TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ THI CÔNG MÔ HÌNH XE TỰ HÀNH

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **ĐẶNG PHÚC HƯNG**
MSSV: 15119093
NGÔ HỮU HẬU
MSSV: 15119086
HOÀNG CHIẾM THÀNH
MSSV: 15119128

TP. HỒ CHÍ MINH – 06/2019

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

THIẾT KẾ VÀ THI CÔNG MÔ HÌNH XE TỰ HÀNH

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **ĐẶNG PHÚC HƯNG**
MSSV: 15119093
NGÔ HỮU HẬU
MSSV: 15119086
HOÀNG CHIẾM THÀNH
MSSV: 15119128

Hướng dẫn**: THS. HUỲNH HOÀNG HÀ**

TP. HỒ CHÍ MINH – 06/2019

THÔNG TIN KHÓA LUẬN TỐT NGHIỆP

1. Thông tin sinh viên

Họ và tên sinh viên: Đặng Phúc Hưng MSSV: 15119093

Email: dphung.ute@gmail.com Điện thoại: 0968357400

Họ và tên sinh viên: Hoàng Chiếm Thành MSSV: 15119128

Email: h.chiemthanh@gmail.com Điện thoại: 0968271550

Họ và tên sinh viên: Ngô Hữu Hậu MSSV: 15119086

Email: nhhauuu@gmail.com Điện thoại: 0961838625

2. Thông tin đề tài

- Tên của đề tài: Thiết Kế Và Thi Công Mô Hình Xe Tự Hành
- Đơn vị quản lý: Bộ môn Kỹ Thuật Máy Tính - Viễn Thông, Khoa Điện Điện Tử, Trường Đại Học Sư Phạm Kỹ Thuật Tp. Hồ Chí Minh.
- Thời gian thực hiện: Từ ngày 20 / 02 / 2019 đến ngày 15 / 06 / 2019
- Thời gian bảo vệ trước hội đồng: Ngày 18-19 / 06 / 2019

3. Lời cam đoan của sinh viên

Chúng tôi – Đặng Phúc Hưng, Hoàng Chiếm Thành và Ngô Hữu Hữu Hậu cam đoan KLTN là công trình nghiên cứu của bản thân chúng tôi dưới sự hướng dẫn của thạc sỹ Huỳnh Hoàng Hà. Kết quả công bố trong KLTN là trung thực và không sao chép từ bất kỳ công trình nào khác.

Tp.HCM, ngày 06 tháng 06 năm 2019

SV thực hiện đồ án

(*Ký và ghi rõ họ tên*)

Đặng Phúc Hưng Ngô Hữu Hậu Hoàng Chiếm Thành

Giảng viên hướng dẫn xác nhận quyển báo cáo đã được chỉnh sửa theo đề nghị được ghi trong biên bản của Hội đồng đánh giá Khóa luận tốt nghiệp.

……

Xác nhận của Bộ Môn

Tp.HCM, ngày … tháng … năm 20…

Giáo viên hướng dẫn

(Ký, ghi rõ họ tên và học hàm - học vị)

BẢN NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(Dành cho giảng viên hướng dẫn)

Đề tài: **Thiết Kế Và Thi Công Mô Hình Xe Tự Hành**

Sinh viên thực hiện: 1. Đặng Phúc Hưng MSSV: 15119093
2. Hoàng Chiếm Thành MSSV: 15119128
3. Ngô Hữu Hậu MSSV: 15119086

Giảng viên hướng dẫn: Ths. Huỳnh Hoàng Hà

Nhận xét bao gồm các nội dung sau đây:

1. Tính hợp lý trong cách đặt vấn đề và giải quyết vấn đề; ý nghĩa khoa học và thực tiễn:
Đặt vấn đề rõ ràng, mục tiêu cụ thể; đề tài có tính mới, cấp thiết; đề tài có khả năng ứng dụng, tính sáng tạo.
Đặt vấn đề hợp lý

2. Phương pháp thực hiện/ phân tích/ thiết kế:
Phương pháp hợp lý và tin cậy dựa trên cơ sở lý thuyết; có phân tích và đánh giá phù hợp; có tính mới và tính sáng tạo.
Đề tài có tính mới

3. Kết quả thực hiện/ phân tích và đánh giá kết quả/ kiểm định thiết kế:
Phù hợp với mục tiêu đề tài; phân tích và đánh giá kiểm tra thiết kế hợp lý; có tính sáng tạo kiểm định chặt chẽ và đảm bảo độ tin cậy.
Kết quả phù hợp với mục tiêu

4. Kết luận và đề xuất:
Kết luận phù hợp với cách đặt vấn đề; đề xuất mang tính cải tiến và thực tiễn; kết luận có đóng góp mới mẻ, đề xuất sáng tạo và thuyết phục.
Kết luận phù hợp

5. Hình thức trình bày và bố cục báo cáo:
Văn phong nhất quán, bố cục hợp lý, cấu trúc rõ ràng, đúng định dạng mẫu; có tính hấp dẫn, thể hiện năng lực tốt, văn bản trau chuốt.
Bố cục hợp lý

6. Kỹ năng chuyên nghiệp và tính sáng tạo:
Thể hiện các kỹ năng giao tiếp, kỹ năng làm việc nhóm, và các kỹ năng chuyên nghiệp khác trong việc thực hiện đề tài.
Làm việc nhóm tốt

7. Tài liệu trích dẫn
Tính trung thực trong việc trích dẫn tại liệu tham khảo; tính phù hợp của các tài liệu trích dẫn, trích dẫn theo đúng chỉ dẫn APA.
Trích dẫn đúng mẫu

8. Đánh giá về sự trùng lặp của đề tài
Cần khẳng định đề tài có trùng lặp hay không? Nếu có, đề nghị ghi rõ mức độ, tên đề tài, nơi công bố, năm công bố của đề tài đã công bố.
Không trùng lặp

9. Những nhược điểm và thiếu sót, những điểm cần được bổ sung và chỉnh sửa* Còn một số lỗi trong cuốn báo cáo và hệ thống

10. Nhận xét tinh thần, thái độ học tập, nghiên cứu của sinh viên
Siêng năng, có cố gắng

Đề nghị của giảng viên hướng dẫn
Ghi rõ: "Báo cáo đạt không đạt yêu cầu của một khóa luận tốt nghiệp kỹ sư, và được phép không được phép bảo vệ khóa luận tốt nghiệp"
Được phép bảo vệ

Tp. HCM, ngày 06 tháng 06 năm 2019
Người nhận xét
(Ký và ghi rõ họ tên)

Huỳnh Hoàng Hà

* *Giáo viên hướng dẫn có thể ghi tiếp vào mặt sau*

LỜI CẢM ƠN

Trên thực tế, không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, chúng tôi đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, chúng tôi xin gửi đến toàn thể quý thầy cô Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM nói chung và các thầy cô giáo trong Khoa Điện - Điện Tử nói riêng, đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức nền tảng và kiến thức chuyên ngành quan trọng, giúp nhóm chúng tôi có được cơ sở lý thuyết để thực hiện đồ án này.

Chúng tôi xin chân thành cảm ơn thầy Huỳnh Hoàng Hà đã tận tâm hướng dẫn chúng tôi trong suốt quá trình thực hiện đồ án. Những buổi nói chuyện, thảo luận cùng thầy giúp chúng tôi có những hướng đi đúng đắn trong quá trình nghiên cứu. Một lần nữa, chúng tôi xin chân thành cảm ơn thầy.

Mặc dù đề tài lần này dựa trên mô hình thực tế, nhưng với kiến thức của chúng tôi còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, chúng tôi rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để kiến thức của chúng tôi trong lĩnh vực này được hoàn thiện hơn.

Sau cùng, chúng tôi xin kính chúc quý thầy cô thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Xin chân thành cám ơn!

Tp.HCM, ngày 06 tháng 06 năm 2019

SV thực hiện đồ án
(*Ký và ghi rõ họ tên*)

Đặng Phúc Hưng Ngô Hữu Hậu Hoàng Chiếm Thành

TÓM TẮT

Ngày nay với sự phát triển của khoa học công nghệ, nhất là trong lĩnh vực điều khiển và tự động, rất nhiều máy, hệ thống tự động thông minh ra đời, đã làm thay dổi về mọi mặt cuộc sống của con ngời. Cụ thể trong công nghiệp là các máy tự động, các trạm sản xuất linh hoạt, các nhà máy thông minh… Trong dân dụng là các thiết bị phục vụ trong các gia đình như máy giặt, tủ lạnh, các hệ thống bảo vệ, chiếu sáng tự động… Trong trao đổi, mua án cũng xuất hiện các thiết bị tự động như máy ATM, các máy bán cà phê…. Với những chiếc máy này việc mua bán, trao đổi của con người trở nên thuận tiện hơn. Trong báo cáo này, nhóm muốn đề cập đến một trong những chiếc máy tự động ứng dụng trong giao thông, đó là mô hình xe tự hành.

Đề tài "Thiết kế và thi công mô hình xe tự hành" – ý tưởng từ những chiếc ô tô tự lái, được nhóm nghiên cứu và thi công trên mô hình xe tự hành chạy trên môi trường lý tưởng. Mô hình này sử dụng mô hình Deep Learning với giải thuật Convolutional Nẻual Network (CNN). Áp dụng thuật toán trên nhóm đưa vào đề tài để xử lý nhận dạng làn đường, biển báo dừng và đèn tín hiệu giao thông. Dưới đây là nghiên cứu chi tiết hơn và kết quả cụ thể được nhóm thu thập trong thời gian thực hiện đồ án.

MỤC LỤC

DANH MỤC HÌNH

DANH MỤC BẢNG

CÁC TỪ VIẾT TẮT

ANN: Artificial Neural Network (mạng nơ-ron nhân tạo).

CNNs: Convolutional Neural Network(mạng nơ-ron tích chập).

CNTK: The Microsoft Cognitive Toolkit (bộ công cụ nhận thức của Microsoft)

ReLU: Rectified Linear Unit

CHƯƠNG 1

GIỚI THIỆU

1.1 GIỚI THIỆU

Xu hướng hiện nay là hiện đại hoá các hệ thống trong công nghiệp lẫn các thiết bị trong đời sống. Nói cách khác, các thiết bị đang ngày càng số hoá để đáp ứng chất lượng cho hệ thống và dễ dàng điều khiển hoặc sử dụng.

Trong cuộc cách mạng công nghiệp lần thứ 4, hầu hết các quốc gia đều đã và đang tập trung vào một lĩnh vực tuy mới mà cũ, đó chính là Trí tuệ nhân tạo. Lĩnh vực này ngày nay đang được ứng dụng ở khắp mọi nơi trong đời sống mà gần đây nhất là xe tự hành. Hàng loạt ông lớn như Google, Apple, Tesla, Uber, BWM, Honda, Ford... cùng hàng ngàn công ty khác đang tham gia nghiên cứu và phát triển công nghệ xe tự hành. Trong 5 năm qua, tổng số tiền đầu tư vào lĩnh vực này đã vượt quá 50 tỉ USD hay nói cách khác cứ mỗi ngày thì lĩnh vực này lại được đầu tư 30 triệu USD. Theo số liệu thống kê, 90% số vụ tai nạn liên quan đến xe tải loại lớn ở Mỹ là do lỗi của tài xế. Theo Morgan Stanley- một công ty tài chính lớn của Mỹ đã dự đoán việc sử dụng loại xe tự hành có thể giảm 90% số vụ tai nạn, cứu sống 30.000 người và giảm 2,12 triệu người bị thương mỗi năm, tiết kiệm 38 giờ di chuyển, 134 tỷ USD tiền xăng mỗi năm trên lãnh thổ nước Mỹ. Đây là một con số biết nói để cho thấy tầm quan trọng của xe tự hành trong đời sống tương lai. Chính những điều này đã thôi thúc nhóm thiết kế một chiếc xe tự hành có thể nhận biết được đèn giao thông, biển báo và đi đúng làn đường.

Với mong muốn mang kiến thức bản thân vào thực tiễn, nhóm đã chọn đề tài "Thiết kế và thi công mô hình xe tự hành" làm đề tài nghiên cứu đồ án.

1.2 MỤC TIÊU CỦA ĐỀ TÀI

Mục tiêu chính của đề tài là thiết kế ra một chiếc xe có khả năng tự đi theo làn chỉ định. Xe sử dụng camera Pi v1 để lấy ảnh đầu vào, sau đó đưa ảnh vào Raspberry Pi 3 để tiến hành xử lý bằng một số thuật toán xử lý ảnh và máy học. Xe chỉ được phép đi trong một làn cố định.

Khi nghiên cứu thực hiện đề tài, nhóm muốn áp dụng những kiến thức của bản thân về vi điều khiển, ngôn ngữ lập trình nhằm tạo ra những sản phẩm, thiết bị có khả năng ứng dụng trong thực tế.

Ngoài ra quá trình nghiên cứu thực hiện đề tài là cơ hội để nhóm tự kiểm tra lại những kiến đã được học ở trường, đồng thời phát huy tính sáng tạo, khả năng giải quyết một vấn đề theo yêu cầu đặt ra.

1.3 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đề tài Thiết Kế Và Thi Công Xe Tự Hành được thực hiện với các đối tượng được đưa vào nghiên cứu ứng dụng như:

- Raspberry Pi 3 Model B+.
- Raspberry Pi Camera v1.
- Module điều khiển động cơ L298N.
- Servo SG90.

Phạm vi ngiên cứu của đề tài giới hạn trong việc thiết kế mô hình xe có thể nhận biết được đèn giao thông, biển báo và đi đúng làn đường.

1.4 BỐ CỤC

Nôi dung gồm có 5 phần chính:

Chương 1: Giới thiệu đề tài: Nêu tầm quan trọng của đề tài, tình hình hiện nay và giới thiệu sơ lược nội dung cho người xem nắm rõ.

Chương 2: Cở sở lý thuyết và tổng quan về đề tài: Trình bày tổng quan về cơ sở lý thuyết các đối tượng được sử dụng trong hệ thống.

Chương 3: Phân tích và thiết kế hệ thống: Từ yêu cầu đề tài, lựa chọn các linh kiện và cách kết nối chúng. Đề ra phương pháp lập trình, lưu đồ giải thuật và nguyên lý làm việc của hệ thống.

Chương 4: Kết quả và đánh giá: Kết quả tổng quan sau khi thực hiện đề tài.

Chương 5 : Kết luận và hướng phát triển: Đưa ra các kết luận về các vấn đề đã hoàn thành và chưa hoàn thành, các mặc hạn chế. Trình bày hướng phát triển của đề tài.

CHƯƠNG 2

TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

2.1 CÁC MÔ-ĐUN VÀ LINH KIỆN CHÍNH

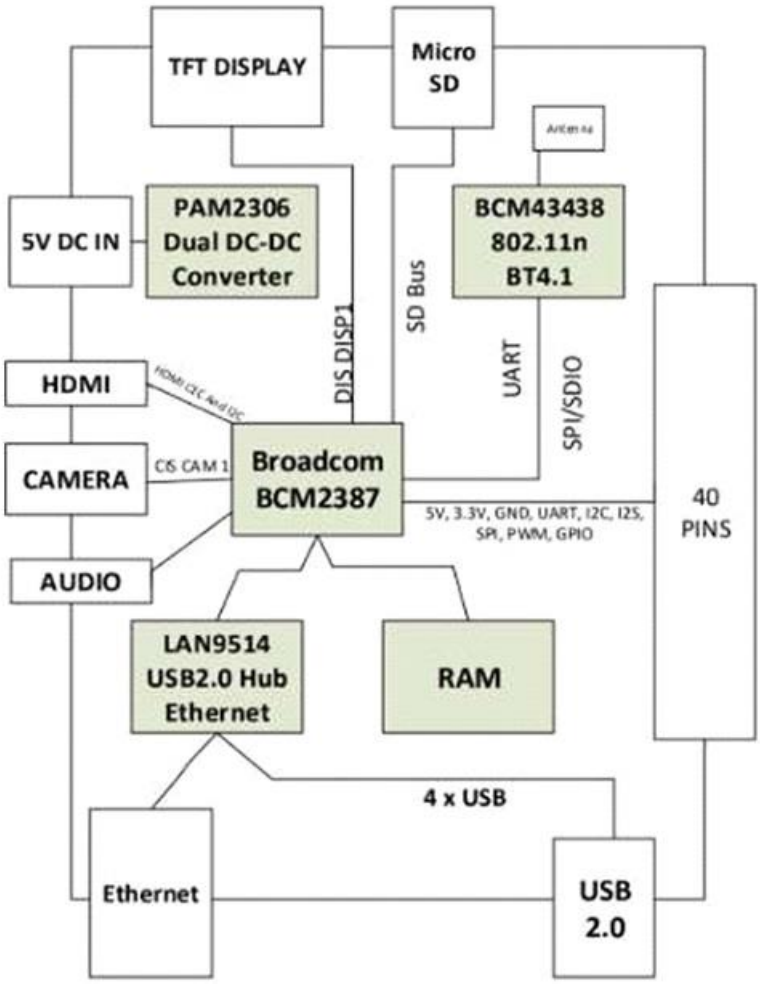
2.1.1 Máy tính nhúng Raspberry Pi [2]

Raspberry Pi là một chiếc máy tính tí hon giá chỉ từ 35$ chạy hệ điều hành Linux ra mắt vào tháng 2 năm 2012. Ban đầu Raspberry Pi được phát triển dựa trên ý tưởng tiến sĩ Eben Upton tại đại học Cambridge muốn tạo ra một chiếc máy tính giá rẻ để học sinh có thể dễ dàng tiếp cận và khám phá thế giới tin học. Dự định khiêm tốn của ông đến cuối đời là có thể bán được tổng cộng 1000 bo mạch cho các trường học. Vậy thì điều gì đã làm nên thành công ngoài sức tưởng tượng của Raspberry Pi khi đã bán được hơn một triệu bo mạch chỉ trong vòng chưa đầy một năm.

Hình 2.1: Raspberry Pi 3 Model B.

Hình 2.1 là một Raspberry phiên bản Raspberry Pi 3 Module B. Raspberry Pi là một máy tính siêu nhỏ, chỉ có kích thước như 1 chiếc thẻ ATM rút tiền. Bạn

chỉ cần 1 bàn phím, 1 tivi hoặc 1 màn hình có cổng HDMI/DVI, 1 nguồn USB 5V và 1 dây micro USB là đã có thể sử dụng RPi như 1 máy tính bình thường. Với Pi, bạn có thể sử dụng các ứng dụng văn phòng, nghe nhạc, xem phim độ nét cao (tới 1024p)...



Hình 2.2: Cấu trúc của Raspberry Pi 3 Model B.

Hình 2.2 thể hiện tổng quan về cấu trúc của một Raspberry Pi 3 Model B. Raspberry Pi 3 Model B hoạt động với vi xử lý Broadcom BCM2837 chạy ở tốc độ 1.2GHz có kiến trúc ARM Cortex-A53 64-bit cùng với 1 Gb RAM. Đây là vi xử lý SoC (system-on-chip) tức là hầu hết mọi thành phần của hệ thống gồm CPU, GPU cũng như audio, communication chip đều được tích hợp trong một. Bởi vì RAM được tích hợp sẵn trong đế chip nên không thể nâng cấp RAM cho

Pi. SoC này khác với CPU ở trong PC thông thường ở chỗ nó được chế tạo dựa trên kiến trúc tập lệnh (Instruction Set Architect – ISA) là ARM chứ không phải kiến trúc x86 như của Intel. ARM có ISA dạng rút gọn RISC và tiêu thụ điện năng rất thấp nên phù hợp với thiết bị di động.

ARM dẫn đầu trong mảng thiết bị di động. Lấy ví dụ như chip ARM trên Raspberry Pi: toàn bộ mạch hoạt động với nguồn 5V, 700mA tức là chỉ tiêu hao 3.5W mỗi giờ trong khi một laptop cũng ngốn ít nhất vài chục Watt. Thiết kế này bảo đảm Raspberry Pi hoạt động với sức mạnh vừa phải trong khi vẫn giữ được hình dáng nhỏ gọn do không cần quạt tản nhiệt và do đó, ARM có mặt trong hầu hết điện thoại di động thời nay.

Cắm Raspberry Pi vào cổng USB của máy tính có thể cấp nguồn cho Pi hoạt động ở mức bình thường, không sử dụng các kết nối internet như LAN và wifi.

