

**VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY  
UNIVERSITY OF INFORMATION TECHNOLOGY  
FACULTY OF COMPUTER ENGINEERING**

**---oOo---**



**Thiết kế hệ thống nhúng  
CE224.P12**

**ĐỒ ÁN MÔN HỌC: XỬ LÝ ẢNH NHẬN BIẾT LỖI  
HƯ HỎNG TRÊN CONTAINER**

**STUDENT NAME:**

Nguyễn Quang Dũng – 22520286

Trần Trí Dũng – 22520293

Chu Thành Hiếu – 22520429

**Lecturer:** Nguyễn Duy Xuân Bách

# Mục lục

<b>1. Giới thiệu</b>	3
<b>2. Thực trạng và ứng dụng</b>	3
2.1. Thực trạng	3
2.2. Ứng dụng liên quan	4
<b>3. Phân tích ưu điểm và hạn chế</b>	6
3.1. Ưu điểm	6
3.2. Hạn chế	6
<b>4. Hướng phát triển</b>	7
4.1. Tổng quan về mục tiêu ban đầu	7
4.2. Thử nghiệm và kết quả, giới hạn đề tài	7
4.3. Chi phí thực hiện	9
<b>5. Lịch trình dự án và phân công nhiệm vụ</b>	10
5.1. Ước tính thời gian thực hiện	10
5.2. Các mốc hoàn thành task	10
5.3. Phân công thành viên	11
5.4. Quản lý và theo dõi tiến độ	12
<b>6. Sản phẩm</b>	12
6.1. Giải pháp xử lý ảnh	12
6.1.1. Mô hình Yolo v10	12
6.1.2. Xây dựng và kết quả huấn luyện mô hình	13
6.2. Thiết kế hệ thống phần mềm và phần cứng	19
6.3. Đầu ra của sản phẩm	20
<b>7. Kết luận</b>	21

## 1. Giới thiệu

Báo cáo này trình bày những tìm hiểu về nghiên cứu và phát triển hệ thống xử lý ảnh để nhận biết lỗi hư hỏng của container của nhóm. Đồ án này kết hợp giữa nhúng, xử lý ảnh và trí tuệ nhân tạo nhằm tạo ra một giải pháp tự động, hiệu quả cho việc kiểm tra lỗi trên container.

## 2. Thực trạng và ứng dụng

### 2.1. Thực trạng

Container là sản phẩm không thể thiếu đối với ngành công nghiệp vận tải, hàng hóa hiện nay, giúp cho việc vận chuyển, bảo quản hàng hóa được an toàn và tiện lợi, đẩy mạnh sự phát triển thương mại toàn cầu.

Bối cảnh trên thế giới:

- Ngày nay, khoảng 90% hàng hóa lưu thông và vận chuyển được đóng trong các container và xếp lên các tàu chuyên trở thành từng cụm. Hơn 200 triệu container được chuyên chở hàng năm.

Bối cảnh trong nước:

- Ngành công nghiệp logistics đóng vai trò quan trọng đối với nền kinh tế Việt Nam. Theo Hiệp hội Chủ tàu Việt Nam, hiện cả nước có 10 công ty vận tải container, sở hữu 48 tàu với tổng sức chứa 39.519 TEUs và tổng trọng tải 548.236 DWT. Tại các khu vực cảng, số lượng container tồn đọng khá lớn: TPHCM có 5.800 container, Hải Phòng có 1.500, Đà Nẵng có 186 và Vũng Tàu có 120 container. Tuy nhiên, số lượng container lớn cũng mang lại nhiều thách thức trong công tác bảo dưỡng và kiểm tra.
- Ngoài việc tuân thủ tiêu chuẩn ISO và tiêu chuẩn BIC, Chính phủ Việt Nam đã ban hành các quy định về kiểm tra container trước khi sử dụng. Theo quy định, container phải được kiểm tra định kỳ để đảm bảo an toàn và chất lượng. Các tiêu chuẩn kiểm tra bao gồm: kiểm tra kết cấu, thiết bị nâng hạ, khóa và phụ kiện, vỏ container, cùng các thiết bị liên quan khác.

Hiện nay, việc kiểm tra và phát hiện lỗi hư hỏng trên container đang được thực hiện bằng cách kết hợp giữa kiểm tra thủ công và hệ thống tự động. Tuy nhiên, do giới hạn về hiệu suất và khả năng của con người, các sai sót trong quá trình kiểm tra và đánh giá là khó tránh khỏi, dẫn đến hậu quả nghiêm trọng khi hàng hóa, lương thực và nhu yếu phẩm bị ảnh hưởng bởi container không đạt chuẩn. Vì vậy, xu hướng đang dần chuyển sang sử dụng các hệ thống kiểm tra tự động hoàn toàn, dựa trên công nghệ xử lý ảnh và trí tuệ nhân tạo để nâng cao độ chính xác, hiệu quả cũng như năng suất công việc.

## 2.2 Ứng dụng liên quan

a) Kiểm tra chất lượng trong sản xuất: Hệ thống xử lý ảnh được sử dụng để phát hiện các khiếm khuyết trên bề mặt sản phẩm trong quá trình sản xuất.

([AOI là gì? Nguyên lý và tầm quan trọng của Aoi trong sản xuất công nghệ cao - LedHD](#))

([Image-Based Surface Defect Detection Using Deep Learning: A Review | J. Comput. Inf. Sci. Eng. | ASME Digital Collection](#))

### Ưu điểm:

- Tiết kiệm thời gian và công sức của công nhân.
- Xác định lỗi một cách nhanh chóng và chính xác, từ đó cải thiện chất lượng và độ tin cậy của sản phẩm

### Nhược điểm:

- Nếu sản phẩm có hình thù phức tạp gây khó khăn cho việc thu thập dữ liệu vì phải lấy nhiều chi tiết và góc của sản phẩm
- Điều kiện ánh sáng ảnh hưởng tới việc chuẩn đoán hình ảnh chính xác

b) Kiểm tra an ninh tại cảng biển: Sử dụng công nghệ này để kiểm tra nhanh chóng và chính xác các container, phát hiện các dấu hiệu bất thường hoặc hư hỏng.

([EconStor: A first step towards automated image-based container inspections](#))

([Kiểm tra container theo C-TPAT \(itvc-global.com\)](#))

### Ưu điểm:

- Tốc độ và hiệu quả: Giúp kiểm tra nhanh chóng một lượng lớn container, giảm thời gian so với việc kiểm tra thủ công.
- Độ chính xác cao: Phân tích hình ảnh tự động giúp phát hiện các dấu hiệu bất thường hoặc hư hỏng mà mắt thường có thể bỏ sót.
- Giảm phụ thuộc vào lao động con người, từ đó giảm chi phí và rủi ro liên quan đến con người.

### Nhược điểm:

- Chi phí triển khai ban đầu cao: Đầu tư vào công nghệ và thiết bị quét hình ảnh có thể tốn kém.
- Hệ thống hình ảnh cần được bảo trì thường xuyên để đảm bảo hoạt động tốt.

- Hạn chế về điều kiện ánh sáng hoặc môi trường: Trong điều kiện ánh sáng yếu hoặc thời tiết khắc nghiệt, hệ thống có thể gặp khó khăn trong việc chụp và phân tích hình ảnh chính xác.

c) Bảo trì dự đoán: Phát hiện sớm các dấu hiệu hư hỏng để lên kế hoạch bảo trì, tránh các sự cố nghiêm trọng.

( [Assets Predictive Maintenance Using Convolutional Neural Networks | IEEE Conference Publication | IEEE Xplore](#))

([Bảo trì dự đoán \(PdM\) là gì? Kỹ thuật và cách thực hiện hiệu quả nhất \(congnghepviet.com.vn\)](#))

#### **Ưu điểm:**

- Phát hiện sớm các sự cố: Bằng cách phân tích dữ liệu và hình ảnh, hệ thống có thể dự đoán và phát hiện hư hỏng trước khi nó trở thành vấn đề nghiêm trọng.
- Giảm chi phí bảo trì: Bảo trì chỉ được thực hiện khi cần thiết, thay vì theo lịch cố định, giúp tiết kiệm chi phí và tối ưu hóa thời gian hoạt động của thiết bị.
- Tăng tuổi thọ thiết bị: Bằng cách phát hiện và xử lý sớm, thiết bị có thể được bảo trì đúng thời điểm, kéo dài tuổi thọ.

