

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM

KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ỨNG DỤNG CHẤM CÔNG BẰNG NHẬN DẠNG KHUÔN MẶT

SỬ DỤNG DEEP LEARNING

Giảng viên hướng dẫn: ThS. Hồ Đắc Quán

Sinh viên thực hiện:

Trần Quang Trung

17009171

Hồ Duy Quang

17014761

TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2021

INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY

FACULTY OF INFORMATION TECHNOLOGY

TIMEKEEPING APPLICATION BY FACIAL RECOGNITION USING DEEP LEARNING

Major: Computer Science

Instructor: MSc. Ho Dac Quan

Student:

Tran Quang Trung

17009171

Ho Duy Quang

17014761

HO CHI MINH CITY, JUNE 2021

TIMEKEEPING APPLICATION BY FACIAL RECOGNITION USING DEEP LEARNING

ABSTRACT

Reason for choosing the topic:

At the present time, due to the complicated situation of Covid-19 pandemic in the world, it is very necessary to limit contact with public devices. However, workers in enterprises still have to use public devices to time attendance every day, especially Biometric Fingerprint Time Clocks or Magnetic Stripe Cards. Companies mainly use these two types of public devices because they have high accuracy - each person will have different fingerprint or different barcode card, these technologies ensure high accuracy and security. On the other hand, besides the advantages of these two attendance devices, during the current epidemic, the frequent use of Biometric Fingerprint Time Clocks or Magnetic Stripe Cards can make employees get viruses when interacting with public devices. Because of the above reasons, our group has learned, researched and wanted to implement the topic “Timekeeping Application by Facial recognition using Deep learning”.

Problems:

- Manage the real time employees take attendance.
- Identify and ensure the correct employee for attendance.
- Save attendance data of employees to calculate salaries for employees.

Methods:

There are a lot of ways to study and implement this research. However, our group mainly used technologies: Visual Studio Code, Google Colab, Python 3.8, Tkinter,... and read a lot documents which related to the research to learn the problems encountered. Then, we learned from experience and built a realtime face attendance system.

LỜI CAM ĐOAN

Chúng tôi xin cam đoan việc nghiên cứu đề tài được thực hiện bởi nhóm chúng tôi cùng với giáo viên hướng dẫn luận án và chưa được sử dụng để báo cáo cho bất kì khóa luận hay môn học nào.

Chúng tôi xin cam đoan mọi sự giúp đỡ đã được cảm ơn, các thông tin trích dẫn luận văn đã được ghi chú đầy đủ và bảo đảm rõ nguồn gốc không xâm phạm quyền tác giả.

TP.Hồ Chí Minh, ngày..... tháng 06 năm 2021

TÁC GIẢ KHÓA LUẬN

Hồ Duy Quang

Trần Quang Trung

LỜI CẢM ƠN

Quá trình thực hiện báo cáo tốt nghiệp được xem là cột mốc quan trọng nhất trong quãng đời sinh viên của mỗi con người vì đây sẽ là tiền đề nhằm trang bị cho chúng em những kiến thức, kỹ năng nghiên cứu và đặc biệt là những kinh nghiệm quý báu trước khi bước ra xã hội lập nghiệp. Để hoàn thành tốt được đề tài “Ứng dụng chấm công khuôn mặt áp dụng công nghệ Deep learning“, chúng em đã nhận được rất nhiều sự giúp đỡ nhiệt tình từ nhà trường, quý thầy cô, gia đình, bạn bè và đồng nghiệp.

Trước hết, chúng em xin chân thành cảm ơn Ban lãnh đạo khoa Khoa học máy tính của trường Đại học Công nghiệp thành phố Hồ Chí Minh đã tạo điều kiện thuận lợi cho chúng em được học tập và thực hiện đề tài báo cáo tốt nghiệp này.

Bên cạnh đó, chúng em xin gửi lời cảm ơn đến quý thầy cô trong khoa Khoa học máy tính đã tận tâm giảng dạy, truyền đạt kiến thức, trang bị cho chúng em những kinh nghiệm quý báu trong những năm học vừa qua.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Ths. Hồ Đắc Quán – giảng viên trường Đại học Công nghiệp Thành phố Hồ Chí Minh đã luôn tận tình hướng dẫn, chỉ bảo và giúp đỡ chúng em trong suốt quá trình hoàn thành báo cáo tốt nghiệp. Những lời hướng dẫn, góp ý của thầy vô cùng trân quý, thúc đẩy chúng em không ngừng cố gắng để có thể hoàn thành tốt khóa luận này.

Cuối cùng, chúng em rất mong nhận được sự đóng góp ý kiến, nhận xét và sự chỉ dẫn của các quý thầy cô giáo.

TP. Hồ Chí Minh, ngày....., tháng 06 năm 2021

Nhóm thực hiện

Hồ Duy Quang

Trần Quang Trung

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày.... tháng 06 năm 2021

GIÁO VIÊN HƯỚNG DẪN

ThS. HỒ ĐẮC QUÁN

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIỆN 1

[illegible]

TP. Hồ Chí Minh, ngày.... tháng 06 năm 2021

GIÁO VIÊN PHẢN BIỆN 1

NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIÁO VIÊN PHẢN BIÊN 2

[illegible]

TP. Hồ Chí Minh, ngày.... tháng 06 năm 2021

GIÁO VIÊN PHẢN BIỆN 2

MỤC LỤC

DANH MỤC HÌNH ẢNH	1
DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT	4
LỜI NÓI ĐẦU	5
CHƯƠNG I. PHẦN MỞ ĐẦU	6
1.1 Tổng quan và mô tả đề tài khóa luận	6
1.1.1 Tổng quan	6
1.1.2 Mô tả đề tài khóa luận	8
1.2 Mục tiêu	10
1.3 Các yêu cầu chức năng	10
CHƯƠNG II. CƠ SỞ LÝ THUYẾT	11
2.1 Convolutional Neural Network (CNNs)	11
2.1.1 Lớp tích chập (Convolution Layer)	11
2.1.2 Lớp tổng hợp (Pooling Layer)	12
2.1.3 Hàm phi tuyến – ReLU (Rectified Linear Unit)	13
2.1.4 Fully Connected	13
2.2 One-shot Learning	15
2.3 Learning Similarity	15
2.4 Siamese Network	17
2.5 Hàm mất mát Triplet Loss	19
2.6 Cách chọn bộ ba ảnh của Triplet Loss	22
2.7 Công nghệ phát hiện khuôn mặt với MediaPipe	23
2.7.1 Tổng quát	23
2.7.2 Đầu ra của MediaPipe	23
2.7.3 Hướng dẫn cài đặt trên python	24
2.7.4 Kết quả nhận diện	25

2.7.5	<i>Nhận xét</i>	25
CHƯƠNG III. XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH		26
3.1	Xây dựng mạng Siamese Network với kiến trúc VGG16.	26
3.2	Bộ dữ liệu dùng để huấn luyện cho đề tài	28
3.3	Giới thiệu về mạng pretrained FaceNet	29
3.4	Đánh giá mô hình tự train sử dụng kiến trúc mạng VGG16.	31
3.4.1	<i>Kết quả training sau 450 epochs.</i>	31
3.4.2	<i>Đánh giá độ chính xác của mô hình VGG16</i>	31
3.5	Đánh giá mô hình pretrain FaceNet	33
CHƯƠNG IV. PHÂN TÍCH VÀ XÂY DỰNG ỨNG DỤNG		34
4.1	Mô tả User Case ứng dụng	34
4.1.1	<i>Mô hình Use case</i>	34
4.1.2	<i>Danh sách các tác nhân và mô tả</i>	50
4.1.3	<i>Danh sách Use case và mô tả</i>	50
4.2	Đặc tả các yêu cầu chức năng	51
4.2.1	<i>UC001_DangNhap</i>	51
4.2.2	<i>UC002_DanhSachNhanVien</i>	53
4.2.3	<i>UC003_TimKiemNhanVien</i>	54
4.2.4	<i>UC004_ThemNhanVien</i>	55
4.2.5	<i>UC005_ChiTietNhanVien</i>	56
4.2.6	<i>UC006_CapNhatNhanVien</i>	58
4.2.7	<i>UC007_ThemKhuonMat</i>	59
4.2.8	<i>UC008_XemDuLieuKhuonMat</i>	61
4.2.9	<i>UC009_HuanLuyenKhuonMat</i>	62
4.2.10	<i>UC010_DanhSachChamCong</i>	63
4.2.11	<i>UC011_LocChamCongTheoNgay</i>	64
4.2.12	<i>UC012_Checkin</i>	65
4.2.13	<i>UC013_Checkout</i>	67

4.2.14	<i>UC014_ChiTietChamCong</i>	68
4.2.15	<i>UC015_CapNhatChamCong</i>	70
4.3	Chuẩn bị	72
4.3.1	<i>Môi trường</i>	72
4.3.2	<i>Thư viện</i>	72
4.4	Kết quả chương trình	73
4.4.1	<i>Màn hình đăng nhập</i>	73
4.4.2	<i>Màn hình Dashboard</i>	74
4.4.3	<i>Màn hình danh sách nhân viên</i>	76
4.4.4	<i>Màn hình thêm nhân viên</i>	77
4.4.5	<i>Màn hình thêm khuôn mặt</i>	80
4.4.6	<i>Màn hình xem dữ liệu khuôn mặt</i>	81
4.4.7	<i>Màn hình chi tiết nhân viên</i>	81
4.4.8	<i>Màn hình chấm công</i>	82
4.4.9	<i>Màn hình check-in</i>	83
4.4.10	<i>Màn hình check-out</i>	84
4.4.11	<i>Màn hình chi tiết chấm công</i>	85
CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		86
5.1	Kết quả đạt được	86
5.2	Hạn chế của khóa luận	86
5.3	Hướng phát triển	87
TÀI LIỆU THAM KHẢO		88

DANH MỤC HÌNH ẢNH

Hình 2.1. Ảnh minh họa bài toán tích chập hai ma trận	11
Hình 2.2. Ảnh minh họa lớp tích chập (tác giả Shashi Rekha)	12
Hình 2.3. Ảnh minh họa lớp Pooling	12
Hình 2.4. Ảnh minh họa lớp ReLU	13
Hình 2.5. Ảnh minh họa cho lớp Fully Connected	14
Hình 2.6. Ảnh minh họa mô hình CNN hoàn chỉnh	14
Hình 2.7. Ảnh minh họa thuật toán Learning Similarity	16
Hình 2.8. Ảnh minh họa mô tả Learning Similarity	17
Hình 2.9. Ảnh minh họa luồng xử lý tính khoảng cách hai bức ảnh	19
Hình 2.10. Mô tả bài toán Triplet Loss	20
Hình 2.11. Ảnh minh họa cho thuật toán Triplet Loss	20
Hình 2.12. Ảnh minh họa cách chọn ảnh Negative	23
Hình 2.13. Kết quả mô hình phát hiện khuôn mặt MediaPipe	25
Hình 3.1. Kiến trúc mạng VGG16	26
Hình 3.2. Kiến trúc mạng được pre-trained dựa trên VGG16	27
Hình 3.3. Bộ dữ liệu khuôn mặt VN-Celeb	28
Hình 3.4. Biểu đồ hiển thị số khuôn mặt của mỗi người	29
Hình 3.5. Bảng thiết kế kiến trúc của mạng FaceNet năm 2015	30
Hình 3.6. Biểu đồ Loss sau khi huấn luyện 450 epochs	31
Hình 3.7. Biểu đồ thể hiện độ chính xác mô hình dự đoán hai người	32
Hình 3.8. Biểu đồ thể hiện độ chính xác mô hình dự đoán ba người	32
Hình 3.9. Biểu đồ hiển thị độ chính xác FaceNet dự đoán ba người	33
Hình 4.1. Biểu đồ usecase ứng dụng chấm công bằng nhận dạng khuôn mặt	34
Hình 4.2. Biểu đồ activity đăng nhập	35
Hình 4.3. Biểu đồ activity danh sách nhân viên	36
Hình 4.4. Biểu đồ activity tìm kiếm nhân viên	37
Hình 4.5. Biểu đồ activity thêm nhân viên	38
Hình 4.6. Biểu đồ activity chi tiết nhân viên	39

Hình 4.7. Biểu đồ activity cập nhật nhân viên	40
Hình 4.8. Biểu đồ activity thêm khuôn mặt nhân viên	41
Hình 4.9. Biểu đồ activity xem dữ liệu khuôn mặt	42
Hình 4.10. Biểu đồ activity huấn luyện khuôn mặt	43
Hình 4.11. Biểu đồ activity danh sách chấm công.....	44
Hình 4.12. Biểu đồ activity lọc danh sách chấm công	45
Hình 4.13. Biểu đồ activity check-in	46
Hình 4.14. Biểu đồ activity check-out	47
Hình 4.15. Biểu đồ activity chi tiết chấm công nhân viên	48
Hình 4.16. Biểu đồ activity cập nhật chi tiết chấm công	49
Hình 4.17. Biểu đồ sequence đăng nhập.....	52
Hình 4.18. Biểu đồ sequence danh sách nhân viên.....	53
Hình 4.19. Biểu đồ sequence tìm kiếm nhân viên	54
Hình 4.20. Biểu đồ sequence thêm nhân viên.....	56
Hình 4.21. Biểu đồ sequence chi tiết nhân viên.....	57
Hình 4.22. Biểu đồ sequence cập nhật thông tin nhân viên	59
Hình 4.23. Biểu đồ sequence thêm khuôn mặt	60
Hình 4.24. Biểu đồ sequence xem dữ liệu khuôn mặt	62
Hình 4.25. Biểu đồ sequence huấn luyện mô hình nhận dạng.....	63
Hình 4.26. Biểu đồ sequence danh sách chấm công	64
Hình 4.27. Biểu đồ sequence lọc danh sách chấm công theo ngày	65
Hình 4.28. Biểu đồ sequence check-in.....	66
Hình 4.29. Biểu đồ sequence check-out.....	68
Hình 4.30. Biểu đồ sequence chi tiết chấm công	69
Hình 4.31. Biểu đồ sequence cập nhật thông tin chấm công	71
Hình 4.32. Giao diện đăng nhập ứng dụng	73
Hình 4.33. Giao diện lỗi đăng nhập	74
Hình 4.34. Giao diện bảng điều khiển	75
Hình 4.35. Giao diện danh sách nhân viên	76
Hình 4.36. Kết quả tìm kiếm nhân viên	76

Hình 4.37. Giao diện thêm nhân viên	77
Hình 4.38. Giao diện lỗi nhập dữ liệu khi thêm nhân viên	78
Hình 4.39. Hình ảnh đã nhập đầy đủ thông tin	79
Hình 4.40. Danh sách sau khi thêm nhân viên	79
Hình 4.41. Giao diện màn hình thêm khuôn mặt	80
Hình 4.42. Giao diện xem dữ liệu khuôn mặt	81
Hình 4.43. Giao diện chi tiết nhân viên	81
Hình 4.44. Giao diện danh sách chấm công	82
Hình 4.45. Kết quả sau khi lọc ngày chấm công	82
Hình 4.46. Giao diện màn hình check-in	83
Hình 4.47. Kết quả sau khi check-in	83
Hình 4.48. Giao diện màn hình check-out	84
Hình 4.49. Kết quả sau khi check-in và check-out	85
Hình 4.50. Giao diện chi tiết chấm công	85

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh
A	Anchor
P	Positive
N	Negative
CNN	Convolutional Neural Network
ReLU	Rectified Linear Units
NN	Neural Network
API	Application Programming Interface

LỜI NÓI ĐẦU

Ngày nay, cùng với những phát triển vượt bậc về mặt khoa học công nghệ, trí tuệ nhân tạo đã ngày càng phát triển và thu được những thành tựu to lớn chứng tỏ được vai trò quan trọng trong xã hội và đời sống. Một trong các lĩnh vực trí tuệ nhân tạo là lĩnh vực về nhận dạng khuôn mặt, đây là lĩnh vực đã được ứng dụng rất trong nhiều lĩnh vực liên quan về y tế, giao thông, dây chuyền sản xuất, giám sát an ninh, và đặc biệt là lĩnh vực nhận dạng bằng sinh trắc học.

Phát hiện và nhận dạng khuôn mặt là lĩnh vực thú vị và nhiều thách thức. Mặc dù nhiều phương pháp hiệu quả đã được đề xuất trong thời gian gần đây, việc giải quyết bài toán tổng quát vẫn là một vấn đề còn nhiều

Với tinh thần của những sinh viên năm cuối ngành Khoa học máy tính, sự khát khao được tìm tòi, học hỏi, nghiên cứu về lĩnh vực trí tuệ nhân tạo đặc biệt là về nhận dạng khuôn mặt nhóm chúng tôi đã có mong muốn làm một đề tài về ứng dụng nhận dạng khuôn mặt.

Ngay lúc này, tình hình dịch bệnh Covid-19 hiện nay đang diễn biến phức tạp, việc tiếp xúc với các thiết bị công cộng cần hạn chế. Với doanh nghiệp, công việc chấm công của nhân viên cần phải thao tác với thiết bị công cộng như máy quét vân tay, máy quét thẻ từ,... Điều này làm cho việc đảm bảo an toàn dịch bệnh gây khó khăn, chính vì lý do này mà công nghệ chấm công bằng khuôn mặt là lựa chọn tốt nhất hiện nay với việc không phải tiếp xúc với các thiết bị công cộng, thời gian chấm công nhanh chóng và có độ chính xác ổn định. Qua quá trình nghiên cứu, nhóm chúng tôi đã biết được ứng dụng chấm công bằng nhận dạng khuôn mặt đang được rất nhiều doanh nghiệp áp dụng và sử dụng.

Từ những lý do trên mà nhóm đã quyết định nghiên cứu và xây dựng một ứng dụng chấm công bằng nhận dạng khuôn mặt sử dụng Deep learning. Hi vọng với những gì mà nhóm chúng tôi nghiên cứu và thực hiện có thể đóng góp cho việc phát triển và tham khảo để phục vụ cho những bước phát triển sau này của ứng dụng chấm công bằng khuôn mặt.

CHƯƠNG I. PHẦN MỞ ĐẦU

1.1 Tổng quan và mô tả đề tài khóa luận

1.1.1 Tổng quan

Hệ thống nhận dạng khuôn mặt (Facial recognition system) [2] là công nghệ có khả năng xác định danh tính dựa vào khuôn mặt của đối tượng. Hệ thống nhận dạng khuôn mặt được sử dụng xác thực danh tính của một người nào đó trong tấm ảnh, videos hoặc thời gian thực (real-time).

Nhận dạng khuôn mặt là một loại bảo mật sinh trắc học, các loại sinh trắc học hiện nay vẫn đang được sử dụng như nhận dạng vân tay, giọng nói, móng mắt. Công nghệ này chủ yếu được dùng trong bảo mật và thực thi pháp luật, nhưng ngày nay công nghệ nhận dạng khuôn mặt đang được sử dụng nhiều hơn ở các lĩnh vực khác.

Ban đầu hệ thống là một dạng ứng dụng trong máy tính, nhưng với sự phát triển của khoa học công nghệ ngày nay, hệ thống nhận dạng khuôn mặt đã được ứng dụng và triển khai trên nhiều nền tảng hơn như: điện thoại thông minh (smart phone), Iot, robot và các công nghệ phần cứng khác.

Hệ thống nhận dạng khuôn mặt thường được sử dụng để xác thực danh tính người dùng thông qua Face ID của iPhone hay iPad, máy giám sát tự động (automatic face tracking), máy chấm công tự động, quản lý ra vào.

Nhận dạng khuôn mặt là một bài toán phức tạp, nó đòi hỏi xử lý các vấn đề:

- Phát hiện tất cả các khuôn mặt có trong bức hình.
- Tập trung vào từng khuôn mặt đảm bảo rằng có thể nhận biết được đó là cùng một người từ các góc nhìn khác nhau hoặc điều kiện sáng tối khác nhau.
- Lựa chọn các đặc trưng (feature) trên từng khuôn mặt.
- So sánh các đặc trưng này với những người khác để có thể định danh được người đó.

Ưu nhược điểm của công nghệ nhận dạng khuôn mặt.**Ưu điểm**

- Thực hiện nhận dạng hàng loạt vì nhận dạng khuôn mặt không cần tương tác của đối tượng để hoạt động.
- Là công nghệ sinh trắc học nhanh nhất.
- Trong nhiều trường hợp, đối tượng hoàn toàn không hay biết về quá trình này nên sẽ không cảm thấy “bị giám sát” hoặc cảm thấy sự riêng tư của mình bị xâm phạm.

Nhược điểm

- Khó khăn để thực hiện trong các điều kiện nhất định.

Ví dụ: Ralph Gross, một nhà nghiên cứu tại Viện Mellon Robotics Carnegie, mô tả một trở ngại liên quan đến các góc nhìn của khuôn mặt: “*Face recognition has been getting pretty good at full frontal faces and 20 degrees off, but as soon as you go towards profile, there've been problems.*”. (trích dẫn từ [2])

- Các điều kiện khác mà nhận dạng khuôn mặt không làm việc tốt được gồm thiếu ánh sáng, đeo kính mát, tóc dài hoặc các đối tượng mà một phần khuôn mặt bị che mất và các hình ảnh có độ phân giải thấp.
- Một nhược điểm nghiêm trọng nữa là nhiều hệ thống sẽ kém hiệu quả nếu biểu cảm trên khuôn mặt bị thay đổi. Ngay cả khi bạn cười cũng có thể làm hệ thống giảm độ chính xác.

Ví dụ: Canada vào năm 2009 chỉ cho phép để khuôn mặt “không cảm xúc” khi chụp ảnh hộ chiếu.

- Độ chính xác và tính khả dụng không được đảm bảo.

Ví dụ: Để có được hình ảnh đối chiếu, hệ thống yêu cầu khách hàng phải quay ít nhất 35° về phía camera và không sử dụng khẩu trang, mũ, nón, tóc phải được vén cao không quá chân mày,... Trong dịch vụ bán lẻ, yêu cầu này hoàn toàn không khả

thi. Chính vì vậy rất khó để cam kết độ chính xác về thông tin khi sử dụng camera nhận dạng khuôn mặt.

Hệ thống chấm công (Time and attendance system) được sử dụng để theo dõi và giám sát khi nhân viên bắt đầu làm việc đến khi tan làm. Hệ thống chấm công cho phép người quản lý theo dõi giờ làm việc nhân viên của họ, đến làm hay đi muộn, nghỉ làm hay về sớm.

Các hình thức của hệ thống chấm công

Hệ thống thủ công là một hệ thống cũ sử dụng “máy ghi thời gian” tên tiếng anh là Time Recorder hay được gọi là Clock Card Machine sử dụng phiếu để ghi thời gian, hiện nay hình thức này đã dần bị loại bỏ và được thay bằng các hệ thống tự động rẻ hơn.

Hệ thống tự động là hệ thống chấm công tự động hiện đại yêu cầu nhân viên thao tác (chạm, vuốt,...) để nhận dạng danh tính và ghi lại giờ làm việc khi ra vào khu vực làm việc. Các hệ thống tự động hiện đại ngày nay thường sử dụng công nghệ sinh trắc học (đầu đọc tĩnh mạch, quét vân tay, nhận dạng khuôn mặt) và các thiết bị cảm ứng.

Hệ thống trên ứng dụng là công nghệ cho phép sử dụng các tính năng của *hàng rào địa lý* (Geo-fence) dựa trên ứng dụng. Giúp nhân viên chấm công ra vào khi nhân viên ở trong phạm vi kết nối internet riêng. Hệ thống nhận dạng khuôn mặt cũng có những công nghệ phần mềm dựa trên ứng dụng này, loại bỏ sự cần thiết của máy quét vân tay.

1.1.2 Mô tả đề tài khóa luận

Với sự phát triển vượt bậc của khoa học công nghệ, sự ra đời của hệ thống chấm công bằng khuôn mặt còn có tên gọi là Face ID đã mang đến giải pháp đột phá khắc phục những hạn chế chấm công theo phương thức truyền thống. Đây được coi là một trong những phương pháp chấm công cao cấp nhất. Hệ thống chấm công nhận dạng khuôn mặt là hệ thống chấm công áp dụng công nghệ nhận dạng khuôn

mặt để xác thực danh tính khi chấm công thay cho việc chấm công bằng dấu vân tay hay thẻ từ.

Cách thức hoạt động: Khi đối tượng lọt vào phạm vi quét của camera, hệ thống sẽ tự động xác định chính xác vị trí khuôn mặt và thực hiện các thuật toán để trích xuất các đặc trưng trên khuôn mặt. Trong quá trình di chuyển, đối tượng sẽ giúp hệ thống dễ dàng xác nhận khuôn mặt hơn. Tất cả các quá trình trên diễn ra chưa tới 1 giây.

Quá trình mã sinh trắc học nhận diện khuôn mặt hoạt động bao gồm

- Lấy mẫu: Lấy ảnh mẫu của đối tượng từ camera của hệ thống.
- Phân tích: Hình ảnh được đưa vào sẽ được trích xuất các đặc trưng riêng biệt và được chuyển thành dữ liệu.
- So sánh: Các dữ liệu thu được ở trên sẽ được so sánh với các dữ liệu đã có ở cơ sở dữ liệu.
- Dự đoán kết quả: Hệ thống sau đó sẽ quyết định kết quả so sánh có phù hợp hay không.

Ưu nhược điểm của hệ thống chấm công nhận diện khuôn mặt

Ưu điểm

- Tính bảo mật cao, hạn chế tối đa việc chấm công hộ.
- Có thể chấm công cho nhiều người do người dùng không cần phải tương tác với thiết bị.
- Tốc độ xử lý nhanh, phù hợp với công ty đông nhân viên.
- Khắc phục các nhược điểm của các loại máy chấm công khác như nhân viên quên thẻ, thẻ giấy bị rách, máy quét bị trầy xước (đối với máy chấm công bằng vân tay),...

Nhược điểm

- Chi phí thiết bị khá cao so với các công nghệ vân tay hay thẻ từ.
- Khi có sự thay đổi về ánh sáng, góc chụp khuôn mặt, nét mặt, phụ kiện trên khuôn mặt đều có thể ảnh hưởng đến kết quả dự đoán.

1.2 Mục tiêu

Mục đích: Tạo ra ứng dụng “chấm công bằng nhận dạng khuôn mặt” trong thực tế. Từ đó, so sánh với các phương thức chấm công khác để đưa ra nhận xét về tính khả thi. Do đó, nội dung đề tài là tìm hiểu các phương pháp, công nghệ và công cụ để hiện thực ứng dụng chấm công nhận dạng khuôn mặt.

Đối tượng và phạm vi: Các nhân viên làm việc trong cùng một công ty.

Để đạt được mục tiêu, cần tập trung tìm hiểu

- Mô hình mạng CNNs.
- One-shot learning.
- Learning Similarity.
- Mô hình phát hiện khuôn mặt MediaPipe.
- Siamese neural network và triplet loss function.
- Mô hình nhận dạng khuôn mặt FaceNet.
- Nghiên cứu Tkinter để xây dựng ứng dụng.

1.3 Các yêu cầu chức năng

Ứng dụng chấm công bằng nhận dạng khuôn mặt có các chức năng chính như sau:

- Thêm nhân viên.
- Thêm dữ liệu khuôn mặt cho nhân viên (phát hiện khuôn mặt).
- Chỉnh sửa thông tin nhân viên.
- Huấn luyện nhận dạng khuôn mặt nhân viên.
- Nhận dạng chấm công cho nhân viên (check-in và check-out).
- Chỉnh sửa thông tin chấm công.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1 Convolutional Neural Network (CNNs)

Convolutional Neural Network [7] là một trong những thuật toán Deep Learning tốt nhất để giải quyết các bài toán về thị giác máy tính như phân lớp, nhận dạng, phát hiện đối tượng trong bức ảnh, ... Trong kiến trúc chính của mạng CNN sẽ bao gồm nhiều thành phần được kết nối với nhau theo các lớp như: Convolution, Pooling, Fully Connected, ...

2.1.1 Lớp tích chập (Convolution Layer)

Lớp tích chập là một lớp không thể thiếu khi nhắc đến mạng CNN. Bởi vì nó sẽ luôn là lớp đầu tiên để trích xuất các đặc trưng từ hình ảnh đầu vào.

Convolutional sử dụng một cửa sổ trượt (Sliding Windows) trên một ma trận và nhân tích chập với các giá trị trong cửa sổ trượt theo giá trị Kernel của lớp tích chập đó tương tự như hình bên dưới.

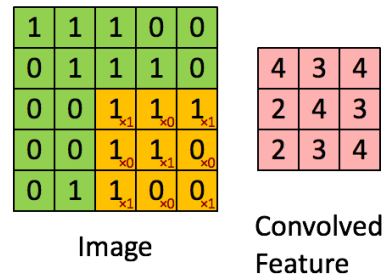
Dưới đây là một ma trận 5x5 với các giá trị 0 và 1 và được nhân tích chập với một ma trận bộ lọc (kernel matrix) 3x3

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

5 x 5 - Image Matrix
3 x 3 - Filter Matrix

Hình 2.1. Ảnh minh họa bài toán tích chập hai ma trận

Và đây là kết quả sau khi sử dụng phép tích chập lên một ma trận, ta sẽ nhận được một ma trận đặc trưng được trích xuất từ ma trận gốc.



Hình 2.2. Ảnh minh họa lớp tích chập (tác giả Shashi Rekha)

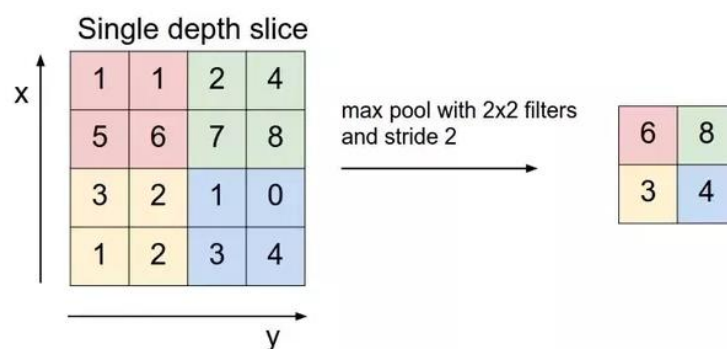
2.1.2 Lớp tổng hợp (Pooling Layer)

Lớp tổng hợp thường được sử dụng ngay sau lớp tích chập với mục đích là đơn giản hóa thông tin đầu ra, nhưng vẫn giữ lại những đặc trưng quan trọng trong ma trận với mục đích giảm bớt các số lượng dữ liệu cần tính toán cho một bức ảnh.

