ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN

ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

GIẢI PHÁP TỰ ĐỘNG HOÁ XUẤT NHẬP KHO HÀNG

Người hướng dẫn: **TS. NGUYỄN HOÀNG MAI**

Sinh viên thực hiện:

1. NGUYỄN VĂN KHIÊM – MSSV: 105200367 – LỚP: 20TDHCLC1

2. CHU VĂN HIẾU – MSSV: 105200360 – LỚP: 20THDCLC1

Đà Nẵng, 7/2023

TÓM TẮT

Tên đề tài : Giải pháp tự động hoá xuất nhập kho hàng

Sinh viên thực hiện : Nguyễn Văn Khiêm

Chu Văn Hiếu

Số thẻ SV : 105200367

105200360

Lớp : 20TDHCLC1

Nội dung đề tài : Đề tài "Giải pháp tự động hoá xuất nhập kho hàng" tập trung vào việc xây dựng hệ thống định vị xe nâng trong kho hàng thông minh, sử dụng camera theo dõi vị trí xe và hàng hoá theo thời gian thực.

Hệ thống sử dụng mô hình thị giác máy tính với YOLOv8 để nhận diện xe nâng từ hình ảnh của các camera, từ đó thu thập các toạ độ pixel của xe và chuyển đổi sang toạ độ thực tế bằng phép biến đổi phối cảnh thông qua ma trận đồng nhất.

Ngoài ra, giao diện điều khiển và sử lý ảnh được xây dựng bằng ngôn ngữ Python kết hợp với PyQt5, cho phép kết nối với cơ sở dữ liệu MySQL để lưu trữ thông tin hàng hoá qua mã QR

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Nguyễn Văn Khiêm	105200367	20TDHCLC1	Kỹ thuật điều khiển và Tự động hoá
2	Chu Văn Hiếu	105200360	20TDHCLC1	Kỹ thuật điều khiển và Tự động hoá

1. Tên đề tài đồ án:

Giải pháp tự động hoá xuất nhập kho hàng

2. *Đề tài thuộc diện: ☐Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*

3. Các số liệu và dữ liệu ban đầu:

……..……..……...
...……
…..…….…..……..……

4. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Chu Văn Hiếu	Tìm hiểu đề tài, nghiên cứu tài liệu tham khảo, các bài viết liên quan đến đề tài Thực nghiệm mô hình Viết báo cáo Lắp đặt mô hình
2	Nguyễn Văn Khiêm	

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Chu Văn Hiếu	Xác định phương pháp xác định toạ độ xe nâng, và phương pháp lưu trữ thông tin hàng hoá Lựa chọn các phần mềm sử dụng trong dự án Nghiên cứu, sử dụng phần mềm MySQL để lưu thông tin hàng hoá
2	Nguyễn Văn Khiêm	Xác định quy trình vận hành hệ thống

		Thiết kế, tính toán các thông số cho nhà kho Huấn luyện hệ thống nhận diện xe nâng Thiết kế giao diện hiển thị, tương tác giữa người dùng và hệ thống Lập trình python cho hệ thống

5. *Họ và tên người hướng dẫn:* **TS.Nguyễn Hoàng Mai**
6. *Ngày giao nhiệm vụ đồ án: 25/2/2025*
7. *Ngày hoàn thành đồ án: 15/06/2025*

Đà Nẵng, ngày tháng 6 năm 2025

Trưởng Bộ môn Tự động hóa **Người hướng dẫn**

TS. Giáp Quang Huy

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ……

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

(Phiếu dành cho người hướng dẫn/sinh viên)

Họ tên sinh viên: _____ Số thẻ SV : _____

Tên đề tài ĐATN: _____

Họ tên người HD: _____ Đơn vị: _____

Tuần	**Ngày**	**Khối lượng**		**GVHD ký tên**
		đã thực hiện (%)	**tiếp tục thực hiện (%)**	
1				
2				
3				
4		Duyệt lần 1: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN □ Không tiếp tục thực hiện ĐATN □		
5				
6				
7				
8		Duyệt lần 2: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN □ Không tiếp tục thực hiện ĐATN □		
9				

10				
11				
12		Duyệt lần 3: Đánh giá khối lượng hoàn thành _____ % : Được tiếp tục làm ĐATN □ Không tiếp tục thực hiện ĐATN □		
13				
14				
15				

LỜI NÓI ĐẦU

Ngày nay, việc áp dụng công nghệ hiện đại vào việc vận hành và quản lý kho hàng thông minh đã và đang trở nên xu thế phổ biến khắp thế giới. Nhằm đáp ứng như cầu thực tiễn trong lĩnh vực tự động hoá , nhóm em đã thực hiện đề tài *"Giải pháp tự động hoá xuất nhập kho hàng"* nhằm mục tiêu phát triển một hệ thống hỗ trợ theo dõi và định vị phương tiện di chuyển trong nhà kho cùng với các mặt hàng được lưu trữ, dựa trên việc xử lý hình ảnh từ camera.

Trong quá trình thực hiện, nhóm đã kết hợp kiến thức về thị giác máy tính, kỹ thuật biến đổi phối cảnh và ngôn ngữ lập trình Python để thiết kế hệ thống. Với kiến thức còn hạn chế, nhóm không tránh khỏi những thiếu sót. Kính mong quý thầy cô góp ý để chúng em được hoàn thiện hơn về mặt kiến thức cũng như kỹ năng.

Chúng em xin gửi lời cảm ơn đến thầy Nguyễn Hoàng Mai đã tận tình hướng dẫn, giúp đỡ nhóm em trong quá trình thực hiện đồ án này. Đồng thời chúng em cũng gửi lời cảm ơn đến quý thầy cô bộ môn Tự Động Hoá nói riêng và tất cả các thầy cô trong khoa Điện nói chung, đã trang bị cho chúng em những kiến thức cần thiết và hữu ích trong suốt những năm học qua

Chúng em xin chân thành cảm ơn !

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Kính gửi hội đồng bảo về đồ án tốt nghiệp ngành Kỹ thuật và Tự Động Hoá. Nhóm chúng em gồm 2 thành viên là Nguyễn Văn Khiêm, mã số sinh viên: 105200367, và Chu Văn Hiếu, mã số sinh viên: 105200360. Chúng em cam đoan đề tài "Giải pháp tự động hoá xuất nhập kho hàng" được thực hiện dưới sự hướng dẫn và giám sát của thầy Nguyễn Hoàng Mai. Tất cả nội dung và kết quả của đồ án này hoàn toàn trung thực và không sao chép từ bất kỳ công trình nghiên cứu nào khác. Nếu bị phát hiện vi phạm trong liêm chính học thuật, chúng em xin chịu toàn bộ trách nhiệm.

Sinh viên thực hiện

MỤC LỤC

DANH SÁCH HÌNH ẢNH

MỞ ĐẦU

Ở đề tài này, nhóm chúng em muốn xây dựng hệ thống định vị chính xác vị trí của xe nâng trong nhà kho cũng như các hàng hoá bằng cách sử dụng công nghệ thị giác máy tính và hệ thống camera. Trong đề tài sẽ sử dụng 4 camera đặt ở 4 góc nhà kho để quan sát toàn bộ khu vực kho và tính toán ra toạ độ thực của xe nâng, ngoài ra với mỗi kệ hàng sẽ có một camera đặt ở phía trước để theo dõi hàng hoá.

Nhóm đặt mục tiêu xây dựng thuật toán tính toán toạ độ thực của xe bằng biến đổi phối cảnh, sử dụng YOLOv8 để nhận diện vị trí xe trong các hình ảnh từ camera, ngoài ra hiển thị toạ độ, dữ liệu của xe nâng và hàng hoá qua giao diện giám sát thân thiện với người dùng. Đồng thời tích hợp kết nối cơ sở dữ liệu để lưu thông tin hàng hoá

Đối tượng nghiên và phạm vi nghiên cứu : Hệ thống định vị và theo dõi xe nâng trong môi trường kho hàng dệt may. Chỉ xử lý một xe đã được huấn luyện sẵn từ YOLOv8 trong mỗi khung hình, camera đặt ở vị trí cố định, không di chuyển

Các phương pháp dùng để nghiên cứu dự án này bao gồm: Phân tích lý thuyết, đưa ra giải pháp, áp dụng giải pháp cho nhà kho thực tế, xây dựng mô hình thực tế thu nhỏ và mô phỏng.

Cấu trúc của đồ án này gồm các chương chính như sau :

- Chương 1 : Tổng quan về đề tài
- Chương 2 : Phương pháp xác định toạ độ xe nâng và lưu trữ hàng hoá
- Chương 3 : Thiết kế hệ thống tự động hoá cho kho hàng dệt may
- Chương 4 : Mô phỏng và thực nghiệm

CHƯƠNG 1 : TỔNG QUAN VỀ ĐỀ TÀI

1.1 Lý do chọn đề tài :

Ngày nay, với sự phát triển của khoa học kỹ thuật, ngành điều khiển và tự động hoá đang đóng vai trò rất quan trọng trong nhiều lĩnh vực với những ưu điểm như tăng năng suất hiệu quả trong công việc, giảm chi phí lao động, cải thiện độ chính xác,… Vì vậy, chúng ta cần nắm bắt sử dụng nó một cách hiệu quả nhằm góp phần vào sự nghiệp phát triển khoa học kỹ thuật của đất nước

Hệ thống quản lý và lưu kho tự động trong nhiều doanh nghiệp ở Việt Nam vẫn còn phụ thuộc nhiều vào các quy trình thủ công. Quá trình xuất nhập hàng hoá chủ yếu được ghi chép bằng tay hoặc sử dụng các công cụ đơn giản như Excel, từ đó dễ dẫn đến việc hàng hoá bị thất lạc cũng như khó kiểm soát. Ngoài ra, cũng không có hệ thống để giám sát vị trí xe hàng, khiến cho việc biết chính xác toạ độ xe và hàng mà xe đang chở trở nên khó khăn, từ đó không thể tối ưu hoá quãng đường di chuyển. Ví dụ cụ thể ở công ty cổ phần dệt may Hoà Thọ - Đà Nẵng, việc lưu trữ và theo dõi hàng hoá cũng như vị trí của xe nâng hoàn toàn dựa vào quá trình thủ công. Và việc di chuyển xe nâng trong nhà kho không được sắp xếp tổ chức chặt chẽ sẽ gây cản trở trong việc tối ưu hoá vận chuyển cũng như đảm bảo an toàn cho nhân viên tại công ty.

Hình 1. 1 Kho lưu trữ hàng hoá của nhà máy sợi

Hình 1. 2 Dùng xe nâng để di chuyển hàng hoá

Các doanh nghiệp hiện đại ngày càng đòi hỏi sự chính xác và tốc độ cũng như hiệu quả trong quản lý kho hàng . Việc mắc lỗi trong quá trí di chuyển, giám sát kho hàng sẽ gây ra nhiều tổn thất cho chi phí. Đồng thời việc quản lý một cách thủ công cũng có thể gây ra thất thoát và thiếu minh bạch trong quá trình hoạt động. Khi phải xác định vị trí của từng kiện hàng bằng các phương pháp thủ công truyền thống, thì sẽ mất nhiều thời gian để có thể tìm và xác định vị trí của kiện hàng cần lấy, hơn nữa, việc này sẽ làm chậm quá trình xuất nhập kho , dễ gây thất lạc hàng hoá và nhầm lẫn trong qua trình xử lý hàng. Ngoài ra, việc tiết kiệm điện năng, giảm nhân công là nhu cầu rất cần thiết cho nhiều nhà máy, xí nghiệp hiện nay, nên việc áp dụng hệ thống lưu kho tự động đang là yếu tố được chú trọng, nhắm dễ dàng quản lý hàng hoá và chi tiêu

Ở một số quốc gia trên thế giới, họ đã áp dụng nhiều phương pháp hiện đại như dùng robot hoặc drone để vận chuyển hàng hoá, trí tuệ nhân tạo và dữ liệu lớn,… Tuy nhiên những phương pháp này đòi hỏi chi phí rất cao và nhiều thời gian nghiên cứu, rất khó để áp dụng cho tình trạng ở Việt Nam hiện tại.

Hình 1. 3 Phương pháp lưu kho thông minh bằng robot

Chính vì vậy, nhóm em đã lựa chọn đề tài "Giải pháp tự động hoá xuất nhập kho hàng" để giúp giải quyết các vấn đề đã kể trên. Đây là đề tài yêu cầu hiểu biết về các lĩnh vực quan trọng trong xã hội hiện đại như thiết kế hệ thống, phân tích dữ liệu và lập trình điều khiển, có sự kết hợp của phần cứng lẫn phần mềm, giúp giải quyết được vấn đề thực tế bằng việc sử dụng công nghệ mới. Ngoài ra, đây cũng là đề tài có tính ứng dụng cao, có thể triển khai từ mô hình nhỏ đến nhà kho thực tế.

1.2 Nội dung thực hiện :

1.2.1 Giải pháp cho vấn đề lưu và xuất nhập kho :

Ở dự án này, nhóm chúng em sẽ xây dựng mô hình nhà kho, sau đó sử dụng 4 camera cố định ở 4 góc của nhà kho để quan sát và theo dõi vị trí của xe nâng và tính toán toạ độ. Đồng thời sẽ có thêm 1 camera để quét mã của hàng hoá, sau đó thông tin của hàng (mã hàng, xuất xứ…) sẽ được lưu trữ vào phần mềm cơ sở dữ liệu, và một camera giám sát kho hàng. Tất cả những dữ kiện trên sẽ được hiển thị qua một màn hình giám sát.

- Ưu điểm của việc sử dụng 4 camera để giám sát xe nâng :
  - Sử dụng 4 camera sẽ cho tầm nhìn bao quát, không có điểm mù
  - Tính toán ra chính xác toạ độ của xe nâng bằng việc tổng hợp toạ đồ từ hình ảnh các camera
  - Hạn chế tình trạng không nhìn thấy xe trên màn hình giám sát
  - Nếu bị lỗi một camera, hệ thống vẫn có thể hoạt động dựa trên 3 camera còn lại

- Ưu điểm của việc giám sát hàng hoá
  - Kiểm soát hàng tồn kho chính xác : Giúp biết chính xác số lượng hàng hoá còn trong kho. Tránh thiếu hụt hoặc thất thoát hàng.
  - Tối ưu hoá việc lưu trữ và di chuyển : Việc sắp xếp hàng hoá hợp lý giúp tăng hiệu suất quản lý xuất/nhập kho. Giảm thời gian tìm kiếm hàng thủ công
  - Theo giõi hàng hoá theo thời gian thực
  - Tăng độ bảo mật : Việc giám sát hàng hoá giúp dễ dàng phát hiện và truy vết hàng bị mất, sai vị trí. Ngoài ra còn có thể ghi lại lịch sử thao tác
  - Dễ dàng tích hợp với các hệ thống khác
  - Tiết kiệm chi phí : Giảm tổn thất chi phí do sai sót thủ công gây ra, đồng thời nâng cao năng suất lao động của nhân viên

1.2.2 Tính đổi mới của hệ thống :

- Tự động hoá giám sát và quản lý kho bằng camera
  - Không cần đến sức người để nhập thông tin và kiểm tra hàng hoá
  - Việc lưu dữ liệu vào hệ quản trị hoàn toàn tự động
- Kết hợp công nghệ thị giác máy tính và định vị không cần cảm biến phần cứng
  - Hệ thống sử dụng thuật toán AI (YOLOv8) và camera góc rộng để theo dõi vị trí xe nâng mà không cần gắn cảm biến
  - Sử dụng camera tính toán toạ độ là phương pháp linh hoạt, chính xác và không tốn quá nhiều chi phí
- Tích hợp nhiều công nghệ mã nguồn mở hiện đại
  - Sử dụng các thư viện tiên tiến như YOLOv8 (nhận diện vật thể), OpenCV (nhận diện ảnh, đọc mã QR), PyQt5 (giao diện), MySQL(cơ sở dữ liệu)
  - Giải pháp này không bị phụ thuộc vào một thiết bị riêng biệt, từ đó dễ mở rộng và triển khai

1.2.3 Các phương pháp để phân loại sản phẩm

- Phương pháp 1 : Dùng mã vạch 1D (Barcode)

Mã vạch 1D là một hệ thống đại diện cho dữ liệu bằng cách sử dụng một chuỗi các dấu vạch và khoảng trống có kích thước khác nhau. Thông thường, chúng được in trên các sản phẩm và có thể được quét bằng máy quét mã vạch. Mã vạch 1D cho phép thông tin về sản phẩm, như giá cả, nguồn gốc, và thông tin quản lý tồn kho, được tự động hóa và theo dõi một cách nhanh chóng và chính xác.

Cấu tạo cơ bản của mã vạch 1D :

- Mã vạch lưu dữ liệu bằng cách mã hoá thông tin thành các vạch đen và các khoảng trắng
- Mỗi độ rộng khác nhau giữa vạch và khoảng trắng tương ứng với chữ cái hoặc ký tự số
- Dùng các thiết bị quét mã như máy quét laser, camera,… để đọc mã vạch
- Có ký hiệu giúp máy quét biết vị trí ban đầu và kết thúc của mã
- Dãy số hoặc chữ bên dưới là phần hiển thị cho người đọc trong trường hợp cần đối chiếu hoặc nhập tay

Hình 1. 4 Mã vạch 1D

Ưu điểm của mã vạch 1D :

- Quét mã vạch rất nhanh, nên từ đó giúp tăng tốc độ xử lý hàng hoá
- Máy quét đọc chính xác dữ liệu được mã hoá, hạn chế sai sót tự việc nhập thủ công
- Chi phí rẻ : In mã vạch rất rẻ và có thể in trên nhiều loại giấy khác nhau như nhãn gián, bao bì, giấy thường,…
- Có thể tích hợp với nhiều phần mềm khác nhau
- Dễ sử dụng : Chỉ cần đưa máy quét vào mã là có thể đọc được thông tin

- Được sử dụng rộng rãi nên dễ kết nối và chia sẽ dữ liệu

Nhược điểm của mã vạch 1D :

- Lưu trữ dữ liệu bị hạn chế : Các loại mã vạch 1D chỉ chứa được một lượng dữ liệu rất nhỏ (thường dưới 30 ký tự). Vì vậy không phù hợp nếu muốn lưu nhiều thông tin của sản phẩm
- Dễ bị lỗi khi quét : Nếu mã vạch bị trầy xước, hoặc gặp lỗi kích thước khi in thì máy sẽ không quét được
- Cần đọc theo hướng cố định : Với mã vạch 1D, máy quét cần đọc theo chiều ngang đúng hướng. Dễ gặp lỗi nếu mã bị xoay lệch
- Bảo mật thấp : Dữ liệu trong mã vạch thường không được mã hoá hoặc bảo vệ. Nên chỉ cần có máy quét thì bất cứ ai cũng có thể đọc thông tin hàng hoá
- Không phù hợp trong môi trường khắc nghiệt : Ở môi trường có độ ẩm cao hoặc nhiều bụi, mã vạch sẽ dễ bị bong tróc

- Phương pháp 2 : Dùng mã QR (mã vạch 2D) :

Mã QR (Quick Response Code) là một dạng mã vạch 2D (hai chiều) được phát triển để mã hóa và lưu trữ thông tin nhiều hơn so với mã vạch 1D truyền thống. QR Code có thể chứa dữ liệu phức tạp như văn bản, URL, thông tin liên hệ, mã sản phẩm, và thậm chí là dữ liệu nhị phân (file nhỏ).

Cấu tạo của mã QR :

- Ba ô vuông lớn ở ba góc (góc trên bên phải, góc trên bên trái, góc dưới bên trái) đóng vai trò làm vùng định vị giúp máy quét xác định vị trí và hướng của mã QR, đảm bảo mã có thể quét được chính xác dù có góc lệch
- Vùng dữ liệu là khu vực các ô vuông nhỏ trong mã QR, nơi chứa dữ liệu của hàng hoá. Các ô này có giá trị 1 (trắng) hoặc 0 (đen) với nhiều kích thước khác nhau
- Vũng tĩnh là khoảng trống xung quanh mã QR giúp máy quét nhận diện mã QR mà không bị nhiễu bởi mỏi trường xung quanh.
- Mã QR dùng thuật toán sửa lỗi để bảo vệ dữ liệu khỏi việc bị hỏng trong quá trình quét

- Các ô vuông nhỏ thường xuất hiện ở mã QR có kích thước lớn giúp cải thiện độ chính xác khi quét mã ở các góc

Hình 1. 5 Mã QR

Ưu điểm của mã QR :

- Lưu trữ dữ liệu lớn hơn nhiều so với mã vạch 1D : Có thể chứa đến 7000 chữ số hoặc 4000 ký tự. Có thể mã hoá văn bản, URL hoặc thậm chí các tài liệu và video
- Quét nhanh và dễ dàng : Có thể dùng điện thoại thông minh hoặc máy quét để quét
- Không yêu cầu phải chỉnh theo một hướng nhất định khi quét vì mã QR có thể quét được từ mọi góc độ
- Có khả năng sửa lỗi cao bằng mã sửa lỗi (giúp mã vẫn có thể đọc được dù bị hỏng một phần).
- Đa dạng ứng dụng : Mã QR có thể dùng để liên kết đến các website, video, ngoài ra còn có thể dùng để thanh toán di động và dùng làm thẻ liên hệ
- Chi phí rẻ : In mã QR rất rẻ và dễ dàng
- Dễ tạo và chia sẻ mã QR : Có thể tạo mã QR bằng nhiều công cụ miễn phí và chia sẻ qua nhiều hình thức khác nhau
- Tính bảo mật cao hơn mã vạch 1D : Có thể bảo vệ mã QR bằng mật khẩu hoặc mã hoá thông tin nếu cần

Nhược điểm của mã QR :

- Phụ thuộc vào kết nối internet nếu mã QR chứa đường liên kết đến website. Nếu không có internet thì sẽ không truy cập được thông tin mã QR cung cấp
- Dễ lừa đảo : Người dùng có thể bị lừa đảo hoặc tấn công mạng nếu truy cập vào các mã QR chưa thông tin nhạy cảm hoặc những website giả mạo
- Không hỗ trợ dữ liệu quá phức tạp : Dù có thể lưu trữ dữ liệu lớn, mã QR có thể gặp khó khăn nếu dữ liệu quá dài.
- Mất thẩm mỹ : Trong một số ứng dụng, mã QR có thể làm mất tính thẩm mỹ ở những mặt hàng cao cấp được thiết kế bao bì đẹp mắt
- Nếu mã QR chứa dữ liệu tĩnh như URL, nếu địa chỉ thay đổi thì cần phải tạo lại mã QR mới và phân phát lại mã mới này

➔ Sau khi đánh giá những tiêu chí trên, nhóm chúng em đã quyết định sử dụng mã QR để quản lý hàng hoá vì những ưu điểm vượt trội so với mã vạch 1D (barcode)[1]

1.2.4 Các phương pháp lưu trữ thông tin hàng hoá :

- Phương pháp 1 : Lưu trữ bằng file cục bộ

File cục bộ là tệp tin được lưu trữ trên thiết bị điện tử của người dùng thay vì lưu trên đám may hay một máy chủ từ xa. Ví dụ về file cục bộ là file Word, Excel,…

Phương pháp lưu trữ bằng file cục bộ là phương pháp lưu trữ dữ liệu trực tiếp vào ổ đĩa của máy tính hoặc thiết bị đang chạy chương trình, thay vì lưu vào cơ sở dữ liệu hoặc hệ thống lưu trữ đám mây. Dữ liệu có thể được lưu dưới nhiều định dạng khác nhau như: .txt, .csv, .json, .xml, .xlsx[2]

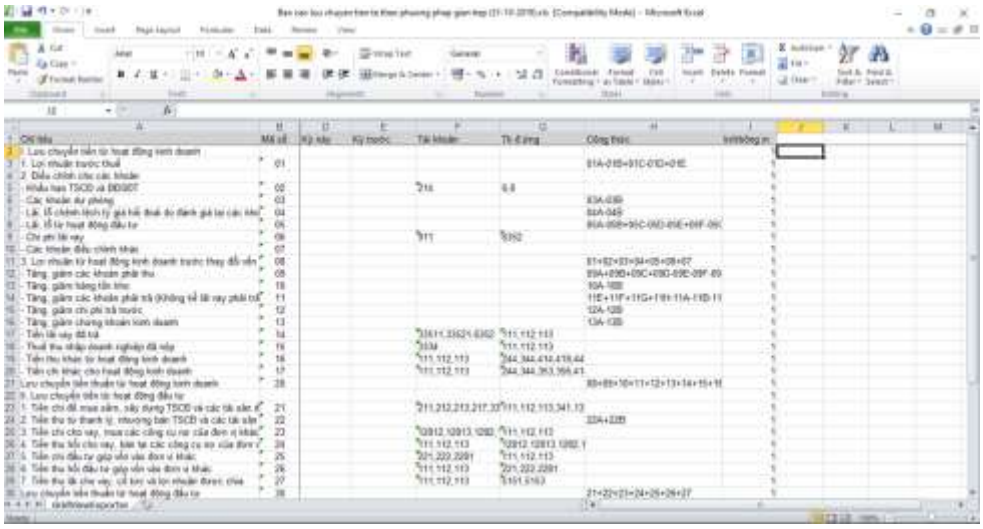
Ưu điểm của lưu trữ bằng file cục bộ :

- Dễ triển khai, không cần cài đặt các phần mềm phức tạp hay hệ quản trị cơ sở dữ liệu
- Phù hợp cho người mới bắt đầu
- Dễ sao lưu và chia sẻ với người dùng khác
- Dễ đọc, kiểm tra và chỉnh sửa. Ngoài ra có thể thao tác thủ công một cách đơn giản nếu cần thiết

- Không cần kết nối internet : Tất cả dữ liệu trong file cục bộ có thể dùng hoàn toàn offline
- Phù hợp để làm mô hình nhỏ và thử nghiệm
- Có thể thay đổi cấu trúc file mà không ảnh hưởng nhiều
- Không tốn phí : Dữ liệu được lưu trữ trên máy tính cá nhân hoàn toàn miễn phí. Cũng không cần đăng ký hay mua tài khoản.

Nhược điểm của lưu trữ bằng file cục bộ :

- Không phù hợp với hệ thống thực tế : Khi cần xử lý nhiều thông tin sẽ khiến quá trình bị chậm và dễ gặp lỗi
- Khó truy vấn, tìm kiếm và cập nhật dữ liệu
- Khó quản lý : Không có khả năng kiểm soát dữ liệu bị trùng và sai định dạng
- Khó mở rộng : Việc nâng cấp file thành hệ thống quản lý chuyên nghiệp khi hệ thống phát triển là phức tạp và tốn công sức
- Dễ xảy ra lỗi mất dữ liệu nếu không cẩn thận
- Tính bảo mật thấp : File có thể truy cập và chỉnh sửa dễ dàng bất cứ lúc nào. Không có mã hoá dữ liệu



Hình 1. 6 Phương pháp lưu trữ bằng file cục bộ

- Phương pháp 2 : Lưu trữ bằng đám mây

Cloud, hay còn gọi là điện toán đám mây, là mô hình điện toán sử dụng công nghệ máy tính và phát triển dựa vào mạng Internet. Ở mô hình điện toán

này, mọi khả năng liên quan đến công nghệ thông tin đều được cung cấp dưới dạng các "dịch vụ", cho phép người sử dụng truy cập các dịch vụ công nghệ từ một nhà cung cấp nào đó "trong đám mây" mà không cần phải có các kiến thức, kinh nghiệm về công nghệ đó, cũng như không cần quan tâm đến các cơ sở hạ tầng phục vụ công nghệ đó.

