ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

#### TRƯỜNG ĐẠI HỌC SÀI GÒN

# HUỲNH GIA HÂN

**LÂM QUỐC PHÚ**

**ỨNG DỤNG AI TRONG PHÂN LOẠI CÀ CHUA**

###### **KHÓA LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC**

**TP. HỒ CHÍ MINH, THÁNG 05 NĂM 2023.**

ỦY BAN NHÂN DÂN TP. HỒ CHÍ MINH

#### TRƯỜNG ĐẠI HỌC SÀI GÒN

# HUỲNH GIA HÂN

**LÂM QUỐC PHÚ**

**ỨNG DỤNG AI TRONG PHÂN LOẠI CÀ CHUA**

###### **KHÓA LUẬN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**TRÌNH ĐỘ ĐÀO TẠO: ĐẠI HỌC**

NGƯỜI HƯỚNG DẪN: TS. TRẦN QUANG HUY

**TP. HỒ CHÍ MINH, THÁNG 05 NĂM 2023.**

**LỜI CAM ĐOAN**

Chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em, các số liệu và kết quả nghiên cứu nêu trong khóa luận là trung thực và chưa từng được công bố trong bất kì một công trình nào khác.

**Tác giả khóa luận**

**LỜI CẢM ƠN**

Chúng em xin chân thành gửi lời cảm ơn đến thầy TS. Trần Quang Huy đã giúp cho chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này. Thầy đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến giúp nhóm chúng em hoàn thành tốt khoá luận của mình. Trong thời gian một học kỳ thực hiện đề tài, nhóm đã vận dụng những kiến thức nền tảng đã được tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm đã vận dụng tối đa những gì đã thu thập được để hoàn tất báo cáo khoá luận một cách tốt nhất.

Tuy nhiên, trong quá trình thực hiện, nhóm không tránh khỏi những thiếu sót. Chính vì vậy, nhóm rất mong muốn nhận được những sự góp ý từ phía thầy cô phản biện, nhằm hoàn thiện những kiến thức để nhóm chúng em có thể học tập thêm.

Một lần nữa, chúng em chân thành cảm ơn!

**TP. Hồ Chí Minh, tháng 5 năm 2023**

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH VÀ SƠ ĐỒ 1](#_Toc135695338)

[DANH MỤC CHỮ VIẾT TẮT VÀ CÁC CỤM TỪ ĐƯỢC SỬ DỤNG 2](#_Toc135695339)

[DANH MỤC BẢNG BIỂU 3](#_Toc135695340)

[LỜI MỞ ĐẦU 4](#_Toc135695341)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 6](#_Toc135695342)

[1.1 Tổng quan về đề tài 6](#_Toc135695343)

[1.2 Mục đích khoá luận tốt nghiệp 7](#_Toc135695344)

[1.3 Đối tượng nghiên cứu 7](#_Toc135695345)

[1.4 Phạm vi nghiên cứu 8](#_Toc135695346)

[1.5 Phương pháp nghiên cứu 8](#_Toc135695347)

[1.5.1 Phương pháp lý luận: 8](#_Toc135695348)

[1.5.2 Phương pháp thực tiễn: 8](#_Toc135695349)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 9](#_Toc135695350)

[2.1 Xử lý ảnh là gì 9](#_Toc135695351)

[2.2 Thị giác máy tính 10](#_Toc135695352)

[2.2.1 Giới thiệu về thị giác máy tính 10](#_Toc135695353)

[2.2.2. Thị giác máy tính hoạt động như thế nào 10](#_Toc135695354)

[2.3 CNN 11](#_Toc135695355)

[2.3.1 Giới thiệu về CNN 11](#_Toc135695356)

[2.3.2. Kiến trúc cơ bản của CNN 12](#_Toc135695357)

[2.4. Deep learning 14](#_Toc135695358)

[2.4.1. Sơ lược về học sâu 14](#_Toc135695359)

[2.4.2. Deep learning hoạt động như thế nào? 15](#_Toc135695360)

[2.4.3. Ảnh hưởng của deep learning đối với xử lý ảnh 16](#_Toc135695361)

[2.5 YOLO 17](#_Toc135695362)

[2.5.1. Giới thiệu mạng YOLO 17](#_Toc135695363)

[2.5.2 Kiến trúc mạng YOLO 17](#_Toc135695364)

[2.5.3 Nguyên lý hoạt động của mạng YOLO 19](#_Toc135695365)

[2.6. YOLOv8 20](#_Toc135695366)

[2.6.1. Giới thiệu mạng YOLOv8 20](#_Toc135695367)

[2.6.2. Điểm khác của YOLOv8 so với các 21](#_Toc135695368)

[2.6.2.1 Anchor 21](#_Toc135695369)

[2.6.2.2. Kết cấu mới 23](#_Toc135695370)

[2.6.2.3. Đóng mở rộng khảm 25](#_Toc135695371)

[CHƯƠNG 3: XÂY DỰNG MÔ HÌNH DỰA VÀO AI VÀ IoT 27](#_Toc135695372)

[3.1. Mô hình hệ thống 27](#_Toc135695373)

[3.2. Chuẩn bị bộ dữ liệu 28](#_Toc135695374)

[3.2.1. Bộ dữ liệu 1 28](#_Toc135695375)

[3.2.2. Bộ dữ liệu 2 29](#_Toc135695376)

[3.3. Huấn luyện mô hình 32](#_Toc135695377)

[3.3.1. Huấn luyện mô hình với bộ dữ liệu 1 32](#_Toc135695378)

[3.3.2 Huấn luyện mô hình với bộ dữ liệu 2 34](#_Toc135695379)

[3.5. Mô hình kết nối từ Camera đến hệ thống 35](#_Toc135695380)

[3.6. Phát triển chức năng ước lượng khối lượng trái cà chua 38](#_Toc135695381)

[3.7. Phát triển chức năng ước lượng ngày thu hoạch 40](#_Toc135695382)

[3.8. Giao diện hệ thống 44](#_Toc135695383)

[3.9. Cách chương trình hoạt động 45](#_Toc135695384)

[CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM 50](#_Toc135695385)

[4.1. Kết quả 50](#_Toc135695386)

[4.1.1. Kết quả huấn luyện bộ dữ liệu 1 50](#_Toc135695387)

[4.1.2. Kết quả huấn luyện bộ dữ liệu 2 50](#_Toc135695388)

[4.2. Thực nghiệm và đánh giá 51](#_Toc135695389)

[4.2.1. Thực nghiệm với điều kiện ngoài trời buổi sáng 52](#_Toc135695390)

[4.2.2. Thực nghiệm với điều kiện ngoài trời buổi tối có đèn 53](#_Toc135695391)

[Kết luận và kiến nghị 55](#_Toc135695392)

[Kết luận: 55](#_Toc135695393)

[Kiến nghị: 55](#_Toc135695394)

[TÀI LIỆU THAM KHẢO 58](#_Toc135695395)

# DANH MỤC HÌNH ẢNH VÀ SƠ ĐỒ

[Hình 1: Định nghĩa về CNN 12](#_Toc135697239)

[Hình 2: Kiến trúc cơ bản của CNN 12](#_Toc135697240)

[Hình 3: Các lớp của mạng nơ-ron tích chập 13](#_Toc135697241)

[Hình 4: Deep learning đối với Machine Learning và AI 14](#_Toc135697242)

[Hình 5: Các lớp của Deep Learning 15](#_Toc135697243)

[Hình 6: Các lớp của YOLO 18](#_Toc135697244)

[Hình 7: Cách mô hình YOLO xử lý ảnh 19](#_Toc135697245)

[Hình 8:Phác thảo của nhiệm vụ phát hiện đối tượng 22](#_Toc135697246)

[Hình 9: Minh họa về Anchor Boxes 23](#_Toc135697247)

[Hình 10: C2F Module YOLOv8 mới 24](#_Toc135697248)

[Hình 11: Minh họa về đóng mở rộng khảm 26](#_Toc135697249)

[Hình 12: Minh họa mô hình 27](#_Toc135697250)

[Hình 13: Minh họa file label của bộ dữ liệu 29](#_Toc135697251)

[Hình 14: Minh họa file label của dữ liệu 31](#_Toc135697252)

[Hình 15: Phác thảo mô hình phần cứng hệ thống sử dụng 35](#_Toc135697253)

[Hình 16: Các giai đoạn phát triên của cà chua 41](#_Toc135697254)

[Hình 17: Giao diện của hệ thống 44](#_Toc135697255)

[Hình 18: Cách chương trình hoạt động 45](#_Toc135697256)

[Hình 19: Các biến Firebase sử dụng khi giao tiếp với ESP8266 46](#_Toc135697257)

[Hình 20: Các biến Firebase sử dụng khi giao tiếp với ESP32-Cam 47](#_Toc135697258)

[Hình 21: Các biến Firebase sử dụng khi giao tiếp với ESP32-Cam 47](#_Toc135697259)

[Hình 22: Kết quả huấn luyện bộ dữ liệu 1 50](#_Toc135697260)

[Hình 23: Kết quả huấn luyện bộ dữ liệu 2 51](#_Toc135697261)

[Hình 24: Ảnh chụp thực nghiệm với điều kiện ngoài trời buổi sáng 52](#_Toc135697262)

[Hình 25: Ảnh chụp thực nghiệm với điều kiện ngoài trời buổi tối có đèn 53](#_Toc135697263)

# DANH MỤC CHỮ VIẾT TẮT VÀ CÁC CỤM TỪ ĐƯỢC SỬ DỤNG

|  |  |  |
| --- | --- | --- |
| **Từ** | **Dạng đầy đủ** | **Ý nghĩa** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 

# DANH MỤC BẢNG BIỂU

Bảng 1: Tham số sử dụng khi huấn luyện mô hình với bộ dữ liệu 1 33

[Bảng 2: Tham số sử dụng khi huấn luyện mô hình với bộ dữ liệu 2 35](#_Toc135697820)

[Bảng 3: Các thiết bị hệ thống sử dụng 37](#_Toc135697821)

[Bảng 4:Các biến Firebase sử dụng khi giao tiếp với ESP-8266 47](#_Toc135697822)

[Bảng 5:Các biến Firebase sử dụng khi giao tiếp với ESP32-Cam 48](#_Toc135697823)

[Bảng 6:Các biến Firebase sử dụng khi giao tiếp với giao diện 49](#_Toc135697824)

[Bảng 7: Kết quả thực nghiệm ban ngày 53](#_Toc135697825)

[Bảng 8: Kết quả thực nghiệm buổi tối 54](#_Toc135697826)

# LỜI MỞ ĐẦU

Cà chua là một loại rau quả phổ biến và được tiêu thụ rộng rãi trên khắp thế giới. Tuy nhiên, việc phân loại cà chua theo màu sắc và ước lượng khối lượng vẫn đang là một thách thức lớn đối với người nông dân và các nhà sản xuất. Những sai sót trong quá trình phân loại cà chua có thể dẫn đến lãng phí tài nguyên, giảm chất lượng sản phẩm và ảnh hưởng đến giá trị thương mại của chúng.

Trong bối cảnh đó, Trí tuệ nhân tạo (AI) đã trở thành một công nghệ tiên tiến được áp dụng để giải quyết vấn đề này. Việc ứng dụng AI trong phân loại cà chua theo màu sắc và ước lượng khối lượng đang ngày càng được quan tâm và sử dụng rộng rãi trong lĩnh vực nông nghiệp.

Trong đề tài này, nhóm nghiên cứu thiết kế và xây dựng một hệ thống thu thập dữ liệu ảnh cà chưa từ xa thông qua mô hình Internet kết nối vạn vật (IoT). Dữ liệu ảnh trái cà chua thu thập được từ hệ thống sẽ được lưu trữ trên Firebase và sau đó sẽ được truyền về máy tính để nhận dạng, phân tích, đánh giá chất lượng, mức độ phát triển và ước lượng thời gian thu hoạch và sản lượng của vườn cà chua. Quá trình đó được thực hiện thông qua các kỹ thuật xử lý ảnh và AI. Bên cạnh đó, nhóm còn tiến hành xây dựng mô hình chăm sóc vườn cà chua từ xa thông qua hệ thống IoT. Các hoạt động chăm sóc bao gồm đo nhiệt độ, độ ẩm của môi trường xung quanh, cường độ chiếu sáng, độ ẩm không khí. Ngoài ra, nhóm còn thiết lập các chế độ tưới nước tự động, điều khiển mái che để làm mát. Toàn bộ qui trình trên có thể được thiết lập tự động và vận hành bằng tay tùy theo mong muốn của người sử dụng.

Hi vọng với đề tài này, chúng ta sẽ có được những hiểu biết mới về sự ứng dụng của AI trong việc nhận dạng, chăm sóc cà chua, đóng góp vào việc nâng cao hiệu quả và năng suất của quá trình sản xuất

Khoá luận sẽ có nội dụng gồm 4 chương như sau:

**CHƯƠNG 1: GIỚI THIỆU**

Tại chương này, chúng em sẽ giới thiệu về đề tài cũng như các thông tin liên quan đến khoá luận.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

Ở chương 2, chúng em trình bày về định nghĩa, kiến trúc mạng yolo và nguyên lý hoạt động của mạng YOLO, chuẩn bị thiết bị.

**CHƯƠNG 3: XÂY DỰNG MÔ HÌNH DỰA VÀO AI VÀ IoT**

Tiếp tục với chương 3, chúng em sẽ giới thiệu mô hình hệ thống, quá trình chuẩn bị bộ dữ liệu, huấn luyện mô hình và chương trình phát hiện trái cà chua sử dụng hình ảnh từ camera, ước lượng khối lượng quả cà chua.

**CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM**

Trình bày kết quả quá trình huấn luyện đạt được, thực nghiệm và đánh giá.

# 

# 

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

## 1.1 Tổng quan về đề tài

Cà chua là một loại rau có giá trị dinh dưỡng cao và được sử dụng rộng rãi trong ẩm thực và y học. Một số lợi ích của cà chua đã được các nhà nghiên cứu chứng minh như: cải thiện sức khỏe tim mạch [1], tăng cường miễn dịch[2], hỗ trợ sức khỏe làn da[3], giảm nguy cơ ung thư[4]. Chính vì thế, cà chua đã và đang được nghiên cứu để có được phương pháp canh tác và chăm sóc phù hợp[5][6][7]. Đặc biệt, việc áp dụng công nghệ trong trồng cây cà chua giúp tăng năng suất và chất lượng sản phẩm[8][9].

