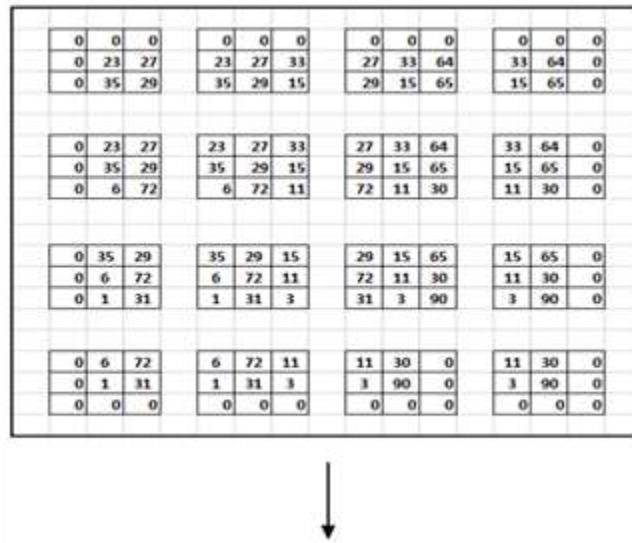
### 3.3.1. Kiến trúc mô hình

Hệ thống sử dụng mô hình **LBPH (Local Binary Patterns Histogram)**, một thuật toán nhận diện khuôn mặt đơn giản nhưng hiệu quả, đặc biệt phù hợp với ứng dụng trong điều kiện thực tế như ánh sáng thay đổi hoặc khuôn mặt ở các góc độ khác nhau.

Thuật toán LBPH hoạt động bằng cách:

- Phân chia ảnh thành các lưới ô nhỏ.
- Mỗi ô sẽ tạo ra một histogram biểu diễn mẫu texture dựa trên các nhị phân cục bộ.
- Ghép nối tất cả histogram lại để tạo thành đặc trưng tổng thể của khuôn mặt.

- So sánh histogram đầu vào với histogram đã lưu từ các mẫu để xác định danh tính.



**Hình 3.1. Mô hình LBPH**

### 3.3.2. Quá trình huấn luyện mô hình

Dữ liệu huấn luyện là tập ảnh khuôn mặt được xử lý và lưu trữ trong thư mục faces/user. Tập dữ liệu này được đưa vào huấn luyện như sau:

```
model.train(faces, np.array(ids))
```

Trong đó:

- faces: là danh sách các ảnh khuôn mặt đã được chuẩn hóa (200x200 pixels, grayscale).
- ids: là danh sách các nhãn (label) tương ứng, đại diện cho từng người (người số 1, số 2...).

Sau khi huấn luyện, mô hình sẽ được lưu lại để phục vụ cho việc nhận diện ở bước chấm công:

```
model.save("face_recognizer.xml")
```

### 3.3.3. Nhận diện khuôn mặt

Khi người dùng đứng trước camera để chấm công, hệ thống thực hiện:

- Mở webcam, thu ảnh theo thời gian thực.
- Dùng haarcascade\_frontalface\_default.xml để phát hiện khuôn mặt.
- Cắt ảnh khuôn mặt, chuyển về grayscale, resize về 200x200.
- Dùng mô hình LBPH đã huấn luyện để nhận diện:

```
label, confidence = model.predict(face)
```



Nếu độ confidence (tin cậy) nhỏ hơn một ngưỡng nhất định (ví dụ: 50), hệ thống coi là nhận diện thành công và hiển thị tên người dùng cùng thời gian chấm công.

if confidence < 50:

```
cv2.putText(frame, "Nhan vien: {}".format(label), ...)
```

### 3.3.4. Ưu điểm của mô hình LBPH

- Tốc độ xử lý nhanh, thích hợp với ứng dụng thời gian thực như chấm công.
- Khả năng nhận diện tốt ngay cả với điều kiện ánh sáng thay đổi nhẹ.
- Không đòi hỏi phần cứng mạnh như các mô hình deep learning (CNN, ResNet...).

