**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN MÔN HỌC**

**XÂY DỰNG HỆ THỐNG ỨNG DỤNG MÔ HÌNH DEEP LEARNING YOLO ĐỂ PHÁT HIỆN ĐỐI TƯỢNG TRONG ẢNH**

**GIẢNG VIÊN HƯỚNG DẪN: ThS Võ Quang Hoàng Khang**

**SINH VIÊN THỰC HIỆN 1: Nguyễn Phú Sang**

**MÃ SỐ SINH VIÊN: 21023391**

**SINH VIÊN THỰC HIỆN 2: Lưu Chí Tài**

**MÃ SỐ SINH VIÊN: 21080681**

*TP HỒ CHÍ MINH, Tháng 05 năm 2025*

**INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**GRADUATION THESIS**

**BUILDING A SYSTEM USING YOLO DEEP LEARNING MODEL TO DETECTING OBJECTS IN IMAGES**

**SUPERVISOR: MSc. Vo Quang Hoang Khang**

**STUDENT NAME: Nguyen Phu Sang**

**STUDENT CODE: 21023391**

**STUDENT NAME: Luu Chi Tai**

**STUDENT CODE: 21080681**

*HO CHI MINH CITY, Month 05 year 2025*

**TÓM TẮT**

Trong thời đại số hóa, lượng dữ liệu hình ảnh và video tăng nhanh chóng, kéo theo nhu cầu xử lý và trích xuất thông tin từ ảnh trở nên ngày càng quan trọng. Một trong những bài toán phổ biến và thiết thực là phát hiện đối tượng trong ảnh nó đóng vai trò quan trọng trong nhiều ứng dụng như giám sát an ninh, xe tự hành, chăm sóc sức khỏe, thương mại điện tử, giao thông v.v. Với đồ án này tập trung vào việc xây dựng một hệ thống ứng dụng mô hình Deep Learning – cụ thể là mô hình YOLO (You Only Look Once) – để thực hiện nhiệm vụ phát hiện đối tượng trong ảnh đầu vào.

Trong quá trình thực hiện, nhóm chúng em đã tìm hiểu lý thuyết về bài toán phát hiện đối tượng, đặc điểm kỹ thuật của mô hình YOLO qua các phiên bản, và lựa chọn phiên bản phù hợp để triển khai. Dữ liệu sử dụng là bộ dữ liệu công khai như Pascal VOC. Hệ thống được xây dựng bằng ngôn ngữ Python, sử dụng các thư viện như PyTorch, OpenCV và Ultralytics YOLO.

Kết quả thực nghiệm cho thấy mô hình có khả năng phát hiện chính xác các đối tượng trong ảnh với độ chính xác phù hợp với yêu cầu ứng dụng thời gian thực. Hệ thống được đánh giá là hiệu quả, dễ triển khai và có tiềm năng ứng dụng rộng rãi trong thực tế.

**LỜI CẢM ƠN**

Trong suốt thời gian làm đồ án, nhóm chúng em đã luôn nhận được nhiều sự quan tâm, hướng dẫn và giúp đỡ tận tình của các thầy cô giáo trong khoa Công nghệ thông tin cùng với sự động viên giúp đỡ từ bạn bè và gia đình.

Lời đầu tiên nhóm chúng em xin chân thành cảm ơn Ban giám hiệu Trường Đại học Công nghiệp thành phố Hồ Chí Minh, Ban chủ nhiệm khoa Công nghệ thông tin đã luôn tận tình quan tâm giúp đỡ nhóm chúng em trong suốt thời gian học tập tại trường.

Đặc biệt nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy hướng dẫn ThS. Võ Quang Hoàng Khang đã trực tiếp hướng dẫn, giúp đỡ nhóm chúng em hoàn thành khoá luận này.

Nhóm chúng em cũng xin gửi lời cảm ơn đến gia đình, người thân và bạn bè đã luôn ở bên quan tâm, giúp đỡ và động viên chúng nhóm chúng em hoàn thành khoá luận tốt nghiệp này.

Chúng nhóm chúng em kính chúc Thầy Cô và các bạn sức khỏe, thành công trong học tập.

