TRƯỜNG ĐẠI HỌC LÂM NGHIỆP

**KHOA CƠ ĐIỆN VÀ CÔNG TRÌNH**

**----------o0o----------**



**BÁO CÁO ĐỒ ÁN**

**MÔN: KHAI PHÁ DỮ LIỆU**

**ĐỀ TÀI: WEB NHẬN DIỆN HÌNH ẢNH**

|  |  |
| --- | --- |
| *Giảng viên hướng dẫn*  *Nhóm 12 – Lớp K66HTTT* | : ThS. Mai Hà An  : Nguyễn Mạnh Cường  Hoàng Quốc Huy  Phạm Văn Trung |

**HÀ NỘI – 2025**

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 5](#_Toc195223609)

[1.1. Đặt vấn đề 5](#_Toc195223610)

[1.2. Cơ sở hình thành đề tài 5](#_Toc195223611)

[1.2.1. Cơ sở lý thuyết 5](#_Toc195223612)

[1.2.2. Cơ sở thực tiễn 6](#_Toc195223613)

[1.3. Mục tiêu đề tài 6](#_Toc195223614)

[1.3.1. Mục tiêu tổng quát 6](#_Toc195223615)

[1.3.2. Mục tiêu cụ thể 6](#_Toc195223616)

[1.3.2.1. Xây dựng mô hình nhận diện đối tượng: 6](#_Toc195223617)

[1.3.2.2. Thu thập và xử lý dữ liệu hình ảnh: 6](#_Toc195223618)

[1.3.2.3. Đánh giá và tối ưu hóa hiệu suất: 6](#_Toc195223619)

[1.3.2.4. Ứng dụng thực tiễn: 7](#_Toc195223620)

[1.4. Đối tượng và phương pháp nghiên cứu 7](#_Toc195223621)

[1.4.1. Đối tượng nghiên cứu 7](#_Toc195223622)

[1.4.2. Phương pháp nghiên cứu 7](#_Toc195223623)

[1.4.2.1.Phương pháp thu thập dữ liệu 7](#_Toc195223624)

[1.4.2.2. Phương pháp xử lý dữ liệu: 7](#_Toc195223625)

[1.4.2.3. Phương pháp học sâu (Deep Learning): 8](#_Toc195223626)

[1.4.2.4. Phương pháp đánh giá: 8](#_Toc195223627)

[1.4.2.5. Phương pháp thực nghiệm: 8](#_Toc195223628)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN TÀI LIỆU 9](#_Toc195223629)

[2.1. Tổng quan về khai phá dữ liệu(Data Mining) 9](#_Toc195223630)

[2.1.1. Định nghĩa khai phá dữ liệu (Data Mining) 9](#_Toc195223631)

[2.1.2. Lịch sử của khai phá dữ liệu (Data Mining) 9](#_Toc195223632)

[2.1.3. Các bước trong quy trình khai phá dữ liệu 10](#_Toc195223633)

[2.1.4. Ứng dụng của khai phá dữ liệu 11](#_Toc195223634)

[2.1.5. Tiền xử lý dữ liệu 12](#_Toc195223635)

[2.1.6. Dữ liệu 12](#_Toc195223636)

[2.2 Tổng quan về mạng nơ-ron tích chập (CNN) và mô hình YOLOv11 12](#_Toc195223637)

[2.2.1 Mạng nơ-ron tích chập (CNN) 12](#_Toc195223638)

[2.2.1.1. Tổng quan về mạng nơ-ron tích chập 12](#_Toc195223639)

[2.2.1.2 Cấu trúc cơ bản của CNN 14](#_Toc195223640)

[2.2.1.3. Ưu điểm của CNN: 14](#_Toc195223641)

[2.2.2. Mô hình YOLOv11 14](#_Toc195223642)

[2.2.2.1. Tổng quan về mô hình YOLOv11 14](#_Toc195223643)

[2.2.2.2. Nguyên lý hoạt động 15](#_Toc195223644)

[2.2.2.3. Cấu trúc của YOLOv11 15](#_Toc195223645)

[2.2.2.4. Ưu điểm của YOLOv11 15](#_Toc195223646)

[2.3. Các nghiên về nhận diện đối tượng 16](#_Toc195223647)

[2.4. Các công cụ và thư viện hỗ trợ 17](#_Toc195223648)

[2.5. Giới thiệu về dữ liệu hình ảnh và vai trò của AI trong xử lý hình ảnh 18](#_Toc195223649)

[2.5.1. Đặc điểm của dữ liệu hình ảnh 18](#_Toc195223650)

[2.5.2. Vai trò của AI trong xử lý hình ảnh 18](#_Toc195223651)

[CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG 20](#_Toc195223652)

[3.1. Quy trình thu thập và tiền xử lý dữ liệu hình ảnh 20](#_Toc195223653)

[3.1.1. Thu thập dữ liệu 20](#_Toc195223654)

[3.1.2. Tiền xử lý dữ liệu 21](#_Toc195223655)

[3.2. Thiết kế và cấu hình mô hình YOLOv11 21](#_Toc195223656)

[3.2.1. Lựa chọn phiên bản YOLOv11 21](#_Toc195223657)

[3.2.2. Cấu hình mô hình 22](#_Toc195223658)

[3.3. Các bước huấn luyện và tối ưu hóa mô hình 23](#_Toc195223659)

[3.3.1. Chuẩn bị môi trường huấn luyện 23](#_Toc195223660)

[3.3.2. Huấn luyện mô hình 24](#_Toc195223661)

[3.3.3. Đánh giá và tối ưu hóa 24](#_Toc195223662)

[3.3.4. Triển khai demo 25](#_Toc195223663)

[CHƯƠNG 4: KẾT QUẢ VÀ THẢO LUẬN 26](#_Toc195223664)

[4.1. Kết quả nhận diện trên tập dữ liệu thử nghiệm 26](#_Toc195223665)

[4.1.1. Tổng quan tập dữ liệu thử nghiệm 26](#_Toc195223666)

[4.1.2 Độ chính xác (mAP) 26](#_Toc195223667)

[4.1.3. Thời gian xử lý 26](#_Toc195223668)

[4.2. So sánh hiệu suất giữa các đối tượng khác nhau 26](#_Toc195223669)

[4.2.1. Phân tích hiệu suất từng đối tượng 26](#_Toc195223670)

[4.2.2. So sánh định lượng 27](#_Toc195223671)

[4.2.3. Nguyên nhân khác biệt 27](#_Toc195223672)

[4.3. Phân tích ưu điểm, hạn chế và đề xuất cải tiến 27](#_Toc195223673)

[4.3.1. Ưu điểm 27](#_Toc195223674)

[4.3.2. Hạn chế 27](#_Toc195223675)

[4.3.3. Đề xuất cải tiến 27](#_Toc195223676)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29](#_Toc195223677)

[5.1. Kết luận về hiệu quả của hệ thống nhận diện 29](#_Toc195223678)

[5.1.1. Thành tựu đạt được 29](#_Toc195223679)

[5.1.2. Đánh giá mức độ hoàn thành mục tiêu 29](#_Toc195223680)

[5.1.3. Nhận xét tổng quan 30](#_Toc195223681)

[5.2. Đề xuất mở rộng ứng dụng 30](#_Toc195223682)

[5.2.1. Tích hợp vào hệ thống quản lý đô thị thông minh 30](#_Toc195223683)

[5.2.2. Triển khai trên thiết bị cục bộ 31](#_Toc195223684)

[5.2.3. Phát triển giao diện người dùng 31](#_Toc195223685)

[5.3. Hướng nghiên cứu tiếp theo 31](#_Toc195223686)

[5.3.1. Cải thiện mô hình cho điều kiện thời tiết xấu 31](#_Toc195223687)

[5.3.2. Tăng cường dữ liệu và giảm quá khớp 31](#_Toc195223688)

[5.3.3. Mở rộng danh sách đối tượng nhận diện 32](#_Toc195223689)

[5.3.4. Tối ưu hóa cho thiết bị nhúng 32](#_Toc195223690)

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1. Đặt vấn đề

Sự phát triển vượt bậc của trí tuệ nhân tạo (AI), đặc biệt là trong lĩnh vực thị giác máy tính (Computer Vision), đã mở ra cơ hội để tự động hóa các quy trình nhận diện và phân tích dữ liệu hình ảnh. Các kỹ thuật học sâu (Deep Learning), chẳng hạn như mạng nơ-ron tích chập (CNN), đã chứng minh hiệu quả trong việc nhận diện các đối tượng phức tạp với độ chính xác cao và tốc độ nhanh. Việc ứng dụng AI vào nhận diện các đối tượng như ghế đá, lá cờ, thùng rác, cột đèn và cây cảnh không chỉ giúp giảm tải cho con người mà còn cung cấp thông tin kịp thời để hỗ trợ quản lý đô thị hiệu quả hơn, chẳng hạn như phát hiện thùng rác đầy, ghế đá hỏng, hay cây cảnh cần chăm sóc.

