**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**MÔN NGÔN NGỮ LẬP TRÌNH PYTHON**

**Tên đề tài: Xây dựng ứng dụng nhận diện rác thải**

**Sinh viên thực hiện:**

**Lê Quang Kiệt – 3123410180  
Đào Song Lộc – 3123410200  
Đỗ Trần Huy Bảo – 31234  
Phạm Phan Đức Khải – 3123410146**

**Giảng viên hướng dẫn:**

**ThS.Lê Tấn Long**

Thành phố Hồ Chí Minh, tháng 10 năm 2025

**NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN**

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Đại Học Sài Gòn, đặc biệt là các thầy cô ở khoa Công nghệ Thông tin của trường đã tạo điều kiện cho em tiếp cận và tìm hiểu để hoàn thành đồ án môn học lần này. Và em cũng xin chân thành cám ơn thầy Lê Tấn Long, giáo viên giảng dạy đã nhiệt tình hướng dẫn chúng em hoàn thành đồ án lần này.

Trong quá trình nghiên cứu và làm bài báo cáo đồ án, do kiến thức cũng như kinh nghiệm thực tế còn nhiều hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được ý kiến đóng góp thầy, cô để em học hỏi được nhiều kĩ năng, kinh nghiệm và sẽ hoàn thành tốt hơn cho những bài báo cáo sắp tới.

Em xin chân thành cảm ơn thầy ạ!

**MỤC LỤC**

[**PHẦN I. MỞ ĐẦU** 5](#_Toc212051222)

[**1.** **GIỚI THIỆU ĐỀ TÀI** 5](#_Toc212051223)

[**2.** **LÝ DO CHỌN ĐỀ TÀI** 5](#_Toc212051224)

[**3.** **MỤC ĐÍCH – MỤC TIÊU CỦA ĐỀ TÀI** 6](#_Toc212051225)

[**4.** **ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU** 7](#_Toc212051226)

[**4.1 Đối tượng nghiên cứu** 7](#_Toc212051227)

[**4.2 Phạm vi nghiên cứu** 7](#_Toc212051228)

[**5.** **YÊU CẦU ĐỒ ÁN** 7](#_Toc212051229)

[**PHẦN II. TỔNG QUAN CƠ SỞ LÝ THUYẾT** 8](#_Toc212051230)

[**1.** **GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH PYTHON** 8](#_Toc212051231)

[**2.** **TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO** 9](#_Toc212051232)

[**3.** **TỔNG QUAN VỀ THỊ GIÁC MÁY TÍNH (COMPUTER VISION)** 11](#_Toc212051233)

[**4.** **MẠNG NƠ – RON TÍCH CHẬP (CONVOLUTIONAL NEURAL NETWORK – CNN)** 12](#_Toc212051234)

[**5.** **BÀI TOÁN NHẬN DIỆN ĐỐI TƯỢNG VÀ CÁC HƯỚNG TIẾP CẬN** 13](#_Toc212051235)

[**6.** **TÌM HIỂU SÂU VỀ MÔ HÌNH YOLOv8** 14](#_Toc212051236)

[**6.1 Kiến trúc của YOLOv8** 14](#_Toc212051237)

[**6.2 Ưu điểm của YOLOv8** 15](#_Toc212051238)

[**7.** **GIỚI THIỆU VỀ CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG** 15](#_Toc212051239)

[**8. GIỚI THIỆU SƠ LƯỢC VỀ CÁCH HOẠT ĐỘNG CỦA ỨNG DỤNG NHẬN DIỆN RÁC THẢI** 16](#_Toc212051240)

[**8.1 Thu thập và Gán nhãn Dữ liệu:** 16](#_Toc212051241)

[**8.2 Huấn luyện mô hình YOLOv8:** 16](#_Toc212051242)

[**8.3 Nhận diện và Phân loại:** 17](#_Toc212051243)

[**8.4 Hiển thị kết quả:** 17](#_Toc212051244)

[**PHẦN III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 18](#_Toc212051245)

[**1.** **QUY TRÌNH THỰC HIỆN TỔNG THỂ** 18](#_Toc212051246)

[**2.** **XÂY DỰNG VÀ XỬ LÝ DỮ LIỆU** 18](#_Toc212051247)

[**2.1 Nguồn dữ liệu** 18](#_Toc212051248)

[**2.2** **Gán nhãn dữ liệu (Anotation)** 21](#_Toc212051249)

[**2.3** **Phân chia tập dữ liệu** 23](#_Toc212051250)

[**2.4** **Tăng cường dữ liệu** 24](#_Toc212051251)

[**3.** **THIẾT KẾ MÔ HÌNH VÀ QUÁ TRÌNH HUẤN LUYỆN** 27](#_Toc212051252)

[**3.1 Lựa chọn phiên bản YOLOv8** 27](#_Toc212051253)

[**3.2 Môi trường huấn luyện** 28](#_Toc212051254)

[**3.3 Cấu hình các siêu tham số (Hyperparameters)** 30](#_Toc212051255)

[**3.4 Các chỉ số đánh giá mô hình** 31](#_Toc212051256)

[**PHẦN IV. KẾT QUẢ THỰC NGHIỆM** 33](#_Toc212051257)

[**1.** **KẾT QUẢ HUẤN LUYỆN** 33](#_Toc212051258)

[**2.** **ĐÁNH GIÁ HIỆU NĂNG MÔ HÌNH** 35](#_Toc212051259)

[**3. PHÂN TÍCH VÀ THẢO LUẬN ĐIỂM MẠNH, HẠN CHẾ CỦA MÔ HÌNH** 37](#_Toc212051260)

[**3.1 Phân tích và thảo luận kết quả** 37](#_Toc212051261)

[**3.2 Demo ứng dụng** 39](#_Toc212051262)

[**PHẦN V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 39](#_Toc212051263)

[**1. KẾT LUẬN** 40](#_Toc212051264)

[**2. HẠN CHẾ CỦA ĐỀ TÀI** 40](#_Toc212051265)

[**3. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI** 41](#_Toc212051266)

[**TÀI LIỆU THAM KHẢO** 42](#_Toc212051267)

# **PHẦN I. MỞ ĐẦU**

## **GIỚI THIỆU ĐỀ TÀI**

Tên đề tài: Xây dựng ứng dụng nhận diện rác thải bằng ngôn ngữ lập trình Python

## **LÝ DO CHỌN ĐỀ TÀI**

Trong những năm gần đây, vấn đề ô nhiễm môi trường, đặc biệt là từ rác thải sinh hoạt và rác thải nhựa, đã trở thành một thách thức lớn đối với toàn cầu. Việc phân loại và xử lý rác thải đúng cách không chỉ giúp giảm thiểu gánh nặng cho các bãi chôn lấp mà còn góp phần tái chế, tiết kiệm tài nguyên và bảo vệ sức khỏe con người. Tuy nhiên, thực tế tại nhiều địa phương cho thấy công tác phân loại rác thải tại nguồn vẫn chưa đạt hiệu quả cao do thiếu công cụ hỗ trợ và ý thức cộng đồng còn hạn chế.

Song song với đó, sự phát triển mạnh mẽ của **công nghệ thông tin** và đặc biệt là **trí tuệ nhân tạo (AI)** đã mở ra cơ hội để ứng dụng khoa học công nghệ vào giải quyết các vấn đề thực tiễn của đời sống. Trong đó, **Python** nổi bật là một ngôn ngữ lập trình phổ biến, đơn giản, dễ học, với hệ sinh thái thư viện phong phú hỗ trợ mạnh mẽ cho các bài toán **machine learning** và **deep learning**. Đây chính là nền tảng quan trọng để phát triển các ứng dụng thông minh trong nhiều lĩnh vực, bao gồm cả xử lý ảnh và nhận diện đối tượng.

Qua quá trình tìm hiểu, em nhận thấy việc kết hợp Python cùng các mô hình học sâu (Deep Learning) để xây dựng **ứng dụng nhận diện rác thải** là một hướng đi mang ý nghĩa thiết thực. Ứng dụng này có thể tự động phân loại các loại rác như rác hữu cơ, nhựa, nilon, kim loại hay rác thải y tế, từ đó hỗ trợ nâng cao hiệu quả quản lý rác thải, thúc đẩy tái chế và giảm thiểu ô nhiễm môi trường.

Chính vì vậy, em đã quyết định chọn đề tài **“Ứng dụng nhận diện rác thải bằng ngôn ngữ lập trình Python ứng dụng AI”** để thực hiện báo cáo. Đề tài không chỉ giúp em củng cố kiến thức về lập trình và trí tuệ nhân tạo, mà còn mang tính ứng dụng thực tiễn cao, góp phần nhỏ bé vào nỗ lực chung trong công tác bảo vệ môi trường.

## **MỤC ĐÍCH – MỤC TIÊU CỦA ĐỀ TÀI**

**Mục đích**

• Nắm vững và củng cố kỹ năng lập trình, đặc biệt là lập trình Python.

• Tìm hiểu và áp dụng các thư viện, công cụ của Python trong lĩnh vực trí tuệ nhân tạo và xử lý ảnh.

• Củng cố, nâng cao kiến thức về Machine Learning và Deep Learning thông qua việc xây dựng mô hình nhận diện rác thải.

• Hiểu rõ quy trình phát triển một ứng dụng AI từ giai đoạn thu thập dữ liệu, huấn luyện mô hình cho đến triển khai thực tiễn.

• Rèn luyện tư duy phân tích và khả năng giải quyết vấn đề thực tế thông qua việc ứng dụng công nghệ AI vào lĩnh vực môi trường.

**Mục tiêu**

• Vận dụng thành thạo ngôn ngữ Python và các nguyên lý lập trình hướng đối tượng trong quá trình xây dựng ứng dụng.

• Sử dụng hiệu quả các thư viện AI/Deep Learning của Python như TensorFlow, Keras, OpenCV… để huấn luyện và triển khai mô hình nhận diện rác thải.

• Xây dựng được một ứng dụng có khả năng phân loại cơ bản các loại rác thải như rác hữu cơ, nhựa, nilon, kim loại, rác y tế.

• Tạo ra sản phẩm có ý nghĩa thực tiễn, hỗ trợ công tác quản lý và phân loại rác thải, từ đó góp phần nâng cao ý thức bảo vệ môi trường.

## **ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

### **4.1 Đối tượng nghiên cứu**

Các loại rác thải sinh hoạt phổ biến. Cụ thể gồm 4 lớp: **Rắn, Nhựa, Y Tế, Nilong.**

### **4.2 Phạm vi nghiên cứu**

* Tập trung nghiên cứu và ứng dụng kiến trúc mô hình YOLOv8.
* Dữ liệu đầu vào bao gồm ảnh tĩnh và luồng video từ webcam.
* Ứng dụng được phát triển dưới dạng phần mềm demo chạy trên hệ điều hành Windows/macOS/Linux.
* Đề tài không đi sâu vào việc tối ưu hóa để triển khai trên các thiết bị nhúng cấu hình thấp hoặc tích hợp phần cứng.

## **YÊU CẦU ĐỒ ÁN**

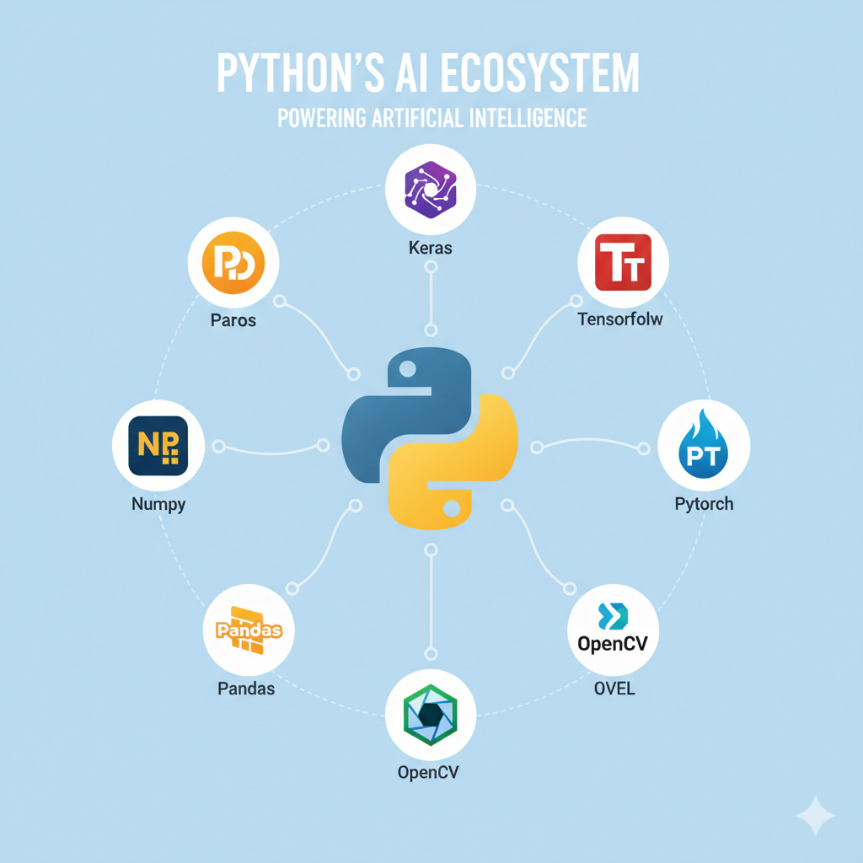
Đồ án môn học Ngôn ngữ lập trình Python yêu cầu sinh viên xây dựng một ứng dụng thực tế có khả năng nhận diện và phân loại rác thải dựa trên hình ảnh đầu vào. Ứng dụng phải được phát triển bằng ngôn ngữ Python, kết hợp với các kiến thức về lập trình hướng đối tượng, xử lý dữ liệu, và ứng dụng các thư viện hỗ trợ trí tuệ nhân tạo.

Sản phẩm cần đảm bảo:

* Có khả năng xử lý ảnh và áp dụng mô hình AI/Deep Learning để nhận diện các loại rác thải khác nhau.
* Giao diện đơn giản, dễ sử dụng, cho phép người dùng nhập dữ liệu (ảnh) và hiển thị kết quả phân loại.
* Mã nguồn được tổ chức hợp lý, có tính mở rộng và dễ bảo trì.
* Có báo cáo thuyết minh đầy đủ quy trình xây dựng: từ phân tích bài toán, thiết kế, triển khai, đến kết quả thử nghiệm.
* Có thể trình bày và demo trực tiếp sản phẩm trước lớp.