## 3.4. Xây dựng giao diện

Giao diện người dùng đóng vai trò trung gian giữa người sử dụng và hệ thống, giúp người dùng dễ dàng tương tác với phần mềm. Trong dự án này, giao diện đồ họa được xây dựng bằng thư viện **Tkinter** – một thư viện GUI chuẩn của Python, kết hợp với **OpenCV** để hiển thị hình ảnh từ camera và xử lý khuôn mặt thời gian thực.

### 3.4.1. Công nghệ sử dụng

- Tkinter: Tạo cửa sổ chính, nút bấm, nhãn (label), khung hiển thị, v.v.
- OpenCV: Xử lý hình ảnh từ webcam, nhận diện khuôn mặt, vẽ khung, hiển thị ảnh.
- PIL (Pillow): Chuyển đổi ảnh OpenCV sang định dạng phù hợp để hiển thị trong giao diện Tkinter.
- csv, os: Quản lý dữ liệu chấm công và lưu trữ ảnh khuôn mặt người dùng.



Hình 3.2. Giao diện phần mềm chấm công.

Giao diện phần mềm chấm công được thiết kế với cấu trúc đơn giản, trực quan, tập trung vào tính tiện dụng và hiệu quả sử dụng. Cụ thể:

- Thanh tiêu đề: hiển thị tên hệ thống và khẩu hiệu (“Hệ thống nhận diện khuôn mặt”) giúp người dùng dễ nhận biết chức năng ứng dụng.
- Khung xem camera (live): hiển thị trực tiếp hình ảnh từ webcam để người dùng kiểm tra khuôn mặt trước khi nhận diện.
- Nút chức năng: bao gồm các nút chính như:
  - “Train Data” – tiến hành huấn luyện dữ liệu khuôn mặt đã thu thập.
  - “Take Images” – cho phép người dùng mới đăng ký bằng cách chụp ảnh khuôn mặt.
  - “Track Images” – nhận diện và ghi nhận chấm công dựa trên hình ảnh webcam.
  - “Quit” – thoát ứng dụng.
- Khung nhập thông tin: người dùng có thể nhập ID và tên để đăng ký nhận diện khuôn mặt mới.
- Thông báo hệ thống: hiển thị thông tin hướng dẫn, kết quả nhận diện hoặc trạng thái hiện tại của hệ thống.

**Hình 3.3. Giao diện đăng ký khuôn mặt**

### Chức năng đăng ký khuôn mặt

Người dùng có thể đăng ký khuôn mặt của mình thông qua giao diện “Nhập Thông Tin”. Sau khi điền đầy đủ **Mã số** và **Họ tên**:

- Người dùng nhấn nút “**Bật Camera**” để kích hoạt webcam.
- Hệ thống sẽ tự động thu thập nhiều ảnh khuôn mặt từ các khung hình của webcam và lưu lại dưới mã số tương ứng.
- Sau khi thu thập đủ ảnh, người dùng nhấn “**Huấn Luyện**” để bắt đầu quá trình huấn luyện mô hình nhận diện khuôn mặt.
- Mô hình sẽ sử dụng các ảnh khuôn mặt đã thu thập để huấn luyện và cập nhật cơ sở dữ liệu nhận diện.