Lớp tổng hợp có nhiều cách thể hiện như:

- Max-Pooling (lấy các đặc trưng lớn nhất trong ma trận).
- Average-Pooling (lấy đặc trưng trung bình).
- Sum-Pooling (lấy đặc trưng tổng).

Trong đó lớp tổng hợp thường được sử dụng nhất cho việc lấy ra giá trị đặc trưng lớn nhất trong ma trận.

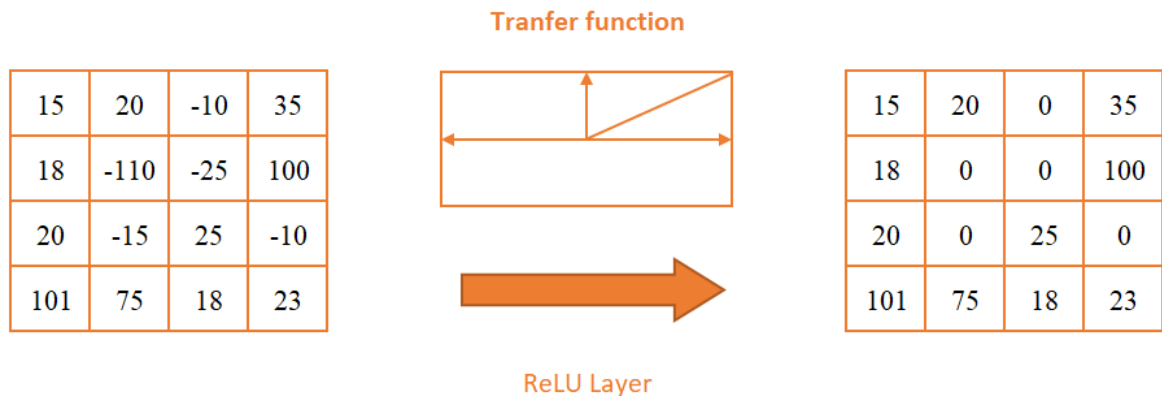


Hình 2.3. Ảnh minh họa lớp Pooling

Thông thường, lớp tổng hợp thường có kích thước là 2x2 và stride = 2 và padding = 0. Khi đó dữ liệu cần tính toán trong ma trận sẽ ít đi bởi vì kích thước của ma trận sẽ giảm đi một nửa so với ban đầu (từ 4x4 thành 2x2).

2.1.3 Hàm phi tuyến – ReLU (Rectified Linear Unit)

Hàm phi tuyến ReLU thường được sử dụng với vai trò là hàm kích hoạt trong lớp ẩn của mạng Neural như lớp tích chập, lớp tổng hợp, ... Tác dụng của hàm này là biến đổi những giá trị âm trong ma trận đặc trưng thành 0 với kích thước ma trận không đổi.

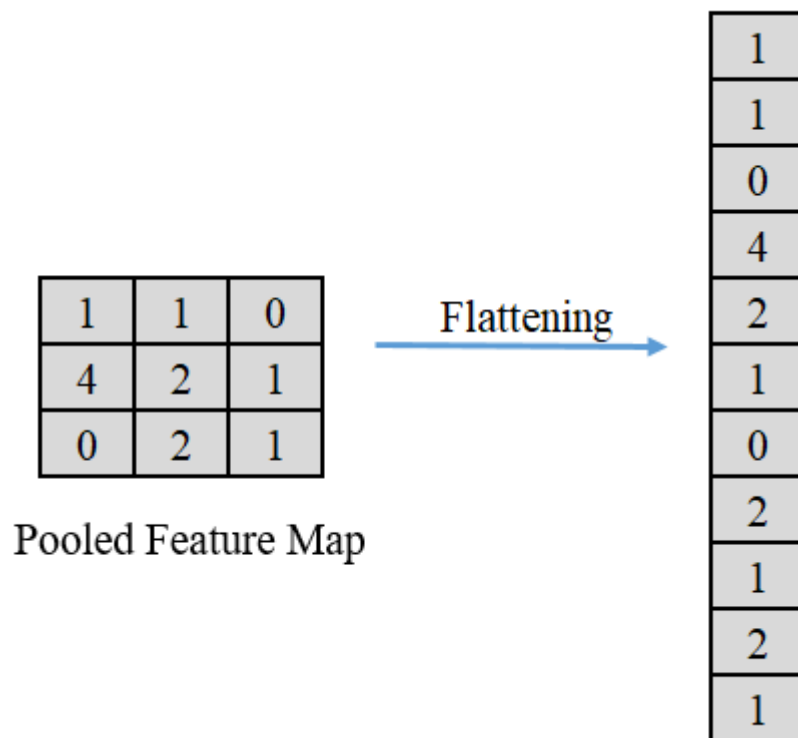


Hình 2.4. Ảnh minh họa lớp ReLU

Ngoài hàm phi tuyến ReLU chúng ta còn có một số hàm phi tuyến khác nhau Tanh, Sigmoid, Linear, ... Nhưng thông dụng nhất vẫn là hàm ReLU.

2.1.4 Fully Connected

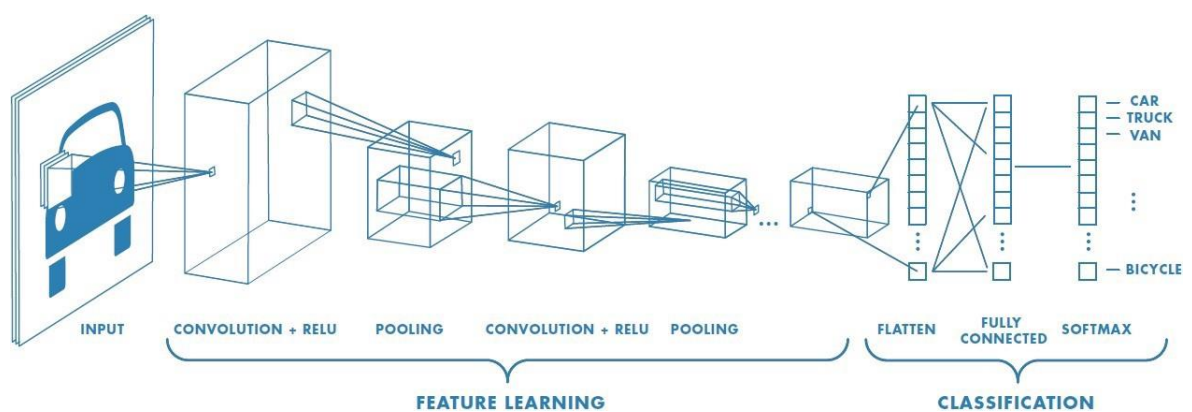
Sau khi ảnh được truyền qua nhiều lớp tích chập và lớp tổng hợp thì mô hình của chúng ta đã học được tương đối các đặc điểm của một bức ảnh (ví dụ mắt, mũi, miệng, khuôn mặt, ...) thì tensor của lớp cuối cùng với kích thước (chiều cao * chiều rộng * độ sâu) sẽ được giãn ra và chuyển thành 1 vector với kích thước (chiều cao * chiều rộng * độ sâu).



Hình 2.5. Ảnh minh họa cho lớp Fully Connected

Sau đó ta dùng lớp Fully Connected để kết hợp với các đặc điểm của ảnh để ra được kết quả phân lớp của mô hình CNN.

Dưới đây là mô hình hoàn chỉnh của một mạng CNN.



Hình 2.6. Ảnh minh họa mô hình CNN hoàn chỉnh

2.2 One-shot Learning

One-Shot Learning [3] là phương pháp học có giám sát sử dụng mô hình Convolutional Neural Network (CNN) phân lớp. Và mỗi đối tượng chỉ cần một hoặc một vài tấm ảnh là chúng ta có thể đưa vào mạng CNN để huấn luyện. Và kết quả sau khi huấn luyện là chúng ta sẽ sử dụng nó để nhận dạng những đối tượng đó lại một lần nữa. Tức là, từ một ảnh đầu vào của một đối tượng (biển báo giao thông, con chó, con mèo,...), chúng ta sẽ sử dụng mô hình được huấn luyện để dự đoán và phân lớp lại những đối tượng này.

Tuy nhiên, nhược điểm của phương pháp này là chúng ta phải huấn luyện lại mô hình mỗi khi chúng ta muốn thêm một đối tượng mới vào trong mô hình để dự đoán. Bởi vì, một mô hình CNN phân lớp sẽ có lớp Output tương ứng với số lượng lớp mà chúng ta muốn mô hình có thể dự đoán. Và đây là một nhược điểm vô cùng lớn đối với các bài toán nhận dạng khuôn mặt, bởi vì số lượng nhân viên trong công ty không cố định và luôn thay đổi theo thời gian.

Để khắc phục được nhược điểm này, chúng ta sẽ đi qua phương pháp Learning Similarity.

2.3 Learning Similarity

Thông thường, chúng ta sẽ xây dựng một mô hình Convolutional Neural Network (CNN) phân lớp để xử lý các bài toán về nhận dạng chủ thể trong bức ảnh. Và sau khi xây dựng và huấn luyện các mô hình này, chúng ta sẽ sử dụng nó để dự đoán các chủ thể có trong bức ảnh theo các lớp tương ứng mà chúng ta cần mô hình này học.

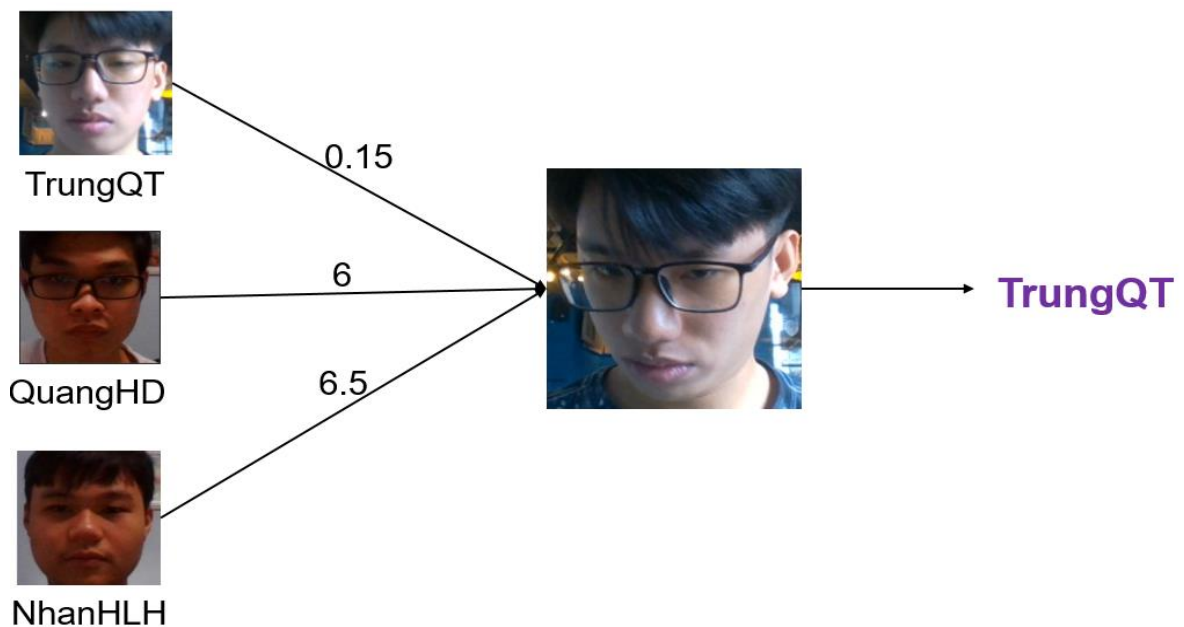
Tuy nhiên, đối với bài toán nhận dạng khuôn mặt phục vụ cho việc chấm công thì phương pháp sử dụng các mô hình CNN phân lớp sẽ không được tối ưu. Vì chúng ta không biết được số lượng nhân viên của công ty sẽ thay đổi như thế nào theo thời gian, và việc sử dụng mô hình CNN phân lớp cho trường hợp này sẽ khiến

chúng ta phải huấn luyện và chỉnh sửa lại số lượng tập phân lớp mỗi khi có sự thay đổi về nhân sự và điều này sẽ mất rất nhiều thời gian và công sức.

Vì vậy, để giải quyết được vấn đề này, chúng ta sẽ sử dụng phương pháp Learning Similarity. Đây là phương pháp, chúng ta sẽ sử dụng phép đo khoảng cách giữa 2 bức ảnh với nhau trong không gian Euclide N chiều. Nếu 2 bức ảnh cùng là một người thì khoảng cách phải nhỏ hơn ngưỡng cho phép và ngược lại.

$$\begin{cases} d(img1, img2) \leq \tau \rightarrow same \\ d(img1, img2) > \tau \rightarrow difference \end{cases}$$

Để có thể hiểu rõ hơn về phương pháp này, dưới đây là hình minh họa:

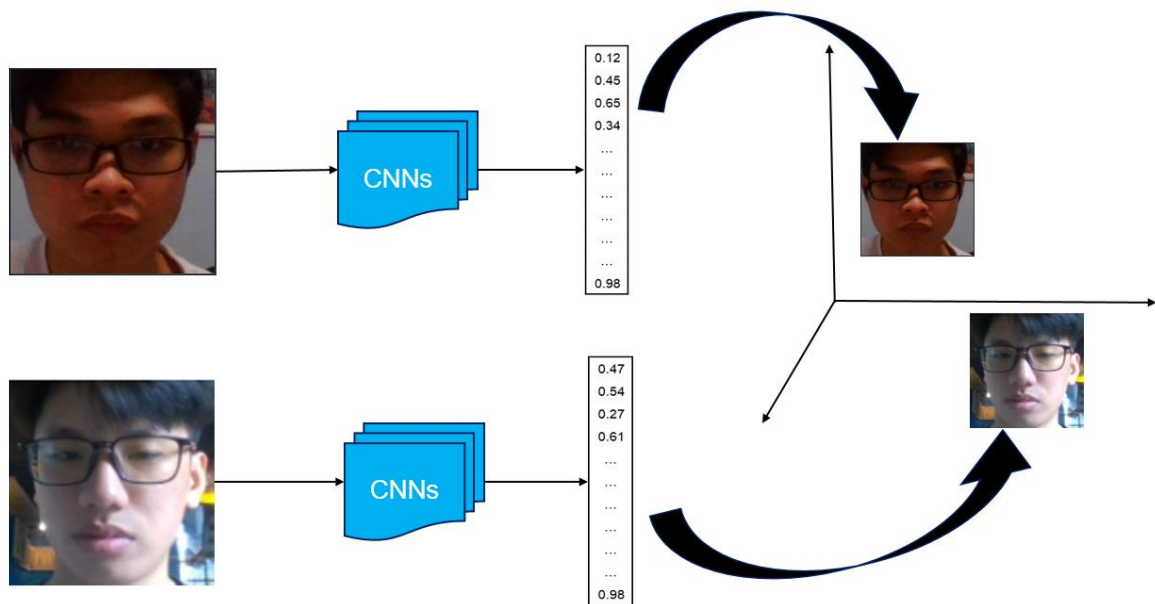


Hình 2.7. Ảnh minh họa thuật toán Learning Similarity

Như vậy, chúng ta có thể thấy được ưu điểm của phương pháp Learning Similarity này không phụ thuộc vào số lượng tập phân lớp. Do đó chúng ta không cần phải huấn luyện lại mô hình mỗi khi có sự thay đổi về mặt nhân sự.

Điểm mấu chốt nằm ở mô hình này là chúng ta cần xây dựng một mô hình CNN dùng để trích xuất vector đặc trưng của khuôn mặt trong bức ảnh và chiếu lên không gian Euclide N chiều. Tức là đầu vào của chúng ta sẽ là một bức ảnh, và đầu

ra của chúng ta sẽ là một vector N đặc trưng và được dùng để chiếu lên không gian Euclid N chiều.



Hình 2.8. Ảnh minh họa mô tả *Learning Similarity*

Sau đó, chúng ta sẽ so sánh khoảng cách giữa 2 bức hình với nhau và sẽ đưa ra quyết định đó là bức ảnh khuôn mặt của người nào giống như hình minh họa phía trên.

Và để xây dựng và huấn luyện được một mô hình trích xuất đặc trưng khuôn mặt và biểu diễn trên không gian Euclidean N chiều thì chúng ta sẽ cùng tìm hiểu về mạng Siamese Network.

2.4 Siamese Network

Siamese Network là tên gọi chung của các mô hình CNN dùng để trích xuất đặc trưng của đối tượng trong bức ảnh và ra quyết định thuộc về lớp nào thông qua tính toán khoảng cách giữa 2 vector đặc trưng của 2 bức ảnh được chiếu lên không gian Euclidean N chiều.

Siamese Network sử dụng các lớp giống như các mạng CNN chúng ta thường bắt gặp. Tuy nhiên, các mạng CNN này chúng ta sẽ loại bỏ lớp phân loại cuối cùng,

và thay vào đó là lớp Embedding cuối cùng phục vụ cho việc trích xuất đặc trưng của bức ảnh.

Sau khi xây dựng và huấn luyện một mô hình Siamese Network thì chúng ta có thể sử dụng nó giống như phương pháp Learning Similarity như trên:

Bước 1: Chúng ta sẽ thu thập các khuôn mặt của nhân viên có trong công ty.

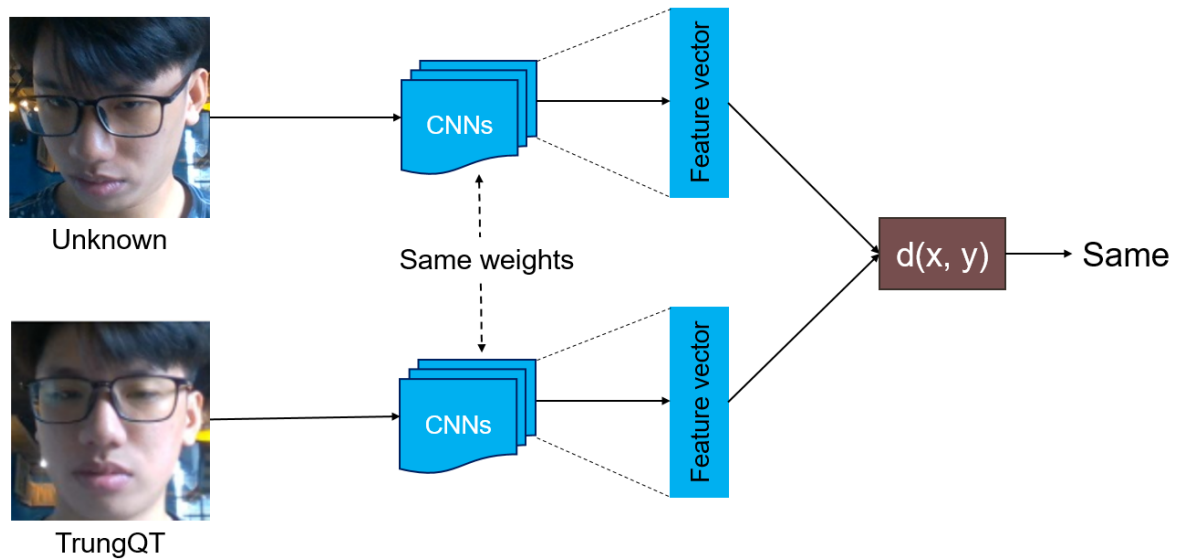
Bước 2: Chúng ta sử dụng mô hình CNN để trích xuất các vector đặc trưng của các khuôn mặt trên và gán nhãn đúng với chủ nhân của nó và lưu vào cơ sở dữ liệu. Ở đây, chúng ta sẽ gọi những vector này là $F(x)$.

Bước 3: Khi có một bức ảnh mới dùng để nhận diện. Chúng ta lại tiếp tục trích xuất vector đặc trưng của khuôn mặt mới ra. Ở đây chúng ta sẽ gọi vector này là $F(y)$.

Bước 4: Chúng ta sẽ tính toán khoảng cách giữa vector $F(y)$ so với từng vector $F(x)$ trong cơ sở dữ liệu.

Bước 5: Điều kiện để ra quyết định.

- Điều kiện 1: Khoảng cách giữa $F(x_i)$ và $F(y)$ phải là nhỏ nhất so với các vector còn lại trong cơ sở dữ liệu
- Điều kiện 2: Khoảng cách giữa $F(x_i)$ và $F(y)$ phải nằm trong ngưỡng cho phép. Nếu ngoài ngưỡng cho phép thì gán nhãn là “Unknown”, tức là khuôn mặt này không có trong cơ sở dữ liệu của nhân viên trong công ty.



Hình 2.9. Ảnh minh họa luồng xử lý tính khoảng cách hai bức ảnh

Mục tiêu chính của Siamese Network đó là tìm ra biểu diễn vector của ảnh trong không gian Euclide N chiều nên không nhất thiết phải lựa chọn hàm mất mát (loss function) là Binary Cross Entropy như các bài toán phân loại. Tuy nhiên, trên thực tế việc lựa chọn hàm mất mát là Binary Cross Entropy vẫn sẽ tìm ra được biểu diễn vector tốt cho ảnh trên không gian N chiều.

Việc lựa chọn hàm mất mát sẽ có tác động rất lớn tới độ chính xác để tìm ra vector đặc trưng cho bức ảnh trên không gian Euclide. Vì vậy, để giải quyết vấn đề này chúng ta sẽ tìm hiểu về hàm mất mát Triplet Loss.

2.5 Hàm mất mát Triplet Loss

Khi chúng ta tạo ra một mạng Siamese Network để trích xuất đặc trưng, thì trong quá trình huấn luyện, các bức ảnh sẽ được trích xuất ra các vector đặc trưng với kích thước bằng nhau, và những vector này sẽ là đầu vào cho hàm mất mát Triplet Loss để đánh giá khoảng cách giữa các vector trong lúc huấn luyện. Có thể nói hàm mất mát Triplet Loss chính là phương pháp để đánh giá khoảng cách giữa các vector trong lúc huấn luyện.

Để dễ hình dung, hàm mất mát Triplet Loss chọn ra trong mô hình 3 vector đã được gán nhãn và chia làm thành 3 phần **Anchor**, **Positive** và **Negative**. Ở đó,

Anchor là ảnh mà mô hình chọn để đánh giá khoảng cách giữa 2 ảnh **Positive** và **Negative**. Ảnh **Positive** là ảnh được cho là giống với ảnh gốc nhất (ảnh của cùng là một người so với ảnh của **Anchor**) và ảnh **Negative** là ảnh được cho là khác nhất so với ảnh gốc (ảnh của một người khác với ảnh của **Anchor**). Kí hiệu lần lượt là **A**, **P** và **N**.



Hình 2.10. Mô tả bài toán Triplet Loss

Khi mô hình chọn ra 3 vector đặc trưng này cũng là lúc 3 vector này nằm rải rác đâu đó trong không gian Euclide bởi vì mô hình vẫn còn đang trong quá trình huấn luyện nên chưa đưa ra được vị trí chính xác cho các vector. Khi đó hàm mất mát sẽ giải quyết được vấn đề này bằng cách dùng ảnh Anchor để so sánh với ảnh **Positive** và **Negative**. Nếu ảnh **Positive** nằm rất xa so với **Anchor** (tức là $d(\mathbf{A}, \mathbf{P})$ rất lớn) thì hàm mất mát sẽ kéo ảnh **Positive** về lại gần **Anchor** và nếu ảnh **Negative** nằm rất gần so với ảnh **Anchor** (tức là $d(\mathbf{A}, \mathbf{N})$ rất nhỏ) thì hàm mất mát sẽ đẩy ảnh **Negative** ra xa và sau đó cập nhật lại bộ trọng số cho mô hình.



Hình 2.11. Ảnh minh họa cho thuật toán Triplet Loss

Hàm mất mát Triplet Loss luôn lấy 3 bức ảnh làm bộ đầu vào, và trong mọi trường hợp khi đánh giá khoảng cách giữa bộ 3 bức ảnh trên thì chúng ta đều kì vọng rằng:

$$d(\mathbf{A}, \mathbf{P}) < d(\mathbf{A}, \mathbf{N})$$

Để làm cho khoảng cách giữa về trái và về phải lớn hơn, chúng ta sẽ cộng thêm vào về trái một hệ số α không âm rất nhỏ, khi đó (1) sẽ thành:

$$d(\mathbf{A}, \mathbf{P}) + \alpha \leq d(\mathbf{A}, \mathbf{N})$$

$$\rightarrow \|f(\mathbf{A}) - f(\mathbf{P})\|_2^2 + \alpha \leq \|f(\mathbf{A}) - f(\mathbf{N})\|_2^2$$

$$\rightarrow \|f(\mathbf{A}) - f(\mathbf{P})\|_2^2 - \|f(\mathbf{A}) - f(\mathbf{N})\|_2^2 + \alpha \leq 0$$

Như vậy hàm mất mát chúng ta nhận được sẽ là:

$$L(\mathbf{A}, \mathbf{P}, \mathbf{N}) = \sum_{i=0}^n \|f(\mathbf{A}_i) - f(\mathbf{P}_i)\|_2^2 - \|f(\mathbf{A}_i) - f(\mathbf{N}_i)\|_2^2 + \alpha$$

Trong đó \mathbf{A} là ảnh gốc, \mathbf{P} là mẫu giống ảnh gốc, \mathbf{N} là mẫu khác ảnh gốc và α là số lượng các bộ 3 được đưa vào hàm mất mát để huấn luyện.

Sẽ không ảnh hưởng gì nếu mô hình nhận diện đúng ảnh **Negative** và **Positive** là cùng cặp hay khác cặp với ảnh **Anchor**. Mục tiêu của chúng ta là làm giảm thiểu các trường hợp mà mô hình nhận diện sai ảnh **Negative** thành **Positive** nhất có thể. Do đó, để loại bỏ ảnh hưởng của các trường hợp nhận diện đúng **Negative** và **Positive** lên hàm mất mát, ta sẽ điều chỉnh giá trị biểu thức của hàm mất mát về 0.

Tức là nếu:

$$\|f(\mathbf{A}) - f(\mathbf{P})\|_2^2 - \|f(\mathbf{A}) - f(\mathbf{N})\|_2^2 + \alpha \leq 0$$

Được điều chỉnh về 0. Khi đó hàm mất mát sẽ trở thành:

$$L(\mathbf{A}, \mathbf{P}, \mathbf{N}) = \sum_{i=0}^n \max(\|f(\mathbf{A}_i) - f(\mathbf{P}_i)\|_2^2 - \|f(\mathbf{A}_i) - f(\mathbf{N}_i)\|_2^2 + \alpha, 0)$$

Như vậy, khi áp dụng hàm mất mát trên vào các mô hình CNN trích xuất đặc trưng, thì chúng ta có thể yên tâm rằng có thể tạo ra các biểu diễn vector tốt nhất cho mỗi bức ảnh lên không gian Euclide. Điều này có nghĩa, các bức ảnh thuộc cùng một lớp sẽ nằm rất gần nhau và những ảnh không thuộc cùng một lớp sẽ nằm rất xa khi được biểu diễn trong không gian Euclide.

Một chú ý quan trọng khi huấn luyện mô hình Siamese Network sử dụng hàm mất mát Triplet Loss đó là chúng ta cần phải đảm bảo cặp ảnh **A**, **P** là thuộc về cùng một chủ nhân và ảnh **N** là được chọn ngẫu nhiên trong các nhãn còn lại. Vì thế, điều kiện tiên quyết đối với bộ dữ liệu được dùng để huấn luyện là cần phải có ít nhất 2 bức ảnh đối với 1 người.

2.6 Cách chọn bộ ba ảnh của Triplet Loss

Trong quá trình huấn luyện mô hình Siamese Network sử dụng hàm mất mát Triplet Loss, chúng ta cần quan tâm đến cách chọn bộ 3 **Anchor**, **Positive** và **Negative** sao cho phù hợp.

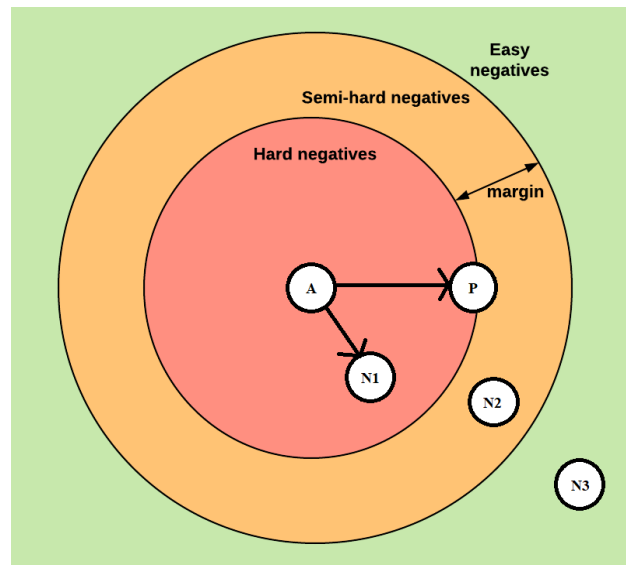
Dựa trên mô tả về hàm mất mát Triplet Loss ở phần trên, Triplet Loss còn được chia thành 3 cách chọn ảnh **Negative**:

Easy Triplets: $d(\mathbf{A}, \mathbf{P}) + \alpha \leq d(\mathbf{A}, \mathbf{N})$, tức là chọn ảnh **Negative** có khoảng cách đến **Anchor** luôn lớn hơn hoặc bằng so với khoảng cách từ **Anchor** đến **Positive**.

Hard Triplets: $d(a, n) < d(a, p)$, tức là chọn ảnh **Negative** có khoảng cách đến **Anchor** luôn nhỏ hơn khoảng cách từ **Anchor** đến **Positive**. Điều này sẽ khiến cho mạng Neural cần phải học nhiều hơn để đưa ra quyết định tốt hơn.