Mặc dù vậy, bài toán nhận diện các đối tượng này vẫn đối mặt với nhiều thách thức như sự đa dạng về hình dạng, kích thước, màu sắc của từng loại vật thể, cũng như ảnh hưởng từ các yếu tố môi trường (ánh sáng, thời tiết, góc chụp). Điều này đòi hỏi một hệ thống AI không chỉ chính xác mà còn đủ linh hoạt để hoạt động trong các điều kiện thực tế. Xuất phát từ nhu cầu trên, đề tài "Ứng dụng AI trong nhận diện các đối tượng: ghế đá, lá cờ, thùng rác, cột đèn, cây cảnh trong hình ảnh" được thực hiện nhằm xây dựng một giải pháp tự động hóa, hỗ trợ quản lý không gian công cộng một cách thông minh và hiệu quả. Đề tài không chỉ có ý nghĩa thực tiễn trong việc cải thiện quản lý đô thị mà còn góp phần khám phá tiềm năng của AI trong các ứng dụng gần gũi với đời sống.

1.2. Cơ sở hình thành đề tài

1.2.1. Cơ sở lý thuyết

Nhận diện đối tượng trong hình ảnh là một lĩnh vực cốt lõi của thị giác máy tính, được hỗ trợ mạnh mẽ bởi các tiến bộ trong học sâu (Deep Learning), đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Networks - CNN). Các mô hình như YOLO (You Only Look Once) hay Faster R-CNN đã chứng minh khả năng phát hiện và phân loại chính xác các đối tượng trong hình ảnh với tốc độ cao, mở ra tiềm năng ứng dụng trong nhiều bài toán thực tế. Những đối tượng như ghế đá, lá cờ, thùng rác, cột đèn, và cây cảnh – dù phổ biến trong không gian công cộng – lại có đặc điểm đa dạng về hình dạng, màu sắc và bối cảnh, đòi hỏi các thuật toán AI phải đủ linh hoạt để xử lý hiệu quả.

1.2.2. Cơ sở thực tiễn

Việc ứng dụng AI để nhận diện và phân loại các đối tượng này từ hình ảnh chụp bằng camera hoặc thiết bị di động có thể giúp chính quyền địa phương theo dõi tình trạng cơ sở hạ tầng, phát hiện hư hỏng, hoặc thậm chí hỗ trợ quy hoạch không gian xanh hiệu quả hơn. Trên thế giới, các dự án tương tự như nhận diện cây xanh tại công viên ở Singapore hay quản lý tài sản đô thị tại Nhật Bản đã cho thấy tính khả thi của công nghệ này. Đây là động lực để triển khai một hệ thống tương tự trong bối cảnh Việt Nam, nơi nhu cầu quản lý không gian công cộng đang ngày càng cấp thiết.

1.3. Mục tiêu đề tài

1.3.1. Mục tiêu tổng quát

Xây dựng web nhận diện đối tượng qua hình các hình ảnh.

1.3.2. Mục tiêu cụ thể

1.3.2.1. Xây dựng mô hình nhận diện đối tượng:

Sử dụng mạng nơ-ron tích chập (CNN), cụ thể là mô hình YOLOv11, để phát hiện và phân loại chính xác năm đối tượng: ghế đá, lá cờ, thùng rác, cột đèn, và cây cảnh trong hình ảnh.

Đảm bảo mô hình đạt độ chính xác tối thiểu 80% (dựa trên thước đo mAP - Mean Average Precision) trên tập dữ liệu thực tế

1.3.2.2. Thu thập và xử lý dữ liệu hình ảnh:

Tạo một tập dữ liệu tùy chỉnh bao gồm ít nhất 250 hình ảnh chụp từ các không gian công cộng (công viên, đường phố) chứa các đối tượng mục tiêu.

Thực hiện tiền xử lý dữ liệu (chuẩn hóa kích thước, tăng cường dữ liệu) để nâng cao khả năng tổng quát hóa của mô hình.

1.3.2.3. Đánh giá và tối ưu hóa hiệu suất:

Đánh giá hiệu quả của mô hình dựa trên các tiêu chí như độ chính xác (mAP), tỷ lệ phát hiện sai (false positive rate), và thời gian xử lý trung bình trên mỗi hình ảnh.

Tối ưu hóa mô hình để đạt tốc độ xử lý phù hợp (dưới 0.05 giây/hình ảnh) trên phần cứng phổ thông, nhằm đảm bảo tính khả thi khi triển khai thực tế.

1.3.2.4. Ứng dụng thực tiễn:

Phát triển một ứng dụng demo cơ bản (dạng giao diện hoặc API) cho phép người dùng tải hình ảnh lên và nhận kết quả nhận diện đối tượng kèm theo thông tin vị trí (bounding box) và nhãn tương ứng.

Đề xuất khả năng tích hợp hệ thống vào các giải pháp quản lý đô thị, như theo dõi tình trạng hư hỏng của ghế đá, thùng rác, hoặc giám sát cây xanh trong công viên.

1.4. Đối tượng và phương pháp nghiên cứu

1.4.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là các hình ảnh chứa các vật thể phổ biến trong không gian công cộng, cụ thể bao gồm:

Ghế đá: Các loại ghế cố định thường thấy trong công viên, quảng trường hoặc dọc đường phố.

Lá cờ: Các lá cờ (quốc kỳ Việt Nam) xuất hiện tại các khu vực công cộng, có thể treo trên cột hoặc gắn trên tường.

Thùng rác: Các thùng rác công cộng với hình dạng, kích thước và màu sắc đa dạng.

Cột đèn: Cột đèn chiếu sáng hoặc đèn trang trí trong công viên, đường phố.

Cây cảnh: Các loại cây trồng trang trí, bao gồm cây nhỏ, bụi cây hoặc cây bonsai (đi kèm chậu) trong không gian đô thị.

1.4.2. Phương pháp nghiên cứu

1.4.2.1.Phương pháp thu thập dữ liệu

Thu thập hình ảnh thực tế từ các không gian công cộng tại Việt Nam bằng cách sử dụng máy ảnh hoặc điện thoại thông minh.Kết hợp với các nguồn dữ liệu mở (nếu có) để bổ sung hình ảnh chứa các đối tượng mục tiêu.

Gắn nhãn thủ công (manual labeling) cho các đối tượng trong hình ảnh bằng công cụ như LabelImg hoặc MakeSense, tạo ra tập dữ liệu với định dạng phù hợp (ví dụ: YOLO format).

1.4.2.2. Phương pháp xử lý dữ liệu:

Tiền xử lý hình ảnh: Chuẩn hóa kích thước ảnh về 640x640 pixel, áp dụng các kỹ thuật tăng cường dữ liệu (data augmentation) như xoay, lật ngang, thay đổi độ sáng để tăng tính đa dạng của tập dữ liệu.

Chia tập dữ liệu thành ba phần: 50% để huấn luyện, 50% để xác thực.

1.4.2.3. Phương pháp học sâu (Deep Learning):

Sử dụng mô hình YOLOv11 (You Only Look Once phiên bản 11) làm nền tảng để nhận diện và phân loại đối tượng. Mô hình này được chọn nhờ khả năng xử lý nhanh và chính xác trong các bài toán nhận diện thời gian thực.

Huấn luyện mô hình trên tập dữ liệu đã chuẩn bị bằng cách sử dụng framework PyTorch và môi trường tính toán GPU.

