

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP

NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên: Phạm Thành Lâm

Giảng viên hướng dẫn: TS. HỒ THỊ HƯƠNG THƠM

Hải Phòng - 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

**XÂY DỰNG MÔ HÌNH AI HỖ TRỢ PHÁT HIỆN
MỘT SỐ LOẠI TRÁI CÂY BỊ HƯ HỎNG
GIÚP LOẠI BỎ KỊP THỜI**

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

NGÀNH: Công nghệ thông tin

Sinh viên: Phạm Thành Lâm

Giáo Viên Hướng Dẫn: TS. Hồ Thị Hương Thơm

HẢI PHÒNG – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Tên đề tài: XÂY DỰNG MÔ HÌNH AI HỖ TRỢ PHÁT HIỆN MỘT SỐ LOẠI TRÁI CÂY BỊ HƯ HỎNG GIÚP LOẠI BỎ KỊP THỜI

Sinh viên làm đề tài tốt nghiệp

Sinh viên : Phạm Thành Lâm

Mã SV: 2112111015

Lớp : CT2501C

Ngành : Công nghệ thông tin

Tên đề tài : Xây dựng mô hình hỗ trợ phát hiện một số loại trái cây hư hỏng giúp loại bỏ kịp thời.

Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a) Mô tả tóm tắt đề tài

 Tìm hiểu về mô hình phát hiện đối tượng bằng YOLO11 cho bài toán phát hiện loại trái cây, thiết kế xây dựng chương trình phát hiện loại trái cây.

b) Nội dung hướng dẫn

- Tìm hiểu về bài toán phát hiện đối tượng nói chung và phát hiện đối tượng bằng YOLO11.
- Phân tích, thu thập dữ liệu hình ảnh trái cây để huấn luyện phát hiện đối tượng.
- Cài đặt mô hình phát hiện đối tượng bằng YOLO11.
- Thử nghiệm mô hình trên hình ảnh và video để phát hiện trái cây.
- Nhận xét đánh giá, kết luận và đưa ra khuyến nghị khi ứng dụng.

c) Kết quả cần đạt được

Xây dựng thành công mô hình phát hiện và phân loại trái cây bằng YOLO11: Mô hình có khả năng nhận dạng chính xác các loại trái cây khác nhau (như cam, táo, xoài, chuối, dưa hấu, v.v.) trên hình ảnh hoặc video. Hệ thống có thể khoanh vùng (bounding box) từng đối tượng và hiển thị tên loại trái cây kèm độ tin cậy (confidence score).

Phát triển ứng dụng phát hiện trái cây trên nền tảng Android: Ứng dụng được xây dựng bằng Android Studio, tích hợp mô hình YOLO11 giúp người dùng có thể nhận diện trái cây trực tiếp qua camera điện thoại hoặc từ hình ảnh/video có sẵn. Kết quả nhận dạng được hiển thị trực quan, dễ sử dụng và có khả năng hoạt động thời gian thực.

Đánh giá và phân tích hiệu quả mô hình: Thực hiện đo đạc các chỉ số đánh giá như Precision, Recall, mAP (mean Average Precision) để xác định độ chính xác và hiệu năng của mô hình trên tập dữ liệu huấn luyện và kiểm thử.

Tối ưu mô hình cho thiết bị di động: Chuyển đổi mô hình YOLO11 sang định dạng ONNX hoặc TensorFlow Lite, giúp giảm dung lượng và tăng tốc độ xử lý khi triển khai trên điện thoại Android.

Hoàn thiện báo cáo đồ án tốt nghiệp: Viết báo cáo trình bày đầy đủ quy trình thực hiện, từ khâu thu thập và gán nhãn dữ liệu, huấn luyện mô hình, đánh giá kết quả, xây dựng ứng dụng, đến triển khai thực tế, đảm bảo nội dung rõ ràng, có hình ảnh minh họa và phân tích cụ thể.

CÁC CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Hồ Thị Hương Thơm

Học hàm, học vị : Tiến sĩ

Cơ quan công tác : Trường Đại Học Hàng Hải Việt Nam

Nội dung hướng dẫn :

- Tìm hiểu về bài toán phát hiện đối tượng nói chung và phát hiện phát hiện đối tượng bằng YOLO11.

- Phân tích, thu thập dữ liệu hình ảnh trái cây để huấn luyện phát hiện đối tượng.

- Cài đặt mô hình phát hiện đối tượng bằng YOLO11.

- Thử nghiệm mô hình trên hình ảnh và video để phát hiện trái cây.

- Nhận xét đánh giá, kết luận và đưa ra khuyến nghị khi ứng dụng.

Kết quả cần đạt được:

- Đã xây dựng được mô hình phát hiện đối tượng trái cây bằng YOLO11.

- Xây dựng được ứng dụng phát hiện loại trái cây trên hình ảnh và video.

- Viết báo cáo đồ án tốt nghiệp.

Đề tài tốt nghiệp được giao ngày tháng năm

Yêu cầu phải hoàn thành xong

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

TS. Hồ Thị Hương Thơm

Hải Phòng, ngày tháng năm 2025

TRƯỞNG KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Hồ Thị Hương Thơm

Đơn vị công tác: Khoa Công nghệ thông tin - Trường Đại học Hàng Hải

Họ và tên sinh viên : Phạm Thành Lâm **Ngành:** Công nghệ Thông tin

Nội dung hướng dẫn : Toàn bộ đề tài

1.Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

.....
.....
.....
.....
.....

2.Đánh giá chất lượng của đề án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

.....
.....
.....
.....
.....

3.Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ

Không được bảo vệ

Điểm hướng dẫn

Hải Phòng, ngày tháng năm 2025

Giảng viên hướng dẫn

(ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên :.....

Đơn vị công tác:.....

Họ và tên sinh viên:.....Ngành:.....

Đề tài tốt nghiệp:.....

.....

1. Phần nhận xét của giảng viên chấm phản biện

.....
.....
.....
.....

2. Những mặt còn hạn chế

.....
.....
.....
.....

3. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm phản biện

Hải Phòng, ngày.....tháng năm 2025

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong quá trình làm đồ án vừa qua vì được sự chỉ dẫn nhiệt tình của thầy\cô TS. Hồ Thị Hương Thơm – Trường Đại học Hàng Hải Hải, em đã hoàn thành đồ án của mình. Mặc dù em đã cố gắng với sự tận tâm của thầy\cô, nhưng vì thời gian và khả năng nên đồ án của em vẫn còn không tránh được những điều thiếu sót.

Em xin chân thành và bày tỏ lòng biết ơn sâu sắc đến thầy\cô vì đã tận tình chỉ bảo, hướng dẫn và giành thời gian quý báu của mình cho em trong thời gian qua để em có thể hoàn thành đồ án của mình đúng thời hạn.

Em xin cảm ơn tất cả thầy cô giáo trong khoa Công nghệ thông tin vì đã truyền đạt cho em rất nhiều các kiến thức nền tảng, chuyên ngành, chuyên môn và chuyên sâu cực kì vững chắc trong những năm qua để em có thể hoàn thành được đồ án này.

Em xin cảm ơn Trường Đại học Quản lý và Công nghệ Hải Phòng vì không ngừng hỗ trợ và đào tạo những điều kiện tốt nhất trong những năm vừa qua để em có thể học và thực hiện tốt đồ án.

Em xin chân thành cảm ơn !

LỜI CAM ĐOAN

Sinh viên xin cam đoan đề tài “Xây dựng mô hình hỗ trợ phát hiện một số loại trái cây hư hỏng giúp loại bỏ kịp thời” là công trình nghiên cứu độc lập dưới sự hướng dẫn của TS. Hồ Thị Hương Thom – Giảng viên Khoa Công nghệ thông tin Trường Đại Học Hàng Hải Việt Nam. Đề tài là sự cố gắng, quyết tâm của cá nhân sinh viên trong việc tiếp cận kiến thức chuyên môn mới để hoàn thành đạt mục tiêu của đề tài. Các dữ liệu đề nghiên cứu thực nghiệm là trung thực có nguồn gốc trích dẫn rõ ràng.

Sinh viên xin hoàn toàn chịu trách nhiệm với lời cam đoan này

Người cam đoan

Phạm Thành Lâm

Mục lục

LỜI CẢM ƠN.....	7
LỜI CAM ĐOAN	8
Danh mục hình.....	12
Bảng cụm từ viết tắt.....	14
CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN	1
1.1. Đặt vấn đề.....	1
1.2. Phát hiện đối tượng	3
1.2.1. Khái niệm:	3
1.2.2. Các phương pháp phát hiện đối tượng phổ biến	4
1.3. Phương pháp mô tả đặc trưng(HOG).....	7
1.4. Mạng nơ-ron	8
1.5. Faster R-CNN	9
1.6. Giới thiệu về YOLO và các phiên bản.....	11
1.6.1. YOLOv1	11
1.6.2. YOLOv2	11
1.6.3. YOLOv3	12
1.6.4. YOLOv4	14
1.6.5. YOLOv5	15
1.6.6. YOLOv6	16
1.6.7. YOLOv7	17
1.6.8. YOLOv8	18
1.7. Phạm vi nghiên cứu.....	20
1.8. Giải pháp đề xuất	24
1.9. Ý nghĩa khoa học và thực tiễn của đề tài.....	24
CHƯƠNG 2: PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG.....	26
2.1. Ứng dụng của YOLO11	26
2.2. Sơ đồ triển khai.....	28
2.3. Thành phần chính của mô hình.....	29

2.4. Phương pháp phát hiện bằng YOLO11	30
2.5. Ứng dụng YOLO11 trong phát hiện trái cây hư hỏng.....	32
2.6. Hoạt động của mô hình sau huấn luyện.....	33
2.7. Các chỉ số đánh giá hiệu quả mô hình	33
2.8. Các kỹ thuật tăng cường dữ liệu (Data Augmentation).....	34
2.9. Các thách thức khi triển khai thực tế	34
CHƯƠNG 3: CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ	35
3.1. Môi trường thử nghiệm	35
3.1.1. Python.....	35
3.1.2. Đặc tính.....	36
3.1.3. Các phiên bản của Python	37
3.2. Visual Studio Code.....	38
3.2.1. Khái niệm.....	38
3.2.2. Các đặc trưng	38
3.3. Google Collab	40
3.3.1. Khái niệm.....	40
3.3.2. Những đặc trưng của Google Colab	40
3.4. Androi studio.....	41
3.4.1. Tổng quan về Android Studio.....	41
3.4.2. Cấu trúc và thành phần chính của Android Studio.....	42
3.4.3. Ưu điểm của Android Studio	43
3.4.4. Ứng dụng thực tế trong phát triển phần mềm	43
3.4.5. Kết luận.....	43
3.5. Giới thiệu về tập huấn luyện dữ liệu.....	44
3.6. Đào tạo mô hình phát hiện loại trái cây	45
3.7. Giá trị các biểu đồ	52
3.7.1. Kết quả thử nghiệm mô hình đã huấn luyện trên desktop	53
.....	54
3.7.2. Kết quả thử nghiệm mô hình đã huấn luyện trên android	54
3.8. Đánh giá mô hình sau khi thử nghiệm.....	56

3.8.1. Kết quả huấn luyện	56
3.8.2. Ưu điểm của mô hình:	57
3.8.3. Nhược điểm của mô hình:	57
3.8.4. Đề xuất cải tiến:	57
TÀI LIỆU THAM KHẢO	61
A. Tài liệu tiếng Việt.....	61
B. Tài liệu tiếng Anh	61

Danh mục hình

Hình 1.1. Các loại trái cây.	3
Hình 1.2. Ảnh phương pháp truyền thống.	5
Hình 1.3. Ảnh phương pháp học sâu.	6
Hình 1.4. Kiến trúc hệ thống HOG để phát hiện đối tượng.	7
Hình 1.5. Mô hình R-CNN.	9
Hình 1.6. Faster R-CNN – Object Detection Algorithm.	10
Hình 2.2. Ứng dụng YOLOv3 cho tủ lạnh thông minh trong ẩm thực và nhà hàng.	13
Hình 2.3. Phát hiện đối tượng nhỏ với YOLOv5 trong phân tích giao thông.....	16
Hình 2.4. Ứng dụng hệ thống AI được đào tạo để phát hiện máy bay với YOLOv7	17
Hình 2.5. Kiến trúc YOLOv8.	20
Hình 1.7. ảnh những quả cam bị mốc.	21
Hình 1.8. Ảnh những táo bị sâu.	21
Hình 1.9. Ảnh quả chuối bị thâm.	22
Hình 1.10. Ảnh quả ôi bị hỏng.	22
Hình 1.11. Ảnh chùm nho bị hỏng.	23
Hình 1.12. Ảnh quả xoài bị hỏng.	23
Hình 2.1. Sơ đồ triển khai.	28
Hình 2.6. Cấu trúc YOLO11.	31
Hình 3.1. Ngôn Ngữ Python.....	38
Hình 3.2. Phần mềm VS Code.....	39
Hình 3.3. Google Collab	41
Hình 3.4. Tập ảnh dữ liệu huấn luyện.....	45
Hình 3.5. Giao diện Google Collab	46
Hình 3.6. Ấn chọn biểu tượng drive trong phần tệp.....	47
Hình 3.7: kết nối qua đoạn mã	47
Hình 3.8. đoạn mã để bắt đầu train.....	47
Hình 3.9. Code thư viện và chuyển đổi sang file tfile để chạy trên di động.....	48
Hình 3.10. Giao diện chương trình phát hiện trên laptop.....	49

Hình 3.11. Giao diện chương trình phát triển trên android studio.....	50
Hình 3.12. Giao diện chương trình phát triển trên android studio.....	51
Hình 3.13. Biểu đồ thống kê sau thi đào tạo.....	52
Hình 3.14. Ảnh đã phát hiện được trái cây hỏng trên chương trình desktop	54
Hình 3.15. Ảnh đã phát hiện được trái cây hỏng trên ứng dụng di động.....	55

Bảng cụm từ viết tắt

Từ viết tắt	Định nghĩa / Diễn giải
AI	Trí tuệ nhân tạo (Artificial Intelligence)
YOLO	You Only Look Once
R-CNN	Region-based CNN ³ (Mạng nơ-ron tích chập theo vùng)
SSD	Single Shot Detector ⁵ hoặc Single Shot MultiBox Detector
FAO	Food and Agriculture Organization
CNN	Mạng nơ-ron tích chập
SIFT	Scale-Invariant Feature Transform
RPN	Region Proposal Network (Mạng đề xuất vùng)
HOG	Histogram of Oriented Gradient (Phương pháp mô tả đặc trưng)
mAP	mean Average Precision (độ chính xác trung bình)
FPS	frames per second (khung hình mỗi giây)
IoU	Intersection over Union
BoF	Bag of Freebies (Túi quà tặng)
BoS	Bag of Specials (Túi đặc biệt)
SPPF	Spatial Pyramid Pooling - Fast
C2PSA	Cross-Stage Partial with Spatial Attention
C3K2	Cross-Stage Partial with Kernel size 2
VS Code	Visual Studio Code
IDE	Integrated Development Environment (Môi trường phát triển tích hợp)

Bảng 3.2. Bảng thử nghiệm phân loại hình ảnh tĩnh nhiều loại quả

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN

Công nghệ phát hiện đối tượng trong ảnh và video hiện nay chủ yếu dựa trên các mô hình học sâu, trong đó YOLO (You Only Look Once) là một trong những phương pháp nổi bật nhờ tốc độ nhanh và độ chính xác cao. YOLO hoạt động theo cơ chế “một lần nhìn”, tức là mô hình dự đoán trực tiếp vị trí và lớp của đối tượng ngay trên toàn bộ ảnh, giúp giảm thời gian xử lý và phù hợp cho các ứng dụng thời gian thực. Bên cạnh YOLO, nhiều phương pháp khác cũng được sử dụng như R-CNN, Fast R-CNN, Faster R-CNN hay SSD. Các phương pháp R-CNN hoạt động theo hai giai đoạn: tạo vùng đề xuất và phân loại, nên thường cho độ chính xác cao nhưng tốc độ chậm hơn. SSD (Single Shot Detector) cũng là mô hình một giai đoạn giống YOLO, nhưng khả năng nhận diện các đối tượng nhỏ và tốc độ tối ưu kém hơn trong nhiều bài toán thực tế. Nhìn chung, YOLO nổi bật nhờ sự cân bằng tốt giữa tốc độ và độ chính xác, nên được lựa chọn rộng rãi trong các hệ thống giám sát, phân loại và theo dõi đối tượng hiện đại.

1.1. Đặt vấn đề

Trong thời đại công nghiệp 4.0, công nghệ trí tuệ nhân tạo (AI) ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, giao thông thông minh, an ninh giám sát và đặc biệt là nông nghiệp thông minh (smart agriculture). Nông nghiệp hiện đại không chỉ tập trung vào nâng cao năng suất mà còn chú trọng đến chất lượng sản phẩm nhằm đáp ứng nhu cầu ngày càng cao của người tiêu dùng. Trong đó, trái cây tươi là một trong những loại nông sản có giá trị kinh tế cao nhưng dễ bị hư hỏng trong quá trình thu hoạch, bảo quản và phân phối.

Theo báo cáo của FAO (Food and Agriculture Organization), có đến 45% lượng trái cây toàn cầu bị thất thoát hoặc hư hỏng trước khi đến tay người tiêu dùng, gây tổn thất kinh tế hàng tỷ USD mỗi năm. Đặc biệt, trong các chuỗi cung ứng dài, việc vận chuyển và bảo quản trái cây thường gặp nhiều rủi ro dẫn đến suy giảm chất

lượng. Trong khi đó, phương pháp kiểm tra thủ công hiện tại thường tốn thời gian, yêu cầu nhiều lao động, và độ chính xác phụ thuộc nhiều vào kinh nghiệm cá nhân của người kiểm tra.