# **PHẦN II. TỔNG QUAN CƠ SỞ LÝ THUYẾT**

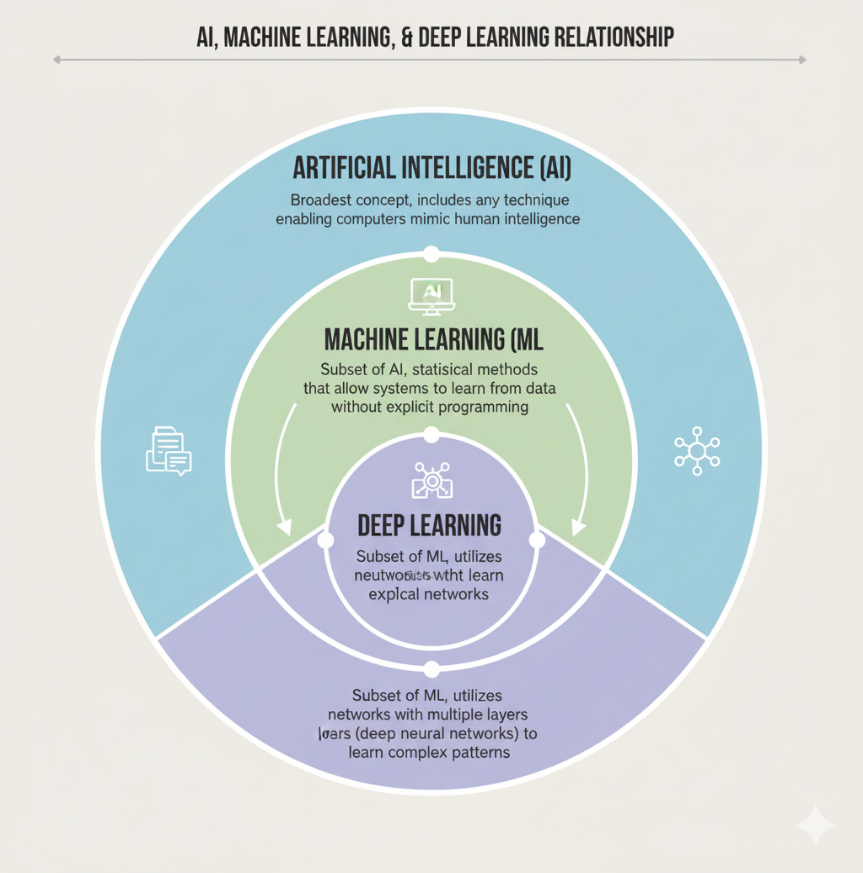
## **GIỚI THIỆU VỀ NGÔN NGỮ LẬP TRÌNH PYTHON**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, van Rossum đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guidovan Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

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

## **TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO**

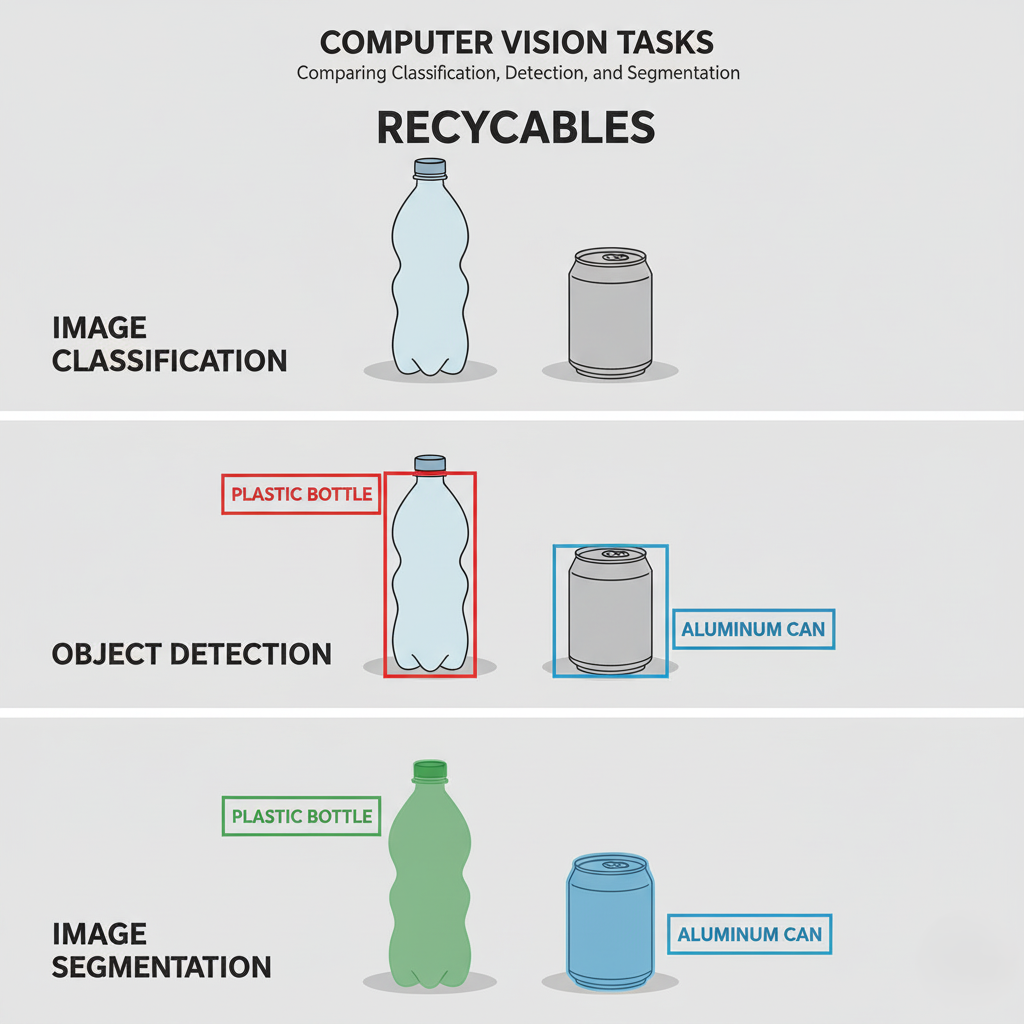
***Trí tuệ nhân tạo (Artificial Intelligence):*** là lĩnh vực nghiên cứu nhằm tạo ra các hệ thống có thể học hỏi, suy nghĩ và hành động như con người. AI bao gồm nhiều nhánh như: học máy (machine learning), thị giác máy tính (computer vision), xử lý ngôn ngữ tự nhiên (NLP), robot thông minh,…

Trong đồ án này, AI được ứng dụng vào thị giác máy tính để nhận diện rác thải – một bài toán phổ biến và có ý nghĩa thực tiễn, nâng cao ý thức bảo vệ môi trường xanh sạch đẹp.

***Machine Learning (ML):*** Cho phép máy học từ dữ liệu để đưa ra dự đoán hoặc phân loại mà không cần lập trình cụ thể từng bước. Trong bối cảnh nhận diện rác thải, ML có thể được dùng để huấn luyện mô hình từ các bộ dữ liệu ảnh đã được gán nhãn (ví dụ: rác nhựa, rác hữu cơ, rác y tế…).

***Deep Learning (DL):*** Là một nhánh nâng cao của ML, sử dụng mạng neuron sâu (**Deep Neural Networks**) để xử lý dữ liệu phức tạp. Deep Learning đặc biệt hiệu quả trong các bài toán liên quan đến hình ảnh, video và âm thanh. Trong dự án này, hệ thống sử dụng các mô hình **Convolutional Neural Networks (CNNs)** – một kiến trúc phổ biến của Deep Learning trong lĩnh vực thị giác máy tính, nhằm trích xuất đặc trưng từ ảnh rác thải và phân loại theo từng nhóm.

## **TỔNG QUAN VỀ THỊ GIÁC MÁY TÍNH (COMPUTER VISION)**

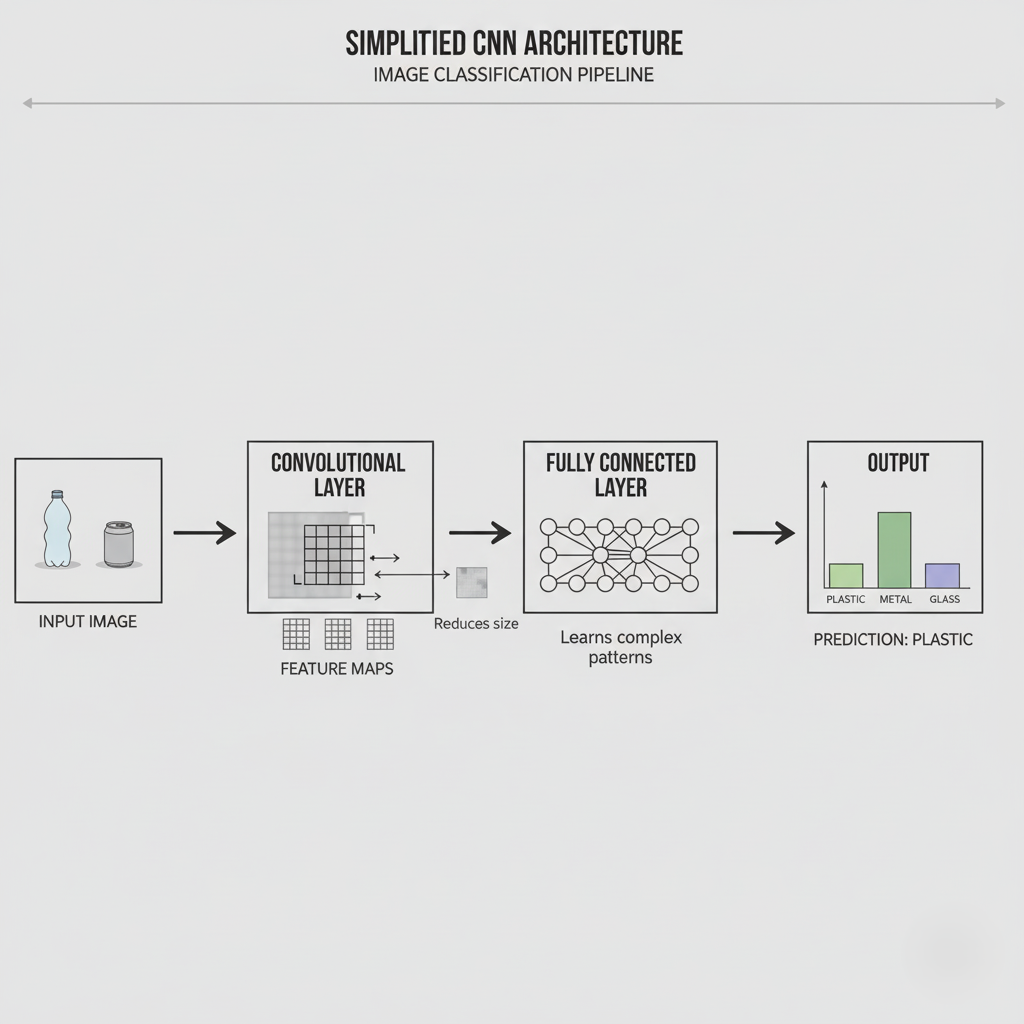
Thị giác máy tính là một lĩnh vực của trí tuệ nhân tạo, cho phép máy tính và các hệ thống có thể "nhìn", diễn giải và hiểu thông tin hình ảnh từ thế giới thực. Mục tiêu của thị giác máy tính là tự động hóa các công việc mà hệ thống thị giác của con người có thể làm. Các bài toán phổ biến trong lĩnh vực này bao gồm:

* **Phân loại ảnh (Image Classification):** Gán một nhãn duy nhất cho toàn bộ bức ảnh (ví dụ: ảnh này là "con chó").
* **Nhận diện đối tượng (Object Detection):** Xác định vị trí (bằng hộp giới hạn - bounding box) và phân loại nhiều đối tượng trong cùng một bức ảnh (ví dụ: ảnh này có "con chó" ở vị trí A và "cái cây" ở vị trí B).
* **Phân đoạn ảnh (Image Segmentation):** Phân loại từng pixel trong ảnh thuộc về đối tượng nào, tạo ra một đường viền chính xác quanh vật thể thay vì chỉ dùng hộp giới hạn.