Semi-Hard Triplets: $d(a, p) < d(a, n) < d(a, p) + \alpha$, tức là chọn ảnh **Negative** có khoảng cách đến **Anchor** nằm trong khoảng từ **Anchor** đến **Positive** và từ **Anchor** đến **Positive** cộng thêm một khoảng α

Dưới đây là hình minh họa cho các cách chọn ảnh **Negative**:



Hình 2.12. Ảnh minh họa cách chọn ảnh Negative

Việc chọn bộ ba ảnh **A**, **P** và **N** sẽ ảnh hưởng rất nhiều trong quá trình huấn luyện ra một mô hình trích xuất vector đặc trưng tốt và biểu diễn nó trong không gian Euclide.

2.7 Công nghệ phát hiện khuôn mặt với MediaPipe

2.7.1 Tổng quát

MediaPipe Face Detection [6] là giải pháp nhận diện khuôn mặt tốc độ nhanh đi kèm với 6 điểm mốc và hỗ trợ đa khuôn mặt. MediaPipe Face Detection dựa trên BlazeFace (một công cụ dò tìm khuôn mặt nhẹ và hoạt động tốt được thiết kế riêng cho sự tính toán GPU trên di động), hiệu suất siêu thời gian thực của của công cụ dò tìm khuôn mặt cho phép nó được áp dụng vào bất kỳ thực nghiệm nào yêu cầu chính xác vùng khuôn mặt để làm đầu vào cho các mô hình khác sử dụng.

2.7.2 Đầu ra của MediaPipe

Là một tập hợp của các khuôn mặt đã phát hiện, mỗi khuôn mặt được biểu diễn dưới dạng một thông báo hiển thị, thông báo này chứa một đường viền hình hộp với 6 điểm (mắt phải, mắt trái, mũi, miệng, tai trái, tai phải).

Đường viền hình hộp gồm x_{\min} và width (cả hai đều được chuẩn hóa [0.0, 1.0] theo chiều rộng hình ảnh) và y_{\min} và height (cả hai đều được chuẩn hóa [0.0, 1.0] theo chiều cao hình ảnh). Mỗi điểm chính bao gồm x và y được chuẩn hóa [0.0, 1.0] theo chiều rộng và cao của hình ảnh (trích dẫn từ [6]).

Ở phần đầu ra này, chúng ta có thể tùy chỉnh lại để phù hợp với yêu cầu của các ứng dụng khác nhau. Đối với giao diện của ứng dụng trên desktop, chúng tôi đã lược bỏ các điểm chấm trên khuôn mặt người và mở rộng lấy cả phần tóc của khuôn mặt dùng để huấn luyện cho mô hình định danh mặt người.

2.7.3 Hướng dẫn cài đặt trên python

Chúng ta có thể kích hoạt môi trường ảo Python bằng công cụ Anaconda hoặc thực hiện lệnh sau:

```
$ python -m venv mp_env && source mp_env/bin/activate
```

Cài đặt gói MediaPipe Python sử dụng pip trong python

```
(mp_env)$ pip install MediaPipe
```

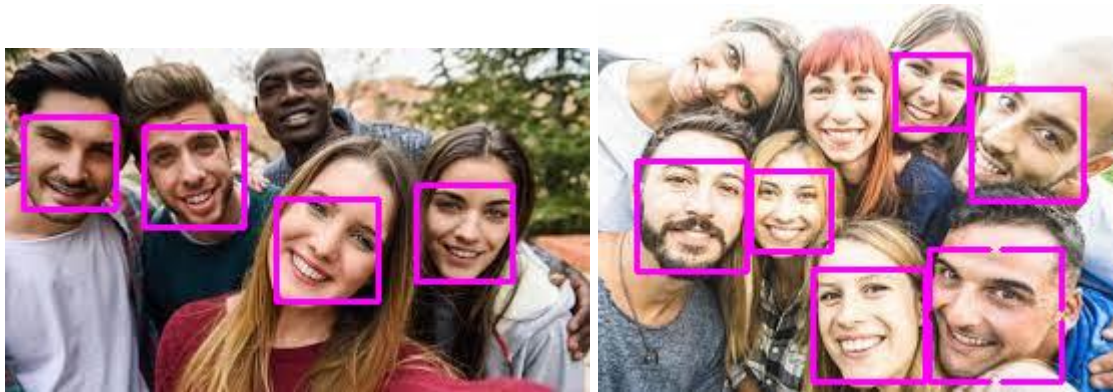
Để gọi được thư viện MediaPipe trong python, chúng ta nhúng mã nguồn

```
import MediaPipe as mp  
  
mp_face_mesh = mp.solutions.face_mesh
```

Mẹo: Sử dụng lệnh deactivate để thoát khỏi môi trường ảo Python.

2.7.4 Kết quả nhận diện

Kết quả thử nghiệm dựa trên hai tấm ảnh bất kỳ có chứa khuôn mặt người, kết quả cho thấy MediaPipe có thể phát hiện nhiều khuôn mặt nhưng vẫn còn hạn chế.



Hình 2.13. Kết quả mô hình phát hiện khuôn mặt MediaPipe

2.7.5 Nhận xét

Ta có thể thấy tốc độ phát hiện khuôn mặt của MediaPipe rất nhanh và tiêu thụ ít tài nguyên máy tính nên phù hợp với các ứng dụng sử dụng thời gian thực (real time).

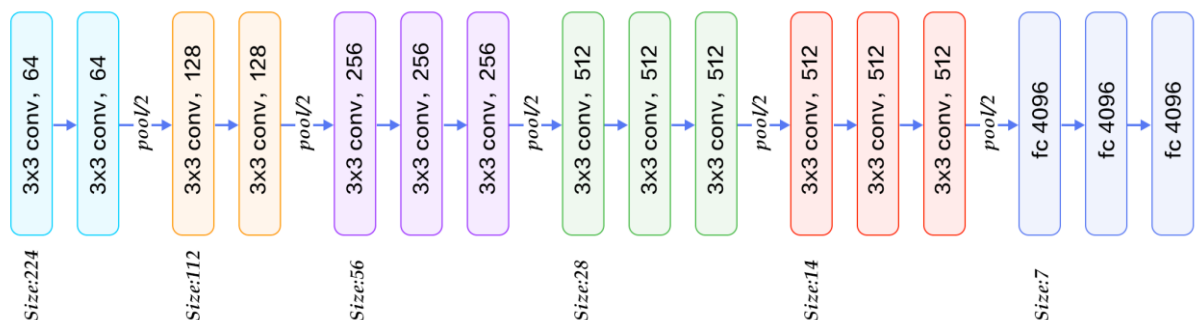
Khi áp dụng vào ứng dụng để sử dụng, thời gian xử lý trung bình của MediaPipe là một phần năm giây (tức là 0.2 giây) trên một khung hình (frame) với dung lượng RAM tiêu thụ là 0 và yêu cầu xử lý CPU trung bình là 10%.

Nhưng mô hình vẫn còn mặt hạn chế về khoảng cách khi quét trong phạm vi camera xa, MediaPipe không thể phát hiện được khuôn mặt

CHƯƠNG III. XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH

3.1 Xây dựng mạng Siamese Network với kiến trúc VGG16.

VGG16 là một mạng CNN được đề xuất bởi K. Simonyan và A. Zisserman của trường đại học Oxford. Model sử dụng mạng VGG16 sau khi train trong bộ dữ liệu ImageNet gồm 14 triệu ảnh và 1000 lớp khác nhau và cho ra độ chính xác là 92,7%.



Hình 3.1. Kiến trúc mạng VGG16

Kiến trúc của mạng VGG16 bao gồm 21 lớp: 13 lớp Convolution với kernel 3x3, 5 lớp Pooling với kernel 2x2 và 3 lớp Fully Connection.

Sau khi có được kiến trúc của mạng VGG16, tôi có tinh chỉnh đôi chút đó là không lấy những lớp Fully Connection và chỉ lấy 15 lớp đầu tiên tính từ lớp Convolution đầu tiên và sau đó thêm 1 lớp Pooling ngay sau lớp Convolution cuối cùng và cuối cùng là những lớp dùng để trích xuất đặc trưng của ảnh.

Hình dưới đây là mô tả về kiến trúc mạng mà tôi dùng để huấn luyện:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 180, 180, 3)]	0
block1_conv1 (Conv2D)	(None, 180, 180, 64)	1792
block1_conv2 (Conv2D)	(None, 180, 180, 64)	36928
block1_pool (MaxPooling2D)	(None, 90, 90, 64)	0
block2_conv1 (Conv2D)	(None, 90, 90, 128)	73856
block2_conv2 (Conv2D)	(None, 90, 90, 128)	147584
block2_pool (MaxPooling2D)	(None, 45, 45, 128)	0
block3_conv1 (Conv2D)	(None, 45, 45, 256)	295168
block3_conv2 (Conv2D)	(None, 45, 45, 256)	590080
block3_conv3 (Conv2D)	(None, 45, 45, 256)	590080
block3_pool (MaxPooling2D)	(None, 22, 22, 256)	0
block4_conv1 (Conv2D)	(None, 22, 22, 512)	1180160
block4_conv2 (Conv2D)	(None, 22, 22, 512)	2359808
block4_conv3 (Conv2D)	(None, 22, 22, 512)	2359808
block4_pool (MaxPooling2D)	(None, 11, 11, 512)	0
block5_conv1 (Conv2D)	(None, 11, 11, 512)	2359808
max_pooling2d (MaxPooling2D)	(None, 5, 5, 512)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 128)	1638528
lambda (Lambda)	(None, 128)	0
Total params: 11,633,600		
Trainable params: 11,633,600		
Non-trainable params: 0		

Hình 3.2. Kiến trúc mạng được pre-trained dựa trên VGG16

3.2 Bộ dữ liệu dùng để huấn luyện cho đề tài

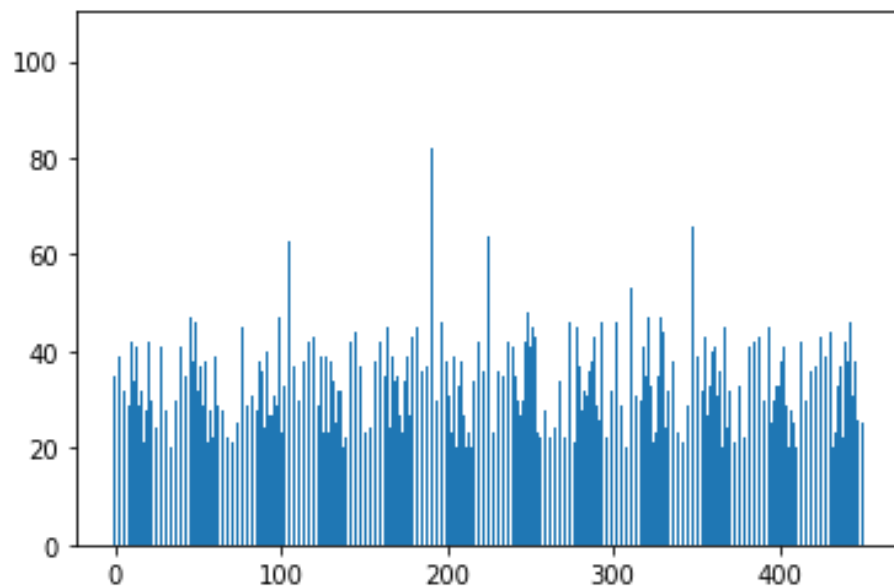
Bộ dataset được sử dụng là VN-Celeb. Với hơn 23000 bức ảnh của hơn 1000 người nổi tiếng tại Việt Nam.

Tuy nhiên vì giới hạn Ram và Vram của Colab nên không thể sử dụng hết số lượng ảnh trên để training. Vì vậy, chỉ có thể lấy được ảnh của 451 người bất kỳ với số lượng khuôn mặt một người là lớn hơn 20 tấm. Và tổng cộng là 15490 tấm ảnh cho 451 người.



Hình 3.3. Bộ dữ liệu khuôn mặt VN-Celeb

Để cho dễ hình dung, đây là biểu đồ tần suất số lượng hình ảnh của từng người trong 451 người được sử dụng để phục vụ mục đích training. Nhìn vào biểu đồ, ta có thể dễ dàng nhận ra được là số lượng hình ảnh thấp nhất của một người là 20 tấm ảnh.



Hình 3.4. Biểu đồ hiển thị số khuôn mặt của mỗi người

3.3 Giới thiệu về mạng pretrained FaceNet

FaceNet [3] lần đầu tiên được giới thiệu vào ngày 17 tháng 6 năm 2015, đây là một sản phẩm của Google. FaceNet sử dụng một kiến trúc khá phức tạp và cũng được biết đến như là mạng Siamese Network. Bởi vì, chính FaceNet cũng chính là một mạng CNN dùng để trích xuất vector đặc trưng của những bức ảnh và sử dụng hàm mất mát Triplet Loss.

FaceNet được huấn luyện dựa trên bộ dữ liệu cực kỳ lớn, từ 100 – 200 triệu tấm hình của 8 triệu người khác nhau.

Dưới đây là bản thảo thiết kế kiến trúc của mạng FaceNet trong năm 2015.

layer	size-in	size-out	kernel	param	FLPS
conv1	$220 \times 220 \times 3$	$110 \times 110 \times 64$	$7 \times 7 \times 3, 2$	9K	115M
pool1	$110 \times 110 \times 64$	$55 \times 55 \times 64$	$3 \times 3 \times 64, 2$	0	
rnorm1	$55 \times 55 \times 64$	$55 \times 55 \times 64$		0	
conv2a	$55 \times 55 \times 64$	$55 \times 55 \times 64$	$1 \times 1 \times 64, 1$	4K	13M
conv2	$55 \times 55 \times 64$	$55 \times 55 \times 192$	$3 \times 3 \times 64, 1$	111K	335M
rnorm2	$55 \times 55 \times 192$	$55 \times 55 \times 192$		0	
pool2	$55 \times 55 \times 192$	$28 \times 28 \times 192$	$3 \times 3 \times 192, 2$	0	
conv3a	$28 \times 28 \times 192$	$28 \times 28 \times 192$	$1 \times 1 \times 192, 1$	37K	29M
conv3	$28 \times 28 \times 192$	$28 \times 28 \times 384$	$3 \times 3 \times 192, 1$	664K	521M
pool3	$28 \times 28 \times 384$	$14 \times 14 \times 384$	$3 \times 3 \times 384, 2$	0	
conv4a	$14 \times 14 \times 384$	$14 \times 14 \times 384$	$1 \times 1 \times 384, 1$	148K	29M
conv4	$14 \times 14 \times 384$	$14 \times 14 \times 256$	$3 \times 3 \times 384, 1$	885K	173M
conv5a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256, 1$	66K	13M
conv5	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256, 1$	590K	116M
conv6a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256, 1$	66K	13M
conv6	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256, 1$	590K	116M
pool4	$14 \times 14 \times 256$	$7 \times 7 \times 256$	$3 \times 3 \times 256, 2$	0	
concat	$7 \times 7 \times 256$	$7 \times 7 \times 256$		0	
fc1	$7 \times 7 \times 256$	$1 \times 32 \times 128$	maxout p=2	103M	103M
fc2	$1 \times 32 \times 128$	$1 \times 32 \times 128$	maxout p=2	34M	34M
fc7128	$1 \times 32 \times 128$	$1 \times 1 \times 128$		524K	0.5M
L2	$1 \times 1 \times 128$	$1 \times 1 \times 128$		0	
total				140M	1.6B

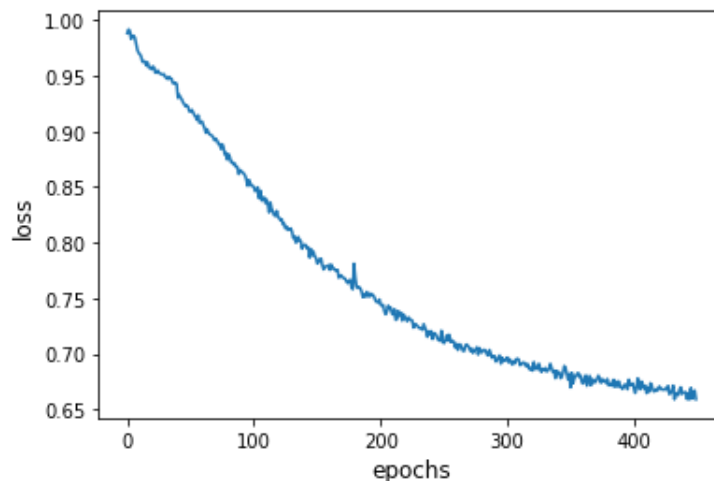
Hình 3.5. Bảng thiết kế kiến trúc của mạng FaceNet năm 2015

Sau quá trình huấn luyện, FaceNet được kiểm thử trên bộ dữ liệu khuôn mặt khác có tên là **Labeled Face in the Wild** thì mô hình FaceNet có độ chính xác lên đến $99,63\% \pm 0,15$.

3.4 Đánh giá mô hình tự train sử dụng kiến trúc mạng VGG16.

3.4.1 Kết quả training sau 450 epochs.

Biểu đồ Loss sau khi training 450 epochs trong thời gian là hơn 20 tiếng đồng hồ sử dụng Colab.



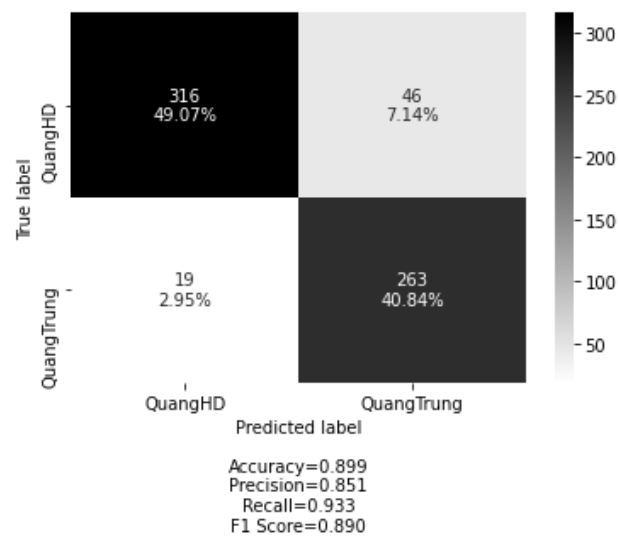
Hình 3.6. Biểu đồ Loss sau khi huấn luyện 450 epochs

3.4.2 Đánh giá độ chính xác của mô hình VGG16

Phương pháp kiểm tra bằng cách tính khoảng cách giữa các vector đặc trưng với nhau:

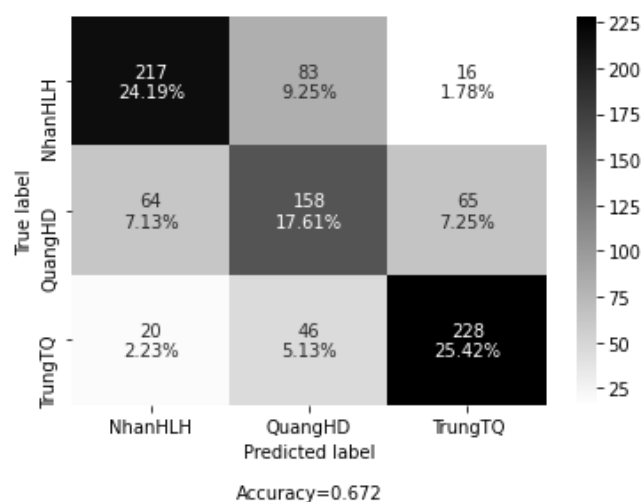
- Cách kiểm tra là chúng ta sẽ sử dụng một bộ dataset mới, với mỗi người khoảng 300 tấm và gán labels tương ứng.
- Trong đó, chúng ta sẽ lấy ra khoảng 10 – 30 tấm để trích xuất đặc trưng bằng mạng Neural vừa được train cho các khuôn mặt đó. Ở đây được gọi là tập train.
- Sau đó, chúng ta sẽ trích xuất đặc trưng số lượng ảnh còn lại và sẽ đem đi so sánh với các bức ảnh đã được gán labels bằng cách so sánh độ tương tự bằng cách tính khoảng cách giữa các đặc trưng với nhau. Tập này được gọi là tập test.
- Cách so sánh:

- Với từng bức ảnh, chúng ta sẽ so sánh toàn bộ với các bức ảnh đã được train. Và nếu khoảng cách của bức ảnh đó gần với bức ảnh nào nhất thì chúng ta sẽ gán labels cho nó tương ứng với bức cần so sánh.
- Sau đó chúng ta sẽ sử dụng Confusion Matrix để đo độ chính xác giữa kết quả dự đoán và kết quả gốc.
- Với bộ dataset có 2 người thì độ chính xác cho ra là: 89,9%



Hình 3.7. Biểu đồ thể hiện độ chính xác mô hình dự đoán hai người

- Với bộ dataset có 3 người thì độ chính xác cho ra là: 67,2%

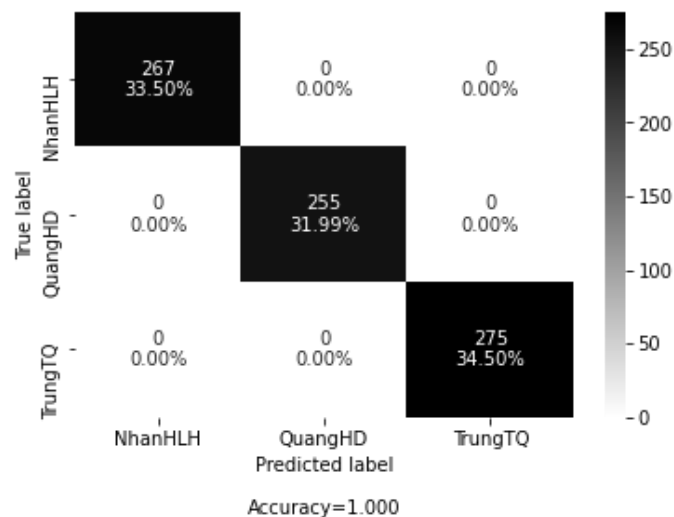


Hình 3.8. Biểu đồ thể hiện độ chính xác mô hình dự đoán ba người

Ở mô hình tự train sử dụng kiến trúc của mạng VGG16 cho ra độ chính xác trong khoảng 65% – 75% tùy vào điều kiện ánh sáng, số lượng khuôn mặt được học và số lượng người cần học. Và đây cũng là một kết quả được cho là tạm chấp nhận được bởi vì số lượng người huấn luyện vẫn còn rất ít so với những mạng lớn như FaceNet và kiến trúc của mạng cũng chỉ ở mức đơn giản mà thôi.

3.5 Đánh giá mô hình pretrain FaceNet

Chúng ta kiểm thử trên tập dữ liệu 3 người với cùng phương pháp tính độ chính xác trên



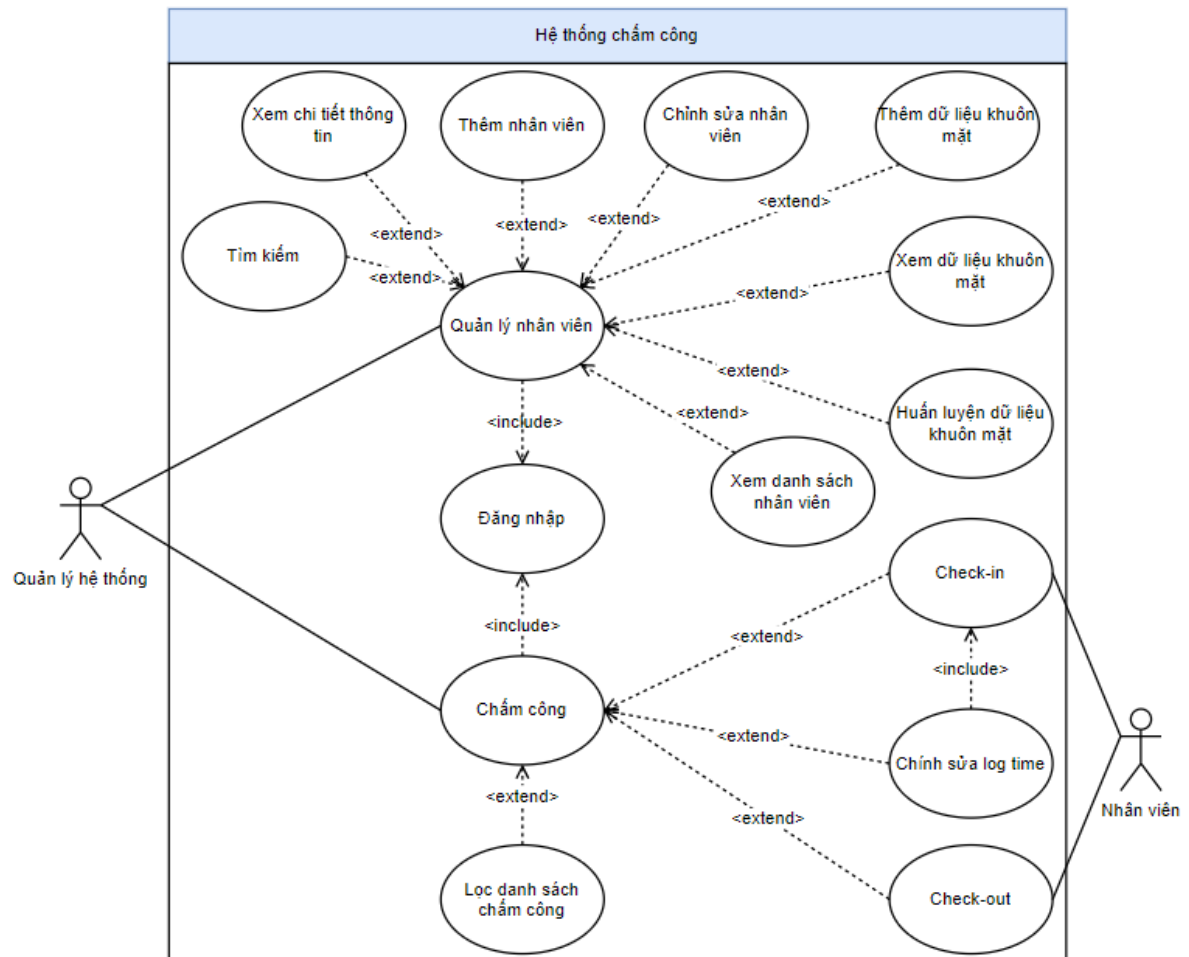
Hình 3.9. Biểu đồ hiển thị độ chính xác FaceNet dự đoán ba người

Ta có thể nhận xét được, ở mô hình pretrain FaceNet đã xây dựng một mô hình trích xuất đặc trưng khuôn mặt với độ chính xác rất cao và đây là một mô hình rất đáng tin cậy để áp dụng vào hệ thống nhận diện khuôn mặt thật nếu không muốn đầu tư vào mô hình quá nhiều.

