ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

Samsung Innovation Campus

Giảng viên: Cao Văn Chung

Nhận diện và phân loại rác

Thành viên nhóm

Nguyễn Vĩ Chí Cường	(SIC251022)
Đỗ Tiến Dũng	(SIC250976)
Vũ Đức Duy	(SIC251039)
Nguyễn Tiến Dũng	(SIC251018)

Hà Nội, 2025

Lời nói đầu

Rác thải đang trở thành một trong những thách thức môi trường lớn nhất của thế giới hiện đại. Không chỉ gây ảnh hưởng trực tiếp đến sức khỏe con người, rác thải còn tác động tiêu cực đến hệ sinh thái, đe dọa sự sống của các loài động – thực vật và làm suy thoái môi trường tự nhiên. Theo Báo cáo hiện trạng môi trường quốc gia năm 2024, mỗi năm tại Việt Nam phát sinh khoảng 28 triệu tấn rác thải sinh hoạt, nhưng chỉ khoảng 85% trong số đó được thu gom và xử lý đúng cách. Phần còn lại bị thải bỏ không kiểm soát, gây ô nhiễm môi trường và lãng phí tài nguyên.

Trước thực trạng đáng báo động đó, việc ứng dụng công nghệ để hỗ trợ phát hiện, phân loại và quản lý rác thải là một hướng đi thiết thực và cấp bách. Trong bài tập này, chúng em nghiên cứu và phát triển một mô hình nhận diện và phân loại rác thải. Mục tiêu chính là xây dựng một hệ thống có khả năng tự động phát hiện các loại rác thải khác nhau – từ rác nhựa, kim loại cho đến chai lọ – trong các điều kiện môi trường đa dạng.

Quy trình nghiên cứu được triển khai qua bốn giai đoạn chính: thu thập và gán nhãn dữ liệu ảnh, huấn luyện và mô hình, nhận diện và phân loại, đưa ra quyết định xử lý. Thông qua mô hình này, chúng em mong muốn góp phần hỗ trợ nhận diện và phân loại rác thải. Nhóm rất mong nhận được các ý kiến đóng góp quý báu từ thầy và các bạn để tiếp tục hoàn thiện và phát triển dự án.

Phân công công việc

STT	Họ và Tên	Công Việc Phụ Trách
1	Nguyễn Vĩ Chí Cường	Code train mô hình, tìm siêu tham số, web, viết báo cáo.
2	Vũ Đức Duy	Viết báo cáo, làm slide thuyết trình.
3	Đỗ Tiến Dũng	Code web, viết báo cáo, làm slide thuyết trình.
4	Nguyễn Tiến Dũng	Viết báo cáo, làm slide thuyết trình.

Mục lục

1 GIỚI THIỆU VẤN ĐỀ

1.1 Khái quát vấn đề

Trong thời đại hiện nay, môi trường đang trở thành vấn đề cấp bách của toàn xã hội. Sự phát triển nhanh chóng của khoa học kỹ thuật cùng nhu cầu sống ngày càng tăng đã làm gia tăng đáng kể lượng rác thải, gây ô nhiễm đất, nước, không khí và ảnh hưởng tiêu cực đến sức khỏe cộng đồng cũng như hệ sinh thái.

Trước thực trạng đó, việc ứng dụng công nghệ, đặc biệt là trí tuệ nhân tạo, vào phát hiện và phân loại rác thải là một giải pháp hiệu quả. Các hệ thống thông minh giúp xử lý rác nhanh chóng, chính xác, đồng thời nâng cao ý thức bảo vệ môi trường của người dân và mở ra hướng nghiên cứu, ứng dụng thiết thực.

Rác thải bao gồm nhiều loại như nhựa, kim loại, thủy tinh, chất hữu cơ,... phát sinh từ các nguồn như sinh hoạt, công nghiệp, nông nghiệp. Nếu không được xử lý đúng cách, chúng sẽ gây hại nghiêm trọng đến môi trường. Việc thống kê, phân tích và phân loại rác thải tại nguồn là yếu tố then chốt trong quản lý rác hiện đại.

Nhận thức rõ về lượng và tác hại của rác giúp cộng đồng chủ động hơn trong phân loại, tái chế và giảm thiểu chất thải. Đây cũng là cơ sở để xây dựng chính sách môi trường và phát triển các giải pháp công nghệ bền vững, góp phần bảo vệ hành tinh cho thế hệ tương lai.

Hình 1: Hình minh họa về các loại rác thải

1.2 Mục tiêu và ý nghĩa của đề tài

Đề tài nhằm xây dựng một hệ thống nhận diện và phân loại rác thải tự động từ hình ảnh. Hệ thống cho phép người dùng tải lên ảnh, sử dụng mô hình học sâu đã huấn luyện để xác định loại rác (nhựa, kim loại, giấy, v.v.) và hiển thị kết quả trực quan kèm thông tin nhãn, độ tin cậy và vùng phát hiện. Kết quả được lưu trữ để phục vụ phân tích và cải tiến sau này.

Về ý nghĩa, hệ thống hỗ trợ phân loại rác tại nguồn, góp phần nâng cao hiệu quả tái chế và giảm ô nhiễm môi trường. Đề tài thể hiện vai trò của AI trong giải quyết các vấn đề môi trường theo hướng bền vững.

1.3 Bài toán nhận diện và phân loại rác thải qua hình ảnh

Hiện nay, lượng rác thải do con người tạo ra ngày càng gia tăng, trong khi công tác thu gom và phân loại vẫn chủ yếu thực hiện thủ công, chưa được tự động hóa hay tối ưu hóa. Phần lớn rác thải được thu gom theo phương thức không phân loại tại nguồn, mà được tập hợp lẫn lộn và vận chuyển đến các điểm tập kết. Tại đó, quá trình phân loại mới diễn ra để quyết định liệu rác sẽ được tái chế hay xử lý tiêu hủy. Điều này không chỉ làm giảm hiệu quả của công tác xử lý rác mà còn gây mất nhiều thời gian và nguồn lực.

Đặc biệt, việc phân loại rác thủ công luôn tiềm ẩn nguy cơ gây ô nhiễm và ảnh hưởng trực tiếp đến sức khỏe của người lao động và môi trường xung quanh. Do đó, việc xây dựng một hệ thống phân loại rác thải tự động là cần thiết để nâng cao hiệu suất, đảm bảo vệ sinh và hướng đến bảo vệ môi trường một cách bền vững.

Trong đề tài này, chúng em đề xuất một mô hình phân loại rác thải dựa trên hình ảnh, sử dụng công nghệ thị giác máy tính và học sâu (deep learning), sử dụng YOLOv11, Faster R-CNN, Rt-DETR để train. Ý tưởng cốt lõi là nhận diện và phân loại các loại rác thải thông qua hình ảnh chụp từ camera hoặc thiết bị đầu vào.