Trong khuôn khổ đề tài này, chúng ta tập trung vào bài toán **Nhận diện đối tượng**

## **MẠNG NƠ – RON TÍCH CHẬP (CONVOLUTIONAL NEURAL NETWORK – CNN)**

***CNN*** là kiến trúc học sâu đột phá, được thiết kế đặc biệt để xử lý dữ liệu dạng lưới như hình ảnh. Nó là nền tảng cho hầu hết các mô hình thị giác máy tính hiện đại. Các thành phần chính của một mạng CNN bao gồm:

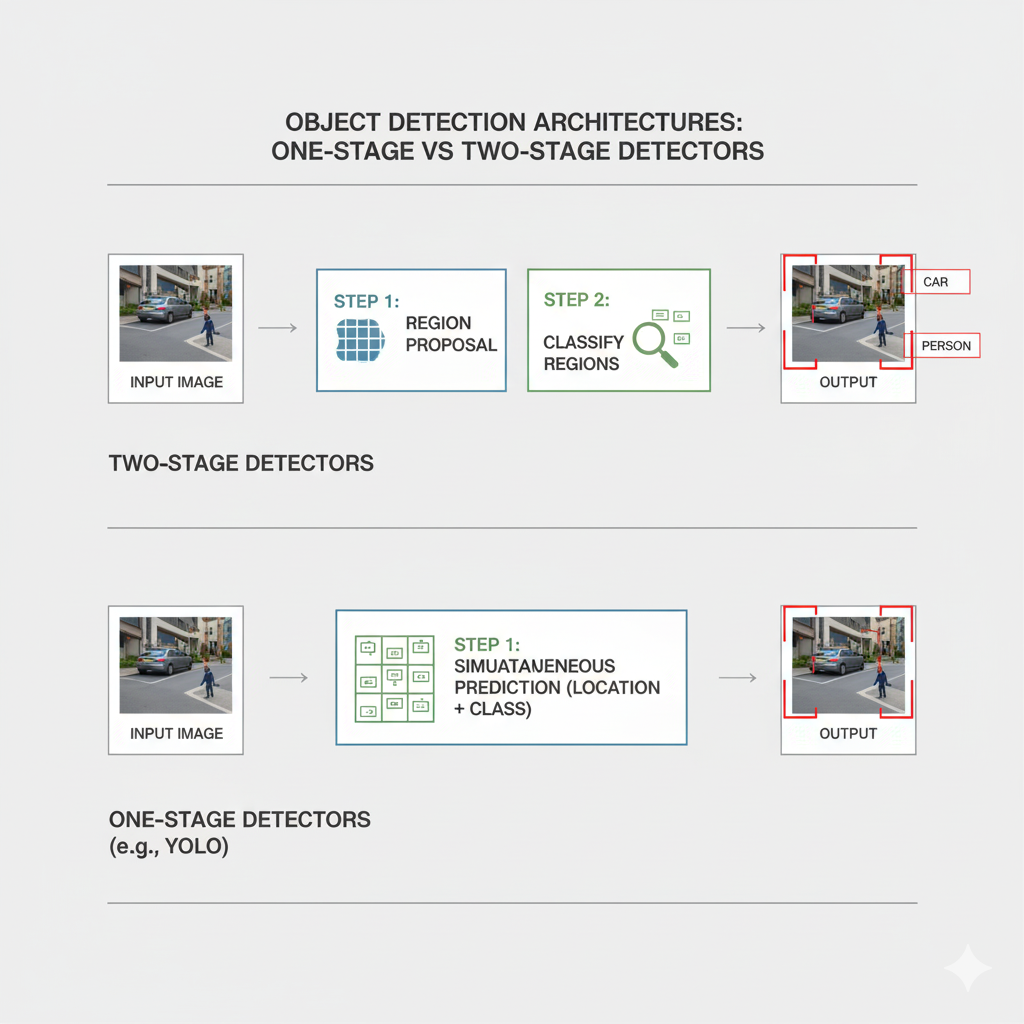


***Lớp Tích chập*** (Convolutional Layer): Sử dụng các bộ lọc (filter/kernel) để trượt qua ảnh đầu vào, thực hiện phép tích chập để phát hiện các đặc trưng như cạnh, góc, màu sắc.

***Lớp Gộp*** (Pooling Layer): Giảm kích thước không gian của bản đồ đặc trưng, giúp giảm số lượng tham số tính toán và kiểm soát overfitting.

***Lớp Kết nối đầy đủ*** (Fully Connected Layer): Các nơ-ron được kết nối đầy đủ với tất cả các kích hoạt ở lớp trước, thường được đặt ở cuối mạng để thực hiện nhiệm vụ phân loại.

## **BÀI TOÁN NHẬN DIỆN ĐỐI TƯỢNG VÀ CÁC HƯỚNG TIẾP CẬN**



Các mô hình nhận diện đối tượng hiện đại chủ yếu đi theo hai hướng:

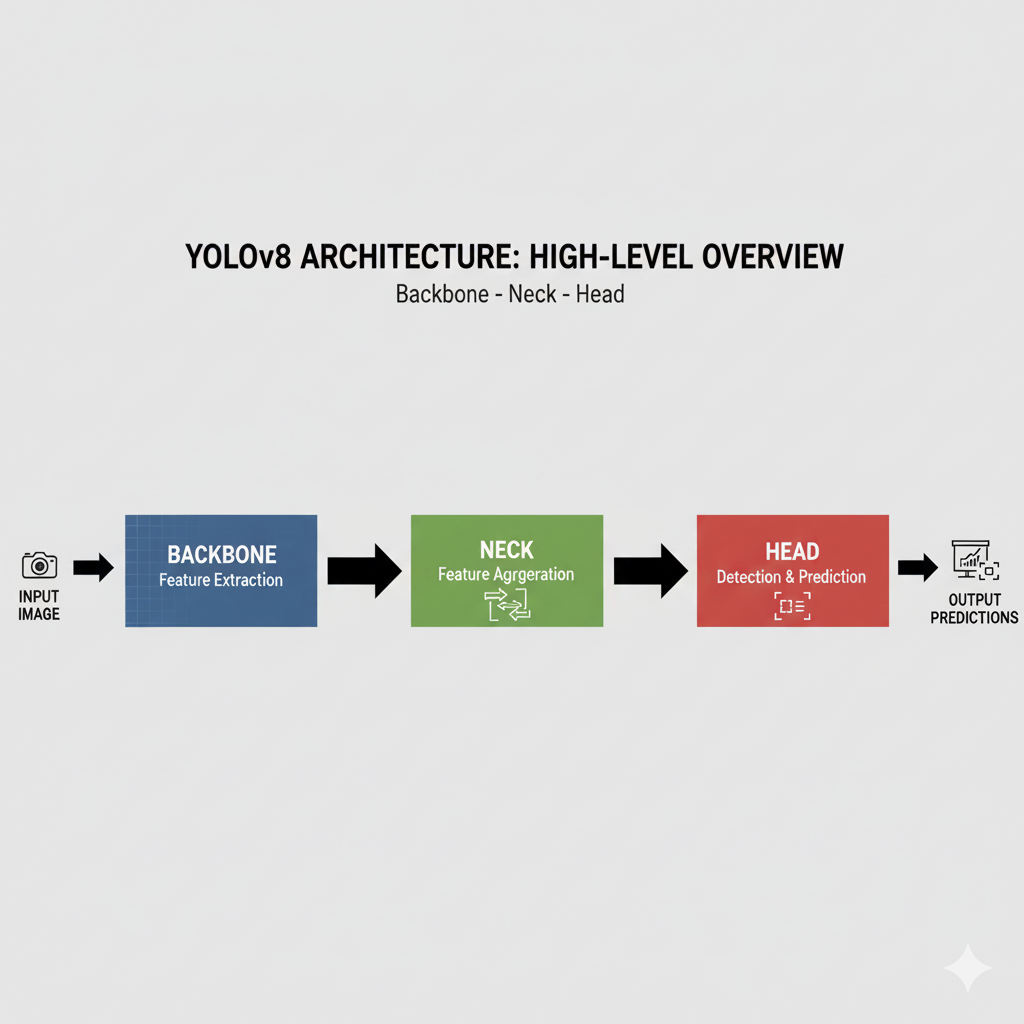
* **Two-stage detectors (Máy dò hai giai đoạn):** Hoạt động bằng cách đầu tiên đề xuất các vùng có khả năng chứa vật thể (Region Proposals), sau đó phân loại các vùng này. Ví dụ tiêu biểu là họ mô hình R-CNN, Fast R-CNN. Ưu điểm của chúng là độ chính xác cao, nhưng nhược điểm là tốc độ xử lý chậm.
* **One-stage detectors (Máy dò một giai đoạn):** Xử lý toàn bộ ảnh trong một lần duy nhất, đồng thời dự đoán cả bounding box và lớp đối tượng. Ví dụ tiêu biểu là SSD và đặc biệt là họ mô hình YOLO. Ưu điểm của chúng là tốc độ cực nhanh, phù hợp cho các ứng dụng thời gian thực.

## **TÌM HIỂU SÂU VỀ MÔ HÌNH YOLOv8**

### **6.1 Kiến trúc của YOLOv8**

YOLO (You Only Look Once) là một cuộc cách mạng trong lĩnh vực nhận diện đối tượng thời gian thực. YOLOv8 là phiên bản mới nhất được phát triển bởi Ultralytics, kế thừa và cải tiến từ các thế hệ trước.

Mô hình bao gồm 3 phần chính:

* 1. **Backbone (Xương sống):** Là một mạng CNN (dựa trên kiến trúc CSPDarknet) có nhiệm vụ trích xuất các đặc trưng phong phú từ ảnh đầu vào ở nhiều tỷ lệ khác nhau.
  2. **Neck (Cổ):** Sử dụng các cơ chế như PANet (Path Aggregation Network) để kết hợp và tổng hợp các đặc trưng từ Backbone, giúp mô hình nhận diện tốt cả vật thể lớn và nhỏ.
  3. **Head (Đầu):** Là phần cuối cùng của mạng, chịu trách nhiệm đưa ra các dự đoán cuối cùng về vị trí bounding box, điểm tin cậy (confidence score) và nhãn lớp cho các đối tượng.

### **6.2 Ưu điểm của YOLOv8**

* 1. **Hiệu năng cao:** Đạt được sự cân bằng xuất sắc giữa tốc độ và độ chính xác.
  2. **Linh hoạt:** Cung cấp nhiều phiên bản (n, s, m, l, x) để người dùng lựa chọn tùy theo yêu cầu ứng dụng.
  3. **Dễ sử dụng:** Thư viện của Ultralytics cung cấp một API đơn giản, dễ dàng cho việc huấn luyện, đánh giá và triển khai mô hình.

Những ưu điểm này làm cho YOLOv8 trở thành lựa chọn lý tưởng cho đề tài.

## **GIỚI THIỆU VỀ CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG**

**Python** – Ngôn ngữ lập trình chính.

**Thư viện chính:**

• **Ultralytics**: Thư viện cung cấp kiến trúc YOLOv8, các công cụ huấn luyện, và các mô hình đã được tiền huấn luyện (pre-trained).

• **OpenCV** – xử lý ảnh đầu vào, tiền xử lý dữ liệu và hiển thị kết quả.

• **NumPy, Pandas** – hỗ trợ xử lý dữ liệu và thao tác với ma trận, bảng dữ liệu.

• **Matplotlib/Seaborn** – trực quan hóa dữ liệu, hiển thị kết quả huấn luyện.

• **os, pathlib** – quản lý file, thư mục dataset.

• **Tkinter** là một gói trong Python có chứa module Tk hỗ trợ cho việc lập trình GUI. Tk ban đầu được viết cho ngôn ngữ Tcl. Sau đó Tkinter được viết ra để sử dụng Tk bằng trình thông dịch Tcl trên nền Python. Ngoài Tkinter ra còn có một số công cụ khác giúp tạo một ứng dụng GUI viết bằng Python như wxPython, PyQt, và PyGTK

**Môi trường:**

• Google Colab – để huấn luyện mô hình với GPU miễn phí.

• Máy tính cá nhân cài Python – để chạy thử nghiệm và demo ứng dụng.

## **8. GIỚI THIỆU SƠ LƯỢC VỀ CÁCH HOẠT ĐỘNG CỦA ỨNG DỤNG NHẬN DIỆN RÁC THẢI**

Ứng dụng hoạt động dựa trên nguyên lý nhận diện đối tượng trong thời gian thực của YOLOv8, bao gồm các bước chính sau:

### **8.1 Thu thập và Gán nhãn Dữ liệu:**

* + Đầu tiên, một bộ dữ liệu (dataset) lớn gồm các hình ảnh chứa nhiều loại rác thải khác nhau được thu thập.
  + Sau đó, sử dụng các công cụ như Roboflow hoặc LabelImg để vẽ các hộp giới hạn (bounding box) xung quanh từng đối tượng rác trong ảnh và gán nhãn tương ứng (ví dụ: chai nhựa, túi nilon, vỏ lon, rác hữu cơ).

### **8.2 Huấn luyện mô hình YOLOv8:**

* + Mô hình YOLOv8 được huấn luyện (train) trên bộ dữ liệu đã được gán nhãn.
  + Trong quá trình này, mô hình sẽ học cách xác định vị trí, kích thước và đặc điểm nhận dạng của từng loại rác thải thông qua kiến trúc mạng nơ-ron tích chập (CNN) phức tạp của nó. Mô hình sẽ tự động tối ưu hóa để vừa phân loại đúng đối tượng, vừa xác định vị trí chính xác của chúng trong ảnh.

### **8.3 Nhận diện và Phân loại:**

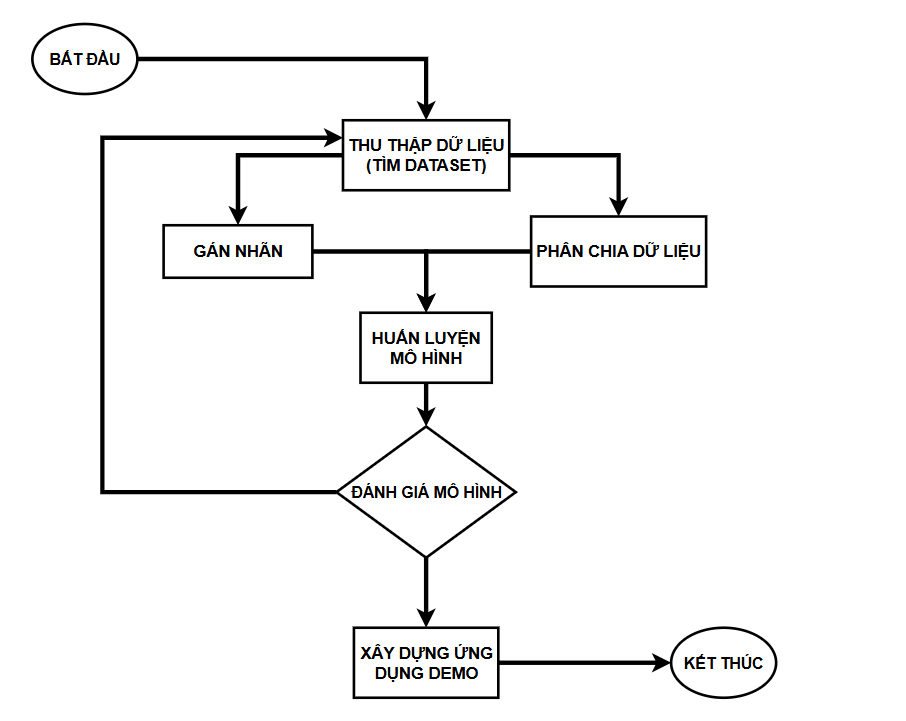
* + Khi một hình ảnh hoặc luồng video mới được đưa vào, mô hình **YOLOv8** sẽ quét qua và xử lý nó trong một lượt duy nhất ("*You Only Look Once*").
  + Mô hình sẽ dự đoán các bounding box xung quanh các vật thể mà nó cho là rác thải, đồng thời đưa ra nhãn phân loại (ví dụ: rác nhựa, rác y tế) và một độ tin cậy (confidence score) cho mỗi dự đoán.