Sự phát triển mạnh mẽ của thị giác máy tính (Computer Vision) kết hợp với học sâu (Deep Learning) mở ra cơ hội ứng dụng công nghệ hiện đại vào việc giám sát chất lượng trái cây. Các mô hình phát hiện đối tượng (Object Detection) dựa trên mạng nơ-ron tích chập (CNN) như R-CNN, SSD, YOLO đã đạt được nhiều thành tựu trong việc nhận diện và định vị đối tượng. Trong đó, YOLO nổi bật với ưu điểm phát hiện thời gian thực, tốc độ nhanh, độ chính xác cao, phù hợp để triển khai trong các dây chuyền sản xuất và giám sát tự động.

Tuy nhiên, bài toán phát hiện trái cây hư hỏng lại khó hơn so với bài toán nhận diện loại trái cây. Lý do là:

- **Đặc điểm biến đổi phức tạp:** Trái cây hư hỏng có thể biểu hiện ở nhiều dạng khác nhau như dập nát, thối, mốc, đổi màu, khó xác định bằng đặc trưng cố định.

- **Sự tương đồng thị giác:** Một số trái cây có vết sần tự nhiên hoặc vùng tối màu dễ nhầm lẫn với tình trạng hư hỏng.

- **Môi trường thu thập đa dạng:** ảnh chụp trong điều kiện ánh sáng, góc chụp, nền ảnh khác nhau ảnh hưởng trực tiếp đến hiệu quả nhận diện.



Hình 1.1. Các loại trái cây.

Để giải quyết những thách thức này, nghiên cứu đề xuất xây dựng hệ thống phát hiện trái cây hư hỏng dựa trên mô hình YOLO11 – phiên bản mới nhất trong họ YOLO, với khả năng tối ưu về tốc độ và độ chính xác, đặc biệt hiệu quả với các đối tượng nhỏ hoặc bị che khuất. Mục tiêu của hệ thống là tự động phát hiện và phân loại trái cây bình thường và hư hỏng, từ đó hỗ trợ giảm thất thoát sau thu hoạch, nâng cao hiệu quả quản lý trong chuỗi cung ứng và đảm bảo an toàn thực phẩm.

1.2. Phát hiện đối tượng

1.2.1. Khái niệm:

Phát hiện đối tượng (Object Detection) là một trong những bài toán nền tảng của thị giác máy tính. Khác với phân loại ảnh (Image Classification) chỉ đưa ra nhãn của toàn bộ ảnh, phát hiện đối tượng phải đồng thời xác định sự hiện diện và vị trí cụ thể của đối tượng trong ảnh hoặc video thông qua bounding box.

Cụ thể, bài toán này phải trả lời đồng thời hai câu hỏi:

1. Đối tượng cần tìm có xuất hiện trong ảnh/video hay không?
2. Nếu có, vị trí cụ thể của đối tượng nằm ở đâu trong ảnh?

Phát hiện đối tượng là nền tảng quan trọng cho nhiều ứng dụng như:

- **Giao thông thông minh:** nhận diện phương tiện, người đi bộ, biển báo.
- **An ninh giám sát:** phát hiện xâm nhập, theo dõi hành vi.
- **Y tế:** hỗ trợ chẩn đoán từ ảnh X-quang, MRI.
- **Nông nghiệp:** phân loại cây trồng, phát hiện sâu bệnh, giám sát chất lượng nông sản.

1.2.2. Các phương pháp phát hiện đối tượng phổ biến

Để phát hiện đối tượng, có thể sử dụng các kỹ thuật xử lý ảnh truyền thống hoặc mạng học sâu hiện đại. Mỗi phương pháp có ưu điểm và nhược điểm riêng, phù hợp với các yêu cầu và điều kiện khác nhau.

a. Phương pháp truyền thống

Trước khi học sâu bùng nổ, phát hiện đối tượng chủ yếu dựa trên đặc trưng thủ công (hand-crafted features):

- **SIFT (Scale-Invariant Feature Transform):** trích xuất đặc trưng cục bộ bền vững trước thay đổi kích thước và góc nhìn.
- **Haar Cascade Classifier:** phát hiện nhanh khuôn mặt dựa trên đặc trưng.

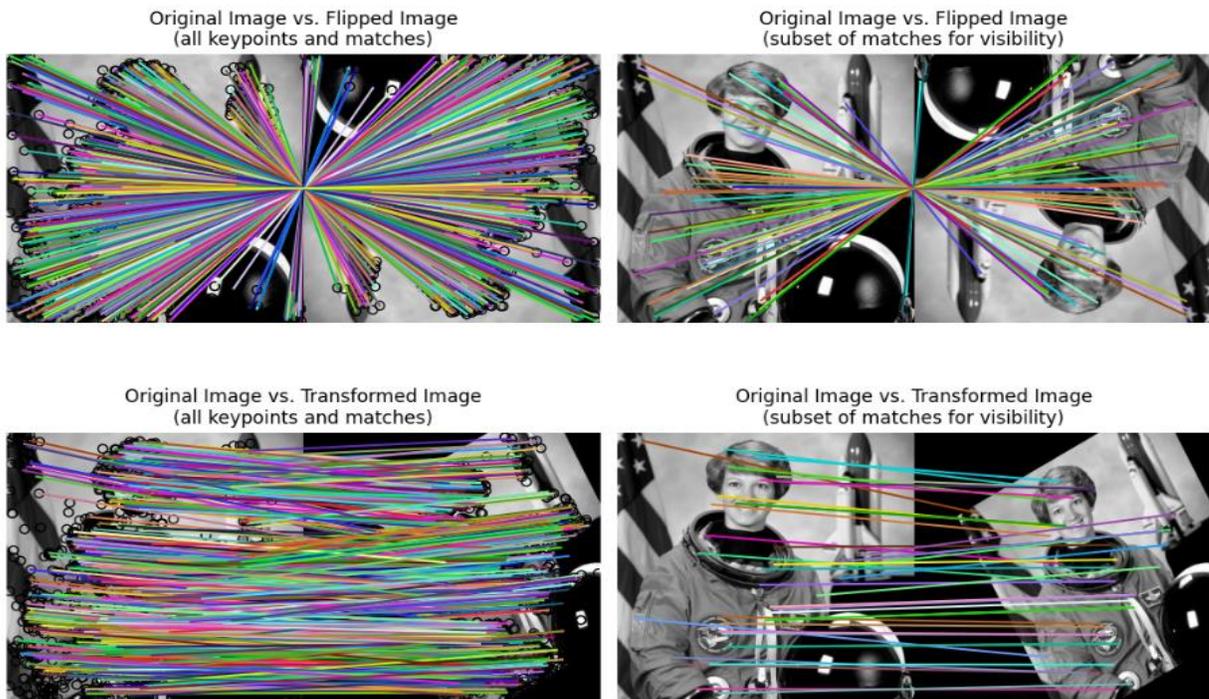
Ưu điểm:

- Tốc độ xử lý nhanh, không yêu cầu nhiều tài nguyên.
- Dễ triển khai trên thiết bị hạn chế tính toán.

Nhược điểm:

- Độ chính xác thấp trong môi trường phức tạp.
- Khó mở rộng cho nhiều loại đối tượng khác nhau.

- Không thích ứng tốt với ánh sáng thay đổi, che khuất hoặc nhiễu nền.



Hình 1.2. Ảnh phương pháp truyền thống.

b. Phương pháp học sâu hiện đại

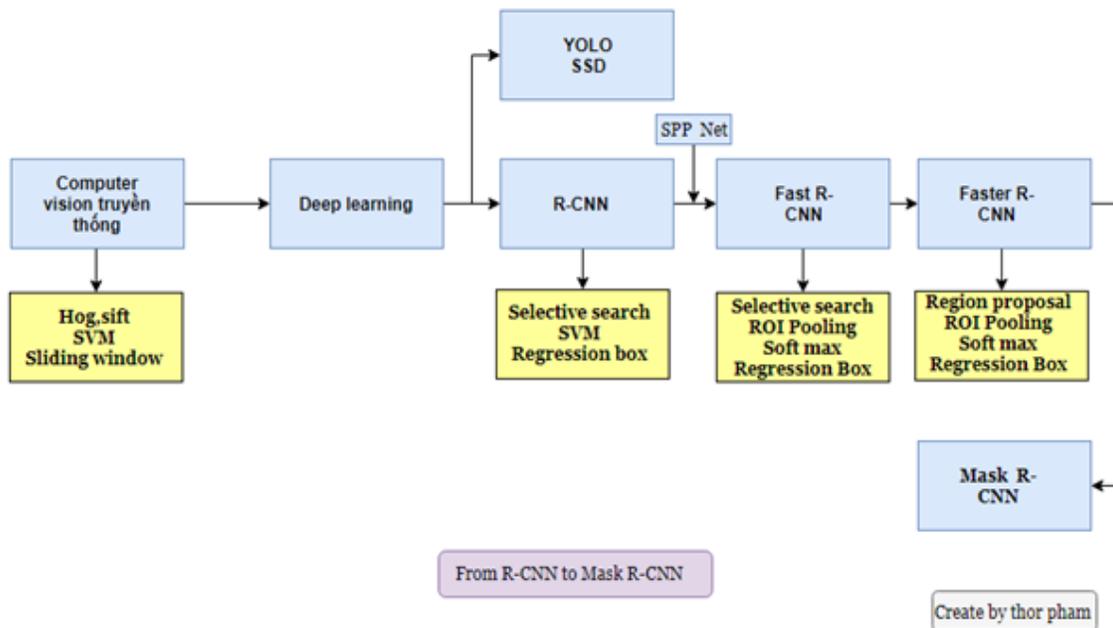
Các phương pháp học sâu thường phụ thuộc vào học có giám sát hoặc không giám sát, với các phương pháp học có giám sát là tiêu chuẩn trong các nhiệm vụ thị giác máy tính. Hiệu suất của các phương pháp này bị giới hạn bởi sức mạnh tính toán của GPU, đang tăng nhanh từng năm.

Ưu điểm:

- Độ chính xác cao.
- Khả năng học đặc trưng mạnh mẽ từ dữ liệu phức tạp.
- Phù hợp với các ứng dụng thực tế có nhiều đối tượng và môi trường đa dạng.

Nhược điểm:

- Yêu cầu lượng dữ liệu lớn để huấn luyện.
- Đòi hỏi GPU có khả năng tính toán mạnh.

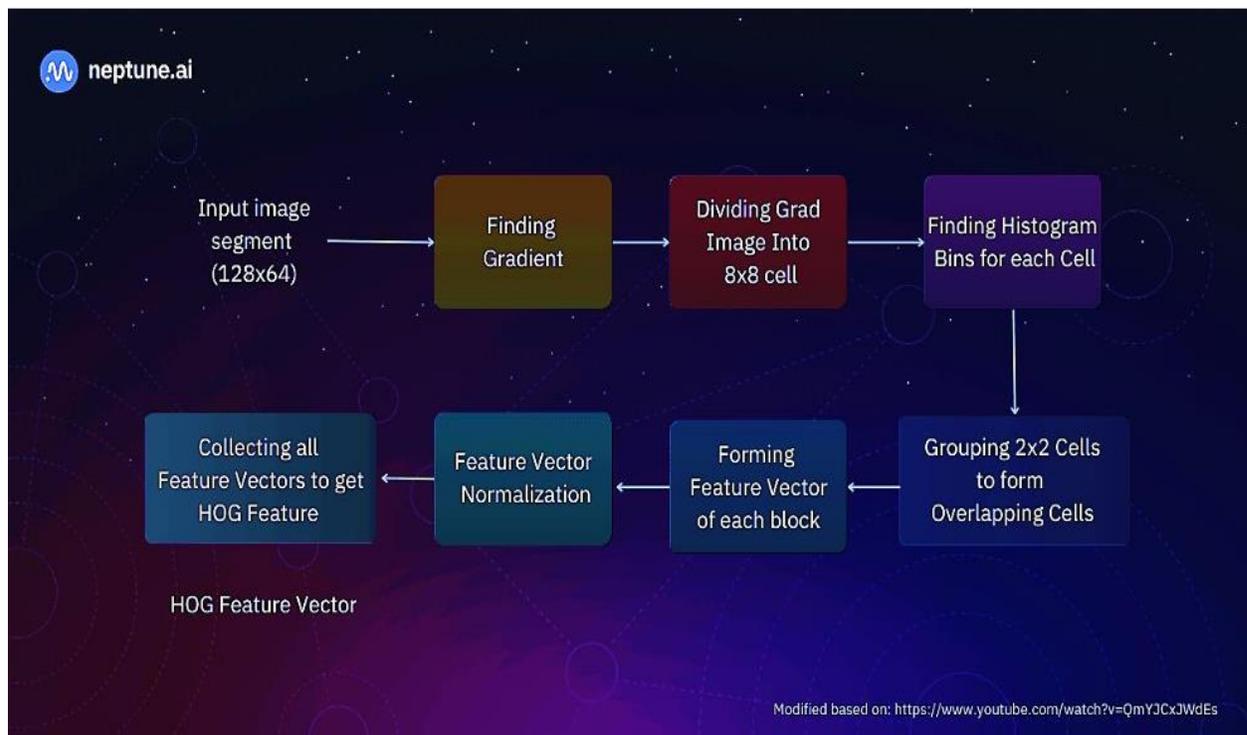


Hình 1.3. Ảnh phương pháp học sâu.

Sự xuất hiện của mạng nơ-ron tích chập (CNN) đã tạo bước ngoặt cho Object Detection. Các mô hình tiêu biểu:

- **R-CNN (Region-based CNN)**: sinh vùng đề xuất, sau đó phân loại bằng CNN.
- **Fast R-CNN và Faster R-CNN**: cải tiến tốc độ bằng cách chia sẻ đặc trưng, Faster R-CNN tích hợp Region Proposal Network (RPN).
- **SSD (Single Shot MultiBox Detector)**: dự đoán bounding box trực tiếp từ đặc trưng, tốc độ nhanh.
- **YOLO (You Only Look Once)**: coi phát hiện như một bài toán hồi quy duy nhất, dự đoán bounding box và nhãn trong một lần tính toán.
- **HOG (Phương pháp mô tả đặc trưng)**: sử dụng thông tin về sự phân bố của các cường độ gradient hoặc của hướng biên để mô tả các đối tượng cục bộ trong ảnh.

1.3. Phương pháp mô tả đặc trưng(HOG)



Hình 1.4. Kiến trúc hệ thống HOG để phát hiện đối tượng.

Phương pháp mô tả đặc trưng (Histogram of Oriented Gradient – HOG) là một trong những phương pháp phát hiện đối tượng lâu đời nhất. Nó được giới thiệu lần đầu tiên vào năm 1986. Mặc dù có một số phát triển trong thập kỷ sau đó, cách tiếp cận này không trở nên phổ biến cho đến năm 2005 khi nó bắt đầu được sử dụng trong nhiều nhiệm vụ liên quan đến thị giác máy tính. HOG sử dụng trình trích xuất đặc trưng để xác định các đối tượng trong hình ảnh.

Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient hoặc của hướng biên để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là cell và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường

độ trong một vùng lớn hơn cell, gọi là các khối và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả sau bước chuẩn hóa sẽ là một vec tor đặc trưng có tính bất biến đối với các thay đổi về điều kiện ánh sáng.

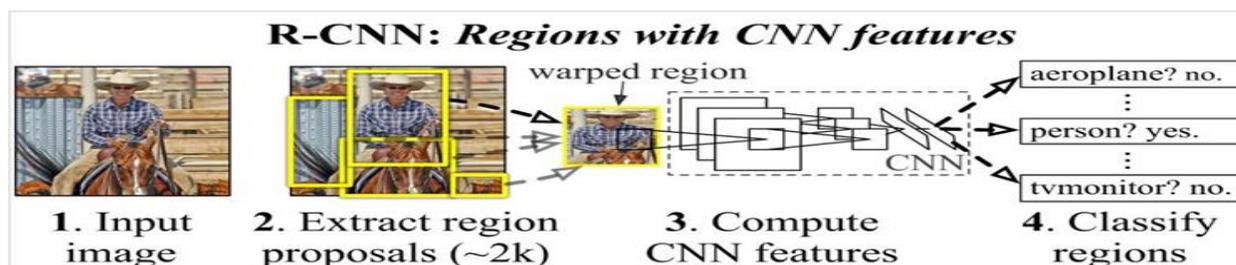
Trước khi tahiều kiến trúc tổng thể của HOG, hãy xem cách nó hoạt động. Đối với một điểm ảnh cụ thể trong hình ảnh, biểu đồ tần suất của gradient được tính bằng cách xem xét các giá trị dọc và ngang để có được các vector đối tượng. Với sự trợ giúp của độ lớn gradient và các góc gradient, ta có thể có được giá trị rõ ràng cho điểm ảnh hiện tại bằng cách khám phá các thực thể khác trong môi trường xung quanh ngang và dọc của chúng.

Như thể hiện trong biểu diễn hình ảnh ở trên, chúng tôi sẽ xem xét một phân đoạn hình ảnh có kích thước cụ thể. Bước đầu tiên là tìm gradient bằng cách chia toàn bộ tính toán của hình ảnh thành các biểu diễn gradient của 8×8 ô. Với sự trợ giúp của 64 vector gradient đạt được, ta có thể chia mỗi ô thành các góc và tính toán biểu đồ cho diện tích cụ thể. Quá trình này làm giảm kích thước của 64 vector xuống kích thước nhỏ hơn là 9 giá trị.

Khi chúng tôi có được kích thước của các giá trị biểu đồ điểm 9 cho mỗi ô, chúng tôi có thể chọn tạo chồng chéo cho các khối ô. Các bước cuối cùng là tạo các khối đặc trưng, chuẩn hóa các vector đặc trưng thu được và thu thập tất cả các vector đặc trưng để có được đặc trưng HOG tổng thể.

1.4. Mạng nơ-ron

Mạng nơ-ron tích chập theo vùng (R-CNN)



Hình 1.5. Mô hình R-CNN.