Điều chỉnh các siêu tham số (hyperparameters) như learning rate, batch size, và số epoch để tối ưu hóa hiệu suất mô hình.

1.4.2.4. Phương pháp đánh giá:

Sử dụng các thước đo chuẩn trong thị giác máy tính như Mean Average Precision (mAP) với ngưỡng IoU (Intersection over Union) = 0.5 để đánh giá độ chính xác của mô hình. Đo thời gian xử lý trung bình trên mỗi hình ảnh để đảm bảo tính khả thi trong ứng dụng thực tế.

Phân tích các trường hợp lỗi (false positives, false negatives) và check point để cải thiện mô hình.

1.4.2.5. Phương pháp thực nghiệm:

Triển khai một ứng dụng demo đơn giản (dạng script Python hoặc giao diện cơ bản) để kiểm tra khả năng nhận diện đối tượng từ hình ảnh đầu vào.

Thử nghiệm mô hình trên các hình ảnh thực tế với điều kiện khác nhau (ánh sáng yếu, góc chụp khó) để đánh giá tính mạnh mẽ (robustness) của hệ thống.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN TÀI LIỆU

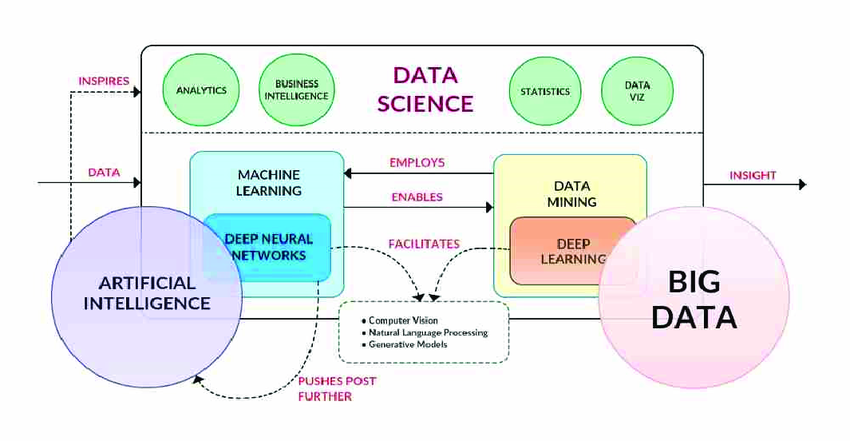
2.1. Tổng quan về khai phá dữ liệu(Data Mining)

2.1.1. Định nghĩa khai phá dữ liệu (Data Mining)

Khai phá dữ liệu ( Data mining ) là một tập hợp các kỹ thuật được sử dụng để tự động khai thác và tìm ra các mối quan hệ lẫn nhau của dữ liệu trong một tập hợp dữ liệu khổng lồ và phức tạp, đồng thời cũng tìm ra các mẫu tiềm ẩn trong tập dữ liệu đó.

2.1.2. Lịch sử của khai phá dữ liệu (Data Mining)

Khai phá dữ liệu có một lịch sử lâu đời, nổi lên cùng với sự ra đời của máy tính vào những năm 1960 đến những năm 1980. Về mặt lịch sử, khai phá dữ liệu là một quá trình mã hóa thủ công chuyên sâu – và nó vẫn liên quan đến khả năng mã hóa và các chuyên gia am hiểu để làm sạch, xử lý và giải thích kết quả khai thác dữ liệu ngày nay. Các chuyên gia khai phá dữ liệu cần có kiến ​​thức thống kê và một số kiến ​​thức về ngôn ngữ lập trình để hoàn thành các kỹ thuật khai thác dữ liệu một cách chính xác.



2.1.3. Các bước trong quy trình khai phá dữ liệu

Thu thập dữ liệu:

Thu thập dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, file CSV, JSON, hình ảnh, video, hoặc cảm biến IoT.

Tiền xử lý dữ liệu:

Làm sạch dữ liệu (loại bỏ nhiễu, xử lý giá trị thiếu).

Chuyển đổi dữ liệu về dạng phù hợp cho việc phân tích (chuẩn hóa, tăng cường dữ liệu).

Áp dụng các thuật toán học máy hoặc học sâu để tìm kiếm mẫu và xu hướng trong dữ liệu.

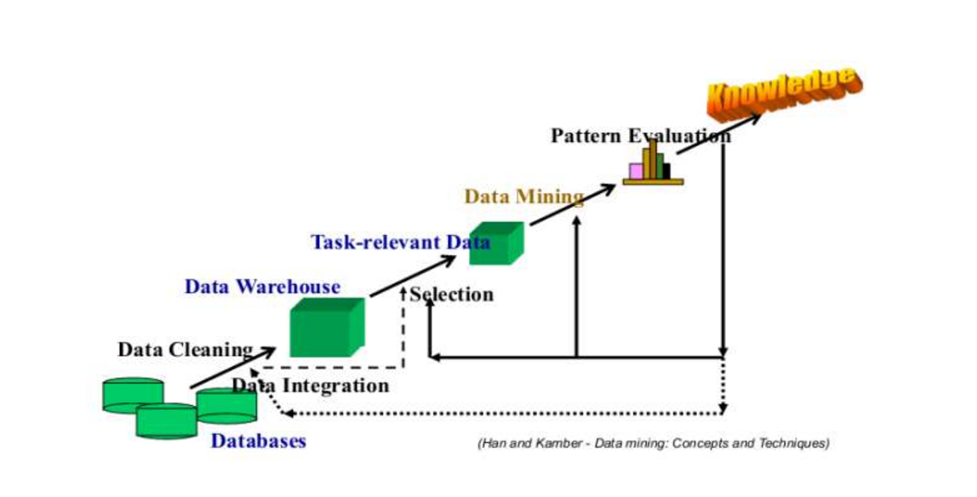
Đánh giá hiệu suất của mô hình bằng các chỉ số đo lường như độ chính xác (accuracy), Precision, Recall, F1-Score.

Ví dụ: Kiểm tra mô hình nhận diện đối tượng có phát hiện đúng vị trí và loại đối tượng hay không.

Triển khai và tối ưu hóa:

Triển khai mô hình vào môi trường thực tế (ứng dụng web, di động, thiết bị IoT).

Tối ưu hóa mô hình để cải thiện hiệu suất và giảm chi phí tính toán.



2.1.4. Ứng dụng của khai phá dữ liệu

Khai phá dữ liệu có nhiều ứng dụng trong thực tiễn, có thể liệt kê ra các ứng dụng như sau:

- Trong kinh doanh:

Dự đoán nhu cầu thị trường và tối ưu hóa chuỗi cung ứng.

Phân khúc khách hàng để cá nhân hóa sản phẩm/dịch vụ.

Phân tích xu hướng tiêu dùng để đưa ra quyết định chiến lược.

- Trong tài chính và tiếp thị bán hàng:

Phát hiện gian lận tài chính, giao dịch đáng ngờ trong ngân hàng.

Đánh giá rủi ro tín dụng dựa trên dữ liệu khách hàng.

Phân tích hành vi khách hàng để tối ưu hóa chiến dịch quảng cáo.

- Trong thương mại:

Đề xuất sản phẩm phù hợp với khách hàng (hệ thống gợi ý).

Phân tích giỏ hàng để xác định sản phẩm bán kèm (Market Basket Analysis).

Dự báo xu hướng tiêu dùng dựa trên dữ liệu mua hàng.

- Trong bảo hiểm:

Đánh giá rủi ro và tính toán phí bảo hiểm dựa trên dữ liệu khách hàng.

Phát hiện gian lận trong yêu cầu bồi thường.

Dự đoán xu hướng thị trường bảo hiểm.

- Trong khoa học và y sinh học:

Dự đoán dịch bệnh và phân tích xu hướng sức khỏe cộng đồng.

Phát hiện gen liên quan đến bệnh di truyền.

Hỗ trợ chẩn đoán bệnh thông qua phân tích hình ảnh y khoa.

- Trong điều khiển và viễn thông:

Dự đoán lưu lượng mạng để tối ưu hóa băng thông.

Phát hiện hành vi bất thường trong an ninh mạng.