Raspberry Pi là một máy tính, để máy tính này hoạt động người dùng cần cài đặt hệ điều hành. Trong thế giới mã nguồn mở linux, có rất nhiều phiên bản hệ điều hành tùy biến (distro) khác nhau. Tùy theo nhu cầu và mục đích, cũng như khả năng học hỏi mà người dùng sẽ sử dụng distro phù hợp với mình.

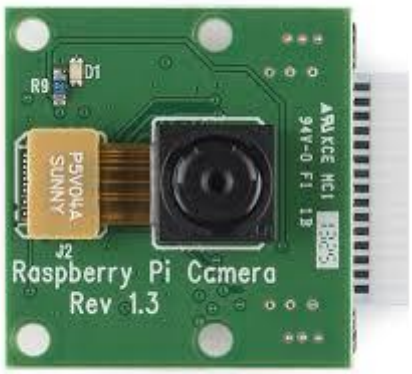
Thông số chi tiết của Raspberry Pi 3 Model B:

- Bộ xử lý BCM2837 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU.
- Bộ nhớ RAM 1GB (LPDDR2 SDRAM).
- On-board Wireless LAN - 2.4 GHz 802.11 b/g/n (BCM43438).
- On-board Bluetooth 4.1 + HS Low-energy (BLE) (BCM43438).
- 4 cổng USB 2.0.
- 40 chân GPIO.
- Cổng HDMI.
- Jack 3.5.
- Camera interface (CSI).
- Display interface (DSI).
- Khe MicroSD.
- VideoCore IV multimedia/3D graphics core @ 400MHz/300MHz.

2.1.2 Module camera Raspberry Pi

Raspberry Pi Camera là module camera được chính Raspberry Pi Foundation thiết kế và đưa vào sản xuất đại trà từ tháng 5/2013. Camera module ra đời đã làm thoả lòng rất nhiều tín đồ yêu thích Raspberry.Trước khi xuất hiện camera, điều duy nhất người sử dụng có thể làm để thêm khả năng nhận biết hình ảnh, quay phim, chụp hình cho Raspberry Pi Camera là sử dụng 1 webcam cắm vào cổng USB. Với các webcam Logotech tích hợp sẵn định dạng xuất mjpeg sẽ giúp Raspberry xử lý nhanh hơn. Nhưng các webcam Logitech lại có giá thành khá cao, nhất là các webcam có độ phân giải lớn.

Raspberry Pi camera được tích hợp camera 5 megapixel có độ nhạy sáng cao, có thể chụp tốt ở nhiều điều kiện ánh sáng khác nhau, cả trong nhà và ngoài trời. Điểm đặc biệt mà camera mang lại đó là chụp hình độ nét cao trong lúc quay phim. Hình 2.3 là một chiếc module camera Raspberry Pi V1.3.



Hình 2.3: Module Camera Raspberry Pi.

Thông số:

- Cảm biến: OV5647
- Độ phân giải: 5MP
- Độ phân giải hình: 2592x1944 pixel.
- Quay phim HD 1080P 30, 720P 60, VGA 640x480P 60.
- Lens: Fixed Focus.

- Conector: Ribon conector.
- Kích thước: 25x24x9mm.

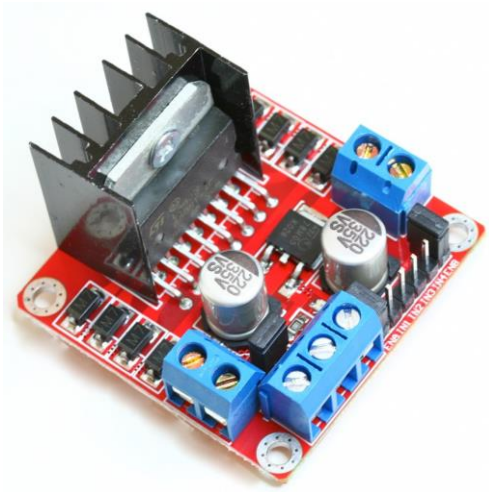
2.1.3 Servo

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0o - 180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực lớn (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,... Hình ảnh động cơ servo như hình 2.4.

Hình 2.4: Động cơ servo SG90.

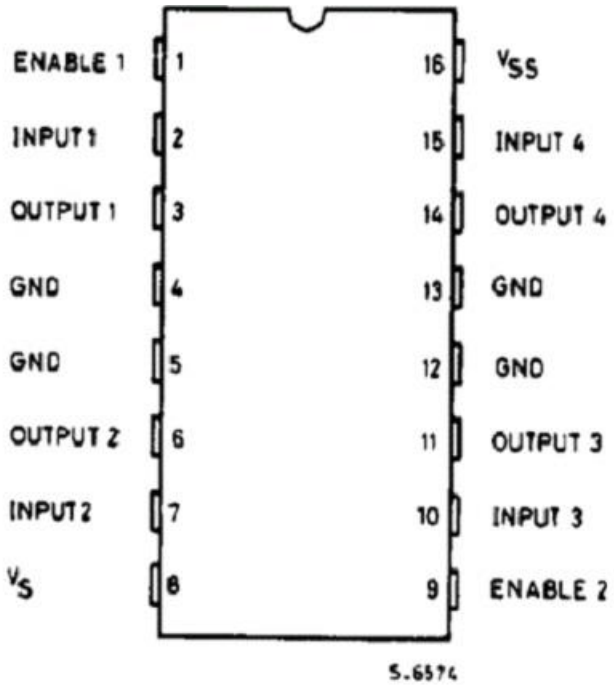
Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.

2.1.4 Module điều khiển động cơ L298N



Hình 2. 5: Module điều khiển động cơ L298.

Hình 2.5 là một chiếc module L298N sử dụng IC cầu H L293D điều khiển. Mạch cầu H điều khiển động cơ L298N cho phép người dùng kiểm soát tốc độ và hướng của hai động cơ DC hoặc kiểm soát một động cơ bước lưỡng cực một cách dễ dàng.



Hình 2.6: Sơ đồ chân IC L293D.

Bảng 2.1: Thông số IC L293D.

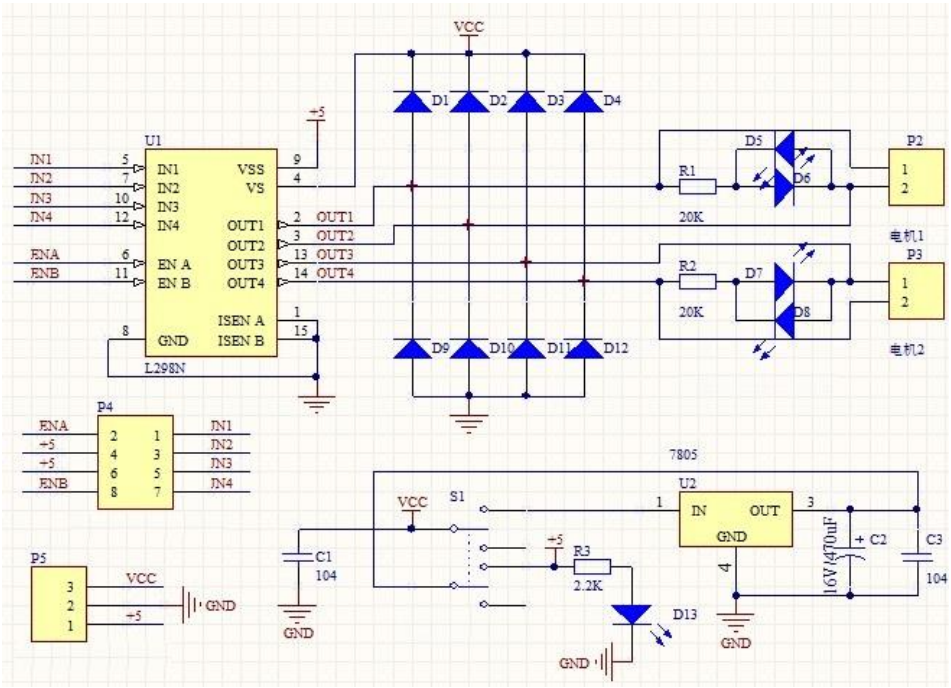
Thông số	Giá trị
Điện áp nguồn	4,5 - 36V
Điện áp ra tải max	36 V
Dòng ra	600mA – 1200mA
Số đầu ra	4

IC L293D là một IC cầu, có thể điều khiển cùng lúc 2 động cơ hoạt động riêng biệt. IC L293D có 4 kênh chấp nhận ngõ vào TTL hoặc DTL. Hỗ trợ các thiết bị hoạt động có tần số lên đến 5 kHz.

Cấu tạo của module L298N bao gồm các chân :

- 12V, 5V: Đây là 2 chân cấp nguồn trực tiếp đến động cơ.
- GND: chân này là GND của nguồn cấp cho Động cơ.

- 2 Jump A enable và B enable, để như hình, Gồm có 4 chân Input. IN1, IN2, IN3, IN4. Chức năng các chân này em sẽ giải thích ở bước sau. Output A: nối với động cơ A. bạn chú ý chân +, -. Nếu nối ngược thì động cơ sẽ chạy ngược. Và chú ý nếu nối động cơ bước, phải đấu nối các pha cho phù hợp.



Hình 2.7: Sơ đồ nguyên lí mạch điều khiển động cơ.

Nguyên lí hoạt động: Hai chân ENA và ENB dùng để điều khiển các mạch cầu H trong L298. Nếu ở mức logic “1” (nối với nguồn 5V) thì cho phép mạch cầu H hoạt động, nếu ở mức logic “0” thì mạch cầu H không hoạt động. Khi ENA = 0: Động cơ không 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 ngay tức thì.

2.1.5 Động cơ DC giảm tốc

Động cơ DC giảm tốc V1 là loại được lựa chọn và sử dụng nhiều nhất hiện nay cho các thiết kế Robot đơn giản. Động cơ DC giảm tốc V1 có chất lượng và giá thành vừa phải cùng với khả năng dễ lắp ráp của nó đem đến chi phí tiết kiệm và sự tiện dụng cho người sử dụng, các bạn khi mua động cơ giảm tốc V1 có thể mua thêm gá bắt động cơ vào thân Robot cũng như bánh xe tương thích.

Hình 2.8: Động cơ DC giảm tốc.

Thông số kỹ thuật:

- Điện áp hoạt động : 3-9VDC.
- Dòng điện tiêu thụ: 110-140mA.
- Tỉ số truyền: 1:120.
- Số vòng/1phút.
- 50 vòng/ 1 phút tại 3VDC.
- 83 vòng/ 1 phút tại 5VDC.
- Moment: 1.0KG.CM.

2.2 VÀI NÉT VỀ XỬ LÍ ẢNH

2.2.1 Ảnh số [1]

Ảnh số là một tập hợp của nhiều điểm ảnh, hay còn gọi là pixel. Mỗi điểm ảnh biểu diễn một màu sắc nhất định (hay độ sáng với ảnh đen trắng) tại một điểm duy nhất, có thể xem một điểm ảnh giống như một chấm nhỏ trong một tấm ảnh màu. Bằng phương pháp đo lường và thống kê một lượng lớn các điểm ảnh,

chúng ta hoàn toàn có thể tái cấu trúc các điểm ảnh này thành một ảnh mới gần giống với ảnh gốc. Có thể nói pixel gần giống như các phần tử có cấu trúc hạt trên một ảnh thông thường nhưng được sắp xếp theo từng hàng và cột và chứa các thông tin khác nhau.

Ảnh được biểu diễn dưới dạng một ma trận hai chiều với các pixel được xác định bởi cặp tọa độ (x, y), trong đó, giá trị của pixel tại tọa độ nhất định biểu diễn độ sáng (ảnh đen trắng) hay màu nhất định (ảnh màu).

Mỗi điểm ảnh tương ứng với một phần của một đối tượng vật lý trong thế giới ba chiều. Đối tượng này được mô tả bởi một vài nguồn sáng mà trong đó chúng được phản chiếu một phần và hấp thụ một phần bởi vật thể. Phần phản chiếu có thể thu được bằng các cảm biến để mô tả lại khung cảnh tương ứng và nó được ghi lại như là đặc trưng của điểm ảnh, hay nói cách khác, các giá trị này phụ thuộc vào từng loại cảm biến được dùng để phản ánh khung cảnh từ nguồn sáng phản chiếu.

Có 2 dạng quan trọng trong ảnh số được dùng với nhiều mục đích khác nhau là ảnh màu và ảnh đen trắng (hay còn gọi là ảnh xám). Trong đó, ảnh màu được cấu trúc từ các pixel màu trong khi ảnh đen trắng được xây dựng từ các pixel có giá trị mức xám khác nhau.

Ảnh đen trắng: với một ảnh đen trắng được xây dựng từ nhiều pixel mà tại đó biểu diễn một giá trị nhất định tương ứng với một mức xám. Những mức xám này trải dài trong một khoảng từ đen sang trắng với bước nhảy rất mịn, thông thường là 256 mức xám khác nhau theo tiêu chuẩn. Do mắt người chỉ có thể phân biệt một cách rõ ràng với khoảng 200 mức xám khác nhau nên vì thế hoàn toàn có thể nhận xét sự thay đổi liên tục các mức xám.

Ảnh màu: một ảnh màu thường được tạo thành từ nhiều pixel mà trong đó mỗi pixel được biểu diễn bởi ba giá trị tương ứng với các mức trong các kênh màu đỏ (Red), xanh lá (Green) và xanh dương (Blue) tại một vị trí cụ thể. Các kênh màu Red, Green và Blue (trong không gian màu RGB) là những màu cơ bản mà từ đó có thể tạo ra các màu khác nhau bằng phương pháp pha trộn. Với

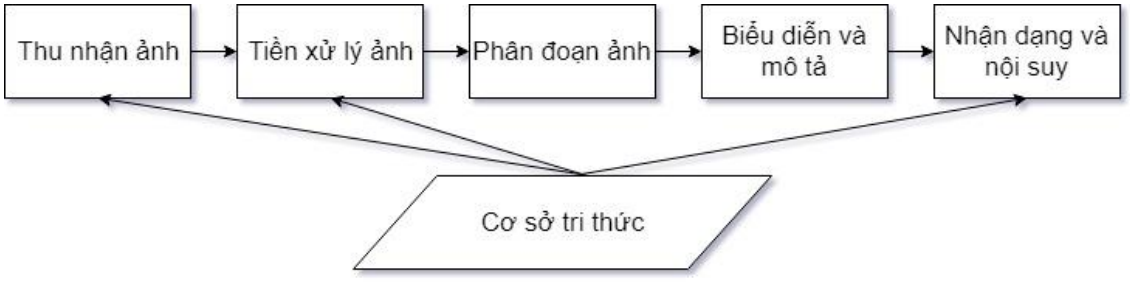
việc chuẩn hóa 256 (28) mức cho từng kênh màu chính, từ đó có thể thấy một pixel màu có thể biểu diễn được một trong (28) 3 =16777216 màu khác nhau. Từ đó có thể thấy rằng với 1 pixel thì chỉ cần 1 byte cho việc lưu trữ đối với ảnh đen trắng và 3 bytes đối với ảnh màu.

Ảnh nhị phân: chỉ sử dụng duy nhất một bit để biểu diễn một pixel. Do một bit chỉ có thể xác lập hai trạng thái là đóng và mở hay 1 và 0 tương ứng với hai màu là đen và trắng. Do đặc trưng trên mà ảnh nhị phân ít khi được sử dụng trong thực tế.

Ảnh chỉ số: một vài ảnh màu (hay đen trắng) được tạo thành từ một bảng màu có sẵn bị giới hạn, điển hình thường dùng là tập 256 màu khác nhau. Những ảnh này được gọi là ảnh màu chỉ số hóa do dữ liệu dành cho mỗi pixel bao gồm chỉ số có sẵn chỉ rõ màu trong tập có sẵn ứng với pixel đang xem xét.

2.2.2 Quy trình xử lý ảnh số

Để dễ tưởng tượng, xét các bước cần thiết trong xử lý ảnh số. Đầu tiên, ảnh tự nhiên từ thế giới ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu CCIR). Gần đây, 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 hiện nay là một thí dụ gần gũi). Mặt khác, ảnh cũng có thể tiếp nhận từ vệ tinh; có thể quét từ ảnh chụp bằng máy quét ảnh. Hình 2.9 dưới đây mô tả các bước cơ bản trong xử lý ảnh số.



Hình 2.9: Các bước cơ bản trong xử lý ảnh số.

Sơ đồ này bao gồm các thành phần sau:

Phần thu nhận ảnh (Image Acquisition).

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hoá (như loại CCD – Change Coupled Device) là loại photodiot tạo cường độ sáng tại mỗi điểm ảnh.

Camera thường dùng là loại quét dòng ; ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh).

Tiền xử lý (Image Processing).

Sau bộ thu nhận, ảnh có thể nhiễu độ tương phản thấp nên cần đưa vào bộ tiền xử lý đểnâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng độ tương phản để làm ảnh rõ hơn, nét hơn.

Phân đoạn (Segmentation) hay phân vùng ảnh.

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu, chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

Biểu diễn ảnh số (Image Representation).

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lận cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Selection) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phân biệt ký tự này với ký tự khác.

Nhận dạng và nội suy ảnh số (Image Recognition and Interpretation).

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể được nội suy thành mã điện thoại. Có nhiều cách phân loai ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản: Nhận dạng theo tham số và nhận dạng theo cấu trúc.

Một số đối tượng nhận dạng khá phổ biến hiện nay đang được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người…

Cơ sở tri thức (Knowledge Base).

Như đã nói ở trên, ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiễu. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy.

Mô tả (Biểu diễn ảnh).

Ảnh sau khi số hoá sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòi hỏi dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ. Thông thường, các ảnh thô đó được đặc tả (biểu diễn) lại (hay đơn giản là mã hoá) theo các đặc điểm của ảnh được gọi là các đặc trưng ảnh (Image Features) như: biên ảnh (Boundary), vùng ảnh (Region). Một số phương pháp biểu diễn thường dùng: Biểu diễn bằng mã chạy (Run-Length Code), biểu diễn bằng mã xích (Chaine - Code) và biểu diễn bằng mã tứ phân (Quad-Tree Code).

2.2.3 Trích đặc trưng và một số phép biến đổi ảnh

Trích chọn đặc trưng là cơ sở của tra cứu ảnh dựa vào nội dung.Theo nghĩa rộng, các đặc trưng có thể bao gồm cả các đặc trưng dựa vào văn bản và các đặc trưng trực quan như màu, kết cấu, hình dạng. Trong phạm vi đặc trưng trực quan, các đặc trưng có thể được phân loại tiếp thành các đặc trưng chung và các đặc trưng lĩnh vực cụ thể. Các đặc trưng trực quan chung gồm màu, kết cấu, và hình dạng trong khi các đặc trưng lĩnh vực cụ thể là phụ thuộc ứng dụng. Các đặc trưng lĩnh vực cụ thể bao gồm nhiều tri thức lĩnh vực. Nhìn chung, không tồn tại một biểu diễn đơn tốt nhất cho một đặc trưng đã cho. Với mọi đặc trưng được cho tồn tại nhiều biểu diễn mô tả đặc trưng từ các cảnh huống khác nhau.

Màu sắc

Màu là đặc trưng trực quan quan trọng đầu tiên và đơn giản nhất cho việc đánh chỉ số và tra cứu các ảnh. Nó cũng là đặc trưng được sử dụng phổ biến nhất trong tra cứu ảnh dựa vào nội dung. Một ảnh màu tiêu biểu được thu từ một camera số, hoặc được tải xuống từ Internet thường có ba kênh màu (các ảnh xám chỉ có một kênh, các ảnh đa phổ có thể có nhiều hơn ba kênh). Tuy nhiên, các giá trị của dữ liệu ba chiều (3 kênh màu) từ ảnh màu không cho chúng ta một mô tả chính xác của màu trong ảnh, nhưng cho vị trí của các điểm ảnh này trong không gian màu.

Các điểm ảnh có các giá trị sẽ xuất hiện khác nhau về màu trong các không gian màu khác nhau.

Hình dạng

Các đặc trưng về hình dạng có thể chia thành các đặc trưng toàn cục hoặc các đặc trưng cục bộ.

- Đặc trưng toàn cục là các đặc trưng thuộc tính thu được từ toàn bộ hình dạng ảnh như chu vi, tính tròn, tính hướng trục...
- Đặc trưng cục bộ là đặc trưng thu được từ việc thao tác với một phần của ảnh, không phụ thuộc vào toàn bộ ảnh....