Các mạng nơ-ron tích chập dựa trên khu vực là một cải tiến trong quy trình phát hiện đối tượng từ các phương pháp HOG và SIFT (biến đổi đặc trưng không đổi theo tỉ lệ) trước đây. Trong các mô hình R-CNN, tác giả cố gắng trích xuất các đặc trưng thiết yếu nhất (thường là khoảng 2000 đặc trưng) bằng cách sử dụng các đặc trưng chọn lọc. Quá trình lựa chọn các trích xuất quan trọng nhất có thể được thực hiện với sự trợ giúp của thuật toán tìm kiếm chọn lọc, nhằm đạt được các đề xuất khu vực quan trọng hơn.

Quy trình làm việc của thuật toán tìm kiếm chọn lọc để chọn các đề xuất khu vực quan trọng nhất bao gồm việc tạo nhiều phân đoạn phụ trên một hình ảnh cụ thể và chọn các mục ứng cử viên cho nhiệm vụ của bạn. Thuật toán tham lam sau đó có thể được sử dụng để kết hợp các mục hiệu quả phù hợp cho một quy trình định kỳ, nhằm kết hợp các phân đoạn nhỏ hơn thành các phân đoạn lớn hơn phù hợp.

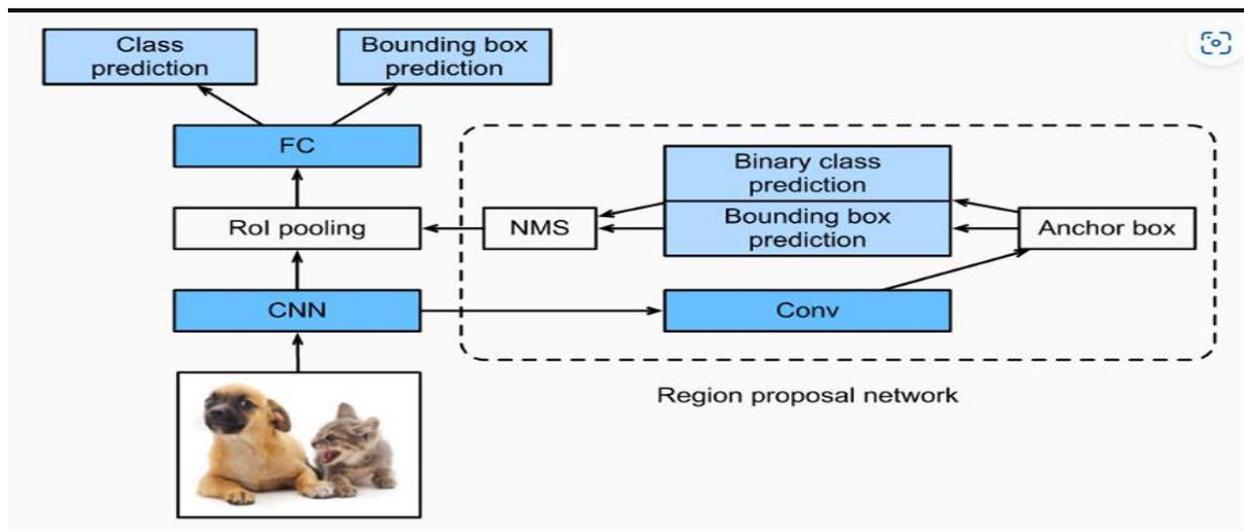
Khi thuật toán tìm kiếm chọn lọc hoàn thành thành công, nhiệm vụ tiếp theo của bạn là trích xuất các đặc trưng và đưa ra dự đoán thích hợp. Sau đó, bạn có thể đưa ra các đề xuất ứng cử viên cuối cùng và các mạng nơ-ron tích chập có thể được sử dụng để tạo vectơ đặc trưng n chiều (2048 hoặc 4096) làm đầu ra. Với sự trợ giúp của mạng nơ-ron tích chập được đào tạo trước, bạn có thể đạt được nhiệm vụ trích xuất đặc trưng một cách dễ dàng.

Bước cuối cùng của R-CNN là đưa ra các dự đoán thích hợp cho hình ảnh và gắn nhãn hộp giới hạn tương ứng cho phù hợp. Để có được kết quả tốt nhất cho mỗi nhiệm vụ, các dự đoán được thực hiện bằng cách tính toán mô hình phân loại cho từng nhiệm vụ, trong khi mô hình hồi quy được sử dụng để sửa phân loại hộp giới hạn cho các vùng được đề xuất.

1.5. Faster R-CNN

R-CNN, một mô hình phát hiện đối tượng, đã đạt được kết quả đáng chú ý nhưng lại gặp phải vấn đề về tốc độ. Để khắc phục điều này, Fast R-CNN đã được giới thiệu. Trong

Fast R-CNN, toàn bộ hình ảnh được truyền qua một mạng nơ-ron tích chập đã được huấn luyện trước, thay vì chỉ xem xét các phân đoạn con của hình ảnh.



Hình 1.6. Faster R-CNN – Object Detection Algorithm.

Tiếp theo, Faster R-CNN, một phiên bản nâng cấp của Fast R-CNN, đã được ra mắt. Faster R-CNN đã thay thế thuật toán tìm kiếm chọn lọc, được sử dụng trong R-CNN và Fast R-CNN để tính toán các đề xuất vùng, bằng một mạng đề xuất vùng vượt trội hơn. Mạng đề xuất vùng (RPN) tính toán hình ảnh từ một phạm vi rộng và các tỷ lệ khác nhau để tạo ra các đầu ra hiệu quả.

RPN giảm thời gian tính toán biên xuống còn khoảng 10 ms cho mỗi hình ảnh. Mạng này bao gồm một lớp tích chập từ đó tácó thể lấy được các bản đồ đặc trưng cần thiết của mỗi pixel. Đối với mỗi bản đồ đặc trưng, tácó nhiều hộp neo có các tỷ lệ, kích thước và tỷ lệ khía cạnh khác nhau. Đối với mỗi hộp neo, tácó dự đoán một lớp nhị phân cụ thể và tạo ra một hộp giới hạn cho nó.

Thông tin sau đó được truyền qua giảm thiểu tối đa để loại bỏ bất kỳ dữ liệu không cần thiết nào vì nhiều chồng chéo được tạo ra khi tạo bản đồ đặc trưng. Đầu ra từ giảm thiểu tối đa được truyền qua vùng quan tâm, và phần còn lại của quá trình và tính toán tương tự như cách hoạt động của Fast R-CNN.

1.6. Giới thiệu về YOLO và các phiên bản

1.6.1. YOLOv1

YOLO chia hình ảnh thành nhiều lưới và tính toán điểm tin cậy và các hộp bao cho mỗi ô lưới, phản ánh xác suất của một đối tượng nằm trong ô lưới đó. Tiếp theo, dựa trên xác suất của đối tượng lớn hơn không, thuật toán tính toán các xác suất lớp tương ứng và nhân chúng với xác suất đối tượng để tạo ra một điểm xác suất tổng thể và hộp bao. Với kiến trúc này, YOLOv1 đã vượt qua R-CNN với độ chính xác trung bình (mean average precision - mAP) là 63.4 và tốc độ suy luận (inference speed) là 45 khung hình mỗi giây (FPS) trên bộ dữ liệu Pascal Visual Object Classes 2007 mã nguồn mở.

Các điểm nổi bật của YOLOv1:

- Phân chia hình ảnh thành lưới: Hình ảnh được chia thành nhiều ô lưới để phân tích.
- Tính toán điểm tin cậy và hộp bao: Mỗi ô lưới sẽ tính toán điểm tin cậy và hộp bao cho các đối tượng có thể nằm trong ô đó.
- Tích hợp xác suất lớp: Nếu xác suất của đối tượng lớn hơn không, thuật toán sẽ tính toán các xác suất lớp tương ứng và nhân chúng với xác suất đối tượng.
- Hiệu suất vượt trội: YOLOv1 đạt mAP 63.4 và tốc độ suy luận 45 FPS, vượt trội hơn so với R-CNN trên bộ dữ liệu Pascal VOC 2007.

1.6.2. YOLOv2

Năm 2016, Joseph Redmon và Ali Farhadi đã giới thiệu YOLOv2, có khả năng phát hiện hơn 9000 loại đối tượng. YOLOv2 mang đến một số cải tiến quan trọng, trong đó có việc giới thiệu anchor boxes – các hộp bao định sẵn được gọi là priors mà mô hình sử dụng để xác định vị trí lý tưởng của một đối tượng. Thuật toán tính toán điểm IoU cho hộp bao dự đoán so với anchor box. Nếu IoU đạt đến ngưỡng nhất định, mô hình sẽ tạo ra dự đoán.

Các điểm nổi bật của YOLOv2:

- Anchor Boxes: Sử dụng các hộp bao định sẵn (priors) để cải thiện độ chính xác trong việc xác định vị trí đối tượng. Điều này giúp mô hình dễ dàng định vị các đối tượng có hình dạng và kích thước khác nhau.

- Intersection over Union (IoU): Thuật toán tính toán điểm IoU giữa hộp bao dự đoán và anchor box. Nếu điểm IoU vượt qua ngưỡng xác định, mô hình sẽ tạo ra dự đoán cuối cùng cho vị trí và loại đối tượng.

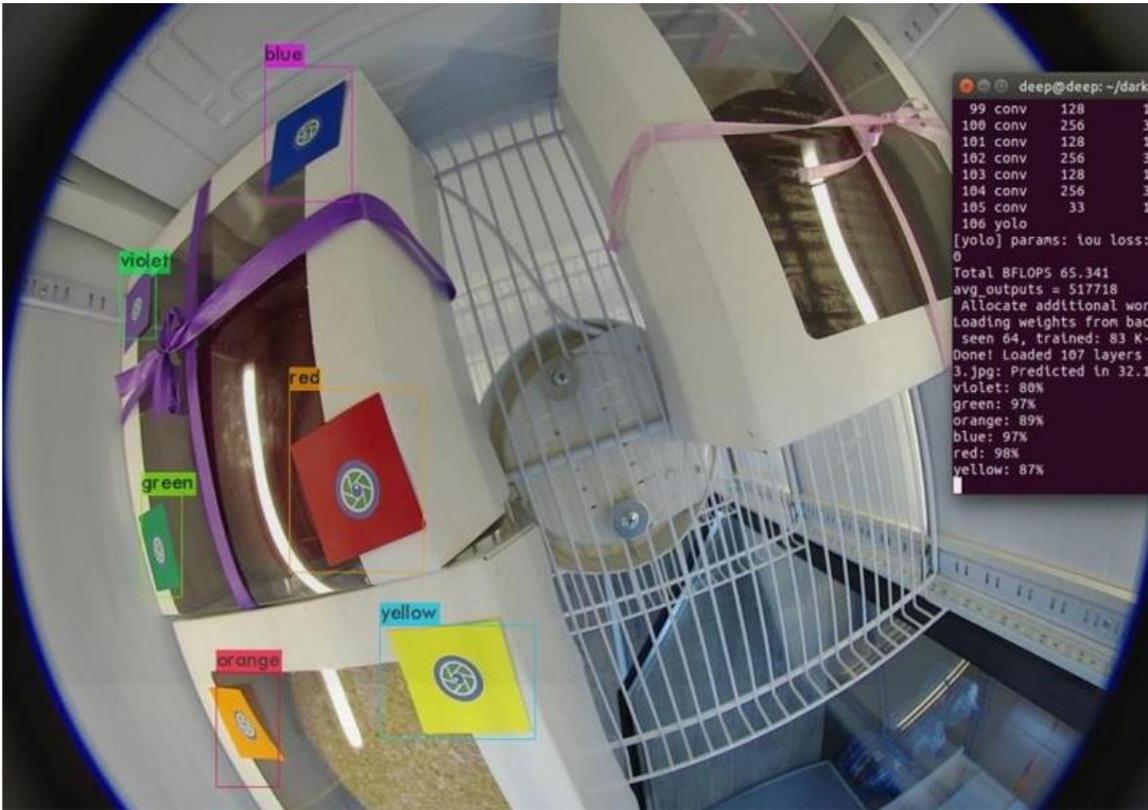
- Hiệu suất vượt trội: YOLOv2 đạt được độ chính xác trung bình (mean Average Precision - mAP) là 76.8 và tốc độ suy luận 67 khung hình mỗi giây (FPS) trên bộ dữ liệu VOC 2007.

Nhờ các cải tiến này, YOLOv2 đã nâng cao khả năng phát hiện đối tượng, không chỉ mở rộng số lượng loại đối tượng có thể nhận diện mà còn cải thiện tốc độ và độ chính xác của quá trình phát hiện. Điều này giúp YOLOv2 trở thành một công cụ mạnh mẽ trong các ứng dụng thị giác máy tính yêu cầu phát hiện đối tượng nhanh chóng và chính xác.

1.6.3. YOLOv3

Năm 2018 Joseph Redmon và Ali Farhadi đã công bố một bài báo giới thiệu YOLOv3, phiên bản mới với độ chính xác cao hơn các phiên bản trước đó.

YOLOv3 đạt được độ chính xác trung bình (mAP) là 28.2 với thời gian xử lý chỉ 22 mili giây.



Hình 2.2. Ứng dụng YOLOv3 cho tủ lạnh thông minh trong ẩm thực và nhà hàng.

Darknet-53 Backbone

YOLOv3 sử dụng Darknet-53 làm kiến trúc xương sống (backbone) cho mô hình. Darknet-53 là một mạng nơ-ron tích chập sâu với 53 lớp, được thiết kế để cân bằng giữa độ chính xác và hiệu suất tính toán.

a. Logistic Classifiers:

Để dự đoán các lớp, YOLOv3 sử dụng bộ phân loại logistic thay vì softmax. Việc sử dụng logistic classifiers giúp mô hình có thể dự đoán nhiều nhãn cho một đối tượng, phù hợp với các tình huống mà một đối tượng có thể thuộc nhiều loại cùng một lúc.

b. Binary Cross-entropy Loss:

YOLOv3 áp dụng hàm mất mát Binary Cross-entropy cho việc huấn luyện, thay vì sử dụng các hàm mất mát truyền thống. Binary Cross-entropy loss giúp cải thiện quá trình tối ưu hóa và độ chính xác của mô hình.

c. Hiệu suất:

Với những cải tiến trên, YOLOv3 đạt mAP 28.2 và có khả năng xử lý với tốc độ 22 mili giây cho mỗi hình ảnh, tương đương với khoảng 45 khung hình mỗi giây (FPS). Điều này cho thấy YOLOv3 không chỉ tăng cường độ chính xác mà còn duy trì tốc độ xử lý nhanh chóng.

Nhờ những cải tiến về kiến trúc và thuật toán, YOLOv3 trở thành một trong những mô hình phát hiện đối tượng hàng đầu, được sử dụng rộng rãi trong nhiều ứng dụng thị giác máy tính đòi hỏi độ chính xác và tốc độ cao.

1.6.4. YOLOv4

Năm 2020, Alexey Bochkovskiy. phát hành YOLOv4, giới thiệu khái niệm Túi quà tặng (BoF) và Túi đặc biệt (BoS). BoF là một tập hợp các kỹ thuật tăng cường dữ liệu giúp tăng độ chính xác mà không mất thêm chi phí suy luận. (BoS tăng cường đáng kể độ chính xác với chi phí tăng nhẹ). Mô hình đạt được 43,5 mAP ở tốc độ 65 FPS trên bộ dữ liệu COCO.

YOLOv4 sử dụng một số đặc trưng sáng tạo hoạt động cùng nhau để tối ưu hóa hiệu suất của nó. Chúng bao gồm Weighted-Residual-Connections, Cross Stage-Partial-connections, Cross-mini-Batch Normalization, Self-adversarial training, Mish-activation, Mosaic data augmentation, DropBlock regularization, và CIOU loss. Các đặc trưng này được kết hợp để đạt được kết quả hiện đại.

Một máy dò đối tượng điển hình bao gồm một số bộ phận bao gồm đầu vào, xương sống, cổ và đầu. Xương sống của YOLOv4 được đào tạo trước trên ImageNet và được sử dụng để dự đoán các lớp và hộp giới hạn của các đối tượng. Xương sống có thể là từ một số mô hình bao gồm VGG, ResNet, ResNeXt hoặc DenseNet. Phần cổ của máy dò được sử dụng để thu thập bản đồ địa vật từ các giai đoạn khác nhau và thường bao gồm một số đường dẫn từ dưới lên và một số đường dẫn từ trên xuống. Phần đầu là những gì được sử dụng để thực hiện phát hiện và phân loại đối tượng cuối cùng.

1.6.5. YOLOv5

Không có tài liệu nghiên cứu chính thức, Ultralytics đã phát hành YOLOv5 cũng vào năm 2020. Mô hình này rất dễ đào tạo vì nó được triển khai trong PyTorch. Kiến trúc mô hình sử dụng khối kết nối một phần giai đoạn chéo làm xương sống cho luồng gradient tốt hơn để giảm chi phí tính toán. YOLOv5 sử dụng tệp YAML thay vì tệp cấu hình trong cấu hình mô hình.

Các đặc trưng chính:

- Chia tách không neo Ultralytics Head: Các mô hình phát hiện đối tượng truyền thống dựa vào các hộp neo được xác định trước để dự đoán vị trí đối tượng. Tuy nhiên, YOLOv5u hiện đại hóa cách tiếp cận này. Bằng cách áp dụng phân chia không neo Ultralytics Đầu tiên, nó đảm bảo một cơ chế phát hiện linh hoạt và thích ứng hơn, do đó nâng cao hiệu suất trong các tình huống đa dạng.

- Tối ưu hóa sự đánh đổi độ chính xác-tốc độ: Tốc độ và độ chính xác thường kéo theo hướng ngược nhau. Nhưng YOLOv5u thách thức sự đánh đổi này. Nó cung cấp một sự cân bằng hiệu chuẩn, đảm bảo phát hiện thời gian thực mà không ảnh hưởng đến độ chính xác. Đặc trưng này đặc biệt vô giá đối với các ứng dụng yêu cầu phản hồi nhanh, chẳng hạn như xe tự hành, robot và phân tích video thời gian thực.

