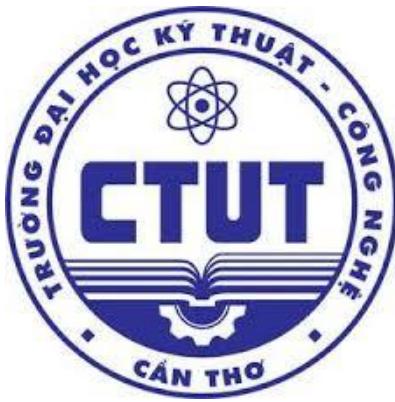


ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ
KHOA KỸ THUẬT CƠ KHÍ

----------



NGUYỄN MINH PHƯỚC

1700534

**MÔ HÌNH MOBILE ROBOT BÁM SÁT ĐỐI TƯỢNG, HỖ
TRỢ THỦ TỤC GIẤY TỜ HÀNH CHÍNH CHO SINH VIÊN
TẠI KHOA KỸ THUẬT CƠ KHÍ**

Ngành: Công nghệ kỹ thuật Cơ điện tử

Mã số: 7510204

LUẬN VĂN ĐẠI HỌC

TP.CẦN THƠ, tháng 06 năm 2021

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ CẦN THƠ**

Cán bộ hướng dẫn: ThS. NGUYỄN LÊ THẾ DUY

Luận văn đại học được bảo vệ tại Khoa Kỹ Thuật Cơ Khí Trường Đại Học Kỹ Thuật Công Nghệ Cần Thơ ngày 28/06/2021.

Thành phần Ban chấm đánh giá luận văn đại học gồm:

1. Trưởng ban: ThS. ĐƯỜNG KHÁNH SƠN
2. Ủy viên phản biện: ThS. PHÓ HOÀNG LINH
3. Thư ký ban chấm: ThS. NGUYỄN LÊ THẾ DUY

Xác nhận của Ban chấm đánh giá sau khi luận văn đã được sửa chữa.

TRƯỞNG BAN

ỦY VIÊN PHẢN BIỆN

THƯ KÝ

ThS. Đường Khánh Sơn

ThS. Phó Hoàng Linh

ThS. Nguyễn Lê Thế Duy

TRƯỜNG ĐẠI HỌC CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
KỸ THUẬT – CÔNG NGHỆ CÀN THƠ Độc lập – Tự do – Hạnh phúc
KHOA KỸ THUẬT CƠ KHÍ
NHIỆM VỤ LUẬN VĂN ĐẠI HỌC

Họ tên sinh viên: NGUYỄN MINH PHƯỚC MSSV: 1700534
Ngày, tháng, năm sinh: 06/05/1999 Nơi sinh: Sóc Trăng
Ngành: Công nghệ kỹ thuật Cơ điện tử Mã số: 7510204

TÊN ĐỀ TÀI:

MÔ HÌNH MOBILE ROBOT BÁM SÁT ĐỐI TƯỢNG, HỖ TRỢ THỦ TỤC GIẤY
TỜ HÀNH CHÍNH CHO SINH VIÊN TẠI KHOA KỸ THUẬT CƠ KHÍ

NHIỆM VỤ VÀ NỘI DUNG

❖ Nhiệm vụ:

Ứng dụng công nghệ xử lý ảnh vào mobile robot phát hiện đối tượng theo khuôn mặt, màu đỏ và điều khiển robot di chuyển bám sát đối tượng.

❖ Nội dung:

- Thiết kế, chế tạo mô hình mobile robot.
- Thiết kế mạch điều khiển mobile robot.
- Xây dựng chương trình xử lý ảnh bằng ngôn ngữ Python và điều khiển mobile robot sử dụng giải thuật PID cho động cơ DC.

CÁN BỘ HƯỚNG DẪN: ThS. NGUYỄN LÊ THẾ DUY

THỜI GIAN GIAO ĐỀ TÀI: 15/02/2021

THỜI GIAN HOÀN THÀNH: 18/06/2021

Cần Thơ, ngày 10 tháng 07 năm 2021

CÁN BỘ HƯỚNG DẪN

BỘ MÔN CƠ ĐIỆN TỬ

KHOA KỸ THUẬT CƠ KHÍ

LỜI CAM ĐOAN

Tôi xin cam đoan đây là đề tài nghiên cứu do tôi thực hiện với sự hướng dẫn của ThS. Nguyễn Lê Thế Duy và không sao chép từ tài liệu hay công trình đã có trước đó. Tất cả những nội dung có sự tham khảo từ các tài liệu khác đều được ghi lại trong phần tài liệu tham khảo.

Cần Thơ, ngày 10 tháng 07 năm 2021

Sinh viên thực hiện

Nguyễn Minh Phước

LỜI CẢM ƠN

Lời đầu tiên, tôi xin gửi đến Thầy Nguyễn Lê Thế Duy – Giảng viên hướng dẫn đề tài lời cảm ơn chân thành và sâu sắc nhất, thầy đã quan tâm và tạo mọi điều kiện thuận lợi nhất về cơ sở vật chất cũng như các tài liệu tham khảo đồng thời cũng là người định hướng, và góp ý các ưu khuyết điểm trong suốt quá trình thực hiện luận văn, để luận văn vào bước hoàn thiện hơn.

Đồng thời, tôi xin chân thành cảm ơn quý thầy, cô giảng viên Khoa Kỹ Thuật Cơ Khí nói chung và thầy cô Trường Đại Học Kỹ thuật – Công nghệ Cần Thơ nói riêng, đã dạy dỗ tôi trong suốt quãng thời gian ngồi trên ghế giảng đường Đại học, truyền đạt rất nhiều kiến thức hữu ích và tạo mọi điều kiện thuận lợi nhất để tôi có thể hoàn thành tốt đề tài luận văn này.

Đặc biệt tôi xin gửi lời biết ơn sâu sắc đến người thân, gia đình, cha mẹ đã tạo mọi điều kiện tốt về kinh tế cũng như tình thần cho tôi có một quá trình nghiên cứu hiệu quả.

Tôi cũng xin cảm ơn đến bạn bè, anh chị khóa trước đã hỗ trợ cũng như động viên và chia sẻ kinh nghiệm cho tôi trong suốt quá trình thực hiện luận văn.

Xin trân trọng cảm ơn.

Cần Thơ, Ngày 10 tháng 07 năm 2021

Nguyễn Minh Phước

TÓM TẮT LUẬN VĂN

Robot nói chung và mobile robot nói riêng là một ngành khoa học phát triển từ giữa thế kỷ XX, và có tốc độ phát triển ngày càng nhanh chóng nhờ vào các thành tựu của khoa học kỹ thuật. Với sự phát triển đa dạng và ngày càng hoàn thiện, mobile robot được ứng dụng trong rất nhiều lĩnh vực như trong công nghiệp, y tế, quân sự, thám hiểm, phục vụ sinh hoạt trong gia đình.

Việc tích hợp Computer Vision (CV) vào mobile robot đem lại cho robot khả năng nhận biết môi trường làm việc, nhờ đó tăng cường khả năng linh hoạt trong khi hoạt động.

Đề tài luận văn “ Mô Hình Mobile Robot Bám Sát Đồi Tượng, Hỗ Trợ Thủ Tục Giấy Tờ Hành Chính Cho Sinh Viên Tại Khoa Kỹ Thuật Cơ Khí” thực hiện giải quyết những nhiệm vụ bao gồm các nội dung như sau: Tìm hiểu về mobile robot sử dụng CV, thiết kế và điều khiển mobile robot sử dụng CV phát hiện đồi tượng (khuôn mặt và màu đỏ) xác định vị trí đồi tượng trong khung hình, điều khiển robot bám theo đồi tượng, xây dựng giải thuật PID điều khiển động cơ DC.

BẢNG KÍ HIỆU, CHỮ VIẾT TẮT

Chữ viết tắt	Chữ đầy đủ
API	Application Programming Interface
BGR	Blue Green Red
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CV	Computer Vision
GPU	Graphics Processing Unit
HSV	Hue Saturation Value
HTML	Hypertext Markup Language
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
OOP	Object Oriented Programming
PEP	Python Enhancement Proposal
PHP	Hypertext Preprocessor
SSD	Single Shot Object Detectors
TPU	Tensor Processing Unit
UART	Universal Asynchronous Receiver / Transmitter

MỤC LỤC

LỜI CAM ĐOAN	I
LỜI CẢM ƠN	II
TÓM TẮT LUẬN VĂN	III
BẢNG KÍ HIỆU, CHỮ VIẾT TẮT	IV
MỤC LỤC.....	i
DANH MỤC HÌNH ẢNH	vi
DANH MỤC BẢNG.....	ix
MỞ ĐẦU.....	1
CHƯƠNG 1 : TỔNG QUAN.....	3
1.1 Giới thiệu về mobile robot	3
1.1.1 Sơ lược về robot.....	3
1.1.2 Mobile robot	3
1.2 Giới thiệu Computer Vision và mobile robot sử dụng Computer Vision	5
1.2.1 Computer Vision.....	5
1.2.2 Mobile robot sử dụng Computer Vision.....	8
1.3 Vấn đề tập trung nghiên cứu của đề tài	9
CHƯƠNG 2 : ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU	10
2.1 Đối tượng nghiên cứu.....	10
2.1.1 Phần mềm Inventor	10
2.1.2 Tổng quan Arduino	11
2.1.2.1 Giới thiệu Arduino	11
2.1.2.2 Phần cứng của Arduino.....	12
2.1.2.3 Một số loại Arduino điển hình	13
2.1.2.4 Ứng dụng của Arduino.....	14
2.1.3 Phần mềm IDE Arduino	15

2.1.3.1 Khái niệm IDE Arduino.....	15
2.1.3.2 Cấu trúc chương trình	16
2.1.4 Ngôn ngữ Python	17
2.1.4.1 Khái niệm Python	17
2.1.4.2 Lịch sử hình thành	19
2.1.4.3 Ứng dụng Python	20
2.1.5 Phần mềm PyCharm	22
2.1.5.1 Khái niệm PyCharm.....	22
2.1.5.2 Ứng dụng	23
2.1.6 Thị giác máy tính - Computer Vision (CV).....	23
2.2 Phương pháp nghiên cứu.....	25
2.2.1 Phương thức truyền dữ liệu I2C	25
2.2.1.1 Giới thiệu về giao tiếp I2C.....	25
2.2.1.2 Đặc điểm	26
2.2.2 Phương thức truyền dữ liệu UART	26
2.2.2.1 Giới thiệu về giao tiếp UART.....	26
2.2.2.2 Đặc điểm	28
2.2.3 Bộ điều khiển PID	28
2.2.3.1 Lý thiết về bộ điều khiển PID	28
2.2.3.2 Các phương pháp chỉnh định tham số bộ điều khiển PID	30
CHƯƠNG 3 : THIẾT KẾ VÀ XÂY DỰNG ROBOT.....	32
3.1 Phân tích động học mobile robot	32
3.1.1 Cơ sở thiết kế	34
3.2 Thiết kế mô hình	34
3.2.1 Bản thiết kế mô hình trên Autodesk Inventor.....	34

3.2.2 Thiết kế cơ khí	38
3.2.2.1 Thiết kế khung xe.....	38
3.2.2.2 Động cơ di chuyển	39
3.2.2.3 Bánh xe di chuyển được sử dụng.....	40
3.3 Thiết kế mạch điện điều khiển	40
3.3.1 Sơ đồ mạch điều khiển robot	40
3.3.1.1 Arduino Mega2560 (Master)	43
3.3.1.2 Arduino Uno R3 (Slave)	46
3.3.1.3 Module L298N	48
3.3.1.4 Module LM2596	50
3.3.1.5 Pin Lipo Lithium Polymer	50
3.4 Mô hình hoàn chỉnh.....	51
3.5 Mô phỏng và chạy thực nghiệm	52
3.5.1 Mô phỏng Simulink	52
3.5.2 Thực nghiệm bộ điều khiển PID.....	53
CHƯƠNG 4 : ÚNG DỤNG CÔNG NGHỆ XỬ LÝ ẢNH TRONG LUẬN VĂN	54
4.1 Xử lý ảnh	54
4.1.1 Mục đích và nội dung chương trình.....	54
4.1.2 Giới thiệu về hệ thống xử lý ảnh	54
4.1.3 Sơ lược xử lý ảnh trong đề tài	55
4.1.4 Thiết bị thu nhận ảnh đầu vào	56
4.2 Xử lý ảnh trong phát hiện khuôn mặt.....	57
4.2.1 TensorFlow object detection API	57
4.2.1.1 Tensorflow	57
4.2.1.2 API	58

MỤC LỤC

4.2.2 Mô hình SSD MobileNet V2	58
4.2.2.1 Mạng MobileNet.....	58
4.2.2.2 Thuật toán SSD (Single Shot Object Detectors).....	60
4.2.3 Giá trị đè tài sử dụng	62
4.2.4 Lưu đồ giải thuật.....	63
4.3 Xử lý ảnh trong phát hiện màu.....	65
4.3.1 Không gian màu.....	65
4.3.1.1 Không gian RGB.....	65
4.3.1.2 Không gian màu HSV	66
4.3.1.3 Không gian màu YCbCr	69
4.3.2 Giá trị màu đè tài sử dụng.....	70
4.3.3 Lưu đồ giải thuật.....	71
4.4 Thư viện hỗ trợ chính	74
4.4.1 Thư viện OpenCV.....	74
4.4.2 Thư viện Tensorflow	75
4.4.3 Thư viện Numpy.....	76
CHƯƠNG 5 : PHẦN KẾT HỢP ĐÈ TÀI.....	78
5.1 Thiết kế mô hình	78
5.1.1 Cơ sở thiết kế	78
5.1.2 Bản vẽ tổng thể	78
5.2 Sơ đồ điều khiển.....	79
5.3 Mô hình thực tế	81
5.4 Đánh giá	82
CHƯƠNG 6 : KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN	84
6.1 Kết quả	84

MỤC LỤC

6.1.1 Kết quả phát hiện khuôn mặt	84
6.1.2 Kết quả phát hiện màu	87
6.1.3 Điều khiển mobile robot	89
6.2 Đánh giá	89
6.3 Kết luận	90
6.3.1 Kết quả đạt được	90
6.3.2 Các hạn chế của đề tài	90
6.4 Hướng phát triển	90
TÀI LIỆU THAM KHẢO	92
PHỤ LỤC	93

DANH MỤC HÌNH ẢNH

Hình 1.1: Một số mobile robot trong thực tế	5
Hình 1.2: Robot hệ có camera trực tiếp tay máy(a) và robot có camera cố định(b) ...	7
Hình 1.3: Một số mobile robot sử dụng Computer Vision	9
Hình 2.1: Giao diện phần mềm AUTODESK INVENTOR	11
Hình 2.2: Arduino Mega 2560 và Arduino Uno R3	12
Hình 2.3: Phần cứng của Arduino Uno R3	12
Hình 2.4: Các Arduino phổ biến	13
Hình 2.5: Một số Arduino cao cấp	14
Hình 2.6: Logo Arduino IDE	15
Hình 2.7: Giao diện Arduino IDE	16
Hình 2.8: Logo Python	17
Hình 2.9: Một số ứng dụng Python	20
Hình 2.10: Logo PyCharm	22
Hình 2.11: Sơ đồ giao tiếp I2C	26
Hình 2.12: Cách truyền dữ liệu UART	27
Hình 2.13: Sơ đồ biểu diễn truyền data.....	28
Hình 2.14: Sơ đồ khối bộ điều khiển PID	29
Hình 2.15: Mô hình hệ thống điều khiển với PID.....	30
Hình 3.1: Mô hình hóa Mobile robot	32
Hình 3.2: Sơ đồ khối thiết kế mô hình	34
Hình 3.3: Bản vẽ tổng thể mobile robot.....	35
Hình 3.4:Bản vẽ liên kết khâu chuyên động	36
Hình 3.5: Bệ đỡ động cơ với khung robot	37
Hình 3.6: Nhôm định hình 20x20	38
Hình 3.7: Mô hình xây dựng thực tế	38
Hình 3.8: Động cơ DC M555.....	39
Hình 3.9: Bánh chủ động(a) và bánh caster(b)	40
Hình 3.10: Sơ đồ khối mạch điều khiển.....	41

DANH MỤC HÌNH ẢNH

Hình 3.11: Sơ đồ Layout mạch điều khiển	41
Hình 3.12: Sơ đồ đấu nối dây.....	42
Hình 3.13: Mạch điều khiển sau khi gia công.....	42
Hình 3.14: Sơ đồ chân Arduino Mega 2560	45
Hình 3.15: Sơ đồ chân Arduino Uno R3	48
Hình 3.16: Sơ đồ nguyên lý mạch cầu H	49
Hình 3.17: Sơ đồ chân Module L298N	49
Hình 3.18: Module LM2596	50
Hình 3.19: Pin Lipo Lithium Polymer	51
Hình 3.20 Mô hình mobile robot hoàn chỉnh.....	51
Hình 3.21: Mô hình mô phỏng Simulink robot di chuyển thẳng	52
Hình 3.22: Mô hình mô phỏng simulink robot chuyển động xoay tại chỗ	52
Hình 3.23: Tốc độ quay của động cơ không tải	53
Hình 4.1: Các bước trong xử lý ảnh.....	55
Hình 4.2: Sơ đồ quá trình xử lý ảnh trong đè tài.....	56
Hình 4.3: Webcam Logitech Pro 9000.....	57
Hình 4.4: Biểu diễn chức năng Tensorflow	58
Hình 4.5: Mô hình kiến trúc mạng mobileNet	59
Hình 4.6: Một số gói mô hình điển hình	60
Hình 4.7: Giai đoạn trích xuất feature map.....	61
Hình 4.8: Hàm trích xuất tọa độ của chương trình.....	62
Hình 4.9: Tọa độ phát hiện khuôn mặt.....	62
Hình 4.10: Lưu đồ giải thuật phát hiện khuôn mặt	63
Hình 4.11: Lưu đồ giải thuật điều khiển robot bám sát khuôn mặt	64
Hình 4.12: Logo RGB	65
Hình 4.13: Không gian màu HSV	66
Hình 4.14: Bảng màu từ RGB sang HSV	68
Hình 4.15: Bản tinh chỉnh giá trị HSV.....	69
Hình 4.16: Chuyển đổi RGB sang YcbCr và ngược lại	70
Hình 4.17: Tọa độ phát hiện màu.....	71

DANH MỤC HÌNH ẢNH

Hình 4.18: Lưu đồ giải thuật phát hiện hiện màu đỏ	72
Hình 4.19: Lưu đồ giải thuật điều khiển robot bám sát màu đỏ về bãi đỗ.....	73
Hình 4.20: Logo OpenCV	74
Hình 4.21: Logo Tensorflow	75
Hình 5.1: Bản vẽ tổng thể mô hình chung	78
Hình 5.2: Sơ đồ điều khiển toàn hệ thống.....	79
Hình 5.3: Sơ đồ đấu nối dây.....	81
Hình 5.4: Mô hình kết hợp thực tế	82
Hình 5.5: Dữ liệu được lưu trong thư mục.....	83
Hình 6.1: Kết quả phát hiện khuôn mặt chưa trích lọc	84
Hình 6.2: Kết quả phát hiện khuôn mặt phần trăm cao nhất.....	84
Hình 6.3: Kết quả phát hiện khuôn mặt ngược sáng.....	85
Hình 6.4: Kết quả phát hiện khuôn mặt khoảng cách 5m	85
Hình 6.5: Kết quả phát hiện khuôn mặt góc nghiêng.....	86
Hình 6.6: Kết quả tinh chỉnh giá trị vùng màu đỏ.....	87
Hình 6.7: Kết quả phát hiện màu đỏ buổi sáng	87
Hình 6.8: Kết quả phát hiện màu đỏ buổi tối	87
Hình 6.9: Kết quả phát hiện màu đỏ ngược sáng	88
Hình 6.10: Kết quả phát hiện màu đỏ ở khoảng cách 5m	88
Hình 6.11: Mô hình Mobile robot hoàn chỉnh	89

DANH MỤC BẢNG

Bảng 2.1: Thông số chỉnh định PID bằng phương pháp Ziegler-Nichols	30
Bảng 6.1: Kết quả thực nghiệm phát hiện khuôn mặt.....	86

MỞ ĐẦU

Lý do chọn đề tài:

Trong thời đại hội nhập, thời đại của công nghiệp hóa, hiện đại hóa đất nước, ngành cơ điện tử ngày càng có nhiều đóng góp rất to lớn vào cuộc cách mạng khoa học kỹ thuật hiện đại, đặc biệt là sự ra đời và phát triển nhanh chóng của kỹ thuật số cũng như hệ thống nhúng làm cho kỹ thuật điện tử cũng như các vi mạch điều khiển phong phú đa dạng hơn. Hiện tại phần lớn trong công việc đều đưa kỹ thuật hiện đại vào rộng rãi trong mọi lĩnh vực của hoạt động sản xuất, kinh tế đời sống xã hội,... Chúng được thể hiện qua các nhà máy, khu công nghiệp,... vận hành các thiết bị điện, điều khiển cánh tay robot, lập trình robot làm những công việc thường ngày ngoài ra còn có cả trong đời sống xã hội và gia đình một cách chi tiết, linh hoạt và đồng bộ.

Ngày nay, ngoài ứng dụng sơ khai ban đầu của robot trong chế tạo máy thì ứng dụng khác trong y tế, chăm sóc sức khỏe, nông nghiệp đóng tàu, xây dựng, an ninh quốc phòng đang là động lực cho sự phát triển robot. Đặc biệt nhất là robot y tế đang được quan tâm vì cần phục vụ cho nhu cầu y tế hiện tại.

Đề tài được phát triển từ Đề án kỹ thuật điều khiển và Đề án thiết kế hệ thống cơ điện tử, đề tài ở hai đề án chỉ dừng lại ở việc điều khiển từ xa bằng tay được điều khiển thông qua sóng Wifi, chính vì thế tôi đã quyết định phát triển và chọn đề tài:”**MÔ HÌNH MOBILE ROBOT BÁM SÁT ĐỐI TƯỢNG, HỖ TRỢ THỦ TỤC GIẤY TỜ HÀNH CHÍNH CHO SINH VIÊN TẠI KHOA KỸ THUẬT CƠ KHÍ**”.

Đề tài được kết hợp với đề tài của bạn Lê Trung Kiên để tạo ra một sản phẩm hoàn chỉnh.

Mục tiêu và phạm vi nghiên cứu:

- + Mục tiêu nghiên cứu:

Thiết kế chế tạo mobile robot bám đối tượng, bằng xử lý ảnh qua phát hiện khuôn mặt và màu đỏ, giao tiếp UART với Arduino, giao tiếp I2C và sử dụng giải thuật PID điều khiển động cơ DC.

Theo dõi các yếu tố ảnh hưởng đến quá trình phát hiện khuôn mặt và màu đỏ trong thực nghiệm và tốc độ di chuyển của robot khi sử dụng thuật điều khiển PID.

- + Phạm vi nghiên cứu:

Đề tài áp dụng tổng hợp các kiến thức chuyên ngành Cơ Điện Tử và đại diện ở các môn: Kỹ thuật lập trình, vi điều khiển, kỹ thuật số,... ngoài ra còn các kiến thức tự tìm hiểu và bổ sung gồm thiết kế 3D, ngôn ngữ C, Python,...

Mục tiêu đề tài là tìm hiểu nghiên cứu chính là phát hiện và bám sát đối tượng (khuôn mặt, màu đỏ) giao tiếp với vi điều khiển và điều khiển động cơ DC dùng giải thuật PID.

Kích thước mô hình mobile robot (DxRxH) 420x330x102 mm với bán kính hoạt động 210 mm, mobile robot phát hiện và bám sát đối tượng sử dụng Computer Vision qua ngôn ngữ Python và giao tiếp xuống vi điều khiển.

- + Giới hạn đề tài

Đề tài được giới hạn bởi yếu tố ánh sáng, phải đảm bảo ánh sáng thích hợp để camera phát hiện được đối tượng. Hệ thống phát hiện đối tượng chỉ phát hiện một đối tượng trong khung hình. Robot di chuyển giới hạn trên mặt phẳng và không vật cản.

Ý nghĩa khoa học và thực tiễn của đề tài:

Chủ đề xử lý ảnh đang được phát triển mạnh mẽ trong xã hội hiện tại, và được đầu tư vào mobile robot trong nghiên cứu khoa học, khám phá, đời sống, học tập,... nhất là trong lĩnh vực y tế đang được chú trọng.

Mobile robot ứng dụng Computer Vision giúp con người kiểm soát tốt trong nhiều lĩnh vực vận chuyển hàng hóa, quan sát các môi trường khó quan sát, độc hại,...

Bố cục đề tài:

Chương 1: TỔNG QUAN ĐỀ TÀI

Chương 2: ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

Chương 3: THIẾT KẾ VÀ XÂY DỰNG ROBOT

Chương 4: ỨNG DỤNG CÔNG NGHỆ XỬ LÝ ẢNH TRONG LUẬN VĂN

Chương 5: PHẦN KẾT HỢP ĐỀ TÀI

Chương 6: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

CHƯƠNG 1 : TỔNG QUAN

1.1 Giới thiệu về mobile robot

Thuật ngữ robot được sinh ra từ trên sân khấu, không phải trong phân xưởng sản xuất. Những robot xuất hiện lần đầu tiên trên ở trên New York vào ngày 09/10/1922 trong vở “Rossum’s Universal Robot” của nhà soạn kịch người Tiệp Karen Kapek viết năm 1921, còn từ robot là cách gọi tắt của từ robota - theo tiếng Tiệp có nghĩa là công việc lao dịch.

Robot hay người máy là một loại máy có thể thực hiện những công việc một cách tự động bằng sự điều khiển của máy tính. Robot là một tác nhân cơ khí, nhân tạo, thường là một hệ thống cơ khí-điện tử.

Với sự xuất hiện và chuyển động của mình, robot gây cho người ta cảm giác rằng nó giắc quan giống như con người. Từ "robot" (người máy) thường được hiểu với hai nghĩa: rõ bót cơ khí và phần mềm tự hoạt động. Về lĩnh vực người máy, Nhật Bản là nước đi đầu thế giới về lĩnh vực này.

Robot là một ngành khoa học phát triển từ giữa thế kỷ XX, là tâm điểm của một cuộc cách mạng lớn sau Internet và có tốc độ phát triển ngày càng nhanh chóng nhờ vào các thành tựu của khoa học kỹ thuật. Với sự phát triển đa dạng ngày càng hoàn thiện nên robot được ứng dụng trong rất nhiều các lĩnh vực như công nghiệp, y tế, quân sự, thám hiểm không gian, học tập, phục vụ sinh hoạt trong gia đình... Robot là sản phẩm công nghệ có độ phức tạp cao, chứa hàm lượng tri thức vô cùng phong phú về tất cả các lĩnh vực của khoa học và công nghệ.

1.1.2 Mobile robot

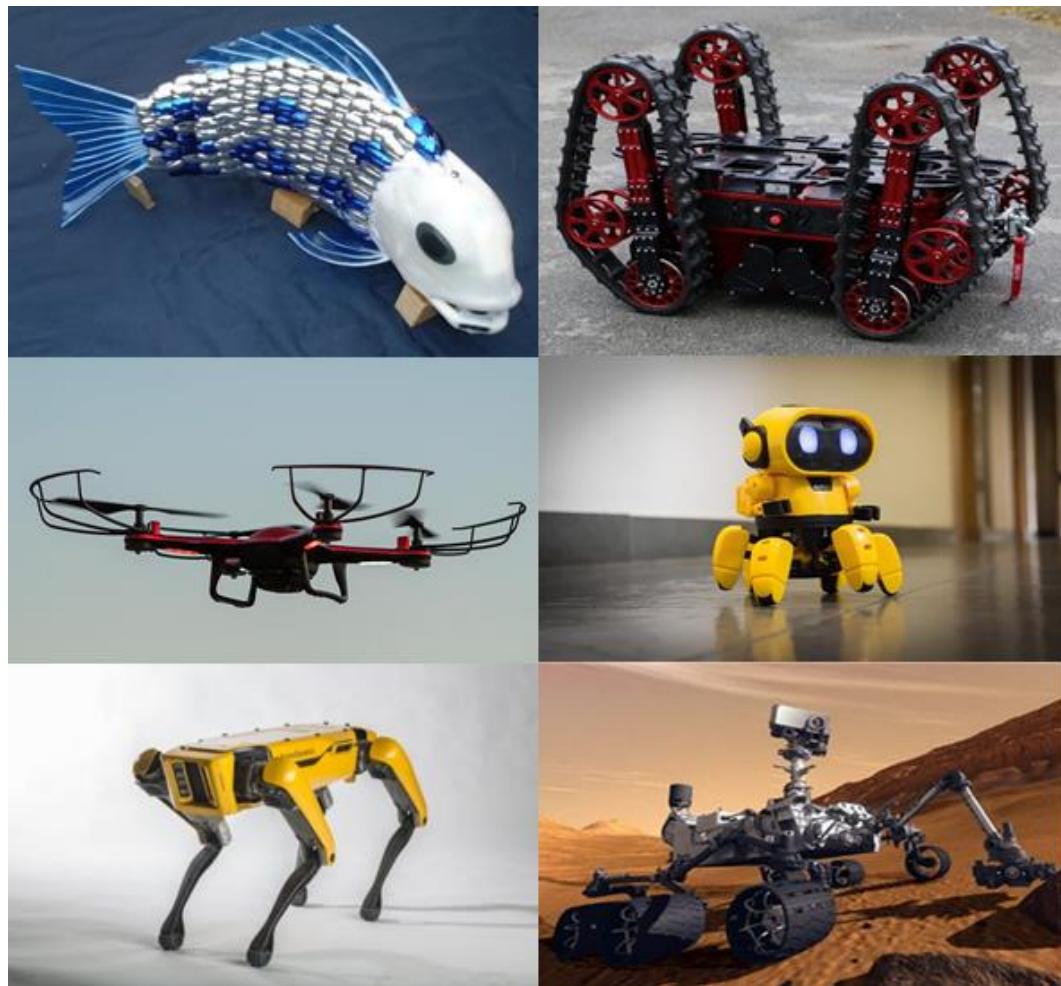
Mobile robot là một loại robot có thể tự di chuyển và vận động trong môi trường để thực hiện các công việc đã được lập trình. Nó có vai trò quan trọng trong ngành khoa học robot, những năm gần đây với các thành tựu trong lĩnh vực cơ khí và điện tử, mobile robot phát triển ngày càng mạnh mẽ, đa dạng về chủng loại, hoàn thiện về chức năng nên chúng ngày càng đóng một vai trò quan trọng trong cuộc sống, được sử dụng trong nhiều lĩnh vực đặc biệt là trong quân sự, cứu hộ, thám hiểm, và phục

vụ sinh hoạt hằng ngày. Hiện nay, mobile robot đang được ứng dụng rất nhiều trong thực tế, từ robot vận chuyển hàng hóa trong các tòa nhà, nhà máy, cửa hàng, sân bay, thư viện, robot thực hiện việc quét dọn đường phố, robot làm việc trong môi trường nguy hiểm như núi lửa, vùng bị phóng xạ, nhà máy hạt nhân, robot an ninh, canh gác, robot do thám dùng trong quân sự, robot thực hiện việc khám phá vũ trụ, di chuyển và lấy mẫu phân tích trên các hành tinh, robot phục vụ, hỗ trợ người già, người tàn tật và robot phục vụ sinh hoạt trong gia đình...

Những ứng dụng thực tế đòi hỏi những robot di động có tính tự động cao và những kỹ thuật hiện đại, bao gồm sự đa dạng của những cảm biến rẻ mà đáng tin cậy và tính toán điện tử công suất làm tăng tính tự động hóa của mobile robot. Tính tự động hóa có nghĩa là robot phải dựa vào chính khả năng của nó để đề xuất ra những dữ liệu vận hành có ích từ bộ phận cảm biến và tự nó đưa ra quyết định thích hợp. Một trong các yếu tố cơ bản của mobile robot là khả năng định hướng tốt trong phạm vi môi trường chưa xác định, bằng cách sử dụng những quan sát thích hợp từ môi trường.

Mobile robot rất đa dạng và ngày càng phổ biến. Có nhiều cách phân loại mobile robot như sau:

- + Theo môi trường hoạt động: trên nền đất, trong môi trường nước, trong không trung,..
- + Theo phương thức di chuyển: bánh xe, chân, bánh xích,...



Hình 1.1: Một số mobile robot trong thực tế

1.2 Giới thiệu Computer Vision và mobile robot sử dụng Computer Vision

1.2.1 Computer Vision

Xử lý ảnh hay còn gọi là thị giác máy (Computer Vision) là một lĩnh vực liên quan đến việc xử lý các dữ liệu thu được từ ảnh để giúp máy tính có được các thông tin cần thiết từ ảnh. Từ góc nhìn kỹ thuật thì nhiệm vụ của xử lý ảnh là cố gắng tiếp cận với sự tự động hóa mà hệ thống thị giác của con người có thể thực hiện.