1.3.1 Quy trình cơ bản của hệ thống

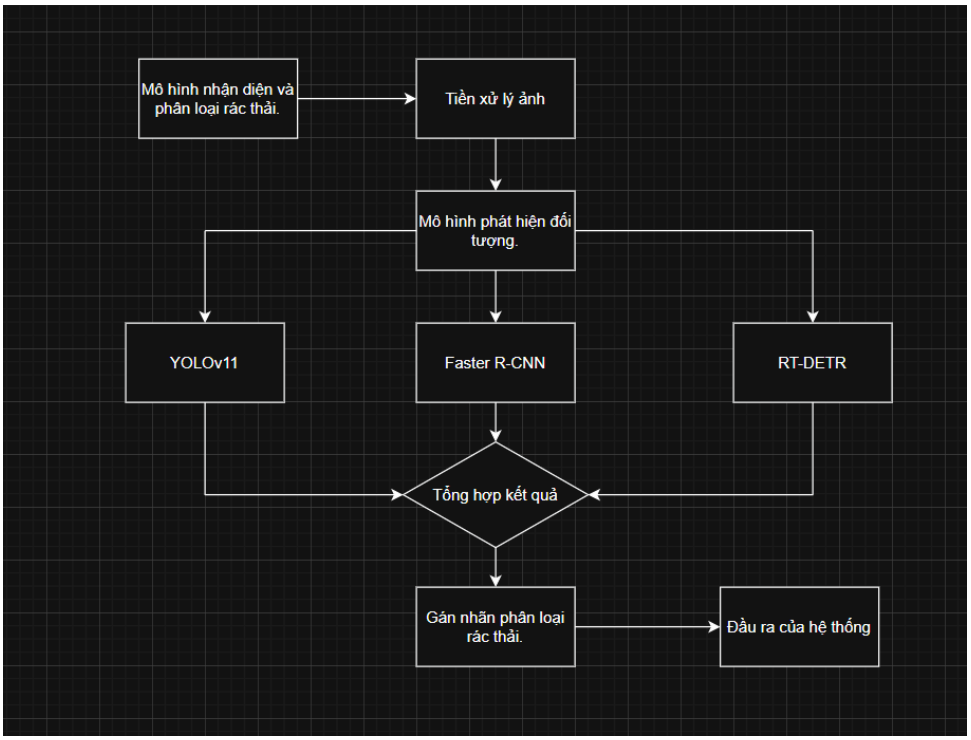
1. **Thu thập và gán nhãn dữ liệu ảnh:** Các hình ảnh rác thải được thu thập và gán nhãn tương ứng (ví dụ: nhựa, kim loại, giấy, thực phẩm, v.v.), nhằm phục vụ cho quá trình huấn luyện mô hình.
2. **Huấn luyện mô hình phân loại:** Mô hình học sâu sẽ được huấn luyện để phân biệt giữa các loại rác thải dựa trên đặc trưng hình ảnh.
3. **Nhận diện và phân loại:** Khi hệ thống đã được huấn luyện, nó có thể tự động nhận diện loại rác thải có trong ảnh mới và hỗ trợ phân loại nhanh chóng, chính xác.
4. **Đưa ra quyết định xử lý:** Dựa vào kết quả phân loại, hệ thống có thể đưa ra hướng xử lý phù hợp như tái chế, tiêu hủy, hoặc thu gom riêng.

Việc ứng dụng mô hình nhận diện rác thải từ hình ảnh không chỉ nâng cao hiệu quả phân loại mà còn mở ra khả năng triển khai các giải pháp công nghệ cao vào công tác quản lý rác thải tại các đô thị, khu công nghiệp và hộ gia đình.

1.3.2 Sơ đồ hoạt động mô hình

Mô hình phân loại rác thải bằng hình ảnh là một hệ thống ứng dụng trí tuệ nhân tạo (AI), cụ thể là thị giác máy tính (Computer Vision) và học sâu (Deep Learning), nhằm tự động phát hiện và phân loại các loại rác như: nhựa, kim loại, giấy, thực phẩm,... từ ảnh đầu vào.

Hệ thống này giúp thay thế quy trình phân loại rác thủ công, tăng tốc độ và độ chính xác trong phân loại tại nguồn hoặc trong dây chuyền xử lý rác thải.



Hình 2: Hình minh họa mô hình

2 Dữ liệu

2.1 Thu thập dữ liệu

Nguồn dữ liệu của dự án được lấy từ nền tảng Roboflow, một công cụ phổ biến trong cộng đồng học máy, cho phép tạo và chia sẻ các tập dữ liệu thị giác máy tính.Bộ dữ liệu được lựa chọn bao gồm 22668 ảnh, trong đó có 19824 ảnh dùng để huấn luyện, 1935 ảnh dùng để kiểm định, 909 ảnh dùng để kiểm tra.

2.2 Nguồn dữ liệu

Link : https://app.roboflow.com/detectionclassificationgarbage/detection-garbage-jkww2

2.3 Tổ chức dữ liệu

Dữ liệu sau khi được xử lý được tổ chức thành ba thư mục riêng biệt: train, valid và test, tương ứng với các tập huấn luyện, xác thực và kiểm tra, theo tỷ lệ xấp xỉ 87% – 9% – 4%. Mỗi hình ảnh được lưu dưới định dạng .jpg, kèm theo một tệp .txt chứa nhãn tương ứng. Mỗi ảnh trong tập dữ liệu đều đi kèm với một tệp nhãn chứa thông tin về tên lớp và tọa độ của các hộp giới hạn (bounding boxes) đã được chuẩn hóa. Việc ghép nối giữa ảnh và nhãn được thực hiện một cách đồng nhất: không có ảnh nào thiếu nhãn và ngược lại, không tồn tại nhãn không tương ứng với ảnh.

2.3.1 Phân tích tập dữ liệu

1. **TRAIN SET :** Là phần lơn nhất trong dữ liệu. Mô hình sẽ "Học" từ đây –> phát hiện đặc trưng hình ảnh, phân biệt loại rác thải.
2. **VALID SET :** Dùng để theo dõi hiệu suất mô hình trong khi huấn luyện. Giúp phát hiện hiện tượng overfitting (mô hình học quá khớp với TRAIN SET nhưng kém ở dữ liệu mới).
3. **TEST SET :** Không được dùng trong quá trình huấn luyện. Chỉ dùng một lần duy nhất sau khi mô hình đã hoàn thành để đánh giá khách quan xem nó hoạt động như thế nào với dữ liệu hoàn toàn mới.

2.3.2 Tiền xử lý dữ liệu

Trong quá trình huấn luyện mô hình học sâu cho bài toán nhận diện và phân loại rác thải, bước tiền xử lý dữ liệu đóng vai trò then chốt nhằm chuẩn hóa đầu vào, đồng thời nâng cao độ chính xác và khả năng tổng quát hóa của mô hình. Do dữ liệu thu thập có sự khác biệt về kích thước, định hướng và bố cục, nhóm tiến hành áp dụng một số kỹ thuật xử lý ảnh cơ bản để đảm bảo tính nhất quán và chất lượng dữ liệu đầu vào.

- **Auto-Orient:** Tự động điều chỉnh hướng ảnh dựa trên metadata có sẵn nhằm đảm bảo tất cả ảnh được căn chỉnh đúng chiều, hạn chế sai lệch trong nhận diện đối tượng.

- **Resize (640×640):** Tất cả ảnh được co giãn về kích thước cố định là `640×640` pixels. Việc chuẩn hóa kích thước giúp đảm bảo tính đồng nhất khi đưa ảnh vào mô hình YOLO, đồng thời cân bằng giữa độ phân giải và chi phí tính toán.

Việc tiền xử lý này không chỉ giúp đồng bộ dữ liệu mà còn góp phần tăng hiệu quả huấn luyện, giảm thời gian xử lý và hạn chế lỗi do dữ liệu không nhất quán gây ra.

2.3.3 Tăng cường dữ liệu

Do số lượng ảnh gốc trong tập huấn luyện còn hạn chế và sự đa dạng trong điều kiện môi trường (ánh sáng, góc chụp, bố cục), nhóm áp dụng kỹ thuật tăng cường dữ liệu nhằm sinh thêm các biến thể từ ảnh gốc, giúp mô hình học được nhiều đặc trưng hơn, giảm hiện tượng overfitting và nâng cao khả năng nhận diện trong thực tế.

- **Flip (Horizontal):** Lật ngang ảnh để mô phỏng góc nhìn đối xứng của vật thể. Kỹ thuật này giúp mô hình không bị phụ thuộc vào vị trí xuất hiện của đối tượng trong khung hình.
- **Rotate (90°):** Xoay ảnh theo ba hướng: theo chiều kim đồng hồ, ngược chiều kim đồng hồ và xoay 180 độ. Điều này cho phép mô hình nhận diện được rác thải ở nhiều tư thế và góc nhìn khác nhau.
- **Zoom (Phóng to):** Ảnh được phóng to với mức tối đa 20% (maximum zoom), giúp mô phỏng khoảng cách chụp gần – xa. Không thực hiện crop nhỏ (minimum zoom là 0%) để đảm bảo giữ nguyên toàn bộ vật thể trong ảnh.
- **Số ảnh tăng cường:** Với mỗi ảnh gốc, hệ thống tạo ra 3 ảnh tăng cường khác biệt bằng cách kết hợp ngẫu nhiên các phép biến đổi trên.

3 Các phương pháp giải quyết

3.1 YOLO11

3.1.1 Tổng quan về YOLOv11

YOLO (You Only Look Once) là dòng mô hình học sâu chuyên dùng cho bài toán phát hiện đối tượng (object detection) theo cơ chế một bước (one-stage detection). Các phiên bản mới của YOLO, đặc biệt là YOLOv11, đã có nhiều cải tiến đáng kể nhằm gia tăng độ chính xác mà vẫn giữ được tốc độ xử lý nhanh – yếu tố then chốt trong các hệ thống xử lý thời gian thực như phân loại rác.

YOLOv11 kế thừa tư tưởng "nhìn một lần – nhận diện tất cả" (one-shot prediction) và được nâng cấp đáng kể với:

- **Backbone hiện đại:** YOLOv11 sử dụng backbone lai giữa ConvNet và attention module như Transformer hoặc EfficientNet để tăng khả năng trích xuất đặc trưng hình ảnh từ nhiều cấp độ.
- **Neck module hiệu quả:** BiFPN (Bidirectional Feature Pyramid Network) giúp tổng hợp thông tin từ các tầng khác nhau, tăng hiệu quả phát hiện vật thể có kích thước nhỏ – điều phổ biến trong bài toán phân loại rác.
- **Detection head anchor-free:** Thay vì dựa vào anchor boxes truyền thống, YOLOv11 sử dụng cơ chế anchor-free để giảm độ phức tạp, đồng thời tăng khả năng phát hiện các vật thể có hình dạng không ổn định, chẳng hạn như túi nilon nhăn, giấy vo tròn, hay rác hữu cơ.
- **Khả năng tích hợp attention:** Nhiều cấu hình của YOLOv11 cho phép tích hợp các cơ chế như CBAM hoặc Self-Attention để tăng hiệu quả nhận diện trong môi trường có nhiễu nền hoặc điều kiện ánh sáng kém.

Nhờ những đặc điểm trên, YOLOv11 rất phù hợp với bài toán phân loại rác thải, nơi dữ liệu hình ảnh thường có nhiều đối tượng nhỏ, hình dạng bất quy tắc và góc chụp không đồng nhất.

3.1.2 Cấu hình huấn luyện

Trong dự án này, nhóm sử dụng mô hình YOLOv11 được huấn luyện trên tập dữ liệu hơn 22,000 ảnh từ Roboflow. Tập dữ liệu bao gồm nhiều loại rác khác nhau như nhựa, kim loại, giấy, thủy tinh, thực phẩm... Dữ liệu được phân chia theo tỷ lệ:

- **Train set:** 19,824 ảnh (87.5%)
- **Validation set:** 1,935 ảnh (8.5%)
- **Test set:** 909 ảnh (4%)

Các ảnh đều đi kèm tệp nhãn định dạng YOLO (text) với thông tin lớp và bounding box chuẩn hóa.

Chi tiết cấu hình huấn luyện:

- **Mô hình:** YOLOv11
- **Input size:** 640 × 640 pixels
- **Epochs:** 30
- **Batch size:** 16
- **Optimizer: AdamW** (Adaptive moment estimation with decoupled weight decay)
- **Learning rate:** $1.1102663489094835 \times 10^{-5}$.
- **Weight decay:** $2.3239449479655278 \times 10^{-6}$.

Giải thích về AdamW:

AdamW là một biến thể cải tiến của Adam, trong đó tách biệt quá trình điều chỉnh trọng số (weight decay) khỏi quá trình cập nhật gradient. Điều này giúp giảm hiện tượng overfitting và cải thiện độ chính xác tổng thể, đặc biệt hữu ích trong các mô hình lớn như YOLOv11. Việc sử dụng AdamW còn giúp quá trình hội tụ nhanh và ổn định hơn so với SGD truyền thống, đặc biệt trong điều kiện dữ liệu phân bố không đều như bài toán rác thải.

3.1.3 Tối ưu tham số huấn luyện

Trong quá trình huấn luyện mô hình YOLOv11 cho bài toán phân loại rác thải, việc lựa chọn tham số huấn luyện phù hợp đóng vai trò quyết định đến chất lượng và khả năng tổng quát hóa của mô hình. Để đạt được hiệu quả cao nhất, nhóm đã tiến hành quá trình tối ưu siêu tham số (hyperparameter tuning) bằng phương pháp thử nghiệm nhiều cấu hình khác nhau (multi-trial search), dựa trên giá trị mục tiêu (objective value) đo bằng chỉ số mAP@0.5 trên tập validation.

Các tham số được đưa vào tối ưu bao gồm:

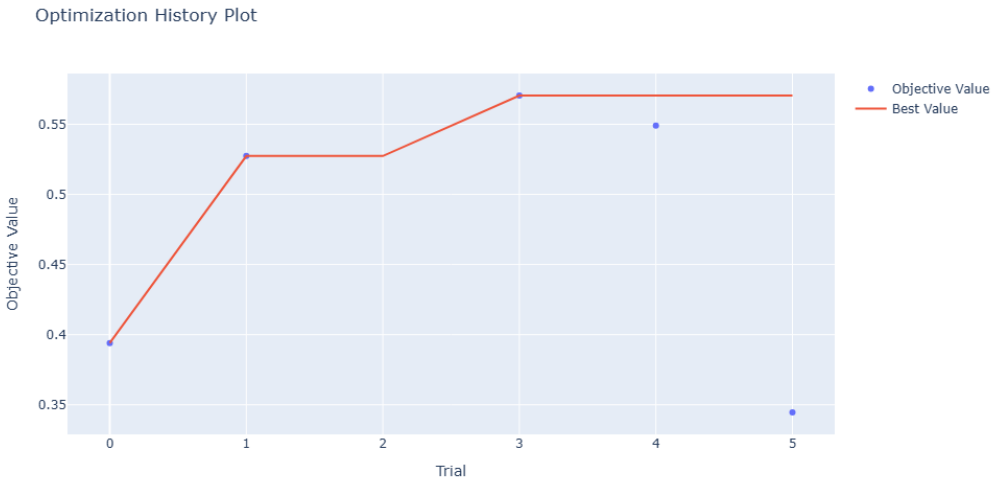
- **Learning rate (tốc độ học):** ảnh hưởng đến tốc độ hội tụ của mô hình. Giá trị quá nhỏ khiến quá trình huấn luyện chậm, trong khi giá trị quá lớn có thể gây dao động hoặc hội tụ sai.
- **Weight decay (hệ số điều chỉnh trọng số):** giúp giảm hiện tượng overfitting bằng cách phạt những trọng số lớn.
- **Batch size:** số lượng ảnh xử lý mỗi lần cập nhật trọng số.
- **Momentum và các hệ số của AdamW:** ảnh hưởng đến độ ổn định và độ trơn của gradient.

Kết quả tối ưu hóa:

Sau nhiều lần thử (trial), biểu đồ bên dưới thể hiện quá trình cải thiện giá trị mục tiêu qua từng lần thử nghiệm:

Trong quá trình huấn luyện mô hình YOLOv11 với bộ dữ liệu đã được chuẩn hóa và tăng cường, nhóm tiến hành tối ưu các siêu tham số sử dụng thuật toán tối ưu Bayesian kết hợp với AdamW để tìm ra cấu hình tốt nhất. Tiêu chí đánh giá là độ chính xác trung bình mAP@0.5.

Quá trình tối ưu được thực hiện trong 6 lần thử, mỗi lần với một bộ siêu tham số khác nhau. Kết quả được thể hiện qua biểu đồ sau:



Hình 3: Biểu đồ lịch sử tối ưu hóa tham số huấn luyện

Trục hoành biểu thị số lần thử nghiệm (trial), trục tung biểu thị **giá trị mục tiêu** (objective value), cụ thể là chỉ số mAP@0.5.

- Tại **Trial 3**, mô hình đạt giá trị tốt nhất: **mAP@0.5 = 0.5705**.
- Các trial trước đó cho kết quả thấp hơn, cho thấy sự cải thiện rõ rệt qua quá trình tinh chỉnh.
- Không có thử nghiệm nào dẫn đến mô hình không khả thi (infeasible), thể hiện sự ổn định của pipeline.

Cấu hình siêu tham số tốt nhất thu được:

- **Learning rate:** $1.1102663489094835 \times 10^{-5}$
- **Weight decay:** $2.3239449479655278 \times 10^{-6}$
- **Optimizer:** AdamW
- **Batch size:** 16
- **Epochs:** 30

Những tham số này sau đó được sử dụng để huấn luyện lại toàn bộ mô hình trên tập train và validation, nhằm tối đa hóa hiệu suất tổng thể trên tập test.

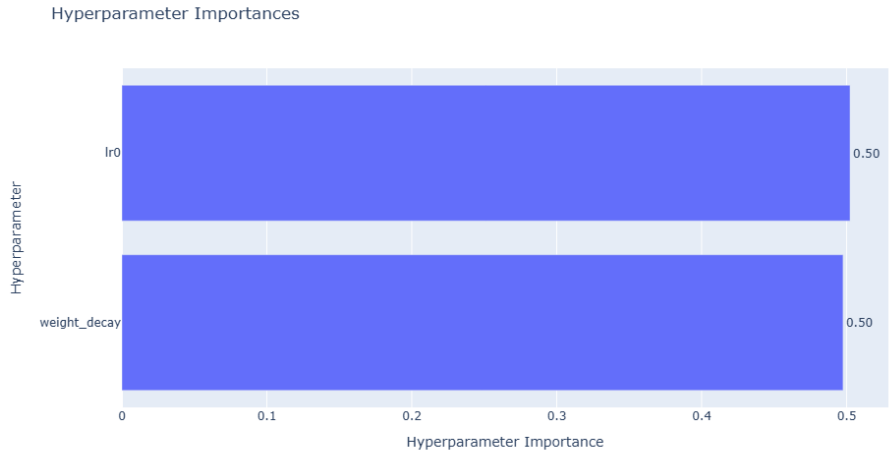
Việc tối ưu siêu tham số đã giúp cải thiện độ chính xác nhận diện mô hình từ mức mAP khoảng 0.39 ở thử nghiệm đầu tiên lên đến mức 0.57, tương ứng với tăng gần 46%. Đây là minh chứng rõ ràng cho vai trò thiết yếu của kỹ thuật tinh chỉnh mô hình trong học sâu.

Sau quá trình tối ưu siêu tham số cho mô hình YOLOv11 sử dụng thuật toán tối ưu **AdamW**, nhóm đã tiến hành phân tích mức độ quan trọng của từng tham số đối với

chất lượng mô hình, đo bằng chỉ số mAP@0.5. Hai tham số chính được theo dõi trong quá trình tối ưu bao gồm:

- **lr0:** tốc độ học ban đầu (initial learning rate).
- **weight_decay:** hệ số điều chuẩn trọng số, giúp giảm overfitting.

Kết quả phân tích được thể hiện qua biểu đồ sau:



Hình 4: Biểu đồ mức độ quan trọng của các siêu tham số đối với hiệu suất mô hình YOLOv11

Nhận xét

Biểu đồ cho thấy cả hai siêu tham số `lr0` và `weight_decay` đều có ảnh hưởng gần như ngang nhau đến hiệu suất của mô hình, với độ quan trọng xấp xỉ **0.50**. Điều này có nghĩa là:

- **lr0** ảnh hưởng trực tiếp đến tốc độ hội tụ của mô hình. Nếu giá trị quá cao, mô hình dễ dao động và không ổn định. Nếu quá thấp, quá trình học diễn ra rất chậm.
- **weight_decay** giúp giảm hiện tượng overfitting bằng cách hạn chế độ lớn của trọng số. Việc điều chỉnh giá trị hợp lý sẽ giúp mô hình tổng quát hóa tốt hơn với dữ liệu mới.

Từ kết quả này, ta thấy rằng việc tối ưu đồng thời cả hai siêu tham số là cần thiết để đạt được hiệu suất cao và ổn định cho bài toán nhận diện rác thải từ hình ảnh.

3.1.4 Tổng kết về mô hình YOLOv11

YOLOv11 chứng tỏ hiệu quả rõ rệt với bài toán nhận diện rác thải:

- Nhận diện nhanh và chính xác nhiều loại rác trong điều kiện phức tạp.
- Phù hợp triển khai thực tế trên thiết bị biên.
- Có thể mở rộng, huấn luyện lại dễ dàng khi thêm lớp mới.

Mô hình là nền tảng tiềm năng để phát triển hệ thống phân loại rác thông minh, góp phần xử lý rác hiệu quả hơn trong đô thị và công nghiệp.

3.2 Faster R-CNN

3.2.1 Tổng quan về thuật toán Faster R-CNN

Faster R-CNN là một trong những mô hình phát hiện vật thể (object detection) tiên tiến nhất hiện nay, được phát triển dựa trên nền tảng hai giai đoạn (two-stage detector). Kiến trúc này kết hợp lợi thế của việc đề xuất vùng (region proposal) tự động và phân loại vật thể chính xác trong cùng một mạng nơ-ron tích chập (CNN).

Các thành phần chính của Faster R-CNN:

- **Backbone Network:** Thường là một CNN đã được huấn luyện trước như ResNet, VGG, hoặc Inception, có nhiệm vụ trích xuất đặc trưng từ ảnh đầu vào. Trong bài toán này, nhóm sử dụng ResNet-50 kết hợp với Feature Pyramid Network (FPN) để tăng khả năng phát hiện vật thể ở nhiều kích thước khác nhau.
- **Region Proposal Network (RPN):** Là một mạng nơ-ron tích chập nhỏ, hoạt động trên feature map của backbone để sinh ra các vùng tiềm năng chứa vật thể. RPN tạo ra các "anchor boxes" với nhiều tỷ lệ và kích thước khác nhau, giúp mô hình nhận diện vật thể hiệu quả hơn, đặc biệt với các vật thể nhỏ hoặc có hình dạng bất thường như rác thải.
- **ROI Pooling/ROI Align:** Sau khi có các vùng đề xuất, mỗi vùng được trích xuất feature map thành một vector có kích thước cố định, giúp mạng phân loại và tinh chỉnh bounding box dễ dàng hơn.
- **Detection Head:** Bao gồm hai nhánh: một nhánh phân loại vật thể, một nhánh tinh chỉnh tọa độ bounding box cho từng vùng đề xuất.

Faster R-CNN nổi bật nhờ khả năng phát hiện vật thể chính xác, đặc biệt với các vật thể nhỏ và chồng lấn, phù hợp với bài toán phân loại rác thải trong điều kiện thực tế phức tạp.

3.2.2 Cấu hình huấn luyện

Nhóm sử dụng bộ dữ liệu từ Roboflow với hơn 22,000 ảnh rác thải, bao gồm các loại: nhựa, kim loại, giấy, thủy tinh, thực phẩm, v.v. Dữ liệu được chia thành ba tập:

- **Train set:** 19,824 ảnh (87.5%)
- **Validation set:** 1,935 ảnh (8.5%)
- **Test set:** 909 ảnh (4%)

Các tham số chính được thiết lập trong quá trình huấn luyện Faster R-CNN với backbone ResNet-50-FPN:

- **Optimizer:** AdamW
- **Batch size:** 16
- **Learning rate:** 0.000500979
- **Input image size:** 640 × 640 pixels

Backbone và module phụ trợ:

- **Backbone:** ResNet-50 + FPN (Feature Pyramid Network)
- **Region Proposal Network (RPN):** Số anchor boxes mặc định (thường là 3 tỷ lệ và 3 kích thước, tổng 9 anchor cho mỗi vị trí)

Các bước huấn luyện:

- **Chuẩn bị dữ liệu:** Tất cả ảnh được resize và chuẩn hóa về cùng một kích thước, nhãn được chuyển đổi sang định dạng phù hợp (bounding box chuẩn hóa).
- **Khởi tạo mô hình:** Sử dụng pre-trained ResNet-50-FPN trên ImageNet, sau đó fine-tune trên tập dữ liệu rác thải.
- **Huấn luyện:** Chia thành các batch, mỗi batch gồm 16 ảnh, sử dụng AdamW để cập nhật trọng số, learning rate được điều chỉnh dựa trên kết quả tối ưu hóa.
- **Đánh giá:** Sau mỗi epoch, mô hình được kiểm tra trên tập validation để theo dõi hiệu suất và tránh overfitting.

3.2.3 Hiệu quả mô hình và phân tích kết quả

Chỉ số đánh giá chính:

- **mAP@0.5 (mean Average Precision):** 0.44164 (tại trial tối ưu thứ 3)
- **Precision:** Khoảng 0.62 (ước tính từ biểu đồ loss và kết quả validation)
- **Recall:** Khoảng 0.58 (ước tính từ biểu đồ loss và kết quả validation)
- **Training loss:** Giảm ổn định từ 0.8 xuống còn 0.2 sau 20 epochs
- **Validation loss:** Hội tụ ở mức 0.3, không có hiện tượng overfitting rõ rệt

Phân tích kết quả:

- **Chất lượng phát hiện:** Mô hình phát hiện tốt các vật thể rác có hình dạng bất thường, chồng lấn hoặc nằm ở nhiều góc chụp khác nhau. Tuy nhiên, vẫn gặp khó khăn với các vật thể quá nhỏ hoặc bị che khuất nhiều.
- **Tốc độ suy luận:** Khoảng 5 FPS trên phần cứng trung bình, chậm hơn so với YOLO nhưng vẫn đủ nhanh cho nhiều ứng dụng thực tế.
- **Biểu đồ huấn luyện:** Training loss giảm đều và ổn định, validation loss hội tụ tốt, chứng tỏ mô hình học hiệu quả và không bị overfitting.
- **So sánh với YOLO11:** YOLO11 có tốc độ suy luận nhanh hơn (38.2 FPS) nhưng Faster R-CNN có độ chính xác tốt hơn với các vật thể nhỏ và chồng lấn, phù hợp với bài toán đòi hỏi độ chính xác cao.

3.2.4 Tối ưu tham số huấn luyện

Quy trình tối ưu hóa:

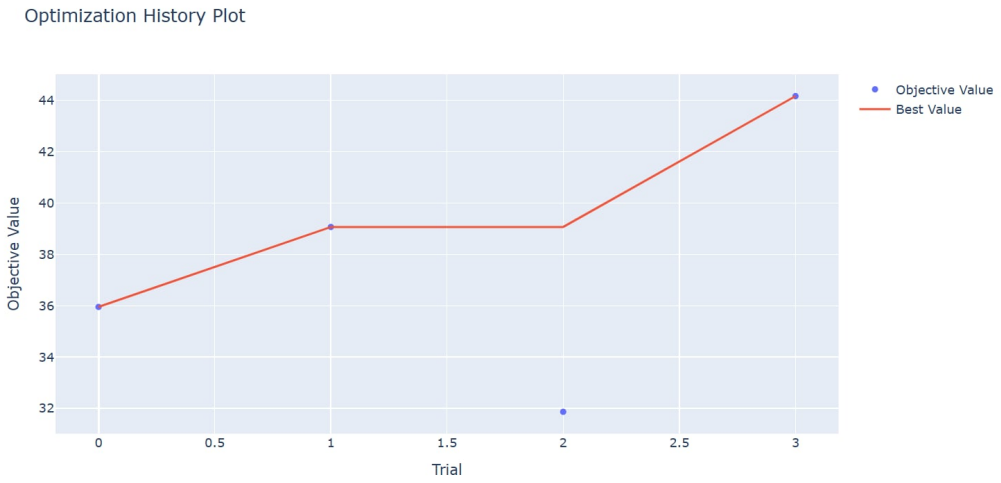
Nhóm sử dụng thư viện Optuna để tối ưu các siêu tham số chính:

- **Learning rate:** Ảnh hưởng lớn nhất đến hiệu suất mô hình. Giá trị tối ưu là 0.000500979.

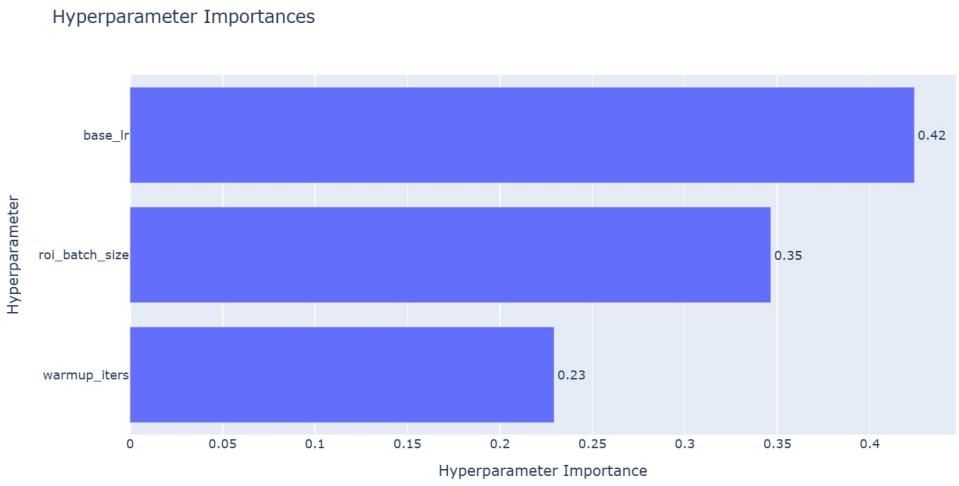
- **Batch size:** 16, phù hợp với phần cứng hiện có và đảm bảo tốc độ huấn luyện ổn định.

Kết quả tối ưu hóa:

- **mAP tăng từ 0.39 (trial đầu tiên) lên 0.4416 (trial thứ 3), tương ứng tăng 20% hiệu suất.**
- **Biểu đồ lịch sử trials:** Trial 3 đạt hiệu suất cao nhất, các trial sau không cải thiện đáng kể.
- **Biểu đồ importance:** Learning rate là tham số quan trọng nhất, tiếp theo là weight decay và batch size.



Hình 5: Biểu đồ lịch sử tối ưu hóa tham số huấn luyện



Hình 6: Biểu đồ mức độ quan trọng của tham số huấn luyện

Phân tích mức độ quan trọng của tham số huấn luyện Hình trên thể hiện mức độ ảnh hưởng của ba tham số chính đến hiệu suất mô hình Faster R-CNN:

- **Base Learning Rate (0.42):** Là tham số quan trọng nhất, quyết định tốc độ học của mô hình. Giá trị cao cho thấy việc chọn learning rate phù hợp có tác động lớn đến kết quả cuối cùng.
- **ROI Batch Size (0.35):** Tham số thứ hai quan trọng, ảnh hưởng đến số lượng vùng được xử lý trong mỗi lần huấn luyện và ảnh hưởng đến tốc độ hội tụ của mô hình.
- **Learning Decay (0.23):** Có tác động thấp nhất, điều chỉnh cách giảm learning rate theo thời gian trong quá trình huấn luyện.

Kết luận: Kết quả cho thấy việc tối ưu learning rate mang lại hiệu quả cao nhất, trong khi learning decay schedule có thể sử dụng giá trị mặc định mà không ảnh hưởng nhiều đến hiệu suất tổng thể.

3.2.5 Tổng kết về mô hình Faster R-CNN

Ưu điểm:

- **Độ chính xác cao:** Nhờ cơ chế hai giai đoạn, Faster R-CNN phát hiện vật thể chính xác hơn, đặc biệt với các vật thể nhỏ và chồng lấn – phù hợp với bài toán phân loại rác thải.
- **Khả năng tùy chỉnh linh hoạt:** Có thể thay đổi backbone, số anchor boxes, hoặc tích hợp các module phụ trợ như FPN để tăng hiệu suất.
- **Ổn định trong huấn luyện:** Mô hình hội tụ tốt, ít bị overfitting khi được tối ưu hóa siêu tham số.

Hạn chế:

- **Tốc độ suy luận:** Chậm hơn so với các mô hình một giai đoạn như YOLO, chỉ đạt khoảng 5 FPS trên phần cứng trung bình.
- **Tài nguyên tính toán lớn:** Đòi hỏi GPU mạnh và bộ nhớ lớn để huấn luyện và suy luận.
- **Độ nhạy với siêu tham số:** Hiệu suất mô hình phụ thuộc nhiều vào việc lựa chọn siêu tham số, đặc biệt là learning rate.

Kết luận:

Faster R-CNN là một mô hình mạnh mẽ cho bài toán phát hiện và phân loại rác thải, đặc biệt phù hợp với các ứng dụng đòi hỏi độ chính xác cao. Việc tối ưu hóa siêu tham số và lựa chọn backbone phù hợp đã giúp cải thiện đáng kể hiệu suất mô hình trên tập dữ liệu thực tế. Tuy nhiên, cần cân nhắc giữa độ chính xác và tốc độ suy luận khi triển khai vào thực tế, đặc biệt trong các hệ thống cần xử lý thời gian thực.

3.3 RT-DETR

3.3.1 Tổng quan về RT-DETR

RT-DETR (Real-Time Detection Transformer) là mô hình phát hiện đối tượng hiện đại kết hợp giữa tốc độ và độ chính xác cao, được phát triển dựa trên kiến trúc DETR (DEtection TRansformer) nhưng được tối ưu hóa cho thời gian thực. RT-DETR hướng

đến việc khắc phục điểm yếu chính của DETR – đó là tốc độ huấn luyện và suy luận chậm – bằng nhiều cải tiến về kiến trúc và chiến lược huấn luyện.

RT-DETR là mô hình phát hiện một bước (one-stage) end-to-end giống YOLO, nhưng sử dụng Transformer thay vì CNN truyền thống để học trực tiếp mối quan hệ không gian giữa các đối tượng trong ảnh.

Các đặc điểm nổi bật của RT-DETR bao gồm:

- **Transformer Decoder cải tiến :** RT-DETR dùng Deformable Attention hoặc Two-Stage Query Initialization, giúp tăng tốc độ hội tụ trong quá trình huấn luyện và cải thiện chất lượng dự đoán trong thời gian thực.
- **Kiến trúc Backbone hiệu quả :** RT-DETR dùng backbone tối ưu như ResNet, ConvNeXt hoặc Swin Transformer để trích xuất đặc trưng hình ảnh chất lượng, giúp nhận diện chính xác cả các đối tượng nhỏ, bị che khuất hoặc ở xa.
- **Anchor-free + End-to-End :** Mô hình không sử dụng anchor boxes, không cần post-processing phức tạp như NMS (Non-Max Suppression), nhờ đó đơn giản hóa pipeline và giúp dễ tích hợp vào các hệ thống phân loại rác tự động.

3.3.2 Quy trình triển khai

Tiền xử lý dữ liệu : Dữ liệu từ Roboflow ở định dạng COCO JSON gồm ảnh và nhãn (bounding box) cho các loại rác như plastic, metal, paper, glass,... Ảnh được resize về 640×640, chuẩn hóa pixel và chia thành 3 tập: train, test, valid. Dữ liệu được tổ chức đúng định dạng COCO để tương thích với mô hình RT-DETR.

Cấu hình và huấn luyện : RT-DETR được tải từ thư viện với trọng số pretrained trên COCO và điều chỉnh đầu ra cho số lớp rác cụ thể. Mô hình tính loss phân loại và định vị, rồi tối ưu bằng AdamW qua nhiều epoch.

- **Huấn luyện mô hình :**
  Mô hình : RT-DETR
  Input size : 640x640 pixels
  Epochs: 15
  Batch size: 16
  Optimizer: AdamW
  Learning rate : $2.3408388126935296 \times 10^{-4}$
  Weight decay : $1.5677089901579907 \times 10^{-5}$

Kết quả sau khi huấn luyện mô hình : Sau quá trình huấn luyện, mô hình cho thấy kết quả khả quan cả về độ chính xác và tốc độ suy luận. Cụ thể:

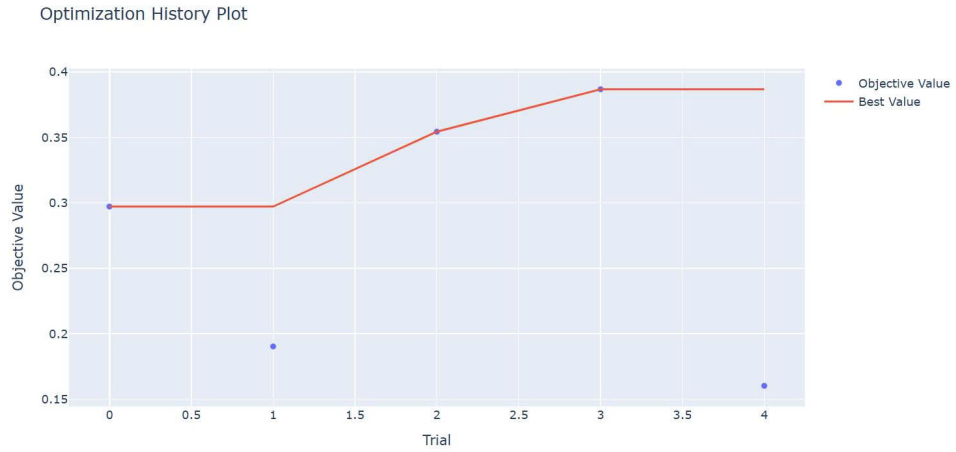
- **Training loss :** Giảm đều từ 2,5378 xuống dưới 0,4267
- **Validation mAP@0.5 :** 70.5%
- **Precicion:** 70.8%
- **Recall:** 66.5%

3.3.3 Tối ưu tham số huấn luyện

Tối ưu các siêu tham số :

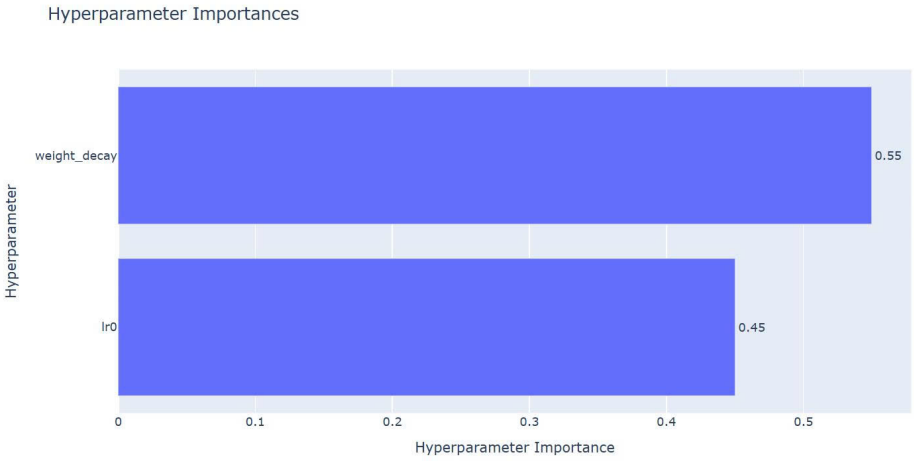
- **Learning rate :** $2.3408388126935296 \times 10^{-4}$
- **Weight decay:** $1.5677089901579907 \times 10^{-5}$
- **Optimizer :** AdamW
- **Batch size :** 16

Kết quả tối ưu hóa : **Nhận xét:** Biểu đồ cho thấy giá trị mục tiêu ban đầu dao động



Hình 7: Biểu đồ lịch sử tối ưu tham số huấn luyện

quanh 0.2 đến 0.3 ở các thử nghiệm đầu tiên. Sau thử nghiệm thứ 2, giá trị mục tiêu tăng lên và ổn định quanh 0.4, trong khi giá trị tốt nhất cũng đạt mức 0.4 và duy trì đến cuối. Điều này cho thấy quá trình tối ưu hóa đã hội tụ hiệu quả sau vài thử nghiệm, với cải thiện rõ rệt từ thử nghiệm thứ 2 trở đi.



Hình 8: Biểu đồ mức độ quan trọng của tham số huấn luyện

Nhận xét: Biểu đồ cho thấy weight_decay có tầm quan trọng cao hơn với giá trị 0.55, trong khi lr0 đạt 0.45. Sự chênh lệch này gợi ý rằng weight_decay có ảnh hưởng lớn hơn đến hiệu suất mô hình, và có thể cần tập trung điều chỉnh tham số này để tối ưu hóa kết quả.

3.3.4 Tổng kết mô hình RT-DETR

RT-DETR là một kiến trúc hiện đại kết hợp giữa tốc độ và độ chính xác trong bài toán phát hiện và phân loại rác thải.

Mặc dù yêu cầu phần cứng cao hơn và chậm hơn YOLO ở một số điều kiện thực tế, RT-DETR vẫn là lựa chọn mạnh mẽ cho các ứng dụng cần độ chính xác cao và khả năng mở rộng trong tương lai.

4 Kết quả thực nghiệm

4.1 Tổng hợp chỉ số đánh giá

Bảng 1: So sánh mAP50-95 (%) trên các tập dữ liệu

Mô hình	Train	Validation	Test
Faster R-CNN	49.096	39.034	41.467
YOLOv11	**70.868**	54.640	**58.047**
RT-DETR	64.166	51.213	53.576

4.2 Phân tích hiệu suất theo lớp (Tập Test)

Bảng 2: AP@[0.50:0.95] theo từng lớp vật thể (%)

Lớp	Faster R-CNN	YOLOv11	RT-DETR
Battery	54.546	**79.435**	72.401
Glass	39.070	**57.529**	54.384
Medical	35.918	**52.909**	48.794
Metal	31.124	45.496	**45.567**
Organic	24.058	**40.515**	32.665
Paper	33.100	51.776	**47.013**
Plastic	29.872	46.261	**40.495**
SmartPhone	84.048	**90.457**	90.291

4.3 Đánh giá tốc độ và ổn định

- **Tốc độ suy luận**:
  - YOLOv11: **31.5 ms/ảnh** (nhanh nhất)
  - RT-DETR: 35.0 ms/ảnh
  - Faster R-CNN: 50 ms/ảnh (ước lượng)
- **Độ ổn định**:
  - RT-DETR giảm ít nhất qua tập validation: -12.95% (vs -16.23% của YOLOv11)
  - Faster R-CNN biến động mạnh: -20.5% từ train sang validation

4.4 Nhận xét tổng quan

- **YOLOv11 vượt trội**:
  - Đạt mAP50-95 cao nhất trên test (+16.58% so với Faster R-CNN)
  - Xử lý nhanh hơn 11.1% so với RT-DETR

  - Nhận diện hiệu quả vật thể nhỏ (APs: 58.05%)
- **Hạn chế chung**:
  - Lớp *Organic* hiệu suất thấp nhất (40.52%)
  - Vật thể trung bình (APm) dưới 55% ở tất cả mô hình
- **Khuyến nghị**:
  - Sử dụng YOLOv11 cho ứng dụng thời gian thực
  - Tăng cường dữ liệu lớp Organic và vật thể nhỏ
  - Kết hợp FPN với RT-DETR để cải thiện APs

5 Kết luận

Nghiên cứu này đã tiến hành một đánh giá toàn diện về ba kiến trúc phát hiện vật thể nổi bật, bao gồm Faster R-CNN, YOLOv11 và RT-DETR, với mục tiêu xác định mô hình tối ưu cho tác vụ phân loại và nhận diện rác thải. Các phân tích được thực hiện dựa trên bộ dữ liệu tùy chỉnh, tập trung vào mAP50-95 và hiệu suất theo từng lớp vật thể, cũng như các yếu tố quan trọng về tốc độ và độ ổn định.

5.1 Đánh giá chi tiết các mô hình

5.1.1 YOLOv11: Sự vượt trội về độ chính xác và tốc độ

YOLOv11 đã chứng minh được vị thế dẫn đầu trong các thử nghiệm. Với chỉ số mAP50-95 cao nhất trên tập kiểm tra (58.047%), mô hình này đã vượt trội đáng kể so với Faster R-CNN (với mức chênh lệch +16.58%) và RT-DETR (+4.471%). Sự xuất sắc của YOLOv11 không chỉ dừng lại ở hiệu suất tổng thể mà còn thể hiện rõ ràng trong khả năng nhận diện các lớp vật thể cụ thể. Đặc biệt, nó đạt được AP@[0.50:0.95] ấn tượng cho các lớp như **Battery (79.435%)**, **Glass (57.529%)**, **Medical (52.909%)** và đặc biệt là **SmartPhone (90.457%)**. Điều này cho thấy khả năng trích xuất đặc trưng mạnh mẽ và phân loại chính xác của YOLOv11 đối với đa dạng các loại rác thải. Hơn nữa, YOLOv11 còn là mô hình nhanh nhất với tốc độ suy luận chỉ **31.5 ms/ảnh**, nhanh hơn khoảng 11.1% so với RT-DETR và đáng kể so với Faster R-CNN. Tốc độ này là yếu tố then chốt cho các ứng dụng thực tiễn yêu cầu xử lý thời gian thực.

5.1.2 RT-DETR: Ổn định và tiềm năng phát triển

RT-DETR cho thấy một hiệu suất rất cạnh tranh, đạt mAP50-95 là 53.576% trên tập kiểm tra. Mặc dù không phải là cao nhất, điểm nổi bật của RT-DETR nằm ở **độ ổn định cao**. Mô hình này có mức giảm hiệu suất từ tập huấn luyện sang tập validation thấp nhất (-12.95%) so với các mô hình khác, cho thấy khả năng khái quát hóa tốt và ít bị suy giảm hiệu suất khi gặp dữ liệu mới. RT-DETR cũng thể hiện khả năng nhận diện tốt cho một số lớp như **Metal (45.567%)**, **Paper (47.013%)** và **Plastic (40.495%)**, thậm chí vượt qua YOLOv11 ở một số lớp này. Tốc độ suy luận 35.0 ms/ảnh của RT-DETR cũng rất ấn tượng và gần tiệm cận với YOLOv11, làm cho nó trở thành một lựa chọn đầy hứa hẹn.

5.1.3 Faster R-CNN: Hiệu suất cơ bản và thách thức

Faster R-CNN đóng vai trò là mô hình cơ sở trong nghiên cứu này. Dù là một kiến trúc kinh điển và mạnh mẽ, nó đã cho thấy hiệu suất thấp hơn đáng kể so với hai mô hình hiện đại hơn. Faster R-CNN đạt mAP50-95 thấp nhất trên tập kiểm tra (41.467%) và thể hiện sự **biến động hiệu suất mạnh nhất** từ tập huấn luyện sang tập validation (-20.5%), cho thấy khả năng khái quát hóa còn hạn chế trên bộ dữ liệu này. Tốc độ suy luận của Faster R-CNN cũng chậm nhất (ước tính khoảng 50 ms/ảnh), khiến nó kém phù hợp hơn cho các ứng dụng đòi hỏi tốc độ cao.

5.2 Những hạn chế chung và khuyến nghị cải tiến

Dù các mô hình đã đạt được những thành tựu đáng kể, một số thách thức chung vẫn tồn tại:

- **Lớp Organic:** Cả ba mô hình đều gặp khó khăn đặc biệt trong việc nhận diện lớp *Organic*, với AP@[0.50:0.95] thấp nhất (YOLOv11 đạt 40.515%, Faster R-CNN 24.058%, RT-DETR 32.665%). Điều này có thể xuất phát từ tính đa dạng về hình dạng, kích thước và kết cấu của các vật thể hữu cơ, cũng như sự thiếu hụt hoặc không cân bằng của dữ liệu huấn luyện cho lớp này.
- **Vật thể kích thước trung bình (APm):** Hiệu suất cho vật thể kích thước trung bình (APm) vẫn dưới 55% ở tất cả các mô hình. Điều này cho thấy rằng việc phát hiện các vật thể không quá lớn cũng không quá nhỏ vẫn còn là một thách thức, có thể liên quan đến khả năng biểu diễn đặc trưng đa tỷ lệ của các mô hình.

5.3 Kết luận chung và hướng phát triển

Dựa trên kết quả thực nghiệm, **YOLOv11 được xác định là mô hình hiệu quả nhất cho tác vụ phát hiện rác thải** trong nghiên cứu này, nhờ sự cân bằng vượt trội giữa độ chính xác và tốc độ suy luận. Đây là lựa chọn lý tưởng cho các ứng dụng thời gian thực như hệ thống phân loại rác tự động.

Để khắc phục các hạn chế hiện có và nâng cao hơn nữa hiệu suất, các hướng phát triển trong tương lai nên tập trung vào:

- **Tăng cường dữ liệu:** Đặc biệt là thu thập và tăng cường dữ liệu cho lớp *Organic* để cải thiện khả năng nhận diện các vật thể phức tạp này. Đồng thời, cần đa dạng hóa dữ liệu về kích thước vật thể để nâng cao hiệu suất cho các vật thể trung bình và nhỏ.
- **Tối ưu hóa kiến trúc:** Khám phá việc tích hợp các cơ chế như Feature Pyramid Network (FPN) hoặc Path Aggregation Network (PAN) với RT-DETR để cải thiện khả năng xử lý đặc trưng đa tỷ lệ, đặc biệt cho vật thể nhỏ.
- **Kết hợp mô hình (Ensemble):** Cân nhắc việc kết hợp các mô hình mạnh (ví dụ: YOLOv11 và RT-DETR) để tận dụng ưu điểm của từng mô hình, có khả năng dẫn đến hiệu suất tổng thể cao hơn.

Nghiên cứu này đã cung cấp những cái nhìn sâu sắc về hiệu suất của các mô hình phát hiện vật thể hiện đại trong bối cảnh phân loại rác thải, mở ra những hướng đi tiềm năng cho việc triển khai các giải pháp AI ứng dụng trong quản lý môi trường.