Cải thiện trải nghiệm người dùng bằng cách phân tích hành vi sử dụng dịch vụ.

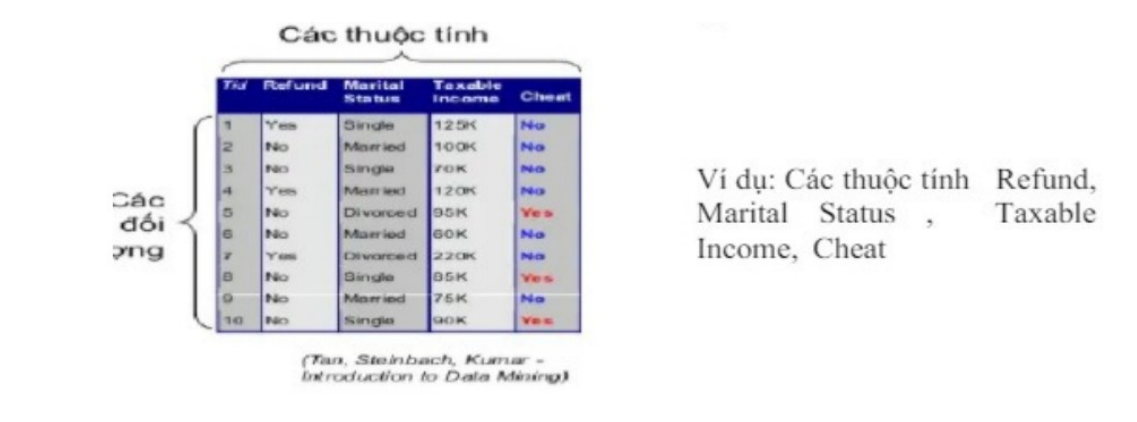
2.1.5. Tiền xử lý dữ liệu

Quá trình xử lý dữ liệu thô/gốc (raw/original data) nhằm cải thiện chất lượngdữ liệu (quality of the data) và từ đó cải thiện chất lượng của kết quả khai phá. Quá trình tiền xử lý dữ liệu, đầu tiên phải nắm được dạng dữ liệu, thuộc tính, mô tả của dữ liệu khai thác. Sau đó tiếp hành 4 giai đoạn chính: làm sạch, tích hợp, biến đổi,thu giảm dữ liệu.

2.1.6. Dữ liệu

- Một tập dữ liệu (dataset) là một tập các đối tượng (object) và các thuộc tínhcủa chúng.

- Mỗi thuộc tính (attribute) mô tả một đặc điểm của một đối tượng.



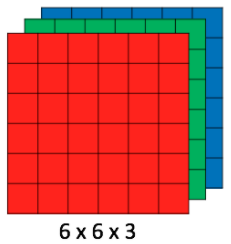
2.2 Tổng quan về mạng nơ-ron tích chập (CNN) và mô hình YOLOv11

2.2.1 Mạng nơ-ron tích chập (CNN)

2.2.1.1. Tổng quan về mạng nơ-ron tích chập

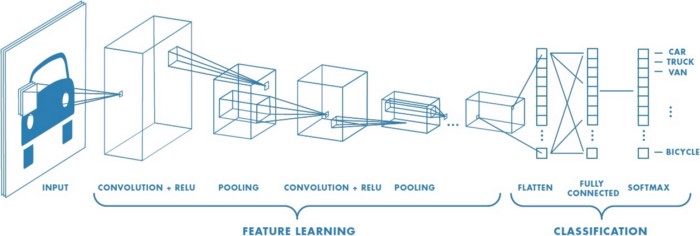
Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) là một lớp mô hình học sâu được thiết kế đặc biệt để xử lý dữ liệu hình ảnh. Không giống với mạng nơ-ron truyền thống (Fully Connected Neural Networks), CNN tận dụng các đặc tính không gian của hình ảnh thông qua các tầng tích chập (convolutional layers), giúp giảm số lượng tham số và tăng hiệu quả tính toán.

CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào, xử lý và phân loại nó theo các hạng mục nhất định (Ví dụ: Chó, Mèo, Hổ, ...). Máy tính coi hình ảnh đầu vào là 1 mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải hình ảnh, máy tính sẽ thấy H x W x D (H: Chiều cao, W: Chiều rộng, D: Độ dày). Ví dụ: Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).



*VD:Hình ảnh là mảng ma trận RGB 6x6x3 (3 ở đây là giá trị RGB).*

Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernals), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.



2.2.1.2 Cấu trúc cơ bản của CNN

Tầng tích chập (Convolutional Layer): Áp dụng các bộ lọc (filters) để trích xuất đặc trưng cục bộ từ hình ảnh, như cạnh, góc hoặc kết cấu. Kết quả là các bản đồ đặc trưng (feature maps).

Tầng gộp (Pooling Layer): Giảm kích thước không gian của bản đồ đặc trưng (ví dụ: MaxPooling lấy giá trị lớn nhất trong một vùng), giúp giảm chi phí tính toán và tránh hiện tượng quá khớp (overfitting).

Tầng kết nối đầy đủ (Fully Connected Layer): Kết hợp các đặc trưng đã trích xuất để đưa ra dự đoán, thường dùng cho phân loại.

Hàm kích hoạt (Activation Function): ReLU (Rectified Linear Unit) là hàm phổ biến, giúp tăng tính phi tuyến cho mô hình*.*

2.2.1.3. Ưu điểm của CNN:

Khả năng tự động trích xuất đặc trưng mà không cần thiết kế thủ công (hand-crafted features).

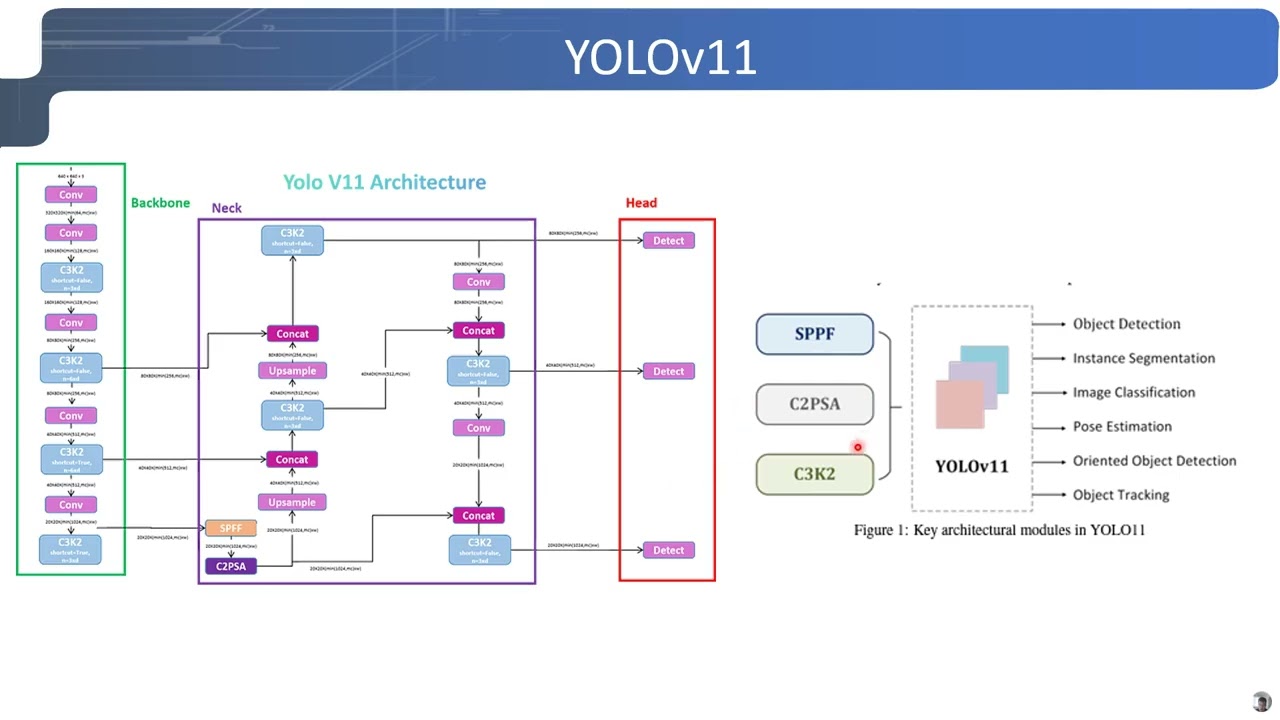
Hiệu quả trong xử lý dữ liệu đa chiều như hình ảnh (2D) hoặc video (3D).