CHƯƠNG IV. PHÂN TÍCH VÀ XÂY DỰNG ỨNG DỤNG

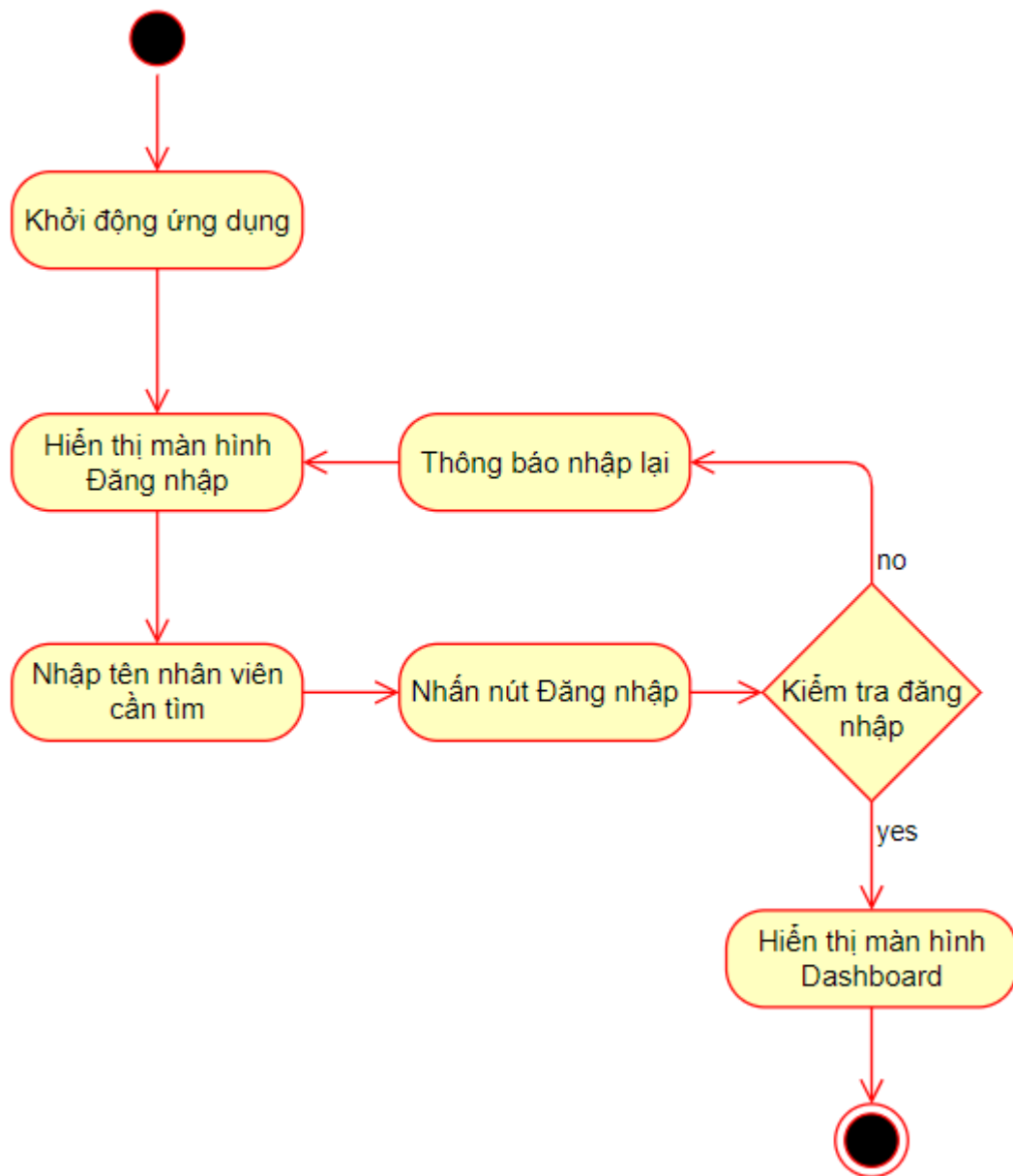
4.1 Mô tả User Case ứng dụng

4.1.1 Mô hình Use case



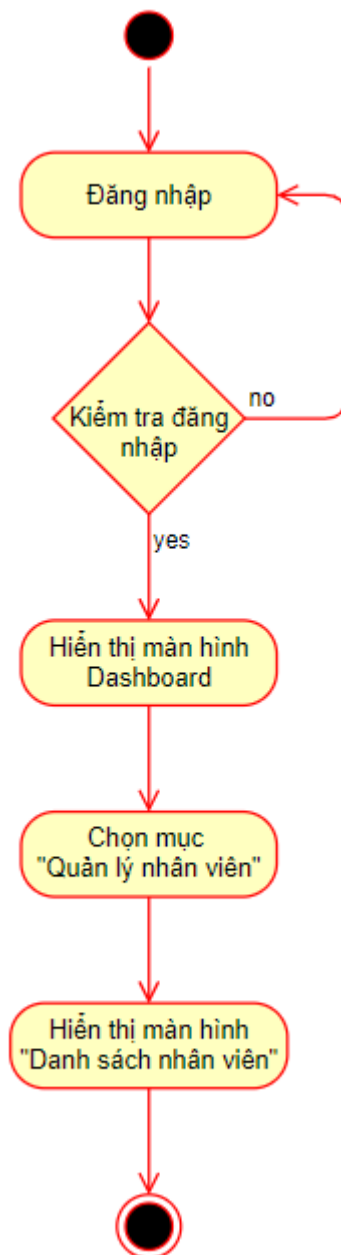
Hình 4.1. Biểu đồ usecase ứng dụng chấm công bằng nhận dạng khuôn mặt

Đăng nhập



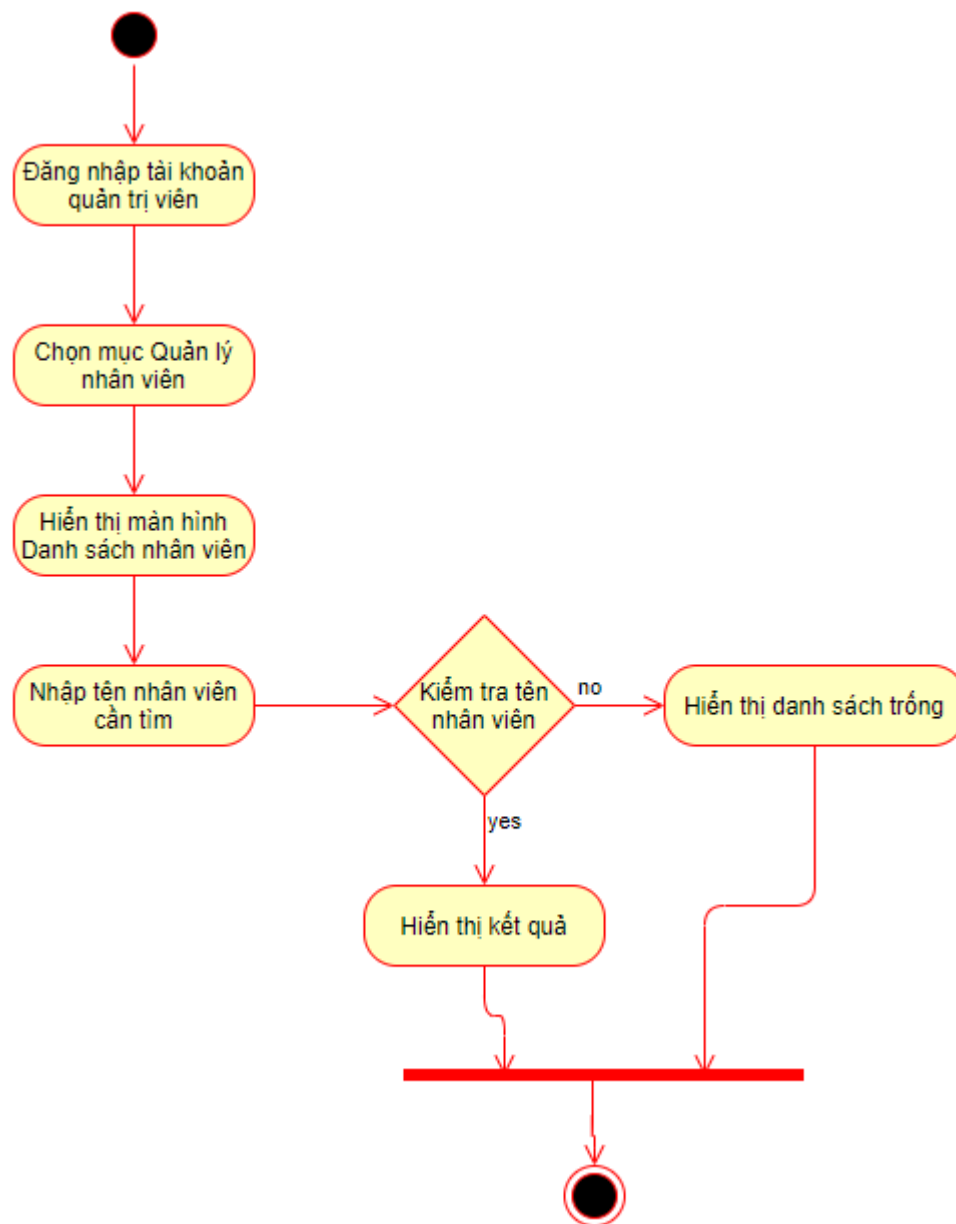
Hình 4.2. Biểu đồ activity đăng nhập

Danh sách nhân viên



Hình 4.3. Biểu đồ activity danh sách nhân viên

Tìm kiếm nhân viên



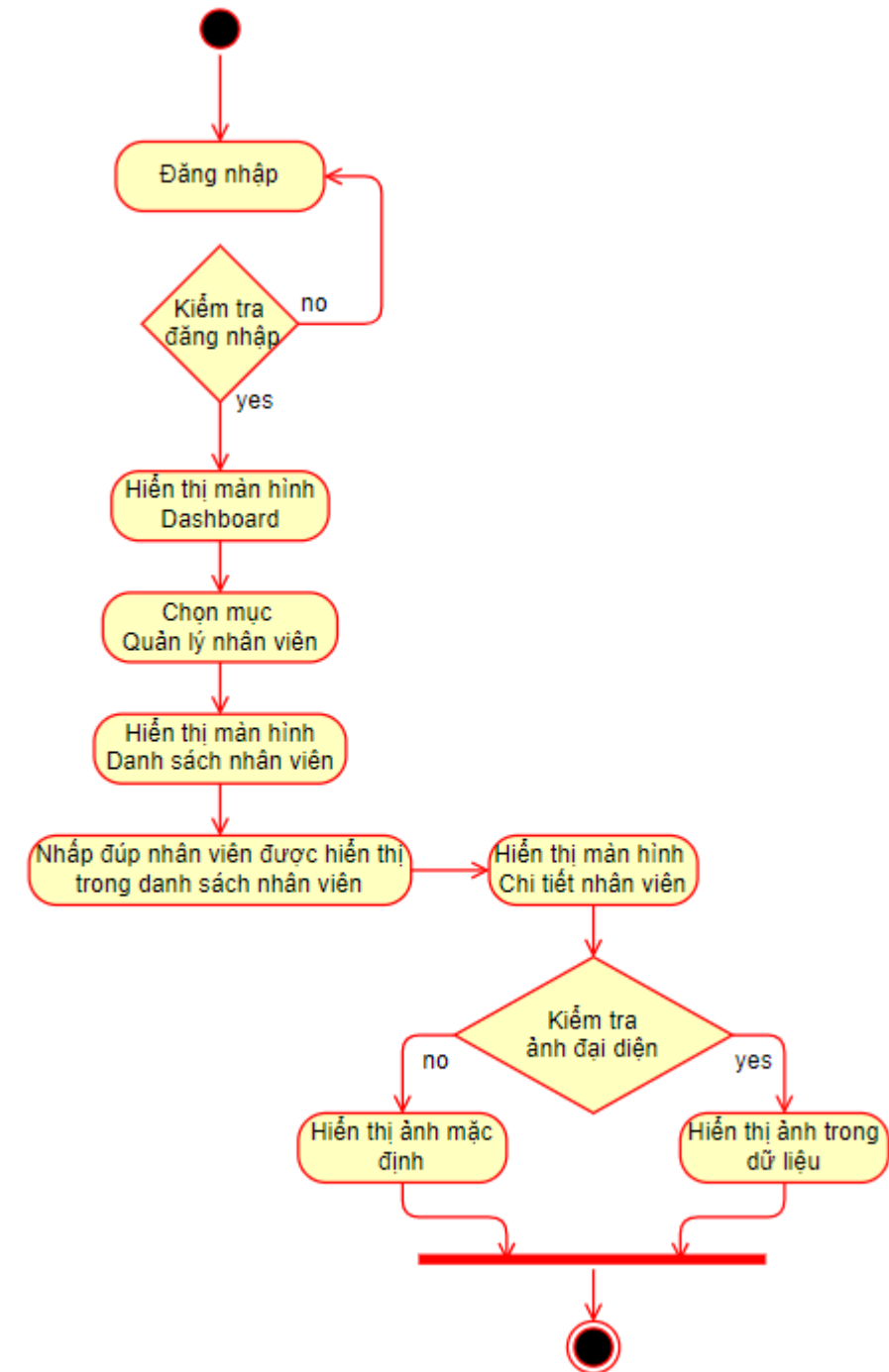
Hình 4.4. Biểu đồ activity tìm kiếm nhân viên

Thêm nhân viên



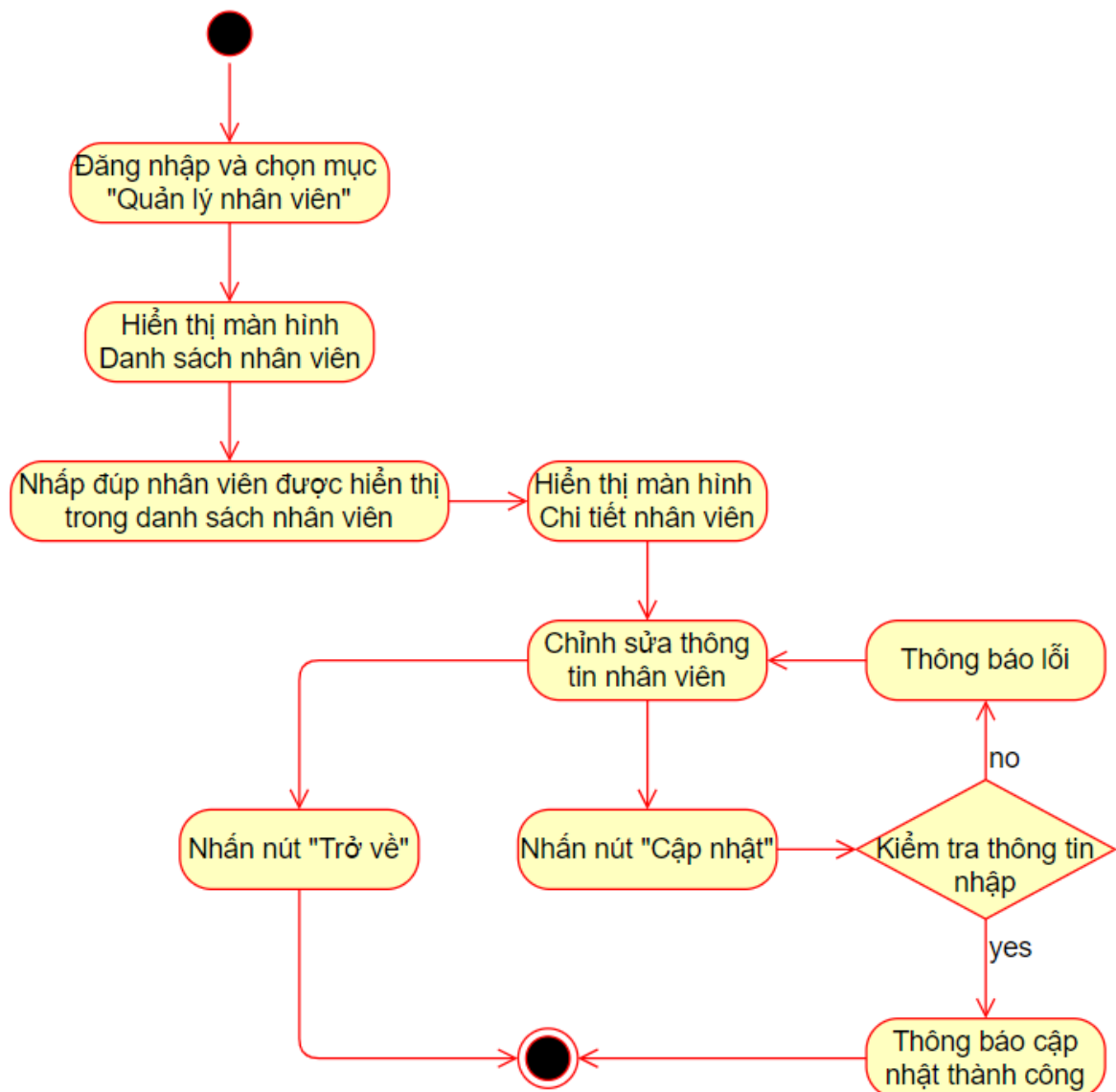
Hình 4.5. Biểu đồ activity thêm nhân viên

Chi tiết nhân viên



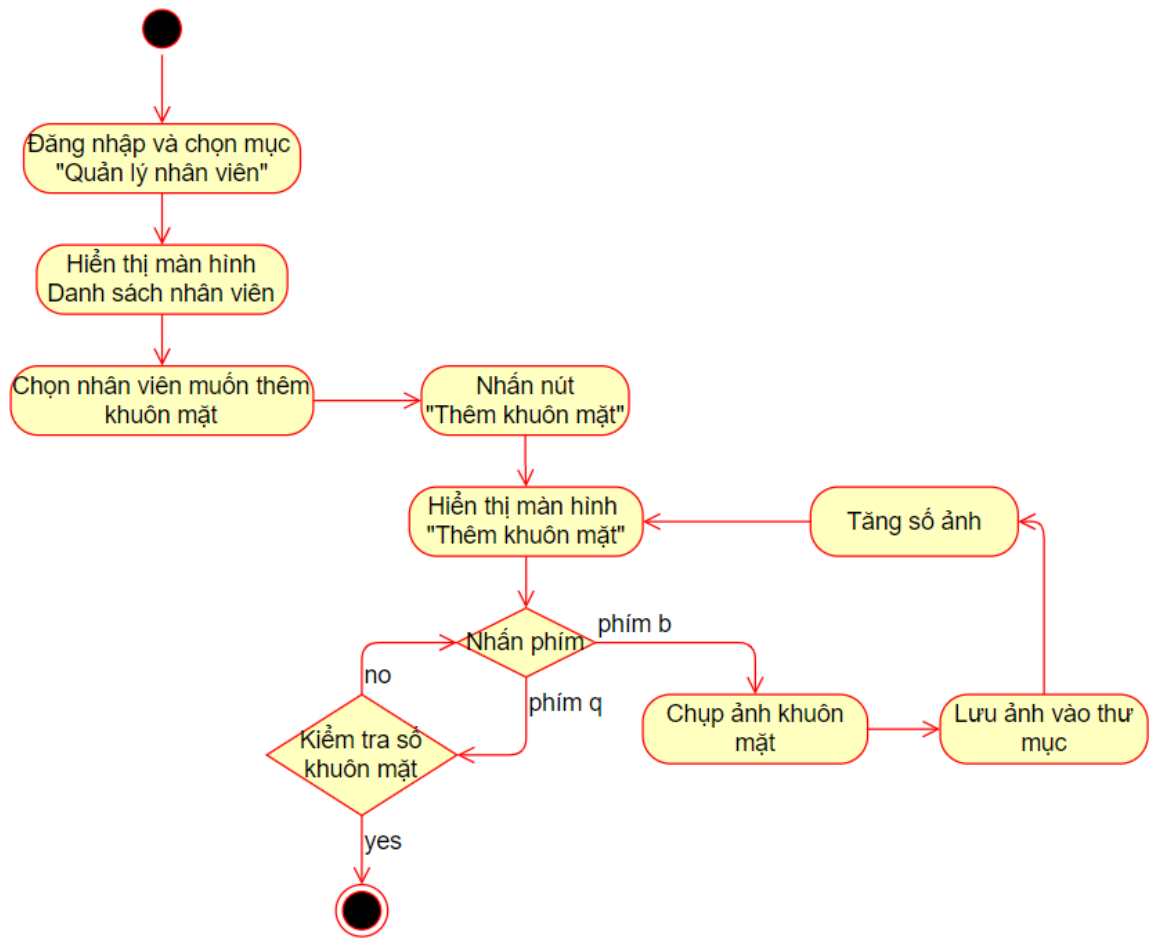
Hình 4.6. Biểu đồ activity chi tiết nhân viên

Cập nhật nhân viên



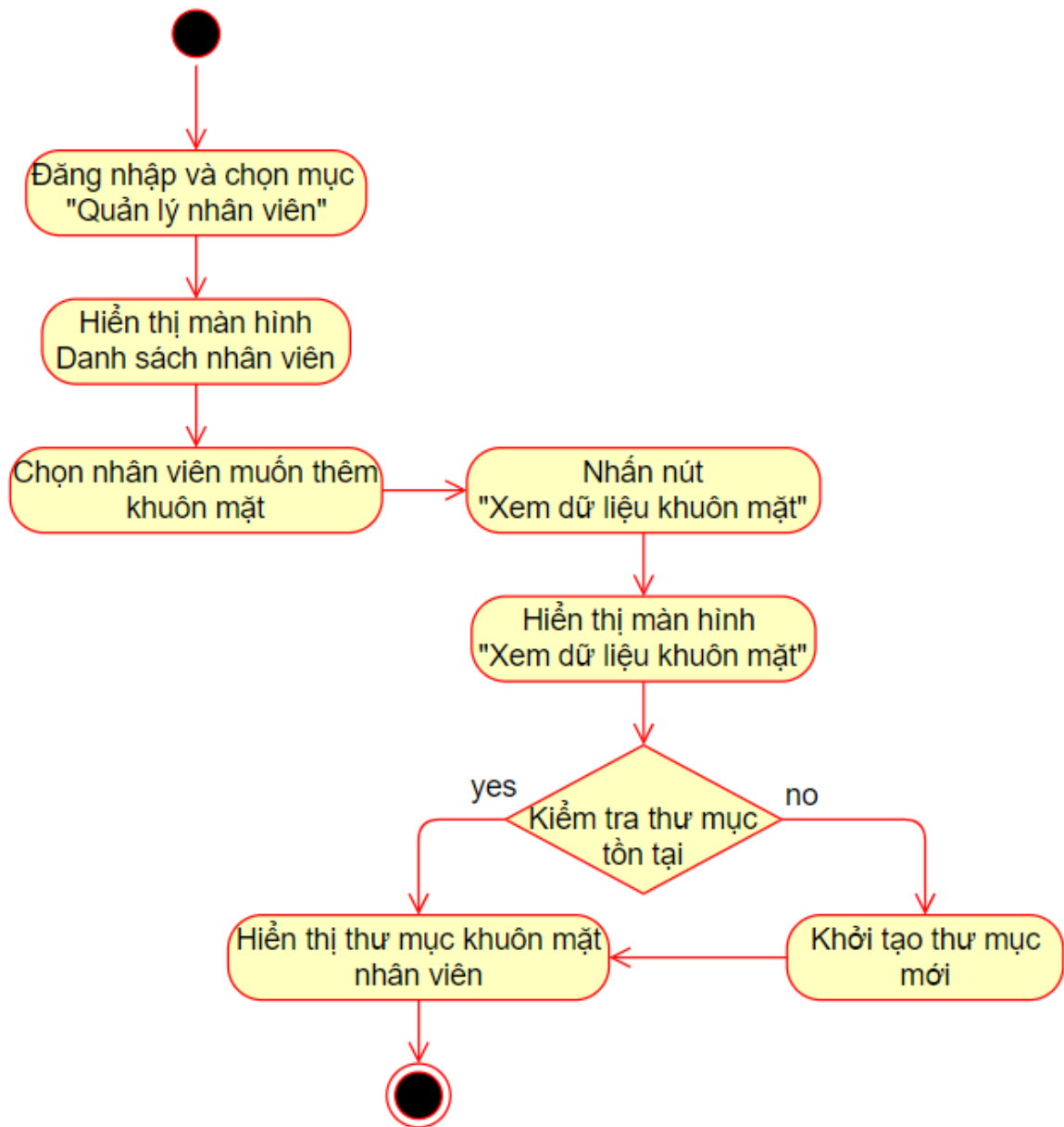
Hình 4.7. Biểu đồ activity cập nhật nhân viên

Thêm khuôn mặt nhân viên



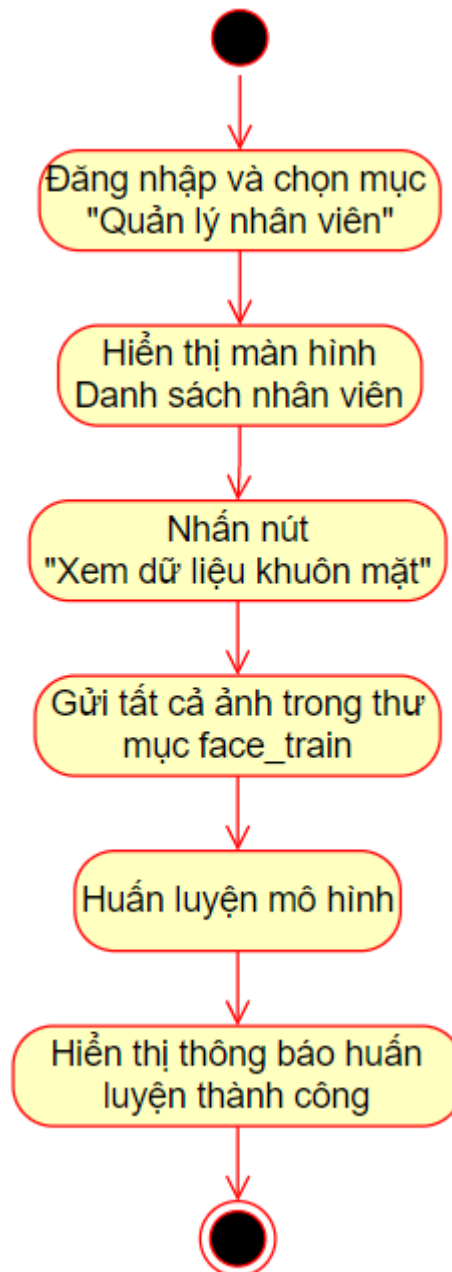
Hình 4.8. Biểu đồ activity thêm khuôn mặt nhân viên

Xem dữ liệu khuôn mặt



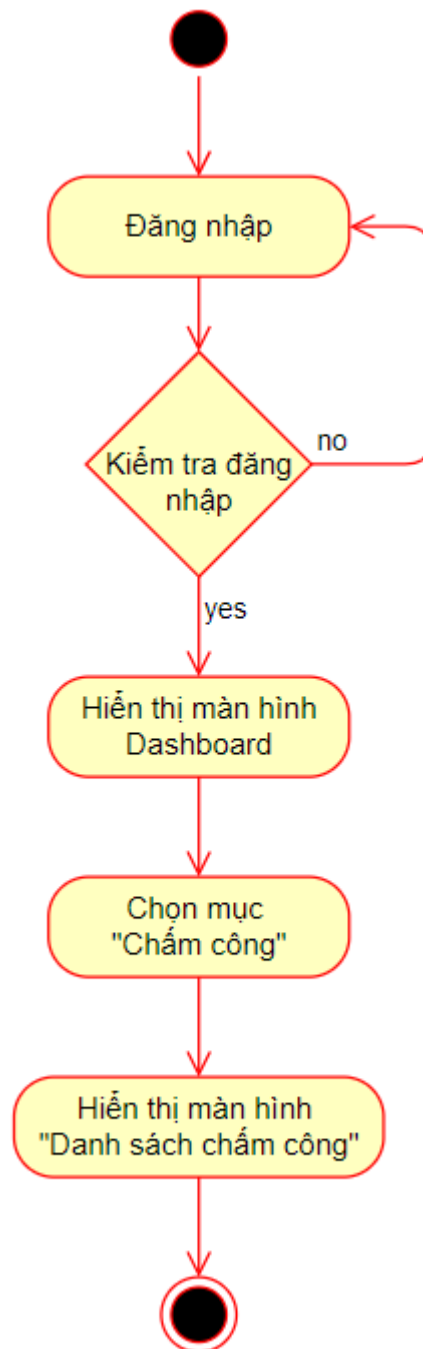
Hình 4.9. Biểu đồ activity xem dữ liệu khuôn mặt

Huấn luyện dữ liệu khuôn mặt



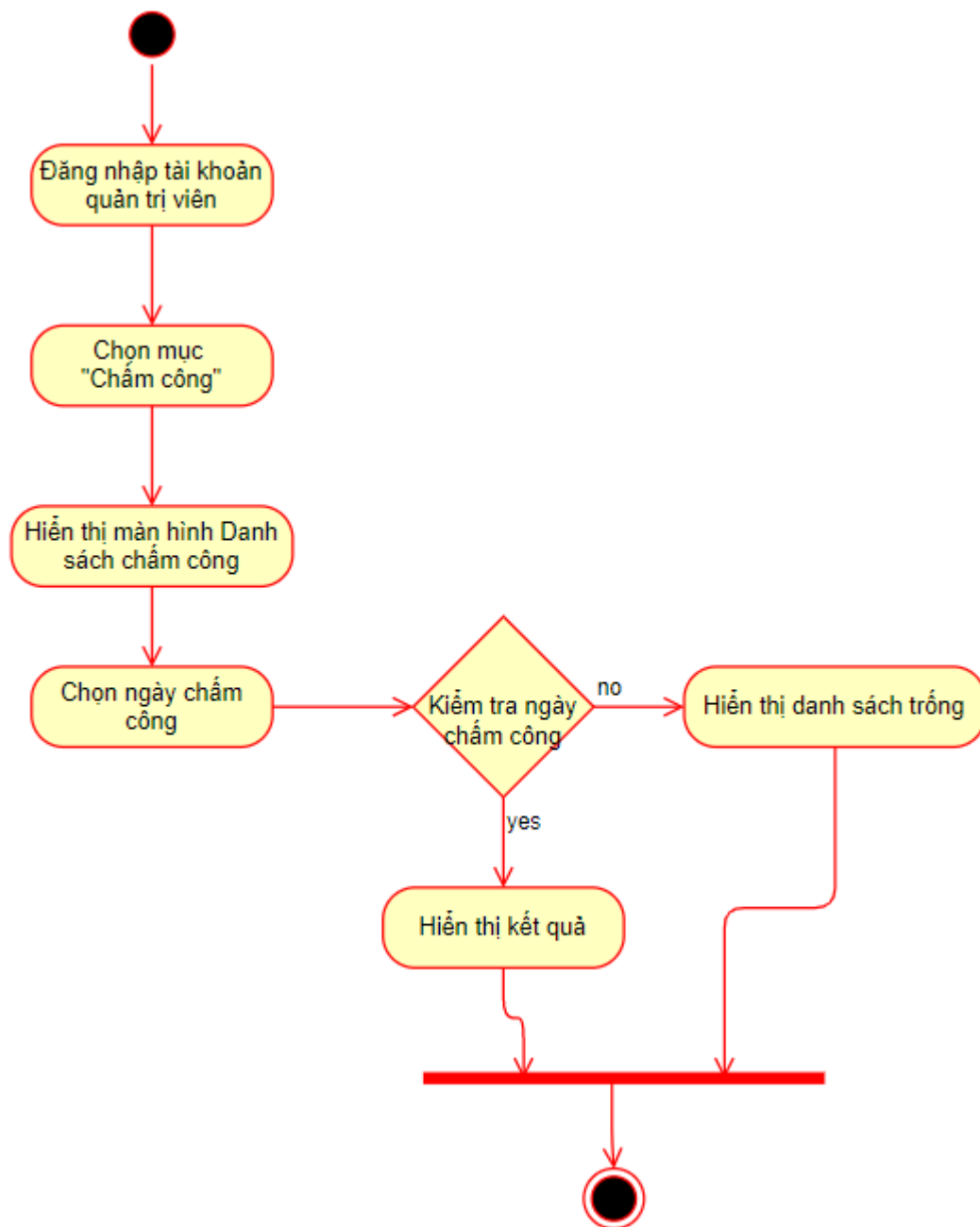
Hình 4.10. Biểu đồ activity huấn luyện khuôn mặt

