

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Ứng dụng phát hiện người và hành động lấy đồ trong
cửa hàng tiện lợi

HOÀNG THÀNH LÂM

lam.ht176798@sis.hust.edu.vn

Ngành Công nghệ thông tin

Giảng viên hướng dẫn: PGS. TS. Đặng Văn Chuyết

Chữ kí GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2022

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin sinh viên

1. Họ và tên: Hoàng Thành Lâm
2. Điện thoại liên lạc: 0704127556
3. Email: lam.ht176798@sis.hust.edu.vn
4. Lớp: Application Specialist-K62
5. Đồ án tốt nghiệp thực hiện tại: Trường Công nghệ Thông tin và Truyền thông.

2. Mục đích của đồ án tốt nghiệp

Tìm hiểu các phương pháp phổ biến dùng cho bài toán phát hiện, theo dõi vật thể và bài toán nhận diện mặt người. Xây dựng và áp dụng những mô hình phù hợp cho bài toán "Phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi".

3. Nhiệm vụ cụ thể của đồ án tốt nghiệp

- Tìm hiểu về các bài toán phát hiện đối tượng
- Tìm hiểu về phương pháp theo dõi đối tượng.
- Phát triển phương pháp phát hiện hành động lấy đồ tại cửa hàng tiện lợi.
- Tìm hiểu bài toán nhận diện và phân biệt khách hàng thông qua khuôn mặt.

4. Cam kết của sinh viên

Tôi - Hoàng Thành Lâm - cam kết đồ án tốt nghiệp là công trình nghiên cứu của bản thân, thực hiện dưới sự hướng dẫn của PGS. TS. Đặng Văn Chuyết. Các kết quả trình bày trong đồ án tốt nghiệp là trung thực, không sao chép bất trì công trình nghiên cứu nào khác.

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của đồ án tốt nghiệp và ý kiến cho phép bảo vệ:

Hà Nội, tháng 8 năm 2022

Giáo viên hướng dẫn

Tác giả

Đặng Văn Chuyết

Hoàng Thành Lâm

LỜI CẢM ƠN

Lời đầu tiên, tôi xin được gửi lời cảm ơn sâu sắc nhất đến PSG. TS. Đặng Văn Chuyết và TS. Đặng Tuấn Linh, hai thầy đã luôn nhiệt tình quan tâm, giúp đỡ và hỗ trợ tôi trong suốt quá trình thực hiện đồ án tốt nghiệp. Các thầy đã giúp đỡ tôi giải quyết các vấn đề khó khăn về mặt ý tưởng cũng như công nghệ và hướng phát triển. Nhờ đó, tôi đã có thể hoàn thành đồ án tốt nghiệp đúng với mong đợi của bản thân mình.

Tiếp theo, tôi xin gửi lời cảm ơn chân thành đến Đại học Bách Khoa Hà Nội nói chung và Trường Công nghệ Thông tin và Truyền thông nói riêng vì đã luôn tạo điều kiện học tập, nghiên cứu phù hợp nhất để chúng tôi phát triển bản thân. Đồng thời, nhà trường đã luôn tận tình dạy bảo, truyền đạt nhiều kiến thức, nguồn cảm hứng và kinh nghiệm bổ ích trong học tập, cuộc sống và công việc.

Cuối cùng, tôi xin gửi lời cảm ơn đến những người bạn trong nhóm của mình: Trần Xuân Đức, Cao Quốc Khánh, Lương Đào Quang Anh và Lê Hoàng đã đồng hành cùng tôi suốt hai năm qua, cùng nhau nghiên cứu, làm việc để hoàn thành sản phẩm của chúng tôi. Đặc biệt là lời cảm ơn tới gia đình vì đã luôn tin tưởng, tạo nguồn động lực lớn lao thúc đẩy tôi vững bước tiến lên, tiếp tục thử thách bản thân và làm những điều ý nghĩa.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong ngành kinh doanh bán lẻ, ứng dụng của ngành công nghệ thông tin đang trở nên đa dạng và phổ biến hơn. Một trong số đó là việc áp dụng camera giám sát để hỗ trợ người quản lý, nhân viên cửa hàng theo dõi sản phẩm cũng như cửa hàng. Nhờ đó, người quản lý, nhân viên cửa hàng dễ dàng kiểm soát sản phẩm, phòng chống trộm cắp. Các hệ thống camera giám sát hiện nay không hoạt động hoàn toàn độc lập, luôn cần có sự điều khiển và quan sát của con người. Trong khi đó, các cửa hàng tiện lợi vừa và nhỏ mặc dù có khả năng lắp đặt hệ thống camera giám sát nhưng không có đủ kinh phí để duy trì việc theo dõi liên tục. Vì thế, trong đồ án này, tôi sẽ phát triển một bài toán giúp phát hiện người và hành động lấy đồ trong cửa hàng nhằm mục đích hỗ trợ người quản lý, nhân viên cửa hàng theo dõi, quản lý sản phẩm cũng như nhận cảnh báo khi phát hiện hành vi lấy trộm đồ trong cửa hàng tiện lợi. Mục tiêu của tôi là tạo ra một giải pháp và xây dựng một bài toán đảm bảo độ chính xác và đặc biệt có thể triển khai trên nền tảng hệ thống camera giám sát có sẵn hiện nay.

Để tiến hành thực nghiệm, bài toán sẽ được triển khai trên hệ thống camera thông minh mà chúng tôi đã xây dựng. Hệ thống này có khả năng tích hợp nhiều bài toán thị giác máy tính khác nhau và được triển khai trên nhiều nền tảng như ứng dụng web, ứng dụng dành cho máy tính, ứng dụng dành cho thiết bị di động.

Bài toán về cơ bản đã giải quyết được các vấn đề đã đặt ra. Mô hình phát hiện và nhận diện khách hàng đã đạt độ chính xác 82%, mô hình phát hiện hành động lấy đồ nhận diện chính xác 17 lần trong 20 lần thử nghiệm trong điều kiện không bị che khuất. Ứng dụng cung cấp tính năng theo dõi các dòng camera trực tiếp, gửi thông báo cho người dùng trong thời gian thực về cảnh báo trộm cắp và hết đồ trên kệ hàng. Với những kết quả đạt được như vậy, ứng dụng cho thấy được tiềm năng của mình trong việc giám sát và đảm bảo an toàn cho cửa hàng tiện lợi.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	5
2.1 Khảo sát hiện trạng	5
2.2 Tổng quan chức năng	6
2.2.1 Biểu đồ use case tổng quát	6
2.2.2 Biểu đồ use case phân rã Đăng ký, Đăng nhập	7
2.2.3 Biểu đồ use case phân rã Store manager	8
2.2.4 Quy trình nghiệp vụ	9
2.3 Đặc tả chức năng	11
2.3.1 Đặc tả use case Đăng nhập.....	11
2.3.2 Đặc tả use case Chọn xem dòng camera	11
2.3.3 Đặc tả use case Thêm vùng theo dõi sản phẩm.....	12
2.3.4 Đặc tả use case Xóa vùng theo dõi sản phẩm	12
2.3.5 Đặc tả use case Điều chỉnh vùng nhận diện khách hàng	13
2.4 Yêu cầu phi chức năng	13
2.4.1 Yêu cầu về kỹ thuật	13
2.4.2 Yêu cầu về giao diện.....	13
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	14
3.1 Công nghệ phát triển Mô-đun "Store manager".....	15
3.1.1 YOLOv5	15

3.1.2 CSRT	16
3.1.3 MTCNN.....	16
3.1.4 Facenet	17
3.1.5 Phát hiện chuyển động.....	17
3.1.6 Docker	18
3.1.7 Tensorflow	18
3.1.8 Torch, Pytorch	19
3.1.9 FastAPI.....	19
3.1.10 OpenCV	19
3.1.11 Numpy	20
3.1.12 Multithread.....	20
3.1.13 Firebase Cloud Messaging	21
3.2 Công nghệ phát triển ứng dụng trên thiết bị Andoird.....	21
3.2.1 Java	21
3.2.2 HttpURLConnection	22
3.2.3 LibVLC	22
CHƯƠNG 4. THIẾT KẾ ỨNG DỤNG	23
4.1 Thiết kế kiến trúc.....	23
4.1.1 Lựa chọn kiến trúc phần mềm	23
4.1.2 Thiết kế luồng hoạt động	24
4.2 Thiết kế chi tiết.....	28
4.2.1 Thiết kế giao diện	28
4.3 Phát triển bài toán "Phát hiện người và hành động lấy đồ"	32
4.3.1 Bài toán phát hiện sản phẩm.....	32
4.3.2 Bài toán theo dõi sản phẩm và phát hiện hành động lấy đồ	34
4.3.3 Bài toán nhận người qua dữ liệu khuôn mặt	35

CHƯƠNG 5. THỰC NGHIỆM VÀ ĐÁNG GIÁ	36
5.1 Xây dựng ứng dụng.....	36
5.1.1 Thư viện và công cụ sử dụng.....	36
5.1.2 Kết quả đạt được	36
5.1.3 Minh họa các chức năng chính	37
5.2 Kiểm thử.....	39
5.2.1 Kiểm thử chức năng đăng nhập	39
5.2.2 Kiểm thử chức năng xem dòng camera đã qua xử lý	40
5.3 Triển khai	41
5.3.1 Chuẩn bị	41
5.3.2 Thực hiện thử nghiệm.....	43
CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	49
6.1 Mô hình phát hiện sản phẩm trong cửa hàng tiện lợi.....	49
6.1.1 Đặt vấn đề	49
6.1.2 Giải pháp	49
6.1.3 Kết quả đạt được	50
6.2 Xây dựng bộ dữ liệu huấn luyện mô hình phát hiện sản phẩm	50
6.2.1 Đặt vấn đề	50
6.2.2 Giải pháp	50
6.2.3 Kết quả đạt được	51
6.3 Xây dựng mô hình nhận diện khác hàng thông qua khuôn mặt	51
6.3.1 Đặt vấn đề	51
6.3.2 Giải pháp	51
6.3.3 Kết quả đạt được	52
6.4 Xây dựng bài toán phát hiện hành động lấy đồ.....	52
6.4.1 Đặt vấn đề	52

6.4.2 Giải pháp	52
6.4.3 Kết quả đạt được	53
CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	54
7.1 Kết luận	54
7.2 Hướng phát triển.....	55
TÀI LIỆU THAM KHẢO.....	58

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ Use Case tổng quan	6
Hình 2.2	Biểu đồ Use Case Đăng ký, Đăng nhập	7
Hình 2.3	Biểu đồ Use Case Store manager	8
Hình 2.4	Biểu đồ hoạt động quy trình sử dụng chức năng về hình ảnh . .	9
Hình 2.5	Biểu đồ hoạt động quy trình sử dụng chức năng về camera . .	10
Hình 3.1	Kiến trúc tổng quan hệ thống camera thông minh	14
Hình 3.2	Các phần đảm nhiệm trong hệ thống	15
Hình 4.1	Kiến trúc tổng quan hệ thống camera thông minh	23
Hình 4.2	Luồng đăng nhập	24
Hình 4.3	Luồng truy cập chức năng "Store manager"	25
Hình 4.4	Luồng xem dòng camera trong chức năng "Store manager" . .	26
Hình 4.5	Luồng tương tác với dòng camera trong chức năng "Store manager"	27
Hình 4.6	Màn hình chính khi chưa đăng nhập	28
Hình 4.7	Màn hình chính khi đăng nhập	28
Hình 4.8	Màn hình đăng nhập	29
Hình 4.9	Màn hình đăng ký	29
Hình 4.10	Màn hình chọn các dịch vụ về hình ảnh khi chưa đăng nhập . .	30
Hình 4.11	Màn hình chọn các dịch vụ về hình ảnh khi đăng nhập	30
Hình 4.12	Màn hình chọn các dịch vụ về camera khi chưa đăng nhập . .	31
Hình 4.13	Màn hình chọn các dịch vụ về camera khi đăng nhập	31
Hình 4.14	Màn hình chức năng "Store manager"	32
Hình 4.15	So sánh tốc độ của các mô hình YOLO	33
Hình 4.16	Bộ dữ liệu tự thu thập	33
Hình 4.17	Giá trị hàm mất mát của mô hình xác định vật thể	34
Hình 4.18	Giá trị mAP của mô hình xác định vật thể	34
Hình 5.1	Giao diện chức năng đăng nhập	37
Hình 5.2	Giao diện menu sau khi đăng nhập	37
Hình 5.3	Giao diện chức năng "Store manager"	38
Hình 5.4	Đăng nhập	43
Hình 5.5	Chọn chức năng "Store manager"	43

Hình 5.6 Chọn camera để quan sát	44

Hình 5.7 Khách hàng thứ nhất bước vào cửa hàng	44
Hình 5.8 Khách hàng thứ hai bước vào cửa hàng	44
Hình 5.9 Khách hàng thứ ba bước vào cửa hàng	44
Hình 5.10 Chọn camera để quan sát	45
Hình 5.11 Chưa thêm vùng theo dõi sản phẩm	45
Hình 5.12 Đã thêm vùng theo dõi sản phẩm	45
Hình 5.13 Nhận thông báo vùng theo dõi trống	45
Hình 5.14 Chọn camera để quan sát	46

Hình 5.15 Khách hàng lấy sản phẩm khỏi kệ	46
Hình 5.16 Khách hàng lấy sản phẩm khỏi kệ	46
Hình 5.17 Chọn camera để quan sát	47
Hình 5.18 Khách hàng khi chưa thanh toán	47
Hình 5.19 Khách hàng sau khi đứng tại quầy thu ngân để thanh toán	47
Hình 5.20 Chọn camera để quan sát	48
Hình 5.21 Khách hàng ra khỏi cửa hàng khi đã thanh toán	48
Hình 5.22 Khách hàng ra khỏi cửa hàng khi chưa thanh toán	48

Hình 5.23 Nhận thông báo có khách hàng chưa thanh toán rời khỏi cửa hàng	48

Hình 6.1 So sánh tốc độ của các mô hình YOLO	50
Hình 6.2 Bộ dữ liệu tự thu thập	51

DANH MỤC BẢNG BIỂU

Bảng 2.1	Use case Đăng nhập	11
Bảng 2.2	Use case Chọn xem dòng camera	11
Bảng 2.3	Use case Thêm vùng theo dõi sản phẩm	12
Bảng 2.4	Use case Xóa vùng theo dõi sản phẩm	12
Bảng 2.5	Use case Điều chỉnh vùng nhận diện khách hàng	13
Bảng 5.1	Danh sách thư viện và công cụ sử dụng	36
Bảng 5.2	Danh sách kết quả đạt được	37
Bảng 5.3	Thông kê thông tin sản phẩm	37
Bảng 5.4	Kiểm thử chức năng đăng nhập	39
Bảng 5.5	Kiểm thử chức năng xem dòng camera đã qua xử lý	40

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
Camera	Thiết bị ghi hình ảnh
CNN	Mạng thần kinh tích chập (Convolutional neural network)
Container	Môi trường thực thi
CPU	Bộ xử lý trung tâm (Central Processing Unit)
Crash	Dừng đột ngột do sự cố hệ thống
Framework	Các đoạn code viết sẵn và được cấu tạo thành một bộ khung
GPU	Bộ xử lý đồ họa (Graphics Processing Unit)
Real-time	Thời gian thực

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong những năm gần đây, ngành công nghệ thông tin đang phát triển rất mạnh mẽ [1], có thể nói việc áp dụng công nghệ vào mọi mặt đời sống đang là xu hướng hiện nay. Đặc biệt, với sự phát triển về khả năng xử lý và giá thành rẻ hơn của các phần cứng, việc áp dụng thành tựu của công nghệ, cụ thể là thành tựu của ngành trí tuệ nhân tạo đã và đang tạo ra nhiều giải pháp hỗ trợ công việc của con người [2]. Trong đó, lĩnh vực thị giác máy tính đang là một trong số những lĩnh vực phát triển và có nhiều ứng dụng nhất hiện nay [3]. Lĩnh vực thị giác máy tính là một nhánh của ngành trí tuệ nhân tạo, chuyên nghiên cứu và tạo ra những thuật toán xử lý, phân tích hình ảnh để đưa ra các thông tin quan trọng.

Hiện nay, việc lắp camera theo dõi cho cửa hàng tiện lợi ngày càng phổ biến [4]. Nhờ đó, nhân viên hay người quản lý cửa hàng có thể dễ dàng theo dõi toàn bộ cửa hàng để phát hiện các sự cố. Tuy nhiên, hiện nay tác dụng chính của camera chỉ dùng trong trường hợp cần trích xuất hình ảnh sau khi có người phát hiện sự cố đã xảy ra trong một thời gian, lý do là không có người theo dõi hình ảnh qua camera liên tục. Vì thế nên khi có sự cố xảy ra cửa hàng không thể phát hiện và khắc phục ngay lập tức được. Đặc biệt là vấn đề trộm cắp trong cửa hàng hay khi các gian hàng đã bán hết sản phẩm. Nếu việc phát hiện và khắc phục các sự cố trên không kịp thời thì sẽ dẫn tới thiệt hại lớn đến cho cửa hàng.

Chính vì thế, để giải quyết vấn đề này, tôi đã chọn ra một số sản phẩm thông dụng trong cửa hàng làm sản phẩm thử nghiệm và lựa chọn đề tài: "**Ứng dụng phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi**" để tiếp cận bài toán và xây dựng một ứng dụng giúp cảnh báo sớm những sự cố xảy ra cho cửa hàng.

Để tiến hành thực nghiệm, bài toán được triển khai trên hệ thống camera thông minh mà chúng tôi đã xây dựng. Hệ thống này có khả năng tích hợp nhiều bài toán thị giác máy tính khác nhau và được triển khai trên nhiều nền tảng như ứng dụng web, ứng dụng dành cho máy tính, ứng dụng dành cho thiết bị di động. Về tổng quan, hệ thống mà chúng tôi xây dựng gồm ba phần chính. Phần thứ nhất sử dụng Raspberry Pi như một thiết bị biên, những thiết bị này tích hợp camera, vừa có thể ghi hình, vừa có thể xử lý các bài toán nhẹ như tăng sáng, khử sương mù trước khi truyền dòng video về máy chủ chính. Phần thứ hai là máy chủ chính. Chúng tôi sử dụng một máy tính như một máy chủ, tại đây hệ thống sẽ nhận các dòng video từ thiết bị biên và thực hiện các bài toán khác nhau đối với các dịch vụ chuyên biệt

được tích hợp sẵn. Phần thứ ba là các ứng dụng phần mềm để tương tác với người dùng. Có ba ứng dụng được phát triển, đó là ứng dụng web, ứng dụng dành cho máy tính và ứng dụng dành cho thiết bị di động. Ứng dụng sẽ cung cấp các dịch vụ được tích hợp trên máy chủ chính cho người dùng.

1.2 Mục tiêu và phạm vi đề tài

Hiện tại trên thị trường có nhiều nhà cung cấp dịch vụ camera giám sát với chi phí rẻ kèm theo một số dịch vụ nhất định [5] như truy cập hình ảnh camera từ xa hay lưu trữ hình ảnh để truy xuất sau. Mặt khác, các cửa hàng tiện lợi hiện nay cũng đã bắt đầu lắp đặt nhiều hơn những camera giám sát để tiện lợi trong việc quản lý cũng như theo dõi cửa hàng.

Tuy nhiên các cửa hàng tiện lợi vừa và nhỏ thường không có nhân viên theo dõi camera liên tục nên có thể bỏ qua một số hành vi trộm cắp hay việc hết sản phẩm trên các gian hàng.

Nhận thấy những điểm hạn chế, ứng dụng này sẽ được xây dựng để khắc phục những hạn chế đã nêu thông qua những chức năng sau:

- Phát hiện và theo dõi sản phẩm trên kệ hàng, thông báo khi kệ hàng rỗng.
- Nhận diện khách hàng trong cửa hàng và hành động lấy sản phẩm khỏi kệ hàng, thông báo khi khách hàng rời khỏi cửa hàng khi chưa thanh toán.

Để xây dựng bài toán với những chức năng trên, tôi đặt ra bốn mục tiêu chính.

Mục tiêu thứ nhất là chuẩn bị bộ dữ liệu, tìm mô hình phù hợp để phát hiện và theo dõi những sản phẩm trên kệ hàng. Tôi mong muốn giải pháp phải có độ chính xác cao đồng thời có độ trễ không quá lớn.

Mục tiêu thứ hai là nghiên cứu và tìm hiểu giải pháp nhận diện, phân biệt khách hàng thông qua hình ảnh. Vì bài toán áp dụng cho cửa hàng tiện lợi vừa và nhỏ nên tôi mong muốn giải pháp có thể nhận diện và phân biệt cùng lúc 20 người với độ chính xác cao.

Mục tiêu thứ ba là xây dựng thuật toán phát hiện hành vi lấy đồ dựa trên việc theo dõi sản phẩm. Từ đó xác định khách hàng đã lấy sản phẩm thông qua giải pháp nhận diện khách hàng đã nêu.