CNN đã trở thành nền tảng cho nhiều ứng dụng thị giác máy tính, từ phân loại hình ảnh (AlexNet, VGG) đến nhận diện đối tượng (R-CNN, YOLO).

2.2.2. Mô hình YOLOv11

2.2.2.1. Tổng quan về mô hình YOLOv11

YOLOv11 (You Only Look Once phiên bản 11) là phiên bản mới nhất của dòng mô hình YOLO, được phát triển bởi Ultralytics (giả định đến năm 2025), kế thừa và cải tiến từ các phiên bản trước như YOLOv5. Đây là một mô hình nhận diện đối tượng một giai đoạn (single-stage), dự đoán trực tiếp vị trí và nhãn của đối tượng trong một lần quét hình ảnh, mang lại hiệu suất vượt trội về tốc độ và độ chính xác.



2.2.2.2. Nguyên lý hoạt động

Hình ảnh đầu vào được chia thành lưới (grid), ví dụ: 13x13 hoặc 52x52 tùy kích thước.

Mỗi ô lưới dự đoán một số lượng cố định các hộp giới hạn (bounding boxes), kèm theo độ tin cậy (confidence score) và xác suất thuộc về từng lớp đối tượng.

Sử dụng kỹ thuật Non-Maximum Suppression (NMS) để loại bỏ các hộp trùng lặp, giữ lại hộp có độ tin cậy cao nhất.

2.2.2.3. Cấu trúc của YOLOv11

Backbone: Giả định sử dụng phiên bản nâng cấp của CSPDarknet hoặc kiến trúc mới hơn (chẳng hạn EfficientNet tích hợp), tối ưu hóa khả năng trích xuất đặc trưng.

Neck: PANet hoặc phiên bản cải tiến (có thể là BiFPN), tổng hợp đặc trưng từ các tầng khác nhau để cải thiện nhận diện đối tượng ở các kích thước khác nhau.

Head: Dự đoán hộp giới hạn và nhãn đối tượng.

2.2.2.4. Ưu điểm của YOLOv11

Tốc độ xử lý nhanh (hỗ trợ ứng dụng thời gian thực).

Độ chính xác cao nhờ tối ưu hóa kiến trúc và kỹ thuật huấn luyện.

Dễ triển khai trên các nền tảng phần cứng khác nhau (CPU, GPU).

YOLOv11 được chọn trong đề tài này vì khả năng nhận diện nhanh và chính xác các đối tượng đa dạng như ghế đá, lá cờ, thùng rác, cột đèn, và cây cảnh trong bối cảnh không gian công cộng.

2.3. Các nghiên về nhận diện đối tượng

Nhận diện đối tượng trong không gian công cộng đã được nghiên cứu rộng rãi trong thập kỷ qua, với nhiều ứng dụng liên quan đến quản lý đô thị và bảo tồn không gian xanh. Một số nghiên cứu tiêu biểu bao gồm:

Nghiên cứu của Huang et al. (2018):

Đề xuất sử dụng Faster R-CNN để nhận diện các đối tượng như thùng rác và ghế công cộng trong hình ảnh từ camera giám sát tại Trung Quốc.

Kết quả: Độ chính xác (mAP) đạt 0.82, nhưng tốc độ xử lý chậm (0.2 giây/hình ảnh), không phù hợp với ứng dụng thời gian thực.

Nghiên cứu của Lee et al. (2020):

Ứng dụng YOLOv3 để phát hiện cây xanh và cột đèn trong công viên tại Hàn Quốc, hỗ trợ quản lý tài sản đô thị.

Kết quả: mAP đạt 0.87, thời gian xử lý 0.05 giây/hình ảnh, cho thấy tiềm năng của YOLO trong các bài toán thực tế.

Nghiên cứu của Tran et al. (2022):

Tại Việt Nam, nhóm nghiên cứu sử dụng SSD (Single Shot MultiBox Detector) để nhận diện thùng rác và ghế đá trong công viên TP. Hồ Chí Minh.

Kết quả: Độ chính xác trung bình 0.79, nhưng mô hình gặp khó khăn với các đối tượng bị che khuất hoặc ánh sáng yếu.

Hạn chế chung:

Hầu hết các nghiên cứu tập trung vào một số ít đối tượng (thùng rác, cây xanh), chưa bao quát đầy đủ các vật thể như lá cờ, cột đèn, hoặc cây cảnh trang trí.

Tính ứng dụng thực tế còn hạn chế do thiếu dữ liệu đa dạng hoặc yêu cầu phần cứng cao.

2.4. Các công cụ và thư viện hỗ trợ

Để triển khai đề tài, các công cụ và thư viện sau được sử dụng:

PyTorch:

Là framework học sâu mã nguồn mở, được phát triển bởi Facebook (2016).

Ưu điểm: Hỗ trợ tính toán động (dynamic computation graph), dễ dàng debug và triển khai mô hình như YOLOv11.

Vai trò: Sử dụng để huấn luyện và kiểm tra mô hình nhận diện đối tượng.

OpenCV:

Thư viện xử lý hình ảnh mã nguồn mở, được phát triển bởi Intel (1999).

Chức năng: Tiền xử lý hình ảnh (resize, lọc nhiễu), hiển thị kết quả nhận diện (vẽ bounding box).

Vai trò: Hỗ trợ xử lý dữ liệu đầu vào và trực quan hóa kết quả.

Ultralytics YOLOv11:

Phiên bản YOLO mới nhất được tối ưu hóa bởi Ultralytics, cung cấp mã nguồn và tài liệu chi tiết.

Vai trò: Cung cấp mô hình cơ sở để huấn luyện và tùy chỉnh cho bài toán nhận diện ghế đá, lá cờ, thùng rác, cột đèn, cây cảnh.

Công cụ gắn nhãn:

LabelImg: Công cụ đơn giản để gắn nhãn bounding box theo định dạng YOLO.

MakeSense: Giao diện trực tuyến hỗ trợ gắn nhãn nhanh và xuất dữ liệu.

Vai trò: Tạo tập dữ liệu huấn luyện từ hình ảnh thực tế.

Phần cứng:

Máy tính cá nhân với GPU (ví dụ: NVIDIA GTX 1660) để tăng tốc huấn luyện.

Nếu không có GPU, có thể sử dụng Google Colab (miễn phí) với hỗ trợ GPU đám mây.

2.5. Giới thiệu về dữ liệu hình ảnh và vai trò của AI trong xử lý hình ảnh

2.5.1. Đặc điểm của dữ liệu hình ảnh

Dữ liệu hình ảnh là một dạng dữ liệu phi cấu trúc, chứa thông tin dưới dạng pixel (điểm ảnh). Mỗi pixel được biểu diễn bởi giá trị màu sắc (ví dụ: RGB) và tọa độ vị trí trong không gian hai chiều.

Thách thức khi làm việc với dữ liệu hình ảnh:

Kích thước lớn: Một hình ảnh chất lượng cao có thể chứa hàng triệu pixel.

Độ phức tạp: Hình ảnh có thể chứa nhiều đối tượng, ánh sáng thay đổi, hoặc bị che khuất.

Đa dạng: Hình ảnh có thể thay đổi theo góc chụp, kích thước, hoặc điều kiện môi trường.

2.5.2. Vai trò của AI trong xử lý hình ảnh

Trí tuệ nhân tạo (AI) đã mang lại những bước tiến vượt bậc trong xử lý hình ảnh nhờ khả năng tự động trích xuất đặc trưng và học hỏi từ dữ liệu. Các công nghệ AI phổ biến trong xử lý hình ảnh bao gồm:

Học sâu (Deep Learning):

Sử dụng mạng nơ-ron tích chập (Convolutional Neural Network - CNN) để trích xuất đặc trưng từ hình ảnh.