Để phát hiện và theo dõi trái cà chua trên cây, phương pháp thông thường được sử dụng là sử dụng Drone để thu thập hình ảnh từ trên cao, sau đó phân tích để phát hiện và đánh giá tình trạng trái cà chua. Tuy nhiên, phương pháp này lại có những hạn chế nhất định. Trong môi trường thực tế, Drone có thể bị hạn chế bởi thời tiết và chi phí đầu tư cao[10]. Vì vậy, việc sử dụng Drone để giám sát và phát hiện bệnh hại trên trái cà chua có thể không hiệu quả trong điều kiện thời tiết xấu như gió mạnh, mưa hoặc sương mù. Điều này có thể ảnh hưởng đến khả năng sử dụng Drone trong việc giám sát và phát hiện các bệnh hại. Bên cạnh đó, giá thành ban đầu của Drone và các cảm biến kỹ thuật số rất cao, do đó việc đầu tư vào Drone và các thiết bị liên quan có thể không phù hợp với ngân sách của các nông dân hoặc nhà nghiên cứu có nguồn lực hạn chế.

Một giải pháp khác là sử dụng hệ thống phát hiện và theo dõi trái cà chua thông qua các thuật toán xử lý hình ảnh và AI. Hệ thống này sử dụng trực tiếp dữ liệu hình ảnh hoặc video từ các hệ thống camera giám sát để phát hiện trái cà chua, sau đó đưa qua các thuật toán xử lý hình ảnh và AI để phân tích và đánh giá tình trạng trái cà chua[11]. Một ưu điểm của hệ thống này là có thể tận dụng dữ liệu từ các hệ thống camera giám sát có sẵn và giá thành đầu tư ban đầu thấp hơn so với Drone.

Hiện nay, các mô hình học sâu, đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks - CNN)[12] đã trở thành kỹ thuật phổ biến và hiệu quả nhất cho các bài toán nhận dạng hình ảnh. CNN đã chứng minh được khả năng xử lý ảnh với độ chính xác cao và độ phức tạp tính toán thấp, bằng cách tìm hiểu các đặc trưng ẩn trong ảnh và tạo ra mô hình dự đoán. Kỹ thuật này đã được áp dụng rộng rãi trong các ứng dụng thực tế như phân loại ảnh y tế, nhận dạng khuôn mặt, nhận dạng biển báo giao thông, và nhiều ứng dụng khác.Ngoài ra, các mô hình học sâu khác như mạng nơ-ron tái cấu trúc (Recurrent Neural Networks - RNN)[13], mạng nơ-ron sinh (Generative Neural Networks - GAN)[14] và nhanh hơn nhưng rất chính xác như SSD [15] và YOLO [16] cũng đang được nghiên cứu và áp dụng rộng rãi trong các bài toán khác như xử lý ngôn ngữ tự nhiên, tạo ảnh nhân tạo, tạo âm nhạc, và nhiều bài toán khác.

Nhận dạng đối tượng trong thị giác máy tính bao gồm 2 giai đoạn chính:

* Phân loại hình ảnh (Image Classification) là quá trình phân loại ảnh thành các lớp khác nhau dựa trên nội dung của ảnh. Nhiệm vụ này tập trung vào việc phân loại đối tượng trong ảnh vào một số lớp đã được xác định trước, ví dụ như xe hơi, chó, mèo, người, cây cối, đồ vật, v.v.
* Định vị đối tượng (Object Localization) là quá trình xác định vị trí và giới hạn của đối tượng trong ảnh. Nhiệm vụ này yêu cầu hệ thống phát hiện được đối tượng trong ảnh và tìm ra vị trí của đối tượng đó. Định vị đối tượng thường được sử dụng để giải quyết các vấn đề liên quan đến định vị đối tượng và phát hiện đối tượng trong ảnh.

Trong bài báo trình bày cơ sở lý thuyết mạng YOLO và quy trình huấn luyện mô hình mạng YOLOv8 để nhận dạng trái cà chua. Sử dụng mô hình đã được huấn luyện để xây dựng chương trình nhận dạng trái cà chua, sau đó thực hiện các thực nghiệm nhận dạng từ đó đưa ra đánh giá hiệu năng của mô hình.

## 1.2 Mục đích khoá luận tốt nghiệp

Tìm hiểu và nghiên cứu sâu hơn về xử lý ảnh và các ứng dụng trong thực tế.

Từ đó thiết kế và thi công mô hình phân loại cà chua.

Điều khiển và giám sát hoạt động của mô hình.

## 1.3 Đối tượng nghiên cứu

Quả cà chua

Kỹ thuật xử lý ảnh.

Trí tuệ nhân tạo (AI) và internet kết nối vạn vật (IoT)

## 1.4 Phạm vi nghiên cứu

Ứng dụng AI và IoT trong việc xây dựng mô hình thu thập dữ liệu quả cà chua trực tiếp trên cây một cách tự động. Dữ liệu đó sẽ được nhận dạng và phân tích dựa trên các kỹ thuật xử lý ảnh và AI.

Điều khiển và giám sát quá trình chăm sóc vườn cà chua thông qua hệ thống IoT

## 1.5 Phương pháp nghiên cứu

### 1.5.1 Phương pháp lý luận:

Nghiên cứu tài liệu về Python.

Nghiên cứu về ứng dụng Pycharm.

Nghiên cứu về xử lý ảnh.

Nghiên cứu về AI.

Tìm kiếm, thu thập các bộ dữ liệu.

### 1.5.2 Phương pháp thực tiễn:

Cài đặt chạy thử thuật toán.

Thử nghiệm, so sánh kết quả trên dữ liệu thực tế.

Đánh giá và kiểm tra tính đúng đắn của phương pháp.

# 

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Xử lý ảnh là gì

Xử lý hình ảnh là một phương pháp thực hiện các thao tác trên hình ảnh để nâng cao chất lượng hoặc trích xuất thông tin hữu ích. Đây là một dạng xử lý tín hiệu, trong đó đầu vào là hình ảnh và đầu ra có thể là hình ảnh hoặc các đặc trưng/tính năng liên quan đến hình ảnh đó. Hiện nay, xử lý ảnh là một trong những công nghệ đang phát triển nhanh chóng và là lĩnh vực nghiên cứu cốt lõi trong các ngành kỹ thuật và khoa học máy tính.

Quá trình xử lý hình ảnh cơ bản bao gồm ba bước chính:

* Nhập hình ảnh thông qua các công cụ thu thập hình ảnh
* Phân tích và xử lý hình ảnh
* Đầu ra trong đó kết quả có thể là hình ảnh được cải thiện hoặc báo cáo dựa trên phân tích hình ảnh.

Có hai loại phương pháp được sử dụng để xử lý ảnh là xử lý ảnh tương tự và kỹ thuật số. Xử lý hình ảnh tương tự có thể được sử dụng cho các bản cứng như bản in và ảnh. Trong khi đó, các nhà phân tích hình ảnh sử dụng các nguyên tắc giải thích cơ bản khác nhau trong khi sử dụng các kỹ thuật này. Các kỹ thuật xử lý hình ảnh kỹ thuật số giúp xử lý các hình ảnh kỹ thuật số bằng cách sử dụng máy tính. Ba giai đoạn chung mà tất cả các loại dữ liệu đều phải trải qua khi sử dụng kỹ thuật số là tiền xử lý, tăng cường và hiển thị, khai thác thông tin.

Trong đó xử lý ảnh kỹ thuật số, một hình ảnh có thể được định nghĩa là một hàm hai chiều, f(x, y), trong đó x và y là tọa độ không gian (mặt phẳng), và độ lớn của f tại bất kỳ cặp tọa độ nào (x, y) được gọi là độ sáng hoặc mức xám của hình ảnh tại điểm đó. Khi x, y và giá trị độ lớn của f đều là các số hữu hạn, ta gọi hình ảnh đó là hình ảnh kỹ thuật số. Lĩnh vực xử lý ảnh kỹ thuật số đề cập đến việc xử lý các hình ảnh kỹ thuật số bằng cách sử dụng máy tính kỹ thuật số. Lưu ý rằng một hình ảnh kỹ thuật số được tạo thành từ một số hữu hạn các phần tử, mỗi phần tử có một vị trí và giá trị cụ thể. Những phần tử này được gọi là các phần tử hình ảnh, pels và pixels. Pixel là thuật ngữ được sử dụng phổ biến nhất để chỉ các phần tử của một hình ảnh kỹ thuật số.[17]

## 2.2 Thị giác máy tính

### 2.2.1 Giới thiệu về thị giác máy tính

Thị giác máy tính là một lĩnh vực trí tuệ nhân tạo (AI) cho phép máy tính và hệ thống lấy thông tin có ý nghĩa từ hình ảnh kỹ thuật số, video và các đầu vào trực quan khác — và thực hiện hành động hoặc đưa ra đề xuất dựa trên thông tin đó. Nếu AI cho phép máy tính suy nghĩ, thì thị giác máy tính cho phép chúng nhìn, quan sát và hiểu.

Thị giác máy tính hoạt động giống như thị giác của con người, ngoại trừ con người có một khởi đầu thuận lợi. Thị giác của con người có lợi thế về thời gian tồn tại của bối cảnh để đào tạo cách phân biệt các vật thể, khoảng cách của chúng, liệu chúng có đang chuyển động hay không và liệu có điều gì đó không ổn trong hình ảnh hay không.

Thị giác máy tính huấn luyện máy móc thực hiện các chức năng này, nhưng nó phải thực hiện điều đó trong thời gian ngắn hơn nhiều với máy ảnh, dữ liệu và thuật toán thay vì võng mạc, dây thần kinh thị giác và vỏ não thị giác. Vì một hệ thống được đào tạo để kiểm tra sản phẩm hoặc theo dõi tài sản sản xuất có thể phân tích hàng nghìn sản phẩm hoặc quy trình trong một phút, nhận thấy các lỗi hoặc sự cố không thể nhận thấy nên hệ thống này có thể nhanh chóng vượt qua khả năng của con người.

### 2.2.2. Thị giác máy tính hoạt động như thế nào

Thị giác máy tính cần rất nhiều dữ liệu. Nó chạy đi chạy lại các phân tích dữ liệu cho đến khi phân biệt được các điểm khác biệt và cuối cùng là nhận ra hình ảnh. Ví dụ, để huấn luyện máy tính nhận dạng lốp ô tô, nó cần được cung cấp một lượng lớn hình ảnh lốp và các mục liên quan đến lốp để tìm hiểu sự khác biệt và nhận ra lốp, đặc biệt là lốp không có khuyết tật.

Hai công nghệ thiết yếu được sử dụng để thực hiện điều này: một loại máy học được gọi là [học sâu](https://www.ibm.com/cloud/watson-studio/deep-learning) và mạng thần kinh tích chập (CNN).

Máy học sử dụng các mô hình thuật toán cho phép máy tính tự dạy về ngữ cảnh của dữ liệu trực quan. Nếu đủ dữ liệu được cung cấp thông qua mô hình, máy tính sẽ “xem xét” dữ liệu và dạy chính nó phân biệt hình ảnh này với hình ảnh khác. Các thuật toán cho phép máy tự học chứ không phải do ai đó lập trình để máy nhận dạng hình ảnh.

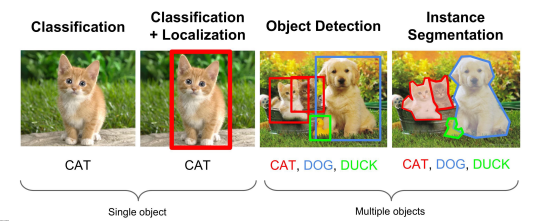
CNN giúp mô hình học máy hoặc học sâu “nhìn” bằng cách chia hình ảnh thành các pixel được gắn thẻ hoặc nhãn. Nó sử dụng các nhãn để thực hiện phép chập (một phép toán trên hai hàm để tạo ra hàm thứ ba) và đưa ra dự đoán về những gì nó “nhìn thấy”. Mạng thần kinh chạy các phép chập và kiểm tra tính chính xác của các dự đoán của nó trong một loạt các lần lặp lại cho đến khi các dự đoán bắt đầu trở thành sự thật. Sau đó, nó nhận ra hoặc nhìn thấy hình ảnh theo cách tương tự như con người.

Giống như con người tạo ra một hình ảnh ở khoảng cách xa, CNN trước tiên phân biệt các cạnh cứng và hình dạng đơn giản, sau đó điền thông tin khi chạy lặp lại các dự đoán của nó. Một CNN được sử dụng để hiểu các hình ảnh đơn lẻ. Mạng nơ-ron tái phát (RNN) được sử dụng theo cách tương tự cho các ứng dụng video để giúp máy tính hiểu các hình ảnh trong một loạt khung hình có liên quan với nhau như thế nào.[19]

## 2.3 CNN

### 2.3.1 Giới thiệu về CNN

Mạng nơ-ron tích chập (CNNs) là các thuật toán Trí tuệ nhân tạo dựa trên mạng nơ-ron đa tầng, học các đặc trưng có liên quan từ các hình ảnh và có khả năng thực hiện nhiều nhiệm vụ như phân loại đối tượng, phát hiện và phân đoạn.