Các đặc trưng hình ảnh của các đối tượng hoặc các vùng đã được sử dụng trong nhiều hệ thống tra cứu ảnh dựa vào nội dung. So với các đặc trưng màu, các đặc trưng hình dạng thường được mô tả sau khi các ảnh được phân lại thành các vùng hoặc các đối tượng. Do phân đoạn ảnh mạnh và chính xác là khó đạt được, sử dụng các đặc trưng hình dạng cho tra cứu ảnh bị giới hạn đối với các ứng dụng chuyên biệt, ở đó các đối tượng hoặc các vùng đã có sẵn.

Color Histogram

Color Histogram là một dạng đặc trưng toàn cục biểu diễn phân phối của các màu trên ảnh. Color Histogram thống kê số lượng các pixel có giá trị nằm trong một khoảng màu nhất định cho trước. Color Histogram có thể được tính trên các ảnh dạng RGB hoặc HSV, thông dụng là HSV. Cách tính là với mỗi kênh màu nhất định, chia kênh màu này thành "n bin". Sau đó, thống kê số lượng pixel trên ảnh có giá trị màu thuộc về "n bin" này. Cuối cùng, nối các bin của các kênh màu lại với nhau để tạo thành đặc trưng.

Ưu điểm:

- Tính toán nhanh, đơn giản, thích hợp trong các ứng dụng thời gian thực.
- Có thể nói Color Histogram là bất biến, vì khi một ảnh bị xoay đi thì phân phối màu hầu như là không đổi.

Khuyết điểm:

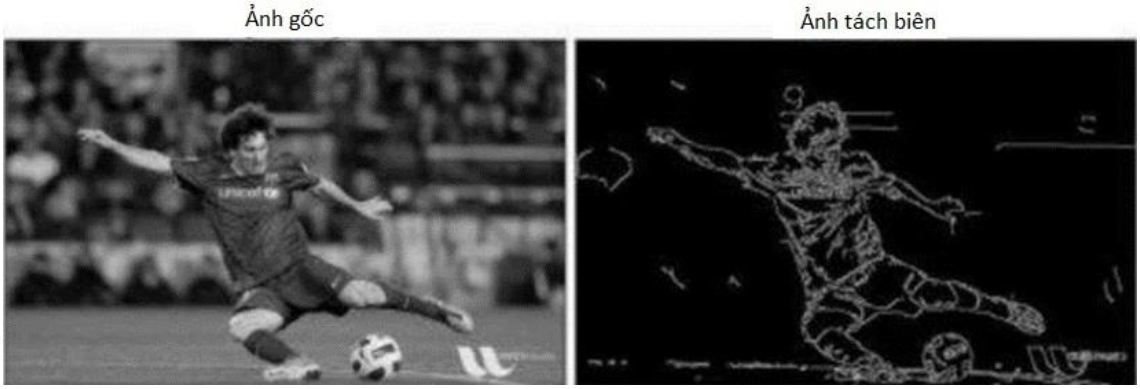
- Color Histogram không nói lên được sự tương đồng về hình dáng, cấu trúc của ảnh. Các ảnh dù cho rất khác nhau nhưng vẫn có phân phối màu giống nhau.
- Color Histogram dễ bị nhiễu với thay đổi về cường độ sáng.

Thuật toán Canny

Phát hiện cạnh Canny là một trong những phương pháp phát hiện biên được sử dụng để tìm tất cả các điểm cạnh trong hình ảnh và đầu ra là một bản đồ nhị phân. Canny sử dụng một gradient trên hình ảnh để tìm những thay đổi mạnh về cường độ pixel. Đây là những đường nét có khả năng là đường viền trong hình ảnh đầu vào. Đầu ra sẽ là một ảnh nhị phân cho thấy những đường nét của hình.

Trong thuật toán Canny, các đạo hàm bậc một được tính theo x và y và sao đó kết hợp thành bốn đạo hàm hướng. Các điểm nơi những đạo hàm hướng này là cực trị địa phương sau đó là các ứng viên cho tập hợp thành các cạnh. Trong đồ án này sử dụng Canny để phát hiện ranh giới làn đường trong khung hình.

Việc phát hiện cạnh cơ bản sử dụng vector gradient của một hình ảnh cường độ để phát hiện ranh giới làn đường có độ tương phản cao trong ảnh. Đây là một trong những phương pháp tốt nhất và hiệu quả trong nhiều phương pháp phát hiện biên.



Hình 2.10: Hình tách biên sử dụng Canny.

Contour Tracking.

Contour có thể được giải thích đơn giản là một đường cong nối tất cả các điểm liên tục (dọc theo đường biên), có cùng màu hoặc cường độ. Các contour là một công cụ hữu ích để phân tích hình dạng và phát hiện và nhận dạng đối tượng. Contour tracking là một trong nhiều kỹ thuật tiền xử lý được thực hiện trên hình ảnh kỹ thuật số để trích xuất thông tin về hình dạng chung của chúng. Khi contour của một mẫu nhất định được trích xuất, các đặc điểm khác nhau của nó sẽ được kiểm tra và sử dụng làm các tính năng mà sau này sẽ được sử dụng trong phân loại mẫu. Do đó, việc tính toán contour tracking sẽ tạo ra các tính năng chính xác hơn, điều này sẽ tăng cơ hội phân loại chính xác một mẫu nhất định.

2.3 NGÔN NGỮ LẬP TRÌNH VÀ MỘT SỐ THƯ VIỆN SỬ DỤNG

2.3.1 Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình được sử dụng phổ biến ngày nay từ trong môi trường học đường cho tới các dự án lớn. Ngôn ngữ phát triển nhiều loại ứng dụng, phần mềm khác nhau như các chương trình chạy trên desktop, server, lập trình các ứng dụng web... Ngoài ra Python cũng là ngôn ngữ ưa thích trong xây dựng các chương trình trí tuệ nhân tạo trong đó bao gồm machine learning. Ban đầu, Python được phát triển để chạy trên nền Unix, nhưng sau này, nó đã chạy trên mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Python do Guido van Rossum tạo ra năm 1990. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. 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.

Python là ngôn ngữ có hình thức đơn giản, cú pháp ngắn gọn, sử dụng một số lượng ít các từ khoá, do đó Python là một ngôn ngữ dễ học đối với người mới bắt đầu tìm hiểu. Python là ngôn ngữ có mã lệnh (source code hay đơn giản là code) không mấy phức tạp. Cả trường hợp bạn chưa biết gì về Python bạn cũng có thể suy đoán được ý nghĩa của từng dòng lệnh trong source code. Python có nhiều ứng dụng trên nhiều nền tảng, chương trình phần mềm viết bằng ngôn ngữ Python có thể được chạy trên nhiều nền tảng hệ điều hành khác nhau bao gồm Windows, Mac OSX và Linux.

2.3.2 Thư viện OpenCV.

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở, nó là miễn phí cho những ai bắt đầu tiếp cận với các học thuật. OpenCV được ứng dụng trong nhiều lĩnh vực như cho thị giác máy tính hay xử lý ảnh và máy học. Thư viện được lập trình trên các ngôn ngữ cấp cao: C++, C, Python,

hay Java và hỗ trợ trên các nền tảng Window, Linux, Mac OS, iOS và Android. OpenCV đã được tạo ra tại Intel vào năm 1999 bởi Gary Bradsky, và ra mắt vào năm 2000. Opencv có rất nhiều ứng dụng: Nhận dạng ảnh, xử lý hình ảnh, phục hồi hình.

ảnh/video, thực tế ảo,... Ở đề tài này thư viện OpenCV được chạy trên ngôn ngữ Python. OpenCV được dùng làm thư viện chính để xử lý hình ảnh đầu vào và sau đó đi nhận dạng ảnh.

2.3.3 Thư viện Scikit-learn.

Scikit-learn (viết tắt là sklearn) là một thư viện mã nguồn mở dành cho học máy - một ngành trong trí tuệ nhân tạo, rất mạnh mẽ và thông dụng với cộng đồng Python, được thiết kế trên nền NumPy và SciPy. Scikit-learn chứa hầu hết các thuật toán machine learning hiện đại nhất, đi kèm với documentations, luôn được cập nhật.

Ưu điểm của sklearn:

- Hỗ trợ hầu hết các thuật toán của machine learning một cách đơn giản, hiệu quả mà chúng ta không cần phải mất công ngồi cài đặt lại.
- Có tài liệu hướng dẫn sử dụng.
- Độ tin cậy cao do scikit-learn được xây dựng bởi các chuyên gia hàng đầu.
- Có nguồn dữ liệu phong phú: iris, digit, …

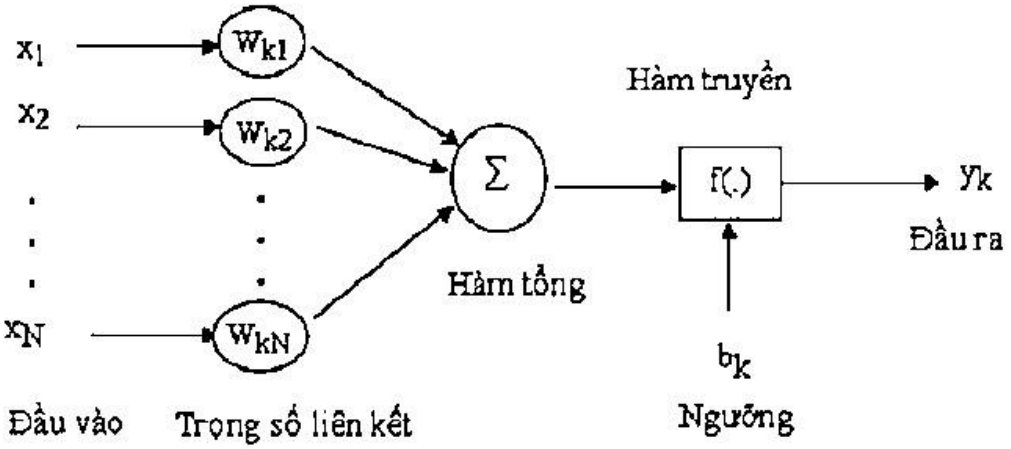
2.4 MẠNG NEURAL

2.4.1 Giới thiệu về mạng Neural [3]

Định nghĩa: Mạng nơron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một

quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron.

Cấu trúc neural nhân tạo:



Hình 2.11: Cấu tạo một neural.

Các thành phần cơ bản của một nơron nhân tạo bao gồm:

- Tập các đầu vào: Là các tín hiệu vào (input signals) của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.
- Tập các liên kết: Mỗi liên kết được thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là w_{kj}. Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.
- Bộ tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.
- Ngưỡng (còn gọi là một độ lệch - bias): Ngưỡng này thường được đưa vào như một thành phần của hàm truyền.
- Hàm truyền (Transfer function): Hàm này được dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.

- Đầu ra: Là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.

Xét về mặt toán học, cấu trúc của một nơron k, được mô tả bằng cặp biểu thức sau:

$$u_k = \sum_{j=1}^{p} w_{kj} x_j$$

và $$y_k = f(u_k - b_k)$$

Hình 2.12: Biểu thức cấu trúc một nơron.

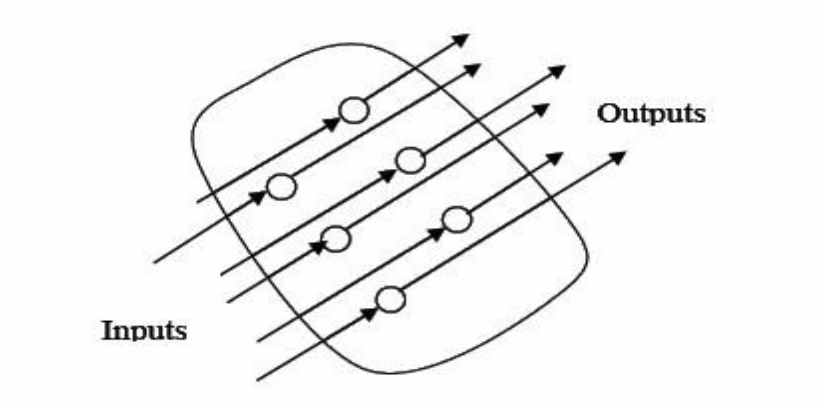
Trong đó: $x_1, x_2, ..., x_p$: là các tín hiệu vào; ($w_{k1}, w_{k2},..., w_{kp}$) là các trọng số liên kết của nơron thứ k; u_k là hàm tổng; b_k là một ngưỡng; f là hàm truyền và y_k là tín hiệu đầu ra của nơron.

Như vậy nơron nhân tạo nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả tới hàm truyền), và cho một tín hiệu đầu ra (là kết quả của hàm truyền).

2.4.2 Một số kiểu mạng Neural [3]

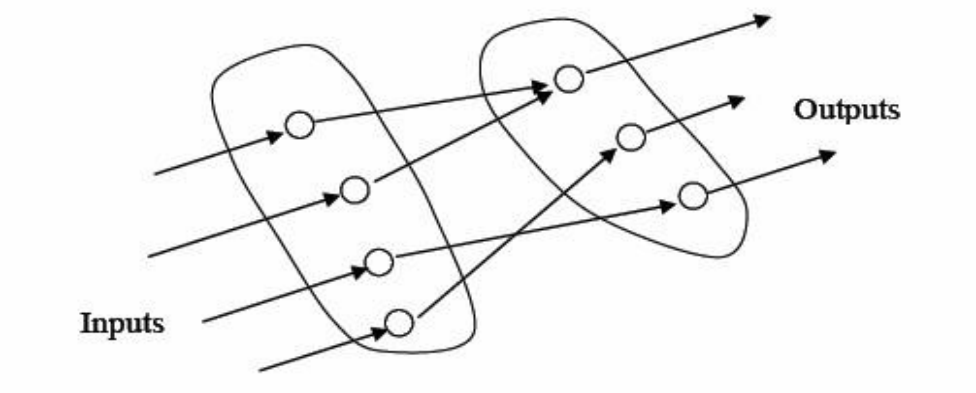
Cách thức kết nối các nơron trong mạng xác định kiến trúc (*topology*) của mạng. Các nơron trong mạng có thể kết nối đầy đủ (*fully connected*) tức là mỗi nơron đều được kết nối với tất cả các nơron khác, hoặc kết nối cục bộ (*partially connected*) chẳng hạn chỉ kết nối giữa các nơron trong các tầng khác nhau. Người ta chia ra hai loại kiến trúc mạng chính:

Tự kết hợp (*autoassociative*): là mạng có các nơron đầu vào cũng là các nơron đầu ra. Mạng Hopfield là một kiểu mạng tự kết hợp.



Hình 2.13: Mạng tự kết hợp.

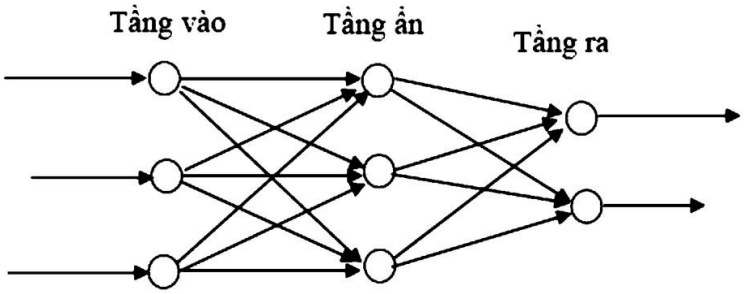
Kết hợp khác kiểu (*heteroassociative*): Hình 2.14 là mạng có tập nơron đầu vào và đầu ra riêng biệt. Perceptron, các mạng Perceptron nhiều tầng (MLP: MultiLayer Perceptron), mạng Kohonen, … thuộc loại này.



Hình 2.14: Mạng kết hợp khác kiểu.

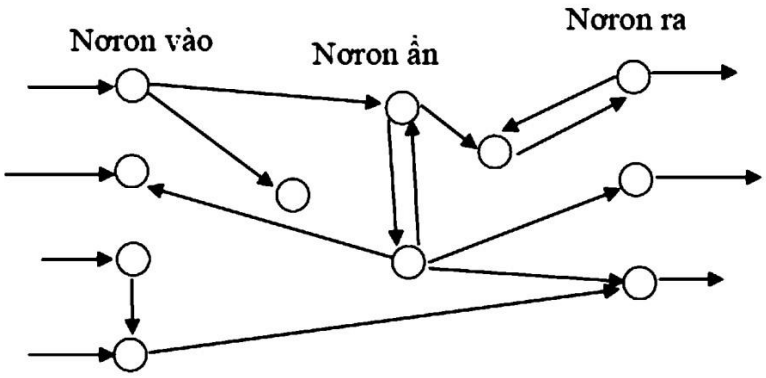
Ngoài ra tùy thuộc vào mạng có các kết nối ngược (*feedback connections*) từ các nơron đầu ra tới các nơron đầu vào hay không, người ta chia ra làm 2 loại kiến trúc mạng.

Kiến trúc truyền thẳng (*feedforward architechture*): Hình 2.15 là kiểu kiến trúc mạng không có các kết nối ngược trở lại từ các nơron đầu ra về các nơron đầu vào; mạng không lưu lại các giá trị output trước và các trạng thái kích hoạt của nơron. Các mạng nơron truyền thẳng cho phép tín hiệu di chuyển theo một đường duy nhất; từ đầu vào tới đầu ra, đầu ra của một tầng bất kì sẽ không ảnh hưởng tới tầng đó. Các mạng kiểu Perceptron là mạng truyền thẳng.



Hình 2.15: Mạng truyền thẳng.

Kiến trúc phản hồi (*Feedback architecture*): Hình 2.16 là kiểu kiến trúc mạng có các kết nối từ nơron đầu ra tới nơron đầu vào. Mạng lưu lại các trạng thái trước đó, và trạng thái tiếp theo không chỉ phụ thuộc vào các tín hiệu đầu vào mà còn phụ thuộc vào các trạng thái trước đó của mạng. Mạng Hopfield thuộc loại này.



Hình 2.16: Mạng phản hồi.

2.5 MẠNG NEURAL TÍCH CHẬP

2.5.1 Định nghĩa mạng Neural tích chập

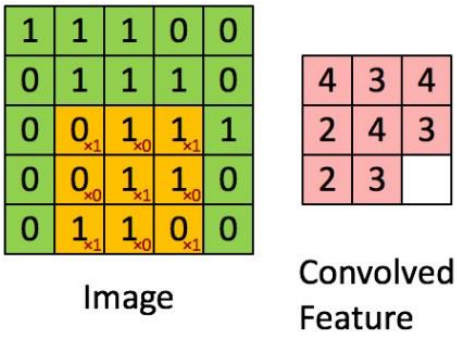
Những năm gần đây, ta đã chứng kiến được nhiều thành tựu vượt bậc trong ngành Thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động.

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Trong luận văn này, chúng ta sẽ trình bày về Convolution (tích chập) cũng như ý tưởng của mô hình CNNs trong phân lớp chữ viết áp dụng trong bài toán nhận dạng biển số xe (Image Classification).

2.5.2 Tích chập (Convolution) [3]

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kĩ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận. Bạn có thể theo dõi cơ chế của tích chập qua hình minh họa bên dưới.



Hình 2.17: Minh họa tích chập.

Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

Sliding window còn có tên gọi là kernel, filter hay feature detector. Ở đây, ta dùng một ma trận filter 3×3 nhân từng thành phần tương ứng (element-wise) với ma trận ảnh bên trái. Gía trị đầu ra do tích của các thành phần này cộng lại. Kết quả của tích chập là một ma trận (convoled feature) sinh ra từ việc trượt ma trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái.

Ta có thể làm mờ bức ảnh ban đầu bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm.

Ngoài ra, ta có thể phát hiện biên cạnh bằng cách tính vi phân (độ dị biệt) giữa các điểm ảnh lân cận.

2.5.3 Mô hình mạng neural tích chập [3][8]

Bây giờ, Chúng ta đã biết thế nào là convolution. Vậy CNNs là gì? CNNs chỉ đơn giản gồm một vài layer của convolution kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) như ReLU hay tanh để tạo ra thông tin trừu tượng hơn (abstract/higher-level) cho các layer tiếp theo.