#### **Nhược điểm:**

- Đòi hỏi nhiều dữ liệu: Hệ thống yêu cầu lượng lớn dữ liệu để có thể phân tích và dự đoán chính xác.
- Chi phí cài đặt ban đầu: Cần đầu tư vào hệ thống cảm biến và các giải pháp AI để thu thập và phân tích dữ liệu.
- Khả năng dự đoán không phải lúc nào cũng hoàn hảo: Trong một số trường hợp, hệ thống có thể tạo ra kết quả dự đoán không chính xác nếu dữ liệu không đầy đủ hoặc không đại diện.

d) Kiểm tra chất lượng trong ngành thực phẩm: Phát hiện các sản phẩm không đạt tiêu chuẩn hoặc bị hư hỏng trong quá trình đóng gói.( [Computer vision technology for food quality assurance - ScienceDirect](#))

#### **Ưu điểm:**

- Phát hiện sai sót chính xác: Hệ thống có thể phát hiện những lỗi mà mắt người có thể bỏ sót, giúp đảm bảo chất lượng sản phẩm.
- Tốc độ kiểm tra nhanh: Tự động hóa quá trình kiểm tra giúp đẩy nhanh tốc độ kiểm tra chất lượng sản phẩm, giảm thời gian sản xuất.

- Tính nhất quán: Công nghệ máy tính đảm bảo rằng tất cả các sản phẩm đều được kiểm tra với cùng một tiêu chuẩn và độ chính xác cao.

#### **Nhược điểm:**

- Chi phí đầu tư cao: Triển khai công nghệ thị giác máy tính có thể tốn kém về chi phí thiết bị và công nghệ.
- Yêu cầu bảo trì thường xuyên: Hệ thống kiểm tra cần được bảo dưỡng và cập nhật thường xuyên để hoạt động hiệu quả.
- Khả năng bị hạn chế trong một số môi trường: Hệ thống có thể gặp khó khăn khi hoạt động trong các điều kiện ánh sáng kém hoặc môi trường không ổn định.

### **3. Phân tích ưu điểm và hạn chế**

#### **3.1. Ưu điểm**

- Tăng tốc độ kiểm tra: Hệ thống tự động có thể xử lý nhiều container (tập trung xử lý dữ liệu hình ảnh với độ phân giải lớn, xử lý dữ liệu đồng thời với nhiều đầu vào được ghi hình từ nhiều cảm biến camera) trong thời gian ngắn (thời gian thực thì một vòng lặp xử lý ảnh ngắn, đạt hiệu suất real time).
- Độ chính xác cao: Giảm thiểu sai sót do con người gây ra.
- Hoạt động 24/7: Không bị giới hạn bởi thời gian làm việc của con người.
- Tiết kiệm chi phí dài hạn: Giảm chi phí nhân công và tăng hiệu quả.
- Khả năng phát hiện lỗi nhỏ: Có thể phát hiện các lỗi mà mắt thường khó nhận biết, đồng thời hạn chế những sai sót chủ quan thường xảy ra đối với việc kiểm tra từ con người.

#### **3.2. Hạn chế**

- Chi phí đầu tư ban đầu cao: Cần đầu tư vào hệ thống camera, phần cứng xử lý (xử lý trực tiếp trên bo mạch và hệ thống server quản lý dữ liệu thô và dữ liệu đã qua xử lý), và phần mềm AI.
- Yêu cầu dữ liệu huấn luyện lớn: Cần có bộ dữ liệu đa dạng và phong phú để huấn luyện mô hình AI.
- Khó khăn trong việc xử lý các trường hợp đặc biệt: Một số loại hư hỏng phức tạp có thể gây khó khăn cho hệ thống.
- Cần bảo trì và cập nhật thường xuyên: Đảm bảo hệ thống luôn hoạt động chính xác và cập nhật với các loại lỗi mới.
- Duy trì trạng thái kết nối mạng wifi nếu như có sử dụng thiết bị kết nối thông qua wifi

## 4. Hướng phát triển

### 4.1. Tổng quan về mục tiêu ban đầu

Phát triển 1 mô hình nhận diện lỗi của container, phát triển hệ thống tự động hóa bằng cách nhúng mô hình xử lý phân loại ảnh đã được huấn luyện trên máy tính xuống bo mạch để thực hiện nhận diện, từ đó phát hiện ra lỗi.

Sau khi đã phát hiện và thống kê được lỗi của container, tiến hành xử lý dữ liệu đã thu thập được, đưa ra thống kê thiệt hại đã được chuyển đổi từ con số thực tế sang đơn vị tiền tệ và xuất thông báo cho người dùng.

Hệ thống sẽ được áp dụng vào một dây chuyền xử lý quy củ trong bối cảnh thực tế, trong đó bao gồm các bước sau:

- Xe đầu máy chở container vào kho, đi ngang qua và dừng lại tại khu vực để đánh giá tình trạng của container
- Cảm biến một cảm biến được sử dụng để nhận diện sự có mặt của container, cũng như đảm bảo container đang ở đúng vị trí phù hợp để thực hiện đánh giá.
- Sau khi xe và container đã ổn định trong khu vực cần thiết và phù hợp (có thể sử dụng hàm delay để căng chỉnh thời gian chờ cho xe và container ổn định), tiến hành thu lại hình ảnh thông qua hệ thống camera, xử lý dữ liệu.
- Lưu lại dữ liệu đánh giá (thời gian, hình ảnh thô, hình ảnh đã gán chỉ định lỗi bởi hệ thống, thống kê lỗi, thống kê chi phí sửa chữa).

### 4.2. Thử nghiệm và kết quả, giới hạn đề tài

Bài toán nhận diện lỗi hư hỏng trên container được cả nhóm đánh giá là tương đối phức tạp, phụ thuộc nhiều vào góc độ của dữ liệu hình ảnh thu thập được. Điều kiện môi trường và ngoại cảnh trong quá trình thực hiện thu thập dữ liệu cũng bị ảnh hưởng đáng kể đến hiệu suất mô hình.

Bên cạnh đó, các mức độ nghiêm trọng đối với lỗi hư hỏng trên container khác nhau không chỉ phụ thuộc vào số lượng, phân lớp lỗi, mà còn mang cả yếu tố kích thước, góc nhìn tương đối từ dữ liệu thu thập được.

Đối với phương pháp sử dụng bộ lọc (filter):

- Sử dụng bộ lọc để quét bề mặt của container, sau đó thông qua các bộ lọc về màu sắc, độ sáng, làm nổi bật lên chi tiết hư hỏng của container, sau đó dùng một mô hình máy học đơn giản để phân loại (SVM, KNN, ...).

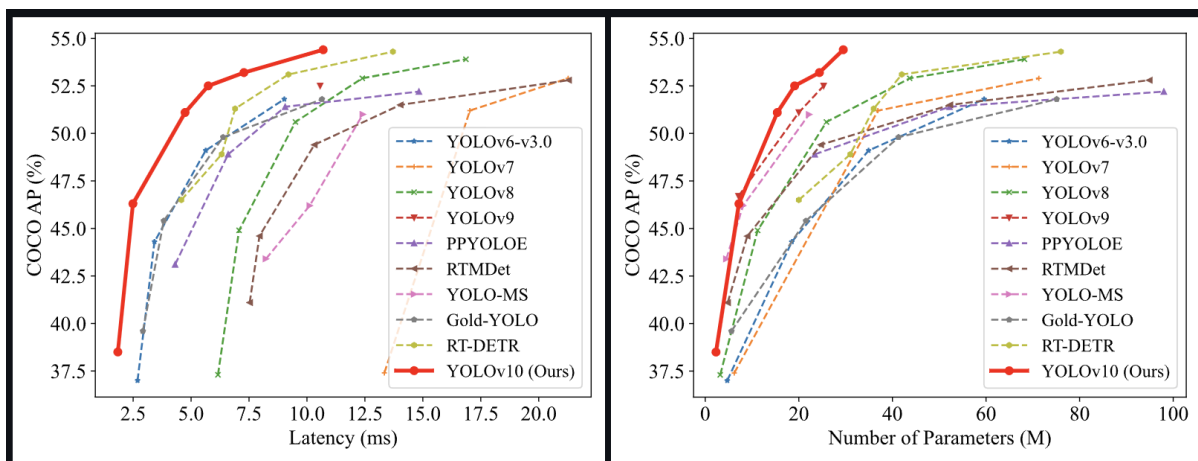
- Tuy nhiên, để hiện thực hóa bài toán thông qua phương pháp này, cần phải cố định container hay cố định góc lấy ảnh và đảm bảo điều kiện hiện thực hiện thu thập hình ảnh phải tối ưu nhất để làm nổi bật các chi tiết hồng học của container.