Các nhiệm vụ của xử lý ảnh bao gồm thu thập, xử lý, phân tích và hiểu được các loại ảnh, và từ đó trích lọc ra các lớp dữ liệu về thế giới thực để có thể thu được dữ liệu dạng số. Các lĩnh vực con của xử lý ảnh bao gồm tái tạo cảnh, phát hiện sự kiện, nhận diện vật thể, ước lượng chuyển động. Xử lý ảnh được ứng dụng rộng rãi trong nhiều lĩnh vực từ gia đình cho đến công nghiệp. Một số ứng dụng của xử lý ảnh có thể được kể đến như:

- Tự động chẩn đoán: Ví dụ trong các lĩnh vực chế tạo.
 - Trợ giúp con người trong các nhiệm vụ phân loại, nhận diện.
 - Kiểm soát, điều khiển quá trình: Ví dụ trong lĩnh vực robot công nghiệp.
 - Phát hiện sự kiện. Ví dụ: Các lĩnh vực giám sát bằng thị giác hoặc đếm người.
 - Tương tác. Ví dụ: Trong các lĩnh vực hỗ trợ tương tác giữa người và máy.
 - Mô hình hóa đối tượng hoặc môi trường xung quanh. Ví dụ: Trong các lĩnh vực y tế.
 - Định hướng. Ví dụ: Trong các loại phương tiện tự hành hoặc mobile robot
- Trong thế giới kỹ thuật số ngày nay, hình ảnh và video phổ biến khắp mọi nơi. Không những thế, với sự phát triển nhanh chóng của khoa học công nghệ, các thiết bị xử lý, tính toán trở nên không những mạnh mẽ hơn mà giá thành cũng hợp lý hơn. Điều này tạo điều kiện thuận lợi rất lớn cho việc tạo ra các ứng dụng về xử lý hình ảnh, video. Từ các loại máy tính để bàn, đến laptop, và cả các loại điện thoại di động thông minh nhỏ bé đều có sức mạnh để xử lý các tác vụ về hình ảnh, video. Do đó, thị giác máy tính là một hướng nghiên cứu đầy tiềm năng, là xu hướng nghiên cứu của nhiều kỹ sư, nhà nghiên cứu, nhà khoa học trên thế giới.

Hơn nữa, robot ngày càng đóng vai trò quan trọng trong công nghiệp và gia đình. Chúng sẽ thực hiện những công việc nguy hiểm hoặc lặp đi lặp lại, nhảm chán, và những công việc mà yêu cầu tốc độ xử lý và độ chính xác vượt quá khả năng của con người. Khi người máy ngày càng trở nên tinh vi hơn, thị giác máy tính đóng vai trò ngày càng quan trọng. Người ta sẽ đòi hỏi người máy không những phát hiện, nhận dạng các đối tượng đã biết, mà còn “hiểu” được những gì chúng “thấy” và đưa ra những hành động phù hợp. Ngày nay, phần lớn các robot hoạt động trong môi trường làm việc ít thay đổi và ứng dụng của robot bị hạn chế khi phải hoạt động trong các môi trường không thể mô tả, nắm rõ trước. Hạn chế này là do thiếu các cảm biến thích hợp để nhận biết môi trường, các yếu tố bên ngoài... Thị giác (vision) trong trường hợp này là một loại cảm biến linh động và có nhiều ưu thế lớn so với các loại cảm biến thông thường khác, nó có thể cung cấp rất nhiều thông tin về môi trường hoạt động cho robot.

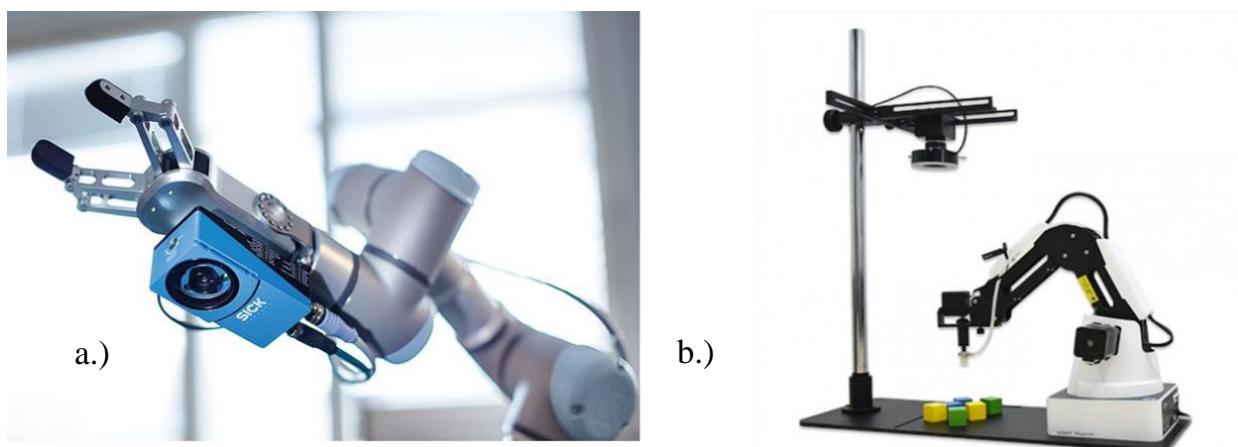
Robot có thị giác có ưu thế lớn trong việc nhận biết môi trường làm việc, do đó nó được sử dụng phổ biến trong nhiều lĩnh vực như công nghiệp, quốc phòng và nghiên cứu. Đặc biệt là các hệ thống cần quan sát, theo dõi mục tiêu, hệ thống phân loại sản phẩm trong công nghiệp, những hệ thống thông minh hỗ trợ lái xe tự động, các hệ thống giám sát và theo dõi chuyển động...

Hệ thống thị giác có thể được phân thành hai loại, theo như cấu trúc của hệ, đó là hệ có camera gắn cố định và hệ có camera gắn trực tiếp trên tay máy hoặc trên thân robot mỗi loại có chức năng và ưu việt riêng:

+ Hệ có camera gắn cố định: camera được gắn cố định so với hệ toạ độ thực, thu thập ảnh của cả mục tiêu và cả môi trường. Mục tiêu của hệ này là cung cấp tín hiệu điều khiển sao cho tay máy, robot đạt được vị trí mong muốn. Hoặc ứng dụng trong các loại hệ thống phân loại sản phẩm, đếm sản phẩm...

+ Hệ có camera gắn trực tiếp trên tay máy hoặc trên thân robot: mục đích của cấu trúc này là điều khiển tay máy (robot) sao cho ảnh của mục tiêu cố định hay di động luôn được duy trì ở vị trí mong muốn trên mặt phẳng ảnh thu được. Trên cơ sở ảnh thu được từ camera, được số hoá và tích hợp trong vòng điều khiển phản hồi, hệ thống điều khiển tay máy, robot.

Ngoài ra còn một số được trang bị kết hợp cả hệ camera gắn cố định và trực tiếp với nhau, giúp xác định vị trí chính xác đồng thời môi trường chuyển động của robot của hệ linh hoạt hơn, do hai camera giúp tăng tầm nhìn của robot.



Hình 1.2: Robot hệ có camera trực tiếp tay máy(a) và robot có camera cố định(b)

1.2.2 Mobile robot sử dụng Computer Vision

Việc tích hợp Computer Vision vào robot đem lại cho robot khả năng nhận biết môi trường làm việc rất tốt, nhờ đó, tăng khả năng linh hoạt trong khi hoạt động. Khi đó mobile robot có thể hoạt động, đáp ứng tốt hơn với sự thay đổi của môi trường, có thể tránh được vật cản, thực thi nhiệm vụ định vị và gấp đôi tượng, phát hiện chuyển động và theo dõi giám sát... Các ứng dụng này chủ yếu dành cho mobile robot tự hành thông minh, sử dụng trong nghiên cứu khoa học, khám phá và thám hiểm...

Hệ thống thị giác có thể được phân thành hai lớp, theo như cấu trúc của hệ, đó là hệ có camera gắn cố định và hệ có camera gắn trên tay máy (eye-in-hand). Trong hệ camera cố định, camera được gắn cố định so với hệ toạ độ thực, thu thập ảnh của cả mục tiêu và cả môi trường. Mục tiêu của hệ này là cung cấp tín hiệu điều khiển sao cho tay máy đạt được vị trí mong muốn. Mục đích của cấu trúc eye-in-hand là điều khiển tay máy sao cho ảnh của mục tiêu cố định hay di động luôn được duy trì ở vị trí mong muốn trên mặt phẳng ảnh thu được. Trên cơ sở ảnh thu được từ camera, được số hoá và tích hợp trong vòng điều khiển phản hồi, hệ thống điều khiển các khớp của tay máy.

Các nghiên cứu của môi trường động bị bỏ khá xa so với của môi trường tĩnh do bị ảnh hưởng khá lớn của tốc độ tính toán cũng như độ chính xác của việc phân tích ảnh. Do vậy có nhiều thuật toán để cải thiện tốc độ xử lý và xem xét đến các quá trình nhiễu tác động lên quá trình thu thập hình ảnh.

Một thách thức của thị giác máy là việc phân loại đối tượng. Một robot có thể phải đối mặt với nhiều đối tượng khác nhau, trong khi đó chỉ có một mục tiêu quan tâm, còn các đối tượng khác thì không. Để nhận biết được thông tin về vị trí của mục tiêu trong môi trường động, các đặc trưng của mục tiêu rất quan trọng. Các điểm lỗ, các góc cạnh, các đặc điểm hình học của mục tiêu được phân tích qua quá trình nhận dạng. Đặc tính về trọng tâm của mục tiêu có thể tính dễ dàng qua moment bậc nhất trong môi trường tĩnh, nhưng trong môi trường động việc này rất khó vì chi phí thời gian tính toán lớn.



Hình 1.3: Một số mobile robot sử dụng Computer Vision

1.3 Vấn đề tập trung nghiên cứu của đề tài

Việc tích hợp Computer Vision vào robot cho thấy lợi ích mang lại từ robot rất cần thiết với xã hội trong nhiều lĩnh vực. Ngoài giúp con người quan sát trong những môi trường khó khăn còn tăng cường tự động hóa, giám sát trong công nghiệp.

Chính vì những lợi ích mang lại trên, đề tài tập trung nghiên cứu là:

- Thiết kế chế tạo mô hình mobile robot.
- Xây dựng chương trình xử lý ảnh phát hiện khuôn mặt và phát hiện màu đỏ.
- Áp dụng bộ điều khiển PID điều khiển động cơ DC.
- Dùng chương trình xử lý ảnh gửi tín hiệu đến vi điều khiển, điều khiển động cơ DC.

CHƯƠNG 2 : ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

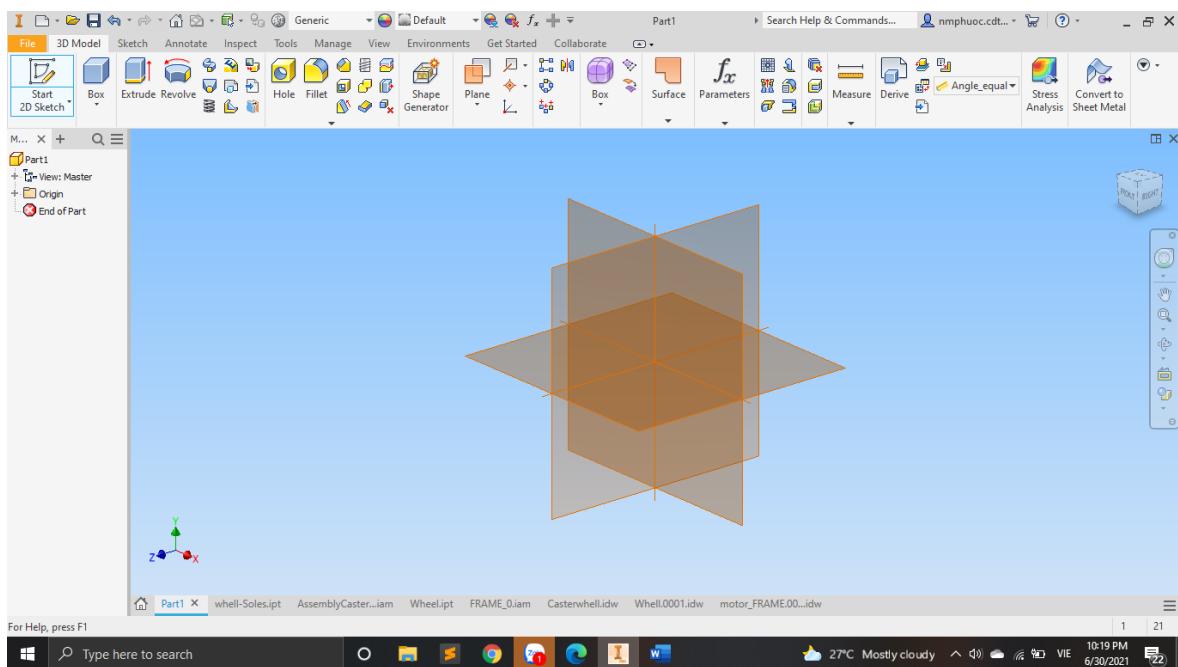
2.1 Đối tượng nghiên cứu

2.1.1 Phần mềm Inventor

Autodesk Inventor được phát triển bởi công ty phần mềm Autodesk USA, là phần mềm thiết kế mô hình 3D phổ biến hiện nay. Đây là phần mềm được phát triển chuyên cho thiết kế các sản phẩm cơ khí, có giao diện trực quan, giúp người dùng thuận tiện khi sử dụng với những tính năng nổi trội như sau:

- Xây dựng dễ dàng mô hình 3D của chi tiết (Part).
- Thiết lập các bản 2D từ mô hình 3D nhanh chóng và chuẩn xác (Drawing).
- Tạo bản vẽ lắp từ các chi tiết đã thiết kế một cách tối ưu (Assembly).
- Mô phỏng quá trình tháo lắp các chi tiết từ bản vẽ lắp hoàn chỉnh một cách trực quan và sinh động (Presentation).
- Thiết kế nhanh các chi tiết kim loại dạng tấm (Sheet metal).
- Thiết kế các chi tiết máy như: Trục, bộ truyền bánh răng, bộ truyền đai, bộ truyền xích, mối ghép bulông-đai óc, cam, chốt, then, ốc bi, lò xo,... một cách nhanh chóng trong môi trường Assembly.
- Thiết kế nhanh và chính xác các loại khuôn mẫu (Mold Design).
- Thiết kế nhanh các đường ống phức tạp (Pipe&Tupe).
- Cho phép sử dụng thư viện các loại dây điện và cáp điện để chạy dây với bán kính uốn phù hợp trong thiết kế điện (Cable & Wiring):.
- Mô phỏng động và động lực học của cơ cấu máy (Dynamic simulation).
- Phân tích ứng suất, tối ưu hóa thiết kế sản phẩm (Analysis Stress and Optimize).
- Thiết kế nhanh các sản phẩm nhựa (Inventor plastic & tooling).
- Có thư viện chi tiết đa dạng và chuẩn hóa (Content center).
- Liên kết được với nhiều phần mềm CAD khác.

Những tiện ích của Autodesk Inventor mang lại hiệu quả khá cao trong việc xây dựng mô hình, vì thế robot được xây dựng trên môi trường Inventor.



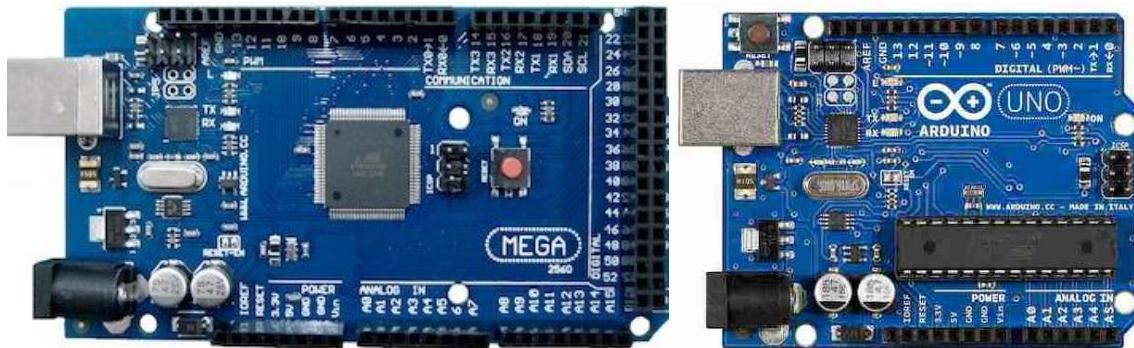
Hình 2.1: Giao diện phần mềm AUTODESK INVENTOR

2.1.2 Tổng quan Arduino

2.1.2.1 Giới thiệu Arduino

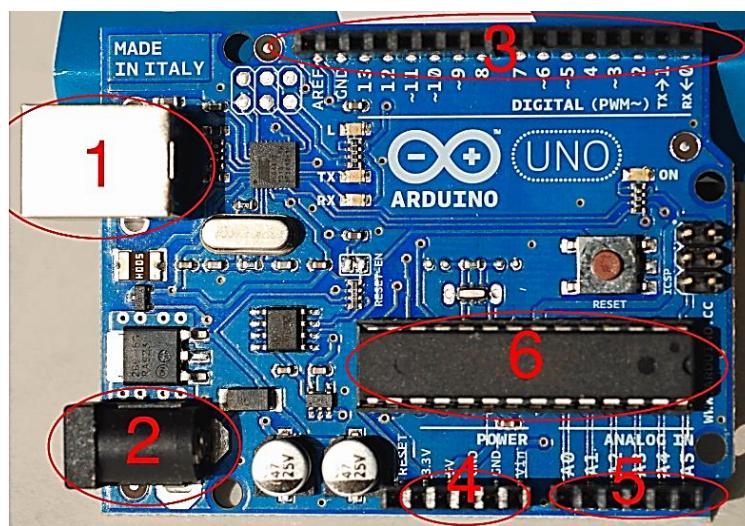
Arduino là một board mạch vi điều khiển do một nhóm giáo sư và sinh viên Ý thiết kế và đưa ra đầu tiên vào năm 2005. Mạch Arduino được sử dụng để cảm nhận và điều khiển nhiều đối tượng khác nhau. Nó có thể thực hiện nhiều nhiệm vụ từ lấy tín hiệu từ cảm biến đến điều khiển đèn, động cơ, và nhiều đối tượng khác. Ngoài ra mạch còn có khả năng kết nối với nhiều module khác nhau như module đọc thẻ từ, ethernet shield, sim900A,để tăng khả ứng dụng của mạch.

Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM, Atmel 32-bit,... Hiện phần cứng của Arduino có tất cả 6 phiên bản, tuy nhiên phiên bản thường được sử dụng nhiều nhất là Arduino Uno và Arduino Mega. Arduino là vi điều khiển được sử dụng rộng rãi trên thế giới với những ưu điểm như rẻ, tương thích được với nhiều hệ điều hành, chương trình lập trình đơn giản, rõ ràng, dễ sử dụng, sử dụng mã nguồn mở và có thể kết hợp với nhiều module khác nhau.



Hình 2.2: Arduino Mega 2560 và Arduino Uno R3

2.1.2.2 Phản ứng của Arduino



Hình 2.3: Phản ứng của Arduino Uno R3

Mỗi loại Arduino sẽ có cấu hình phản ứng khác nhau, tuy nhiên nó vẫn có đầy đủ các phản ứng cơ bản như trên Arduino UNO R3, gồm có 6 phản ứng cơ bản:

1. Cổng USB (loại B): Đây là cổng giao tiếp để ta upload code từ máy tính lên vi điều khiển. Đồng thời nó cũng là giao tiếp serial để truyền dữ liệu giữa vi điều khiển với máy tính.
2. Jack nguồn: Để chạy Arduino thì có thể lấy nguồn từ cổng USB ở trên, nhưng không phải lúc nào cũng có thể cắm với máy tính được. Lúc đó, ta cần một nguồn từ 9V đến 12V.
3. Hàng header thứ nhất: Đánh số từ 0 đến 12 là hàng digital pin, nhận vào hoặc xuất ra các tín hiệu số. Ngoài ra có một pin mass (GND) và pin điện áp tham chiếu (AREF).

4. Hàng header thứ hai: Chủ yếu liên quan đến điện áp mass (GND), nguồn.
5. Hàng header thứ ba: Các chân để nhận vào hoặc xuất ra các tín hiệu analog.

Ví dụ như đọc thông tin của các thiết bị cảm biến.

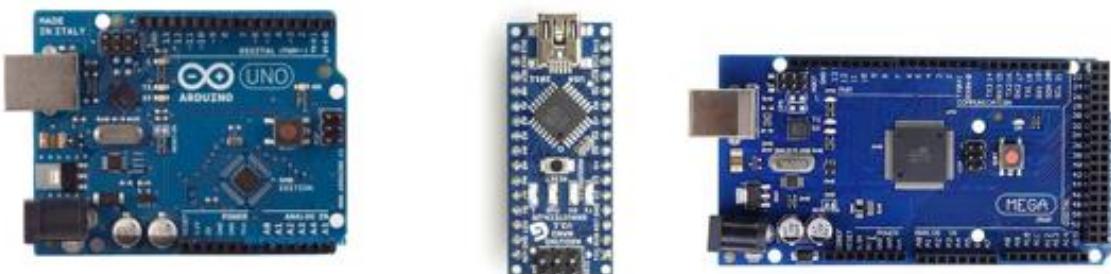
6. Vi điều khiển AVR: Đây là bộ xử lý trung tâm của toàn bo mạch. Với mỗi mẫu Arduino khác nhau thì con chip này khác nhau. Ở con Arduino Uno này thì sử dụng ATMega328.

Đối với các Arduino khác tùy vào kích thước và phần cứng sẽ có cách sắp xếp linh kiện khác nhau, tuy nhiên vẫn giữ cấu hình cơ bản như Arduino UNO R3, một số Arduino có kích thước nhỏ sẽ không có jack cáp nguồn DC.

2.1.2.3 Một số loại Arduino điển hình

Arduino gồm rất nhiều phiên bản với các cấu hình khác nhau nhưng hầu như vẫn đảm nhận được các chức năng cơ bản tương đương nhau.

Ba loại Arduino được sử dụng phổ biến nhất là Arduino UNO R3, Nano, Mega 2560:



Hình 2.4: Các Arduino phổ biến

Bên cạnh đó còn nhiều loại Arduino được nâng cao hơn nhằm đáp ứng nhu cầu sử dụng cao hơn trong công nghiệp như: Arduino Yun, Arduino Ethernet, Arduino Tre, Arduino Robot,...



Hình 2.5: Một số Arduino cao cấp

2.1.2.4 Ứng dụng của Arduino

Arduino có nhiều ứng dụng trong đời sống, trong học tập phục vụ hỗ trợ những người bắt đầu nghiên cứu với vi điều khiển, trong việc chế tạo các thiết bị điện tử chất lượng cao. Một số ứng dụng có thể kể đến như:

- Lập trình Robot: Arduino chính là một phần quan trọng trong trung tâm xử lý giúp điều khiển được hoạt động của Robot.
- Lập trình máy bay không người lái. Có thể nói đây là ứng dụng có nhiều kỳ vọng trong tương lai.
- Game tương tác: Chúng ta có thể dùng Arduino để tương tác với Joystick, màn hình,... để chơi các trò như Tetrix, phá gạch, Mario,... và nhiều game rất sáng tạo.
- Arduino cũng được ứng dụng trong máy in 3D và nhiều ứng dụng khác tùy thuộc vào khả năng sáng tạo của người sử dụng.

Trong đề tài, sử dụng một board Arduino Mega 2560 (Master) làm trung tâm giao tiếp với máy tính, truyền tín hiệu điều khiển và hai board Arduino Uno (Slave) làm nhiệm vụ điều khiển trực tiếp hai động cơ truyền động cho robot. Có thể nói rằng Arduino đảm nhiệm vai trò là bộ xử lý trung tâm cho toàn bộ Robot, vừa đảm nhiệm

vai trò nhận tín hiệu điều khiển từ máy tính vừa xử lý và xuất tín hiệu để động cơ hoạt động.

2.1.3 Phần mềm IDE Arduino

2.1.3.1 Khái niệm IDE Arduino



Hình 2.6: Logo Arduino IDE

Arduino IDE được viết tắt (Arduino integrated Development Environment) là một phần mềm mã nguồn mở chủ yếu được sử dụng để soạn thảo văn bản và biên dịch mã vào module Arduino. Đây là một phần mềm Arduino chính thức, giúp cho việc biên dịch mã trở nên dễ dàng mà ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được.

Nó có các phiên bản cho các hệ điều hành như MAC, Windows, Linux và chạy trên nền tảng Java đi kèm với các chức năng và lệnh có sẵn đóng vai trò quan trọng để gỡ lỗi, chỉnh sửa và biên dịch mã trong môi trường.

Có rất nhiều các module Arduino như Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều module khác. Mỗi module chứa một bộ vi điều khiển trên bo mạch được lập trình và chấp nhận thông tin dưới dạng mã.

Mã chính, còn được gọi là sketch, được tạo trên nền tảng IDE sẽ tạo ra một file Hex, sau đó được chuyển và tải lên trong bộ điều khiển trên bo. Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình chỉnh sửa và Trình biên dịch, phần đầu sử dụng để viết mã được yêu cầu và phần sau được sử dụng để biên dịch và tải mã lên module Arduino. Môi trường này hỗ trợ cả ngôn ngữ C và C++. Khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và sau đó được gửi đến bo mạch bằng cáp USB. Mỗi board Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file hex và chạy

theo mã được viết. Các thư viện rất hữu ích để thêm chức năng bổ sung vào module Arduino. Có một danh sách các thư viện có thể thêm bằng cách nhấp vào nút Sketch trong thanh menu và đi tới Include Library.



Hình 2.7: Giao diện Arduino IDE

2.1.3.2 Cấu trúc chương trình

Một chương trình Arduino cơ bản có 2 phần chính: void setup() và void loop(). Phần setup() gọi là phần “cài đặt” dùng để chuẩn bị cho một chương trình Arduino. Các câu lệnh của phần này được đặt trong cặp dấu ngoặc nhọn ngay sau void setup().

Vd: Cài đặt chân 13 của board Arduino hoạt động ở chế độ xuất tín hiệu thì viết như sau:

```
void setup() {
    // put your setup code here, to run once:
    pinMode(13,OUTPUT);
}
```

Phần loop() là nơi chứa mã thực thi chính. Những lệnh trong phần này sẽ chạy liên tục. Các câu lệnh của phần này được đặt trong cặp dấu ngoặc nhọn ngay sau void loop()

Ví dụ: Điều khiển một bóng LED nối với chân 13 nhấp nháy liên tục thì viết như sau:

```
void loop() {
```

```
// put your main code here, to run repeatedly:  
digitalWrite(13, HIGH);  
delay(1000);  
digitalWrite(13, LOW);  
delay(1000);  
}
```

2.1.4 Ngôn ngữ Python

2.1.4.1 Khái niệm Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức có cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.



Hình 2.8: Logo Python

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động, do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của

Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

+ Đặc điểm:

Ngôn ngữ lập trình đơn giản, dễ học: Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép người dùng tập trung vào những giải pháp chứ không phải cú pháp. Miễn phí, mã nguồn mở: có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

Khả năng di chuyển, chạy trên nhiều nền tảng: Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux. Khả năng mở rộng và có thể nhúng: Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, người dùng có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

Ngôn ngữ thông dịch cấp cao: Không giống như C/C++, với Python, người dùng không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu, không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào. Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến: Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của người dùng trở nên dễ hơn rất nhiều, đơn giản vì không phải tự viết tất cả code.

Hướng đối tượng: Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan.

Với OOP, người dùng có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

2.1.4.2 Lịch sử hình thành

Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI (Centrum voor Wiskunde en Informatica - Trung tâm Toán-Tin học tại Amsterdam, Hà Lan). Vì vậy, các phiên bản Python đầu tiên đều do CWI phát hành. Phiên bản cuối cùng phát hành tại CWI là 1.2.

Vào năm 1995, Guido chuyển sang CNRI (Corporation for National Research Initiatives) ở Reston, Virginia. Tại đây, ông phát hành một số phiên bản khác. Python 1.6 là phiên bản cuối cùng phát hành tại CNRI.

Sau bản phát hành 1.6, Guido rời bỏ CNRI để làm việc với các lập trình viên chuyên viết phần mềm thương mại. Tại đây, ông có ý tưởng sử dụng Python với các phần mềm tuân theo chuẩn GPL. Sau đó, CNRI và FSF (Free Software Foundation - Tổ chức phần mềm tự do) đã cùng nhau hợp tác để làm bản quyền Python phù hợp với GPL. Cùng năm đó, Guido được nhận Giải thưởng FSF vì Sự phát triển Phần mềm tự do (Award for the Advancement of Free Software).

Phiên bản 1.6.1 ra đời sau đó là phiên bản đầu tiên tuân theo bản quyền GPL. Tuy nhiên, bản này hoàn toàn giống bản 1.6, trừ một số sửa lỗi cần thiết.

Python 2: vào năm 2000, Guido và nhóm phát triển Python dời đến BeOpen.com và thành lập BeOpen PythonLabs team. Phiên bản Python 2.0 được phát hành tại đây. Sau khi phát hành Python 2.0, Guido và các thành viên PythonLabs gia nhập Digital Creations.

Python 2.1 ra đời kế thừa từ Python 1.6.1 và Python 2.0. Bản quyền của phiên bản này được đổi thành Python Software Foundation License. Từ thời điểm này trở đi, Python thuộc sở hữu của Python Software Foundation (PSF), một tổ chức phi lợi nhuận được thành lập theo mẫu Apache Software Foundation.

Python 3, còn gọi là Python 3000 hoặc Py3K: Dòng 3.x sẽ không hoàn toàn tương thích với dòng 2.x, tuy vậy có công cụ hỗ trợ chuyển đổi từ các phiên bản 2.x sang 3.x. Nguyên tắc chủ đạo để phát triển Python 3.x là "bỏ cách làm việc cũ nhằm

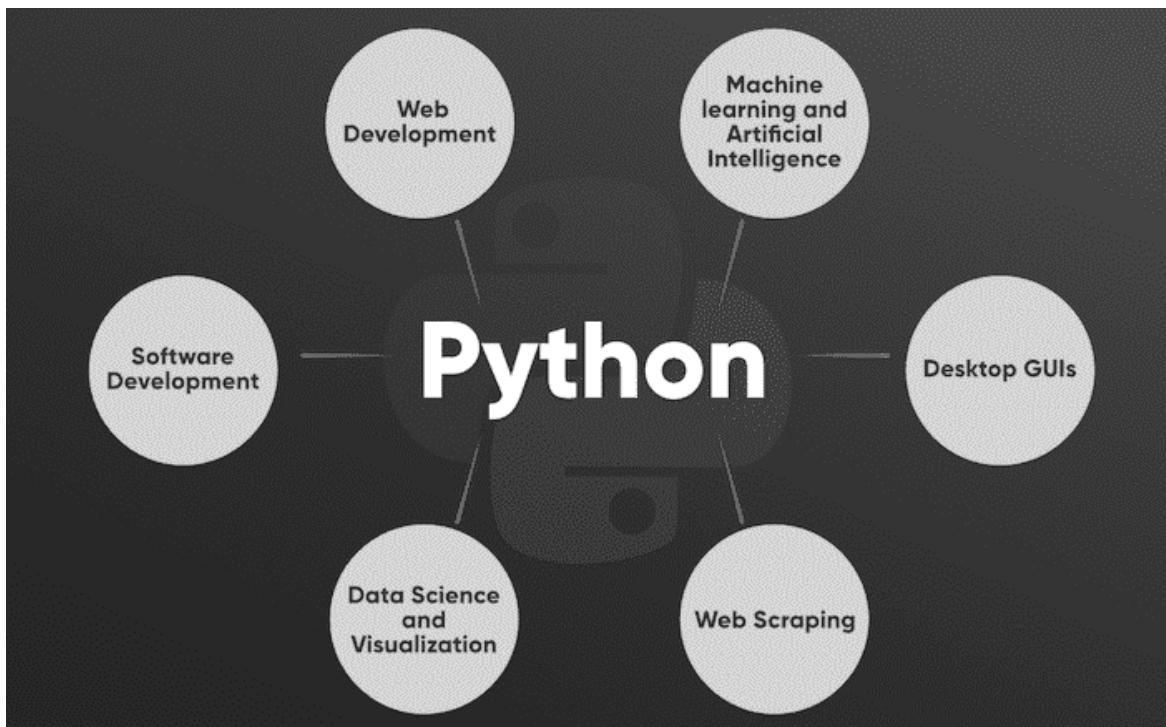
hạn chế trùng lặp về mặt chức năng của Python". Trong PEP (Python Enhancement Proposal) có mô tả chi tiết các thay đổi trong Python.

Đề tài sử dụng Python phiên bản 3.7.9 cho việc xây dựng chương trình xử lý ảnh.

2.1.4.3 Ứng dụng Python

Như đã mô tả giới thiệu, về Python là ngôn ngữ lập trình hướng đối tượng bậc cao, dùng để phát triển website và nhiều ứng dụng khác. Với cú pháp cực kỳ đơn giản, Python là lựa chọn hoàn hảo cho các lập trình viên trong nhiều lĩnh vực lập trình phần mềm, sau đây là những ứng dụng nổi bật và được sử dụng rộng rãi.