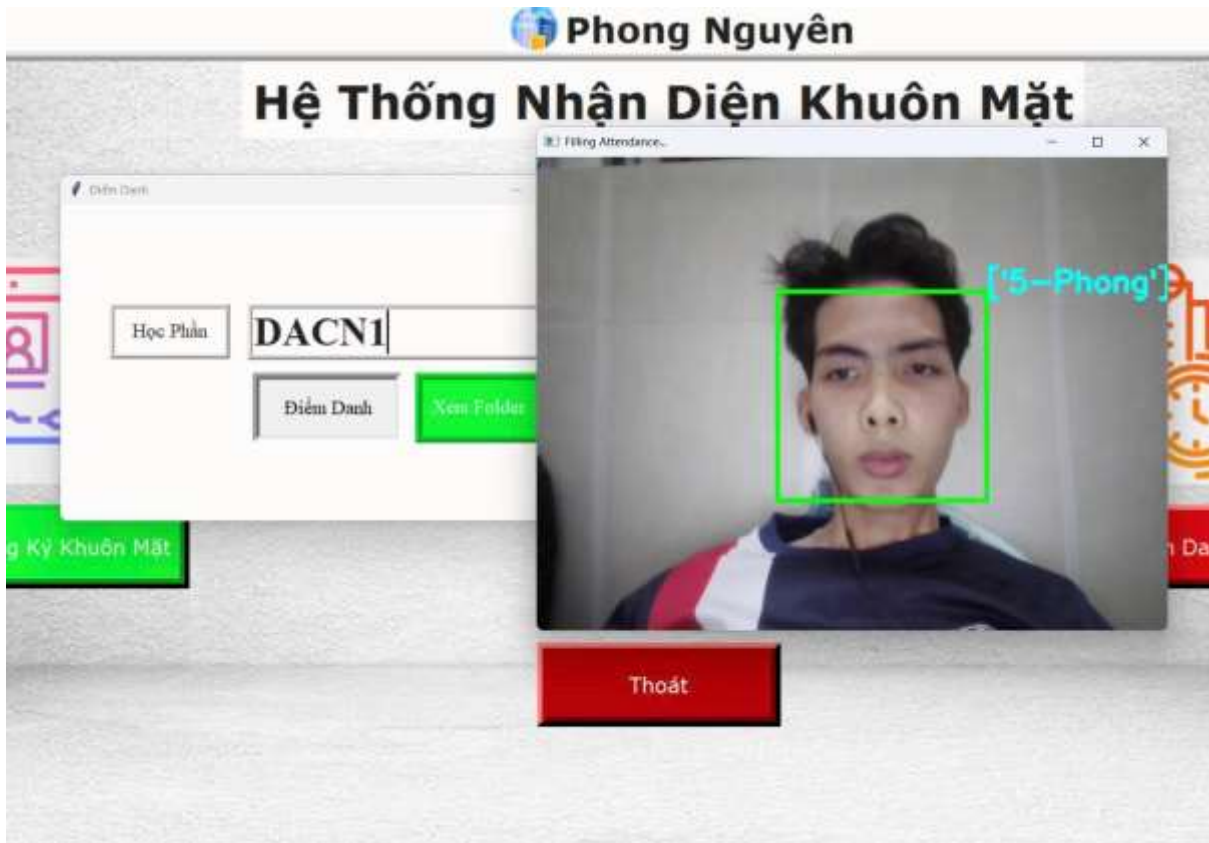
#### 3.4.2. Cấu trúc giao diện chính

Giao diện chính gồm các phần:

- Khung hiển thị ảnh webcam: Hiển thị hình ảnh thời gian thực từ camera.
- Ô nhập tên sinh viên: Người dùng nhập họ tên để thêm dữ liệu hoặc chấm công.
- Các nút chức năng:
  - Đăng ký khuôn mặt: Lưu ảnh khuôn mặt của sinh viên vào thư mục dataset/.
  - Huấn Luyện Dữ Liệu: Chạy quá trình huấn luyện mô hình LBPH dựa trên ảnh đã thu thập.
  - Điểm Danh: Mở webcam, nhận diện khuôn mặt và ghi thông tin chấm công vào file CSV.
  - Xem Danh Sách Điểm Danh: Mở file chấm công đã lưu.

#### 3.4.3. Tích hợp xử lý ảnh vào giao diện

Trong hàm `update_frame()`, ảnh từ webcam được lấy qua OpenCV và hiển thị lên giao diện Tkinter thông qua `ImageTk.PhotoImage`. Các khung màu xanh (detected face) được vẽ để hiển thị vùng khuôn mặt nhận diện được.