- Nhiều mô hình Pre-training: Hiểu rằng các nhiệm vụ khác nhau đòi hỏi các bộ công cụ khác nhau, YOLOv5u cung cấp rất nhiều mô hình được đào tạo trước. Cho dù bạn đang tập trung vào Suy luận, Xác thực hay Đào tạo, có một mô hình được thiết kế riêng đang chờ bạn. Sự đa dạng này đảm bảo bạn không chỉ sử dụng giải pháp một kích thước phù hợp với tất cả, mà còn là một mô hình được tinh chỉnh đặc biệt cho thử thách độc đáo của bạn.



Hình 2.3. Phát hiện đối tượng nhỏ với YOLOv5 trong phân tích giao thông

1.6.6. YOLOv6

Là một phiên bản không chính thức khác được giới thiệu vào năm 2022 bởi Meituan – một nền tảng mua sắm của Trung Quốc. Công ty nhắm mục tiêu mô hình cho các ứng dụng công nghiệp với hiệu suất tốt hơn so với người tiền nhiệm của nó. Những thay đổi dẫn đến việc YOLOv6n đạt được mAP là 37.5 ở 1187 FPS trên tập dữ liệu COCO và YOLOv6s đã đạt được 45 mAP ở 484 FPS.

Các đặc trưng chính:

- Mô-đun nổi hai chiều: YOLOv6 giới thiệu một mô-đun ở cổ của máy dò, tăng cường tín hiệu nội địa hóa và mang lại hiệu suất tăng với sự suy giảm tốc độ không đáng kể.

- Chiến lược đào tạo có sự hỗ trợ của neo (Anchor-Assisted Training): Mô hình này đề xuất Anchor-Assisted Training để tận hưởng những lợi ích của cả mô hình dựa trên neo và không có neo mà không ảnh hưởng đến hiệu quả suy luận.

- Thiết kế xương sống và cổ nâng cao: Bằng cách đào sâu YOLOv6 để bao gồm một giai đoạn khác ở xương sống và cổ, mô hình này đạt được hiệu suất hiện đại trên bộ dữ liệu COCO ở đầu vào có độ phân giải cao.

- Chiến lược tự chung cất: Một chiến lược tự chung cất mới được thực hiện để tăng hiệu suất của các mô hình YOLOv6 nhỏ hơn, tăng cường nhánh hồi quy phụ trợ trong quá trình đào tạo và loại bỏ nó theo suy luận để tránh sự suy giảm tốc độ rõ rệt.

1.6.7. YOLOv7

Vào tháng 7/2022, một nhóm các nhà nghiên cứu đã công bố mô hình mã nguồn mở YOLOv7, máy dò đối tượng nhanh nhất và chính xác nhất với mAP là 56,8% ở FPS dao động từ 5 đến 160. YOLOv7 dựa trên Mạng tổng hợp lớp hiệu quả mở rộng, giúp cải thiện đào tạo bằng cách cho phép mô hình học các đặc trưng đa dạng với tính toán hiệu quả.



Hình 2.4. Ứng dụng hệ thống AI được đào tạo để phát hiện máy bay với YOLOv7

Các đặc trưng chính

- Tham số hóa lại mô hình: YOLOv7 đề xuất một mô hình tái tham số hóa theo kế hoạch, đây là một chiến lược áp dụng cho các lớp trong các mạng khác nhau với khái niệm đường dẫn lan truyền gradient.

- Phân công nhân động: Việc đào tạo mô hình với nhiều lớp đầu ra trình bày một vấn đề mới: "Làm thế nào để chỉ định các mục tiêu động cho đầu ra của các

nhánh khác nhau?" Để giải quyết vấn đề này, YOLOv7 giới thiệu một phương pháp gán nhãn mới được gọi là gán nhãn có hướng dẫn từ thô đến mịn.

- Mở rộng và chia tỷ lệ hợp chất: YOLOv7 đề xuất các phương pháp "mở rộng" và "chia tỷ lệ hợp chất" cho máy dò đối tượng thời gian thực có thể sử dụng hiệu quả các tham số và tính toán. - Hiệu quả: Phương pháp do YOLOv7 đề xuất có thể giảm hiệu quả khoảng 40% thông số và 50% tính toán của máy dò đối tượng thời gian thực hiện đại, đồng thời có tốc độ suy luận nhanh hơn và độ chính xác phát hiện cao hơn.

1.6.8. YOLOv8

Là một mô hình tiên tiến, hiện đại, YOLO v8 Được xây dựng dựa trên sự thành công của các phiên bản trước, giới thiệu các đặc trưng và cải tiến mới để nâng cao hiệu suất, tính linh hoạt và hiệu quả. YOLO v8 hỗ trợ đầy đủ các tác vụ AI thị giác, bao gồm phát hiện, phân đoạn, ước tính tư thế, theo dõi và phân loại. Tính linh hoạt này cho phép người dùng tận dụng YOLOv8. Khả năng của họ trên các ứng dụng và lĩnh vực đa dạng.

Kiến trúc của yolo v8

YOLOv8 được thiết kế để khắc phục những hạn chế của các phiên bản trước đồng thời duy trì sự cân bằng giữa tốc độ và độ chính xác. Một cải tiến đáng chú ý của YOLOv8 là thiết kế mô-đun và có thể mở rộng, cho phép mô hình linh hoạt thích ứng với nhiều yêu cầu khác nhau.

Cấu trúc của YOLOv8 được chia thành ba thành phần chính: xương sống, cổ và đầu. Xương sống chịu trách nhiệm trích xuất các đặc trưng từ hình ảnh đầu vào, với các tùy chọn như CSPDarknet53 và EfficientDet để phù hợp với nhu cầu cụ thể. Phần cổ kết nối xương sống với đầu, đóng vai trò quan trọng trong việc hợp nhất các đặc trưng. Phần đầu đảm nhận việc dự đoán các hộp giới hạn, lớp đối tượng và điểm tin cậy.

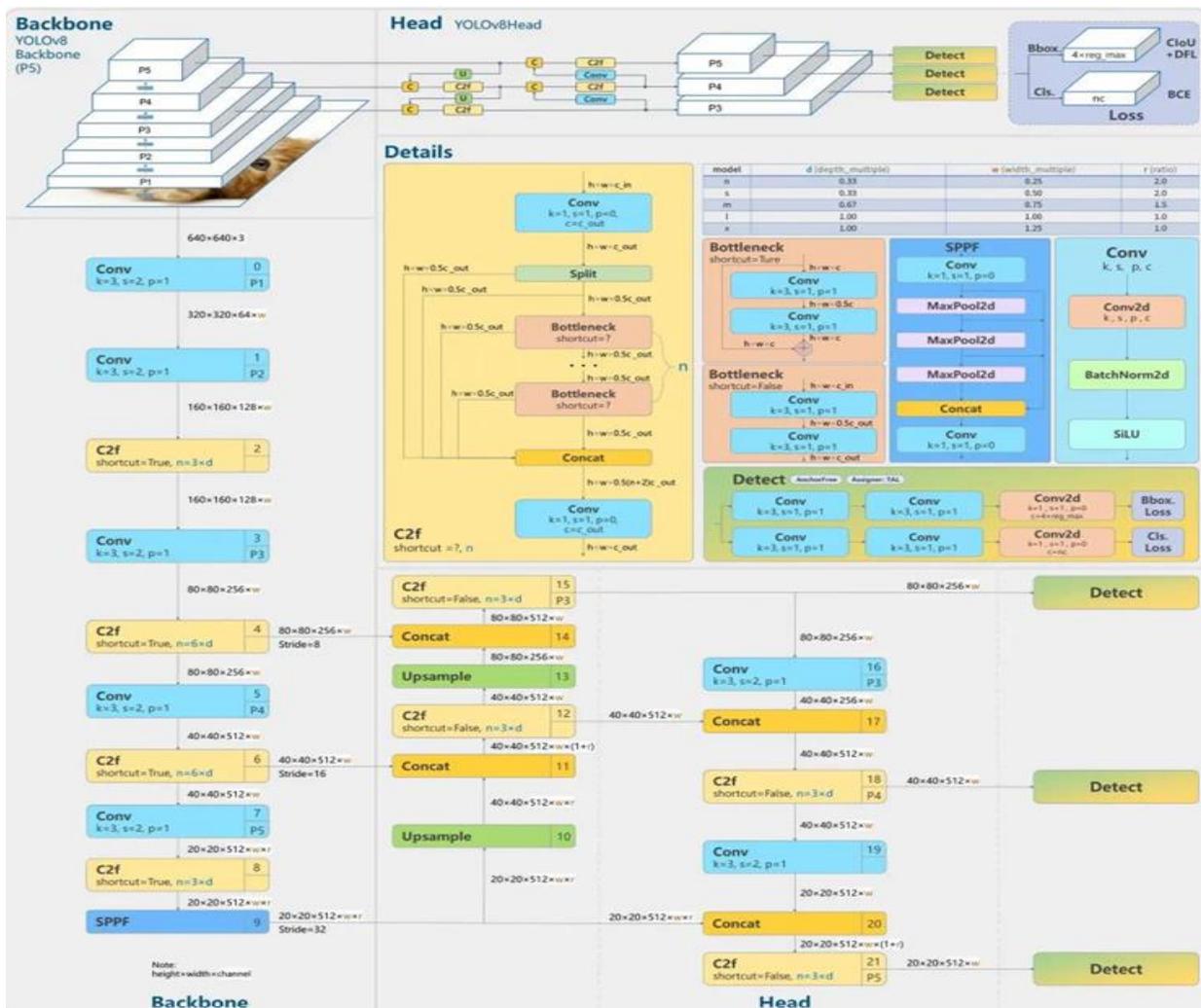
Một khía cạnh quan trọng khác của YOLOv8 là khả năng mở rộng mô hình. Nó cung cấp các biến thể khác nhau như YOLOv8-tiny và YOLOv8x, khác nhau về

kích thước và độ phức tạp tính toán. Điều này cho phép người dùng chọn mô hình phù hợp với yêu cầu của mình, dù là trong môi trường hạn chế tài nguyên hay ứng dụng hiệu suất cao.

YOLOv8 cũng giới thiệu những cải tiến trong chiến lược đào tạo, bao gồm việc sử dụng tối ưu hóa Adam chỉnh lưu (RAdam) và phương pháp phát hiện đối tượng dựa trên neo hoặc không neo. Những cải tiến này giúp quá trình đào tạo hội tụ nhanh hơn và nâng cao hiệu suất trong các nhiệm vụ phát hiện đối tượng.

Hơn nữa, YOLOv8 có hệ thống cấu hình linh hoạt, cho phép người dùng dễ dàng tùy chỉnh các thông số như kích thước đầu vào, hộp neo và độ phức tạp của mô hình. Tính linh hoạt này làm cho YOLOv8 dễ dàng thích ứng với các bộ dữ liệu và kịch bản ứng dụng đa dạng.

Tóm lại, kiến trúc của YOLOv8 nổi bật với thiết kế mô-đun, các biến thể có thể mở rộng, xương sống được cải thiện và các chiến lược đào tạo nâng cao. Những đặc điểm này cùng nhau góp phần vào sự thành công của YOLOv8 trong việc phát hiện đối tượng thời gian thực, khiến nó trở thành lựa chọn phổ biến cho các nhà nghiên cứu và chuyên gia trong lĩnh vực thị giác máy tính.



Hình 2.5. Kiến trúc YOLOv8.

1.7. Phạm vi nghiên cứu

Trong đề tài này, đối tượng nghiên cứu là trái cây hư hỏng, với mục tiêu:

1. Xây dựng mô hình có khả năng phân biệt trái cây bình thường và trái cây hư hỏng.
2. Ứng dụng mô hình vào hệ thống giám sát trong kho, siêu thị và chuỗi cung ứng.



Hình 1.7. ảnh những quả cam bị mốc.



Hình 1.8. Ảnh những táo bị sâu.



Hình 1.9. Ảnh quả chuối bị thâm.



Hình 1.10. Ảnh quả ổi bị hỏng.



Hình 1.11. Ảnh chùm nho bị hỏng.



Hình 1.12. Ảnh quả xoài bị hỏng.

Phạm vi:

- Thu thập và xử lý tập dữ liệu ảnh trái cây hư hỏng.
- Gán nhãn dữ liệu bằng Roboflow hoặc LabelImg.
- Huấn luyện mô hình YOLO11.
- Đánh giá dựa trên các chỉ số: Precision, Recall, mAP.
- Ứng dụng thử nghiệm trên ảnh và video thực tế.

Đối tượng phát hiện:

- Trái cây bình thường: cam, xoài, táo, chuối, nho, ổi, dưa hấu.
- Trái cây hư hỏng: có dấu hiệu thối, mốc, dập nát.

1.8. Giải pháp đề xuất

Giải pháp được lựa chọn là YOLO11 – phiên bản mới nhất trong họ YOLO, nổi bật với khả năng phát hiện nhanh, chính xác và hiệu quả trên dữ liệu có nhiều đối tượng nhỏ. Các bước chính:

- Thu thập dữ liệu (ảnh trái cây bình thường & hư hỏng).
- Tiền xử lý và gán nhãn dữ liệu.
- Huấn luyện mô hình YOLO11.
- Đánh giá mô hình bằng Precision, Recall, mAP.
- Ứng dụng mô hình để phát hiện trái cây hư hỏng trong thực tế.

1.9. Ý nghĩa khoa học và thực tiễn của đề tài

Ý nghĩa khoa học:

- Đóng góp vào lĩnh vực thị giác máy tính trong nông nghiệp, cụ thể là nhận diện trái cây hư hỏng – một bài toán phức tạp hơn nhận diện loại trái cây.

- Cung cấp dữ liệu và mô hình tham khảo cho các nghiên cứu tương lai về phát hiện nông sản hư hỏng.

- Kiểm chứng hiệu quả của YOLO11 – phiên bản mới – trong một bài toán thực tiễn.

Ý nghĩa thực tiễn:

- Hỗ trợ giảm thiểu thất thoát sau thu hoạch, nâng cao lợi nhuận cho nông dân và doanh nghiệp.

- Tăng hiệu quả quản lý chuỗi cung ứng trái cây, từ khâu bảo quản, vận chuyển đến phân phối.

- Nâng cao chất lượng trái cây đến tay người tiêu dùng, góp phần bảo đảm an toàn thực phẩm.

- Góp phần thúc đẩy ứng dụng công nghệ 4.0 trong nông nghiệp thông minh tại Việt Nam.

CHƯƠNG 2: PHƯƠNG PHÁP PHÁT HIỆN ĐỐI TƯỢNG

Trong chương này, đề tài sẽ trình bày các phương pháp và kỹ thuật được sử dụng để phát hiện và phân loại đối tượng trong hình ảnh. Phương pháp phát hiện đối tượng đóng vai trò quan trọng, quyết định đến độ chính xác và hiệu quả của mô hình trong quá trình nhận dạng. Việc lựa chọn thuật toán phù hợp, quy trình xử lý dữ liệu và cơ chế huấn luyện mô hình là những yếu tố then chốt ảnh hưởng trực tiếp đến chất lượng kết quả. Chương này sẽ tập trung mô tả chi tiết các bước xây dựng mô hình, từ tiền xử lý dữ liệu, trích xuất đặc trưng, cho đến kiến trúc mạng học sâu dùng để phát hiện đối tượng.