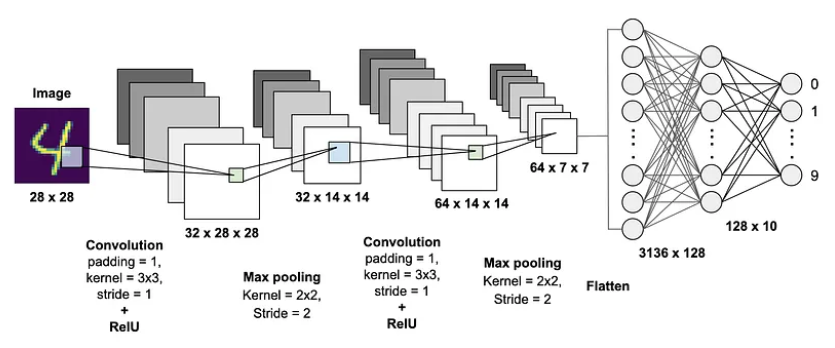


Hình 1: Định nghĩa về CNN

(nguồn: https://www.inf.ufpr.br/todt/IAaplicada/CNN\_Presentation.pdf)

Lợi thế của mạng nơ-ron tích chập so với các thuật toán phân loại khác (SVM, K-NN, Random-Forest và các thuật toán khác) là CNN học các đặc trưng tốt nhất để biểu diễn đối tượng trong các hình ảnh và có khả năng tổng quát hóa cao, có thể phân loại chính xác các ví dụ mới chỉ với vài ví dụ trong tập huấn luyện.[18]

### 2.3.2. Kiến trúc cơ bản của CNN

****

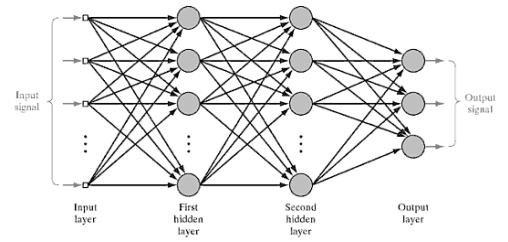
Hình 2: Kiến trúc cơ bản của CNN

(nguồn:[https://becominghuman.ai/building-a-convolutiona](https://becominghuman.ai/building-a-convolutional-neural-network-cnn-model-for-image-classification-116f77a7a236)

[l-neural-network-cnn-model-for-image-classification-116f77a7a236](https://becominghuman.ai/building-a-convolutional-neural-network-cnn-model-for-image-classification-116f77a7a236))

Một mạng nơ-ron tích chập thường được bố trí bởi bốn loại lớp sau:

* Lớp tích chập (Convolutional): Một lớp tích chập được tạo thành bởi một tập hợp các bộ lọc, còn được gọi là nhân, trượt qua dữ liệu đầu vào.
* Lớp gộp (Pooling): hoặc lớp giảm mẫu, được áp dụng để giảm chiều của các bản đồ đặc trưng sao cho giữ lại thông tin quan trọng nhất từ các bản đồ đặc trưng đó.
* Lớp Relu (ReLU): Các lớp ReLU (Rectified Linear Units) là các lớp kích hoạt được kết nối sau lớp tích chập để tạo ra tính phi tuyến trong mạng.
* Lớp kết nối đầy đủ (Fully Connected): là một MultiLayer Perceptron (MLP), được tạo thành bởi ba loại lớp: lớp đầu vào, lớp ẩn và lớp đầu ra.

****

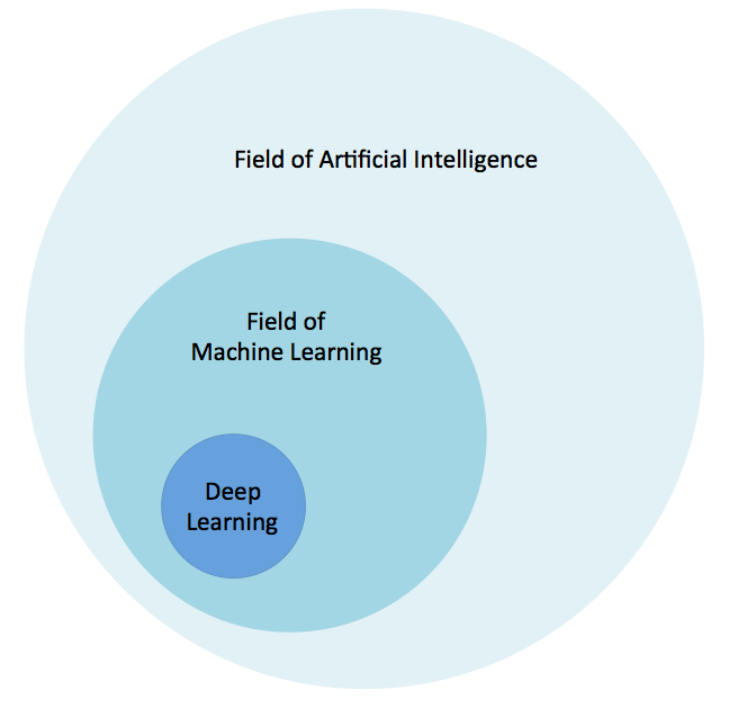
Hình 3: Các lớp của mạng nơ-ron tích chập

(nguồn : https://www.inf.ufpr.br/todt/IAaplicada/CNN\_Presentation.pdf)

* + Lớp đầu vào nhận các đặc trưng được tạo ra bởi CNN.
  + Lớp ẩn là một chuỗi các nơ-ron với trọng số sẽ được học trong quá trình huấn luyện. Một MLP được tạo thành bởi một hoặc nhiều lớp ẩn.
  + Lớp đầu ra cũng là một chuỗi các nơ-ron. Tuy nhiên, nó có một hàm kích hoạt khác. Thông thường, hàm softmax được sử dụng để tạo ra xác suất của mỗi danh mục trong phạm vi bài toán.

## 2.4. Deep learning

### 2.4.1. Sơ lược về học sâu



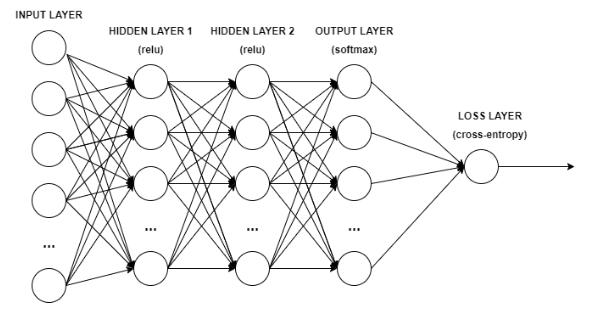
Hình 4: Deep learning đối với Machine Learning và AI

(nguồn: https://ybox.vn/ky-nang/tomo-gioi-thieu-ve-deep-learning-ymh4xrcgrr)

Deep Learning liên quan đến các thuật toán lấy cảm hứng từ cấu trúc và chức năng của não nhằm bắt chước trí thông minh của con người được gọi là Mạng thần kinh nhân tạo. Cụ thể hơn, nó bắt chước các chức năng phức tạp của não người để sử dụng dữ liệu phi cấu trúc để giải mã ý nghĩa và dạy cho máy móc, từ đó giúp máy móc tìm ra các giải pháp để hỗ trợ đưa ra các quyết định tự động và tạo ra kết quả bắt chước các mẫu hành vi của con người. Các mô hình học sâu có thể đạt được độ chính xác hiện đại, đôi khi vượt quá hiệu suất ở cấp độ con người.

Deep learning đã cho ra đời các kết quả đáng chú ý trong nhiều lĩnh vực, bao gồm nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên, phân tích hình ảnh, và tìm kiếm trên web. Deep learning cũng đã được áp dụng rộng rãi trong các ứng dụng công nghệ mới như xe tự lái và chẩn đoán y tế.

### 2.4.2. Deep learning hoạt động như thế nào?

****

Hình 5: Các lớp của Deep Learning

(Nguồn: Michael A.Nielsen,”Neural Network and Deep Learning”)

Các mạng nơ-ron sâu được cấu thành từ nhiều lớp (layer) nơ-ron liên kết với nhau. Mỗi lớp bao gồm một số lượng nơ-ron (neuron) nhất định và mỗi nơ-ron có một trọng số (weight) tương ứng với các liên kết giữa các lớp.

* Các lớp nơ-ron đầu tiên được gọi là lớp đầu vào (input layer) và nhận dữ liệu đầu vào.
* Lớp cuối cùng được gọi là lớp đầu ra (output layer) và sản xuất kết quả dự đoán.
* Các lớp ở giữa được gọi là lớp ẩn (hidden layer) Các lớp ẩn này xử lý thông tin ở các cấp độ khác nhau, thích ứng với hành vi của mình khi nhận được thông tin mới. Các mạng học sâu có hàng trăm lớp ẩn có thể được dùng để phân tích một vấn đề từ nhiều góc độ khác nhau. Các lớp ẩn trong mạng nơ-ron chuyên sâu hoạt động theo cùng một cách. Nếu một thuật toán học sâu đang cố gắng phân loại một hình ảnh động vật, mỗi lớp ẩn của thuật toán này sẽ xử lý một đặc điểm khác nhau của con vật và cố gắng phân loại chính xác nó.

Trong quá trình huấn luyện, mô hình deep learning được cung cấp dữ liệu huấn luyện (training data) để học từ đó và cập nhật các trọng số của các liên kết giữa các nơ-ron trong mạng nơ-ron sâu. Quá trình này được thực hiện bằng cách sử dụng thuật toán lan truyền ngược (backpropagation), trong đó các đạo hàm (derivatives) của hàm mất mát (loss function) được tính toán để điều chỉnh các trọng số.

Hàm mất mát (loss function) được sử dụng để đo lường sự khác biệt giữa kết quả dự đoán của mô hình và giá trị thực tế của dữ liệu. Mục tiêu của quá trình huấn luyện là tìm ra các trọng số tối ưu để giảm thiểu hàm mất mát. Khi đạt được một mức độ giảm mất mát đủ lớn và không có dấu hiệu của overfitting (quá khớp), mô hình được coi là đã học tốt và có thể được sử dụng để dự đoán trên dữ liệu mới.

### 2.4.3. Ảnh hưởng của deep learning đối với xử lý ảnh

Deep learning đã có những ảnh hưởng rất lớn đối với xử lý ảnh, giúp cho việc xử lý ảnh trở nên nhanh chóng và chính xác hơn. Trước khi deep learning ra đời, các phương pháp xử lý ảnh truyền thống thường dựa trên việc định nghĩa các đặc trưng (features) của ảnh bằng cách sử dụng các thuật toán tiền xử lý (preprocessing), và sau đó sử dụng các phương pháp phân loại (classification) để phân loại ảnh vào các lớp khác nhau. Tuy nhiên, cách tiếp cận này có những hạn chế như yêu cầu các chuyên gia phải định nghĩa và lựa chọn các đặc trưng thủ công, và không thể tự động học hỏi được các đặc trưng mới.

Với deep learning, các mô hình học máy có khả năng tự động học hỏi các đặc trưng từ dữ liệu ảnh mà không cần định nghĩa chúng thủ công. Điều này cho phép các mô hình deep learning tự động tìm ra các đặc trưng quan trọng của ảnh và cải thiện tính chính xác của phân loại ảnh.

## 2.5 YOLO

### 2.5.1. Giới thiệu mạng YOLO

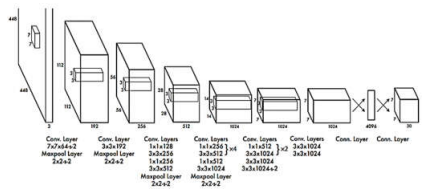
YOLO (You Only Look Once) là một mô hình deep learning cho bài toán nhận diện đối tượng trên ảnh và video. YOLO là một trong những mô hình đầu tiên sử dụng phương pháp end-to-end để giải quyết bài toán nhận diện đối tượng, nghĩa là đầu vào và đầu ra của mô hình là ảnh và danh sách các đối tượng được nhận diện trên ảnh đó.

Mô hình YOLO được phát triển bởi nhóm nghiên cứu tại Đại học Washington vào năm 2016 và đã trở thành một trong những mô hình phổ biến nhất trong lĩnh vực nhận diện đối tượng. Các phiên bản YOLO tiếp theo được phát triển với nhiều cải tiến và sự kết hợp với các kỹ thuật deep learning khác, bao gồm YOLOv2, YOLOv3 và YOLOv4.

Một trong những ưu điểm của YOLO là tốc độ xử lý nhanh, do việc chỉ thực hiện một lần forward propagation cho một ảnh và các bounding box được dự đoán trực tiếp trên toàn bộ ảnh đó. Điều này giúp cho YOLO được sử dụng rộng rãi trong các ứng dụng thời gian thực như giám sát an ninh, xe tự lái, robot, v.v.

### 2.5.2 Kiến trúc mạng YOLO

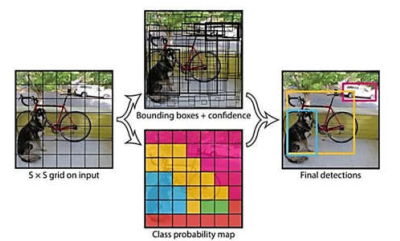
YOLO có kiến trúc bao gồm phần trích xuất đặc trưng (Feature Extractor) và phát hiện đối tượng (Extra Layers). Phần trích xuất đặc trưng là các lớp convolution sẽ cho đầu ra là các bản đồ đặc trưng (Feature map). Phần phát hiện vật thể bao gồm các lớp kết nối đầy đủ (Fully connected layers) dùng để phát hiện, dự đoán nhãn và tọa độ Bounding Box của đối tượng trên các bản đồ đặc trưng (Feature map) của phần Feature Extractor. Hình ảnh cần nhận dạng sẽ được đi qua khâu trích xuất đặc trưng để có đầu ra ra là bản đồ đặc trưng (Feature map) với các kích thước khác nhau. Các bản đồ đặc trưng sau đó được đưa qua khâu phát hiện đối tượng (Extra Layers) để dự đoán ra các thông tin về tên đối tượng, vị trí và kích thước của Bounding Box bao quanh đối tượng.