Hình 1. 7 Lưu trữu bằng đám mây

Lưu trữ bằng đám mây là hình thức lưu trữ dữ liệu trên các máy chủ từ xa (thường của bên thứ ba như Google, Amazon, Microsoft...), thay vì lưu trong ổ cứng máy tính cá nhân. Người dùng có thể truy cập, lưu, cập nhật, chia sẻ dữ liệu từ bất kỳ đâu, miễn là có kết nối internet.[3]

Ưu điểm của lưu trữ bằng đám mây :

- Dễ dàng truy cập : Vì dữ liệu được lưu online nên người dùng có thể truy cập mọi lúc mọi nơi, miễn là có kết nối internet
- Hỗ trợ nhiều người dùng cùng lúc : Cho phép nhiều người dùng truy cập, chỉnh sửa dữ liệu đồng thời
- Tự động sao lưu và phục hồi : Hệ thống đám mây có chức năng backup tự động, giảm nguy cơ mất dữ liệu khi thiết bị gặp sự cố
- Bảo mật cao : Hỗ trợ mã hoá dự liệu, xác thực người dùng. An toàn hơn so với phương pháp lưu file cục bộ
- Khả năng mở rộng linh hoạt : Có thể mở rộng dung lượng và hiệu năng theo yêu cầu. Phù hợp với cả hệ thống lớn và nhỏ.
- Kết nối dễ dàng với các thiết bị điện tử

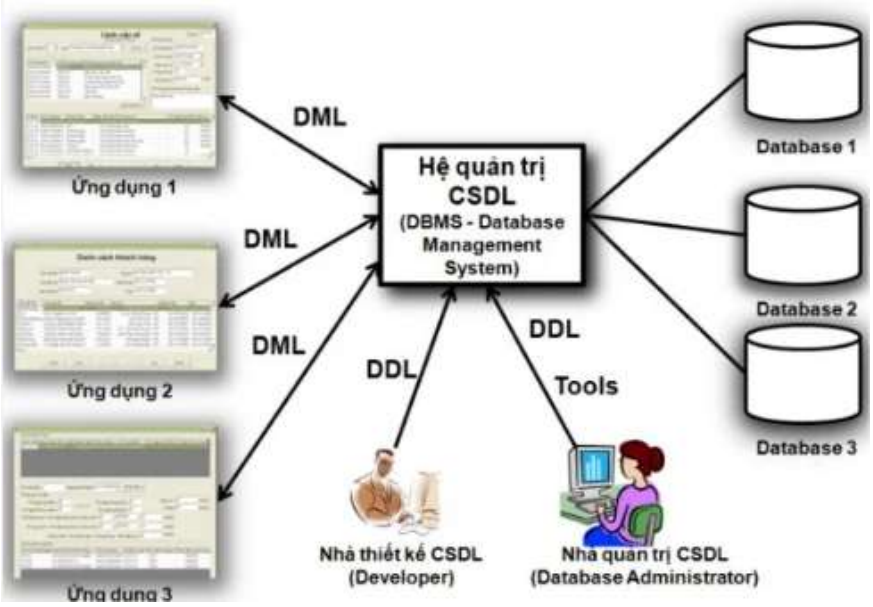
Nhược điểm của lưu trữ bằng đám mây :

- Phụ thuộc vào kết nối internet : Không có mạng thì sẽ không truy cập và chỉnh sửa được dữ liệu
- Chi phí tăng theo nhu cầu : Dịch vụ đám mây thường chỉ miễn phí với dung lượng nhỏ, nếu làm hệ thống dung lượng lớn thì sẽ tốn kém hơn
- Quyền riêng tư không đảm bảo : Dữ liệu được lưu ở bên thứ ba, nên dễ có nguy cơ bị rò rỉ thông tin. Ngoài ra độ tin cậy cũng phụ thuộc vào nhà cung cấp
- Công đoạn đăng ký tài khoản phức tạp

- Phương pháp 3 : Lưu trữ bằng phần mềm cơ sở dữ liệu

Cơ sở dữ liệu (Database) là một tập hợp các dữ liệu có tổ chức liên quan đến nhau, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

Là cách lưu trữ và quản lý dữ liệu bằng các hệ thống có tổ chức, có khả năng truy vấn, cập nhật, và kiểm soát dữ liệu một cách hiệu quả. Đây là cách lưu trữ phổ biến và chuyên nghiệp nhất trong các hệ thống phần mềm hiện nay, từ website, phần mềm quản lý đến hệ thống doanh nghiệp.[4]



Hình 1. 8 Lưu trữ bằng phần mềm cơ sở dữ liệu

Ưu điểm của lưu trữ bằng phần mềm cơ sở dữ liệu :

- Dễ kiểm soát dữ liệu và dễ mở rộng : Dữ liệu được tổ chức theo bảng (SQL) hoặc dạng tài liệu (NoSQL), giúp truy vấn, thống kê dữ liệu dễ dàng. Phù hợp với mô hình hàng hoá có nhiều dữ liệu
- Đảm bảo tính toàn vẹn và chính xác : Giảm lỗi dữ liệu bị trùng, thiếu hoặc sai định dạng
- Dễ truy xuất : Có thể truy vấn, tìm file hoặc lọc file dễ dàng theo nhiều tiêu chí bằng câu lệnh SQL
- Dễ tích hợp với nhiều ngôn ngữ lập trình (python, C#, java,…)
- Hỗ trợ nhiều người dùng đồng thời : Cho phép nhiều người sử dụng cùng lúc mà không bị xung đột dữ liệu
- Bảo mật tốt : Có thể phân quyền người dùng, hỗ trợ mã hoá và kiểm soát truy cập. Dữ liệu được sao lưu định kỳ. Giảm nguy cơ file bị lạc hoặc vô tình bị xoá khi lưu trên máy
- Sao lưu và phục hồi dữ liệu đơn giản : Toàn bộ dữ liệu đều nằm trong CSDL, nên chỉ cần backup CSDL là không lo về việc bị mất dữ liệu
- Dễ mở rộng và tích hợp với các phần mềm quản lý kho, website,…
- Chi phí rẻ : Nhiều phần mềm cơ sở dữ liệu có phiên bản miễn phí cho người dùng những dự án nhỏ có thể sử dụng

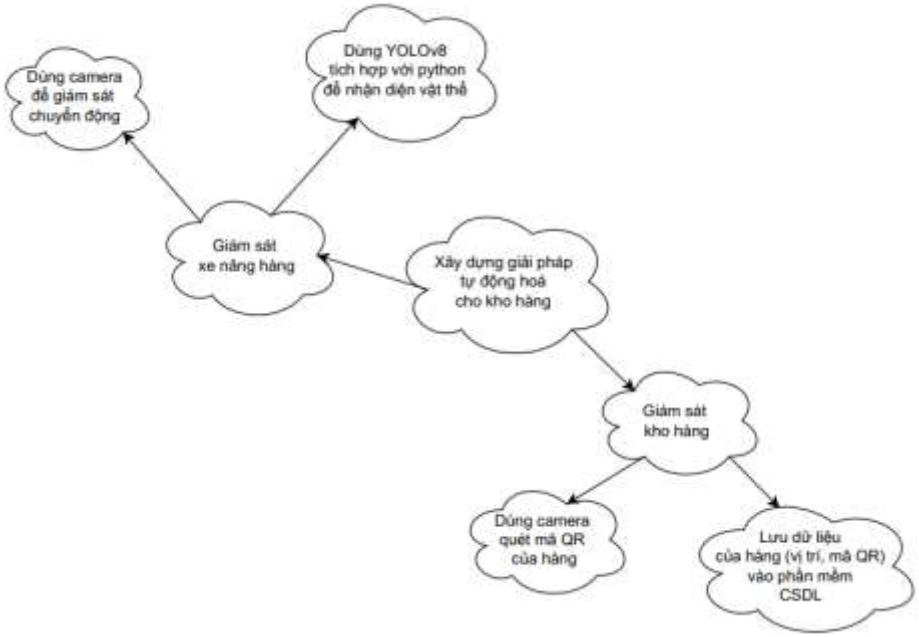
Nhược điểm của lưu trữ bằng phần mềm cơ sở dữ liệu

- Phức tạp khi triển khai ban đầu : Cần cài đặt và cấu hình ban đầu
- Tốn tài nguyên hệ thống hơn so với việc dùng file cục bộ
- Cần sao lưu dữ liệu định kỳ, nếu không có thể mất dữ liệu do lỗi hệ thống hoặc phần mềm
- Với hệ thống lớn, cần server mạnh hoặc hosting

➔ Từ những tiêu chí của các phương pháp trên, nhóm chúng em lựa chọn phương pháp lưu trữ thông tin hàng hoá bằng phần mềm cơ sở dữ liệu vì những ưu điểm vượt trội so với các phương pháp còn lại.

1.2.4 Tổng quan về hệ thống được đề xuất :

Sơ đồ tổng quát hệ thống của đề tài được phác thảo như hình 1.9



Hình 1. 9 Sơ đồ tổng quát của hệ thống

Nguyên lý chung của hệ thống :

- Hệ thống sử dụng 4 camera để theo dõi xe nâng, dùng YOLOv8 để nhận diện xe, sau đó tính toán toạ độ của xe khi xe di chuyển
- Sử dụng camera để quét mã QR được in trên hàng để nhận biết loại hàng
- Thông tin hàng hoá (vị trí, mã QR) và lịch sử xuất nhập kho được lưu vào phần mềm cơ sở dữ liệu
- Tất cả những dữ liệu trên được hiển thị trên một màn hình giám sát. Có thể dùng thiết bị điện tử để theo dõi

1.3 Kết quả dự kiến :

Kết quả mong muốn đạt được của nhóm khi thực hiện dự án này như sau :

- Xây dựng thành công mô hình dùng camera để xác định toạ độ của xe nâng trên mặt phẳng : Hệ thông sẽ sử dụng 4 camera đặt ở 4 góc để theo dõi vị trí xe nâng, đồng thời tính toán ra toạ độ của xe theo thời gian thực
- Nhận diện xe nâng từ nhiều góc nhìn khác nhau, không có điểm mù : Việc sử dụng đồng thời 4 camera giúp tránh việc không theo dõi được xe ở một vài góc nhất định
- Theo dõi vị trí lưu trữ và lịch sử xuất nhập kho hàng bằng phần mềm cơ sở dữ liệu MySQL.

- Tối ưu hoá không gian lưu trữ hàng hoá. Sắp xếp hàng hoá một cách hợp lý bằng việc phân loại hàng với sự hỗ trợ của mã QR, giúp tiết kiệm không gian kho, từ đó có thể lưu nhiều mặt hàng hơn.
- Tạo giao diện dễ dùng và đơn giản, hiển thị đầy đủ thông tin cần thiết để giám sát kho hàng
- Tăng độ chính xác : Việc dùng mã QR sẽ giúp giảm sai sót xảy ra trong quá trình hoạt động, đồng thời giúp hệ thống làm việc nhanh và chính xác hơn
- Tối ưu hoá chi phí cho dự án : Hệ thống hoạt động tự động sẽ rút ngắn thời gian của quá trình lưu kho và giảm thiểu chi phí gây ra do sai sót. Ngoài ra, hệ thống cũng giúp hạn chế tối đa việc thất thoát hàng hoá.

1.4 Kết luận :

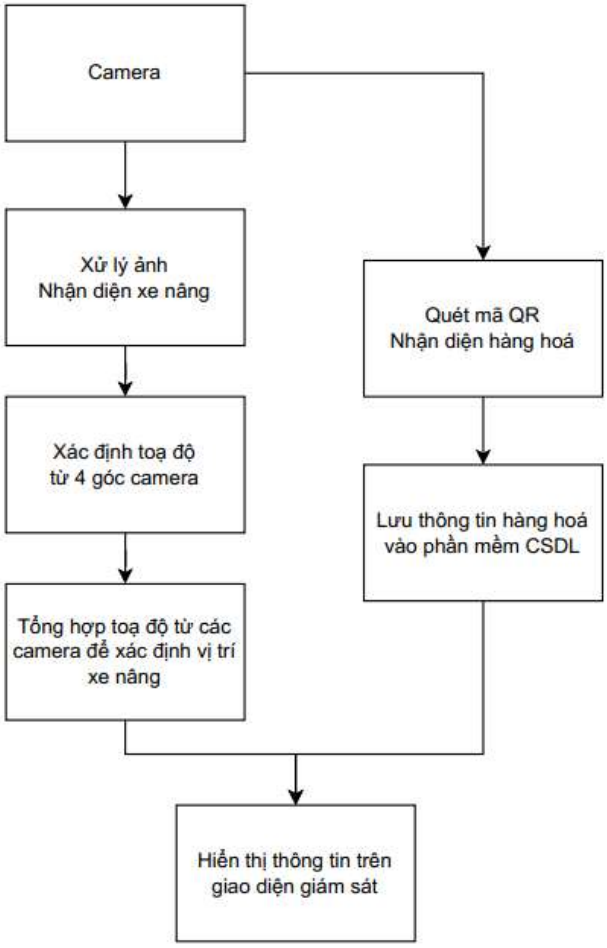
Ở chương 1, nhóm chúng em đã đưa ra tổng quan về đề tài "Xây dựng hệ thống tự động hoá trong kho hàng dệt may". Nêu được tính cấp thiết của dự án dựa trên hiện trạng thực tế tại các nhà máy. Nhóm cũng đã cung cấp giải pháp cho vấn đề này và đưa ra kết quả đạt được dự kiến.

Chương 2 sẽ đi sâu vào đối tượng chính của dự án, bao gồm cấu trúc, nguyên lý hoạt động và điều khiển của hệ thống. Ngoài ra sẽ nêu cụ thể các thiết bị (phần cứng, phần mềm) được sử dụng trong hệ thống

CHƯƠNG 2 : PHƯƠNG PHÁP XÁC ĐỊNH TOẠ ĐỘ XE NÂNG VÀ LƯU TRỮ HÀNG HOÁ

2.1 Cấu trúc của hệ thống :

Hệ thống có sơ đồ khối như hình 2.1



Hình 2. 1 Sơ đồ khối của hệ thống

Hệ thống sử dụng camera để xử lý ảnh, nhận diện và tính toán toạ độ xe nâng, đồng thời cũng dùng một camera để quét mã QR được in trên hàng hoá, thông tin hàng hoá sẽ được lưu vào phần mềm CSDL. Tất cả những dữ liệu về xe nâng và hoàng hoá sẽ được hiển thị trên giao diện giám sát

2.2 Một vài định nghĩa :

Trước khi đi vào nguyên lý hoạt động, cần tìm hiểu về một số định nghĩa

- **Thế nào là thị giác máy tính ?**

Thị giác máy tính (computer vision) là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng, ví dụ trong các dạng quyết định.Việc phát triển lĩnh vực này có bối cảnh từ việc sao chép các khả năng thị giác con người bởi sự nhận diện và hiểu biết một hình ảnh mang tính điện tử. Sự nhận diện hình ảnh có thể xem là việc giải quyết vấn đề của các biểu tượng thông tin từ dữ liệu hình ảnh qua cách dùng các mô hình được xây dựng với sự giúp đỡ của các ngành lý thuyết học, thống kê, vật lý và hình học.Thị giác máy tính cũng được mô tả là sự tổng thể của một dải rộng các quá trình tự động và tích hợp và các thể hiện cho các nhận thức thị giác.

Thị giác máy tính là một môn học khoa học liên quan đến lý thuyết đằng sau các hệ thống nhân tạo có trích xuất các thông tin từ các hình ảnh. Dữ liệu hình ảnh có thể nhiều dạng, chẳng hạn như chuỗi video, các cảnh từ đa camera, hay dữ liệu đa chiều từ máy quét y học. Thị giác máy tính còn là một môn học kỹ thuật, trong đó tìm kiếm việc áp dụng các mô hình và các lý thuyết cho việc xây dựng các hệ thống thị giác máy tính.

Các lĩnh vực con của thị giác máy tính bao gồm tái cấu trúc cảnh, dò tìm sự kiện, theo dõi video, nhận diện bố cục đối tượng, học, chỉ mục, đánh giá chuyển động và phục hồi ảnh.

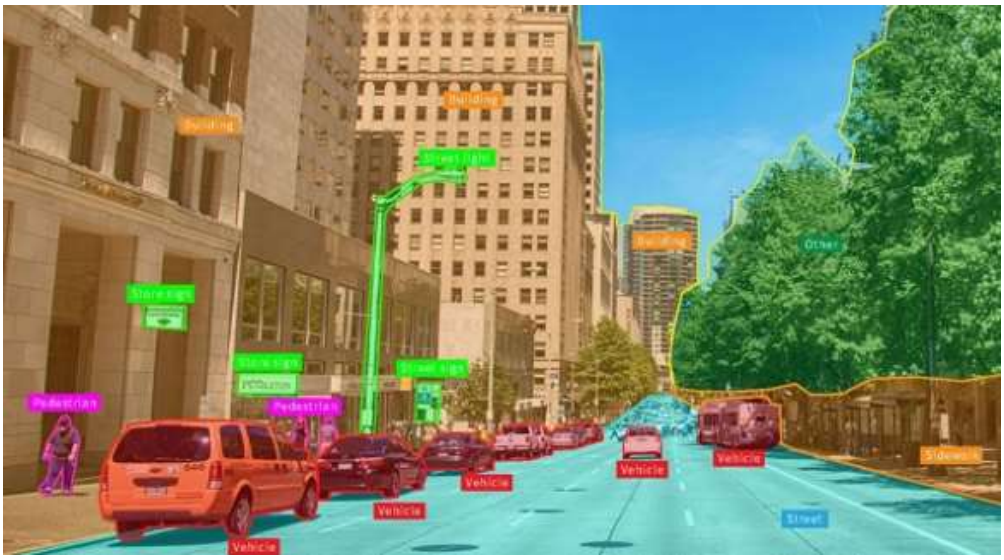
Nguyên lý hoạt động của thị giác máy tính :

Thị giác máy tính được phát triển và hoạt động dựa trên các kỹ thuật và thuật toán phức tạp nhằm mô phỏng quá trình nhận diện và xử lý hình ảnh của con người như sau:

- Thu thập hình ảnh (Image Acquisition): Bước đầu tiên trong quá trình Computer Vision là thu thập dữ liệu hình ảnh hoặc video từ nhiều nguồn khác nhau như camera, cảm biến hoặc các cơ sở dữ liệu hình ảnh đã có sẵn.
- Tiền xử lý (Preprocessing): Trước khi bắt đầu phân tích, các chuyên gia sẽ tiền xử lý hình ảnh để làm sạch dữ liệu, loại bỏ nhiễu, sửa các biến dạng, điều chỉnh độ sáng hoặc độ tương phản để cải thiện chất lượng hình ảnh. Mục tiêu của tiền

xử lý là đảm bảo các thuật toán nhận được đầu vào hình ảnh chính xác và đáng tin cậy.

- Trích xuất đặc trưng (Feature Extraction): Trong bước này, hệ thống sẽ nhận diện và trích xuất các đặc điểm đặc trưng của hình ảnh như các cạnh, góc, kết cấu, hình dạng hoặc phân bố màu sắc. Việc này giúp đơn giản hóa dữ liệu, giữ lại các thông tin quan trọng và loại bỏ những chi tiết không cần thiết, từ đó tạo ra đầu vào hiệu quả cho các bước phân tích và phân loại sau.
- Phân loại đối tượng (Object Classification): Sau khi trích xuất các đặc trưng và hoàn thiện huấn luyện mô hình học sâu (deep learning), hệ thống có thể phân loại và nhận diện các đối tượng trong hình ảnh bằng cách so sánh các đặc trưng đã trích xuất với các mẫu mà nó đã học được trong quá trình huấn luyện. Đây là quá trình giúp máy tính phân biệt các đối tượng khác nhau, chẳng hạn như nhận diện hình ảnh của một con mèo hay con chó.
- Nhận diện đối tượng (Object Identification): Nhận diện đối tượng nhằm xác định chính xác các đối tượng cụ thể trong một bức tranh. Nhiệm vụ của giai đoạn này là nhận diện các đối tượng hoặc cá thể trong một cảnh giúp hệ thống không chỉ phân loại mà còn biết chính xác đối tượng nào đang xuất hiện trong hình ảnh.
- Theo dõi đối tượng (Object Tracking): Quá trình theo dõi đối tượng trong thị giác máy tính liên quan đến việc theo dõi sự chuyển động của một đối tượng qua các khung hình liên tiếp trong video. Hệ thống sẽ xác định vị trí của đối tượng trong mỗi khung hình và duy trì sự liên tục trong suốt quá trình chuyển động.



Hình 2. 2 Thị giác máy tính giúp theo dõi đối tượng

Các công nghệ của thị giác máy tính :

Công nghệ thị giác máy tính dựa vào nhiều thuật toán khác nhau để thực hiện các nhiệm vụ như nhận diện đối tượng, theo dõi chuyển động, và phân loại hình ảnh. Một số công nghệ nổi bật trong lĩnh vực này bao gồm:

- Convolutional Neural Networks (CNNs) : Convolutional Neural Networks (CNNs) làm công nghệ thị giác máy tính sử dụng các phép toán học để trích xuất mẫu từ hình ảnh và ước lượng những gì chúng đang nhìn thấy. Quá trình này được lặp lại cho đến khi hệ thống xác nhận độ chính xác của ước lượng. CNNs có thể nhận diện các đặc điểm nổi bật trong hình ảnh như các cạnh, góc và các mẫu phức tạp khác giúp máy tính "hiểu" và phân tích hình ảnh một cách chính xác.
- Deep Learning và Transfer Learning : Sự ra đời của Deep Learning cho phép máy tính học hỏi về dữ liệu hình ảnh một cách độc lập mà không cần sự can thiệp nhiều từ con người. Các nhà khoa học chỉ cần phát triển một thuật toán tốt và máy tính sẽ tự động xử lý phần còn lại. Ngoài ra, Transfer Learning là một khái niệm cho phép công nghệ thị giác máy tính sử dụng mô hình đã được huấn luyện từ trước để bắt đầu. Điều này giúp giảm thiểu thời gian và chi phí huấn luyện, vì mô hình đã có sẵn các kiến thức cơ bản.
- Edge Detection và Feature Extraction : Edge Detection là một trong những kỹ thuật nổi bật trong việc trích xuất đặc trưng, đóng vai trò xác định biên giới của một đối tượng và trích xuất các đặc điểm của nó. Quá trình này sử dụng các thuật toán để nhận diện sự khác biệt trong độ sáng các điểm ảnh (sau khi chuyển dữ liệu thành hình ảnh xám). Mục tiêu cuối cùng của Edge Detection là nhận diện đối tượng trong hình ảnh.
- Optical Flow and Motion Estimation : Optical Flow là một kỹ thuật trong thị giác máy tính giúp xác định cách mỗi điểm trong một hình ảnh hoặc đoạn video di chuyển so với mặt phẳng hình ảnh. Kỹ thuật này có thể ước lượng tốc độ di chuyển của các đối tượng. Trong khi đó, motion estimation dự đoán vị trí của các đối tượng trong các khung hình tiếp theo của một chuỗi video. Cả hai kỹ thuật này đều rất quan trọng trong việc theo dõi đối tượng và dẫn đường tự động, đặc biệt trong các hệ thống tự lái và robot.

- Image Registration and Stitching : Image registration và Image stitching là hai kỹ thuật quan trọng trong thị giác máy tính có nhiệm vụ kết hợp nhiều hình ảnh lại với nhau. Image registration hỗ trợ căn chỉnh các hình ảnh, trong khi Image stitching sẽ chồng các hình ảnh lên nhau để tạo thành một hình ảnh duy nhất.

Hình 2. 3 Thị giác máy tính giúp xác định chuyển động và dự đoán vị trí

Quy trình xử lý ảnh trong thị giác máy tính :

Quy trình xử lý ảnh thường gồm 5 giai đoạn chính

- Thu nhận hình ảnh (Image Acquisition) : Là bước đầu tiên để lấy hình ảnh từ nguồn như camera webcam, file ảnh, video,… Hình ảnh thu được là dạng số, tập hợp các điểm ảnh pixel
- Tiền xử lý ảnh (Image Preprocessing) :Giúp làm sạch và chuẩn hóa ảnh để dễ phân tích. Ví dụ như cân bằng ánh sáng, làm mờ ảnh, lọc nhiễu,…
- Trích xuất đặc trưng (Feature Extraction) :Là bước rút ra các thông tin đặc trưng từ ảnh như: cạnh, đường viền, điểm đặc trưng, đối tượng,…
- Phân tích và Nhận dạng (Object Detection/Recognition) : Tùy vào bài toán, có thể là: phân loại ảnh, phát hiện đối tượng , phân loại đối tượng,…
- Hiển thị và diễn giải kết quả : vẽ vùng giới hạn, gửi dữ liệu sang phần mềm quản lý, lưu video hoặc kết quả nhận dạng

Ưu nhược điểm của thị giác máy tính :

Ưu điểm

- Tăng cường tự động hóa: Các hệ thống dựa trên thị giác máy tính có thể thay thế con người trong những công việc như kiểm tra chất lượng sản phẩm trong dây chuyền sản xuất hay nhận diện khuôn mặt trong hệ thống an ninh.
- Độ chính xác cao: Các thuật toán của Computer Vision có thể đạt được độ chính xác rất cao trong việc nhận diện và phân loại đối tượng, vượt qua khả năng của con người trong một số tình huống cụ thể.
- Ứng dụng rộng rãi: Ứng dụng của Computer Vision rất đa dạng, từ y tế (phân tích hình ảnh y khoa) đến giao thông (hệ thống lái xe tự động) và giải trí (nhận diện người chơi trong trò chơi video).

Nhược điểm

- Chi phí và yêu cầu tài nguyên cao: Việc phát triển các hệ thống thị giác máy tính đòi hỏi phần cứng mạnh mẽ và tài nguyên tính toán lớn, điều này có thể làm tăng chi phí triển khai.
- Khó khăn trong việc xử lý hình ảnh chất lượng thấp: Các thuật toán thị giác máy tính đôi khi gặp khó khăn trong việc nhận diện đối tượng khi hình ảnh bị mờ, ánh sáng yếu, hoặc có nhiều yếu tố nhiễu.
- Vấn đề bảo mật và quyền riêng tư: Các ứng dụng như nhận diện khuôn mặt có thể dẫn đến các vấn đề về bảo mật và quyền riêng tư nếu không được kiểm soát đúng cách.

Một số ứng dụng thực tiễn của thị giác máy tính:

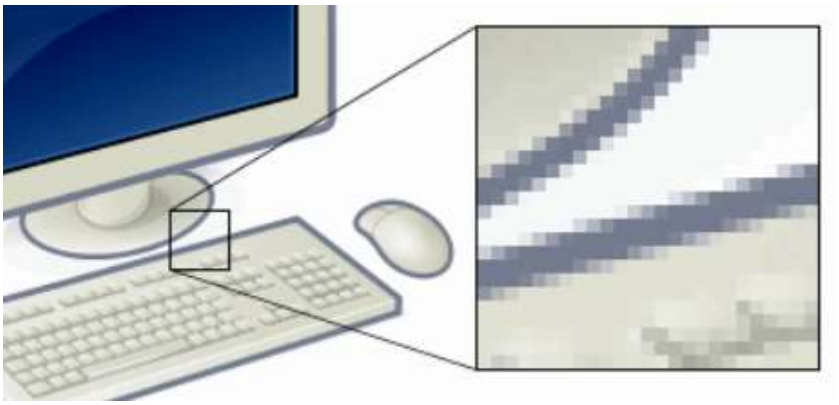
- Nhận diện khuôn mặt
- Lái xe tự động
- Phân tích hình ảnh y khoa
- Robot và tự động hoá
- Thương mại điện tử

Trong dự án này, ta có thể dùng thị giác máy tính để thu hình từ các camera, sau đó phát hiện xe bằng YOLOv8, xử lý ảnh và nhận diện xe, tiếp theo là biến đổi phối cảnh từ mỗi camera để xác định nhiều vị trí thực tế, cuối cùng là tính toán toạ độ trung bình từ toạ độ của các camera[5].

- **Pixel là gì ?**

Pixel là một đơn vị nhỏ nhất của một hình ảnh trên màn hình hoặc sensor ảnh. Pixel là viết tắt của "picture element" (phần tử hình ảnh). Mỗi pixel đại diện cho một điểm ảnh trên màn hình hoặc trong một hình ảnh số. Mỗi điểm ảnh là một mẫu của một hình ảnh ban đầu, nhiều điểm ảnh hơn thường cung cấp đại diện chính xác hơn của bản gốc.