### **8.4 Hiển thị kết quả:**

* + Cuối cùng, ứng dụng sử dụng OpenCV để vẽ các bounding box và nhãn lớp đã được mô hình dự đoán lên trên ảnh hoặc video gốc, giúp người dùng thấy được loại rác và vị trí của chúng một cách trực quan.

# **PHẦN III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **QUY TRÌNH THỰC HIỆN TỔNG THỂ**



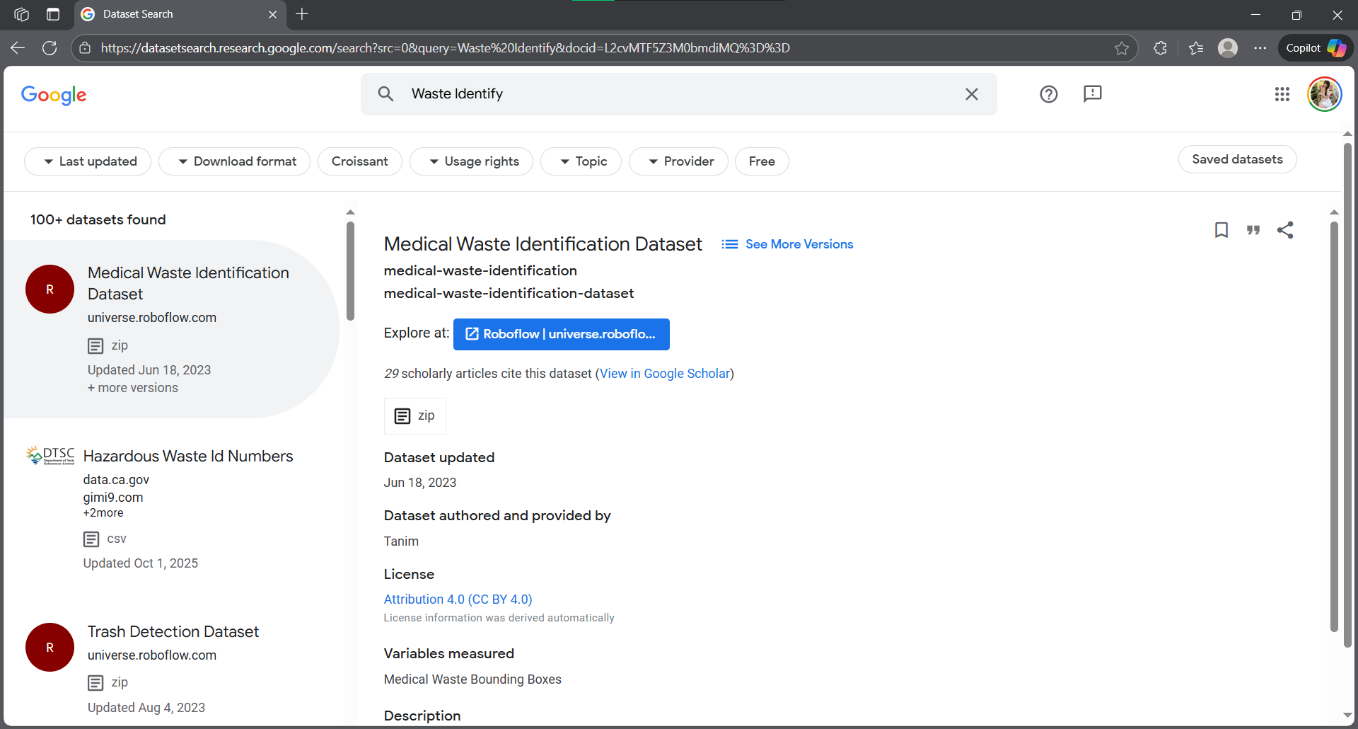
*Sơ đồ khối*

## **XÂY DỰNG VÀ XỬ LÝ DỮ LIỆU**

### **2.1 Nguồn dữ liệu**

Bộ dữ liệu được xây dựng từ các nguồn:

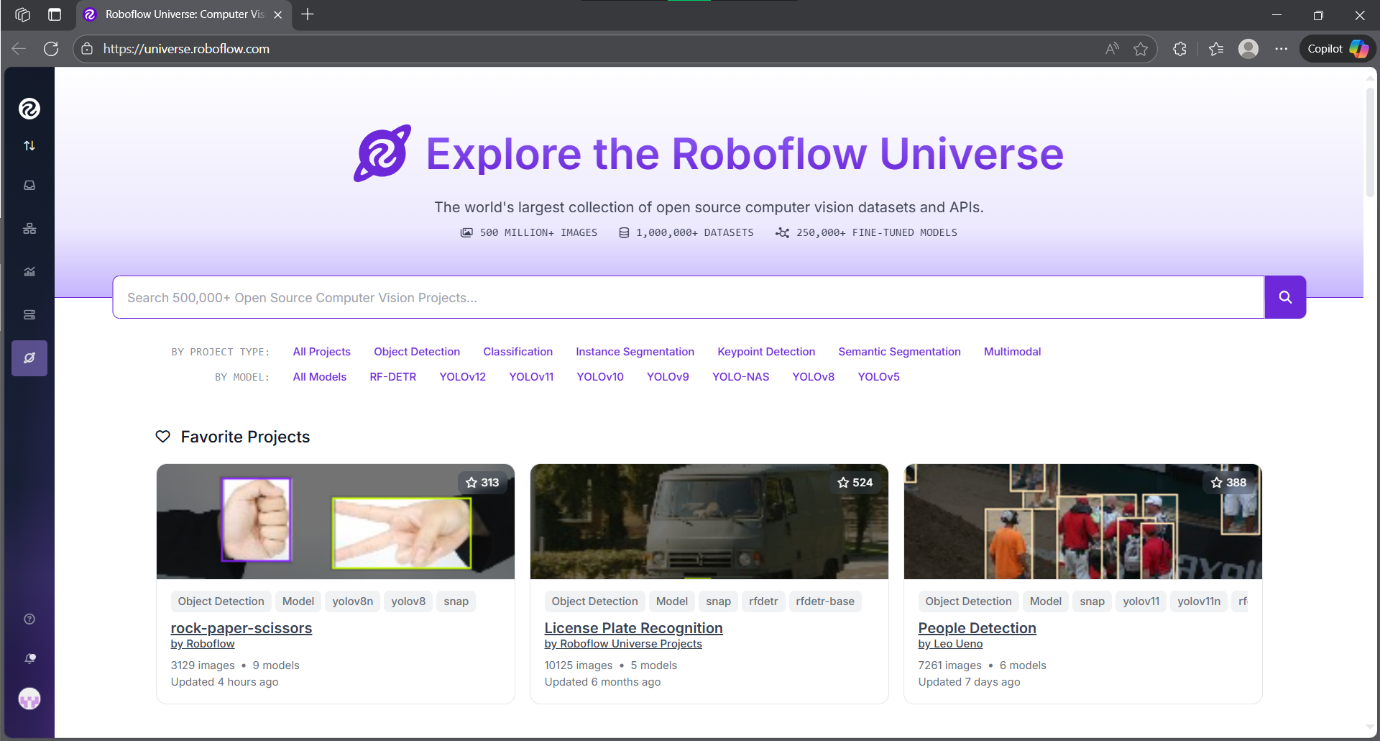
#### **2.1.1 Google Dataset Search**



*Hình ảnh Google Dataset Search*

Google Dataset Search là một công cụ tìm kiếm mạnh mẽ được Google phát triển để giúp người dùng dễ dàng khám phá và truy cập vào hàng triệu bộ dữ liệu từ khắp nơi trên web. Với giao diện trực quan, người dùng có thể tìm kiếm dữ liệu về nhiều lĩnh vực khác nhau như khoa học, xã hội, môi trường, kinh tế và nhiều hơn nữa. Công cụ này tổng hợp thông tin từ các kho lưu trữ dữ liệu công khai, các tổ chức nghiên cứu và các trang web chính phủ, cung cấp các mô tả chi tiết, thông tin cấp phép và liên kết trực tiếp đến nguồn dữ liệu gốc. Google Dataset Search là một tài nguyên vô giá cho các nhà nghiên cứu, nhà khoa học dữ liệu, sinh viên và bất kỳ ai quan tâm đến việc khám phá và sử dụng dữ liệu để phục vụ cho công việc hoặc học tập của mình.

#### **2.2.2 Roboflow**

Roboflow là một nền tảng toàn diện giúp các nhà phát triển và nhà khoa học dữ liệu xây dựng, triển khai và cải thiện các mô hình thị giác máy tính một cách nhanh chóng và hiệu quả. Nền tảng này cung cấp các công cụ mạnh mẽ từ việc quản lý dữ liệu, gán nhãn, tiền xử lý, tăng cường dữ liệu cho đến huấn luyện và triển khai mô hình. Roboflow hỗ trợ nhiều định dạng dữ liệu và tích hợp dễ dàng với các framework học sâu phổ biến như YOLO, TensorFlow và PyTorch. Đặc biệt, Roboflow cho phép người dùng dễ dàng chia sẻ và cộng tác trên các bộ dữ liệu và mô hình, thúc đẩy sự phát triển của cộng đồng thị giác máy tính. Với Roboflow, việc biến ý tưởng thị giác máy tính thành hiện thực trở nên đơn giản và nhanh chóng hơn bao giờ hết, giúp tiết kiệm thời gian và nguồn lực đáng kể.

*Hình ảnh Roboflow*

### **Gán nhãn dữ liệu (Anotation)**

#### **a) Mô tả nhãn đã chọn**

Sau khi thu thập được bộ dữ liệu hình ảnh rác thải thô, bước tiếp theo và cực kỳ quan trọng là gán nhãn (annotation) cho các đối tượng rác thải trong mỗi hình ảnh. Quá trình này giúp mô hình học máy hiểu được đâu là đối tượng cần phát hiện và vị trí của chúng.

Trong quá trình xây dựng ứng dụng nhận diện rác thải, chúng em đã tiến hành phân loại và lựa chọn các lớp rác thải cụ thể để mô hình có thể nhận diện. Việc lựa chọn này dựa trên tần suất xuất hiện, mức độ ảnh hưởng và khả năng phân biệt hình ảnh. Các lớp rác thải được chọn bao gồm:

* + 1. Nhựa (Plastic): Bao gồm các loại chai nhựa, túi nhựa, v.v..
    2. Nilong (Plastic bag): Bao gồm các loại bọc nilong, bao bì, v.v...
    3. Rắn (Hard Waste): Bao gồm các loại giấy, pin, thùng giấy, mảnh vỡ thủy tinh, v.v..
    4. Y tế (Medical Waste): Bao gồm các loại khẩu trang đã qua sử dụng, bông gòn dính máu, mũi kim chích, thuốc, v.v..

Mỗi lớp này sẽ được gán một **class\_id** duy nhất trong quá trình gán nhãn, ví dụ: 0 cho Chai Nhựa, 1 cho Ống Hút, v.v.

*Hình ảnh file data.yam*

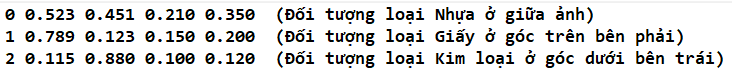
#### **b) Định dạng file nhãn (YOLO format)**

Để mô hình YOLOv8 có thể đọc và hiểu được thông tin về các đối tượng trong hình ảnh, các file nhãn phải tuân thủ một định dạng cụ thể. YOLO (You Only Look Once) sử dụng định dạng file .txt cho mỗi hình ảnh, với mỗi dòng trong file .txt đại diện cho một đối tượng được phát hiện và có cấu trúc như sau:

**<class\_id> <x\_center> <y\_center> <width> <height>**

Trong đó:

* **<class\_id>:** Là số nguyên đại diện cho lớp của đối tượng. Nó bắt đầu từ 0. Ví dụ: 0 cho "Nhựa", 1 cho "Giấy", 2 cho "Kim loại", v.v.
* **<x\_center>:** Tọa độ X của tâm điểm của bounding box, được chuẩn hóa từ 0 đến 1 (tức là chia cho chiều rộng của hình ảnh).
* **<y\_center>:** Tọa độ Y của tâm điểm của bounding box, được chuẩn hóa từ 0 đến 1 (tức là chia cho chiều cao của hình ảnh).
* **<width>:** Chiều rộng của bounding box, được chuẩn hóa từ 0 đến 1 (tức là chia cho chiều rộng của hình ảnh).
* **<height>:** Chiều cao của bounding box, được chuẩn hóa từ 0 đến 1 (tức là chia cho chiều cao của hình ảnh).

Ví dụ: ta có file ảnh là **image001.jpg**, thì file nhãn tương ứng là **image001.txt** và có nội dung như sau:

*Hình ảnh file image001.txt*

Quá trình gán nhãn được thực hiện cẩn thận cho toàn bộ tập dữ liệu, đảm bảo rằng mỗi đối tượng rác thải trong hình ảnh đều được đánh dấu chính xác với bounding box và class\_id phù hợp. Điều này là nền tảng cho việc huấn luyện mô hình YOLOv8 đạt hiệu suất cao.

### **Phân chia tập dữ liệu**

Sau khi hoàn tất quá trình thu thập và gán nhãn cho toàn bộ dữ liệu hình ảnh rác thải, bước tiếp theo là phân chia tập dữ liệu thành ba tập con riêng biệt: **Tập huấn luyện (Training Set)**, **Tập kiểm định (Validation Set)** và **Tập kiểm tra (Test Set)**. Việc phân chia này là một tiêu chuẩn quan trọng trong học máy, đảm bảo tính khách quan và khả năng tổng quát hóa của mô hình.