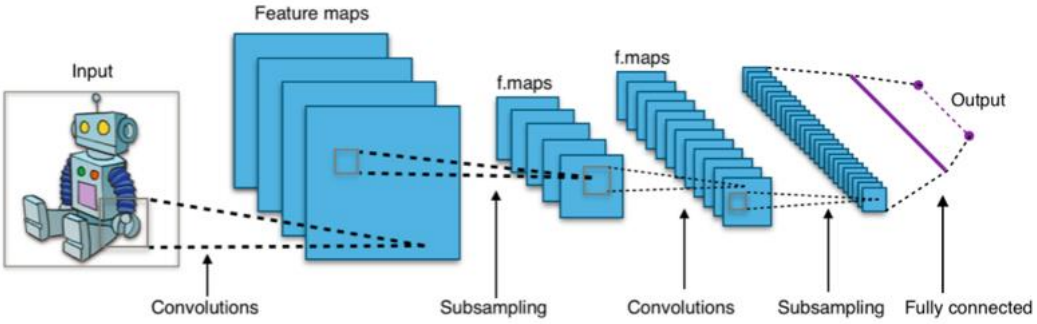
Trong mô hình Feedforward Neural Network (mạng nơ-ron truyền thẳng), các layer kết nối trực tiếp với nhau thông qua một trọng số w (weighted vector).

Các layer này còn được gọi là có kết nối đầy đủ (fully connected layer) hay affine layer.

Trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng ảnh cục bộ của nơ-ron layer trước đó.

Mỗi layer như vậy được áp đặt các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Tuy nhiên, ta sẽ không đi sâu vào khái niệm của các layer này.

Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các filter. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự ***raw pixel*** > ***edges*** > ***shapes*** > ***facial*** > ***high-level features***. Layer cuối cùng được dùng để phân lớp ảnh.



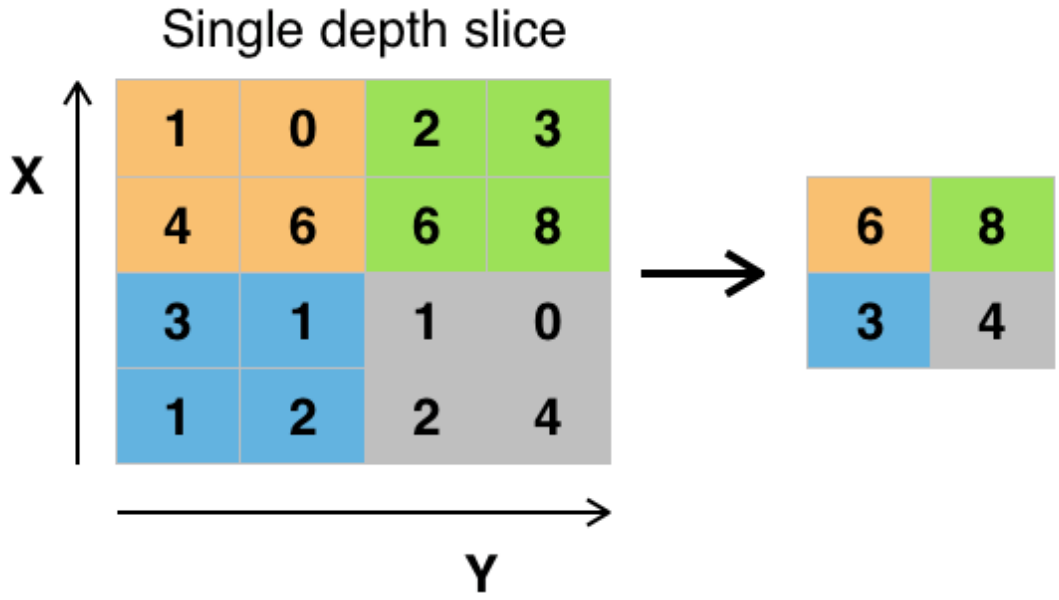
Hình 2.18: Mô hình mạng neural tích chập. (Nguồn Wikimedia)

Hình ảnh trên cho thấy rằng ta đã cung cấp một hình ảnh như một đầu vào cho mạng, trải qua nhiều kết cấu, mẫu phụ, một lớp được kết nối đầy đủ và cuối cùng xuất ra một cái gì đó.

Lớp tích chập tính toán đầu ra của các nơ-ron được kết nối với các vùng cục bộ hoặc các trường tiếp nhận trong đầu vào, mỗi máy tính một sản phẩm chấm giữa các trọng số của chúng và một trường tiếp nhận nhỏ mà chúng được kết nối với khối lượng đầu vào. Mỗi tính toán dẫn đến trích xuất một bản đồ đặc trưng từ hình ảnh đầu vào. Nói cách khác, hãy tưởng tượng có một hình ảnh được biểu diễn dưới dạng ma trận 5x5 của các giá trị và bạn lấy một ma trận 3x3 và trượt cửa sổ 3x3 hoặc kernel xung quanh hình ảnh. Tại mỗi vị trí của ma trận đó, nhân các giá trị của cửa sổ 3x3 của bạn với các giá trị trong hình ảnh hiện đang được che bởi cửa sổ. Kết quả là, nhận được một số duy nhất đại diện cho tất cả các giá trị trong cửa sổ hình ảnh đó. Sử dụng lớp này để lọc: khi cửa sổ di chuyển qua hình ảnh, kiểm tra các mẫu trong phần đó của hình ảnh. Điều này hoạt động vì các bộ lọc, được nhân với các giá trị được đưa ra bởi tích chập.

Mục tiêu của việc lấy mẫu con là để có được một đại diện đầu vào bằng cách giảm kích thước của nó, giúp giảm quá mức. Một trong những kỹ thuật của mẫu phụ là gộp chung tối đa. Với kỹ thuật này, ta chọn giá trị pixel cao nhất từ một vùng tùy thuộc vào kích thước của nó. Nói cách khác, gộp nhóm tối đa lấy giá trị lớn nhất từ cửa sổ của hình ảnh hiện được bao phủ bởi kernel. Ví dụ:ta có thể có một nhóm tổng hợp kích thước tối đa 2 x 2 sẽ chọn giá trị cường độ pixel tối đa từ vùng 2 x 2. Ta có quyền nghĩ rằng lớp gộp sau đó hoạt động rất giống với lớp

chập! Lấy một kernel hoặc một cửa sổ và di chuyển nó qua hình ảnh; Sự khác biệt duy nhất là chức năng được áp dụng cho kernel và cửa sổ hình ảnh không tuyến tính.



Hình 2.19: Tổng hợp tối đa (Nguồn: Wikipedia).

Mục tiêu của lớp được kết nối đầy đủ là làm phẳng các tính năng cấp cao được học bởi các lớp chập và kết hợp tất cả các tính năng. Nó chuyển đầu ra được làm phẳng cho lớp đầu ra nơi ta sử dụng trình phân loại softmax hoặc sigmoid để dự đoán nhãn lớp đầu vào.

CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling).

Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên. Ta phân biệt được một con chó

với một con mèo nhờ vào các đặc trưng từ mức độ thấp (có 4 chân, có đuôi) đến mức độ cao (dáng đi, hình thể, màu lông).

CHƯƠNG 3

THIẾT KẾ HỆ THỐNG

3.1 PHÂN TÍCH HỆ THỐNG

3.1.1 Yêu cầu hệ thống

Mô hình được thiết kế với các yêu cầu sau:

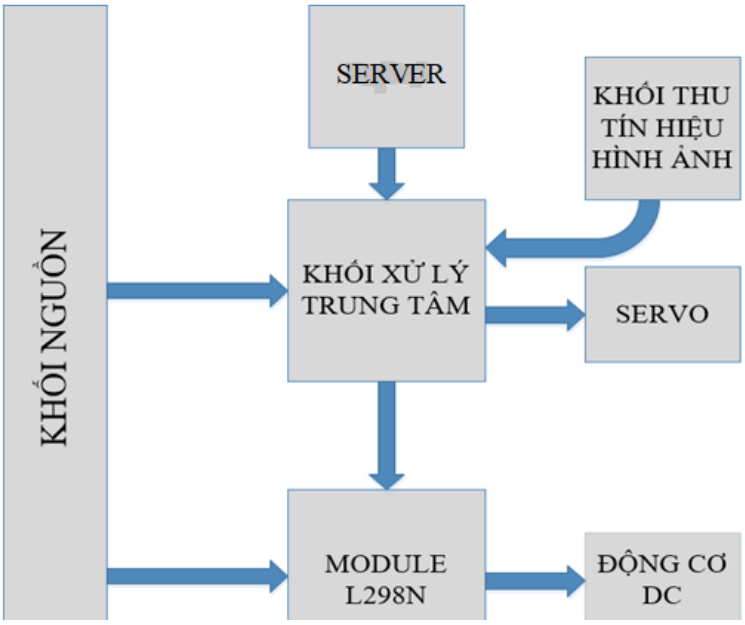
- Xe hoạt động tự động chính xác dựa vào 2 làn đường.
- Xe nhận diện được biển báo Stop và dừng, phát hiện và thực hiện theo đèn giao thông.
- Mô hình xử lí hướng, nhận diện biển báo và đèn tín hiệu qua camera.
- Mô hình sử dụng công nghệ học sâu, model được đào tạo trên máy chủ.
- Hệ thống có thể dễ chỉnh sửa và nâng cấp trong chương trình.

3.1.2 Sơ đồ khối hệ thống

Hệ thống bao gồm các khối chính sau:

- Khối xử lí trung tâm: có chức năng thu thập dữ liệu từ khối nhận diện kết hợp với các model đã đào tạo từ server, tính toán và điều khiểu khối động cơ và khối servo.
- Khối nhận diện: có chức năng thu nhận hình ảnh đầu vào để gửi đến bộ xử lí trung tâm.
- Servo: là đơn vị quay góc để rẽ hướng cho xe.
- Khối động cơ: bao gồm mạch điều khiển động cơ và 2 motor DC, có chức năng chuyển động cho xe đi thẳng về phía trước.
- Server: nơi lưu trữ dữ liệu để đào tạo model.

- Khối nguồn: cung cấp nguồn ổn định cho bộ xử lý trung tâm và module điều khiển động cơ.

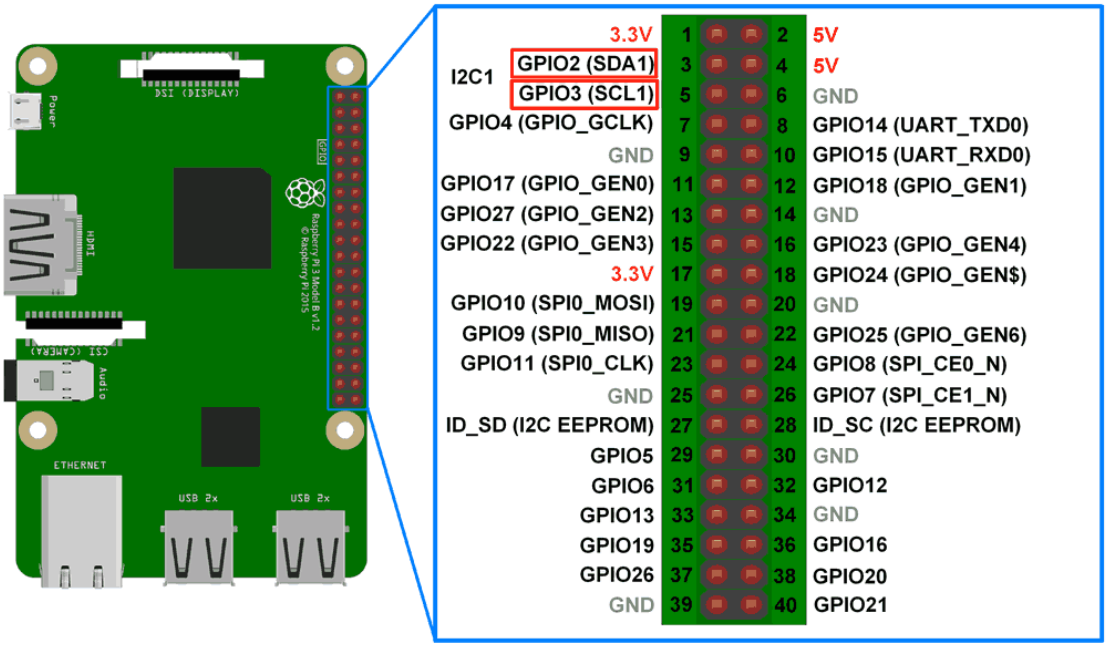


Hình 3.1: Sơ đồ khối hệ thống.

3.2 THIẾT KẾ PHẦN CỨNG

3.2.1 Khối xử lý trung tâm

Raspberry Pi 3 Model B được chọn làm đơn vị xử lí trung tâm trên xe. Nguyên nhân vì Raspberry Pi 3 Model B giá phù hợp với ứng dụng. Nếu như sử dụng Arduino hay các dòng vi xử lý khác thì không đủ tài nguyên để thực hiện xử lý ảnh. Nếu dùng những kit chuyên dụng cho xử lý ảnh của NVIDIA thì giá thành rất cao. Trong phạm vi nghiên cứu của đồ án thì việc dùng Raspberry Pi 3 Model B là phù hợp.



Hình 3.2: Sơ đồ chân Raspberry pi 3.

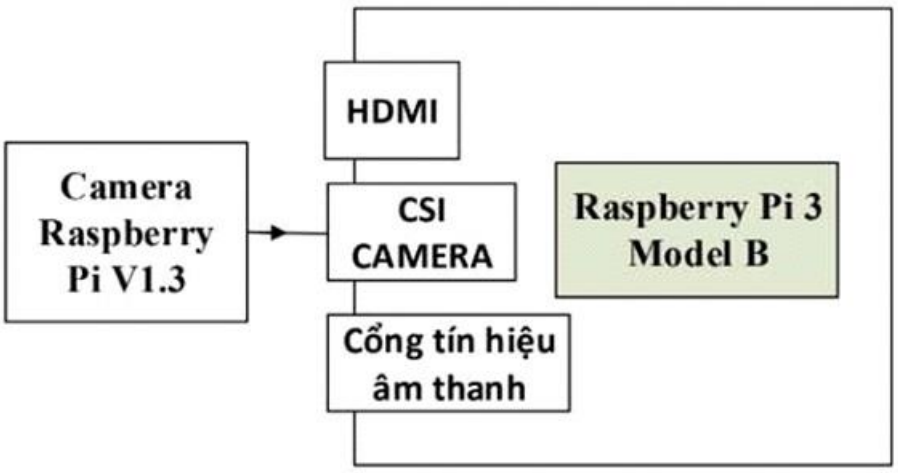
Máy tính nhúng Raspberry có CPU tộc độ xử lý lên tới 1.2GHz chính vì vậy việc sử dụng vào mô hình là hợp lý. Máy tính hỗ trợ rất tốt việc giao tiếp các thiết bị ngoại vi và model từ bên ngoài.

Cổng CSI Camera: Mắt của mô hình này chính là Camera được kết nối thông qua cổng CSI Camera có 15 chân.

Nguồn: Dòng hoạt động của Raspberry 500-1000mA, module Camera sử dụng 250mA. Tổng dòng tiêu thụ được khoảng 1250mA. Chính vì vậy nhóm sử dụng sạc dự phòng có dòng ra 2A để cấp cho máy tính nhúng cho Raspberry.

3.2.2 Khối nhận diện

Khối nhận diện có chức năng thu nhận hình ảnh đầu vào để gửi đến bộ xử lí trung tâm, nhóm chọn camera Raspberry Pi V1.3có độ phân giải 5 Megapixels để vừa có chất lượng tốt vừa dễ xử lý ảnh.



Hình 3.3: Sơ đồ kết nối khối thu tín hiệu hình ảnh.

Cùng với kích thước nhỏ gọn phù hợp cho việc thí nghiệm trong mô hình nhỏ. Camera Pi được kết nối trực tiếp với máy tính nhúng Raspberry thông qua cổng giao tiếp ngoại vi CSI Camera có 15 chân.

Raspberry Pi camera được tích hợp camera 5 megapixel có độ nhạy sáng cao, có thể chụp tốt ở nhiều điều kiện ánh sáng khác nhau, cả trong nhà và ngoài trời. Điểm đặc biệt mà camera mang lại đó là chụp hình độ nét cao trong lúc quay phim.

3.2.3 Khối động cơ Servo

Servo có chức năng nhận dữ liệu từ bộ xử lí trung tâm và tạo ra góc quay phù hợp để xe rẽ phải, trái hay đi thẳng. Ở đây động cơ Servo SG90 được chọn làm động cơ lái.

3.2.4 Khối động cơ DC

Khối động cơ bao gồm 2 động cơ DC giảm tốc cho 2 bánh xe và được điều khiển thông qua Module điều khiển động cơ .

Module điều khiển động cơ L298

Hiện nay trên thị trường module dùng điều khiển động cơ DC như Shield L298, Shield VNH2SP30, Shield L293D... Trong hệ thống sử dụng Module L298N vì trong hệ thống xe chỉ điều khiển 2 động cơ DC, Module L298N thỏa

mãn yêu cầu trên bởi khả năng điều khiển được tối đa 2 động cơ DC, giá bán trên thị trường không quá đắt.

Module L298N sử dụng IC cầu H L293D điều khiển. Mạch cầu H điều khiển động cơ L298N cho phép người dùng kiểm soát tốc độ và hướng của hai động cơ DC hoặc kiểm soát một động cơ bước lưỡng cực một cách dễ dàng.

Động cơ DC giảm tốc

Động cơ DC giảm tốc bao gồm động cơ DC và hộp giảm tốc. Mục đích để làm giảm tốc độ vòng quay và tăng momen xoắn. Trong đồ án này sử dụng động cơ DC giảm tốc V1 1:120. Nguyên nhân vì giá thành rẻ, bền và đáp ứng được nhu cầu của đồ án.

Bảng 3.1: Thông số động cơ DC giảm tốc V1.

Thông số	Giá trị
Điện áp hoạt động	3 – 9 VDC
Dòng tiêu thụ	110 – 140 mA
Tỉ số truyền	1:120
Tốc độ	50v/phút tại 3V 83v/phút tại 5V
Momen	1kg.cm

3.2.5 Khối nguồn

Một mạch điện thì yêu cầu tối thiểu là phải có nguồn cung cấp với giá trị điện áp, dòng điện phù hợp và ổn định để các linh kiện có thể hoạt động với tuổi thọ lâu nhất.Với các hệ thống xe mô hình, điều khiển từ xa hiện nay, phần lớn năng lượng cho khối nguồn được cấp từ pin. Trên thị trường hiện nay có rất nhiều loại pin gồm loại sạc được và không sạc được như pin Akaline, NiCd, Li-ion, Lipo, acquy…

Trong hệ thống xe của đồ án sử dụng loại pin cell 18650 3.7V 4200mAh. Khối nguồn gồm 3 cell pin Li-ion 18650 mắc nối tiếp đủ cho toàn bộ hệ thống hoạt động ổn định trong một thời gian dài.

Khối nguồn cấp nguồn cho Raspberry Pi 3 và 2 động cơ DC:

- Raspberry Pi yêu cầu điện áp 5 VDC, dòng 2A nên sử dụng nguồn sạc dự phòng.
- Module L298N, để đảm bảo ổn định cho hệ thống, nguồn cấp cho động cơ thông qua L298N được lấy từ nguồn, dòng cấp cho động cơ cũng lấy từ nguồn.

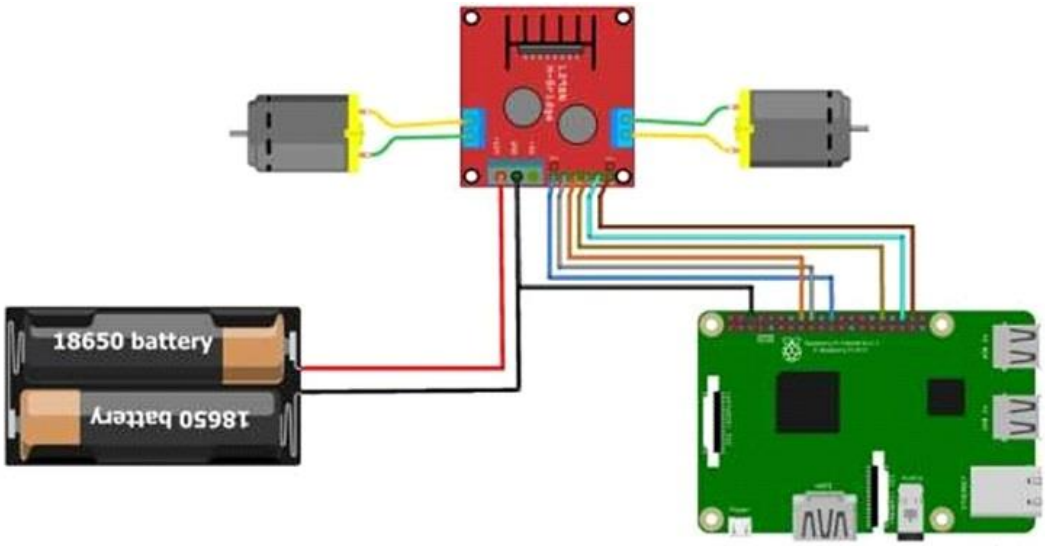
Với yêu cầu của Module L298N, Khối nguồn cần cấp được nguồn VDC. Dòng cần cấp được tính toán như sau:

Icc = 2 * IDC motor = 2 * 140 = 280 (mA)

Trong đó:

- Icc là dòng điện cần cấp.
- IDC motor là dòng điện cấp cho 1 động cơ DC.