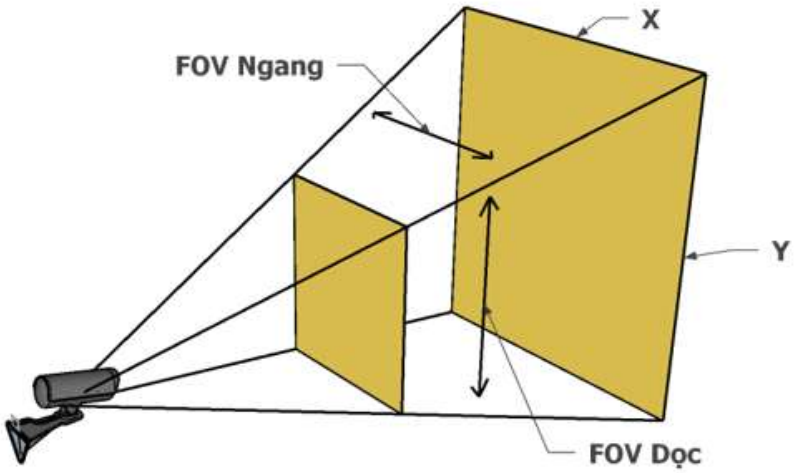
Cường độ của mỗi điểm ảnh có thể thay đổi. Các pixel được sắp xếp cạnh nhau với cường độ màu giống hoặc khác nhau, tạo nên hình ảnh mà mắt người có thể nhìn thấy được. Càng nhiều Pixel trong 1 bức ảnh thì bức hình đó càng chi tiết. Khi phóng to 1 bức ảnh, đến 1 mức nào đó sẽ nhìn thấy pixel[6]



Hình 2. 4 Các pixel trong một hình ảnh được hiển thị dưới dạng ô vuông nhỏ

- **Góc nhìn FOV là gì ?**

Field of View (FOV) – hay còn gọi là góc nhìn, trường nhìn, là góc tạo bởi hai đường thẳng xuất phát từ điểm đặt thiết bị (như camera, mắt người, cảm biến,…) đến hai mép ngoài cùng của vùng mà thiết bị đó có thể nhìn thấy hoặc ghi nhận được.



Hình 2. 5 Góc nhìn FOV

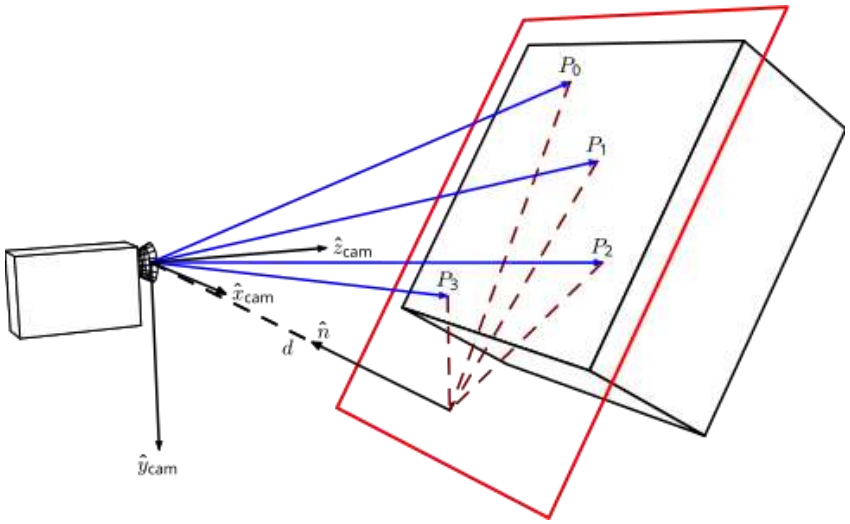
Trong nhiếp ảnh và quay phim, trường nhìn là phạm vi góc của cảnh. Có thể quan sát được mà ống kính hoặc máy ảnh có thể chụp được. Nó thường được biểu thị bằng độ hoặc radian. Trường nhìn rộng hơn có nghĩa là có thể nhìn thấy nhiều bối cảnh hơn, hữu ích khi chụp phong cảnh hoặc nhóm đông người. Trường nhìn hẹp hơn thường được sử dụng để chụp ảnh từ xa. Hoặc khi nhiếp ảnh gia hoặc đạo diễn video muốn tập trung vào một chủ đề nhỏ hơn.[7]

- **Phương pháp biến đổi đồng nhất là gì ?**

Biến đổi đồng nhất là phép biến đổi hình học trong không gian 2D, dùng để ánh xạ một mặt phẳng này sang mặt phẳng khác. Thường được dùng trong thị giác máy tính để :

- Ghép ảnh
- Hiệu chỉnh phối cảnh
- Xác định vị trí của vật thể trên mặt phẳng khi thay đổi góc nhìn

Biến đổi đồng nhất là phép biến đổi tuyến tính trong không gian toạ độ đồng nhất, thay vì biểu diễn toạ độ điểm là (x,y)[8]



Hình 2. 6 Phương pháp biến đổi đồng nhất

2.3 Nguyên lý hoạt động tổng quát :

Để xác định được toạ độ của xe nâng trong nhà kho, cần sử dụng phương pháp biến đổi đồng nhất thường được sử dụng để chuyển đổi toạ độ giữa các hệ quy chiếu khác nhau trong không gian 2D hoặc 3D.

Trong hệ toạ độ đồng nhất, mọi điểm được biểu diễn dưới dạng $(x, y) \rightarrow (wx, wy, w), w \neq 0$. Tất cả đều biểu diễn cùng một điểm vì khi chia cho w, vẫn thu được kết quả (x,y)

Các bước xác định toạ độ như sau

- Bước 1: Dạng tổng quát của công thức biến đổi phối cảnh đồng nhất

$$\begin{bmatrix} X' \\ Y' \\ W \end{bmatrix} = H \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.1)$$

Trong đó

- (u,v) : toạ độ pixel trong ảnh trả về
- (X',Y',W): toạ độ đồng nhất
- W là hệ số tỷ lệ, giúp chuẩn hoá toạ độ
- H : ma trận đồng nhất có kích thước 3x3

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{32} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (2.2)$$

- Bước 2 : Chuyển về toạ độ thực

Sau khi nhân ma trận
$$\begin{aligned} X' &= h_{11}u + h_{12}v + h_{13} \\ Y' &= h_{21}u + h_{22}v + h_{23} \\ \mathrm{W} &= h_{31}u + h_{32}v + h_{33} \end{aligned} \quad (2.3)$$

Lúc này toạ độ thực là
$$\begin{aligned} X &= \frac{X'}{\mathrm{W}} = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}} \\ Y &= \frac{Y'}{\mathrm{W}} = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}} \end{aligned} \quad (2.4)$$

➔ Mỗi camera đều thực hiện các bước như vậy để xác định toạ độ riêng cho hình ảnh của từng camera

- Bước 3 : Sau khi có toạ độ riêng từ hình ảnh từng camera, ta tính giá trị trung bình để lấy kết quả gần đúng nhất

$$\begin{aligned} x_{xe} &= \frac{x_1 + x_2 + x_3 + x_4}{4} \\ y_{xe} &= \frac{y_1 + y_2 + y_3 + y_4}{4} \end{aligned} \quad (2.5)$$

2.3.1 Các bước tìm ma trận H :

Các hệ số của ma trận H được tính từ 4 cặp điểm tương ứng giữa ảnh và mặt phẳng thực. 4 điểm đó cần phải phân bố đều trên mặt phẳng, không thẳng hàng. Nên chọn 4 điểm nằm tại 4 góc của khung hình để phủ kín vùng ảnh. Giả sử ta có 4 điểm tương ứng

Ảnh (pixel)	Thực tế (mét)
(u_1, v_1)	(x_1, y_1)
(u_2, v_2)	(x_2, y_2)
(u_3, v_3)	(x_3, y_3)
(u_4, v_4)	(x_4, y_4)

- Bước 1 : Viết 2 phương trình cho mỗi điểm

  Từ công thức chiếu ở trên

$$X_i = \frac{X'}{W} = \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + 1} \Rightarrow (h_{31}u_i + h_{32}v_i + 1)x_i = h_{11}u_i + h_{12}v_i + h_{13}$$
$$Y_i = \frac{Y'}{W} = \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + 1} \Rightarrow (h_{31}u_i + h_{32}v_i + 1)y_i = h_{21}u_i + h_{22}v_i + h_{23} \quad (2.6)$$

  Ở công thức này ta cho hệ số h_{33}=1 để loại bỏ vô định

- Bước 2 : Chuyển về dạng tuyến tính

  Chuyển từng phương trình về dạng tuyến tính với 8 ẩn. Giả sử với một ẩn i

$$h_{11}u_i + h_{12}v_i + h_{13} - h_{31}u_i x_i - h_{32}v_i x_i = x_i$$
$$h_{21}u_i + h_{22}v_i + h_{23} - h_{31}u_i y_i - h_{32}v_i y_i = y_i \quad (2.7)$$

  Tách hệ số, viết dưới dạng ma trận

$$\begin{bmatrix} u_i & v_i & 1 & 0 & 0 & 0 & -u_i x_i & -v_i x_i \\ 0 & 0 & 0 & u_i & v_i & 1 & -u_i y_i & -v_i y_i \end{bmatrix} \times \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2.8)$$

- Bước 3 : Xây dựng hệ phương trình

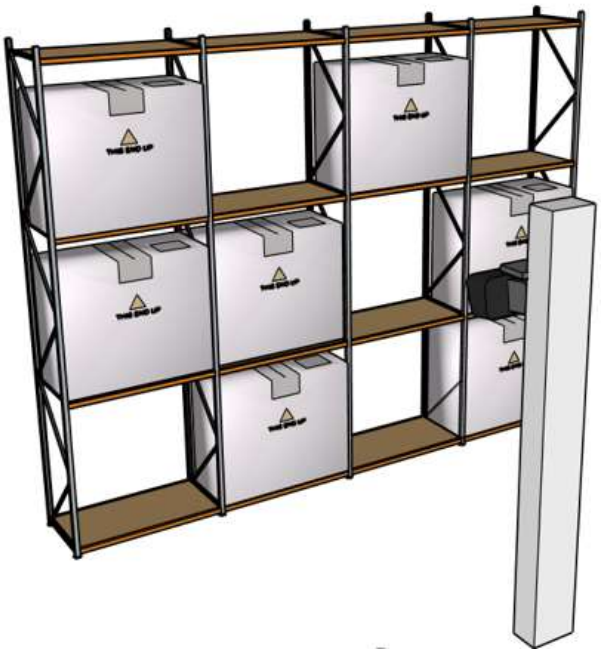
  Lặp lại bước 2 với các điểm còn lại điểm, ta có hệ phương trình 8 ẩn, sau khi giải xong, thế các giá trị vào ma trận H

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{32} \\ h_{31} & h_{32} & 1 \end{bmatrix} \quad (2.9)$$

2.3 Cách xác định vị trí hàng hoá trong kho :

Để xác định vị trí hàng hoá trong kho, mỗi kệ hàng sẽ lắp đặt 1 camera ở phía trước, Camera này sẽ đóng vai trò theo dõi hàng hoá. Mỗi ô hàng hoá sẽ được python xác định toạ độ dưới dạng pixel. Sau khi quét mã QR của hàng hoá, xe nâng đưa hàng vào ô, camera sẽ nhận diện có hàng hoá được đưa vào, sau đó lưu thông tin mã QR và vị trí chứa hàng, cùng với thời gian nhập hàng vào phần mềm cơ sở dữ liệu MySQL.

Tương tự khi xuất hàng, loại hàng, thời gian xuất hàng cũng sẽ được lưu vào cơ sở dữ liệu.



Hình 2. 7 Camera gắn phía trước kệ hàng

Hình 2. 8 Mã QR dán trên kiện hàng

2.4 Các thành phần được dùng trong hệ thống :

2.3.1 Các phần mềm sử dụng

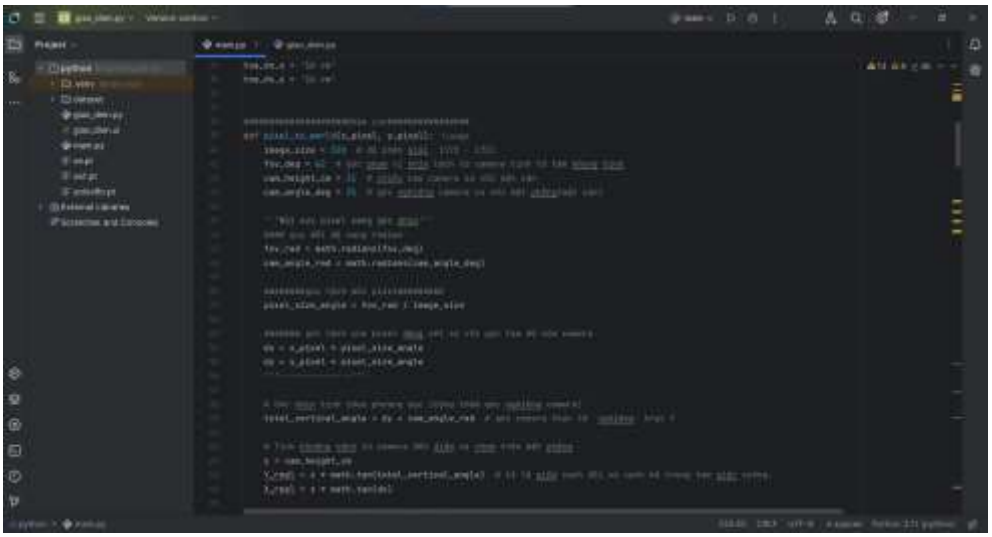
2.3.2.1 Giới thiệu về phần mềm Pycharm

PyCharm là một phần mềm IDE (Integrated Development Environment – Môi trường phát triển tích hợp) chuyên dùng để lập trình Python. PyCharm nổi tiếng nhờ việc giúp lập trình viên code nhanh hơn, chính xác hơn và dễ dàng quản lý dự án lớn.

PyCharm cũng tương tự như Python, là một trong những IDE được nhiều người dùng ưa chuộng và sử dụng rộng rãi trong giới lập trình. Nhờ sự tiện lợi và hiệu quả cao, PyCharm đã được nhiều doanh nghiệp có tầm ảnh hưởng lớn sử dụng làm IDE Python như: Symantec, Twitter, Pinterest,...[9]

Các ưu điểm của Pycharm :

- Giao diện thông minh : Trực quan, dễ dùng cho cả người mới lập trình và những người lập trình lâu năm
- Debugger mạnh mẽ : Hỗ trợ debug trực quan, xem giá trị biến,… Dễ dàng kiểm tra và sửa lỗi trong chương trình
- Quản lý project tốt : Pycharm giúp tổ chức các file python, thư viện, tài nguyên,… một cách gọn gàng
- Hỗ trợ web frameworks : Tích hợp tốt với nhiều phần mềm và website như PyQt5, Django,…
- Hỗ trợ viết code: Pycharm gợi ý hàm, biến, đối tượng,… trong quá trình viết, giúp rút ngắn thời gian làm việc
- Kiểm tra và cảnh báo lỗi : Phát hiện lỗi cú pháp, cảnh báo biến không dùng, hàm sai số,… trong thời gian thực
- Có phiên bản miễn phí đủ cho lập trình cơ bản



Hình 2. 9 Giao diện phần mềm Pycharm

Trong dự án này, Python đóng vai trò làm trung tâm điều khiển và xử lý chính trong toàn bộ hệ thống nhận diện, tính toán toạ độ xe nâng và quét mã QR hàng hoá. Các công dụng của Python như sau :

- Dùng để lấy hình ảnh theo thời gian thực từ các camera, đưa hình ảnh qua YOLOv8 để nhận diện xe nâng, trích xuất bounding box (vùng giới hạn) của xe trong hình ảnh
- Tính toán toạ độ thật của xe nâng
- Nhận diện và quét mã QR
- Lưu trữ dữ liệu vào MySQL

2.3.2.2 Giới thiệu về YOLOv8

YOLOv8 là một mô hình trí tuệ nhân tạo (AI) dùng để nhận diện và phát hiện đối tượng trong hình ảnh hoặc video. Ngoài ra YOLOv8 có thể xác định vị trí của từng đối tượng bằng cách vẽ hộp giới hạn (bounding box). Mô hình YOLO nổi tiếng với khả năng phát hiện nhanh, chính xác các đối tượng trong hình ảnh và thường được ứng dụng rộng rãi trong giám sát an ninh, phát hiện phương tiện giao thông, phân tích video và nhiều lĩnh vực khác.[10]

Đặc điểm nội bật của YOLOv8

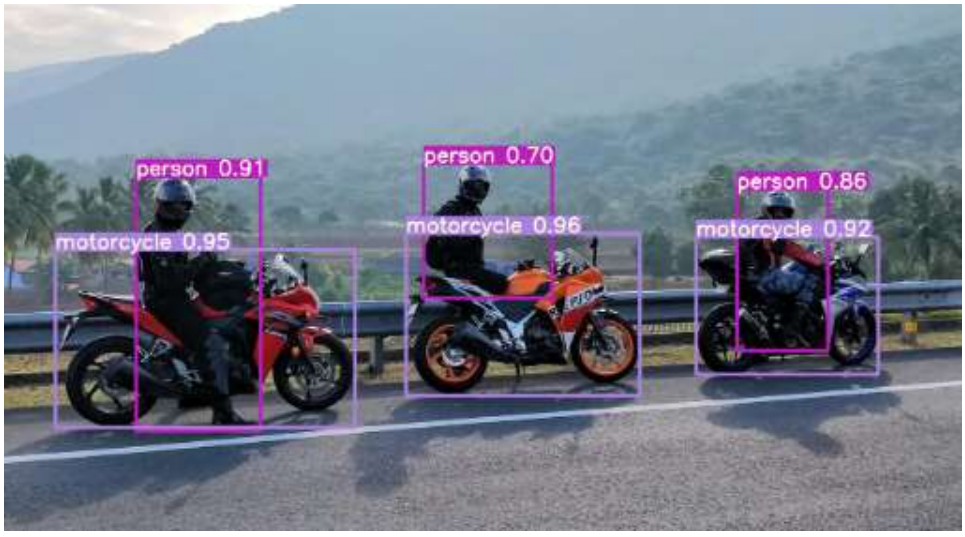
- Nhanh và chính xác: YOLO xử lý toàn bộ ảnh trong một lần (one-shot), khác với các mô hình khác cần thời gian xử lý lâu hơn. Ngoài ra YOLO có thể đạt tốc độ xủ lý trên 30fps

- Xử lý nhiều đối tượng cùng lúc: YOLO có thể phát hiện nhiều đối tượng trong một ảnh một cách đồng thời và phân loại từng loại rõ ràng. Dễ mở rộng cho các ứng dụng như đếm số người, theo dõi xe nâng, nhận diện hàng hoá,…
- Đơn giản, dễ dùng: Dễ hiểu và dễ triển khai với lượng mã nguồn không phức tạp
- Hỗ trợ nhiều công cụ và nền tảng: Có thể tích hợp vào python, C++, OpenCV,…
- Cho phép xuất file sang nhiều định dạng khác nhau

Trong dự án này, YOLO đóng vai trò giúp phát hiện xe từ hình ảnh lấy từ các camera. Từ mỗi khung hình, YOLO sẽ phát hiện xe, trả toạ độ xe dưới dạng pixel, tức là bounding box của xe. Đây là dữ liệu gốc để tính toán vị trí xe nâng trên mặt phẳng thật.

Ngoài ra , YOLO còn có thể lọc dữ liệu đầu vào bằng cách bỏ qua các vật thể không liên quan, chỉ dữ lại hình ảnh được train từ trước là xe nâng.

Việc xử lý hình ảnh bắt đầu từ việc đọc ảnh từ camera, dùng thư viện hỗ trợ để lấy khung hình từ camera, sau đó huấn luyện (train) để phát hiện vật thể, tiếp theo sẽ đến giai đoạn tính tâm vùng giới hạn (bounding box), lọc đối tượng theo nhãn (nếu quan sát nhiều đối tượng), và cuối cùng là tính toán để chuyển toạ độ pixel sang toạ độ thật



Hình 2. 10 Ứng dụng của YOLOv8

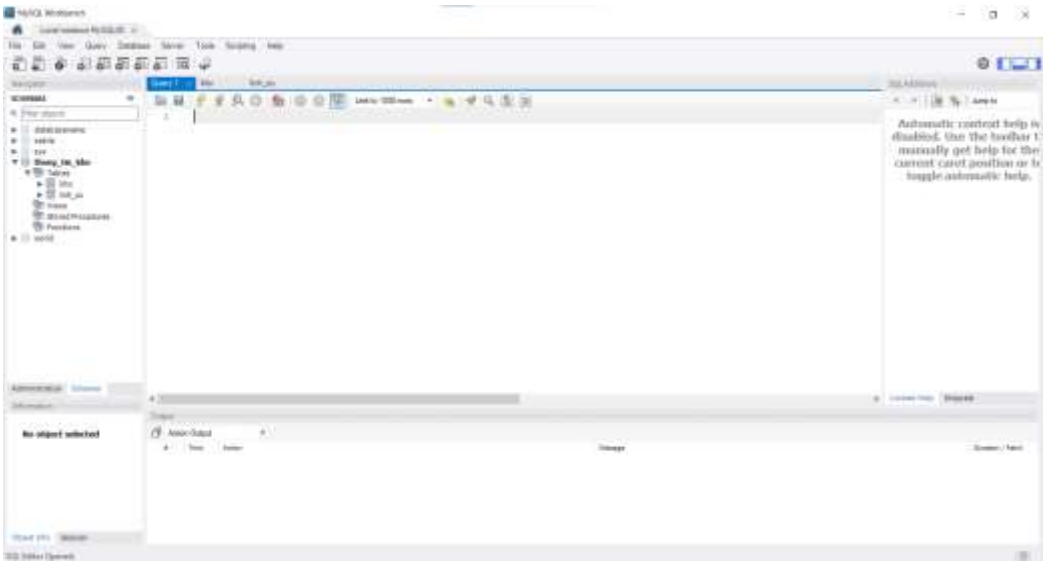
2.3.2.3 Giới thiệu về phần mềm MySQL

MySQL là phần mềm giúp tạo ra các cơ sở dữ liệu (database), và quản lý dữ liệu trong đó bằng ngôn ngữ SQL (Structured Query Language). MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.

Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.[11]

Các ưu điểm của MySQL

- Lưu trữ dữ liệu lớn
- Hiệu suất cao và ổn định : MySQL xử lý dữ liệu nhanh, hoạt động trên cả hệ thống lớn, nhiều người có thể truy cập cùng lúc
- Hỗ trợ đa nền tảng : Có thể cài đặt và sử dụng trên nhiều hệ điều hành khác nhau. Thêm vào đó, có thể tích hợp nhiều ngôn ngữ lập trình như python, Java,…
- Dễ học, dễ sử dụng : MySQL sử dụng các cú pháp đơn giản, phổ biến, phù hợp cho người mới bắt đầu
- Giao diện dòng lệnh đơn giản, nhiều công cụ hỗ trợ
- Thích hợp cho cả dự án nhỏ đến lớn
- Hoàn toàn miễn phí : MySQL là phần mềm có mã nguồn mở, cho phép sử dụng hoàn toàn miễn phí
- Bảo mật cao : Có hỗ trợ phân quyền người dùng
- Dễ sao lưu và phục hồi trong trường hợp xảy ra lỗi



Hình 2. 11 Giao diện phần mềm MySQL

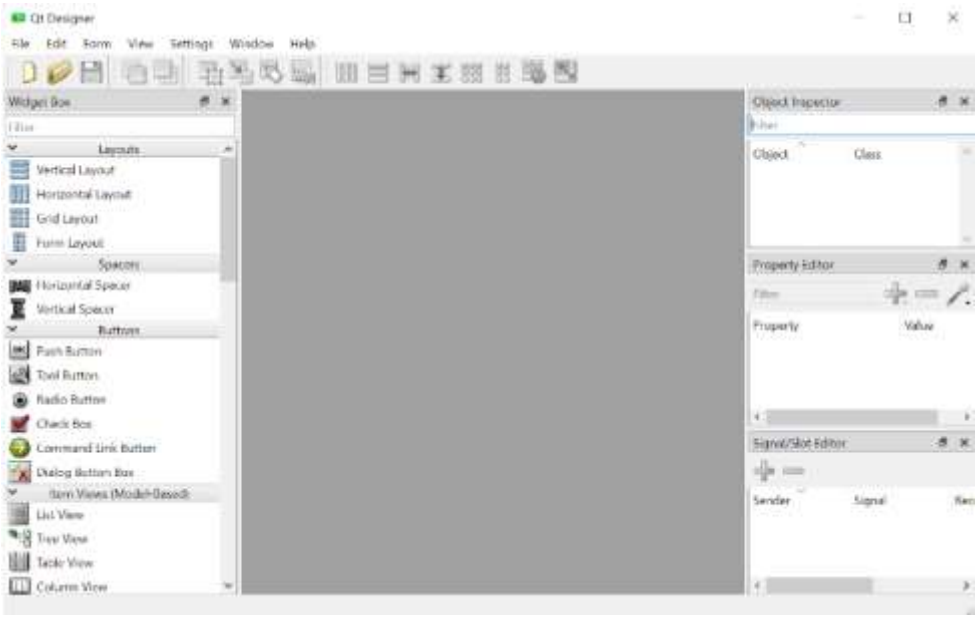
Trong dự án này, phần mềm MySQL đóng vai trò quản lý hệ cơ sở dữ liệu, chịu trách nhiệm lưu trữ, truy xuất và quản lý toàn bộ thông tin liên quan đến hoạt động kho. Các ứng dụng cụ thể bao gồm :

- Lưu thông tin hàng hoá (Mã hàng, loại hàng, ngày xuất/nhập kho)
- Kết nối và phục vụ giao diện người dùng (GUI)
- Có thể thống kê, báo cáo kho hàng cho người quản lý

2.3.2.4 Giới thiệu về Qt Designer :

Qt designer là công cụ đồ hoạ giúp thiết kế giao diện người dùng (GUI – Graphical User Interface) cho các ứng dụng viết bằng Qt – một framework phát triển phần mềm đa nền tảng phổ biến [12]. Các ưu điểm của Qt Designer :

- Giai diện kéo – thả : Thay vì viết mã thủ công, người dùng có thể tạo giao diện bằng cách kéo thả các nút bấm, nhãn, bảng,… hay còn gọi là widget
- Có thể chuyển thành mã python : Qt Designer lưu giao diện được thiết kế dưới dạng file XML, có thể chuyển thành mã python hoặc nạp trực tiếp trong python
- Tăng tốc độ phát triển phần mềm : Giúp tiết kiệm thời gian viết GUI bằng code, dễ kiểm tra bố cục giao diện một cách trực quan



Hình 2. 12 Giao diện phần mềm Qt Designer

Trong dự án này, Qt Designer đóng vai trò làm giao diện hiển thị thông tin bao gồm hình ảnh của các camera, toạ độ của xe nâng, vị trí hàng hoá và lịch sử xuất nhập kho của hàng.

2.4 Kết luận :

Ở chương 2, nhóm chúng em đã đưa ra cấu trúc và nguyên lý hoạt động của hệ thống, cách tính toán toạ độ của xe nâng dựa trên hình ảnh từ 4 camera. Ngoài ra cũng đã đưa ra các thành phần phần cứng và phần mềm dùng trong dự án này.

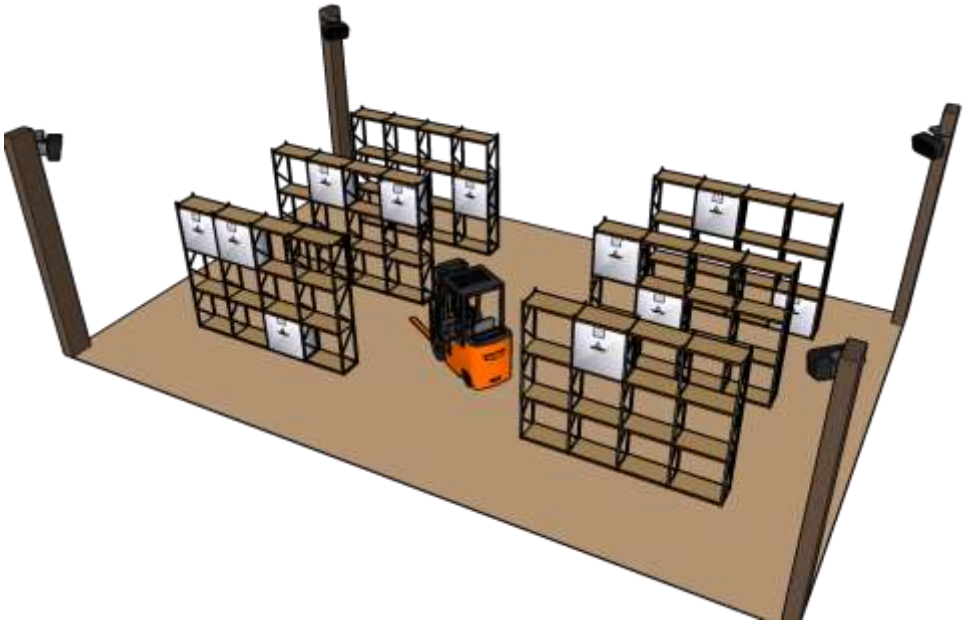
Trong chương 3, nhóm sẽ đi vào phần thiết kế cụ thể của đề tài, tính toán dựa theo thông số của nhà kho thực tế.