2.1. Ứng dụng của YOLO11

YOLO11 là mô hình phát hiện đối tượng một bước, hoạt động bằng cách chia ảnh đầu vào thành các ô lưới và dự đoán trực tiếp vị trí, kích thước hộp chứa và nhãn của đối tượng trong một lần xử lý. Mô hình sử dụng kiến trúc hiện đại để đạt độ chính xác cao và tốc độ xử lý nhanh, phù hợp cho các ứng dụng thời gian thực.

a. Mục tiêu của bài toán phát hiện và phân loại trái cây

Trong nông nghiệp hoặc sản xuất, việc tự động nhận dạng và phân loại trái cây giúp:

- Xác định loại trái cây (cam, táo, chuối, xoài, v.v.).
- Phân biệt trái chín và chưa chín.
- Kiểm tra chất lượng, kích thước, màu sắc.
- Hỗ trợ tự động hóa dây chuyền đóng gói hoặc kiểm tra sản phẩm.

b. Ứng dụng của YOLO11 trong bài toán này

Phát hiện và nhận dạng nhiều loại trái cây

YOLO11 có thể được huấn luyện trên bộ dữ liệu gồm nhiều loại trái cây khác nhau. Sau khi huấn luyện, mô hình có khả năng:

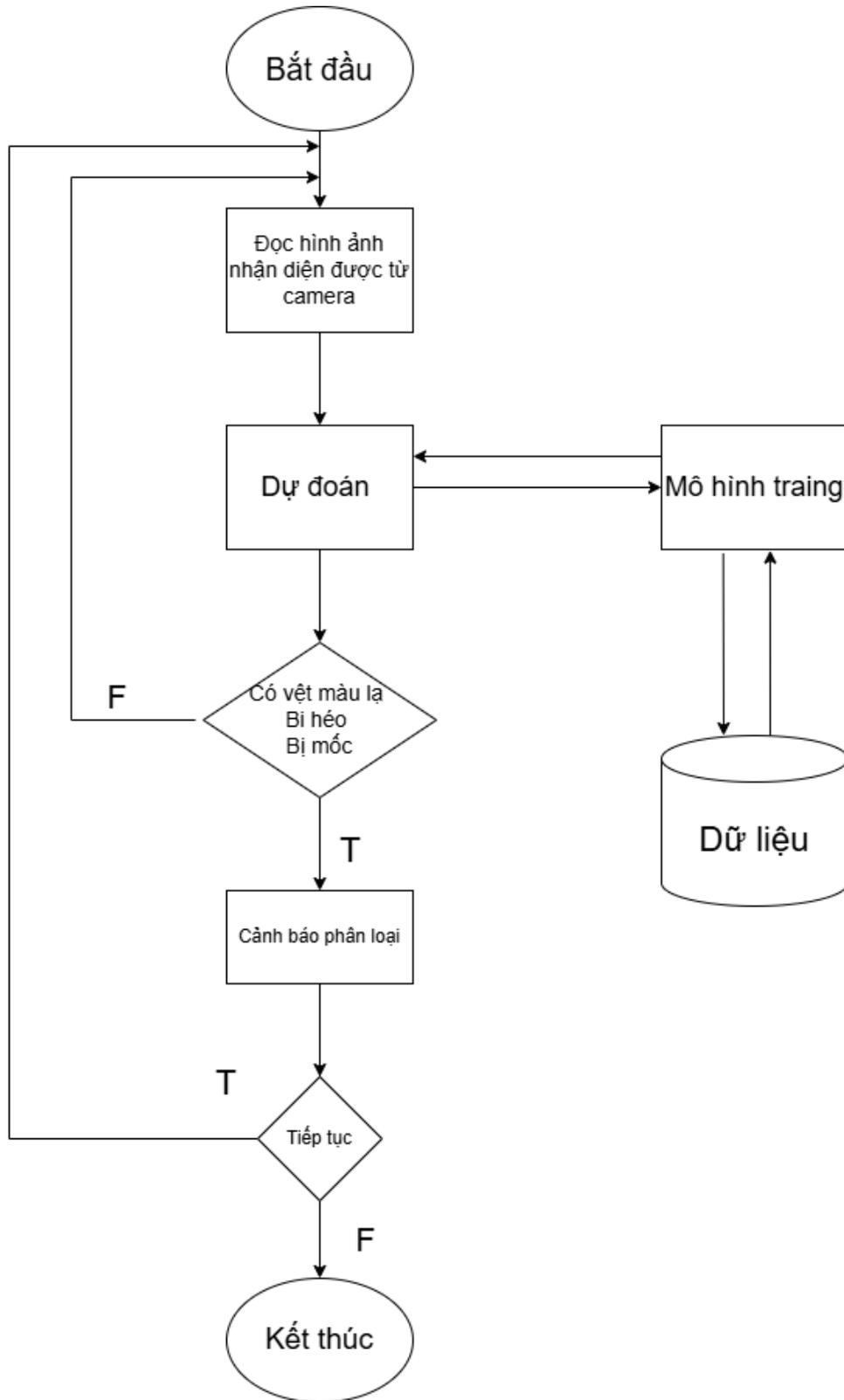
- Nhận diện nhiều loại trái cây cùng lúc trong một ảnh.
- Khoanh vùng (bounding box) từng trái cây kèm theo tên loại (label) và độ tin cậy (confidence).

c. Ứng dụng trong robot hoặc thiết bị thông minh

YOLO11 có thể được tích hợp với camera và robot để:

- Tự động hái trái chín, tránh hái nhầm trái non.
- Theo dõi số lượng trái cây trong quá trình thu hoạch hoặc vận chuyển.
- Ứng dụng trong nhà máy, siêu thị, hoặc hệ thống phân loại tự động.

2.2. Sơ đồ triển khai



Hình 2.1. Sơ đồ triển khai.

2.3. Thành phần chính của mô hình

Quy trình này bắt đầu và kết thúc như sau:

a. Bắt Đầu (Start): Bắt đầu quy trình

- Quy trình được khởi động và hệ thống bắt đầu thực hiện quá trình kiểm tra chất lượng hoa quả.

b. Đọc hình ảnh từ camera:

- Hệ thống thu nhận hình ảnh hoa quả thông qua camera. Đây là nguồn dữ liệu đầu vào để tiến hành phân tích và nhận diện tình trạng của hoa quả.

c. Dự đoán (Predict):

- Hình ảnh thu được sẽ được đưa vào **mô hình đã được huấn luyện** (Machine Learning hoặc Deep Learning).

- Mô hình sử dụng **tập dữ liệu đã được chuẩn bị** để tiến hành phân loại hoặc dự đoán tình trạng của hoa quả.

- Kết quả trả về là **dự đoán về chất lượng**, ví dụ: bình thường, có vết màu lạ, bị héo, bị mốc,...

d. Hệ thống kiểm tra kết quả dự đoán thông qua bộ tiêu chí:

- Hoa quả **có vết màu lạ?**

- Hoa quả **bị héo?**

- Hoa quả **bị mốc?**

e. Xử lý kết quả

Trường hợp F (False – Không phát hiện vấn đề)

- Nếu không có dấu hiệu bất thường, hệ thống coi hoa quả là **đạt chất lượng**.

- Quy trình quay lại bước **Đọc hình ảnh từ camera** để tiếp tục giám sát hoặc kiểm tra quả khác.

Trường hợp T (True – Có vấn đề chất lượng)

- Khi phát hiện hoa quả **có dấu hiệu hỏng hoặc bất thường**, hệ thống chuyển sang bước cảnh báo.

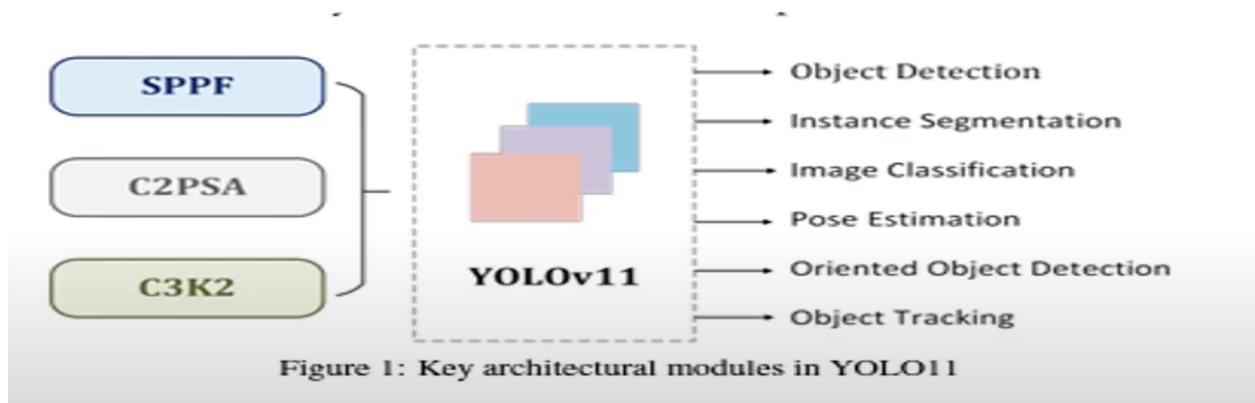
- **Kích hoạt chương trình cảnh báo**: phát âm thanh hoặc hiển thị thông báo để thông báo trái cây bị hỏng.

f. Kết thúc (End)

Nếu người dùng chọn không tiếp tục, quy trình sẽ dừng lại và hệ thống kết thúc hoạt động.

2.4. Phương pháp phát hiện bằng YOLO11

YOLO11 là mô hình phát hiện đối tượng được thiết kế theo cấu trúc gọn nhẹ nhưng hiệu quả, gồm ba phần chính: backbone để trích xuất đặc trưng, neck để kết hợp thông tin ở nhiều mức độ và head để dự đoán vị trí cũng như nhãn đối tượng. Nhờ các khối xử lý được tối ưu hóa và cơ chế dự đoán nhanh, YOLO11 cho phép phát hiện vật thể với độ chính xác cao và tốc độ xử lý tốt, phù hợp cho các bài toán nhận dạng trong thời gian thực.



Các thành phần chính của YOLO11

SPPF (Spatial Pyramid Pooling - Fast): Đây là một mô-đun được cải tiến từ SPP. SPPF tổng hợp các đặc trưng ở nhiều tỷ lệ khác nhau bằng cách sử dụng nhiều lớp max-pooling liên tiếp. Điều này cho phép mô hình thu thập thông tin từ các khu vực có kích thước khác nhau của ảnh đầu vào, giúp cải thiện khả năng phát hiện các đối tượng đa dạng về kích thước, đặc biệt là các đối tượng nhỏ.

C2PSA (Cross-Stage Partial with Spatial Attention): C2PSA là một khối mới bổ sung cơ chế attention (chú ý) vào kiến trúc. Cơ chế này giúp mô hình tập trung vào các vùng quan trọng nhất của hình ảnh, thay vì xử lý toàn bộ một cách đồng đều. Việc kết hợp spatial attention với cấu trúc cross-stage partial giúp cải thiện hiệu quả, cho phép mô hình lọc bỏ thông tin không liên quan và tăng cường các đặc trưng quan trọng để phát hiện đối tượng chính xác hơn trong các cảnh phức tạp.

C3K2 (Cross-Stage Partial with Kernel size 2): C3K2 là một khối tính toán hiệu quả, thay thế cho khối C2F được sử dụng trong YOLOv8. C3K2 sử dụng các lớp tích chập với kernel size nhỏ để trích xuất đặc trưng một cách hiệu quả hơn. Điều này giúp tăng tốc độ xử lý và cải thiện khả năng học của mô hình.

Ưu điểm YOLO11:

- Nhẹ hơn, nhanh hơn YOLOv8.
- Phát hiện tốt hơn đối tượng nhỏ.
- Thuật toán NMS cải tiến, giảm false positive.
- Dễ triển khai đa nền tảng (Python, ONNX, TensorRT).

Nhược điểm YOLO11:

Là một mô hình hiện đại, các biến thể YOLO11 lớn nhất đòi hỏi tài nguyên tính toán đáng kể để đạt được độ chính xác tối đa, mặc dù chúng vẫn rất hiệu quả so với lớp hiệu suất của chúng.

2.5. Ứng dụng YOLO11 trong phát hiện trái cây hư hỏng

Phát hiện trái cây hư hỏng (thối, dập, mốc...) là một bài toán quan trọng trong chuỗi cung ứng nông sản để giảm lãng phí và nâng cao chất lượng. Mô hình YOLO11 (phiên bản mới nhất của dòng YOLO do Ultralytics phát triển) cung cấp giải pháp phát hiện đối tượng thời gian thực với độ chính xác và tốc độ phù hợp cho ứng dụng trên dây chuyền, thiết bị di động hoặc hệ thống giám sát nhà kho.

Mô hình YOLO11 (phiên bản mới nhất của dòng YOLO do Ultralytics phát triển) mang lại nhiều cải tiến vượt trội trong việc xử lý bài toán này. YOLO11 sử dụng kiến trúc mạng CNN lai Transformer giúp mô hình vừa học được đặc trưng cục bộ (vết thối, chỗ dập, mốc trắng, vết nứt trên vỏ...) vừa nhận biết được ngữ cảnh tổng thể của quả (hình dạng, màu sắc tổng thể, vùng sáng – tối). Nhờ đó, YOLO11 có thể phân biệt hiệu quả giữa các khuyết điểm tự nhiên và những dấu hiệu hư hỏng thực sự.

Trong quá trình triển khai, mô hình YOLO11 được huấn luyện trên tập dữ liệu gồm nhiều loại trái cây khác nhau (chuối, táo, cam, xoài, v.v.) với các trạng thái bình thường và hư hỏng ở nhiều mức độ. Sau khi huấn luyện, YOLO11 có thể phát hiện và khoanh vùng chính xác vị trí phần bị hư trong ảnh đầu vào, đồng thời phân loại tình trạng từng quả theo các mức như “tốt”, “bị dập nhẹ”, “hư hỏng nặng”, “mốc”.

Với khả năng xử lý thời gian thực và tốc độ cao, YOLO11 rất phù hợp để tích hợp vào:

- **Dây chuyền sản xuất tự động:** camera giám sát liên tục dòng trái cây đi qua băng tải và hệ thống sẽ tự động loại bỏ quả hư.

- **Thiết bị di động:** cho phép người dùng hoặc nông dân quét nhanh từng lô hàng bằng điện thoại để kiểm tra chất lượng.

- **Hệ thống giám sát kho:** kiểm tra tình trạng bảo quản và phát hiện sớm trái cây hư hỏng lan rộng.

Ngoài ra, YOLO11 còn hỗ trợ khả năng tối ưu hóa mô hình để triển khai trên các thiết bị có cấu hình thấp (như Raspberry Pi hoặc Jetson Nano), giúp việc ứng dụng trong thực tế dễ dàng và tiết kiệm chi phí hơn.

Nhờ đó, việc áp dụng YOLO11 không chỉ nâng cao hiệu quả giám sát và phân loại trái cây, mà còn góp phần xây dựng hệ thống chuỗi cung ứng nông sản thông minh, đảm bảo chất lượng sản phẩm và giảm thiểu thiệt hại kinh tế do hàng hư hỏng.

2.6. Hoạt động của mô hình sau huấn luyện

Sau khi huấn luyện, mô hình có thể dự đoán:

- Input: quay trực tiếp qua camera hoặc ảnh hoặc video chứa trái cây.
- Output: bounding box + nhãn lớp (bình thường/hư hỏng).

Ví dụ:

- Một quả xoài có vết thối → gán nhãn “Quả xoài hỏng”
- Một chùm nho tươi → gán nhãn “Nho bình thường”

2.7. Các chỉ số đánh giá hiệu quả mô hình

Để đánh giá hiệu quả của hệ thống phát hiện trái cây hư hỏng, sử dụng các chỉ số sau:

- **Precision (Độ chính xác):** tỷ lệ dự đoán đúng trong số các dự đoán dương tính.
- **Recall (Độ nhạy):** tỷ lệ dự đoán đúng trong số các mẫu thực tế dương tính.
- **F1-score:** trung bình điều hòa giữa Precision và Recall, cân bằng hai yếu tố.

- **MAP (mean Average Precision)**: chỉ số phổ biến trong Object Detection, đo lường mức độ chính xác tổng thể khi phát hiện nhiều đối tượng ở các ngưỡng IoU khác nhau.

Việc sử dụng đồng thời các chỉ số này giúp đánh giá mô hình toàn diện, tránh trường hợp chỉ tốt ở một khía cạnh.

2.8. Các kỹ thuật tăng cường dữ liệu (Data Augmentation)

Vì dữ liệu trái cây hư hỏng thường không phong phú, cần áp dụng các phương pháp tăng cường dữ liệu:

- **Xoay ảnh (rotation)**: tạo sự đa dạng về góc nhìn.
- **Lật ngang/dọc (flip)**: giúp mô hình học được tính đối xứng.
- **Thay đổi độ sáng/độ tương phản**: mô phỏng điều kiện ánh sáng khác nhau.
- **Gaussian noise / blur**: mô phỏng ảnh mờ, nhiễu khi chụp.
- **CutMix / Mosaic augmentation**: ghép nhiều ảnh vào cùng một khung để tăng độ đa dạng dữ liệu.

2.9. Các thách thức khi triển khai thực tế

- **Chi phí hạ tầng**: huấn luyện mô hình yêu cầu GPU mạnh, tốn chi phí.
- **Khả năng triển khai trên thiết bị IoT**: cần tối ưu mô hình để chạy trên máy tính nhúng (Raspberry Pi, Jetson Nano).
- **Chất lượng dữ liệu**: dữ liệu không đồng đều, có thể ảnh hưởng đến độ chính xác.
- **Tính thích ứng**: mô hình có thể cần được huấn luyện lại khi triển khai cho nhiều loại trái cây mới.

CHƯƠNG 3: CÀI ĐẶT, THỬ NGHIỆM VÀ ĐÁNH GIÁ

Phần này tập trung mô tả quy trình triển khai mô hình, từ bước cài đặt môi trường cho đến việc tiến hành các thử nghiệm thực tế. Các cấu hình hệ thống, thiết lập thư viện và dữ liệu sử dụng trong quá trình kiểm thử sẽ được trình bày rõ ràng nhằm đảm bảo khả năng tái hiện kết quả. Song song với đó, các phép đo đánh giá và phân tích kết quả nhận dạng sẽ được sử dụng để phản ánh mức độ hiệu quả của mô hình trong những điều kiện khác nhau. Đây là cơ sở quan trọng để xem xét khả năng ứng dụng thực tiễn cũng như đề xuất hướng cải thiện ở giai đoạn tiếp theo.

3.1. Môi trường thử nghiệm

3.1.1. Python

a. Khái niệm

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

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

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay

vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

3.1.2. Đặc tính

Python đang trở nên phổ biến trong cộng đồng lập trình nhờ các đặc tính sau:

- Ngôn ngữ thông dịch: Python được xử lý trong thời gian chạy bởi trình thông dịch Python.
- Ngôn ngữ hướng đối tượng: Nó hỗ trợ các đặc trưng và kỹ thuật lập trình hướng đối tượng.
- Ngôn ngữ lập trình tương tác: Người dùng có thể tương tác trực tiếp với trình thông dịch python để viết chương trình.
- Ngôn ngữ dễ học: Python rất dễ học, đặc biệt là cho người mới bắt đầu.
- Cú pháp đơn giản: Việc hình thành cú pháp Python rất đơn giản và dễ hiểu, điều này cũng làm cho nó trở nên phổ biến.
- Mã nguồn mở: Python cho phép người dùng có thể sử dụng, chỉnh sửa và phân phối mã nguồn một cách tự động.
- Di động: Mã Python có thể chạy trên nhiều nền tảng phần cứng có cùng giao diện.
- Có thể mở rộng: Người dùng có thể thêm các mô-đun cấp thấp vào trình thông dịch Python.
- Có thể cải tiến: Python cung cấp một cấu trúc cải tiến để hỗ trợ các chương trình lớn sau đó là shell-script.
- Hỗ trợ cộng đồng: python có một cộng đồng lớn với nhiều người dùng và nhà phát triển trên khắp thế giới đóng góp vào việc phát triển và cải tiến Python. Cộng đồng này cũng cung cấp hỗ trợ và tài liệu để giúp người dùng python giải quyết các vấn đề một cách nhanh chóng và hiệu quả.