Mục tiêu thứ tư là dựa vào những công nghệ đã nghiên cứu, xây dựng một chương trình tích hợp các giải pháp với nhau để tạo thành một ứng dụng hoàn chỉnh, sau đó đánh giá và kiểm thử chính ứng dụng này.

Ngoài ra, để tiến hành thực nghiệm, tôi cần triển khai bài toán lên hệ thống camera thông minh mà chúng tôi đã phát triển. Cụ thể là tích hợp bài toán vào máy

chủ chính và thêm chức năng trên ứng dụng Android liên kết với bài toán đó.

Vì lý do bản quyền hình ảnh nên việc trích xuất hình ảnh camera hay quay phim tại các cửa hàng tiện lợi gặp khó khăn. Do đó, trong đồ án này, tôi sẽ dùng những hình ảnh mô phỏng lại cấu trúc của cửa hàng tiện lợi và gian hàng với một số sản phẩm nhất định.

1.3 Định hướng giải pháp

Tất cả vấn đề này đều được cân nhắc kỹ lưỡng sao cho đạt được độ chính xác cao và độ trễ thấp nhất. Do đó tôi đưa ra những giải pháp sau.

Đối với **mục tiêu thứ nhất**, tôi lựa chọn mô hình YOLOv5 [6], đây là mô hình mới nhất trong những phiên bản YOLO [7], sẽ được giải thích ở chương 4. Mô hình này có nhiều cải tiến so với các phiên bản cũ, mang lại độ chính xác tương đối tốt và cũng rất nhẹ. Đối với vấn đề theo dõi vị trí sản phẩm, tôi lựa chọn thuật toán CSRT [8], sẽ được giải thích ở chương 4. Thuật toán này mang lại độ chính xác cao để giúp tính toán đưa ra kết quả của hành động lấy đồ một cách chính xác.

Đối với **mục tiêu thứ hai**, tôi dùng mạng MTCNN [9] để phát hiện khuôn mặt của khách hàng và đưa qua mô hình Facenet [10] để phân biệt, chi tiết sẽ được giải thích ở chương 4.

Đối với **mục tiêu thứ ba**, vì gian hàng là cố định nên tôi sẽ dựa vào việc tính toán sự thay đổi vị trí của sản phẩm để đưa ra kết quả, chi tiết sẽ được giải thích ở chương 4.

Đối với **mục tiêu thứ tư**, tôi áp dụng công nghệ xử lý đa luồng để tiến hành chạy song song nhiều dòng camera khác nhau, kết hợp các kết quả nhận được từ các luồng để đưa ra kết quả của bài toán.

Ngoài ra, trong bài toán phát triển ứng dụng Android, tôi lựa chọn xây dựng ứng dụng Android trên nền tảng Java [11]. Đây là ngôn ngữ lâu đời, dễ tiếp cận và tích hợp nhiều thư viện tốt.

Bằng cách tích hợp các giải pháp đã nêu cùng với công nghệ sẽ được nêu trong quyển đồ án này, tôi sẽ thiết kế và tạo ra ứng dụng phát hiện, theo dõi sản phẩm, phát hiện hành vi lấy đồ, cảnh báo hành vi trộm cắp và thông báo hết hàng cho cửa hàng tiện lợi.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Ở chương 2 tôi sẽ khảo sát hiện trạng, phân tích và thiết kế ứng dụng sao cho phù hợp với bài toán.

Chương 3, là chương tôi trình bày về các công nghệ sử dụng, lý do sử dụng cũng như tôi đã sử dụng nó ở đâu trong ứng dụng.

Tiếp theo, tại chương 4 tôi sẽ trình bày sâu hơn về thiết kế, xây dựng ứng dụng và tiến hành đánh giá, kiểm thử ứng dụng.

Ở chương 5, tôi sẽ trình bày quá trình phát triển ứng dụng cùng với những giải pháp và đóng góp nổi bật, cũng như những khó khăn gặp phải.

Cuối cùng, tôi sẽ đưa ra kết luận cũng như hướng phát triển tại chương 6.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

Hiện nay trên thị trường có nhiều phương pháp chống trộm cắp phổ biến tại các cửa hàng nói chung [12], đó là sử dụng hệ thống cổng từ an ninh kết hợp với tem từ gắn trên sản phẩm, hệ thống camera giám sát và gương cầu lồi thường dùng để quan sát. Mỗi phương pháp đều có những ưu và nhược điểm nhất định. Đối với hệ thống cổng từ an ninh thì độ chính xác của phương pháp này cao, dễ dàng phát hiện sản phẩm được mang ra khỏi cửa hàng trái phép. Tuy nhiên, chi phí để triển khai hệ thống này khá cao và chỉ áp dụng với một số mặt hàng nhất định nên không phù hợp để triển khai trong cửa hàng tiện lợi vừa và nhỏ. Với với hệ thống camera giám sát thông thường và gương cầu lồi, phương pháp này có chi phí triển khai thấp và dễ dàng lắp đặt cũng như sử dụng, nhưng để hệ thống hoạt động hiệu quả thì luôn phải có người theo dõi thường xuyên.

Nhận thấy rằng tại khá nhiều cửa hàng tiện lợi vừa và nhỏ hiện nay đều được lắp đặt camera quan sát quanh cửa hàng, việc tích hợp một hệ thống có khả năng theo dõi sản phẩm, phát hiện người lấy đồ và cảnh báo khi có hành vi trộm cắp là một việc hoàn toàn khả thi, phù hợp và tiết kiệm chi phí xét trên phương diện là một giải pháp hỗ trợ các hệ thống chính. Ứng dụng của tôi không phải là một ứng dụng thay thế hoàn toàn các phương pháp đảm bảo an toàn truyền thống mà là hỗ trợ trên những giải pháp có sẵn, hỗ trợ nhân viên giám sát cửa hàng một cách tự động.

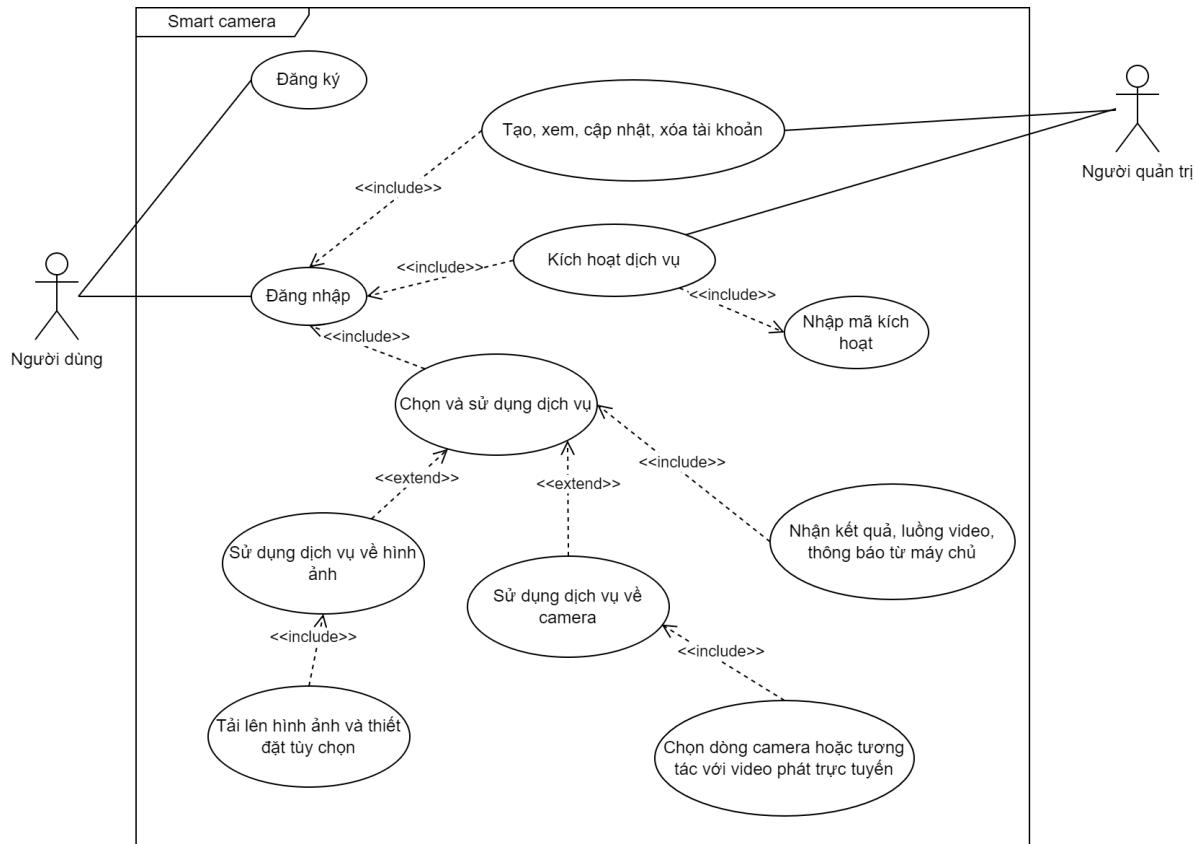
Để bài toán trở nên hiệu quả trong việc đảm bảo an toàn cho cửa hàng tiện lợi, ứng dụng mà tôi phát triển cần có các tính năng chính sau:

- Lựa chọn các camera riêng lẻ hoặc camera tổng lắp trong cửa hàng để quan sát.
- Điều chỉnh vùng nhận diện, theo dõi khách hàng trên camera.
- Thêm và xóa vùng theo dõi sản phẩm trên camera.
- Nhận thông báo khi có khách hàng rời khỏi cửa hàng khi chưa thanh toán.
- Nhận thông báo khi có kệ hàng hết sản phẩm.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát

Through the analysis, I have already analyzed the main functions of the application and presented them through the general use case diagram in Figure 2.1. The application includes two main actors:



Hình 2.1: Biểu đồ Use Case tổng quan

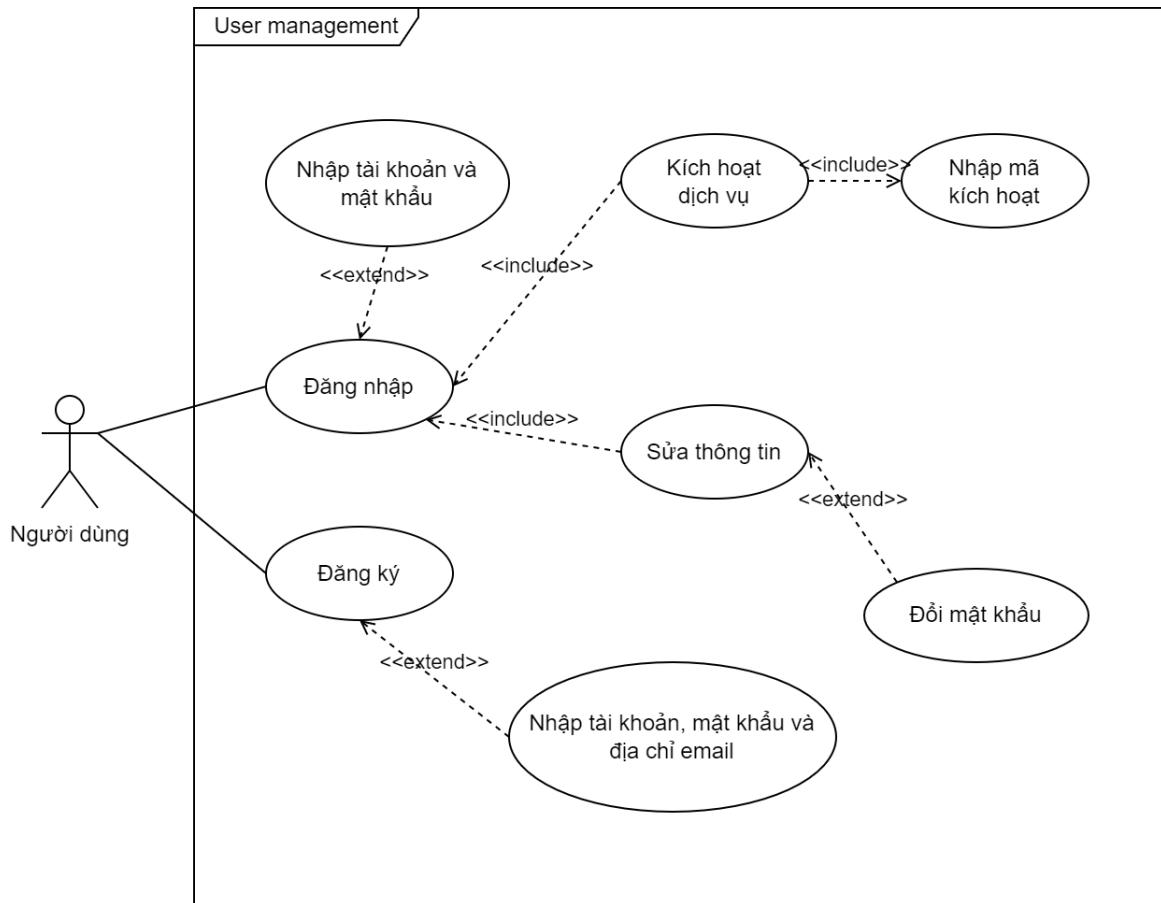
- Người dùng

Người dùng có thể đăng ký, đăng nhập tài khoản để sử dụng các dịch vụ của ứng dụng.Thêm vào đó, người dùng có thể kích hoạt những dịch vụ có sẵn bằng cách nhập mã kích hoạt. Với phần dịch vụ, có hai loại dịch vụ chính. Dịch vụ thứ nhất là dịch vụ về hình ảnh, với dịch vụ này, người dùng có thể gửi hình ảnh lên máy chủ với mỗi tùy chọn để nhận về kết quả mong muốn. Dịch vụ thứ hai là dịch vụ về camera, với dịch vụ này, người dùng có thể xem và tương tác với dòng video phát trực tuyến từ các camera được liên kết.

- Người quản trị

Người quản trị có thể quản lý các tài khoản người dùng. Ngoài ra, người quản trị còn có thể kích hoạt những dịch vụ cho từng người dùng.

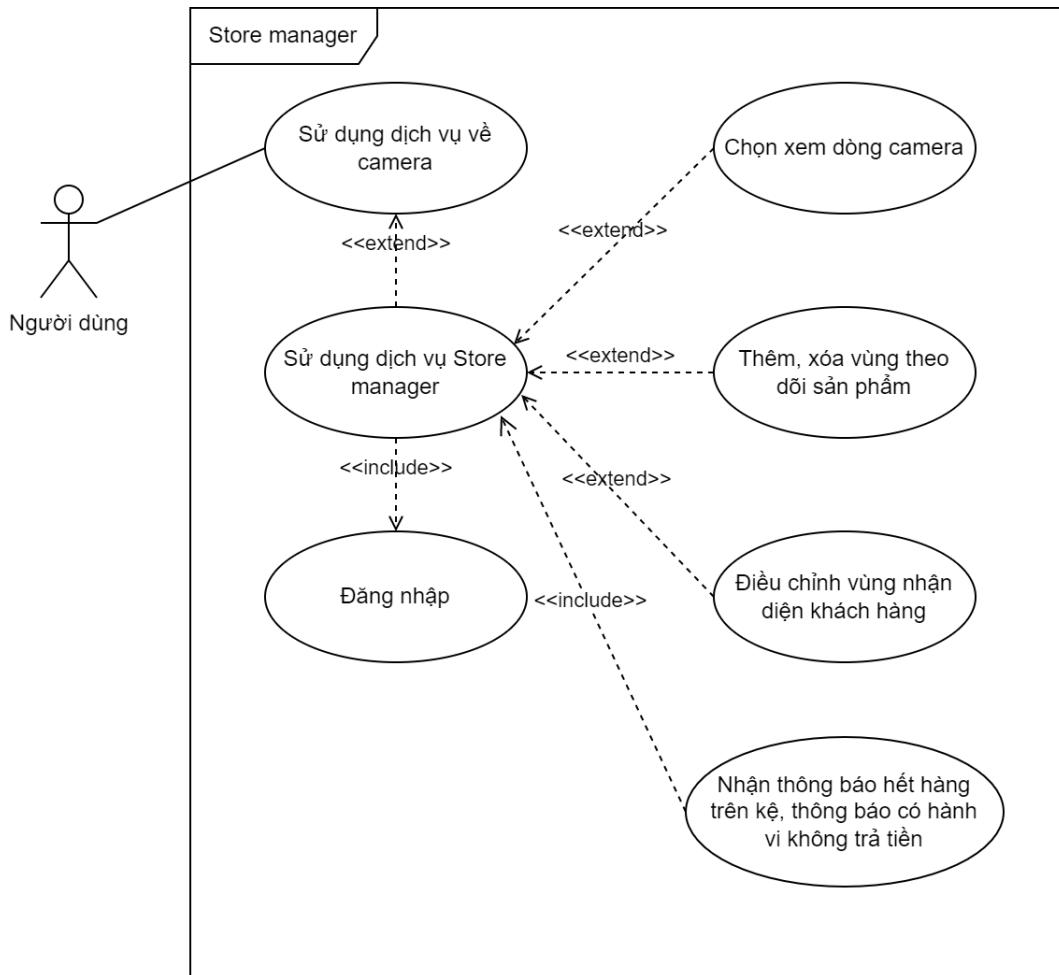
2.2.2 Biểu đồ use case phân rã Đăng ký, Đăng nhập



Hình 2.2: Biểu đồ Use Case Đăng ký, Đăng nhập

Để đăng nhập và đăng ký tài khoản, người dùng cần nhập các trường cần thiết như tên tài khoản, mật khẩu và địa chỉ mail. Tại use case này, sau khi đăng nhập, người dùng có thể thực hiện các chức năng như sửa thông tin tài khoản hay kích hoạt dịch vụ.

2.2.3 Biểu đồ use case phân rã Store manager

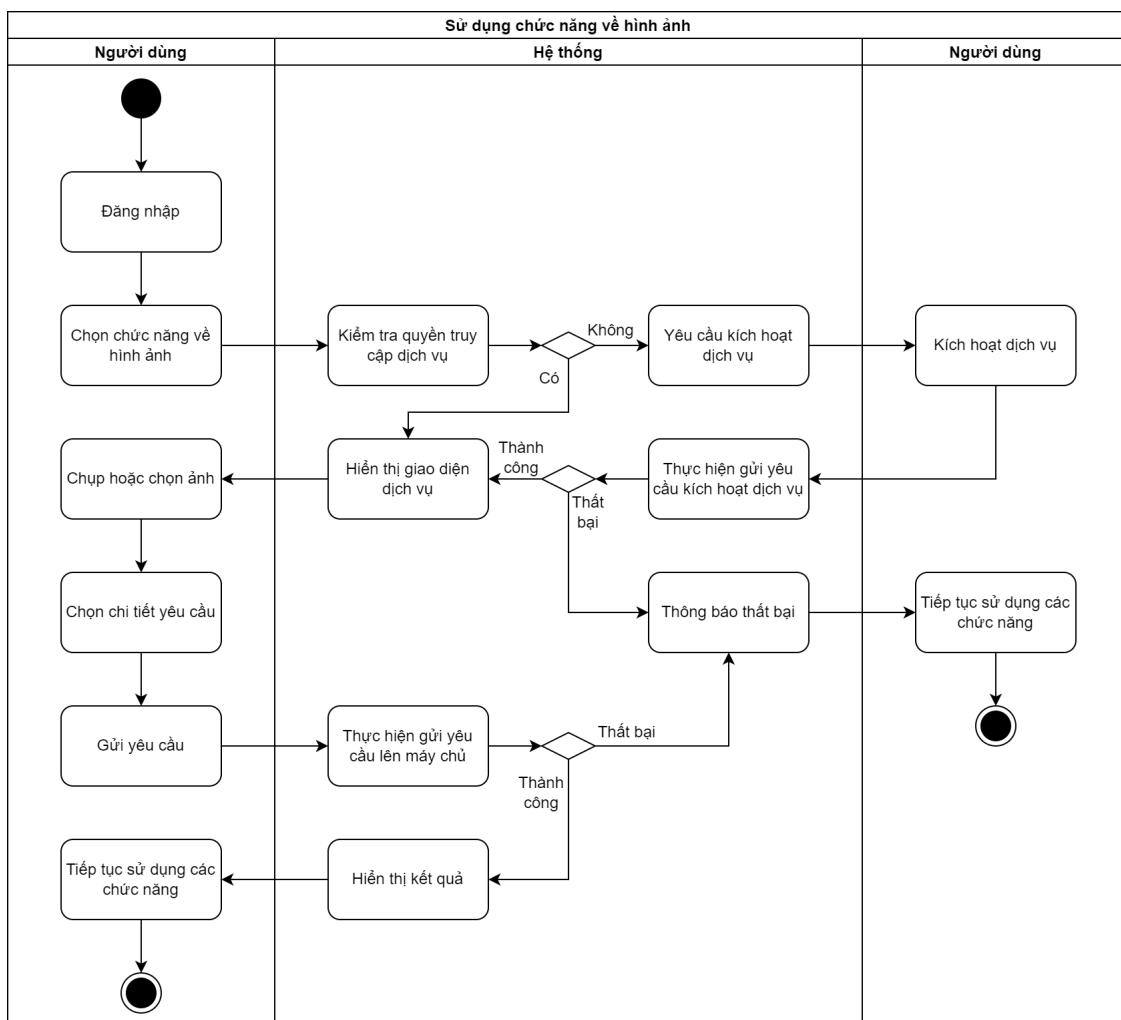


Hình 2.3: Biểu đồ Use Case Store manager

Use case Store manager là dịch vụ sử dụng bài toán "phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi" mà tôi đã phát triển, đây là một trong những dịch vụ về camera. Tại use case này, người dùng có thể thực hiện chọn xem dòng video được liên kết. Thêm vào đó là điều chỉnh các chức năng của dịch vụ như thêm, xóa vùng theo dõi sản phẩm và điều chỉnh vùng nhận diện khách hàng. Ngoài ra, người dùng sẽ nhận được thông báo khi hết hàng trên kệ hoặc có hành vi không trả tiền trong cửa hàng.