CHƯƠNG 3 : THIẾT KẾ HỆ THỐNG TỰ ĐỘNG HOÁ CHO KHO HÀNG DỆT MAY

3.1 Thiết kế hệ thống cho nhà kho :

Thiết kế giải pháp "Tự động hoá cho kho hàng dệt may" cho một kho hàng với các kích thước giả định như sau:

- Chiều dài: 30 mét
- Chiều rộng: 15 mét
- Chiều cao : 8 mét
- Tổng diện tích : 450m^2
- Kích thước kiện hàng : 80cm x 40cm x 40cm



Hình 3. 1 Sơ đồ nhà kho

3.1.1 Tính toán chiều cao lắp đặt camera :

Để tính chiều cao lắp đặt camera cho hệ thống, cần cân nhắc các yếu tố

- Trường nhìn (field of view – FOV) của camera
- Kích thước vùng cần quan sát (chiều dài và chiều rộng của mặt sàn) : cần xác định yếu tố này vì trường nhìn của camera có giới hạn. Camera chỉ có thể nhìn thấy một vùng nhất định, càng đặt camera xa vùng cần nhìn, thì

vùng bao phủ càng lớn, tuy nhiên độ chi tiết lại giảm. Nên cần xác định trước kích thước vùng mong muốn quan sát được để tính được nên đặt camera cao bao nhiêu là đủ

Để thực hiện dự án cho nhà kho, cần lựa chọn camera phù hợp trong các hệ thống giám sát công nghiệp, có giá thành hợp lý, độ phân giải cao đủ nhận diện vùng và vật thể lớn, ngoài ra cần có góc nhìn rộng và có độ bền cao, dùng trong lâu dài được.

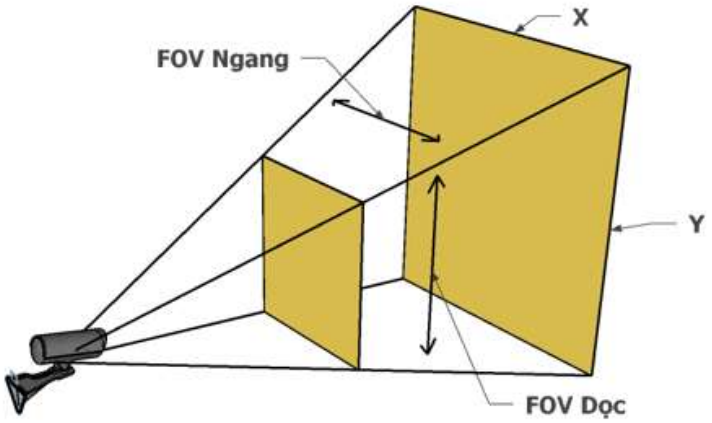
Từ những yêu cầu trên, lựa chọn loại camera IMOU Cruiser SE+ 4MP. Ưu điểm của camera này :

- Giá thành không quá đắt : Khoảng 1 triệu đồng, so với các loại cao cấp hơn, camera này vẫn cung cấp đủ chất lượng dùng
- Độ phân giải 2560x1440 (Full HD), đủ để quan sát xe nâng.
- Góc nhìn rộng (Khoảng 120^0 ngang và 90^0 dọc) : Bao phủ được một góc lớn, phù hợp để quan sát vùng rộng, hình ảnh ổn định.
- Có tiêu chuẩn chống bụi nước IP66 nên kháng bụi và nước tốt, dùng được lâu dài.
- Kích thước ảnh tối đa : 2560 x 1400 pixels

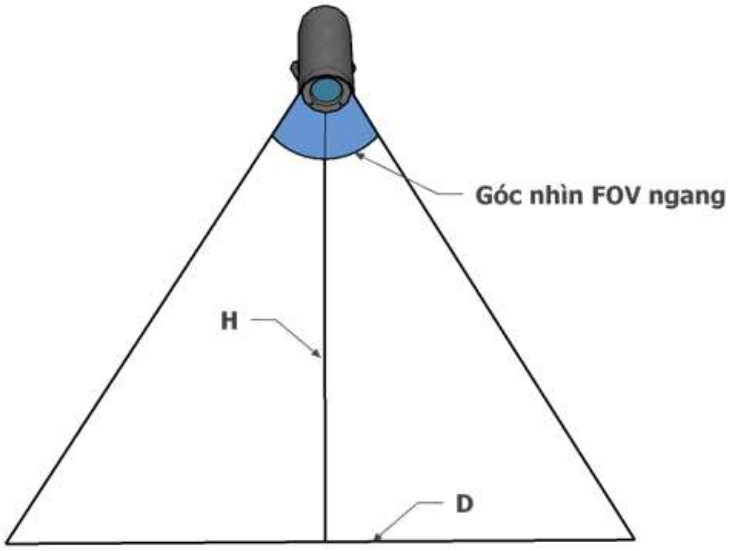
Hình 3. 2 Camera IMOU Cruiser SE+ 4MP

Với kích thước 15 x 30 mét của nhà kho, chọn góc nhìn của mỗi camera bao trọn vùng có chiều ngang 25 mét.

Camera tạo một hình nón nhìn (FOV). Khi xét mặt cắt dọc của hình nón thì sẽ được một hình tam giác cân.



Hình 3. 3 Góc nhìn FOV



Hình 3. 4 Mặt cắt của góc nhìn FOV

Với : H là chiều cao lắp camera

D : Chiều ngang vùng cần quan sát

Góc nhìn ngang FOV = 120^0 → FOV'= 60^0

Vì muốn camera bao trọn vùng có chiều ngang 25 mét, nên $D = 25$

Ta có: $\tan\left(FOV'\right) = \dfrac{\dfrac{D}{2}}{H} \Rightarrow D = 2 \times H \times \tan(FOV')$ (3.1)

$$\Rightarrow H = \frac{D}{2 \times \tan\left(FOV'\right)} = \frac{25}{2 \times \tan\left(60\right)} \approx 7.22\left(m\right) \quad (3.2)$$

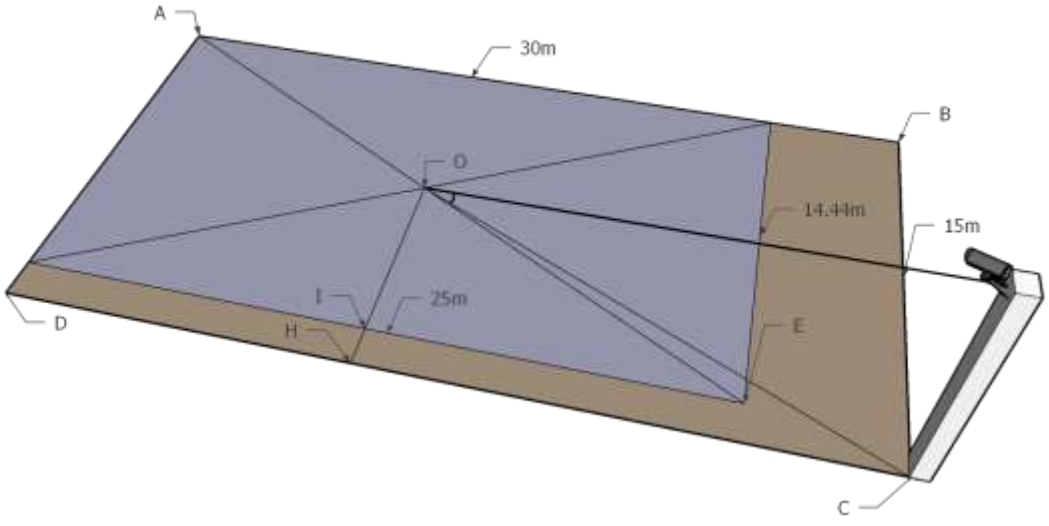
Khi đó, chiều rộng quan sát được của camera khi lắp đặt ở độ cao này là

$$\mathrm{W} = 2 \times H \times \tan\left(\frac{90}{2}\right) = 2 \times 7.22 \times \tan\left(45\right) = 14.44\left(m\right) \quad (3.3)$$

Vậy khi lắp đặt camera ở độ cao khoảng 7.22 mét, thì sẽ quan sát được vùng có kích thước là 14.44 x 25 mét

3.1.2 Tính góc nghiêng của camera so với mặt phẳng:

Sau khi đã xác định chiều cao lý tưởng để lắp camera, cần xác định góc nghiêng θ của camera so với mặt phẳng để quan sát được vùng có kích thước mong muốn.



Hình 3. 5 Góc nghiêng của camera so với mặt phẳng

Ta có : $\theta = \arctan\left(\frac{H}{OC}\right)$ với H là chiều cao của camera so với mặt phẳng, và OC là khoảng cách từ camera đến tâm của vùng nhìn thấy. Vậy cần tính khoảng cách này để xác định góc nghiêng θ. Cách tính khoảng cách như sau :

Ta có :
$$IE = HD = \frac{25}{2} = 12.5(m)$$
$$HC = DC - HD = 30 - 12.5 = 17.5(m) \quad (3.4)$$

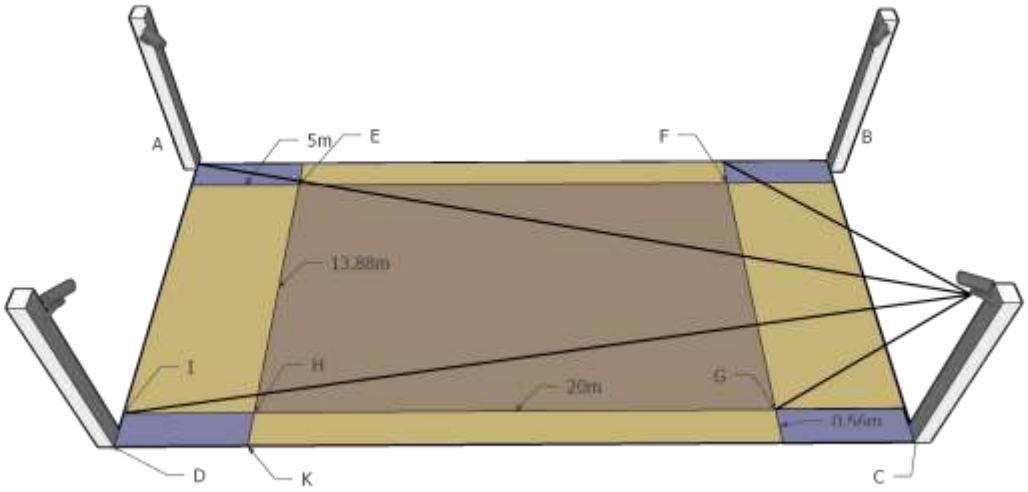
$$OI = \frac{14.44}{2} = 7.22(m)$$
$$IH = 15 - 14.44 = 0.56(m)$$
$$\Rightarrow OH = 0.56 + 0.72 = 7.78(m) \quad (3.5)$$
$$\Rightarrow OC = \sqrt{OH^2 + HC^2} = \sqrt{7.78^2 + 17.5^2} = 14.15(m)$$

$$\Rightarrow \theta = \arctan\left(\frac{H}{OC}\right) = \frac{7.22}{14.15} = 27^0 \quad (3.6)$$

- Kết luận: Vậy cần lắp camera cao 7.22 mét so với mặt phẳng, và có góc nghiêng $\theta = 27^0$ so với mặt phẳng để quan sát được vùng có kích thước 14.44 x 25 mét

3.1.3 Độ bao phủ của 4 camera trong nhà kho:

Hình 3.6 thể hiện độ bao phủ của cả 4 camera trong nhà kho. Khu trung tâm của nhà kho được quan sát bởi 4 camera, trong khi đó, vùng ngoài rìa được quan sát bởi 2 camera và 4 vùng nằm ở chân camera chỉ được quan sát bởi camera đối diện



Hình 3. 6 Độ bao phủ của 4 camera

- Tính diện tích vùng bao phủ bởi 4 camera:

Ta có :
$$\begin{aligned} &DK = DC - KC = 30 - 25 = 5(m) \\ &\Rightarrow HG = DC - 2 \times DK = 30 - 2 \times 5 = 20(m) \end{aligned} \quad (3.7)$$

$$\begin{aligned} &ID = AD - AI = 15 - 14.44 = 0.56(m) \\ &\Rightarrow EH = AD - 2 \times AI = 15 - 2 \times 0.56 = 13.88(m) \end{aligned} \quad (3.8)$$

Vậy diện tích được bao phủ bởi 4 camera là :

$$S_{4cam} = HG \times EH = 20 \times 13.88 = 277.6\left(m^2\right) \quad (3.9)$$

Suy ra phần trăm diện tích nhà kho được quan sát bởi 4 camera là :

$$\%S_{4cam} = \frac{S_{4cam}}{S_{tong}} \times 100\% = \frac{277.6}{450} \times 100\% \approx 61.69\% \quad (3.10)$$

Vậy có 61.69% diện tích nhà kho được bao phủ bởi 4 camera

- Tiếp theo, tính diện tích các khu vực ngoài rìa được bao phủ bởi 2 camera:

Ta có:

$$S_{2cam} = (IH \times EH \times 2) + (HK \times HG \times 2) = (5 \times 13.88 \times 2) + (0.56 \times 20 \times 2) = 161.2(m^2)$$

(3.22)

Suy ra phần trăm diện tích nhà kho được quan sát bởi 2 camera là :

$$\%S_{2cam} = \frac{S_{2cam}}{S_{tong}} \times 100\% = \frac{161.2}{450} \times 100\% \approx 35.82\% \quad (3.11)$$

Vậy có 35.82% diện tích nhà kho được bao phủ bởi 4 camera

- Tính diện tích các khu vực ở 4 góc được quan sát bởi 1 camera :

$$S_{1cam} = IH \times HK \times 4 = 5 \times 0.56 \times 4 = 11.2(m^2) \quad (3.12)$$

Suy ra phần trăm diện tích nhà kho được quan sát bởi 1 camera là :

$$\%S_{1cam} = \frac{S_{1cam}}{S_{tong}} \times 100\% = \frac{11.2}{450} \times 100\% \approx 2.49\% \quad (3.13)$$

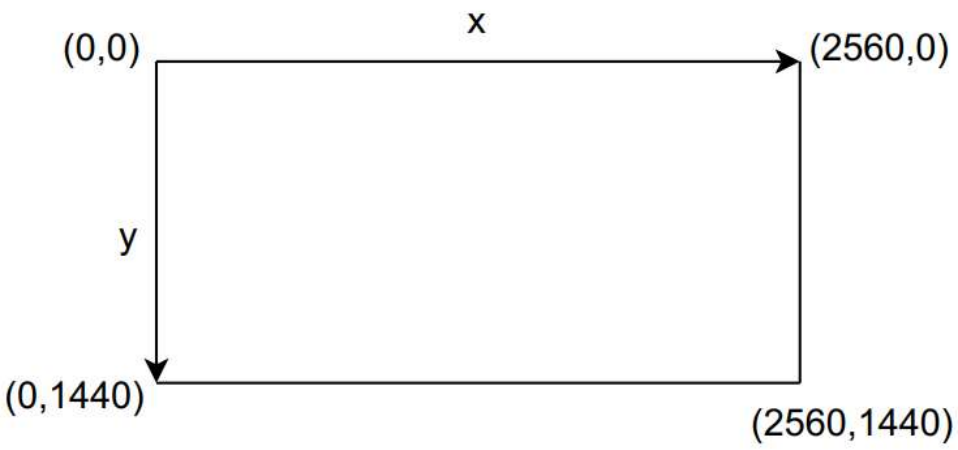
Vậy có 2.49% diện tích nhà kho được bao phủ bởi 4 camera

3.1.4 Tính toạ độ của xe nâng:

Trên hình ảnh lấy từ camera khi nhận diện xe nâng, mỗi vị trí của xe sẽ có toạ độ pixel riêng, đây là cách định danh vị trí của một điểm ảnh, được xác định theo hệ trục toạ độ 2D. Trong hình ảnh kỹ thuật số, toạ độ pixel (x,y) nghĩa là :

- Gốc toạ độ (0,0) nằm ở góc trên bên trái của ảnh
- Trục x tăng dần từ trái sang phải
- Trục y tăng dần từ trên xuống dưới

Camera sử dụng trong mô phỏng này có kích thước ảnh tối đa 2560 x 1440 pixels.Có nghĩa là toạ độ của điểm ảnh trong hình ảnh thu được như sau.



Hình 3. 7 Toạ độ pixel của ảnh

Như đã nêu ở chương 2, để xác định được toạ độ của xe nâng trong nhà kho, cần dùng phương pháp biến đổi ma trận đồng nhất cho toạ độ pixel thu được từ các camera, sau đó tính giá trị trung bình.

Để tính được toạ độ thực, trước hết cần tính các hệ số của ma trận đồng nhất H cho từng camera

Giả sử ở camera 1 , ta lấy 4 điểm ảnh theo toạ độ thực, với gốc toạ độ là điểm trên trái của khung hình, A(5,3), B(25,4.5), C(7.5, 13.5), D(27,12). Lúc này tỉ lệ chuyển đổi là

Tỉ lệ X (theo chiều ngang) : $1m \leftrightarrow \frac{2560}{30} \approx 85.33 pixel$

Tỉ lệ Y (theo chiều dọc) : $1m \leftrightarrow \frac{1440}{15} \approx 96 pixel$

Sau đó chuyển đổi toạ độ thực sang pixel

Ảnh (pixel)	Thực tế (mét)
(427,288)	(5,3)
(2133,432)	(25,4.5)
(640,1296)	(7.5,13.5)
(2304,1152)	(27,12)

- Bước 1 : Viết 2 phương trình cho mỗi điểm từ công thức chiếu

$$X_1 = \frac{X'}{W} = \frac{427h_{11} + 288h_{12} + h_{13}}{427h_{31} + h_{32}288 + 1} \Rightarrow (427h_{31} + 288h_{32} + 1) \times 5 = 427h_{11} + 288h_{12} + h_{13}$$

$$Y_1 = \frac{Y'}{W} = \frac{427h_{21} + 288h_{22} + h_{23}}{427h_{31} + 288h_{32} + 1} \Rightarrow (427h_{31} + 288h_{32} + 1) \times 3 = 427h_{21} + 288h_{22} + h_{23}$$

$$X_2 = \frac{X_2'}{W} = \frac{2133h_{11} + 432h_{12} + h_{13}}{2133h_{31} + 432h_{32} + 1} \Rightarrow (2133h_{31} + 432h_{32} + 1) \times 25 = 2133h_{11} + 432h_{12} + h_{13}$$

$$Y_2 = \frac{Y_2'}{W} = \frac{2133h_{21} + 432h_{22} + h_{23}}{2133h_{31} + 432h_{32} + 1} \Rightarrow (2133h_{31} + 432h_{32} + 1) \times 4.5 = 2133h_{21} + 432h_{22} + h_{23} \quad (3.14)$$

$$X_3 = \frac{X_3'}{W} = \frac{640h_{11} + 1296h_{12} + h_{13}}{640h_{31} + 1296h_{32} + 1} \Rightarrow (640h_{31} + 1296h_{32} + 1) \times 7.5 = 640h_{11} + 1296h_{12} + h_{13}$$

$$Y_3 = \frac{Y_3'}{W} = \frac{640h_{21} + 1296h_{22} + h_{23}}{640h_{31} + 1296h_{32} + 1} \Rightarrow (640h_{31} + 1296h_{32} + 1) \times 13.5 = 640h_{21} + 1296h_{22} + h_{23}$$

$$X_4 = \frac{X_4'}{W} = \frac{2304h_{11} + 1152h_{12} + h_{13}}{2304h_{31} + 1152h_{32} + 1} \Rightarrow (2304h_{31} + 1152h_{32} + 1) \times 27 = 2304h_{11} + 1152h_{12} + h_{13}$$

$$Y_4 = \frac{Y_4'}{W} = \frac{2304h_{21} + 1152h_{22} + h_{23}}{2304h_{31} + 1152h_{32} + 1} \Rightarrow (2304h_{31} + 1152h_{32} + 1) \times 12 = 2304h_{21} + 1152h_{22} + h_{23}$$

- Bước 2 : Chuyển về dạng tuyến tính

$$\begin{aligned}
&427h_{11} + 288h_{12} + h_{13} - 427 \times 5 \times h_{31} - 288 \times 5 \times h_{32} = 5 \\
&427h_{21} + 288h_{22} + h_{23} - 427 \times 3 \times h_{31} - 288 \times 3 \times h_{32} = 3 \\
&2133h_{11} + 432h_{12} + h_{13} - 2133 \times 25 \times h_{31} - 432 \times 25 \times h_{32} = 25 \\
&2133h_{21} + 432h_{22} + h_{23} - 2133 \times 4.5 \times h_{31} - 432 \times 4.5 \times h_{32} = 4.5 \\
&640h_{11} + 1296h_{12} + h_{13} - 640 \times 7.5 \times h_{31} - 1296 \times 7.5 \times h_{32} = 7.5 \\
&640h_{21} + 1296h_{22} + h_{23} - 640 \times 13.5 \times h_{31} - 1296 \times 13.5 \times h_{32} = 13.5 \\
&2304h_{11} + 1152h_{12} + h_{13} - 2304 \times 27 \times h_{31} - 1152 \times 27 \times h_{32} = 27 \\
&2304h_{21} + 1152h_{22} + h_{23} - 2304 \times 12 \times h_{31} - 1152 \times 12 \times h_{32} = 12
\end{aligned} \quad (3.15)$$

- Bước 3 : Tách hệ số, viết dưới dạng ma trận

$$\begin{bmatrix} 427 & 288 & 1 & 0 & 0 & 0 & -427\times 5 & -288\times 5 \\ 0 & 0 & 0 & 427 & 288 & 1 & -427\times 3 & -288\times 3 \end{bmatrix} \times \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 2133 & 432 & 1 & 0 & 0 & 0 & -2133\times 25 & -432\times 25 \\ 0 & 0 & 0 & 2133 & 432 & 1 & -2133\times 4.5 & -432\times 4.5 \end{bmatrix} \times \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 25 \\ 4.5 \end{bmatrix} \quad (3.16)$$

$$\begin{bmatrix} 640 & 1296 & 1 & 0 & 0 & 0 & -640\times 7.5 & -1296\times 7.5 \\ 0 & 0 & 0 & 640 & 1296 & 1 & -640\times 13.5 & -1296\times 13.5 \end{bmatrix} \times \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 7.5 \\ 13.5 \end{bmatrix}$$

$$\begin{bmatrix} 2304 & 1152 & 1 & 0 & 0 & 0 & -2300\times 27 & -1152\times 27 \\ 0 & 0 & 0 & 2304 & 1152 & 1 & -2304\times 12 & -1152\times 12 \end{bmatrix} \times \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} 27 \\ 12 \end{bmatrix}$$

- Bước 4 : Xây dựng hệ phương trình

  Ta có hệ phương trình 8 ẩn, sau khi giải xong, thế các giá trị vào ma trận H. Vì quá trình giải tay rất phức tạp nên có thể dùng python để giải ra ma trận H

$$H = \begin{bmatrix} 0.0125 & -0.006 & -0.2017 \\ 0.001 & 0.0075 & -0.1341 \\ 0 & 0 & 1 \end{bmatrix}$$

Tính ma trận H ở các camera còn lại bằng cách tương tự, mỗi ma trận H được dùng để tính toạ độ thực của xe ở mỗi camera

- Bước 5 : Sau khi có ma trận H, thay các giá trị của ma trận H để tính toạ độ

$$X = \frac{X'}{\mathrm{W}} = \frac{h_{11}u + h_{12}v + h_{13}}{h_{31}u + h_{32}v + h_{33}}$$
$$Y = \frac{Y'}{\mathrm{W}} = \frac{h_{21}u + h_{22}v + h_{23}}{h_{31}u + h_{32}v + h_{33}} \quad (3.17)$$

- Bước 6 : Tính trung bình toạ độ ở 4 camera để lấy giá trị gần đúng nhất

$$x_{xe} = \frac{x_1 + x_2 + x_3 + x_4}{4}$$
$$y_{xe} = \frac{y_1 + y_2 + y_3 + y_4}{4} \quad (3.18)$$

3.1.5 Tính sai số khi bị khuất camera:

Giả sử trường hợp 1 camera gặp sự cố, không hiển thị được hình ảnh hay toạ độ, hoặc camera bị che khuất. Thì sẽ có sai số do thiếu camera. Cách tính sai số như sau :

- Tính khoảng cách giữa điểm này và gốc toạ độ sẽ là

$$E = \sqrt{x_{xe}^{\ 2} + y_{xe}^{\ 2}} \quad (3.19)$$

Tuỳ vào camera nào bị mất, độ sai số sẽ thay đổi

- Giả sử bị mất camera số 1. Lúc này toạ độ trung bình sẽ là:

$$x_{3cam} = \frac{x_2 + x_3 + x_4}{3}$$
$$y_{3cam} = \frac{y_2 + y_3 + y_4}{3} \quad (3.20)$$

Suy ra sai số là:

$$E_3 = \sqrt{(x_{3cam} - x_{xe})^2 + (y_{3cam} - y_{xe})^2} \quad (3.29)$$

Suy ra phần trăm sai số khi mất camera số 1 trong trường hợp này là :

$$\%E_3 = \frac{E_3}{E} \times 100\% \quad (3.21)$$

- Với trường hợp 2 camera bị mất hình ảnh, giả sử mất cam 1 và cam 2, toạ độ trung bình lúc sẽ là:

$$x_{2cam} = \frac{x_3 + x_4}{2}$$
$$y_{2cam} = \frac{y_3 + y_4}{2} \quad (3.22)$$

Khi đó khoảng cách sai số là:

$$E_2 = \sqrt{(x_{2cam} - x_{xe})^2 + (y_{2cam} - y_{xe})^2} \quad (3.32)$$

Suy ra phần trăm sai số khi mất 2 camera trong trường hợp này là:

$$\%E_2 = \frac{E_2}{E} \times 100\% \quad (3.23)$$

- Nếu xe di chuyển đến góc nhà kho, lúc đó sẽ chỉ có 1 camera nhận diện được xe, giả sử chỉ có camera 4 quan sát được xe, lúc đó toạ độ trung bình là:

$$x_{1cam} = x_4$$
$$y_{2cam} = y_4$$

Khi đó sai số là:

$$E_1 = \sqrt{(x_{1cam} - x_{xe})^2 + (y_{1cam} - y_{xe})^2} \quad (3.24)$$

Suy ra phần trăm sai số khi mất cả 3 camera trong trường hợp này là:

$$\%E_1 = \frac{E_1}{E} \times 100\% \quad (3.25)$$

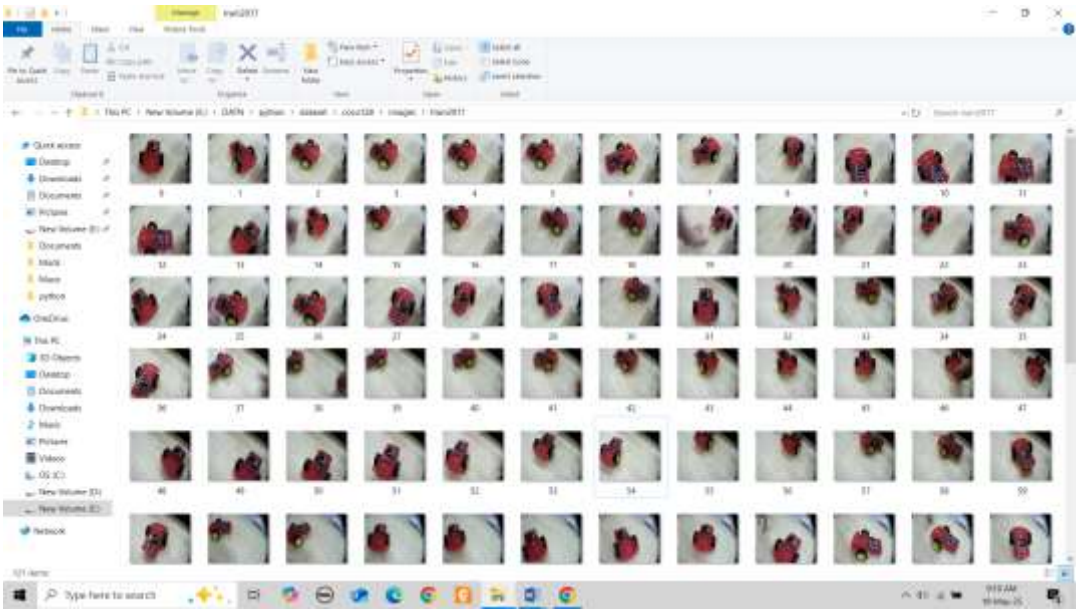
3.2 Huấn luyện (train) hệ thống nhận diện xe nâng:

YOLOv8 có thể nhận diện các vật thể phổ biến như ô tô, chó, mèo,… Tuy nhiên, xe nâng không có sẵn trong mô hình hoặc hệ thống, nên cần phải huấn luyện mô hình để camera học cách nhận diện đúng đối tượng cần xử lý. Sau khi nhận ảnh đầu vào, YOLO sẽ thực hiện giai đoạn tiền xử lý ảnh, bao gồm resize về kích thước chuẩn, chuyển ảnh thành dạng mảng số,…Sau đó sẽ chia ảnh thành một lưới và dự đoán trong mỗi ô

xem có vật thể không, và xác định toạ độ vùng giới hạn (bounding box), tiếp theo sẽ lọc và trả lại kết quả

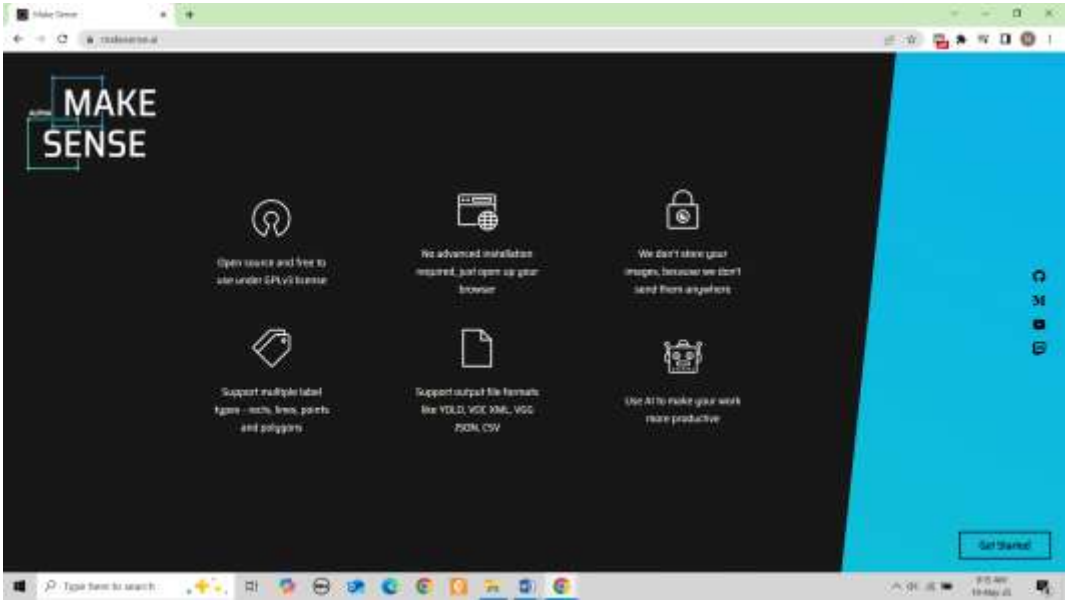
Các bước để huấn luyện hệ thống nhận diện xe nâng như sau

- Bước 1: Chụp ảnh đối tượng cần huấn luyện. Vì AI không biết trước được về hình dáng của vật thể, nên cần phải học từ ví dụ cụ thể. Càng cung cấp nhiều ảnh về đối tượng, độ chính xác càng tăng, ngoài ra, việc có nhiều ảnh chụp từ các góc khác nhau, với ánh sáng khác nhau làm tăng độ đa dạng



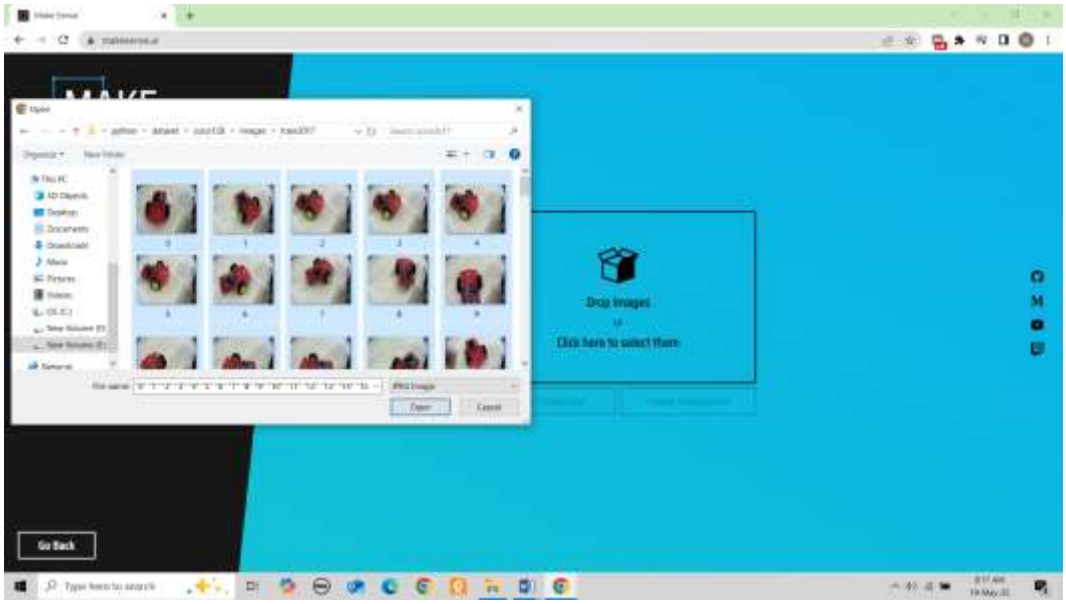
Hình 3. 8 Chụp ảnh đối tượng cần nhận diện

- Bước 2 : Truy cập vào trang web makesense AI, đây là một công cụ gán nhãn ảnh miễn phí, dùng để gán nhãn cho đối tượng trong ảnh, tạo dữ liệu huấn luyện cho mô hình như YOLO, ngoài ra công cụ này chạy trực tiếp trên trình duyện nên không cần cài đặt và tốn phí.



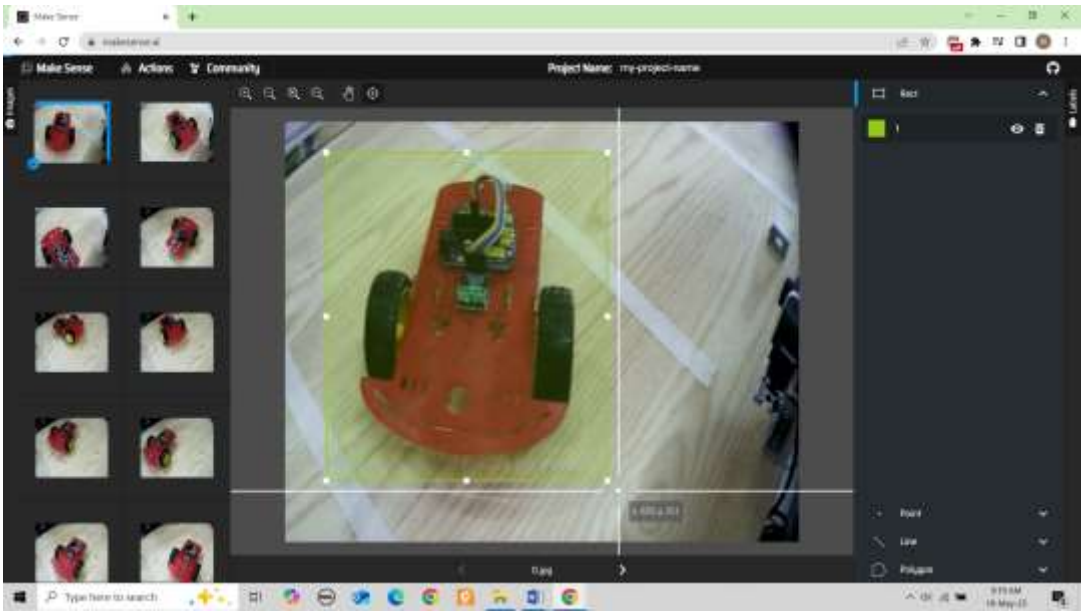
Hình 3. 9 Giao diện trang web Makesense AI

- Bước 3: Chọn các ảnh chụp đối tượng từ trước, sau đó chọn object detection để bắt đầu nhận diện vật thể



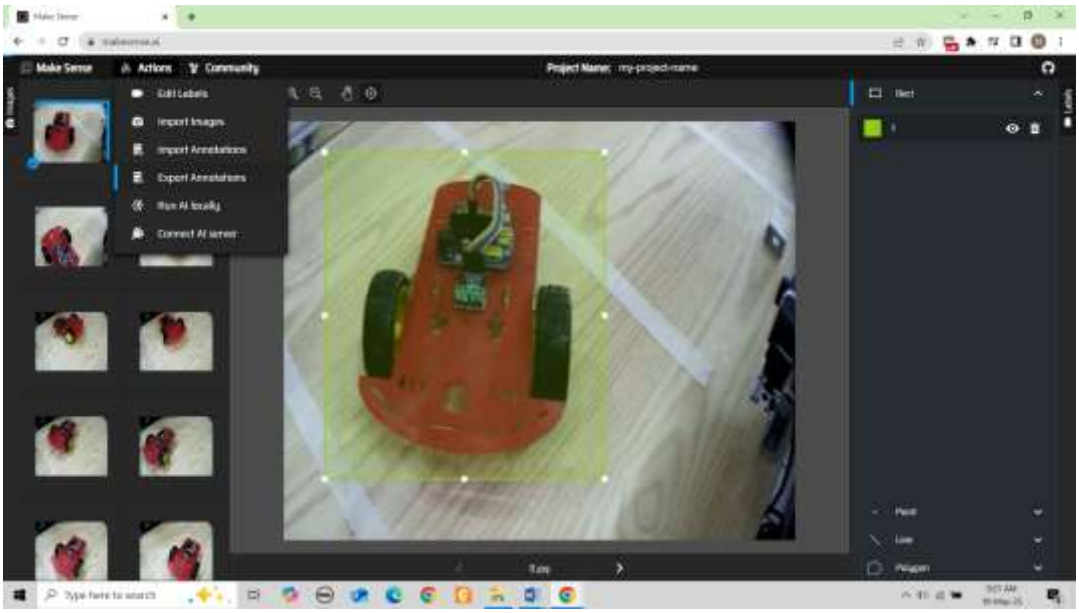
Hình 3. 10 Chọn ảnh các đối tượng đã được chụp và tải lên

- Bước 4: Tiến hành nhận diện xe từ ảnh. Lặp lại bước này cho tất cả các ảnh đã chụp



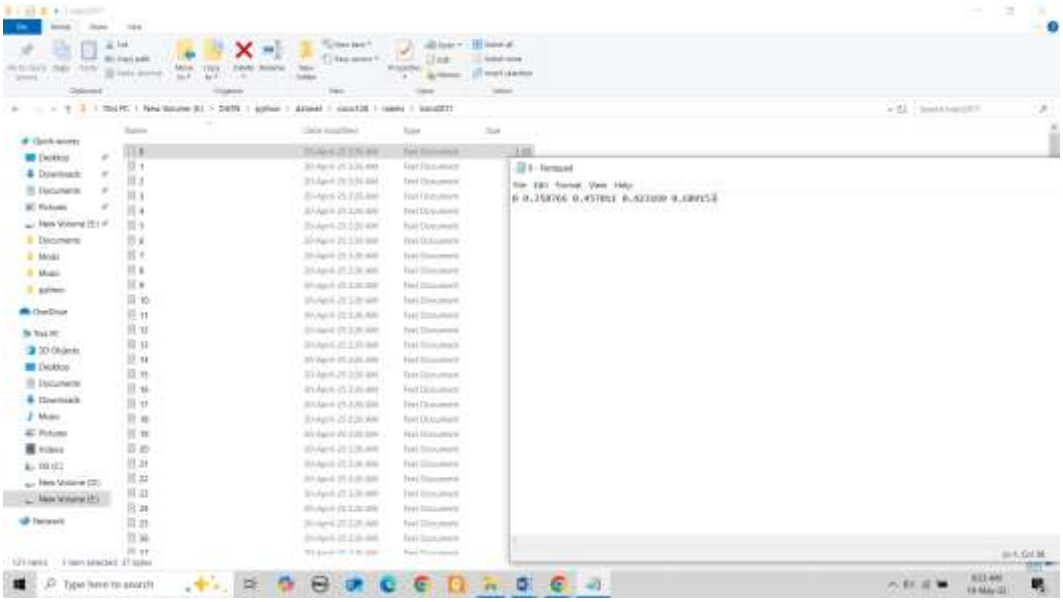
Hình 3. 11 Tiến hành nhận diện đối tượng

- Bước 5: Sau khi nhận diện xong, vào Action → Export để xuất file về máy tính



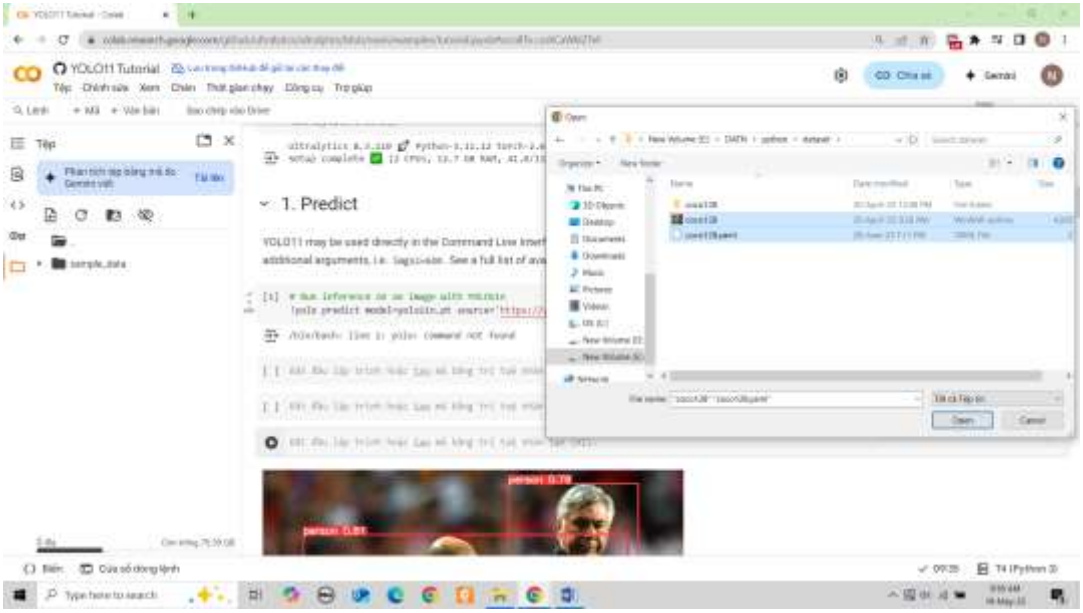
Hình 3. 12 Xuất file về máy tính

File sẽ được xuất về máy, số file sẽ tương ứng với số ảnh được huấn luyện, và mỗi file sẽ hiển thị id, toạ độ xe trong tấm ảnh, với thứ tự là class id, x_center,y_center, width, height (tất cả được chuẩn hoá 0→1)

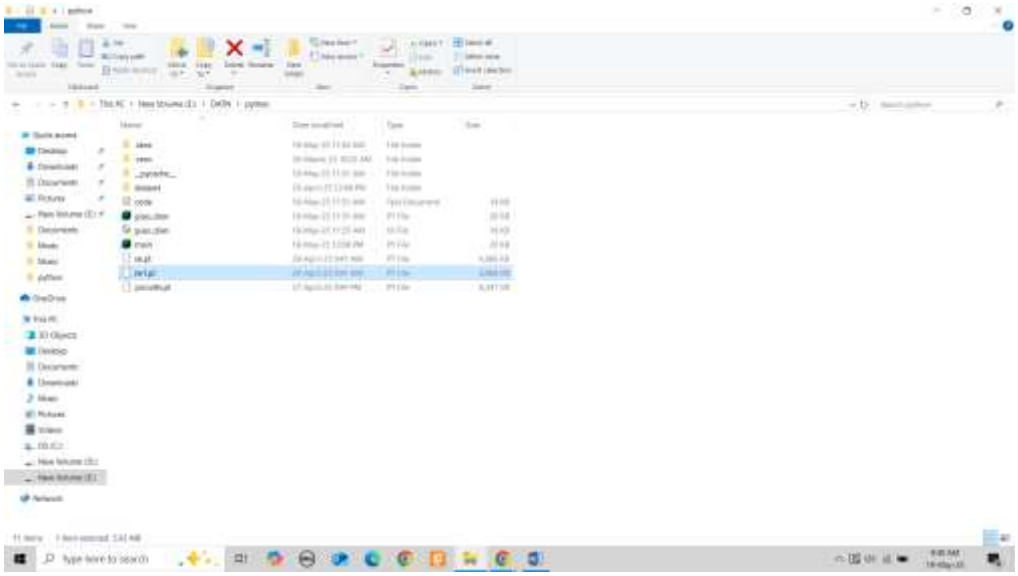


Hình 3. 13 Cấu trúc của file được lưu về máy

- Bước 6: Vào trang web YOLOv8, tải file vừa mới huấn luyện lên và chạy file.Sau đó lưu file vừa mới chạy về máy, sẽ có được file có dạng .pt. Đây là file mà YOLOv8 sẽ dùng để nhận diện.

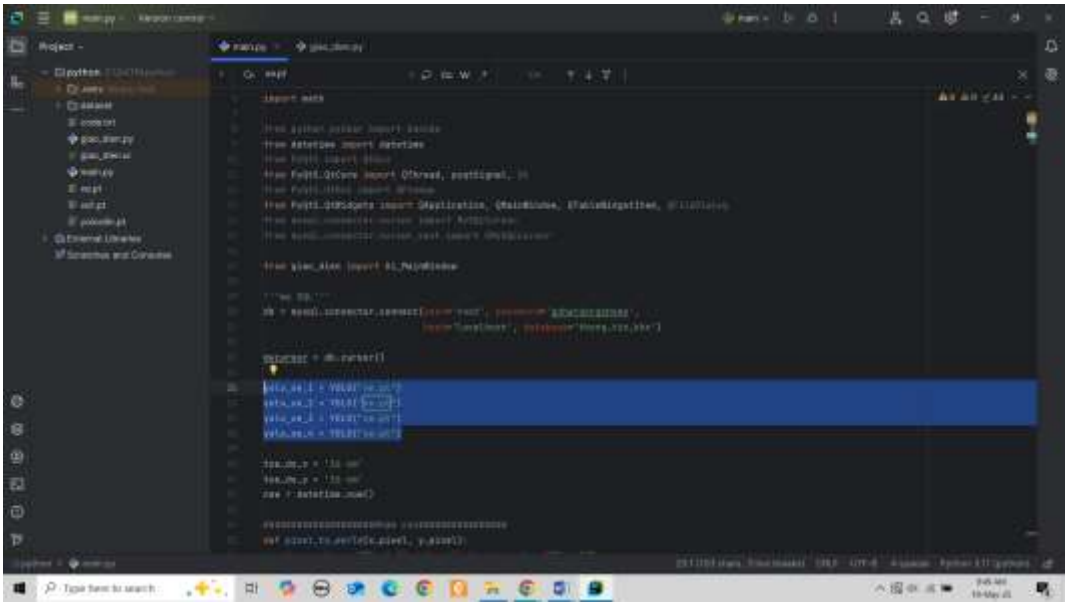


Hình 3. 14 Giao diện của trang web YOLOv8



Hình 3. 15 File được lưu sau khi train thành công

- Bước 7: Nhập code vào python để camera nhận diện xe nâng sử dụng YOLOv8. Vì dùng 4 camera khác nhau nên tạo 4 bản sao của mô hình ".pt". Mỗi biến sẽ dùng để xử lý ảnh từ một camera, giúp tránh xung đột nếu xử lý song song (Phần giải thích code chi tiết sẽ nằm ở phụ lục)



Hình 3. 16 Nhập code vào python để nhận diện vật thể

3.3 Xác định vị trí hàng hoá trong kho :

Hình 3. 17 Camera gắn phía trước kệ hàng

Giả sử kệ hàng gồm 3 tầng và mỗi tầng gồm 4 ô, kiện hàng có kích thước 80cm x 40cm x 40cm như đã đề cập ở đầu chương, vậy cần lắp camera sao cho có thể giám sát khu vực có kích thước tối thiểu 320cm x 120cm. Dùng công thức trường nhìn (FOV):

$$D = \frac{320}{2 * \tan\left(\frac{\theta}{2}\right)} \tag{3.36}$$

với D là khoảng cách từ camera đến kệ hàng, θ là góc nhìn ngang (FOV ngang) . Giả sử muốn đặt camera cách kệ hàng khoảng 1.5 đến 2.5 mét, thì khi đó góc nhìn ngang sẽ có giá trị từ 65^0 đến 94^0

Tương tự với góc nhìn dọc, ta có :

$$D = \frac{120}{2 * \tan\left(\frac{\theta}{2}\right)} \tag{3.37}$$

Khi đó, góc nhìn dọc sẽ có giá trị từ 27^0 đến 43^0

Dựa theo những yếu tố trên, có thể chọn loại camera Hikvision DS-2CD1023G0E-I. Với những thông số phù hợp :

- Độ phân giải : 1920 x 1080
- Góc nhìn ngang : 103^0
- Góc nhìn dọc : 55^0
- Chống nước bụi : Chuẩn IP67, chống nước và bụi tốt
- Giá thành rẻ : Khoảng 500.000 đến 1.000.000 VNĐ

Hình 3. 18 Camera Hikvision DS-2CD1023G0E-I

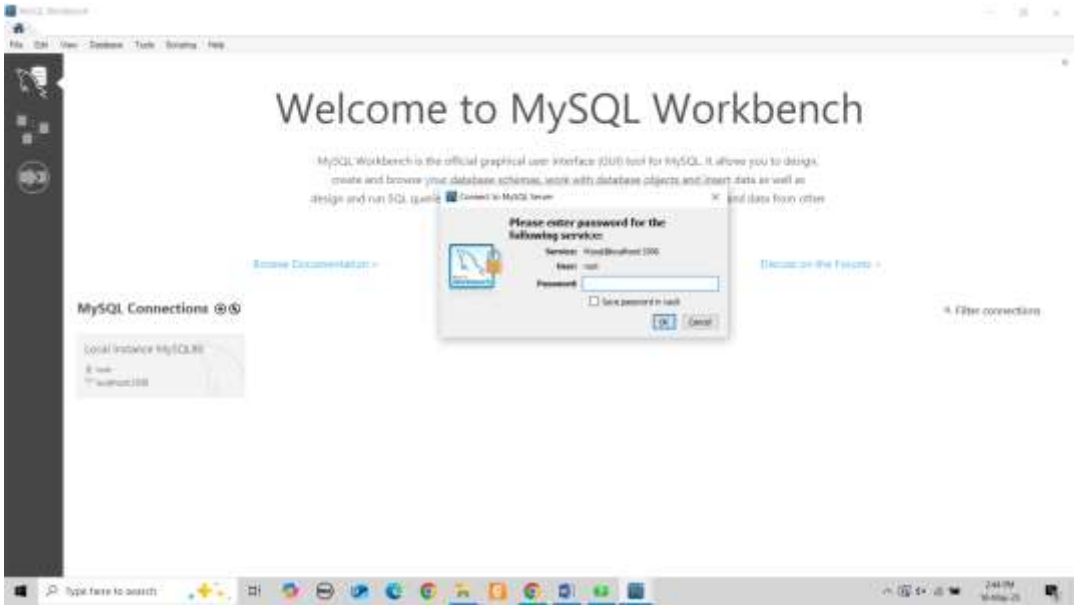
3.3.1 Phương pháp lưu thông tin hàng hoá trong kho:

Trước khi lưu thông tin hàng hoá, cần tạo mã QR với các dữ liệu về kiện hàng. Ghi những dữ liệu muốn được hiển thị khi quét mã QR, sau đó lưu mã QR về thiết bị,

Hình 3. 19 Tạo mã QR

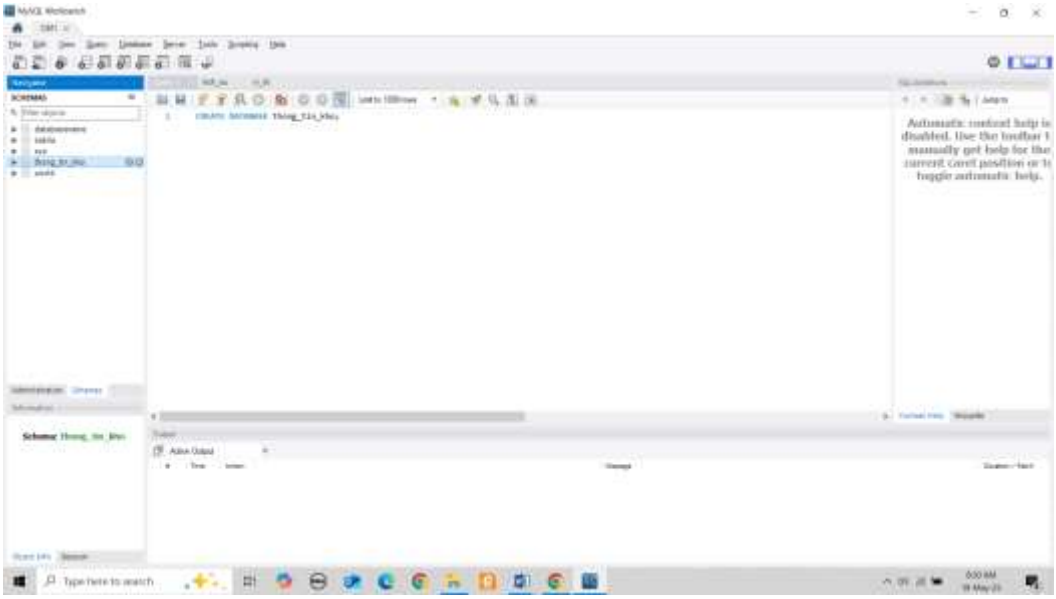
Việc lưu dữ liệu hàng hoá trong kho được thực hiện bởi phần mềm cơ sở dữ liệu MySQL. Các bước tiến hành như sau :

- Bước 1: Khởi động phần mềm MySQL, sau đó đăng nhập bằng tài khoản tạo trước.



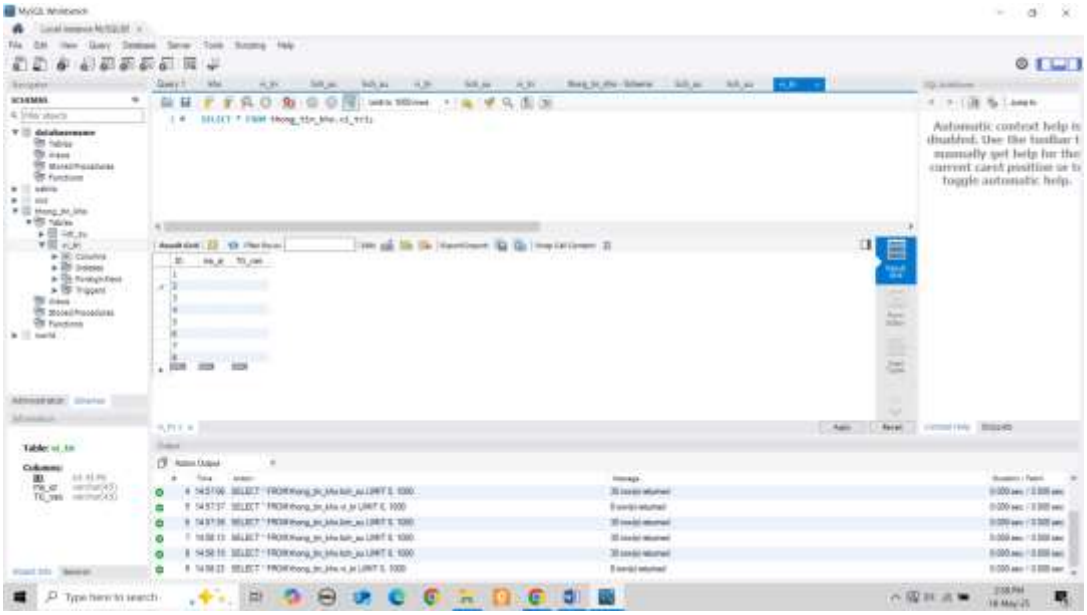
Hình 3. 20 Khởi động phần mềm MySQL

- Bước 2 :Tạo cơ sở dữ liệu đặt tên là thông tin kho, đây là cơ sở dữ liệu chính để làm việc.