Xây dựng website với Framework của Python: Các web frameworks Django và Flask ngày càng trở nên phổ biến thì việc phát triển web bằng Python trở nên dễ dàng hơn bao giờ hết. Flask cung cấp cho người dùng các thành phần cốt lõi thường được sử dụng nhất của khung ứng dụng web như URL routing, request & response object, template... Django được thiết kế để giúp các nhà phát triển đưa các ứng dụng từ ý tưởng đến hoàn thành càng nhanh càng tốt, nó có đầy đủ các thư viện/module cần thiết.



Hình 2.9: Một số ứng dụng Python

Python dùng để tạo nguyên mẫu phần mềm: Bên cạnh mảng lập trình ứng dụng web, Python còn được ứng dụng trong tạo nguyên mẫu phần mềm. Nếu nói về tốc độ thì Python sẽ chậm hơn so với các ngôn ngữ biên dịch như Java và C++. Python sẽ không phải lựa chọn tốt nhất nếu người dùng đang bị giới hạn về nguồn lực, yêu cầu hiệu quả cao. Tuy nhiên, bên cạnh đó Python lại là một ngôn ngữ lập trình tuyệt vời để tạo ra những nguyên mẫu - bản chạy thử.

Ứng dụng trong khoa học và tính toán: Ngoài việc lập trình ứng dụng web, tạo nguyên mẫu phần mềm, Python còn được ứng dụng trong khoa học và tính toán. Python sở hữu thư viện cho khoa học, tính toán số liệu như NumPy, SciPy. Chúng được sử dụng cho mục đích chung trong việc tính toán. Bên cạnh đó, trong thư viện của mình, Python còn có EarthPy cho khoa học Trái Đất, AstroPy cho thiên văn học... Python cũng được ứng dụng trong khai thác dữ liệu, deep learning, machine learning.

Viết tool để tự động hóa công việc: Chắc hẳn những ai đam mê công nghệ chắc chắn biết những tool thú vị ở trong Google Chrome Extension hoặc vô vàn tool hữu ích có thể download trên bất kỳ trang web nào. Nếu như trước kia, những tool này cần phải viết bằng các ngôn ngữ khó như Java, PHP... thì hiện nay, với Python, tốc độ sản xuất tool có thể tăng lên gấp 10 lần. Python sẽ giúp người dùng nhanh chóng tạo ra các công cụ như tạo từ điển, công cụ dò quét trang web, công cụ xử lý dữ liệu file, gửi email...

Ngôn ngữ lập trình Python là một trong những ngôn ngữ lập trình được sử dụng nhiều nhất trong việc phát triển trí tuệ nhân tạo. Python được nhiều người lựa chọn để lập trình trí tuệ nhân tạo vì tính phổ biến, dễ học và khả năng tương thích với công nghệ 4.0. Nó có thể được sử dụng một cách trọn vẹn với các cấu trúc dữ liệu và các thuật toán AI được sử dụng thường xuyên khác. Rất nhiều thư viện Python có thể và đang được sử dụng cho lĩnh vực Machine Learning, Deep Learning và AI. Có thể kể tới: Tensorflow, Theano và PyTorch. Càng nhiều lập trình viên làm việc trong lĩnh vực này thì số lượng nguồn (resources) và thư viện (libraries) lại càng tăng lên.

2.1.5 Phần mềm PyCharm

2.1.5.1 Khái niệm PyCharm

Phần mềm PyCharm cung cấp một bộ công cụ hoàn chỉnh cho các nhà phát triển Python chuyên nghiệp. PyCharm được xây dựng xung quanh một trình soạn thảo mã sâu sắc, và một trình sửa lỗi cho cái nhìn rõ ràng về hoạt động của mã. PyCharm cung cấp khả năng tích hợp với các công cụ công tác như hệ thống kiểm soát phiên bản và các tracker. Trình biên tập chuyên nghiệp mở rộng các yếu tố cần thiết bằng cách tích hợp liền mạch với các khuôn khổ web, các công cụ JavaScript, ảo hóa và hỗ trợ containerization.

Một khía cạnh quan trọng của chương trình là hiểu được nền tảng mã đang đưa vào. PyCharm đảm bảo người dùng có thể khám phá dự án, chỉ với một vài thao tác trên phím, nó cung cấp cho người dùng một cái nhìn tổng quan về cấu trúc dự án và cho phép truy cập vào các tài liệu có liên quan ngay từ trình soạn thảo. Hiểu được một nền tảng code nhanh hơn có nghĩa là thúc đẩy nhanh hơn quá trình phát triển của người dùng.

Pycharm là một giải pháp phù hợp cho Python developers vì IDE này hỗ trợ nhiều extensions, môi trường ảo (Virtual Environment), nhiều tính năng thông minh như bộ code completion, tự động thực lè, phát hiện văn bản trùng lặp và kiểm tra lỗi. Ngoài ra còn có các tính năng tìm kiếm mã nguồn thông minh để tìm kiếm từng từ.



Hình 2.10: Logo PyCharm

2.1.5.2 Ứng dụng

PyCharm giúp chúng ta viết mã chất lượng cao hơn. Nó bao gồm các lược đồ màu (color schemes) cho từ khóa, lớp và hàm. Điều này giúp tăng khả năng đọc và hiểu mã. Giúp xác định lỗi một cách dễ dàng. Cung cấp tính năng tự động hoàn thành và hướng dẫn để hoàn thành mã.

Nó giúp các nhà phát triển chỉnh sửa và cải thiện mã với ít nỗ lực và thời gian hơn. Với điều hướng mã, lập trình viên có thể dễ dàng điều hướng đến một hàm, lớp hoặc tệp. Một lập trình viên có thể xác định vị trí một phần tử, một ký hiệu hoặc một biến trong mã nguồn trong thời gian ngắn. Hơn nữa, bằng cách sử dụng chế độ thấu kính, nhà phát triển có thể kiểm tra và vá lỗi toàn bộ mã nguồn một cách kỹ lưỡng.

Nó có lợi thế là thực hiện các thay đổi hiệu quả và nhanh chóng đối với cả biến cục bộ (local variables) và biến toàn cục (global variables). Tái cấu trúc trong PyCharm cho phép các nhà phát triển cải thiện cấu trúc bên trong mà không thay đổi hiệu suất bên ngoài của code. Pycharm cũng giúp phân chia các lớp và chức năng mở rộng tốt hơn với sự trợ giúp của phương pháp trích xuất.

Nó giúp các nhà phát triển tạo các ứng dụng web bằng Python. Nó hỗ trợ các công nghệ web phổ biến như HTML, CSS và JavaScript. Các nhà phát triển có lựa chọn chỉnh sửa trực tuyến với IDE này. Đồng thời, họ có thể xem trước trang web đã cập nhật/đã tạo. Các nhà phát triển có thể theo dõi các thay đổi trên trình duyệt web trực tiếp. PyCharm cũng hỗ trợ AngularJS và NodeJS để phát triển các ứng dụng web.

PyCharm hỗ trợ các thư viện khoa học của Python như Matplotlib, NumPy và Anaconda. Các thư viện khoa học này giúp xây dựng các dự án về Khoa học Dữ liệu và Học máy. Hỗ trợ các biểu đồ tương tác giúp các nhà phát triển hiểu dữ liệu tốt hơn. Nó có khả năng tích hợp với những công cụ khác nhau như IPython, Django và Pytest. Sự tích hợp này giúp thúc đẩy các giải pháp độc đáo.

2.1.6 Thị giác máy tính - Computer Vision (CV)

Thị giác máy tính là một lĩnh vực đã và đang phát triển mạnh mẽ trên thế giới hiện nay. Khái niệm về CV có liên quan tới nhiều ngành học và có nhiều hướng nghiên cứu khác nhau. Đây là một trong những hướng nghiên cứu rộng, đa dạng và có nhiều ứng dụng trong thực tế, có thể kể đến như xử lý ảnh y khoa, robot tự hành,

xử lý ảnh viễn thám, phát giác và theo dõi chuyển động, an ninh, quân sự (phát giác chuyển động, nhận dạng mặt, vân tay, võng mạc, xe, chiến hạm...) cũng như áp dụng trong các ngành nghiên cứu khác...

Trong thế giới kỹ thuật số ngày nay, hình ảnh và video phổ biến khắp mọi nơi. Không những thế, với sự phát triển nhanh chóng của khoa học công nghệ, các thiết bị xử lý, tính toán trở nên không những mạnh mẽ hơn mà giá thành cũng hợp lý hơn. Điều này tạo điều kiện thuận lợi rất lớn cho việc tạo ra các ứng dụng về xử lý hình ảnh, video. Từ các loại máy tính để bàn, đến laptop, và cả các loại điện thoại di động thông minh nhỏ bé điều có sức mạnh để xử lý các tác vụ về hình ảnh, video. Do đó, thị giác máy tính là một hướng nghiên cứu đầy tiềm năng, là xu hướng nghiên cứu của nhiều kỹ sư, nhà nghiên cứu, nhà khoa học trên thế giới.

Thị trường thị giác máy tính đang ngày càng mở rộng và liên tục phát triển. Việc sử dụng thư viện tiêu chuẩn sẽ giúp ta có thể tạo ra ứng dụng mới và các giải pháp xử lý dễ dàng hơn, giảm thời gian cho người lập trình. Nhờ đó việc phát, tạo ra các mẫu ứng dụng bằng các thư viện được cập nhật và phát triển thường xuyên trở nên dễ dàng và thuận tiện hơn. Lĩnh vực thị giác máy tính là một lĩnh vực rộng và phức tạp, nhờ có việc chia sẻ mã nguồn mở và cung cấp thư viện như thế này đã là góp phần rất lớn đẩy mạnh sự phát triển của thị giác máy tính trong những năm gần đây và kể cả tương lai. Nhờ nó mà việc thực hiện được các thuật toán rất phức tạp như dò tìm khuôn mặt, nhận diện vật thể,... trở nên đơn giản hơn bao giờ hết.

Có rất nhiều thư viện thị giác máy mã nguồn mở, mỗi thư viện có một nét riêng của nó mà người dùng có thể khai thác sử dụng để áp dụng cho ứng thị giác máy tính của mình. Hiện nay trên thế giới có nhiều thư viện mã nguồn mở cho người dùng sử dụng, ở đây luận văn nghiên cứu và sử dụng thư viện OpenCV của Intel vì tính phổ biến và những khả năng của nó có thể phục vụ cho thực hiện xử lý ảnh, video. Được viết bằng C++ và thao tác interface bằng Python nên phần performance của TensorFlow cực kỳ tốt. Đối tượng sử dụng nó cũng đa dạng không kém: từ các nhà nghiên cứu, nhà khoa học dữ liệu và dĩ nhiên không thể thiếu các lập trình viên.

2.2 Phương pháp nghiên cứu

2.2.1 Phương thức truyền dữ liệu I2C

2.2.1.1 Giới thiệu về giao tiếp I2C

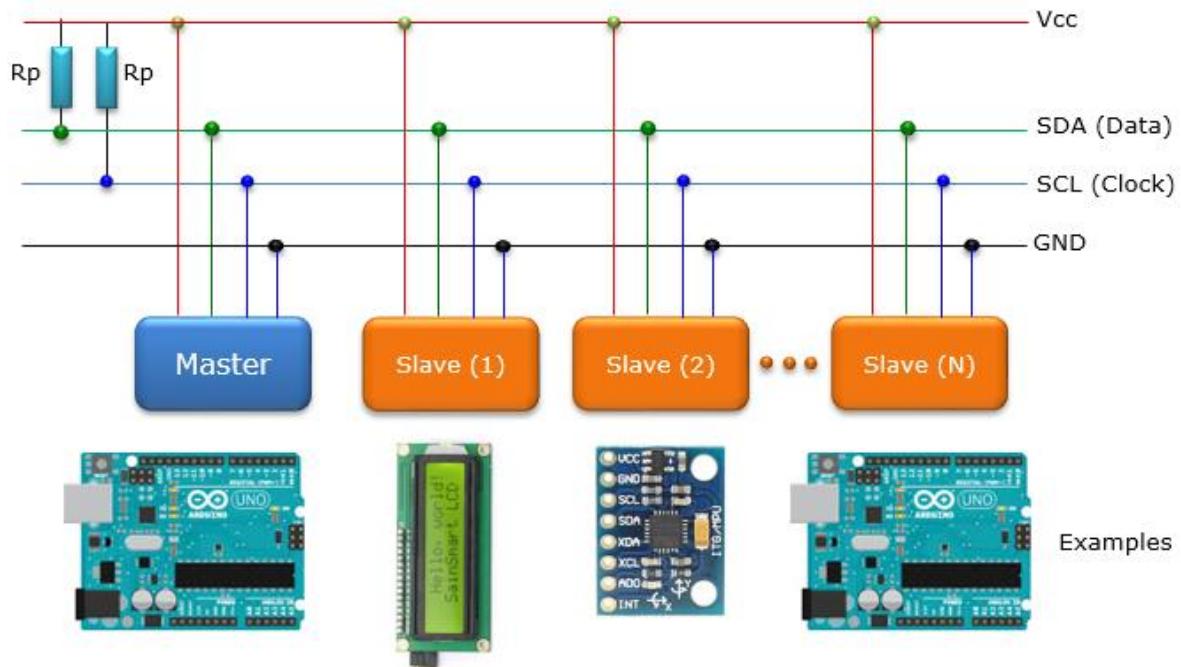
I2C là tên viết tắt của cụm từ tiếng anh “Inter-Integrated Circuit”. Nó là một giao thức giao tiếp được phát triển bởi Philips Semiconductors để truyền dữ liệu giữa một bộ xử lý trung tâm với nhiều IC trên cùng một board mạch chỉ sử dụng hai đường truyền tín hiệu.

Do tính đơn giản của nó nên loại giao thức này được sử dụng rộng rãi cho giao tiếp giữa vi điều khiển và mảng cảm biến, các thiết bị hiển thị, thiết bị IoT, EEPROMs, v.v ...

I2C là giao thức truyền tin nối tiếp đồng bộ sử dụng 2 dây: SDA và SCL trong đó SDA (serial data) là đường truyền nhận dữ liệu và SCL(serial clock) là đường truyền xung nhịp. Các đường SDA và SCL trên các thiết bị có cấu hình cực máng hở (open-drain) hoặc collector hở (open-collector) vì thế chúng cần điện trở kéo lên (pull-up resistor) khoảng từ 1 đến 10($k\Omega$). Ở trạng thái nghỉ thì 2 chân này ở mức cao. Lý do sử dụng một hệ thống cực máng hở (open drain) là để không xảy ra hiện tượng ngắn mạch, điều này có thể xảy ra khi một thiết bị cố gắng kéo đường dây lên cao và một số thiết bị khác cố gắng kéo đường dây xuống thấp

Tốc độ truyền: 100Kbits/s ở chế độ chuẩn (Standard mode), có thể lên tới 400Kbits/s ở chế độ nhanh(Fast mode) và cao nhất là 3,4Mbits/s ở chế độ cao tốc (High-speed mode).

Trên đường truyền giao tiếp I2C: Có thể là một chủ-một tớ (One Master One Slave), một chủ- nhiều tớ (One Master-Multi Slave) hoặc nhiều chủ-nhiều tớ (Multi Master –Multi Slave). Một thiết bị tớ Slave cũng có thể trở thành chip chủ nếu nó có khả năng.



Hình 2.11: Sơ đồ giao tiếp I2C

2.2.1.2 Đặc điểm

Sau đây là một số đặc điểm quan trọng của giao thức giao tiếp I2C:

- Chỉ cần có hai đường bus (dây) chung để điều khiển bất kỳ thiết bị / IC nào trên mạng I2C
- Không cần thỏa thuận trước về tốc độ truyền dữ liệu như trong giao tiếp UART. Vì vậy, tốc độ truyền dữ liệu có thể được điều chỉnh bất cứ khi nào cần thiết
- Cơ chế đơn giản để xác thực dữ liệu được truyền
- Sử dụng hệ thống địa chỉ 7 bit để xác định một thiết bị / IC cụ thể trên bus I2C
- Các mạng I2C dễ dàng mở rộng. Các thiết bị mới có thể được kết nối đơn giản với hai đường bus chung I2C.

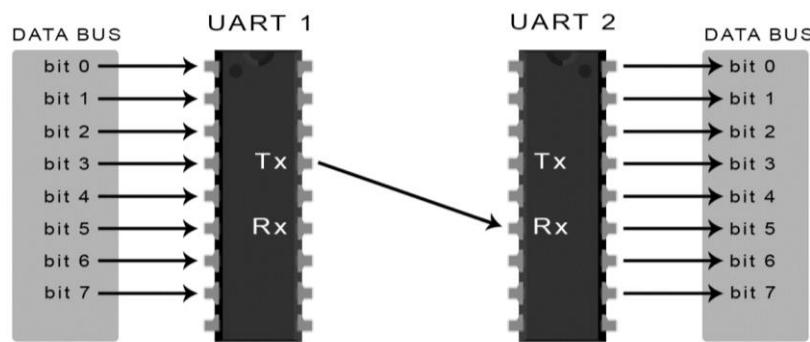
2.2.2 Phương thức truyền dữ liệu UART

2.2.2.1 Giới thiệu về giao tiếp UART

UART có tên đầy đủ là Universal Asynchronous Receiver – Transmitter. Nó là một mạch tích hợp được sử dụng trong việc truyền dẫn dữ liệu nối tiếp giữa máy tính và các thiết bị ngoại vi.

UART có chức năng chính là truyền dữ liệu nối tiếp. Trong UART, giao tiếp giữa hai thiết bị có thể được thực hiện theo hai phương thức là giao tiếp dữ liệu nối tiếp và giao tiếp dữ liệu song song.

Trong giao tiếp UART cơ bản, thiết bị truyền và thiết bị nhận giao tiếp theo cách thức như sau: Phần cứng – hardware- UART sẽ chuyển đổi dữ liệu song song nhận được từ vi xử lý, vi điều khiển và chuyển chúng thành dữ liệu nối tiếp. Dữ liệu nối tiếp này sẽ được truyền đến thiết bị nhận và tại đây, hardware UART sẽ chuyển đổi ngược lại thành dữ liệu song song để truyền về vi điều khiển, vi xử lý của thiết bị nhận.



Hình 2.12: Cách truyền dữ liệu UART

Các chân sử dụng cho giao tiếp UART được gọi là TX ở thiết bị truyền và RX ở thiết bị nhận. Đồng thời, có các thanh ghi – shift registers – được hiểu như là một phần của UART hardware (2 loại thanh ghi được sử dụng ở đây là: Transmitter Shift Register và Receiver Shift Register).

Trong giao tiếp UART, dữ liệu được truyền không đồng bộ, nghĩa là không cần tín hiệu clock hoặc các tín hiệu timing khác để đồng bộ, kiểm tra dữ liệu giữa thiết bị truyền và thiết bị nhận. Thay vào đó, UART sử dụng các bit đặc biệt được gọi là Start và Stop bits.

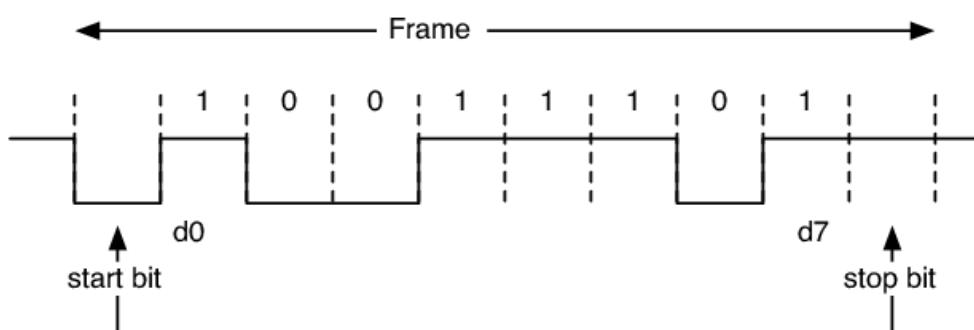
Thiết bị nhận UART sẽ nhận dữ liệu từ chân RX và xác định đâu là dữ liệu thực sau khi loại trừ start và stop bits. Parity bit được sử dụng để kiểm tra độ chính xác của dữ liệu. Phụ thuộc vào sự chia cắt của start, parity và stop bits từ gói dữ liệu, tất cả dữ liệu sẽ được chuyển từ nối tiếp sang song song và được lưu dưới các thanh ghi

– shift register. Những dữ liệu song song này sẽ được truyền đến vi điều khiển thông qua data bus.

Chúng ta đã tìm hiểu các bits đồng bộ và parity bit and data bits. Tuy nhiên, có 1 yếu tố quan trọng trong quá trình truyền nhận dữ liệu UART là Baud Rate.

Baud Rate: là tốc độ dùng để truyền và nhận dữ liệu. Cá thiết bị truyền và thiết bị nhận cần giao tiếp trên cùng một tốc độ để quá trình truyền nhận được hoàn thành.

Baud Rate được đo bởi số bit / giây (bits per second). Một số baud rates tiêu chuẩn là: 4800 bps, 9600 bps, 19200 bps, 115200 bps, Trong đó, tốc độ 9600 bps là tốc độ thường được sử dụng nhiều nhất.



Hình 2.13: Sơ đồ biểu diễn truyền data

2.2.2.2 Đặc điểm

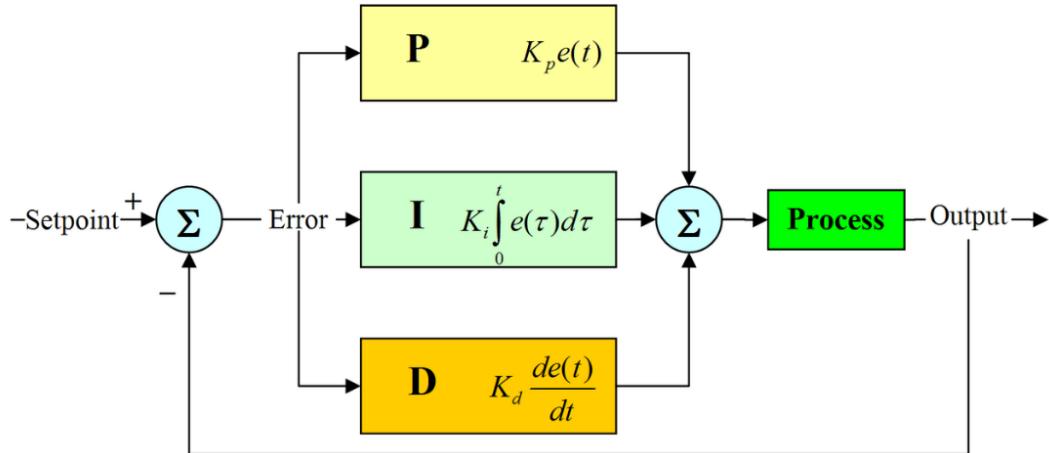
- + Đặc điểm thiết yếu:
- Chỉ cần hai dây để truyền nhận song song dữ liệu
- Không cần tín hiệu clock hay bất kỳ tín hiệu đồng bộ nào khác
- Parity bit đảm bảo dữ liệu được truyền đi chính xác
- + Tuy nhiên vẫn còn bị hạn chế bởi :
- Kích thước gói dữ liệu bị giới hạn
- Tốc độ truyền chậm hơn khi so sánh với kiểu truyền dữ liệu song song
- Thiết bị truyền và thiết bị nhận cần phải đồng nhất một số thông số với nhau

2.2.3 Bộ điều khiển PID

2.2.3.1 Lý thuyết về bộ điều khiển PID

PID được coi là là bộ điều khiển lý tưởng của các hệ thống điều khiển quy trình hiện đại. Nó được sử dụng hầu hết trong các ứng dụng điều khiển quá trình tự động

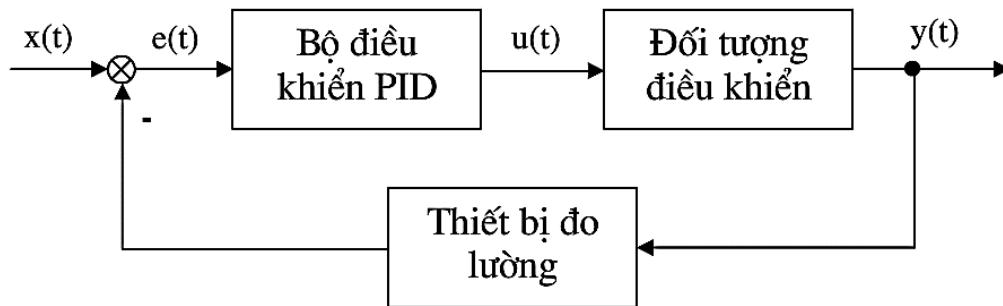
trong công nghiệp hiện nay. Để điều chỉnh lưu lượng, nhiệt độ, áp suất, vv... Giảm sai số xác lập đến mức tối thiểu nhất, hạn chế độ dao động, giảm thời gian xác lập và độ vọt lố.



Hình 2.14: Sơ đồ khái bô điều khiển PID

Bộ điều khiển PID được sử dụng khá rộng rãi để điều khiển đối tượng SISO (một vào, một ra) theo nguyên lý hồi tiếp. Lý do mà bộ điều khiển PID được sử dụng rộng rãi là vì tính đơn giản của nó cả về cấu trúc lẫn nguyên lý làm việc. Bộ điều khiển PID có nhiệm vụ đưa sai lệch $e(t)$ của hệ thống về 0 sao cho quá trình quá độ thỏa mãn các yêu cầu cơ bản về chất lượng. Nguyên lý làm việc của bộ điều khiển PID được mô tả một cách định tính như sau:

- Nếu sai lệch $e(t)$ càng lớn thì thông qua thành phần $u_P(t)$ tín hiệu điều chỉnh $u(t)$ càng lớn (vai trò của khâu khuếch đại tỉ lệ K_P).
- Nếu sai lệch $e(t)$ chưa bằng 0 thì thông qua thành phần $u_I(t)$ bộ điều khiển PID vẫn còn tạo tín hiệu điều chỉnh $u(t)$ (vai trò của khâu tích phân T_I)
- Nếu sự thay đổi của sai lệch $e(t)$ càng lớn thì thông qua thành phần $u_D(t)$ bộ điều khiển PID sẽ phản ứng càng nhanh (vai trò của khâu vi phân T_D)



Hình 2.15: Mô hình hệ thống điều khiển với PID

2.2.3.2 Các phương pháp chỉnh định tham số bộ điều khiển PID

Do từng thành phần của bộ điều khiển PID có những ưu nhược điểm khác nhau, và không thể đồng thời đạt được tất cả các chỉ tiêu chất lượng một cách tối ưu, nên cần lựa chọn, thỏa hiệp giữa các yêu cầu chất lượng và mục đích điều khiển. Việc lựa chọn tham số cho bộ điều khiển PID cũng phụ thuộc vào đối tượng điều khiển và các phương pháp xác định thông số.

+ Phương pháp chỉnh định bằng tay

- Đặt $K_I = K_D = 0$. Tăng K_P đến khi hệ thống dao động tuần hoàn.
- Đặt thời gian tích phân bằng chu kỳ dao động.
- Điều chỉnh lại giá trị K_P cho phù hợp.
- Nếu có dao động thì điều chỉnh giá trị K_D .

+ Phương pháp Ziegler-Nichols

- Đặt $K_I = K_D = 0$. Tăng K_P đến khi hệ thống dao động tuần hoàn.
- Đặt giá trị K_P này = K_C
- Đo chu kỳ dao động P_C

Bảng 2.1: Thông số chỉnh định PID bằng phương pháp Ziegler-Nichols

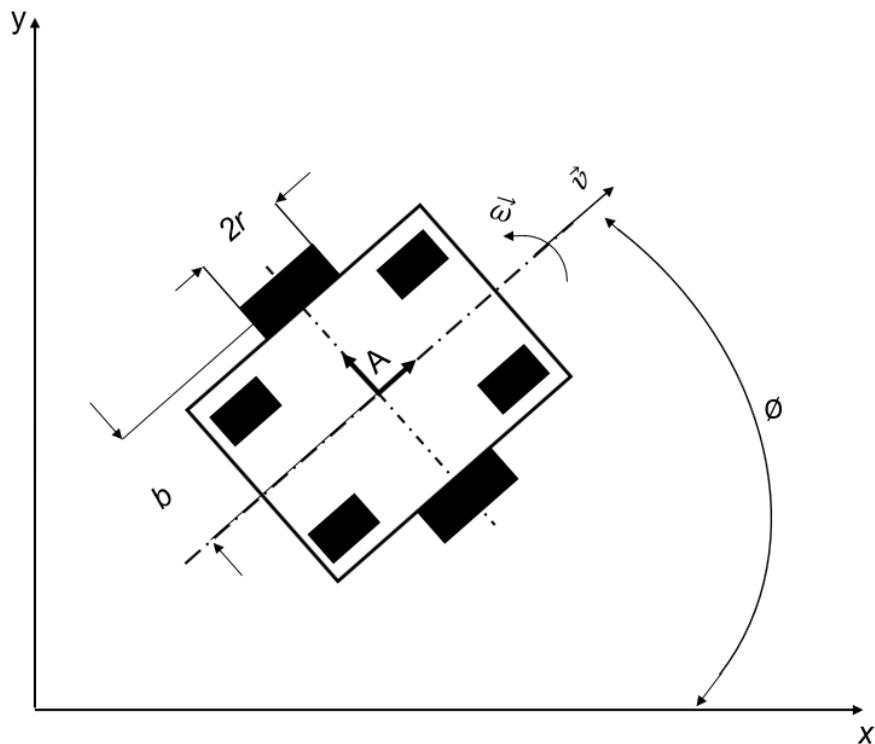
Thông số Bộ ĐK	K_P	K_I	K_D
P	$0.5K_C$	-	-
PI	$0.45K_C$	$1.2K_P/P_C$	-
PID	$0.6K_C$	$2K_P/P_C$	$K_P P_C / 8$

+ Phương pháp chỉnh định bằng phần mềm

- Dùng phần mềm để tự động chỉnh định thông số PID (thực hiện trên mô hình toán, kiểm nghiệm trên mô hình thực).
- Ví dụ dùng giải thuật di truyền (GA) để tìm thông số sao cho sai số đo được nhỏ hơn giá trị yêu cầu.

CHƯƠNG 3 : THIẾT KẾ VÀ XÂY DỰNG ROBOT

3.1 Phân tích động học mobile robot



Hình 3.1: Mô hình hóa Mobile robot

Trong đó:

$A(x,y)$: tọa độ giao điểm của 2 bánh xe với trục robot.

ϕ : góc tạo bởi trục robot với trục x.

v : tốc độ dài.

ω : tốc độ góc.

ω_r : tốc độ góc bánh phải.

ω_l : tốc độ góc bánh trái.

b : khoảng cách giữa trục robot với bánh xe.

r : bán kính của bánh xe.

Giả sử các bánh xe lăn không trượt. Phương trình động học của mobile robot:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.1)$$

Mỗi quan hệ giữa v , ω với tốc độ góc của 2 bánh xe chủ động là:

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} 1/r & b/r \\ 1/r & -b/r \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.2)$$

Từ (3.2) ta có:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ r/2b & -r/2b \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (3.3)$$

Từ (3.1), (3.3) trên ta được:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} &= \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r/2 & r/2 \\ r/2b & -r/2b \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \\ \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} &= \begin{bmatrix} (rcos\phi)/2 & (rcos\phi)/2 \\ (rsin\phi)/2 & (rsin\phi)/2 \\ r/2b & -r/2b \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \end{aligned} \quad (3.4)$$

Các chuyển động cơ bản cần thực hiện của mobile robot gồm chuyển động thẳng và chuyển động xoay tại chỗ.

Để mobile robot di chuyển thẳng, góc ϕ tạo với trục robot với trục x không đổi, suy ra $\dot{\phi} = 0$. Từ phương trình động học mobile robot trên ta được:

$$\dot{\phi} = \frac{r(\omega_r - \omega_l)}{2b}$$

Suy ra, để robot di chuyển thẳng ta phải điều khiển sao cho $\omega_r = \omega_l$, có nghĩa là hai bánh trái và bánh phải có cùng tốc độ góc, tùy thuộc vào tốc độ di chuyển mong muốn mà chọn giá trị tốc độ góc bánh trái và bánh phải cho hợp lý với $v = \omega_r = \omega_l$.

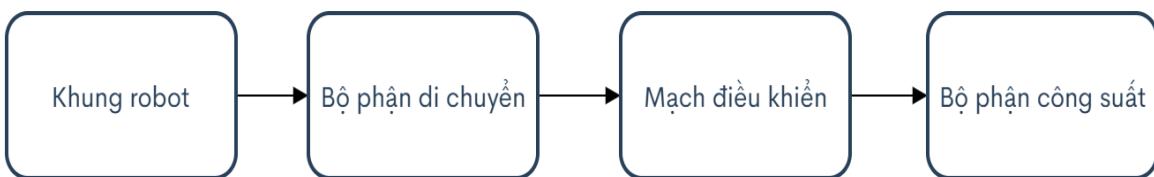
Để thực hiện chuyển động xoay tại chỗ quanh tâm A, Tọa độ A(x,y) không đổi, tức là $\dot{x} = 0, \dot{y} = 0$.

$$\begin{aligned} \dot{x} &= \frac{(\omega_r + \omega_l)(rcos\phi)}{2} \\ \dot{y} &= \frac{(\omega_r + \omega_l)(rsin\phi)}{2} \end{aligned}$$

Do đó để thực hiện chuyển động xoay tại chỗ $\omega_r = -\omega_l$, tức là hai bánh trái và phải xoay cùng tốc độ, ngược chiều nhau. Tốc độ góc của robot trong trường hợp này là $\omega = \omega_r r/b$.