Danh sách chấm công



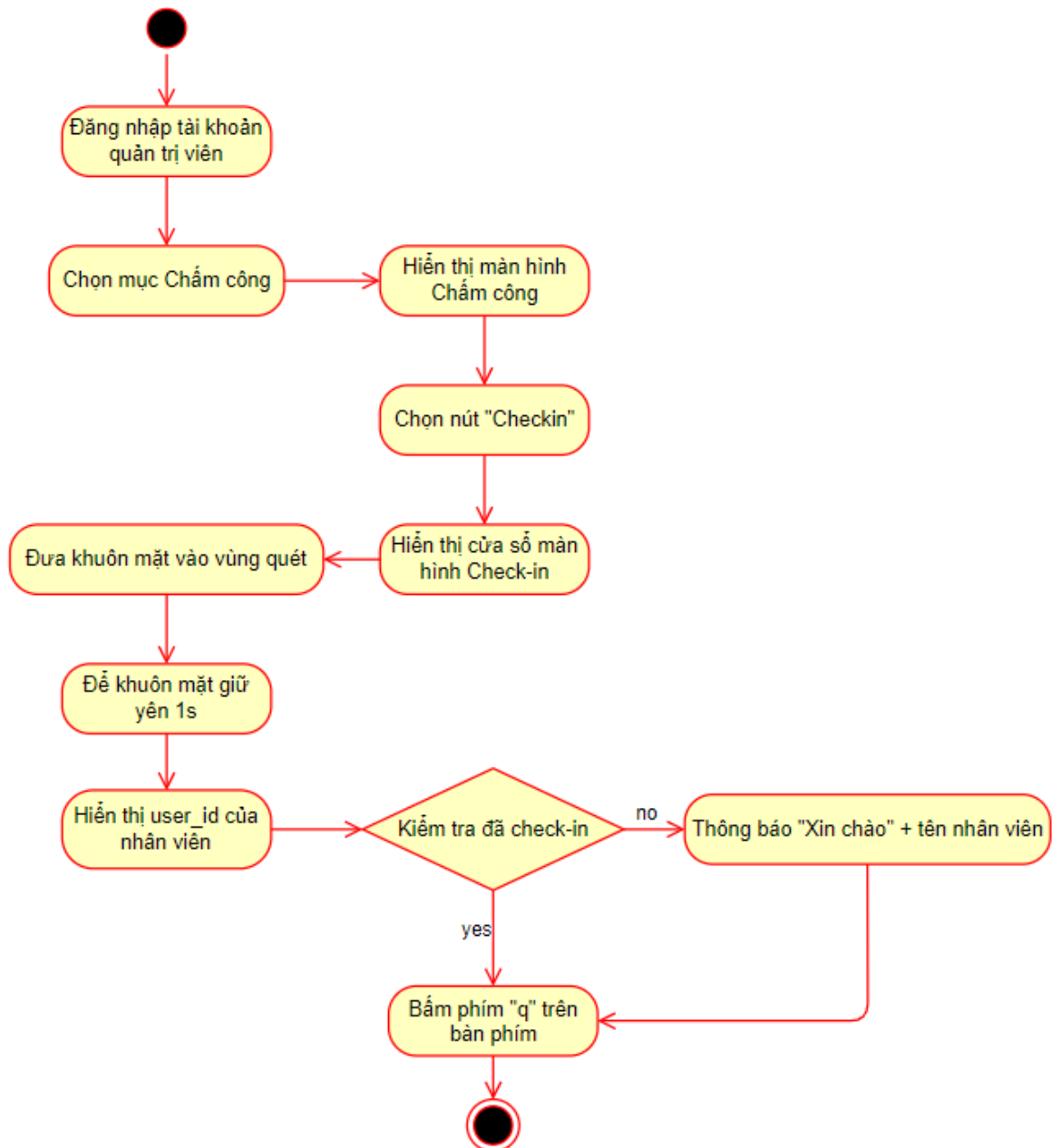
Hình 4.11. Biểu đồ activity danh sách chấm công

Lọc danh sách chấm công



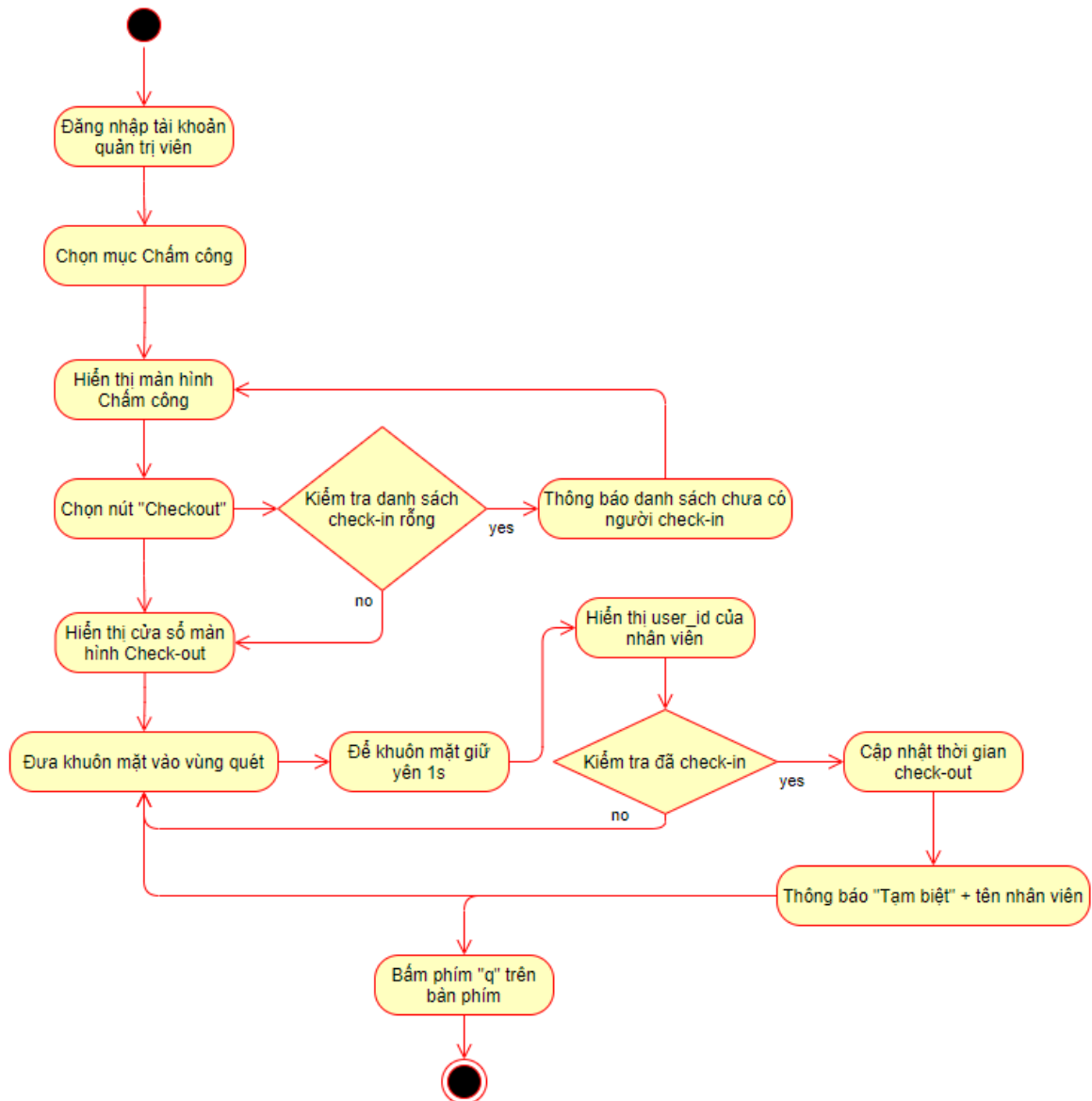
Hình 4.12. Biểu đồ activity lọc danh sách chấm công

Check-in



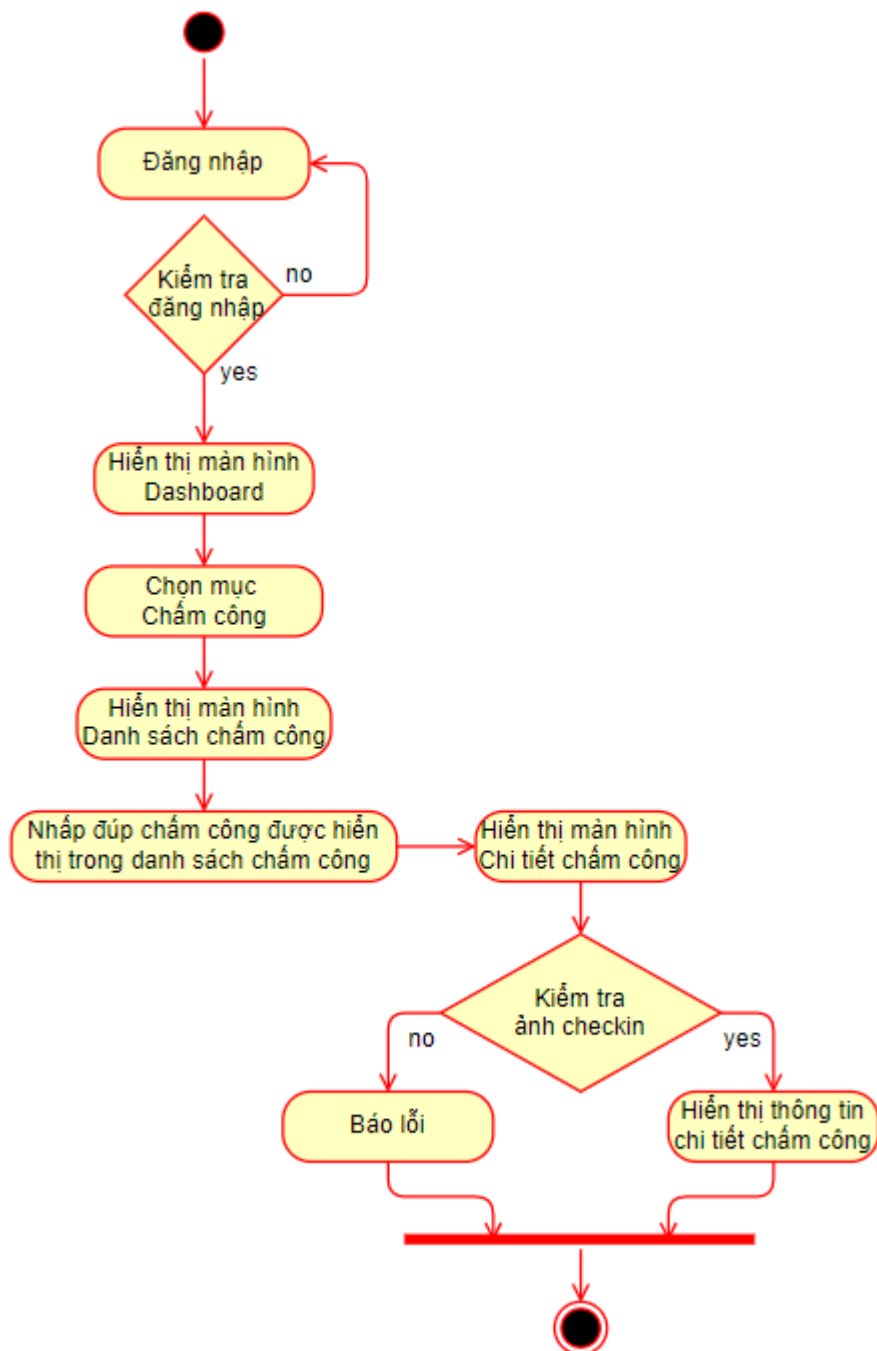
Hình 4.13. Biểu đồ activity check-in

Check-out



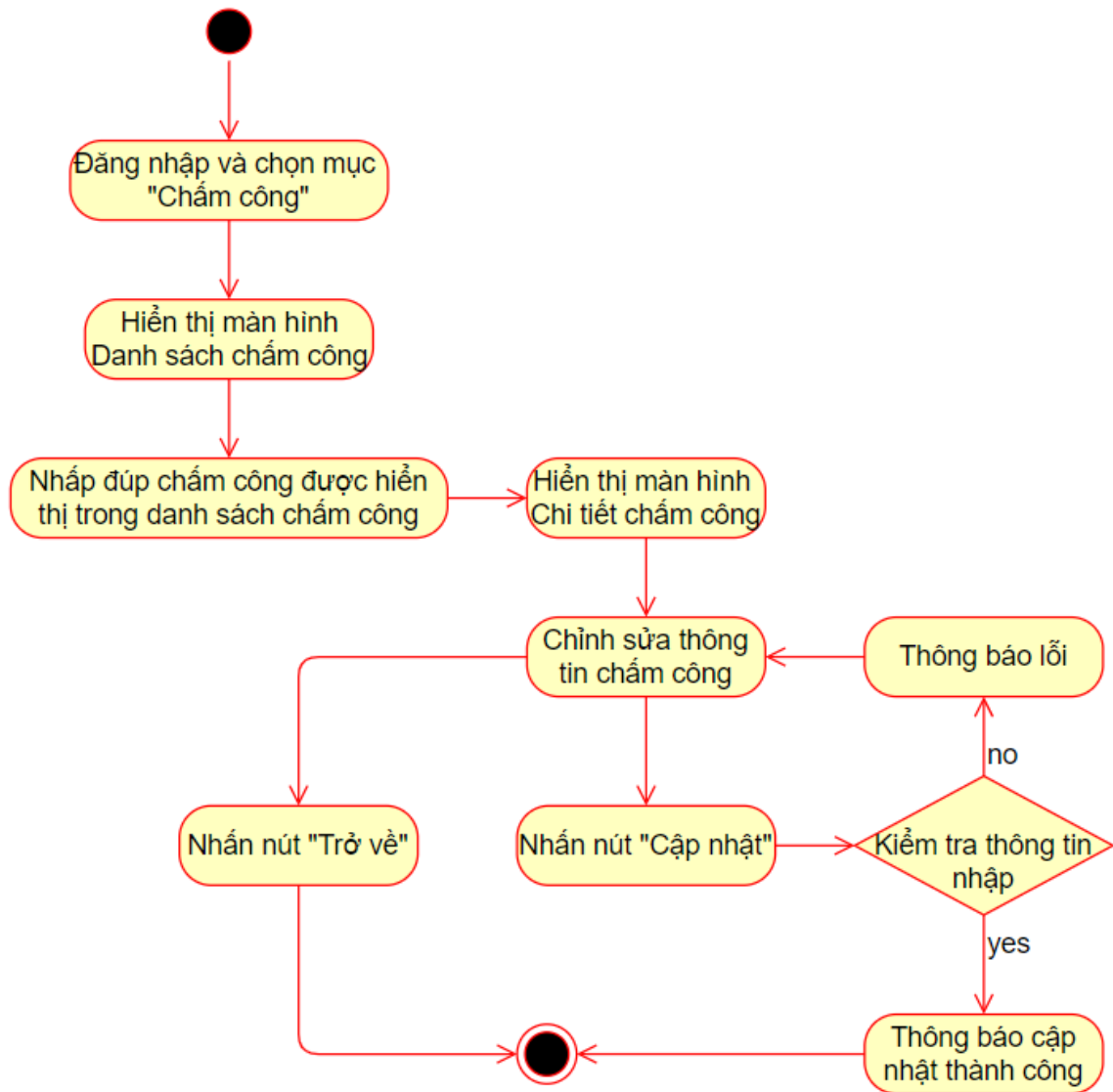
Hình 4.14. Biểu đồ activity check-out

Chi tiết chấm công của nhân viên



Hình 4.15. Biểu đồ activity chi tiết chấm công nhân viên

Cập nhật chi tiết thời gian chấm công



Hình 4.16. Biểu đồ activity cập nhật chi tiết chấm công

4.1.2 Danh sách các tác nhân và mô tả

Tác nhân	Mô tả tác nhân	Ghi chú
Quản lý	Quản lý nhân viên và chấm công.	
Nhân viên	Check-in và check-out.	

4.1.3 Danh sách Use case và mô tả

ID	Tên Use case	Mô tả ngắn gọn Use case	Chức năng	Ghi chú
UC001	UC001_DangNhap	Chỉ được thao tác khi là quản trị viên	Đăng nhập	
UC002	UC002_DanhSachNhanVien		Xem danh sách	
UC003	UC003_TimKiemNhanVien		Tìm kiếm	Theo tên
UC004	UC004_ThemNhanVien		Thêm nhân viên	
UC005	UC005_ChiTietNhanVien		Xem chi tiết	Nhân viên
UC006	UC006_CapNhatNhanVien		Cập nhật	
UC007	UC007_ThemKhuonMat		Thêm khuôn mặt	
UC008	UC008_XemDuLieuKhuonMat		Xem dữ liệu khuôn mặt	

UC009	UC009_HuanLuyenKhuonMat		Huấn luyện khuôn mặt	
UC010	UC010_DanhSachChamCong		Xem danh sách	
UC011	UC011_LocChamCong		Tìm kiếm	Theo ngày
UC012	UC012_Checkin		Check-in	
UC013	UC013_Checkout		Check-out	
UC014	UC014_ChiTietChamCong	Chỉ được thao tác khi là quản trị viên	Xem chi tiết	Chấm công
UC015	UC015_CapNhatChamCong		Cập nhật	

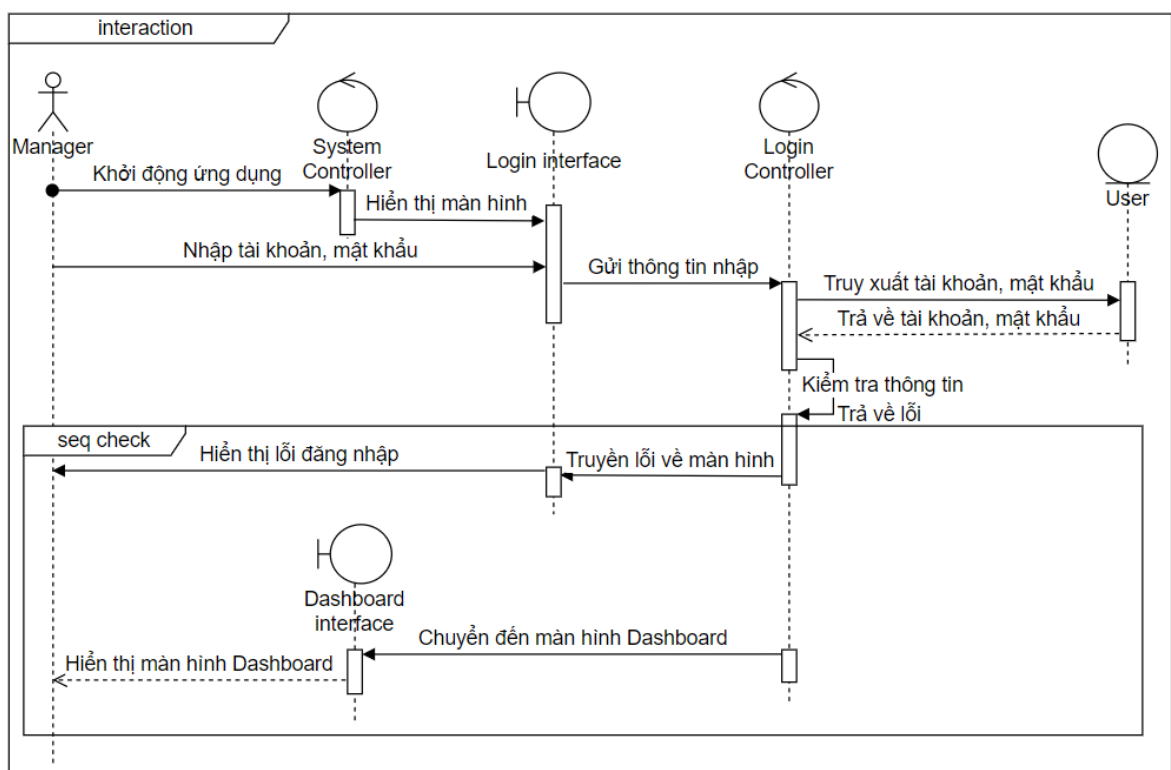
4.2 Đặc tả các yêu cầu chức năng

4.2.1 UC001_DangNhap

Use case: UC001_DangNhap	
Mục đích:	Chỉ cho phép người có tài khoản vào ứng dụng.
Mô tả:	Người dùng thực hiện đăng nhập tài khoản của ứng dụng.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý được cấp tài khoản.
Điều kiện sau:	Quản lý có thể vào ứng dụng để sử dụng và quản lý ứng dụng.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý mở ứng dụng. 2. Sau khi hiện cửa sổ form đăng nhập, quản lý nhập tài

	<p>khoản và mật khẩu của mình.</p> <p>3. Quản lý nhấn nút Đăng nhập.</p> <p>4. Màn hình sẽ được chuyển sang màn hình Dashboard.</p>
Luồng sự kiện phụ (Alternative Flows):	3.2. Nếu dữ liệu nhập hợp lệ, ứng dụng sẽ chuyển đến màn hình Dashboard.
Luồng sự kiện ngoại lệ (Exception Flows):	3.1. Nếu tài khoản hoặc mật khẩu bị sai, ứng dụng sẽ thông báo cho quản lý nhập lại.

Biểu đồ sequence

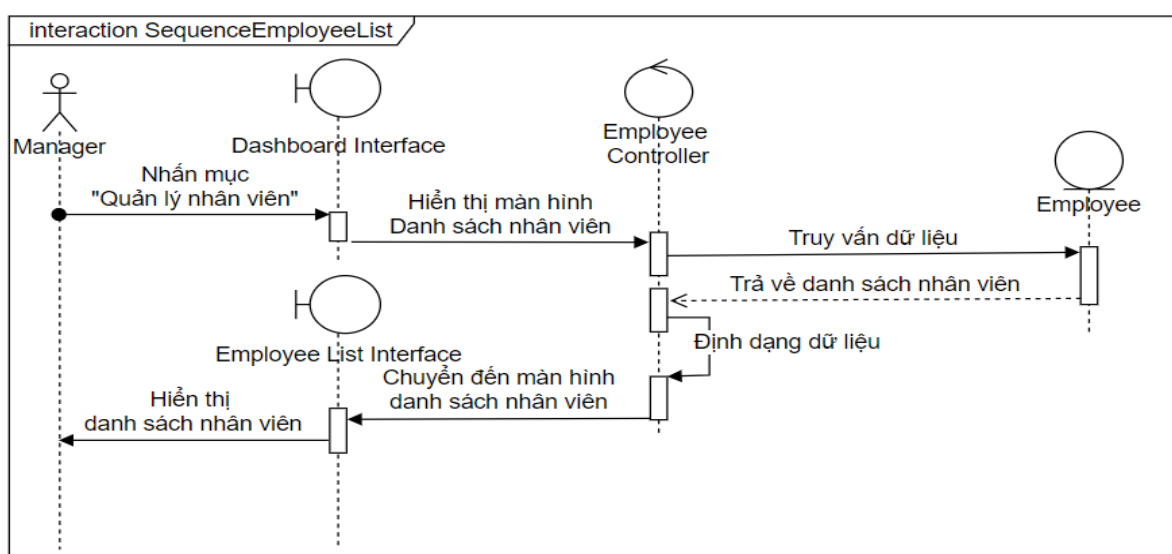


Hình 4.17. Biểu đồ sequence đăng nhập

4.2.2 UC002_DanhSachNhanVien

Use case: UC002_ DanhSachNhanVien	
Mục đích:	Cho phép quản lý nhân viên trong công ty.
Mô tả:	Quản lý nhấn vào mục “Quản lý nhân viên”.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý sau khi đăng nhập ứng dụng.
Điều kiện sau:	Quản lý có thể xem danh sách nhân viên của công ty.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Sau khi đăng nhập thành công. 2. Quản lý nhấn vào mục “Quản lý nhân viên” trên màn hình Dashboard. 3. Màn hình “Danh sách nhân viên” được hiển thị. 4. Danh sách nhân viên sẽ hiển thị trên màn hình “Danh sách nhân viên”.

Biểu đồ sequence

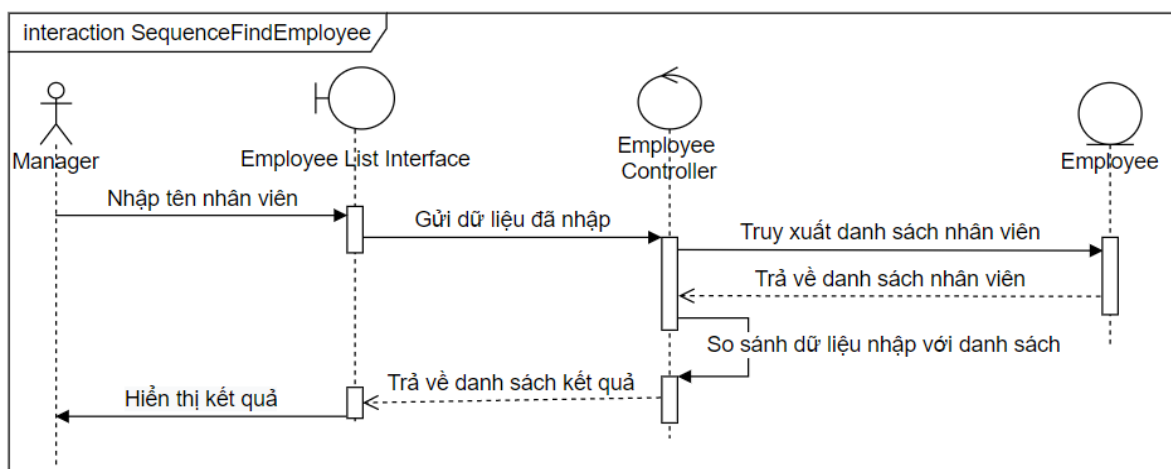


Hình 4.18. Biểu đồ sequence danh sách nhân viên

4.2.3 UC003_TimKiemNhanVien

Use case: UC003_TimKiemNhanVien	
Mục đích:	Quản lý quản lý nhân viên một cách dễ dàng hơn.
Mô tả:	Quản lý nhập tên của nhân viên.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý đã đăng nhập vào ứng dụng.
Điều kiện sau:	Quản lý có thể tìm kiếm nhân viên dễ dàng.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý vào màn hình “Danh sách nhân viên”. 2. Quản lý nhập tên của nhân viên (có dấu). 3. Kết quả hiển thị theo tên nhân viên chứa từ khóa. 4. Màn hình sẽ được chuyển sang màn hình Dashboard.
Luồng sự kiện phụ (Alternative Flows):	3.1. Nếu không có tên của nhân viên, danh sách nhân viên sẽ hiển thị trống.

Biểu đồ sequence

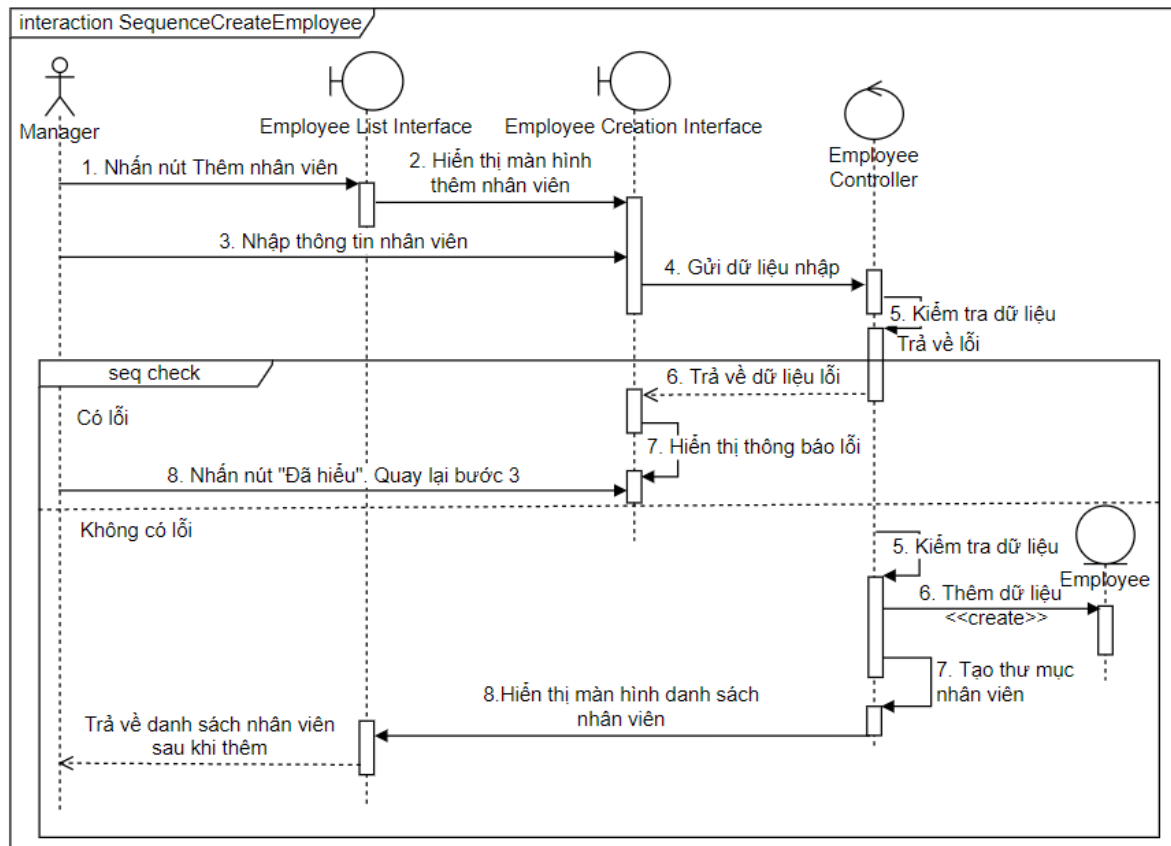


Hình 4.19. Biểu đồ sequence tìm kiếm nhân viên

4.2.4 UC004_ThemNhanVien

Use case: UC004_ThemNhanVien	
Mục đích:	Cho phép quản lý thêm nhân viên mới vào danh sách.
Mô tả:	Quản lý nhập thông tin của nhân viên mới và chọn đăng ký.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý được cấp tài khoản.
Điều kiện sau:	Quản lý có thể thêm nhân viên mới vào danh sách.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập thành công, nhấn vào mục “Quản lý nhân viên”. 2. Nhấn chọn “Thêm nhân viên”. 3. Quản lý nhập thông tin của nhân viên và nhấn “Đăng ký”. 4. Màn hình sẽ được chuyển sang màn hình “Danh sách nhân viên”.
Luồng sự kiện phụ (Alternative Flows):	3.2. Nếu dữ liệu nhập hợp lệ, dữ liệu nhân viên sẽ được thêm vào danh sách và chuyển đến màn hình “Danh sách nhân viên”.
Luồng sự kiện ngoại lệ (Exception Flows):	3.1. Nếu thông tin không hợp lệ ứng dụng sẽ thông báo cho quản lý.