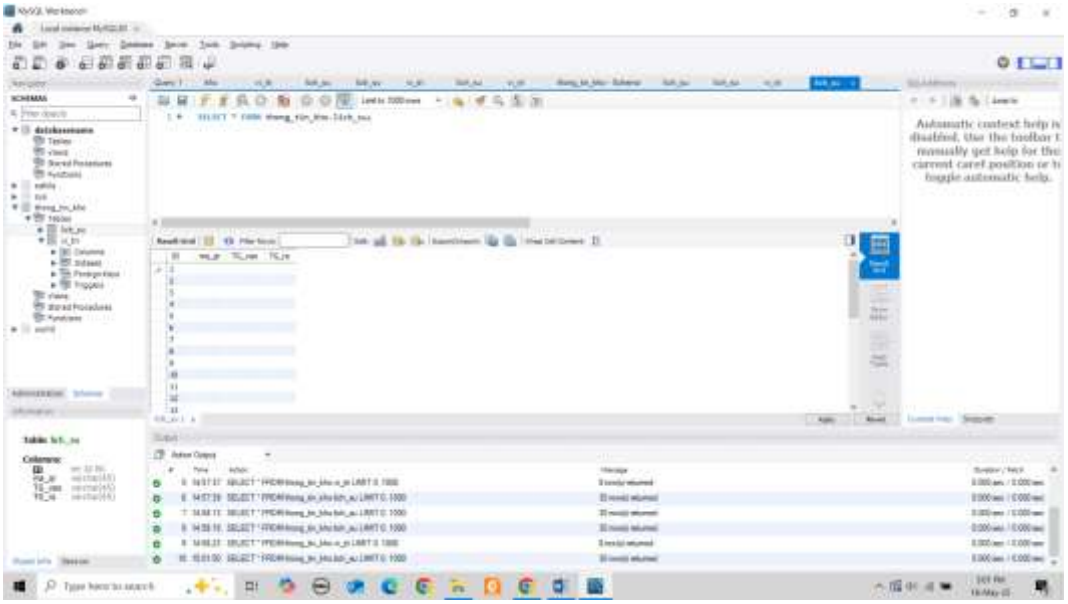
Hình 3. 21 Tạo cơ sở dữ liệu mới

- Bước 3: Tạo 2 bảng vị trí và lịch sử. Bảng vị trí có cấu trúc gồm 3 cột , ID, mã QR và thời gian nhập hàng. Khi một mã QR được quét và xác định vị trí đặt vào kho thì dữ liệu sẽ được đưa vào bảng này



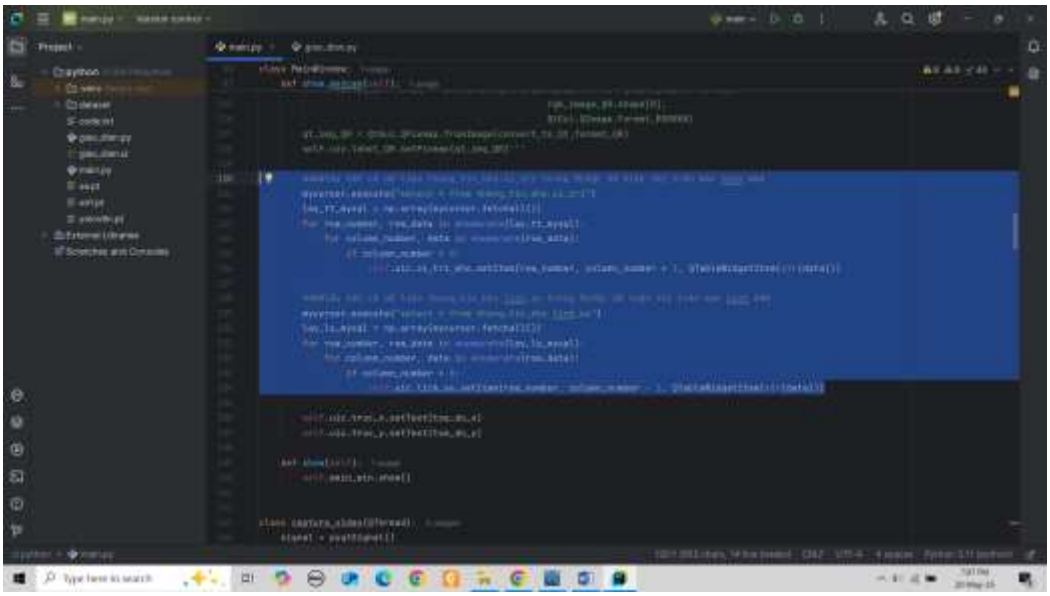
Hình 3. 22 Bảng "vị trí" trong cơ sở dữ liệu

Bảng lịch sử bao gồm 3 cột : Mã QR, thời gian nhập hàng, thời gian xuất hàng. Có thể xem đây là bảng theo dõi chuyện động của hàng hoá. Mỗi dòng dữ liệu trong bảng lịch sử đại diện cho một lần quét mã QR khi hàng hoá vào hoặc ra khỏi kho. Kết hợp với camera dùng YOLOv8 dữ liệu sẽ được tự động ghi vào bảng này.



Hình 3. 23 Bảng "lịch sử" trong cơ sở dữ liệu

- Bước 4: Nhập code vào python để lấy dữ liệu từ MySQL và hiển thị lên giao diện người dùng (Phần code sẽ được giải thích chi tiết ở phụ lục)



Hình 3. 24 Nhập code để hiển thị dữ liệu của MySQL lên giao diện

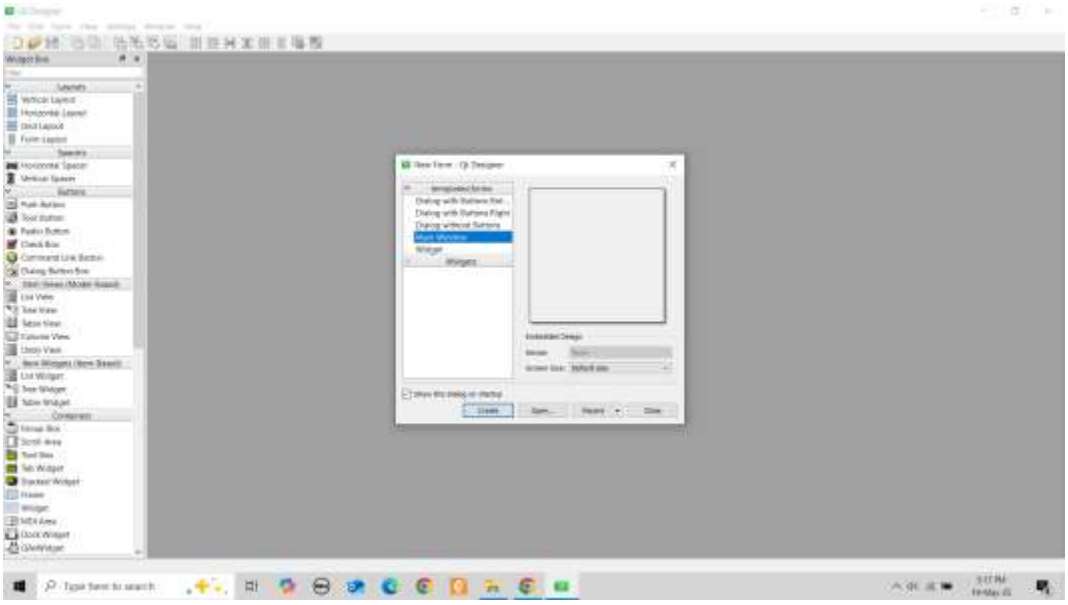
3.3.2 Thiết kế giao diện bằng Qt designer:

Việc sử dụng Qt designer để thiết kế giao diện mang lại nhiều lợi ích như :

- Tiết kiệm thời gian lập trình: Không cần viết từng dòng code để tạo nút
- Chỉ cần thả - kéo để tạo giao diện, dễ dàng điều khiển và sắp xếp
- Trực quan, dễ hình dung: Giao diện hiển thị sẽ giống như khi chạy chương trình thật
- Tách biệt giao diện và code: Không trộn lẫn file code với giao diện, giúp tổ chức mã tốt hơn, dễ bảo trì về sau
- Dễ dàng cập nhật, chỉnh sửa : Nếu cần thay đổi bố cục, chỉ cần mở lại phần mềm, chỉnh sửa và lưu, không cần phải sửa code phức tạp

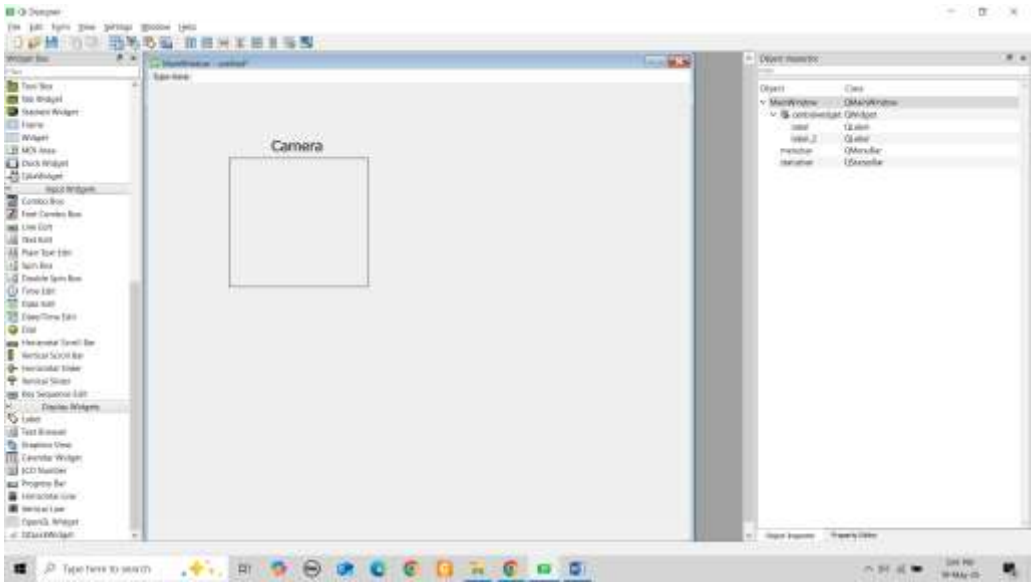
Các bước thiết kế giao diện trên Qt Designer:

- Bước 1: Mở phần mềm Qt Designer. Chọn Main Window để bắt đầu thiết kế giao diện



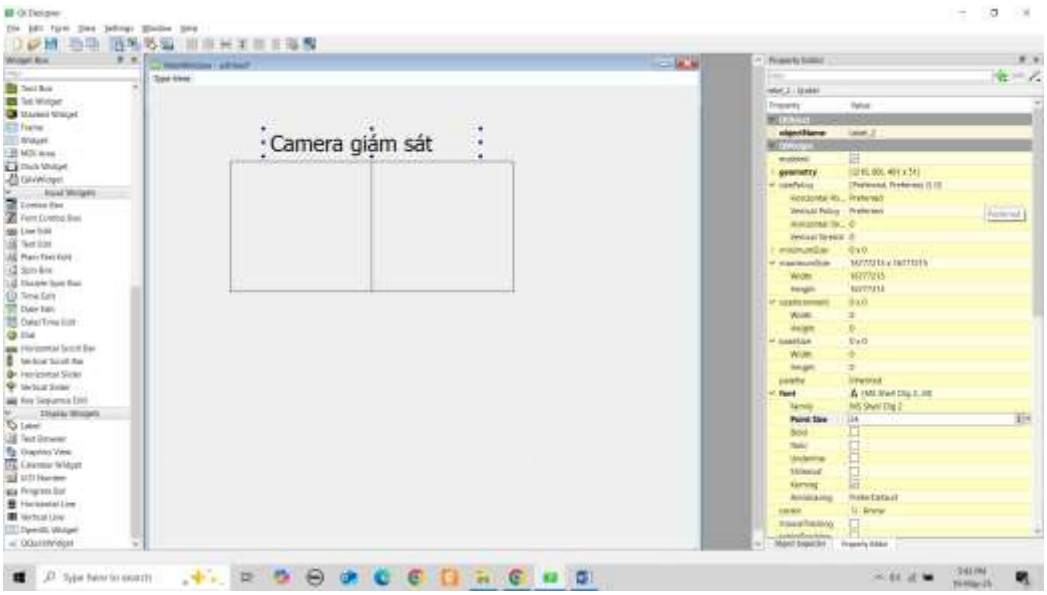
Hình 3. 25 Mở phần mềm Qt Designer

- Bước 2: Chọn Label để làm phần màn hình hiển thị hình ảnh từ camera, và văn bản chú thích. Label đóng vai trò như một "nhãn" để hiển thị thông tin cho người dùng



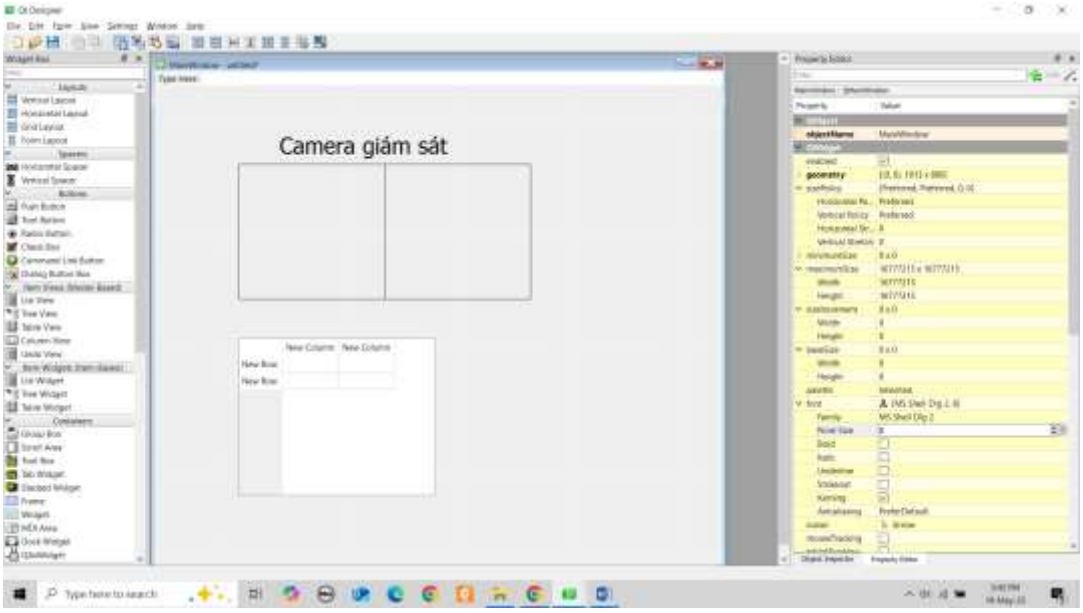
Hình 3. 26 Tạo các nhãn hiển thị

- Bước 3: Ở mục Property Editor, nằm bên phải góc dưới giao diện Qt Designer, cho phép người dùng xem và chỉnh sửa các thuộc tính như tên nhãn, chỉnh văn bản hiển thị, thêm hình ảnh, màu sắc, thiết lập hành vi



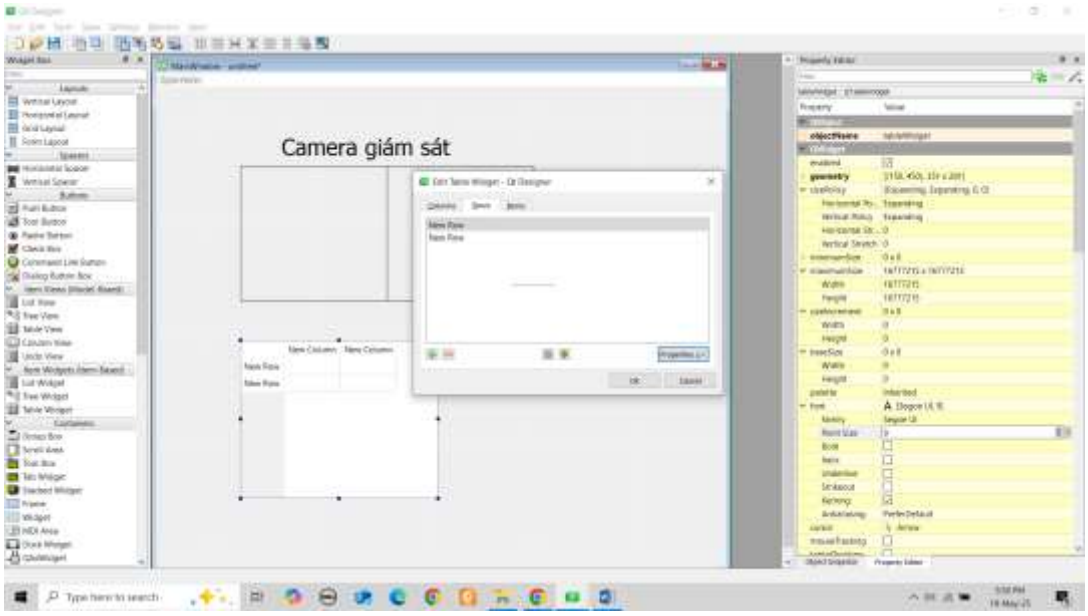
Hình 3. 27 Chỉnh sửa các thuộc tính của giao diện

- Bước 4: Để tạo bảng, chọn vào mục Table Widget



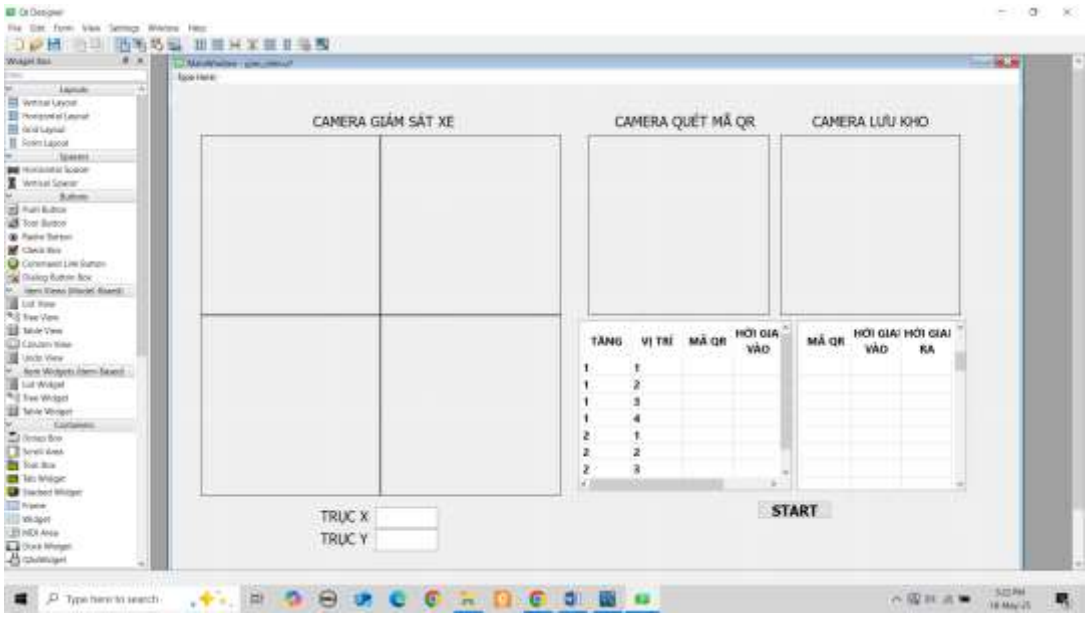
Hình 3. 28 Tạo bảng trong giao diện

Ở đây ta có thể tuỳ chỉnh số hàng, cột của bảng, và đặt tên cho chúng.



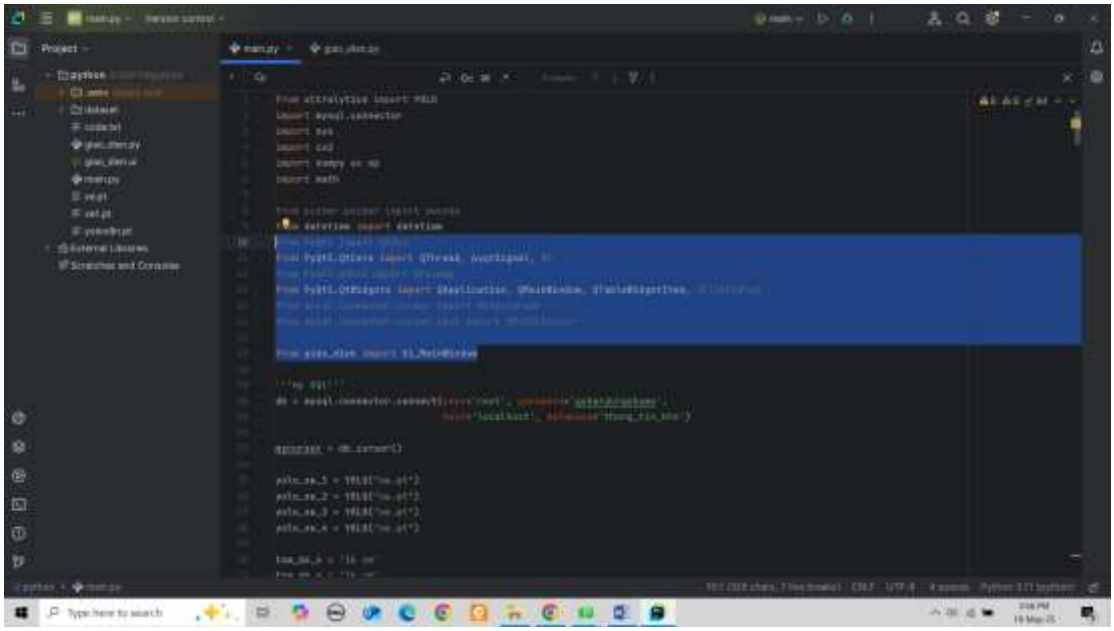
Hình 3. 29 Chỉnh sửa bảng trong giao diện

- Bước 5: Sau khi thiết kế xong, ta sẽ được màn hình như dưới đây. Ở màn hình giám sát sẽ bao gồm 4 màn hình hiển thị hình ảnh giám sát xe từ 4 camera đặt ở 4 góc nhà kho. 1 màn hình hiển thị mã QR quét được, và một màn hình hiển thị hình ảnh giám sát kho hàng. Ngoài ra sẽ có 2 bảng tương ứng là bảng vị trí, gồm vị trí của ô hàng, thông tin mã QR và thời gian nhập hàng. Bảng còn lại là bảng lịch sử, gồm thông tin hàng, và thời gian xuất/nhập hàng



Hình 3. 30 Giao diện hoàn chỉnh

- Bước 6: Gọi giao diện trong chương trình python (Phần giải thích code cụ thể sẽ nằm trong phụ lục)



Hình 3. 31 Gọi giao diện trong python

3.4 Quy trình vận hành hệ thống:

- Tổng hợp lại các thành phần dùng trong hệ thống :
  - Camera đặt ở 4 góc để quan sát xe nâng
  - Mã QR được dán lên hàng hoá để định danh sản phẩm. Dùng 1 camera để quét hàng và 1 camera để giám sát hàng ở trên kệ
  - Dùng python để nhận ảnh từ camera, tổng hợp, tính toán toạ độ xe nâng
  - Dùng YOLOv8 nhận diện xe nâng
  - Lưu thông tin hàng hoá vào phần mềm cơ sở dữ liệu MySQL
  - Hiển thị dữ liệu bao gồm toạ độ xe, thông tin hàng hoá lên màn hình hiển thị PyQt5
- Quy trình vận hành chi tiết như sau :

Đầu tiên khi khởi động hệ thống, phần mềm nhận diện python được chạy, hệ thống kết nối với các camera thông qua cổng USB, và kết nối với cơ sở dữ liệu MySQL. Các camera lấy ảnh đồng thời và truyền hình ảnh về python. Tiếp theo đó là quá trình nhận diện xe nâng bằng YOLOv8. Phần mềm sẽ phát hiện xe trong ảnh, và xác định vị

trí đối tượng bằng vùng giới hạn (bounding box). Dựa vào vị trí của vùng giới hạn mỗi ảnh để tính được toạ độ của xe nâng.

Mã QR sẽ được quét trước khi xe lấy hàng. Các thông tin của hàng hoá, bao gồm ID hàng hoá, tên hàng hoá, thời gian xuất nhập kho được lưu vào cơ sở dữ liệu MySQL. Các dữ liệu như hình ảnh thời gian thực từ camera, vị trí xe nâng, thông tin hàng mới quét sẽ hiển thị trên giao diện Qt Designer.

- Tiêu chí để lựa chọn phương pháp thiết kế hệ thống :
  - Tính chính xác định vị cao : 4 camera gắn ở 4 góc cho phép quan sát toàn bộ mặt phẳng, tránh điểm mù. Ngoài ra, giúp theo dõi chính xác vị trí xe theo thời gian thực. So với hệ thống dùng 1 camera trên cao, phương pháp này góc có thể bị ít che khuất hơn
  - Dễ triển khai và mở rộng : Mỗi camera hoạt động độc lập, dễ xử lý. Dùng YOLOv8 và python giúp dễ nâng cấp và chỉnh sửa nếu cần
  - Khả năng nhận diện mạnh mẽ : YOLOv8 có thể nhận diện vật thể theo thời gian thực, nhanh và chính xác. Nếu muốn nhận diện vật thể mới, có thể training lại
  - Tính linh hoạt của phần mềm : Phần mềm pycharm dễ viết chương trình xử lý ảnh, giao diện và tính toán toạ độ. Dễ tích hợp thư viện đọc mã QR, và có thể phát triển giao diện GUI (với PyQt5) nhanh chóng.
  - Lưu trữ và truy xuất hiệu quả : Dùng MySQL hỗ trợ truy xuất nhanh, cấu trúc rõ ràng. Có thể truy vấn nhiều lệnh dễ dàng (Tìm hàng theo mã QR, xem lịch sử xuất nhập kho, xem vị trí hiện tại của kho hàng). Dễ kết nối với những giao diện web, công cụ báo cáo,…
  - Chi phí thấp : Dùng camera phổ thông, phần mềm mã nguồn mở.
  - Đảm bảo tính an toàn lao động : Đặt camera ở vị trí có thể dễ dàng vệ sinh, không ảnh hưởng đến môi trường làm việc xung quanh

3.5 Kết luận :

Chương 3 đã đi tính toán hệ thống với số liệu cụ thể, và nêu các bước để thiết kế hệ thống, cách huấn luyện để nhận diện xe, cách lưu thông tin hàng hoá cũng như thiết kế giao diện hiển thị

Chương 4 sẽ đi vào phần mô phỏng và nhận xét kết quả cho mô hình của hệ thống, kèm theo phụ lục giải thích code được dùng trong dự án.

CHƯƠNG 4 : MÔ PHỎNG VÀ THỰC NGHIỆM

4.1 Xây dựng và tính toán mô hình cho hệ thống :

4.1.1 Mô hình của hệ thống:

Để chứng minh hoạt động của hệ thống thì nhóm đã lắp đặt một mô hình có kích thước mô phỏng quá trình theo dõi toạ độ xe nâng và hàng hoá ở trong kho, mô hình bao gồm :

- Sàn kho có kích thước 60cm x 80cm
- 4 camera dùng để giám sát và tính toán toạ độ được lắp đặt ở 4 góc của nhà kho.
- Kệ hàng gồm 2 tầng, mỗi tầng 4 ô chứa hàng
- Một camera đặt trước kệ hàng để giám sát hàng hoá

Hình 4. 1 Mô hình hệ thống

Như đã đề cập ở chương 3, cần xác định kích thước vùng cần camera quan sát vì trường nhìn của camera có giới hạn. Camera chỉ có thể nhìn thấy một vùng nhất định, càng đặt camera xa vùng cần nhìn, thì vùng bao phủ càng lớn, tuy nhiên độ chi tiết lại giảm. Nên cần xác định trước kích thước vùng mong muốn quan sát được để tính được nên đặt camera cao bao nhiêu là đủ

Với kích thước 60 x 80 cm của mô hình nhà kho, chọn góc nhìn của mỗi camera bao trọn vùng có chiều ngang 65 cm.