Để kiểm tra sự chính xác phương trình trên được mô phỏng trên phần mềm Matlab Simulink.

3.1.1 Cơ sở thiết kế



Hình 3.2: Sơ đồ khái niệm mô hình

Mobile robot được thiết kế trong luận văn gồm có các bộ phận chính như sau:

Bộ phận khung, vỏ: Phần khung, vỏ có tác dụng liên kết các bộ phận khác của robot. Đồng thời, nó cũng có tác dụng trong việc tăng độ cứng vững, bảo vệ robot khỏi các tác động bên ngoài.

Bộ phận truyền động: Mobile robot cần được truyền động bằng hai bánh chủ động và bốn bánh dẫn hướng. Các bánh chủ động được điều khiển bởi hai động cơ DC M555.

Bộ phận mạch điều khiển: Module điều khiển trung tâm có vai trò điều khiển hoạt động của mobile robot thông qua xử lý tín hiệu từ cảm biến encoder, điều khiển động cơ qua L298N và giao tiếp với máy tính qua cổng COM. Board Arduino Mega 2560 được sử dụng làm Master làm nhiệm vụ nhận tín hiệu từ máy tính và hai Arduino Uno làm Slave nhận tín hiệu từ Master để điều khiển động cơ.

Bộ phận công suất: Bộ phận công suất có tác dụng cung cấp năng lượng cho robot hoạt động. Bộ phận này phải cung cấp công suất đủ lớn cho toàn bộ hệ thống. Do đó, mobile robot được sử dụng pin Lipo Lithium làm nguồn cung cấp năng lượng cho toàn hệ thống.

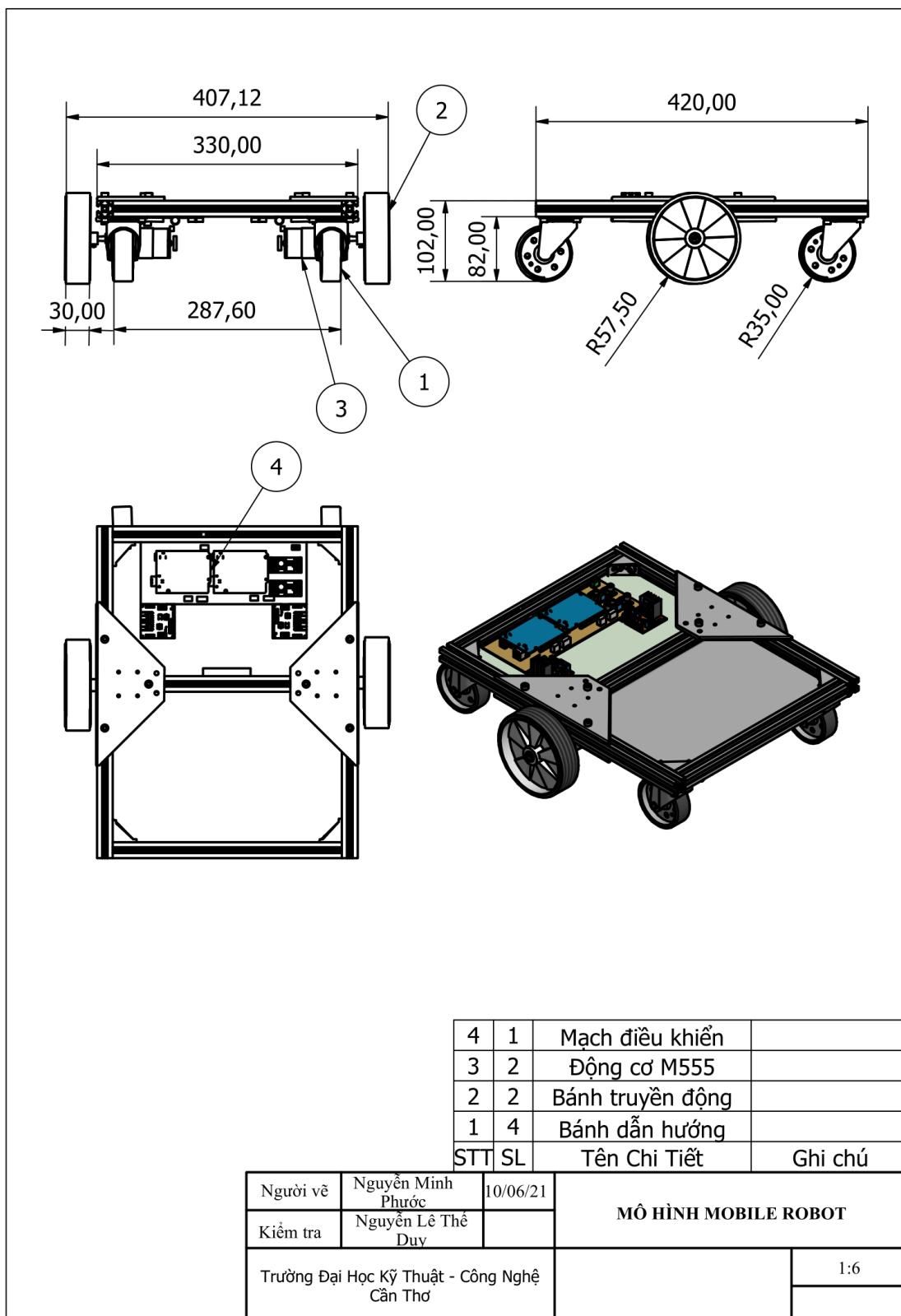
3.2 Thiết kế mô hình

3.2.1 Bản thiết kế mô hình trên Autodesk Inventor

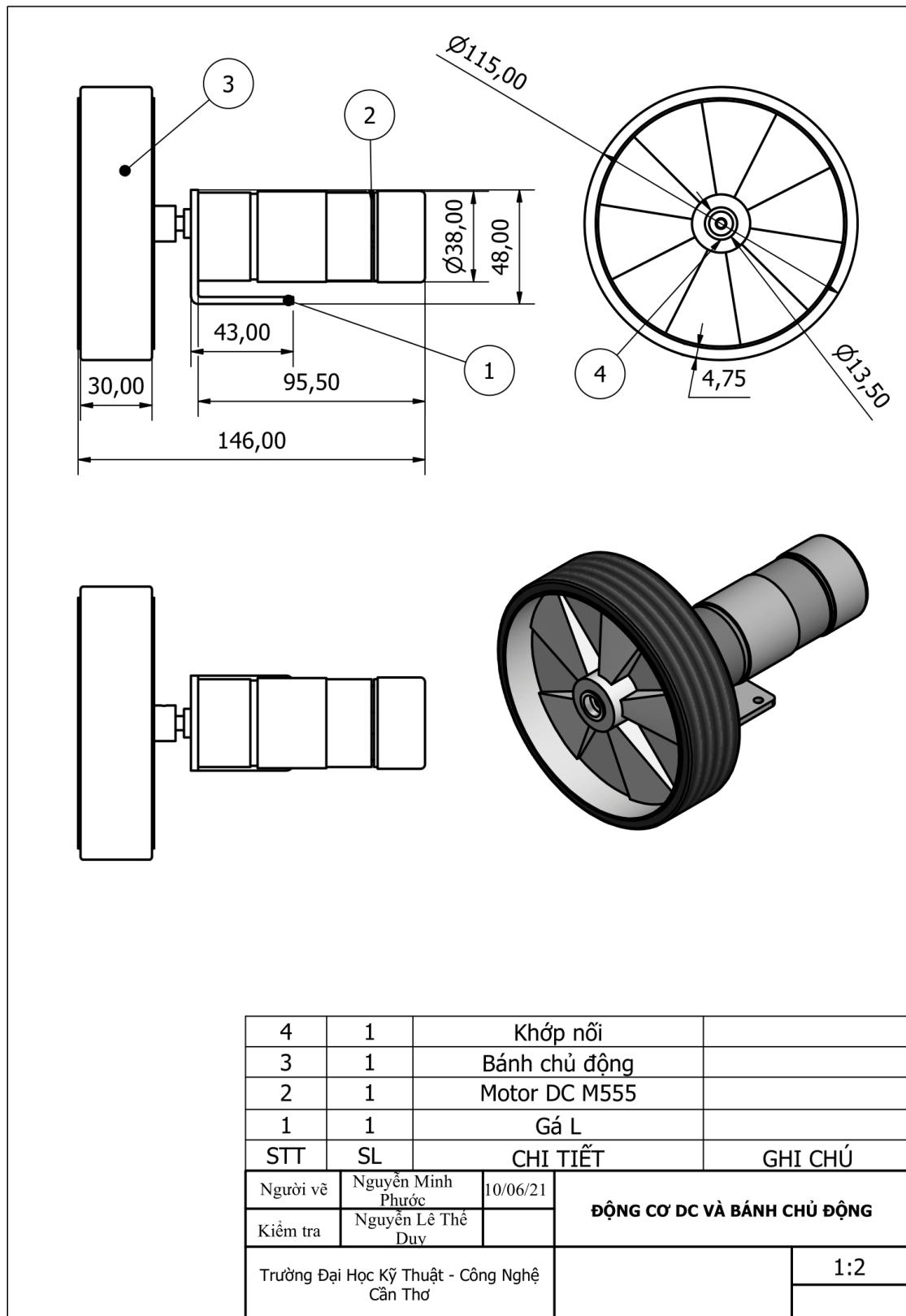
Bản vẽ tổng thể mô hình mobile robot trên Autodesk Inventor:

Khung robot được liên kết với nhau bởi các ke góc L và được cố định bằng ốc lục giác và con trượt. Các bánh caster còn lại đặt ở 4 góc khung và cũng được cố định như khung robot (Xem hình 3.2).

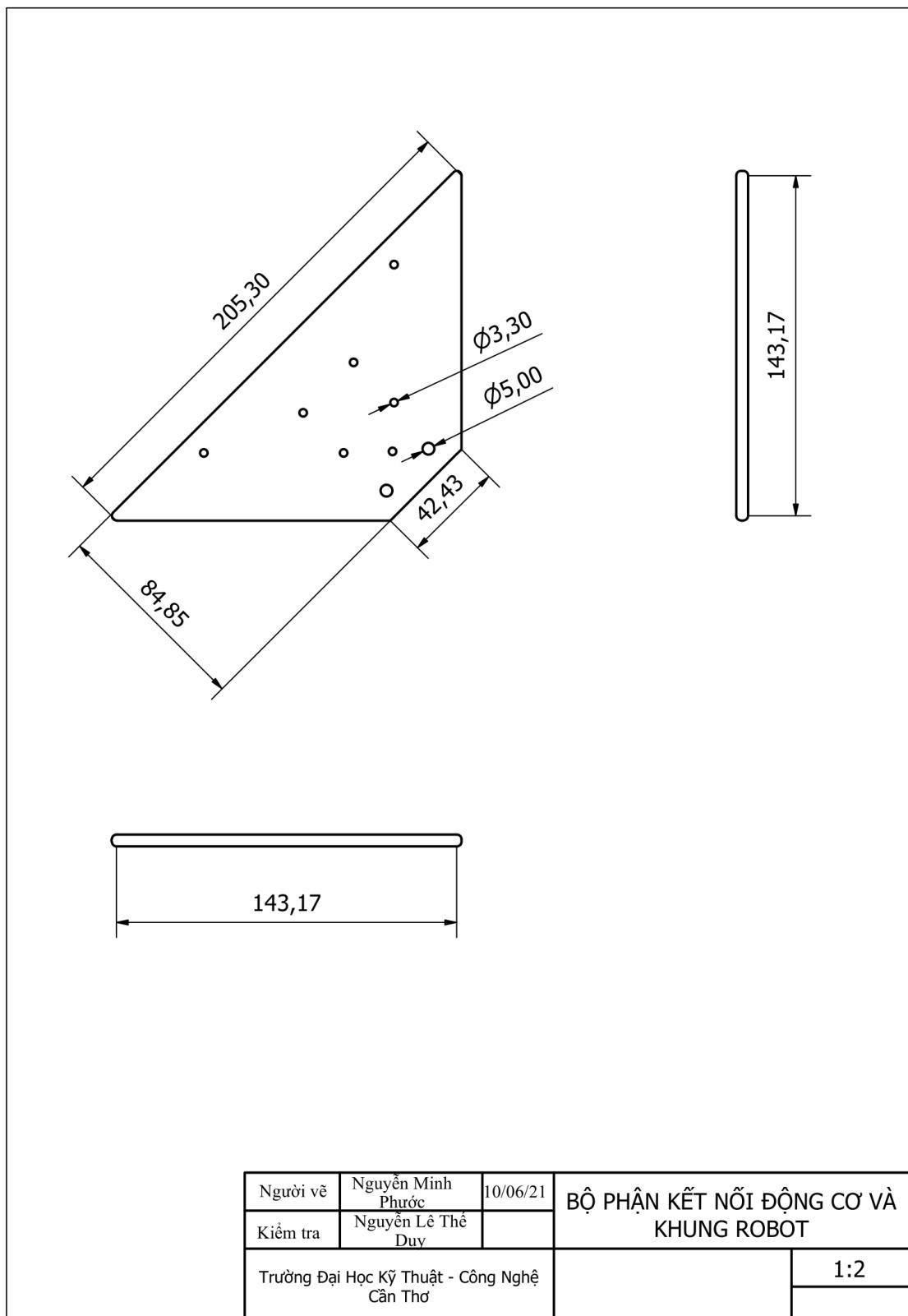
Động cơ DC được liên kết với bánh chủ động bằng khớp nối qua trục động cơ, động cơ được cố định với khung robot bằng gá L (Xem hình 3.3).



Hình 3.3: Bản vẽ tổng thể mobile robot



Hình 3.4:Bản vẽ liên kết khâu chuyển động

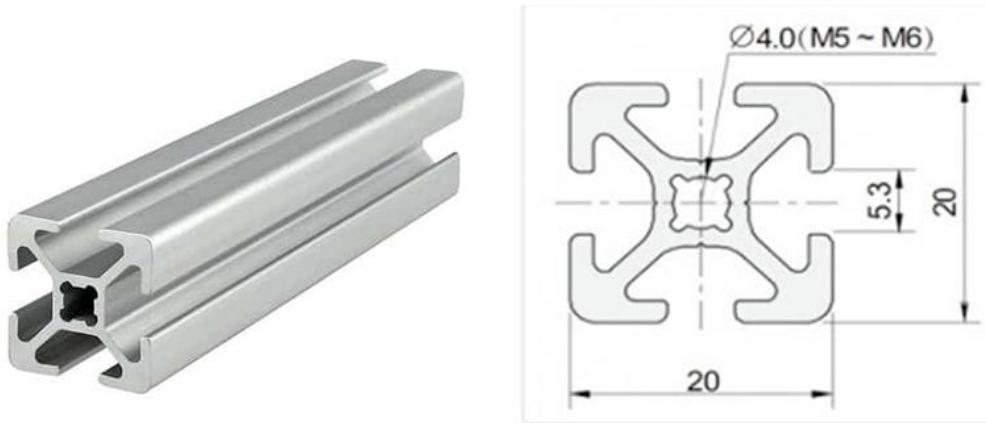


Hình 3.5: Bệ đỡ động cơ với khung robot

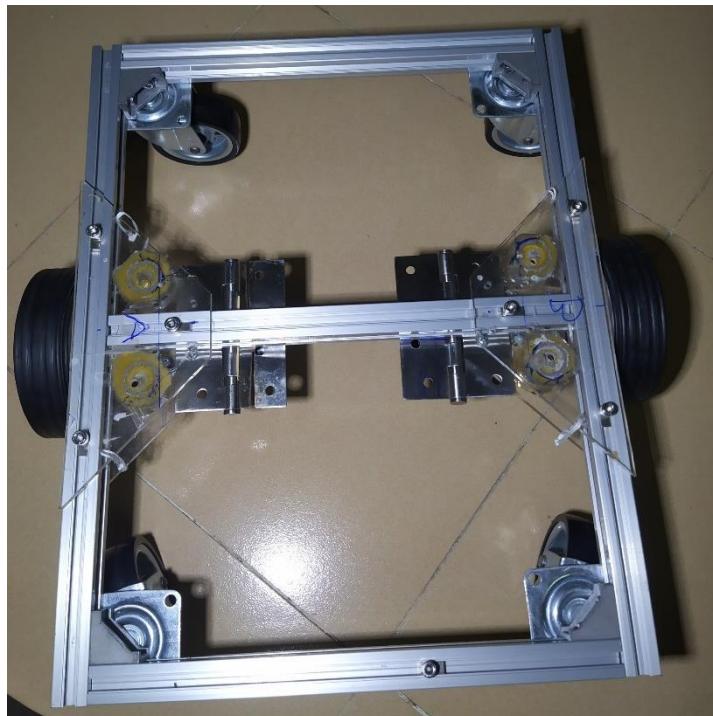
3.2.2 Thiết kế cơ khí

3.2.2.1 Thiết kế khung xe

Phần khung xe là nơi lắp ráp các chi tiết của robot. Yêu cầu cơ bản của khung là độ cứng vững, độ chính xác về vị trí tương đối của các bề mặt lắp ghép và khả năng chịu tải tốt. Chính vì thế khung xe được chế tạo bằng nhôm định hình, các thanh nhôm được lắp với nhau bằng các bulong và ke góc. Nhôm định hình có khối lượng nhẹ, các gân tăng cứng nên khả năng chịu tải đảm bảo. Nhôm được chọn cho thiết kế là nhôm định hình 20x20cm.



Hình 3.6: Nhôm định hình 20x20



Hình 3.7: Mô hình xây dựng thực tế

3.2.2.2 Động cơ di chuyển

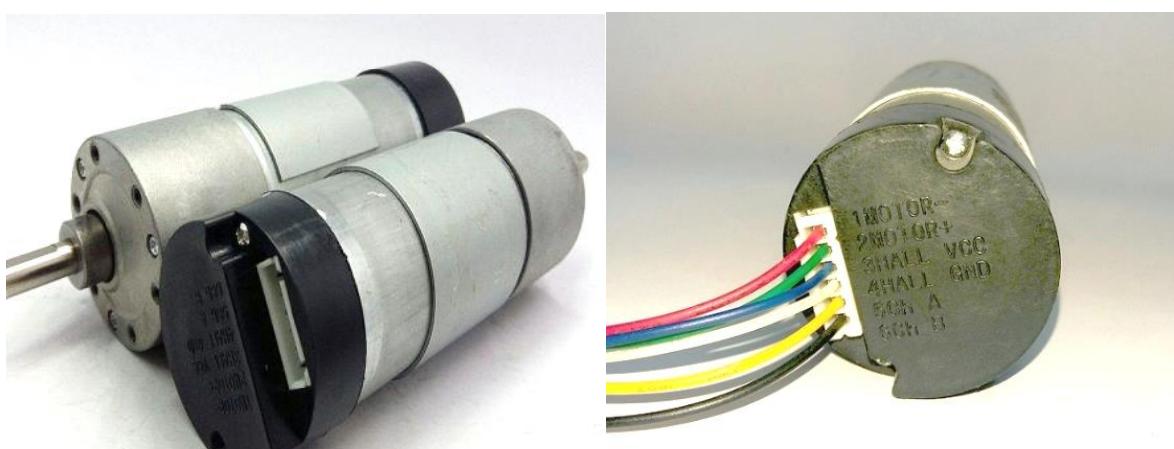
Động cơ điện ảnh hưởng trực tiếp đến khả năng làm việc, độ tin cậy và tính an toàn của robot. Việc lựa chọn cần đảm bảo để robot hoạt động đạt công suất yêu cầu, đảm bảo khả năng di chuyển của robot, tiết kiệm điện năng cũng như tăng tuổi thọ cho robot.

Theo yêu cầu thiết kế, mobile robot không yêu cầu về tốc độ di chuyển. Tuy nhiên cần lựa chọn động cơ momen lớn để chịu tải và giúp di chuyển không bị rung lắc ảnh hưởng đến chất lượng ảnh khi di chuyển.

Từ những yêu cầu kỹ thuật trên cùng những yếu tố khác như: kích thước, giá thành, khối lượng... Tôi đã chọn động cơ DC M555 cho robot di chuyển.

❖ Thông số kỹ thuật của động cơ M555

- Điện áp vận hành: 12 VDC
- Công suất: 17W
- Khối lượng: 350 gam
- Tốc độ động cơ: 6000 (vòng/phút)
- Tốc độ qua hộp số: 33 (vòng/phút)
- Bộ giảm tốc hộp số: 180:1
- Encoded: Hall sensor 12 xung, 2 kênh AB, Nguồn 3~5VDC



Hình 3.8: Động cơ DC M555

❖ Sơ đồ bô trí dây:

1. Motor DC -

2. Motor DC +
3. Vcc encoder: 3~5v
4. Gnd encoder
5. Kênh A
6. Kênh B

3.2.2.3 Bánh xe di chuyển được sử dụng

Robot dùng bốn bánh dẫn hướng caster, và hai bánh truyền động được đặt giữa khung robot, mục đích bốn bánh dẫn hướng giúp trọng lực phân bố đều cho độ ổn định khi di chuyển, hiệu suất truyền động tốt, bán kính khi quay nhỏ giúp di chuyển linh hoạt. Do cần tải trọng phù hợp nên bánh truyền động được chọn là bánh nhựa với đường kính 115mm và bánh caster với đường kính 70mm.



Hình 3.9: Bánh chủ động(a) và bánh caster(b)

3.3 Thiết kế mạch điện điều khiển

3.3.1 Sơ đồ mạch điều khiển robot

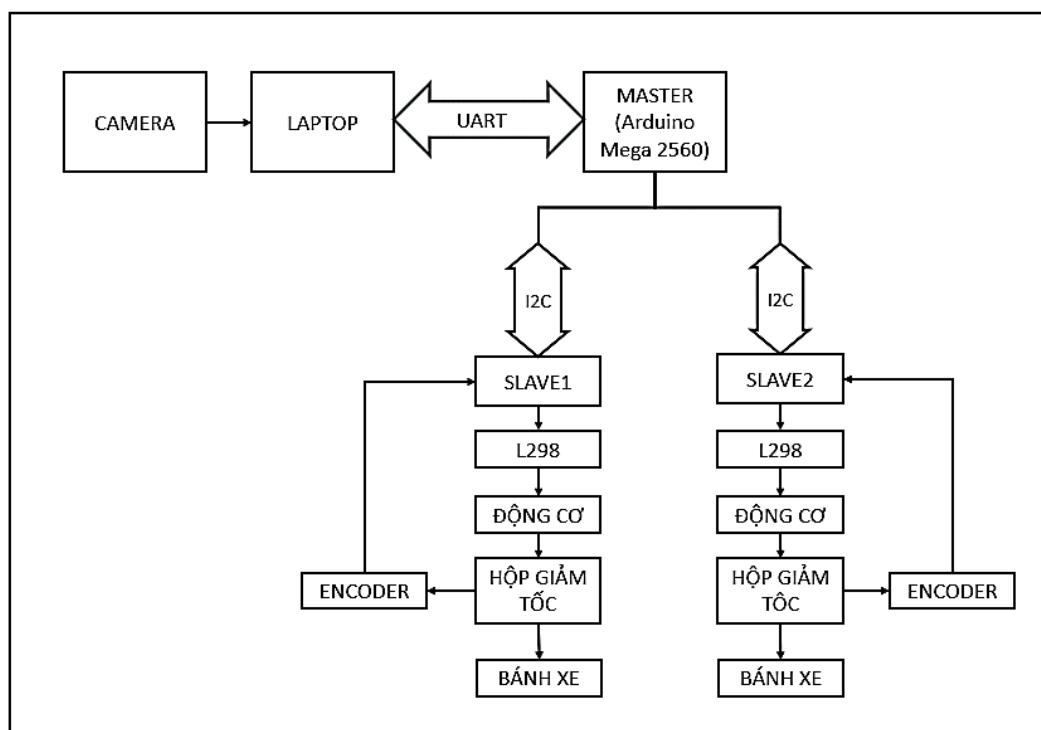
Sơ đồ khối mạch điều khiển được thiết kế như sau:

Giải thích sơ đồ mạch điều khiển (Xem hình 3.10) :

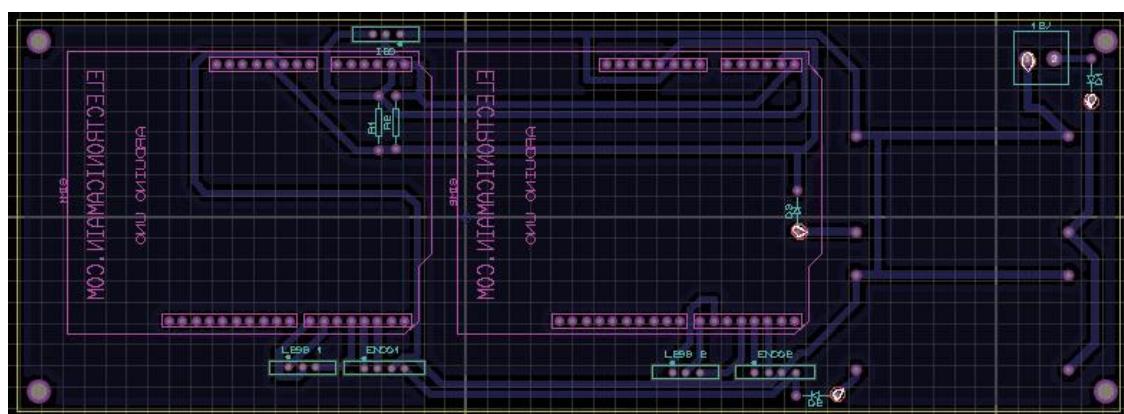
Đầu tiên ảnh được nhận từ đầu vào camera, sau khi thu nhận hình ảnh, ảnh sẽ được xử lý thông qua chương trình phân tích, tính toán trong chương trình được xây dựng trong laptop, sau khi xử lý ảnh, hệ thống truyền tín hiệu xuống master qua giao tiếp uart, khi master nhận được tín hiệu, master sẽ ra lệnh xuống slave qua giao tiếp

I2C, slave sau khi nhận lệnh từ master sẽ trực tiếp điều khiển động cơ qua hộp giảm tốc cuối cùng là bánh xe. Slave sẽ nhận phản hồi từ encoder trả về khi động cơ chuyển động thuận hoặc nghịch từ đó đưa điều khiển động cơ phù hợp.

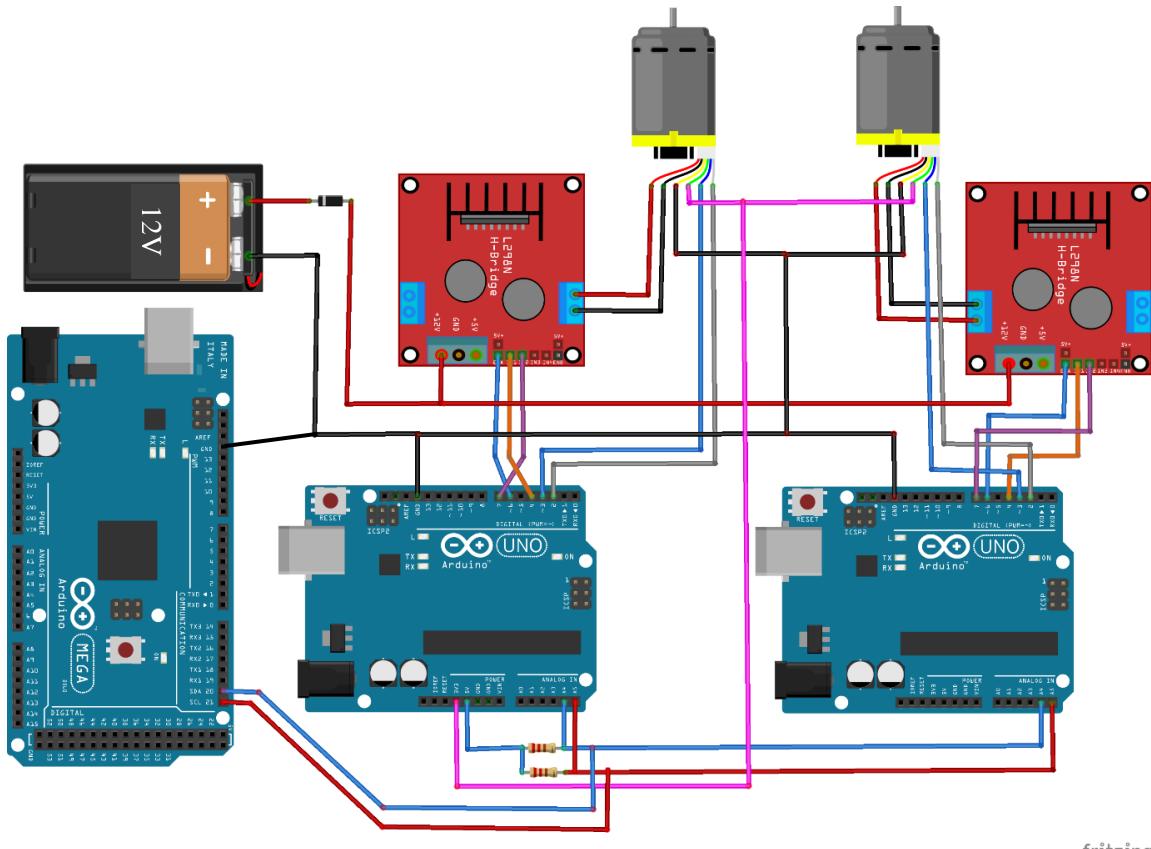
Robot cần có một mạch điện thành một hệ thống thống nhất, mạch lạc và ổn định do đó mạch điện robot cần đảm bảo yêu cầu sau: Đảm bảo đủ công suất cho robot hoạt động, ổn định tiết kiệm và tính linh hoạt cao trong việc thay thế sử dụng. Để làm được điều đó tôi đã dùng Proteus trong việc thiết kế đi dây cho mạch điều khiển và phần mềm Fritzing thực hiện sơ đồ đấu nối dây.



Hình 3.10: Sơ đồ khái niệm mạch điều khiển



Hình 3.11: Sơ đồ Layout mạch điều khiển



fritzing

Hình 3.12: Sơ đồ đấu nối dây

Mạch điều khiển sau khi gia công:



Hình 3.13: Mạch điều khiển sau khi gia công

3.3.1.1 Arduino Mega2560 (Master)

Trong đề tài, Arduino Mega 2560 được chọn làm trung tâm quản lý điều khiển. Board Arduino Mega2560(Master) nhận tín hiệu điều khiển từ Python thông qua giao tiếp cổng COM và gửi dữ liệu xuống cho các slave điều khiển động cơ.

Arduino Mega2560 sẽ thực hiện các nhiệm chính sau:

- Giao tiếp và nhận tín hiệu từ Python.
- Giao tiếp và truyền lệnh với hai slave Arduino Uno R3.

Board có các đặc tính kỹ thuật để đáp ứng cho mục đích luận văn như sau:

❖ **Thông số kỹ thuật:**

- Vi điều khiển : ATmega 2560.
- Điện áp hoạt động : 5V.
- Nguồn ngoài (giắc tròn DC) 7-9V. Không nên cấp nguồn 12v vì sẽ gây hỏng IC ổn áp.
- Số chân Digital: 54 (15 chân PWM).
- Số chân Analog: 16.
- Giao tiếp UART: 4 bộ Serial.
- Giao tiếp SPI: 1 bộ (chân 50 đến chân 53) dùng với thư viện SPI của Arduino.
- Giao tiếp I2C: 1 bộ.
- Ngắt ngoài : 6 chân.
- Bộ nhớ Flash : 256 Kb, 8 Kb sử dụng cho Bootloader.
- SRAM: 8 Kb.
- Xung clock: 16MHz.

❖ **Chức năng của các chân Arduino Mega 2560**

- Chân Vin: Đây là điện áp đầu vào được cung cấp cho board mạch Arduino Uno. Khác với chân 5V được cung cấp qua cổng USB. Chân này được sử dụng để cung cấp điện áp toàn mạch thông qua jack nguồn, thông thường khoảng từ 7 đến 12VDC.
 - Chân 5V: Chân này được sử dụng để cung cấp điện áp đầu ra. Arduino Uno được cấp nguồn bằng ba cách đó là USB, chân Vin của board mạch hoặc jack nguồn.

- USB: Dùng để giao tiếp với máy tính thông qua cáp USB chúng ta có thể nạp chương trình cho Arduino hoạt động. Ngoài ra, USB còn là nguồn cho Arduino hỗ trợ điện áp khoảng 5 VDC trong khi Vin và jack nguồn hỗ trợ điện áp trong khoảng từ 7V đến 20VDC.

- Chân GND: Chân mass chung cho toàn mạch Arduino.
- Chân điều khiển:

RESET: Arduino Mega 2560 có sẵn mạch reset với nút ấn để thiết lập lại hệ thống và chân này có thể được sử dụng khi kết nối các thiết bị khác để thiết lập lại bộ điều khiển.