Biểu đồ sequence



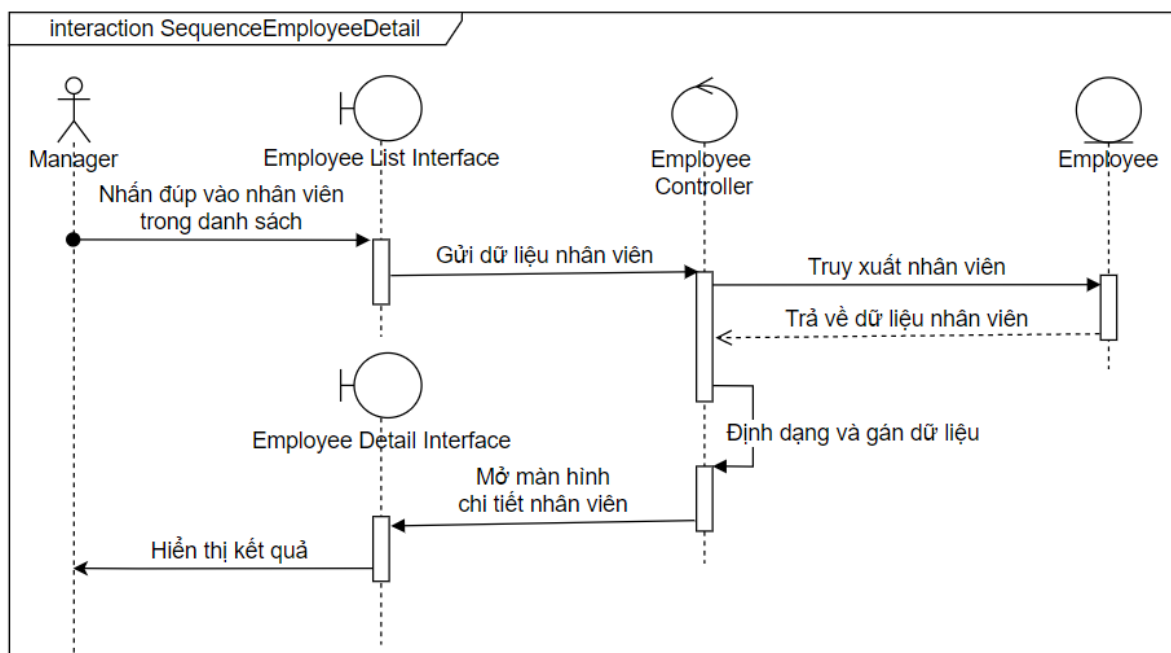
Hình 4.20. Biểu đồ sequence thêm nhân viên

4.2.5 UC005_ChiTietNhanVien

Use case: UC005_ChiTietNhanVien	
Mục đích:	Cho phép quản lý xem chi tiết thông tin của nhân viên và có thể chỉnh sửa thông tin sai.
Mô tả:	Quản lý nhấn đúp vào nhân viên cần xem thông tin.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý sau khi đăng nhập đăng nhập và nhấn đúp vào nhân viên trong danh sách nhân viên.
Điều kiện sau:	Quản lý có thể xem chi tiết và chỉnh sửa thông tin nhân viên.
Luồng sự kiện	1. Quản lý sau khi đăng nhập thành công và nhấn vào

chính (Basic flows)	<p>mục “Quản lý nhân viên”.</p> <ol style="list-style-type: none"> Quản lý nhấn đúp vào nhân viên trong danh sách nhân viên được hiển thị ở màn hình “Danh sách nhân viên”. Thông tin chi tiết của nhân viên được hiện ra. Quản lý nhấn nút “Trở về” để quay lại màn hình “Quản lý nhân viên”.
Luồng sự kiện phụ (Alternative Flows):	<ol style="list-style-type: none"> Nếu nhân viên chưa thêm khuôn mặt, ứng dụng sẽ dùng ảnh mặc định. Nếu nhân viên đã thêm khuôn mặt, ứng dụng sẽ dùng khuôn mặt đầu tiên của nhân viên.

Biểu đồ sequence

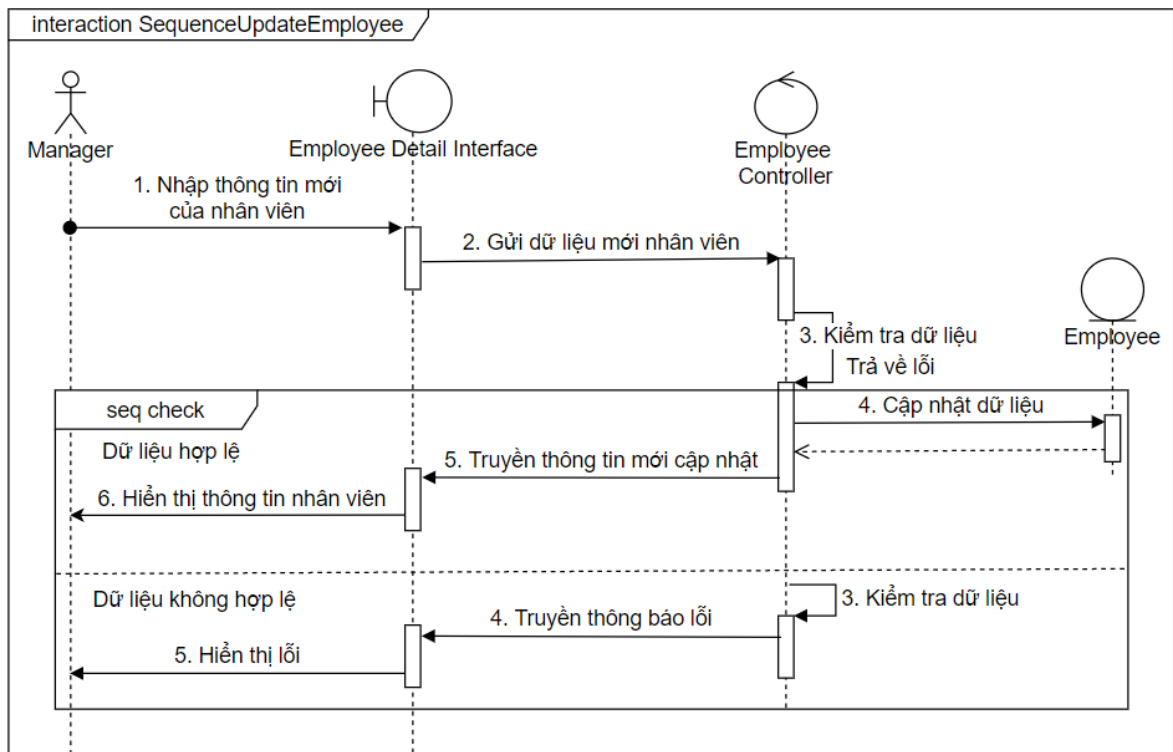


Hình 4.21. Biểu đồ sequence chi tiết nhân viên

4.2.6 UC006_CapNhatNhanVien

Use case: UC006_CapNhatNhanVien	
Mục đích:	Cho phép quản lý chỉnh sửa thông tin chi tiết của nhân viên.
Mô tả:	Quản lý thay đổi thông tin của nhân viên và nhấn “Cập nhật”.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý đăng nhập thành công.
Điều kiện sau:	Quản lý chỉnh sửa thông tin nhân viên thành công.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập thành công và nhấn mục “Quản lý nhân viên”. 2. Quản lý nhấn đúp vào nhân viên trong danh sách nhân viên được hiển thị ở màn hình “Danh sách nhân viên”. 3. Thông tin chi tiết của nhân viên được hiện ra. 4. Quản lý có thể nhấn đúp để cập nhật hình đại diện cho nhân viên. 5. Quản lý chỉnh sửa thông tin của nhân viên và nhấn cập nhật. 6. Ứng dụng thông báo cho quản lý cập nhật thành công.
Luồng sự kiện phụ (Alternative Flows):	3.2. Nếu thông tin chỉnh sửa hợp lệ, ứng dụng sẽ cập nhật lại dữ liệu mới cho nhân viên và thông báo cập nhật thành công.
Luồng sự kiện ngoại lệ	3.1. Nếu thông tin chỉnh sửa không hợp lệ, ứng dụng sẽ thông báo cho quản lý.

Biểu đồ sequence



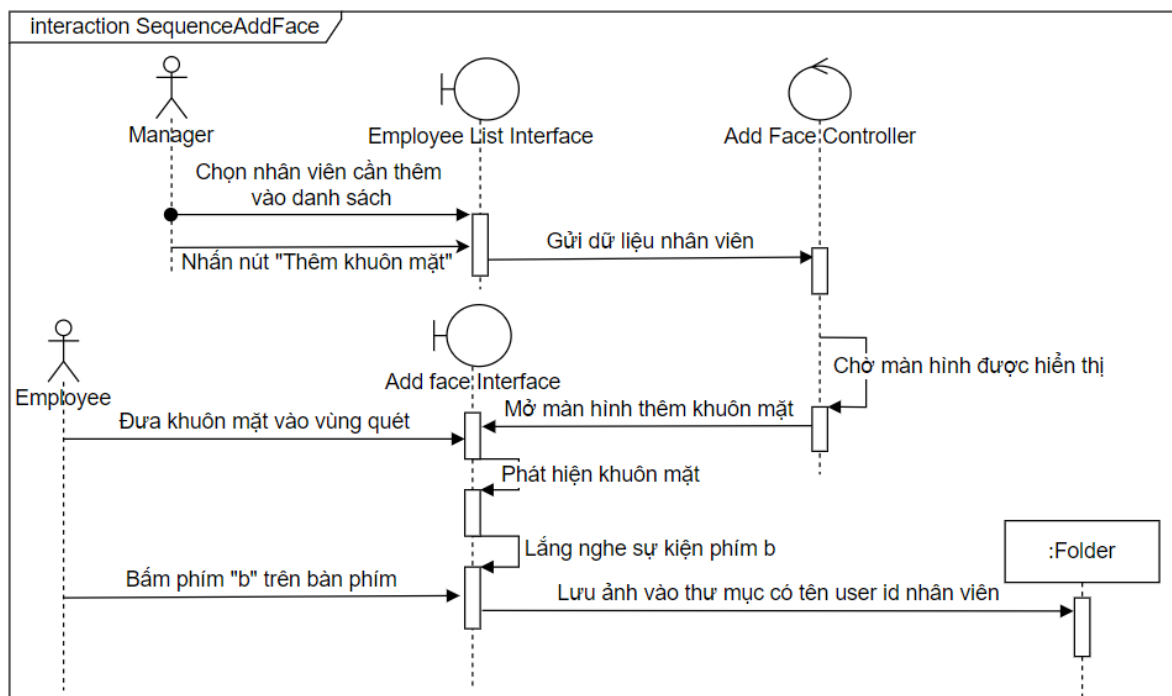
Hình 4.22. Biểu đồ sequence cập nhật thông tin nhân viên

4.2.7 UC007_ThemKhuonMat

Use case: UC007_ThemKhuonMat	
Mục đích:	Cho phép quản lý thêm dữ liệu khuôn mặt của nhân viên để phục vụ cho việc huấn luyện mô hình nhận dạng khuôn mặt.
Mô tả:	Quản lý chọn “Thêm khuôn mặt” trong màn hình “Danh sách nhân viên”.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý phải đăng nhập trước
Điều kiện sau:	Quản lý thêm dữ liệu khuôn mặt cho nhân viên.
Luồng sự kiện chính (Basic)	1. Quản lý sau khi đăng nhập nhấn vào “Quản lý nhân viên”.

flows)	<p>2. Chọn nhân viên cần thêm khuôn mặt ở danh sách hiển thị nhân viên.</p> <p>3. Quản lý nhấn nút “Thêm khuôn mặt”, cửa sổ “Thêm khuôn mặt” hiển thị.</p> <p>4. Quản lý nhấn phím “b” trên bàn phím để thêm khuôn mặt và nhấn phím “q” để thoát khỏi cửa sổ.</p>
Luồng sự kiện phụ (Alternative Flows):	4.2. Nếu số khuôn mặt trên 100 có thể nhấn “q” để thoát khỏi cửa sổ.
Luồng sự kiện ngoại lệ	4.1. Nếu số khuôn mặt chưa đạt tới 100 thì không thể thoát khỏi cửa sổ.

Biểu đồ sequence

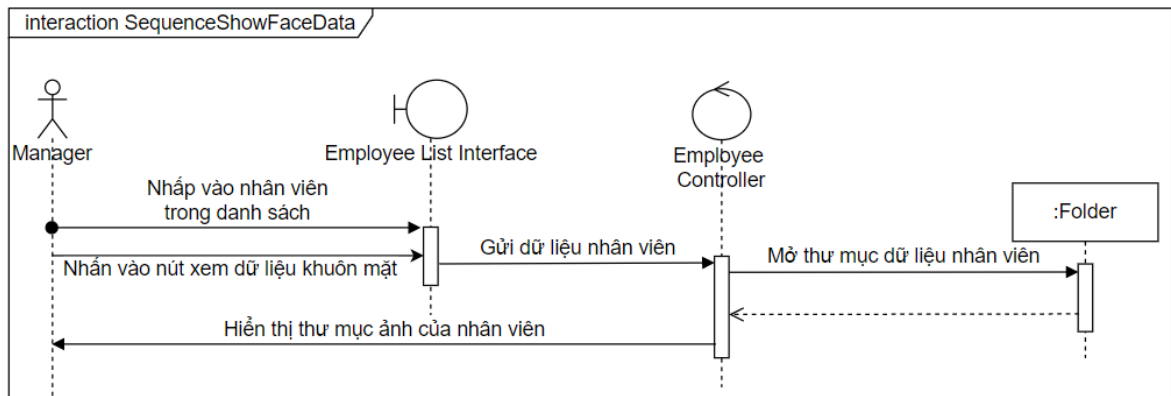


Hình 4.23. Biểu đồ sequence thêm khuôn mặt

4.2.8 UC008_XemDuLieuKhuonMat

Use case: UC008_XemDuLieuKhuonMat	
Mục đích:	Cho phép quản lý xem dữ liệu khuôn mặt của nhân viên.
Mô tả:	Quản lý chọn nhân viên ở trong danh sách nhân viên được hiển thị ở màn hình “Danh sách nhân viên”.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý sau khi đăng nhập, nhấn vào mục “Quản lý nhân viên” chọn chức năng “Xem dữ liệu khuôn mặt”.
Điều kiện sau:	Quản lý có thể xem dữ liệu khuôn mặt.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập, nhấn vào mục “Quản lý nhân viên”. 2. Quản lý chọn nhân viên cần xem dữ liệu được hiển thị trong danh sách nhân viên. 3. Quản lý nhấn nút “Xem dữ liệu khuôn mặt”. 4. Hiển thị thư mục trên file explorer có tên nhân viên đó.
Luồng sự kiện phụ (Alternative Flows):	3.2. Nếu có thư mục chứa dữ liệu, ứng dụng sẽ hiển thị thư mục của nhân viên đó.
Luồng sự kiện ngoại lệ (Exception Flows):	3.1. Nếu chưa có thư mục chứa dữ liệu, ứng dụng sẽ tạo thư mục rỗng với tên là user id của nhân viên đó.

Biểu đồ sequence

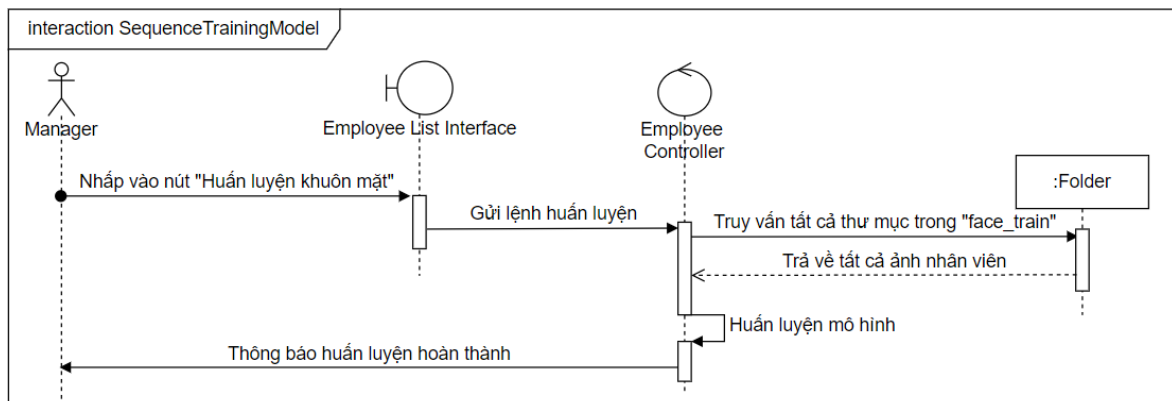


Hình 4.24. Biểu đồ sequence xem dữ liệu khuôn mặt

4.2.9 UC009_HuanLuyenKhuonMat

Use case: UC009_HuanLuyenKhuonMat	
Mục đích:	Huấn luyện mô hình để nhận dạng khuôn mặt của nhân viên.
Mô tả:	Quản lý nhấn nút “Huấn luyện khuôn mặt” để thực hiện huấn luyện, sau khi hoàn tất ứng dụng sẽ thông báo cho quản lý.
Tác nhân:	Quản lý.
Điều kiện trước:	Phải có dữ liệu ảnh của nhân viên trong thư mục “face_train”.
Điều kiện sau:	Mô hình nhận dạng khuôn mặt sẽ được cập nhật nếu có dữ liệu nhân viên mới vào hay thay đổi dữ liệu cũ.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập thành công, chọn mục “Quản lý nhân viên”. 2. Màn hình Danh sách nhân viên hiển thị ra, quản lý nhấn vào nút “Huấn luyện khuôn mặt”. 3. Ứng dụng sẽ load trong vòng 10 – 30s tùy vào lượng dữ liệu ảnh, để huấn luyện tập dữ liệu nhận dạng. 4. Ứng dụng thông báo cho quản lý sau khi hoàn tất.

Biểu đồ sequence

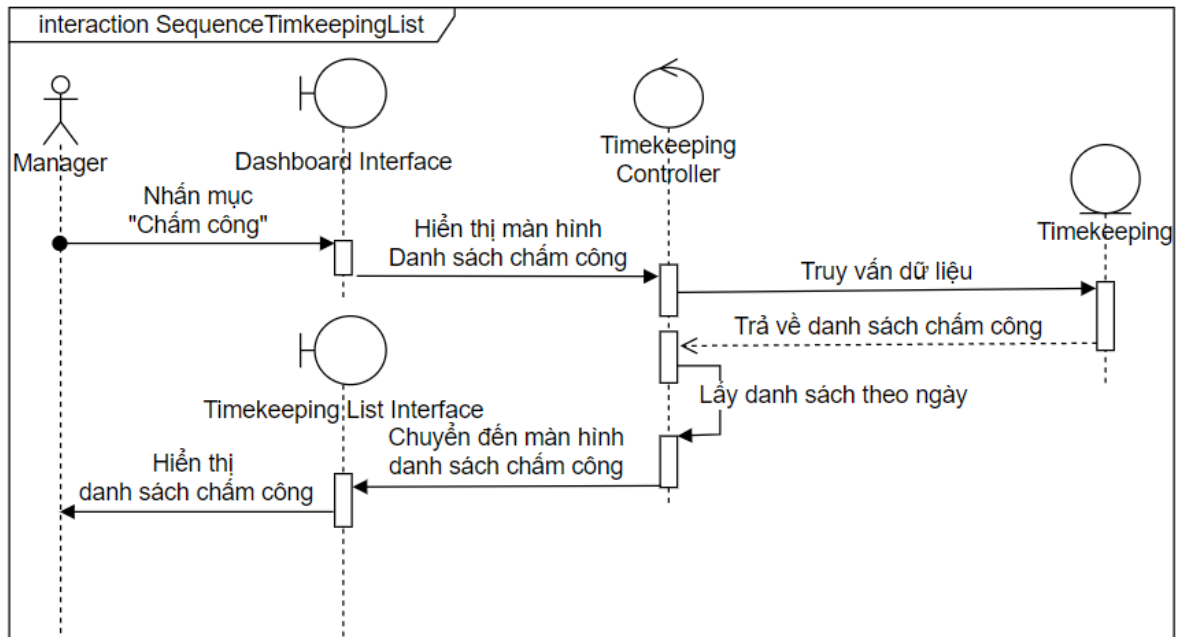


Hình 4.25. Biểu đồ sequence huấn luyện mô hình nhận dạng

4.2.10 UC010_DanhSachChamCong

Use case: UC010_DanhSachChamCong	
Mục đích:	Cho phép quản lý xem danh sách nhân viên đã chấm công.
Mô tả:	Quản lý chọn mục “Chấm công” ở bảng điều khiển.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý đăng nhập thành công.
Điều kiện sau:	Quản lý có thể vào ứng dụng để sử dụng và quản lý ứng dụng.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập ứng dụng, chọn mục “Chấm công”. 2. Ứng dụng truy xuất dữ liệu chấm công của ngày hiện tại và truyền dữ liệu tới màn hình “Chấm công”. 3. Màn hình “Chấm công” được hiển thị. 4. Danh sách chấm công được hiển thị trong màn hình “Chấm công”.
Luồng sự kiện ngoại lệ	3.1. Nếu nhân viên chưa thêm khuôn mặt, ứng dụng sẽ không hiển thị trên danh sách chấm công.

Biểu đồ sequence.



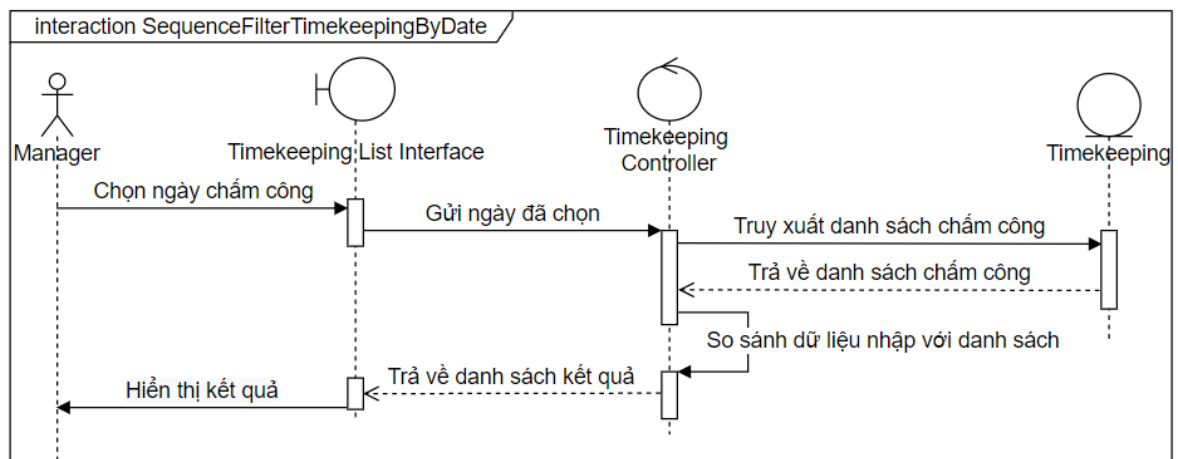
Hình 4.26. Biểu đồ sequence danh sách chấm công

4.2.11 UC011_LocChamCongTheoNgay

Use case: UC011_LocChamCongTheoNgay	
Mục đích:	Cho phép quản lý lọc danh sách chấm công theo ngày.
Mô tả:	Quản lý sau khi vào danh sách chấm công của ứng dụng, chọn ngày để lọc danh sách các ngày trước đó.
Tác nhân:	Quản lý.
Điều kiện trước:	Quản lý đăng nhập thành công.
Điều kiện sau:	Quản lý có thể xem lại danh sách chấm công những ngày trước đó.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý chọn mục “Chấm công” trên màn hình Dashboard. 2. Sau khi hiển thị màn hình “Danh sách chấm công”.

	<p>3. Quản lý chọn “Ngày chấm công” để lọc danh sách.</p> <p>4. Ứng dụng hiển thị kết quả theo ngày quản lý đã chọn.</p>
--	--

Biểu đồ sequence.



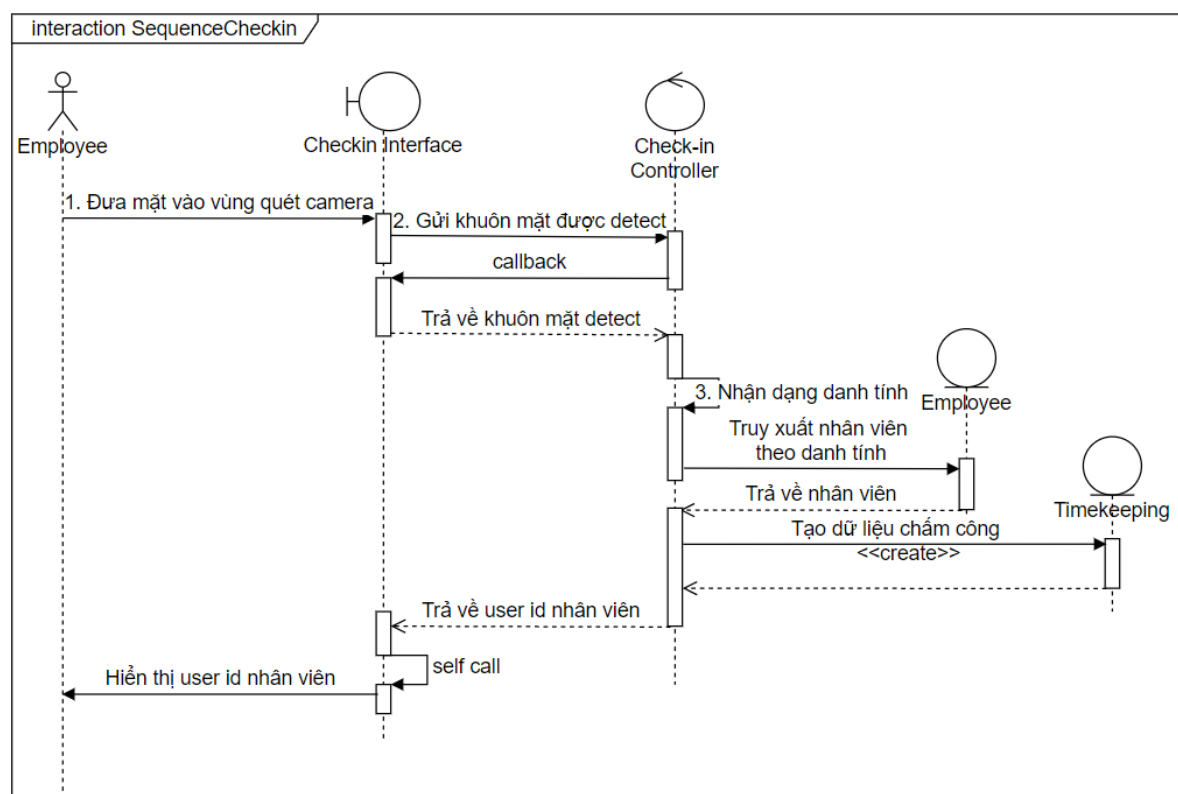
Hình 4.27. Biểu đồ sequence lọc danh sách chấm công theo ngày

4.2.12 UC012_Checkin

Use case: UC012_Checkin	
Mục đích:	Cho phép nhân viên trong công ty check-in bằng khuôn mặt.
Mô tả:	Quản lý mở ứng dụng để nhân viên vào check-in khuôn mặt.
Tác nhân:	Nhân viên.
Điều kiện trước:	Nhân viên phải có thông tin và dữ liệu khuôn mặt.
Điều kiện sau:	Nhân viên chấm công vào (check-in) thành công.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Nhân viên vào vùng quét của camera. 2. Ứng dụng phát hiện khuôn mặt và xác nhận danh tính. 3. Sau khi xác nhận nhân viên có trong danh sách nhân viên, ứng dụng sẽ thông báo bằng giọng nói với tin

	<p>nhấn “Xin chào” + tên nhân viên.</p> <p>4. Quản lý bấm giữ phím “q” trên bàn phím để thoát.</p>
Luồng sự kiện phụ (Alternative Flows):	2.1. Ứng dụng chỉ tắt khi quản lý bấm phím “q”, nếu không camera sẽ mở và nhận dạng khi có gương mặt lọt vào phạm vi quét của nó.
Luồng sự kiện ngoại lệ (Exception Flows):	2.1. Nếu khuôn mặt được xác nhận là “Unknown”, ứng dụng sẽ hiển thị trên màn hình.