Đối với phương pháp sử dụng bo mạch esp32 và chạy trực tiếp mô hình FOMO:

- Việc chạy trực tiếp mô hình nhận diện trên bo mạch esp32 hứa hẹn nhiều tiềm năng, nhất là khi esp32 là bo mạch có hỗ trợ kết nối wifi giúp truyền tải dữ liệu.
- Tuy nhiên, mô hình FOMO (9216 tham số) dù được xây dựng để có thể chạy trên những bo mạch đơn giản với tốc độ xử lý chậm, thể hiện hiệu suất tương đối kém trên bo mạch esp32 cam với thời gian thực thi lên đến 5369 ms cho khâu phân loại (classification), đồng thời độ chính xác cho bài toán nhận diện người cơ bản (1 class) cũng mang độ chính xác không ổn định và tương đối thấp.
- Nhìn chung, mô hình FOMO tuy không phù hợp với bài toán nhận diện hư hỏng trên container, tuy nhiên bo mạch esp32 vẫn có thể được tận dụng bởi tính tiện lợi (nhỏ gọn, dễ lắp đặt) cũng như giá thành phù hợp.

Vì những điều trên, để đạt được hiệu năng xử lý đáp ứng đủ tốc độ, quy mô mong muốn của giải pháp, nhóm quyết định sử dụng bo mạch raspberry pi 5 và mô hình YOLOv10 cho giải pháp này.

- Về bo mạch raspberry pi, nhóm quyết định sử dụng Pi 5 thay vì thế hệ thứ 4, bởi hiệu năng có được từ raspberry pi 5 so với 4 tương đối lớn so với giá thành (tốc độ xử lý nhanh gấp 2 lần).
- Về mô hình YOLO, nhóm chọn YOLOv10 dù cho dựa trên kết quả đánh giá, độ chính xác mạng lại từ mô hình v10 so với các phiên bản đời trước nhìn chung là không được cải thiện quá nhiều, tuy nhiên, YOLOv10 được thiết kế không chỉ cải thiện độ chính xác, mà còn được tinh giảm và thay đổi cấu trúc bên trong để gia tăng tốc độ thực thi. Vì thế, khi thực hiện nhúng xuống một bo mạch, YOLOv10 là lựa chọn tối ưu hơn.





*Biểu đồ so sánh, mô hình YOLOv10 với các phiên bản đời trước, có thể thấy v10 có độ trễ và số lượng tham số gần như thấp nhất trong các mô hình, trong khi vẫn đảm bảo độ chính xác (AP) của mô hình được train trên tập dữ liệu COCO gần như lớn nhất. (nguồn: <https://github.com/THU-MIG/yolov10>)*

- Đồng thời, nhóm cũng sử dụng mô hình pretrain yolov10 nano để kiểm tra hiệu năng của bo mạch raspberry, đối với các định dạng mô hình khác nhau (model format). Đối với format pytorch, thời gian thực thi đạt xấp xỉ 500ms (tức 1 – 2 fps), còn đối với định dạng NCNN (một định dạng phù hợp hơn cho các thiết bị nhúng, thời gian thực thi đạt xấp xỉ chỉ còn dưới 100ms, (tức > 10 fps).
- Trên thực tế, con số 100ms đối với bài toán này là tương đối dư thừa, tuy nhiên, việc có được một khoản dự phòng về khả năng tính toán như thế này cho phép chúng ta mở rộng mô hình của như tích hợp các tác vụ xử lý đi kèm khác, trong đó có thể là lắp đặt hệ thống quan sát nhiều góc độ với nhiều camera để thu thập hình ảnh, sử dụng esp32 và kết nối thông qua tính năng wifi.

#### 4.3. Chi phí thực hiện

Thành phần	Giá(VND)	Chức năng
Raspberry Pi 5 4GB (thẻ microSD có sẵn)	2.090.000	- Xử lý chính cho hệ thống. - Chạy mô hình deep learning. - Xử lý và phân tích hình ảnh.
Hộp Raspberry Pi 5 nhôm	235.000	- Bảo vệ Raspberry Pi 5. - Tản nhiệt hiệu quả.
Nguồn adapter 5.1V 5A cho Raspberry Pi 5	395.000	- Cung cấp nguồn điện ổn định cho Raspberry Pi 5.
Kit phát triển Wifi BLE ESP32 Camera ESP32-CAM và đế nạp	225.000	- Chụp ảnh và truyền dữ liệu không dây. - Mở rộng khả năng giám sát từ xa.
Cảm biến vật cản hồng ngoại khoảng cách 1-7cm	6.000	- Phát hiện vật thể gần. - Kích hoạt quá trình bắt đầu thu thập dữ liệu.
Camera Module 65 Degree FoV Fixed Focus OV5647 5MP	105.000	- Chụp ảnh độ phân giải ổn định. - Góc nhìn rộng để bao quát container.
Dây nối các loại	/	- Kết nối các thiết bị cần dây nối như cảm biến,...

Tổng chi phí: 3.056.000 VND

Chức năng tổng thể của hệ thống:

1. Raspberry Pi 5 đóng vai trò là bộ não của hệ thống, xử lý chính và chạy mô hình deep learning để phân tích hình ảnh container.
2. Camera Module OV5647 5MP cung cấp hình ảnh, sử dụng cho việc thu thập hình ảnh thô của container để phân tích.
3. ESP32-CAM có thể được sử dụng như một camera thứ hai, cho phép chụp ảnh từ nhiều góc độ hoặc trong trường hợp cần giám sát từ xa.
4. Cảm biến vật cản hồng ngoại được sử dụng để nhận biết và xác định thời điểm xe đầu máy chở container cần được kiểm tra đã đến và dừng tại vị trí phù hợp, từ đó phát tính hiệu chờ đến khi ổn định khung hình, sau đó bắt đầu thu thập hình ảnh.
5. Hộp nhôm bảo vệ Raspberry Pi 5 khỏi môi trường bên ngoài và giúp tản nhiệt hiệu quả.
6. Nguồn adapter đảm bảo cung cấp đủ điện cho toàn bộ hệ thống hoạt động ổn định.

Hệ thống này thiết kế một quy trình kiểm tra tình trạng hư hỏng của container một cách tự động, sử dụng cảm biến để xác định thời điểm thực thi, sử dụng hệ thống camera có thể lắp đặt mở rộng dễ dàng để thu thập hình ảnh, sử dụng mô hình học sâu YOLOv10 để nhận diện và định vị lỗi hư hỏng một cách chính xác và nhanh chóng, đồng thời lưu trữ dữ liệu đã thu thập được để đối chiếu và tùy ý sử dụng về sau.

## **5. Lịch trình dự án và phân công nhiệm vụ**

### **5.1. Ước tính thời gian thực hiện**

Tổng thời gian dự kiến cho dự án: 10 tuần (bao gồm quá trình khảo sát đề tài, thực trạng, thử nghiệm các phương hướng đề xuất, xây dựng ý tưởng và xây dựng hoàn thành mô hình hoàn chỉnh sau cùng).