2.2.4 Quy trình nghiệp vụ

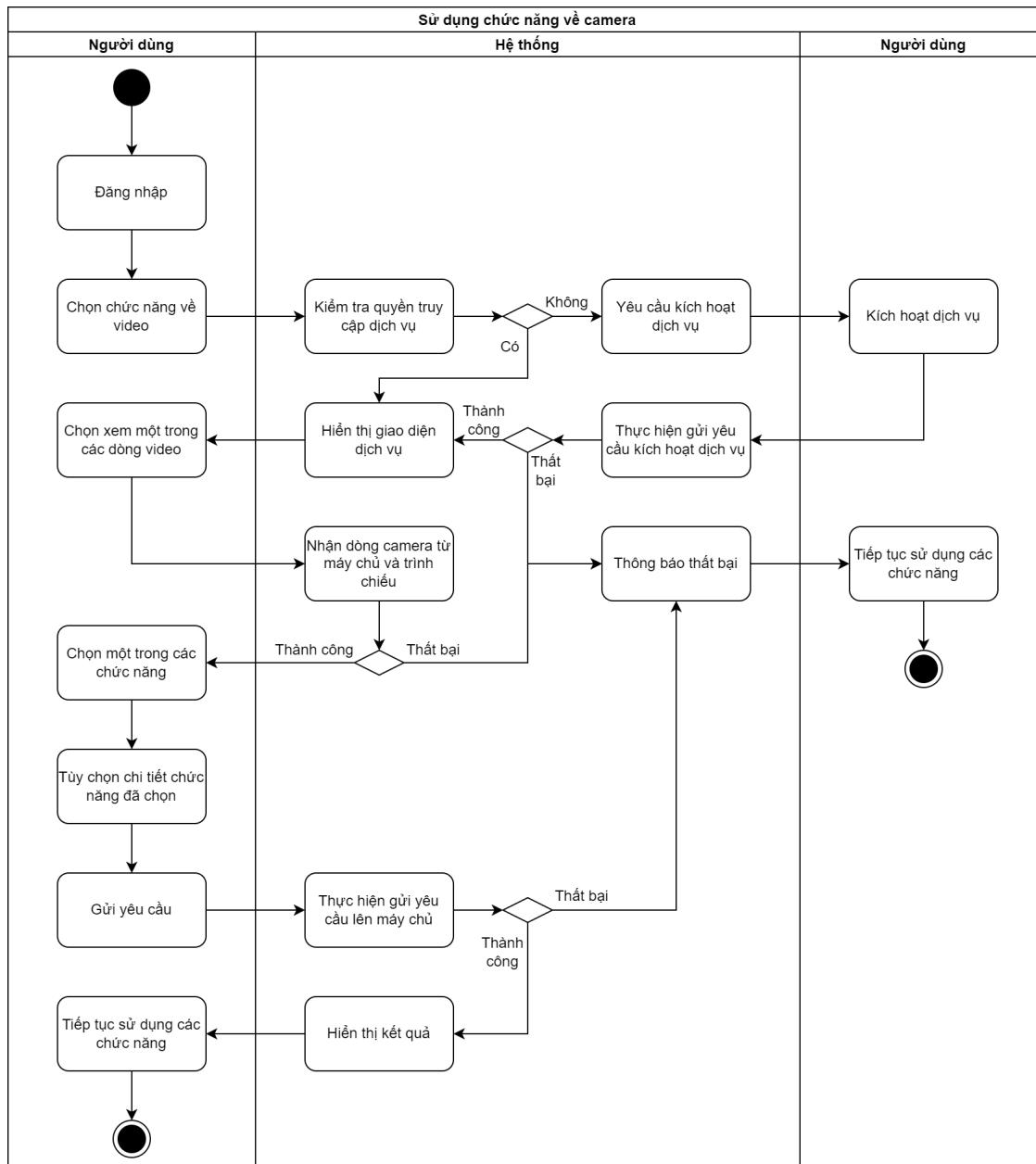
a, Quy trình sử dụng chức năng về hình ảnh



Hình 2.4: Biểu đồ hoạt động quy trình sử dụng chức năng về hình ảnh

Biểu đồ trong hình 2.4 mô tả quy trình sử dụng chức năng dịch vụ về hình ảnh.

b, Quy trình sử dụng chức năng về camera



Hình 2.5: Biểu đồ hoạt động quy trình sử dụng chức năng về camera

Biểu đồ trong hình 2.5 mô tả quy trình sử dụng chức năng dịch vụ về camera.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Đăng nhập

Bảng 2.1: Use case Đăng nhập

Use case đăng nhập			
Mã use case	UC001		
Tác nhân	Người dùng, hệ thống		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn chức năng đăng nhập
	2	Hệ thống	Hiển thị giao diện nhập thông tin đăng nhập
	3	Người dùng	Nhập tài khoản bao gồm mật khẩu và tên đăng nhập
	4	Người dùng	Nhấn nút "Đăng nhập"
	5	Hệ thống	Hiển thị đăng nhập thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Hiển thị thông báo sai thông tin đăng nhập
Hậu điều kiện	Không		

2.3.2 Đặc tả use case Chọn xem dòng camera

Bảng 2.2: Use case Chọn xem dòng camera

Use case chọn xem dòng camera			
Mã use case	UC002		
Tác nhân	Người dùng, hệ thống		
Tiền điều kiện	Chọn sử dụng dịch vụ Store manager		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Mở danh sách camera và chọn camera muốn xem
	2	Hệ thống	Gửi yêu cầu đến máy chủ
	3	Hệ thống	Thực hiện phát dòng camera nhận được
	STT	Thực hiện bởi	Hành động
	3a	Hệ thống	Hiển thị thông báo dòng camera bị lỗi
Hậu điều kiện	Không		

2.3.3 ĐẶC TẢ USE CASE Thêm vùng theo dõi sản phẩm

Bảng 2.3: Use case Thêm vùng theo dõi sản phẩm

Use case thêm vùng theo dõi sản phẩm			
Mã use case	UC003		
Tác nhân	Người dùng, hệ thống		
Tiền điều kiện	Sử dụng dịch vụ Store manager và đã chọn xem dòng camera		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn chức năng thêm vùng theo dõi sản phẩm
	2	Người dùng	Chọn vùng mong muốn trên camera
	3	Hệ thống	Gửi yêu cầu lên máy chủ
	4	Hệ thống	Hiển thị dòng camera với vùng theo dõi đã thêm
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Hiển thị thông báo lỗi khi thêm vùng theo dõi sản phẩm
Hậu điều kiện	Không		

2.3.4 ĐẶC TẢ USE CASE Xóa vùng theo dõi sản phẩm

Bảng 2.4: Use case Xóa vùng theo dõi sản phẩm

Use case xóa vùng theo dõi sản phẩm			
Mã use case	UC004		
Tác nhân	Người dùng, hệ thống		
Tiền điều kiện	Sử dụng dịch vụ Store manager và đã chọn xem dòng camera		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn chức năng xóa vùng theo dõi sản phẩm
	2	Người dùng	Nhập id vùng muốn xóa
	3	Người dùng	Thực hiện ấn nút "Xóa vùng"
	4	Hệ thống	Gửi yêu cầu lên máy chủ
	5	Hệ thống	Hiển thị dòng camera với vùng theo dõi đã xóa
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a	Hệ thống	Hiển thị thông báo lỗi xóa vùng theo dõi sản phẩm
Hậu điều kiện	Không		

2.3.5 Đặc tả use case Điều chỉnh vùng nhận diện khách hàng

Bảng 2.5: Use case Điều chỉnh vùng nhận diện khách hàng

Use case điều chỉnh vùng nhận diện khách hàng			
Mã use case	UC005		
Tác nhân	Người dùng, hệ thống		
Tiền điều kiện	Sử dụng dịch vụ Store manager và đã chọn xem dòng camera		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn chức năng điều chỉnh vùng nhận diện khách hàng
	2	Người dùng	Chọn vùng mong muốn trên camera
	3	Hệ thống	Gửi yêu cầu lên máy chủ
	4	Hệ thống	Hiển thị dòng camera với vùng theo dõi sau khi điều chỉnh
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Hiển thị thông báo lỗi khi điều chỉnh vùng nhận diện khách hàng
Hậu điều kiện	Không		

2.4 Yêu cầu phi chức năng

2.4.1 Yêu cầu về kỹ thuật

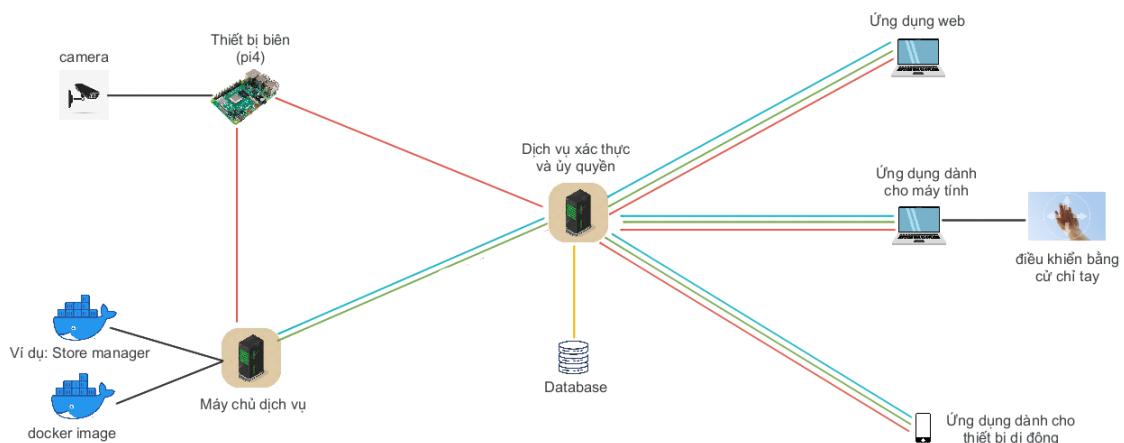
- Người dùng cần phải đăng nhập để sử dụng các chức năng của ứng dụng
- Đảm bảo hình ảnh cá nhân người dùng không bị lưu lại hay lan truyền trên internet.
- Độ trễ của dòng camera và thông báo tới người dùng không quá thấp, đảm bảo real time.
- Mô-đun xử lý bài toán hoạt động ổn định, đưa ra kết quả chính xác.

2.4.2 Yêu cầu về giao diện

- Giao diện đẹp, đơn giản, dễ sử dụng.
- Đảm bảo thao tác tối thiểu khi thực hiện các chức năng nhưng không tạo cảm giác khó hiểu cho người dùng.
- Đảm bảo các chức năng chạy ổn định, không crash giữa chừng.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Để tiến hành thực nghiệm, bài toán được triển khai trên hệ thống camera thông minh mà chúng tôi đã xây dựng. Hệ thống này có khả năng tích hợp nhiều bài toán thị giác máy tính khác nhau và được triển khai trên nhiều nền tảng như ứng dụng web, ứng dụng dành cho máy tính, ứng dụng dành cho thiết bị di động.



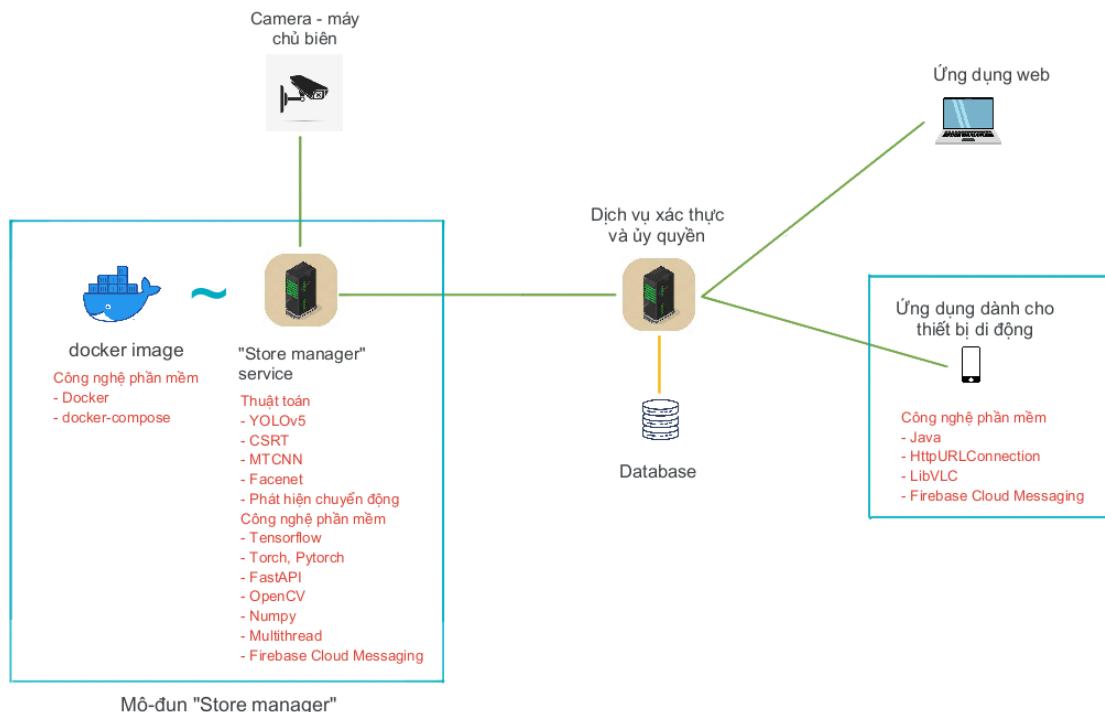
Hình 3.1: Kiến trúc tổng quan hệ thống camera thông minh

Hệ thống gồm có ba phần chính được mô tả trong hình 3.1.

Phần thứ nhất bao gồm các thiết bị biên, những thiết bị này tích hợp camera, vừa có thể ghi hình, vừa có thể xử lý các bài toán nhẹ như tăng sáng, khử sương mù trước khi truyền dòng video về máy chủ chính.

Phần thứ hai là máy chủ chính, tại đây hệ thống sẽ nhận các dòng video từ thiết bị biên và thực hiện các bài toán khác nhau đối với các dịch vụ chuyên biệt được tích hợp sẵn.

Phần thứ ba là các ứng dụng phần mềm để tương tác với người dùng. Có ba ứng dụng được phát triển, đó là ứng dụng web, ứng dụng dành cho máy tính và ứng dụng dành cho thiết bị di động. Ứng dụng sẽ cung cấp các dịch vụ được tích hợp trên máy chủ chính cho người dùng.



Hình 3.2: Các phần đảm nhiệm trong hệ thống

Trong hệ thống này, tôi đảm nhiệm hai phần được đánh dấu bằng khung màu xanh trong hình 3.2. Đó việc phát triển bài toán "Phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi" để tích hợp thành một dịch vụ cung cấp đến người dùng. Đồng thời phát triển ứng dụng dành cho thiết bị di động, cụ thể là ứng dụng dành cho thiết bị Android. Tôi đã nghiên cứu và lựa chọn nhiều công nghệ sao cho phù hợp với mục đích này. Vì vậy trong chương này tôi xin được trình bày các công nghệ đã sử dụng, lý do sử dụng công nghệ đó và nó được sử dụng ở đâu trong ứng dụng của mình.

3.1 Công nghệ phát triển Mô-đun "Store manager"

3.1.1 YOLOv5

a, Khái niệm

You only look once (YOLO) là một mô hình CNN để phát hiện vật thể mà một ưu điểm nổi trội là nhanh hơn nhiều so với những mô hình cũ. YOLOv5 là một mô hình thuộc họ mô hình YOLO. YOLOv5 được công bố gần đây với những so sánh ban đầu cho thấy độ chính xác tương đương YOLOv4 và có tốc độ nhanh hơn khi thực hiện dự đoán.

b, Lý do lựa chọn

YOLOv5 là phiên bản mới nhất trong các mô hình YOLO, nó có nhiều cải tiến về tốc độ xử lý. Ngoài ra, YOLOv5 còn có thể dễ dàng tùy chỉnh sự cân bằng giữa tốc độ xử lý và độ chính xác sao cho phù hợp với mục đích sử dụng. Việc huấn luyện mô hình cũng vô cùng dễ dàng.

c, Nơi sử dụng

YOLOv5 được sử dụng để phát hiện những sản phẩm trong cửa hàng, từ đó đưa vào theo dõi.

3.1.2 CSRT

a, Khái niệm

Object Tracking [13] là bài toán theo dõi một hoặc nhiều đối tượng chuyển động theo thời gian trong một video. Hiểu một cách đơn giản nhất, nó là bài toán ở mức độ cao hơn so với object detection, khi đối tượng được xử lí không đơn giản là một hình ảnh mà là một chuỗi các hình ảnh. CSRT là một trong những thuật toán tracking được OpenCV [14] cung cấp. CSRT Tracker có tốc độ chậm nhưng độ chính xác lại cao.

b, Lý do lựa chọn

CSRT mang lại độ chính xác cao, vẫn duy trì theo dõi được vật thể khi bị che khuất một phần.

c, Nơi sử dụng

Trong đồ án này, tôi sử dụng CSRT tracker để thực hiện theo dõi vị trí của các sản phẩm trong cửa hàng

3.1.3 MTCNN

a, Khái niệm

MTCNN là viết tắt của Multi-task Cascaded Convolutional Networks. Nó bao gồm 3 CNN xếp chồng lên nhau và đồng thời hoạt động khi phát hiện khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đóng một vai trò khác nhau trong nhiệm vụ.

b, Lý do lựa chọn

MTCNN phổ biến và dễ sử dụng vì kết quả hiện đại trên nhiều bộ dữ liệu điểm chuẩn, và vì nó có thể nhận ra các đặc điểm khuôn mặt khác như miệng hoặc mắt, được gọi là phát hiện điểm khuôn mặt. Sử dụng cách tiếp cận MTCNN này, hệ thống đề xuất của chúng tôi có thể đạt được khả năng suy luận nhanh chóng.

c, Nơi sử dụng

Tôi sử dụng MTCNN để thực hiện bài toán phát hiện khuôn mặt. Từ đó lấy kết quả làm đầu vào cho bài toán nhận diện khách hàng trong cửa hàng.

3.1.4 Facenet**a, Khái niệm**

FaceNet là một hệ thống nhận dạng khuôn mặt được công bố vào năm 2015 bởi Google, đã đạt được kết quả tốt trên nhiều bộ dữ liệu khuôn mặt. Đối với các tác vụ nhận dạng khuôn mặt người, FaceNet đã được chứng minh là hiệu quả, đạt độ chính xác cao. FaceNet có thể được sử dụng để trích xuất các đặc điểm có chất lượng cao từ các khuôn mặt bằng cách sử dụng mạng nơ-ron tích chập phức tạp với hàm mã hóa bộ ba, được gọi là nhúng khuôn mặt, có thể được đưa vào thuật toán máy vector hỗ trợ để đào tạo hệ thống nhận dạng khuôn mặt.

b, Lý do lựa chọn

Facenet dễ sử dụng, mô hình chỉ cần thu thập dữ liệu ảnh được căn chỉnh tối ưu về vùng cắt của mặt để đưa ra kết quả tốt. Ngoài ra, Facenet luôn đáp ứng một hiệu suất vô cùng ổn định với một số lượng dữ liệu ảnh rất lớn.

c, Nơi sử dụng

Trong đồ án này, tôi sử dụng Facenet để trích xuất đặc trưng của khuôn mặt khách hàng, từ đó tiến hành phân tích, đưa ra kết quả để nhận diện khách hàng trong cửa hàng.

3.1.5 Phát hiện chuyển động**a, Khái niệm**

Phát hiện chuyển động là quá trình phát hiện sự thay đổi vị trí của đối tượng nào đó hoặc sự thay đổi của môi trường xung quanh. Trong đồ án này tôi thực hiện phương pháp so sánh hai khung hình liền kề nhau trong dòng video để phát hiện chuyển động.

b, Lý do lựa chọn

Tốc độ xử lý của phương pháp này nhanh, đảm bảo được vấn đề về real time.

c, Nơi sử dụng

Phát hiện chuyển động được áp dụng trong việc phát hiện các sự kiện xảy ra trong dòng video nhận được từ camera, nếu không thấy sự chuyển động, các bài toán sẽ được tắt tạm thời cho đến khi có sự kiện diễn ra để tiết kiệm tài nguyên hệ thống.