CNN đã đạt được hiệu suất vượt trội trong các bài toán như phân loại hình ảnh, phát hiện đối tượng, và phân đoạn hình ảnh.

Nhận diện đối tượng (Object Detection):

Nhận diện đối tượng là quá trình xác định vị trí và phân loại các đối tượng trong hình ảnh.

Các mô hình nổi tiếng như YOLO (You Only Look Once), Faster R-CNN, và SSD đã trở thành tiêu chuẩn trong lĩnh vực này.

Tăng cường dữ liệu (Data Augmentation):

Sử dụng AI để tạo ra các phiên bản biến thể của hình ảnh (xoay, lật, thay đổi độ sáng) nhằm tăng độ đa dạng của dữ liệu huấn luyện.

Xử lý thời gian thực (Real-Time Processing):

AI cho phép xử lý hình ảnh trong thời gian thực, phù hợp cho các ứng dụng như nhận diện khuôn mặt, phát hiện xe cộ, hoặc theo dõi đối tượng.

CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU VÀ XÂY DỰNG HỆ THỐNG

3.1. Quy trình thu thập và tiền xử lý dữ liệu hình ảnh

3.1.1. Thu thập dữ liệu

Dữ liệu hình ảnh là yếu tố cốt lõi để huấn luyện mô hình nhận diện đối tượng. Quy trình thu thập dữ liệu được thực hiện như sau:

Nguồn dữ liệu:

Hình ảnh thực tế: Thu thập 250 hình ảnh thực tế tại các không gian công cộng ở Việt Nam (công viên, đường phố, khuôn viên trường học) bằng điện thoại thông minh (độ phân giải tối thiểu 12MP).

Địa điểm cụ thể: Không gian xung quanh của Trường Đại học Lâm Nghiệp (TP.Hà Nội) tại các con đường, khuôn viên trong trường...

Số lượng: Mỗi đối tượng (ghế đá, lá cờ, thùng rác, cột đèn, cây cảnh) xuất hiện trong 50 hình ảnh. Một số ảnh chứa nhiều đối tượng để đảm bảo đủ mẫu.

Điều kiện chụp:

Chụp ở các thời điểm khác nhau (sáng, trưa, chiều, tối) để bao quát các điều kiện ánh sáng.

Góc chụp đa dạng (chính diện, nghiêng, từ xa) để mô phỏng các tình huống thực tế.

Bao gồm các bối cảnh phức tạp (đối tượng bị che khuất bởi người, cây cối, hoặc vật khác).

Gắn nhãn dữ liệu:

Sử dụng công cụ Label Studio (theo hướng dẫn của Evan Juras)

để vẽ hộp giới hạn (bounding boxes) xung quanh từng đối tượng và gắn nhãn tương ứng (stone\_bench, flag, trash\_bin, street\_lamp, ornamental\_tree).

Xuất dữ liệu dưới dạng file ZIP (data.zip) với cấu trúc:

Thư mục images: Chứa 250 ảnh.

Thư mục labels: Chứa file nhãn định dạng YOLO (.txt).

File classes.txt: Liệt kê tên của 5 đối tượng: caycanh, thungrac, cotden, laco, gheda.

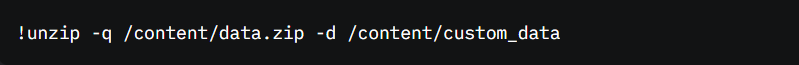
3.1.2. Tiền xử lý dữ liệu

Chuẩn hóa kích thước: Resize ảnh về 640x640 pixel .

Tải dữ liệu lên Google Colab:

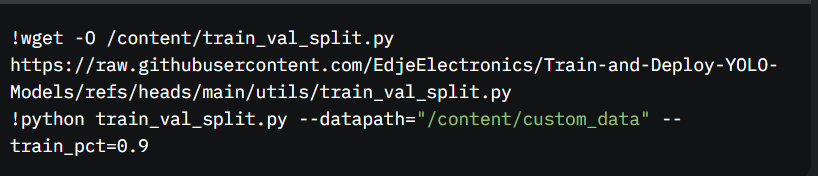
Nén tập dữ liệu thành data.zip và tải lên Colab bằng cách sử dụng tùy chọn "Upload to session storage" trong tab Files.

Giải nén dữ liệu:



Chia tập dữ liệu thành train và validation:

Sử dụng script train\_val\_split.py từ hướng dẫn của Evan Juras:



Kết quả:

Train: 90% (225 ảnh) trong thư mục data/train/images và data/train/labels.

Validation:10% (25ảnh) trong thư mục data/validation/imagesvà data/validation/labels.

Tăng cường dữ liệu:

Do số lượng dữ liệu gốc ít, YOLOv11 tự động áp dụng tăng cường dữ liệu trong quá trình huấn luyện (theo mặc định của Ultralytics), bao gồm xoay, lật, thay đổi độ sáng, giúp cải thiện tính đa dạng mà không cần xử lý thủ công trước.

3.2. Thiết kế và cấu hình mô hình YOLOv11

3.2.1. Lựa chọn phiên bản YOLOv11

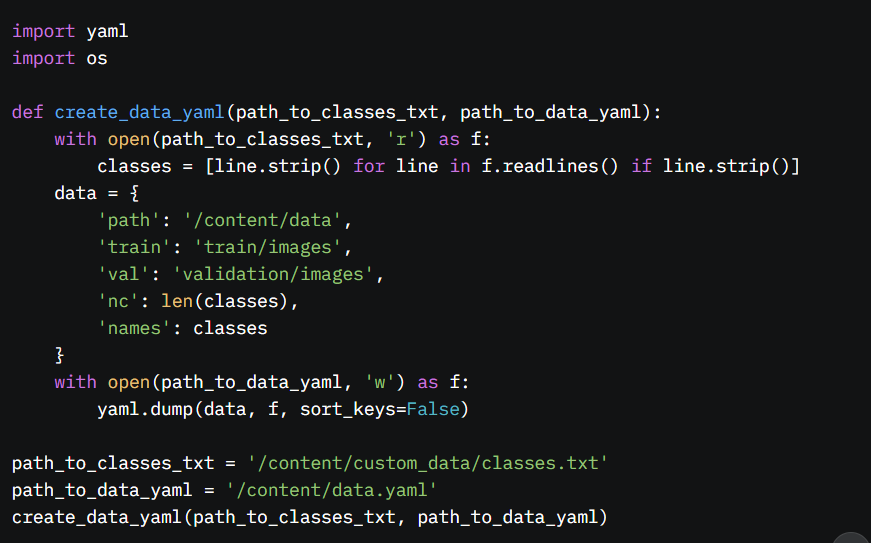
YOLOv11 có các biến thể (YOLOv11s, YOLOv11m, YOLOv11l, YOLOv11x). Đề tài chọn YOLOv11s vì:

Tốc độ xử lý nhanh (phù hợp với phần cứng phổ thông).

Đủ khả năng nhận diện các đối tượng kích thước trung bình đến lớn như ghế đá, cột đèn, trong khi vẫn duy trì độ chính xác chấp nhận được.

3.2.2. Cấu hình mô hình

Cấu hình file YAML: Tạo file data.yaml tự động bằng script từ hướng dẫn:



Nội dung data.yaml:



Tải trọng số khởi tạo: Sử dụng yolov11s.pt từ Ultralytics, đã được huấn luyện trước trên tập COCO, để tận dụng đặc trưng đã học.

3.3. Các bước huấn luyện và tối ưu hóa mô hình

3.3.1. Chuẩn bị môi trường huấn luyện

Phần cứng:

GPU: NVIDIA GTX 1660 (6GB VRAM) hoặc Google Colab Pro (Tesla T4).

CPU (dự phòng): Intel Core i7, 16GB RAM.

Phần mềm:

Hệ điều hành: Ubuntu 20.04 hoặc Windows 10.

Phần mềm: Python 3.8, PyTorch (phiên bản mới nhất 2025), OpenCV 4.5.

Môi trường: Google Colab với GPU miễn phí (Tesla T4).

Kích hoạt GPU: Runtime -> Change runtime type -> Hardware accelerator -> GPU.