Vì muốn camera bao trọn vùng có chiều ngang 65 cm, nên $D = 65$

Camera trong mô hình dùng có trường nhìn ngang 100^0 và trường nhìn dọc 84^0

Ta có: $\tan\left(FOV'\right) = \dfrac{\dfrac{D}{2}}{H} \Rightarrow D = 2 \times H \times \tan(FOV')$ (3.1)

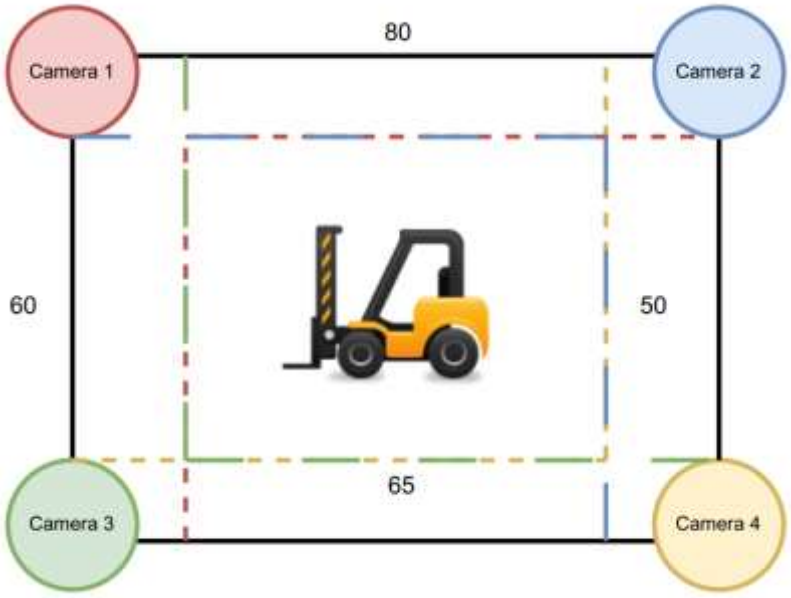
$$\Rightarrow H = \frac{D}{2 \times \tan\left(FOV'\right)} = \frac{65}{2 \times \tan\left(50\right)} \approx 27.3\left(cm\right) \quad (3.2)$$

Khi đó, chiều rộng quan sát được của camera khi lắp đặt ở độ cao này là

$$\mathrm{W} = 2 \times H \times \tan\left(\frac{84}{2}\right) = 2 \times 27.3 \times \tan\left(42\right) \approx 50\left(cm\right) \quad (3.3)$$

Vậy khi lắp đặt camera ở độ cao khoảng 27.3 cm, thì sẽ quan sát được vùng có kích thước là 50 x 65 cm

Hình 4.2 thể hiện cấu trúc mô hình cũng như vùng bao phủ của các camera

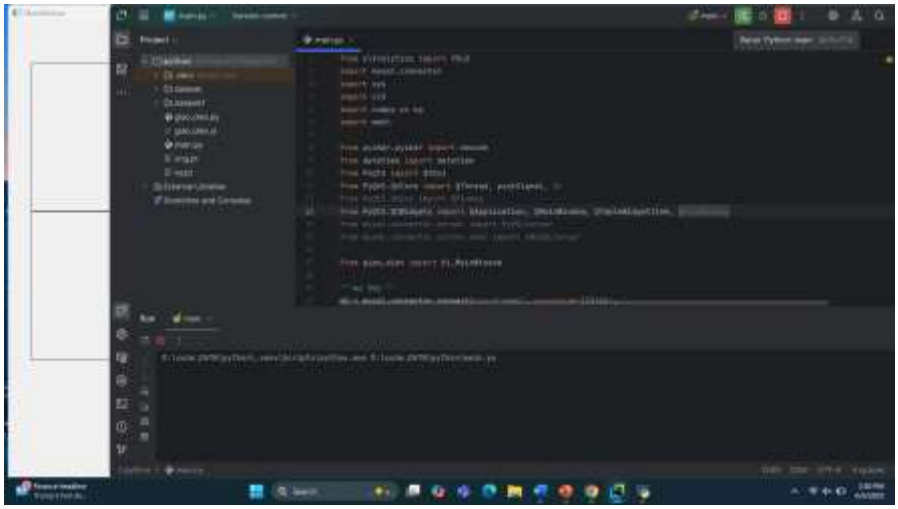


Hình 4. 2 Sơ đồ thể hiện góc quan sát của camera

Ở vùng trung tâm kho là khu vực xe nâng được giám sát bởi 4 camera, các vùng bên ngoài sẽ được giám sát bởi 2 camera, và cuối cùng là ở khu vực dưới chân các camera, xe nâng sẽ được giám sát bởi camera ở phía đối diện.

4.1.2 Tiến hành mô phỏng :

- Bước 1 : Chạy chương trình chính



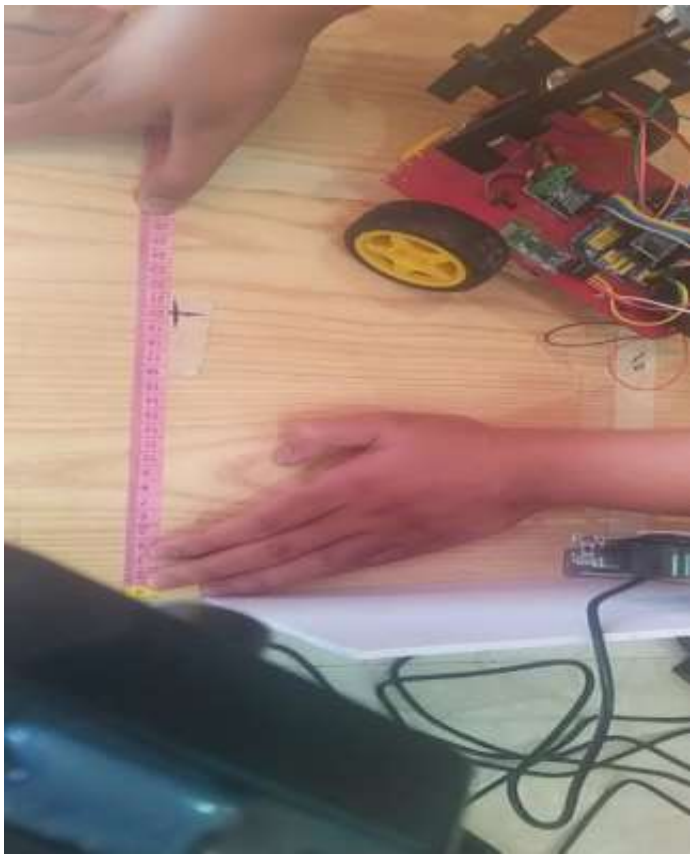
Hình 4. 3 Chạy chương trình chính

- Bước 2 : Bấm nút Start, các camera bắt đầu hoạt động, xác định và tính toán toạ độ của xe dựa trên hình ảnh thu về từ các camera

Hình 4. 4 Chạy các camera xác định toạ độ

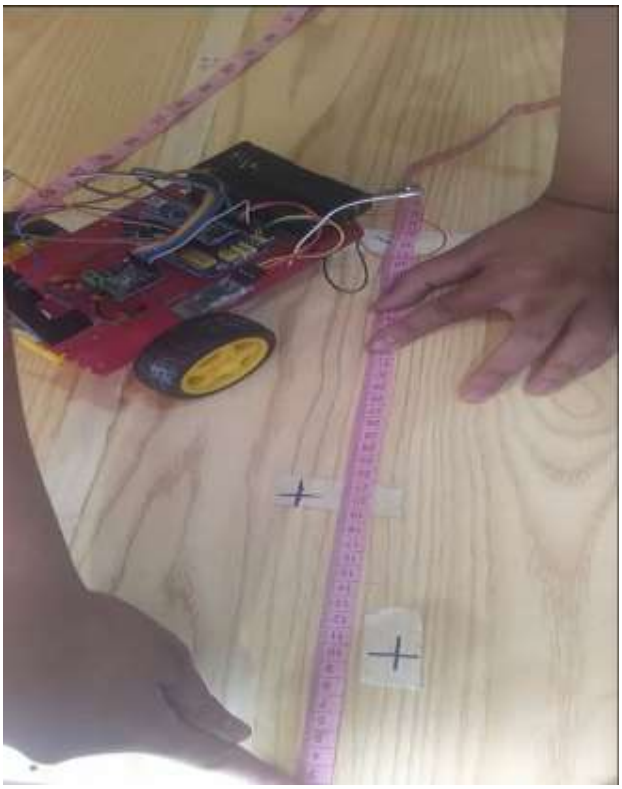
Trong lần mô phỏng này, camera hiển thị toạ độ của xe trên mặt phẳng là trục x: 26cm và trục y 29cm

- Bước 3 : Kiểm tra độ chính xác của toạ độ bằng cách đo thủ công



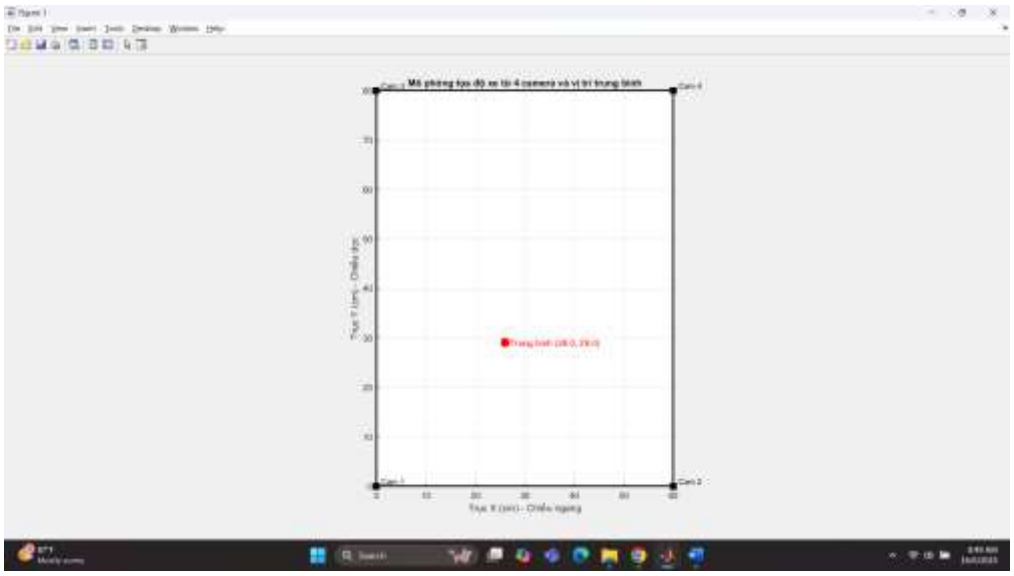
Hình 4. 5 Kiểm tra độ chính xác của trục X

Khoảng cách từ tâm của xe đến trục toạ độ theo trục x có giá trị gần đúng với 26cm



Hình 4. 6 Kiểm tra độ chính xác của trục Y

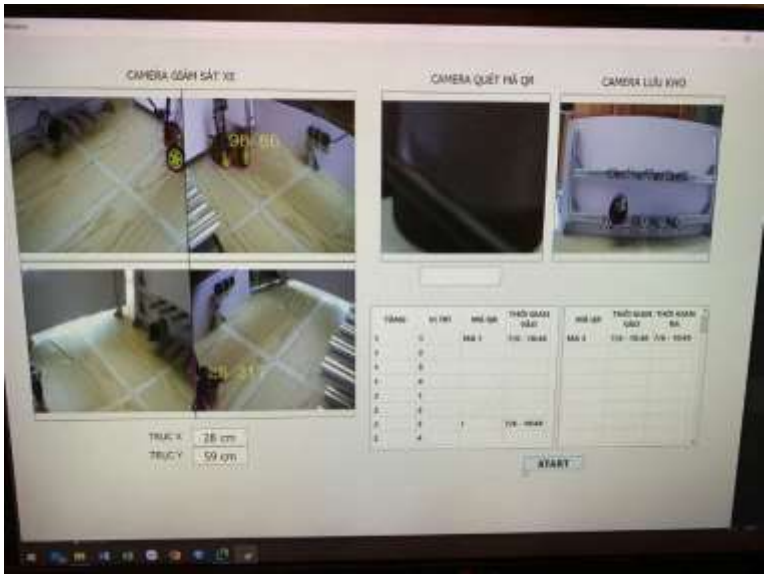
Khoảng cách từ tâm chiếc xe đến gốc toạ độ theo trục y có giá trị gần đúng với 29cm.



Hình 4. 7 Mô phỏng Matlab vị trí của xe

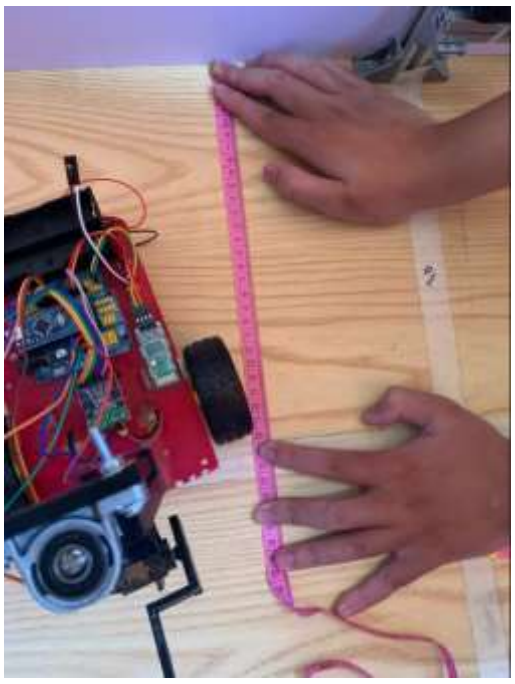
Hình ảnh mô phỏng trên Matlab vị trí của xe

- Kiểm tra độ chính xác trường hợp xe nằm ở vị trí dễ bị khuất



Hình 4. 8 Xe nằm ở vị trí khuất

Trong trường hợp này, toạ độ trả về là 28 cm với trục x, và 59cm ở trục y

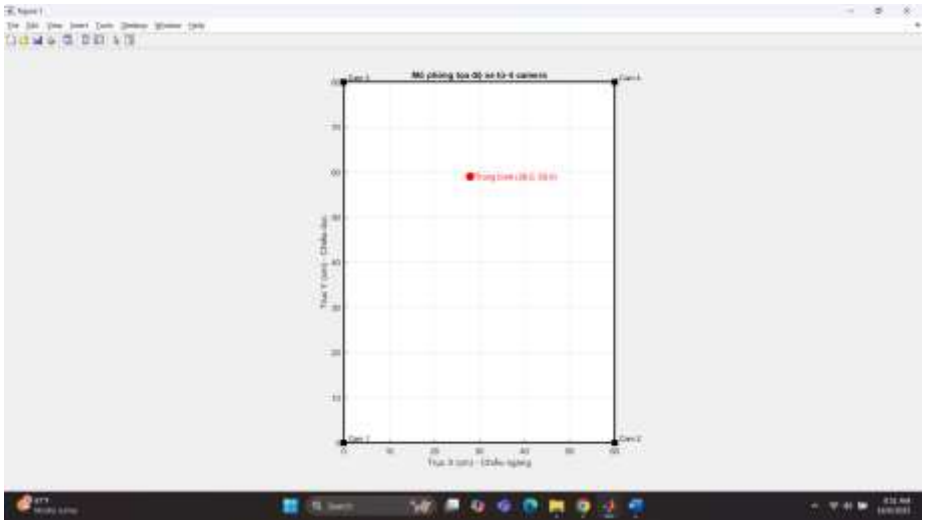


Hình 4. 9 Kiểm tra độ chính xác của trục x

Kiểm tra toạ độ thực tế, khoảng cách từ tâm xe đến gốc toạ độ theo trục x có giá trị gần đúng 28 cm

Hình 4. 10 Kiểm tra độ chính xác theo trục y

Kiểm tra toạ độ thực tế, khoảng cách từ tâm xe đến trục toạ độ theo trục y có giá trị sai số không đáng kể



Hình 4. 11 Mô phỏng Matlab vị trí của xe

Hình ảnh trên Matlab mô phỏng vị trí của xe

Kết quả xác định toạ độ xe từ hình ảnh camera cho thấy độ chính xác tương đối cao, với sai số nhỏ so với giá trị thực tế. Chứng tỏ hệ thống xử lý ảnh hoạt động ổn định và phương pháp tính toán toạ độ thực bằng ma trận đồng nhất là hiệu quả.

- Bước 4 : Tiến hành lưu trữ hàng hoá bằng mã QR

Hình 4. 12 Quét mã QR

Đầu tiên dùng camera để quét mã QR được dán trên hàng.Khi camera quét mã được in trên hàng, hệ thông sẽ nhận diện loại mã được thiết kế sẵn của kiện hàng

Khi quét, hệ thống nhận diện và hiện ra tên loại mã ở trên màn hình giám sát, sau đó xe sẽ đưa hàng vào ô trống

Hình 4. 13 Hiển thị tên loại hàng

Hình 4. 14 Hệ thống lưu hàng vào đúng vị trí

Khi hàng được bỏ và kho, hệ thống lập tức ghi vị trí mà kiện hàng vừa được quét mã nằm trong. Ở trường hợp này, kiện hàng được đưa vào ô số 1, tầng 1, nên ở trong bảng dữ liệu lưu loại hàng "Ma 1" vào ô tương ứng, kèm theo là ngày và giờ được đưa vào kho

Tiếp theo, quét mã QR cho hàng tiếp theo, loại mã lần này là "Ma 3"

Hình 4. 15 Hiển thị tên loại hàng

Hình 4. 16 Hệ thống lưu hàng vào đúng vị trí

Khi xe đặt hàng vào trong vị trí số 3, tầng 2, lập tức hệ thống lưu thông tin của hàng vào vị trí tương ứng

Hình 4. 17 Hệ thống lưu hàng được lấy ra khỏi kho

Khi hàng được lấy ra khỏi kho, thông tin sẽ được lưu vào bảng dữ liệu, gồm loại hàng, thời gian nhập hàng vào trong kho và thời gian lấy hàng ra khỏi kho. Ngoài ra dữ

liệu ở bảng nhập kho sẽ bị xoá đi, để người dùng biết rằng hàng này không còn nằm trong kho

Kiểm tra hoạt động của hệ thống khi lưu nhiều hàng cùng một lúc. Lưu 2 kiện hàng ở tầng 1 và 2 kiện hàng ở tầng 2, sau đó lấy mỗi kiện hàng ở mỗi tầng ra

Hình 4. 18 Hệ thống lưu hàng được đưa vào

Hình 4. 19 Hệ thống ghi nhận hàng được lấy ra

→ Kết luận : Quá trình lưu trữ hàng hoá khi xuất và nhập kho diễn ra ổn định

4.2 Nhận xét về kết quả :

Nhóm đã dùng mô hình để mô phỏng quá trình theo dõi xe nâng và tính toán toạ độ của xe, cùng với đó là giám sát kho hàng bằng việc lưu dữ liệu xuất nhập vào cơ sở dữ liệu. Qua quá trình thử nghiệm, hệ thống cho thấy kêt quả định vị có độ chính xác cao so với giá trị thực tế, đảm bảo yêu cầu về độ chính xác

Đồng thời, mô hình đã được tích hợp với phần mềm cơ sở dữ liệu MySQL giúp quản lý thông tin hàng hoá một cách hiệu quả. Việc lưu trữ dữ liệu diễn ra ổn định và chính xác.

4.3 Kết luận :

Ở chương 4 này, nhóm đã tiến hành mô phỏng hệ thống bằng việc xây dựng một mô hình nhỏ, tiến hành chạy chương trình để theo dõi hoạt động và kiểm tra độ tin cậy của hệ thống. Đồng thời cũng thử nghiệm hệ thống với nhiều trường hợp khác nhau có thể xảy ra tại nhà kho thực tế.

KẾT LUẬN

Sau khi kết thúc quá trình nghiên cứu và thực hiện đề tài "Giải pháp tự động hoá xuất nhập kho hàng", nhóm chúng em đã thực hiện được việc xây dựng hệ thống nhận diện, tính toán chính xác toạ độ thực tế của xe nâng từ hình ảnh các camera, theo dõi vị trí hàng hoá trong kho, cũng như lịch sử xuất nhập hàng

Những đóng góp của đề tài bao gồm :

- Về mặt thực tiễn : Đề tài có tính ứng dụng cao, có thể mở rộng và áp dụng cho nhà kho trong thực tế
- Về mặt kỹ thuật công nghệ : Kết hợp nhiều phần mềm công nghệ tiên tiến khác nhau như YOLOv8, Qt Designer, cơ sở dữ liệu MySQL
- Về mặt học thuật : Ứng dụng thành công thị giác máy tính, biến đổi phối cảnh trong bài toán thực tế

TÀI LIỆU THAM KHẢO

[1] Ha Phan, "Phân biệt điểm giống nhau và khác nhau giữa Barcode và QR Code",HaPhanCompany,27/06/2022, [Online], truy cập tại: https://www.haphan.com/News/41140/phan-biet-diem-giong-nhau-va-khac-nhau-giua-barcode-va-qr-code , [Truy cập ngày : 25/02/2025]

[2] FPTC, "Lưu trữ đám mây và lưu trữ cục bộ: Cái nào tốt hơn?", fptcamera, 05/09/2023, [Online], truy cập tại: https://fptcamera.com.vn/luu-tru-dam-may-va-luu-tru-cuc-bo-cai-nao-tot-hon?srsltid=AfmBOop09WzXtCkqFjpu-9GeR4Nt0B9Xbo4KGl7ffJv226JpklA5HG7j , [Truy cập ngày : 26/02/2025]

[3] Tuan Nguyen, "Lưu trữ đám mây là gì? Những lợi ích mà lưu trữ đám mây mang lại", fptshop, 2023, [Online], truy cập tại: https://fptshop.com.vn/tin-tuc/danh-gia/luu-tru-dam-may-la-gi-nhung-loi-ich-ma-luu-tru-dam-may-mang-lai-152628 , [Truy cập ngày: 26/02/2025]

[4] Lê Chi, "Phần mềm quản lý cơ sở dữ liệu là gì? Các tính năng chính và ưu điểm", fptshop, 2023, [Online], truy cập tại: https://fptshop.com.vn/tin-tuc/thu-thuat/phan-mem-quan-ly-co-so-du-lieu-151986 , [Truy cập ngày: 26/02/2025]

[5] Nguyễn Minh Hải, "Computer Vision là gì? Lợi ích và ứng dụng nổi bật của thị giác máy tính", VNPT AI, 18/01/2025, [Online], truy cập tại: https://vnptai.io/vi/blog/detail/computer-vision-la-gi , [Truy cập ngày: 26/02/2025]

[6] Alexander S.Gillis, "What is a pixel?", TechTarget, 30/08/2022, [Online], truy cập tại: https://www.techtarget.com/whatis/definition/pixel , [Truy cập ngày: 27/02/2025]

[7] FPTC, "Trường nhìn (FOV) là gì: Tìm hiểu những điều cơ bản", fptcamera, 29/10/2023, [Onine], truy cập tại: https://fptcamera.com.vn/truong-nhin-fov-la-gi-tim-hieu-nhung-dieu-co-ban?srsltid=AfmBOorY3ybDupnSpucjCuEn-uPaFsDZhNmhKMOUM4B8D-T6kC4M3FyC , [Truy cập ngày: 28/02/2025]

[8] Geeksforgeeks, "What is Homography? How to estimate homography between two images?" , Geeksforgeeks, 24/09/2024, [Online], truy cập tại:

https://www.geeksforgeeks.org/computer-vision/what-is-homography-how-to-estimate-homography-between-two-images/ , [Truy cập ngày: 05/03/2025]

[9] Tô Thị Hát, "Pycharm là gì? Gợi ý cách cài đặt và sử dụng Pycharm đơn giản và nhanh chóng", Fptshop, 2024, [Online], truy cập tại: https://fptshop.com.vn/tin-tuc/thu-thuat/pycharm-167326 , [Truy cập ngày: 07/03/2025]

[10] Vinbigdata, "YOLOv8 có gì nâng cấp so với các phiên bản trước?", Vinbigdata, 16/08/2023, [Online], truy cập tại: https://vinbigdata.com/cong-nghe-hinh-anh/yolov8-co-gi-nang-cap-so-voi-cac-phien-ban-truoc.html , [Truy cập ngày: 07/03/2025]

[11] Gia Hân, "Tìm hiểu MySQL là gì? Cơ chế hoạt động và cách thức cài đặt MySQL trên Window và Server", Fptshop, 2024, [Online], truy cập tại: https://fptshop.com.vn/tin-tuc/danh-gia/tim-hieu-mysql-la-gi-172297 , [Truy cập ngày: 07/03/2025]

[12] Leonadis Pozo Ramos, "Qt Designer and Python: Build Your GUI Applications Faster", Real Python, [Online], truy cập tại: https://realpython.com/qt-designer-python/ , [Truy cập ngày: 09/03/2025]

Phụ lục

Giải thích code chi tiết

```
from ultralytics import YOLO # Import mô hình YOLOv8 từ thư viện ultralystic, dùng để nhận diện vật thể
import mysql.connector # Kết nối cơ sở dữ liệu MySQL
import sys # Dùng để tương tác với trình thông dịch (thoát chương trình, xử lý dòng lệnh)
import cv2 # Thư viện OpenCV, dùng để xử lý hình ảnh từ các camera
import numpy as np # Xử lý mảng số
import math # Dùng các hàm toán học

from pyzbar.pyzbar import decode # Dùng thư viện pyzbar để đọc mã QR
from datetime import datetime # Dùng để lấy thời gian thực
from PyQt5 import QtGui # Dùng để khởi tạo giao diện người dùng
from PyQt5.QtCore import QThread, pyqtSignal, Qt # Xử lý nhiều luồng chạy song song
from PyQt5.QtGui import Qpixmap # Xử lý hình ảnh để hiển thị lên giao diện
from PyQt5.QtWidgets import QApplication, QMainWindow, QTableWidgetItem, QfileDialog # Thêm giữ liệu vào bảng
from mysql.connector.cursor import MySQLCursor # Kết nối và thao tác với CSDL
from mysql.connector.cursor_cext import CMySQLCursor # Kết nối và thao tác với CSDL

from giao_dien import Ui_MainWindow # Nhập giao diện người dùng được tạo

db = mysql.connector.connect(user='root', password='123456',
                             host='localhost', database='thong_tin_kho') # Kết nối đến tài khoản CSDL

mycursor = db.cursor() # Nhận diện con trỏ để thực thi các lệnh SQL

yolo_xe = YOLO("xe.pt") # Tải mô hình YOLO để nhận diện xe
yolo_ong = YOLO("ong.pt") # Tải mô hình YOLO để nhận diện hàng

toa_do_x = 0 # Lưu vị trí x hiện tại của đối tượng
```

```
toa_do_y = 0 # Lưu vị trí x hiện tại của đối tượng
now = datetime.now() # Lưu thời điểm hiện tại

vi_tri = [0,0,0,0,0,0,0,0] # Mảng chứa trạng thái của 8 vị trí ô hàng
vi_tri_SS = [0,0,0,0,0,0,0,0] # Trạng thái so sánh, dùng để kiểm tra thay đổi giữa 2 lần quét mã
vi_tri_ao = [0,0,0,0,0,0,0,0] # Mảng dùng để xử lý trung gian

x = [0,0,0,0] # Lưu toạ độ x của vật thể từ 4 camera
y = [0,0,0,0] # Lưu toạ độ y của vật thể từ 4 camera

global ma_qr # Mã QR đọc được
x_TB = 0 # Toạ độ x trung bình
y_TB = 0 # Toạ độ y trung bình
i = 0 # Biến đệm

bien_vao = 0 # Đếm số hàng đi vào kho
bien_ra = 0 # Đếm số hàng đi ra khỏi kho

width, height = 300, 420 # Kích thước ảnh sau khi biến đổi phối cảnh
pts2 = np.float32([[0, 0], [width, 0], [0, height], [width, height]]) # Ma trận sau khi biến đổi

pts1_1 = np.float32([[224, 7], [315, 65], [108, 67], [210, 164]]) # Toạ độ pixel của 4 điểm góc từ ảnh camera 1
matrix_1 = cv2.getPerspectiveTransform(pts1_1, pts2) # Ma trận biến đổi dùng để chuyển vùng nhìn xiên từ camera thành 1 vùng nhìn thẳng

pts1_2 = np.float32([[54, 108], [143, 63], [147, 205], [248, 119]]) # Toạ độ pixel của 4 điểm góc từ ảnh camera 2
matrix_2 = cv2.getPerspectiveTransform(pts1_2, pts2) # Ma trận biến đổi dùng để chuyển vùng nhìn xiên từ camera thành 1 vùng nhìn thẳng

pts1_3 = np.float32([[166, 19], [257, 56], [42, 105], [155, 185]]) # Toạ độ pixel của 4
```

```
điểm góc từ ảnh camera 3
matrix_3 = cv2.getPerspectiveTransform(pts1_3, pts2) # Ma trận biến đổi dùng để chuyển vùng nhìn xiên từ camera thành 1 vùng nhìn thẳng

pts1_4 = np.float32([[94, 111], [190, 83], [197, 252], [311, 178]]) # Toạ độ pixel của 4 điểm góc từ ảnh camera 4
matrix_4 = cv2.getPerspectiveTransform(pts1_4, pts2) # Ma trận biến đổi dùng để chuyển vùng nhìn xiên từ camera thành 1 vùng nhìn thẳng

def vao(): # Phát hiện có đội tượng vào vùng lưu trữ
    now = datetime.now() # Lấy thời gian thực
    global ma_qr # Mã QR được quét
    for n in range(8):
        if vi_tri[n] != vi_tri_SS[n]:
            vi_tri[n] = vi_tri_SS[n]
            mycursor.execute("UPDATE thong_tin_kho.vi_tri SET ma_qr = %s, TG_vao = %s where ID = %s",
                             (ma_qr, str(now.day) + '/' + str(now.month) + ' - ' + str(now.hour) + ':' + str(now.minute), n + 1))
            db.commit()
            break

    ma_qr = " "

#So sánh vi_tri và vi_tri_SS để phát hiện ô có hàng được nhập vào ; cạp nhật lại vi_tri, xoá mã qr và thời gian vào
def ra(): # Phát hiện đối tượng ra khỏi vùng lưu trữ
    now = datetime.now() # Lấy thời gian thực
    for n in range(8):
        if vi_tri[n] != vi_tri_SS[n]:
            vi_tri[n] = vi_tri_SS[n]
            mycursor.execute("select * from thong_tin_kho.vi_tri where ma_qr > %s and ID = %s",
                             ("", n + 1))
```

```
            result = np.array(mycursor.fetchall())

            mycursor.execute("UPDATE thong_tin_kho.vi_tri SET ma_qr = %s, TG_vao
= %s where ID = %s",
                ("", "", n + 1))
            db.commit()

        #So sánh vi_tri và vi_tri_SS để phát hiện có hàng rời đi ; tìm dòng trong vi_tri
có mã qr của hàng vừa xuất, cập nhật lại vi_tri.

            mycursor.execute('select * from thong_tin_kho.lich_su where ma_qr = """) #
Cập nhật bảng lich_su
            result_ls = np.array(mycursor.fetchall())
            mycursor.execute("UPDATE thong_tin_kho.lich_su SET ma_qr = %s, TG_vao
= %s, TG_ra = %s where ID = %s",
                (str(result[0][1]), str(result[0][2]), str(now.day) + '/' + str(now.month) + ' - ' +
str(now.hour) + ':' + str(now.minute), int(result_ls[0][0]))) # Giá trị thời gian thực gồm
ngày, tháng , giờ, phút
            db.commit()
            break

class MainWindow: # Khởi tạo giao diện
    def __init__(self):
        self.main_win = QMainWindow() # Tạo cửa sổ chính
        self.uic = Ui_MainWindow() # Gọi lớp thiết kế giao diện
        self.uic.setupUi(self.main_win) # Liên kết layout thiết kế với cửa sổ chính

        self.uic.Button_start.clicked.connect(self.start_capture_video) # Bắt đầu lấy hình
ảnh từ camera khi bấm nút start

        self.thread = {} # Dùng để lưu nhiều luồng từ các camera khác nhau

    def start_capture_video(self): # Bắt đầu nhận diện ảnh
        self.thread[1] = capture_video(index=1)
        self.thread[1].start() # Khởi động luồng
```

```
        cv2.waitKey(5) # Đợi 5ms
        self.thread[1].signal.connect(self.show_wedcam) # Kết nối tín hiệu từ luồng về giao diện

        global vi_tri
        mycursor.execute('select * from thong_tin_kho.vi_tri where ma_qr != """) # Lấy dữ liệu từ MySQL
        result = np.array(mycursor.fetchall()) # Truy vấn các vị trí đã có hàng
        #print(len(result), vi_tri)
        if len(result) > 0 :
            for n in range(len(result)):
                vi_tri[int(result[n][0]) - 1] = 1
        #print(len(result), vi_tri)  # Cập nhật lại vị trí, chuyển ID từ 1-based (SQL) sang 0-based (Python)

def show_wedcam(self, cv_img_xe_1, cv_img_xe_2, cv_img_xe_3, cv_img_xe_4, cv_img_qr, cv_img_kho): # Nhận hình ảnh từ 6 camera (4 camera xe, 1 camera quét QR, 1 camera giám sát kho)
    #cv2.waitKey(5)

    rgb_image_xe_1 = cv2.cvtColor(cv_img_xe_1, cv2.COLOR_BGR2RGB) # OpenCV dùng định dạng màu BGR, chuyển đổi sang RGB để phù hợp với giao diện hiển thị
    convert_to_Qt_format_xe_1 = QtGui.QImage(rgb_image_xe_1.data, rgb_image_xe_1.shape[1],rgb_image_xe_1.shape[0],QtGui.QImage.Format_RGB888) # Tạo Qimage từ dữ liệu ảnh gốc của camera 1
    qt_img_xe_1 = QtGui.QPixmap.fromImage(convert_to_Qt_format_xe_1) # Chuyển Qimage thành Qpixmap (ảnh được hiển thị trong giao diện)
    self.uic.label_xe_1.setPixmap(qt_img_xe_1) # Gán ảnh vào label_xe_1 trên giao diện
    #cv2.waitKey(5) # Đợi 5ms để tránh quá tải

    rgb_image_xe_2 = cv2.cvtColor(cv_img_xe_2, cv2.COLOR_BGR2RGB) # OpenCV dùng định dạng màu BGR, chuyển đổi sang RGB để phù hợp với giao diện
```

```
hiển thị
    convert_to_Qt_format_xe_2 = QtGui.QImage(rgb_image_xe_2.data,
rgb_image_xe_2.shape[1], rgb_image_xe_2.shape[0],
QtGui.QImage.Format_RGB888) # Tạo Qimage từ dữ liệu ảnh gốc của camera 2
    qt_img_xe_2 = QtGui.QPixmap.fromImage(convert_to_Qt_format_xe_2) # Chuyển
Qimage thành Qpixmap (ảnh được hiển thị trong giao diện)
    self.uic.label_xe_2.setPixmap(qt_img_xe_2) # Gán ảnh vào label_xe_2 trên giao
diện
    #cv2.waitKey(5) # Đợi 5ms để tránh quá tải

    rgb_image_xe_3 = cv2.cvtColor(cv_img_xe_3, cv2.COLOR_BGR2RGB) #
OpenCV dùng định dạng màu BGR, chuyển đổi sang RGB để phù hợp với giao diện
hiển thị
    convert_to_Qt_format_xe_3 = QtGui.QImage(rgb_image_xe_3.data,
rgb_image_xe_3.shape[1], rgb_image_xe_3.shape[0],
QtGui.QImage.Format_RGB888) # Tạo Qimage từ dữ liệu ảnh gốc của camera 3
    qt_img_xe_3 = QtGui.QPixmap.fromImage(convert_to_Qt_format_xe_3) # Chuyển
Qimage thành Qpixmap (ảnh được hiển thị trong giao diện)
    self.uic.label_xe_3.setPixmap(qt_img_xe_3) # Gán ảnh vào label_xe_3 trên giao
diện
    #cv2.waitKey(5) # Đợi 5ms để tránh quá tải

    rgb_image_xe_4 = cv2.cvtColor(cv_img_xe_4, cv2.COLOR_BGR2RGB) #
OpenCV dùng định dạng màu BGR, chuyển đổi sang RGB để phù hợp với giao diện
hiển thị
    convert_to_Qt_format_xe_4 = QtGui.QImage(rgb_image_xe_4.data,
rgb_image_xe_4.shape[1],
rgb_image_xe_4.shape[0],QtGui.QImage.Format_RGB888) # Tạo Qimage từ dữ liệu
ảnh gốc của camera 4
    qt_img_xe_4 = QtGui.QPixmap.fromImage(convert_to_Qt_format_xe_4) # Chuyển
Qimage thành Qpixmap (ảnh được hiển thị trong giao diện)
    self.uic.label_xe_4.setPixmap(qt_img_xe_4) # Gán ảnh vào label_xe_4 trên giao
diện
    #cv2.waitKey(5) # Đợi 5ms để tránh quá tải
```

```
    rgb_image_qr = cv2.cvtColor(cv_img_qr, cv2.COLOR_BGR2RGB) # OpenCV dùng định dạng màu BGR, chuyển đổi sang RGB để phù hợp với giao diện hiển thị
    convert_to_Qt_format_qr = QtGui.QImage(rgb_image_qr.data, rgb_image_qr.shape[1],rgb_image_qr.shape[0],QtGui.QImage.Format_RGB888) # Tạo Qimage từ dữ liệu ảnh gốc của camera quét mã QR
    qt_img_qr = QtGui.QPixmap.fromImage(convert_to_Qt_format_qr) # Chuyển Qimage thành Qpixmap (ảnh được hiển thị trong giao diện)
    self.uic.label_ma_qr.setPixmap(qt_img_qr) # Gán ảnh vào label_ma_qr trên giao diện
    #cv2.waitKey(2)

rgb_image_kho = cv2.cvtColor(cv_img_kho, cv2.COLOR_BGR2RGB) # OpenCV dùng định dạng màu BGR, chuyển đổi sang RGB để phù hợp với giao diện hiển thị
convert_to_Qt_format_kho = QtGui.QImage(rgb_image_kho.data, rgb_image_kho.shape[1], rgb_image_kho.shape[0],QtGui.QImage.Format_RGB888) # Tạo Qimage từ dữ liệu ảnh gốc của camera giám sát kho

qt_img_kho = QtGui.QPixmap.fromImage(convert_to_Qt_format_kho) # Chuyển Qimage thành Qpixmap (ảnh được hiển thị trong giao diện)
self.uic.label_kho.setPixmap(qt_img_kho) # Gán ảnh vào label_kho trên giao diện
#cv2.waitKey(5)

self.uic.hien_ma_qr.setText(ma_qr) # Gán chuỗi ma_qr lên giao diện
mycursor.execute("select * from thong_tin_kho.vi_tri") # Truy vấn dữ liệu bảng vi_tri trong database thong_tin_kho
        lay_tt_mysql = np.array(mycursor.fetchall())
        for row_number, row_data in enumerate(lay_tt_mysql):
            for column_number, data in enumerate(row_data):
                if column_number > 0: # Bỏ cột ID đầu tiên
                    self.uic.vi_tri_kho.setItem(row_number, column_number + 1, QTableWidgetItem(str(data))) # Dịch cột sang phải một ô, hiện dữ liệu lên bảng vi_tri_kho
        mycursor.execute("select * from thong_tin_kho.lich_su") # Truy vấn bảng lich_su
```

```
        lay_ls_mysql = np.array(mycursor.fetchall())
        for row_number, row_data in enumerate(lay_ls_mysql):
            for column_number, data in enumerate(row_data):
                if column_number > 0: # Bỏ cột ID đầu tiên
                    self.uic.lich_su.setItem(row_number, column_number - 1,
QTableWidgetItem(str(data))) # Dịch cột sang trái một ô

        self.uic.truc_x.setText(str(int(toa_do_x)) + ' cm') # Hiển thị toạ độ x của vật thể
        self.uic.truc_y.setText(str(int(toa_do_y)) + ' cm') # Hiển thị toạ độ của vật thể

class capture_video(QThread):
    signal =
pyqtSignal(np.ndarray,np.ndarray,np.ndarray,np.ndarray,np.ndarray,np.ndarray) #
Dùng để nhận dạng ảnh từ các camera bằng các luồng riêng

    def __init__(self, index): # Khởi tạo luồng với biến index
        self.index = index
        print("start threading", self.index) # Giúp kiểm tra xem luồng có hoạt động không
        super(capture_video, self).__init__()

def run(self):  # Bắt đầu xử lý video trong luồng
    global ma_qr # Khởi tạo biến ma_qr

    ma_qr = " "
    goc_rad = math.radians(35)  # tính tan góc nghiêng
    kq = 4.5 / math.tan(goc_rad)  # phần bù sau khi hiệu chỉnh từ pixel ra thực
    # Mở kết nối camera 1, độ phân giải ảnh 320x320 pixel
    cap_xe_1 = cv2.VideoCapture(0, cv2.CAP_DSHOW)
    cap_xe_1.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
    cap_xe_1.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)
    #cv2.waitKey(5)

   # Mở kết nối camera 2, độ phân giải ảnh 320x320 pixel
```

```
cap_xe_2 = cv2.VideoCapture(5, cv2.CAP_DSHOW)
cap_xe_2.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cap_xe_2.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)
#cv2.waitKey(5)


# Mở kết nối camera 3, độ phân giải ảnh 320x320 pixel
cap_xe_3 = cv2.VideoCapture(1, cv2.CAP_DSHOW)
cap_xe_3.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cap_xe_3.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)
#cv2.waitKey(5)


# Mở kết nối camera 4, độ phân giải ảnh 320x320 pixel
cap_xe_4 = cv2.VideoCapture(2, cv2.CAP_DSHOW)
cap_xe_4.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cap_xe_4.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)
#cv2.waitKey(5)


# Mở kết nối camera quét mã QR, độ phân giải ảnh 320x320 pixel
cap_qr = cv2.VideoCapture(4, cv2.CAP_DSHOW)
cap_qr.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cap_qr.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)


# Mở kết nối camera giám sát kho, độ phân giải ảnh 320x320 pixel
cap_kho = cv2.VideoCapture(3, cv2.CAP_DSHOW)
cap_kho.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cap_kho.set(cv2.CAP_PROP_FRAME_HEIGHT, 320)
#cv2.waitKey(5)

while True: # Vòng lặp để liên tục xử lý camera

# Các biến global cho phép sử dụng và cập nhật giá trị trong suốt chương trình
#global do_vi_tri, i
global i
```

```
    global toa_do_x, toa_do_y
    global x_TB, y_TB
    global bien_vao, bien_ra
    cv2.waitKey(5) # Tạm dừng 5ms giữa các lần đọc camera
    success_xe_1, img_xe_1 = cap_xe_1.read() # Lấy khung hình mới từ camera 1
    #print("ok")
    success_xe_2, img_xe_2 = cap_xe_2.read() # Lấy khung hình mới từ camera 2
    #print("ok")
    success_xe_3, img_xe_3 = cap_xe_3.read() # Lấy khung hình mới từ camera 3
    #print("ok")
    success_xe_4, img_xe_4 = cap_xe_4.read() # Lấy khung hình mới từ camera 4
    #print("ok")
    success_qr, img_qr = cap_qr.read() # Lấy khung hình mới từ camera quét mã QR
    #print("ok")
    success_kho, img_kho = cap_kho.read() # Lấy khung hình mới từ camera giám sát
kho
    #print("ok")
    cv2.waitKey(5) # Tạm dừng 5ms giữa các lần đọc camera

    result_xe_1 = yolo_xe(img_xe_1) # Dùng mô hình YOLO đã được huấn luyện để
phát hiện xe trong ảnh từ camera 1
cv2.waitKey(5)
for r_xe_1 in result_xe_1: # Lấy các vùng giới hạn (bounding box) từ kết quả của
YOLO
    boxes_xe_1 = r_xe_1.boxes
    for box_xe_1 in boxes_xe_1:
        if box_xe_1.conf[0] > 0.6: # Chỉ xử lý nếu độ tin cậy > 0.6, tránh nhận diện
nhầm vật thể

            # Lấy tâm của vùng giới hạn
            x1, y1, x2, y2 = box_xe_1.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
            z1 = int(x1 + ((x2 - x1) / 2))
            z2 = int(y1 + ((y2 - y1) / 2))
            print(z1, "  ", z2) # In ra toạ độ trung tâm của vật
```

```
            cv2.rectangle(img_xe_1, (x1, y1), (x2, y2), (255, 0, 255), 1)
            if box_xe_1.cls[0] == 0:
                points_1 = np.array([[[z1, z2]]], dtype=np.float32) # Biến điểm z1,z2 thành vị trí trong mặt phẳng thực tế
                transformed_points_1 = cv2.perspectiveTransform(points_1, matrix_1) # Dùng ma trận đồng nhất đã hiệu chỉnh
                x_new, y_new = transformed_points_1[0][0] # Chuyển toạ độ pixel sang cm
                x[0] = 60 - x_new / 10 + kq
                y[0] = 60 - y_new / 10 + kq


            # Vẽ khung, toạ độ lên ảnh

cv2.putText(img_xe_1, str(z1) + ' ' + str(z2), (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255), 1)
cv2.waitKey(5)

result_xe_2 = yolo_xe(img_xe_2) # Dùng mô hình YOLO đã được huấn luyện để phát hiện xe trong ảnh từ camera 2
cv2.waitKey(5)
for r_xe_2 in result_xe_2: # Lấy các vùng giới hạn (bounding box) từ kết quả của YOLO
    boxes_xe_2 = r_xe_2.boxes
    for box_xe_2 in boxes_xe_2:
if box_xe_2.conf[0] > 0.6: # Chỉ xử lý nếu độ tin cậy > 0.6, tránh nhận diện nhầm vật thể

            # Lấy tâm của vùng giới hạn
            x1, y1, x2, y2 = box_xe_2.xyxy[0]
            x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
            z1 = int(x1 + ((x2 - x1) / 2))
            z2 = int(y1 + ((y2 - y1) / 2))
            print(z1, ' ', z2) # In ra toạ độ trung tâm của vật
            cv2.rectangle(img_xe_2, (x1, y1), (x2, y2), (255, 0, 255), 1)
            if box_xe_2.cls[0] == 0:
```

```
            points_2 = np.array([[[z1, z2]]], dtype=np.float32) # Biến điểm z1,z2 thành vị
trí trong mặt phẳng thực tế
                transformed_points_2 = cv2.perspectiveTransform(points_2, matrix_2) #
Dùng ma trận đồng nhất đã hiệu chỉnh
                x_new, y_new = transformed_points_2[0][0] # Chuyển toạ độ pixel sang cm
                x[1] = 30 - x_new / 10 - kq
                y[1] = 60 - y_new / 10 + kq
               # Vẽ khung, toạ độ lên ảnh
                cv2.putText(img_xe_2, str(z1) + ' ' + str(z2), (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1,
                            (0, 255, 255), 1)
cv2.waitKey(5)

result_xe_3 = yolo_xe(img_xe_3) # Dùng mô hình YOLO đã được huấn luyện để phát
hiện xe trong ảnh từ camera 3
cv2.waitKey(5)
for r_xe_3 in result_xe_3: # Lấy các vùng giới hạn (bounding box) từ kết quả của
YOLO
   boxes_xe_3 = r_xe_3.boxes
   for box_xe_3 in boxes_xe_3:
 if box_xe_3.conf[0] > 0.6: # Chỉ xử lý nếu độ tin cậy > 0.6, tránh nhận diện nhầm vật
thể

          # Lấy tâm của vùng giới hạn
          x1, y1, x2, y2 = box_xe_3.xyxy[0]
          x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
          z1 = int(x1 + ((x2 - x1) / 2))
          z2 = int(y1 + ((y2 - y1) / 2))
          cv2.rectangle(img_xe_3, (x1, y1), (x2, y2), (255, 0, 255), 1)
          if box_xe_3.cls[0] == 0:
          points_3 = np.array([[[z1, z2]]], dtype=np.float32) # Biến điểm z1,z2 thành vị
trí trong mặt phẳng thực tế
                transformed_points_3 = cv2.perspectiveTransform(points_3, matrix_3) #
Dùng ma trận đồng nhất đã hiệu chỉnh
```

```
            x_new, y_new = transformed_points_3[0][0] # Chuyển toạ độ pixel sang cm
            x[2] = x_new / 10 + kq
            y[2] = y_new / 10 + kq
# Vẽ khung, toạ độ lên ảnh
cv2.putText(img_xe_3, str(z1) + ' ' + str(z2), (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1,
                    (0, 255, 255), 1)
cv2.waitKey(5)

result_xe_4 = yolo_xe(img_xe_4) # Dùng mô hình YOLO đã được huấn luyện để phát hiện xe trong ảnh từ camera 4
cv2.waitKey(5)
for r_xe_4 in result_xe_4: # Lấy các vùng giới hạn (bounding box) từ kết quả của YOLO
   boxes_xe_4 = r_xe_4.boxes
   for box_xe_4 in boxes_xe_4:
 if box_xe_4.conf[0] > 0.6: # Chỉ xử lý nếu độ tin cậy > 0.6, tránh nhận diện nhầm vật thể

        # Lấy tâm của vùng giới hạn
         x1, y1, x2, y2 = box_xe_4.xyxy[0]
         x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
         z1 = int(x1 + ((x2 - x1) / 2))
         z2 = int(y1 + ((y2 - y1) / 2))
         print(z1, ' ', z2) # In ra toạ độ trung tâm của vật
         cv2.rectangle(img_xe_4, (x1, y1), (x2, y2), (255, 0, 255), 1)
         if box_xe_4.cls[0] == 0:
        points_4 = np.array([[[z1, z2]]], dtype=np.float32) # Biến điểm z1,z2 thành vị trí trong mặt phẳng thực tế
            transformed_points_4 = cv2.perspectiveTransform(points_4, matrix_4) # Dùng ma trận đồng nhất đã hiệu chỉnh
            x_new, y_new = transformed_points_4[0][0] # Chuyển toạ độ pixel sang cm
            x[3] = 30 + x_new / 10 - kq
            y[3] = y_new / 10 + kq
```

```
            # Vẽ khung, toạ độ lên ảnh
            cv2.putText(img_xe_4, str(z1) + ' ' + str(z2), (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1,
                        (0, 255, 255), 1)
cv2.waitKey(5)


# Tính trung bình toạ độ từ các camera, nếu camera nào không nhận diện được xe
(x[n]=0) thì bỏ qua, sau đó lưu toạ độ vào toa_do_x = toa_do_y
toa_do_x = toa_do_y = 0
x_TB = y_TB = 0

for n in range(4):
    if x[n] != 0:
        toa_do_x += x[n]
        x_TB +=1

    if y[n] != 0:
        toa_do_y += y[n]
        y_TB += 1

if x_TB > 1:
    toa_do_x /= x_TB

if y_TB > 1:
    toa_do_y /= y_TB

for barcode in decode(img_qr): # Trả về danh sách các QR code hoặc barcode được
phát hiện trong ảnh
    ma_qr = barcode.data.decode('utf-8') # Lấy dữ liệu từ mã QR và chuyển về dạng
byte
    pts = np.array([barcode.polygon], np.int32) # Trả về dữ liệu 4 điểm tạo thành hình
dạng của mã QR
    pts = pts.reshape((-1, 1, 2)) # Định dạng lại cho đúng yêu cầu của cv2.polylines
    cv2.polylines(img_qr, [pts], True, (0, 0, 255), 5)  # Vẽ khung đa giác quanh mã QR
```

```
    pts2 = barcode.rect # Trả về toạ độ của hình chữ nhật bao quanh mã QR
    cv2.putText(img_qr, barcode.data.decode('utf-8'), (pts2[0], pts2[1]),
cv2.FONT_HERSHEY_SIMPLEX, 0.9,(255, 0, 145), 2) # Ghi nội dung của mã QR

result_ong = yolo_ong(img_kho, stream=False) # Chạy mô hình YOLO để nhận diện
hàng
        cv2.waitKey(5)
    # Mỗi 4 khung hình thì gán dữ liệu từ vi_tri_ao sang vi_tri_SS để giữ lại vị trí hàng
vừa nhận được, reset lại vi_tri_ao để đếm lại từ đầu

  if i >= 4:
          i = 0
          for n in range(8):
             vi_tri_SS[n] = vi_tri_ao[n]
             vi_tri_ao[n] = 0


        for r_ong in result_ong:
           boxes_ong = r_ong.boxes
           for box_ong in boxes_ong:
           if box_ong.conf[0] > 0.6: # Chỉ xét các vùng giới hạn có độ tin cậy cao hơn
60%

                    # Tính toạ độ trung tâm vùng giới hạn của hàng
                    x1, y1, x2, y2 = box_ong.xyxy[0]
                    x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)
                    z1 = int(x1 + ((x2 - x1) / 2))
                    z2 = int(y1 + ((y2 - y1) / 2))
                    #print(z1, "  ", z2)
                    cv2.rectangle(img_kho, (x1, y1), (x2, y2), (255, 0, 255), 1)
                    if box_ong.cls[0] == 0: # Xác định vị trí trong kho

                      # So sánh vị trí trung tâm (z1,z2) của vật thể với toạ độ mặc định để xác
định vật đang nằm vị trí nào trong kho, khi phái hiện đúng vị trí nào thì gái số tương
ứng để ghi nhận vị trí đó đang có hàng
                        if z2 > 180 and z2 < 225 :
                           if z1 > 130 and z1 < 150 :
```

```
                    cv2.putText(img_kho, '1', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX,1, (0, 255, 255), 1)
                    vi_tri_ao[0] = 1
                elif z1 > 160 and z1 < 180:
                    cv2.putText(img_kho, '2', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255),1)
                    vi_tri_ao[1] = 1
                elif z1 > 200 and z1 < 225:
                    cv2.putText(img_kho, '3', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255), 1)
                    vi_tri_ao[2] = 1
                elif z1 > 230 and z1 < 260:
                    cv2.putText(img_kho, '4', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255), 1)
                    vi_tri_ao[3] = 1

            elif z2 > 80 and z2 < 120:
                if z1 > 130 and z1 < 150 :
                    vi_tri_ao[4] = 1
                    cv2.putText(img_kho, '5', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX,1, (0, 255, 255), 1)
                elif z1 > 155 and z1 < 190:
                    vi_tri_ao[5] = 1
                    cv2.putText(img_kho, '6', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255),1)
                elif z1 > 200 and z1 < 225:
                    vi_tri_ao[6] = 1
                    cv2.putText(img_kho, '7', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255), 1)
                elif z1 > 230 and z1 < 260:
                    vi_tri_ao[7] = 1
                    cv2.putText(img_kho, '8', (z1, z2),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 255, 255), 1)
```

```
# Nếu xe tiến vào vùng x > 32, tức lối ra vào, bật bien_vao, bien_ra để xử lý xuất nhập kho
        if toa_do_x > 32 and bien_vao == 0:
            bien_vao = 1
            bien_ra = 1
# Nếu có mã QR và xe đã vào toạ độ x < 20, gọi hàm vao() để ghi nhận nhập hàng và reset biến vào
        if ma_qr != " " and bien_vao == 1 and toa_do_x < 20:
            vao()
            bien_vao = 0
# Nếu xe ở vùng x < 20 và bien_ra được bật thì ghi nhận xuất kho
        if bien_ra == 1 and toa_do_x < 20:
            ra()
            bien_ra = 0

        print(bien_vao)

        i = i + 1
        cv2.waitKey(5)

self.signal.emit(img_xe_1, img_xe_2, img_xe_3, img_xe_4, img_qr, img_kho) # Gửi tín hiệu cập nhật hình ảnh
        cv2.waitKey(5)

def stop(self): # Dừng luồng xử lý
    print("stop threading", self.index)
    self.terminate()

if __name__ == "__main__": # Đảm bảo đoạn mã bên trong chỉ chạy khi file được thực hiện trực tiếp
    app = QApplication(sys.argv) # Tạo Qapplication dùng để chạy giao diện
    main_win = MainWindow() # Tạo cửa sổ chính của ứng dụng
    main_win.show() # Hiển thị cửa số chính của ứng dụng lên màn hình
    sys.exit(app.exec()) # Dùng để kết thúc hoàn toàn chương trình khi đóng giao diện
```