Hình 6: Các lớp của YOLO

(nguồn: https://jst-haui.vn/media/30/uffile-upload-no-title30930.pdf?fbclid=IwAR1BYUxs7TqzZHSJFSulsSai0wKsx-e8fuZAcL4pua5Z3pjExqmwc7THDwU)

Dữ liệu đầu vào là hình ảnh hoặc một frame được trích xuất từ Video. Ảnh đầu vào sau đó được chia thành mạng lưới có S x S ô (hay còn gọi là grid size). Thông thường grid size sẽ là 3 x 3, 7 x 7, 9 x 9, …



Hình 7: Cách mô hình YOLO xử lý ảnh

(Nguồn:https://jst-haui.vn/media/30/uffile-upload-no-title30930.pdf?fbclid=IwAR1BYUxs7TqzZHSJFSulsSai0wKsx-e8fuZAcL4pua5Z3pjExqmwc7THDwU)

### 2.5.3 Nguyên lý hoạt động của mạng YOLO

YOLO sẽ phân chia hình ảnh thành các ô vuông dạng lưới (grid) và xác định xem trong mỗi ô vuông liệu có tâm của vật thể cần xác định hay không. Nếu có, mô hình YOLO sẽ khoanh vùng đối tượng bằng hộp mốc (Anchor Box), sau khi có kết quả lọc chính xác thì sẽ xuất ra bounding box. Thuật toán sử dụng một mạng nơ ron đơn với đặc trưng có được từ các feature map của các lớp tích chập để dự đoán bounding box ở mỗi ô và xác suất loại đối tượng nào đang được chứa bên trong. Sau cùng, ta sẽ có rất nhiều bounding box được thuật toán đưa ra với kích thước khác nhau. Sử dụng thuật toán Non-Maxima Suppresstion (NMS) ta có thể loại được các hầu hết các bounding box là miền bao của cùng một đối tượng, có tỉ lệ thấp và giữ lại các bounding box có tỉ lệ khớp cao.

Đầu ra của YOLO khi dự đoán 1 Bounding Box là một vector gồm các thành phần gồm: xác xuất dự báo có vật thể (P0) được định nghĩa Pr(Object)∗IOU(pred,truth), tọa độ tâm kích thước chiều rộng, cao của Bouding Box (x, y, w, h), các vector phân phối xác xuất dự báo của các classes. Kích thước vector đầu ra được tính bằng (5 + số classes) ví dụ khi ta huấn luyện nhận dạng 50 đối tượng(classes) thì kích thước vector đầu ra dự đoán cho mỗ Bounding Box sẽ có kích thước là 55. Như vậy đầu ra của cả mô hình là một ma trận 3 chiều có kích thước S×S×B×(5+C) trong đó SxS là kích thước của các Feature map ở phần trích xuất đặc trưng, B là số lượng Bounding Box tương ứng là số Anchor Box và tại mỗi ô(cell) áp dụng 3 Anchor Box như vậy B = 3, C là số classes.

Mô hình YOLO hiện tại có 5 phiên bản YOLOv1 [1], YOLOv2 [2], YOLOv3 [3] do Joseph Redmon phát triển từ 2016, YOLOv4 [4] do Alexey Bochkovskiy phát triển năm 2020, YOLOv5 do Glenn Jocher phát hành trong cùng năm 2020 [5],YOLO v6 được đề xuất vào năm 2022 bởi Li, YOLOv7 và YOLOv8 vừa được phát hành cách đây không lâu. Trong đó, YOLOv8 được phát hành vào tháng 3/2023 là phiên bản YOLO mới nhất của Ultralytics. Là một mô hình tiên tiến, hiện đại (SOTA), YOLOv8 được xây dựng dựa trên sự thành công của các phiên bản trước, giới thiệu các tính năng và cải tiến mới để nâng cao hiệu suất, tính linh hoạt và hiệu quả. YOLOv8 hỗ trợ đầy đủ các tác vụ AI về thị giác, bao gồm [phát hiện](https://docs.ultralytics.com/tasks/detect/), [phân đoạn](https://docs.ultralytics.com/tasks/segment/), [ước tính tư thế](https://docs.ultralytics.com/tasks/pose/), [theo dõi](https://docs.ultralytics.com/modes/track/) và [phân loại](https://docs.ultralytics.com/tasks/classify/) . Tính linh hoạt này cho phép người dùng tận dụng các khả năng của YOLOv8 trên nhiều ứng dụng và miền khác nhau.

## 2.6. YOLOv8

### 2.6.1. Giới thiệu mạng YOLOv8

YOLOv8 là mô hình YOLO tiên tiến nhất có thể được sử dụng cho các tác vụ phát hiện đối tượng, phân loại hình ảnh và phân đoạn phiên bản. YOLOv8 được phát triển bởi [Ultralytics](https://ultralytics.com/?ref=blog.roboflow.com) , người cũng đã tạo ra mô hình YOLOv5 có sức ảnh hưởng và định hình ngành. YOLOv8 bao gồm nhiều thay đổi và cải tiến về kiến ​​trúc và trải nghiệm của nhà phát triển so với YOLOv5.

Loạt mô hình YOLO [(You Only Look Once)](https://blog.roboflow.com/guide-to-yolo-models/) đã trở nên nổi tiếng trong thế giới thị giác máy tính. Sự nổi tiếng của YOLO là do độ chính xác đáng kể của nó trong khi vẫn duy trì kích thước mô hình nhỏ. Các mô hình YOLO có thể được đào tạo trên một GPU duy nhất, giúp nhiều nhà phát triển có thể tiếp cận mô hình này. Những người thực hành máy học có thể triển khai nó với chi phí thấp trên phần cứng biên hoặc trên đám mây.

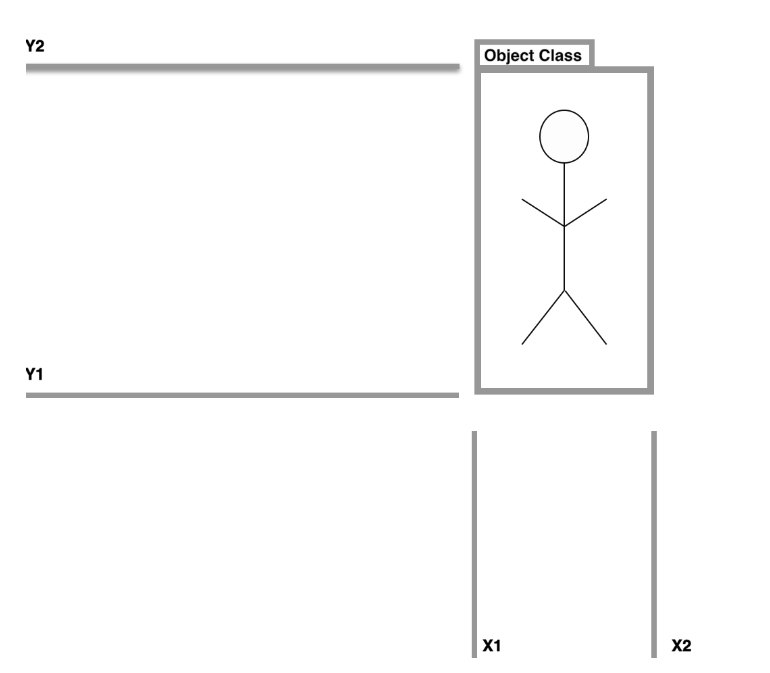
So với những phiên bản YOLO trước:

1. YOLOv8 có tỷ lệ chính xác cao được đo bằng COCO và Roboflow 100.
2. YOLOv8 đi kèm với rất nhiều tính năng thuận tiện cho nhà phát triển, từ CLI dễ sử dụng đến gói Python có cấu trúc tốt.
3. Có một cộng đồng lớn xung quanh YOLO và một cộng đồng đang phát triển xung quanh mô hình YOLOv8, nghĩa là có nhiều người trong giới thị giác máy tính có thể hỗ trợ bạn khi bạn cần hướng dẫn.

### 2.6.2. Điểm khác của YOLOv8 so với các

#### 2.6.2.1 Anchor

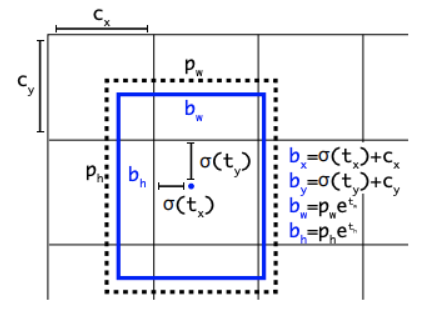
Trong việc phát hiện đối tượng, chúng ta đang tìm cách nhận dạng và xác định vị trí của các đối tượng khi chúng xuất hiện trong một hình ảnh. Phát hiện đối tượng khác biệt với việc phân loại hình ảnh vì có thể có nhiều đối tượng thuộc cùng một hoặc các lớp khác nhau xuất hiện trong hình ảnh, và phát hiện đối tượng cố gắng dự đoán chính xác tất cả các đối tượng này.



Hình 8:Phác thảo của nhiệm vụ phát hiện đối tượng

(nguồn: https://blog.roboflow.com/what-is-an-anchor-box/)

Anchor boxes là một phần trong các mô hình phát hiện đối tượng, được sử dụng để giúp dự đoán vị trí và kích thước của các đối tượng trong hình ảnh. Anchor boxes là các hộp giới hạn (bounding boxes) có kích thước và tỷ lệ được chọn trước đó, đại diện cho những đặc điểm khác nhau của các đối tượng cần được phát hiện. Các mô hình truyền thống sử dụng anchor boxes để dự đoán vị trí của các đối tượng bằng cách xác định các offset (độ lệch) từ anchor boxes.



Hình 9: Minh họa về Anchor Boxes

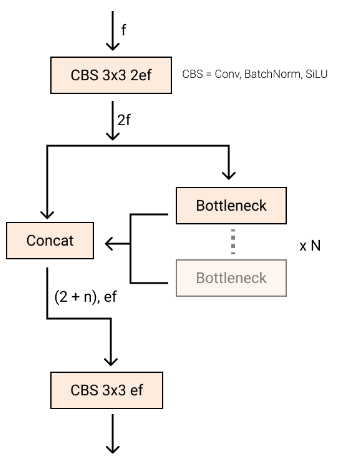
(nguồn:https://blog.roboflow.com/whats-new-in-yolov8/?fbclid=IwAR1iQwFobeodSDiJFd1jxfoAs-JZJgZmHeDJ3wPsGQwnAw\_HnjktsAvd-pw)

Anchor-free là một phương pháp khác trong phát hiện đối tượng, trong đó không sử dụng anchor boxes. Thay vào đó, mô hình anchor-free dự đoán trực tiếp vị trí và kích thước của các đối tượng bằng cách sử dụng các điểm đặc trưng (keypoints) hoặc các điểm tâm (center points). Phương pháp này giúp tránh những khó khăn và hạn chế của việc chọn và điều chỉnh anchor boxes, và có thể đem lại kết quả chính xác hơn trong việc dự đoán vị trí đối tượng.

YOLOv8 là một mô hình anchor-free. Điều này có nghĩa là nó dự đoán trực tiếp tâm của đối tượng thay vì độ lệch từ một hộp neo đã biết trước.

#### 2.6.2.2. Kết cấu mới

Convolution đầu tiên của nhánh gốc (stem) được thay thế bằng một convolution kích thước 3x3. Khối xây dựng chính cũng đã thay đổi và C2f được thay thế bằng C3. Mô-đun được tóm tắt trong hình dưới đây, trong đó "f" là số tính năng (features), "e" là tỷ lệ mở rộng (expansion rate) và CBS là một khối gồm Conv, BatchNorm và SiLU.



Hình 10: C2F Module YOLOv8 mới

(Nguồn:https://blog.roboflow.com/whats-new-in-yolov8/?fbclid=IwAR1iQwFobeodSDiJFd1jxfoAs-JZJgZmHeDJ3wPsGQwnAw\_HnjktsAvd-pw)

Trong C2f, tất cả đầu ra từ Bottleneck (tên đặc biệt cho hai convolution kích thước 3x3 với kết nối dư) được nối lại. Trong khi đó, trong C3 chỉ có đầu ra của Bottleneck cuối cùng được sử dụng.

Bottleneck giống như trong YOLOv5, nhưng kích thước kernel của convolution đầu tiên đã được thay đổi từ 1x1 thành 3x3. Từ thông tin này, chúng ta có thể thấy rằng YOLOv8 đang bắt đầu quay trở lại khối ResNet được định nghĩa từ năm 2015.

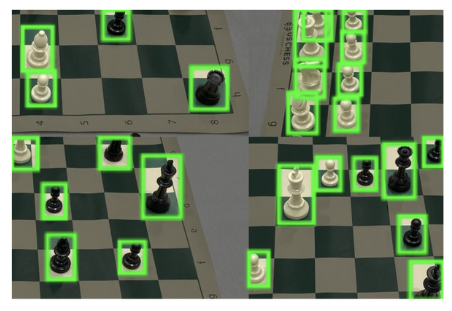
Trong phần neck, các tính năng (features) được nối trực tiếp mà không yêu cầu các kênh có cùng kích thước. Điều này giảm số lượng tham số và tổng kích thước của các tensor.

#### 2.6.2.3. Đóng mở rộng khảm

Nghiên cứu deep learning thường tập trung vào kiến trúc mô hình, nhưng quy trình huấn luyện trong YOLOv5 và YOLOv8 là một phần quan trọng của sự thành công của chúng.

YOLOv8 gia tăng ảnh trong quá trình huấn luyện trực tuyến. Ở mỗi epoch, mô hình nhìn thấy một biến thể hơi khác nhau của những bức ảnh mà nó đã được cung cấp.

Một trong những kỹ thuật gia tăng đó được gọi là gia tăng mosaic. Điều này liên quan đến việc ghép nối bốn hình ảnh với nhau, buộc mô hình phải học về các đối tượng ở các vị trí mới, trong trường hợp bị che khuất một phần và với các điểm ảnh xung quanh khác nhau.