Biểu đồ sequence

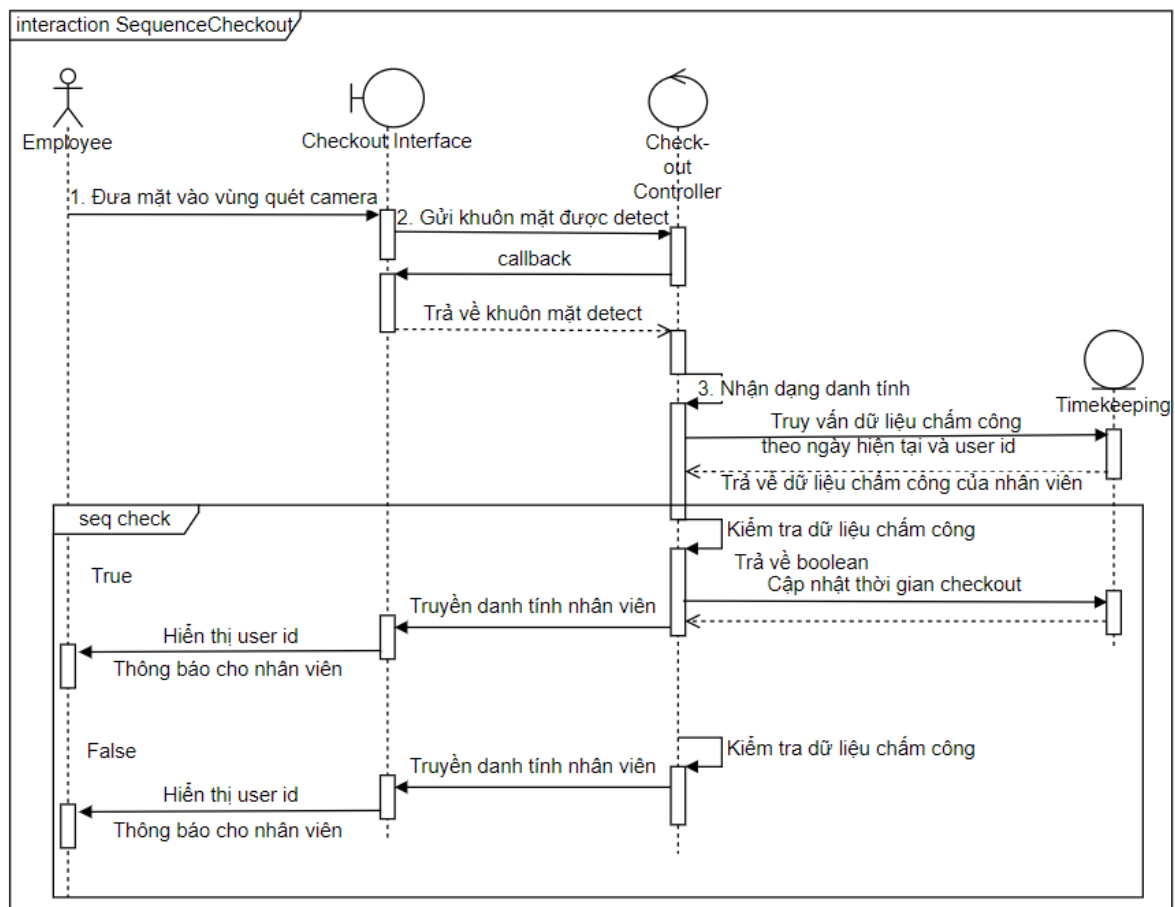


Hình 4.28. Biểu đồ sequence check-in

4.2.13 UC013_Checkout

Use case: UC013_Checkout	
Mục đích:	Cho phép nhân viên trong công ty check-out bằng khuôn mặt.
Mô tả:	Quản lý mở ứng dụng để nhân viên vào check-out khuôn mặt.
Tác nhân:	Nhân viên.
Điều kiện trước:	Nhân viên vào phạm vi quét của camera.
Điều kiện sau:	Nhân viên check-out thành công.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Nhân viên đưa khuôn mặt vào phạm vi quét của camera. 2. Ứng dụng hiển thị user id của nhân viên. 3. Ứng dụng kiểm tra nhân viên đã check-in chưa. 4. Bấm phím “b” trên bàn phím để thoát khỏi cửa sổ.
Luồng sự kiện phụ (Alternative Flows):	3.2. Nếu nhân viên đã check-in, ứng dụng sẽ cập nhật thời gian check-out.
Luồng sự kiện ngoại lệ (Exception Flows):	3.1. Nếu nhân viên chưa check-in, ứng dụng không lưu thời gian check-out của nhân viên.

Biểu đồ sequence



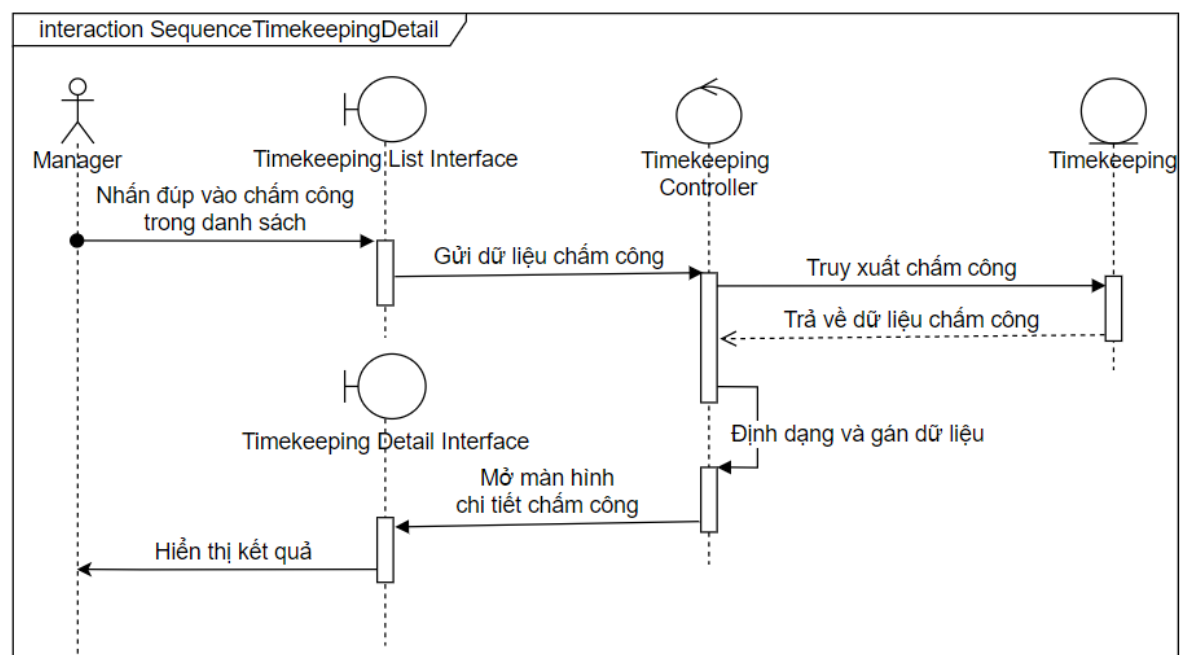
Hình 4.29. Biểu đồ sequence check-out

4.2.14 UC014_ChiTietChamCong

Use case: UC014_ChiTietChamCong	
Mục đích:	Cho phép quản lý xem chi tiết thông tin chấm công của từng nhân viên.
Mô tả:	Quản lý vào mục “Chấm công”, sau khi ứng dụng hiển thị danh sách chấm công, quản lý nhấp đúp vào dòng chấm công trên danh sách.
Tác nhân:	Quản lý.
Điều kiện trước:	Phải có dữ liệu trong danh sách.

Điều kiện sau:	Quản lý có thể xem được chi tiết thông tin chấm công của từng nhân viên.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập thành công, chọn mục “Chấm công”. 2. Sau khi danh sách chấm công hiển thị trong màn hình “Chấm công”, quản lý nhấp đúp vào dòng chấm công trên danh sách. 3. Ứng dụng sẽ tự chuyển sang màn hình “Chi tiết chấm công”. 4. Để quay lại, quản lý có thể nhấn nút “Trở về”.
Luồng sự kiện phụ	3.2. Nếu dữ liệu nhập hợp lệ, ứng dụng sẽ chuyển đến màn hình Dashboard.
Luồng sự kiện ngoại lệ	3.1. Nếu tài khoản hoặc mật khẩu bị sai, ứng dụng sẽ thông báo cho quản lý nhập lại.

Biểu đồ sequence

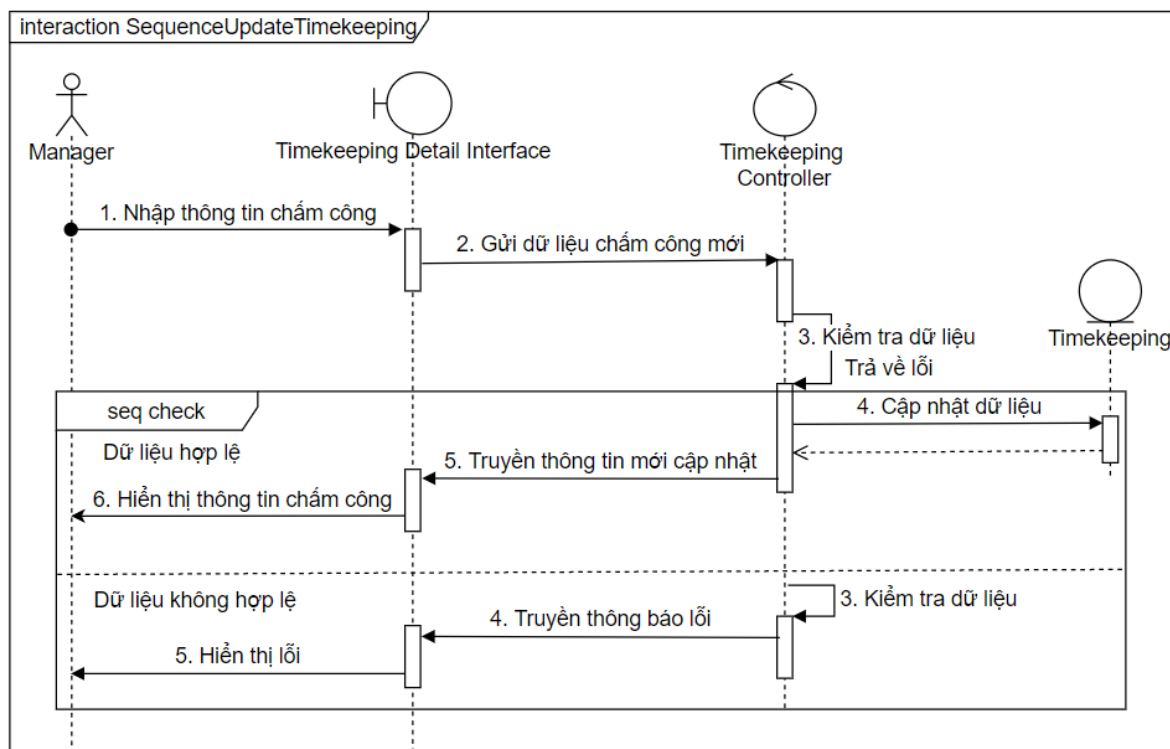


Hình 4.30. Biểu đồ sequence chi tiết chấm công

4.2.15 UC015_CapNhatChamCong

Use case: UC015_CapNhatChamCong	
Mục đích:	Cho phép quản lý chỉnh sửa thông tin chấm công khi có sai sót.
Mô tả:	Quản lý vào màn hình Chi tiết chấm công, chỉnh sửa thông tin và nhấn nút “Cập nhật”.
Tác nhân:	Quản lý.
Điều kiện trước:	Phải có dữ liệu trên danh sách chấm công.
Điều kiện sau:	Quản lý có thể chỉnh sửa thông tin chấm công nếu có sai sót.
Luồng sự kiện chính (Basic flows)	<ol style="list-style-type: none"> 1. Quản lý sau khi đăng nhập thành công, chọn mục “Chấm công”. 2. Hiện thị màn hình Danh sách chấm công, quản lý nhấn đúp vào chấm công trên danh sách chấm công. 3. Màn hình chi tiết được hiển thị, quản lý chỉnh sửa thông tin và nhấn nút “Cập nhật”. 4. Để quay lại, quản lý nhấn nút “Trở về”.
Luồng sự kiện phụ (Alternative Flows):	3.2. Nếu thông tin chỉnh sửa hợp lệ, ứng dụng thông
Luồng sự kiện ngoại lệ (Exception Flows):	3.1. Nếu thông tin chỉnh sửa không hợp lệ, ứng dụng sẽ báo lỗi.

Biểu đồ sequence



Hình 4.31. Biểu đồ sequence cập nhật thông tin chấm công

4.3 Chuẩn bị

4.3.1 Môi trường

Editor: Visual Studio Code 2019.

Ngôn ngữ lập trình: Python 3 (version 3.6 trở lên).

Môi trường ảo: Anaconda.

4.3.2 Thư viện

MediaPipe version 0.8.3.1

Numpy version 1.18.5

PIL: Pillow version 8.2.0

Keras-facenet version 0.3.2

Opencv-python version 4.5.1.48

Openpyxl version 3.0.7

Pandas version 1.2.4

Tensorflow version 2.3.0

Tkcalendar version 1.6.1

XlsxWriter servision 1.4.2

Scipy version 1.4.1

Tkinter

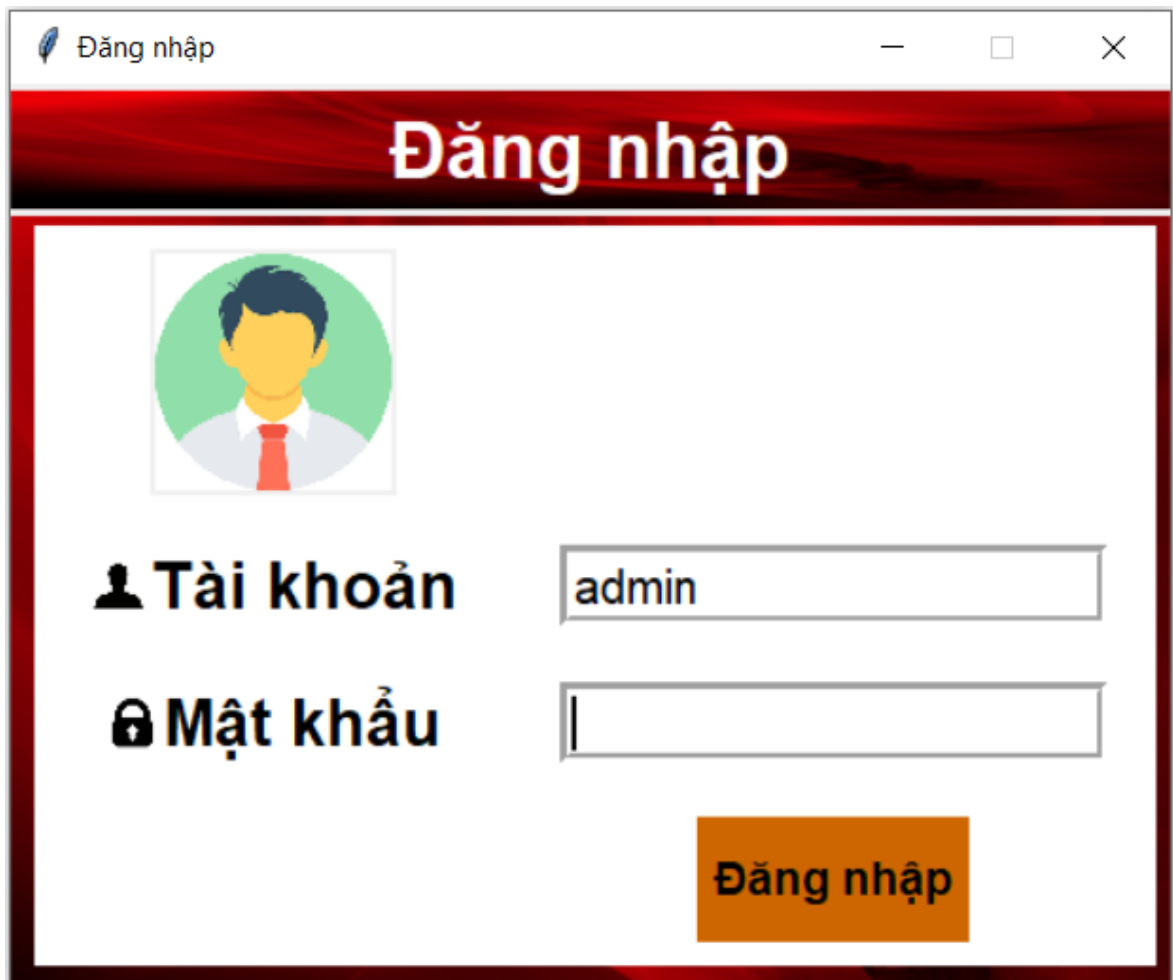
Pickle

Pyttsx3 version 2.90

4.4 Kết quả chương trình

4.4.1 Màn hình đăng nhập

Nhập tài khoản và mật khẩu của quản lý để vào ứng dụng.

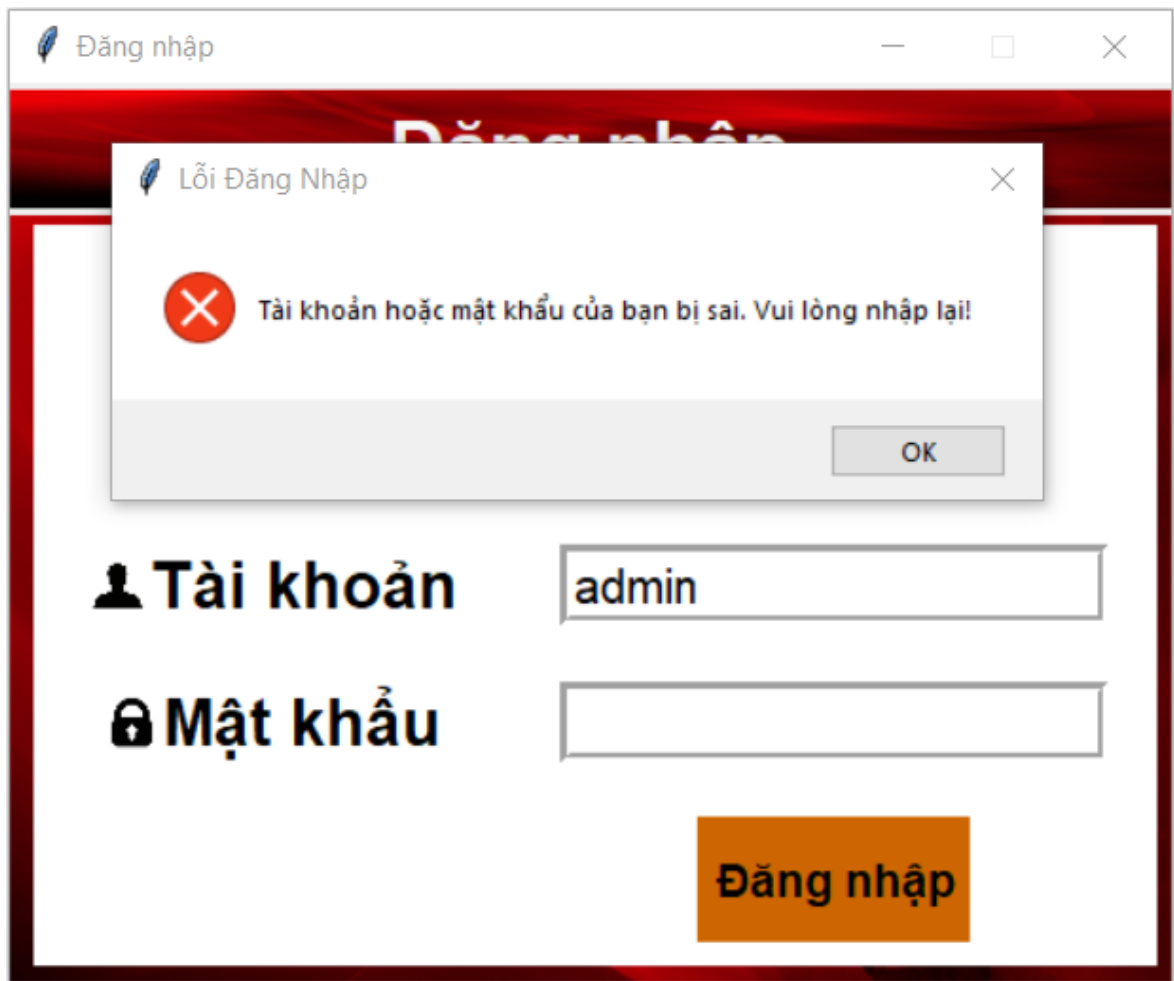


The screenshot shows a login window titled "Đăng nhập" (Login). The window has a red header bar with the title. Below the header, there is a circular profile picture of a person with dark hair, wearing a white shirt and a red tie. To the right of the profile picture, there are two input fields. The first field is labeled "Tài khoản" (Account) and contains the text "admin". The second field is labeled "Mật khẩu" (Password) and is empty. Below the input fields, there is a large orange button labeled "Đăng nhập" (Login).

Hình 4.32. Giao diện đăng nhập ứng dụng

Nếu nhập sai hoặc không nhập đủ thông tin, màn hình sẽ hiển thị lỗi.

Nhấn “OK” hoặc bấm phím Enter trên bàn phím để đăng nhập lại.



Hình 4.33. Giao diện lỗi đăng nhập

Sau khi đăng nhập thành công, ứng dụng sẽ tự động chuyển đến màn hình Dashboard, chứa các mục chức năng quản lý.

4.4.2 Màn hình Dashboard

Hiển thị các mục quản lý (nhân viên, chấm công) và chức năng đăng xuất, quản lý có thể chọn để vào thao tác với các mục này.

Ở mục **Quản lý nhân viên** ta có các chức năng chính như sau:

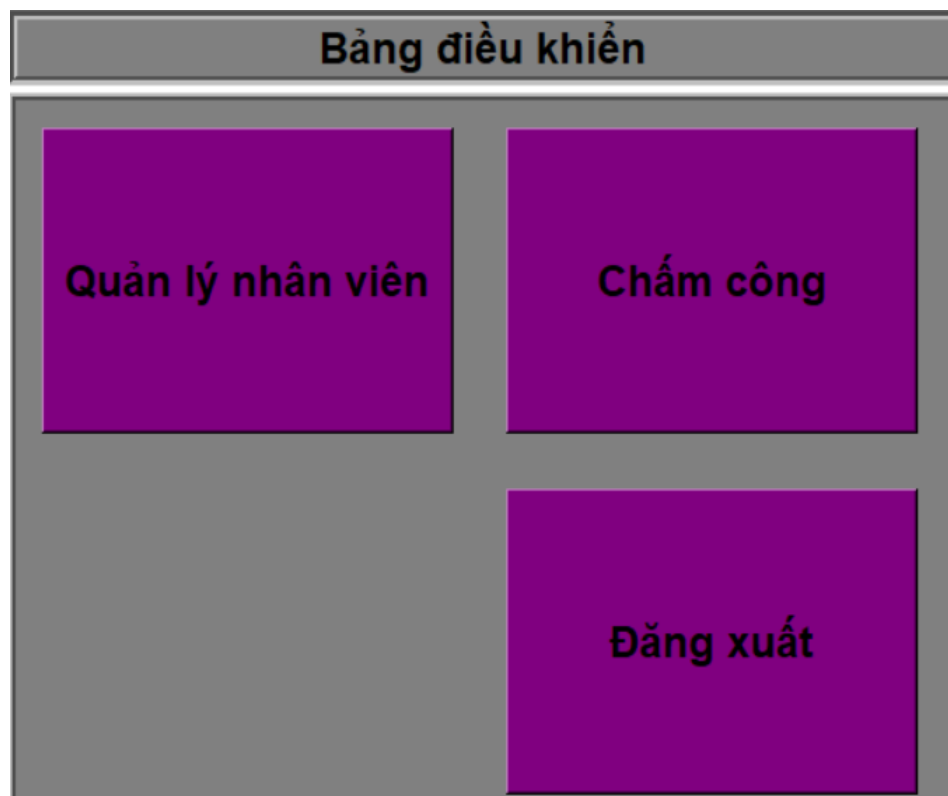
- Xem danh sách nhân viên.
- Tìm kiếm nhân viên.
- Thêm nhân viên mới.

- Xem thông tin chi tiết nhân viên.
- Chỉnh sửa thông tin nhân viên.
- Thêm khuôn mặt.
- Xem dữ liệu khuôn mặt.
- Training khuôn mặt.

Ở mục **Chấm công** ta có các chức năng chính như sau:

- Xem danh sách chấm công.
- Lọc danh sách chấm công theo ngày.
- Check-in.
- Check-out.
- Xem và chỉnh sửa thông tin chi tiết chấm công.

Nếu như không muốn dùng ứng dụng nữa, ta có thể chọn mục Đăng xuất để thoát khỏi màn hình chức năng của ứng dụng.



Hình 4.34. Giao diện bảng điều khiển

4.4.3 Màn hình danh sách nhân viên

STT	Họ và tên	Ngày sinh	Email
1	Hồ Duy Quang	07/05/1999	hoduyquang@gmail.com
2	Trần Quang Trung	06/06/1999	trung@gmail.com

Hình 4.35. Giao diện danh sách nhân viên

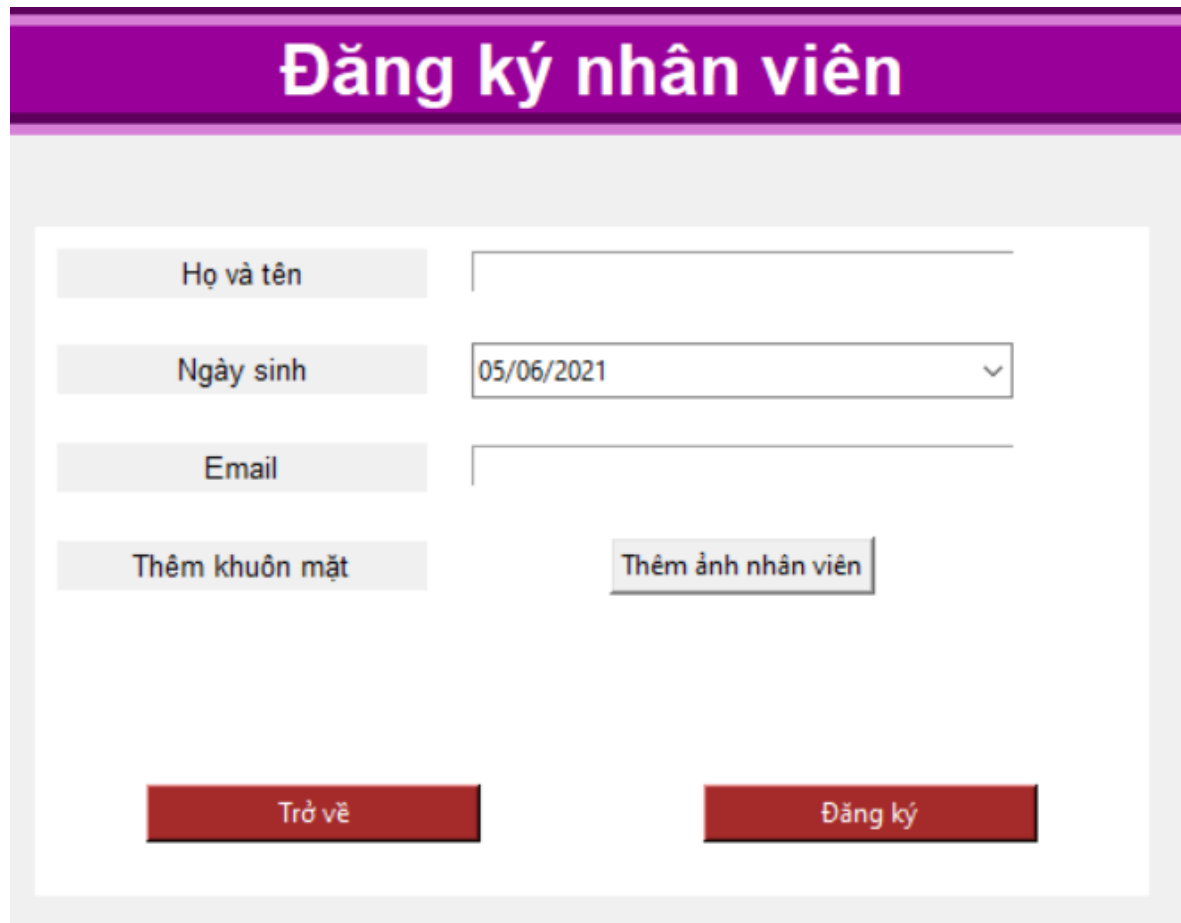
Chức năng tìm kiếm nhân viên bằng tên nhân viên.

Tìm kiếm chữ thường, chữ hoa, nếu tên nhân viên có dấu thì phải viết dấu để có thể tìm kết quả chính xác.

STT	Họ và tên	Ngày sinh	Email
1	Hồ Duy Quang	07/05/1999	hoduyquang@gmail.com

Hình 4.36. Kết quả tìm kiếm nhân viên

4.4.4 Màn hình thêm nhân viên



Hình 4.37. Giao diện thêm nhân viên

Nếu không nhập thông tin đủ, sai định dạng email thì màn hình sẽ hiển thị lỗi.

Sau khi nhập xong thông tin nhân viên mới, bắt buộc phải thêm khuôn mặt cho nhân viên với tối thiểu 5 tấm và tối đa 15 tấm ảnh, dùng dữ liệu ảnh này để huấn luyện cho mô hình nhận dạng khuôn mặt.

Nút thêm ảnh nhân viên sẽ hiển thị màu xanh và đổi thành “Đã thêm ảnh nhân viên” nếu thêm khuôn mặt hoàn tất.

Nhấn nút “Đăng ký” để đăng ký cho nhân viên mới này.

Nhấn nút “Trở về” để thoát khỏi màn hình thêm nhân viên.