3.1.6 Docker

a, Khái niệm

Docker [15] là một nền tảng cung cấp cách để xây dựng, triển khai và chạy ứng dụng một cách dễ dàng bằng các sử dụng containers. Các containers cho phép lập trình viên đóng gói một ứng dụng với tất cả các phần cần thiết và đóng gói tất cả thành một image.

b, Lý do lựa chọn

Doker có khả năng triển khai trên nhiều nền tảng như Linux, Window và có những ưu điểm sau.

- Tính dễ sử dụng: Docker rất dễ để sử dụng đối với lập trình viên, nó tận dụng lợi thế của container để chạy ứng dụng một cách nhanh chóng.
- Tốc độ: Docker container rất nhẹ và nhanh, docker có thể khởi tạo và chạy trong vòng vài giây,

c, Nơi sử dụng

Tôi sử dụng Docker để đóng gói bài toán "phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi" và triển khai nó trên máy chủ, tích hợp như một dịch vụ trong hệ thống camera thông minh nhằm mục đích thực nghiệm.

3.1.7 Tensorflow

a, Khái niệm

Tensorflow [16] là thư viện mã nguồn mở cho machine learning nổi tiếng, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong học máy và học sâu đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều.

b, Lý do lựa chọn

Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép tensorflow xây dựng được nhiều mạng học máy. Dựa vào đó tôi đã lựa chọn ra mạng thích hợp để ứng dụng vào bài toán phát hiện và nhận diện khuôn mặt.

c, Nơi sử dụng

Tensorflow dùng để triển khai mô hình Facenet để xây dựng bài toán nhận diện khuôn mặt.

3.1.8 Torch, Pytorch

a, Khái Niệm

Torch là một thư viện máy học mã nguồn mở, được phát triển bằng ngôn ngữ tập lệnh dựa trên ngôn ngữ lập trình Lua [17]. Nó cung cấp một loạt các thuật toán để học sâu và được tạo ra tại IDIAP ở EPFL [18].

PyTorch [19] là một thư viện học máy mã nguồn mở, được tạo ra dựa trên thư viện Torch và được ứng dụng vào các bài toán như thị giác máy tính, xử lý ngôn ngữ tự nhiên. Nó được phát triển bởi phòng Nghiên cứu AI Facebook [20].

b, Lý do lựa chọn

Torch và Pytorch tích hợp nhiều API mạnh mẽ giúp việc huấn luyện mô hình trở nên dễ dàng hơn.

c, Nơi sử dụng

Tôi sử dụng torch để huấn luyện và triển khai mô hình Yolov5 áp dụng trên bài toán nhận diện sản phẩm. Thư viện Pytorch dùng để triển khai mô hình MTCNN nhằm mục đích xây dựng bài toán phát hiện khuôn mặt.

3.1.9 FastAPI

a, Khái niệm

FastAPI [21] là một web framework được sử dụng để xây dựng các web API có hiệu năng cao và khả năng triển khai nhanh trên nền tảng Python.

b, Lý do lựa chọn

- Cú pháp đơn giản dựa trên ngôn ngữ Python [22]
- Tốc độ xử lý nhanh và có độ trễ thấp.
- So với Flask [23] hoặc Django [24], FastAPI có thể thực hiện đồng thời nhiều yêu cầu từ khách hàng cùng một lúc nhờ cơ chế tự đồng bộ hóa.
- FastAPI hỗ trợ tạo tài liệu API, rất hữu ích cho các nhà phát triển khách hàng.

c, Nơi sử dụng

FastAPI được sử dụng để deploy bài toán đã phát triển lên máy chủ chính, từ đó cung cấp các dịch vụ cho ứng dụng người dùng sử dụng.

3.1.10 OpenCV

a, Khái niệm

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở được viết bằng ngôn ngữ C/C++ [25]. OpenCV cung cấp các hàm giúp xử lý về thị giác máy tính, học máy và xử lý ảnh. OpenCV cung cấp giao diện có thể được sử dụng bởi các

ngôn ngữ C/C++, Python, Java vì vậy hỗ trợ được cho Window, Linux, Mac OS lẫn Android và IOS.

b, Lý do lựa chọn

- OpenCV cung cấp nhiều hàm hỗ trợ cho việc xử lý hình ảnh và đọc ghi video.
- Có độ tương thích cao với ngôn ngữ Python.
- Có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực.

c, Nơi sử dụng

OpenCV được sử dụng để xử lý hình ảnh nhận được từ dòng camera. Ngoài ra thư viện này còn giúp triển khai mô hình tracking CSRT nhằm mục đích theo dõi các sản phẩm.

3.1.11 Numpy

a, Khái niệm

Numpy [26] là một thư viện mã nguồn mở dành cho ngôn ngữ Python nhằm mục đích xử lý, tính toán và tương tác với các chuỗi, ma trận số lớn và đa chiều.

b, Lý do lựa chọn

Numpy cung cấp các hàm tính toán mạnh mẽ đối với các chuỗi, vector.

c, Nơi sử dụng

Dựa vào khả năng tính toán của Numpy, tôi đã sử dụng nó trong việc tính toán sự sai lệch về vị trí của các sản phẩm, từ đó đưa ra các kết luận cần thiết. Ngoài ra, Numpy còn được sử dụng trong việc so sánh các vector đặc trưng nhận được từ mô hình Facenet.

3.1.12 Multithread

a, Khái niệm

Thread là thuật ngữ chỉ về luồng trong lập trình, luồng được hiểu là một tiến trình nhỏ hay còn được gọi là tiến trình con. Multithread là đa luồng, đây là một tiến trình bao gồm nhiều luồng khác nhau chạy đồng thời trong cùng một khoảng thời gian. Các luồng cùng một lúc xử lý các công việc khác nhau giúp cho ứng dụng hoạt động mượt hơn, hiệu quả hơn.

b, Lý do lựa chọn

Đa luồng có thể xử lý nhiều tiến trình hoàn toàn độc lập với nhau, khi một luồng xảy ra ngoại lệ thì các luồng khác đều không bị ảnh hưởng, đảm bảo tính ổn định của chương trình. Mặc dù các luồng chạy độc lập với nhau nhưng hoàn toàn có khả

năng chia sẻ tài nguyên, cùng xử lý một vấn đề một lúc nhằm tăng tốc độ xử lý.

c, Nơi sử dụng

Trong đồ án này, tôi áp dụng Multithread để xử lý cùng lúc nhiều dòng camera khác nhau. Các luồng cùng lúc xử lý thông tin nhận được từ camera và gửi kết quả về luồng xử lý chính để đưa ra kết quả của bài toán.

3.1.13 Firebase Cloud Messaging

a, Khái niệm

Firebase [27] là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây, kèm theo đó là các dịch vụ và hệ thống máy chủ cực kỳ mạnh mẽ của Google. Firebase Cloud Messaging (FCM) [28] là một dịch vụ tích hợp của Firebase. Thông qua FCM, các nhà phát triển có thể gửi thông báo một cách nhanh chóng tới các thiết bị cài đặt ứng dụng của họ.

b, Lý do lựa chọn

Firebase hỗ trợ phát triển ứng dụng một cách đơn giản. Tiếp theo chính là Firebase hỗ trợ cho việc phát triển ứng dụng rất nhanh chóng. Ngoài ra còn cung cấp đa dịch vụ cho mục đích phát triển.

c, Nơi sử dụng

Trong đồ án này, tôi dùng Firebase Cloud Messaging trong việc gửi và nhận thông báo từ máy chủ chính tới các ứng dụng người dùng.

3.2 Công nghệ phát triển ứng dụng trên thiết bị Andoird

3.2.1 Java

a, Khái niệm

Java là một ngôn ngữ lập trình bậc cao, hướng đối tượng, mạnh mẽ và bảo mật cao. Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995.

b, Lý do lựa chọn

Java là một trong những ngôn ngữ lập trình được sử dụng rộng rãi và là ngôn ngữ chính thức để phát triển Android. Android Studio hỗ trợ tốt cho Java. Khi cài đặt thư viện dùng Java, Android Studio làm cho việc cấu hình các thư viện đó trong dự án trở nên đơn giản.

c, Nơi sử dụng

Tôi dùng Java làm ngôn ngữ chính để phát triển ứng dụng dành cho thiết bị di động.

3.2.2 HttpURLConnection

a, Khái niệm

HttpURLConnection [29] là một thư viện được Java hỗ trợ, được sử dụng để thực hiện một yêu cầu duy nhất nhưng kết nối mạng cơ bản với máy chủ HTTP.

b, Lý do lựa chọn

HttpURLConnection cung cấp giao thức kết nối đơn giản và đặc biệt có khả năng tùy chỉnh cao, dễ dàng cấu hình các yêu cầu theo mục đích sử dụng.

c, Nơi sử dụng

Để gửi các yêu cầu và nhận thông tin từ máy chủ chính bằng thiết bị Android, tôi dùng HttpURLConnection để tương tác với máy chủ thông qua các API có sẵn.

3.2.3 LibVLC

a, Khái niệm

LibVLC [30] là thư viện Android nhúng công cụ VLC, cung cấp rất nhiều tính năng đa phương tiện. Có khả năng phát mọi định dạng tệp phương tiện, mọi codec và mọi giao thức phát trực tuyến.

b, Lý do lựa chọn

Thư viện LibVLC hỗ trợ giải mã hiệu quả trên mọi nền tảng, phần cứng khác nhau. Đặc biệt LibVLC có khả năng truyền và phát trực tiếp tới các trình kết xuất ở xa.

c, Nơi sử dụng

Tôi dùng thư viện LibVLC để đọc và phát những dòng camera trên thiết bị di động.

CHƯƠNG 4. THIẾT KẾ ỨNG DỤNG

4.1 Thiết kế kiến trúc

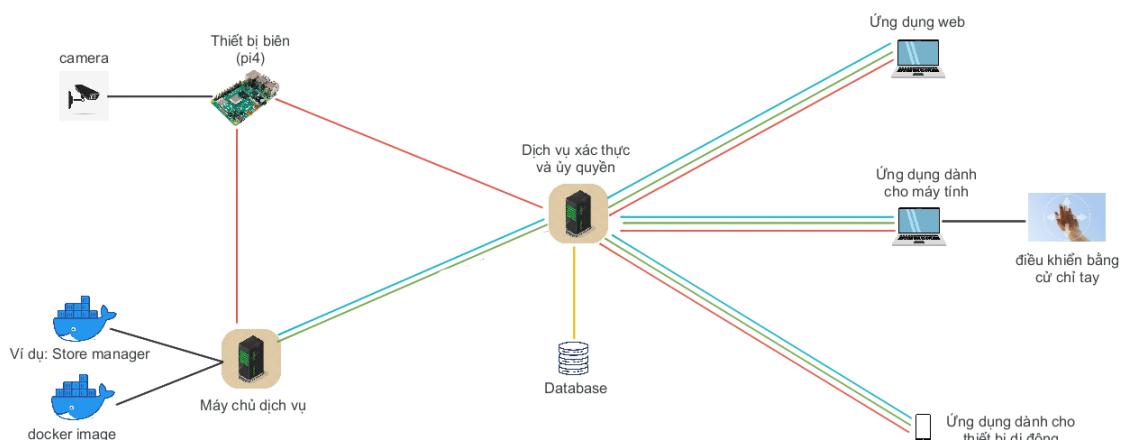
4.1.1 Lựa chọn kiến trúc phần mềm

Để xây dựng một hệ thống hoàn chỉnh, chúng tôi lựa chọn mô hình Client - Server [31] và áp dụng kiến trúc Microservice [32] để tích hợp cùng lúc nhiều dịch vụ và đảm bảo khả năng mở rộng của hệ thống.

Mô hình Client - Server là một mô hình mạng máy tính, trong đó các thiết bị con đóng vai trò như một máy khách, chúng có nhiệm vụ gửi yêu cầu tới máy chủ, và máy chủ sẽ xử lý yêu cầu để trả lại kết quả cho máy khách đó.

Kiến trúc Microservice là kiến trúc chia một ứng dụng thành một bộ các microservice, mỗi microservice là một service có thể được triển khai và chạy độc lập. Các service này tách biệt về mặt mã nguồn, hoạt động và dữ liệu. Do các microservice này là độc lập, chúng không giao tiếp trực tiếp với nhau mà qua một thành phần trung gian được gọi là API gateway.

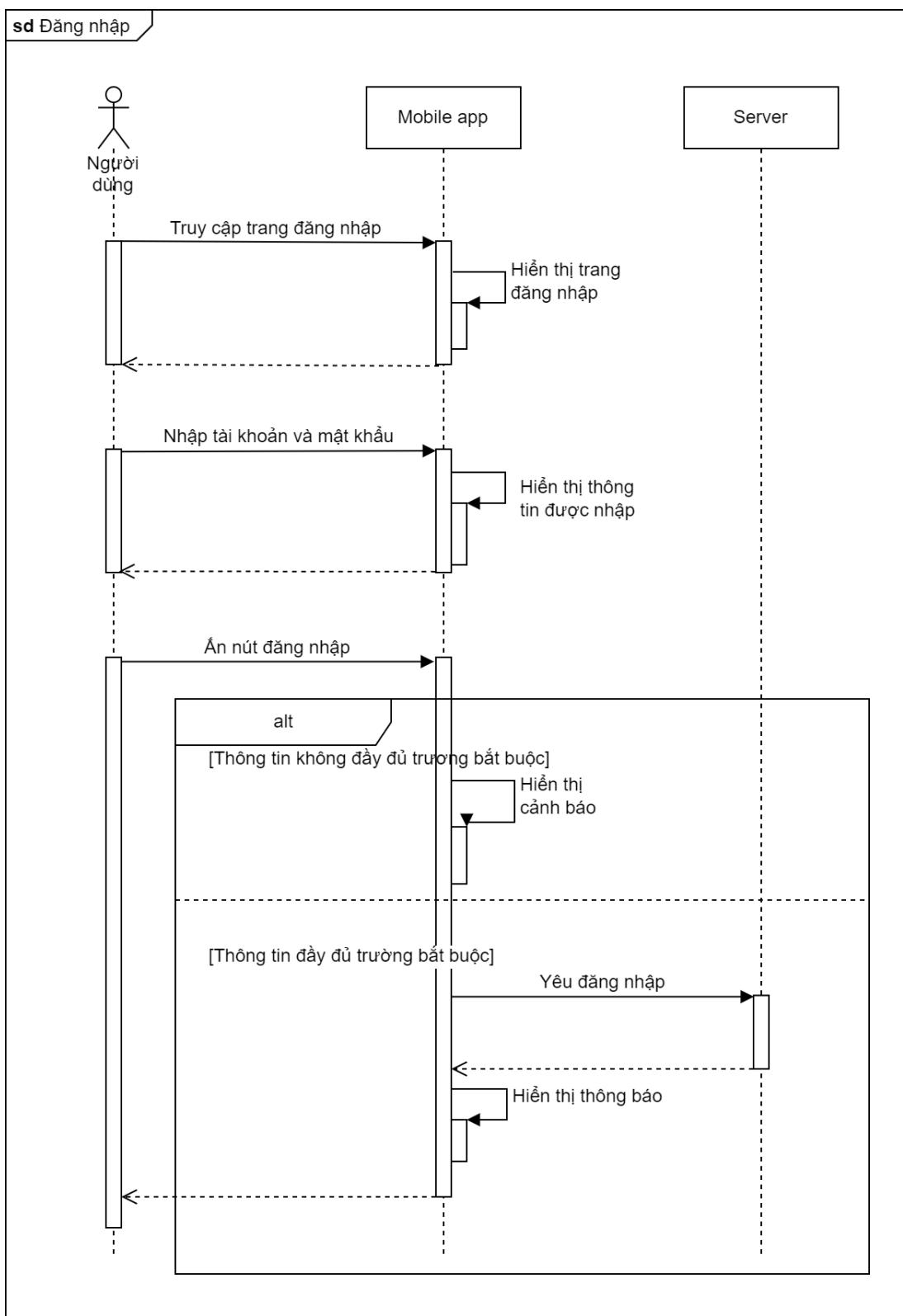
Cụ thể kiến trúc của hệ thống được mô tả như hình 4.1



Hình 4.1: Kiến trúc tổng quan hệ thống camera thông minh

4.1.2 Thiết kế luồng hoạt động

a, Luồng đăng nhập

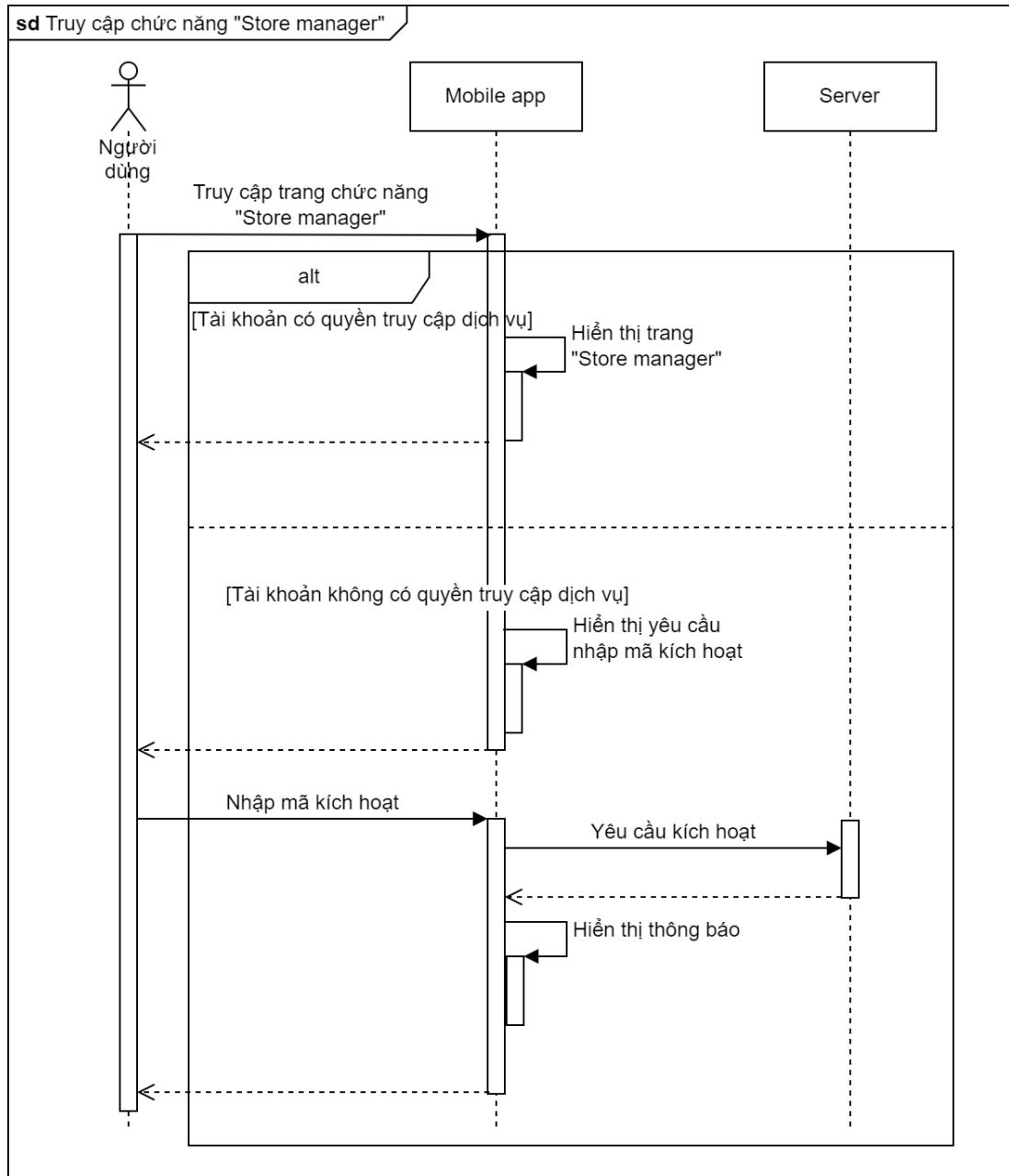


Hình 4.2: Luồng đăng nhập

Hình 4.2 mô tả chi tiết luồng đăng nhập. Để thực hiện đăng nhập, người dùng thực hiện truy cập trang đăng nhập. Sau đó người dùng thực hiện nhập tên đăng

nhập và mật khẩu. Ứng dụng sẽ gửi yêu cầu tới máy chủ để xác thực. Nếu hợp lệ, người dùng sẽ đăng nhập thành công, còn nếu không thành công, ứng dụng sẽ báo lỗi tương ứng cho người dùng.

b, Luồng truy cập chức năng "Store manager"