XTAL1, XTAL2: Thạch anh(16Mhz) được kết nối với xung clock cung cấp cho bộ điều khiển.

AREF: Chân này được dùng khi sử dụng ADC để chuyển đổi tín hiệu với điện áp tham chiếu bên ngoài mà không muốn sử dụng điện áp tham chiếu nội bộ 1.1V hoặc 5V.

- Các chân Digital (70): Chân số: Từ 0-53 (số) và 0-15 (tương tự) có thể được sử dụng làm đầu vào hoặc đầu ra cho thiết bị.

Chân tương tự (16):

Từ 0-15 (analog) có thể được sử dụng như chân đầu vào tương tự cho bộ ADC, nếu không sử dụng nó hoạt động như chân digital bình thường.

❖ **Chân có chức năng thay thế:**

Chân SPI: Chân 22-SS, 23_SCK, 24-MOSI, 25-MISO

Các chân này được sử dụng cho giao tiếp nối tiếp với giao thức SPI để liên lạc giữa 2 thiết bị trao đổi. SPI cho phép bit phải được thiết lập để bắt đầu giao tiếp với các thiết bị khác.

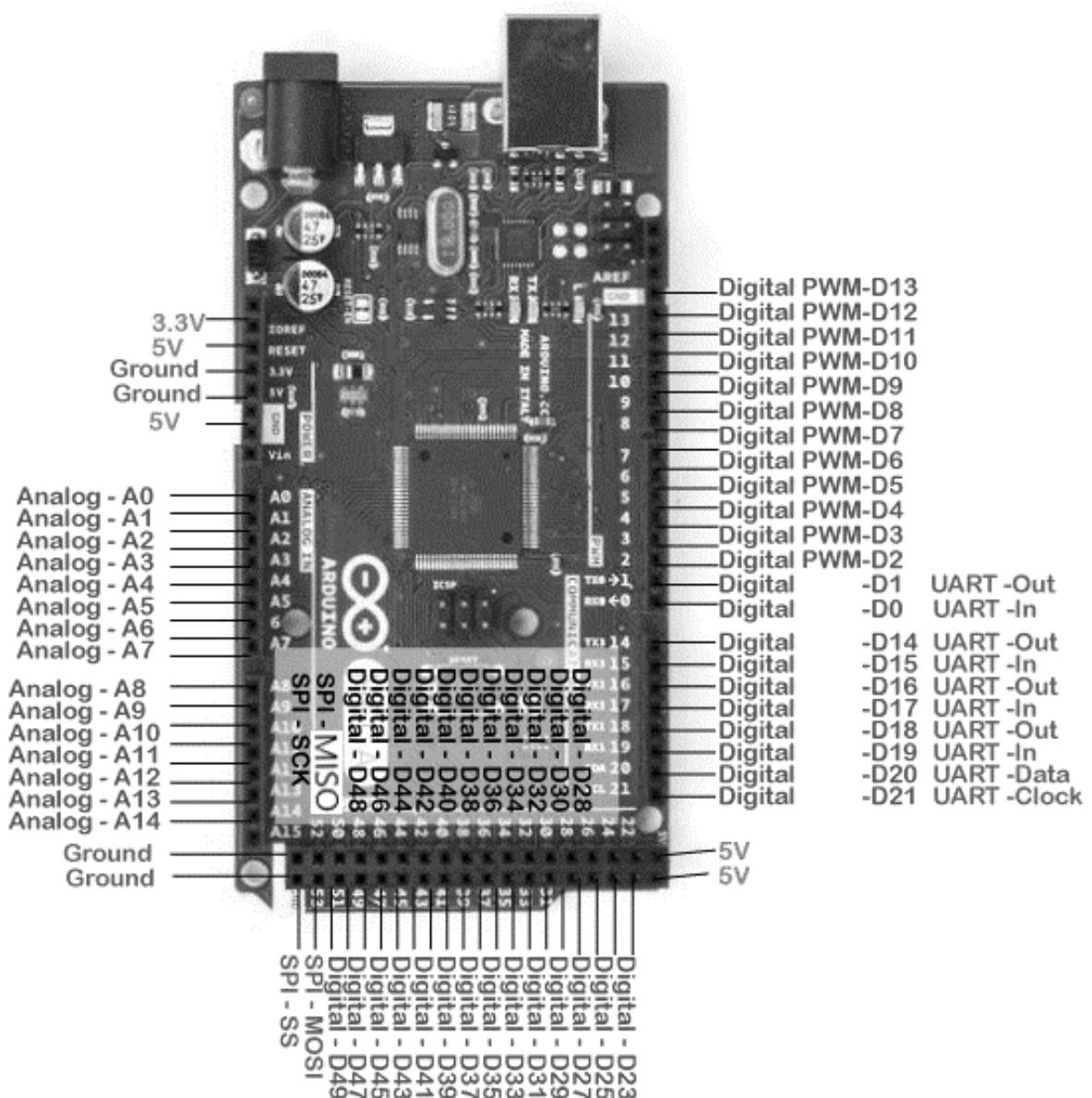
Chân I2C: Chân 20 cho SDA và 21 cho SCK (Tốc độ 400khz) để cho phép liên lạc hai dây với các thiết bị khác.

Chân PWM : Chân 2-13 có thể được sử dụng như đầu ra PWM với hàm analogWrite () để ghi giá trị PWM từ 0-255.

Chân UART: Chân này được sử dụng cho giao tiếp nối tiếp giữa bo mạch với máy tính hoặc hệ thống khác để chia sẻ và ghi dữ liệu.

❖ Chân ngắt :

- Chân digital: 0, 22, 23, 24, 25, 10, 11 ,12 ,13 ,15 ,14.
- Chân analog: 6 ,7 ,8 ,9, 10, 11, 12, 13 ,14, 15.
- Chân này được sử dụng để ngắt. Để kích hoạt chân ngắt phải cài đặt bật ngắt toàn cục.
- Chân ngắt phần cứng: Chân 18 - 21,2,3 ngắt phần cứng được sử dụng cho các ứng dụng ngắt. Ngắt phần cứng phải được bật với tính năng ngắt toàn cục để ngắt quãng từ các thiết bị khác.



Hình 3.14: Sơ đồ chân Arduino Mega 2560

3.3.1.2 Arduino Uno R3 (Slave)

Trong đề tài, Arduino Uno R3 được dùng làm slave nhận lệnh từ master để điều khiển động cơ thông qua giao tiếp I2C.

Arduino Uno thực hiện các nhiệm vụ sau:

- Nhận lệnh từ master.
- Đọc tín hiệu phản hồi từ encoder.
- Xuất tín hiệu ra module L298 để điều khiển động cơ.

Board có các đặc tính kỹ thuật để đáp ứng cho mục đích luận văn như sau:

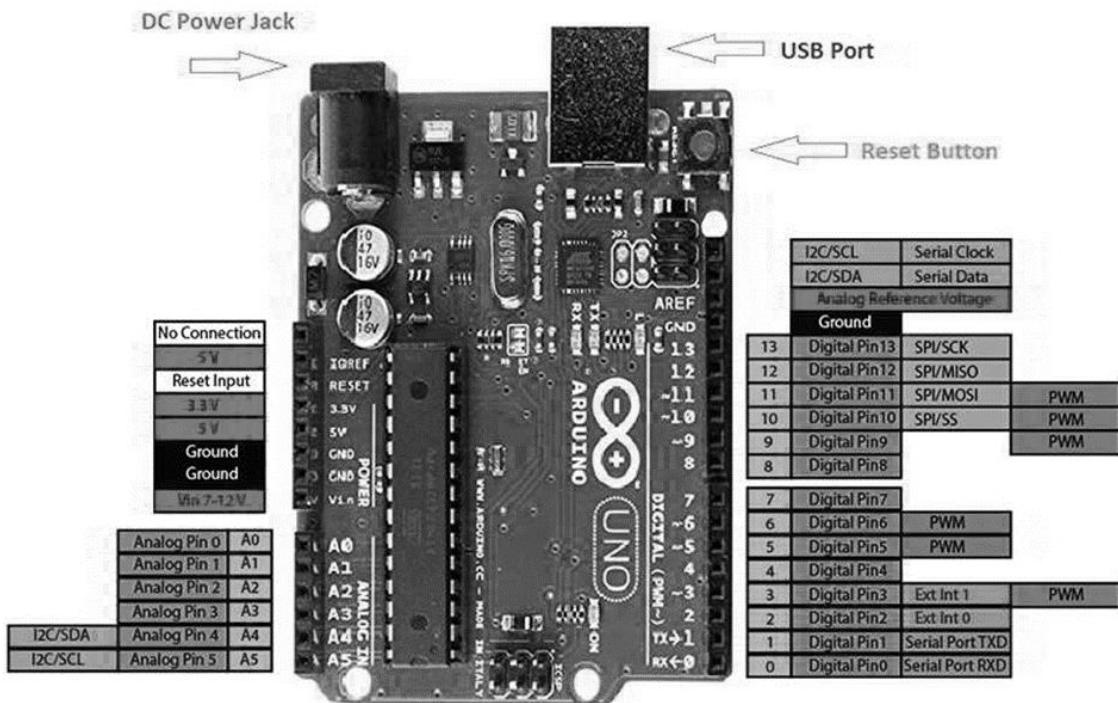
❖ **Thông số kỹ thuật:**

- Chip: ATmega328P.
- Nguồn cấp: từ 7 đến 12 VDC.
- Dòng max chân 5V: 500 mA.
- Dòng max chân 3.3V: 50 mA.
- Dòng max chân I/O: 30 mA.
- Số chân Digital: 14 chân (Bao 6 chân PWM).
- Số chân Analog: 6 chân.
- Flash Memory: 32 kb.
- SRAM: 2 kb.
- EEPROM: 1 kb.
- Clock Speed: 16 Mhz.
- Kích thước: 68.6 x 53.4mm.
- Khối lượng: 25 gam.

❖ **Chức năng của các chân Arduino Uno:**

- Chân Vin: Đây là điện áp đầu vào được cung cấp cho board mạch Arduino Uno. Khác với chân 5V được cung cấp qua cổng USB. Chân này được sử dụng để cung cấp điện áp toàn mạch thông qua jack nguồn, thông thường khoảng từ 7 đến 12VDC.
 - Chân 5V: Chân này được sử dụng để cung cấp điện áp đầu ra. Arduino Uno được cấp nguồn bằng ba cách đó là USB, chân Vin của board mạch hoặc jack nguồn.

- USB: Dùng để giao tiếp với máy tính thông qua cáp USB chúng ta có thể nạp chương trình cho Arduino hoạt động. Ngoài ra, USB còn là nguồn cho Arduino hỗ trợ điện áp khoảng 5VDC trong khi Vin và jack nguồn hỗ trợ điện áp trong khoảng từ 7V đến 20VDC.
- Chân GND: Chân mass chung cho toàn mạch Arduino.
- Chân Reset: Để thiết lập lại toàn bộ và đưa chương trình đang chạy trở về ban đầu. - Chân này rất hữu ích khi Arduino bị treo khi đang chạy chương trình.
- Các chân Digital: Có 14 chân Digital trên board mạch Arduino Uno được ký hiệu từ 0 đến 13 với chức năng input và output sử dụng các hàm pinMode(), digitalWrite(), digitalRead() để điều khiển các chân này.
- Chân PWM: Được cung cấp bởi các chân 3,5,6,9,10,11. Các chân này được cấu hình để cung cấp PWM đầu ra 8 bit.
- Các chân Analog: Có 6 chân Analog trên board mạch Arduino Uno được ký hiệu từ A0 đến A5. Được sử dụng để đo điện áp tương tự trong khoảng từ 0 đến 5V.
- Chân Rx, Tx: Dùng để thực hiện giao tiếp nối tiếp thông qua hai chân 0 (Rx) và chân 1 (Tx).
- Chân LED: Arduino Uno tích hợp đèn LED kết nối với chân 13. Cung cấp mức logic HIGH tương ứng ON và LOW tương ứng OFF.
- Chân AREF: Chân này được gọi là tham chiếu tương tự, được sử dụng để cung cấp điện áp tham chiếu cho các đầu vào tương tự.
- Chân SPI: Được dùng để giao tiếp ngoại vi nối tiếp. Các chân 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) cung cấp liên lạc SPI với sự trợ giúp của thư viện SPI.
- Chân ngắt ngoài: Chân 2 và 3 được sử dụng để cung cấp các ngắt ngoài.
- Chân I2C: Dùng để giao tiếp I2C với các thiết bị. Chân A4 (SDA) và A5 (SCL) được sử dụng cho mục đích này.



Hình 3.15: Sơ đồ chân Arduino Uno R3

3.3.1.3 Module L298N

Trong đề tài, Module L298N được dùng để làm module điều khiển trung gian giữa Slave và động cơ DC vì Module L298N được tích hợp mạch cầu H bên trong nên có thể điều khiển tốc độ và chiều quay của động cơ DC một cách dễ dàng. Mỗi module có khả năng điều khiển được hai động cơ DC với dòng tối đa cho mỗi động cơ là 2A. Ngoài ra, Module L298N còn tích hợp IC nguồn 78M05 để tạo ra nguồn 5 VDC để cấp cho các module khác.

❖ **Nguyên lý hoạt động:**

Mạch cầu H có hai loại, mạch cầu H dùng BJT và FET. Hai dạng mạch cầu H này có nguyên lý hoạt động giống như nhau, chỉ khác nhau là linh kiện cấu tạo nên mạch cầu H như thế nào và việc đóng mở Transistor hay FET.

Mạch cầu H điều khiển bởi bốn tín hiệu đóng và mở các van đó là các tín hiệu 1, tín hiệu 2. Mạch cầu H điều khiển được hai chiều quay của động cơ.

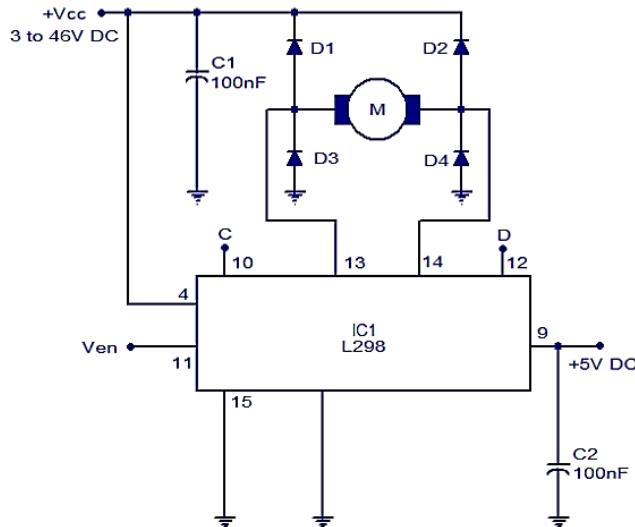
Khi ENA = 0: Động cơ quay với mọi đầu vào.

Khi ENA = 1:

- INT1 = 1; INT2 = 0: Động cơ quay thuận.

- INT1 = 0; INT2 = 1: Động cơ quay nghịch.
- INT1 = INT2: Động cơ dừng.

Với ENB cũng tương tự với INT3, INT4.

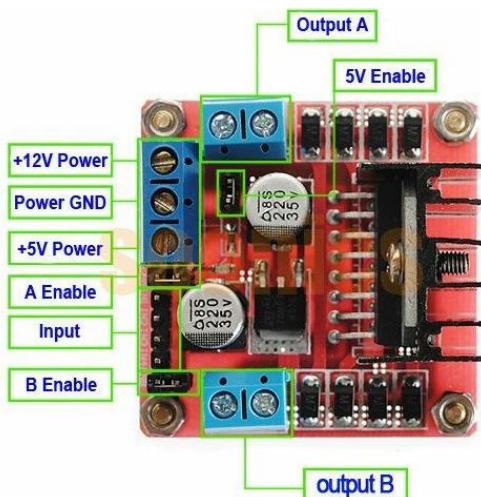


Hình 3.16: Sơ đồ nguyên lý mạch cầu H

❖ **Thông số kỹ thuật của Module L298N:**

- Điện áp điều khiển: từ 5 đến 12 Volt.
- Dòng tối đa cho mỗi cầu H là: 2A.
- Điện áp của tín hiệu điều khiển: từ 5 đến 7 Volt.
- Dòng của tín hiệu điều khiển: 0 ~ 36mA.
- Công suất hao phí: 20W (khi nhiệt độ T = 75°C).

❖ **Sơ đồ chân của Module L298N được thể hiện (xem hình 3.13):**



Hình 3.17: Sơ đồ chân Module L298N

3.3.1.4 Module LM2596



Hình 3.18: Module LM2596

❖ **Thông số kỹ thuật:**

- Điện áp đầu vào: Từ 3V đến 30V.
- Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 30V.
- Dòng đáp ứng tối đa là 3A.
- Hiệu suất: 92%
- Công suất: 15W

Module LM2596 trong hệ thống dùng để giảm áp từ nguồn pin, cấp nguồn cho Arduino và encoder động cơ hoạt động.

3.3.1.5 Pin Lipo Lithium Polymer

❖ **Thông số kỹ thuật:**

- Dung lượng :~10000mAh
- Điện áp định mức : 3.7 V
- Điện áp tối đa : 4.2 V
- Điện áp cắt: 2.5 V
- Công suất : 37Wh
- Dòng điện tối đa: 3C
- Nội trở : < 40 Ohm
- Nhiệt Độ xả: 20 ° C ~ 60 ° C
- Chu kỳ nạp xả: >1000 lần

- Kích thước: 18 x 65 x 131(mm)

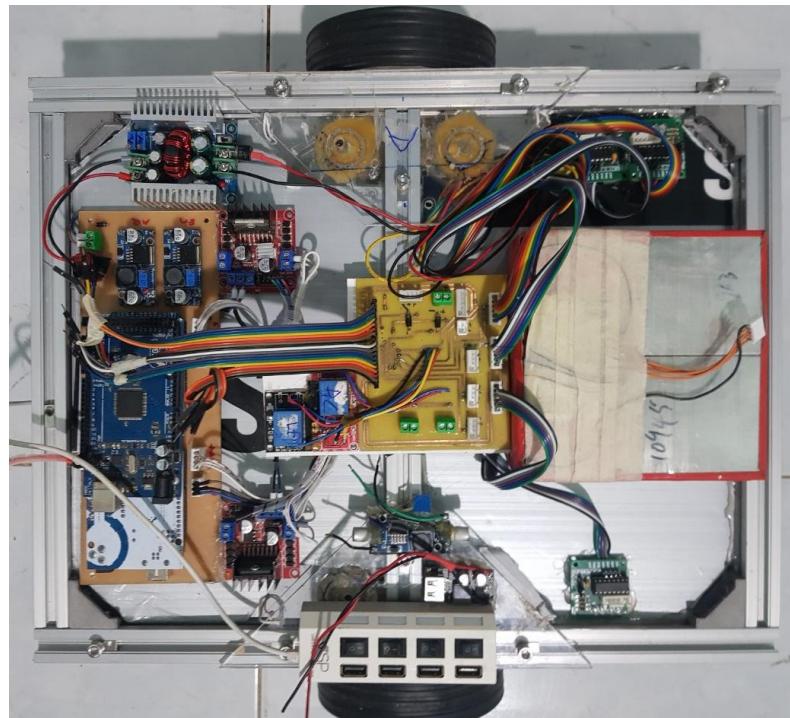
Robot được sử dụng bốn cell pin để cấp nguồn cho toàn bộ hệ thống bao gồm nguồn cho Arduino, cảm biến và động cơ.



Hình 3.19: Pin Lipo Lithium Polymer

3.4 Mô hình hoàn chỉnh

Mô hình được thiết kế hoàn chỉnh gồm bốn bánh điều hướng được bố trí ở bốn góc và hai bánh dẫn động ở giữa giúp robot linh hoạt hơn. Linh kiện được bố trí phù hợp để thuận tiện trong việc điều khiển và di chuyển của robot.



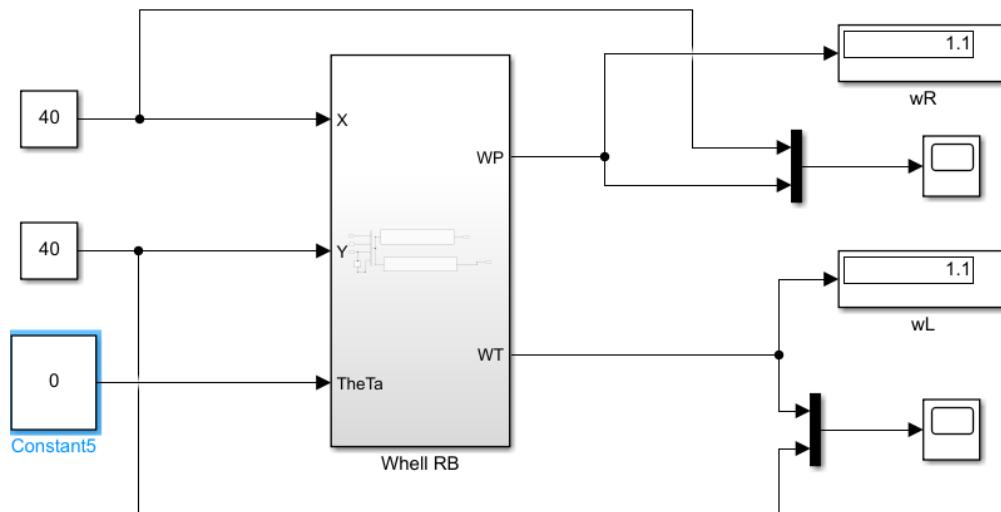
Hình 3.20 Mô hình mobile robot hoàn chỉnh

3.5 Mô phỏng và chạy thực nghiệm

3.5.1 Mô phỏng Simulink

Mô phỏng kiểm tra phương trình động học của robot(mục 3.1) trên Simulink.

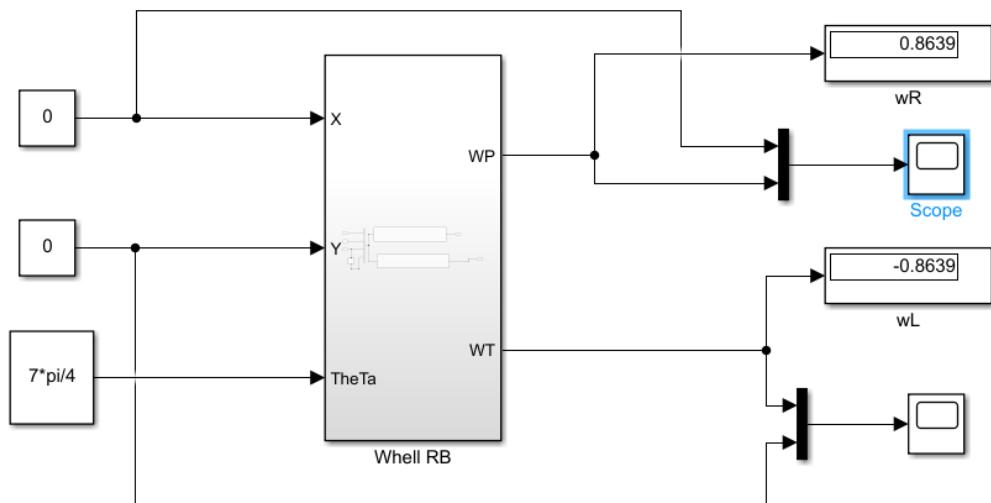
- ❖ Trường hợp robot di chuyển thẳng: góc tạo bởi trục robot và trục x không đổi, tương đương $\dot{\theta} = 0$.



Hình 3.21: Mô hình mô phỏng Simulink robot di chuyển thẳng

Kết quả mô phỏng cho thấy tốc độ góc của bánh trái và bánh phải như nhau với $\omega_r = \omega_l = 1.1 \text{ rad/s}$.

- ❖ Trường hợp robot thực hiện chuyển động xoay tại chổ: Tọa độ A(x,y) không đổi, tương đương $\dot{x} = 0, \dot{y} = 0$.



Hình 3.22: Mô hình mô phỏng simulin robot chuyển động xoay tại chổ

Kết quả mô phỏng cho thấy tốc độ góc của bánh trái và bánh phải bằng nhau nhưng ngược chiều với $-\omega_r = \omega_l = 0.86 rad/s$.

3.5.2 Thực nghiệm bộ điều khiển PID

Bộ điều khiển PID được áp dụng vào điều khiển động cơ của robot nhằm mục đích tạo sự ổn định tốc độ quay cho hai bánh xe robot, với các thông số động cơ đã được chọn (tiêu mục 3.2.2.2) bộ điều khiển PID được thực nghiệm trực tiếp và chỉnh định bằng tay các thông số được chọn như sau:

- ❖ Bánh trái thông số chỉnh định thực nghiệm được chọn $K_p = 7.2$, $K_i = 0.035$, $K_d = 0.01$.
- ❖ Bánh phải thông số chỉnh định thực nghiệm được chọn $K_p = 6.8$, $K_i = 0.032$, $K_d = 0.008$.
 - Các trường hợp chạy thực nghiệm trực tiếp:
- ❖ Robot chạy không tải: Tốc độ vòng quay bánh xe qua quan sát như sau..



Hình 3.23: Tốc độ quay của động cơ không tải

- ❖ Robot chạy mặt phẳng nền gạch: Khi cho robot di chuyển thẳng, kết quả robot chạy lệch về trái hoặc phải so với phương ban đầu (5-10 độ) tùy thuộc vào mặt phẳng di chuyển.

CHƯƠNG 4 : ỨNG DỤNG CÔNG NGHỆ XỬ LÝ ẢNH TRONG LUẬN VĂN

4.1 Xử lý ảnh

4.1.1 Mục đích và nội dung chương trình

Chương trình xử lý ảnh được viết bằng ngôn ngữ Python kết hợp với việc sử dụng thư viện OpenCV và một số thư viện hỗ trợ khác có mục đích tìm đường và dẫn hướng cho robot trong quá trình di chuyển theo thời gian thực. Để đạt được những mục tiêu đó, chương trình cần phải giải quyết được các nội dung sau:

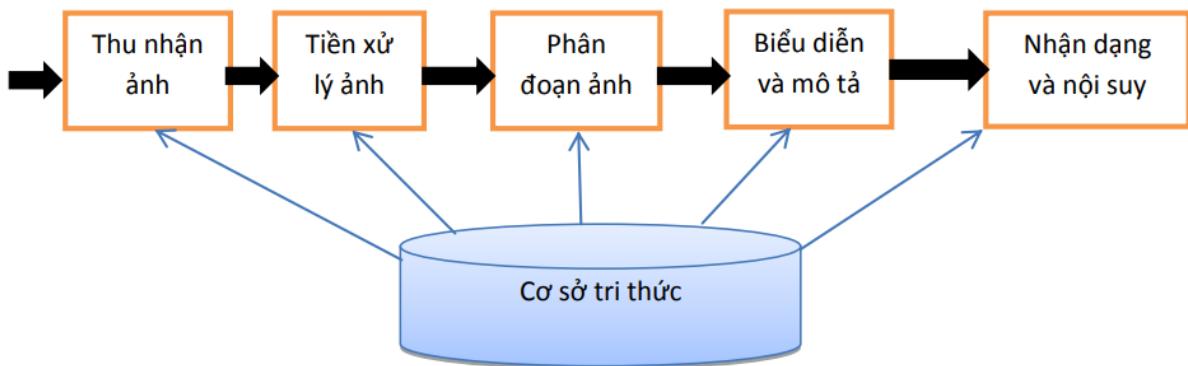
- Khả năng tương tác với đối tượng, phát hiện và bám theo đối tượng.
- Giao tiếp với thiết bị ngoại vi, gửi tín hiệu dẫn hướng cho robot.

4.1.2 Giới thiệu về hệ thống xử lý ảnh

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học còn khá mới so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính riêng chuyên dụng cho nó ngày càng đa dạng và mở rộng.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số - một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tính chập, các biến đổi Fourier, Laplace, các bộ lọc... Thứ hai, các công cụ như Đại số tuyến tính, xác suất, thống kê. Một số kiến thức cần thiết khác như Trí tuệ nhân tạo, mạng nơron nhân tạo cũng được đề cập đến trong quá trình phân tích và đặc biệt là nhận dạng – OCR.

Ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh, scan, ...). Trước đây, các tấm ảnh thu được qua máy chụp hình với phim âm bản, không có tác dụng trong xử lý ảnh. Với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo – máy ảnh số, chụp qua điện thoại ... là những ví dụ gần gũi. Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh, có thể quét từ ảnh chụp bằng máy quét ảnh. Dưới đây mô tả các bước cơ bản trong xử lý ảnh.



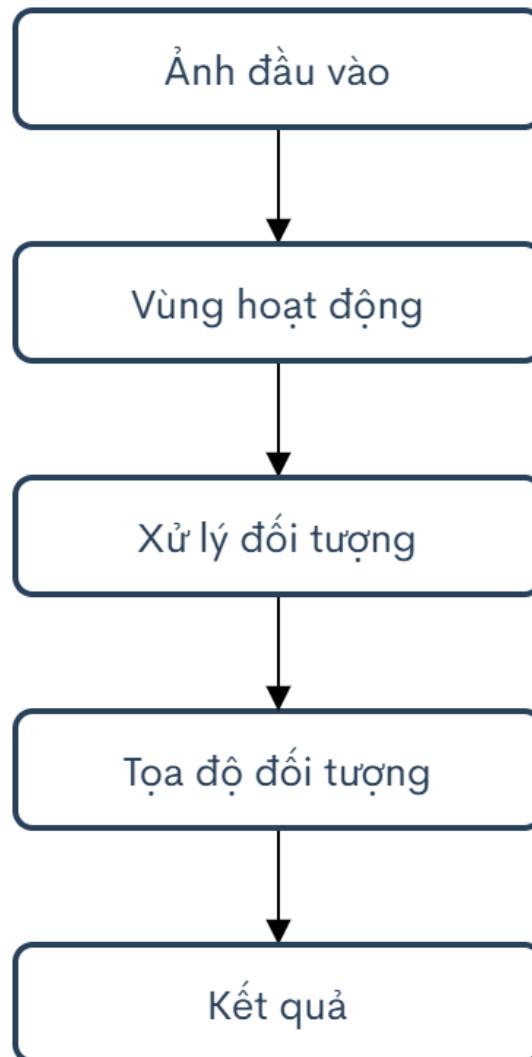
Hình 4.1: Các bước trong xử lý ảnh

4.1.3 Sơ lược xử lý ảnh trong đề tài

Trọng tâm của đề tài là xử lý ảnh, các quá trình điều khiển robot di chuyển đều liên quan đến xử lý ảnh, để xác định vùng làm việc Robot cho đến tọa độ của đối tượng bao gồm phát hiện khuôn mặt và màu đỏ. Xử lý ảnh cũng được xem như là thị giác của Robot, bởi thế Robot có thể di chuyển và chuyển động.

Trong đề tài: "Mô hình mobile robot bám sát đối tượng, hỗ trợ thủ tục giấy tờ hành chính cho sinh viên tại khoa Kỹ Thuật Cơ Khí" Hình ảnh được xử lý như sau: (Xem hình 4.2)

- Thu nhận ảnh đầu vào: Ảnh đầu vào được thu nhận qua camera.
- Vùng hoạt động: Tiến trình này sẽ tiến hành nhận dạng vùng hoạt động của Robot thông qua ảnh đầu vào.
- Xử lý đối tượng: Đến đây đối tượng sẽ được xác định bằng các thuật toán được xây dựng sẵn trong các gói hoặc thư viện hỗ trợ.
- Xác định đối tượng: Quá trình này sẽ tiến hành phát hiện đối tượng, ở đây là khuôn mặt và phát hiện màu đỏ dựa trên vùng hoạt động của camera.
- Kết quả: Cuối cùng sau khi phát hiện sẽ đưa ra những thông số cần thiết như tọa độ và độ chính xác đối tượng trong khung hình.



Hình 4.2: Sơ đồ quá trình xử lý ảnh trong đề tài

4.1.4 Thiết bị thu nhận ảnh đầu vào

Như đã nói ở phần trên để nhận ảnh cần máy ảnh, scan, camera,... là nguồn cung cấp dữ liệu ảnh đầu vào cho các ứng dụng xử lý ảnh. Trong đó camera là lựa chọn phù hợp nhất sử dụng trong đề tài, để chọn camera thì phải xét đến các thông số như: độ phân giải, số điểm ảnh, điều kiện hoạt động của thiết bị, tùy theo môi trường thực tế và yêu cầu hệ thống có thể lựa chọn camera phù hợp.

Để tài chọn camera cho hệ thống xử lý ảnh phát hiện khuôn mặt và màu đỏ, được chọn là Webcam Logitech Pro 9000, với ứng dụng công nghệ Rightlight tự động điều chỉnh để được ánh sáng tối ưu, được tích hợp Micro công nghệ RightSound, góc quay rộng (75^0), Độ phân giải 960x720 pixels, 30fps. Đáp ứng phù hợp với yêu cầu của đề tài.