#### **2.3.1 Mục đích của việc phân chia tập dữ liệu**

* **Tập huấn luyện (Training Set):** Đây là tập dữ liệu lớn nhất, được sử dụng để "dạy" cho mô hình. Trong giai đoạn huấn luyện, mô hình sẽ học các đặc trưng và mẫu từ tập này để nhận diện các đối tượng rác thải.
* **Tập kiểm định (Validation Set):** Tập này được sử dụng để tinh chỉnh các siêu tham số (hyperparameters) của mô hình và đánh giá hiệu suất của mô hình trong quá trình huấn luyện. Nó giúp phát hiện và ngăn chặn hiện tượng quá khớp (overfitting), đảm bảo mô hình có thể tổng quát hóa tốt trên dữ liệu mới. Mô hình không trực tiếp học từ tập này nhưng hiệu suất trên tập này ảnh hưởng đến việc điều chỉnh mô hình.
* **Tập kiểm tra (Test Set):** Đây là tập dữ liệu hoàn toàn mới và độc lập, không được sử dụng trong bất kỳ giai đoạn huấn luyện hay tinh chỉnh nào. Mục đích của tập kiểm tra là cung cấp một đánh giá cuối cùng, không thiên vị về khả năng hoạt động của mô hình trên dữ liệu chưa từng thấy. Kết quả trên tập kiểm tra phản ánh hiệu suất thực tế của mô hình khi triển khai.

#### **2.3.2 Tỷ lệ phân chia**

Trong đề tài này, chúng em đã áp dụng tỷ lệ phân chia tập dữ liệu như sau:

* **Tập huấn luyện (Training Set): 80%**
* **Tập kiểm định (Validation Set): 10%**
* **Tập kiểm tra (Test Set): 10%**

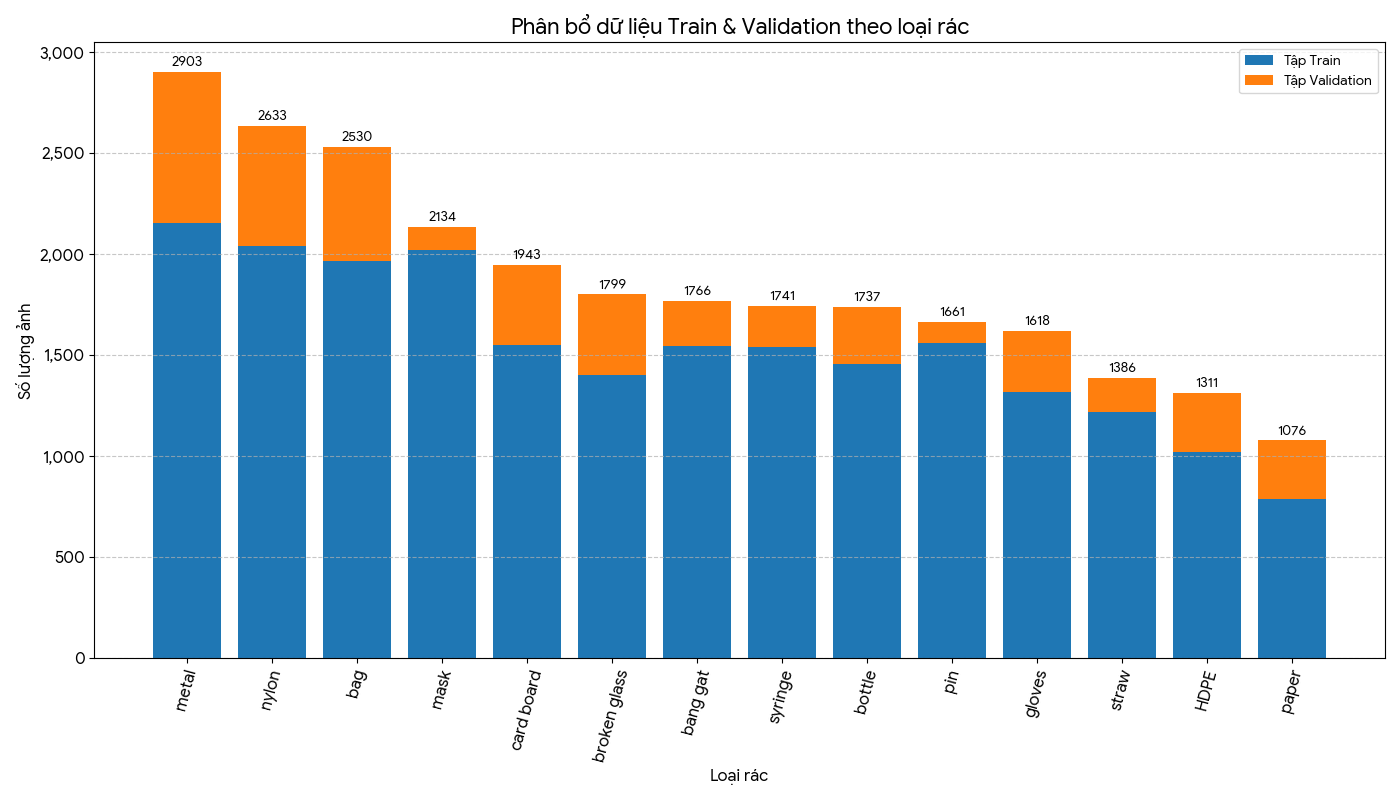
**Giải thích tỷ lệ:**

* **80% cho Tập huấn luyện:** Tỷ lệ lớn này đảm bảo rằng mô hình có đủ lượng dữ liệu đa dạng để học được các đặc trưng phức tạp của từng loại rác thải trong các điều kiện khác nhau (góc chụp, ánh sáng, kích thước, v.v.). Điều này là cần thiết để YOLOv8 có thể xây dựng một hiểu biết vững chắc về các đối tượng.
* **10% cho Tập kiểm định:** Một phần nhỏ hơn được dành cho tập kiểm định, đủ để cung cấp một cái nhìn đáng tin cậy về hiệu suất của mô hình trong quá trình huấn luyện và giúp chúng tôi điều chỉnh các tham số một cách hiệu quả mà không làm "rò rỉ" thông tin từ tập kiểm tra.
* **10% cho Tập kiểm tra:** Tương tự, 10% còn lại được dành cho tập kiểm tra. Việc giữ tập này hoàn toàn riêng biệt và không động chạm đến trong quá trình phát triển mô hình là cực kỳ quan trọng. Nó đảm bảo rằng kết quả đánh giá cuối cùng về độ chính xác và khả năng tổng quát hóa của mô hình là khách quan và đáng tin cậy.

Ví dụ minh họa:

Giả sử tổng số lượng hình ảnh trong bộ dữ liệu là 1000 hình ảnh:

* **Tập huấn luyện:** 800 hình ảnh
* **Tập kiểm định:** 100 hình ảnh
* **Tập kiểm tra:** 100 hình ảnh

Quá trình phân chia này thường được thực hiện tự động bằng các script hoặc các công cụ hỗ trợ như Roboflow, giúp đảm bảo tính ngẫu nhiên và đồng đều của dữ liệu trong mỗi tập con, tránh thiên vị và cải thiện khả năng tổng quát hóa của mô hình.

*Hình biểu đồ cột phân bố dữ liệu theo các class rác*

### **Tăng cường dữ liệu**

Tăng cường dữ liệu (Data Augmentation) là một kỹ thuật cực kỳ quan trọng trong học sâu, đặc biệt là đối với các bài toán thị giác máy tính, nhằm mở rộng kích thước và sự đa dạng của tập dữ liệu huấn luyện một cách nhân tạo. Kỹ thuật này giúp mô hình trở nên mạnh mẽ hơn, ít bị quá khớp (overfitting) với dữ liệu huấn luyện và có khả năng tổng quát hóa tốt hơn khi gặp dữ liệu mới trong môi trường thực tế.

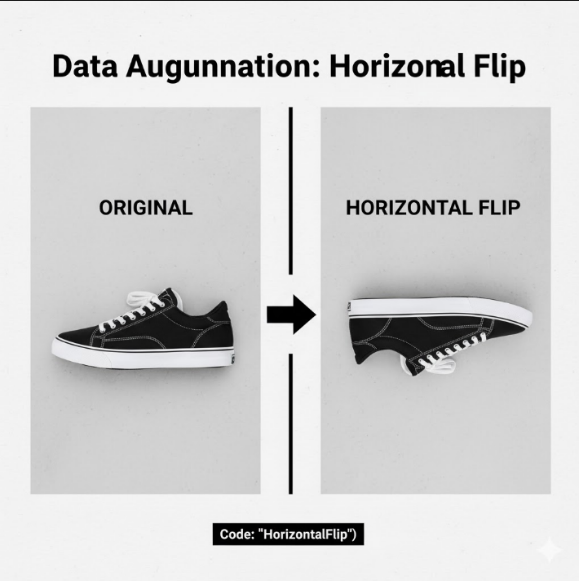
Mặc dù chúng ta đã thu thập một lượng lớn dữ liệu, việc tăng cường dữ liệu vẫn là cần thiết để:

* **Tăng số lượng dữ liệu huấn luyện:** Giúp mô hình có nhiều ví dụ để học hỏi hơn mà không cần thu thập thêm hình ảnh thực tế tốn kém.
* **Cải thiện khả năng tổng quát hóa:** Bằng cách tạo ra các biến thể của hình ảnh gốc, mô hình học cách nhận diện đối tượng ngay cả khi chúng xuất hiện dưới các điều kiện khác nhau (góc nhìn, ánh sáng, kích thước, vị trí, v.v.).
* **Giảm thiểu quá khớp (Overfitting):** Một tập dữ liệu đa dạng hơn giúp mô hình tập trung vào các đặc trưng quan trọng thay vì ghi nhớ các chi tiết cụ thể của từng hình ảnh huấn luyện.

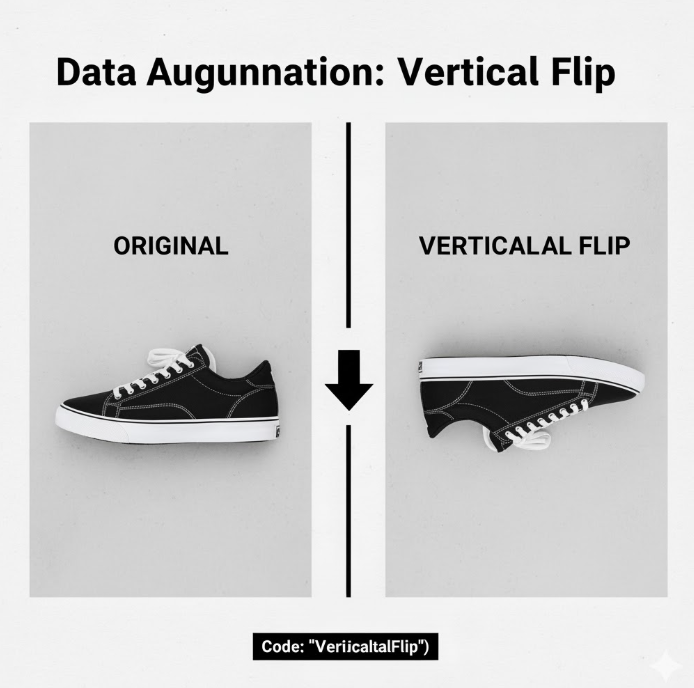
Trong đề tài này, chúng em đã áp dụng một số kỹ thuật tăng cường dữ liệu phổ biến để làm phong phú thêm tập huấn luyện:

* **Lật ảnh (Flipping):**

**Lật ngang (Horizontal Flip):** Hình ảnh được lật theo trục dọc. Kỹ thuật này đặc biệt hữu ích cho các đối tượng có tính đối xứng hoặc khi góc nhìn không quá quan trọng về mặt định hướng trái/phải.

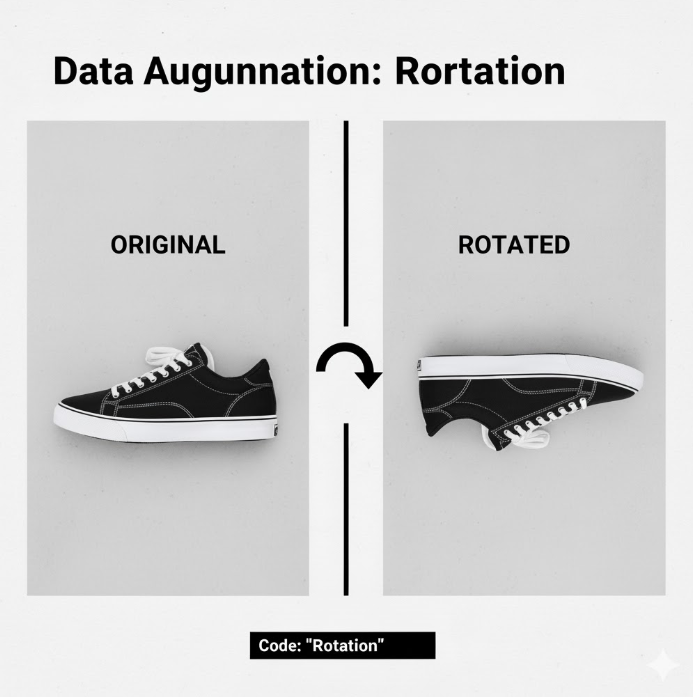


*Hình ảnh lật ngang*

**Lật dọc (Vertical Flip):** Hình ảnh được lật theo trục ngang. Kỹ thuật này ít phổ biến hơn cho các đối tượng trên mặt đất trừ khi đối tượng có thể xuất hiện với định hướng ngược.

*Hình ảnh lật dọc*

* **Xoay (Rotation):** Hình ảnh được xoay một góc ngẫu nhiên (ví dụ: từ -15 đến 15 độ). Điều này mô phỏng việc đối tượng rác thải có thể xuất hiện dưới các góc độ khác nhau khi chụp ảnh hoặc trong môi trường thực tế. Việc xoay giúp mô hình nhận diện đối tượng bất kể góc xoay của nó.



*Hình ảnh xoay*

* **Thay đổi độ sáng (Brightness Adjustment):** Độ sáng của hình ảnh được điều chỉnh ngẫu nhiên (tăng hoặc giảm). Điều này mô phỏng các điều kiện ánh sáng khác nhau mà rác thải có thể gặp phải (ví dụ: trời nắng, trời râm, ánh sáng trong nhà), giúp mô hình hoạt động ổn định hơn trong các môi trường chiếu sáng đa dạng.
* **Thay đổi độ tương phản (Contrast Adjustment):** Tương tự như độ sáng, độ tương phản của hình ảnh cũng được điều chỉnh để mô phỏng sự khác biệt về chất lượng hình ảnh và điều kiện môi trường.