3.2.6 Kết nối phần cứng trên xe



Hình 3.4: Mô phỏng kết nối các mô-đun trên xe.

Bảng 3.2: Công xuất tiêu thụ.

Thành phần	Dòng tiêu thụ	Điện áp hoạt động	Công suất

Raspberry Pi	700 mA	5 V	3.5W
Camera Raspberry Pi	250 mA	5 V	1.25W
Servo	500 mA	4.8V	2.4W
Động cơ	120mA x 2	5V x2	2.4W
Tổng			9.55W

3.2.7 Khối server

Máy chủ là nơi chạy chương trình đào tạo model cho hệ thống, nơi chứa đựng dữ liệu đào tạo gồm các hình ảnh trái, phải, thẳng. Để thuận tiện nhóm quyeestt định sử dụng PC Laptop để làm máy chủ.

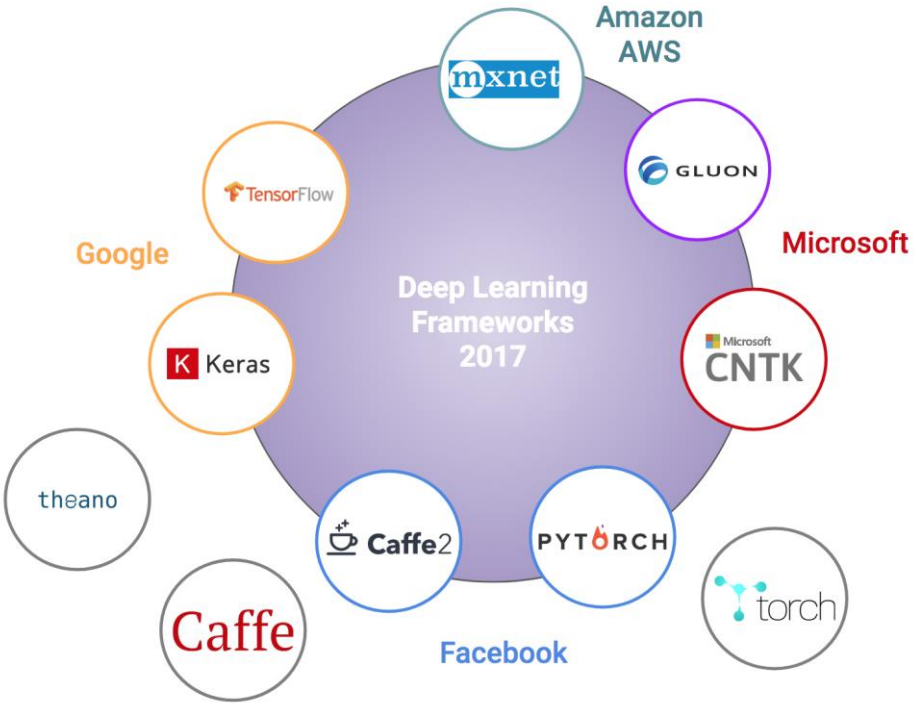
Hình 3.5: Hình ảnh laptop (Nguồn: Ảnh mạng).

3.3 THIẾT KẾ PHẦN MỀM

3.3.1 Model CNN với Keras [7]

Kể từ 2012 khi deep learning có bước đột phá lớn, hàng loạt các thư viện hỗ trợ deep learning ra đời. Cùng với đó, ngày càng nhiều kiến trúc deep learning ra đời, khiến cho số lượng ứng dụng và các bài báo liên quan tới deep learning tăng lên chóng mặt.

Các thư viện deep learning thường được đầu tư bởi những hãng công nghệ lớn: Google (Keras, TensorFlow), Facebook (Caffe2, Pytorch), Microsoft (CNTK), Amazon (Mxnet), Microsoft và Amazon cũng đang bắt tay xây dựng Gluon (phiên bản tương tự như Keras). (Các hãng này đều có các dịch vụ cloud computing và muốn thu hút người dùng).

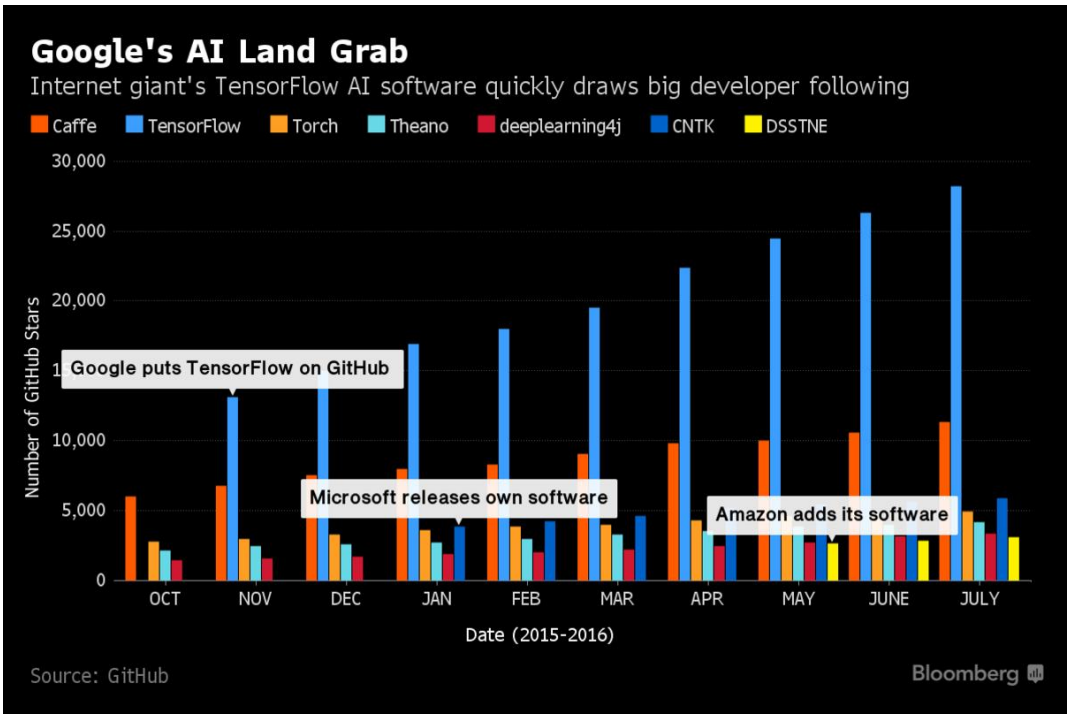


Hình 3.6: Thư viện deep learning và các hãng công nghệ lớn.

(Nguồn : Battle of the Deep Learning frameworks—Part I: 2017, even more frameworks and interfaces)

Hình trên liệt kê các thư viện deep learning và các hãng công nghệ lớn. Việc sử dụng thư viện nào là tốt nhất sẽ tuỳ thuộc vào việc bạn quen với hệ điều hành, ngôn ngữ lập trình nào, bạn sử dụng deep learning vào mục đích nghiên cứu hay ra sản phẩm, bạn sử dụng trên nền tảng phần cứng nào,…

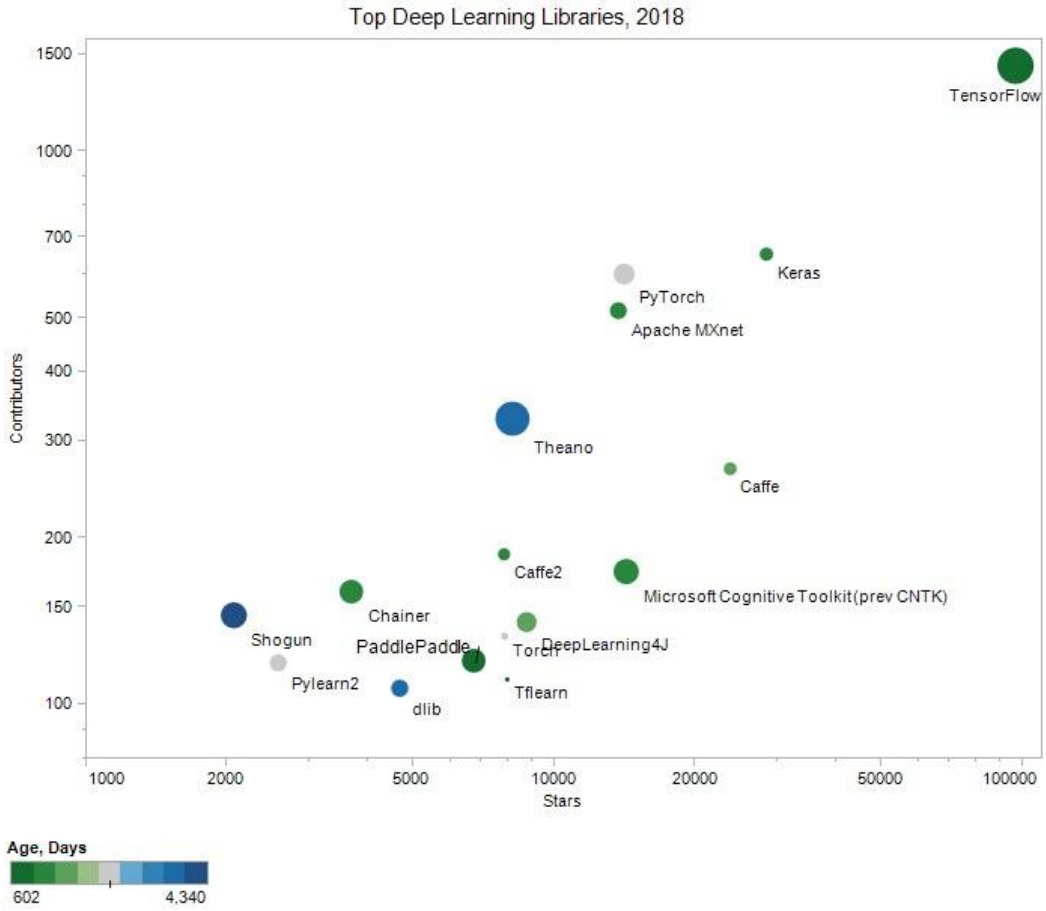
Dưới đây sẽ là một vài thống kê để có thể thấy được thư viện nào được sử dụng nhiều nhất.



Hình 3.7: Số lượng 'stars' trên GitHub repo.

(Nguồn: Machine Learning Frameworks Comparison).

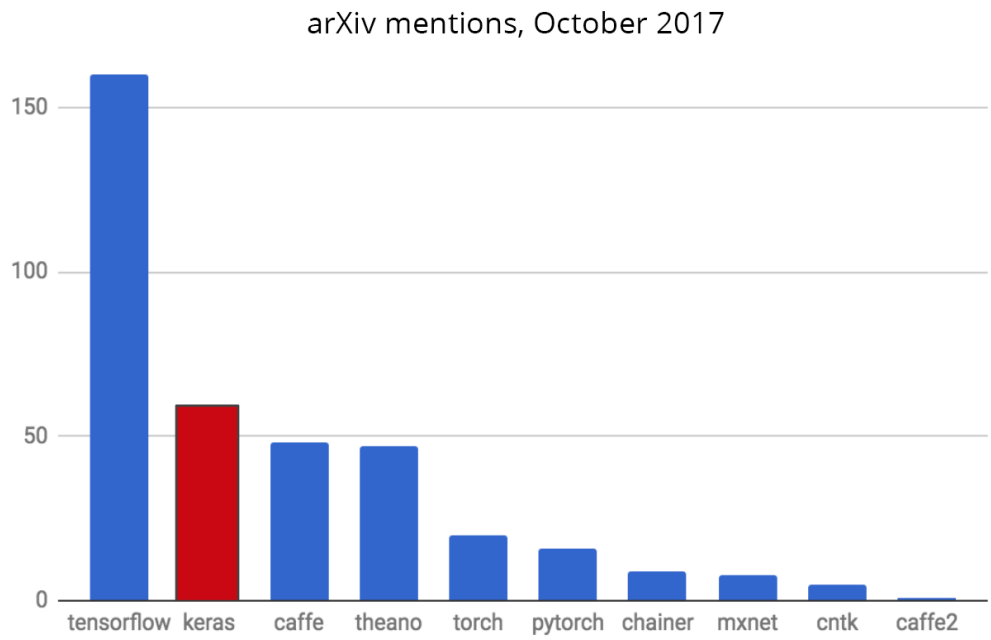
Hình trên cho thấy vào khoảng năm 2015- 2016 thì xếp thứ nhất là TensorFlow, thứ 2 là Caffe.



Hình 3.8: Số lượng 'stars' trên GitHub repo, số lượng 'contributors', và 'tuổi' của thư viện.

(Nguồn: [Top 16 Open Source Deep Learning Libraries and Platforms])

Qua hình thì TensorFlow được xếp thứ nhất, hai là Keras và ba là Caffe (2018).



Hình 3.9: Số lượng các bài báo trên arxiv có đề cập đến mỗi thư viện.

(Nguồn: Why use Keras?)

Số lượng các bài báo trên arxiv đề cập đến thư viện nhiều nhất theo thứ tự 1.TensorFlow, 2. Keras, 3. Caffe.

Những so sánh trên đây chỉ ra rằng TensorFlow, Keras và Caffe là các thư viện được sử dụng nhiều nhất (gần đây có thêm PyTorch rất dễ sử dụng và đang thu hút thêm nhiều người dùng).

Keras được coi là một thư viện 'high-level' với phần 'low-level' (còn được gọi là backend) có thể là TensorFlow, CNTK, hoặc Theano (sắp tới Theano sẽ không được duy trì nâng cấp nữa). Keras có cú pháp đơn giản hơn TensorFlow rất nhiều. Với mục đích giới thiệu về các mô hình nhiều hơn là các sử dụng các thư viện deep learning, tôi sẽ chọn Keras với TensorFlow là 'backend'.

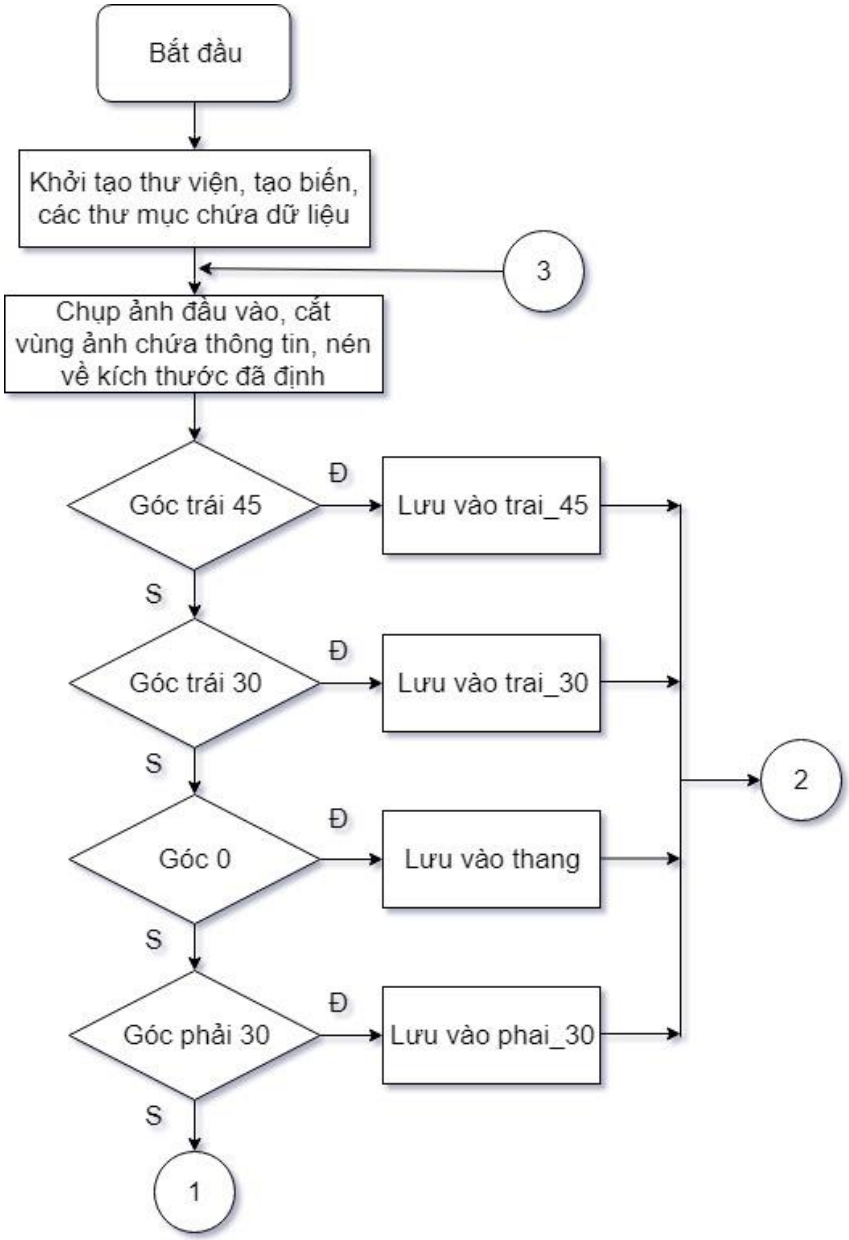
Một số ưu điểm của Keras:

- Keras ưu tiên trải nghiệm của người lập trình.
- Keras đã được sử dụng rộng rãi trong doanh nghiệp và cộng đồng nghiên cứu.
- Keras giúp dễ dàng biến các thiết kế thành sản phẩm.

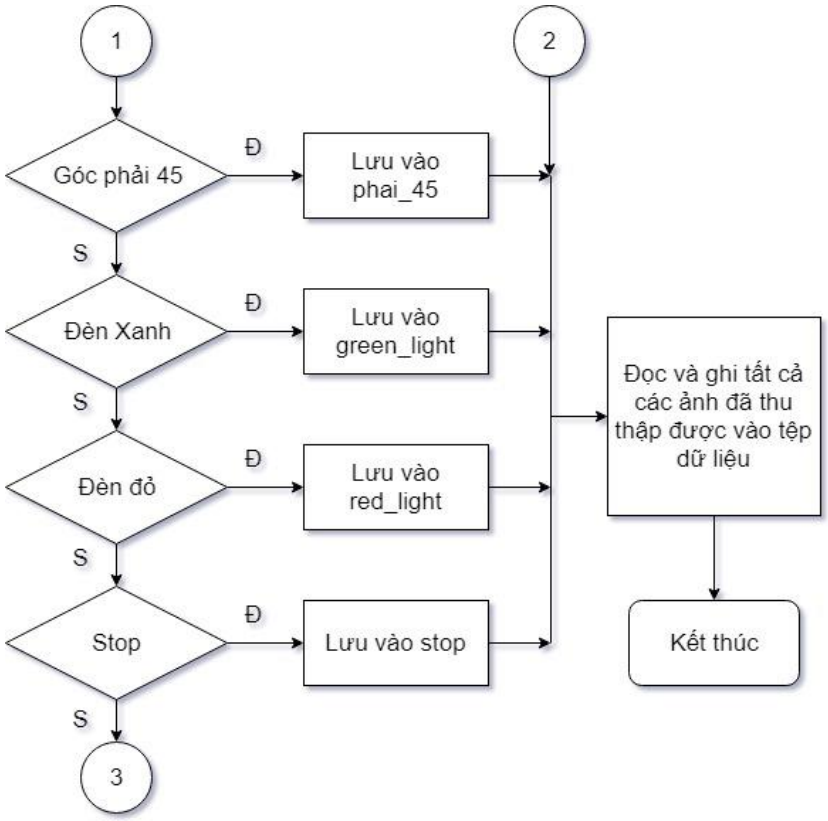
- Keras hỗ trợ huấn luyện trên nhiều GPU phân tán.
- Keras hỗ trợ đa backend engines và không giới hạn bạn vào một hệ sinh thái.