Hình 4.3: Webcam Logitech Pro 9000

4.2 Xử lý ảnh trong phát hiện khuôn mặt

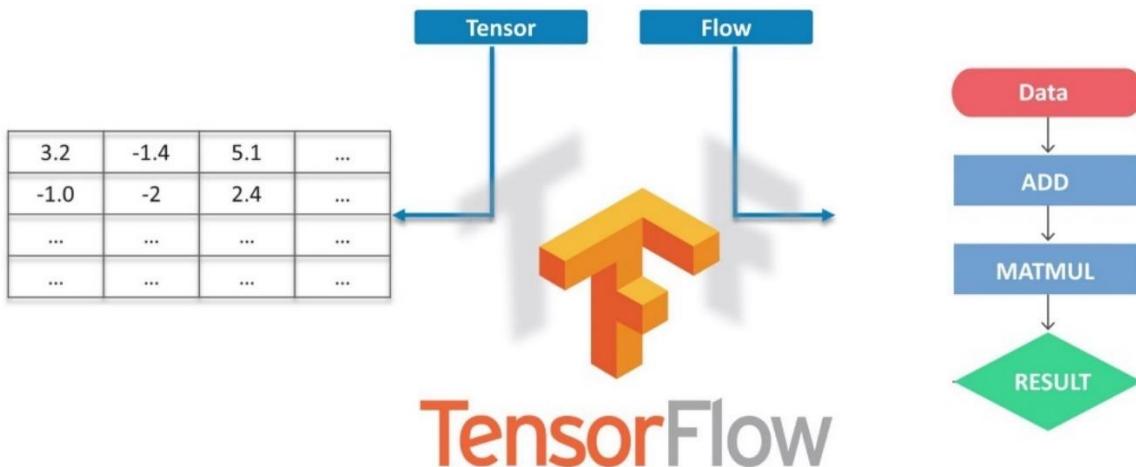
4.2.1 TensorFlow object detection API

4.2.1.1 Tensorflow

Tensorflow là một framework mã nguồn mở cho Machine learning của Google để lập trình luồng dữ liệu trong một loạt các tác vụ. Các nút trong biểu đồ đại diện cho các phép toán, trong khi các cạnh của đồ thị đại diện cho các mảng dữ liệu đa chiều (tensor) được giao tiếp giữa chúng. Tensors chỉ là mảng nhiều chiều, một phần mở rộng của bảng dữ liệu 2 chiều với chiều cao hơn. Có nhiều tính năng của Tensorflow thích hợp cho việc deep learning.

Một số công cụ của Tensorflow:

- TensorBoard: công cụ giúp minh họa các đồ thị tính toán (computational graph), sự thay đổi giá trị của các hàm tính toán (loss, accuracy,...) dưới dạng biểu đồ.
- TensorFlow Serving: công cụ giúp triển khai các mô hình Machine Learning viết bằng TensorFlow thành một sản phẩm thực sự.
- Các API giúp cho việc sử dụng TensorFlow dễ dàng hơn được phát triển bởi những nhà nghiên cứu về Machine Learning trên toàn thế giới (TensorFlow High Level API, TF-Slim, TensorFlow Object Detection API)
- Tập hợp code mẫu giúp cho những người mới học dễ tiếp cận hơn.



Hình 4.4: Biểu diễn chức năng TensorFlow

4.2.1.2 API

API (Application Programming Interface) là một giao diện lập trình ứng dụng. Cung cấp cho người dùng một tập hợp các hoạt động phổ biến được cung cấp sẵn từ các lập trình viên đã tạo ra trước đó.

API phát hiện đối tượng của TensorFlow là một khuôn khổ mã nguồn mở được xây dựng trên TensorFlow để xây dựng, đào tạo và triển khai các mô hình phát hiện đối tượng. Bằng cách sử dụng tính năng học chuyển giao (định vị lại mô hình được đào tạo trước để sử dụng với các mục bên ngoài tập dữ liệu đào tạo ban đầu), API phát hiện đối tượng cung cấp khả năng phát hiện nhiều đối tượng cho các mục tùy chỉnh miễn là có tập dữ liệu được xây dựng có kích thước phù hợp.

4.2.2 Mô hình SSD MobileNet V2

4.2.2.1 Mạng MobileNet

MobileNet được phát triển bởi đội ngũ Google, mô hình sử dụng cách tính chập tích mang tên DSC (Depthwise Separable Convolution) nhằm giảm kích thước mô hình và giảm độ phức tạp tính toán. Do đó, MobileNet thường được sử dụng cho các ứng dụng Computer Vision trên các thiết bị nhỏ gọn như điện thoại thông minh hay thiết bị nhúng.

Kiến trúc mạng mobileNet gồm 30 lớp, mobileNet dùng rất nhiều lớp Depthwise Separable Convolution để giảm số lượng parameter đi nhiều lần khoảng 9 lần.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Hình 4.5: Mô hình kiến trúc mạng mobileNet

Có nhiều gói để sử dụng cho việc phát hiện đối tượng trong trang chủ sử dụng cho quá trình phát hiện và nhận diện đối tượng. Nhưng nếu muốn deploy các mô hình trên cho các ứng dụng real time, ta cần phải có cấu hình cực kỳ mạnh mẽ (GPU / TPU) còn đối với các hệ thống nhúng (Raspberry Pi, Nano pc, etc) hay các ứng dụng chạy trên smartphone, ta cần có một mô hình “nhẹ” hơn.

Để tài sử dụng Laptop cá nhân nên còn hạn chế nên cần một mô hình nhẹ và hiệu quả, sau khi tham khảo một số mô hình (hình 4.6) để tài chọn MobileNet V2 để xây dựng chương trình.

Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco ☆	26	18	Boxes
ssd_mobilenet_v1_quantized_coco ☆	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco ☆	29	16	Boxes
ssd_mobilenet_v1_ppn_coco ☆	26	20	Boxes
ssd_mobilenet_v1_fpn_coco ☆	56	32	Boxes
ssd_resnet_50_fpn_coco ☆	76	35	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	27	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	58	28	Boxes
faster_rcnn_resnet50_coco	89	30	Boxes

Hình 4.6: Một số gói mô hình điển hình

Mô hình SSD Mobile V2 là một multibox- Single-Shot Detection (SSD) mạng nhằm thực hiện phát hiện đối tượng. Mô hình đã được đào tạo từ tập dữ liệu hình ảnh đối tượng chung trong ngữ cảnh (COCO).

Đầu vào mô hình là một đốm màu bao gồm một hình ảnh 1x3x300x300 theo thứ tự RGB. Đầu ra của mô hình là một vectơ điển hình chứa dữ liệu đối tượng được theo dõi. Lưu ý rằng dữ liệu "class_id" hiện là quan trọng và nên được sử dụng để xác định phân loại cho bất kỳ đối tượng được phát hiện nào.

MobileNet V2 là một trong những kiến trúc được ưa chuộng nhất khi phát triển các ứng dụng AI trong computer vision. MobileNet V2 có một số điểm cải tiến so với MobileNet V1 giúp cho nó có độ chính xác cao hơn, số lượng tham số và số lượng các phép tính ít hơn.

4.2.2.2 Thuật toán SSD (Single Shot Object Detectors)

SSD (Single Shot Object Detectors) được hiểu như sau:

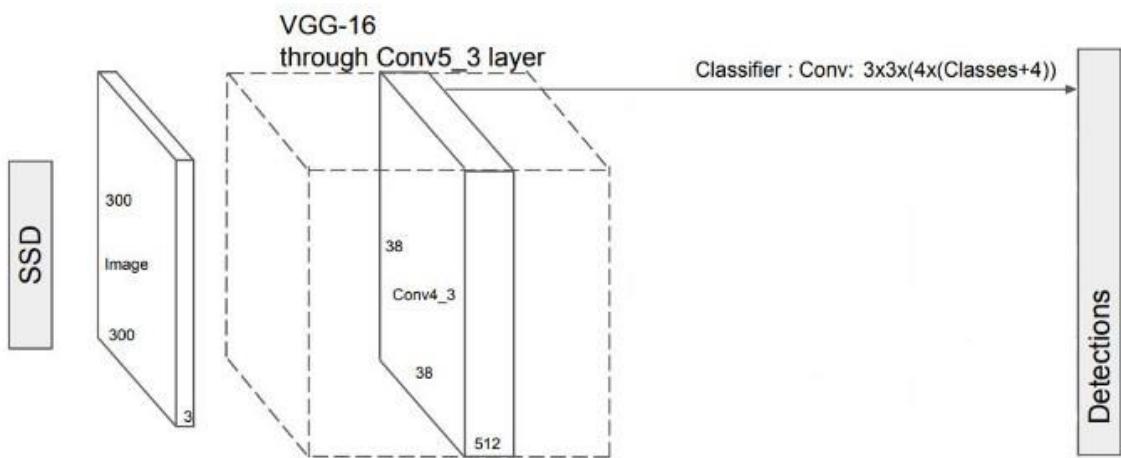
- Single Shot: Có nghĩa là các nhiệm vụ của định vị hóa và phân loại đối tượng được thực hiện trong một lần chuyển tiếp duy nhất của vật thể.
- MultiBox: Tên của một kỹ thuật hộp ràng buộc được phát triển bởi Szegedy et al.

- Detector: Là một bộ dò tìm đối tượng và phân loại các đối tượng được phát hiện.

SSD được thiết kế để phát hiện đối tượng trong thời gian thực. R-CNN nhanh hơn sử dụng mạng để xuất khu vực để tạo các hộp ranh giới và sử dụng các hộp đó để phân loại các đối tượng. Mặc dù được coi là khởi đầu chính xác, toàn bộ quá trình chạy ở 7 khung hình mỗi giây. Thấp hơn nhiều so với những gì cần cho một nhu cầu xử lý thời gian thực. SSD tăng tốc quá trình bằng cách loại bỏ sự cần thiết của mạng để xuất khu vực. Để giải quyết về vấn đề độ chính xác giảm, SSD áp dụng một vài cải tiến bao gồm các feature map đa kích thước và sử dụng các hộp mặc định. Những cải tiến này cho phép SSD tiến gần được với độ chính xác của Faster R-CNN nhưng lại có thể sử dụng hình ảnh có độ phân giải thấp hơn, giúp đẩy tốc độ cao hơn.

Mô hình SSD được chia làm hai giai đoạn:

- Trích xuất feature map (dựa vào mạng cơ sở VGG16) để tăng hiệu quả trong việc phát hiện => thì nên sử dụng ResNet, InceptionNet, hoặc MobileNet;
- Áp dụng các bộ lọc tích chập để có thể phát hiện được các đối tượng.



Hình 4.7: Giai đoạn trích xuất feature map

SSD được thiết kế để phát hiện đối tượng trong thời gian thực. R-CNN nhanh hơn sử dụng mạng để xuất khu vực để tạo các hộp ranh giới và sử dụng các hộp đó để phân loại các đối tượng. Mặc dù được coi là khởi đầu chính xác, toàn bộ quá trình chạy ở 7 khung hình mỗi giây. Thấp hơn nhiều so với những gì cần cho một nhu cầu

xử lý thời gian thực. SSD tăng tốc quá trình bằng cách loại bỏ sự cần thiết của mạng đề xuất khu vực. Để giải quyết về vấn đề độ chính xác giảm, SSD áp dụng một vài cải tiến bao gồm các feature map đa kích thước và sử dụng các hộp mặc định. Những cải tiến này cho phép SSD tiến gần được với độ chính xác của Faster R-CNN nhưng lại có thể sử dụng hình ảnh có độ phân giải thấp hơn, giúp đẩy tốc độ cao hơn.

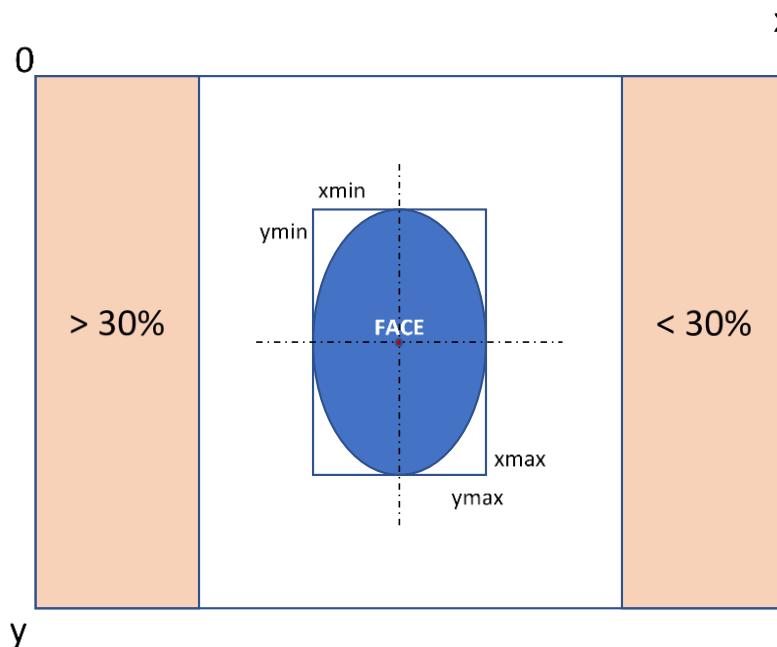
4.2.3 Giá trị đề tài sử dụng

Khi đầu vào là khuôn mặt được phát hiện trong khung hình, thì trong các hàm tính toán trong các bản của ảnh, xuất ra được tọa độ khung hình bao gồm: xmax, ymax, xmin, ymin, min_score_thresh (phần trăm phát hiện khuôn mặt).

```
for box, color in box_to_color_map.items():
    ymin, xmin, ymax, xmax = box
    height, width, channels = image.shape
    ymin = int(ymin*height)
    ymax = int(ymax*height)
    xmin = int(xmin*width)
    xmax = int(xmax*width)
    coordinates_list.append([ymin, ymax, xmin, xmax, (box_to_score_map[box]*100), display_str[counter_for]])
    counter_for = counter_for + 1
```

Hình 4.8: Hàm trích xuất tọa độ của chương trình

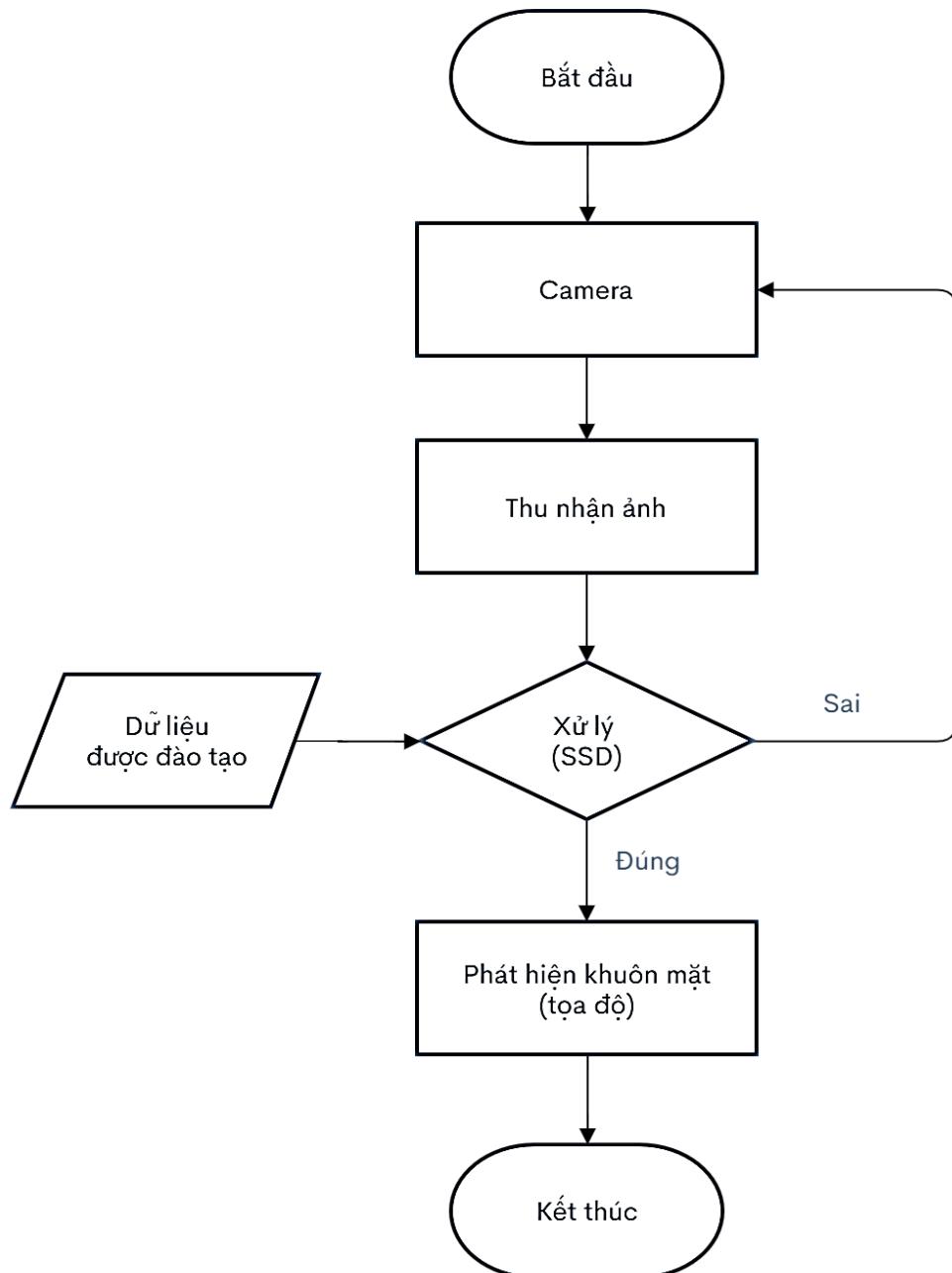
Đề tài sử dụng tọa độ khuôn mặt trên khung hình khi được phát hiện để thực hiện việc điều khiển robot. Được tính toán diện tích khuôn mặt vị trí gần và xa, và lệch trái hoặc phải của khuôn mặt đối với khung hình từ những tọa độ được trích xuất.



Hình 4.9: Tọa độ phát hiện khuôn mặt

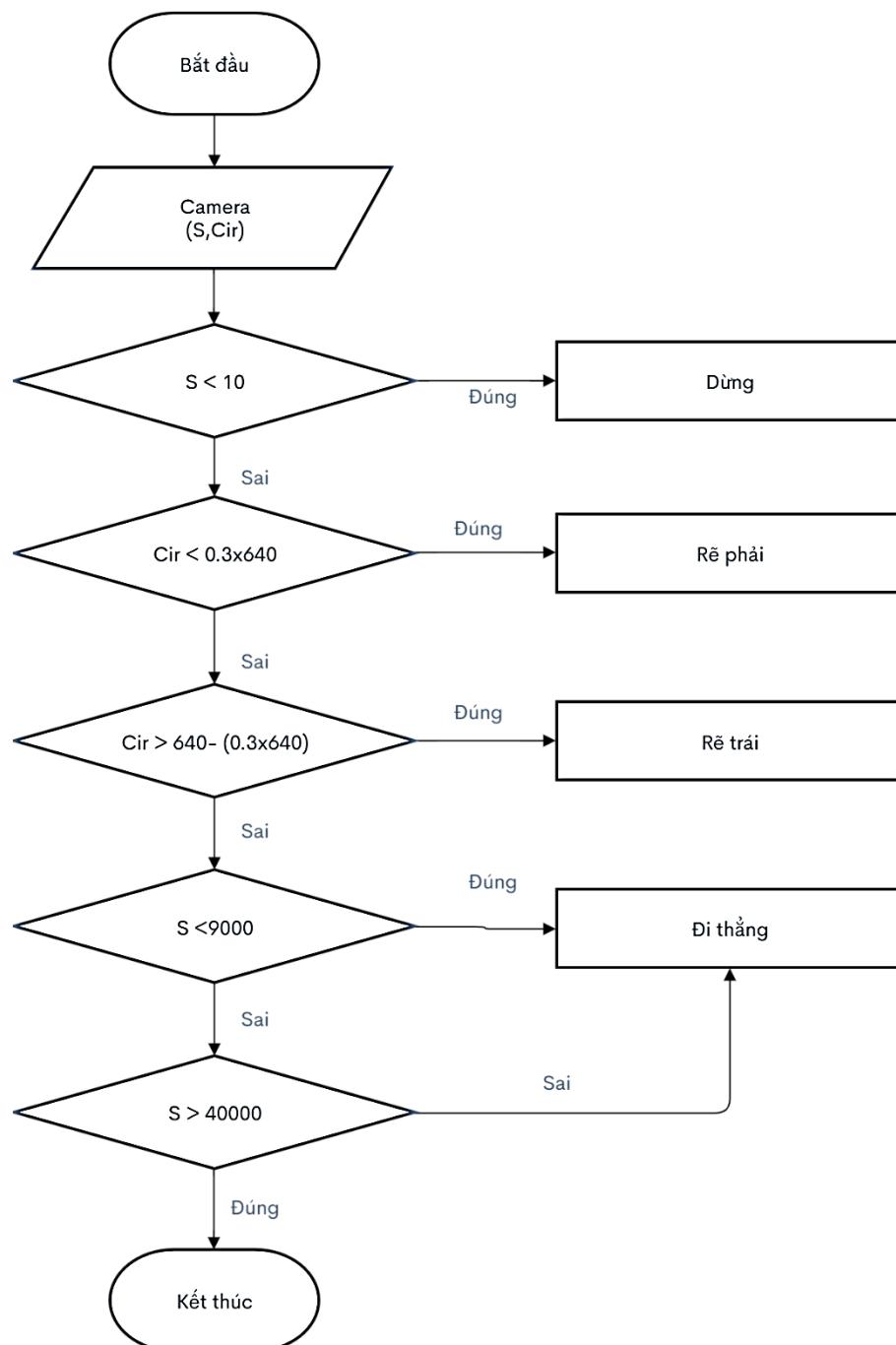
Nếu có nhiều đối tượng, robot không phân biệt được đối tượng chính để bám theo, lúc đó robot bị xung đột, nên biện pháp làm giảm bớt cho nhiều đối tượng là tăng phần trăm phát hiện khuôn mặt đồng thời chỉ xuất ra một tọa độ khi phát hiện được dự đoán là gần giống nhất trong nhiều khuôn mặt.

4.2.4 Lưu đồ giải thuật



Hình 4.10: Lưu đồ giải thuật phát hiện khuôn mặt

Khi bắt đầu camera sẽ thu nhận hình ảnh và đưa vào thuật toán SSD của mô hình ssd mobile V2 chuyển ảnh qua thành các mảng, phân lớp, trích xuất feature map,... tính toán đưa ra phát hiện khuôn mặt đồng thời đưa ra dự đoán phần trăm khuôn mặt được phát hiện, và nếu không phát hiện hệ thống tiếp tục tìm trong khung hình camera.



Hình 4.11: Lưu đồ giải thuật điều khiển robot bám sát khuôn mặt

Khi bắt đầu, chương trình dò tìm khuôn mặt sẽ tìm trong khung hình camera thu được nếu không phát hiện được khuôn mặt, ở đây biểu diễn giá trị đó trên biến diện tích S, Nếu ($S < 10$) tín hiệu gửi đi là không di chuyển, tiếp đến khi phát hiện chương trình sẽ xét mẫu 5s, khi đúng là khuôn mặt sẽ xét đến vị trí tọa độ đối tượng trong khung, nếu đối tượng lệch về trái quá 30% của khung hình, robot sẽ rẽ về trái, và ngược lại lệch phải 30% sẽ rẽ phải để đối tượng về vị trí trung tâm, sau khi cân chỉnh vị trí đối tượng trong khung hình, tiếp đến là tính khoảng cách tương đối nếu ($S < 9000$) sẽ gửi tín hiệu đi thẳng đến khi ($S > 40000$) sẽ kết thúc quá trình phát hiện khuôn mặt.

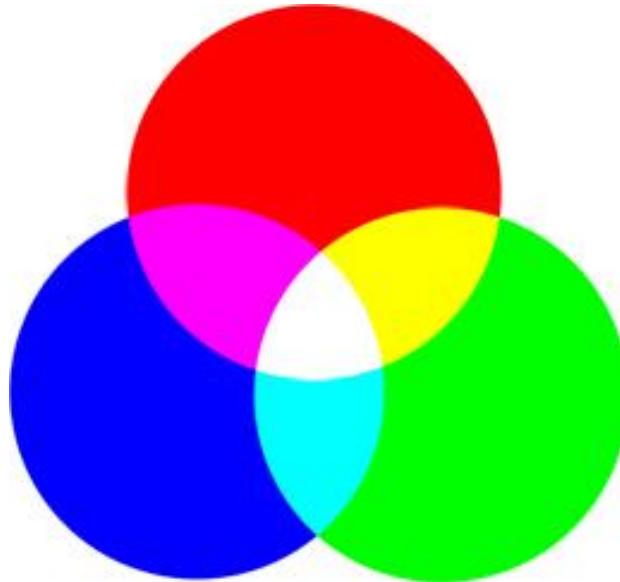
4.3 Xử lý ảnh trong phát hiện màu

4.3.1 Không gian màu

Không gian màu là mô hình toán học dùng để mô tả màu sắc trong thực tế được biểu diễn dưới dạng số học. Trên thực tế có rất nhiều không gian màu khác nhau được mô hình được sử dụng trong nhiều mục đích khác nhau.

Trong đề tài sử dụng hai không gian màu HSV và YcbCr để phát hiện màu đỏ.

4.3.1.1 Không gian RGB



Hình 4.12: Logo RGB

RGB là không gian màu phổ biến dùng trong máy tính, máy ảnh, điện thoại và nhiều thiết bị kỹ thuật số khác nhau. Không gian màu này gần với cách mắt người

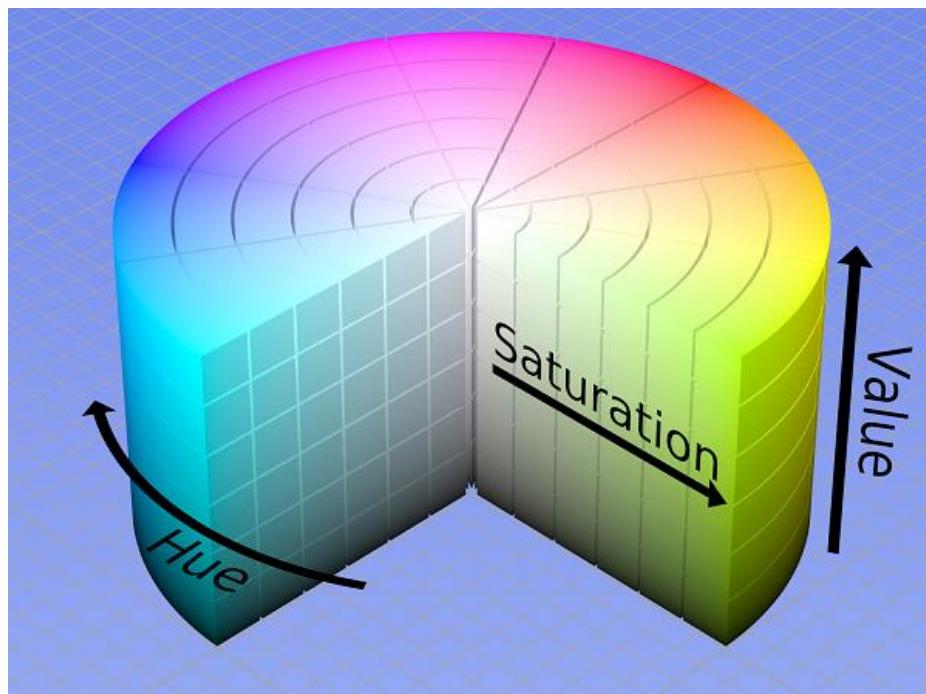
tổng hợp màu sắc. Nguyên lý cơ bản là sử dụng 3 màu sắc cơ bản R (red - đỏ), G (green - xanh lục) và B (blue - xanh lam) để biểu diễn tất cả các màu sắc.

Thông thường, trong mô hình 24 bit mỗi kênh màu sẽ sử dụng 8bit để biểu diễn, tức là giá trị R, G, B nằm trong khoảng 0 - 255. Bộ 3 số này biểu diễn cho từng điểm ảnh, mỗi số biểu diễn cho cường độ của một màu. Với mô hình màu 24bit thì số màu tối đa có thể tạo ra là $255 \times 255 \times 255 = 16581375$ màu. Một điểm cần lưu ý là với các thư viện đọc ảnh và hiển thị ảnh như matplotlib, Pillow thì các ảnh được đọc theo RGB tuy nhiên Opencv đọc ảnh theo các kênh BGR.

Vd: (0, 0, 0) là màu đen, (255, 255, 255) là màu trắng,, (255, 0, 0) là màu đỏ, 0, 255, 0) là màu xanh lá cây, (0, 0, 255) là màu xanh lam,....

Bên cạnh hệ màu RGB thì ta sẽ nghe đến RGBA, thực ra đây là một hệ màu được lấy căn bản từ hệ màu RGB tuy nhiên có thêm một kênh alpha (). Kênh được sử dụng như là kênh mờ, nếu một pixel có giá trị 0% trong kênh của nó thì nó hoàn toàn trong suốt, trong khi giá trị 100% sẽ khiến cho điểm đó bị mờ đục. Điều này được ứng dụng rất nhiều trong việc ghép các ảnh lại và mang một độ chân thực nhất định.

4.3.1.2 Không gian màu HSV



Hình 4.13: Không gian màu HSV

Không gian màu HSV (còn gọi là HSB) là một cách tự nhiên hơn để mô tả màu sắc, dựa trên 3 số liệu:

- H viết tắt của từ HUE có nghĩa là vùng màu
- S viết tắt của từ SATURATION có nghĩa là độ bão hòa màu
- B (hay V) viết tắt của chữ BRIGHT hay VALUE có nghĩa là giá trị hay độ sáng của màu sắc

Giá trị của kênh Hue có phạm vi [0, 179], Saturation và Value có phạm vi [0, 255]. Một số phần mềm sử dụng không gian màu này nhưng với hệ quy đổi khác. Khi đó phải tính toán lại giá trị của từng kênh màu cho phù hợp với hệ quy đổi cơ bản trên.

Ảnh được biểu diễn trong không gian màu BGR, nhưng không gian màu HSV lại dễ dàng xử lý hơn. Dãy màu của kênh Hue là “bảy sắc cầu vồng” từ đỏ đến tím rồi lại về đỏ, nên có thể xác định được màu. Do đó trong hầu hết bài toán đều phải tiến hành chuyển ảnh ban đầu sang không gian màu HSV trước khi xử lý những bước tiếp theo.

Công thức chuyển đổi từ RGB sang HSV: (docs.opencv.org)

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0, \text{otherwise} \\ 0 & \text{if } V = 0 \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \\ 0 & \text{if } R = G = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On Output $\leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$

Các giá trị được chuyển đổi thành kiểu dữ liệu type:

Ảnh 8-bit: $V \leftarrow 255V, S \leftarrow 255S, H \leftarrow H/2$ (để vừa với 0 đến 255)

Ảnh 16 bit: (hiện không được hỗ trợ) $V < -65535V, S < -65535S, H < -H$

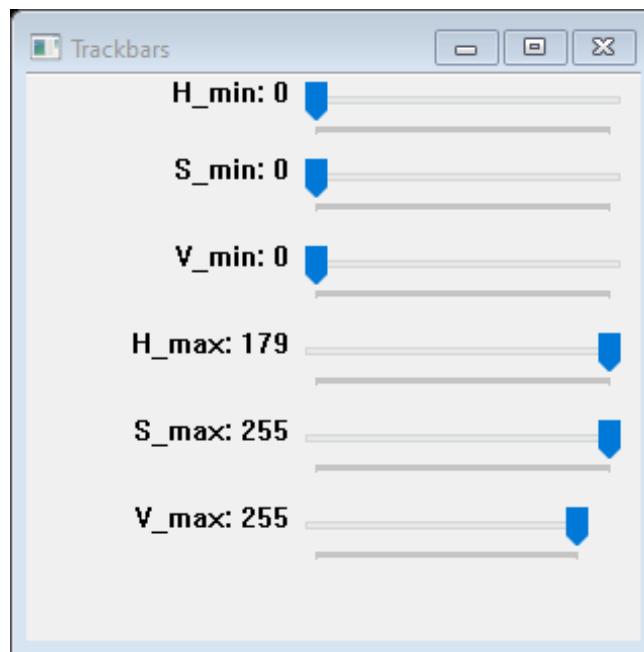
Ảnh 32-bit: H, S và V được giữ nguyên

Màu sắc	Màu sắc Tên	Hex	(R, G, B)	(H, S, V)
	Đen	# 000000	(0,0,0)	(0 °, 0%, 0%)
	trắng	#FFFFFF	(255,255,255)	(0 °, 0%, 100%)
	Màu đỏ	# FF0000	(255,0,0)	(0 °, 100%, 100%)
	Vôi	# 00FF00	(0,255,0)	(120 °, 100%, 100%)
	Màu xanh da trời	# 0000FF	(0,0,255)	(240 °, 100%, 100%)
	Màu vàng	# FFFF00	(255,255,0)	(60 °, 100%, 100%)
	Lục lam	# 00FFFF	(0,255,255)	(180 °, 100%, 100%)
	Đỏ tươi	# FF00FF	(255,0,255)	(300 °, 100%, 100%)
	Bạc	#BFBFBF	(191,191,191)	(0 °, 0%, 75%)
	Màu xám	# 808080	(128,128,128)	(0 °, 0%, 50%)
	BỎ RƠI	# 800000	(128,0,0)	(0 °, 100%, 50%)
	Ôliu	# 808000	(128,128,0)	(60 °, 100%, 50%)
	màu xanh lá	# 008000	(0,128,0)	(120 °, 100%, 50%)
	Màu tím	# 800080	(128,0,128)	(300 °, 100%, 50%)
	Màu mòng két	# 008080	(0,128,128)	(180 °, 100%, 50%)
	Hải quân	# 000080	(0,0,128)	(240 °, 100%, 50%)

Hình 4.14: Bảng màu từ RGB sang HSV

Ứng dụng điển hình nhất của HSV là trong việc lọc màu. Chính vì thế luận văn dùng HSV xác định đối tượng màu đỏ.