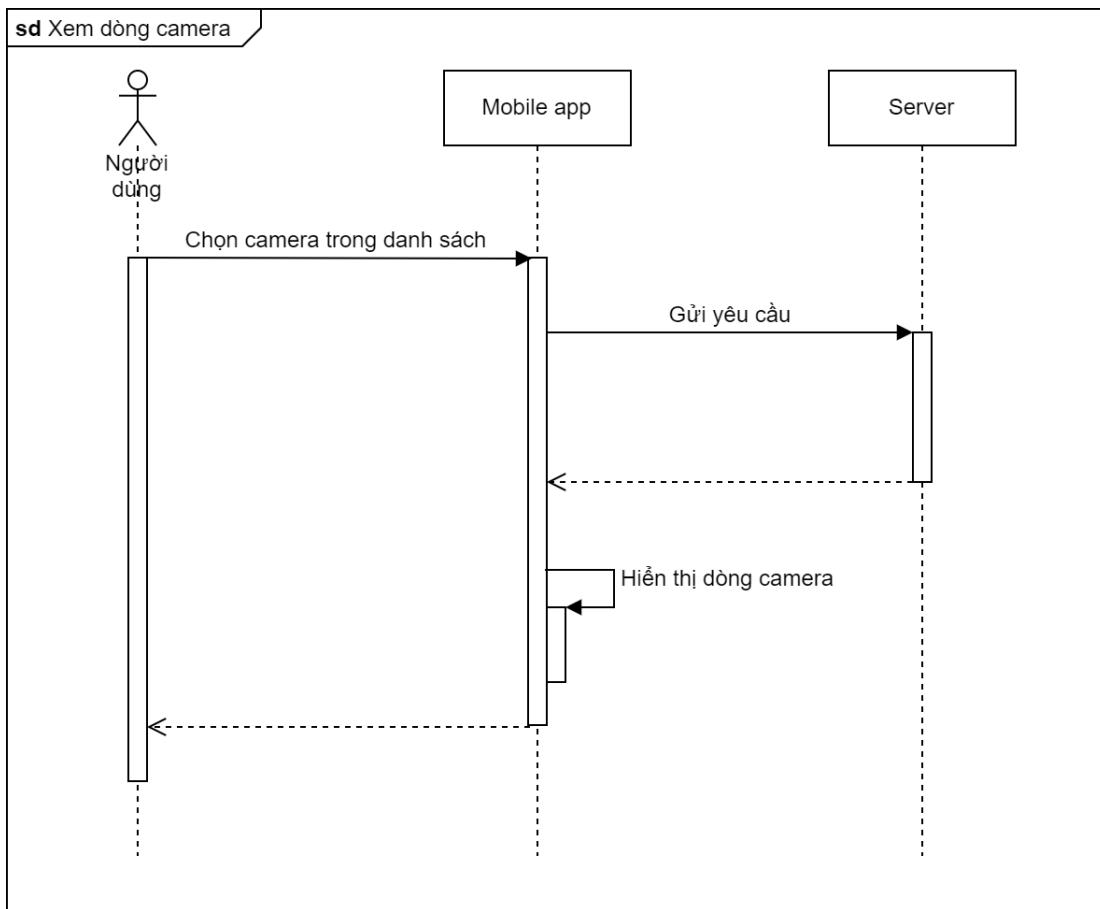


Hình 4.3: Luồng truy cập chức năng "Store manager"

Hình 4.3 mô tả chi tiết luồng truy cập chức năng "Store manager". Để sử dụng chức năng "Store manager", người dùng chọn chức năng đó trên giao diện ứng dụng. Ứng dụng sẽ kiểm tra người dùng có quyền sử dụng chức năng "Store manager" hay không. Nếu có ứng dụng sẽ hiển thị trang chức năng cho người dùng tiếp tục sử dụng. Nếu không, ứng dụng sẽ hiển thị bảng yêu cầu nhập mã kích hoạt dịch vụ. Người dùng tiến hành nhập mã kích hoạt dịch vụ, sau đó ứng dụng sẽ gửi yêu

cầu lên máy chủ để xác thực. Nếu thành công, hệ thống sẽ hiển thị trang chức năng cho người dùng tiếp tục sử dụng ,còn nếu không thành công, ứng dụng sẽ báo lỗi tương ứng cho người dùng.

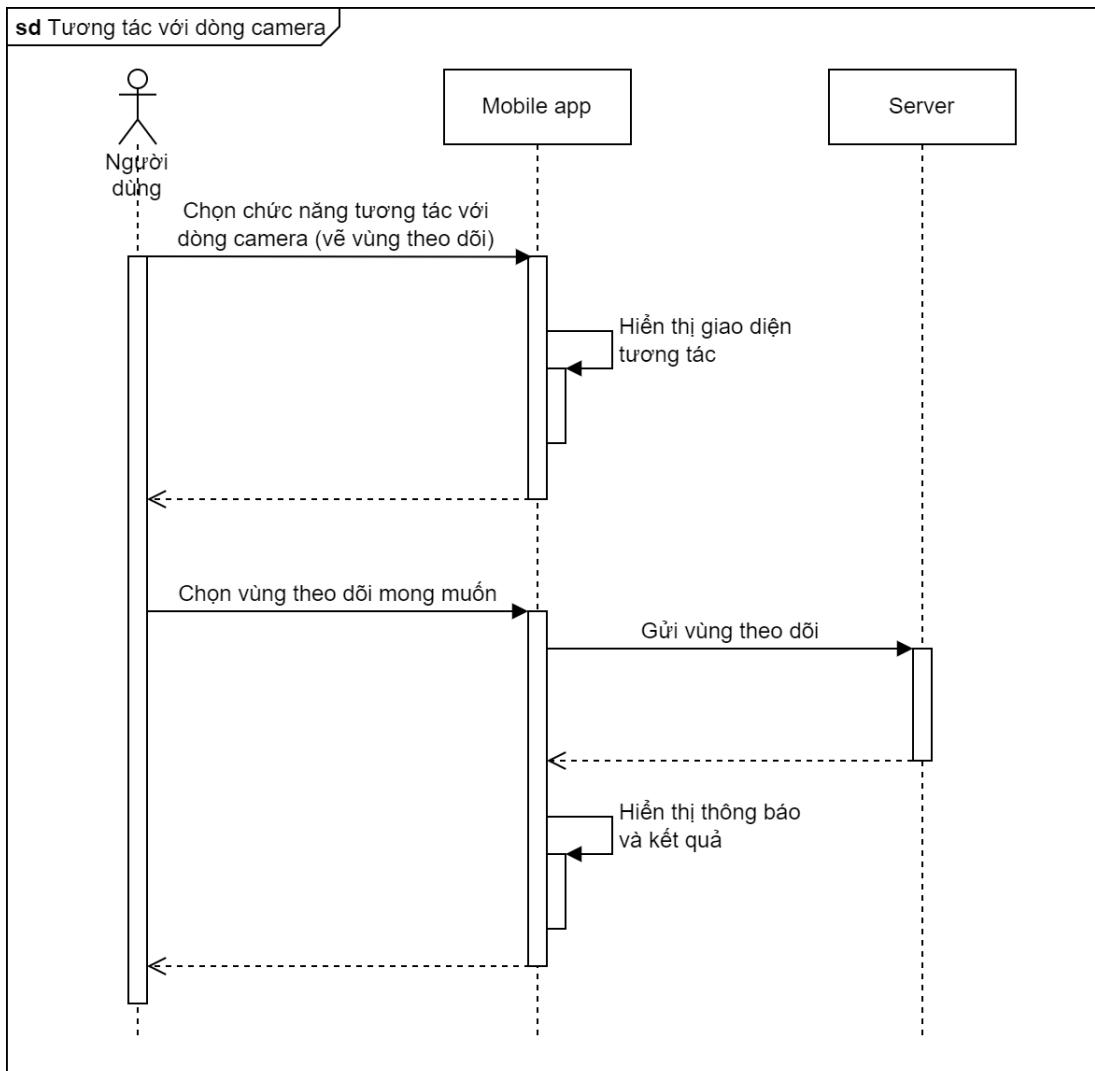
c, Luồng xem dòng camera trong chức năng "Store manager"



Hình 4.4: Luồng xem dòng camera trong chức năng "Store manager"

Hình 4.4 mô tả chi tiết luồng xem dòng camera trong chức năng "Store manager". Trong giao diện chức năng "Store manager", để xem dòng camera, người dùng chọn camera muốn xem trên giao diện. Sau đó hệ thống sẽ gửi yêu cầu lên máy chủ để xác thực. Nếu thành công, hệ thống sẽ phát dòng video cho người dùng theo dõi ,còn nếu không thành công, ứng dụng sẽ báo lỗi tương ứng cho người dùng.

d, Luồng tương tác với dòng camera trong chức năng "Store manager"



Hình 4.5: Luồng tương tác với dòng camera trong chức năng "Store manager"

Hình 4.5 mô tả chi tiết luồng tác với dòng camera trong chức năng "Store manager". Trong giao diện chức năng "Store manager", để tương tác với dòng camera, người dùng chọn chức năng tương tác muốn sử dụng, cụ thể là thêm vùng theo dõi trên camera. Sau đó người dùng thực hiện chọn vùng theo dõi mong muốn. Hệ thống sẽ gửi yêu cầu tới máy chủ để xác thực. Hệ thống sẽ báo kết quả cho người dùng khi có kết quả.

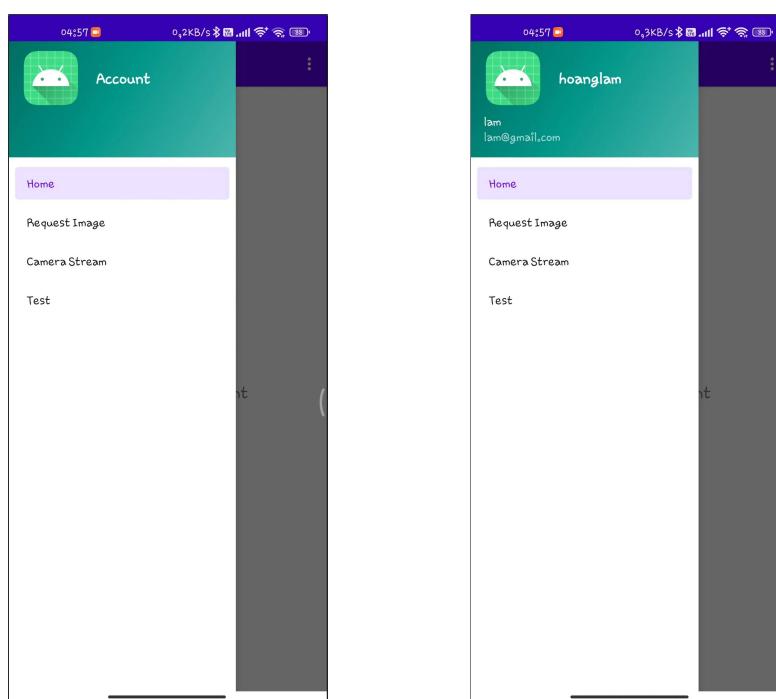
4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Ứng dụng có tổng cộng sáu màn hình chính, bao gồm:

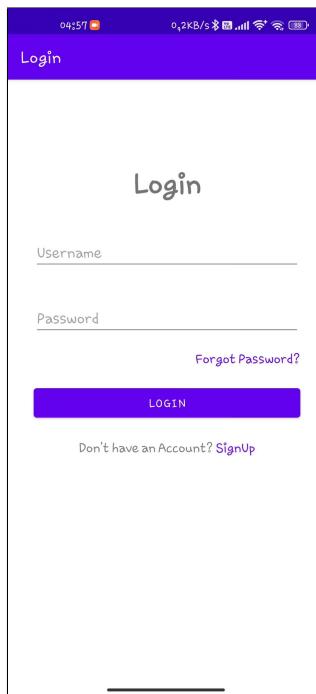
- Màn hình chính gồm menu chọn chức năng.
- Màn hình đăng nhập.
- Màn hình đăng ký.
- Màn hình chọn các dịch vụ về hình ảnh.
- Màn hình chọn các dịch vụ về camera.
- Màn hình chức năng "Store manager"

a, Màn hình chính

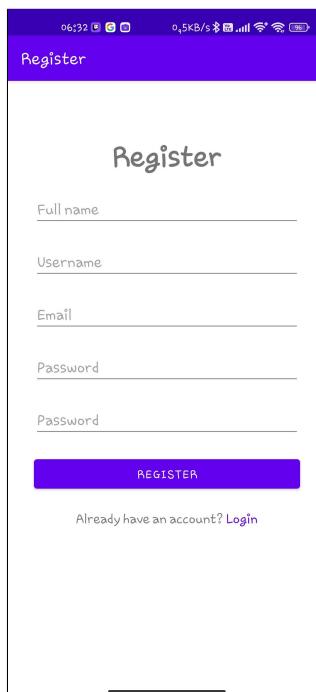


Hình 4.6: Màn hình chính khi chưa đăng nhập **Hình 4.7:** Màn hình chính khi đã đăng nhập

Hình 4.6 là giao diện màn hình chính cùng menu chọn chức năng khi chưa đăng nhập. Sau khi đăng nhập, trên thanh menu sẽ hiện thêm thông tin tài khoản người dùng như hình 4.7

b, Màn hình đăng nhập**Hình 4.8:** Màn hình đăng nhập

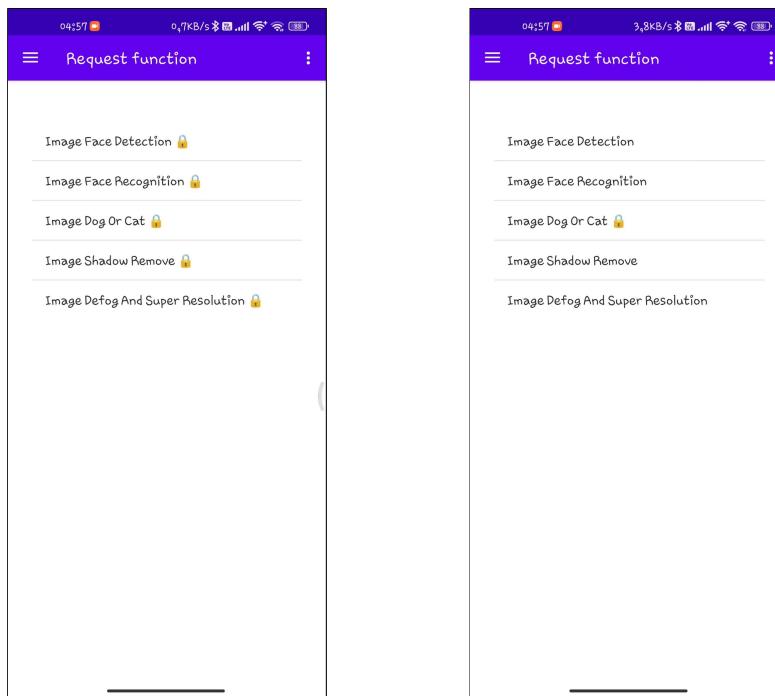
Hình 4.8 là giao diện màn hình đăng nhập. Giao diện bao gồm ô nhập tên đăng nhập, ô nhập mật khẩu và nút đăng nhập. Ngoài ra còn có thêm tùy chọn "Quên mật khẩu" và "Đăng ký"

c, Màn hình đăng ký**Hình 4.9:** Màn hình đăng ký

Hình 4.9 là giao diện màn hình đăng nhập. Giao diện bao gồm ô nhập tên người

dùng, tên đăng nhập, địa chỉ mail, ô nhập mật khẩu và nút đăng ký. Ngoài ra còn có thêm tùy chọn "Đăng nhập" khi người dùng đã có tài khoản.

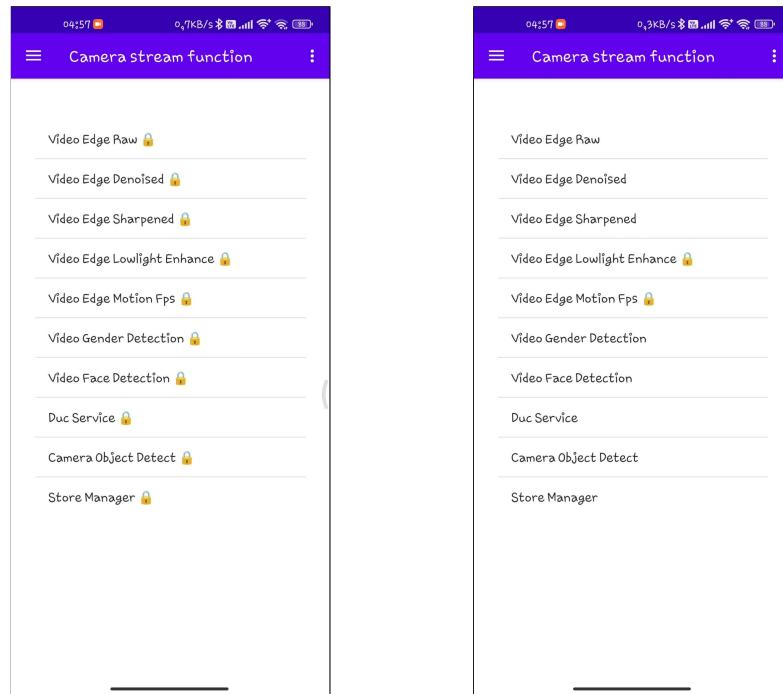
d, Màn hình chọn các dịch vụ về hình ảnh



Hình 4.10: Màn hình chọn các dịch vụ
Hình 4.11: Màn hình chọn các dịch vụ
 về hình ảnh khi chưa đăng nhập về hình ảnh khi đăng nhập

Hình 4.10 là giao diện màn hình chọn các dịch vụ về hình ảnh khi chưa đăng nhập. Khi chưa đăng nhập, các chức năng đều khóa và không sử dụng được. Sau khi đăng nhập, các dịch vụ mà người dùng đã mở khóa sẽ được hiển thị để người dùng sử dụng như hình 4.11

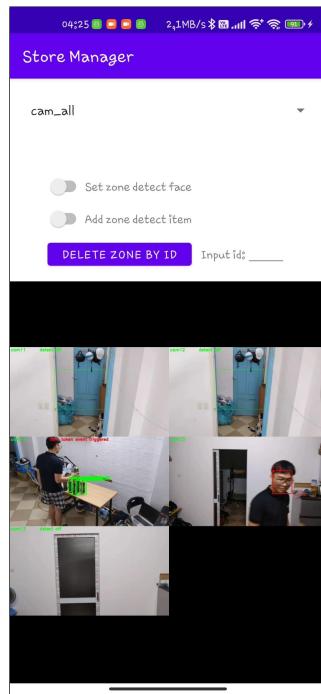
e, Màn hình chọn các dịch vụ về camera



Hình 4.12: Màn hình chọn các dịch vụ
Hình 4.13: Màn hình chọn các dịch vụ
 về camera khi chưa đăng nhập về camera khi đăng nhập

Hình 4.12 là giao diện màn hình chọn các dịch vụ về camera khi chưa đăng nhập. Khi chưa đăng nhập, các chức năng đều khóa và không sử dụng được. Sau khi đăng nhập, các dịch vụ mà người dùng đã mở khóa sẽ được hiển thị để người dùng sử dụng như hình 4.13

f, Màn hình chức năng "Store manager"



Hình 4.14: Màn hình chức năng "Store manager"

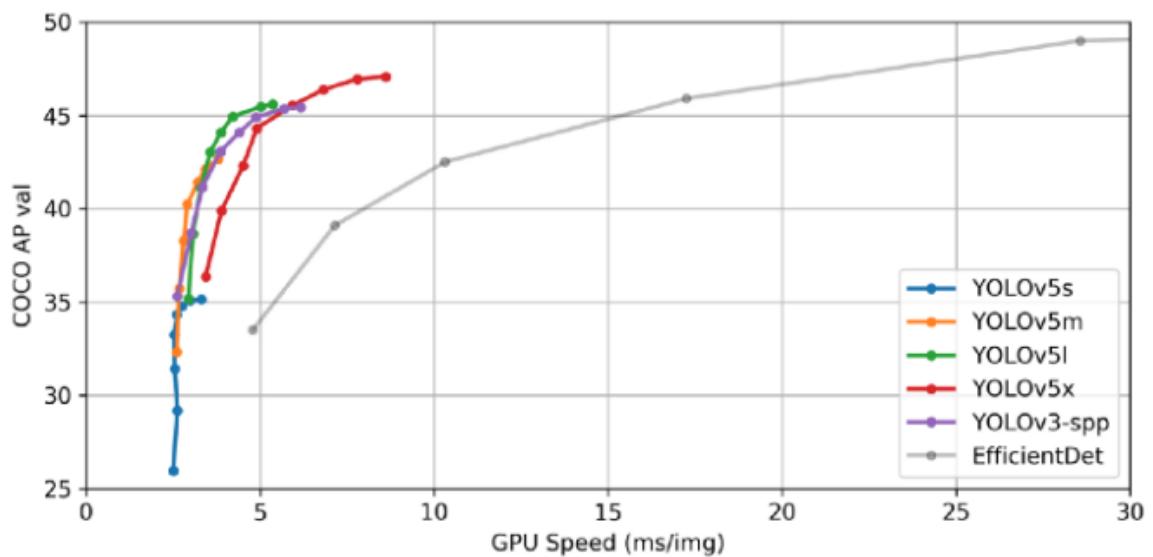
Hình 4.14 là giao diện màn hình chức năng "Store manager". Giao diện bao gồm thanh chọn dòng camera muốn theo dõi cùng màn hình hiển thị camera bên dưới. Ngoài giao diện còn có ba chức năng để người dùng chọn và tương tác với dòng camera.