Với các ưu điểm trên, nhóm thực hiện đề tài lựa chọn thư viện Keras để hỗ trợ cho model CNN.

3.3.2 Thu thập và chuẩn hóa đầu vào



Hình 3.10: Lưu đồ thu thập và chuẩn hóa dữ liệu (P1).



Hình 3.11: Lưu đồ thu thập và chuẩn hóa dữ liệu (P2).

Để có được dữ liệu đầu vào, ta phải tiến hành thu thập dữ liệu hình ảnh và chuẩn hóa dữ liệu theo lưu đồ giải thuật trên.

Đầu tiên ta phải khởi tạo các thư viện của OpenCv hỗ trợ cho việc thu thập dữ liệu. Ở đây ta sử dụng các thư viện cơ bản của OpenCv như là cv2 nó cho phép chúng ta thao tác với các dữ liệu ảnh, nó hỗ trợ các thao tác cơ bản như đọc, cắt , sửa, ghi ảnh. Thư viện numpy hỗ trợ chúng ta với các thao tác trên dữ liệu kiểu cấu trúc mảng. Các thư viện còn lại như Sklearn, Keras, Tensorflow hỗ trợ việc tạo model và train model sẽ được làm rõ ở phần sau.

Sau khi đã khởi tạo các thư việc cần thiết, chúng ta bắt đầu đọc dữ liệu. Ta sử dụng hàm cv2.VideoCapture(0) để đọc ảnh tử camera, số 0 là chỉ sẽ đọc ảnh ở camera mặc định của Raspberry Pi.

Hình 3.12: Ảnh gốc được chụp từ camera.

Ảnh vừa thu được từ camera có kích thước 480x640 pixel, như vậy là kích thước ảnh là quá lớn và dư thừa nhiều thông tin. Ta xác vùng ảnh chứa đủ thông tin và tiến hành cắt lấy phần ảnh cần thiết và loại bỏ phần ảnh có thông tin dư thừa.

Hình 3.13: Xác định vùng ảnh cần thiết và dư thừa.

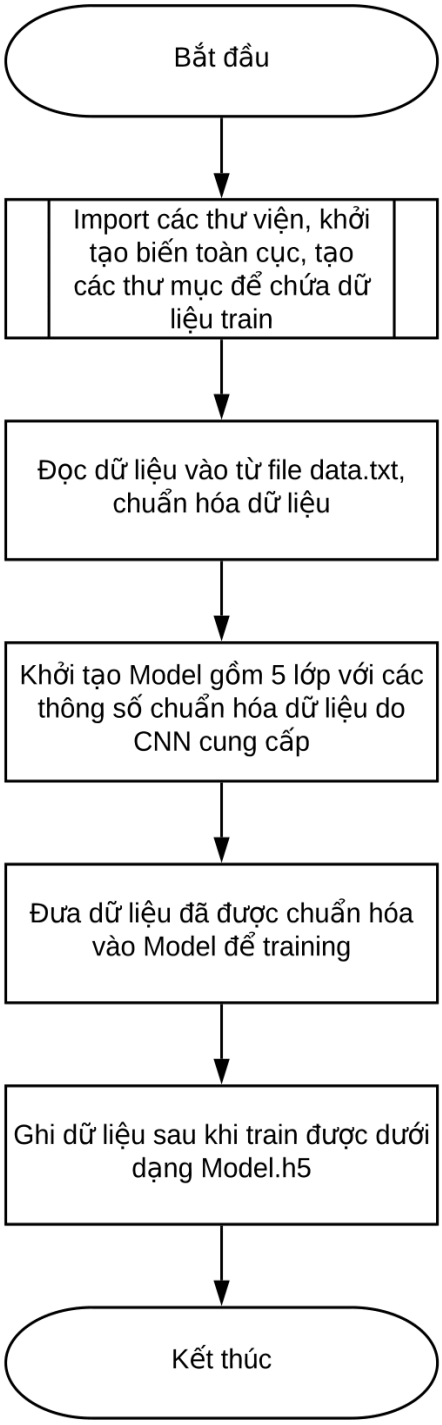
Khi đã có ảnh chưa đủ thông tin ta tiến hành nén ảnh về kích thước nhỏ nhất có thể để tập dữ liệu của chúng ta có đủ thông tin nhưng vẫn có dung lượng phù hợp để tăng tốc độ sử lí của của phần cứng cũng như độ chính xác của phần mềm.

Hình 3.14: Ảnh sau khi nén (*75x200*)

Sau khi đã tiến hành các bước khởi tạo, đọc và chuẩn hóa hình ảnh đầu vào thì ta bắt đầu việc đọc và ghi dữ liệu đã được chuẩn hóa vào tệp tương ứng. Ở mô hình này có tất cả là 7 tệp dữ liệu tương ứng với 2 góc cua trái, 1 góc thẳng, 2 góc thẳng, 1 tệp chứa hình ảnh có biển báo stop, 1 tệp chứa hình ảnh có đèn giao thông màu đỏ. Điều khiển mô hình xe chạy bằng tay theo lane có sẵn, với mỗi góc quay của servo, camera sẽ chụp hình ảnh và lưu vào các tệp tương ứng đã nêu trên. Với trường hợp đèn đỏ hay biển báo stop thì dừng xe lại và nhấn phím để điều khiển việc ghi lại hình ảnh và lưu vào tệp.

Cuối cùng ta đọc hết tất cả các thư mục dữ liệu rồi ghi vào một tệp có tên là data.txt duy nhất theo thứ tự để thuận tiện cho việc load dữ liệu và train model sau này.

3.3.2 Thiết lập các model

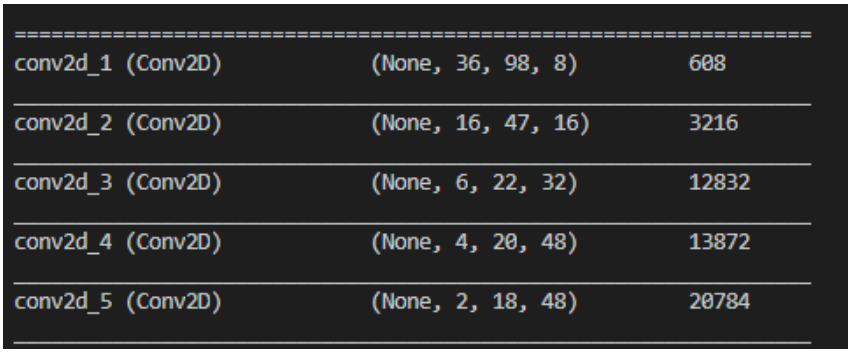


Hình 3.15: Lưu đồ tạo model và tập train.

Sau khi đã có dữ liệu đầu vào ta tiến hành tạo model và đưa vào huấn luyện theo lưu đồ trên.

```
model = Sequential()
model.add(Conv2D(8, kernel_size=(7, 7), activation='relu',strides=(2, 2),input_shape= (60,160,3),padding='valid',))
model.add(Conv2D(16, kernel_size=(5, 5),  activation='relu', strides=(2, 2),padding='valid',))
model.add(Conv2D(32, kernel_size=(5, 5), activation='relu',strides=(2, 2),padding='valid',))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu',strides=(1, 1),padding='valid',))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(number_class, activation='softmax'))# ví dụ 1 matrix thì + giữ nguyên, - thì = 0
model.summary()
```

Đầu tiên ta phải khởi tạo các thư viện cần thiết để thực hiện việc tạo model và huấn luyện. Các thư viện chính được dùng ở đây như là cv2, keras, sklearn, numpy… Khi đã đọc được ảnh từ tệp thì tiến hành tạo model để huấn luyện. Ở đây model gồm 5 lớp, có số chanel lần lượt là 8, 16, 32, 48, 48.



conv2d_1 (Conv2D)	(None, 36, 98, 8)	608
conv2d_2 (Conv2D)	(None, 16, 47, 16)	3216
conv2d_3 (Conv2D)	(None, 6, 22, 32)	12832
conv2d_4 (Conv2D)	(None, 4, 20, 48)	13872
conv2d_5 (Conv2D)	(None, 2, 18, 48)	20784

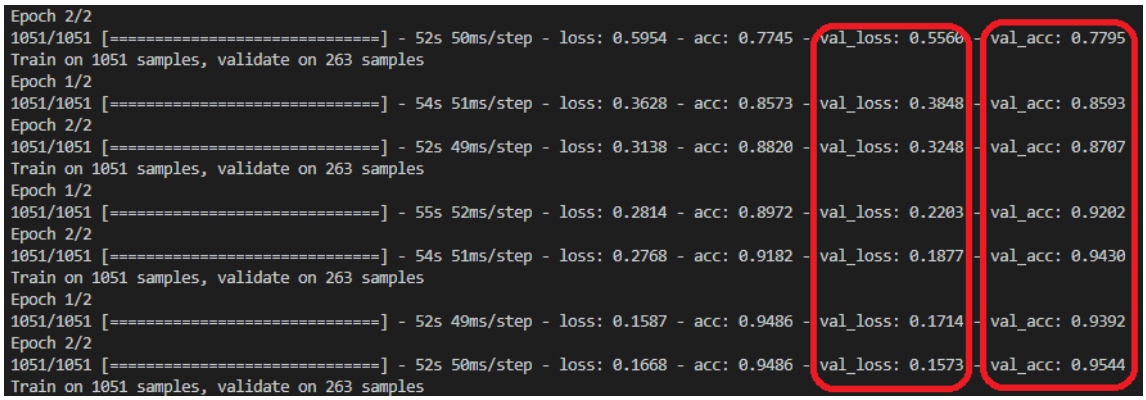
Hình 3.16: Các lớp của model được huấn luyện.

Sau khi tạo xong model ta tiến hành chia dữ liệu để train và test.

X_train, X_test = X[train_index], X[test_index]

y_train, y_test = y[train_index], y[test_index]

Ở đây model sẽ chia tập dữ liệu thành 10 phần, dùng 80% để train và 20% để test.

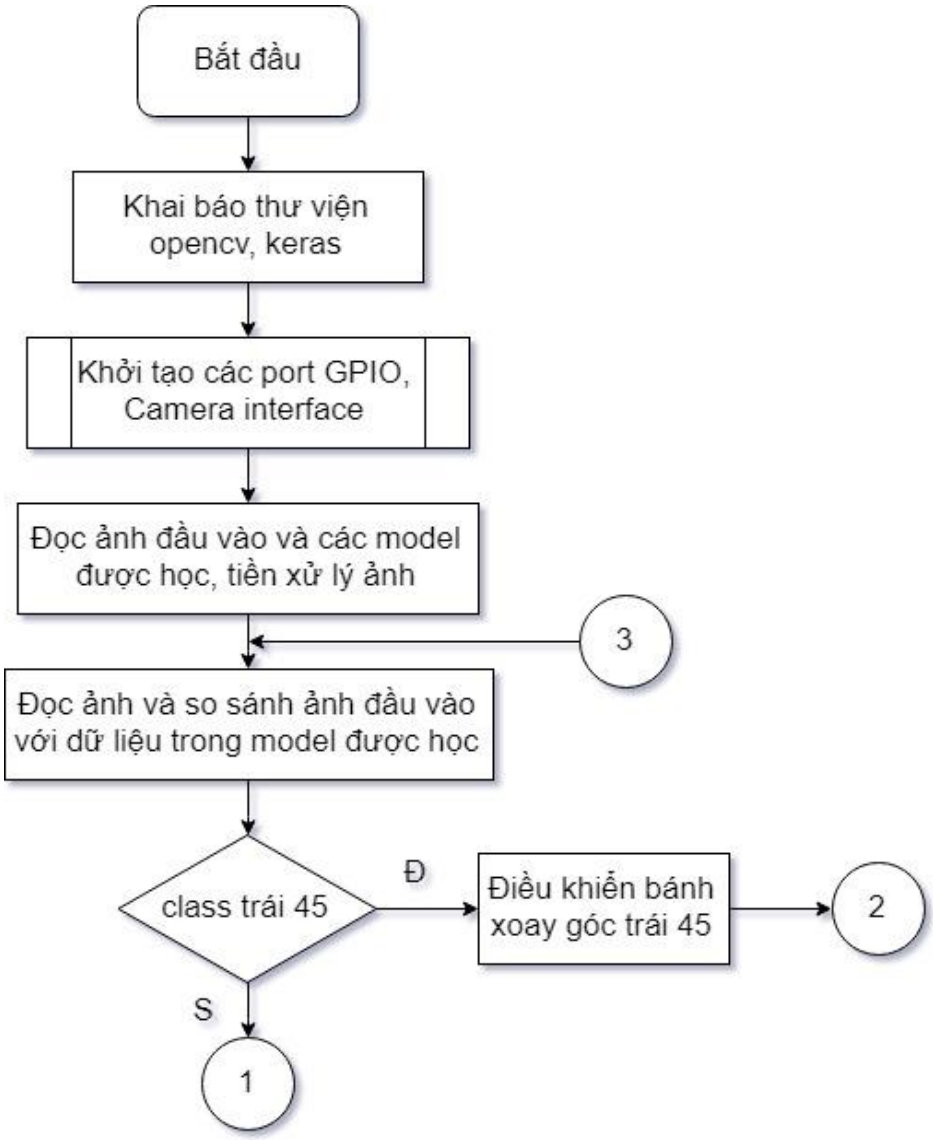


Hình 3.17: Kết quả sau khi huấn luyện.

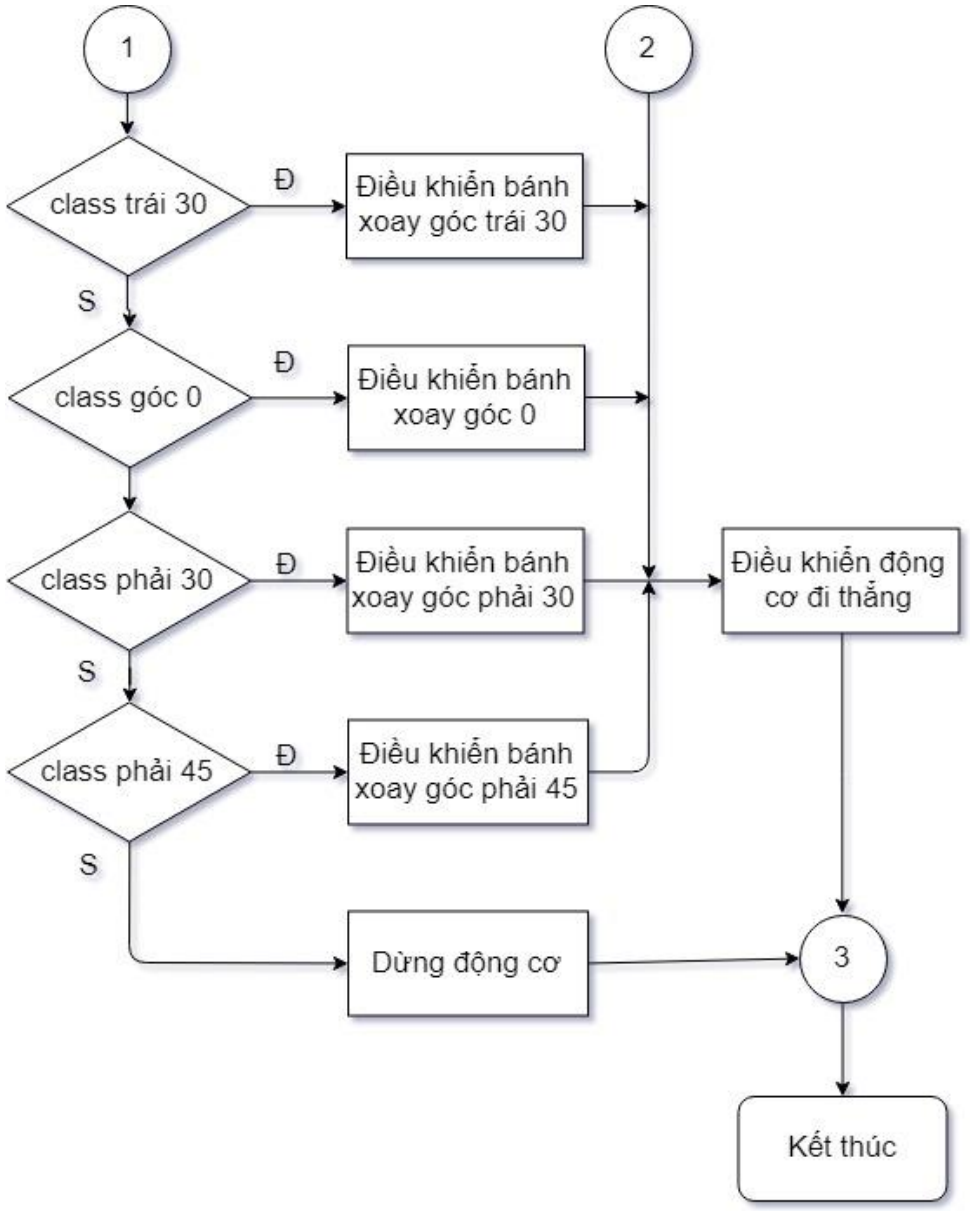
Cuối cùng! Nhóm đã đào tạo mô hình về làn đường và bằng cách quan sát độ chính xác và tổn thất đào tạo, ta có thể nói rằng mô hình đã làm rất tốt vì sau nhiều lần train, độ chính xác đào tạo là 94% và tổn thất đào tạo là khá thấp.

3.3.3 Nhận dạng hình ảnh làn đường

Lưu đồ dò chạy theo làn đường



Hình 3.18: Lưu đồ nhận điện và chạy theo làn đường (P1).



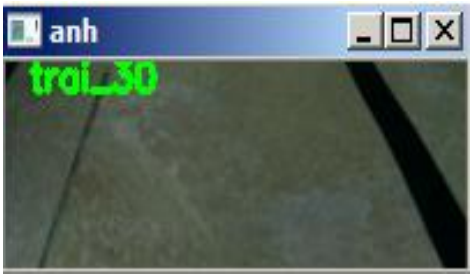
Hình 3.19: Lưu đồ nhận diện và chạy theo làn đường (P1).

Khi đã có model đã được huấn luyện ở phần trước, ta sẽ dùng nó để nhận diện và điều khiển. Đầu tiên phải khởi tạo các thư viện cần thiết giống với các phần trước. Sau đó đọc mode đã được huấn luyện vào chương trình.

Đọc ảnh đầu vào và tiền xử lý giống với phần thu thập dữ liệu để phù hợp với model đã được huấn luyện. Đưa ảnh vào nhận diện, theo như lưu đồ thì có 5 class ứng với thứ tự trong model để nhận diện làn đường và quẹo góc. Nếu nhận diện class bằng 0 tương ứng với góc quay trái 45 độ, tương tự cho đến class bằng

4 thì ứng với góc quay phải 45 độ. Một vài hình ảnh sau khi nhận diện bằng các model đã huấn luyện.

Hình 3.20: Góc thẳng 0^0.