### **5.2. Các mốc hoàn thành task**

#### **1. Tuần 1-2: Nghiên cứu và chuẩn bị**

- Nghiên cứu sâu về các thuật toán xử lý ảnh và deep learning.
- Thử nghiệm và chạy thử các mô hình máy học từ lớn đến nhỏ, từ cơ bản đến phức tạp (sử dụng bộ lọc và mô hình phân loại đơn giản như svm, knn; sử dụng mô hình FOMO, sử dụng mô hình SSD mobilenet, sử dụng mô hình SSD, ...), đồng thời áp dụng các mô hình tương ứng và phù hợp trên các bo mạch (ESP32cam, Aduino, Raspberry pi 5).
- Chốt phương án cuối cùng, bao gồm thiết kế gần hoàn thiện của sản phẩm và các chức năng đi kèm.
- Tiến hành lên danh sách các vật dụng cần thiết, mua đồ để nghiên cứu và làm sản phẩm.
- Chuẩn bị môi trường phát triển và Raspberry Pi 5.

- Chạy thử nghiệm các chức năng cơ bản điều khiển, nhận diện cơ bản trên Raspberry pi.

## 2. Tuần 3-4: Thu thập và xử lý dữ liệu

- Khảo sát và tiềm kiểm tập dữ liệu phù hợp, thu thập và tổ chức dataset về lỗi container.
- Tiềm xử lý và augment dữ liệu.
- Đánh giá bài toán chung, đồng thời đánh giá bài toán nhật diện lỗi trên container, cập nhật hướng phát triển.

## 3. Tuần 5-6: Phát triển mô hình, tối ưu hóa và triển khai

- Lựa chọn và triển khai mô hình deep learning (YOLO v10).
- Huấn luyện mô hình trên dataset đã chuẩn bị.
- Tối ưu hóa mô hình cho Raspberry Pi 5 bằng các chuyển sang các định dạng mô hình phù hợp với cách thiết bị nhúng.
- Đánh giá hiệu suất và độ chính xác của mô hình, đồng thời khoanh vùng những tính chất (ưu điểm và nhược điểm) của mô hình sau khi đã được huấn luyện.
- Xây dựng thuật toán quy đổi dữ liệu đã được nhận diện sang đơn vị thực tế (ở đây là chi phí để sửa chữa container).
- Xây dựng hệ thống hoạt động cùng với tính hiệu của cảm biến.
- Xây dựng và cài đặt tính năng lưu dữ liệu đã thu thập và sử lí (xác định rõ những thực thể cần lưu).

## 4. Tuần 7-8: Kiểm thử và đánh giá, sửa lỗi còn tồn đọng

- Thực hiện việc nhúng toàn bộ các chức năng xuống bo mạch raspberry pi 5.
- Thực hiện kiểm thử toàn diện các chức năng.
- Tiến hành sửa những lỗi lớn ảnh hưởng tới kết quả, hoàn thiện sản phẩm.

## 5. Tuần 9-10: Hoàn thiện và báo cáo

- Tinh chỉnh cuối cùng dựa trên kết quả kiểm thử.
- Chuẩn bị báo cáo và tài liệu dự án.

### 5.3. Phân công thành viên

#### 1. Nguyễn Quang Dũng (Trưởng nhóm)

- Quản lý dự án và phối hợp giữa các thành viên.
- Nghiên cứu và lựa chọn mô hình deep learning.
- Phát triển và tối ưu hóa mô hình.
- Thu thập và xử lý dataset.
- Triển khai hệ thống trên Raspberry Pi 5.

## 2. Chu Thành Hiếu

- Phát triển pipeline xử lý ảnh.
- Hỗ trợ huấn luyện mô hình.
- Chuẩn bị tài liệu và báo cáo dự án.
- Thiết kế và thực hiện các bài kiểm thử.

## 3. Trần Trí Dũng

- Cấu hình và tối ưu hóa Raspberry Pi 5.
- Đánh giá hiệu suất và độ chính xác của hệ thống.
- Sửa các lỗi liên quan đến hiệu suất đầu ra.
- Chạy huấn luyện mô hình.

## 5.4. Quản lý và theo dõi tiến độ

- Họp nhóm hàng tuần để cập nhật tiến độ và giải quyết vấn đề
- Báo cáo tiến độ định kỳ cho giảng viên hướng dẫn

## 6. Sản phẩm

### 6.1. Giải pháp xử lý ảnh

#### 6.1.1. Mô hình Yolo v10

You only look once (Yolo) là thuật toán sử dụng mạng tích chập học sâu (CNN) để nhận dạng vật thể. Đây là một trong những thuật toán nhận dạng vật thể nhanh nhất hiện nay, mặc dù không phải là phương pháp có độ chính xác cao nhất. Các ưu điểm của mô hình Yolo có thể kể đến như sau:

- Nắm khái quát về đặc trưng của từng đối tượng.
- Gọn nhẹ dễ sử dụng, thông số tùy chỉnh nhiều, hỗ trợ nhiều mô hình pretrain, đặc biệt là mô hình nano với kích thước được tinh giảm, phù hợp để chạy trên thiết bị nhúng.
- Mã nguồn mở.
- Nhanh, có thể sử dụng trên thời gian thực trên raspberry pi 5 nếu sử dụng format NCNN cho mô hình.

Mô hình Yolo có nhiều điểm vượt trội hơn so với các hệ thống nhận dạng khác. Trong quá trình chạy, Yolo sẽ xem xét toàn bộ bức ảnh thay vì từng phần rời rạc, vì thế kết quả phát hiện của nó được rút kết thông tin từ toàn bộ nội dung tấm ảnh. Đồng thời kết quả đánh giá của Yolo cũng chỉ cần được thực hiện một lần chạy mạng đánh giá duy nhất, vì thế so với các mô hình học sâu khác, tốc độ thực thi của Yolo là cực kì nhanh, gấp hàng

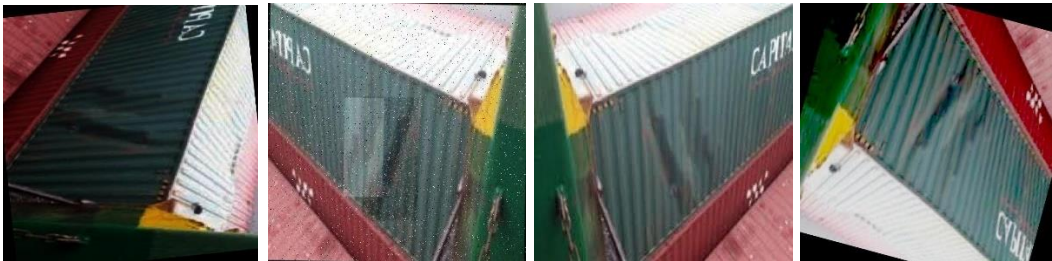
trăm lần so với Fast R\_CNN. Do đó khiến cho Yolo có thể phát hiện vật thể trong ảnh và theo thời gian thực.

Khác với các phiên bản Yolo trước, Yolo v10 có những cải tiến sau nhằm tăng tốc độ thực thi, đáp ứng cho yêu cầu giới hạn về phần cứng của các hệ thống nhúng:

- Gán nhãn kép: Yolo v10 áp dụng kỹ thuật dán nhãn kép (1-1, 1-nhiều) giúp giảm thiểu độ trễ trong quá trình suy luận (vì mô hình chỉ sử dụng 1-1 để dự đoán), nhưng đồng thời vẫn đảm bảo chất lượng của quá trình huấn luyện.
- Giảm thiểu kích thước không gian kênh: Yolo v10 tách rời các hoạt động giảm kích thước không gian và kênh, duy trì hiệu suất nhưng giảm khối lượng tính toán.
- Cơ chế tự chú ý một phần (Partial Self-Attention - PSA): Để tăng cường khả năng nhận diện đối tượng, YOLOv10 tích hợp cơ chế tự chú ý một phần, giúp cân bằng giữa việc trích xuất đặc trưng toàn cục và tiết kiệm tính toán. PSA chỉ xử lý một phần của bản đồ đặc trưng, cải thiện độ chính xác mà không ảnh hưởng nhiều đến hiệu suất.