Hình 11: Minh họa về đóng mở rộng khảm

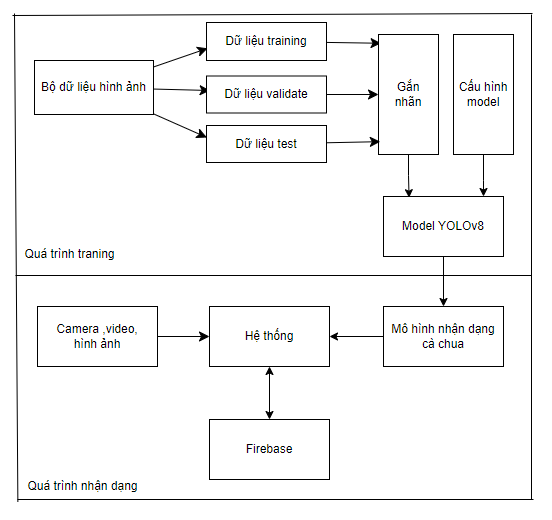
(nguồn:https://blog.roboflow.com/whats-new-in-yolov8/?fbclid=IwAR1iQwFobeodSDiJFd1jxfoAs-JZJgZmHeDJ3wPsGQwnAw\_HnjktsAvd-pw)

Tuy nhiên, đã có những chứng minh kinh nghiệm cho thấy kỹ thuật gia tăng này làm suy giảm hiệu suất nếu được thực hiện trong toàn bộ quy trình huấn luyện. Việc tắt nó trong mười epoch huấn luyện cuối cùng là có lợi.

Loại thay đổi này là ví dụ cho sự chú ý cẩn thận đã được đưa ra cho mô hình YOLO qua thời gian trong kho chứa YOLOv5 và trong nghiên cứu YOLOv8.

# CHƯƠNG 3: XÂY DỰNG MÔ HÌNH DỰA VÀO AI VÀ IoT

## 3.1. Mô hình hệ thống

****

Hình 12: Minh họa mô hình

YOLOv8 yêu cầu một bộ dữ liệu chia thành 3 tập: tập huấn luyện (train), tập xác thực (validation) và tập kiểm tra (test). Việc chia dữ liệu thành các tập này giúp đánh giá khả năng tổng quát hóa của mô hình và tránh việc overfitting (mô hình quá khớp) trên tập huấn luyện. Thông thường, tập huấn luyện được sử dụng để huấn luyện mô hình, tập xác thực được sử dụng để đánh giá và cải thiện mô hình trong quá trình huấn luyện và tập kiểm tra được sử dụng để đánh giá khả năng tổng quát hóa của mô hình sau khi đã hoàn thành quá trình huấn luyện. Các hình ảnh trong tập dữ liệu đã được gán nhãn để xác định vị trí của các quả cà chua. Sau đó, mô hình đã được cấu hình với các thông số và được huấn luyện bằng cách sử dụng tập huấn luyện và tập xác minh. Khi huấn luyện mô hình thành công, bộ trọng số của nó được lưu trữ dưới định dạng "best.pt". Mô hình này được sử dụng để thực thi chương trình nhận dạng quả cà chua trên các hình ảnh, video và từ camera. Kết quả của quá trình nhận dạng phần mềm sẽ được hiển thị trực tiếp với các phân loại "red", "green", "halfRed" và vị trí của quả cà chua được hiển thị trên màn hình. Ngoài ra, phần mềm cũng sẽ lưu lại hình ảnh hoặc video của quá trình nhận dạng để sử dụng trong tương lai.

## 3.2. Chuẩn bị bộ dữ liệu

### 3.2.1. Bộ dữ liệu 1

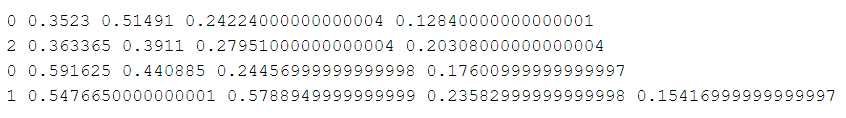
Bộ dữ liệu được sử dụng để huấn luyện được lấy từ … . Bộ dữ liệu ban đầu có 804 hình ảnh bao gồm 6 class là 0, 1, 2, 3, 4, 5 tương ứng với 2 giống cà chua

(cà chua giống thường và cà chua bi) mỗi giống gồm 3 màu: red, green, halfred. Sau khi chọn lọc, nhóm giữ lại giống cà chua thường và 3 màu: red, green, hafred. Do đó, bộ dữ liệu ban đầu còn 305 ảnh và có kích thước 640x640.Để có được tập dữ liệu nhiều hình hơn nhóm đã xử lý, sau đó bộ dữ liệu được nhân lên thành hơn 2100 tấm ảnh. Hình ảnh trong bộ dữ liệu có nhiều quả cà chua và màu sắc quả cà chua khác nhau. Trong 2100 ảnh gồm 2100 ảnh chứa quả cà chua. Mỗi ảnh có thể xuất hiện một hoặc nhiều trái cà chua khác nhau với màu sắc khác nhau, tất cả các quả cà chua đều phải được gán nhãn. Tổng số có … quả cà chua được gán nhãn. Bộ dữ liệu được phân chia 77% cho tập huấn luyện và 19% cho tập xác minh, còn lại 4% cho tập test.

Cấu trúc dữ liệu YOLOv8 yêu cầu bao gồm: 3 folder(train, valid, test) và trong mỗi folder đều có tập hình ảnh và labels tương ứng.

Dữ liệu đầu vào ảnh của YOLOv8 theo format darknet với mỗi 1 file .txt sẽ cho 1 ảnh có chứa đối tượng là label, còn với những ảnh không có đối tượng thì bỏ qua. File .txt có định dạng như sau :

* Mỗi hàng sẽ là một đối tượng
* Mỗi hàng sẽ có format như sau: class x\_center y\_center width height trong đó class x\_center y\_center là tọa độ x, y của tâm khung bao vật thể, width, height là chiều rộng và cao của khung bao vật thể
* Toạ độ của các box sẽ được chuẩn hóa (từ 0-1) theo format xywh
* Class sẽ có là 0, 1, 2(tùy từng ảnh)



Hình 13: Minh họa file label của bộ dữ liệu

### 3.2.2. Bộ dữ liệu 2

Bộ dữ liệu 2 được gộp từ 2 bộ dữ liệu:

1. Bộ dữ liệu đã chọn lọc gồm 305 ảnh và có kích thước 640x640 của bộ dữ liệu 1.
2. Bộ dữ liệu lấy từ …. gồm Bộ dữ liệu bao gồm 1987 hình ảnh. Các loại cà chua đã được chú thích theo định dạng YOLOv8.

Các bước tiền xử lý sau đã được áp dụng cho mỗi hình ảnh:

* Thay đổi kích thước thành 416x416 (kéo căng)

Các bước tăng cường sau đã được áp dụng để tạo ra 3 phiên bản của mỗi hình ảnh nguồn:

* Cắt ngẫu nhiên giữa 0 và 50 phần trăm của hình ảnh
* Điều chỉnh độ sáng ngẫu nhiên giữa -25 và +25 phần trăm

Từ 2 bộ dữ liệu này nhóm nghiên cứu sử dụng trang Roboflow.com để gộp 2 bộ dữ liệu này thành 1. Cuối cùng bộ dữ liệu bao gồm 6705 hình ảnh. Các loại cà chua đã được chú thích theo định dạng YOLOv8.

Các bước tiền xử lý sau đã được áp dụng cho mỗi hình ảnh:

* Tự động xoay dữ liệu pixel (với việc loại bỏ thông tin EXIF-orientation)
* Thay đổi kích thước thành 640x640 (kéo căng)

Các bước tăng cường sau đã được áp dụng để tạo ra 3 phiên bản của mỗi hình ảnh nguồn:

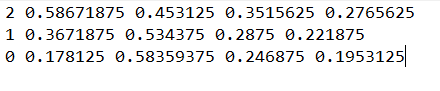
* Xoay ngang với xác suất 50%
* Xoay dọc với xác suất 50%
* Xoay ngẫu nhiên giữa -15 và +15 độ
* Điều chỉnh độ sáng ngẫu nhiên giữa -25 và +25 phần trăm

Tương tự như bộ dữ liệu 1:

Cấu trúc dữ liệu YOLOv8 yêu cầu bao gồm: 3 folder(train, valid, test) và trong mỗi folder đều có tập hình ảnh và labels tương ứng.

Dữ liệu đầu vào ảnh của YOLOv8 theo format darknet với mỗi 1 file .txt sẽ cho 1 ảnh có chứa đối tượng là label, còn với những ảnh không có đối tượng thì bỏ qua. File .txt có định dạng như sau :

* Mỗi hàng sẽ là một đối tượng
* Mỗi hàng sẽ có format như sau: class x\_center y\_center width height trong đó class x\_center y\_center là tọa độ x, y của tâm khung bao vật thể, width, height là chiều rộng và cao của khung bao vật thể
* Toạ độ của các box sẽ được chuẩn hóa (từ 0-1) theo format xywh
* Class sẽ có là 0, 1, 2(tùy từng ảnh)



Hình 14: Minh họa file label của dữ liệu

## 3.3. Huấn luyện mô hình

Mô hình YOLOv8 được tạo bằng cách tải các trọng số được huấn luyện trước đó từ YOLOv8 Darknet. Sau đó, các tầng của mô hình được khởi tạo và kết nối với nhau. Mô hình YOLOv8 được xây dựng trên cơ sở mạng nơ-ron tích chập (CNN) và sử dụng phương pháp áp dụng thang tính để dự đoán các hộp giới hạn (bounding boxes) cho các đối tượng trên ảnh đầu vào. Quá trình huấn luyện mô hình YOLOv8 là quá trình tối ưu hóa các trọng số của mô hình để tối đa hoá độ chính xác của mô hình trên tập dữ liệu huấn luyện. Việc tối ưu hóa này được thực hiện bằng các thuật toán tối ưu như stochastic gradient descent (SGD), Adam, hoặc RMSprop. Trong quá trình huấn luyện, mô hình YOLOv8 sẽ được đưa vào các batch ảnh để tính toán độ lỗi (loss) và cập nhật trọng số. Mỗi epoch (vòng lặp huấn luyện), mô hình sẽ được đánh giá trên tập dữ liệu kiểm tra để đánh giá hiệu suất của mô hình.

Quá trình này sử dụng một lượng lớn tài nguyên của máy tính, vì vậy nhóm nguyên cứu sử dụng Google Colab để đảm bảo chất lượng và hiệu quả khi huấn luyện mô hình.

### 3.3.1. Huấn luyện mô hình với bộ dữ liệu 1

|  |  |  |
| --- | --- | --- |
| Tham số | Giá trị tham số | Ý nghĩa |
| detect |  | lệnh để tiến hành nhận diện đối tượng sử dụng mô hình YOLOv8 |
| train |  | tập dữ liệu huấn luyện được sử dụng để huấn luyện mô hình |
| data | $data | đường dẫn đến tập dữ liệu huấn luyện được định nghĩa trước đó. |
| model | yolov8n.pt | tên của mô hình YOLOv8 được sử dụng để nhận diện đối tượng trên tập dữ liệu huấn luyện. Đường dẫn đến mô hình này cũng được định nghĩa trước đó. |
| epochs | 100 | số lượng epoch (vòng lặp huấn luyện) được sử dụng để huấn luyện mô hình. |
| imgsz | 640 | kích thước của ảnh đầu vào được sử dụng để huấn luyện và nhận diện đối tượng. Kích thước này được đặt là 640x640 pixel. |

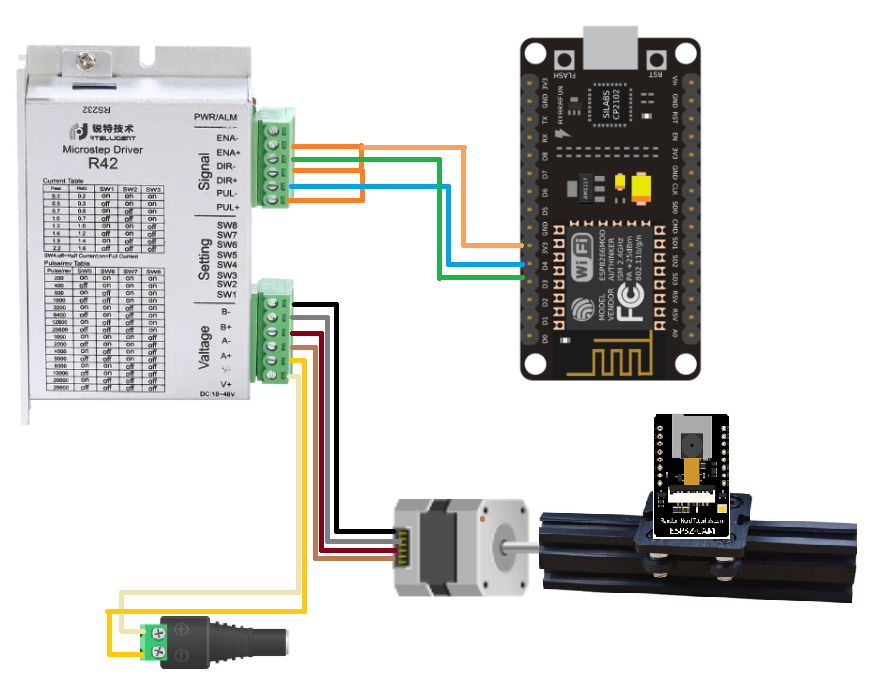
Bảng 1: Tham số sử dụng khi huấn luận mô hình với bộ dữ liệu 1

### 3.3.2 Huấn luyện mô hình với bộ dữ liệu 2

|  |  |  |
| --- | --- | --- |
| Tham số | Giá trị tham số | Ý nghĩa |
| detect |  | lệnh để tiến hành nhận diện đối tượng sử dụng mô hình YOLOv8 |
| train |  | tập dữ liệu huấn luyện được sử dụng để huấn luyện mô hình |
| data | $data | đường dẫn đến tập dữ liệu huấn luyện được định nghĩa trước đó. |
| model | yolov8n.pt | tên của mô hình YOLOv8 được sử dụng để nhận diện đối tượng trên tập dữ liệu huấn luyện. Đường dẫn đến mô hình này cũng được định nghĩa trước đó. |
| epochs | 100 | số lượng epoch (vòng lặp huấn luyện) được sử dụng để huấn luyện mô hình. |
| imgsz | 640 | kích thước của ảnh đầu vào được sử dụng để huấn luyện và nhận diện đối tượng. Kích thước này được đặt là 640x640 pixel. |
| lr0 | 0.009 | Tốc độ học ban đầu của mô hình là 0,009. |
| momentum | 0.9 | tham số momentum cho quá trình tối ưu hóa gradient descent được thiết lập là 0,9. |