**Hình 3.4.** Giao diện ảnh từ webcam được lấy qua OpenCV.

#### **3.4.4. Trải nghiệm người dùng**

Giao diện được thiết kế đơn giản, dễ sử dụng:

- Người dùng chỉ cần nhập tên, nhấn nút là có thể thu thập dữ liệu hoặc chấm công.
- Giao diện phản hồi nhanh, hiển thị ảnh trực quan và cập nhật kết quả ngay trên cửa sổ ứng dụng.
- Các tệp dữ liệu như hình ảnh, mô hình, lịch sử chấm công được lưu trữ rõ ràng trong các thư mục tương ứng (dataset/, model/, attendance.csv).

## CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 4.1. Kết quả đạt được

Sau quá trình nghiên cứu, thiết kế và xây dựng hệ thống, đề tài “Xây dựng phần mềm chấm công bằng nhận diện khuôn mặt” đã đạt được những kết quả cụ thể như sau:

- + Xây dựng được mô hình nhận diện khuôn mặt dựa trên thuật toán LBPH (Local Binary Patterns Histogram), có khả năng phân biệt chính xác khuôn mặt của nhiều nhân viên trong môi trường làm việc.
- + Thu thập và xử lý bộ dữ liệu hình ảnh khuôn mặt nhân viên từ nhiều góc độ, đảm bảo đủ đa dạng để phục vụ quá trình huấn luyện và nhận diện.
- + Thiết kế giao diện người dùng trực quan bằng Tkinter, hỗ trợ việc thêm mới nhân viên, huấn luyện mô hình, chấm công và xem báo cáo lịch sử điểm danh.
- + Tích hợp thành công các thành phần quan trọng như thu thập ảnh, nhận diện, lưu dữ liệu và giao diện quản lý, đảm bảo luồng xử lý liên tục và nhanh chóng.
- + Xây dựng hệ thống lưu trữ dữ liệu chấm công và thông tin nhân viên dưới dạng file CSV, giúp quản lý dễ dàng và thuận tiện cho việc xuất báo cáo.
- + Hệ thống vận hành ổn định trên máy tính cá nhân với khả năng nhận diện khuôn mặt thời gian thực qua webcam hoặc camera IP.

Tổng thể, hệ thống đã cơ bản hoàn thành mục tiêu đề ra, tạo ra một giải pháp chấm công thông minh, tiện lợi, giúp giảm thiểu sai sót và tăng hiệu quả quản lý nhân sự.

### 4.2. Những điểm hạn chế

Mặc dù hệ thống đã đạt được nhiều kết quả tích cực, song trong quá trình phát triển và vận hành thực tế vẫn còn một số hạn chế như sau:

- + Độ chính xác nhận diện khuôn mặt có thể bị ảnh hưởng khi điều kiện ánh sáng kém hoặc khi khuôn mặt có sự thay đổi (đeo khẩu trang, kính, góc nghiêng lớn).
- + Hệ thống hiện chưa hỗ trợ nhận diện trong môi trường có nhiều người cùng lúc (đa đối tượng).
- + Dữ liệu huấn luyện khuôn mặt còn hạn chế về số lượng và độ đa dạng, chưa đủ phong phú để nâng cao khả năng nhận diện trong nhiều tình huống khác nhau.

- + Chưa tích hợp chức năng quản lý nâng cao như phân quyền người dùng, bảo mật dữ liệu cá nhân.
- + Giao diện người dùng còn đơn giản, chưa hỗ trợ các tính năng báo cáo và thống kê trực quan, thuận tiện cho quản lý nhân sự.
- + Chưa có khả năng kết nối với hệ thống chấm công hoặc cơ sở dữ liệu doanh nghiệp hiện có để đồng bộ dữ liệu.