Đăng ký nhân viên

Họ và tên

Ngày sinh

Email

Thêm khuôn mặt

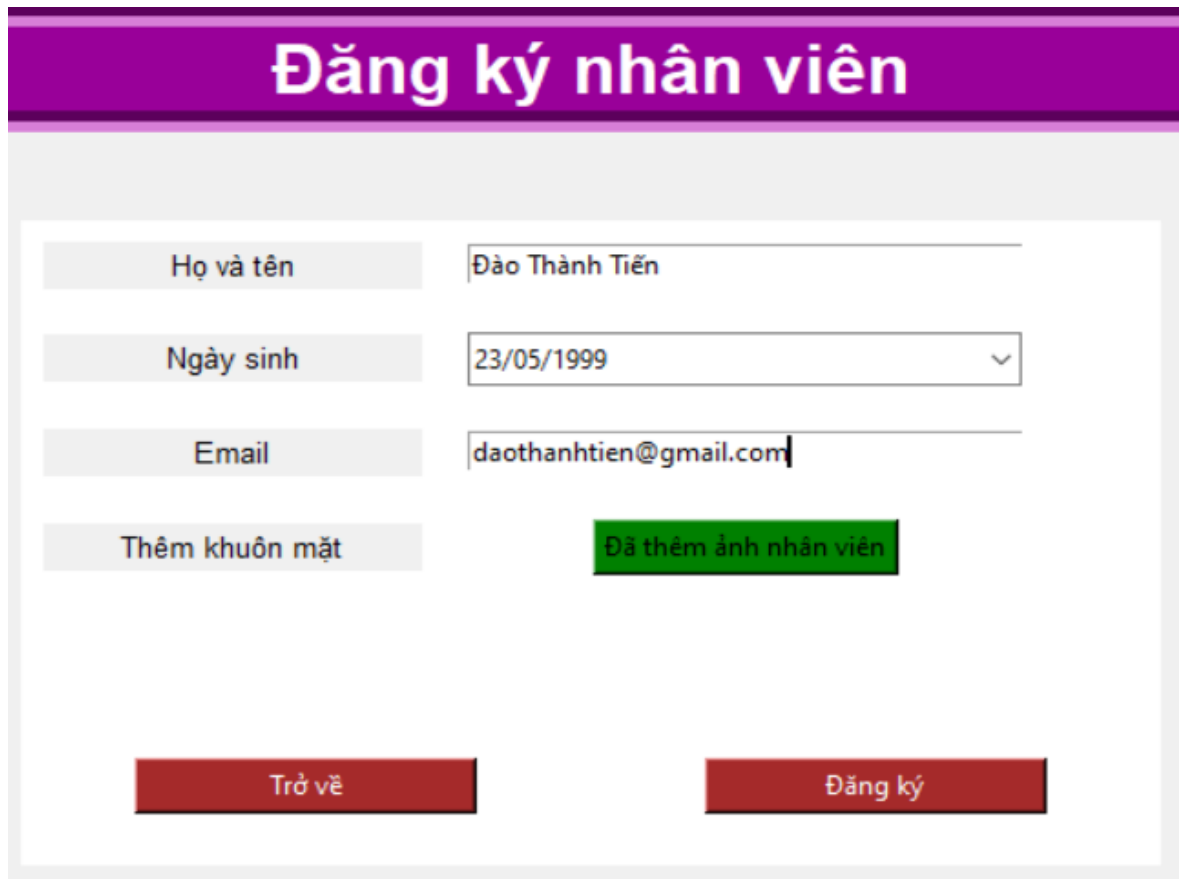
Tên không được để trống

Đã hiểu

Trở về

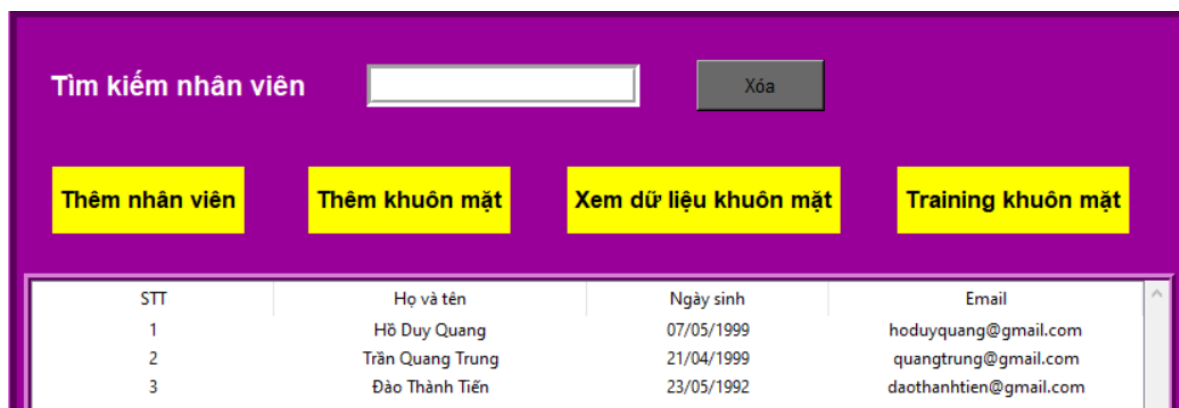
Đăng ký

Hình 4.38. Giao diện lỗi nhập dữ liệu khi thêm nhân viên



Hình 4.39. Hình ảnh đã nhập đầy đủ thông tin

Danh sách nhân viên sẽ thêm vào nhân viên mới được tạo.



STT	Họ và tên	Ngày sinh	Email
1	Hồ Duy Quang	07/05/1999	hoduyquang@gmail.com
2	Trần Quang Trung	21/04/1999	quangtrung@gmail.com
3	Đào Thành Tiến	23/05/1992	daothanhtien@gmail.com

Hình 4.40. Danh sách sau khi thêm nhân viên

Chức năng thêm khuôn mặt cho nhân viên: mục đích để thu thập dữ liệu khuôn mặt của một nhân viên, dùng dữ liệu đó để huấn luyện mô hình mạng nhân tạo nhận dạng nhân viên đó.

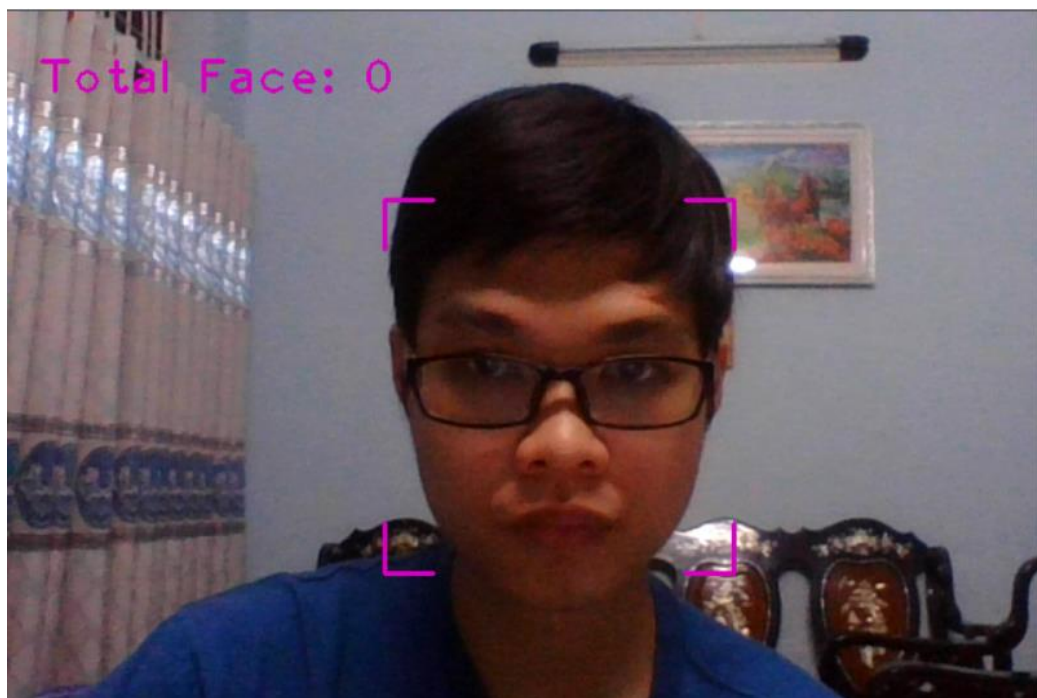
Sau khi thêm xong, quản lý có thể xem lại khuôn mặt bằng cách nhấn vào nút “Xem dữ liệu khuôn mặt”.

4.4.5 Màn hình thêm khuôn mặt

Màn hình thêm khuôn mặt dùng để thu thập dữ liệu khuôn mặt của nhân viên nhằm mục đích huấn luyện mô hình nhận dạng khuôn mặt.

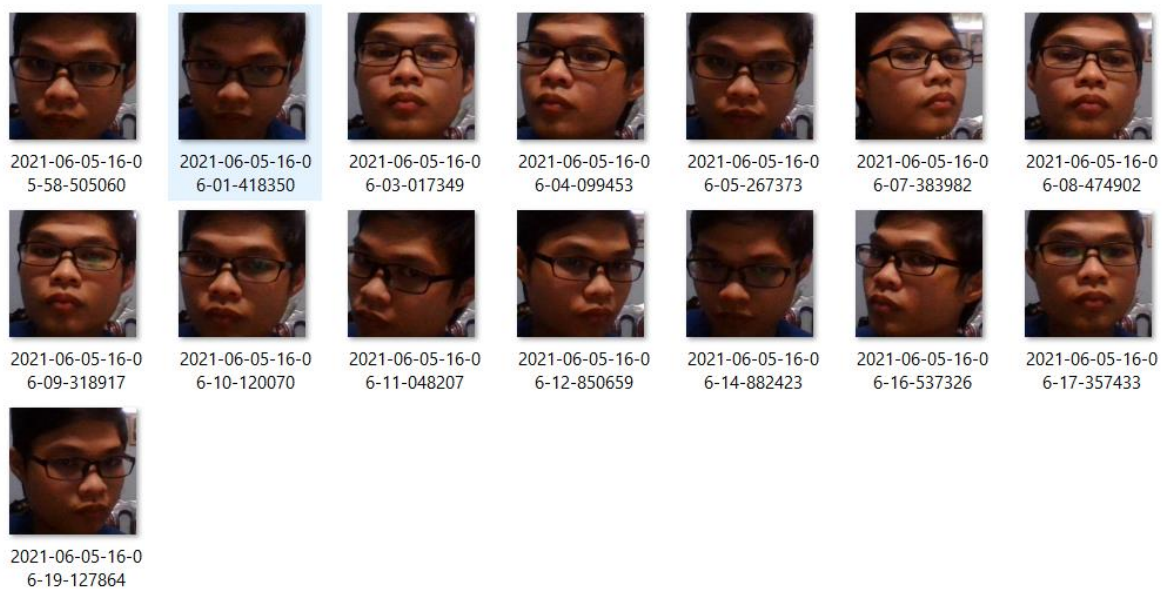
Để vào màn hình này:

- Bước 1: Nhấn chọn nhân viên trong danh sách nhân viên được hiển thị ở màn hình “Danh sách nhân viên”
- Bước 2: Nhấn phím “b” trên bàn phím để lấy dữ liệu khuôn mặt.
- Bước 3: Sau khi đã thu thập trên 100 tấm ảnh, nhấn phím “q” để thoát khỏi cửa sổ này.



Hình 4.41. Giao diện màn hình thêm khuôn mặt

4.4.6 Màn hình xem dữ liệu khuôn mặt



Hình 4.42. Giao diện xem dữ liệu khuôn mặt

4.4.7 Màn hình chi tiết nhân viên

Có thể chỉnh sửa họ tên, ngày sinh, email của nhân viên.

Chi tiết nhân viên

Ảnh gốc



Trở về

Thông tin nhân viên

Số thứ tự	1
Mã nhân viên	QuangHD
Họ và tên	Hồ Duy Quang
Ngày sinh	07/05/1999
Email	hoduyquang@gmail.com

Cập nhật

Hình 4.43. Giao diện chi tiết nhân viên

4.4.8 Màn hình chấm công

Hình 4.44. Giao diện danh sách chấm công

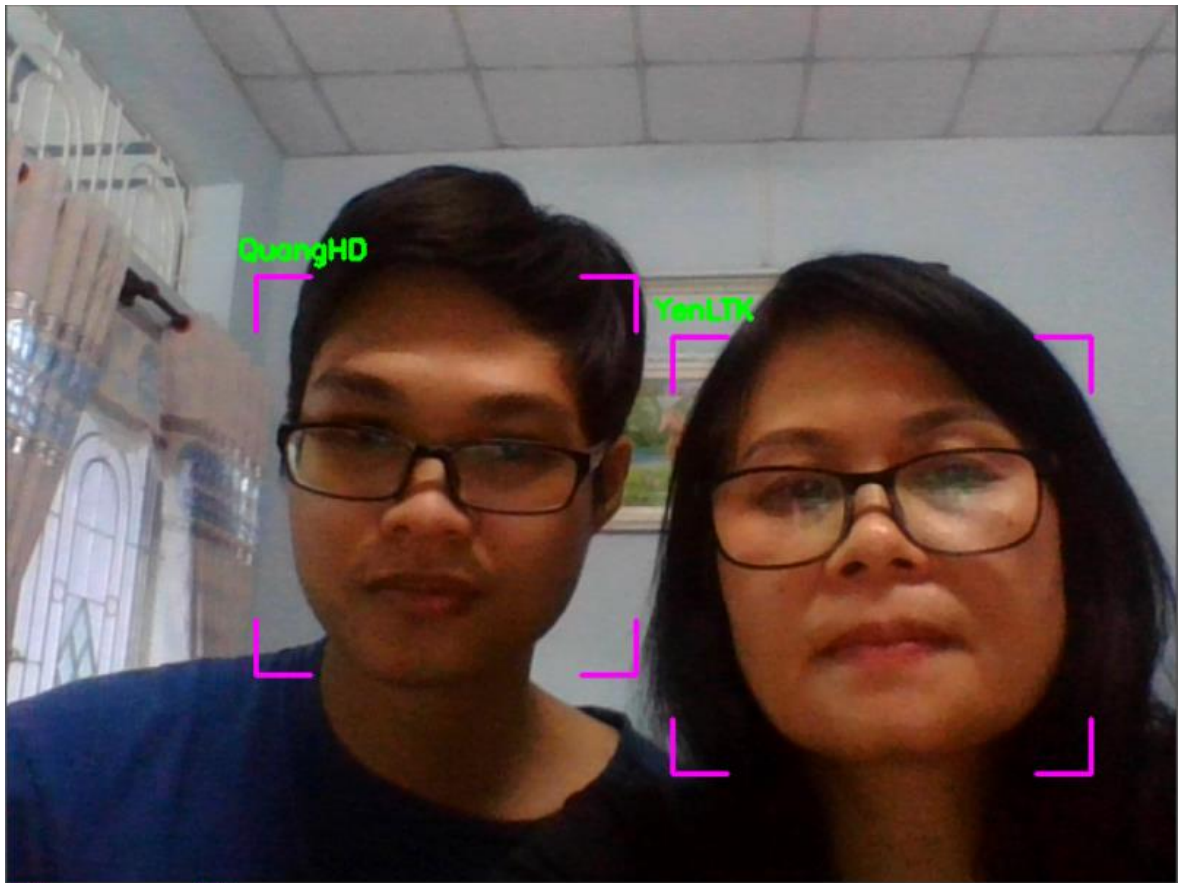
Chức năng lọc danh sách chấm công theo ngày, quản lý chọn ngày trên thanh select box, kết quả danh sách chấm công sẽ hiển thị theo ngày quản lý đã chọn.

Số thứ tự	Ngày chấm công	Mã nhân viên	Giờ checkin	Giờ checkout
1	06/06/2021	QuangHD	11:07:23	
2	06/06/2021	NhanHLH	11:23:51	

Hình 4.45. Kết quả sau khi lọc ngày chấm công

4.4.9 Màn hình check-in

Ứng dụng sẽ hiển thị user id của nhân viên mà nó nhận dạng được và lưu dữ liệu cũng như lưu hình ảnh dùng để nhận dạng. Sau khi lưu thời gian check-in xong, ứng dụng sẽ thông báo bằng cách nói “Xin chào” + {họ và tên} của nhân viên đó. Danh sách sau khi check-in sẽ được cập nhật, khi thoát khỏi màn hình check-in kết quả danh sách sẽ được hiển thị.



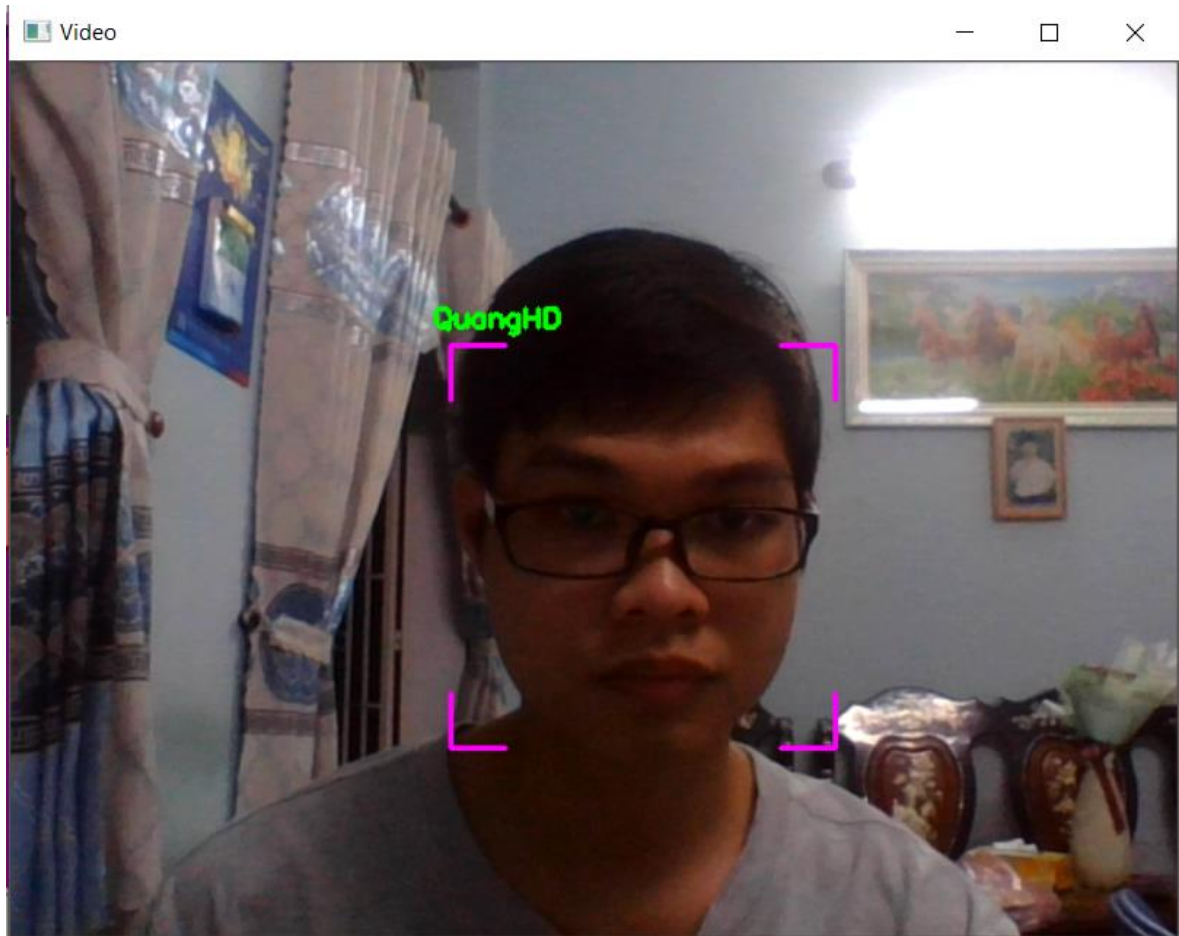
Hình 4.46. Giao diện màn hình check-in

Check-in		Check-out		
Số thứ tự	Ngày chấm công	Mã nhân viên	Giờ checkin	Giờ checkout
1	05/06/2021	QuangHD	23:07:06	

Hình 4.47. Kết quả sau khi check-in

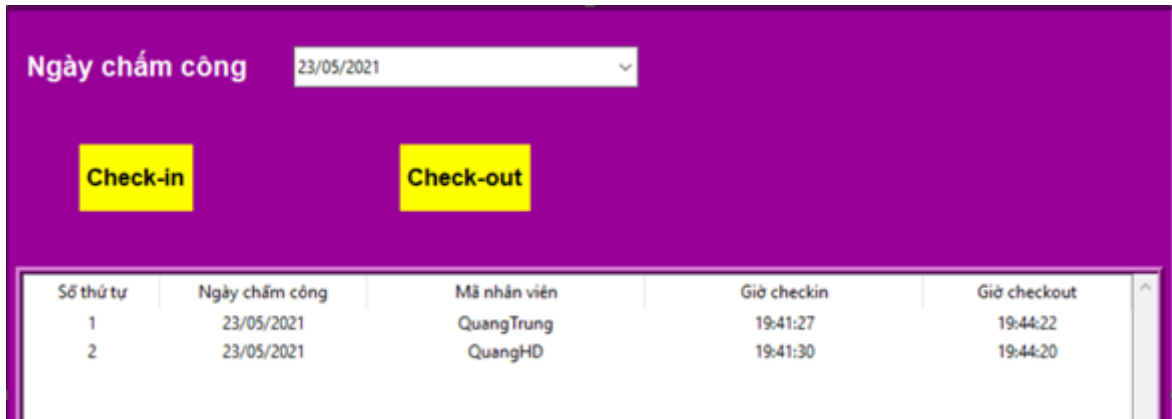
4.4.10 Màn hình check-out

Ứng dụng sẽ hiển thị user id của nhân viên mà nó nhận dạng được và lưu dữ liệu cũng như lưu hình ảnh dùng để nhận dạng. Sau khi lưu thời gian check-out xong, ứng dụng sẽ thông báo bằng cách nói “Tạm biệt” + {họ và tên} của nhân viên đó.



Hình 4.48. Giao diện màn hình check-out

Nếu phát hiện ra khuôn mặt đường viền màu hồng sẽ bao lấy khuôn mặt của nhân viên, nếu nhận dạng được danh tính của nhân viên, ứng dụng sẽ hiển thị user id của nhân viên đó, ngược lại ứng dụng sẽ hiển thị “Unknown”.



The screenshot shows a web interface with a purple header. At the top left, there is a label "Ngày chấm công" (Attendance Date) next to a date picker showing "23/05/2021". Below this, there are two yellow buttons: "Check-in" and "Check-out". Below the buttons is a table with the following data:

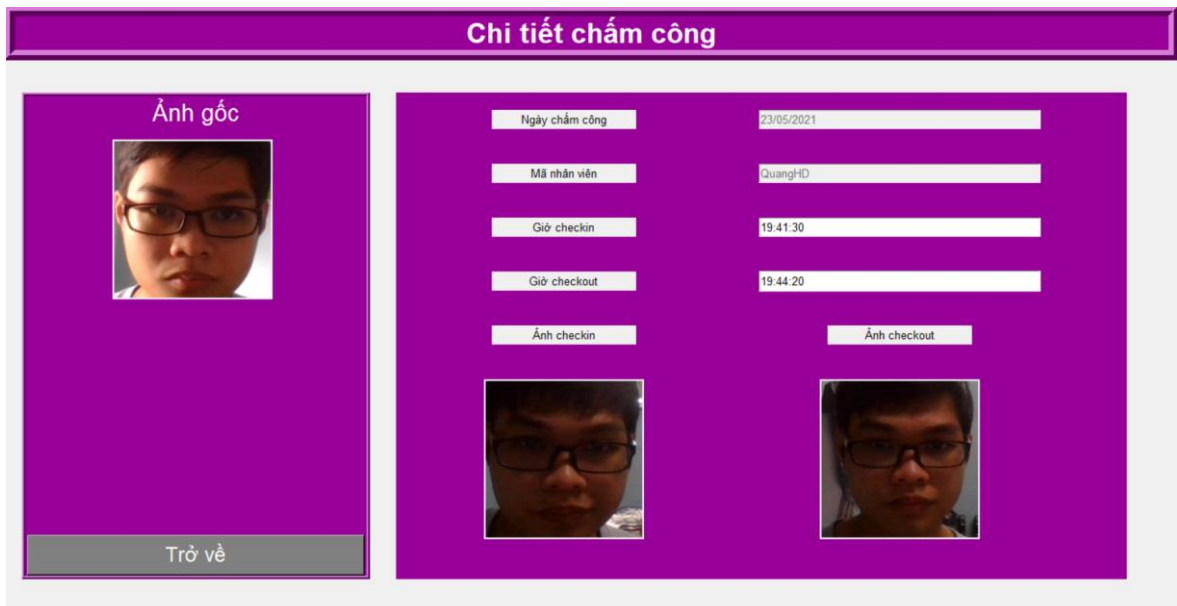
Số thứ tự	Ngày chấm công	Mã nhân viên	Giờ checkin	Giờ checkout
1	23/05/2021	QuangTrung	19:41:27	19:44:22
2	23/05/2021	QuangHD	19:41:30	19:44:20

Hình 4.49. Kết quả sau khi check-in và check-out

Danh sách chấm công sẽ hiển thị thông tin của nhân viên đã check-in hoặc check-out thành công.

4.4.11 Màn hình chi tiết chấm công

Ở đây quản lý có thể chỉnh sửa thông tin chấm công nếu có sai sót từ ứng dụng khi nhận dạng sai danh tính hoặc thời gian check-in và check-out.



The screenshot shows a web interface with a purple header titled "Chi tiết chấm công" (Attendance Details). The interface is divided into two main sections. On the left, there is a section titled "Ảnh gốc" (Original Image) showing a photo of a person with glasses. Below the photo is a button labeled "Trở về" (Back). On the right, there is a form for editing attendance details. The form contains the following fields:

- Ngày chấm công (Attendance Date): 23/05/2021
- Mã nhân viên (Employee ID): QuangHD
- Giờ checkin (Check-in Time): 19:41:30
- Giờ checkout (Check-out Time): 19:44:20
- Ảnh checkin (Check-in Image): A small photo of the same person.
- Ảnh checkout (Check-out Image): A small photo of the same person.

Hình 4.50. Giao diện chi tiết chấm công

CHƯƠNG V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

Sau khi hoàn thành xong ứng dụng chấm công khuôn mặt, nhóm chúng tôi đã đúc kết và có thêm kinh nghiệm thực tiễn về nhiều mặt.

- Hiểu được cách hoạt động của mô hình Siamese Network và hàm loss Triplet Loss.
- Xây dựng được ứng dụng thực tế.
- Xây dựng và huấn luyện được mô hình nhận dạng khuôn mặt.

5.2 Hạn chế của khóa luận

Song song đó thì trong quá trình phát triển và hoàn thiện ứng dụng vẫn còn nhiều mặt hạn chế như:

Đối với hạn chế về lưu trữ và thao tác dữ liệu trên tệp Excel thay vì sử dụng cơ sở dữ liệu SQL như những ứng dụng khác, việc sử dụng tệp Excel để tạo mới hay cập nhật dữ liệu thực tế vẫn có thể sử dụng nếu cấu trúc dữ liệu của ứng dụng không quá phức tạp, nhưng về mặt tốc độ xử lý đối với thời gian thực là một trở ngại khi check-in hoặc check-out ứng dụng phải trễ 1 đến 2 giây để xử lý dữ liệu mới. Đây là kinh nghiệm khi làm ứng dụng cần phải chú ý, nhất là đối với các quan hệ trong dữ liệu phức tạp và yêu cầu cao về thời gian thực.

Đối với mô hình phát hiện khuôn mặt, MediaPipe làm rất tốt nhiệm vụ của mình, phát hiện được khuôn mặt dù ở góc mặt nào kể cả đeo phụ kiện hay bị vật cản che như khẩu trang, mắt kính, đội nón với tốc độ rất nhanh và chính xác. MediaPipe rất tuyệt vì nó hỗ trợ đa nền tảng dù nó là giải pháp cho thiết bị di động, nhưng nó vẫn hỗ trợ Windows, Mac và cả Ubuntu. Nhưng MediaPipe vẫn có hạn chế nhất định của nó khi sự tập trung hỗ trợ cho thiết bị di động, khoảng cách phát hiện khuôn mặt của MediaPipe có giới hạn khi nhóm chúng tôi thử nghiệm với nhiều khoảng cách khác nhau, mô hình khó có thể phát hiện được khuôn mặt nếu cách xa

hơn 1 giây. Và việc phát hiện khuôn mặt vẫn chưa thể xử lý việc chống giả mạo khuôn mặt để chấm công hộ.

Đối với mô hình nhận dạng khuôn mặt, FaceNet thực sự là một công nghệ nhận dạng khuôn mặt rất tốt, với xác suất nhận dạng trên 95%, nhưng độ chính xác sẽ bị ảnh hưởng nếu các khuôn mặt trong cùng một gia đình, thật bất ngờ khi chúng tôi thử nghiệm nhận dạng cùng gia đình mình kết quả hiển thị thay đổi liên tục tên của các thành viên trong gia đình.

5.3 Hướng phát triển

Dựa trên những cơ sở sẵn có trong tương lai, ứng dụng có thể được cải thiện:

Phát triển hệ thống chống giả mạo khuôn mặt cho hệ thống nhận diện.

Về độ chính xác khi nhận dạng khuôn mặt nhân viên trong ứng dụng, chúng ta cần cải thiện bộ dữ liệu bằng cách thêm các góc chụp, hạn chế các thay đổi trên khuôn mặt, điều kiện ánh sáng,... Hiện tại, ứng dụng chưa thể nhận dạng khuôn mặt chính xác khi đeo khẩu trang, mong muốn phát triển tiếp theo của nhóm sẽ có thể cải thiện ứng dụng nhận dạng được khuôn mặt đeo khẩu trang để có thể phù hợp hơn với tình hình đại dịch Covid hiện nay phải luôn đeo khẩu trang.

Thử nghiệm thêm các mô hình pre-trained và thuật toán huấn luyện khác nhau cho bộ dữ liệu của ứng dụng. Bằng cách này chúng ta có thể tìm ra được mô hình nhận dạng tốt hơn có ứng dụng.

Để cải thiện hơn về tốc độ chấm công, việc thay đổi cách thức lưu trữ dữ liệu là một ưu tiên hàng đầu. Ngoài ra, chúng ta có thể cấu trúc lại mô hình dữ liệu để thuận tiện trong việc phát triển chức năng tính tiền lương cho nhân viên.

Không chỉ dừng lại ở nhiệm vụ chấm công, ứng dụng còn có thể cải tiến thêm chức năng giám sát, quản lý ra vào cho nhân viên từng phòng ban.

TÀI LIỆU THAM KHẢO

- [1]. Bhaskar Chaudhary (2015), *Tkinter GUI Application Development Blueprints*.
- [2]. Facial recognition system – Wikipedia (Đã cập nhật lần cuối: May 2021), https://en.wikipedia.org/wiki/Facial_recognition_system
- [3]. Florian Schroff, Dmitry Kalenichenko, James Philbin (Jun 2015), *FaceNet: A Unified Embedding for Face Recognition and Clustering*.
- [4]. Time and attendance – Wikipedia (Đã cập nhật lần cuối: March 2021), https://en.wikipedia.org/wiki/Time_and_attendance
- [5]. Tkinter-GUI-Application-Development-Cookbook (March 2018), <https://github.com/PacktPublishing/Tkinter-GUI-Application-Development-Cookbook>
- [6]. Face Detection – MediaPipe (2020), https://google.github.io/MediaPipe/solutions/face_detection.html
- [7]. CS231n Convolutional Neural Networks for Visual Recognition, <https://cs231n.github.io/convolutional-networks>