Bảng 2: Tham số sử dụng khi huấn luyện mô hình với bộ dữ liệu 2

## 3.5. Mô hình kết nối từ Camera đến hệ thống

****

Hình 15: Phác thảo mô hình phần cứng hệ thống sử dụng

|  |  |  |
| --- | --- | --- |
| Tên thiết bị | Hình ảnh | Chức năng |
| ESP8266 |  | ESP8266 là một module WiFi có khả năng kết nối mạng và phát triển ứng dụng IoT. Nó được sản xuất bởi công ty Espressif Systems và được phát hành vào năm 2014. Module ESP8266 kết hợp một vi xử lý chính mạnh mẽ và tích hợp sẵn Wi-Fi, cho phép nó trở thành một bo mạch vi điều khiển thông minh và kết nối với internet. |
| Microstep Driver R42 |  | Microstep Driver (hay còn gọi là bộ điều khiển bước nhỏ) là một thành phần phần cứng được sử dụng để điều khiển động cơ bước (stepper motor) trong các ứng dụng tự động hóa và robot. |
| Stepper motor |  | Stepper motor (còn được gọi là motor bước) là một loại motor điện được thiết kế để chuyển đổi tín hiệu điện thành chuyển động xoay rõ ràng và chính xác. Stepper motor được điều khiển bằng cách kích hoạt các cuộn dây điện (windings) trong motor theo một thứ tự cụ thể, tạo ra các bước xoay nhất định. |
| ESP32-cam |  | ESP32-CAM là một module phát triển dựa trên vi điều khiển ESP32 và tích hợp camera. Nó được sản xuất bởi công ty Espressif Systems, cùng nhà sản xuất ESP8266.  Module ESP32-CAM cung cấp khả năng kết nối Wi-Fi và Bluetooth BLE, và tích hợp một camera có thể chụp ảnh và quay video. Nó có khả năng xử lý ảnh và gửi dữ liệu hình ảnh thông qua giao thức mạng, làm cho nó trở thành một giải pháp phát triển dễ dàng cho các ứng dụng IoT liên quan đến hình ảnh và video. |
|  |  | Con Lăn Vslot OpenBuilds Mini (OpenBuilds Mini V-Slot Wheel Mount Plate) là một bộ phận trong hệ thống V-Slot của OpenBuilds. V-Slot là một loại hệ thống kết cấu hợp kim nhôm được sử dụng trong các ứng dụng xây dựng máy CNC, máy in 3D và các dự án cơ khí |
| Chuôi cắm Adapter |  | Chuôi cắm cấp nguồn cho microstep driver R42 |

Bảng 3: Các thiết bị hệ thống sử dụng

* Kết nối ESP8266 với mô-đun điều khiển microstep R42 thông qua giao thức UART. ESP8266 có các chân GPIO có thể được cấu hình để hoạt động như các chân giao tiếp UART (TX, RX) để gửi và nhận dữ liệu từ R42.
* Microstep driver R42 điều khiển stepper motor: R42 cho phép bạn điều khiển stepper motor bằng cách tạo ra các xung và tín hiệu điều khiển thích hợp.
* Nhiệm vụ cụ thể của stepper motor bao gồm: Chuyển đổi tín hiệu điện thành chuyển động: Stepper motor nhận tín hiệu điện từ bộ điều khiển và chuyển đổi nó thành chuyển động xoay. Điều này được thực hiện bằng cách kích hoạt các cuộn dây điện trong motor theo một thứ tự cụ thể, tạo ra các bước xoay nhất định.

## 3.6. Phát triển chức năng ước lượng khối lượng trái cà chua

Anchor-free là một phương pháp nhận dạng đối tượng trong việc xác định vị trí và phân loại đối tượng trong hình ảnh mà không sử dụng anchor boxes. Thay vì dựa vào anchor boxes như trong phương pháp anchor-based, anchor-free xác định vị trí và kích thước của đối tượng trực tiếp từ các thông tin trong không gian đầu ra.

Trong phương pháp anchor-free, thông thường có các giá trị sau:

1. Điểm đặc trưng (Key Points): Điểm đặc trưng là các điểm quan trọng trong không gian đầu ra được sử dụng để định vị vị trí của đối tượng. Thông qua việc tính toán từ các điểm đặc trưng, ta có thể xác định vị trí và kích thước của đối tượng.
2. Độ tin cậy (Confidence): Độ tin cậy dùng để ước lượng độ chắc chắn của việc nhận dạng đối tượng tại các điểm đặc trưng. Nó thường được tính dựa trên xác suất rằng một điểm đặc trưng chứa một đối tượng và độ chính xác của dự đoán đối tượng.
3. Bản đồ đặc trưng (Feature Map): Bản đồ đặc trưng là một ma trận được tạo ra từ đầu ra của mạng neural network. Nó chứa các giá trị biểu thị cho thông tin đặc trưng của hình ảnh và được sử dụng để phát hiện và phân loại đối tượng.
4. Các giá trị dự đoán (Predicted Values): Các giá trị dự đoán trong phương pháp anchor-free bao gồm vị trí (x, y) của đối tượng, kích thước (width, height) và độ tin cậy của dự đoán. Các giá trị này được sử dụng để xác định vị trí và đặc điểm của đối tượng trong hình ảnh.

Trong đó, các giá trị dự đoán bao gồm các thông tin sau:

1. Vị trí (x, y): Đây là các giá trị tọa độ (x, y) của điểm gốc (gốc của hộp giới hạn) hoặc tâm của đối tượng trong hình ảnh. Các giá trị này xác định vị trí tương đối của đối tượng trong không gian đầu ra, thường được chuẩn hóa theo kích thước của ô lưới hoặc bản đồ đặc trưng.
2. Kích thước (width, height): Đây là các giá trị chiều rộng (width) và chiều cao (height) của hộp giới hạn hoặc đối tượng. Các giá trị này cũng được chuẩn hóa hoặc dự đoán theo tỷ lệ so với ô lưới hoặc bản đồ đặc trưng.
3. Độ tin cậy (Confidence): Đây là một giá trị đánh giá độ chắc chắn của dự đoán. Nó thể hiện xác suất rằng hộp giới hạn chứa một đối tượng và độ chính xác của dự đoán. Độ tin cậy thường được tính bằng cách sử dụng hàm kích hoạt softmax hoặc hàm sigmoid để đưa giá trị về khoảng từ 0 đến

Các giá trị này được sử dụng để xác định vị trí, kích thước và độ tin cậy của đối tượng trong hình ảnh. Chúng cung cấp thông tin quan trọng để nhận dạng và phân loại đối tượng trong phương pháp anchor-free.

Sau khi xác định được giá trị dự đoán của trái cà chua, nhóm tác giả tính bán kính trung bình của trái cà chua bằng công thức

Trong đó: là bán kính trung bình của trái cà chua

w là width của trái cà chua

h là height của trái cà chua

Sử dụng công thức tính thể tích hình cầu V = (4/3) \* π \* r^3

Trong đó:

* V là thể tích của hình cầu
* r là bán kính của hình cầu
* π là hằng số Pi, có giá trị khoảng 3.14

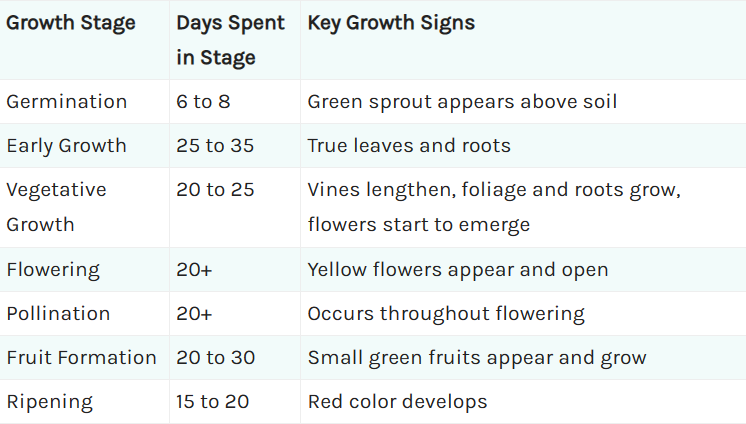
Để tính khối lượng của một vật từ thể tích và khối lượng riêng của nó: m = V \* ρ

Trong đó:

* m là khối lượng của vật (đơn vị khối lượng)
* V là thể tích của vật (đơn vị thể tích)
* ρ là khối lượng riêng của vật (đơn vị khối lượng / đơn vị thể tích)

## 3.7. Phát triển chức năng ước lượng ngày thu hoạch

Chức năng ước lượng ngày thu hoạch là một tính năng khá hữu ích cho các nông dân trồng cây ăn quả, giúp cho các nông dân có thể lên kế hoạch chăm sóc và thu hoạch một cách chính xác và hiệu quả, tăng năng suất và chất lượng của sản phẩm. Vì vậy, việc ước lượng ngày thu hoạch dựa vào màu sắc và khối lượng đã tính toán thông qua kết quả dự đoán hình ảnh là một chức năng vô cùng cần thiết trong hệ thống.



Hình 16: Các giai đoạn phát triên của cà chua

(Nguồn: https://www.thespruce.com/tomato-plant-growth-timeline-7255375)

Quá trình sinh trưởng của cà chua diễn ra một cách tuần tự theo 7 giai đoạn phát triển: nảy mầm, tăng trưởng sớm, tăng trưởng sinh dưỡng, giai đoạn ra hoa, thụ phấn, hình thành quả, giai đoạn chín.[20]

Vì hệ thống nhận biết cà chua qua ảnh 2D và tính toán, do đó để phát triển chức năng này nhóm tác giả sẽ chỉ tập trung vào hai giai đoạn chính là hình thành quả và giai đoạn chín. Theo hình 16, giai đoạn hình thành quả sẽ diễn ra trong khoảng thời gian trung bình là 20 - 30 ngày, giai đoạn chín sẽ diễn ra trong khoảng thời gian trung bình là 15 - 20 ngày. Từ đó, nhóm tác giả chọn thời gian trung bình cho quá trình hình thành và phát triển quả là 25 ngày (1) và thời gian trung bình cho quá trình chín là 18 ngày (2).

Khối lượng trung bình của quả: 40-70 gram , nhóm tác giả sử dụng khối lượng trung bình là 50 gram (3).

Khi cây có quả chín từ ½ đến ⅔ thì tiến hành thu hoạch ngay, để chín quá dễ bị rụng[21] => nhóm tác giả sử dụng số liệu là ⅖ (2 phần màu đỏ, cam, vàng và 3 phần màu xanh) là mức độ có thể thu hoạch được (4).

Từ (1) và (3) => Mỗi ngày cà chua phát triển trung bình (5).

=

Từ (2) và (4) => Thời gian từ lúc đậu quả đến khi thu hoạch (6).

=

Từ (2) và (6) => Độ chín mỗi ngày của quả.

=

Nếu quả được nhận diện là Green và chưa đạt được khối lượng trung bình:

Thời gian ước tính = + 1 + .

Trong đó: là khối lượng trung bình của quả cà chua.

là khối lượng hiện tại của quả đang ước tính.

là khối lượng tăng lên của quả cà chua trong 1 ngày.

là thời gian từ khi đậu quả đến khi thu hoạch được.

Nếu quả được nhận diện là Green nhưng đã đạt hoặc vượt quá khối lượng trung bình:

Thời gian ước tính = + 1.

là độ chín của quả có thể thu hoạch.

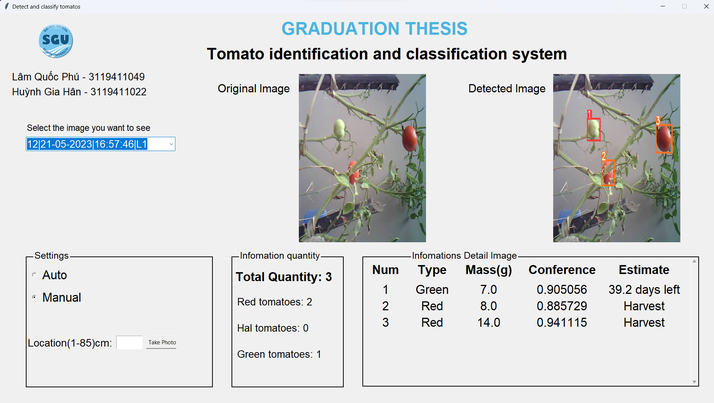
là độ chín hiện tại của quả đang ước tính (dựa trên số lượng màu đỏ, cam vàng trên số lượng màu xanh).

là độ chín tăng mỗi ngày của quả cà chua.

Nếu quả được nhận diện là Half Red và Red thì đã có thể thu hoạch.

## 3.8. Giao diện hệ thống

Giao diện được nhóm thiết kế bằng Tkinter, Tkinter là một thư viện được tích hợp sẵn trong Python, cung cấp các công cụ và lớp đối tượng để thiết kế và xây dựng giao diện người dùng đồ họa (GUI) cho các ứng dụng desktop. Tkinter dựa trên toolkit đồ họa Tk, một toolkit đa nền tảng được sử dụng rộng rãi trong việc phát triển ứng dụng GUI.