#### **4.3. Hướng nghiên cứu**

Để hoàn thiện và nâng cao hiệu quả ứng dụng trong thực tế, nhóm đề xuất một số hướng phát triển sau:

- + Mở rộng và đa dạng hóa bộ dữ liệu khuôn mặt: thu thập thêm hình ảnh từ nhiều điều kiện môi trường khác nhau, nhiều biểu cảm và góc chụp khác nhau để tăng khả năng nhận diện chính xác.
- + Nâng cấp thuật toán nhận diện bằng các mô hình học sâu (Deep Learning) như FaceNet, MTCNN, hoặc các mạng CNN tiên tiến để tăng độ chính xác và khả năng nhận diện trong điều kiện khó khăn.
- + Phát triển tính năng nhận diện đa đối tượng cùng lúc trong khu vực camera giám sát.
- + Xây dựng giao diện quản lý nâng cao, hỗ trợ báo cáo trực quan, thống kê chi tiết về giờ giấc đi làm, vắng mặt, trễ giờ.
- + Tích hợp hệ thống bảo mật, phân quyền người dùng để bảo vệ dữ liệu cá nhân và đảm bảo an toàn thông tin.
- + Kết nối hệ thống với phần mềm quản lý nhân sự hoặc ERP hiện có của doanh nghiệp để đồng bộ dữ liệu tự động.
- + Phát triển ứng dụng trên nền tảng di động, giúp người quản lý có thể theo dõi và quản lý chấm công mọi lúc, mọi nơi.

## KẾT LUẬN

Báo cáo này đã trình bày toàn diện quá trình nghiên cứu, phân tích, thiết kế và triển khai phần mềm **chăm công bằng nhận diện khuôn mặt**, từ khâu lựa chọn công nghệ, xây dựng mô hình nhận diện cho đến thiết kế giao diện và kết nối các thành phần hệ thống. Hệ thống được xây dựng với mục tiêu hiện đại hóa và tự động hóa quy trình chăm công trong môi trường doanh nghiệp, góp phần nâng cao hiệu quả quản lý nhân sự, giảm thiểu sự phụ thuộc vào phương pháp thủ công truyền thống vốn dễ xảy ra sai sót và gian lận.

Trong suốt quá trình phát triển, nhóm đã ứng dụng các công nghệ như OpenCV, Python (Tkinter), và các thuật toán nhận diện khuôn mặt như LBPH để xây dựng nên một hệ thống có khả năng hoạt động ổn định, giao diện thân thiện, và phản hồi thời gian thực. Các chức năng thiết yếu như thu thập ảnh khuôn mặt, huấn luyện mô hình, nhận diện và lưu trữ dữ liệu chăm công đều đã được tích hợp và thử nghiệm thành công trong môi trường thực tế.

Tuy nhiên, hệ thống vẫn còn một số điểm hạn chế nhất định như độ chính xác còn phụ thuộc nhiều vào điều kiện ánh sáng và góc chụp, chưa có khả năng nhận diện nhiều người cùng lúc hay phân quyền quản trị nâng cao. Đây sẽ là tiền đề để nhóm tiếp tục nghiên cứu và cải tiến trong các giai đoạn tiếp theo.

Với nền tảng kỹ thuật vững chắc đã được xây dựng, hệ thống chăm công bằng nhận diện khuôn mặt có tiềm năng ứng dụng thực tiễn cao trong các doanh nghiệp, trường học, tổ chức công lập,... không chỉ giúp tiết kiệm thời gian, nâng cao tính minh bạch mà còn hỗ trợ quá trình chuyển đổi số trong quản lý nhân sự.

Nhóm phát triển cam kết sẽ tiếp tục cập nhật các công nghệ mới, nâng cấp tính năng và tối ưu hiệu suất hệ thống nhằm đáp ứng tốt hơn nhu cầu thực tiễn, hướng đến một giải pháp chăm công thông minh, hiệu quả và bền vững trong thời đại công nghệ 4.0.

## TÀI LIỆU THAM KHẢO

- [1]. OpenCV Documentation - <https://docs.opencv.org>
- [2]. Tkinter GUI Programming Documentation -  
<https://docs.python.org/3/library/tkinter.html>
- [3]. Machine Learning Library- <http://dlib.net>
- [4]. Python CSV Module Documentation - <https://docs.python.org/3/library/csv.html>