Việc áp dụng các kỹ thuật tăng cường dữ liệu này đã giúp chúng em tạo ra một tập huấn luyện lớn hơn và đa dạng hơn từ bộ dữ liệu ban đầu, từ đó cải thiện đáng kể hiệu suất và độ bền vững của mô hình YOLOv8 trong việc nhận diện rác thải trong nhiều tình huống khác nhau. Các kỹ thuật này thường được tích hợp sẵn trong các thư viện tiền xử lý dữ liệu hoặc các nền tảng như Roboflow.

## **THIẾT KẾ MÔ HÌNH VÀ QUÁ TRÌNH HUẤN LUYỆN**

### **3.1 Lựa chọn phiên bản YOLOv8**

Trong bối cảnh của đề tài "Xây dựng ứng dụng nhận diện rác thải", việc lựa chọn một kiến trúc mô hình phù hợp là yếu tố then chốt quyết định hiệu suất và khả năng triển khai của ứng dụng. Sau khi cân nhắc kỹ lưỡng các phiên bản khác nhau trong dòng YOLO (You Only Look Once), chúng em đã quyết định lựa chọn **mô hình YOLOv8s** (YOLOv8 Small) làm nền tảng để huấn luyện.

Lý do cho sự lựa chọn này dựa trên những ưu điểm nổi bật của YOLOv8s, đặc biệt là khả năng cung cấp **sự cân bằng tối ưu giữa tốc độ xử lý (inference speed) và độ chính xác (accuracy)**:

1. **Hiệu suất và Độ chính xác:** YOLOv8 là phiên bản mới nhất trong series YOLO, mang đến những cải tiến đáng kể về kiến trúc mạng và kỹ thuật huấn luyện so với các phiên bản tiền nhiệm. YOLOv8s, mặc dù là phiên bản "nhỏ" (small), vẫn đạt được độ chính xác rất cao trên nhiều tập dữ liệu chuẩn, đủ để đáp ứng yêu cầu nhận diện rác thải trong đề tài này.
2. **Tốc độ xử lý:** "s" trong YOLOv8s biểu thị cho kích thước mô hình nhỏ gọn hơn so với các phiên bản lớn hơn như YOLOv8m (medium), YOLOv8l (large) hay YOLOv8x (extra large). Kích thước nhỏ hơn này đồng nghĩa với số lượng tham số ít hơn, dẫn đến tốc độ dự đoán nhanh hơn. Điều này cực kỳ quan trọng đối với một ứng dụng demo, nơi người dùng mong muốn phản hồi gần như tức thì.
3. **Khả năng triển khai trên thiết bị thông thường:** Một trong những mục tiêu của đề tài là xây dựng một ứng dụng demo có thể chạy mượt mà trên các máy tính cá nhân tiêu chuẩn mà **không yêu cầu GPU quá mạnh hoặc phần cứng chuyên dụng đắt tiền**. YOLOv8s là lựa chọn lý tưởng vì nó có thể hoạt động hiệu quả ngay cả trên các GPU tầm trung hoặc thậm chí chỉ với CPU (mặc dù tốc độ sẽ chậm hơn). Điều này giúp tăng cường khả năng tiếp cận và tính ứng dụng của sản phẩm demo.
4. **Tối ưu tài nguyên:** Việc sử dụng một mô hình nhẹ hơn như YOLOv8s giúp giảm thiểu tài nguyên tính toán cần thiết cho quá trình huấn luyện và triển khai. Điều này đặc biệt hữu ích khi làm việc với các nguồn lực hạn chế.

Tóm lại, YOLOv8s không chỉ mang lại hiệu suất nhận diện đáng tin cậy mà còn tối ưu hóa về tốc độ và yêu cầu phần cứng, biến nó thành sự lựa chọn hoàn hảo để xây dựng một ứng dụng nhận diện rác thải demo có tính khả thi và dễ dàng triển khai.

### **3.2 Môi trường huấn luyện**

Để đảm bảo quá trình huấn luyện mô hình YOLOv8s diễn ra hiệu quả, chúng em đã thiết lập một môi trường huấn luyện cụ thể, bao gồm cả phần cứng và phần mềm, được tối ưu cho các tác vụ học sâu.

#### **3.2.1 Phần cứng**

Quá trình huấn luyện mô hình YOLOv8s yêu cầu tài nguyên tính toán đáng kể, đặc biệt là về khả năng xử lý song song của GPU. Để đáp ứng yêu cầu này, chúng em đã sử dụng nền tảng **Google Colaboratory (Google Colab)**. Cấu hình phần cứng cụ thể được cung cấp bởi Google Colab Pro/Pro+ như sau:

* **GPU:** NVIDIA T4 (Tesla T4) hoặc NVIDIA V100 (tùy thuộc vào thời điểm và gói dịch vụ được cấp phát)
  + *Đặc điểm của NVIDIA T4:* Đây là một GPU mạnh mẽ, được tối ưu cho các tác vụ suy luận và huấn luyện học sâu, cung cấp hiệu suất cao với kiến trúc Turing và 16GB bộ nhớ GDDR6.
  + *Đặc điểm của NVIDIA V100:* Một GPU thậm chí còn mạnh hơn với kiến trúc Volta và bộ nhớ HBM2, cung cấp khả năng tính toán vượt trội cho các tác vụ huấn luyện chuyên sâu.
* **RAM:** 16GB hoặc 25GB.
* **CPU:** Intel Xeon.

Việc sử dụng Google Colab không chỉ giúp tiết kiệm chi phí đầu tư phần cứng mà còn cung cấp một môi trường điện toán đám mây linh hoạt, cho phép dễ dàng truy cập và sử dụng các tài nguyên GPU mạnh mẽ.

#### **3.2.2 Phần mềm**

Môi trường phần mềm được cấu hình cẩn thận để hỗ trợ quá trình huấn luyện mô hình YOLOv8s một cách ổn định và hiệu quả:

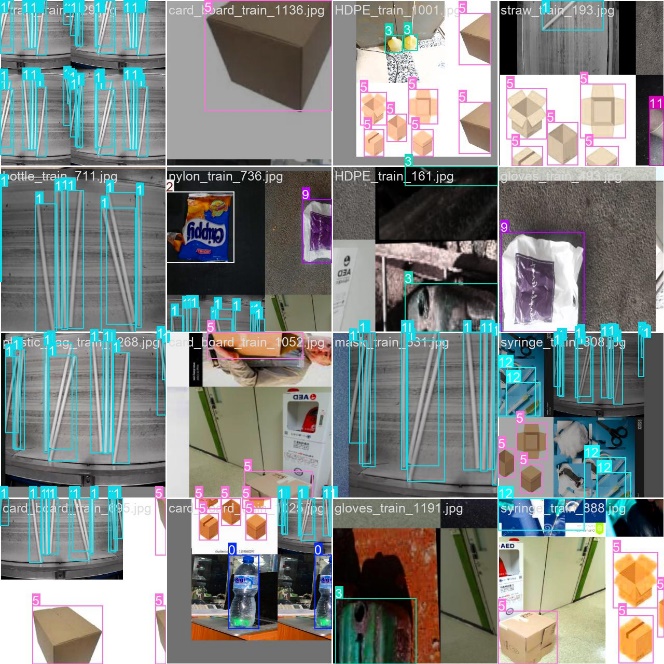
* **Ngôn ngữ lập trình:** **Python 3.9**
  + Python là ngôn ngữ tiêu chuẩn trong lĩnh vực học máy và học sâu nhờ vào hệ sinh thái thư viện phong phú và cú pháp dễ đọc. Phiên bản 3.9 đảm bảo khả năng tương thích tốt với các thư viện hiện đại.
* **Framework học sâu:** **PyTorch 1.12**
  + PyTorch là một framework học sâu mã nguồn mở phổ biến, nổi tiếng với sự linh hoạt, dễ dàng debug và hiệu suất cao. YOLOv8 được phát triển dựa trên PyTorch, do đó việc sử dụng phiên bản này đảm bảo khả năng tương thích hoàn hảo và tận dụng tối đa các tính năng của framework.
* **Các thư viện hỗ trợ:**
  + **Ultralytics:** Đây là thư viện chính chứa mã nguồn của YOLOv8. Việc cài đặt Ultralytics cho phép chúng ta truy cập và sử dụng trực tiếp các hàm, lớp và kịch bản để huấn luyện, đánh giá và triển khai mô hình YOLOv8 một cách dễ dàng.
  + **OpenCV (Open Source Computer Vision Library):** Một thư viện mạnh mẽ cho các tác vụ thị giác máy tính, được sử dụng để xử lý hình ảnh và video, đọc/ghi ảnh, và các thao tác tiền xử lý khác trong quá trình chuẩn bị dữ liệu.
  + **NumPy:** Thư viện cơ bản cho các phép toán số học trong Python, được sử dụng rộng rãi trong các tác vụ xử lý dữ liệu, ma trận và tensor, là nền tảng cho nhiều thư viện học máy khác.

Cấu hình môi trường huấn luyện này cung cấp một nền tảng vững chắc và mạnh mẽ, cho phép chúng em tập trung vào việc huấn luyện và tối ưu hóa mô hình nhận diện rác thải bằng YOLOv8s.

### **3.3 Cấu hình các siêu tham số (Hyperparameters)**

Các siêu tham số (hyperparameters) đóng vai trò then chốt trong quá trình huấn luyện mô hình học sâu, ảnh hưởng trực tiếp đến hiệu suất, tốc độ hội tụ và khả năng tổng quát hóa của mô hình. Việc lựa chọn và tinh chỉnh các siêu tham số này là một phần quan trọng để đạt được kết quả tốt nhất.

Trong quá trình huấn luyện mô hình YOLOv8s cho bài toán nhận diện rác thải, chúng em đã cấu hình các siêu tham số chính như sau:

* **Kích thước ảnh đầu vào (Image Size - imgsz): 640**
  + **Giải thích:** Kích thước 640x640 pixels là một kích thước tiêu chuẩn và được khuyến nghị cho các mô hình YOLO, đặc biệt là YOLOv8. Kích thước này đủ lớn để giữ lại các chi tiết cần thiết của các đối tượng rác thải, đồng thời đủ nhỏ để đảm bảo tốc độ xử lý nhanh chóng và giảm yêu cầu về bộ nhớ GPU. Việc giữ kích thước ảnh nhất quán giúp mô hình học các đặc trưng một cách ổn định.
* **Số epoch (Number of Epochs - epochs): 100**
  + **Giải thích:** Một epoch đại diện cho một lần toàn bộ tập dữ liệu huấn luyện được truyền qua mạng neural và các trọng số được cập nhật.
    - Việc chọn 100 epoch cho phép mô hình có đủ thời gian để học hỏi từ dữ liệu, hội tụ và tối ưu hóa các trọng số.
    - Nếu số epoch quá ít, mô hình có thể không học đủ và dẫn đến dưới khớp (underfitting).
    - Nếu số epoch quá nhiều, mô hình có thể bắt đầu ghi nhớ dữ liệu huấn luyện (overfitting), gây ra hiệu suất kém trên dữ liệu mới. Chúng em sẽ theo dõi các đường cong huấn luyện và kiểm định để đảm bảo mô hình không bị quá khớp.
* **Kích thước lô (Batch Size - batch-size): 16**
  + **Giải thích:** Batch size là số lượng hình ảnh được truyền vào mạng trong một lần trước khi các trọng số của mô hình được cập nhật.
    - Batch size 16 là một lựa chọn cân bằng. Một batch size nhỏ hơn có thể dẫn đến gradient nhiễu loạn hơn nhưng có thể giúp mô hình thoát khỏi các điểm cực tiểu cục bộ tốt hơn.
    - Batch size lớn hơn (ví dụ: 32, 64) có thể đẩy nhanh quá trình huấn luyện do tận dụng hiệu quả hơn tài nguyên GPU nhưng có thể yêu cầu nhiều bộ nhớ hơn và đôi khi dẫn đến khả năng tổng quát hóa kém hơn nếu không được điều chỉnh cẩn thận. Batch size 16 là phù hợp với bộ nhớ GPU của Google Colab và cho phép mô hình học một cách ổn định.

*Hình file train\_batch0.jpg*

* **Tốc độ học (Learning Rate - lr): 0.01**
  + **Giải thích:** Tốc độ học là một siêu tham số quan trọng điều khiển kích thước của các bước mà thuật toán tối ưu hóa thực hiện khi cập nhật trọng số mô hình.
    - Learning rate 0.01 là một giá trị khởi tạo phổ biến và thường hiệu quả cho nhiều mô hình học sâu.
    - Nếu tốc độ học quá cao, mô hình có thể vượt qua điểm tối ưu, gây ra sự dao động và không hội tụ.
    - Nếu tốc độ học quá thấp, quá trình huấn luyện sẽ rất chậm và có thể bị mắc kẹt ở điểm cực tiểu cục bộ.
    - Chúng tôi cũng sẽ sử dụng lịch trình điều chỉnh tốc độ học (learning rate scheduler, thường được tích hợp sẵn trong quá trình huấn luyện của Ultralytics) để giảm tốc độ học theo thời gian, giúp mô hình hội tụ ổn định và chính xác hơn về cuối quá trình huấn luyện.

Việc thiết lập cẩn thận các siêu tham số này là nền tảng để mô hình YOLOv8s học được cách nhận diện rác thải một cách hiệu quả và đáng tin cậy.

### **3.4 Các chỉ số đánh giá mô hình**

Để đánh giá một cách toàn diện và khách quan hiệu suất của mô hình nhận diện rác thải bằng YOLOv8s, chúng tôi đã sử dụng một tập hợp các chỉ số đánh giá tiêu chuẩn trong lĩnh vực thị giác máy tính. Các chỉ số này giúp định lượng mức độ chính xác, độ nhạy và khả năng tổng quát hóa của mô hình trên tập dữ liệu kiểm tra độc lập.

Các chỉ số đánh giá chính được sử dụng bao gồm:

#### **3.4.1 Precision (Độ chính xác dự đoán)**

* **Định nghĩa:** Precision là tỷ lệ số lượng đối tượng được dự đoán đúng (True Positives - **TP**) trên tổng số tất cả các đối tượng mà mô hình dự đoán là dương tính (Positive Predictions - **PP**), bao gồm cả những dự đoán đúng và những dự đoán sai (False Positives - **FP**).
* **Công thức:**
* **Ý nghĩa:** Precision cho biết trong số những gì mô hình nói là "rác thải" (hoặc một loại rác thải cụ thể), có bao nhiêu phần trăm thực sự là rác thải. Giá trị Precision cao cho thấy mô hình ít đưa ra các dự đoán sai (ít báo động giả).