4.3 Phát triển bài toán "Phát hiện người và hành động lấy đồ"

4.3.1 Bài toán phát hiện sản phẩm

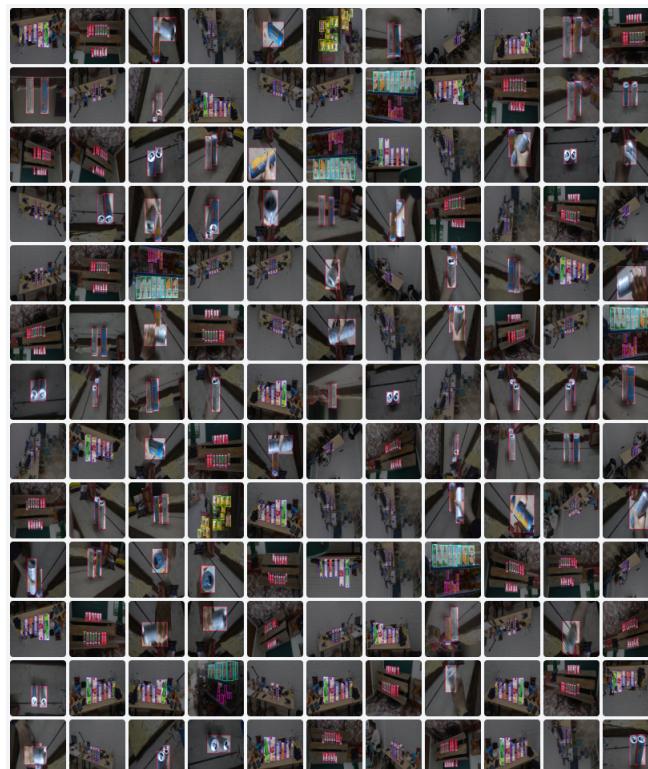
a, Mô hình sử dụng

Với bài toán này, hướng tiếp cận của tôi là tìm một mô hình có khả năng phát hiện và phân loại những sản phẩm. Sau khi tìm hiểu các mô hình thông qua internet, tôi đã chọn mô hình YOLOv5. YOLOv5 là phiên bản mới nhất trong những mô hình YOLO, nó được cải tiến để mang lại tốc độ nhanh hơn trong khi vẫn giữ được độ chính xác cao. YOLOv5 cũng có nhiều phiên bản khác nhau như: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x [33]. Trong đó, mô hình YOLOv5l đảm bảo độ chính xác tốt hơn YOLOv3 và có tốc độ xử lý nhanh hơn, kết quả thể được thể hiện ở hình 4.15, vì vậy tôi sẽ lựa chọn mô hình YOLOv5l cho bài toán của mình.

**Hình 4.15:** So sánh tốc độ của các mô hình YOLO

b, Bộ dữ liệu

Trong đồ án này, tôi đã thu thập bộ dữ liệu gồm 871 ảnh về năm sản phẩm: Lon nước, chai nước, Hộp sữa, hộp hình trụ và gói snack. Bộ dữ liệu được đẩy lên Roboflow để gán nhãn.

**Hình 4.16:** Bộ dữ liệu tự thu thập

c, Huấn luyện mô hình phát hiện sản phẩm

Tiếp theo, tôi sẽ trình bày về quá trình huấn luyện mô hình YOLOv5l. Để huấn luyện mô hình, tôi cài đặt mô hình với các thông số sau.

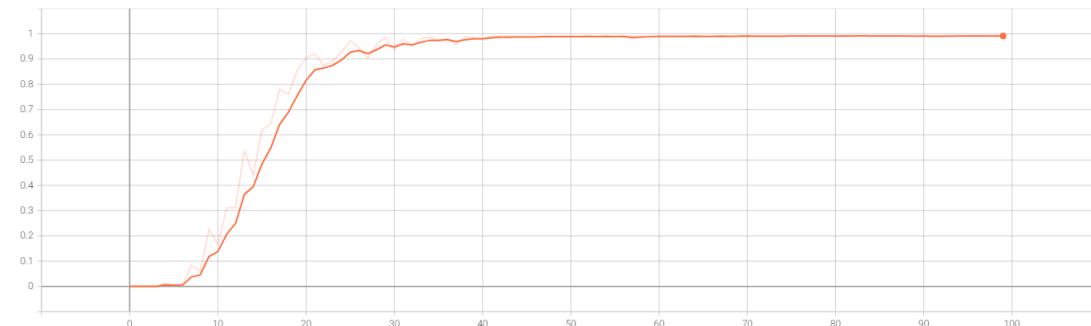
Với dữ liệu, kích thước ảnh đầu vào là 416. Các kỹ thuật tăng cường ảnh đầu vào gồm có: Xoay ảnh, lật ảnh và cắt ảnh.

Với mô hình, tôi dựa vào mẫu tham số của YOLOv5l và thay đổi tham số nc (num of class) thành 5 để phù hợp với bộ dữ liệu đầu vào để huấn luyện. Mô hình được huấn luyện với 100 epochs và batch size bằng 16.

Dưới đây là kết quả huấn luyện.



Hình 4.17: Giá trị hàm mất mát của mô hình xác định vật thể



Hình 4.18: Giá trị mAP của mô hình xác định vật thể

Hình 4.17 là đồ thị biểu thị giá trị hàm mất mát học trên bộ dữ liệu huấn luyện. Hình 4.18 là đồ thị biểu thị giá trị mAP trên bộ dữ liệu đánh giá. Có thể thấy từ vòng lặp 0 đến khoảng vòng lặp thứ 40, giá trị hàm mất mát giảm nhanh, nghĩa là mô hình học nhanh trong giai đoạn đó. Từ vòng lặp 40 trở đi, mô hình học chậm hơn và dựa vào giá trị mAP cao nhất, quá trình huấn luyện đưa ra kết quả tốt nhất.

4.3.2 Bài toán theo dõi sản phẩm và phát hiện hành động lấy đồ

Bài toán theo dõi sản phẩm có chức năng lưu trữ thông tin và kiểm tra tính hợp lý của vị trí tất cả các sản phẩm để quyết định nếu vật thể bị lấy ra khỏi gian hàng hay không. Hiện tại, OpenCV hỗ trợ một số mô hình theo dõi vật thể như

MOSSE Tracker, KCF Tracker, CSRT Tracker. Trong đó tôi lựa chọn sử dụng mô hình CSRT Tracker vì nó có độ chính xác cao nhất. Tuy nhiên CSRT Tracker vốn dĩ chỉ là thuật toán theo dõi một vật thể, vì thế mà ban đầu tôi đã sử dụng kèm theo hàm MultiTracker [34] của OpenCV. Nhưng khi có nhiều vật thể được theo dõi thì số bộ theo dõi tăng lên làm giảm đi FPS một cách đáng kể. Do đó tôi đã kết hợp sử dụng cùng với Multithread để tăng số luồng xử lý cùng một lúc, từ đó cải thiện FPS.

Vị trí của sản phẩm được theo dõi theo thời gian, từ đó tính toán ra vị trí trung bình của sản phẩm. Trong trường hợp sản phẩm bị thay đổi vị trí đột ngột, tôi sẽ so sánh vị trí của sản phẩm so với vị trí trung bình của nó, từ đó xác định sản phẩm có bị lấy đi hay không.

4.3.3 Bài toán nhận người qua dữ liệu khuôn mặt

a, Mô hình phát hiện khuôn mặt

Trong bài toán này, tôi sử dụng mạng MTCNN phát hiện khuôn mặt. Mạng MTCNN gồm 3 lớp mạng khác biệt, tương ứng cho 3 stage chính là P-Net, R-Net và O-Net. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng.

b, Mô hình trích xuất đặc trưng khuôn mặt

Khuôn mặt sau khi được phát hiện sẽ được đưa qua mô hình Facenet để trích xuất vector đặc trưng. Vector đặc trưng là một vector với kích thước cố định, đã được học trong quá trình huấn luyện và đại diện cho một tập các đặc trưng có trách nhiệm trong việc phân loại các đối tượng trong chiều không gian đã được biến đổi. Hiểu đơn giản, dựa vào khoảng cách của hai vector đặc trưng của hai khuôn mặt, ta có thể đưa ra kết luận hai khuôn mặt đó có cùng một người hay không.

c, Phương pháp nhận diện khuôn mặt

Qua thực nghiệm với bộ dữ liệu 20 người gồm từ 50 tới 100 ảnh mỗi người, tôi đã tìm ra khoảng cách giữa các vector đặc trưng để nhận diện khuôn mặt với độ chính xác 82,42%. Hai khuôn mặt được xác định là cùng một người khi khoảng cách giữa hai vector đặc trưng nhỏ hơn 5. Hai khuôn mặt được tính là khác nhau khi khoảng cách giữa hai vector đặc trưng lớn hơn 5,5.

CHƯƠNG 5. THỰC NGHIỆM VÀ ĐÁNG GIÁ

5.1 Xây dựng ứng dụng

5.1.1 Thư viện và công cụ sử dụng

Bảng 5.1: Danh sách thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual studio code v1.66.2	https://code.visualstudio.com/
IDE lập trình	Android studio v4.2.1	https://developer.android.com/studio
Quản lý source code	Github	https://github.com/
Thao tác API	Postman v8.10.0	https://www.postman.com/
Xây dựng API	FastAPI v0.79.0	https://fastapi.tiangolo.com/
Công khai API	Ngrok v3.0.6	https://ngrok.com/
Gán nhãn bộ dữ liệu	Roboflow	https://roboflow.com/
Đào tạo mô hình	Google colab	https://colab.research.google.com/
Quản lý môi trường lập trình	Anaconda v4.10.0	https://www.anaconda.com/
Đóng gói, triển khai code	Docker v4.10.1	https://www.docker.com/

Bảng 5.1 là các thư viện và công cụ cần thiết để phát triển ứng dụng đã được tôi liệt kê cùng với mục đích sử dụng của chúng.

5.1.2 Kết quả đạt được

Sau quá trình thực hiện đồ án, kết quả tôi đã xây dựng được một phương pháp để phát hiện hành động lấy đồ và mô hình phát hiện sản phẩm trong cửa hàng tiện lợi. Đồng thời, áp dụng thành công bài toán theo dõi đối tượng, phát hiện và nhận diện khuôn mặt để làm nêu ứng dụng giúp cảnh báo trường hợp kệ hàng hết sản phẩm và những tình huống có hành vi trộm cắp trong cửa hàng tiện lợi. Các kết quả được viết trong bảng 5.2, thông tin sản phẩm được liệt kê trong bảng 5.3.

Bảng 5.2: Danh sách kết quả đạt được

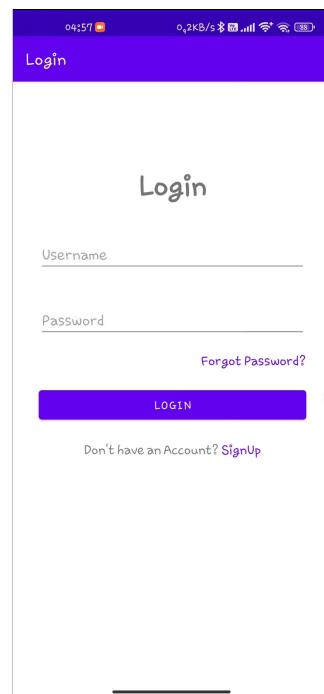
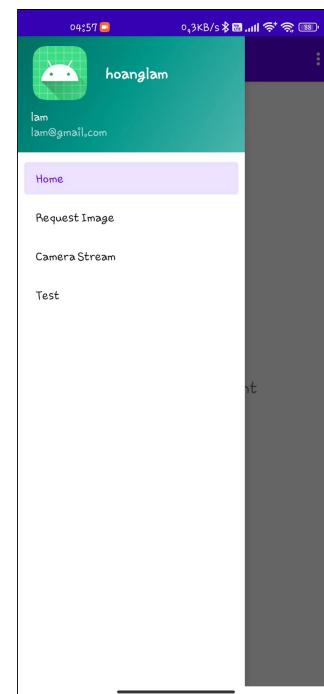
Sản phẩm đóng gói	Thành phần	Vai trò
Bài toán phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi	Một microservice chứa các API để tương tác với bài toán đặt trên máy chủ của hệ thống camera thông minh	Phát hiện, gửi thông báo về trường hợp hết hàng trên kệ và tình huống có hành vi trộm cắp
Ứng dụng Android	Ứng dụng chạy trên thiết bị Android đóng vai trò như máy khách trong hệ thống camera thông minh	Cung cấp bài toán phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi như một dịch vụ cho người dùng sử dụng

Bảng 5.3: Thông kê thông tin sản phẩm

Thông tin	Thông kê
Số dòng code bài toán phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi	1800 dòng
Mô hình phát hiện sản phẩm	93MB
Dung lượng file cài đặt ứng dụng Android	88,1MB

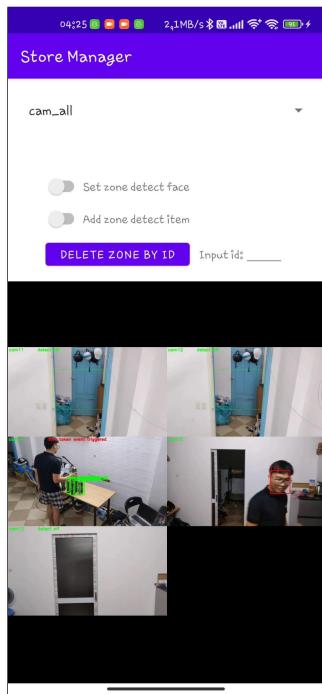
5.1.3 Minh họa các chức năng chính

a, Chức năng đăng nhập

**Hình 5.1:** Giao diện chức năng đăng nhập**Hình 5.2:** Giao diện menu sau khi đăng nhập

Hình 5.1 là giao diện đăng nhập. Trong giao diện này, người dùng sẽ nhập tài khoản và mật khẩu để đăng nhập. Nếu chưa có tài khoản, có thể ấn vào nút đăng ký để tạo tài khoản mới. Hình 5.2 là giao diện chọn chức năng của ứng dụng sau khi đăng nhập.

b, Chức năng "Store manager"



Hình 5.3: Giao diện chức năng "Store manager"

Hình 5.3 là giao diện chức năng "Store manager". Trong giao diện này, người chơi có thể chọn để xem các dòng camera mình mong muốn. Ngoài ra, người dùng có thể tương tác với các dòng camera thông qua chức năng trong giao diện này.

5.2 Kiểm thử

Để đánh giá độ hoàn thiện của ứng dụng, tôi đã sử dụng phương pháp kiểm thử hộp đen để tiến hành kiểm thử áp dụng với 2 chức năng chính của ứng dụng (i) đăng nhập (ii) xem dòng hình ảnh đã qua xử lý. Sau đó tiến hành kiểm thử đối với các yêu cầu phi chức năng. Dưới đây là kết quả kiểm thử.

5.2.1 Kiểm thử chức năng đăng nhập

Mô tả yêu cầu: Đối với chức năng đăng nhập, ứng dụng cần xử lý dữ liệu đầu vào được nhập. Nếu dữ liệu bỏ trống, thông tin nhập sai phải thông báo lỗi đến người dùng. Khi nhập đúng thông tin tài khoản, phải chuyển người dùng đến giao diện sau khi đăng nhập. Chi tiết tại bảng 5.4

Bảng 5.4: Kiểm thử chức năng đăng nhập

STT	Mục đích	Quy trình		Kết quả thực tế	Đánh giá
		Đầu vào	Đầu ra		
1	Kiểm tra bắt buộc nhập dữ liệu	Bỏ trống tài khoản hoặc mật khẩu	Thông báo yêu cầu nhập tất cả các trường	Thông báo yêu cầu nhập tất cả các trường	Đạt
2	Kiểm tra dữ liệu chưa ký tự đặc biệt	Nhập tài khoản hoặc mật khẩu chứa khoảng trống	Thông báo lỗi chứa ký tự đặc biệt	Thông báo lỗi chứa ký tự đặc biệt	Đạt
3	Kiểm tra sai dữ liệu	Nhập tài khoản hoặc mật khẩu không chính xác	Thông báo lỗi sai thông tin đăng nhập	Thông báo lỗi sai thông tin đăng nhập	Đạt
4	Kiểm tra dữ liệu chính xác	Nhập tài khoản và mật khẩu chính xác	Chuyển giao diện về màn hình sau khi đăng nhập	Chuyển giao diện về màn hình sau khi đăng nhập	Đạt

Kết quả cho thấy hệ thống đã đạt được mục tiêu yêu cầu người dùng cần đăng nhập để sử dụng các chức năng của ứng dụng.

5.2.2 Kiểm thử chức năng xem dòng camera đã qua xử lý

Đây là chức năng giúp cho người dùng có thể điều khiển các chức năng liên quan đến camera. Giúp người dùng xem dòng hình ảnh tương ứng với camera và chức năng đã chọn. Sau đây là nội dung kiểm thử. Mô tả yêu cầu: Đối với chức năng này, người dùng cần chọn camera và chức năng trước khi yêu cầu máy chủ trả về dòng hình ảnh. Nếu chưa hoàn tất các bước này, ứng dụng sẽ thông báo lỗi yêu cầu thao tác lại. Sau khi lựa chọn thành công, ứng dụng sẽ hiển thị dòng hình ảnh đã qua xử lý cho người dùng.

Bảng 5.5: Kiểm thử chức năng xem dòng camera đã qua xử lý

STT	Mục đích	Quy trình		Kết quả thực tế	Đánh giá
		Đầu vào	Đầu ra		
1	Kiểm tra đã hoàn tất các lựa chọn	Không lựa chọn camera hoặc chức năng xử lý	Thông báo yêu cầu hoàn tất mọi thao tác	Thông báo yêu cầu hoàn tất mọi thao tác	Đạt
2	Kiểm tra chức năng có phù hợp với camera	Chọn chức năng mà camera đã chọn không hỗ trợ	Thông báo camera không hỗ trợ chức năng vừa chọn	Thông báo camera không hỗ trợ chức năng vừa chọn	Đạt
3	Kiểm tra hoạt động của ứng dụng	Lựa chọn đúng camera và chức năng	Hiển thị dòng camera đã qua xử lý tương ứng với lựa chọn	Hiển thị dòng camera đã qua xử lý tương ứng với lựa chọn	Đạt

Kết quả thu được cho thấy ứng dụng đã có thể đáp ứng mục tiêu điều khiển chức năng camera và xem dòng hình ảnh đã qua xử lý được máy chủ trả về.

5.3 Triển khai

5.3.1 Chuẩn bị

a, Phần cứng

- Máy tính với CPU Intel® Xeon(R) X5650, 24 GB RAM và GPU NVIDIA GeForce GTX 1080 Ti.
- Điện thoại Android với CPU Qualcomm Snapdragon 870 5G, 8 GB RAM và GPU Adreno 650.

b, Chuẩn bị dữ liệu

Dữ liệu để kiểm thử bao gồm 5 video được phát liên tục để thay thế cho 5 dòng camera. Các video đó bao gồm: Video quay cửa vào cửa hàng, video quay cửa ra cửa hàng, video quay quầy thu ngân, video quay gian hàng từ phía trước và video quay gian hàng từ phía sau.

c, Cài đặt các thông số

Ứng dụng được cài đặt với một số tham số như dưới đây:

MTCNN

- thresholds: [0.4, 0.6, 0.6]
- min_face_size: 60
- keep_all: True

YOLOv5

- threshold: 0.65

Phát hiện chuyển động

- area_min: 1500
- area_max: 2000

Khác

- width_scale: 800
- vector_distance_min: 5
- vector_distance_max: 5.5
- delay_frame_object_detect: 600
- frame_item_empty: 1800

Ngoài ra, các mô hình đều được chạy trên CPU.

Các cài đặt của **MTCNN** có ý nghĩa như sau:

- thresholds: Thresholds chính là mức thresholds cho 3 lớp mạng P, R và O của mạng MTCNN.
- min_face_size: thiết đặt cho mô hình chỉ phát hiện khuôn mặt có kích thước lớn hơn 60.
- keep_all: keep_all để xác định việc chúng ta có phát hiện và trả về tất cả các mặt có thể trong bức hình hay không.

Các cài đặt của **YOLOv5** có ý nghĩa như sau:

- threshold: Là ngưỡng để xác định kết quả trả về của mô hình YOLOv5. Chỉ lấy những kết quả có độ chính xác lớn hơn threshold.