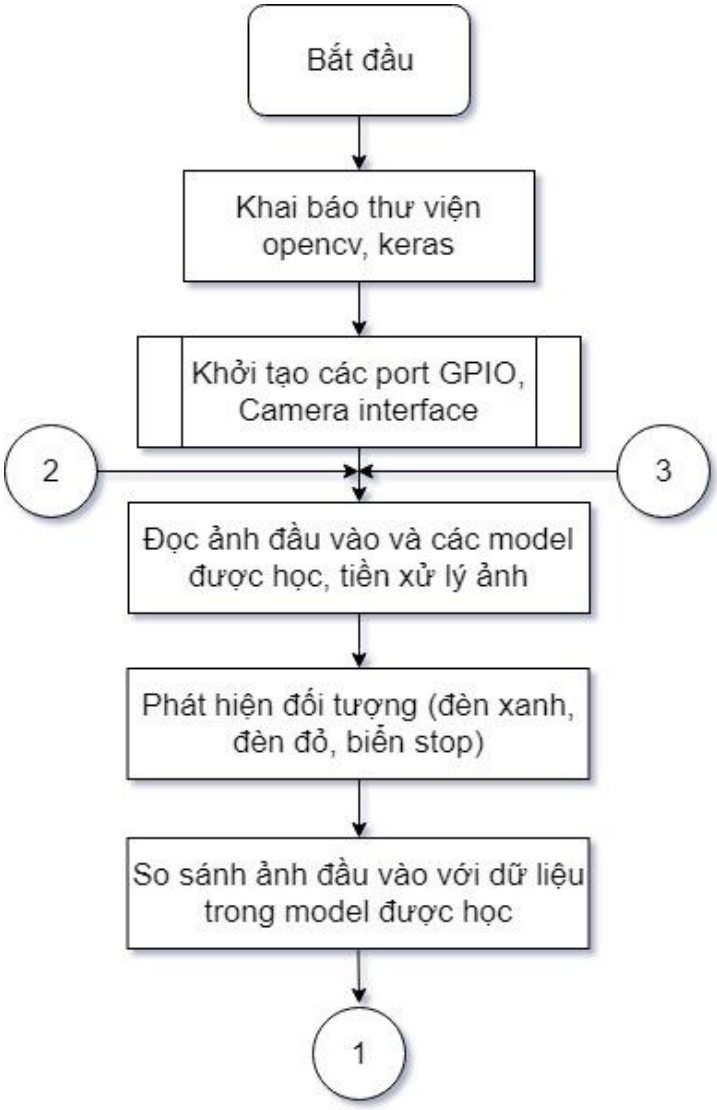
Hình 3.21: Góc trái 30^0.

Hình 3.22: Góc trái 45^0.

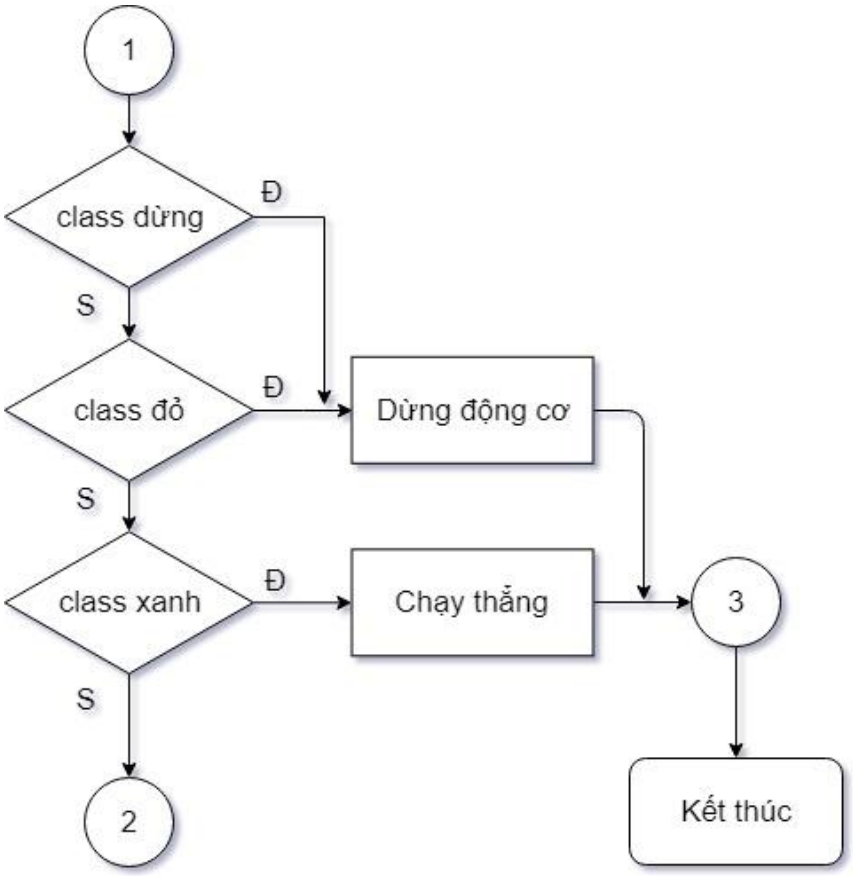
Hình 3.23: Góc phải 45^0.

3.3.4 Phát hiện và nhận diện biển báo và đèn giao thông

Lưu đồ phát hiện và nhận diện



Hình 3.24: Lưu đồ nhận diện biển bảo và đèn tín hiệu (P1).



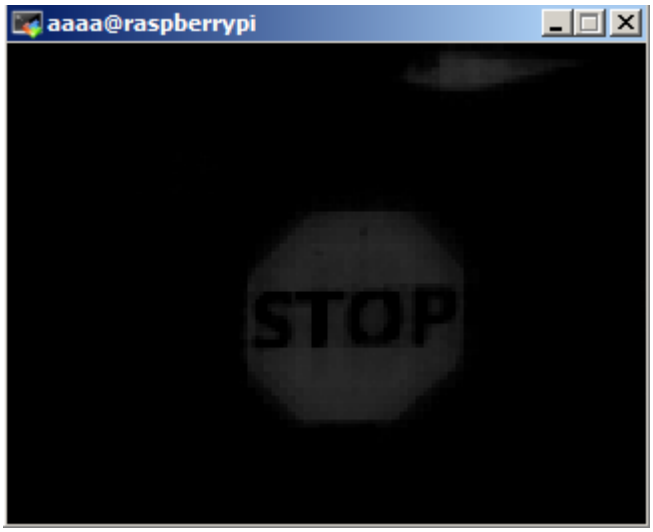
Hình 3.25: Lưu đồ nhận diện biển bảo và đèn tín hiệu (P2).

Việc đầu tiên chúng ta cần phải khởi tạo các thư viện của OpenCv để hỗ trợ cho việc thu thập dữ liệu. Tương tự các phần trên sau khi lấy được ảnh đầu vào ta tiến hành chuẩn hóa theo lưu đồ giải thuật trên.

Đọc ảnh đầu vào và tiền xử lý giống với phần thu thập dữ liệu để phù hợp với model nhận diện biển báo và đèn tín hiệu đã được huấn luyện. Đưa ảnh vào nhận diện, theo như lưu đồ thì có 3 class ứng với thứ tự trong model. Nếu nhận diện class bằng 0 tương ứng với biển báo dừng điều khiển dừng động cơ, class bằng 1 thì ứng với đèn đỏ điều khiển dừng động cơ và class bằng 2 tương ứng với đèn xanh điều khiển là xe chạy thẳng.

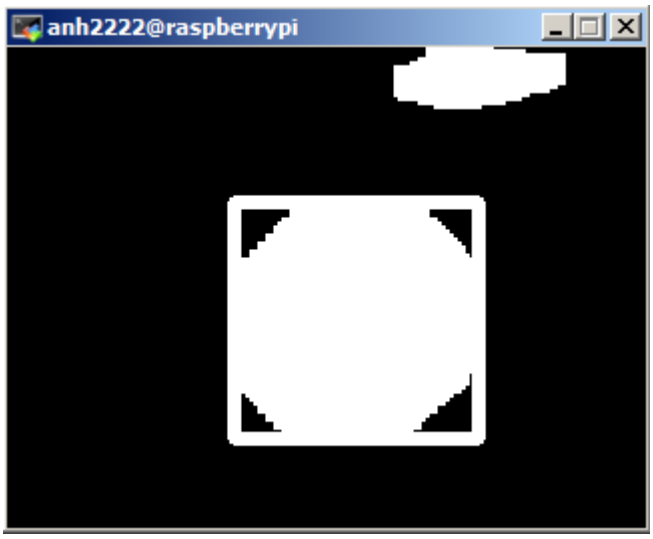
Ảnh đầu vào chưa rất nhiều thông tin, để phát hiện được các đối tượng có trong ảnh ta phải xử lí hình ảnh. Ở đây có 3 đối tượng cần phát hiện đó là biển báo Stop, đèn xanh, đèn đỏ. Ảnh đầu vào ta sẽ chuyển sang ảnh xám, sau đó trừ

đi nền với kênh màu là đỏ và xanh để tách được 2 đối tượng. Sau khi trừ nền và chuyển thành ảnh nhị phân ta tiến hành gộp 2 ảnh lại để phát hiện tất cả các đối tượng có màu xanh và màu đỏ cùng trên một ảnh.



Hình 3.26: Chuyển ảnh đầu vào thành ảnh xám.

Khi đã có ảnh nhị phân của các đối tượng, ta tiến hành tìm đường biên của đối được đó để tách nó ra khỏi ảnh để tiến hành nhận diện. OpenCv có hàm contours hỗ trợ chúng ta tìm được đường biên của đối tượng khi đã phát hiện. Các thông số tọa độ chiều rộng chiều cao của đối tượng được xác định bởi hàm contours.



Hình 3.27: Vẽ đường biên của đối tượng trên ảnh nhị phân.

Sau khi phát hiện được đối tượng và đường biên, ta tiến hành cắt lấy đối tượng trên ảnh nhị phân để đưa vào model training và nhận diện.

Hình 3.28: Cắt đối tượng ra khỏi ảnh nhị phân.

Và cuối cùng là kết quả sau khi tiến hành nhận diện đối tượng sau các quá trình biến đổi và xử lí.

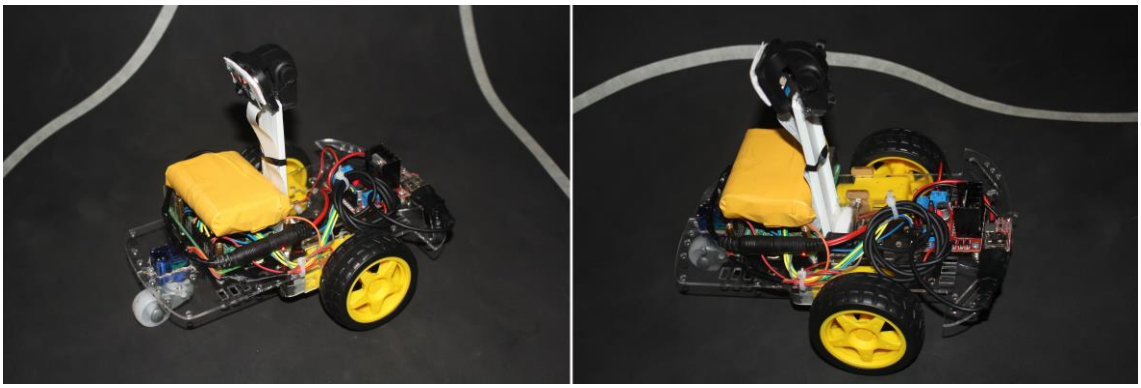
CHƯƠNG 4

KẾT QUẢ VÀ ĐÁNH GIÁ

4.1 KẾT QUẢ

Sau 15 tuần thiết kế và thi công, nhóm đã hoàn thành mô hình xe tự hành. Mô hình xử lý được 8 frame/s đối với chương trình nhận diện làn đường, 3 frame/s đối với chương trình nhận diện biển báo và đèn tín hiệu nên tốc độ xe khá chậm so với dự kiến.

Một số hình ảnh kết quả nhóm thực hiện được thể hiện bên dưới.

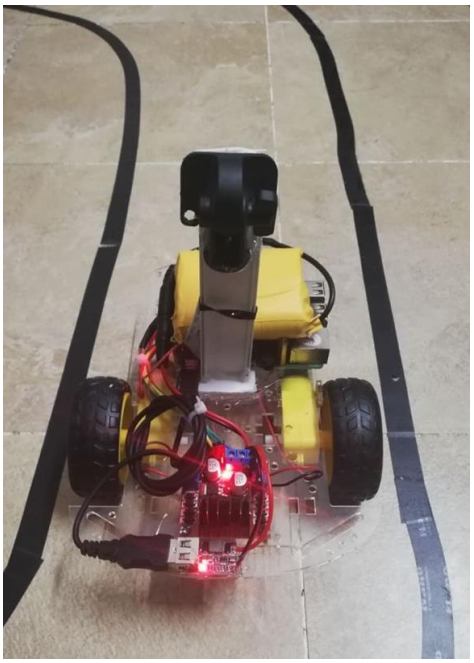


Hình 4.1: Mô hình xe tự hành.

Hình 4.1 là mô hình hoàn thiện của chiếc xe bao gồm đầy đủ các thành phần như đã nêu ở phần thiết kế.

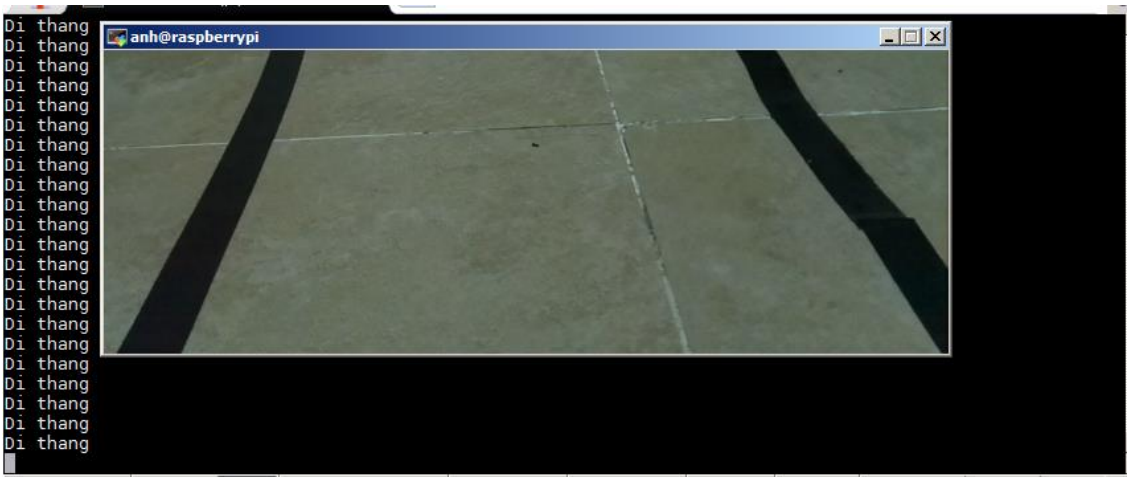
Một số kết quả chụp được từ thực tế và kết quả in ra trên máy.

Đi thẳng



Hình 4.2: Hình ảnh thực tế xe đi thẳng.

Hình trên chụp được lúc xe đang chạy thẳng theo làn lý tưởng.

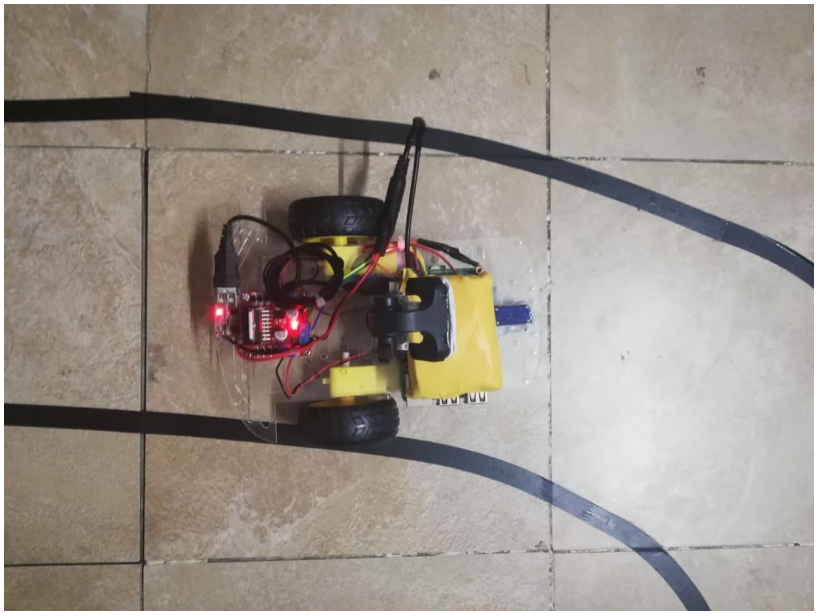


Hình 4.3: Kết quả in ra trên máy tính xe đang đi thẳng.

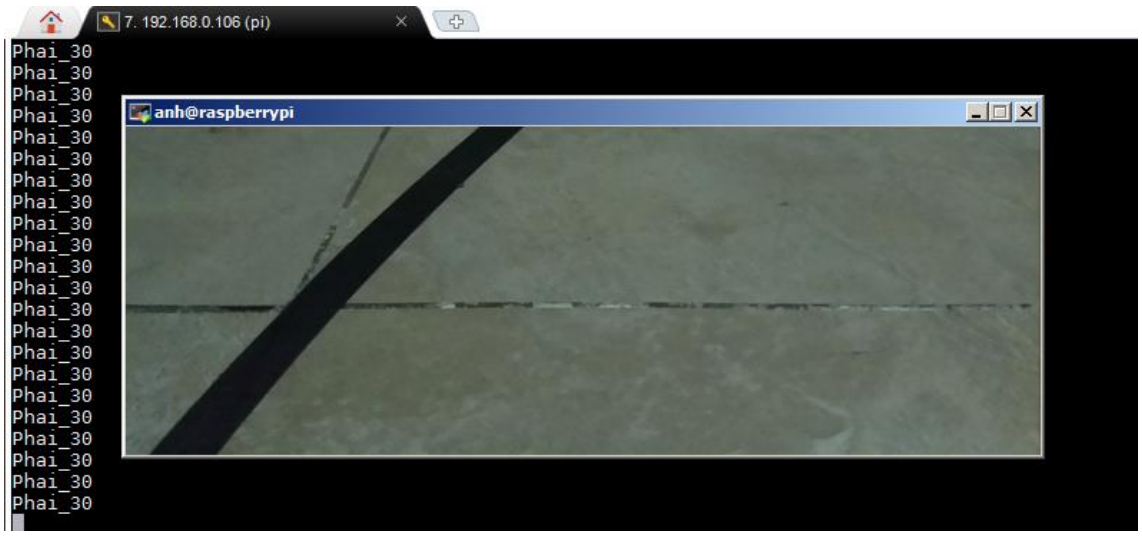
Hình 4.3 là ảnh chụp màn hình lúc xe đang chạy thẳng và in kết quả ra màn hình được nhóm so sánh thực tế. Khi xe chạy, camera tiến hành chụp lại ảnh và gửi về bộ xử lý trung tâm, bộ xử lý trung tâm xử lý để phát hiện ra các đặc trưng, sau đó đem so sánh với các đặc trưng trong model đã train và cho ra kết quả bức ảnh trên là đường thẳng

Quẹo phải

Tương tự như lúc đi thẳng, lúc xe tiến đến góc rẽ phải 30^0 như hình 4.4, bộ xử lý trung tâm so sánh và đưa ra kết quả như hình 4.5.



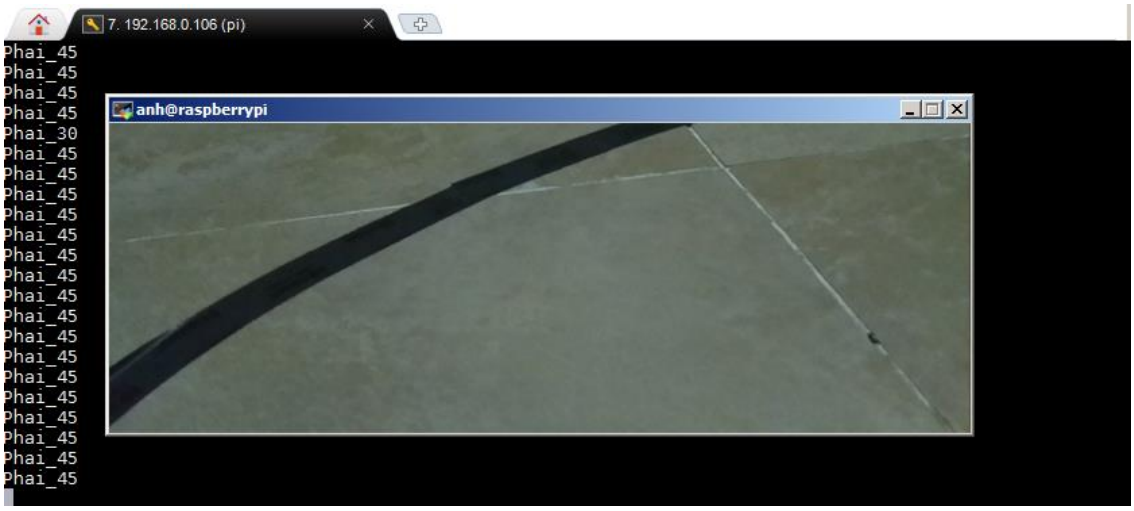
Hình 4.4: Xe đang quẹo góc phải 30^0.