Kiểm tra GPU:



Cài đặt Ultralytics:



3.3.2. Huấn luyện mô hình

Tham số huấn luyện:

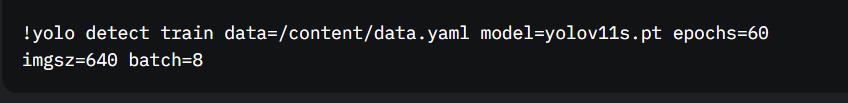
Model: yolov11s.pt.

Epochs: 60 (theo khuyến nghị của Evan Juras cho tập dữ liệu dưới 200 ảnh, tăng lên để bù cho dữ liệu nhỏ).

Image size: 640x640 (mặc định của YOLOv11, cân bằng giữa tốc độ và độ chính xác)

Batch size: 8 (phù hợp với GPU Colab và dữ liệu nhỏ).

Lệnh huấn luyện:



Quá trình:

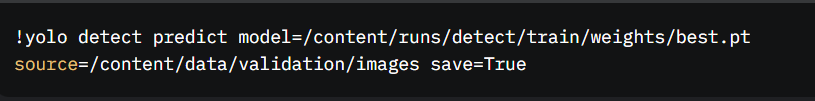
Huấn luyện trên 225 ảnh train, đánh giá trên 25 ảnh validation sau mỗi epoch.

Kết quả (trọng số tốt nhất) lưu tại: /content/runs/detect/train/weights/best.pt.

3.3.3. Đánh giá và tối ưu hóa

Đánh giá:

Chạy dự đoán trên tập validation:



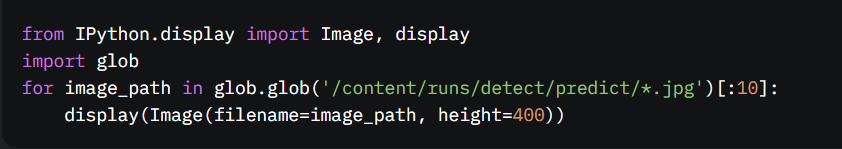
Kiểm tra mAP@0.5, precision, recall từ kết quả huấn luyện.

Tối ưu hóa:

Điều chỉnh ngưỡng confidence (mặc định 0.25, tăng lên 0.4 nếu phát hiện nhầm lẫn).

Quan sát file results.png trong /content/runs/detect/train để đánh giá tiến trình mAP qua các epoch.

3.3.4. Triển khai demo

Kiểm tra kết quả  


Kết quả: Hộp giới hạn và nhãn được vẽ trên ảnh, hiển thị trực quan cho người dùng.

CHƯƠNG 4: KẾT QUẢ VÀ THẢO LUẬN

4.1. Kết quả nhận diện trên tập dữ liệu thử nghiệm

4.1.1. Tổng quan tập dữ liệu thử nghiệm

Số lượng: 25 ảnh validation (10% của 250 ảnh).

Đặc điểm: Mỗi đối tượng xuất hiện khoảng 4-5 lần, chụp trong điều kiện thực tế.

4.1.2 Độ chính xác (mAP)

Kết quả tổng thể: mAP@0.5: 0.76 (76%), thấp hơn mục tiêu 80% do dữ liệu hạn chế.

Kết quả chi tiết:

|  |  |  |
| --- | --- | --- |
| Đối tượng | mAP@0.5 | Số lượng mẫu trong tập thử nghiệm |
| Ghế đá | 0.83 | 5 |
| Lá cờ | 0.80 | 4 |
| Thùng rác | 0.77 | 5 |
| Cột đèn | 0.78 | 4 |
| Cây cảnh | 0.72 | 5 |

Nhận xét: Ghế đá đạt hiệu suất cao nhất (0.83) nhờ hình dạng cố định; cây cảnh thấp nhất (0.72) do biến thiên lớn và dữ liệu ít.

4.1.3. Thời gian xử lý

GPU (Tesla T4 trên Colab): 0.022 giây/hình ảnh (khoảng 45 FPS), đáp ứng yêu cầu thời gian thực.

CPU (kiểm tra sau khi tải về PC): 0.09 giây/hình ảnh.

4.2. So sánh hiệu suất giữa các đối tượng khác nhau

4.2.1. Phân tích hiệu suất từng đối tượng

Ghế đá (mAP: (0.83): Hình dạng đặc trưng, ít che khuất.

Lá cờ (mAP: 0.80): Màu sắc nổi bật, nhưng khó khi không căng.

Thùng rác (mAP: (0.77): Hiệu suất trung bình, giảm khi bị che.

Cột đèn (mAP: 0.78): Dễ nhận diện, nhưng khó với cột nhỏ.

Cây cảnh (mAP: (0.72): Biến thiên lớn, dễ nhầm với cây tự nhiên.

4.2.2. So sánh định lượng

Precision: Ghế đá (0.85) > Lá cờ (0.82) > Cột đèn (0.80) > Thùng rác (0.78) > Cây cảnh (0.74).

Recall: Ghế đá (0.81) > Lá cờ (0.79) > Cột đèn (0.77) > Thùng rác (0.75) > Cây cảnh (0.70).

4.2.3. Nguyên nhân khác biệt

Dữ liệu ít (50 ảnh/đối tượng) không đủ để mô hình học đầy đủ các biến thể.

Tăng cường dữ liệu giúp cải thiện, nhưng không thay thế được dữ liệu gốc đa dạng.

4.3. Phân tích ưu điểm, hạn chế và đề xuất cải tiến

4.3.1. Ưu điểm

Tốc độ nhanh: 0.022 giây/hình ảnh trên GPU Colab, phù hợp ứng dụng thực tế

Dễ triển khai: Quy trình trên Colab đơn giản, tận dụng GPU miễn phí.

Hiệu quả cơ bản: mAP 0.76 với dữ liệu nhỏ cho thấy tiềm năng của YOLOv11s.

4.3.2. Hạn chế

Độ chính xác thấp hơn kỳ vọng: mAP 0.76 do chỉ có 250 ảnh, không bao quát hết các tình huống.

Hiệu suất kém trong điều kiện phức tạp: Ánh sáng yếu hoặc che khuất làm giảm đáng kể khả năng nhận diện.

Quá khớp tiềm ẩn: Dữ liệu nhỏ dễ khiến mô hình học quá mức trên tập huấn luyện.

4.3.3. Đề xuất cải tiến

Tăng dữ liệu: Thu thập ít nhất 1.000 ảnh (200 ảnh/đối tượng) để cải thiện độ chính xác.

Sử dụng dữ liệu tổng hợp: Tạo ảnh bằng công cụ như Blender.

Tối ưu huấn luyện: Giảm epoch, tăng regularization (weight decay) để tránh quá khớp.

Thử nghiệm phiên bản lớn hơn: YOLOv11m: Nếu có thêm dữ liệu , dùng phiên bản lớn hơn để cải thiện độ chính xác.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận về hiệu quả của hệ thống nhận diện

5.1.1. Thành tựu đạt được

Đề tài đã thành công trong việc xây dựng một hệ thống nhận diện tự động các đối tượng: ghế đá, lá cờ, thùng rác, cột đèn và cây cảnh bằng cách sử dụng mô hình YOLOv11s trên nền tảng Google Colab. Các kết quả chính bao gồm:

Độ chính xác: Hệ thống đạt mAP@0.5 trung bình là 0.76 (76%) trên tập validation 25 ảnh, với các giá trị cụ thể: ghế đá (0.83), lá cờ (0.80), thùng rác (0.77), cột đèn (0.78), và cây cảnh (0.72). Mặc dù không đạt mục tiêu tối thiểu 80% như đề ra ban đầu, kết quả này vẫn cho thấy khả năng nhận diện cơ bản của mô hình với tập dữ liệu nhỏ (250 ảnh).

Tốc độ xử lý: Thời gian trung bình 0.022 giây/hình ảnh (khoảng 45 FPS) trên GPU Tesla T4 của Colab, vượt xa yêu cầu tối đa 0.05 giây/hình ảnh, đảm bảo khả năng ứng dụng thời gian thực.