Các cài đặt của **Phát hiện chuyển động** có ý nghĩa như sau:

- Kết quả trả về khi so sánh hai khung hình liên tiếp có giá trị lớn hơn **area_min** và nhỏ hơn **area_max** thì sẽ được xác định là có chuyển động, ngược lại kết quả là không phát hiện thấy chuyển động.

Các tham số còn lại có ý nghĩa như sau:

- width_scale: Kích thước của khung hình sẽ được thay đổi cùng tỷ lệ thành khung hình có độ rộng width_scale trước khi đưa vào bài toán.
- vector_distance_min: Khi so sánh khoảng cách khoảng cách của hai vector đặc trưng của hai khuôn mặt, nếu khoảng cách đó nhỏ hơn **vector_distance_min** thì sẽ được xác định là cùng một người.
- vector_distance_max: Khi so sánh khoảng cách khoảng cách của hai vector đặc trưng của hai khuôn mặt, nếu khoảng cách đó lớn hơn **vector_distance_max** thì sẽ được xác định là một người mới.
- delay_frame_object_detect: Hai lần phát hiện vật phẩm bằng YOLOv5 sẽ cách nhau **delay_frame_object_detect** khung hình.
- frame_item_empty: Khi phát hiện không có sản phẩm trong vùng theo dõi sản phẩm quá **frame_item_empty** khung hình, hệ thống sẽ xác định là đã hết sản phẩm trong vùng đó.

d, Kịch bản thử nghiệm

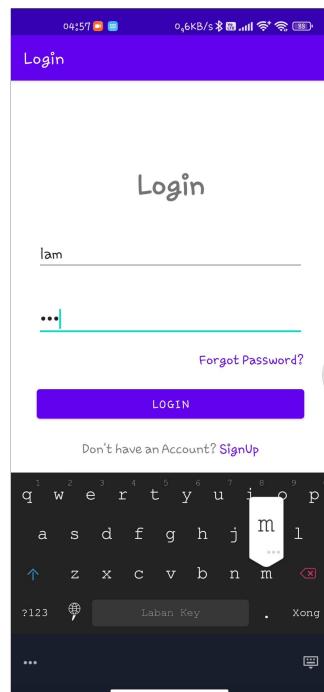
Kịch bản dưới đây là một chuỗi kết hợp các tình huống thực tế mà có thể gặp trong cửa hàng. Tôi sẽ mô phỏng lại các tình huống sau:

- Khách hàng bước vào cửa hàng.
- Khách hàng lấy đồ trong kệ hàng.

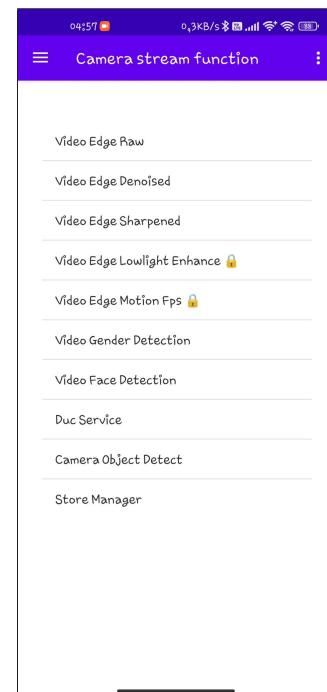
- Khách hàng qua quầy thu ngân để thanh toán.
- Khách hàng rời khỏi cửa hàng sau khi qua quầy thu ngân để thanh toán.
- Khách hàng rời khỏi cửa hàng nhưng chưa đi qua quầy thu ngân để thanh toán.
- Sản phẩm trên kệ hàng bị lấy hết.

5.3.2 Thực hiện thử nghiệm

a, Đăng nhập và lựa chọn sử dụng chức năng "Store manager"



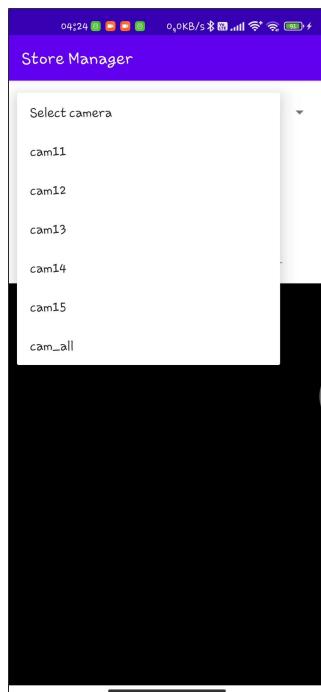
Hình 5.4: Đăng nhập



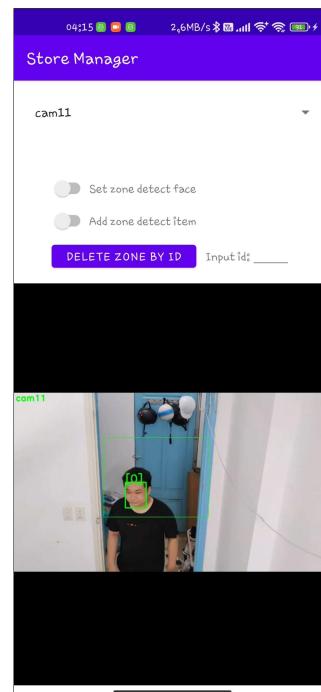
Hình 5.5: Chọn chức năng "Store manager"

Người dùng tiến hành đăng nhập vào ứng dụng bằng tài khoản và mật khẩu được cấp như hình 5.4. Việc đăng nhập diễn ra chính xác. Sau đó lựa chọn chức năng "Store manager" để sử dụng dịch vụ như hình 5.5. Màn hình đã chuyển tới trang của dịch vụ.

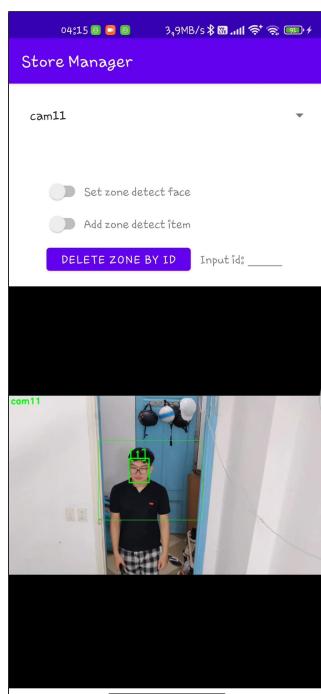
b, Lựa chọn camera đặt tại cửa vào để quan sát



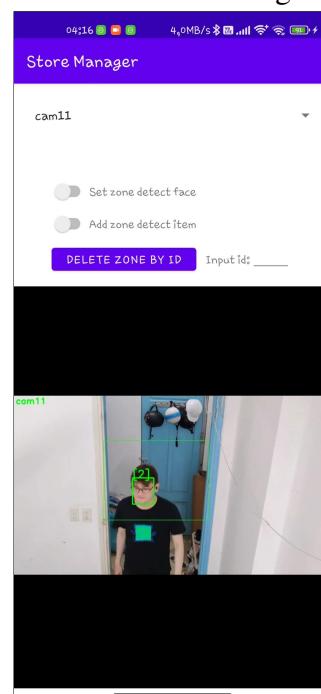
Hình 5.6: Chọn camera để quan sát



Hình 5.7: Khách hàng thứ nhất bước vào cửa hàng



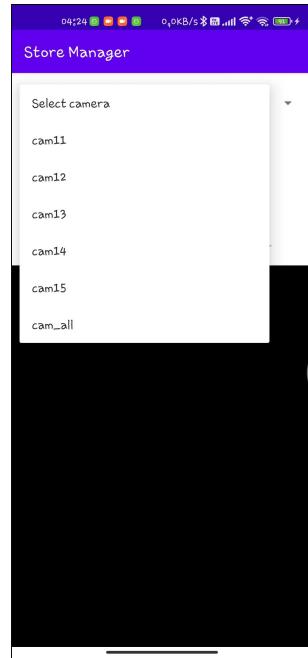
Hình 5.8: Khách hàng thứ hai bước vào cửa hàng



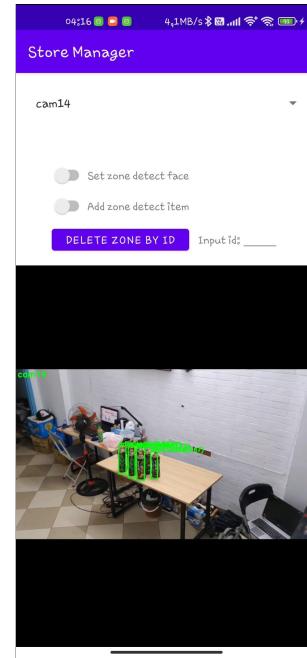
Hình 5.9: Khách hàng thứ ba bước vào cửa hàng

Sau khi tiến chọn camera như hình 5.6, dòng camera đã được hiển thị để tiến hành quan sát. Lúc này, có khách hàng tiến vào cửa hàng, hệ thống đã nhận diện đúng và phân biệt ba khách hàng bằng các ID khác nhau như trong hình 5.7, 5.8 và 5.9.

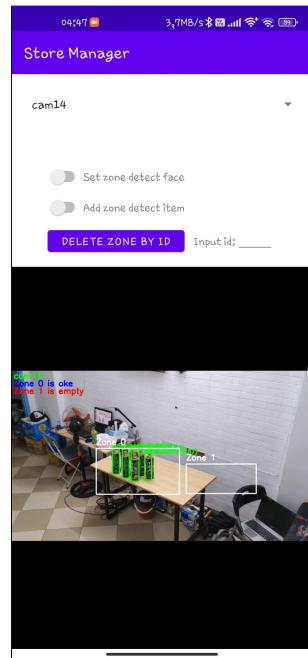
c, Lựa chọn camera đặt phía trước gian hàng để quan sát và thêm vùng theo dõi sản phẩm



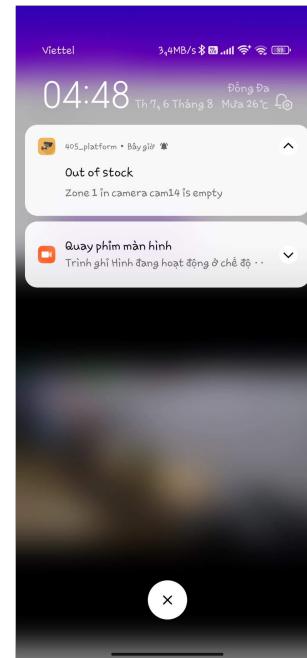
Hình 5.10: Chọn camera để quan sát



Hình 5.11: Chưa thêm vùng theo dõi sản phẩm



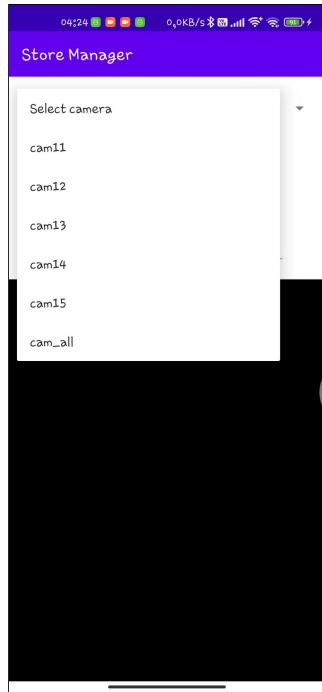
Hình 5.12: Đã thêm vùng theo dõi sản phẩm



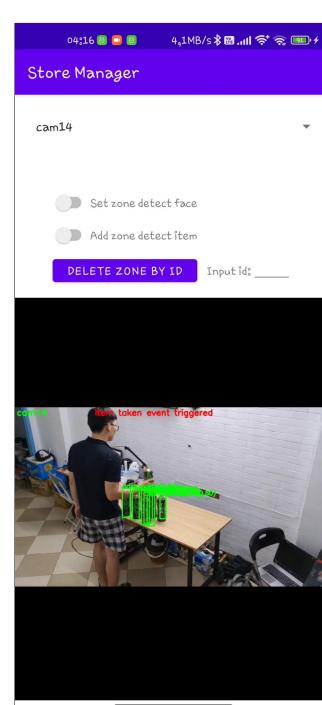
Hình 5.13: Nhận thông báo vùng theo dõi trống

Sau khi tiến chọn camera như hình 5.10, dòng camera đã được hiển thị để tiến hành quan sát. Lúc này, dòng cam hiển thị các sản phẩm nhưng chưa có vùng theo dõi cụ thể cho sản phẩm như trong hình 5.11. Tiếp hàng thêm vùng theo dõi sản phẩm, ta nhận được kết quả như trong hình 5.12. Khi phát hiện vùng theo dõi đã trống, ứng dụng sẽ nhận được thông báo như trong hình 5.13.

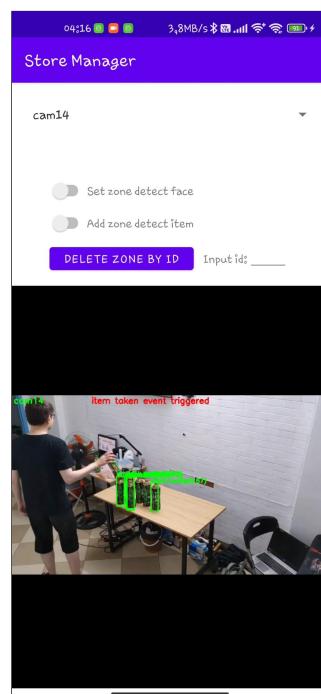
d, Lựa chọn camera đặt phía trước gian hàng để quan sát hành động lấy đồ



Hình 5.14: Chọn camera để quan sát



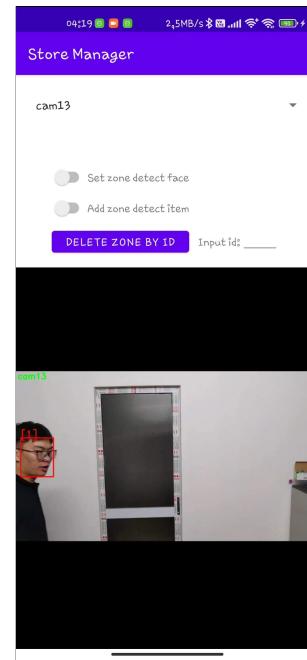
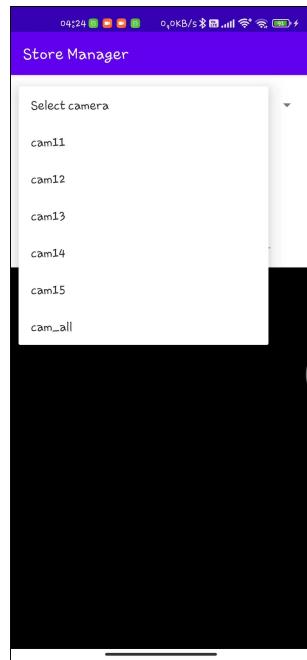
Hình 5.15: Khách hàng lấy sản phẩm khỏi kệ



Hình 5.16: Khách hàng lấy sản phẩm khỏi kệ

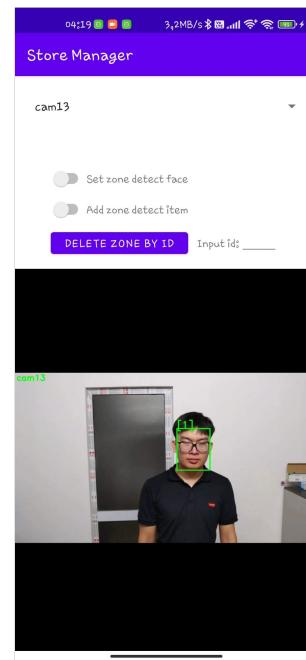
Sau khi tiến chọn camera như hình 5.14, dòng camera đã được hiển thị để tiến hành quan sát. Khi có khách hàng lấy sản phẩm khỏi kệ, hệ thống đã nhận biết được và hiển thị trên camera như hình 5.15 và 5.16.

e, Lựa chọn camera đặt tại quầy thu ngân để quan sát khách hàng thanh toán



Hình 5.17: Chọn camera để quan sát

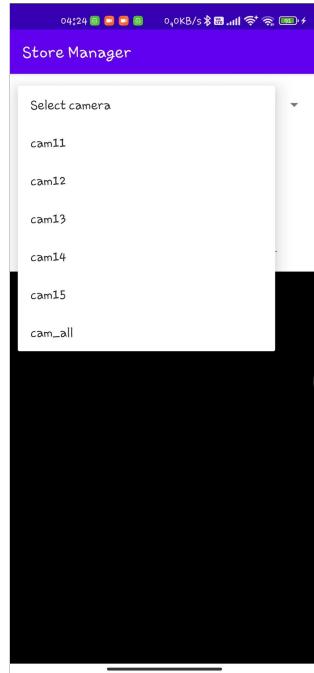
Hình 5.18: Khách hàng khi chưa thanh toán



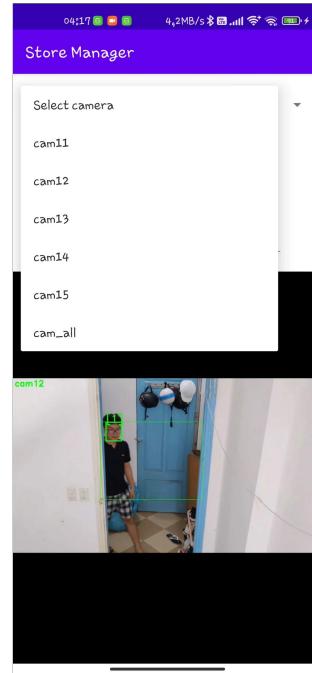
Hình 5.19: Khách hàng sau khi đứng tại quầy thu ngân để thanh toán

Sau khi tiến chọn camera như hình 5.17, dòng camera đã được hiển thị để tiến hành quan sát. Khi có khách hàng bước vào quầy thu ngân khi chưa thanh toán sản phẩm, camera sẽ hiển thị như hình 5.18. Sau khi khách hàng đứng tại quầy thu ngân để thanh toán, camera sẽ hiển thị như hình 5.19.

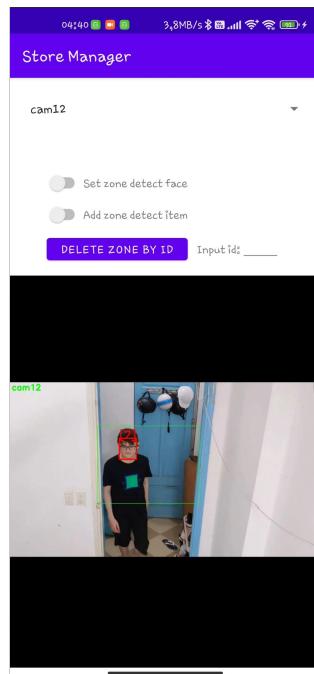
f, Lựa chọn camera đặt tại cửa ra để quan sát



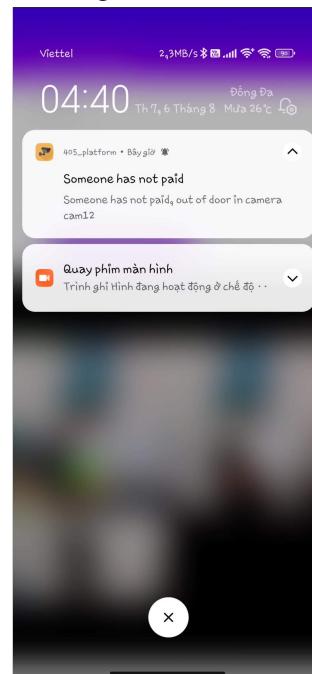
Hình 5.20: Chọn camera để quan sát



Hình 5.21: Khách hàng ra khỏi cửa hàng khi đã thanh toán



Hình 5.22: Khách hàng ra khỏi cửa hàng khi chưa thanh toán



Hình 5.23: Nhận thông báo có khách hàng chưa thanh toán rời khỏi cửa hàng

Sau khi tiến chọn camera như hình 5.20, dòng camera đã được hiển thị để tiến hành quan sát. Lúc này, khi có khách hàng ra khỏi cửa hàng, trên camera sẽ hiển thị xem khách hàng đó đã thanh toán chưa. Nếu đã thanh toán, camera sẽ hiển thị như hình 5.21. Còn nếu chưa thanh toán, camera sẽ hiển thị như hình 5.22, đồng thời ứng dụng sẽ nhận được thông báo như trong hình 5.23.

CHƯƠNG 6. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Dưới đây, tôi sẽ tóm tắt một vài đóng góp nổi bật của tôi trong đồ án này.