Hình 4.5: Kết quả in ra trên máy tính xe đang quẹo góc phải 30^0.

Hai hình trên so sánh kết quả thực tế và kết quả trên máy tính.

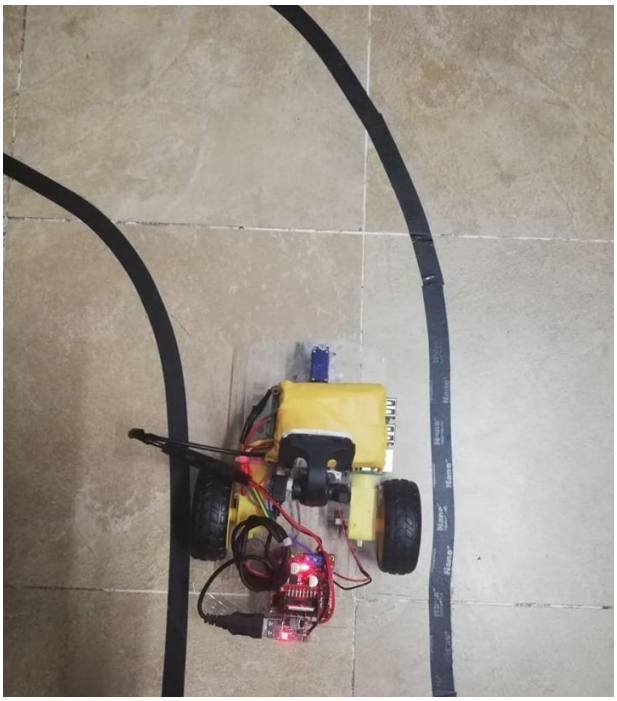
Hình 4.6: Xe đang quẹo góc phải 45^0.



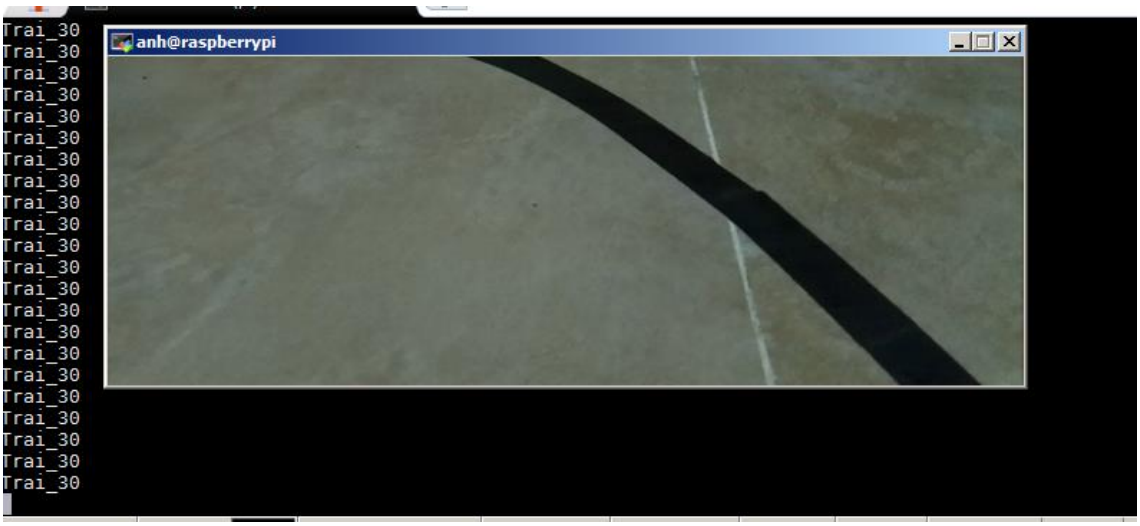
Hình 4.7: Kết quả in ra trên máy tính xe đang quẹo góc phải 45^0.

Các kết quả rẽ trái cũng tương tự như trên.

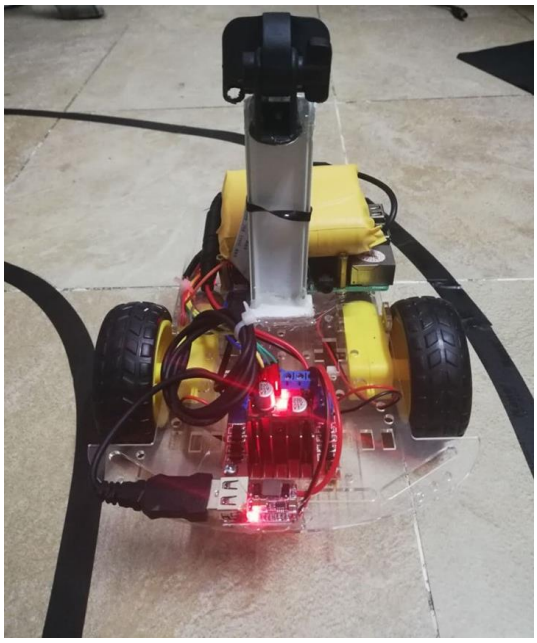
Quẹo trái



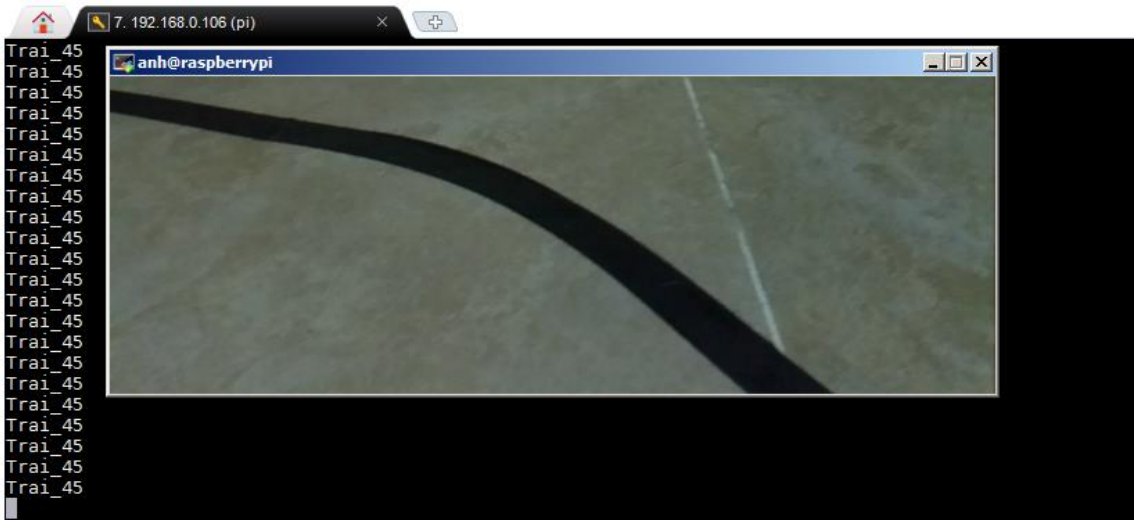
Hình 4.8: Xe đang quẹo góc trái 30^0.



Hình 4.9: Kết quả in ra trên máy tính xe đang quẹo góc trái 30^0.



Hình 4.10: Xe đang quẹo góc trái 45^0.

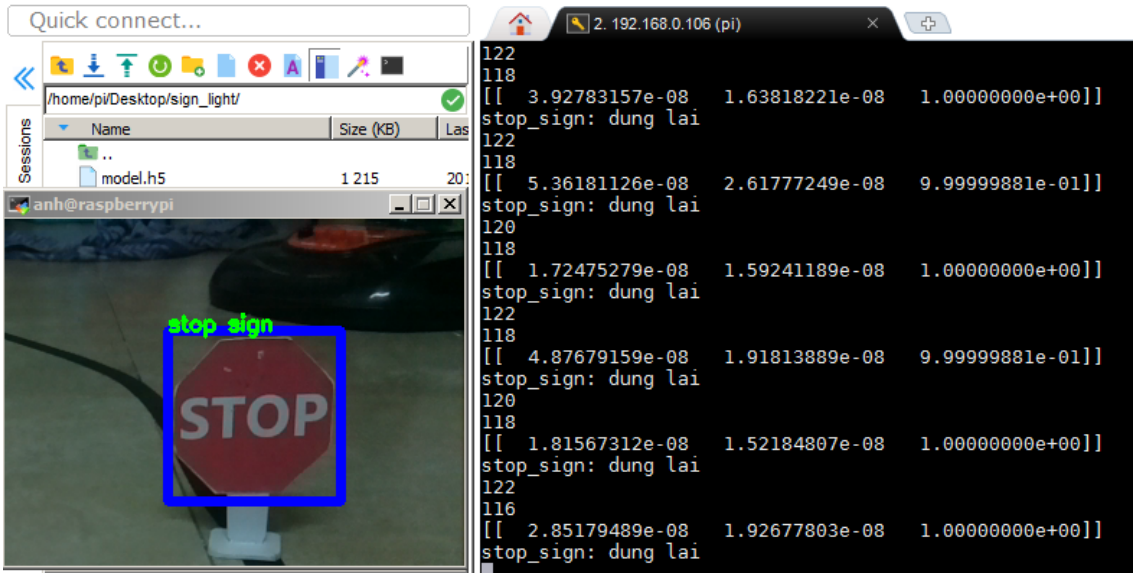


Hình 4.11: Kết quả in ra trên máy tính xe đang quẹo góc trái 45^0.

Biển Stop

Hình 4.12 là kết quả hiển thị trên mà hình máy chủ, đây là trường hợp khi xe gặp biển báo STOP. Khác với quá trình phát hiện lane, với biển báo STOP thì sau khi camera truyền hình ảnh tới bộ xử lý trung tâm, bộ xử lý trung tâm xử lý thuật toán phát hiện cùng biển báo trên bức ảnh rồi mới tìm đặc trưng để so sánh

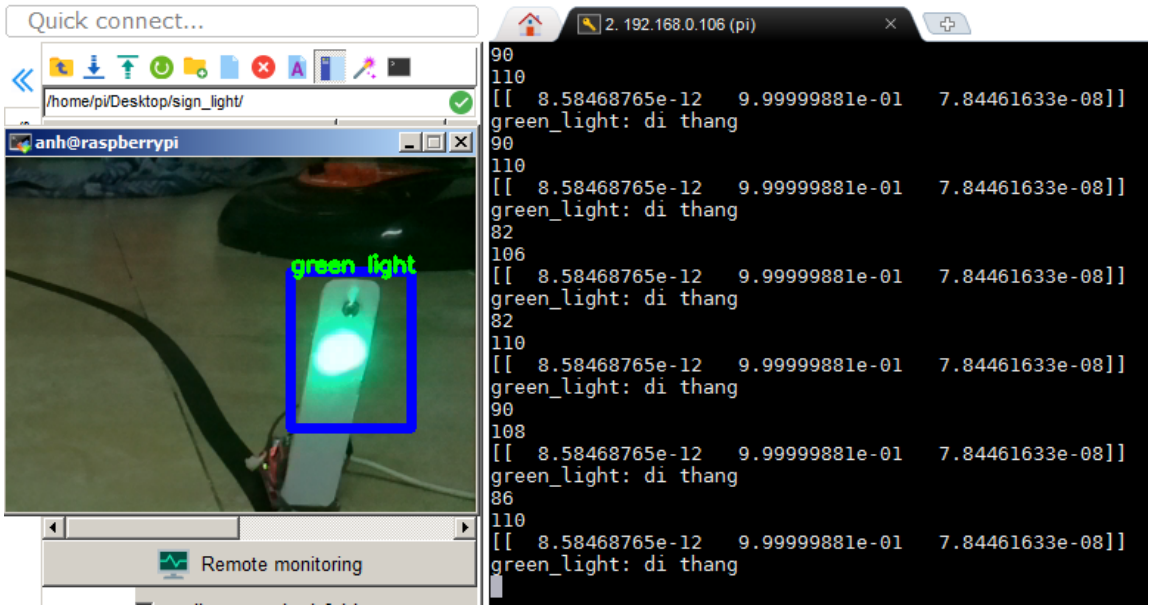
với model. Cuối cùng cho ra kết quả stop_sign, lúc này xe sẽ dừng lại đến khi không thấy biển báo nữa.



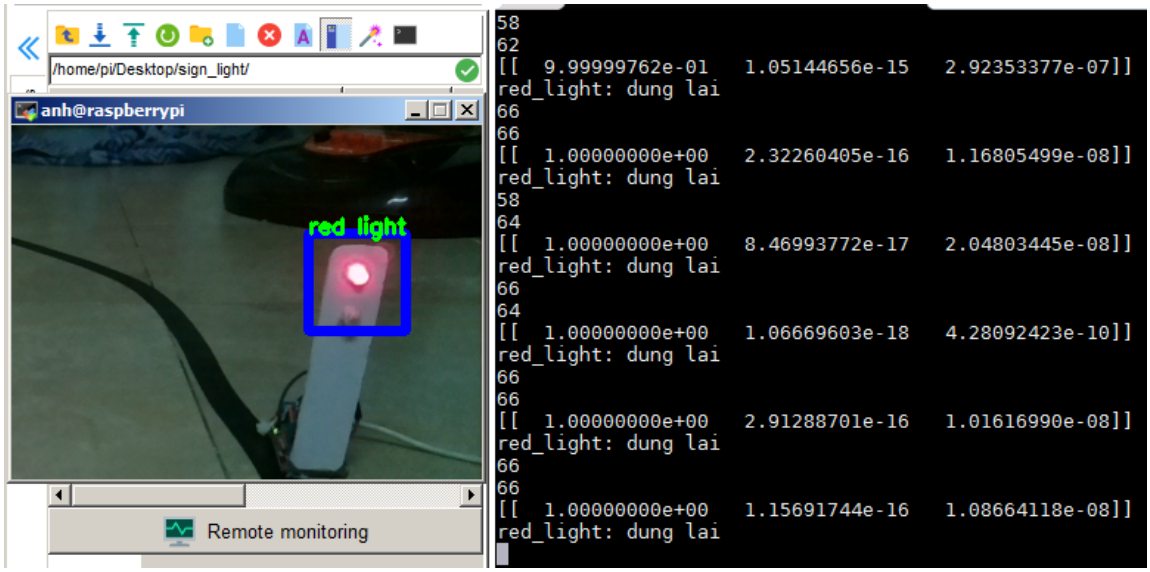
Hình 4.12: Nhận diện biển stop.

Đèn tín hiệu

Tương tự như trên, quá trình nhận diện đèn giao thông giống như việc nhận diện biển báo STOP. Khi gặp đèn xanh, xe tiếp tục đi thẳng như hình 4.13. Khi gặp đèn đỏ, xe dừng lại như hình 4.14.



Hình 4.13: Nhận diện đèn xanh.



Hình 4.14: Nhận diện đèn đỏ.

4.2 ĐÁNH GIÁ

Dựa vào kết quả bên trên thấy được kết quả thực tế và kết quả in ra hoàn toàn đúng. Tuy nhiên nhóm đã thử rất nhiều lần và đây là kết quả chính xác nhất, vẫn

còn những sai số xảy ra trong quá trình xe đang chạy như là xe chạy ngoài làn hay dẫm lên vạch. Vấn đề này vẫn có thể khác phục nhưng không chính xác hoàn toàn, đơn giản vì đó là một chiếc máy được lập trình vẫn thì vẫn có thể sai xót.

Trong nhưng trường hợp được xem là môi trường không lý tưởng đối với mô hình mà nhóm đã thực hiện như là mất một đoạn line do sự cố hay môi trường ánh sáng yếu,…Nếu đối với line bị mất đi những đoạn nhỏ thì độ chính xác khi nhận diện của xe không thay đổi nhiều so với bình thường, vì trong tập dữ liệu train, nhóm đã cố gắng thu thập nhiều ảnh với các trường hợp bị mất line khác nhau. Còn đối với việc mất các đoạn line dài so với kích thước ảnh nhỏ thì hệ thống sẽ có những kết quả sai lệch dẫn đến xe sẽ chạy sai. Cũng như việc mất các đường line, việc ánh sáng quá thấp cũng sẽ ảnh hưởng đến kết quả xử lí ảnh của hệ thống.

Công trình trên nhóm nghiên cứu trên mô hình lý tưởng, chưa thể đem vào áp dụng trong thực tế. Môi trường và điều kiện ánh sáng có ảnh hưởng rất lớn đến mô hình này, chính vì thế vì những lý do trên làm cho xe chạy sai là điều bình thường dựa theo kinh nghiệm nhóm quan sát được trong quá trình thực hiện đồ án.

CHƯƠNG 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

Với sự hỗ trợ nhiệt tình từ giáo viên hướng dẫn nhóm thực hiện đề tài, về cơ bản đã hoàn thành được những mục tiêu đề ra là thiết kế mô hình xe tự hành có thể nhận diện biển báo stop, nhận diện đèn tín hiệu và chạy theo làn đường. Trong quá trình thực hiện, nhóm gặp không ít khó khăn từ phần cứng đến phần mềm. Mặc dù gặp không ít khó khăn như thế nhưng nhóm đã cố gắng nghiên cứu vượt qua và tích lũy cho mình một số kinh nghiệm mới, kiến thức mới.

5.1.1 Ưu điểm

Phần cứng lắp ráp nhanh gọn, láp ráp nhanh chóng.

Với việc áp dụng deep learning vào mô hình cho ra tỷ lệ chính xác cao hơn các phương thức khác.

Thư viện keras dễ sử dụng, được hỗ trợ nhiều, dễ dàng nâng cấp hệ thống.

5.1.2 Nhược điểm

Chi phí phần cứng khá cao.

Còn ảnh hưởng nhiều bởi nhiễu.

Chỉ chạy được trên môi trường lý tưởng.

Phần cứng tự tạo nên chưa đạt độ chính xác cao.

Cấu hình của kit Raspberry pi quá yếu nên việc thứ nhất là không thể thực hiện việc huấn luyện các model ngay trên kit, thứ 2 là không thể cùng xử lý nhiều vấn đề (làn đường, đèn giao thông và biển báo) trên cùng 1 ảnh. Điều này làm cho mô hình chưa được tối ưu về mặt giải pháp.

5.2 HƯỚNG PHÁT TRIỂN ĐỀ TÀI

Thu thập nhiều dữ liệu train để tăng tính chính xác của mô hình.

Cải thiện thuật toán để tăng tốc độ xử lý.

Thiết kế khả năng phát hiện , tránh người và vật cản.

Sử dụng phần cứng chuyên dụng để tăng tốc độ xử lý.

Trang bị thêm cảm biến để thu thập dữ liệu.

Để khắc phục được các nhược điểm đã nêu ra, tăng tốc độ xử lý và đạt kết quả chính xác cao thì cần phải thay thế khối xử lý trung tâm bằng một kit khác có cấu hình mạnh hơn như là board Jetson TK1. Board Jetson TK1 là một máy tính nhúng được thiết kế trên nền tảng máy tính trí tuệ nhân tạo, được sản xuất bởi hãng NVIDIA. Trái tim của Jetson TK1 Developer Kit chính là bộ xử lý di động Tegra K1, là chip di động 192 nhân của NVIDIA, được xây dựng dựa trên kiến trúc NVIDIA Kepler, là GPU hiệu quả về điện năng nhất hiện nay. Toàn bộ 192 nhân có thể lập trình được của Tegra K1 mang lại năng lực tính toán và xử lý đồ hoạ mạnh nhất cho các thiết bị dạng di động,…

TÀI LIỆU THAM KHẢO

[1] Hải, N. T. (2014). In *Giáo trình xử lý ảnh* (pp. 24-27). TP. HCM.

[2] Macpherson, K. (2016). *Instructables*. Raspberry PI L298N Dual H Bridge DC Motor: https://www.instructables.com/id/Raspberry-PI-L298N-Dual-H-Bridge-DC-Motor/

[3] Nielsen, M. (2018, 10 2). *Introducing convolutional networks*. Neural Networks and Deep Learning: http://neuralnetworksanddeeplearning.com/chap6.html#introducing_convolutional_networks

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[5] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014, January). Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning (pp. 647-655).

[6] Battle of the Deep Learning frameworks—Part I: 2017, even more frameworks and interfaces

[7] Keras hompage

[8] Comparison of deep learning software – Wikipedia

[9] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

[10] https://scikit-learn.org/stable/