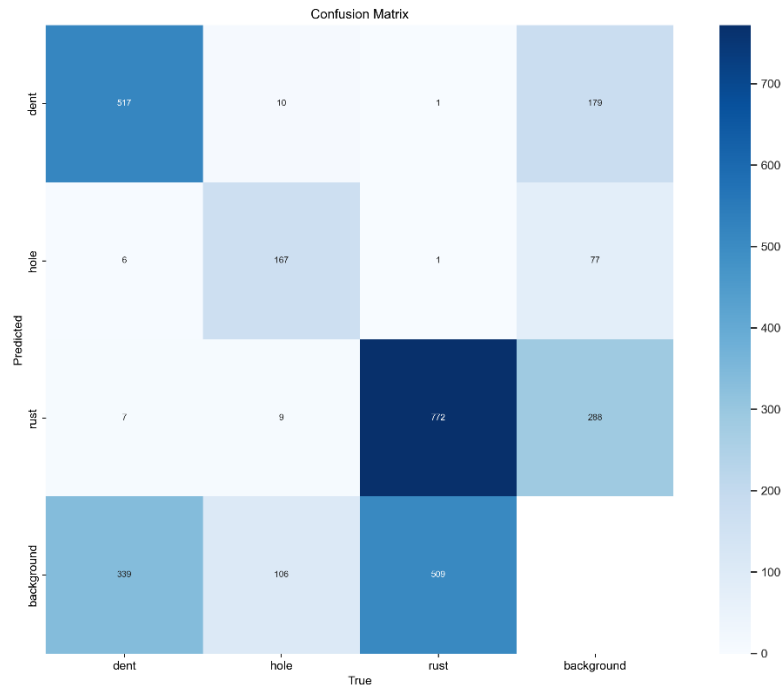
### 6.1.2. Xây dựng và kết quả huấn luyện mô hình

Tập dữ liệu được sử dụng để huấn luyện mô hình cho bài toán này bao gồm 5178 bức hình container, trong đó 4142 tấm được dùng cho tập train và test, 1036 tấm được dùng cho tập valid. Tập dữ liệu này bao gồm 3 class (“dent”, “rust”, “hole”) và đã được áp dụng các phép biến đổi và bộ lọc để làm phong phú tập dữ liệu (data augmentation).



*Sử dụng phương pháp data augmentation để làm giàu dữ liệu cho quá trình huấn luyện*

Mô hình cuối cùng được train 300 epoch, với batch size 16, train trên RTX4060 laptop 8gb vram, thời gian 6.043 giờ.



*Confusion matrix*

## Phân tích Confusion Matrix ( ma trận nhầm lẫn )

### Phát hiện lỗi "Dent" (móp):

- Mô hình phân loại đúng khá nhiều trường hợp "dent," thể hiện qua số lượng cao trên đường chéo của ma trận cho "dent."
- Tuy nhiên, vẫn có một số trường hợp nhầm lẫn, đặc biệt là khi "dent" bị nhầm với "background" (179 trường hợp). Điều này có thể cho thấy mô hình đôi khi gặp khó khăn trong việc phân biệt giữa "dent" và các vùng không có lỗi.

### Phát hiện lỗi "Hole" (lỗ):

- Mô hình phát hiện "hole" khá tốt, với đa số các trường hợp được phân loại chính xác. Tuy nhiên, có một số nhầm lẫn với "rust" và "background," cho thấy có thể đặc điểm hình ảnh của "hole" không luôn rõ ràng, hoặc đôi khi dễ nhầm lẫn với các lỗi khác.

### Phát hiện lỗi "Rust" (rỉ sét):

- Khả năng phát hiện "rust" của mô hình rất tốt, với số lượng lớn nhất trên đường chéo (772), điều này cho thấy độ chính xác cao cho lớp "rust."
- Có một số ít trường hợp nhầm lẫn, nhưng số lượng này thấp hơn so với các lớp khác, điều này cho thấy "rust" có các đặc điểm phân biệt mà mô hình đã học tốt.

#### Phân loại "Background" (nền):

- Có một số nhầm lẫn khi nền bị phân loại nhầm thành các lỗi (chẳng hạn như "dent" và "rust"). Việc phân loại nhầm các vùng nền thành lỗi có thể dẫn đến tỷ lệ dương tính giả cao, làm giảm độ tin cậy khi ứng dụng mô hình vào thực tế.

Tổng quan, ma trận cho thấy mô hình hoạt động tốt nhưng có thể cần tinh chỉnh thêm để giảm các trường hợp nhầm lẫn, đặc biệt là với lớp "background."

Class	Precision	Recall	mAP-50
All	0.68	0.554	0.557
Dent	0.69	0.57	0.562
Hole	0.625	0.538	0.5
Rust	0.725	0.556	0.608

*Bảng tóm tắt kết quả trên mô hình best.pt*

#### **Toàn bộ mô hình (All):**

- Precision: 0.68 – tức là trong các dự đoán của mô hình, 68% là chính xác (ít nhầm lẫn với các lớp khác).
- Recall: 0.554 – mô hình phát hiện được 55.4% các trường hợp thực sự có lỗi trong dữ liệu kiểm thử.
- mAP-50: 0.557 – mức độ chính xác trung bình trên ngưỡng IoU 0.5 là 55.7%, phản ánh hiệu suất tổng quan của mô hình.

#### **Lớp "Dent":**

- Precision: 0.69 – khả năng phân biệt "Dent" tương đối ổn, với độ chính xác dự đoán là 69%.
- Recall: 0.57 – mô hình nhận diện đúng 57% các trường hợp "Dent" thực sự.

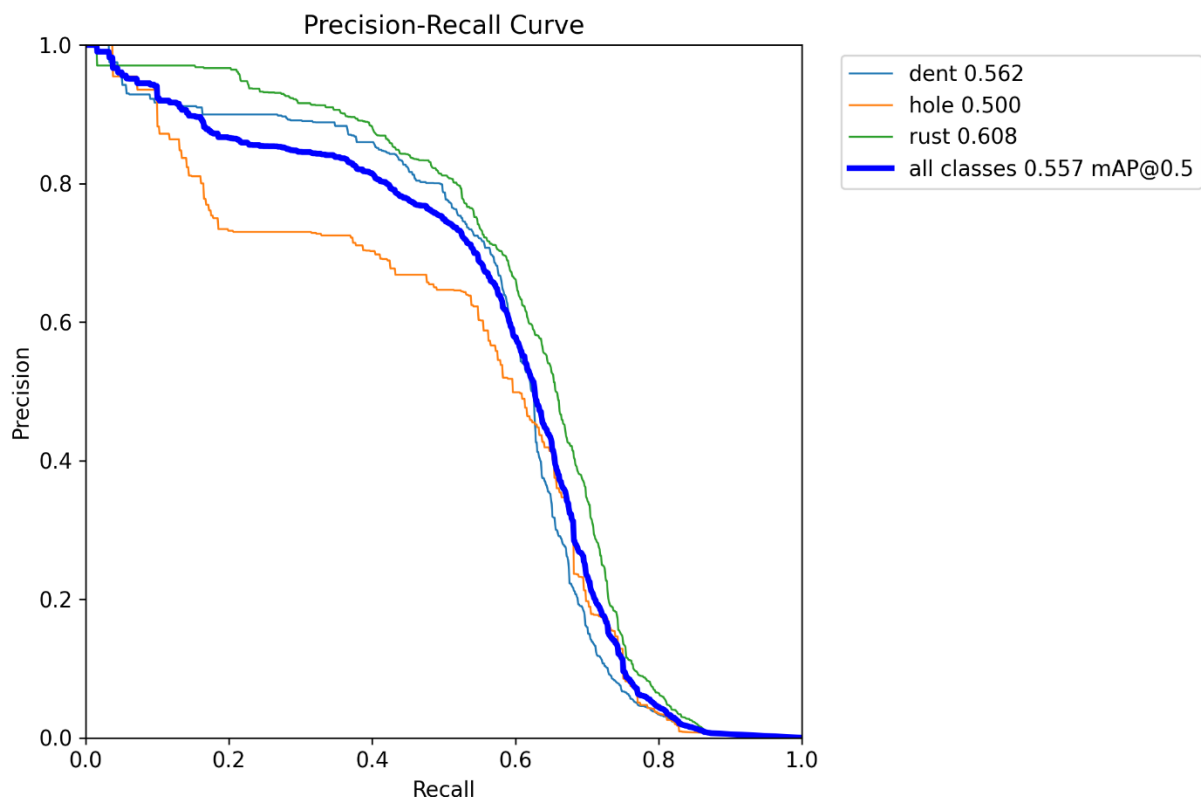
- mAP-50: 0.562 – độ chính xác trung bình của lớp "Dent" là 56.2%, cho thấy mô hình cần cải thiện để phát hiện tốt hơn các lỗi "Dent."

### Lớp "Hole":

- Precision: 0.625 – độ chính xác của dự đoán "Hole" là 62.5%, mức này hơi thấp, chỉ ra rằng mô hình có thể dễ bị nhầm lẫn "Hole" với các lớp khác.
- Recall: 0.538 – mô hình chỉ phát hiện được khoảng 53.8% các trường hợp "Hole" trong dữ liệu kiểm thử, thấp hơn so với các lớp khác.
- mAP-50: 0.5 – độ chính xác trung bình thấp nhất trong ba lớp, điều này cho thấy "Hole" là lớp khó phân biệt nhất với mô hình hiện tại.

### Lớp "Rust":

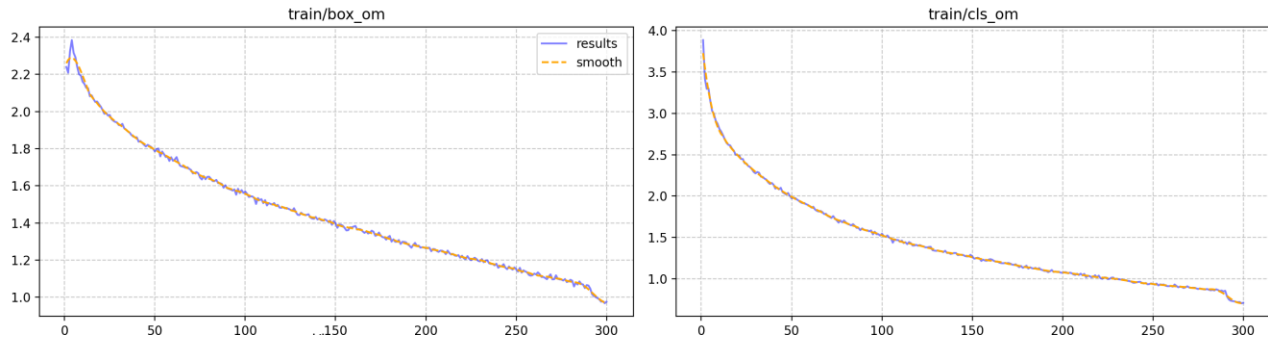
- Precision: 0.725 – mô hình dự đoán "Rust" với độ chính xác cao nhất (72.5%), cho thấy lớp này có đặc điểm dễ phân biệt hơn.
- Recall: 0.556 – phát hiện được 55.6% các trường hợp "Rust," gần bằng với độ recall của toàn bộ mô hình.
- mAP-50: 0.608 – đây là lớp có mAP-50 cao nhất, cho thấy mô hình hoạt động tốt nhất trên lớp "Rust" và có khả năng phân biệt "Rust" tốt hơn so với hai lớp còn lại.



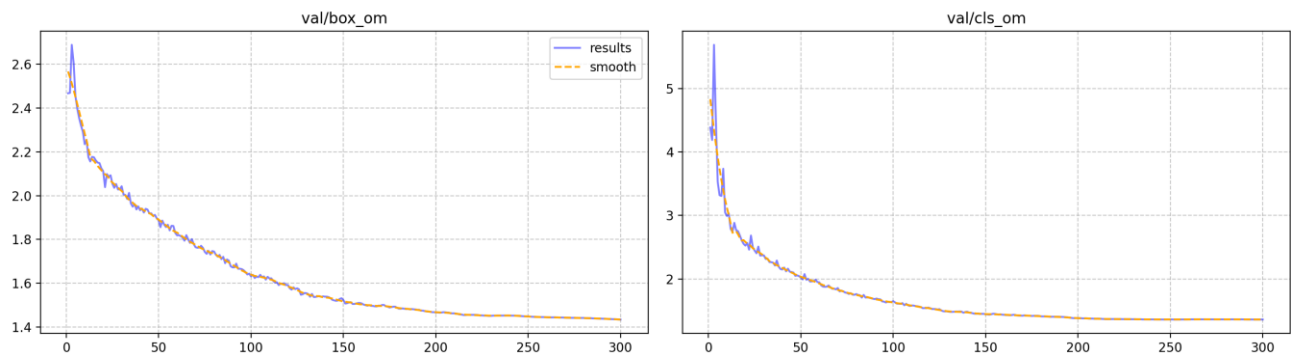


### *Biểu đồ precision theo recall*

Biểu đồ precision theo recall cũng cho thấy các chỉ số của mô hình đã gần như chạm ngưỡng, hội tụ về một điểm, độ chính xác trung bình của cả mô hình (mAP) nhìn chung vẫn tương đối thấp, và điểm mAP của riêng class hole thấp hơn so với 2 class còn lại



*Biểu đồ train loss*

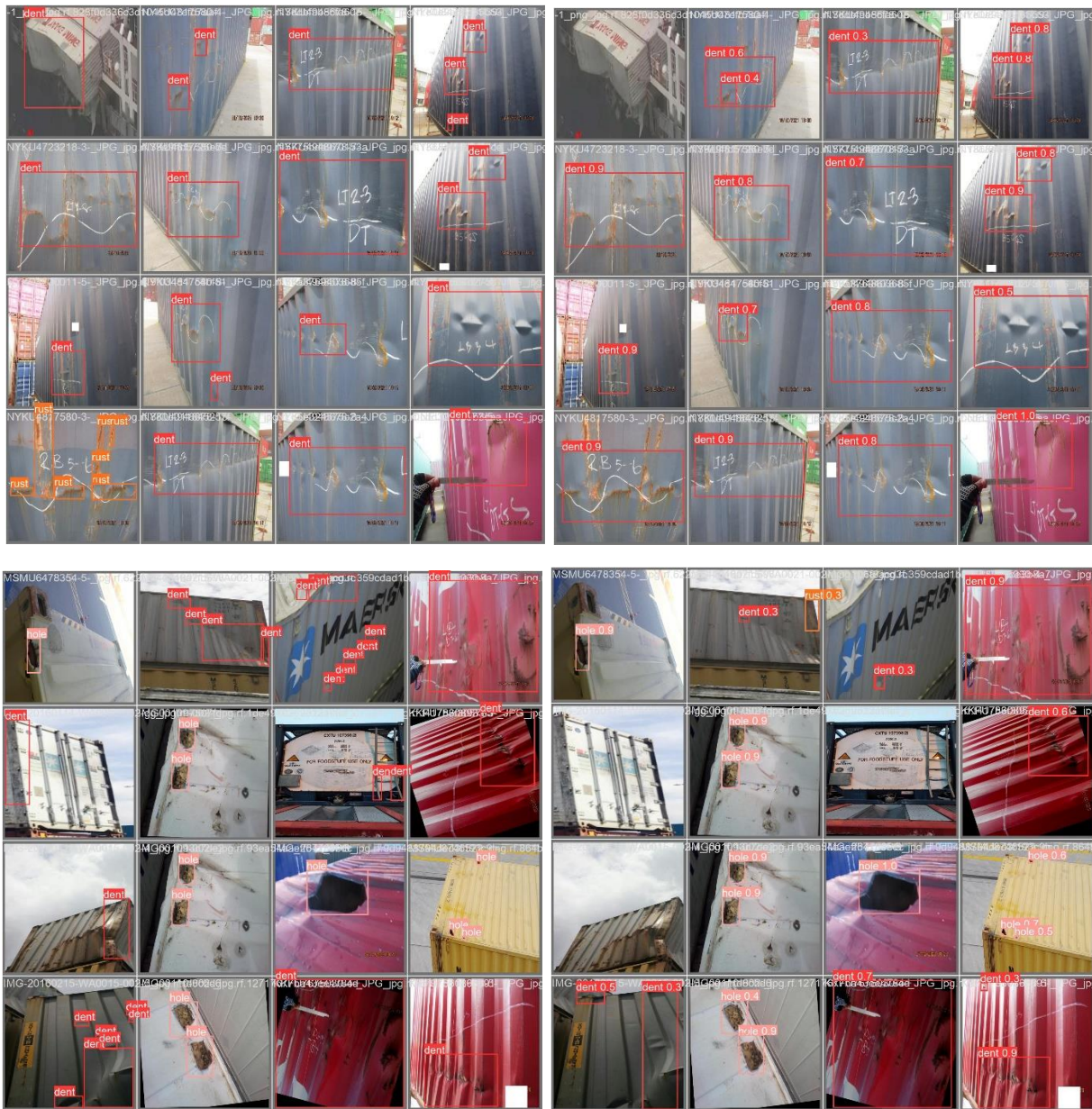


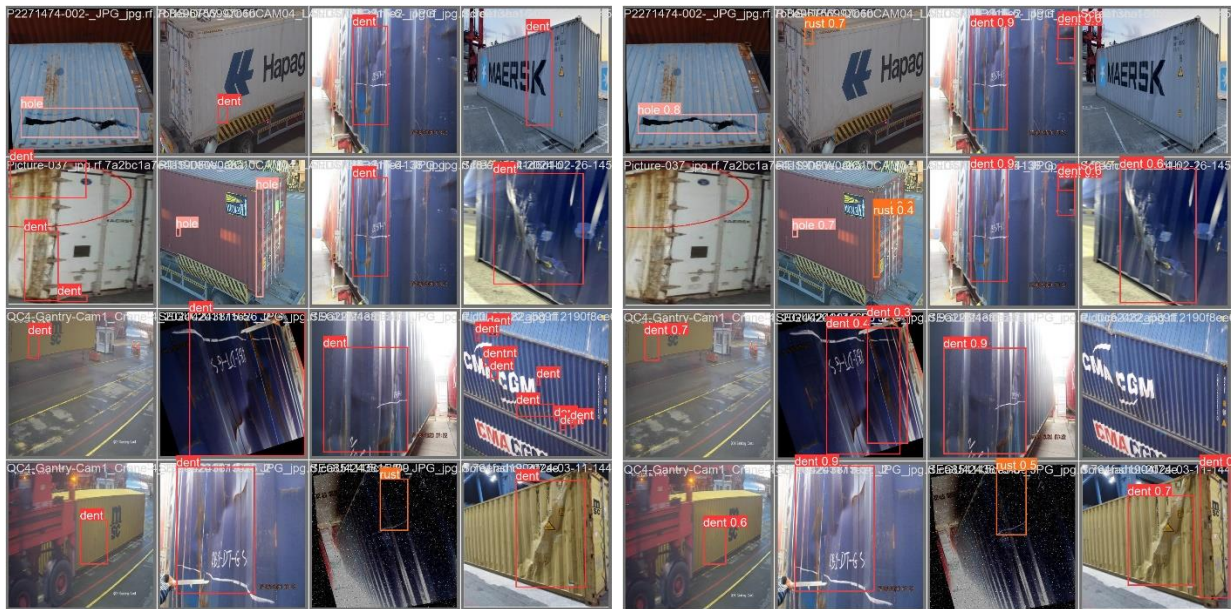
*Biểu đồ valid loss*

Quan sát đồ thị hàm loss qua thời gian (epoch) có thể thấy rằng chỉ số loss của quá trình huấn luyện về cả bài toán classification và localization vẫn đang có xu hướng đi xuống, tuy nhiên ngược lại, chỉ số loss của quá trình đánh giá có vẻ như đã hội tụ và không có xu hướng giảm đáng kể.

So sánh chỉ số loss của quá trình huấn luyện và quá trình đánh giá, chỉ số loss của quá trình huấn luyện nhìn chung thấp hơn so với chỉ số loss của quá trình đánh giá một khoảng không đáng kể, kể cả nếu xét sau 300 epoch. Điều đó có nghĩa mô hình đang có dấu hiệu bị overfit nhẹ.

Dưới đây là một số kết quả dự đoán trong giai đoạn đánh giá mô hình tốt nhất sau khi train 300 epoch:

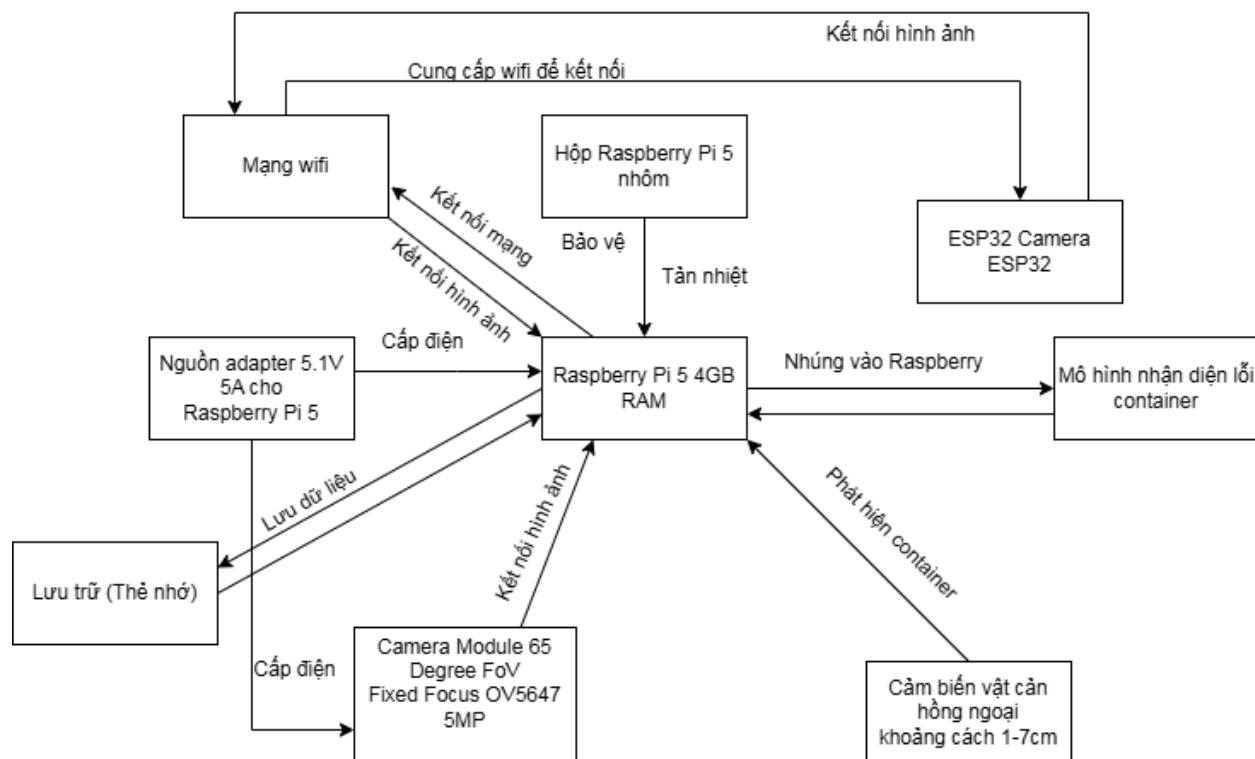




Một số kết quả dự đoán trên mô hình best.pt (bên trái – label, bên phải – dự đoán)

## 6.2. Thiết kế hệ thống phần mềm và phần cứng

Về cơ bản, giải pháp sử dụng bo mạch Raspberry pi 5 và camera để thực hiện lấy dữ liệu và dự đoán những hỏng hóc của container thông qua mô hình Yolo, sơ đồ thiết kế của giải pháp có thể được vẽ như sau:





Hệ thống camera được sử dụng để ghi nhận hình ảnh có thể được mở rộng thông qua việc lắp đặt phần cứng và cập nhật đầu vào camera trên phần mềm. Dữ liệu từ các nguồn camera sẽ được thu lại cùng một thời điểm và được đánh dấu dựa trên ID của container.

Về phần mềm, giải pháp sử dụng ngôn ngữ lập trình python vì đối với các bài toán thị giác máy tính, đặc biệt là các mô hình học sâu phức tạp, ngôn ngữ python có những lợi thế vượt trội hơn so với các ngôn ngữ khác trong việc thiết lập và tương tác với các kiến trúc mạng học sâu.

### **6.3. Đầu ra của sản phẩm**

Sau khi đã thực hiện xong khâu dự đoán dựa trên các bức hình thu thập được, phần mềm sẽ dựa trên danh sách các lỗi hỏng hóc đã kèm theo xác suất chính xác của dự đoán, được lưu trong một biến mảng, để tính toán chi phí hao tổn sau cùng.

Các tham số quan trọng được sử dụng để tính chi phí hao tổn như sau:

Base cost: các lỗi khác nhau sẽ được cài đặt gắn với một base cost ban đầu (USD):

- Dent (vết lõm): 50
- Hole (lỗ): 100
- Rust ( gỉ): 30

Confidence: Đây là độ chính xác của mô hình YOLO trong việc phát hiện hư hỏng, có giá trị từ 0 đến 1. Mô hình sử dụng độ tự tin để xác định mức độ nghiêm trọng của hư hỏng.

Severity: Được xác định dựa trên độ tự tin:

- Severe (Nghiêm trọng): Độ tự tin > 0.8 (multiplier 1.5)
- Moderate (Vừa phải): Độ tự tin từ 0.6 đến 0.8 (multiplier 1.2)
- Minor (Nhẹ): Độ tự tin từ 0.4 đến 0.6 (multiplier 1.0)
- Dim (Mờ nhạt): Độ tự tin < 0.4 (multiplier 0.5)

Penalty: nếu có nhiều hư hỏng cùng loại, hình phạt có thể được áp dụng. Khi một loại hư hỏng xuất hiện nhiều lần, hình phạt sẽ làm giảm chi phí sửa chữa cho các lần tiếp theo. Bội số hình phạt mặc định là 1 (không có hình phạt), nhưng nếu có nhiều hư hỏng trùng loại, bội số hình phạt có thể giảm xuống (ví dụ: 0.96)

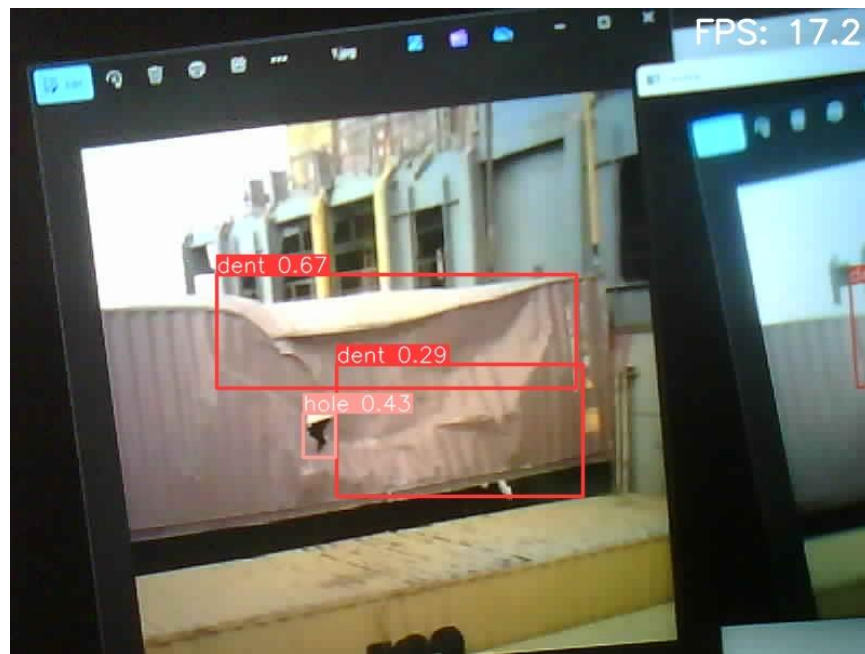
Dựa vào những bội số điều chỉnh như trên, ta có thể viết công thức tính tổng chi phí để sửa chữa container như dưới đây.

$$Total\ Cost = \sum (Base\ Cost \times Confidence \times Severity\ Multiplier \times Penalty\ (nếu\ có))$$

Dữ liệu được tính toán sẽ được tổng hợp và lưu lại vào trong một file CSV (bao gồm cả đường link dẫn đến bức ảnh được dùng để thực hiện dữ đoán).

	A	B	C	D	E	F
1	Container ID	Time	Image	Detail Report	Summary Report	Total Cost
2	1	10/22/2024 17:40	data\1\1a605a35-9062-11ef-913d-005056c00008.jpg	No damage found!	No damage found!	0
3	2	10/22/2024 17:41	data\2\33952cba-9062-11ef-a8c3-005056c00008.jpg	Damage Type: dent Damage Type: dent Base Cost: \$50 Confidence: 0.67 Severity: Moderate Adjusted Cost: \$40.4 Penalty Applied: No ----- Damage Type: hole Base Cost: \$100 Confidence: 0.43 Severity: Minor Adjusted Cost: \$43.08 Penalty Applied: No ----- Damage Type: dent Base Cost: \$50 Confidence: 0.29 Severity: Dim Adjusted Cost: \$6.86 Penalty Applied: Yes (4% reduction) -----	Dent: 1 (cost: 6.46)	6.46
4	3	10/22/2024 17:41	data\3\391ab12d-9062-11ef-aec4-005056c00008.jpg		Dent: 2 (cost: 47.26) Hole: 1 (cost: 43.08) Rust: 0 (cost: 0)	90.34

*Ví dụ cho kết quả dữ liệu được lưu vào trong file CSV sau khi thực hiện dự đoán*



Hình ảnh được sử dụng để dự đoán lỗi cho container có ID = 3

## 7. Kết luận

Dự án "Xử lý ảnh nhận biết lỗi hư hỏng của container" không chỉ mang tiềm năng lớn trong việc cải thiện hiệu quả và độ chính xác của quá trình kiểm tra container mà còn có thể thay đổi cách thức mà ngành công nghiệp vận tải và logistics thực hiện việc kiểm tra

chất lượng container. Các tiến bộ trong công nghệ xử lý ảnh và học máy đã mở ra cơ hội để phát triển các hệ thống tự động có thể nhận diện chính xác các khuyết tật như vết lõm, rỉ sét và các hư hỏng khác mà không cần sự can thiệp trực tiếp của con người.

Thông qua việc áp dụng các phương pháp như học sâu (deep learning), hệ thống có thể học và nhận diện các mẫu hư hỏng với độ chính xác cao, giảm thiểu sai sót do sự mệt mỏi hoặc sự thiếu tập trung của người kiểm tra. Các hướng phát triển, như cải thiện thuật toán để nhận diện các lỗi khó phát hiện hoặc nâng cao khả năng phân tích các yếu tố ảnh hưởng đến độ bền của container, đều đóng góp vào việc tạo ra một hệ thống mạnh mẽ và đáng tin cậy.

Ngoài ra, với khả năng tự động hóa quá trình kiểm tra, dự án sẽ giúp giảm thời gian và chi phí kiểm tra, đồng thời nâng cao năng suất làm việc. Hệ thống này có thể được tích hợp vào các quy trình hiện tại trong các cảng biển, kho bãi, và các trung tâm vận chuyển, mang lại lợi ích lâu dài trong việc nâng cao hiệu quả hoạt động. Với những tính năng tiên tiến và tiềm năng ứng dụng rộng rãi, nhóm kỳ vọng hệ thống sẽ trở thành một công cụ không thể thiếu trong ngành công nghiệp vận tải, logistics, và bảo trì container.

**Source code:**

<https://github.com/DungQuangUiT/Container-damage---Raspberry-Pi-5.git>