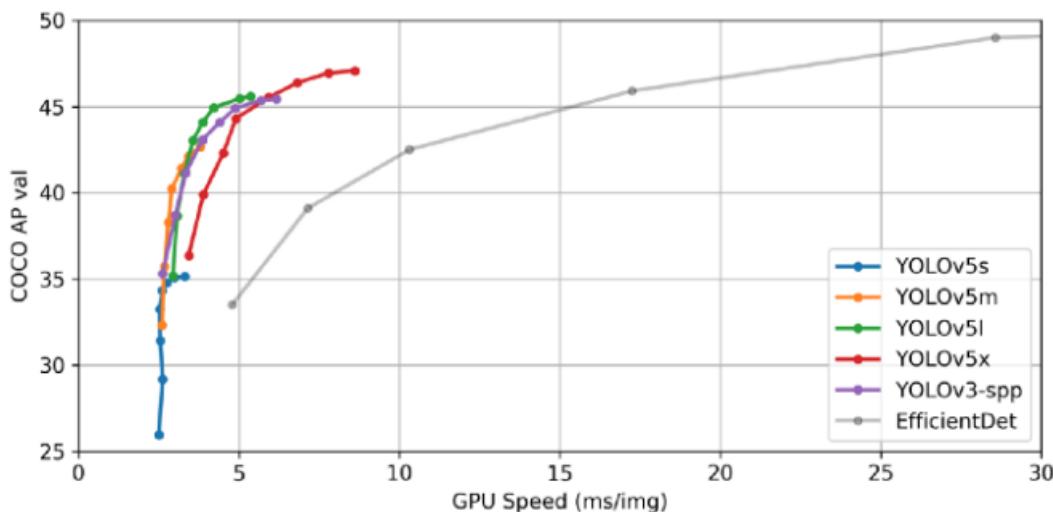
6.1 Mô hình phát hiện sản phẩm trong cửa hàng tiện lợi

6.1.1 Đặt vấn đề

Trong quá trình nghiên cứu đồ án, tôi đã tìm nhiều mô hình có khả năng phát hiện các sản phẩm trong cửa hàng tiện lợi cũng như mô hình đã được huấn luyện trước của chúng. Nhưng nhận thấy rằng mô hình và bộ dữ liệu hướng tới đều nhận diện sản phẩm khi được chụp ở góc nhìn của người mua hàng, điều đó dẫn tới việc mô hình không thể nhận diện sản phẩm qua góc nhìn của camera. Đồng thời bộ dữ liệu chỉ được đánh nhãn chung cho tất cả sản phẩm, nên mô hình chỉ phát hiện được những sản phẩm trên kệ hàng chứ không phân biệt được các sản phẩm với nhau. Để phục vụ cho ý tưởng bài toán sau này, tôi cần một mô hình thích hợp hơn để phát hiện các sản phẩm trong cửa hàng. Mô hình cần đảm bảo độ chính xác, tốc độ xử lý và có thể phát hiện và phân loại nhiều loại sản phẩm khác nhau.

6.1.2 Giải pháp

Để giải quyết bài toán trên, đầu tiên tôi đã đi tìm hiểu những mô hình phát hiện vật thể thông dụng hiện nay. Sau quá trình tìm hiểu, tôi lựa chọn mô hình YOLO vì sự phổ biến cũng và khả dĩ tiếp cận của nó. Cụ thể ở đây, tôi sử dụng mô hình YOLOv5. YOLOv5 là phiên bản mới nhất trong những mô hình YOLO, nó được cải tiến để mang lại tốc độ nhanh hơn trong khi vẫn giữ được độ chính xác cao. YOLOv5 cũng có nhiều phiên bản khác nhau như: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. Trong đó, mô hình YOLOv5l đảm bảo độ chính xác tốt hơn YOLOv3 và có tốc độ xử lý nhanh hơn, kết quả thể được thể hiện ở hình 6.1, vì vậy tôi sẽ lựa chọn mô hình YOLOv5l cho bài toán của mình.



Hình 6.1: So sánh tốc độ của các mô hình YOLO

6.1.3 Kết quả đạt được

Sau khi thử nghiệm với mô hình YOLOv5 bằng một số hình ảnh tự chụp, tôi nhận thấy YOLOv5 là một mô hình dễ tiếp cận, đảm bảo việc dễ dàng huấn luyện cũng như triển khai. Tuy độ chính xác khi phát hiện vật thể chưa cao nhưng mô hình cũng đã phân biệt được hai loại sản phẩm khác nhau. Đồng thời thời gian xử lý của mô hình đạt tốc độ 5 fps, phù hợp với mục đích mà tôi sử dụng mô hình này trong đồ án.

6.2 Xây dựng bộ dữ liệu huấn luyện mô hình phát hiện sản phẩm

6.2.1 Đặt vấn đề

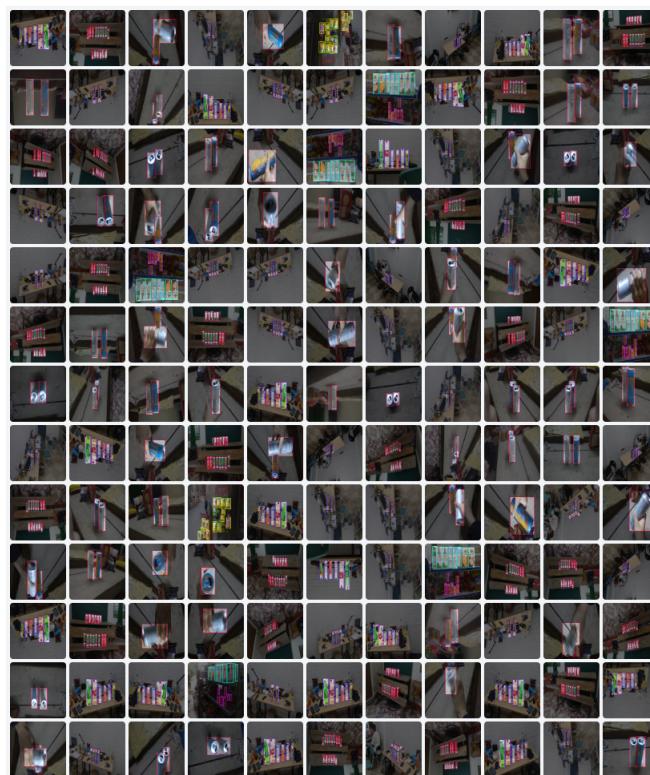
Để có thể huấn luyện mô hình phát hiện sản phẩm đạt độ chính xác cao cả trong quá trình thử nghiệm và thực nghiệm, thì một bộ dữ liệu tốt để huấn luyện và đánh giá là điều không thể thiếu. Bộ dữ liệu phải đảm bảo có số ảnh đủ lớn, đa dạng về sản phẩm, cùng với đó là góc chụp, ánh sáng và độ phân giải phải đa dạng để mô hình được huấn luyện trong nhiều môi trường. Hiện nay, trong quá trình tìm kiếm, tôi nhận thấy chưa có bộ dữ liệu nào đáp ứng được những tiêu chí vừa kể trên.

6.2.2 Giải pháp

Qua quá trình tìm kiếm để giải quyết vấn đề tôi đã nêu trên, tôi có tìm thấy một vài bộ dữ liệu về sản phẩm trong cửa hàng. Tuy nhiên những sản phẩm này không phù hợp với những tiêu chí đã nêu, đồng thời, chúng đều là những sản phẩm khó tìm thấy ở Việt Nam nên việc tiến hành kiểm thử sẽ gặp khó khăn. Đứng trước vấn đề này, tôi đã tự mình chuẩn bị bộ dữ liệu riêng. Bộ dữ liệu tôi chuẩn bị bao gồm 871 ảnh về năm sản phẩm: Lon nước, chai nước, Hộp sữa, hộp hình trụ và gói snack. Sau đó, bộ dữ liệu được đẩy lên Roboflow để gán nhãn.

6.2.3 Kết quả đạt được

Sau khi gán nhãn bộ dữ liệu đã chuẩn bị và áp dụng các kỹ thuật tăng cường ảnh đầu vào như: Xoay ảnh, lật ảnh và cắt ảnh. Tôi đã có bộ dữ liệu bao gồm 1800 ảnh huấn luyện, 174 ảnh xác thực chéo và 95 ảnh kiểm thử. Hình 6.2 mô tả một phần bộ dữ liệu mà tôi đã chuẩn bị



Hình 6.2: Bộ dữ liệu tự thu thập

6.3 Xây dựng mô hình nhận diện khách hàng thông qua khuôn mặt

6.3.1 Đặt vấn đề

Trong đồ án này, để phát hiện được những khách hàng ra vào cửa hàng, từ đó xác định ai là người có hành vi trộm cắp cũng như chưa thanh toán khi rời khỏi cửa hàng, tôi cần một mô hình có thể phát hiện và phân biệt khách hàng thông qua dữ liệu khuôn mặt. Vì bài toán áp dụng cho cửa hàng tiện lợi vừa và nhỏ nên tôi cần một mô hình nhỏ và nhẹ, có khả năng triển khai đơn giản. Đặc biệt, vì khách hàng ra vào cửa hàng là ngẫu nhiên, nên tôi cần một mô hình có thể nhận diện được khách hàng mà không qua huấn luyện trước.

6.3.2 Giải pháp

Qua quá trình tìm hiểu, tôi cần hai mô hình chính cho bài toán này. Đầu tiên là mô hình dùng để phát hiện khuôn mặt, thứ hai là mô hình dùng để trích xuất đặc trưng của khuôn mặt để đưa vào nhận diện.

Đối với mô hình phát hiện khuôn mặt, tôi lựa chọn mạng MTCNN. Mạng MTCNN gồm 3 lớp mạng khác biệt, tương ứng cho 3 stage chính là P-Net, R-Net và O-Net. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng.

Đối với bài toán trích xuất đặc trưng khuôn mặt, tôi lựa chọn mô hình Facenet. Khuôn mặt sau khi được phát hiện sẽ được đưa qua mô hình Facenet để trích xuất vector đặc trưng. Vector đặc trưng là một vector với kích thước cố định, đã được học trong quá trình huấn luyện và đại diện cho một tập các đặc trưng có trách nhiệm trong việc phân loại các đối tượng trong chiều không gian đã được biến đổi. Hiểu đơn giản, dựa vào khoảng cách của hai vector đặc trưng của hai khuôn mặt, ta có thể đưa ra kết luận hai khuôn mặt đó có cùng một người hay không.

6.3.3 Kết quả đạt được

Qua thực nghiệm với bộ dữ liệu 20 người gồm từ 50 tới 100 ảnh mỗi người, tôi đã tìm ra khoảng cách giữa các vector đặc trưng để nhận diện khuôn mặt với độ chính xác 82,42%. Hai khuôn mặt được xác định là cùng một người khi khoảng cách giữa hai vector đặc trưng nhỏ hơn 5. Hai khuôn mặt được tính là khác nhau khi khoảng cách giữa hai vector đặc trưng lớn hơn 5,5.

6.4 Xây dựng bài toán phát hiện hành động lấy đồ

6.4.1 Đặt vấn đề

Qua quá trình tìm hiểu, dự định ban đầu của tôi là sử dụng bài toán phát hiện khung xương, sau đó đưa vào huấn luyện để tạo ra mô hình có khả năng phát hiện hành vi lấy đồ trên kệ hàng. Tuy nhiên, việc tìm kiếm bộ dữ liệu về hành hình lấy đồ trên kệ hàng rất khó khăn. Vì đây là một bài toán lạ sau quá trình tìm kiếm, tôi không tìm thấy bộ dữ liệu nào phù hợp với mục tiêu đề ra. Vì vậy tôi cần một giải pháp khác giúp xác định hành vi lấy đồ trên kệ trong bài toán của mình.

6.4.2 Giải pháp

Nhận thấy, camera giám sát trong cửa hàng tiện lợi luôn được đặt cố định và sản phẩm đặt trên kệ cũng luôn có định cho đến khi có người lấy chúng khỏi kệ hàng. Vì lý do đó, tôi phát triển một mô đun theo dõi sản phẩm và vị trí của chúng theo thời gian. Bằng cách phát hiện vị trí sản phẩm có bị di chuyển hay không, tôi sẽ tính toán, so sánh độ di chuyển với kích thước của sản phẩm mà đưa ra kết quả sản phẩm có bị lấy đi hay không. Để theo dõi sản phẩm liên tục, tôi sử dụng mô hình CSRT Tracker được thư viện OpenCV hỗ trợ. Đầu vào của mô hình này chính là đầu ra của mô hình phát hiện sản phẩm mà tôi đã trình bày ở phần 6.1. Tôi lựa chọn sử dụng mô hình CSRT Tracker vì nó có độ chính xác cao. Tuy nhiên CSRT Tracker vốn dĩ chỉ là thuật toán theo dõi một vật thể, vì thế mà ban đầu tôi đã sử

dụng kèm theo hàm MultiTracker của OpenCV. Nhưng khi có nhiều vật thể được theo dõi thì số bộ theo dõi tăng lên làm giảm đi FPS một cách đáng kể. Do đó tôi đã kết hợp sử dụng cùng với Multithread để tăng số luồng xử lý cùng một lúc, từ đó cải thiện FPS.

6.4.3 Kết quả đạt được

Sau quá trình xây dựng bài toán, tôi đã xây dựng được một phương pháp phát hiện hành động lấy đồ trên kệ đồ đúng như mong muốn. Mô hình phát hiện hành động lấy đồ này đã nhận diện chính xác 17 lần trong 20 lần thử nghiệm trong điều kiện không bị che khuất.

CHƯƠNG 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1 Kết luận

Sau thời gian thực hiện đồ án dưới sự hướng dẫn của PSG. TS. Đặng Văn Chuyết và TS. Đặng Tuấn Linh, tôi đã thành công xây dựng bài toán "Phát hiện người và hành động lấy đồ trong cửa hàng tiện lợi" triển khai bài toán đó thành công trên thiết bị Android trong hệ thống camera thông minh mà chúng tôi phát triển nhằm mục đích thực nghiệm. Ứng dụng đã đạt được những mục tiêu đề ra:

- Cung cấp dịch vụ camera giám sát trong cửa hàng cùng với các chức năng thông minh.
- Thông báo cho người dùng trường hợp hết sản phẩm trên kệ hàng.
- Thông báo cho người dùng khi phát hiện hành vi rời khỏi cửa hàng mà không thanh toán.

Để sử dụng các chức năng quản lý cửa hàng, người dùng cần tiến hành đăng nhập và chọn chức năng "Store manager" tại phần dịch vụ về camera. Trong giao diện này, người dùng có thể chọn từng dòng camera muốn quan sát hoặc quan sát tất cả các dòng camera được liên kết cùng một lúc. Đối với chức năng thông báo khi hết hàng trên kệ, người dùng cần vẽ vùng theo dõi sản phẩm bằng cách tương tác với camera thông qua chức năng của ứng dụng. Đối với chức năng thông báo khi phát hiện hành vi rời khỏi cửa hàng mà không thanh toán, người dùng cần điều chỉnh vùng nhận diện khách hàng sao cho phù hợp với hình ảnh cửa ra vào hiển thị trên camera.

Trong quá trình thực hiện đồ án, tôi đã học được nhiều công nghệ và kiến thức mới. Tôi cũng hiểu thêm về quy trình thiết kế, phát triển ứng dụng, cách xử lý khi gặp khó khăn trong phát triển.

7.2 Hướng phát triển

Vì thời gian phát triển hạn chế, nên đồ án này vẫn chưa được hoàn thiện ở một số tình năng. Từ đánh giá của bản thân và góp ý từ những người có chuyên môn xung quanh, tôi nhận thấy ứng dụng cần có những cải thiện như sau:

- Cải thiện tốc độ xử lý, từ đó đảm bảo không phát hiện thiếu trường hợp vì tốc độ xử lý chậm.
- Cải thiện thuật toán phát hiện hành động lầy đồ, tăng độ chính xác cũng như đảm bảo phát hiện trong nhiều góc quan sát hơn.
- Xử lý tốt hơn trong trường hợp nhiều khách hàng cùng thực hiện hành động một lúc.
- Cải thiện bộ dữ liệu, giúp mô hình phát hiện, theo dõi được nhiều sản phẩm hơn.
- Cải thiện khả năng phản hồi của ứng dụng khi người dùng tương tác.

Trên đây là một số những cải thiện quan trọng mà tôi nghĩ có thể tiếp tục phát triển trong tương lai trước khi có thể đưa ra sử dụng trong thực tế. Tuy hiện nay ứng dụng còn gặp hạn chế về kỹ thuật và thiết kế, nhưng tôi mong rằng trong tương lai không xa ứng dụng có thể được ứng dụng trong thực tế, giúp nhiều người có thể sử dụng camera thông minh để hỗ trợ quản lý những công việc trong cuộc sống.

TÀI LIỆU THAM KHẢO

- [1] E. Stolterman and A. C. Fors, “Information technology and the good life,” in *Information systems research*, Springer, 2004, pp. 687–692.
- [2] J. Shabbir and T. Anwer, “Artificial intelligence and its role in near future,” *arXiv preprint arXiv:1804.01396*, 2018.
- [3] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [4] V. Tsakanikas and T. Dagiuklas, “Video surveillance systems-current status and future trends,” *Computers & Electrical Engineering*, vol. 70, pp. 736–753, 2018.
- [5] *Top 10 cctv camera brands in the world*. [Online]. Available: <https://blog.bizvibe.com/blog/security-and-safety/top-10-cctv-camera-brands-world> (visited on 08/05/2022).
- [6] *Yolov5 documentation*. [Online]. Available: <https://docs.ultralytics.com/> (visited on 08/05/2022).
- [7] *Yolo*. [Online]. Available: <https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html> (visited on 08/05/2022).
- [8] *Csrt tracker documentation*. [Online]. Available: <https://docs.ultralytics.com/> (visited on 08/05/2022).
- [9] K. Z. Z. Z. L. Y. Qiao, “Joint face detection and alignment using multi-task cascaded convolutional networks,” 2016. DOI: 10.1109/LSP.2016.2603342.
- [10] F. S. D. K. J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” pp. 815–823, 2015. DOI: 10.1109/CVPR.2015.7298682.
- [11] *Java documentation*. [Online]. Available: <https://docs.oracle.com/en/java/> (visited on 08/05/2022).
- [12] *10 thiết bị chống trộm hàng đầu mà các cửa hàng bán lẻ nên có*. [Online]. Available: <https://congtuanninhgia.re/10-thiet-bi-chong-trom-hang-dau-macac-cua-hang-ban-le-nen-co/> (visited on 08/05/2022).
- [13] *Object tracking*. [Online]. Available: <https://viso.ai/deep-learning/object-tracking/> (visited on 08/05/2022).
- [14] *The opencv reference manual*, 4.5.3, OpenCV, 2021. [Online]. Available: <https://docs.opencv.org/4.5.3/> (visited on 08/05/2022).

- [15] D. Merkel, “Docker: Lightweight linux containers for consistent development and deployment,” *Linux journal*, vol. 2014, no. 239, p. 2, 2014. [Online]. Available: <https://docs.docker.com/> (visited on 08/05/2022).
- [16] M. Abadi, A. Agarwal, P. Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: <https://www.tensorflow.org/> (visited on 08/05/2022).
- [17] R. Ierusalimschy, *Programming in lua*. Roberto Ierusalimschy, 2006.
- [18] Epfl. [Online]. Available: <https://www.epfl.ch/en/> (visited on 08/05/2022).
- [19] A. Paszke, S. Gross, F. Massa, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf> (visited on 08/05/2022).
- [20] Meta Inc., *Meta ai research*, Feb. 29, 2020. [Online]. Available: <https://ai.facebook.com/> (visited on 08/05/2022).
- [21] *Fastapi documentation*. [Online]. Available: <https://fastapi.tiangolo.com/> (visited on 08/05/2022).
- [22] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [23] *Flask documentation*, Flask, 2021. [Online]. Available: <https://flask.palletsprojects.com/en/2.0.x/> (visited on 08/05/2022).
- [24] *Django documentation*. [Online]. Available: <https://www.djangoproject.com/> (visited on 08/05/2022).
- [25] ISO, *ISO/IEC 14882:2011 Information technology—Programming languages — C++*. Geneva, Switzerland: International Organization for Standardization, Feb. 2012, 1338 (est.) [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=50372 (visited on 08/05/2022).
- [26] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: <https://doi.org/10.1038/s41586-020-2649-2> (visited on 08/05/2022).
- [27] *Firebase*. [Online]. Available: <https://firebase.google.com/> (visited on 08/05/2022).

-
- [28] *Firebase cloud messaging documentation*. [Online]. Available: <https://firebase.google.com/docs/cloud-messaging/> (visited on 08/05/2022).
 - [29] *Android httpurlconnection documentation*. [Online]. Available: <https://developer.android.com/reference/java/net/HttpURLConnection/> (visited on 08/05/2022).
 - [30] VideoLan, *Vlc media player*, 2006. [Online]. Available: <https://www.videolan.org/vlc/index.html> (visited on 08/05/2022).
 - [31] A. Berson, *Client/server architecture*. McGraw-Hill, Inc., 1996.
 - [32] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice architecture: aligning principles, practices, and culture*. " O'Reilly Media, Inc.", 2016.
 - [33] *Yolov5 new version - improvements and evaluation*. [Online]. Available: <https://blog.roboflow.com/yolov5-improvements-and-evaluation/> (visited on 08/05/2022).
 - [34] *Multitracker*. [Online]. Available: https://docs.opencv.org/3.4/d8/d77/classcv_1_1MultiTracker.html (visited on 08/05/2022).