****

Hình 17: Giao diện của hệ thống

Giao diện hệ thống gồm:

* 2 hình ảnh trước xử lý và sau khi xử lý
* Danh sách những lần chụp trước(có thể xem lại)
* Ô cài đặt: chứa 2 lựa chọn và 1 nút nhấn

+Auto: chạy theo giờ đã được cài đặt

+Manual: tự động chạy sau khi nhập nơi muốn đến và nhấn nút chụp hình

* Ô thông tin số lượng của ảnh đã được xử lý: gồm 4 thông tin

+Tổng số lượng trái cà chua đã được xử lý

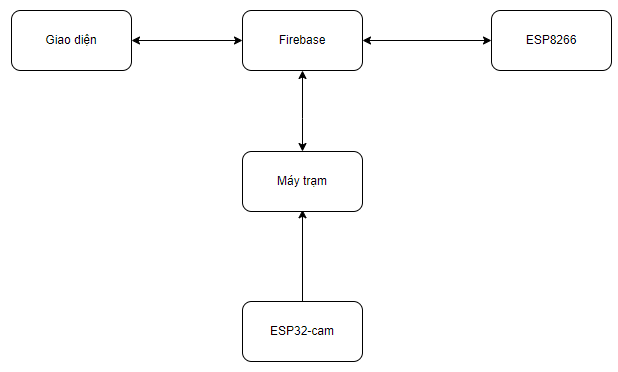
+Tổng số lượng trái cà chua màu đỏ

+Tổng số lượng trái cà chua màu cam

+Tổng số lượng trái cà chua màu xanh

* Ô thông tin chi tiết của ảnh vừa xử lý: gồm 5 cột, 4 hàng
* Cột số thứ tự
* Cột loại: hiển thị loại của các trái cà chua đã xử lý
* Cột khối lượng: hiển thị khối lượng từng trái cà chua đã xử lý
* Cột độ chính xác: hiển thị độ chính xác của từng trái cà chua được xử lý
* Cột ước tính ngày để thu hoạch: hiển thị ngày sẽ có thể thu hoạch từng trái cà chua

## 3.9. Cách chương trình hoạt động

****

Hình 18: Cách chương trình hoạt động

* Firebase giao tiếp với ESP8266

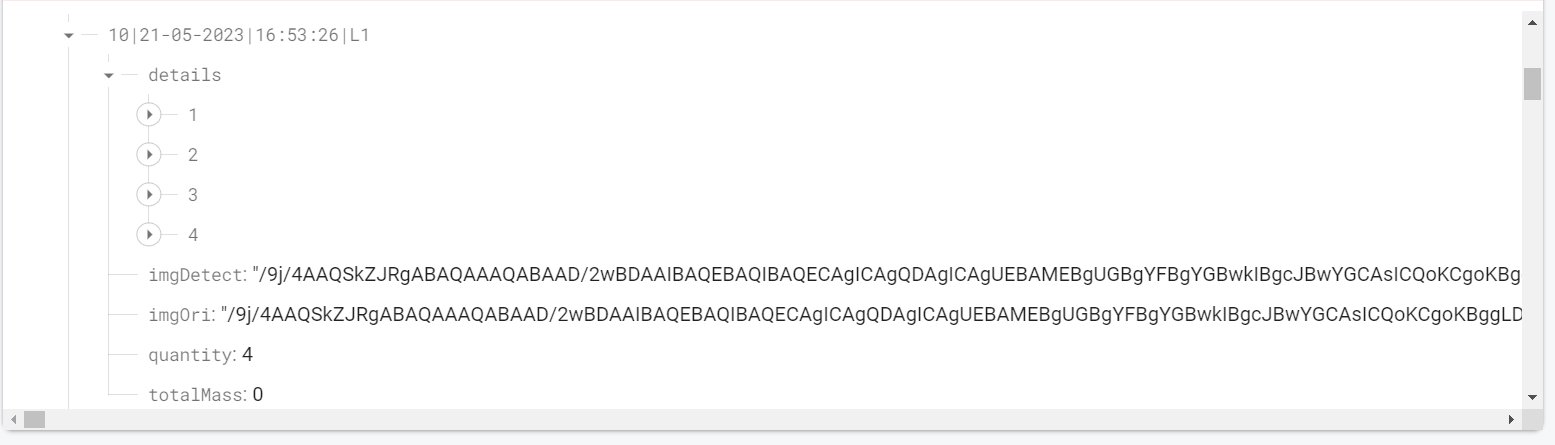


Hình 19: Các biến Firebase sử dụng khi giao tiếp với ESP8266

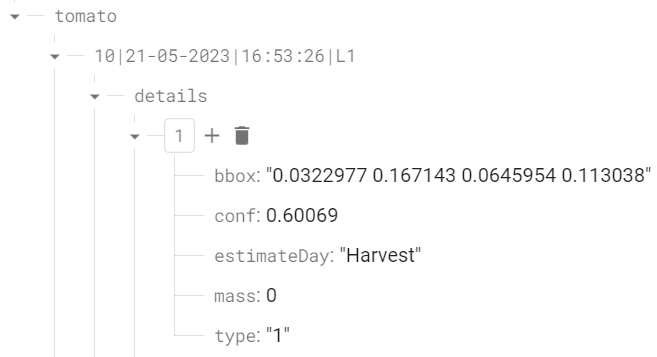
|  |  |  |
| --- | --- | --- |
| Biến | Kiểu dữ liệu | Ý nghĩa |
| addPic | string | biến xét quá trình chụp hình đã hoàn thành hay chưa |
| done | int | biến xét quá trình hoạt động đã hoàn thành hay chưa |
| location | int | vị trí stepper motor sẽ chạy tới |
| run | int | biến điều khiển stepper chạy |
| status | string | trạng thái mà mô hình sẽ hoạt động |
| time… | string | thời gian được chuẩn bị của trạng thái auto |

Bảng 4: Các biến Firebase sử dụng khi giao tiếp với ESP-8266

* Firebase giao tiếp với máy trạm và ESP32-cam: sau khi ESP32-cam chụp hình sẽ được chuyển cho máy trạm xử lý, sau khi máy trái xử lý sẽ truyền kết quả lên Firebase.



Hình 20: Các biến Firebase sử dụng khi giao tiếp với ESP32-Cam



Hình 21: Các biến Firebase sử dụng khi giao tiếp với ESP32-Cam

|  |  |  |
| --- | --- | --- |
| Biến | Tham số | Ý nghĩa |
| imgDetect | string | hình ảnh sau khi xử lý được chuyển về chuỗi base 64 |
| imgOri | string | hình ảnh trước khi xử lý được chuyển về chuỗi base 64 |
| quantity | int | tổng số lượng trái đã được xử lý |
| totalMass | int | tổng khối lượng trái đã được xử lý |
| detail | dictionary | thông tin chi tiết từng trái đã được xử lý |
| bbox | string | vị trí của từng trái sau khi được xử lý |
| conf | int | độ tin cậy |
| estimateDay | string | Số ngày đến ngày thu hoạch được ước tính |
| mass | float | Khối lượng ước tính của trái |
| type | int | Giai đoạn chín của trái |

Bảng 5: Các biến Firebase sử dụng khi giao tiếp với ESP32-Cam

* Firebase giao tiếp với giao diện

|  |  |  |
| --- | --- | --- |
| Biến | Tham số | Ý nghĩa |
| run | int | biến điều khiển stepper chạy |
| imgDetect | string | hình ảnh sau khi xử lý được chuyển về chuỗi base 64 |
| imgOri | string | hình ảnh trước khi xử lý được chuyển về chuỗi base 64 |
| quantity | int | tổng số lượng trái đã được xử lý |
| totalMass | int | tổng khối lượng trái đã được xử lý |
| detail | dictionary | thông tin chi tiết từng trái đã được xử lý |
| bbox | string | vị trí của từng trái sau khi được xử lý |
| conf | int | độ tin cậy |
| estimateDay | string | Số ngày đến ngày thu hoạch được ước tính |
| mass | float | Khối lượng ước tính của trái |
| type | int | Giai đoạn chín của trái |

Bảng 6: Các biến Firebase sử dụng khi giao tiếp với giao diện

# CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM

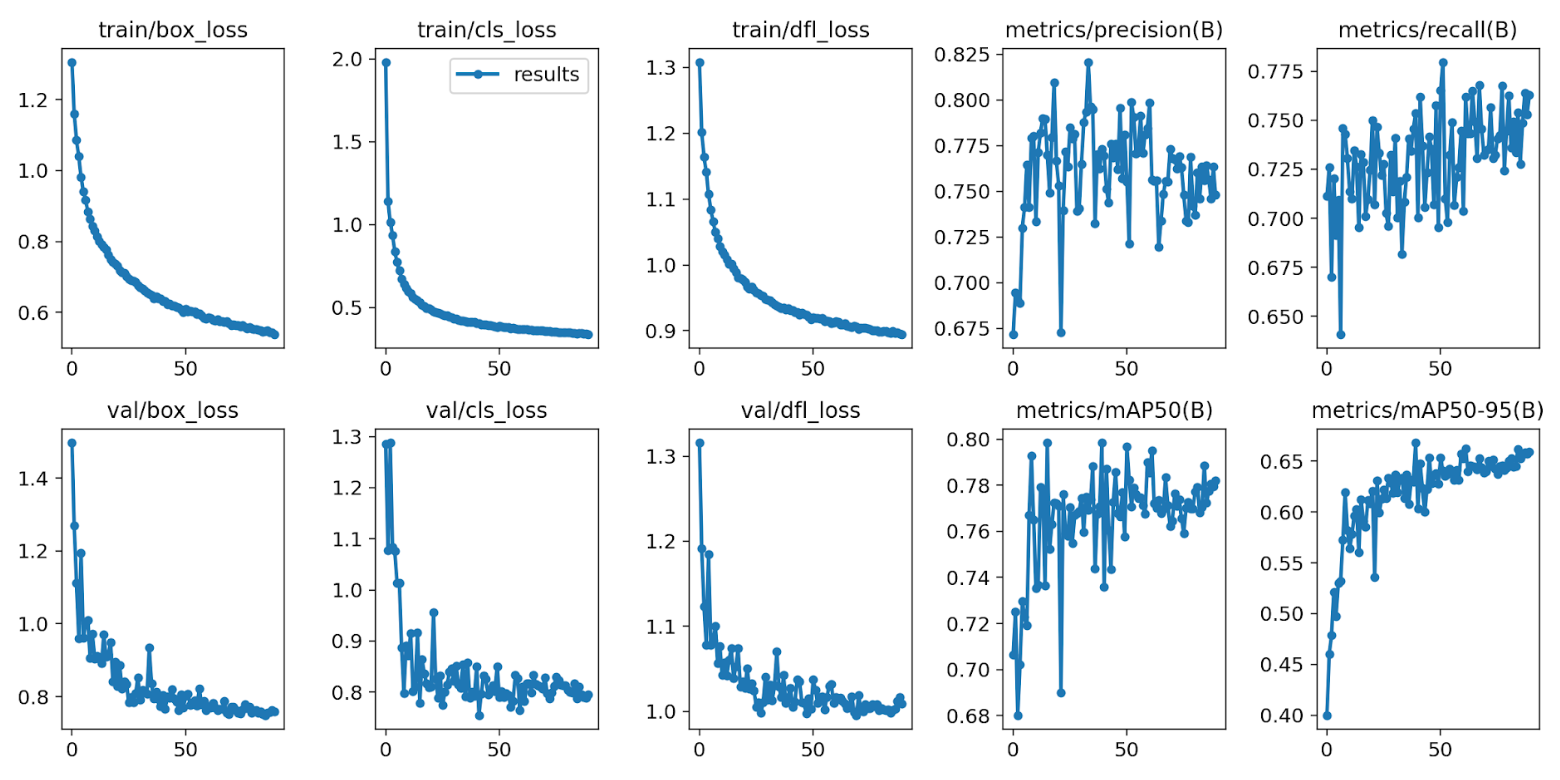
## 4.1. Kết quả

### 4.1.1. Kết quả huấn luyện bộ dữ liệu 1

Kết thúc quá trình huấn luyện có 2 bộ trọng số được mô hình lưu lại bao gồm:

* Bộ trọng số tốt nhất: best.pt
* Bộ trọng số cuối cùng của quá trình huấn luyện: last.pt

Trong đó bộ trọng số best.pt được sử dụng cho chương trình nhận dạng. Thông số chi tiết kết quá training thể hiện trong hình.



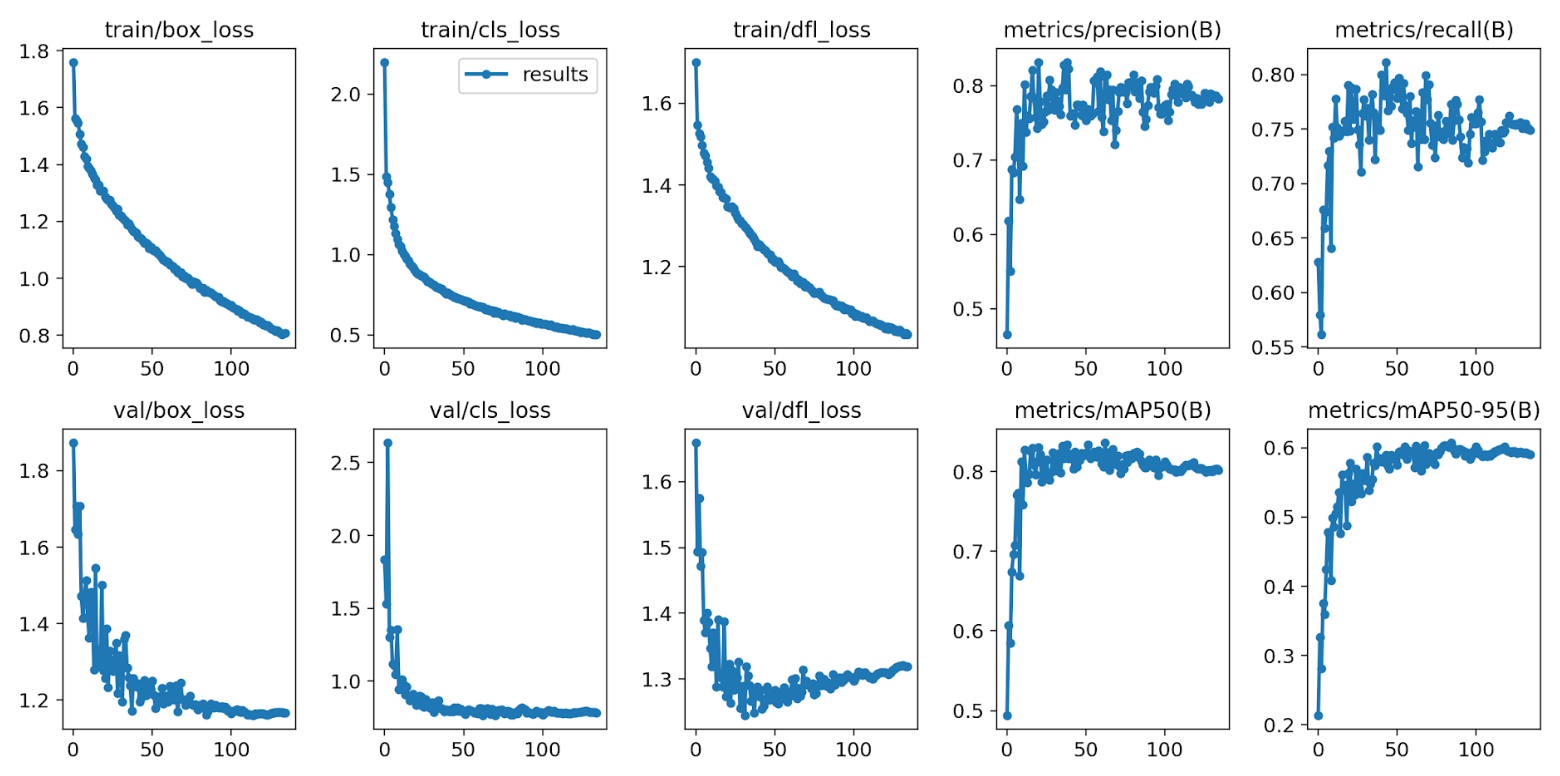
Hình 22: Kết quả huấn luyện bộ dữ liệu 1

### 4.1.2. Kết quả huấn luyện bộ dữ liệu 2

Kết thúc quá trình huấn luyện có 2 bộ trọng số được mô hình lưu lại bao gồm:

* Bộ trọng số tốt nhất: best.pt
* Bộ trọng số cuối cùng của quá trình huấn luyện: last.pt

Trong đó bộ trọng số best.pt được sử dụng cho chương trình nhận dạng. Thông số chi tiết kết quá training thể hiện trong hình.