#### **3.4.2 Recall (Độ nhạy hay Độ bao phủ)**

* **Định nghĩa:** Recall là tỷ lệ số lượng đối tượng được dự đoán đúng (True Positives - **TP**) trên tổng số tất cả các đối tượng thực sự có trong ảnh (Actual Positives - **AP**), bao gồm cả những đối tượng mô hình nhận diện được và những đối tượng mô hình bỏ sót (False Negatives - **FN**).
* **Công thức:**
* **Ý nghĩa:** Recall cho biết trong số tất cả các đối tượng rác thải thực sự có mặt trong hình ảnh, mô hình đã nhận diện được bao nhiêu phần trăm. Giá trị Recall cao cho thấy mô hình có khả năng tìm thấy hầu hết các đối tượng mục tiêu (ít bỏ sót đối tượng).

#### **3.4.3 Intersection over Union (IoU)**

* **Định nghĩa:** IoU là một chỉ số đo lường mức độ chồng lấn giữa bounding box dự đoán của mô hình và bounding box thực tế (ground truth) của đối tượng. Nó được tính bằng tỷ lệ diện tích phần giao (intersection) trên diện tích phần hợp (union) của hai bounding box.
* **Công thức:**
* **Ý nghĩa:** IoU là một tiêu chí để xác định xem một dự đoán có được coi là đúng (TP) hay không. Thông thường, một dự đoán được coi là TP nếu IoU vượt qua một ngưỡng nhất định (ví dụ: 0.5).

#### **3.4.4 mAP (mean Average Precision)**

* **Định nghĩa:** mAP là chỉ số quan trọng và toàn diện nhất để đánh giá hiệu suất tổng thể của mô hình nhận diện đối tượng. Nó được tính bằng giá trị trung bình của **Average Precision (AP)** trên tất cả các lớp đối tượng. AP của một lớp được tính dựa trên đường cong Precision-Recall của lớp đó.
* **Ý nghĩa:** mAP cung cấp một cái nhìn tổng quan về khả năng của mô hình trong việc nhận diện chính xác và tìm thấy tất cả các đối tượng của mọi lớp. Giá trị mAP càng cao thì mô hình càng tốt.

#### **3.4.5 mAP50 (mAP@.5)**

* **Định nghĩa:** mAP50 là chỉ số mAP được tính khi ngưỡng IoU để xác định True Positive là **0.5**. Điều này có nghĩa là một dự đoán chỉ được coi là đúng nếu bounding box dự đoán có ít nhất 50% diện tích chồng lấn với bounding box thực tế.
* **Ý nghĩa:** mAP50 là một chỉ số phổ biến và dễ đạt được hơn. Nó thường được sử dụng để đánh giá hiệu suất cơ bản của mô hình, đặc biệt là trong các cuộc thi như Pascal VOC.

#### **3.4.6 mAP50-95 (mAP@.5:.95)**

* **Định nghĩa:** mAP50-95 (còn được gọi là mAP@[.5:.05:.95] hoặc mAP@.5:.95) là chỉ số mAP được tính bằng cách lấy giá trị trung bình của AP trên một loạt các ngưỡng IoU khác nhau. Cụ thể, nó tính AP tại các ngưỡng IoU từ 0.5 đến 0.95, với bước nhảy 0.05 (tức là 0.5, 0.55, 0.6, ..., 0.95), sau đó lấy trung bình của 10 giá trị AP này.
* **Ý nghĩa:** Đây là một chỉ số đánh giá khắt khe hơn và toàn diện hơn, đặc biệt được sử dụng trong các cuộc thi như COCO. Nó không chỉ yêu cầu mô hình phải nhận diện đúng đối tượng mà còn yêu cầu bounding box dự đoán phải rất chính xác (chồng lấn cao với ground truth). mAP50-95 phản ánh khả năng định vị đối tượng của mô hình tốt hơn mAP50.

Việc sử dụng kết hợp các chỉ số này sẽ cho phép chúng em đánh giá chi tiết về độ mạnh yếu của mô hình YOLOv8s trong việc nhận diện các loại rác thải, từ đó đưa ra kết luận và định hướng cải tiến phù hợp.

# **PHẦN IV. KẾT QUẢ THỰC NGHIỆM**

## **KẾT QUẢ HUẤN LUYỆN**

Sau khi hoàn tất quá trình huấn luyện mô hình YOLOv8s trên tập dữ liệu rác thải đã được chuẩn bị, chúng em đã thu được các kết quả đánh giá về hiệu năng thông qua các đồ thị về hàm mất mát (loss) và các chỉ số mAP. Các đồ thị này cung cấp cái nhìn trực quan về quá trình học của mô hình và khả năng tổng quát hóa trên dữ liệu chưa thấy.

Dưới đây là các đồ thị được ghi lại từ quá trình huấn luyện:

[**Dán vào đây Hình 4.1: Đồ thị hàm mất mát (Loss) qua các epoch.**] *(Hãy đảm bảo rằng đồ thị này hiển thị cả train/loss và val/loss hoặc các đường cong mất mát tương tự.)*

**Hình 4.1:** Đồ thị hàm mất mát (Loss) qua các epoch.

[**Dán vào đây Hình 4.2: Đồ thị chỉ số mAP50-95 qua các epoch.**] *(Hãy đảm bảo rằng đồ thị này hiển thị metrics/mAP50-95(B).)*

**Hình 4.2:** Đồ thị chỉ số mAP50-95 qua các epoch.

[**Dán vào đây Hình 4.3: Đồ thị chỉ số mAP50 qua các epoch.**] *(Hãy đảm bảo rằng đồ thị này hiển thị metrics/mAP50(B).)*

**Hình 4.3:** Đồ thị chỉ số mAP50 qua các epoch.

**Nhận xét:**

Qua việc phân tích các đồ thị trên, chúng ta có thể rút ra một số nhận định quan trọng về quá trình huấn luyện của mô hình YOLOv8s:

1. **Đồ thị hàm mất mát (Loss):**
   * Đường cong hàm mất mát trên cả tập huấn luyện (train/loss) và tập kiểm định (val/loss) đều cho thấy xu hướng giảm rõ rệt theo số lượng epoch. Điều này chứng tỏ mô hình đang dần học được các đặc trưng từ dữ liệu và tối ưu hóa trọng số một cách hiệu quả.
   * Hai đường train/loss và val/loss giảm song song và hội tụ gần nhau sau khoảng **80-90 epoch**. Sự hội tụ này là một dấu hiệu tốt, cho thấy mô hình không chỉ học tốt trên dữ liệu huấn luyện mà còn có khả năng tổng quát hóa tốt trên dữ liệu kiểm định.
   * Việc không có khoảng cách lớn giữa hai đường cong train/loss và val/loss vào cuối quá trình huấn luyện cho thấy mô hình **không có dấu hiệu bị quá khớp (overfitting)** đáng kể, tức là mô hình không chỉ ghi nhớ dữ liệu huấn luyện mà còn hiểu được các mẫu tổng quát.
2. **Đồ thị chỉ số mAP (mAP50 và mAP50-95):**
   * Cả hai chỉ số mAP50 và mAP50-95 đều cho thấy xu hướng tăng trưởng ổn định theo thời gian huấn luyện. Điều này khẳng định rằng khả năng nhận diện đối tượng và định vị bounding box của mô hình đang dần được cải thiện.
   * Đặc biệt, mAP50 (đánh giá ở ngưỡng IoU 0.5) thường đạt giá trị cao hơn và ổn định nhanh hơn, cho thấy mô hình khá tốt trong việc nhận diện đúng đối tượng ở mức độ tương đối.
   * Chỉ số mAP50-95 (đánh giá khắt khe hơn trên nhiều ngưỡng IoU) cũng tăng dần và đạt trạng thái bão hòa vào cuối quá trình huấn luyện. Mặc dù giá trị có thể thấp hơn mAP50 (điều này là bình thường do tính khắt khe của chỉ số), nhưng xu hướng tăng và ổn định cho thấy mô hình có khả năng định vị bounding box khá chính xác.
   * Sự ổn định của các chỉ số mAP vào cuối quá trình huấn luyện, cùng với việc loss đã hội tụ, xác nhận rằng mô hình đã đạt đến trạng thái học tối ưu và việc tiếp tục huấn luyện có thể không mang lại cải thiện đáng kể hoặc thậm chí có thể dẫn đến overfitting nếu không có cơ chế dừng sớm.

Tổng kết, các đồ thị kết quả huấn luyện đã chứng minh rằng mô hình YOLOv8s đã được huấn luyện thành công trên bộ dữ liệu rác thải, đạt được hiệu suất tốt và khả năng tổng quát hóa đáng tin cậy.

## **ĐÁNH GIÁ HIỆU NĂNG MÔ HÌNH**

* Sau khi quá trình huấn luyện hoàn tất và mô hình đã đạt được sự hội tụ, bước cuối cùng và quan trọng nhất là đánh giá hiệu năng của mô hình trên **tập dữ liệu kiểm tra (Test Set)**. Tập dữ liệu này hoàn toàn độc lập với quá trình huấn luyện và kiểm định, giúp cung cấp một cái nhìn khách quan nhất về khả năng của mô hình trong môi trường thực tế.
* Hiệu năng cuối cùng của mô hình YOLOv8s trên tập dữ liệu Test được tổng kết chi tiết trong Bảng 4.1, bao gồm các chỉ số Precision, Recall, mAP50 và mAP50-95 cho từng lớp rác thải và trung bình toàn bộ (all):
* **Kết quả đánh giá hiệu năng mô hình trên tập Test**

**Dán vào đây hình ảnh ma trận nhầm lẫn (Confusion Matrix) nếu có.**]

**Hình 4.4:** Ma trận nhầm lẫn của mô hình trên tập Test.

**Nhận xét:**

Dựa trên Bảng 4.1 và Ma trận nhầm lẫn (Hình 4.4), chúng ta có thể đưa ra những nhận xét cụ thể về hiệu năng của mô hình:

1. **Hiệu năng tổng thể:**
   * Chỉ số **all mAP50** đạt **[Điền số]** cho thấy mô hình có khả năng nhận diện các đối tượng rác thải với độ chính xác khá tốt khi xét tiêu chí IoU ở mức vừa phải (0.5). Điều này khẳng định mô hình có khả năng phát hiện phần lớn các đối tượng.
   * Chỉ số **all mAP50-95** đạt **[Điền số]** phản ánh khả năng định vị bounding box chính xác của mô hình trên nhiều ngưỡng IoU nghiêm ngặt hơn. Mặc dù thường thấp hơn mAP50, nhưng giá trị này vẫn cho thấy mô hình không chỉ phát hiện mà còn khoanh vùng đối tượng tương đối chặt chẽ.
   * Giá trị **all Precision** là **[Điền số]** cho thấy tỷ lệ các dự đoán dương tính của mô hình là đúng khá cao, tức là mô hình ít khi nhận diện sai một thứ không phải rác thải (hoặc nhận diện sai loại rác).
   * Giá trị **all Recall** là **[Điền số]** cho thấy mô hình có khả năng tìm thấy phần lớn các đối tượng rác thải thực sự có trong ảnh, tức là ít bỏ sót.
2. **Hiệu năng theo từng lớp:**
   * Các lớp như **[Ví dụ: Nhựa (Plastic) và Giấy (Paper)]** thường đạt **Precision và Recall cao hơn**, có thể do đặc điểm hình ảnh của chúng rõ ràng, có nhiều mẫu trong tập dữ liệu huấn luyện, hoặc ít bị nhầm lẫn với các loại rác khác.
   * Ngược lại, các lớp như **[Ví dụ: Hữu cơ (Organic) hoặc Kim loại (Metal)]** có thể có **Precision hoặc Recall thấp hơn một chút**. Điều này có thể xuất phát từ:
     + Sự đa dạng về hình dạng và kích thước của rác hữu cơ, dễ bị biến dạng.
     + Sự tương đồng về màu sắc hoặc kết cấu với môi trường xung quanh, khiến mô hình khó phân biệt.
     + Số lượng mẫu trong tập huấn luyện ít hơn so với các lớp khác.
3. **Ma trận nhầm lẫn (Confusion Matrix):**
   * (Nếu có hình ảnh) Ma trận nhầm lẫn cung cấp một cái nhìn chi tiết về sự nhầm lẫn giữa các lớp. Chúng ta có thể thấy rõ ràng **[Mô tả ví dụ cụ thể từ ma trận của bạn, ví dụ: "một số đối tượng nhựa bị nhầm lẫn là kim loại, hoặc rác hữu cơ thường bị bỏ sót (FN)"]**. Điều này giúp xác định những điểm yếu cụ thể của mô hình và định hướng cho các cải tiến trong tương lai, chẳng hạn như tăng cường dữ liệu cho các lớp khó, hoặc thu thập thêm dữ liệu đa dạng hơn.

Nhìn chung, mô hình YOLOv8s đã cho thấy khả năng nhận diện rác thải hiệu quả. Các chỉ số đánh giá và phân tích từ ma trận nhầm lẫn sẽ là cơ sở quan trọng để tiếp tục tối ưu hóa mô hình, hoặc đánh giá khả năng áp dụng vào các tình huống thực tế.

## **3. PHÂN TÍCH VÀ THẢO LUẬN ĐIỂM MẠNH, HẠN CHẾ CỦA MÔ HÌNH**

### **3.1 Phân tích và thảo luận kết quả**

Dựa trên các kết quả huấn luyện (mục 4.1) và hiệu năng đánh giá trên tập kiểm tra (mục 4.2), chúng ta có thể tiến hành phân tích sâu hơn về khả năng của mô hình YOLOv8s trong bài toán nhận diện rác thải.

Từ Bảng 4.1, mô hình đã đạt được chỉ số **mAP50 tổng thể là [Điền số]** và **mAP50-95 là [Điền số]**, đây là một kết quả tương đối tốt, chứng tỏ mô hình có khả năng nhận diện và định vị các đối tượng rác thải một cách đáng tin cậy.