Với bài toán này thì hướng tiếp cận đơn giản nhất chính là lựa chọn giá trị HSV phù hợp bằng cách tạo ra một bản tinh chỉnh giá trị HSV được viết trên Python. Từ kiến thức ở bên trên thì mỗi một màu sẽ được thể hiện tương đối qua một khoảng giá trị với mỗi hệ màu tương ứng.



Hình 4.15: Bản tinh chỉnh giá trị HSV

4.3.1.3 Không gian màu YCbCr

YCbCr, Y'CbCr hoặc Y Pb/Cb Pr/ Cr, cũng được viết là YCBCR hoặc Y'CBCR, là một họ các không gian màu được sử dụng như một phần của đường dẫn hình ảnh màu trong video và hệ thống chụp ảnh kỹ thuật số. Y là thành phần luma và Cb và Cr là các thành phần sắc độ khác nhau của màu xanh và màu đỏ. Y' (với dấu ') được phân biệt với Y, đó là độ chói, nghĩa là cường độ ánh sáng được mã hóa phi tuyến dựa trên các nguyên tắc RGB được hiệu chỉnh gamma.

Các không gian màu Y'CbCr được xác định bằng phép biến đổi tọa độ toán học từ không gian màu RGB tương ứng. Nếu không gian màu RGB bên dưới là tuyệt đối, thì không gian màu Y'CbCr cũng là một không gian màu tuyệt đối, ngược lại nếu không gian RGB không xác định, thì Y'CbCr cũng vậy.

Mục đích chính của không gian màu YcbCr làm bão hòa màu sắc

Công thức tính chuyển đổi RGB sang YcrCb và ngược lại.

+ RGB → YCrCb:

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

$$Cr \leftarrow (R - Y) \cdot 0,713 + \text{delta}$$

$$Cb \leftarrow (B - Y) \cdot 0,564 + \text{delta}$$

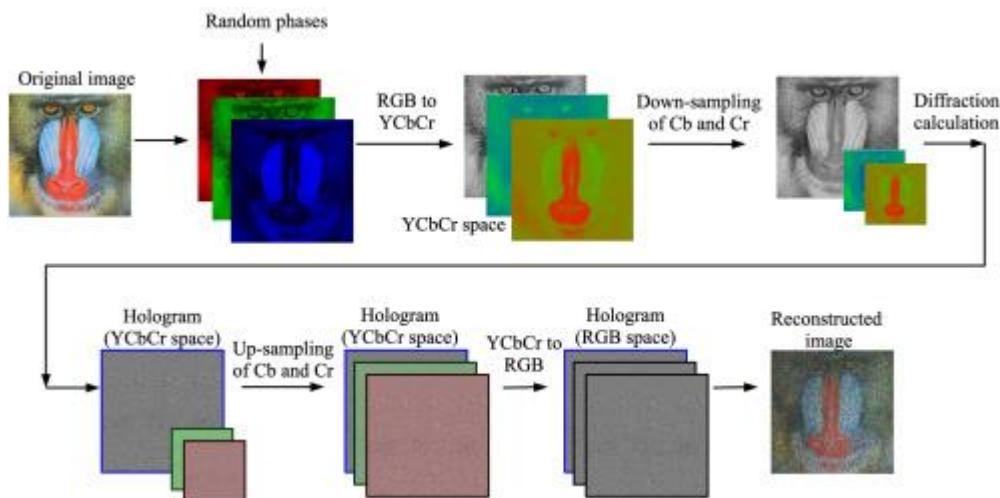
+ $\text{RGB} \leftarrow YCrCb$:

$$R \leftarrow Y + 1.403 \cdot (Cr - \text{delta})$$

$$G \leftarrow Y - 0.714 \cdot (Cr - \text{delta}) - 0.344 \cdot (Cb - \text{delta})$$

$$B \leftarrow Y + 1.773 \cdot (Cb - \text{delta})$$

Trong đó: $\text{delta} = \begin{cases} 128 \text{ đối với ảnh 8 bit} \\ 32768 \text{ với ảnh 16 bit} \\ 0.5 \text{ với ảnh dấu chấm động} \end{cases}$



Hình 4.16: Chuyển đổi RGB sang YcbCr và ngược lại

4.3.2 Giá trị màu đẽ tài sử dụng

Luận văn lựa chọn màu đỏ để phát hiện như sau:

- Chuyển ảnh gốc từ BGR sang HSV.
- Giới hạn vùng mã màu của đỏ, hay những giá trị trong không gian màu HSV “gần” đỏ.
 - Lọc ra những pixel có giá trị nằm trong giới hạn mã màu tìm được, các pixel có giá trị nằm ngoài vùng giới hạn được gán bằng giá trị đặc biệt khác (ví dụ màu đen).

Trong HSV, màu đỏ là màu khá đặc biệt vì giá trị kênh Hue của nó nằm ở 2 vùng biên. Vì thế màu đỏ có đến 2 vùng phạm vi:

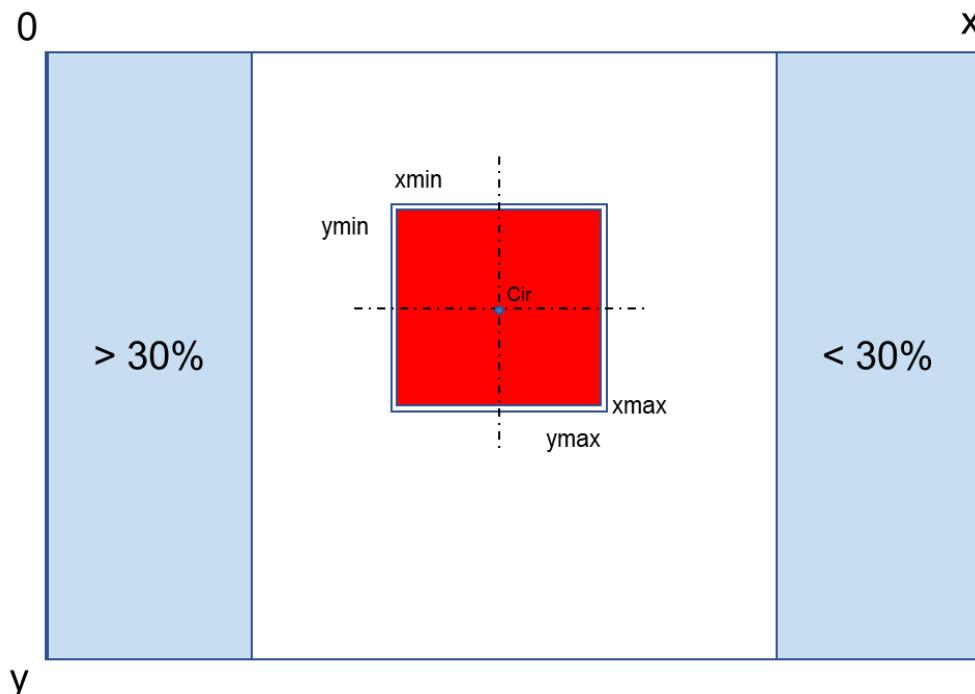
- $[0, 100, 100]$ đến $[10, 255, 255]$
- $[160, 100, 100]$ đến $[179, 255, 255]$

Tuy nhiên tìm các pixel có màu “gần” đỏ là vì trên thực tế, màu đỏ hay bất kỳ màu gì mà ta nói đến không bao giờ có giá trị chính xác cả. Ví dụ các màu đỏ cam,

đỏ tươi, đỏ thẫm, đỏ hồng..., luôn có sự pha trộn màu nhưng ta vẫn gọi đó là màu đỏ. Việc chọn vùng phạm vi của nó cũng mang tính tương đối.

Kết quả sau khi tinh chỉnh giá trị trong HSV phạm vi được chọn: [160, 160, 10], [190 , 255, 255]

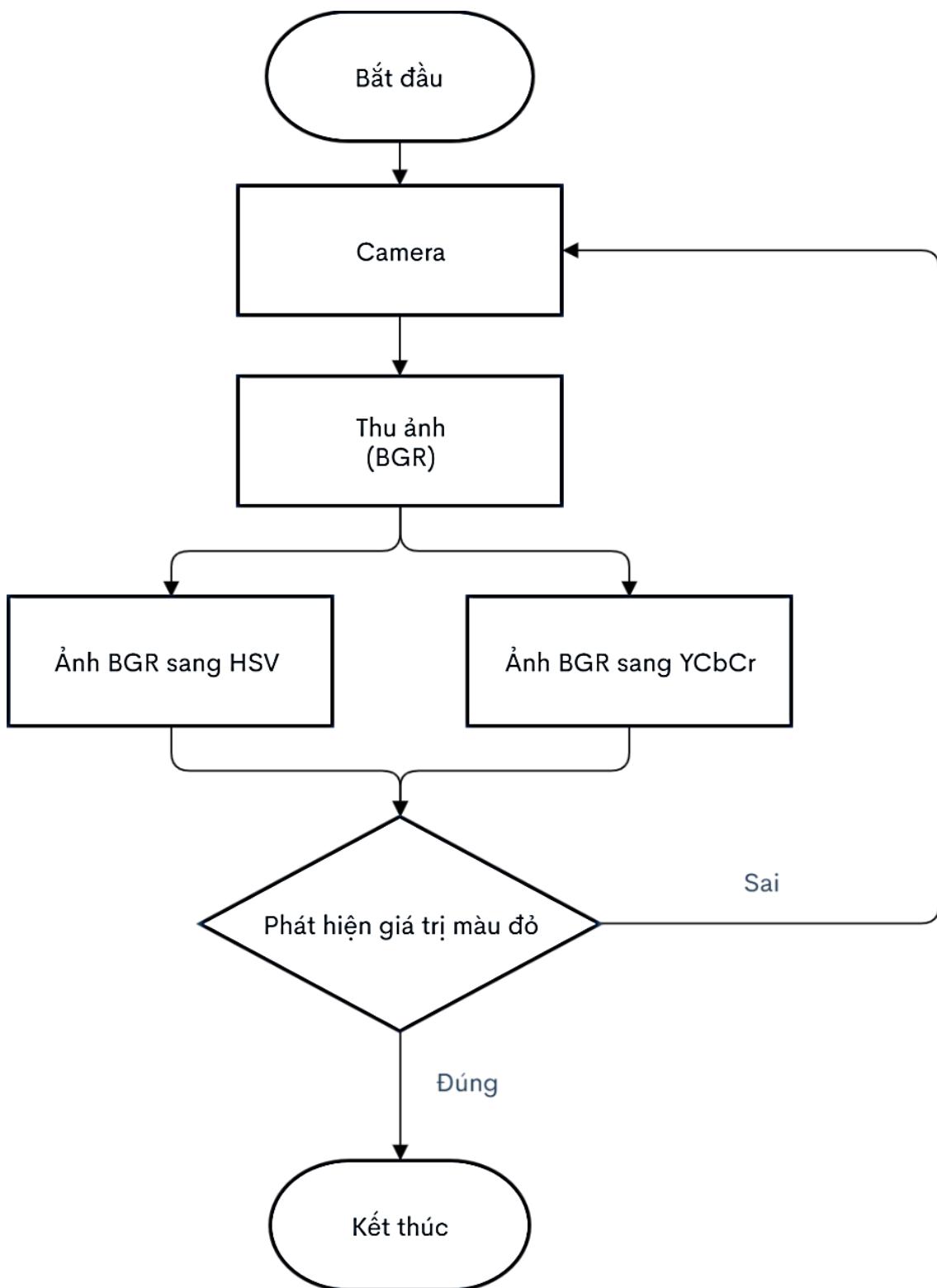
Đến đây chúng ta dùng các tọa độ được trích xuất từ thuật toán trong chương trình, và sử dụng tương tự với phát hiện khuôn mặt, dùng bốn giá trị cơ bản để xác định vị trí của đối tượng màu đỏ: xmax, ymax, xmin, ymin.



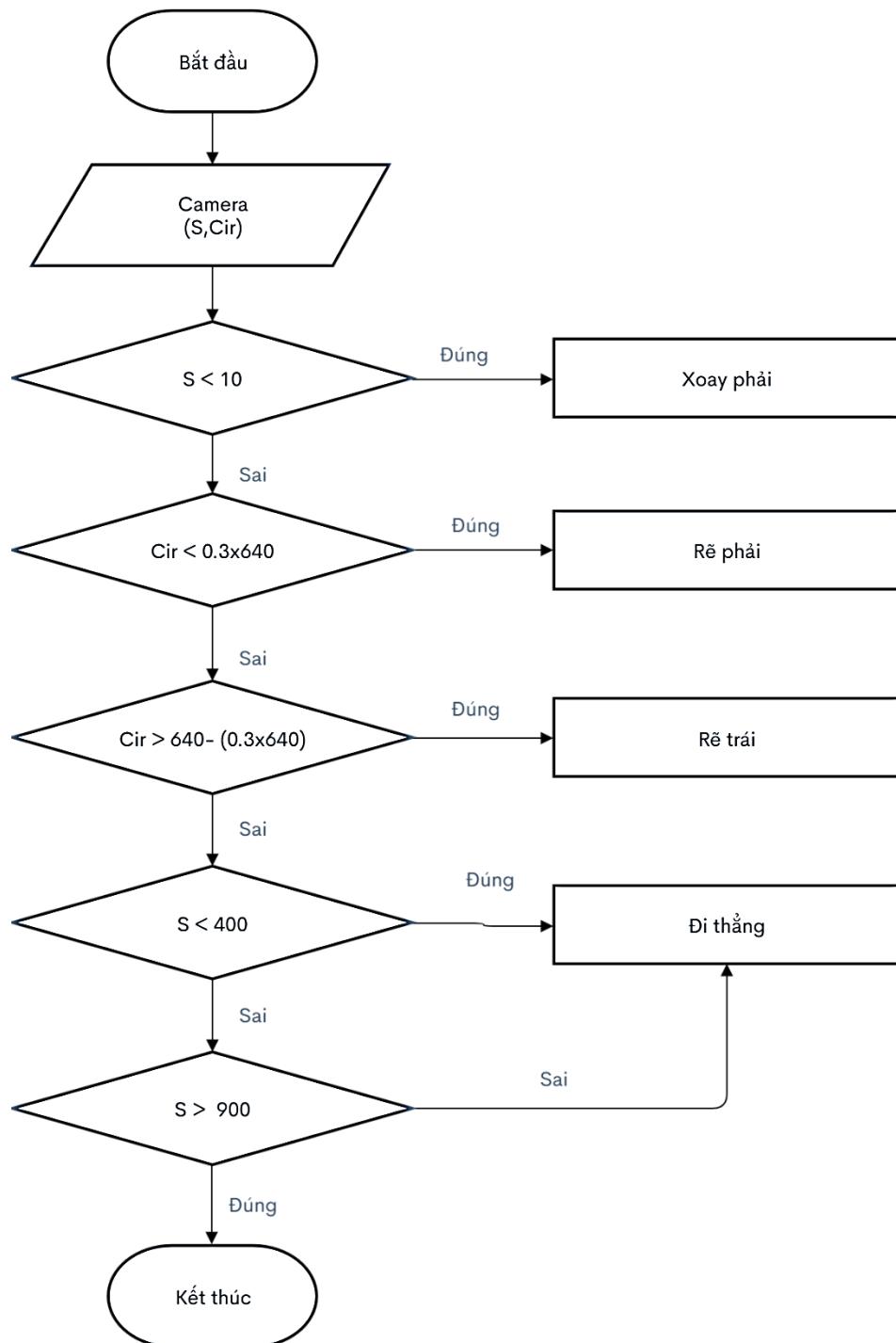
Hình 4.17: Tọa độ phát hiện màu

4.3.3 Lưu đồ giải thuật

Khi bắt đầu camera sẽ bắt đầu thu nhận hình ảnh, hình ảnh thu vào mặc định trong Python là BGR, sau khi nhận ảnh, ảnh sẽ chuyển sang hệ không gian màu HSV và YcbCr để lọc các giá trị màu đỏ, khi trong khung hình nhận được các giá trị trong khoảng màu đỏ sẽ trả ra màn hình và kết thúc, ngược lại nếu không nhận được giá trị sẽ trả về camera, camera tiếp tục thu nhận ảnh đến khi đạt được giá trị và kết thúc (Xem hình 4.18).



Hình 4.18: Lưu đồ giải thuật phát hiện hiện màu đỏ



Hình 4.19: Lưu đồ giải thuật điều khiển robot bám sát màu đỏ về bãi đỗ

S ở đây tương đương với diện tích khi phát hiện, $S < 10$ tức là không phát hiện được đối tượng trong khung hình, lược bỏ một số phát hiện nhỏ khác và song đó sẽ xoay tại chỗ theo hướng cùng chiều kim đồng hồ quét khung hình đến khi tìm thấy đối tượng. $S < 400$ lúc này khung hình đã phát hiện được đối tượng cần tìm hệ thống sẽ tạo một hình chữ nhật bao quanh đối tượng và bắt đầu di chuyển thẳng về phía trước.

Cir ở đây là tâm giữa của đối tượng trong hình khung hình. Nếu đối tượng lệch quá 30% bên trái so với trục x thì robot sẽ phải và ngược lại lệch quá 30% bên phải trục x robot sẽ rẽ trái. Cuối cùng khi robot di chuyển đến vùng diện tích đã chọn, robot sẽ kết thúc.

4.4 Thư viện hỗ trợ chính

4.4.1 Thư viện OpenCV



Hình 4.20: Logo OpenCV

OpenCV được bắt đầu từ Intel năm 1999 bởi Gary Bradsky. OpenCV viết tắt cho Open Source Computer Vision Library. OpenCV là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning, và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo real-time.

OpenCV có thể sử dụng được hầu hết ở các ngôn ngữ như C++, Python, Java,... Tuy nhiên, Python Ngôn ngữ được dùng nhiều để demo / test OpenCV do tính ngắn gọn, ít phải thiết lập. Bên cạnh đó, nếu dùng Python thì cũng có thể code được trên nhiều hệ điều hành.

Tính năng và các module phổ biến của OpenCV:

Theo tính năng và ứng dụng của OpenCV, có thể chia thư viện này thành các nhóm tính năng và module tương ứng như sau:

- Xử lý và hiển thị Hình ảnh/ Video/ I/O (core, imgproc, highgui)
- Phát hiện các vật thể (objdetect, features2d, nonfree)
- Geometry-based monocular hoặc stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)

- CUDA acceleration (gpu)

Luận văn sử dụng thư viện mã nguồn mở OpenCV (Open Source Computer Vision Library) phiên bản 3.2 để hỗ trợ trong quá trình xử lý phát hiện màu đỏ.

4.4.2 Thư viện Tensorflow

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

Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép TensorFlow xây dựng được nhiều neural network. Nó còn cho phép tính toán song song trên nhiều máy tính khác nhau, thậm chí trên nhiều CPU, GPU trong cùng một máy hay tạo ra các dataflow graph – đồ thị luồng dữ liệu để dựng nên các model. Nếu người dùng muốn chọn con đường sự nghiệp trong lĩnh vực A.I. này, nắm rõ những điều cơ bản của TensorFlow thực sự rất quan trọng. Được viết bằng C++ và thao tác interface bằng Python nên phần performance của TensorFlow cực kỳ tốt. Đối tượng sử dụng nó cũng đa dạng không kém: từ các nhà nghiên cứu, nhà khoa học dữ liệu và dĩ nhiên không thể thiếu các lập trình viên.



TensorFlow

Hình 4.21: Logo Tensorflow

Kiến trúc TensorFlow hoạt động được chia thành 3 phần:

- Tiền xử lý dữ liệu

- Dựng model
- Train và ước tính model

Hoạt động của Tensorflow:

TensorFlow cho phép các lập trình viên tạo ra dataflow graph, cấu trúc mô tả làm thế nào dữ liệu có thể di chuyển qua 1 biểu đồ, hay 1 seri các node đang xử lý. Mỗi node trong đồ thị đại diện 1 operation toán học, và mỗi kết nối hay edge giữa các node là 1 mảng dữ liệu đa chiều, hay còn được gọi là ‘tensor’.

TensorFlow cung cấp tất cả những điều này cho lập trình viên theo phương thức của ngôn ngữ Python. Vì Python khá dễ học và làm việc, ngoài ra còn cung cấp nhiều cách tiện lợi để ta hiểu được làm thế nào các high-level abstractions có thể kết hợp cùng nhau. Node và tensor trong TensorFlow là các đối tượng Python, và các ứng dụng TensorFlow bản thân chúng cũng là các ứng dụng Python.

Các operation toán học thực sự thì không được thi hành bằng Python. Các thư viện biến đổi có sẵn thông qua TensorFlow được viết bằng các binary C++ hiệu suất cao. Python chỉ điều hướng lưu lượng giữa các phần và cung cấp các high-level abstraction lập trình để nối chúng lại với nhau.

Luận văn sử dụng thư viện Tensorflow phiên bản 1.15 cho chương trình phát hiện khuôn mặt.

4.4.3 Thư viện Numpy

Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng “core Python” đơn thuần.

NumPy được phát triển bởi Jim Hugunin. Phiên bản ban đầu là Numarray được phát triển, có một số chức năng bổ sung. Năm 2005, Travis Oliphant đã tạo ra gói NumPy bằng cách kết hợp các tính năng của Numarray và gói Numeric.

Sử dụng NumPy, lập trình viên có thể thực hiện các thao tác sau:

- Các phép toán toán học và logic trên mảng.
- Các biến đổi Fourier và các quy trình để thao tác shape.

- Các phép toán liên quan đến đại số tuyến tính. NumPy tích hợp sẵn các hàm cho đại số tuyến tính và tạo số ngẫu nhiên.

Luận văn sử dụng thư viện numpy biên bản 1.17 cho cả phát hiện khuôn mặt và màu đỏ.

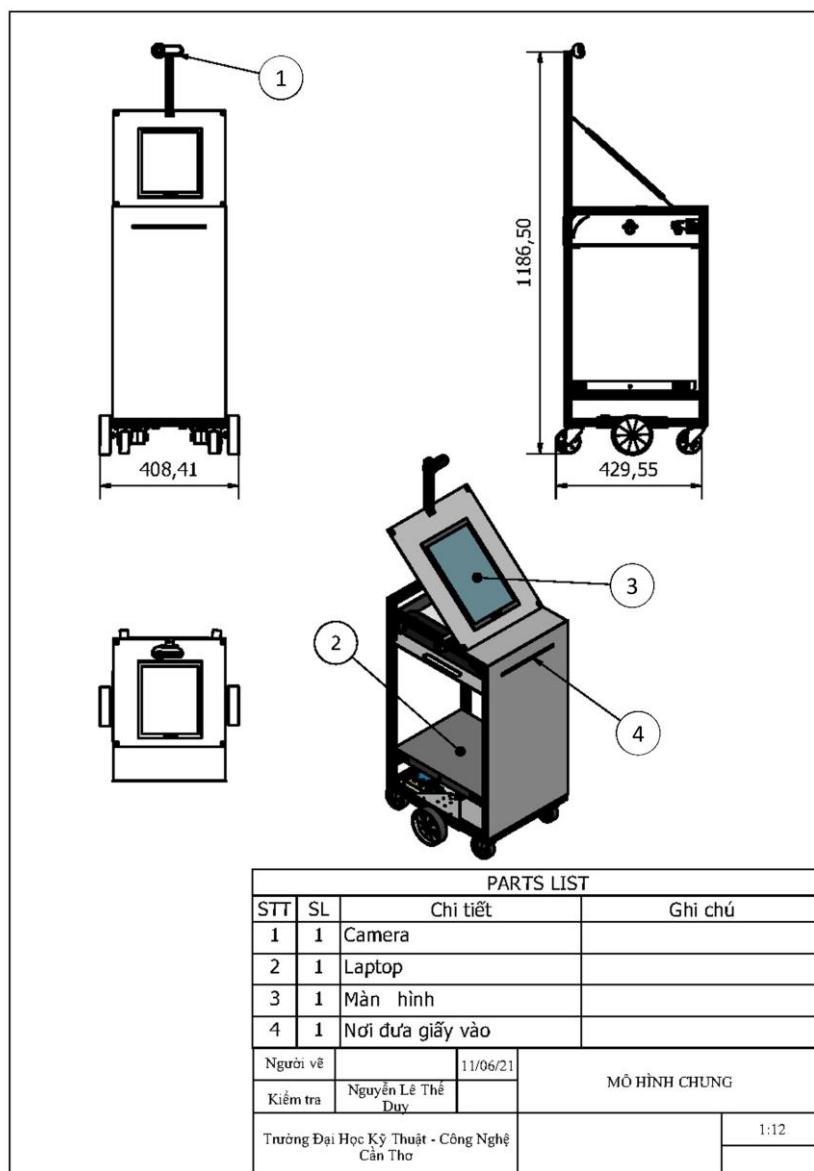
CHƯƠNG 5 : PHẦN KẾT HỢP ĐỀ TÀI

5.1 Thiết kế mô hình

5.1.1 Cơ sở thiết kế

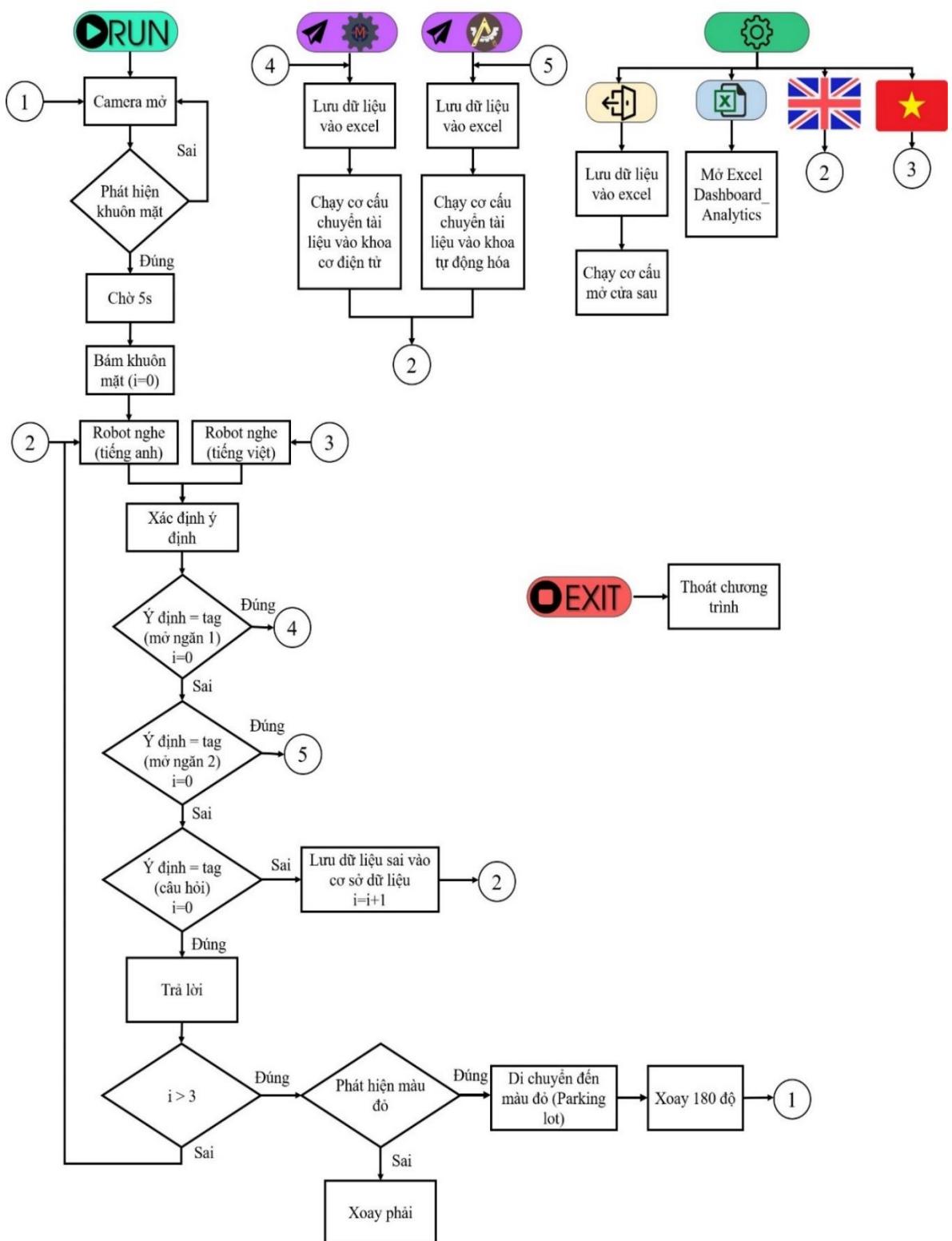
Để mô hình mang tính trực quan và tăng khả năng giao tiếp robot với người dùng thì phải đảm bảo độ cao phù hợp và phải phù hợp với độ tải của động cơ nên độ cao được tăng thêm 400mm.

5.1.2 Bản vẽ tổng thể



Hình 5.1: Bản vẽ tổng thể mô hình chung

5.2 Sơ đồ điều khiển



Hình 5.2: Sơ đồ điều khiển toàn hệ thống

Giải thích sơ đồ các nút điều khiển:

- ❖ **RUN:** Đầu tiên khi kích hoạt camera sẽ được khởi động tiếp đến chạy chương trình xử lý ảnh phát hiện khuôn mặt và bám theo đối tượng, nếu khuôn mặt trong khung hình đúng trong 5s robot sẽ bắt đầu bám theo và điều chỉnh tọa độ khuôn mặt vào giữa khung hình, đến khi đối tượng vào đúng khung hình chương trình xử lý phát hiện khuôn mặt kết thúc và chạy chương trình giọng nói.

Robot kích hoạt mic và chương trình xử lý giọng nói và dùng mô hình phân loại ý định để xác định câu trả lời đầu ra nếu không nghe hoặc không xác định được ý định robot sẽ báo không hiểu và lưu câu hỏi đó vào excel rồi bắt đầu lặp lại nghe lần nữa, ngược lại nếu xác định được ý định robot sẽ trả lời yêu cầu mở ngăn một hoặc hai thì robot sẽ trả lời và đồng thời lưu ảnh người gửi, dữ liệu vào excel và gửi tín hiệu xuống vi điều khiển để tiến hành hoạt động đến khi hoàn tất robot trở lại trạng thái nghe. Trong trường hợp nếu robot nghe quá 3 lần mà không thấy phản hồi robot tự động sẽ chuyển sang chế độ đỗ xe ở vị trí đã định.

Sau khi kết thúc cuộc trò chuyện, robot sẽ bắt đầu xoay phải và chạy chương trình phát hiện màu đỏ, đến khi phát hiện được đối tượng (parking lot), giống như phát hiện khuôn mặt robot di chuyển và cân chỉnh đối tượng và giữa khung hình, bám đến vị trí, sau khi đến đối tượng robot sẽ xoay 180 độ nhằm mục đích hướng robot ra ngoài cho tầm camera quan sát phát hiện đối tượng mới, cuối cùng quay lại chương trình xử lý khuôn mặt.