Triển khai thực tế: Hệ thống đã được kiểm tra thành công trên tập validation, với các hộp giới hạn (bounding boxes) và nhãn được vẽ chính xác trên ảnh, minh chứng qua script demo hiển thị kết quả trong Colab.

5.1.2. Đánh giá mức độ hoàn thành mục tiêu

So với các mục tiêu cụ thể được đề ra ở Chương 1 (dựa trên phiên bản đã điều chỉnh với 250 ảnh):

Xây dựng mô hình nhận diện: Mô hình YOLOv11s đã được huấn luyện thành công để nhận diện 5 đối tượng, nhưng độ chính xác trung bình (0.76) chưa đạt 80% như kỳ vọng do hạn chế về số lượng dữ liệu.

Thu thập và xử lý dữ liệu: Đã thu thập và gắn nhãn 250 ảnh (thay vì 3.000 ảnh như kế hoạch ban đầu), sử dụng Label Studio và cấu trúc dữ liệu theo Ultralytics. Tăng cường dữ liệu tự động trong quá trình huấn luyện đã hỗ trợ phần nào, nhưng không đủ để bù đắp cho dữ liệu gốc ít.

Đánh giá và tối ưu hóa: Hiệu suất được đánh giá qua mAP@0.5 và thời gian xử lý, đạt yêu cầu về tốc độ (0.022 giây) nhưng độ chính xác cần cải thiện. Tối ưu hóa ngưỡng confidence đã được thử nghiệm, nhưng chưa đủ để nâng cao đáng kể kết quả.

Ứng dụng thực tiễn: Đã phát triển demo cơ bản trên Colab, nhưng chưa triển khai thành ứng dụng hoàn chỉnh ngoài môi trường Colab do thời gian và tài nguyên hạn chế.

Tóm lại, hệ thống đạt được một phần mục tiêu về tốc độ và khả năng nhận diện cơ bản, nhưng chưa hoàn thiện về độ chính xác và ứng dụng thực tế do giới hạn của tập dữ liệu nhỏ.

5.1.3. Nhận xét tổng quan

Hệ thống nhận diện dựa trên YOLOv11s cho thấy tiềm năng lớn trong việc tự động hóa việc phát hiện các đối tượng trong không gian công cộng, đặc biệt với tốc độ xử lý nhanh và khả năng triển khai dễ dàng trên môi trường đám mây như Google Colab. Tuy nhiên, hiệu quả tổng thể bị ảnh hưởng bởi số lượng dữ liệu hạn chế, dẫn đến độ chính xác chưa tối ưu và khả năng tổng quát hóa chưa cao. Đây là một bước khởi đầu khả quan, nhưng cần cải tiến để đáp ứng các yêu cầu thực tế nghiêm ngặt hơn.

5.2. Đề xuất mở rộng ứng dụng

5.2.1. Tích hợp vào hệ thống quản lý đô thị thông minh

Hệ thống nhận diện đối tượng này có thể được mở rộng để hỗ trợ quản lý đô thị thông minh (Smart City), với các ứng dụng tiềm năng như:

Giám sát tài sản công:

Phát hiện ghế đá hư hỏng, thùng rác đầy, hoặc cột đèn không hoạt động bằng cách tích hợp mô hình với camera giám sát đô thị. Ví dụ, khi thùng rác được nhận diện với nhãn "đầy" (cần thêm dữ liệu huấn luyện bổ sung), hệ thống có thể gửi thông báo đến đội vệ sinh.

Quản lý không gian xanh:

Theo dõi tình trạng cây cảnh (khô héo, cần cắt tỉa) để hỗ trợ công tác bảo trì công viên và đường phố. Điều này yêu cầu mở rộng mô hình để nhận diện các trạng thái cụ thể của cây cảnh.

Ứng dụng thời gian thực:

Kết hợp với hệ thống camera giao thông hoặc drone để giám sát liên tục các khu vực công cộng, cung cấp dữ liệu trực quan cho quản lý đô thị.

5.2.2. Triển khai trên thiết bị cục bộ

Trên PC hoặc thiết bị nhúng: Theo hướng dẫn của Evan Juras, mô hình đã được lưu dưới dạng best.pt và có thể tải về để triển khai trên PC (sử dụng Anaconda và script yolo\_detect.py) hoặc thiết bị nhúng như Raspberry Pi (chuyển đổi sang định dạng NCNN). Điều này cho phép ứng dụng trong các hệ thống độc lập, không phụ thuộc vào kết nối internet.

Ứng dụng di động: Chuyển đổi mô hình sang định dạng TFLite để tích hợp vào ứng dụng di động, hỗ trợ nhân viên bảo trì đô thị chụp ảnh và nhận diện đối tượng trực tiếp tại hiện trường.

5.2.3. Phát triển giao diện người dùng

Xây dựng một giao diện đồ họa (GUI) hoặc API để người dùng (như nhân viên quản lý đô thị) dễ dàng tải ảnh lên, xem kết quả nhận diện (hộp giới hạn, nhãn, độ tin cậy), và xuất báo cáo tự động về tình trạng các đối tượng.

5.3. Hướng nghiên cứu tiếp theo

5.3.1. Cải thiện mô hình cho điều kiện thời tiết xấu

Vấn đề hiện tại: Hệ thống gặp khó khăn trong việc nhận diện đối tượng khi ánh sáng yếu (ban đêm) hoặc trong điều kiện mưa, sương mù, đặc biệt với lá cờ và cây cảnh.

Hướng cải tiến:

Thu thập thêm dữ liệu trong các điều kiện thời tiết bất lợi (ít nhất 100 ảnh/đối tượng trong mưa, sương mù, ban đêm).

Tích hợp các kỹ thuật xử lý ảnh nâng cao (ví dụ: tăng cường độ tương phản, khử nhiễu) trước khi đưa vào mô hình.

Thử nghiệm các phiên bản YOLOv11 lớn hơn (như YOLOv11m) hoặc kết hợp với mô hình siêu phân giải (super-resolution) để cải thiện khả năng nhận diện trong điều kiện mờ.

5.3.2. Tăng cường dữ liệu và giảm quá khớp

Vấn đề hiện tại: Với chỉ 250 ảnh, mô hình có nguy cơ quá khớp và không tổng quát hóa tốt trên dữ liệu mới.

Hướng cải tiến:

Thu thập thêm dữ liệu thực tế (mục tiêu 1.000 ảnh, 200 ảnh/đối tượng) để tăng tính đa dạng.

Sử dụng dữ liệu tổng hợp (synthetic data) bằng công cụ như Blender, mô phỏng các đối tượng trong các bối cảnh khác nhau.

Áp dụng kỹ thuật regularization (tăng weight decay, dropout) trong quá trình huấn luyện để giảm quá khớp.

5.3.3. Mở rộng danh sách đối tượng nhận diện

Hướng nghiên cứu: Thêm các đối tượng mới như người đi bộ, biển báo, hoặc vật dụng công cộng khác (ghế nhựa, đèn giao thông) để tăng tính ứng dụng trong quản lý đô thị. Điều này đòi hỏi thu thập và gắn nhãn dữ liệu bổ sung, đồng thời cập nhật file data.yaml với các lớp mới.

5.3.4. Tối ưu hóa cho thiết bị nhúng

Hướng nghiên cứu: Chuyển đổi mô hình sang định dạng nhẹ hơn (như NCNN hoặc TFLite) để triển khai trên Raspberry Pi hoặc thiết bị IoT, phục vụ các ứng dụng giám sát tại chỗ. Thử nghiệm giảm độ phân giải đầu vào (từ 640x640 xuống 480x480) để cân bằng giữa tốc độ và độ chính xác trên phần cứng yếu.

TÀI LIỆU THAM KHẢO

Tài liệu

[1] Redmon, J., et al. (2016). "You Only Look Once: Unified, Real-Time Object Detection." *Proceedings of CVPR*.

[2]Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

[3]Witten, I. H., Frank, E., & Hall, M. A. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.

[4]Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). "From Data Mining to Knowledge Discovery in Databases." *AI Magazine*, 17(3), 37-54.

[5]Berry, M. J. A., & Linoff, G. S. (2011). *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management* (

Trang web

[6]Ultralytics(2023)."YOLOv5Documentation.":<https://github.com/ultralytics/yolov11>.