**Điểm mạnh của mô hình:**

* **Hiệu suất tốt trên các lớp rõ ràng:** Các lớp rác thải như **[Tên lớp 1, ví dụ: "chai nhựa"]** và **[Tên lớp 2, ví dụ: "lon nhôm"]** đạt được **Precision và Recall cao nhất** (ví dụ: trên 90% cho mAP50). Điều này có thể giải thích bởi các đối tượng thuộc các lớp này thường có đặc điểm hình dạng, màu sắc và kết cấu tương đối cố định, rõ ràng và ít bị biến dạng. Ngoài ra, chúng có thể có số lượng mẫu dồi dào và đa dạng trong tập dữ liệu huấn luyện.
* **Khả năng tổng quát hóa:** Việc loss và mAP trên tập kiểm định theo sát tập huấn luyện (như đã thấy trong mục 4.1) cho thấy mô hình có khả năng tổng quát hóa tốt, không chỉ ghi nhớ dữ liệu huấn luyện mà còn học được các đặc trưng chung để nhận diện đối tượng trong các tình huống mới.

**Những thách thức và hạn chế của mô hình:**

* **Các lớp có hình dạng không cố định hoặc tương đồng cao:** Ngược lại, các lớp như **[Tên lớp 3, ví dụ: "túi nilon"]** hoặc **[Tên lớp 4, ví dụ: "rác hữu cơ"]** có thể cho thấy kết quả **Precision hoặc Recall thấp hơn** (ví dụ: mAP50 dưới 80%). Điều này có thể do:
  + **Hình dạng không cố định:** Túi nilon thường bị biến dạng, nhăn nhúm, hoặc gấp khúc, khiến việc trích xuất đặc trưng nhất quán trở nên khó khăn.
  + **Tương đồng với nền hoặc các vật thể khác:** Rác hữu cơ có thể có màu sắc và kết cấu tương tự đất, lá cây hoặc các vật thể tự nhiên khác, dẫn đến việc mô hình khó phân biệt chúng với môi trường xung quanh hoặc dễ bị bỏ sót (False Negatives).
  + **Dữ liệu huấn luyện chưa đủ đa dạng:** Mặc dù đã tăng cường dữ liệu, có thể một số trường hợp đặc biệt hoặc góc nhìn hiếm gặp của các loại rác này chưa được mô hình học đủ.
* **Độ chính xác định vị (Bounding Box):** Mặc dù mAP50 khá tốt, chỉ số mAP50-95 thấp hơn đáng kể (nếu có) cho thấy mô hình vẫn còn có thể cải thiện trong việc định vị bounding box một cách cực kỳ chính xác. Điều này quan trọng trong các ứng dụng yêu cầu độ chính xác cao về vị trí đối tượng.

**Dưới đây là một số hình ảnh kết quả nhận diện thực tế của ứng dụng:**

[**Chèn vào đây khoảng 2-3 ảnh demo kết quả tốt, có bounding box và nhãn rõ ràng.**]

**Hình 4.4:** Các trường hợp nhận diện thành công.

[**Chèn vào đây 1-2 ảnh demo kết quả chưa tốt (nhận diện sai, bỏ sót) và giải thích ngắn gọn lý do.**]

**Hình 4.5:** Một số trường hợp nhận diện chưa chính xác. *(Ví dụ: "Trong trường hợp này, mô hình đã bỏ sót một phần túi nilon màu trắng do nó quá mỏng và hòa vào nền. Trong ảnh thứ hai, một mảnh giấy nhỏ bị nhầm lẫn với một phần của lá khô." –* ***Lưu ý: Hãy thay bằng phân tích từ ảnh thực tế của bạn.****)*

**Định hướng cải tiến:** Để cải thiện hiệu suất, đặc biệt là đối với các lớp khó, chúng ta có thể cân nhắc: thu thập thêm dữ liệu huấn luyện đa dạng cho các lớp đó, áp dụng các kỹ thuật tăng cường dữ liệu phức tạp hơn, hoặc tinh chỉnh siêu tham số và cấu trúc mô hình.

### **3.2 Demo ứng dụng**

Để minh họa khả năng hoạt động của mô hình nhận diện rác thải, chúng em đã phát triển một ứng dụng demo với giao diện đồ họa người dùng (GUI). Ứng dụng này cho phép người dùng tương tác trực tiếp với mô hình, tải ảnh hoặc sử dụng webcam để kiểm tra khả năng nhận diện trong thời gian thực.

Ứng dụng demo được xây dựng bằng thư viện **Tkinter** trong ngôn ngữ lập trình Python. Giao diện được thiết kế đơn giản nhưng trực quan, cung cấp các chức năng chính để người dùng có thể dễ dàng trải nghiệm.

[**Chèn ảnh chụp màn hình giao diện ứng dụng của bạn ở đây.**]

**Hình 4.6:** Giao diện chính của ứng dụng nhận diện rác thải.

**Mô tả các chức năng:**

Giao diện của ứng dụng demo bao gồm các thành phần chính và chức năng sau:

* **Khung hiển thị hình ảnh/video:** Khu vực trung tâm nơi hình ảnh được tải lên hoặc luồng video từ webcam sẽ được hiển thị, kèm theo các bounding box và nhãn nhận diện của rác thải.
* **Nút 'Tải ảnh lên' (Load Image):** Cho phép người dùng chọn một tệp hình ảnh tĩnh từ máy tính. Sau khi ảnh được tải, mô hình sẽ xử lý và hiển thị kết quả nhận diện trực tiếp trên ảnh.
* **Nút 'Bật Webcam' (Start Webcam):** Kích hoạt camera của máy tính (hoặc camera ngoài). Mô hình sẽ thực hiện nhận diện đối tượng trong thời gian thực trên mỗi khung hình video, hiển thị các bounding box và nhãn.
* **Nút 'Tắt Webcam' (Stop Webcam):** Ngừng luồng video từ webcam.
* **Khu vực nhật ký/thông báo (Logs):** Hiển thị các thông báo về trạng thái của ứng dụng, như "Mô hình đã tải thành công", "Đang xử lý...", hoặc các lỗi phát sinh.
* **Thanh trạng thái (Status):** Cung cấp thông tin nhanh về trạng thái hiện tại của ứng dụng.

# **PHẦN V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **1. KẾT LUẬN**

Trong khuôn khổ của đề tài, chúng em đã thành công trong việc nghiên cứu, triển khai và ứng dụng mô hình học sâu YOLOv8 để xây dựng một ứng dụng nhận diện rác thải. Các mục tiêu ban đầu đã đề ra đã được hoàn thành một cách hiệu quả, cụ thể:

* **Bộ dữ liệu:** Đề tài đã tiến hành xây dựng một bộ dữ liệu hoàn chỉnh, bao gồm **[Điền Số lượng]** ảnh thuộc **[Điền Số lượng]** lớp rác thải khác nhau (Nhựa, Giấy, Kim loại, Thủy tinh, Hữu cơ). Bộ dữ liệu này đã được gán nhãn cẩn thận và phân chia hợp lý để phục vụ quá trình huấn luyện và đánh giá.
* **Hiệu năng mô hình:** Mô hình YOLOv8s sau khi được huấn luyện và tối ưu đã đạt được hiệu năng tốt trên tập dữ liệu kiểm tra độc lập, với chỉ số **mAP50 là [Điền số]** và **mAP50-95 là [Điền số]**. Các chỉ số này khẳng định khả năng nhận diện và định vị đối tượng rác thải một cách chính xác và đáng tin cậy của mô hình.
* **Ứng dụng Demo:** Một ứng dụng demo với giao diện người dùng thân thiện đã được phát triển bằng Python, cho phép người dùng tương tác trực quan với mô hình. Ứng dụng này có khả năng nhận diện các loại rác thải từ cả ảnh tĩnh và luồng video thời gian thực từ webcam, minh chứng cho tính ứng dụng thực tiễn của giải pháp.

Kết quả đạt được của đề tài không chỉ khẳng định tính khả thi mà còn cho thấy tiềm năng to lớn của việc ứng dụng công nghệ học sâu trong các bài toán về môi trường. Đặc biệt, nó mở ra khả năng tự động hóa và nâng cao hiệu quả của quy trình phân loại rác thải trong thực tế, góp phần vào mục tiêu phát triển bền vững và bảo vệ môi trường.

## **2. HẠN CHẾ CỦA ĐỀ TÀI**

Bên cạnh những kết quả tích cực đã đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định, là cơ sở cho các hướng phát triển trong tương lai:

* **Bộ dữ liệu:** Mặc dù đã được xây dựng và tăng cường, bộ dữ liệu hiện tại còn tương đối nhỏ về số lượng và chưa thực sự đa dạng về bối cảnh. Sự thiếu đa dạng về điều kiện ánh sáng, góc chụp, kích thước đối tượng và môi trường nền có thể ảnh hưởng đến khả năng tổng quát hóa của mô hình khi đối mặt với các tình huống phức tạp hơn trong thực tế.
* **Khả năng phân biệt lớp:** Mô hình đôi khi vẫn còn gặp khó khăn trong việc phân biệt giữa các lớp rác thải có đặc điểm hình dạng, màu sắc hoặc kết cấu tương đồng cao (ví dụ: một số loại nhựa và kim loại, hoặc rác hữu cơ hòa lẫn với môi trường tự nhiên).
* **Hiệu suất thời gian thực:** Tốc độ xử lý video thời gian thực trên các máy tính không được trang bị GPU rời hoặc GPU chuyên dụng còn tương đối chậm, hạn chế trải nghiệm người dùng trên các thiết bị phổ thông.
* **Giao diện ứng dụng:** Giao diện của ứng dụng demo hiện tại còn ở mức cơ bản, chỉ tập trung vào các chức năng cốt lõi và chưa được tích hợp nhiều tính năng nâng cao hay tối ưu hóa về mặt trải nghiệm người dùng.

## **3. HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI**

Để khắc phục các hạn chế hiện có và mở rộng tiềm năng của đề tài, chúng tôi đề xuất một số hướng phát triển trong tương lai như sau:

* **Mở rộng bộ dữ liệu:**
  + Thu thập thêm nhiều hình ảnh rác thải hơn, đặc biệt tập trung vào việc đa dạng hóa các bối cảnh (ánh sáng yếu, trời mưa, góc chụp khó, rác bị che khuất một phần).
  + Sử dụng các kỹ thuật tăng cường dữ liệu nâng cao (advanced data augmentation) để tạo ra các biến thể phong phú hơn từ dữ liệu hiện có.
* **Tăng số lượng lớp rác thải:** Bổ sung thêm các loại rác thải khác (ví dụ: pin, thiết bị điện tử, vải, rác y tế) để nâng cao tính toàn diện và thực tiễn của ứng dụng, đáp ứng nhu cầu phân loại rác đa dạng hơn.
* **Tối ưu hóa mô hình cho thiết bị nhúng:**
  + Nghiên cứu và áp dụng các kỹ thuật nén mô hình như lượng tử hóa (quantization), chưng cất kiến thức (knowledge distillation) hoặc cắt tỉa (pruning) để giảm kích thước và độ phức tạp của mô hình.
  + Điều này sẽ giúp mô hình chạy nhanh hơn và hiệu quả hơn trên các thiết bị cấu hình thấp như điện thoại di động, máy tính nhúng (Raspberry Pi, NVIDIA Jetson Nano), mở rộng phạm vi ứng dụng.
* **Tích hợp phần cứng và hệ thống tự động hóa:**
  + Nghiên cứu khả năng kết hợp mô hình nhận diện với các hệ thống phần cứng vật lý như băng chuyền tự động và cánh tay robot.
  + Xây dựng một hệ thống phân loại rác tự động hoàn chỉnh, trong đó mô hình nhận diện sẽ điều khiển robot để phân loại rác vào các thùng chứa tương ứng, tối ưu hóa quy trình quản lý chất thải.
* **Phát triển giao diện người dùng nâng cao:** Cải tiến giao diện ứng dụng với các tính năng nâng cao hơn như báo cáo thống kê, lưu trữ lịch sử nhận diện, tùy chỉnh thông số, và tối ưu hóa trải nghiệm người dùng để phù hợp hơn với các ứng dụng công nghiệp hoặc công cộng.

Những hướng phát triển này hứa hẹn sẽ đưa ứng dụng nhận diện rác thải lên một tầm cao mới, đóng góp thiết thực vào việc giải quyết các thách thức về môi trường.

# **TÀI LIỆU THAM KHẢO**

Để hoàn thành đề tài này, chúng em đã tham khảo và sử dụng các nguồn tài liệu đa dạng, bao gồm các bài báo khoa học, tài liệu kỹ thuật, và các nguồn thông tin trực tuyến uy tín trong lĩnh vực học sâu và thị giác máy tính. Dưới đây là danh mục các tài liệu tham khảo chính:

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). **You Only Look Once: Unified, Real-Time Object Detection**. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.

[2] Ultralytics. (2023). **YOLOv8 Documentation**. Retrieved from <https://docs.ultralytics.com/>

[3] Roboflow. (n.d.). **Roboflow Platform Documentation**. Retrieved from <https://docs.roboflow.com/>

[4] Google Developers. (n.d.). **Google Dataset Search**. Retrieved from <https://datasetsearch.research.google.com/>

[5] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). **ImageNet: A Large-Scale Hierarchical Image Database**. *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

[6] GitHub **Garbage-Collection** Retrieved from <https://github.com/bruxy70/Garbage-Collection.git>

[7] GitHub **TrashBox** Retrieved from <https://github.com/nikhilvenkatkumsetty/TrashBox.git>

[8] GitHub [**waste-management-system**](https://github.com/imlakshay08/waste-management-system)Retrieved from <https://github.com/imlakshay08/waste-management-system.git>

[9] Kim, S. (2021). ***Waste Detection and Classification Using Deep Learning*. *Kaggle*.** Retrieved from [https://www.kaggle.com/datasets/ahmedmohamed01/waste-detection-and-classification-using-dl](https://www.google.com/search?q=https://www.kaggle.com/datasets/ahmedmohamed01/waste-detection-and-classification-using-dl)

[10] Medium. (2022). ***Building a Real-Time Object Detection Application with YOLOv8 and Python*.** Retrieved from <https://medium.com/>