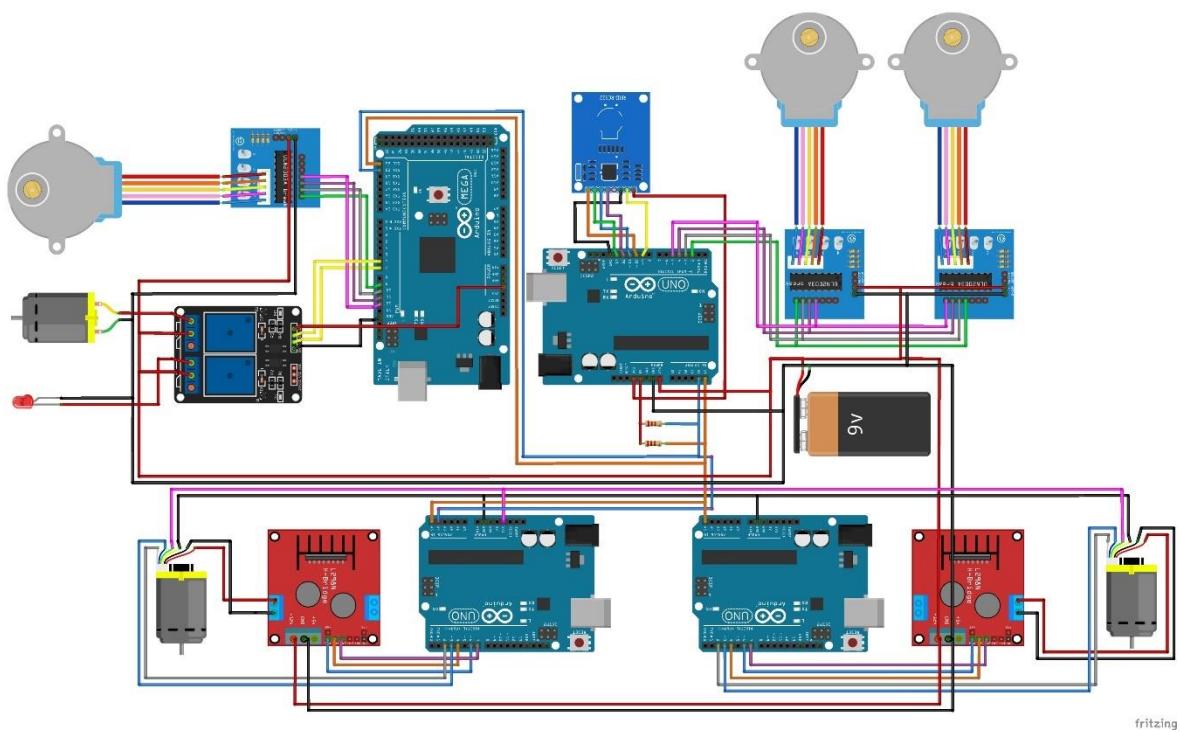
- ❖ **Mechatronic:** Khi kích hoạt chương trình sẽ lưu ảnh người gửi và lưu dữ liệu lên excel sau đó truyền dữ liệu xuống vi điều khiển để tiến hành chạy cơ cấu đưa giấy vào ngăn cơ điện tử khi kết thúc, robot trở lại trạng thái nghe.

- ❖ **Automation:** Tương tự như mechatronic khi kích hoạt sẽ lưu ảnh người gửi và lưu dữ liệu lên excel sau đó truyền dữ liệu xuống vi điều khiển để tiến hành chạy cơ cấu đưa giấy vào ngăn tự động hóa đến khi kết thúc, robot trở lại trạng thái nghe.

- ❖ **Setting:** Khi kích hoạt sẽ hiện ra giao diện chuyển ngôn ngữ và đăng nhập, vào excel xem dữ liệu hoặc mở cổng sau yêu cầu phải nhập đúng tài khoản được cấp.

- ❖ **EXIT:** Dừng và thoát tất cả chương trình.

Sơ đồ mạch điều khiển:



Hình 5.3: Sơ đồ đấu nối dây

Trung tâm điều khiển chính vẫn là Arduino Mega 2560, nhiệm vụ của master là nhận lệnh từ máy tính, truyền lệnh cho các slave đồng thời thực hiện việc thu nhận giấy tờ từ cổng vào, chuyển đổi các ngăn từ yêu cầu người gửi.

Slave công việc là chờ và nhận lệnh từ master, trong đó hai slave vẫn tiếp tục giữ vai trò điều khiển động cơ DC, slave còn lại đảm nhận vai trò mở cổng sau để người sử dụng lấy giấy tờ trong ngăn, đồng thời tích hợp thêm module RFID trong việc mở cổng sau.

5.3 Mô hình thực tế

Mô hình có kích thước (DxRxH) 430x408x1200 với chiều cao 1200mm phù hợp với người dùng tương tác với màn hình điều khiển, và thuận tiện gửi giấy tờ hành chính vào các ngăn của robot khi thực hiện thao tác cũng như giao tiếp với robot.

Robot có khối lượng xấp xỉ 9kg bao gồm toàn bộ hệ thống với tốc độ di chuyển 22 vòng/phút để quá trình dò tìm đối tượng ổn định và chính xác.



Hình 5.4: Mô hình kết hợp thực tế

5.4 Đánh giá

Mô hình khi kết hợp hoạt động tốt sau nhiều lần thực nghiệm, robot di chuyển ổn định, ít rung lắc đảm bảo cho tầm nhìn camera quan sát và phát hiện đối tượng hiệu quả, giao tiếp tốt với người dùng khi tương tác, các ngăn chứa tài liệu hoạt động tốt.

Robot quản lý dữ liệu người sử dụng bao gồm người gửi tài liệu và người quản lý truy cập vào cửa lấy tài liệu được lưu thông tin vào excel, hình ảnh người gửi được lưu vào thư mục quản lý được thể hiện như (hình 5.5)

CHƯƠNG 5: PHÂN KẾT HỢP ĐỀ TÀI

1	STT	Day	Month	Recording Time	Pic-Acc	Faculty
2	1	22	Tháng 06	22/06/2021 10:13	CDT-22_06_2021_10_13_45	Khoa CDT
3	2	22	Tháng 06	22/06/2021 10:46	CDT-22_06_2021_10_46_40	Khoa CDT
4	3	22	Tháng 06	22/06/2021 11:25	CDT-22_06_2021_11_25_44	Khoa CDT
5	4	22	Tháng 06	22/06/2021 11:40	CDT-22_06_2021_11_40_30	Khoa TDH
6						
7						
8						



CDT-22_06_2021_10_13_45.jpg



CDT-22_06_2021_10_46_40.jpg



CDT-22_06_2021_11_25_44.jpg



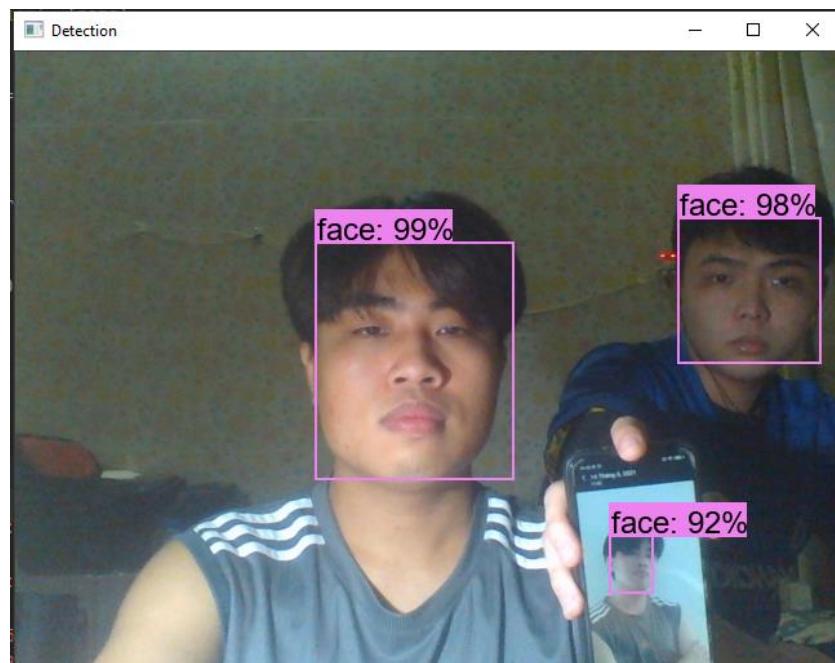
TDH-22_06_2021_11_40_30.jpg

Hình 5.5: Dữ liệu được lưu trong thư mục

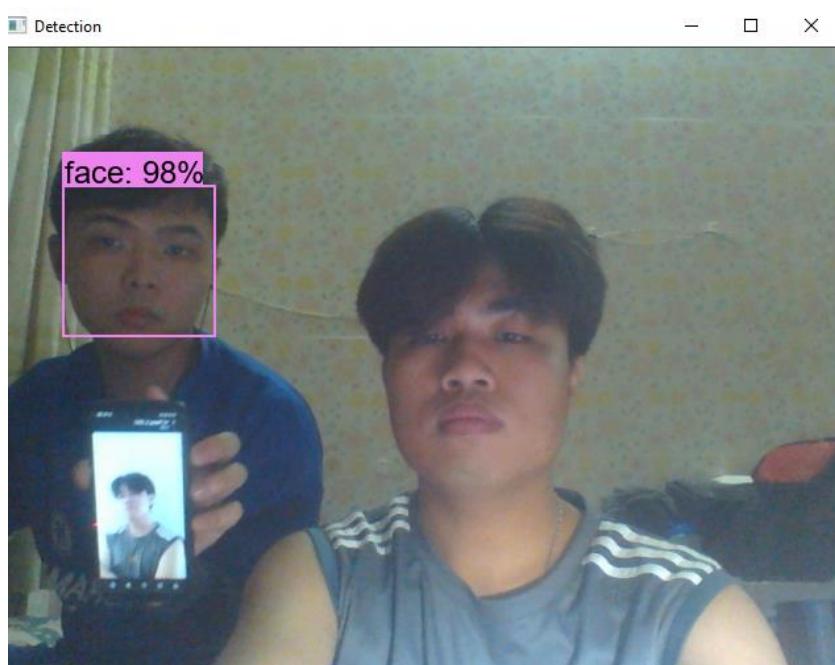
CHƯƠNG 6 : KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết quả

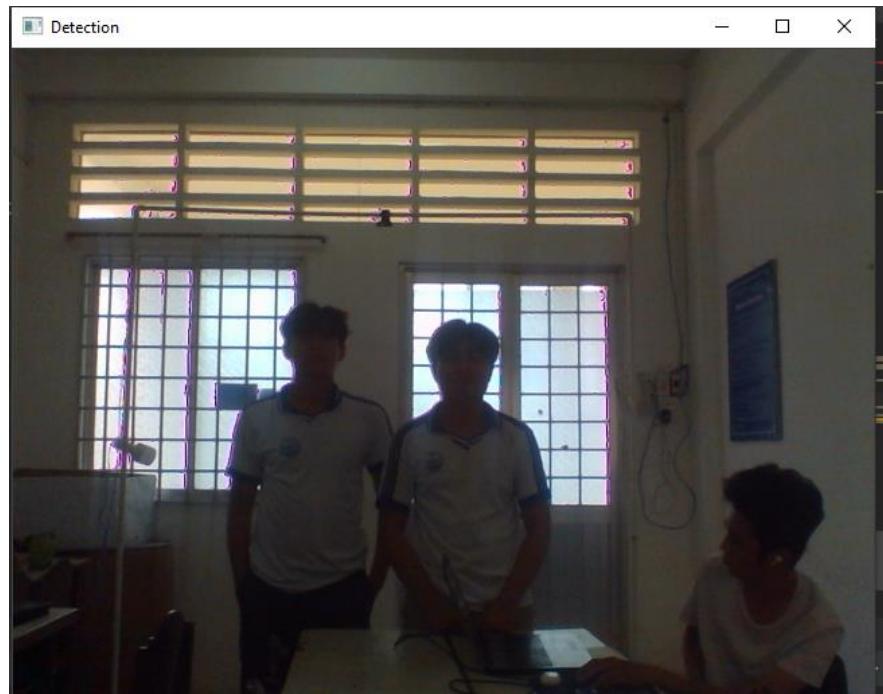
6.1.1 Kết quả phát hiện khuôn mặt



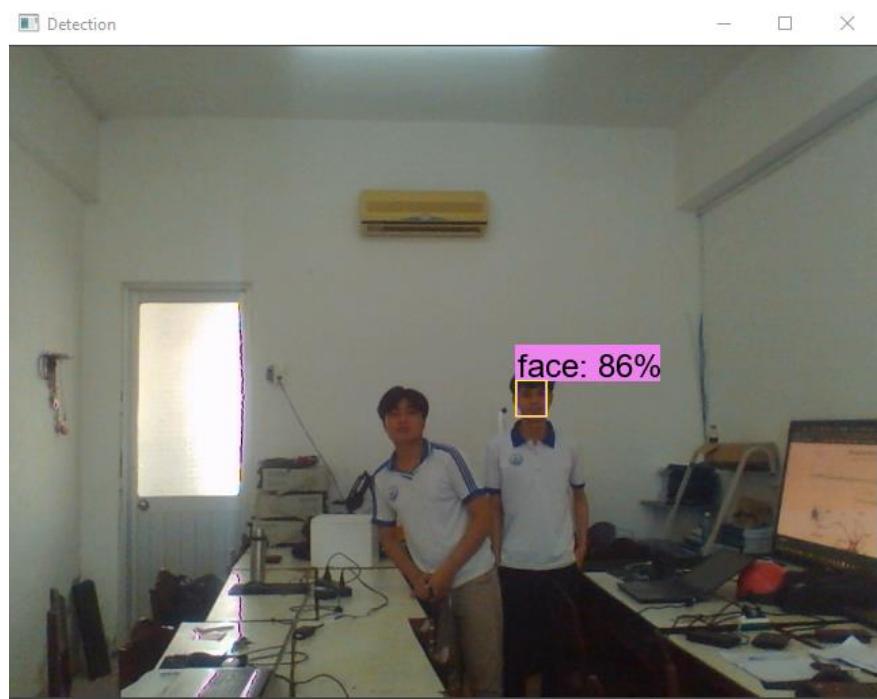
Hình 6.1: Kết quả phát hiện khuôn mặt chưa trích lọc



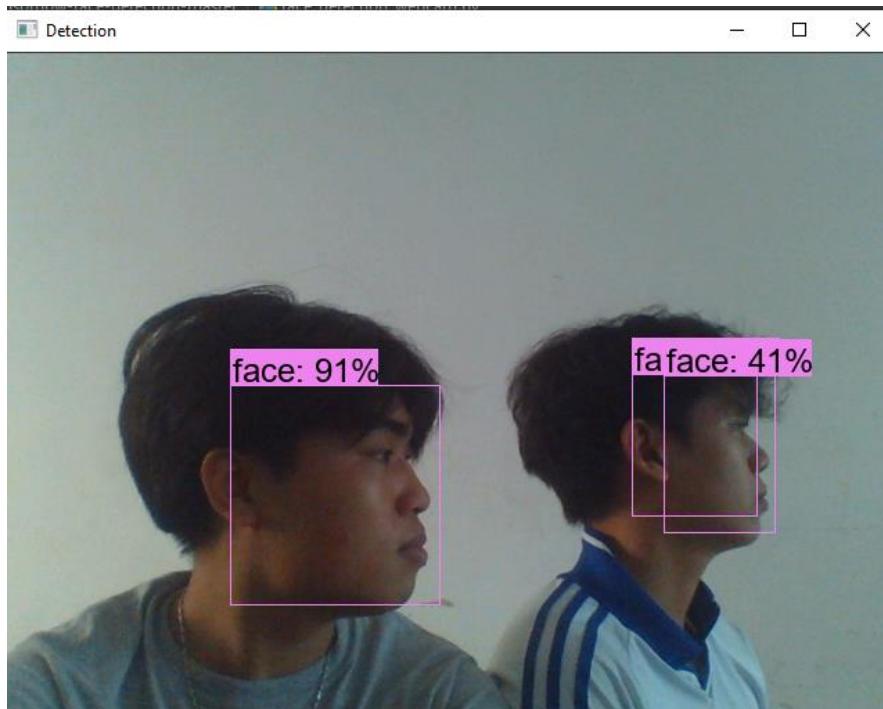
Hình 6.2: Kết quả phát hiện khuôn mặt phần trăm cao nhất



Hình 6.3: Kết quả phát hiện khuôn mặt ngược sáng



Hình 6.4: Kết quả phát hiện khuôn mặt khoảng cách 5m



Hình 6.5: Kết quả phát hiện khuôn mặt góc nghiêng

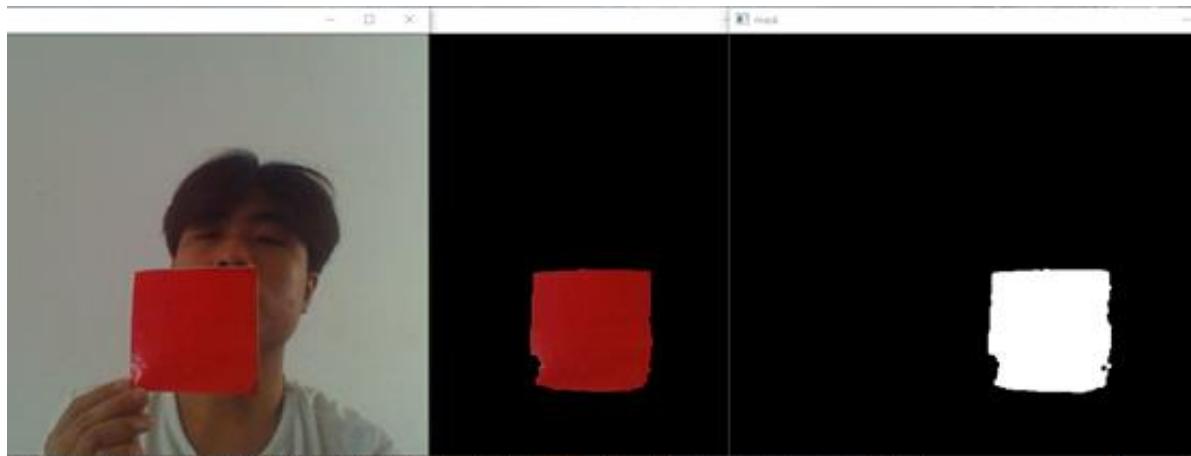
Bảng 6.1: Kết quả thực nghiệm phát hiện khuôn mặt

Số lần thực nghiệm	Dự đoán phần trăm khuôn mặt được phát hiện			
	Trực diện (5m)	Ngược sáng	Góc nghiêng (1m)	Trực diện (1m)
1	75%	0%	41%	97%
2	79%	0%	52%	98%
3	86%	0%	60%	99%
4	82%	0%	76%	96%
5	70%	0%	91%	97%

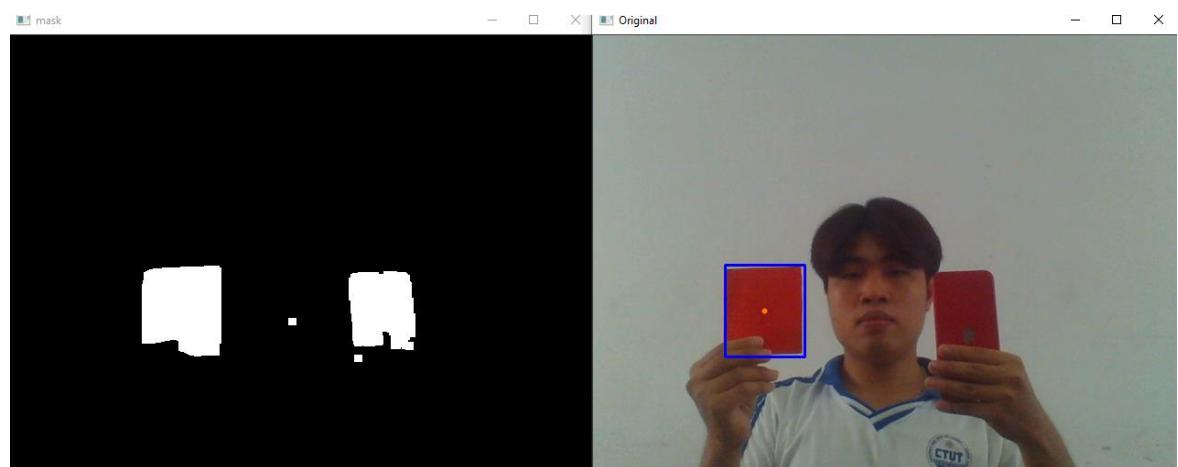
Bảng trên cho thấy trung bình khi phát hiện khuôn mặt trong khung hình khi thực nghiệm 5 lần là:

- Trực diện (5m): 78.5 %
- Ngược sáng: 0%
- Góc nghiêng: 64%
- Trực diện: 97.4 %

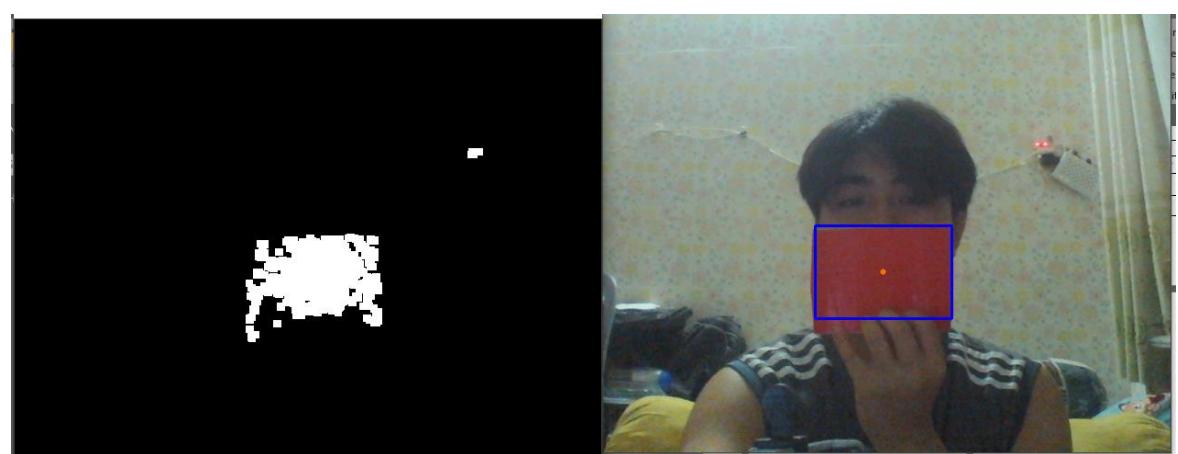
6.1.2 Kết quả phát hiện màu



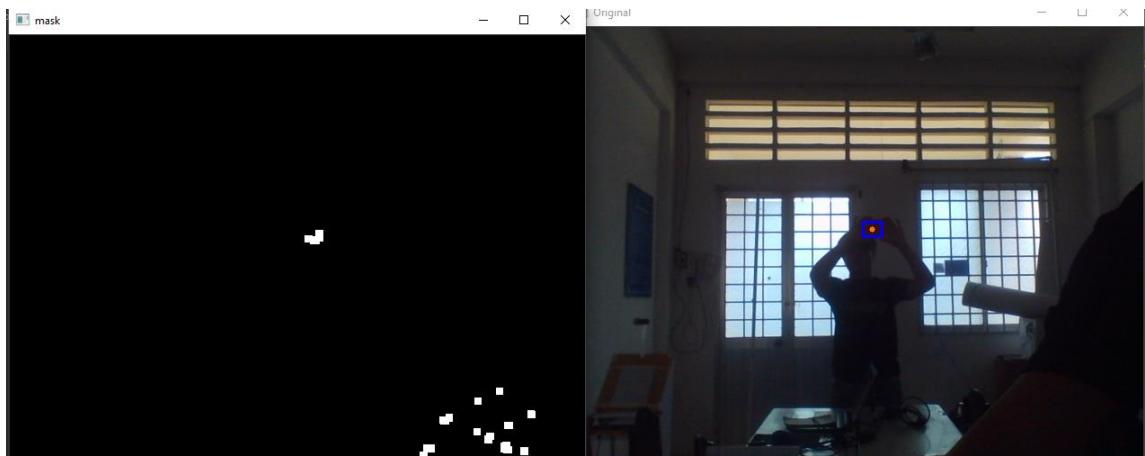
Hình 6.6: Kết quả tinh chỉnh giá trị vùng màu đỏ



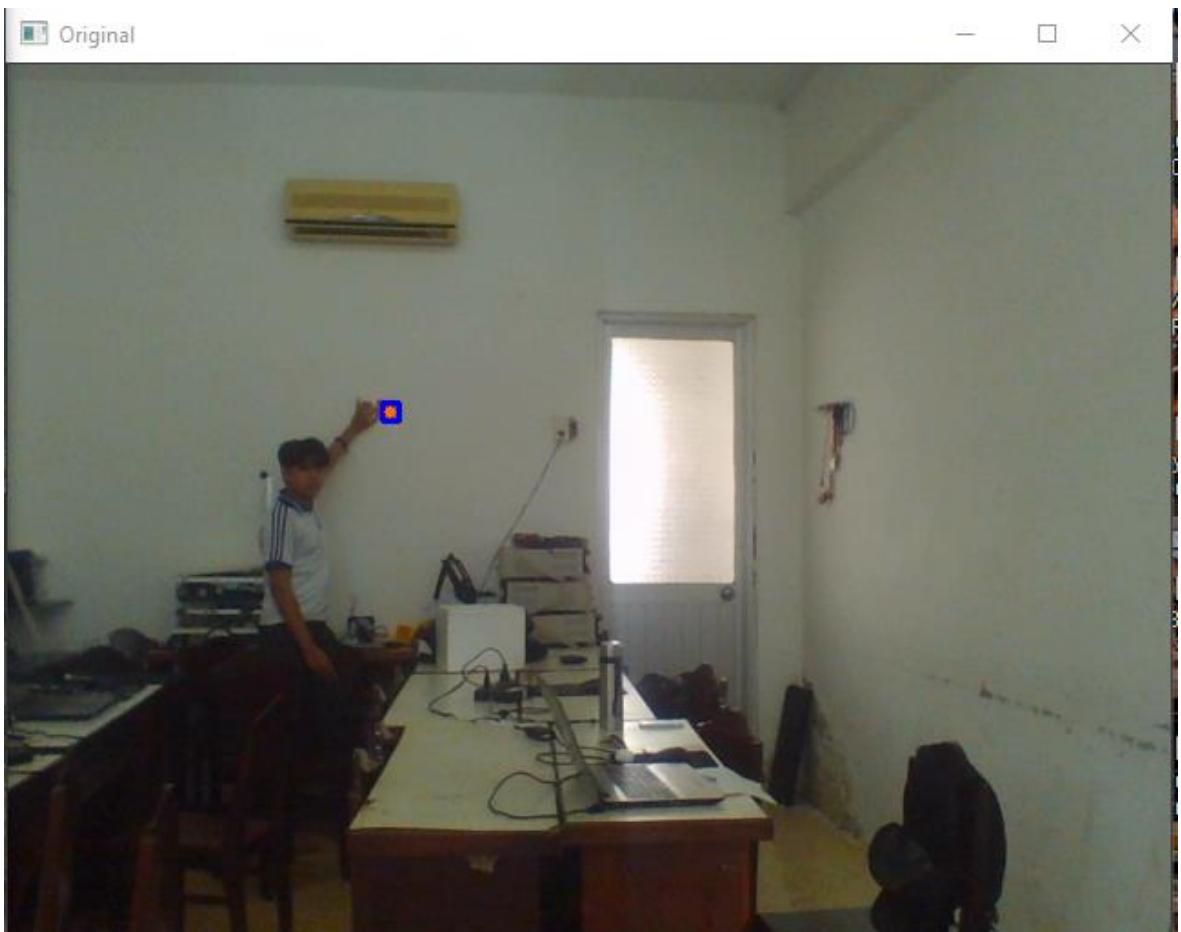
Hình 6.7: Kết quả phát hiện màu đỏ buổi sáng



Hình 6.8: Kết quả phát hiện màu đỏ buổi tối



Hình 6.9: Kết quả phát hiện màu đỏ ngược sáng



Hình 6.10: Kết quả phát hiện màu đỏ ở khoảng cách 5m

6.1.3 Điều khiển mobile robot



Hình 6.11: Mô hình Mobile robot hoàn chỉnh

Vì điều khiển giao tiếp khá tốt từ máy tính thông qua hai chương trình xử lý ảnh, phát hiện và bám sát đối tượng.

6.2 Đánh giá

❖ Phát hiện khuôn mặt

Mô hình phát hiện khuôn mặt sử dụng Tensorflow tốt so với những phát hiện dùng các phát hiện trước đây, tuy nhiên vẫn còn nhầm lẫn trong khi phát hiện và bài toán ánh sáng.

Nhìn chung phát hiện khuôn mặt trong đề tài sử dụng khá tốt trong việc điều khiển và di chuyển mobile robot.

Phát hiện khuôn mặt chỉ thực hiện tốt khi phát hiện với khoảng cách dưới 5m.

❖ Phát hiện màu đỏ

Phát hiện màu đỏ dùng không gian màu HSV và YCbCr giúp nhận diện màu đỏ và “gần đỏ” khá tốt, phát hiện được cả buổi sáng và tối nếu ánh sáng thích hợp, tuy nhiên yếu tố ánh sáng vẫn là mấu chốt giới hạn việc phát hiện màu.

Phát hiện màu đỏ và giao tiếp xuống vi điều khiển điều khiển cho thấy mobile robot hoạt động tốt hơn so với phát hiện khuôn mặt.

❖ Di chuyển mobile robot

Giải thuật PID giúp cho robot di chuyển với tốc độ ổn định, tuy nhiên do nhiều yếu tố khác nhau từ khu băm xung, môi trường,... làm cho robot khi di chuyển thường bị lệch về phía trái hoặc phải.

6.3 Kết luận

6.3.1 Kết quả đạt được

- Xây dựng thiết kế mô hình 3D mobile robot trên phần mềm Autodesk Inventor.
- Chế tạo thành công mô hình mobile robot thực tế.
- Thiết kế mạch điều khiển cho mobile robot.
- Ứng dụng bộ điều khiển PID vào điều khiển tốc độ cho động cơ DC.
- Giao tiếp cổng COM thành công giữa máy tính và Arduino.
- Giao tiếp I2C giữa các board Arduino với nhau.
- Chương trình xử lý phát hiện khuôn mặt và màu đỏ dùng tọa độ và điều khiển vi điều khiển thành công.
- Mobile robot di chuyển và điều chỉnh hướng bám sát đối tượng được phát hiện trong khung hình.

6.3.2 Các hạn chế của đề tài

- Mobile robot chỉ di chuyển được trên các mặt phẳng, chưa tránh được các vật cản bên dưới robot.
- Hệ thống xử lý ảnh chưa xử lý được phát hiện nhiều đối tượng trong khung hình.
- Xử lý phát hiện khuôn mặt còn bị hạn chế bởi góc nghiêng.
- Mobile robot khi hoạt động tạo ra tiếng ồn từ động cơ.

6.4 Hướng phát triển

- Cải tiến mô hình mobile robot và động cơ có công suất cao hơn giúp ổn định khi di chuyển, linh hoạt với địa hình gồ ghề, nhấp nhô,...
- Trang bị thêm cảm biến phát hiện vật cản giúp robot di chuyển linh hoạt
- Đưa khung hình phát hiện lên Internet, kết hợp điều khiển mobile robot bằng tay trên Internet và điện thoại nhằm ứng dụng di chuyển quan sát từ xa.

- Thiết kế thêm tay gấp vật trên mobile robot, nâng cao ứng dụng vào các lĩnh vực khác nhau.
- Bộ dữ liệu khuôn mặt được đào tạo trực tiếp khi phát hiện khuôn mặt mới nhằm nâng cao khả năng nhận biết.
- Dùng máy tính nhúng (raspberry) nhỏ gọn để thực hiện các công việc của máy tính hiện tại.

TÀI LIỆU THAM KHẢO

- [1] Trịnh Tuấn Dương (2020), Luận Văn Thạc Sĩ : “*Nghiên Cứu, Thiết Kế Chế Tạo Mobile Robot Tự Hành Tích Hợp Một Số Công Nghệ Xử Lý Ảnh Và Thuật Toán Hiện Đại*”, Hà Nội.
- [2] PGS.TS Nguyễn Quang Hoan (2006), *Giáo trình Xử lý ảnh*, Học viện công nghệ bưu chính viễn thông.
- [3] TS. Nguyễn Ngọc Phương(2005), Hướng Dẫn Thiết Kế & Lắp Ráp ROBOT, Nhà Xuất Bản Đà Nẵng
- [4] PGS.TS Nguyễn Trường Thịnh (2014), *Giáo Trình Kỹ Thuật Robot*, Nhà Xuất Bản Đại Học Quốc Gia Thành Phố Hồ Chí Minh.
- [5] Mã Trường Thành, Phạm Nguyên Khang, Lâm Chí Nguyên, Triệu Thanh Ngoan, Hà Duy An, Đỗ Thanh Nghị (2013), “*Điều Khiển Robot Pioneer P3-Dx Bám Sát Đối Tượng*”, *Tạp chí khoa học Trường Đại học Cần Thơ*, (Số chuyên đề: Công nghệ Thông tin (2013), 11-19.

Các website tham khảo:

- [6] <http://docs.opencv.org/>
- [7] <https://www.arduino.cc/>
- [8] <https://github.com/tensorflow>
- [9] <https://www.instructables.com/Object-Finding-Personal-Assistant-Robot-Ft-Raspber/#discuss>
- [10] <https://github.com/tensorflow/models>
- [11] <https://www.w3schools.com/python/>
- [12] <http://Arduino.vn/bai-viet/1053-giao-tiep-i2c-voi-nhieu-module>
- [13] <https://arxiv.org/pdf/1801.04381.pdf> (MobileNetV2)
- [14] <https://arxiv.org/pdf/1512.02325.pdf> (SSD)
- [15] <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/>

PHỤ LỤC

Hướng dẫn cài đặt môi trường Tensorflow và phát hiện đối tượng:

<https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/>

Code phát hiện khuôn mặt và phát hiện màu:

Link google drive:

https://drive.google.com/file/d/1fzy_ZSYJiqyor9uKvCPi4pU7KSnjOCVX/view?usp=sharing

QR code:



Code kết hợp đè tài:

Link google drive:

<https://drive.google.com/drive/folders/1SvicVinguXFDEhDMt2Ef1LmmvJPo3cBP?usp=sharing>

QR code:

