

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
Đại Học Quốc Gia Hà Nội



## Phân loại rác thải trong gia đình

Vũ Nguyễn Quỳnh Anh  
Nguyễn Văn Duy  
Đoàn Thị Minh Khuê

Mã học phần: MAT3508  
Học kỳ 1, Năm học 2025-2026

# Thông tin Dự án

[Thông tin này cũng cần được ghi trong README.md của kho GitHub.]

**Học phần:** MAT3508 – Nhập môn Trí tuệ Nhân tạo

**Học kỳ:** Học kỳ 1, Năm học 2025-2026

**Trường:** VNU-HUS (Đại học Quốc gia Hà Nội – Trường Đại học Khoa học Tự nhiên)

**Tên dự án:** Phân loại rác thải trong gia đình

**Ngày nộp:** 30/11/2025

**Báo cáo PDF:** <https://github.com/doanminhkhue/IntroAI-GC/tree/main/report>

**Slide thuyết trình:** <https://github.com/doanminhkhue/IntroAI-GC/tree/main/slide>

**Kho GitHub:** [https://github.com/doanminhkhue/IntroAI\\_garbage\\_classification](https://github.com/doanminhkhue/IntroAI_garbage_classification)

## Thành viên nhóm

| Họ tên              | Mã sinh viên | Tên GitHub    |
|---------------------|--------------|---------------|
| Vũ Nguyễn Quỳnh Anh | 23001830     | quynhanh_2610 |
| Nguyễn Văn Duy      | 23001854     | nvndy27       |
| Đoàn Thị Minh Khuê  | 23001894     | doanminhkhue  |

## Phân công công việc

| Họ tên              | Nhiệm vụ                                 |
|---------------------|------------------------------------------|
| Vũ Nguyễn Quỳnh Anh | Giới thiệu, cơ sở lý thuyết, tiền xử lý  |
| Nguyễn Văn Duy      | Huấn luyện, triển khai mô hình, kết luận |
| Đoàn Thị Minh Khuê  | Chia dữ liệu, thực nghiệm và đánh giá    |

# Danh sách hình vẽ

|     |                                                       |    |
|-----|-------------------------------------------------------|----|
| 1.1 | Minh họa về phân loại rác thải sinh hoạt . . . . .    | 7  |
| 3.1 | Hình minh họa lớp glass(thủy tinh) . . . . .          | 17 |
| 3.2 | Hình minh họa lớp metal(kim loại) . . . . .           | 17 |
| 3.3 | Hình minh họa lớp organic(rác hữu cơ) . . . . .       | 18 |
| 4.1 | Ma trận nhầm lẫn của mô hình MobileNetV2 . . . . .    | 23 |
| 4.2 | Ma trận nhầm lẫn của mô hình EfficientNetB0 . . . . . | 24 |
| 4.3 | Ma trận nhầm lẫn của mô hình YOLOv8 . . . . .         | 25 |
| D.1 | Minh họa ứng dụng camera <sub>demo</sub> . . . . .    | 37 |
| D.2 | Minh họa ứng dụng Streamlit . . . . .                 | 38 |

# Danh sách bảng

|     |                                                          |    |
|-----|----------------------------------------------------------|----|
| 3.1 | Thống kê chi tiết phân bố dữ liệu cho từng lớp . . . . . | 18 |
| 4.1 | Bảng so sánh hiệu năng giữa các mô hình . . . . .        | 26 |
| B.1 | Tham số huấn luyện các mô hình . . . . .                 | 35 |

# Mục lục

|                                                                                                  |           |
|--------------------------------------------------------------------------------------------------|-----------|
| <b>Lời mở đầu</b>                                                                                | <b>6</b>  |
| <b>1 GIỚI THIỆU</b>                                                                              | <b>7</b>  |
| 1.1 Lý do chọn đề tài . . . . .                                                                  | 7         |
| 1.2 Các công trình nghiên cứu liên quan đến phân loại rác thải sinh hoạt trên thế giới . . . . . | 8         |
| 1.2.1 Các phương pháp học máy truyền thống . . . . .                                             | 8         |
| 1.2.2 Các phương pháp học sâu dựa trên CNN . . . . .                                             | 8         |
| 1.2.3 Các nghiên cứu hướng tới ứng dụng thực tế . . . . .                                        | 8         |
| 1.2.4 Nhận xét chung . . . . .                                                                   | 8         |
| 1.3 Mục tiêu của đề tài . . . . .                                                                | 9         |
| 1.4 Đối tượng và phạm vi nghiên cứu . . . . .                                                    | 9         |
| 1.4.1 Đối tượng nghiên cứu . . . . .                                                             | 9         |
| 1.4.2 Phạm vi nghiên cứu . . . . .                                                               | 9         |
| 1.5 Phương pháp nghiên cứu . . . . .                                                             | 9         |
| 1.6 Bố cục báo cáo . . . . .                                                                     | 10        |
| <b>2 CƠ SỞ LÝ THUYẾT</b>                                                                         | <b>11</b> |
| 2.1 Tổng quan về Trí tuệ nhân tạo . . . . .                                                      | 11        |
| 2.2 Học máy và Học sâu . . . . .                                                                 | 11        |
| 2.2.1 Học máy . . . . .                                                                          | 11        |
| 2.2.2 Học sâu . . . . .                                                                          | 11        |
| 2.3 Mạng nơ-ron tích chập . . . . .                                                              | 12        |
| 2.4 Bài toán phân loại ảnh và phát hiện đối tượng . . . . .                                      | 12        |
| 2.4.1 Phân loại ảnh . . . . .                                                                    | 12        |
| 2.4.2 Phát hiện đối tượng . . . . .                                                              | 13        |
| 2.4.3 Sự khác nhau trong ngữ cảnh phân loại rác . . . . .                                        | 13        |
| 2.5 Giới thiệu các mô hình sử dụng trong đề tài . . . . .                                        | 13        |
| 2.5.1 MobileNetV2 . . . . .                                                                      | 13        |
| 2.5.2 EfficientNetB0 . . . . .                                                                   | 14        |
| 2.5.3 YOLOv8 . . . . .                                                                           | 14        |
| <b>3 XÂY DỰNG HỆ THỐNG PHÂN LOẠI RÁC</b>                                                         | <b>16</b> |
| 3.1 Thu thập và Xây dựng bộ dữ liệu . . . . .                                                    | 16        |
| 3.1.1 Nguồn dữ liệu . . . . .                                                                    | 16        |
| 3.1.2 Phân bổ và Tổ chức dữ liệu . . . . .                                                       | 16        |
| 3.1.3 Tiền xử lý và tăng cường dữ liệu . . . . .                                                 | 18        |
| 3.2 Xây dựng và huấn luyện mô hình MobileNetV2 . . . . .                                         | 19        |
| 3.2.1 Thiết kế kiến trúc mô hình . . . . .                                                       | 19        |
| 3.2.2 Tham số huấn luyện . . . . .                                                               | 19        |
| 3.3 Xây dựng và huấn luyện mô hình EfficientNetB0 . . . . .                                      | 20        |

|                           |                                                           |           |
|---------------------------|-----------------------------------------------------------|-----------|
| 3.3.1                     | Thiết kế kiến trúc mô hình . . . . .                      | 20        |
| 3.3.2                     | Chiến lược huấn luyện . . . . .                           | 20        |
| 3.4                       | Xây dựng và huấn luyện mô hình YOLOv8 . . . . .           | 21        |
| 3.4.1                     | Kiến trúc và Cấu hình . . . . .                           | 21        |
| 3.4.2                     | Tham số huấn luyện . . . . .                              | 21        |
| <b>4</b>                  | <b>ĐÁNH GIÁ VÀ SO SÁNH KẾT QUẢ</b>                        | <b>22</b> |
| 4.1                       | Môi trường thực nghiệm . . . . .                          | 22        |
| 4.2                       | Các tiêu chí đánh giá . . . . .                           | 22        |
| 4.3                       | Kết quả huấn luyện các mô hình . . . . .                  | 22        |
| 4.3.1                     | Kết quả với MobileNetV2 . . . . .                         | 22        |
| 4.3.2                     | Kết quả với EfficientNetB0 . . . . .                      | 24        |
| 4.3.3                     | Kết quả với YOLOv8 . . . . .                              | 25        |
| 4.4                       | So sánh các mô hình . . . . .                             | 26        |
| 4.5                       | Nhận xét và đánh giá . . . . .                            | 27        |
| <b>5</b>                  | <b>XÂY DỰNG VÀ TRIỂN KHAI ỨNG DỤNG PHÂN LOẠI RÁC THẢI</b> | <b>28</b> |
| 5.1                       | Mục tiêu triển khai ứng dụng . . . . .                    | 28        |
| 5.2                       | Ứng dụng phân loại rác bằng camera real-time . . . . .    | 28        |
| 5.2.1                     | Mô tả tổng quan . . . . .                                 | 28        |
| 5.2.2                     | Quy trình xử lý . . . . .                                 | 28        |
| 5.2.3                     | Ứng dụng camera sử dụng MobileNetV2 . . . . .             | 29        |
| 5.2.4                     | Ứng dụng camera sử dụng EfficientNetB0 . . . . .          | 29        |
| 5.3                       | Ứng dụng phân loại rác sử dụng Streamlit . . . . .        | 29        |
| 5.3.1                     | Giới thiệu Streamlit . . . . .                            | 29        |
| 5.3.2                     | Chức năng của ứng dụng . . . . .                          | 29        |
| 5.3.3                     | Ứng dụng Streamlit sử dụng MobileNetV2 . . . . .          | 29        |
| 5.3.4                     | Ứng dụng Streamlit sử dụng EfficientNetB0 . . . . .       | 29        |
| 5.4                       | Dánh giá khả năng ứng dụng thực tế . . . . .              | 30        |
| <b>6</b>                  | <b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>                       | <b>31</b> |
| 6.1                       | Kết luận . . . . .                                        | 31        |
| 6.2                       | Hạn chế của đề tài . . . . .                              | 31        |
| 6.3                       | Hướng phát triển . . . . .                                | 32        |
| <b>Tài liệu tham khảo</b> |                                                           | <b>32</b> |
| <b>A</b>                  | <b>Cấu trúc thư mục dự án</b>                             | <b>34</b> |
| <b>B</b>                  | <b>Tham số huấn luyện mô hình</b>                         | <b>35</b> |
| <b>C</b>                  | <b>Kết quả đánh giá</b>                                   | <b>36</b> |
| C.1                       | Kết quả đánh giá MobileNetV2 . . . . .                    | 36        |
| C.2                       | Kết quả đánh giá EfficientNetB0 . . . . .                 | 36        |
| C.3                       | Kết quả đánh giá YOLOv8 . . . . .                         | 36        |
| <b>D</b>                  | <b>Mô tả ứng dụng triển khai</b>                          | <b>37</b> |

# Lời mở đầu

Trong bối cảnh khoa học và công nghệ phát triển mạnh mẽ, Trí tuệ nhân tạo (Artificial Intelligence – AI) ngày càng khẳng định vai trò quan trọng trong nhiều lĩnh vực của đời sống xã hội như y tế, giáo dục, giao thông, sản xuất và bảo vệ môi trường. Việc ứng dụng AI không chỉ giúp nâng cao hiệu quả công việc mà còn góp phần giải quyết nhiều bài toán thực tiễn phức tạp mà con người gặp phải.

Một trong những vấn đề nổi bật hiện nay là tình trạng ô nhiễm môi trường do rác thải sinh hoạt ngày càng gia tăng. Việc phân loại rác không hiệu quả ngay từ nguồn gây khó khăn cho quá trình xử lý, tái chế và làm gia tăng tác động tiêu cực đến môi trường sống. Do đó, nghiên cứu và phát triển các hệ thống hỗ trợ phân loại rác thải tự động đang trở thành một hướng đi cần thiết và có ý nghĩa thực tiễn cao.

Thông qua học phần *Nhập môn Trí tuệ nhân tạo*, nhóm thực hiện đề tài mong muốn vận dụng các kiến thức đã học về học máy, học sâu và thị giác máy tính để xây dựng một hệ thống phân loại rác thải sinh hoạt gia đình dựa trên hình ảnh. Đề tài tập trung nghiên cứu và so sánh hiệu quả của ba mô hình học sâu phổ biến là MobileNetV2, EfficientNetB0 và YOLOv8 nhằm đánh giá khả năng ứng dụng của từng mô hình trong các điều kiện khác nhau.

Mặc dù đã có nhiều cố gắng trong quá trình thực hiện, báo cáo không tránh khỏi những hạn chế do thời gian, kinh nghiệm và tài nguyên còn hạn chế. Nhóm thực hiện rất mong nhận được những ý kiến đóng góp quý báu từ giảng viên để đề tài được hoàn thiện hơn và có thể phát triển trong các nghiên cứu tiếp theo.

Cuối cùng, nhóm xin chân thành cảm ơn các giảng viên hướng dẫn đã tận tình chỉ bảo, hỗ trợ trong suốt quá trình học tập và thực hiện đề tài.

*Xin chân thành cảm ơn!*

# Chương 1

## GIỚI THIỆU

### 1.1 Lý do chọn đề tài

Trong những năm gần đây, ô nhiễm môi trường đang trở thành một trong những vấn đề nghiêm trọng và cấp bách trên phạm vi toàn cầu. Trong đó, rác thải sinh hoạt phát sinh hằng ngày từ các hộ gia đình chiếm một tỷ lệ lớn và gây áp lực không nhỏ lên hệ thống xử lý rác thải. Việc xử lý rác không hiệu quả có thể dẫn đến ô nhiễm đất, nước và không khí, ảnh hưởng trực tiếp đến sức khỏe con người và hệ sinh thái.

Một trong những giải pháp quan trọng nhằm giảm thiểu tác động tiêu cực của rác thải sinh hoạt là phân loại rác ngay từ nguồn. Tuy nhiên, trên thực tế, việc phân loại rác thủ công tại các hộ gia đình vẫn gặp nhiều khó khăn do thiếu kiến thức, ý thức của người dân cũng như sự bất tiện trong quá trình thực hiện. Điều này khiến cho hiệu quả phân loại rác chưa cao và làm tăng chi phí xử lý rác ở các khâu sau.



Hình 1.1: Minh họa về phân loại rác thải sinh hoạt

Với sự phát triển mạnh mẽ của Trí tuệ nhân tạo (Artificial Intelligence – AI), đặc biệt là lĩnh vực Thị giác máy tính (Computer Vision), việc tự động hóa quá trình phân loại rác dựa trên hình ảnh đã trở thành một hướng nghiên cứu tiềm năng. Các mô hình học sâu có khả năng học và nhận diện đặc trưng của từng loại rác với độ chính xác cao, từ đó hỗ trợ hoặc thay thế con người trong công tác phân loại.

Xuất phát từ những vấn đề trên, đề tài “*Phân loại rác thải sinh hoạt trong gia đình*” được lựa chọn với mong muốn nghiên cứu, xây dựng và đánh giá một hệ thống phân loại rác tự động, có tính ứng dụng thực tiễn cao, phù hợp với định hướng phát triển bền vững và bảo vệ môi trường.

## 1.2 Các công trình nghiên cứu liên quan đến phân loại rác thải sinh hoạt trên thế giới

Trong những năm gần đây, bài toán phân loại rác thải sinh hoạt đã thu hút sự quan tâm lớn của cộng đồng nghiên cứu do vai trò quan trọng của nó trong việc nâng cao hiệu quả tái chế và hướng tới phát triển bền vững. Nhiều công trình trên thế giới đã đề xuất các phương pháp khác nhau, từ các thuật toán học máy truyền thống cho đến các mô hình học sâu hiện đại, nhằm tự động hóa quá trình phân loại rác dựa trên hình ảnh.

### 1.2.1 Các phương pháp học máy truyền thống

Trước khi các mô hình học sâu trở nên phổ biến, nhiều nghiên cứu đã áp dụng các thuật toán học máy truyền thống để phân loại rác thải. Các phương pháp này thường dựa trên việc trích xuất đặc trưng thủ công từ hình ảnh, sau đó sử dụng các bộ phân loại như *Support Vector Machine (SVM)*, *Decision Tree*, *Random Forest* hoặc *k-Nearest Neighbors (k-NN)*.

Một số nghiên cứu cho thấy SVM có thể đạt độ chính xác tương đối tốt trong phân loại rác tái chế như giấy, nhựa, kim loại và thủy tinh khi tập dữ liệu không quá lớn. Tuy nhiên, các phương pháp này phụ thuộc nhiều vào chất lượng đặc trưng được thiết kế thủ công và thường gặp hạn chế khi xử lý dữ liệu phức tạp hoặc đa dạng trong môi trường thực tế.

### 1.2.2 Các phương pháp học sâu dựa trên CNN

Với sự phát triển mạnh mẽ của học sâu, đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks – CNN), nhiều công trình đã đạt được kết quả vượt trội trong bài toán phân loại rác thải sinh hoạt. Các kiến trúc CNN phổ biến như *VGG*, *ResNet*, *EfficientNet* và *MobileNet* thường được sử dụng dưới dạng mô hình huấn luyện từ đầu hoặc mô hình tiền huấn luyện kết hợp với kỹ thuật *transfer learning*.

Nghiên cứu của Gupta (2020) đã xây dựng hệ thống phân loại ảnh rác thải dựa trên tập dữ liệu TACO, sử dụng các mô hình CNN tiền huấn luyện như ResNet-50 và VGG-19. Kết quả cho thấy các mô hình học sâu tiền huấn luyện đạt độ chính xác cao hơn đáng kể so với các phương pháp học máy truyền thống, đặc biệt trong bối cảnh dữ liệu có nhiều lớp và sự đa dạng cao.

### 1.2.3 Các nghiên cứu hướng tới ứng dụng thực tế

Bên cạnh việc cải thiện độ chính xác phân loại, nhiều công trình gần đây tập trung vào khả năng triển khai thực tế của hệ thống. Một số nghiên cứu kết hợp các mô hình học sâu với hệ thống quản lý rác thông minh, hướng tới việc tự động hóa quá trình phân loại tại nguồn hoặc tại các trạm xử lý rác.

Chẳng hạn, các nghiên cứu ứng dụng mô hình phát hiện và phân loại đối tượng như YOLO đã chứng minh hiệu quả trong việc vừa nhận diện vừa phân loại rác theo thời gian thực. Ngoài ra, một số công trình còn gắn kết kết quả phân loại với các quy chuẩn quản lý rác của từng quốc gia, nhằm tăng tính ứng dụng trong môi trường thực tế.

### 1.2.4 Nhận xét chung

Từ các công trình nghiên cứu trên có thể thấy rằng học sâu, đặc biệt là các mô hình CNN, hiện đang chiếm ưu thế trong bài toán phân loại rác thải sinh hoạt nhờ khả năng tự động học đặc trưng và đạt độ chính xác cao. Tuy nhiên, các phương pháp học máy truyền thống vẫn có giá trị trong những trường hợp dữ liệu hạn chế hoặc yêu cầu tính toán thấp. Điều này cho thấy

tiềm năng kết hợp và tối ưu các phương pháp khác nhau nhằm xây dựng hệ thống phân loại rác hiệu quả và phù hợp với điều kiện thực tế.

## 1.3 Mục tiêu của đề tài

Mục tiêu chính của đề tài không phải là xây dựng một hệ thống hoàn toàn mới mà tập trung vào việc xây dựng lại hệ thống phân loại rác thải sinh hoạt gia đình bằng hình ảnh dựa trên các thư viện sẵn có. Cụ thể, đề tài hướng đến các mục tiêu sau:

- Xây dựng hệ thống phân loại rác thải sinh hoạt dựa trên dữ liệu hình ảnh.
- Áp dụng và huấn luyện ba mô hình học sâu phổ biến gồm MobileNetV2, EfficientNetB0 và YOLOv8 cho bài toán phân loại rác.
- So sánh hiệu quả của các mô hình thông qua các chỉ số đánh giá nhằm xác định mô hình phù hợp cho từng điều kiện ứng dụng thực tế.
- Triển khai các ứng dụng phân loại rác thải, qua đó đánh giá về độ chính xác của các mô hình.

## 1.4 Đối tượng và phạm vi nghiên cứu

### 1.4.1 Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là các hình ảnh rác thải sinh hoạt gia đình, được thu thập từ nhiều nguồn khác nhau bao gồm Kaggle, GoogleImage và thu thập thủ công, phản ánh các loại rác thường gặp trong đời sống hằng ngày.

### 1.4.2 Phạm vi nghiên cứu

Trong khuôn khổ của môn học *Nhập môn Trí tuệ nhân tạo*, đề tài tập trung vào việc phân loại một số loại rác phổ biến như: giấy, nhựa, kim loại, thủy tinh rác hữu cơ, và rác thải bẩn. Đề tài không đi sâu vào các kỹ thuật xử lý rác sau phân loại mà chỉ tập trung vào bài toán nhận diện và phân loại dựa trên hình ảnh.

## 1.5 Phương pháp nghiên cứu

Để đạt được các mục tiêu đề ra, đề tài sử dụng các phương pháp nghiên cứu sau:

- Thu thập dữ liệu hình ảnh rác thải từ các nguồn công khai và tự xây dựng.
- Tiền xử lý dữ liệu bao gồm chuẩn hóa kích thước ảnh và tăng cường dữ liệu nhằm nâng cao hiệu quả huấn luyện.
- Áp dụng các mô hình học sâu dựa trên mạng nơ-ron tích chập (CNN) và mô hình phát hiện đối tượng để thực hiện phân loại rác.
- Đánh giá hiệu năng của các mô hình thông qua các chỉ số như độ chính xác (Accuracy), Precision, Recall và F1-score.

## 1.6 Bố cục báo cáo

Nội dung báo cáo được tổ chức thành 5 chương chính, được trình bày theo trình tự logic từ cơ sở lý thuyết đến xây dựng hệ thống, đánh giá kết quả và triển khai ứng dụng. Cụ thể như sau:

- **Chương 1: Giới thiệu** Trình bày lý do chọn đề tài, tổng quan các công trình nghiên cứu liên quan đến bài toán phân loại rác thải sinh hoạt, mục tiêu nghiên cứu, đối tượng và phạm vi nghiên cứu, phương pháp nghiên cứu, đồng thời giới thiệu bối cảnh tổng thể của báo cáo.
- **Chương 2: Cơ sở lý thuyết** Trình bày các kiến thức nền tảng liên quan đến trí tuệ nhân tạo, học máy, học sâu, mạng nơ-ron tích chập (CNN), bài toán phân loại ảnh và phát hiện đối tượng. Chương này cũng giới thiệu các mô hình học sâu được sử dụng trong đề tài bao gồm MobileNetV2, EfficientNetB0 và YOLOv8.
- **Chương 3: Xây dựng hệ thống phân loại rác** Mô tả quá trình thu thập và xây dựng bộ dữ liệu, tổ chức và tiền xử lý dữ liệu, cũng như quá trình thiết kế, huấn luyện các mô hình MobileNetV2, EfficientNetB0 và YOLOv8 với các tham số và chiến lược huấn luyện cụ thể.
- **Chương 4: Đánh giá và so sánh kết quả** Trình bày môi trường thực nghiệm, các tiêu chí đánh giá và kết quả huấn luyện của từng mô hình. Đồng thời, chương này tiến hành so sánh các mô hình về độ chính xác, độ ổn định giữa các lớp và khả năng ứng dụng trong thực tế, từ đó đưa ra các nhận xét và đánh giá tổng quát.
- **Chương 5: Xây dựng và triển khai ứng dụng phân loại rác thải** Trình bày quá trình triển khai các ứng dụng phân loại rác thải trong môi trường thực tế, bao gồm ứng dụng nhận diện rác bằng camera real-time và ứng dụng phân loại rác sử dụng giao diện Streamlit, với hai mô hình MobileNetV2 và EfficientNetB0. Chương này cũng đánh giá khả năng ứng dụng thực tế của hệ thống.
- **Chương 6: Kết luận và hướng phát triển** Tổng kết các kết quả đạt được của đề tài, chỉ ra những hạn chế còn tồn tại và đề xuất các hướng phát triển và nghiên cứu tiếp theo trong tương lai.

# Chương 2

## CƠ SỞ LÝ THUYẾT

### 2.1 Tổng quan về Trí tuệ nhân tạo

Trí tuệ nhân tạo là một lĩnh vực rộng lớn của khoa học máy tính, tập trung vào việc tạo ra các hệ thống có khả năng thực hiện các tác vụ đòi hỏi trí thông minh con người.[3]. Trong ngữ cảnh hiện đại, AI không chỉ dừng lại ở các hệ thống dựa trên quy tắc (Rule-based systems) mà đã chuyển dịch mạnh mẽ sang các hệ thống dựa trên dữ liệu (Data-driven).

Trong bài toán phân loại rác thải, AI đóng vai trò thay thế con người trong việc nhận diện các đặc điểm thị giác phức tạp của vật thể, giúp tự động hóa quy trình phân loại với tốc độ và độ chính xác cao mà các phương pháp lập trình truyền thống không thể giải quyết được.

### 2.2 Học máy và Học sâu

#### 2.2.1 Học máy

Học máy (Machine Learning – ML) là một nhánh của AI cho phép máy tính tự động học từ dữ liệu mà không cần lập trình tường minh.[4]. Các thuật toán học máy có thể được chia thành ba nhóm chính:

- Học có giám sát (Supervised Learning)
- Học không giám sát (Unsupervised Learning)
- Học tăng cường (Reinforcement Learning)

Trong bài toán phân loại rác thải, học có giám sát được sử dụng khi các hình ảnh đầu vào đã được gán nhãn tương ứng với từng loại rác.

#### 2.2.2 Học sâu

Học sâu (Deep Learning) là bước tiến hóa của ML, lấy cảm hứng từ cấu trúc mạng nơ-ron sinh học. Điểm đột phá của Deep Learning là khả năng Học biểu diễn (Representation Learning). Mô hình tự động học các đặc trưng từ cấp thấp (cạnh, góc) đến cấp cao (hình dáng vật thể) thông qua nhiều lớp ẩn (hidden layers) mà không cần sự can thiệp thủ công của con người.

Mô hình toán học cơ bản của một nơ-ron nhân tạo (Perceptron) được biểu diễn như sau:

$$y = f \left( \sum_{i=1}^n w_i x_i + b \right)$$

Trong đó:

- $x_i$ : Tín hiệu đầu vào (Input).
- $w_i$ : Trọng số (Weights) cần tối ưu hóa.
- $b$ : Hệ số thiên kiến (Bias).
- $f(\cdot)$ : Hàm kích hoạt phi tuyến (Non-linear Activation Function) như ReLU hoặc Sigmoid, giúp mạng có khả năng học các mối quan hệ phức tạp.

## 2.3 Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (CNN) là nền tảng của nhiều mô hình thị giác máy tính hiện đại [6, 7]. Cấu trúc của CNN được thiết kế để xử lý dữ liệu dạng lưới (như ảnh 2D) thông qua ba thành phần chính:

1. **Lớp tích chập (Convolutional Layer):** Đây là nơi trích xuất đặc trưng. Một "kernel" (bộ lọc) sẽ trượt qua toàn bộ ảnh đầu vào để tạo ra các bản đồ đặc trưng (Feature Maps). Phép toán tích chập tại vị trí  $(i, j)$  được định nghĩa:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Nhờ cơ chế chia sẻ trọng số (Parameter Sharing), CNN giảm thiểu đáng kể số lượng tham số so với mạng kết nối đầy đủ (Fully Connected).

2. **Lớp gộp (Pooling Layer):** Giúp giảm kích thước không gian của dữ liệu (Down-sampling), từ đó giảm chi phí tính toán và kiểm soát hiện tượng quá khớp (Overfitting). Phổ biến nhất là *Max Pooling* (lấy giá trị lớn nhất trong vùng lân cận).
3. **Lớp kết nối đầy đủ (Fully Connected Layer - FC):** Sau khi các đặc trưng không gian được trích xuất và nén lại, chúng được duỗi (flatten) thành vector và đưa vào lớp FC để thực hiện phân loại dựa trên xác suất.

## 2.4 Bài toán phân loại ảnh và phát hiện đối tượng

### 2.4.1 Phân loại ảnh

Phân loại ảnh (Image Classification) là bài toán xác định nhãn của toàn bộ hình ảnh dựa trên nội dung bên trong. Trong bài toán này, mỗi ảnh đầu vào chỉ tương ứng với một nhãn duy nhất.

- **Đầu vào:** Một ma trận ảnh  $I$ .
- **Đầu ra:** Một vector xác suất  $P = [p_1, p_2, \dots, p_n]$  tương ứng với  $n$  lớp, sao cho  $\sum p_i = 1$ .
- **Cơ chế:** Mô hình tập trung vào việc trả lời câu hỏi “Cái gì có trong ảnh?”. Các mô hình như MobileNetV2 hay EfficientNetB0 trong đề tài này hoạt động theo cơ chế này. Chúng nhìn toàn bộ bức ảnh và quyết định xem đó là “Nhựa” hay “Kim loại”.

## 2.4.2 Phát hiện đối tượng

Phát hiện đối tượng là bài toán phức tạp hơn, kết hợp giữa hai tác vụ: Định vị (Localization) và Phân loại (Classification).

- **Đầu vào:** Một ma trận ảnh.
- **Đầu ra:** Một danh sách các đối tượng, mỗi đối tượng bao gồm: Nhãn lớp (Class Label), Độ tin cậy (Confidence Score) và Hộp giới hạn (Bounding Box:  $x, y, w, h$ ).
- **Cơ chế:** Mô hình trả lời câu hỏi “Cái gì có trong ảnh và nó nằm ở đâu?”.

Trong đánh giá phát hiện đối tượng, chỉ số **IoU (Intersection over Union)** được sử dụng để đo độ khớp giữa hộp dự đoán ( $B_p$ ) và hộp thực tế ( $B_{gt}$ ):

$$IoU = \frac{\text{Area}(B_p \cap B_{gt})}{\text{Area}(B_p \cup B_{gt})}$$

## 2.4.3 Sự khác nhau trong ngũ cảnh phân loại rác

Sự khác biệt cốt lõi ảnh hưởng đến việc lựa chọn mô hình cho hệ thống thực tế:

- **Với bài toán Phân loại (MobileNet, EfficientNet):** Giả định rằng mỗi ảnh chỉ chứa một đối tượng rác duy nhất nằm ở trung tâm (ví dụ: người dùng đưa rác vào buồng chụp của thùng rác thông minh). Ưu điểm là mô hình đơn giản, dễ huấn luyện.
- **Với bài toán Phát hiện (YOLOv8):** Phù hợp cho bối cảnh thực tế phức tạp hơn, ví dụ như rác nằm lẩn lộn trên băng chuyền hoặc ảnh chụp môi trường tự nhiên chứa nhiều loại rác khác nhau cùng lúc. YOLOv8 có thể tách biệt từng chai nhựa, lon nước ngọt ra khỏi nền ảnh phức tạp.

## 2.5 Giới thiệu các mô hình sử dụng trong đề tài

### 2.5.1 MobileNetV2

MobileNetV2 giải quyết vấn đề triển khai Deep Learning trên các thiết bị có tài nguyên tính toán hạn chế (điện thoại, nhúng) thông qua hai cải tiến kiến trúc đột phá:

a. **Depthwise Separable Convolution (Tích chập tách biệt chiều sâu)** Thay vì sử dụng phép tích chập tiêu chuẩn tốn kém, MobileNetV2 tách nó thành hai bước:

1. **Depthwise Convolution:** Áp dụng một bộ lọc duy nhất cho mỗi kênh đầu vào.
2. **Pointwise Convolution ( $1 \times 1$  Convolution):** Kết hợp đầu ra của lớp depthwise bằng tích chập  $1 \times 1$ .

*Phân tích hiệu quả tính toán:* Giả sử đầu vào có kích thước  $H \times W \times C$ , và ta dùng  $N$  bộ lọc kích thước  $K \times K$ .

- Chi phí tích chập thường:  $H \cdot W \cdot C \cdot N \cdot K^2$
- Chi phí MobileNet:  $H \cdot W \cdot C \cdot K^2 + H \cdot W \cdot C \cdot N$
- Tỷ lệ giảm tải:

$$\frac{\text{MobileNet Cost}}{\text{Standard Cost}} = \frac{1}{N} + \frac{1}{K^2}$$

Với  $K = 3$  (kernel 3x3), MobileNetV2 giảm khối lượng tính toán đi khoảng **8 đến 9 lần** mà độ chính xác giảm không đáng kể.

**b. Inverted Residuals & Linear Bottlenecks** Khác với kiến trúc ResNet truyền thống (Wide → Narrow → Wide), MobileNetV2 sử dụng cấu trúc ngược lại (Narrow → Wide → Narrow):

- Đầu vào được mở rộng số kênh (Expansion) để tăng khả năng biểu diễn.
- Thực hiện trích xuất đặc trưng bằng Depthwise Conv.
- Nén lại số kênh (Projection). Đặc biệt, lớp Projection cuối cùng sử dụng hàm kích hoạt tuyến tính (Linear Activation) thay vì ReLU để tránh làm mất thông tin khi nén dữ liệu xuống chiều thấp (“Manifold of interest”).

### 2.5.2 EfficientNetB0

EfficientNetB0 là một kiến trúc CNN được xây dựng dựa trên nguyên lý *Compound Scaling*, trong đó độ sâu (depth), độ rộng (width) và độ phân giải (resolution) của mạng được mở rộng một cách đồng thời và cân đối.

Cụ thể, EfficientNet đề xuất các hệ số mở rộng:

$$\text{depth} = \alpha^\phi, \quad \text{width} = \beta^\phi, \quad \text{resolution} = \gamma^\phi$$

với ràng buộc:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

Cách mở rộng này giúp mô hình sử dụng hiệu quả tài nguyên tính toán hơn so với việc chỉ tăng riêng lẻ độ sâu hoặc độ rộng của mạng.

EfficientNetB0 là phiên bản cơ sở của họ EfficientNet, được thiết kế để đạt sự cân bằng giữa độ chính xác và chi phí tính toán. Mô hình sử dụng các khối *MBConv* (Mobile Inverted Bottleneck Convolution), kết hợp giữa Depthwise Separable Convolution và cơ chế *Squeeze-and-Excitation* nhằm tăng cường khả năng học các đặc trưng quan trọng.

Nhờ kiến trúc tối ưu, EfficientNetB0 thường đạt độ chính xác cao hơn so với nhiều mô hình CNN truyền thống trong khi vẫn giữ số lượng tham số ở mức hợp lý, phù hợp cho các bài toán phân loại ảnh yêu cầu độ chính xác cao.

### 2.5.3 YOLOv8

YOLOv8 (You Only Look Once phiên bản 8) là mô hình phát hiện đối tượng thuộc nhóm *one-stage detector*, cho phép thực hiện đồng thời hai nhiệm vụ: xác định vị trí đối tượng và phân loại đối tượng trong cùng một lần suy luận.

Khác với các mô hình hai giai đoạn, YOLOv8 chia ảnh đầu vào thành các ô lưới và trực tiếp dự đoán:

- Tọa độ hộp giới hạn (bounding box),
- Xác suất tồn tại đối tượng,
- Nhãn phân loại của đối tượng.

Hàm mất mát của YOLOv8 là sự kết hợp giữa:

- Loss định vị (Bounding Box Loss),

- Loss phân loại (Classification Loss),
- Loss đối tượng (Objectness Loss).

Việc tối ưu đồng thời các thành phần này giúp mô hình học được cả vị trí và loại đối tượng một cách hiệu quả.

Trong đề tài này, YOLOv8 được sử dụng để đánh giá khả năng phát hiện và phân loại rác thải trong các tình huống thực tế, nơi một hình ảnh có thể chứa nhiều loại rác khác nhau. Nhờ tốc độ suy luận nhanh và độ chính xác cao, YOLOv8 phù hợp cho các hệ thống phân loại rác tự động trong môi trường thực tế và thời gian thực.

# Chương 3

## XÂY DỰNG HỆ THỐNG PHÂN LOẠI RÁC

Chương này trình bày chi tiết quy trình xây dựng hệ thống phân loại rác thải, bao gồm các giai đoạn: thu thập và xử lý dữ liệu, thiết kế kiến trúc mô hình, và thiết lập các chiến lược huấn luyện cho ba mô hình mạng nơ-ron tích chập: MobileNetV2, EfficientNetB0 và YOLOv8.

### 3.1 Thu thập và Xây dựng bộ dữ liệu

#### 3.1.1 Nguồn dữ liệu

Để đảm bảo tính tổng quát và khả năng ứng dụng thực tế, bộ dữ liệu được tổng hợp từ hai nguồn chính:

a. Bộ dữ liệu TrashNet: Đây là nguồn dữ liệu nền tảng (baseline) cho nghiên cứu, được công bố bởi Gary Thung và Mindy Yang (Đại học Stanford). Tập dữ liệu bao gồm 2.527 hình ảnh rác thải chất lượng cao, cung cấp các đặc trưng cơ bản của đối tượng trong điều kiện tiêu chuẩn.

b. Google Image và Ánh thu thập thủ công: Nhằm khắc phục hạn chế về sự đa dạng bối cảnh và điều kiện ánh sáng của TrashNet, dữ liệu được bổ sung thông qua việc thu thập từ Google Images và chụp ảnh thủ công. Việc này giúp mô hình thích nghi tốt hơn với các biến thể rác thải trong môi trường thực tế.

Quá trình chia hình ảnh được tự động hóa với tỉ lệ train : val : test = 70 : 15 : 15.

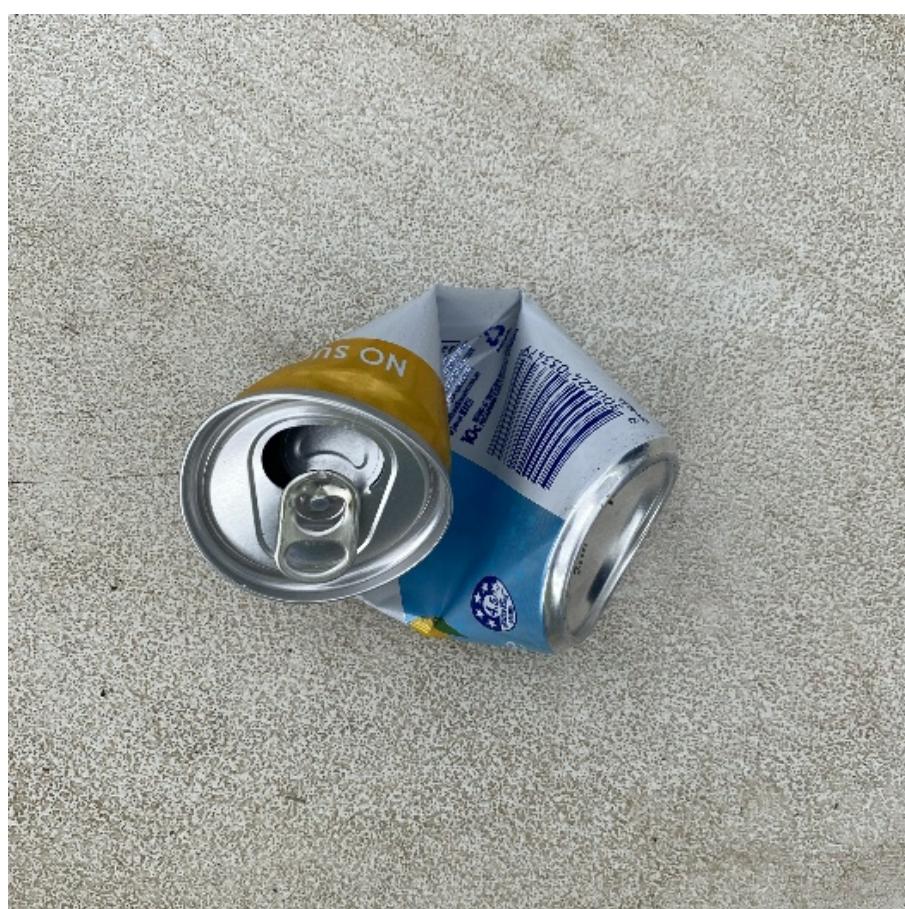
#### 3.1.2 Phân bố và Tổ chức dữ liệu

Sau quá trình tiền xử lý, sàng lọc và gán nhãn, bộ dữ liệu hoàn chỉnh bao gồm 3.773 hình ảnh, được phân loại thành 6 lớp bao gồm glass, metal, organic, paper, plastic, trash.

Dưới đây là một số hình minh họa cho tập dữ liệu:



Hình 3.1: Hình minh họa lớp glass(thủy tinh)



Hình 3.2: Hình minh họa lớp metal(kim loại)



Hình 3.3: Hình minh họa lớp organic(rác hữu cơ)

Số liệu thống kê tại Bảng 3.1 cho thấy sự phân bố dữ liệu giữa các lớp tương đối đồng đều, với số lượng ảnh dao động từ hơn 500 đến hơn 700 ảnh mỗi lớp. Sự cân bằng này đóng vai trò quan trọng trong việc hạn chế hiện tượng thiên lệch (bias) của mô hình đối với các lớp chiếm đa số.

Bảng 3.1: Thống kê chi tiết phân bố dữ liệu cho từng lớp

| STT       | Nhân (Class)      | Tổng số      | Train        | Val        | Test       |
|-----------|-------------------|--------------|--------------|------------|------------|
| 1         | Glass (Thủy tinh) | 536          | 375          | 80         | 81         |
| 2         | Metal (Kim loại)  | 622          | 435          | 93         | 94         |
| 3         | Organic (Hữu cơ)  | 761          | 532          | 114        | 115        |
| 4         | Paper (Giấy)      | 647          | 452          | 97         | 98         |
| 5         | Plastic (Nhựa)    | 601          | 420          | 90         | 91         |
| 6         | Trash (Rác khác)  | 606          | 424          | 90         | 92         |
| Tổng cộng |                   | <b>3.773</b> | <b>2.638</b> | <b>564</b> | <b>571</b> |

### 3.1.3 Tiết xử lý và tăng cường dữ liệu

Trước khi đưa vào các mô hình CNN (MobileNetV2 và EfficientNetB0), dữ liệu trải qua quy trình xử lý thông qua lớp ImageDataGenerator với các kỹ thuật cụ thể sau:

**a. Chuẩn hóa (Normalization):** Giá trị điểm ảnh (pixel intensity) của toàn bộ dữ liệu (Train, Val, Test) được chuẩn hóa từ khoảng [0, 255] về khoảng [0, 1] thông qua phép nhân với hệ số tỉ lệ 1/255. Kỹ thuật này giúp đồng bộ hóa phân phối dữ liệu, hỗ trợ thuật toán tối ưu (Optimizer) hội tụ nhanh và ổn định hơn.

**b. Tăng cường dữ liệu (Data Augmentation):** Đối với tập huấn luyện (Train set), các kỹ thuật tăng cường được áp dụng để giải quyết vấn đề dữ liệu hạn chế và ngăn chặn hiện

tượng quá khớp (Overfitting). Các phép biến đổi bao gồm:

- **Rotation:** Xoay ảnh ngẫu nhiên trong phạm vi 20 độ.
- **Shift:** Dịch chuyển ảnh theo chiều ngang và dọc với tỷ lệ 10% (width/height shift range = 0.1).
- **Shear & Zoom:** Áp dụng biến đổi cắt và phóng to/thu nhỏ với tỷ lệ 20% (range = 0.2).
- **Horizontal Flip:** Lật ảnh theo trục dọc ngẫu nhiên.

Những biến đổi này giúp mô hình học được tính bất biến (invariance) của đối tượng, đảm bảo khả năng nhận diện chính xác dù rác thải xuất hiện ở các tư thế hoặc kích thước khác nhau. Lưu ý rằng tập Validation và Test không áp dụng tăng cường để đảm bảo kết quả đánh giá phản ánh trung thực dữ liệu thực tế.

## 3.2 Xây dựng và huấn luyện mô hình MobileNetV2

### 3.2.1 Thiết kế kiến trúc mô hình

MobileNetV2 là mạng CNN nhẹ sử dụng kỹ thuật *Depthwise Separable Convolution* và *Inverted Residuals*. Trong hệ thống, MobileNetV2 được sử dụng như một backbone trích xuất đặc trưng, sau đó gắn thêm các lớp:

- **Backbone:** Sử dụng mạng MobileNetV2 đã được huấn luyện trước trên bộ dữ liệu ImageNet (`weights='imagenet'`). Lớp phân loại gốc được loại bỏ (`include_top=False`) để xây dựng phần đầu mạng (Head) mới.
- **Global Average Pooling:** Giảm chiều dữ liệu từ bản đồ đặc trưng (Feature Map)  $7 \times 7 \times 1280$  xuống vector đặc trưng 1 chiều (1280, ).
- **Lớp Dense cho phân loại:** Một lớp ẩn kết nối đầy đủ với 128 nơ-ron và hàm kích hoạt ReLU được thêm vào để tăng khả năng biểu diễn phi tuyến tính các đặc trưng rác thải.
- **Dropout để giảm overfitting:** Áp dụng kỹ thuật Dropout với tỷ lệ 50% trước lớp đầu ra. Đây là tỷ lệ khá cao, cho thấy chiến lược kiểm soát overfitting chặt chẽ của mô hình này.
- **Output Layer:** Lớp Dense cuối cùng gồm 6 nơ-ron sử dụng hàm kích hoạt Softmax để phân loại vào 6 nhóm rác.

### 3.2.2 Tham số huấn luyện

Các tham số huấn luyện của MobileNetV2 được cấu hình như sau:

- Kích thước ảnh:  $224 \times 224$
- Batch size: 32
- Optimizer: Adam
- Hàm mất mát: Categorical Crossentropy

Quá trình huấn luyện sử dụng **Early Stopping** để tự động dừng khi độ lỗi trên tập validation không còn cải thiện, giúp tránh hiện tượng quá khớp và tối ưu thời gian huấn luyện.

### 3.3 Xây dựng và huấn luyện mô hình EfficientNetB0

#### 3.3.1 Thiết kế kiến trúc mô hình

EfficientNetB0 là mô hình CNN được thiết kế dựa trên nguyên lý *Compound Scaling*, cho phép cân bằng giữa độ sâu, độ rộng và độ phân giải của mạng. Mô hình được sử dụng làm backbone và gắn thêm các lớp phân loại ở phía trên.

- **Backbone:** EfficientNetB0 (Weights: ImageNet).
- **Global Average Pooling:** Tổng hợp đặc trưng không gian.
- **Dropout (0.2):** Tỷ lệ Dropout được thiết lập ở mức 20%. Mức này thấp hơn so với MobileNetV2 do kiến trúc nội tại của EfficientNet đã tích hợp các cơ chế chuẩn hóa (Batch Normalization) hiệu quả trong từng khối MBConv.
- **Output Layer:** Kết nối trực tiếp đến lớp phân loại 6 nơ-ron (Softmax) mà không qua lớp ẩn trung gian, giúp giảm độ phức tạp tính toán và số lượng tham số.

#### 3.3.2 Chiến lược huấn luyện

Chiến lược huấn luyện EfficientNetB0 tập trung vào việc bảo toàn các đặc trưng cấp cao thông qua quy trình "Warm-up" và "Deep Fine-tuning":

##### Giai đoạn 1 – Huấn luyện phần đầu (Head training):

- Batch size: 16
- Số epoch: 5
- Optimizer: Adam
- Learning rate:  $1 \times 10^{-3}$

Trong giai đoạn này, các lớp backbone được đóng băng (freeze), chỉ huấn luyện các lớp phân loại phía trên. Mục đích: Ổn định trọng số lớp cuối cùng trước khi lan truyền ngược gradient vào các lớp sâu.

##### Giai đoạn 2 – Fine-tuning:

- Mở khóa một phần backbone (trừ khoảng 30 lớp đầu).
- Số epoch: 20
- Optimizer: Adam
- Learning rate:  $1 \times 10^{-5}$
- Callback ModelCheckpoint được sử dụng để lưu lại phiên bản mô hình có độ chính xác trên tập Validation cao nhất (`save_best_only=True`), đảm bảo kết quả cuối cùng là tối ưu nhất.

Chiến lược fine-tuning giúp mô hình học sâu hơn các đặc trưng của rác thải mà vẫn giữ được tính ổn định.

## 3.4 Xây dựng và huấn luyện mô hình YOLOv8

YOLOv8 (You Only Look Once - version 8) là mô hình tiên tiến nhất (SOTA) về tốc độ xử lý thời gian thực. Trong nghiên cứu này, biến thể phân loại hình ảnh (Image Classification) được sử dụng thay vì phát hiện đối tượng.

### 3.4.1 Kiến trúc và Cấu hình

Mô hình được lựa chọn là yolov8n-cls.pt (Nano Classification). Đây là phiên bản nhỏ gọn nhất, sử dụng kiến trúc backbone CSPDarknet53 cải tiến, tối ưu hóa cho tốc độ suy luận (Inference speed) trên các hệ thống nhúng.

Khác với hai mô hình trên, YOLOv8 xử lý dữ liệu theo pipeline nội tại của thư viện Ultralytics với cấu trúc thư mục phân cấp, tích hợp sẵn các kỹ thuật tăng cường dữ liệu đặc thù như Mosaic Augmentation.

### 3.4.2 Tham số huấn luyện

Quá trình huấn luyện được cấu hình thông qua API với các tham số tối ưu hóa cho sự hội tụ:

- **Bộ tối ưu hóa (Optimizer):** Sử dụng AdamW (Adam with Weight Decay). Thuật toán này tách biệt quá trình suy giảm trọng số khỏi quá trình cập nhật gradient, giúp mô hình có khả năng tổng quát hóa (generalization) tốt hơn so với Adam truyền thống.
- **Computational Params:**
  - Số epoch: 50 (Chu kỳ huấn luyện dài hơn để đảm bảo hội tụ sâu).
  - Kích thước ảnh: 224 (Đồng bộ kích thước với các mô hình trước).
  - Batch size: 32
- **Learning rate ban đầu:**  $1 \times 10^{-3}$
- **Early stopping:** Tham số patience=10 được thiết lập nhằm tự động dừng quá trình huấn luyện nếu độ chính xác không cải thiện, giúp tiết kiệm tài nguyên tính toán.
- **Thiết bị huấn luyện:** CPU

Dữ liệu huấn luyện được cung cấp theo định dạng thư mục phù hợp với YOLO, cho phép mô hình học đồng thời vị trí và nhãn của rác thải.

# Chương 4

## ĐÁNH GIÁ VÀ SO SÁNH KẾT QUẢ

### 4.1 Môi trường thực nghiệm

Các thí nghiệm trong đề tài được thực hiện trên môi trường máy tính cá nhân với các thông số cơ bản như sau:

- Ngôn ngữ lập trình: Python
- Thư viện chính: TensorFlow/Keras (đối với MobileNetV2 và EfficientNetB0), Ultralytics YOLOv8
- Nền tảng huấn luyện: Google Colab

Dữ liệu được chia thành ba tập riêng biệt gồm tập huấn luyện (training set), tập kiểm tra (validation set) và tập kiểm thử (test set) nhằm đảm bảo đánh giá khách quan hiệu quả của các mô hình.

### 4.2 Các tiêu chí đánh giá

Để đánh giá hiệu quả của các mô hình, đề tài sử dụng các chỉ số phổ biến trong bài toán phân loại và phát hiện đối tượng, bao gồm:

- **Accuracy (Độ chính xác):** Tỉ lệ dự đoán đúng trên tổng số mẫu.
- **Precision:** Đánh giá mức độ chính xác của các dự đoán dương tính.
- **Recall:** Đánh giá khả năng mô hình phát hiện đầy đủ các mẫu thuộc một lớp.
- **F1-score:** Trung bình điều hòa giữa Precision và Recall.

Đối với YOLOv8, ngoài các chỉ số trên, mô hình còn được đánh giá thông qua khả năng phát hiện đúng vị trí và nhãn của các đối tượng trong ảnh.

### 4.3 Kết quả huấn luyện các mô hình

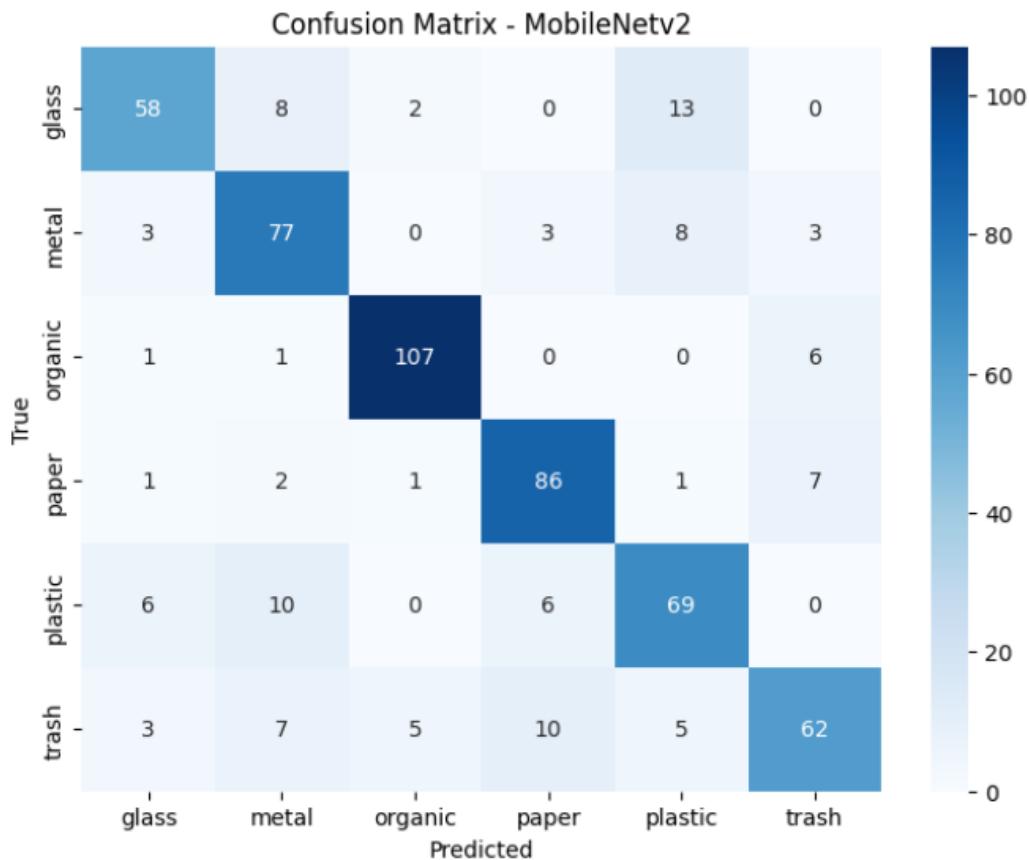
#### 4.3.1 Kết quả với MobileNetV2

Mô hình MobileNetV2 được đánh giá trên tập kiểm thử với các chỉ số như sau:

- Accuracy: 80.39%

- Precision: 80.56%
- Recall: 80.39%
- F1-score: 80.29%

Kết quả cho thấy MobileNetV2 đạt độ chính xác ở mức khá, phù hợp cho các hệ thống yêu cầu tốc độ nhanh và tài nguyên phần cứng hạn chế. Tuy nhiên, độ chính xác chưa cao đối với các lớp rác có đặc điểm hình ảnh tương đồng.



Hình 4.1: Ma trận nhầm lẫn của mô hình MobileNetV2

Hình 4.1 trình bày ma trận nhầm lẫn của mô hình MobileNetV2 trên tập dữ liệu kiểm tra. Từ kết quả thu được, có thể đưa ra một số nhận xét như sau:

- Các giá trị trên đường chéo chính vẫn chiếm ưu thế, cho thấy mô hình MobileNetV2 có khả năng phân loại tương đối tốt các lớp rác thải.
- Lớp *organic* đạt kết quả cao nhất với số lượng mẫu được phân loại đúng lớn, cho thấy mô hình học tốt các đặc trưng rõ ràng của lớp này.
- Tuy nhiên, so với EfficientNetB0 và YOLOv8, mức độ nhầm lẫn của MobileNetV2 giữa các lớp là cao hơn, đặc biệt ở các lớp *plastic*, *trash* và *glass*.
- Một số nhầm lẫn đáng chú ý xảy ra giữa các lớp có đặc trưng hình ảnh tương đồng, điển hình là giữa *plastic* và *metal*, cũng như giữa *trash* và *paper*.
- Nguyên nhân có thể do kiến trúc MobileNetV2 có số lượng tham số ít hơn, dẫn đến khả năng trích xuất đặc trưng phức tạp bị hạn chế.

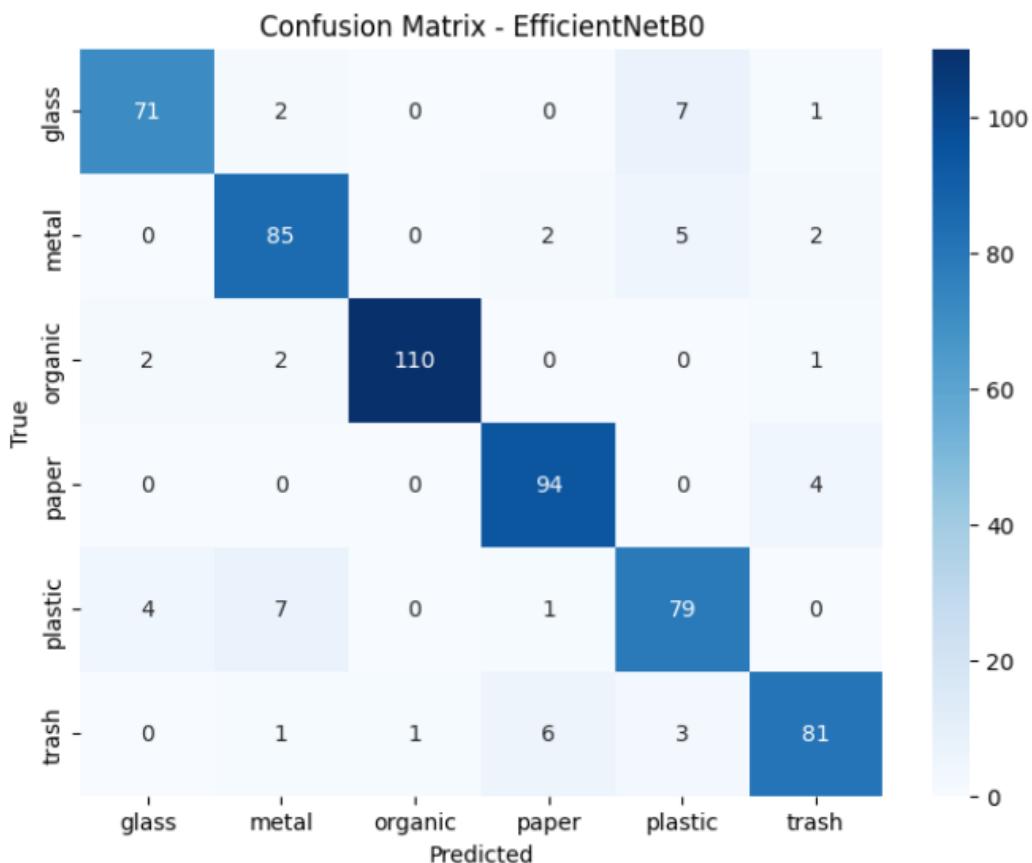
Nhìn chung, MobileNetV2 cho hiệu năng phân loại ở mức chấp nhận được và phù hợp với các hệ thống yêu cầu tài nguyên tính toán thấp.

### 4.3.2 Kết quả với EfficientNetB0

Mô hình EfficientNetB0 đạt kết quả nổi bật trên tập kiểm thử với:

- Test Loss: 0.2375
- Test Accuracy: 92.84%
- Macro F1-score: 0.91

Kết quả phân loại chi tiết cho thấy EfficientNetB0 đạt độ chính xác cao trên tất cả các lớp, đặc biệt là lớp *organic* với F1-score đạt 0.97. Điều này chứng minh hiệu quả của kiến trúc Compound Scaling trong việc nâng cao chất lượng phân loại.



Hình 4.2: Ma trận nhầm lẫn của mô hình EfficientNetB0

Hình 4.2 trình bày ma trận nhầm lẫn của mô hình EfficientNetB0 trên tập dữ liệu kiểm tra. Dựa trên kết quả thu được, có thể đưa ra một số nhận xét như sau:

- Các giá trị lớn tập trung chủ yếu trên đường chéo chính cho thấy mô hình EfficientNetB0 đạt hiệu năng phân loại cao và ổn định.
- Lớp *organic* cho kết quả tốt nhất với số lượng mẫu được phân loại đúng rất cao, cho thấy mô hình học hiệu quả các đặc trưng của rác hữu cơ.
- Các lớp *metal*, *paper*, *plastic* và *trash* đều có số lượng dự đoán đúng lớn, phản ánh khả năng tổng quát hóa tốt của mô hình.
- Một số nhầm lẫn vẫn xảy ra giữa các lớp có đặc trưng hình ảnh tương đồng, điển hình là giữa *plastic* và *metal*, hoặc giữa *trash* và *paper*.

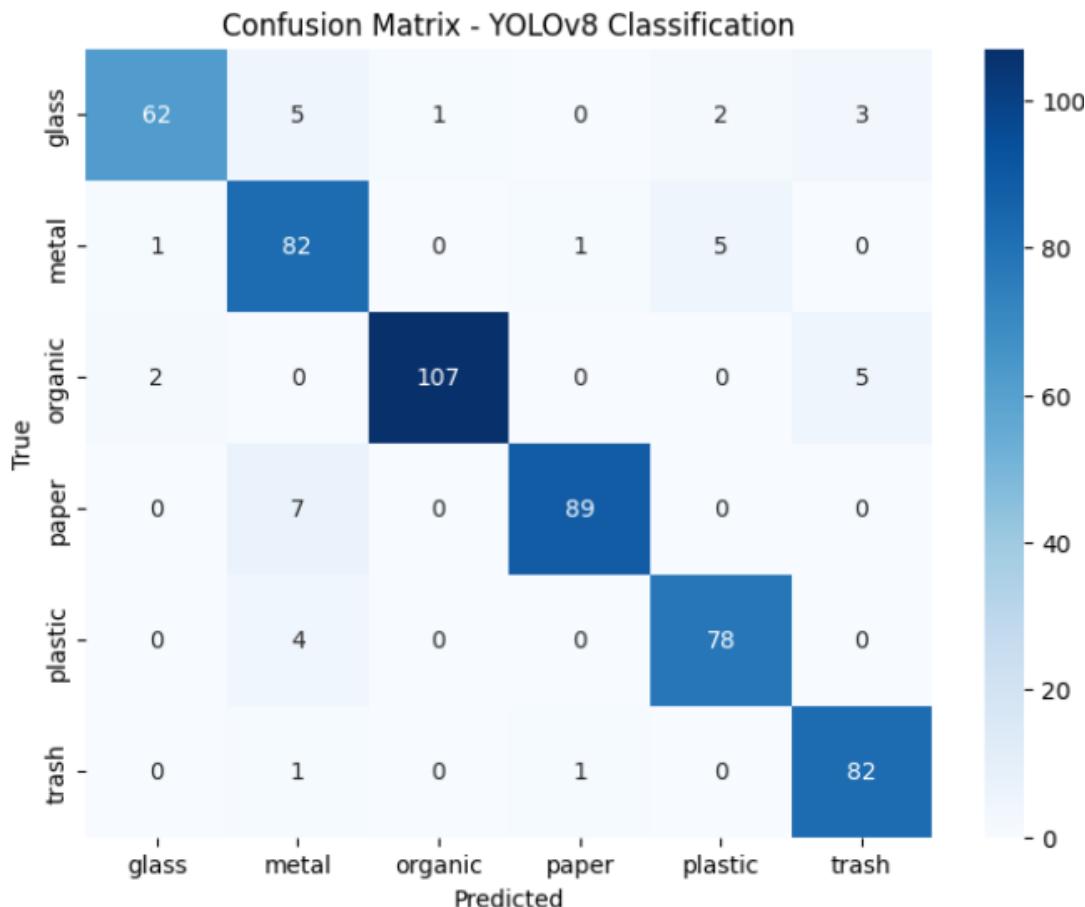
- Tuy nhiên, số lượng các mẫu bị nhầm lẫn là tương đối nhỏ so với tổng số mẫu, cho thấy mô hình vẫn duy trì độ chính xác cao.

Nhìn chung, mô hình EfficientNetB0 cho thấy khả năng phân loại rác thải hiệu quả và là một lựa chọn phù hợp cho bài toán phân loại ảnh nhiều lớp.

#### 4.3.3 Kết quả với YOLOv8

YOLOv8 đạt độ chính xác tổng thể 92.94% (không tính các trường hợp reject). Kết quả đánh giá cho thấy mô hình không chỉ phân loại chính xác mà còn xác định tốt vị trí của các đối tượng rác trong ảnh.

YOLOv8 đạt Macro F1-score là 0.9272 và Weighted F1-score là 0.9299, cho thấy hiệu suất ổn định và đồng đều trên các lớp dữ liệu khác nhau.



Hình 4.3: Ma trận nhầm lẫn của mô hình YOLOv8

Hình 4.3 minh họa ma trận nhầm lẫn của mô hình YOLOv8 trong bài toán phân loại rác thải. Từ kết quả thu được, có thể rút ra một số nhận xét quan trọng như sau:

- Các giá trị lớn tập trung trên đường chéo chính cho thấy mô hình đạt độ chính xác cao, phần lớn các mẫu trong tập test được phân loại đúng.
- Lớp *organic* cho hiệu năng tốt nhất với số lượng mẫu dự đoán đúng cao, rất ít bị nhầm lẫn sang các lớp khác.
- Các lớp *metal*, *paper*, *plastic* và *trash* cũng đạt kết quả khả quan, chứng tỏ mô hình có khả năng tổng quát hóa tốt.

- Một số nhầm lẫn xảy ra giữa các lớp có đặc trưng hình ảnh tương đồng, điển hình là giữa *glass* và *metal*, hoặc *paper* và *metal*.
- Nguyên nhân của hiện tượng này có thể đến từ sự tương đồng về màu sắc, hình dạng hoặc điều kiện ánh sáng trong ảnh.

Nhìn chung, ma trận nhầm lẫn cho thấy mô hình YOLOv8-classification hoạt động ổn định và phù hợp cho bài toán phân loại rác thải trong thực tế.

## 4.4 So sánh các mô hình

Để có cái nhìn tổng quan và lựa chọn mô hình tối ưu cho hệ thống phân loại rác thải, bảng 4.1 dưới đây tóm tắt các chỉ số hiệu năng chính của ba mô hình MobileNetV2, EfficientNetB0 và YOLOv8 trên cùng tập dữ liệu kiểm thử.

Bảng 4.1: Bảng so sánh hiệu năng giữa các mô hình

| Mô hình        | Accuracy (%)  | F1-Score    | Đánh giá sơ bộ                                                     |
|----------------|---------------|-------------|--------------------------------------------------------------------|
| MobileNetV2    | 80.39%        | 0.80        | Tốc độ cao, độ chính xác trung bình, phù hợp thiết bị yếu.         |
| EfficientNetB0 | 92.84%        | 0.91        | Độ chính xác cao, ổn định, cân bằng giữa tốc độ và hiệu năng.      |
| YOLOv8         | <b>92.94%</b> | <b>0.93</b> | Độ chính xác cao nhất, tối ưu cho bài toán thực tế thời gian thực. |

Dựa vào bảng số liệu và các biểu đồ ma trận nhầm lẫn đã phân tích, có thể rút ra các so sánh chi tiết sau:

### Về độ chính xác (Accuracy)

- YOLOv8 và EfficientNetB0 thể hiện sự vượt trội đáng kể so với MobileNetV2. Cả hai đều đạt độ chính xác xấp xỉ 93%, cao hơn khoảng 12.5% so với MobileNetV2 (80.39%). Điều này cho thấy kiến trúc Compound Scaling (của EfficientNet) và kiến trúc CSPDarknet cải tiến (của YOLOv8) có khả năng trích xuất đặc trưng sâu và hiệu quả hơn hẳn so với kiến trúc MobileNetV2 cũ hơn trên tập dữ liệu này.
- MobileNetV2 mặc dù là một mô hình nhẹ, nhưng việc đánh đổi quá nhiều tham số để tối ưu tốc độ đã dẫn đến hạn chế trong việc phân biệt các lớp có đặc điểm thị giác gần giống nhau (như nhựa và kim loại).

### Về độ ổn định giữa các lớp (Class-wise Performance)

- Cả ba mô hình đều hoạt động rất tốt trên lớp *organic* (hữu cơ). Đây là lớp có đặc trưng hình ảnh khác biệt rõ rệt so với các loại rác tái chế (vỏ chai, giấy, lon).
- Sự khác biệt nằm ở khả năng xử lý các lớp "khó". Trong khi MobileNetV2 thường xuyên nhầm lẫn giữa *plastic*, *glass* và *metal*, thì YOLOv8 và EfficientNetB0 đã giảm thiểu đáng kể tỷ lệ này. Đặc biệt, YOLOv8 cho thấy chỉ số F1-score đồng đều nhất giữa các lớp, chứng tỏ mô hình ít bị thiên lệch (bias) hơn.

## Về khả năng ứng dụng thực tế

- MobileNetV2: Phù hợp cho các thiết bị nhúng có cấu hình cực thấp (như vi điều khiển hoặc IoT đời cũ) nơi mà độ chính xác không phải là ưu tiên hàng đầu.
- EfficientNetB0: Là sự lựa chọn cân bằng, phù hợp cho các hệ thống server hoặc ứng dụng di động cần độ tin cậy cao.
- YOLOv8: Đây là ứng cử viên sáng giá nhất cho bài toán này. Với độ chính xác cao nhất (92.94%) và bản chất là mô hình được thiết kế cho tác vụ thời gian thực (Real-time), YOLOv8 đáp ứng tốt cả hai yêu cầu về hiệu năng và tốc độ xử lý khi triển khai trên camera giám sát hoặc robot phân loại rác tự động.

## 4.5 Nhận xét và đánh giá

Dựa trên toàn bộ quá trình thực nghiệm, từ việc huấn luyện đến phân tích ma trận nhầm lẫn và các chỉ số đánh giá, có thể rút ra những nhận định cốt lõi sau về hiệu năng của ba mô hình:

- **Về hiệu năng phân loại:** Cả YOLOv8 và EfficientNetB0 đều chứng tỏ khả năng vượt trội trong việc trích xuất đặc trưng với độ chính xác xấp xỉ 93%. Trong khi đó, MobileNetV2, dù có lợi thế về kiến trúc nhẹ, đã bộc lộ hạn chế khi xử lý các lớp dữ liệu có tính chất thị giác tương đồng (như *plastic* và *glass*), dẫn đến độ chính xác chỉ dừng lại ở mức 80.39%.
- **Quyết định lựa chọn mô hình:** Xét trên sự cân bằng giữa độ chính xác và tốc độ xử lý, nghiên cứu quyết định lựa chọn YOLOv8 là mô hình chính thức để triển khai cho hệ thống. Mô hình này không chỉ đạt độ chính xác cao nhất (92.94%) mà còn đáp ứng tốt nhất yêu cầu về thời gian thực, đảm bảo khả năng tích hợp lên các thiết bị phần cứng hoặc hệ thống nhúng trong giai đoạn tiếp theo.

Việc lựa chọn mô hình phù hợp phụ thuộc vào mục tiêu ứng dụng, tài nguyên phần cứng và yêu cầu về độ chính xác của hệ thống.

# Chương 5

## XÂY DỰNG VÀ TRIỂN KHAI ỨNG DỤNG PHÂN LOẠI RÁC THẢI

### 5.1 Mục tiêu triển khai ứng dụng

Nhằm đánh giá khả năng ứng dụng thực tế của các mô hình đã huấn luyện, đề tài tiến hành xây dựng hai ứng dụng phân loại rác thải dựa trên hình ảnh, bao gồm:

- Ứng dụng phân loại rác thời gian thực sử dụng camera (camera\_demo).
- Ứng dụng giao diện web sử dụng Streamlit cho phép upload ảnh và nhận diện qua camera real-time.

Hai ứng dụng được triển khai với hai mô hình học sâu là MobileNetV2 và EfficientNetB0 nhằm so sánh hiệu quả và khả năng áp dụng trong các tình huống thực tế khác nhau.

### 5.2 Ứng dụng phân loại rác bằng camera real-time

#### 5.2.1 Mô tả tổng quan

Ứng dụng `camera_demo.py` cho phép sử dụng camera của máy tính để thu nhận hình ảnh theo thời gian thực và thực hiện phân loại rác trực tiếp trên từng khung hình. Ứng dụng được triển khai với hai phiên bản tương ứng với hai mô hình MobileNetV2 và EfficientNetB0.

#### 5.2.2 Quy trình xử lý

Quy trình xử lý của ứng dụng camera được thực hiện theo các bước sau:

1. Thu nhận khung hình từ camera thông qua thư viện OpenCV.
2. Tiền xử lý ảnh: resize ảnh về kích thước  $224 \times 224$ .
3. Chuẩn hóa dữ liệu đầu vào phù hợp với từng mô hình.
4. Dưa ảnh vào mô hình đã huấn luyện để dự đoán.
5. Hiển thị nhãn dự đoán và độ tin cậy (confidence) lên khung hình.

### 5.2.3 Ứng dụng camera sử dụng MobileNetV2

Đối với MobileNetV2, ảnh đầu vào được chuẩn hóa bằng cách chia giá trị pixel cho 255.0 để đưa về khoảng [0, 1]. Mô hình dự đoán xác suất thuộc về từng lớp rác, sau đó lớp có xác suất cao nhất được chọn làm kết quả.

Ưu điểm của phiên bản này là tốc độ xử lý nhanh, độ trễ thấp, phù hợp với các thiết bị có tài nguyên phần cứng hạn chế. Tuy nhiên, mô hình không có cơ chế từ chối dự đoán khi độ tin cậy thấp.

### 5.2.4 Ứng dụng camera sử dụng EfficientNetB0

Đối với EfficientNetB0, ảnh đầu vào được tiền xử lý bằng hàm `preprocess_input` đúng theo chuẩn của mô hình. Ngoài ra, ứng dụng sử dụng ngưỡng độ tin cậy (*confidence threshold*) để xác định trường hợp “không có rác”.

Cụ thể, nếu xác suất dự đoán lớn nhất nhỏ hơn ngưỡng xác định, hệ thống sẽ không gán nhãn rác cho khung hình. Cách tiếp cận này giúp giảm các dự đoán sai trong môi trường thực tế khi camera ghi nhận các vật thể không thuộc tập rác đã huấn luyện.

## 5.3 Ứng dụng phân loại rác sử dụng Streamlit

### 5.3.1 Giới thiệu Streamlit

Streamlit là một framework hỗ trợ xây dựng nhanh các ứng dụng web cho học máy và khoa học dữ liệu. Trong đề tài này, Streamlit được sử dụng để xây dựng giao diện thân thiện, cho phép người dùng dễ dàng tương tác với hệ thống phân loại rác.

### 5.3.2 Chức năng của ứng dụng

Ứng dụng Streamlit được xây dựng với các chức năng chính:

- Upload ảnh từ máy tính và thực hiện phân loại rác.
- Phân loại rác theo thời gian thực thông qua camera.
- Hiển thị nhãn dự đoán và độ tin cậy của mô hình.

### 5.3.3 Ứng dụng Streamlit sử dụng MobileNetV2

Trong phiên bản sử dụng MobileNetV2, ảnh được resize về kích thước chuẩn và chuẩn hóa về khoảng [0, 1]. Mô hình được load một lần và lưu trong bộ nhớ đệm bằng `st.cache_resource` nhằm tăng tốc độ xử lý.

Phiên bản này cho tốc độ phản hồi nhanh và giao diện đơn giản, phù hợp cho các thiết bị có cấu hình trung bình.

### 5.3.4 Ứng dụng Streamlit sử dụng EfficientNetB0

Phiên bản Streamlit sử dụng EfficientNetB0 được tích hợp thêm cơ chế từ chối dự đoán dựa trên ngưỡng độ tin cậy. Nếu xác suất dự đoán thấp hơn ngưỡng xác định, hệ thống sẽ hiển thị kết quả “không có rác”.

Cách tiếp cận này giúp nâng cao tính ổn định và độ tin cậy của hệ thống khi triển khai trong môi trường thực tế, nơi dữ liệu đầu vào có thể rất đa dạng và không hoàn toàn giống tập huấn luyện.

## 5.4 Đánh giá khả năng ứng dụng thực tế

Qua quá trình triển khai và thử nghiệm, có thể rút ra một số nhận xét:

- MobileNetV2 phù hợp cho các hệ thống yêu cầu tốc độ cao và tài nguyên hạn chế.
- EfficientNetB0 cho độ chính xác cao hơn và ổn định hơn trong môi trường thực tế.
- Việc kết hợp mô hình học sâu với camera và giao diện Streamlit cho thấy tiềm năng ứng dụng của hệ thống trong phân loại rác thải sinh hoạt tại hộ gia đình.

Các ứng dụng được xây dựng trong đề tài là bước thử nghiệm quan trọng, chứng minh khả năng triển khai thực tế của các mô hình trí tuệ nhân tạo trong bài toán phân loại rác thải.

# Chương 6

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Trong đề tài này, nhóm đã xây dựng và triển khai thành công một hệ thống phân loại rác thải dựa trên các mô hình học sâu (Deep Learning), đáp ứng được các mục tiêu ban đầu về độ chính xác, khả năng triển khai và tính ứng dụng trong thực tế.

Cụ thể, các nội dung chính của đề tài bao gồm:

- Thu thập, tiền xử lý và xây dựng tập dữ liệu ảnh rác thải với nhiều lớp khác nhau như: glass, metal, organic, paper, plastic và trash.
- Triển khai và huấn luyện các mô hình học sâu tiêu biểu gồm MobileNetV2 và EfficientNetB0 cho bài toán phân loại ảnh, cùng với YOLOv8 cho bài toán phát hiện và phân loại rác trong ảnh thực tế.
- Đánh giá hiệu năng của các mô hình thông qua các chỉ số phổ biến như Accuracy, Precision, Recall, F1-score và ma trận nhầm lẫn (Confusion Matrix).
- Xây dựng ứng dụng thử nghiệm cho phép phân loại rác từ ảnh tĩnh và camera real-time thông qua giao diện web Streamlit.

Kết quả thực nghiệm cho thấy MobileNetV2 có ưu điểm về kiến trúc gọn nhẹ và tốc độ suy luận nhanh, phù hợp với các thiết bị có tài nguyên hạn chế. EfficientNetB0 đạt hiệu năng phân loại cao nhất với độ chính xác và khả năng tổng quát hóa tốt. Trong khi đó, YOLOv8 thể hiện khả năng phát hiện và phân loại nhiều loại rác trong cùng một ảnh với độ chính xác cao và tốc độ xử lý nhanh, phù hợp cho các ứng dụng thực tế.

Nhìn chung, đề tài đã chứng minh được tính khả thi của việc ứng dụng các mô hình học sâu trong bài toán phân loại rác thải, góp phần hỗ trợ các hệ thống quản lý rác thông minh và nâng cao hiệu quả phân loại rác tự động.

### 6.2 Hạn chế của đề tài

Bên cạnh các kết quả đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định:

- Tập dữ liệu sử dụng trong nghiên cứu chưa thực sự lớn và đa dạng, một số lớp rác còn mất cân bằng về số lượng mẫu, ảnh hưởng đến khả năng tổng quát của mô hình.

- Các điều kiện môi trường trong dữ liệu thực tế như ánh sáng, góc chụp, nền ảnh còn nhiều nhiễu, gây khó khăn cho mô hình trong một số trường hợp.
- Hệ thống hiện tại chủ yếu được triển khai ở mức độ thử nghiệm, chưa được tối ưu sâu cho các thiết bị nhúng hoặc các hệ thống thời gian thực có tài nguyên hạn chế.

### 6.3 Hướng phát triển

Trong tương lai, đề tài có thể được mở rộng và phát triển theo một số hướng sau:

- Mở rộng và đa dạng hóa tập dữ liệu huấn luyện bằng cách thu thập thêm dữ liệu thực tế hoặc áp dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation).
- Nghiên cứu và thử nghiệm thêm các mô hình học sâu tiên tiến hơn như EfficientNet phiên bản cao hơn, Vision Transformer (ViT) hoặc các mô hình kết hợp (Hybrid Models).
- Tối ưu mô hình để triển khai trên các thiết bị nhúng như Raspberry Pi, Jetson Nano, phục vụ cho các hệ thống phân loại rác thông minh ngoài thực tế.
- Kết hợp hệ thống phân loại rác với các cảm biến và cơ cấu chấp hành (băng chuyền, cánh tay robot) để xây dựng một hệ thống phân loại rác tự động hoàn chỉnh.

# Tài liệu tham khảo

- [1] Lambton College, Canada, Suman Kunwar: Health Care Waste Classification Using Deep Learning Aligned with Nepal's Bin Color Guidelines. Available: <https://arxiv.org/html/2508.07450v1>
- [2] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2021.
- [4] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
- [6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *NIPS*, 2012.
- [8] M. Sandler et al., “MobileNetV2: Inverted residuals and linear bottlenecks,” *CVPR*, 2018.
- [9] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for CNNs,” *ICML*, 2019.
- [10] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” *CVPR*, 2016.
- [11] Ultralytics, *YOLOv8 Documentation*, 2023.

Stanford University, “CS231n: Convolutional Neural Networks for Visual Recognition.” [Online]. Available: <https://cs231n.stanford.edu>

MIT, “Introduction to Deep Learning.” [Online]. Available: <https://introtodeeplearning.com>

# Phụ lục A

## Cấu trúc thư mục dự án

Cấu trúc thư mục của dự án được tổ chức như sau:

```
AI_Project/
  data/
    train/
    val/
    test/
  scripts/
    train_mobilenetv2.ipynb
    train_efficientnetb0.ipynb
    train_yolov8.ipynb
    evaluate.py
  model/
    model.h5
    best_efficientnetb0.keras
  app/
    camera_demo_mobilenet.py
    camera_demo_efficientnet.py
    streamlit_mobilenet.py
    streamlit_efficientnet.py
  figures/
    confusion_mobilenetv2.png
    confusion_efficientnetb0.png
    yolo_detect.jpg
```

Cấu trúc trên giúp tách biệt rõ ràng giữa dữ liệu, mã huấn luyện, mô hình đã train và các ứng dụng triển khai thực tế.

## Phụ lục B

### Tham số huấn luyện mô hình

Bảng dưới đây trình bày các tham số huấn luyện chính của các mô hình được sử dụng trong đề tài.

Bảng B.1: Tham số huấn luyện các mô hình

| Tham số        | MobileNetV2              | EfficientNetB0           | YOLOv8           |
|----------------|--------------------------|--------------------------|------------------|
| Kích thước ảnh | $224 \times 224$         | $224 \times 224$         | $640 \times 640$ |
| Batch size     | (theo notebook)          | (theo notebook)          | (theo YOLO)      |
| Epoch          | (theo notebook)          | (theo notebook)          | (theo YOLO)      |
| Optimizer      | Adam                     | Adam                     | SGD / AdamW      |
| Loss function  | Categorical Crossentropy | Categorical Crossentropy | YOLO Loss        |

## Phụ lục C

### Kết quả đánh giá

#### C.1 Kết quả đánh giá MobileNetV2

- Accuracy: 80.39%
- Precision: 80.56%
- Recall: 80.39%
- F1-score: 80.29%

#### C.2 Kết quả đánh giá EfficientNetB0

- Test Loss: 0.2375
- Test Accuracy: 92.84%
- Macro F1-score: 0.91

#### C.3 Kết quả đánh giá YOLOv8

- Accuracy (không tính reject): 92.94%
- Macro F1-score: 92.72%
- Weighted F1-score: 92.99%

## Phụ lục D

### Mô tả ứng dụng triển khai

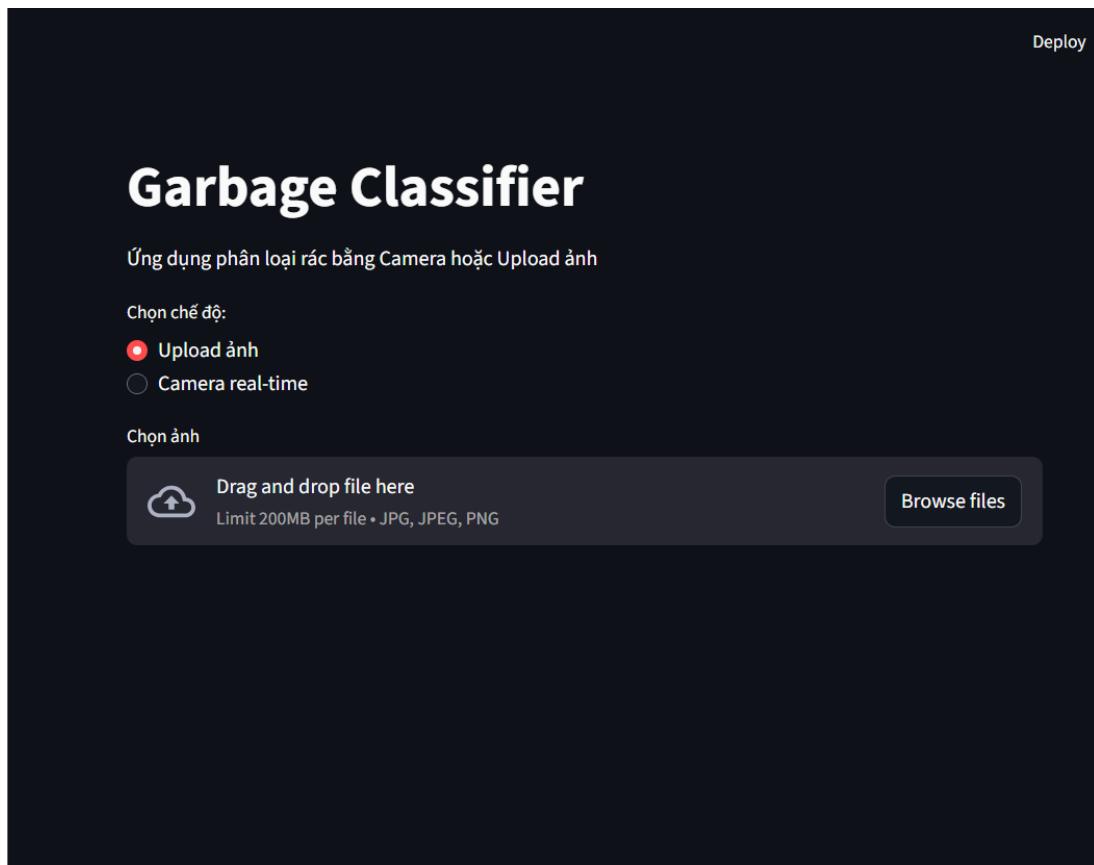
Hai loại ứng dụng đã được xây dựng trong đề tài bao gồm:

- Ứng dụng camera real-time sử dụng OpenCV.



Hình D.1: Minh họa ứng dụng camera<sub>demo</sub>

- Ứng dụng giao diện web sử dụng Streamlit.



Hình D.2: Minh họa ứng dụng Streamlit

Các ứng dụng cho phép phân loại rác trực tiếp từ camera hoặc ảnh tải lên, hiển thị kết quả dự đoán cùng độ tin cậy của mô hình, qua đó chứng minh khả năng ứng dụng thực tế của hệ thống.