3.1.3. Các phiên bản của Python

Python đã được Guido van Rossum tạo ra vào những năm 1980 tại Trung tâm Toán học – Tin học (Centrum Wiskunde & Informatica, CWI) ở Hà Lan như là một ngôn ngữ kế tục ngôn ngữ ABC – một ngôn ngữ được lấy cảm hứng từ SETL (ngôn ngữ được phát triển bởi Jacob T. Schwartz và các đồng nghiệp), có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba. Nó bắt đầu được triển khai vào tháng 12 năm 1989.

Python 2.0 được ra mắt vào ngày 16 tháng 10 năm 2000, với nhiều đặc trưng mới mẻ, bao gồm một bộ dọn rác phát hiện theo chu kỳ và khả năng hỗ trợ Unicode.

Python 3.0 được ra mắt vào ngày mùng 3 tháng 12 năm 2008. Đây là một phiên bản lớn của Python không tương thích ngược hoàn toàn. Nhiều đặc trưng lớn của nó đã được chuyển mã ngược (backport) về loạt phiên bản Python 2.6.x và 2.7.x. Các bản phát hành của Python 3 có đi kèm với công cụ 2to3, có tác dụng tự động hoá việc dịch mã Python 2 sang Python 3.

Python 3.9.2 và 3.8.8 được xúc tiến vì tất cả các phiên bản trước của Python (bao gồm cả 2.7) gặp một số vấn đề bảo mật, có thể dẫn đến thực thi mã từ xa và "đầu độc" bộ nhớ đệm.

Trong năm 2022, Python 3.10.4 và 3.9.12 được xúc tiến cùng với 3.8.13 và 3.7.13, nguyên nhân là do một vài vấn đề về bảo mật. Khi Python 3.9.13 được phát hành vào tháng Năm năm 2022, loạt phiên bản 3.9 (cùng với loạt 3.8 và 3.7) được thông báo rằng sẽ chỉ nhận được các bản vá bảo mật trong tương lai. Vào ngày 7 tháng Chín năm 2022, bốn bản cập nhật mới được phát hành do có khả năng xảy ra một cuộc tấn công từ chối dịch vụ: 3.10.7, 3.9.14, 3.8.14 và 3.7.14.

Tính đến tháng 10 năm 2023, Python 3.12 là bản phát hành ổn định mới nhất. Một số thay đổi đáng chú ý từ bản 3.11 bao gồm các thay đổi về ngôn ngữ và thư viện chuẩn.



Hình 3.1. Ngôn Ngữ Python

3.2. Visual Studio Code

3.2.1. Khái niệm

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn miễn phí, nhẹ và mạnh mẽ, phát triển bởi Microsoft, hoạt động trên đa nền tảng (Windows, macOS, Linux). Nó hỗ trợ nhiều ngôn ngữ lập trình và cung cấp các tính năng hữu ích như gỡ lỗi (debug), tô sáng cú pháp (syntax highlighting), tự động hoàn thành mã, và tích hợp Git. VS Code được ưa chuộng vì khả năng tùy biến cao và kho tiện ích mở rộng phong phú, phù hợp cho cả người mới bắt đầu và lập trình viên chuyên nghiệp. Nó là đa nền tảng, hoạt động trên Microsoft Windows, macOS và Linux.

3.2.2. Các đặc trưng

Visual Studio Code là gì được rất nhiều người tìm hiểu. Đây cũng là một trong các ứng dụng được dân IT “săn đón” và tải về và sử dụng rất nhiều. Visual Studio Code cũng luôn có những cải tiến và tạo ra đa dạng các tiện ích đi kèm từ đó giúp cho các lập trình viên sử dụng dễ dàng hơn. Trong đó có thể kể đến những ưu điểm sau:

1. Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++,...

2. Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.
3. Các tiện ích mở rộng rất đa dạng và phong phú.
4. Hỗ trợ đa nền tảng: Linux, Mac, Windows,..
5. Hỗ trợ đa ngôn ngữ: C/C++, C#, F#, JavaScript, JSON, Visual Basic,HTML,CSS,...
6. Ít dung lượng.
7. Tính năng mạnh mẽ.
8. Intellisense chuyên nghiệp.
9. Giao diện thân thiện.
10. Kiến trúc mạnh mẽ và người dùng có thể khai thác mở rộng.
11. Số lượng người sử dụng lớn tạo nên ộng đồng hỗ trợ rộng rãi.



Hình 3.2. Phần mềm VS Code

3.3. Google Collab

3.3.1. *Khái niệm*

Colaboratory hay còn gọi là Google Colab, là một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, bạn có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp cho bạn.

Colab cung cấp nhiều loại GPU, thường là Nvidia K80s, T4s, P4s and P100s, tuy nhiên người dùng không thể chọn loại GPU trong Colab, GPU trong Colab thay đổi theo thời gian. Vì là dịch vụ miễn phí, nên Colab sẽ có những thứ tự ưu tiên trong việc sử dụng tài nguyên hệ thống, cũng như giới hạn thời gian sử dụng, thời gian sử dụng tối đa lên tới 12 giờ.

3.3.2. *Những đặc trưng của Google Colab*

Google Colab (Google Collaboratory) là môi trường Jupyter Notebook trực tuyến do Google cung cấp, cho phép người dùng viết và chạy mã Python mà không cần cài đặt môi trường phức tạp trên máy tính cá nhân. Giao diện tương tự Jupyter Notebook truyền thống, hỗ trợ cả mã Python và nội dung Markdown giúp trình bày dễ dàng.

Colab hỗ trợ làm việc nhóm và chia sẻ thông qua Google Drive, cho phép nhiều người cùng chỉnh sửa, nhận xét và cộng tác trực tiếp trên cùng một notebook

Một ưu điểm nổi bật là khả năng sử dụng GPU và TPU miễn phí, giúp tăng tốc đáng kể quá trình huấn luyện mô hình học máy và deep learning. Ngoài ra, Colab còn tích hợp chặt chẽ với hệ sinh thái Google Cloud, cho phép truy cập, lưu trữ dữ liệu từ Drive hoặc sử dụng các dịch vụ như BigQuery, Cloud Storage, và các API khác trong quá trình làm việc.



Hình 3.3. Google Collab

3.4. Android studio

Android Studio là môi trường phát triển tích hợp (*Integrated Development Environment – IDE*) chính thức do Google phát hành, được sử dụng rộng rãi để phát triển các ứng dụng cho hệ điều hành Android. Android Studio được xây dựng dựa trên nền tảng IntelliJ IDEA – một IDE nổi tiếng của JetBrains – và được tối ưu hóa dành riêng cho việc phát triển ứng dụng di động Android.

3.4.1. Tổng quan về Android Studio

Android Studio được giới thiệu lần đầu tiên vào năm 2013 tại hội nghị Google I/O, thay thế cho công cụ Eclipse ADT (Android Development Tools) trước đây. Việc ra đời của Android Studio đánh dấu bước chuyển quan trọng trong phát triển ứng dụng Android, mang lại một môi trường làm việc chuyên nghiệp, hiện đại, có khả năng tự động hóa và tích hợp mạnh mẽ.

Công cụ này hỗ trợ lập trình viên trong toàn bộ quá trình phát triển phần mềm, từ thiết kế giao diện, viết mã nguồn, biên dịch, kiểm thử, đến triển khai ứng dụng lên thiết bị thật hoặc giả lập. Android Studio tương thích với các ngôn ngữ lập trình Java, Kotlin (ngôn ngữ chính thức của Google cho Android từ năm 2017) và C/C++ thông qua bộ công cụ Android NDK.

3.4.2. Cấu trúc và thành phần chính của Android Studio

Android Studio bao gồm nhiều thành phần quan trọng giúp tối ưu hóa quy trình phát triển ứng dụng:

a. Môi trường viết mã (Code Editor): Cung cấp khả năng hỗ trợ gợi ý cú pháp thông minh, tự động hoàn thiện mã, phân tích lỗi cú pháp theo thời gian thực và hỗ trợ refactor mã nguồn nhanh chóng. Trình soạn thảo có khả năng nhận diện cấu trúc code, hỗ trợ chuyển hướng dễ dàng giữa các lớp, hàm và tệp tin trong dự án.

b. Trình thiết kế giao diện (Layout Editor): Cho phép thiết kế giao diện ứng dụng một cách trực quan bằng thao tác kéo – thả (Drag and Drop) các thành phần UI như nút bấm, hộp văn bản, hình ảnh, danh sách,... Người dùng có thể xem trước giao diện trên nhiều loại màn hình, kích thước và độ phân giải khác nhau mà không cần chạy ứng dụng.

c. Hệ thống biên dịch Gradle (Gradle Build System): Là công cụ tự động hóa việc biên dịch, đóng gói và quản lý thư viện trong dự án. Gradle giúp lập trình viên dễ dàng khai báo các thư viện phụ thuộc, cấu hình phiên bản SDK, thiết lập bản debug hoặc release chỉ bằng vài dòng lệnh trong file build.gradle.

d. Trình giả lập Android (Android Emulator): Cho phép chạy thử ứng dụng trên các thiết bị ảo (Virtual Devices) mà không cần điện thoại thật. Emulator mô phỏng đầy đủ phần cứng và phần mềm của thiết bị Android, giúp kiểm thử giao diện, hiệu năng và hành vi ứng dụng ở nhiều điều kiện khác nhau.

e. Trình quản lý thiết bị ảo (AVD Manager): Cho phép người dùng tạo và cấu hình các thiết bị ảo với thông số phần cứng khác nhau như độ phân giải màn hình, dung lượng RAM, phiên bản hệ điều hành Android, hay cấu trúc CPU (ARM, x86).

f. Android Profiler: Cung cấp bộ công cụ theo dõi hiệu năng ứng dụng (CPU, bộ nhớ, luồng xử lý, mạng, năng lượng tiêu thụ...) trong thời gian thực, giúp lập trình viên phát hiện các vấn đề gây chậm hoặc tốn pin.

g. Tích hợp Firebase và Google Play Services: Android Studio hỗ trợ kết nối trực tiếp với nền tảng Firebase để thêm các chức năng như xác thực người dùng (Firebase Auth), lưu trữ dữ liệu đám mây (Cloud Firestore), thông báo đẩy (Push Notifications) và phân tích hành vi người dùng (Analytics).

3.4.3. Ưu điểm của Android Studio

Android Studio mang lại nhiều lợi thế nổi bật so với các IDE khác:

- Giao diện trực quan, dễ sử dụng và thân thiện với người mới học.
- Tự động cập nhật SDK, API và thư viện hỗ trợ của Google.
- Hệ thống giả lập mạnh mẽ, tiết kiệm thời gian kiểm thử.
- Hỗ trợ tích hợp Git, GitHub, SVN, giúp quản lý mã nguồn hiệu quả.
- Cho phép phát triển ứng dụng đa nền tảng (điện thoại, máy tính bảng, TV, Wear OS, Auto).
- Khả năng tối ưu hóa hiệu năng, giảm thời gian biên dịch và hỗ trợ phát triển ứng dụng Android hiện đại theo kiến trúc MVVM hoặc Jetpack Compose.

3.4.4. Ứng dụng thực tế trong phát triển phần mềm

Trong các đề tài, Android Studio thường được sử dụng để:

- Xây dựng ứng dụng di động phục vụ học tập, quản lý, giám sát, hoặc nhận dạng đối tượng.
- Tích hợp với mô hình trí tuệ nhân tạo (AI), đặc biệt là mô hình YOLO trong nhận diện hình ảnh.
- Kết nối với các mô hình học sâu (*Deep Learning*) được huấn luyện trên Python để tạo ra ứng dụng thông minh chạy trên điện thoại Android.

3.4.5. Kết luận

Android Studio là công cụ phát triển ứng dụng Android toàn diện, mạnh mẽ và linh hoạt nhất hiện nay. Với khả năng tích hợp đa ngôn ngữ, hỗ trợ kiểm thử tự

động, và khả năng tương thích với các nền tảng AI – IoT – Cloud, Android Studio đóng vai trò then chốt trong việc xây dựng các ứng dụng di động thông minh, đặc biệt trong bối cảnh chuyên đổi số và ứng dụng trí tuệ nhân tạo ngày càng phổ biến.

3.5. Giới thiệu về tập huấn luyện dữ liệu

Roboflow [8] là một framework hỗ trợ các nhà phát triển trong lĩnh vực thị giác máy tính (Computer Vision) thực hiện việc thu thập, xử lý và quản lý dữ liệu hình ảnh một cách hiệu quả. Công cụ này giúp người dùng tạo ra bộ dữ liệu được tổ chức khoa học, thuận tiện cho quá trình huấn luyện mô hình học máy (Machine Learning) và học sâu (Deep Learning [12]).

Roboflow [8] cung cấp sẵn nhiều tập dữ liệu công khai (public datasets) mà người dùng có thể sử dụng trực tiếp, đồng thời cho phép tải lên và tùy chỉnh dữ liệu riêng của mình. Người dùng có thể xử lý dữ liệu theo nhu cầu cá nhân như điều chỉnh kích thước ảnh, thay đổi định dạng, tăng cường dữ liệu (data augmentation) hoặc lọc bỏ nhiễu để nâng cao chất lượng tập dữ liệu đầu vào.

Quy trình làm việc trong Roboflow [8] được thiết kế khoa học và có thể phối hợp nhóm, giúp nhiều người cùng tham gia xây dựng bộ dữ liệu trong cùng một dự án.

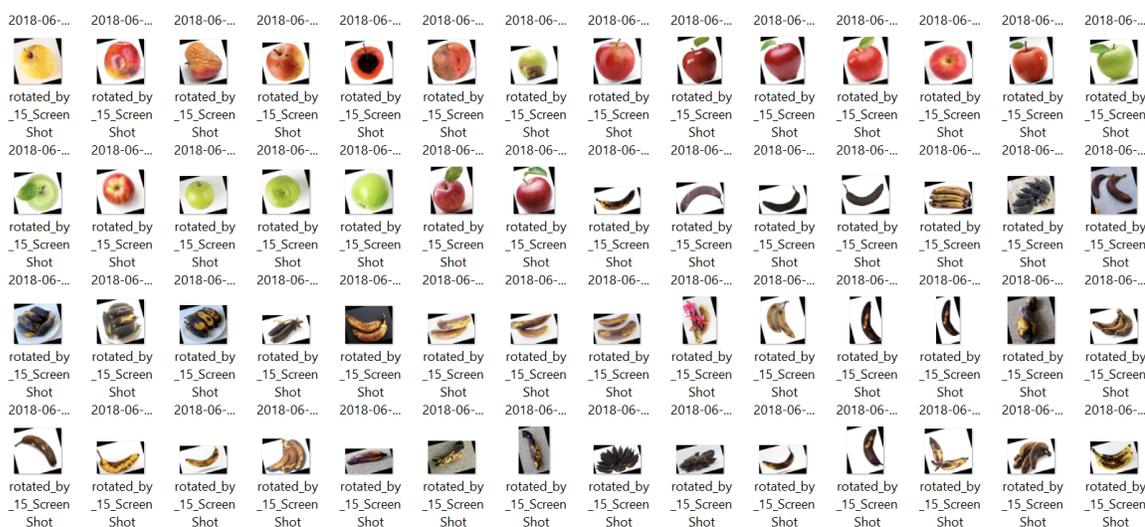
Sau khi hoàn thiện dữ liệu, Roboflow [8] hỗ trợ xuất dữ liệu ở nhiều định dạng chuẩn tương thích với các thư viện và framework phổ biến như TensorFlow, PyTorch, YOLO [9], EfficientDet, MobileNet, v.v.

Đối với quá trình huấn luyện mô hình nhận dạng đối tượng, Roboflow [8] đóng vai trò trung gian giữa khâu chuẩn bị dữ liệu và khâu huấn luyện, giúp người dùng tiết kiệm thời gian xử lý thủ công và dễ dàng tích hợp vào quy trình phát triển mô hình hiện đại.

Hiện nay, Roboflow [8] được ứng dụng rộng rãi trong nhiều lĩnh vực có liên quan đến thị giác máy tính [2], bao gồm:

- Nhận dạng vật thể trong ảnh hoặc video (object detection)
- Phát hiện cháy nổ, khói hoặc người xâm nhập bất hợp pháp
- Theo dõi phương tiện giao thông, quản lý bãi xe
- Dự đoán thiệt hại công trình, mái nhà hoặc cây trồng từ ảnh vệ tinh
- Hỗ trợ xe tự hành và robot trong việc nhận biết môi trường xung quanh

Với những ưu điểm đó, Roboflow [8] trở thành công cụ không thể thiếu trong việc xây dựng và quản lý bộ dữ liệu hình ảnh, góp phần nâng cao độ chính xác và hiệu quả của các mô hình học sâu.



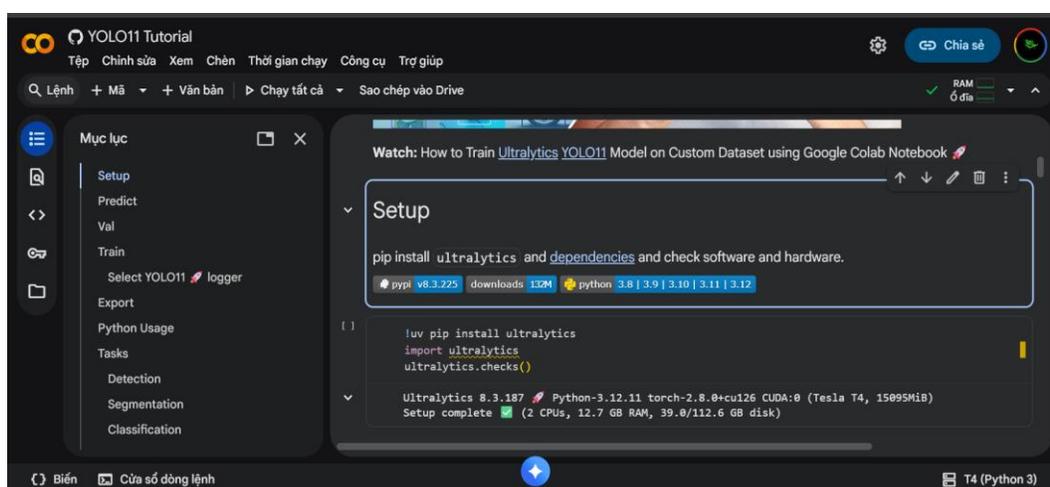
Hình 3.4. Tập ảnh dữ liệu huấn luyện

3.6. Đào tạo mô hình phát hiện loại trái cây

Để huấn luyện mô hình “ Phát hiện loại trái cây “ bằng phương pháp học sâu thì em sử dụng một công cụ hỗ trợ ở đây là Google Colab. Google Colab là một dịch vụ máy tính đám mây của Google cho phép người dùng tạo ra các tập note book Jupyter để thực thi mã Python. Nó cung cấp một môi trường tính toán đám mây miễn

phí, với các đặc trưng như GPU miễn phí, RAM miễn phí đến 12GB có thể mở rộng đến 25.5GB nếu có trả phí, lưu trữ đám mây và nhiều đặc trưng khác.

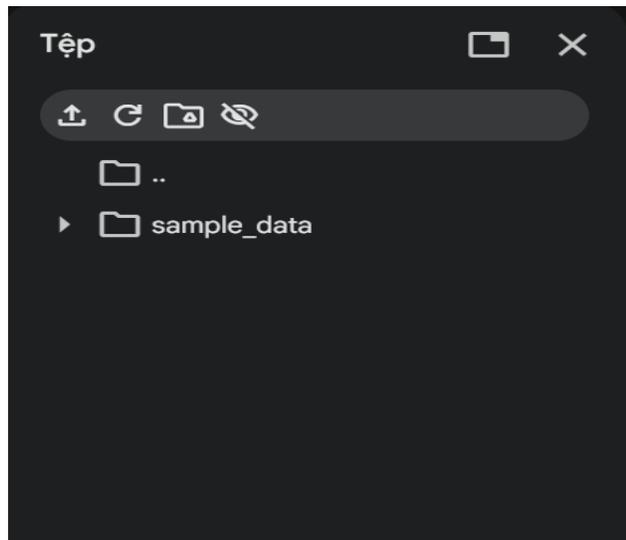
Ngoài ra Google Colab còn liên kết với driver để có thể hỗ trợ người dùng trong việc lưu trữ các dữ liệu quan trọng. Google Colab được thiết kế để hỗ trợ việc phát triển và huấn luyện các mô hình học máy và các ứng dụng AI. Nó cung cấp một môi trường tính toán đám mây miễn phí, mạnh mẽ và linh hoạt cho các nhu cầu học tập và nghiên cứu.



Hình 3.5. Giao diện Google Collab

Vì Google Colab bản miễn phí chỉ cho giới hạn về thời gian chạy nên ta có thể liên kết với drive để lưu trữ để tránh mất dữ liệu sau khi hết phiên.

Ta có thể dễ dàng kết nối với drive với 2 cách:



Hình 3.6. Ấn chọn biểu tượng drive trong phần tệp

```
from google.colab import drive
drive.mount('/content/drive')
```

Hình 3.7: kết nối qua đoạn mã

Sau khi cài xong một vài đoạn mã ta có thể bắt đầu train với đoạn mã sau

```
[ ] # Train YOLO11n on COCO8 for 3 epochs
!yolo train model=yolo11n.pt data=coco8.yaml epochs=3 imgsz=640
```

Hình 3.8. đoạn mã để bắt đầu train

Trong đó:

- !yolo train: Là lệnh huấn luyện mô hình YOLO được chạy trong Jupyter Notebook hoặc Google Colab (dấu ! cho phép chạy lệnh hệ thống).
- Model = yolo11n.pt: Là mô hình nền (pretrained model) do Ultralytics cung cấp, phiên bản YOLO11 Nano – nhỏ, nhẹ, huấn luyện nhanh.
- Data = coco8.yaml: Là tệp cấu hình dữ liệu chứa đường dẫn đến tập huấn luyện, tập kiểm thử và số lớp (classes) của đối tượng.

- Epochs = 3: Là số vòng lặp huấn luyện, nghĩa là toàn bộ tập dữ liệu sẽ được mô hình học qua 3 lần.

- Imgsz = 640: Là kích thước ảnh đầu vào (640×640 pixel). Với YOLO, kích thước này giúp mô hình đạt hiệu quả tốt nhất giữa tốc độ và độ chính xác.

Ta cũng có một số thông số khác nhưng ta cũng không cần phải ghi hết tất cả để có thể chạy được mọi thông tin ta có thể tìm trên trang train của ultralytics. Ultralytics có đầy đủ các thông số nếu ai có yêu cầu chi tiết trong quá trình đào tạo.

Bước tiếp theo là chuyển đổi từ file mô hình đã huấn luyện best.pt chạy trên desktop qua mô hình đã huấn luyện best.tflite để chạy trên android

Đầu tiên cài đặt thư viện Ultralytics

```
!pip install ultralytics

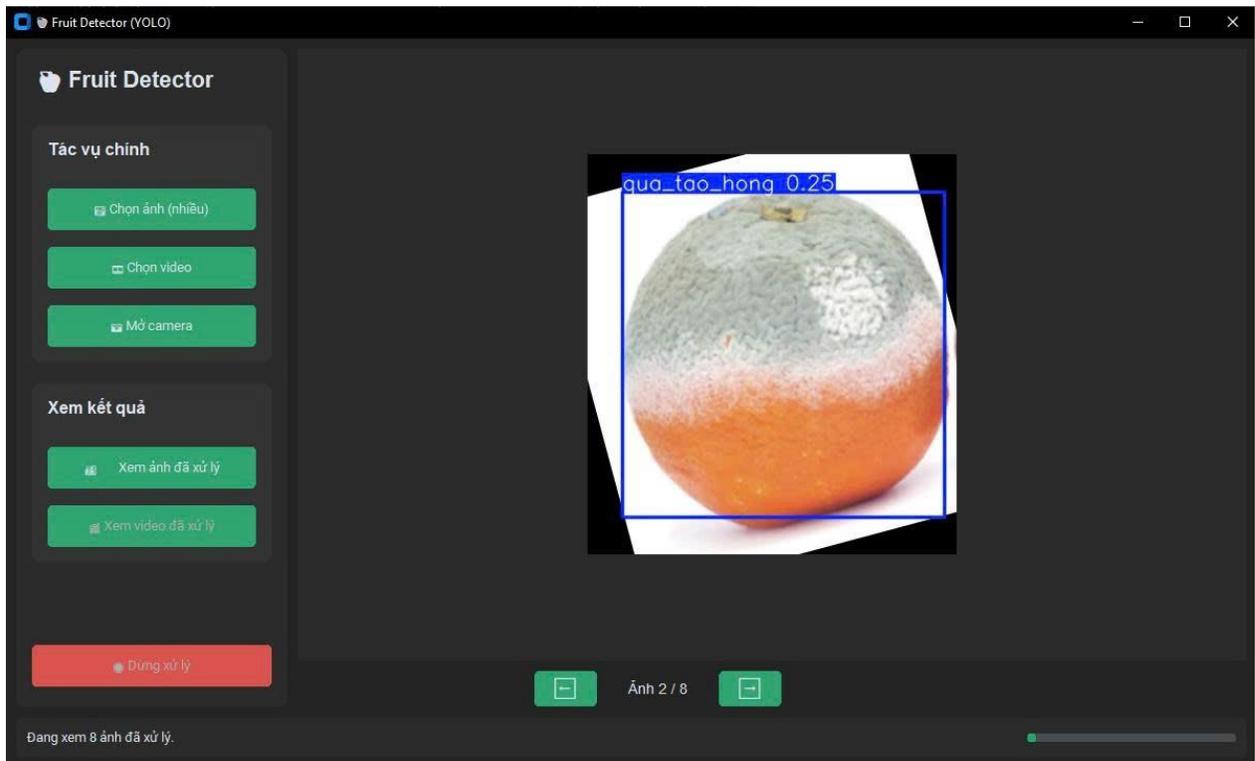
from ultralytics import YOLO

# Load mô hình PyTorch
model = YOLO("best.pt")

# Xuất sang định dạng TensorFlow Lite
model.export(format="tflite")
```

Hình 3.9. Code thư viện và chuyển đổi sang file tflite để chạy trên di động.

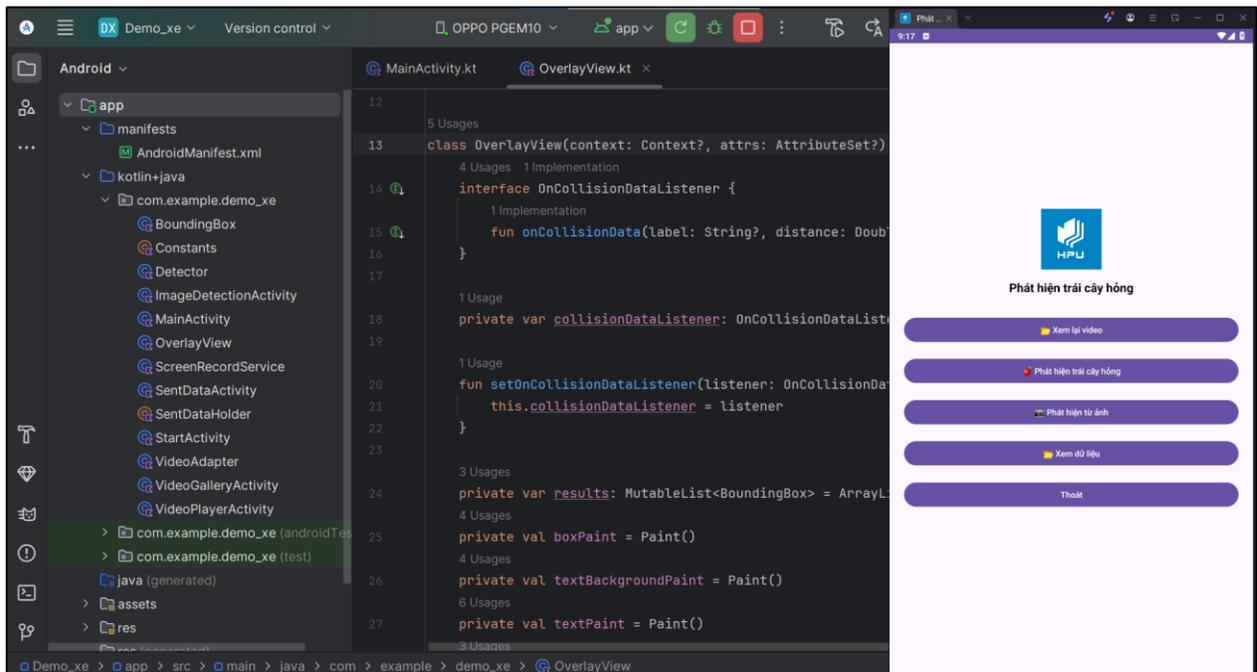
Bước tiếp theo chạy code chương trình phát hiện hoa quả hư hỏng trên desktop.



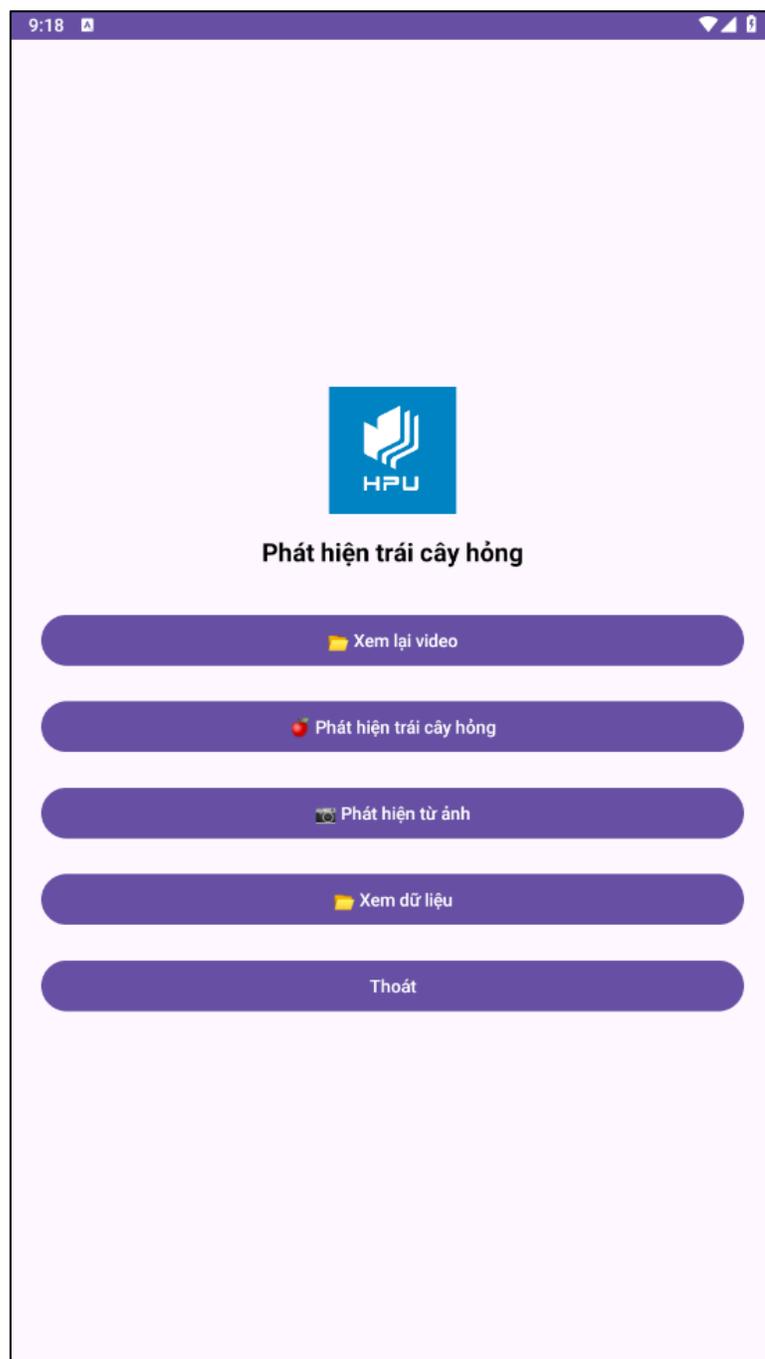
Hình 3.10. Giao diện chương trình phát hiện trên laptop.

- Chọn ảnh (nhiều): Dùng để chọn nhiều hình ảnh từ máy tính và đưa vào hệ thống để mô hình xử lý.
- Chọn video: Cho phép chọn một tệp video để mô hình thực hiện nhận diện trên từng khung hình.
- Mở camera: Bật camera để nhận diện trái cây trực tiếp theo thời gian thực.
- Xem ảnh đã xử lý: Mở thư mục hoặc giao diện hiển thị các ảnh đã được mô hình phân tích.
- Xem video đã xử lý: Xem lại video đã được mô hình nhận diện (trong ảnh nút này đang bị mờ, có thể chưa kích hoạt).
- Dừng xử lý: Ngừng quá trình xử lý ảnh hoặc video đang chạy.
- Nút chuyển ảnh sang trái: Chuyển sang xem ảnh trước đó trong danh sách ảnh đã xử lý.
- Nút chuyển ảnh sang phải: Chuyển sang xem ảnh tiếp theo.

Tiếp đến là chạy chương trình phát hiện trái cây hỏng đã xây dựng trên android



Hình 3.11. Giao diện chương trình phát triển trên android studio.

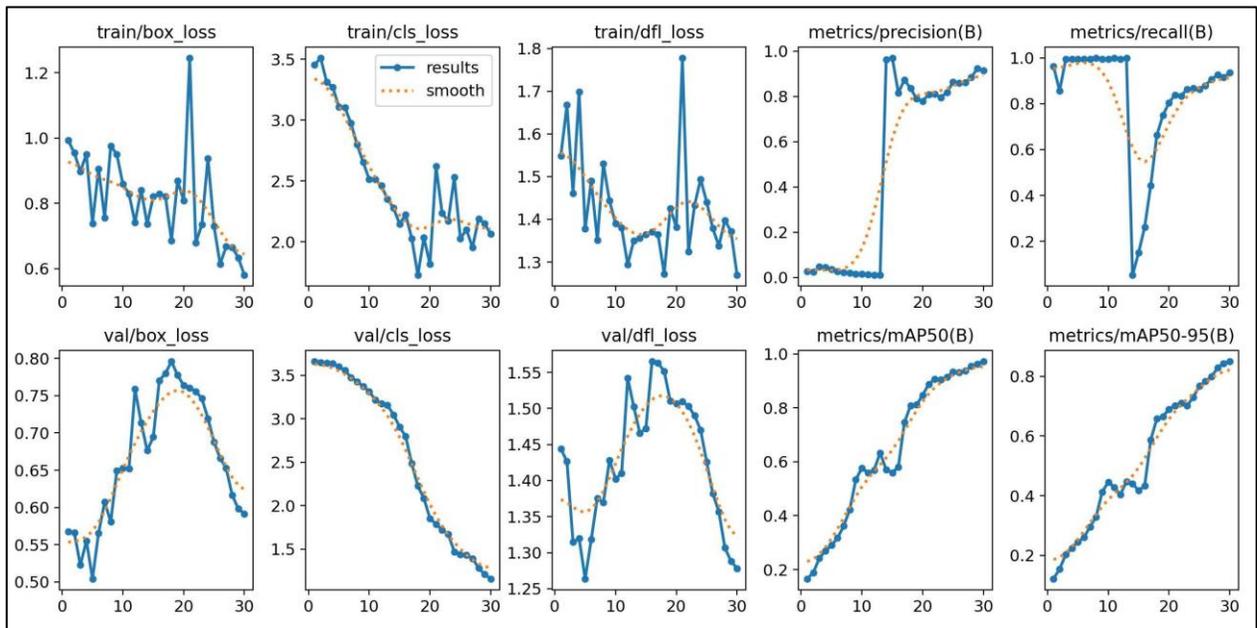


Hình 3.12. Giao diện chương trình phát triển trên android studio.

- Xem lại video: Mở danh sách hoặc thư mục các video đã được ứng dụng xử lý trước đó để xem lại.
- Phát hiện trái cây hỏng: Bắt đầu chế độ nhận diện trái cây hỏng theo thời gian thực (thường từ camera).

- Phát hiện từ ảnh: Cho phép chọn ảnh từ thiết bị để ứng dụng phân tích và phát hiện trái cây hỏng trong ảnh.
- Xem dữ liệu: Mở phân lưu trữ dữ liệu, bao gồm ảnh/video đã xử lý hoặc kết quả nhận diện được ghi lại.
- Thoát: Đóng ứng dụng.

3.7. Giá trị các biểu đồ



Hình 3.13. Biểu đồ thống kê sau thi đào tạo

Theo như hình ta có thể thấy:

- Các giá trị loss (train/box_loss, train/cls_loss, train/df_l_loss) đều có xu hướng giảm dần theo số epoch, thể hiện mô hình đang học được cách dự đoán chính xác hơn cả về vị trí và phân loại đối tượng.

- Loss trên tập validation (val/box_loss, val/cls_loss, val/df_l_loss) cũng giảm tương tự, chứng tỏ mô hình không bị overfitting đáng kể và khả năng tổng quát hóa tốt.

- Độ chính xác trung bình (mAP50 và mAP50–95) tăng đều theo thời gian huấn luyện, đạt giá trị cao ở giai đoạn cuối, phản ánh hiệu suất nhận dạng và định vị đối tượng được cải thiện rõ rệt.

3.7.1. Kết quả thử nghiệm mô hình đã huấn luyện trên desktop

Các chỉ số Precision và Recall đều đạt mức cao và ổn định sau khoảng 10–15 epoch, cho thấy mô hình phân biệt tốt giữa các đối tượng và phát hiện đầy đủ phần lớn các mẫu trong tập dữ liệu.

Tổng thể, mô hình YOLO sau huấn luyện đã đạt được hiệu năng tốt, độ chính xác cao và khả năng nhận dạng ổn định, đủ điều kiện để áp dụng vào giai đoạn kiểm thử và triển khai trong ứng dụng thực tế.

Thực hiện thử công trên 2 loại dữ liệu đầu vào: ảnh tĩnh 1 loại quả và ảnh tĩnh nhiều loại quả.

Đối với ảnh tĩnh 1 loại quả chương trình đem lại độ chính xác trong khoảng 85%

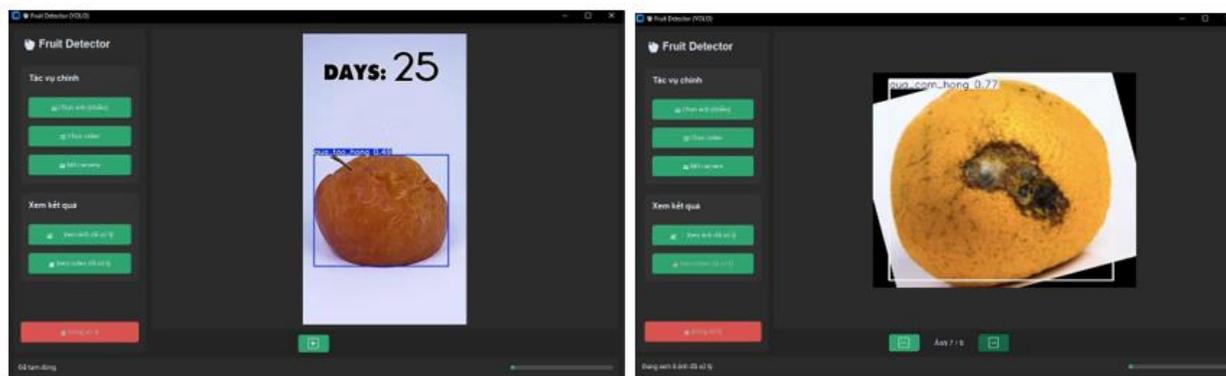
Bảng 3.1. Bảng thử nghiệm phân loại hình ảnh tĩnh 1 loại quả

STT	Tên quả	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Táo	50	41	9	82%
2	Cam	50	43	7	86%
3	Chuối	50	40	10	80%

Đối với ảnh tĩnh nhiều loại quả chương trình đem lại độ chính xác trong khoảng 82%

Bảng 3.2. Bảng thử nghiệm phân loại hình ảnh tĩnh nhiều loại quả

STT	Tên quả	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Táo	50	39	11	78%
2	Cam	25	20	5	80%
3	Chuối	25	22	3	88%



Hình 3.14. Ảnh đã phát hiện được trái cây hồng trên chương trình desktop

3.7.2. Kết quả thử nghiệm mô hình đã huấn luyện trên android

Các chỉ số Precision và Recall đều đạt mức cao và ổn định sau khoảng 10–15 epoch, cho thấy mô hình phân biệt tốt giữa các đối tượng và phát hiện đầy đủ phần lớn các mẫu trong tập dữ liệu.

Tổng thể, mô hình YOLO sau huấn luyện đã đạt được hiệu năng tốt, độ chính xác cao và khả năng nhận dạng ổn định, đủ điều kiện để áp dụng vào giai đoạn kiểm thử và triển khai trong ứng dụng thực tế.

Thực hiện thử công trên 2 loại dữ liệu đầu vào: ảnh tĩnh 1 loại quả và ảnh tĩnh nhiều loại quả.

Đối với ảnh tĩnh 1 loại quả chương trình đem lại độ chính xác trong khoảng 85%

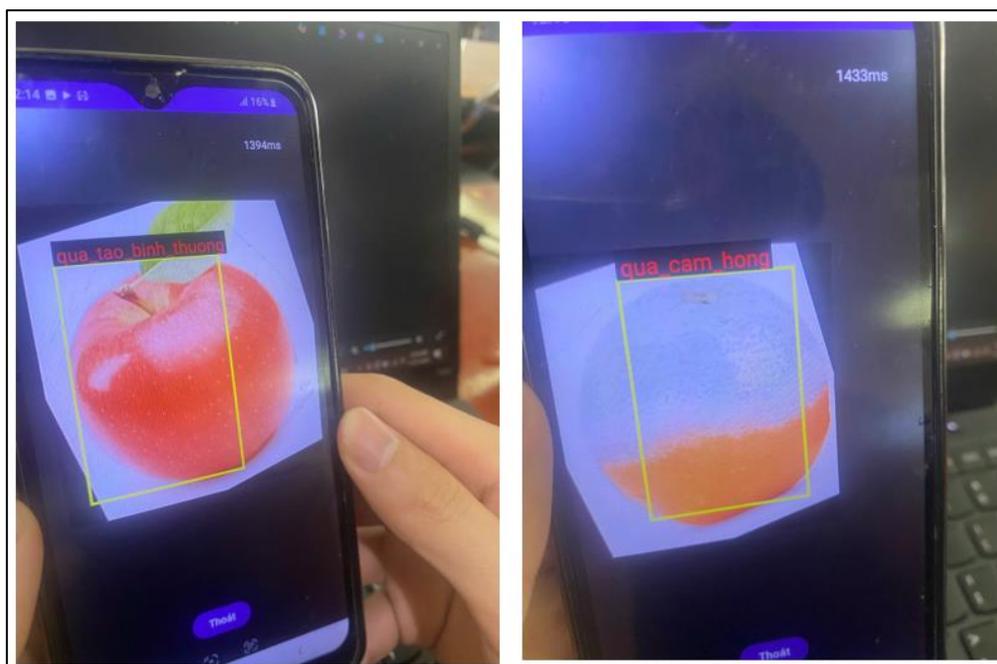
Bảng 3.3. Bảng thử nghiệm phân loại hình ảnh tĩnh 1 loại quả

STT	Tên quả	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Táo	50	38	12	76%
2	Cam	50	40	10	80%
3	Chuối	50	42	8	84%

Đối với ảnh tĩnh nhiều loại quả chương trình đem lại độ chính xác trong khoảng 82%

Bảng 3.4. Bảng thử nghiệm phân loại hình ảnh tĩnh nhiều loại quả

STT	Tên quả	Số lượng ảnh	Nhận đúng	Nhận sai	Độ chính xác
1	Táo	50	40	11	80%
2	Cam	25	18	5	72%
3	Chuối	25	18	3	72%



Hình 3.15. Ảnh đã phát hiện được trái cây hồng trên ứng dụng di động

Kết quả thử nghiệm cho thấy mô hình phân loại trái cây hoạt động khá tốt trên tập dữ liệu ảnh tĩnh, đặc biệt là đối với từng loại quả riêng lẻ. Ở bài toán phân loại ảnh tĩnh 1 loại quả, độ chính xác trung bình đạt khoảng 85%, trong đó cao nhất là quả cam (86%) và thấp nhất là quả chuối (80%). Điều này cho thấy mô hình có khả năng nhận diện tốt khi đối tượng rõ ràng, ít bị che khuất và không có nhiều loại trái cây xuất hiện cùng lúc trong ảnh.

Đối với bài toán phân loại ảnh tĩnh chứa nhiều loại quả, độ chính xác trung bình giảm nhẹ xuống còn 82%. Nguyên nhân là do ảnh có nhiều đối tượng dẫn đến hiện tượng che khuất, nhiều nền hoặc kích thước quả nhỏ hơn, khiến mô hình khó nhận dạng chính xác hơn. Tuy vậy, kết quả vẫn được đánh giá là khả quan, trong đó quả chuối đạt độ chính xác cao nhất (88%), tiếp theo là cam (80%) và cuối cùng là táo (78%).

Tổng quan lại, mô hình huấn luyện đã đạt hiệu suất ở mức tốt, đáp ứng yêu cầu của bài toán. Độ chính xác chỉ giảm nhẹ khi số lượng đối tượng trong ảnh tăng lên, cho thấy mô hình vẫn có khả năng tổng quát hóa trên các trường hợp phức tạp hơn. Đây là cơ sở để tiếp tục cải thiện mô hình trong các phiên bản sau bằng cách mở rộng tập dữ liệu, tăng độ đa dạng mẫu và tối ưu hóa các siêu tham số huấn luyện.

3.8. Đánh giá mô hình sau khi thử nghiệm

Mô hình YOLO11 đã được sử dụng để phân loại trái cây trên Google Colab, với bộ dữ liệu gồm khoảng 250 hình ảnh của 3 loại quả khác nhau. Mô hình đã được huấn luyện trong 50 epochs. Bài báo cáo này sẽ đánh giá các kết quả đạt được, cũng như nêu rõ các ưu điểm và nhược điểm của mô hình.

3.8.1. Kết quả huấn luyện

Độ chính xác:

Dựa trên quá trình huấn luyện 50 epochs, mô hình đạt hiệu năng tốt và ổn định

Ở khoảng epoch thứ 40 trở đi, các chỉ số đánh giá đạt giá trị cao và gần như không còn biến động lớn.

Kiểm tra thủ công trên bảng 3.5:

Khi tiến hành kiểm tra mô hình bằng các hình ảnh thực tế bên ngoài tập huấn luyện, mô hình cho kết quả nhận diện chính xác cao và ổn định.

Bộ ảnh kiểm tra có chất lượng tốt (rõ nét, đủ sáng) cho kết quả nhận diện rất chính xác, trong khi các ảnh chụp từ webcam hoặc điều kiện ánh sáng kém sẽ làm độ chính xác giảm nhẹ.

Điều này cho thấy chất lượng ảnh đầu vào có ảnh hưởng trực tiếp đến hiệu quả của mô hình, tuy nhiên mô hình vẫn chứng tỏ khả năng tổng quát hóa tốt trên dữ liệu thực tế.

3.8.2. Ưu điểm của mô hình:

- Hiệu suất và độ chính xác: YOLO11 đã cho thấy hiệu suất khá tốt với độ chính xác cao đối với các hình ảnh tĩnh chứa một loại quả.

- Khả năng xử lý: Hệ thống xử lý nhanh và chính xác đối với các hình ảnh tĩnh và những video có 1 loại hàng hóa.

3.8.3. Nhược điểm của mô hình:

- Độ chính xác với hình ảnh phức tạp: Với các hình ảnh chứa nhiều loại trái cây khác nhau, độ chính xác chỉ đạt mức trung bình.

- Sai sót trong nhận diện: Mô hình còn gặp khó khăn khi nhận diện các loại trái cây có màu sắc và hình dáng tương tự nhau, hoặc bị che khuất một phần.

3.8.4. Đề xuất cải tiến:

- Thêm dữ liệu đào tạo: Bổ sung thêm dữ liệu đa dạng để cải thiện khả năng nhận diện.

- Nâng cấp mô hình: Sử dụng các phiên bản cao cấp hơn của YOLO như YOLOv12 mới ra mắt vào tháng 2/2025 để tăng độ chính xác.

- Tối ưu hóa tài nguyên: Sử dụng các dịch vụ đám mây khác hoặc phiên bản trả phí của Google Colab để có tài nguyên huấn luyện mạnh mẽ hơn.

- Mô hình YOLO11 đã phần nào đáp ứng được các yêu cầu của đề tài, đặc biệt là đối với các hình ảnh tĩnh. Tuy nhiên, vẫn cần cải thiện thêm để tăng độ chính xác và giảm sai sót trong quá trình nhận diện trái cây. Việc nâng cấp mô hình và bổ sung dữ liệu sẽ là các bước quan trọng để đạt được kết quả tốt hơn trong tương lai.

KẾT LUẬN

Sau thời gian gần 5 tháng nghiên cứu và thực hiện đề án tốt nghiệp với sự hướng dẫn của cô TS. Hồ Thị Hương Thơm – Khoa Công Nghệ Thông Tin trường Đại Học Hàng Hải Việt Nam để hoàn thiện được đề án với đề tài: Phát hiện loại trái cây

Một số mục tiêu mà đề án đã đạt được: Hoàn thành đề tài: ”Phát hiện loại trái cây”. Đào tạo thành công mô hình nhận diện với 3 loại trái cây khác nhau: cam, táo, chuối bằng mô hình YOLOv8

Mô hình phân loại được các loại trái cây với độ chính xác trung bình 75% với 100 epochs nhưng lại chính xác khá cao với ảnh tĩnh với một loại quả là 95% và vẫn còn nhiều sai sót với một số loại quả. Tuy nhiên khi đưa vào thực tiễn có thể đáp ứng được phần nào yêu cầu đặt ra nhưng có thể sẽ không được chính xác. Muốn cải thiện độ chính xác thì ảnh trước khi đưa vào phân loại cần phải được xử lý nâng cao chất lượng ảnh, hoặc ảnh được chụp/ quay từ các thiết bị chuyên dụng có chất lượng cao

Tácó thể phát triển chương trình này thành một mô-đun tiền xử lý hình ảnh cho các loại trái cây và rau củ, để tích hợp vào hệ thống robot phân loại tự động. Hệ thống này sẽ thực hiện bước tiền xử lý trước khi sản phẩm được đưa lên băng chuyền đóng gói hoặc nhận dạng tính tiền tự động trong siêu thị. Việc tự động hóa này sẽ góp phần hiện đại hóa ngành nông nghiệp và công nghiệp đóng gói, hướng tới quy trình sản xuất và đóng gói theo dây chuyền. Điều này sẽ hỗ trợ hiệu quả cho nền sản xuất trái cây và rau củ tại Việt Nam, một đất nước nổi tiếng với nhiều loại trái cây ngon xuất khẩu

Trong quá trình học tập và nghiên cứu, tìm hiểu một lĩnh vực rất mới và nhiều thử thách đối với bản thân sinh viên cho nên đề án của sinh viên sẽ còn có những thiếu sót và hạn chế. Sinh viên rất cầu thị quý thầy cô trong khoa Công nghệ thông tin và Viện đào tạo sau đại học có cái nhìn cảm thông và chia sẻ đóng góp những lời

nhận xét để sinh viên nâng cao sự hiểu biết về lĩnh vực AI vốn là lĩnh vực mới và hoàn thiện hơn đề án của mình. Sinh viên xin trân trọng cảm ơn.

TÀI LIỆU THAM KHẢO

A. Tài liệu tiếng Việt

1. Nguyễn Văn Vy, *Giáo trình Trí tuệ nhân tạo*. Hà Nội: NXB Khoa học và Kỹ thuật, 2023.
 2. Phạm Văn Hùng, *Thị giác máy tính và ứng dụng*. Hà Nội: NXB Bách Khoa, 2022.
 3. Ultralytics, “Tài liệu hướng dẫn sử dụng YOLOv8–v11,” *Trang chủ Ultralytics YOLO*, 2024. Có tại:
<https://docs.ultralytics.com/vi/models/yolo11/>
 4. Nguyễn Hữu Tùng, “Ứng dụng học sâu trong nhận diện khuôn mặt,” *Tạp chí Công nghệ Thông tin & Truyền thông*, 2024.
 5. Python Software Foundation, “Python Documentation,” 2024. Có tại:
<https://docs.python.org/3/>
 6. Google Research, “Hướng dẫn sử dụng Google Colab,” *Google Research Documentation*, 2024. Có tại: <https://colab.research.google.com/>
 7. Microsoft, “Visual Studio Code Documentation,” 2024. Có tại:
<https://code.visualstudio.com/docs/>
-

B. Tài liệu tiếng Anh

8. Ultralytics, “YOLO11 Model Release and Benchmark Report,” 2025. Có tại: <https://github.com/ultralytics/YOLO11>
9. Roboflow, “Computer Vision Dataset Management,” 2024. Có tại:
<https://roboflow.com/>
10. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. Có tại :
<https://www.deeplearningbook.org/>