****

Hình 23: Kết quả huấn luyện bộ dữ liệu 2

## 4.2. Thực nghiệm và đánh giá

Để đánh giá hiệu năng của mô hình, nhóm tác giả đánh giá một thông số là độ chính xác trong nhận dạng. Độ chính xác được đánh giá trong 2 điều kiện: buổi sáng và buổi tối có đèn.

Đánh giá độ chính xác trong nhận dạng nhóm tác giả sử dụng hai thông số gồm:

Precision: Biểu thị độ chính xác trong dự đoán tên và vị trí đối tượng;

Recall: Biểu thị khả năng phát hiện đối tượng trong dữ liệu đầu vào.

Trong hệ thống yêu cầu phát hiện ngọn lửa là nhiệm vụ quan trọng nhất do đó tham số Recall được quan tâm hơn Precision. Recall càng cao khả năng phát hiện lửa càng cao đồng nghĩa là khả năng bỏ sót lửa càng thấp.

Độ chính xác dự đoán đối tượng:

Precision = (1)

Khả năng phát hiện đối tượng:

Recall = (2)

Trong đó:

TP - True Positive: Thực tế có đối tượng, dự đoán có đối tượng;

FN - False Negative: Thực tế có đối tượng, dự đoán không có đối tượng;

TN - True Negative: Thực tế không có đối tượng, dự đoán không có đối tượng;

FP - False Positive: Thực tế không có đối tượng, dự đoán có đối tượng.

### 4.2.1. Thực nghiệm với điều kiện ngoài trời buổi sáng

Thực nghiệm được thực hiện chụp từ camera ngoài trời với điều kiện đầy đủ ánh sáng. Camera chụp hình và gửi cho máy trạm, trích xuất từ hình ảnh kết quả có kích thước 640x480. Hình ảnh sau khi nhận dạng như trong hình

****

Hình 24: Ảnh chụp thực nghiệm với điều kiện ngoài trời buổi sáng

Kết quả chức năng ước lượng khối lượng và ước lượng ngày thu hoạch:

|  |  |  |
| --- | --- | --- |
| Số thứ tự | Khối lượng ước tính | Số ngày đến thu hoạch ước lượng |
| Trái thứ 1 | 34 | Còn 16 ngày |
| Trái thứ 2 | 42 | Đã có thể thu hoạch |
| Trái thứ 3 | 70 | Đã có thể thu hoạch |

Bảng 7: Kết quả thực nghiệm ban ngày

### 4.2.2. Thực nghiệm với điều kiện ngoài trời buổi tối có đèn

Thực nghiệm được thực hiện chụp từ camera ngoài trời với điều kiện ngoài trời buổi tối có đèn. Camera chụp hình và gửi cho máy trạm, trích xuất từ hình ảnh kết quả có kích thước 640x480. Hình ảnh sau khi nhận dạng như trong hình

****

Hình 25: Ảnh chụp thực nghiệm với điều kiện ngoài trời buổi tối có đèn

Kết quả chức năng ước lượng khối lượng và ước lượng ngày thu hoạch

|  |  |  |
| --- | --- | --- |
| Số thứ tự | Khối lượng ước tính | Số ngày đến thu hoạch ước lượng |
| Trái thứ 1 | 39 | Đã có thể thu hoạch |
| Trái thứ 2 | 80 | Đã có thể thu hoạch |

Bảng 8: Kết quả thực nghiệm buổi tối

# 

# Kết luận và kiến nghị

## Kết luận:

Trong đề tài này, nhóm nghiên cứu đã cố gắng thiết kế và xây dựng một hệ thống sử dụng Trí tuệ nhân tạo (AI) và kết nối vạn vật (IoT) để phân loại cà chua theo màu sắc, ước lượng khối lượng và thực hiện các hoạt động chăm sóc vườn cà chua từ xa. Mặc dù còn nhiều hạn chế và thách thức phải đối mặt, đề tài này đóng góp một phần nhỏ trong việc ứng dụng công nghệ vào lĩnh vực nông nghiệp, có thể mang lại một số lợi ích cho quá trình sản xuất cà chua.

Bằng cách sử dụng hệ thống thu thập dữ liệu ảnh cà chua từ xa thông qua mô hình IoT, dữ liệu ảnh được lưu trữ trên Firebase và truyền về máy tính để phân tích. Sử dụng các kỹ thuật xử lý ảnh và AI, hệ thống có khả năng nhận dạng, phân tích và đánh giá chất lượng, mức độ phát triển của cà chua. Điều này có thể cung cấp cho nông dân và nhà sản xuất thông tin hữu ích về tình trạng và tiến độ của vườn cà chua.

Hệ thống cũng cung cấp các chức năng chăm sóc vườn cà chua từ xa thông qua IoT, bao gồm đo nhiệt độ, độ ẩm môi trường, cường độ chiếu sáng và độ ẩm không khí. Hệ thống cũng có khả năng tự động tưới nước và điều khiển mái che để làm mát theo các chế độ đã được thiết lập trước. Tuy nhiên, việc tự động hóa hoạt động này vẫn còn hạn chế và cần được cải thiện.

Mặc dù đề tài này còn nhiều giới hạn và thách thức, nhóm nghiên cứu hy vọng rằng nó có thể mang lại một số kiến thức và hiểu biết mới về ứng dụng của AI trong phân loại cà chua và cung cấp một cách tiếp cận mới trong việc nâng cao hiệu quả và hiệu suất của quá trình sản xuất cà chua.

## Kiến nghị:

Dựa trên đề tài này, có một số kiến nghị nhằm cải thiện và phát triển hệ thống phân loại cà chua và chăm sóc vườn cà chua từ xa sử dụng AI và IoT:

1. Tăng cường khả năng phân tích chất lượng cà chua: Để đảm bảo chất lượng sản phẩm, hệ thống có thể được mở rộng để phân tích các yếu tố khác như hàm lượng đường, acid, chất béo, và hàm lượng chất dinh dưỡng khác trong cà chua. Điều này giúp cung cấp thông tin chi tiết về chất lượng và giá trị dinh dưỡng của sản phẩm.
2. Phát triển giao diện người dùng thân thiện: Để hỗ trợ nông dân và nhà sản xuất, giao diện người dùng của hệ thống cần được thiết kế đơn giản, dễ sử dụng và dễ hiểu. Nên cung cấp các công cụ và chức năng dễ dùng để quản lý và theo dõi các thông tin quan trọng về cà chua và vườn trồng.
3. Tăng cường tính ổn định và độ tin cậy của hệ thống: Để đảm bảo hoạt động liên tục và tin cậy của hệ thống, cần đảm bảo mạng kết nối IoT ổn định và có khả năng chịu tải cao. Đồng thời, hệ thống cần được kiểm tra và đánh giá định kỳ để phát hiện và khắc phục các lỗi và sự cố nhanh chóng.

Ngoài ra, việc chia sẻ kết quả nghiên cứu và thành công trong triển khai hệ thống phân loại cà chua và chăm sóc vườn cà chua từ xa có thể được thực hiện thông qua bài viết khoa học, báo cáo, hoặc tham gia các hội nghị và sự kiện liên quan đến nông nghiệp và công nghệ. Điều này giúp tăng cường nhận thức và khuyến nghị cho các nông dân và nhà sản xuất khác trong việc áp dụng công nghệ vào sản xuất nông nghiệp.

Cuối cùng, việc tiếp tục nghiên cứu và phát triển các ứng dụng AI và IoT trong lĩnh vực nông nghiệp, đặc biệt là trong phân loại cà chua và chăm sóc vườn cà chua từ xa, là một quá trình liên tục. Nhóm nghiên cứu có thể tiếp tục tìm hiểu và thử nghiệm các công nghệ mới, tối ưu hóa thuật toán và mô hình, và thích nghi với các yêu cầu và điều kiện cụ thể của nông nghiệp.

Nhờ những nỗ lực này, chúng ta có thể không chỉ nâng cao hiệu suất và hiệu quả trong quá trình sản xuất cà chua, mà còn mở ra cơ hội để áp dụng công nghệ vào các lĩnh vực nông nghiệp khác, góp phần vào sự phát triển bền vững và nâng cao chất lượng cuộc sống trong cộng đồng nông dân.

# 

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Agarwal S, Fulgoni V, Saman S, Lycopene and heart health. American Journal of Lifestyle Medicine, vol. 11, no. 6, pp. 458-467, Nov/Dec 2017. |
| [2] | Di Matteo M, Esposito E. Tomatoes and Immunity. International Journal of Molecular Sciences, vol. 18, no. 9, pp. 1906, Sep 2017. |
| [3] | Stahl W, Heinrich U, Aust O, Tronnier H, Sies H. Tomatoes and Skin Health. Antioxidants, vol. 4, no. 1, pp. 135-150, Jan 2015. |
| [4] | Palozza P, Simone R, Catalano A, Mele MC. Lycopene and Cancer. Frontiers in Pharmacology, vol. 9, pp. 1-11, Feb 2018 |
| [5] | T. Miralles-Wilhelm, J. R. Garatuza-Payan, K. R. Thorp, M. R. Nayak, T. Hogue, and J. Chavez, "Optimizing irrigation scheduling for tomato production in semiarid regions using remote sensing and crop growth models," Agricultural Water Management, vol. 214, pp. 152-163, Feb. 2019. DOI: 10.1016/j.agwat.2018.10.019.  J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788 |
| [6] | F. F. de Sousa, M. L. S. Nascimento, R. B. L. Teixeira, and L. L. Marques, "Optimization of nitrogen fertilization for greenhouse tomato production using chlorophyll meter and soil analysis," Scientia Horticulturae, vol. 267, 109313, Jan. 2020. DOI: 10.1016/j.scienta.2020.109313. |
| [7] | J. B. Almeida, V. M. Chagas, T. P. Aquino, L. A. Dos Santos, M. L. P. Nogueira, and A. L. A. Sousa, "Assessment of tomato seedling root system traits for early vigour and abiotic stress tolerance," Plant Physiology and Biochemistry, vol. 156, pp. 267-274, Jul. 2020. DOI: 10.1016/j.plaphy.2020.07.004. |
| [8] | Lê, T. T. H., & Lee, S. G. (2019). Effect of a smart fertigation system on tomato yield and quality in greenhouse cultivation. IEEE Access, 7, 107949-107958. |
| [9] | Wang, J., Chen, S., Huang, Y., Zhao, L., Peng, S., & Liu, L. (2018). Application of machine learning for yield prediction and decision-making in tomato production. Computers and Electronics in Agriculture, 152, 268-276. |
| [10] | C. M. Barton, S. C. Bost, and W. W. Wallender, "Weather and low light limit small unmanned aerial vehicle aerial imaging for agricultural applications," Journal of Applied Remote Sensing, vol. 9, no. 1, p. 096016, 2015. |
| [11] | G. v. Rossum, Interviewee, Programming at Python Speed. [Interview]. 27 1 2003. |
| [12] | 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). |
| [13] | I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," vol. 1. MIT Press, 2016, ch. 10. |
| [14] | Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:1701.07875. |
| [15] | Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, 2016. SSD: Single Shot MultiBox Detector.arXiv:1512.02325v5. |
| [16] | Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2016. You Only Look Once: Unified, Real-Time Object Detection.arXiv:1506.02640v5 [cs.CV. |
| [17] | R. C. Gonzalez and R. E. Woods, Digital Image Processing, Second Edition. Pearson Education, Inc., Upper Saddle River, NJ, USA, 2002, pp. 01-02. |
| [18] | E. Todt and B. A. Krinski, "Convolutional Neural Network - CNN," VRI Group - Vision Robotic and Images, Federal University of Parana, Nov. 30, 2019. |
| [19] | IEEE. (2023). Computer Vision. [Online]. Available:<https://www.ibm.com/topics/computer-vision>. [Accessed: 16-May-2023]. |
| [20] | phần 3.7  “Tomato Plant Growth Timeline: From Seed to Harvest,” The Spruce. [Online]. Available: https://www.thespruce.com/tomato-plant-growth-timeline-7255375. [Accessed: 17-May-2023]. |
| [21] | “Hội nông dân tỉnh Ninh Bình,” Kỹ thuật trồng cây cà chua. [Online]. Available: https://bacsicayxanh.vn/cay-trong/ca-chua/ky-thuat/cach-thu-hoach/22. [Accessed: May 22, 2023]. |
|  |  |