Nhóm chúng em xin trân trọng cảm ơn!

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**TP. Hồ Chí Minh, ngày…. tháng …. năm….**

CHỮ KÝ CỦA GIẢNG VIÊN

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 1**

**TP. Hồ Chí Minh, ngày…. Tháng …. năm….**

CHỮ KÝ CỦA GIẢNG VIÊN

**NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN 2**

**TP. Hồ Chí Minh, ngày…. tháng …. năm….**

CHỮ KÝ CỦA GIẢNG VIÊN

**LỜI MỞ ĐẦU**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, trí tuệ nhân tạo (AI) và học sâu (Deep Learning) ngày càng khẳng định vai trò then chốt trong việc giải quyết nhiều bài toán thực tiễn thuộc các lĩnh vực như y tế, giáo dục, giao thông, an ninh, và đặc biệt là xử lý ảnh và video. Trước nhu cầu ngày càng gia tăng đối với các mô hình công nghệ hiện đại, ngành Khoa học máy tính luôn cần đến đội ngũ chuyên gia có kiến thức chuyên sâu và kỹ năng thực hành vững vàng để nghiên cứu, phát triển và triển khai hiệu quả các mô hình máy học và học sâu, từ đó đáp ứng các thách thức thực tế của xã hội hiện đại.

Một trong những ứng dụng nổi bật của học sâu trong lĩnh vực thị giác máy tính là bài toán phát hiện đối tượng (Object Detection). Bài toán này cho phép hệ thống có thể nhận diện và xác định vị trí của các đối tượng xuất hiện trong ảnh hoặc video, từ đó phục vụ hiệu quả cho nhiều mục đích ứng dụng như giám sát an ninh, xe tự hành, kiểm soát giao thông, v.v.

Xuất phát từ ý nghĩa thực tiễn đó, nhóm chúng em đã lựa chọn đề tài:

“Xây dựng hệ thống ứng dụng mô hình Deep Learning YOLO để phát hiện đối tượng trong ảnh” cho đồ án cuối kỳ của mình. Đây là một đề tài có tính phổ biến, được nghiên cứu và ứng dụng rộng rãi trong các hệ thống thực tế hiện nay.

Đồ án tập trung vào việc nghiên cứu và cải tiến mô hình phát hiện đối tượng dựa trên mô hình học sâu tiên tiến YOLO – nổi bật với khả năng xử lý nhanh và độ chính xác cao, phù hợp với yêu cầu của các hệ thống hoạt động trong thời gian thực. Mô hình được huấn luyện và đánh giá trên tập dữ liệu Pascal VOC thông qua các chỉ số đánh giá phổ biến như: độ chính xác trung bình (mAP@0.5), độ chính xác (precision), và độ bao phủ (recall). Kết quả thực nghiệm cho thấy mô hình đề xuất có khả năng phát hiện hiệu quả các đối tượng trong ảnh, từ đó khẳng định tiềm năng ứng dụng thực tiễn của công nghệ này.

Cuối cùng, nhóm chúng em xin gửi lời cảm ơn chân thành đến quý Thầy/Cô – những người đã tận tình hướng dẫn, định hướng và hỗ trợ nhóm chúng em trong suốt quá trình thực hiện đồ án. Đồng thời, nhóm chúng em cũng xin cảm ơn các anh/chị và bạn bè đã giúp đỡ, động viên và tạo điều kiện thuận lợi để nhóm chúng em hoàn thành đề tài một cách tốt nhất.

Mặc dù đã nỗ lực hết sức, nhưng do giới hạn về kiến thức và kinh nghiệm, đồ án không thể tránh khỏi những thiếu sót. Nhóm chúng em rất mong nhận được sự góp ý quý báu từ Thầy/Cô và các bạn để hoàn thiện hơn trong những nghiên cứu và dự án tiếp theo.

**MỤC LỤC**

[**CHƯƠNG 1.**  **TỔNG QUAN** 1](#_Toc198561706)

[**1.1 Lý do chọn đề tài** 1](#_Toc198561707)

[**1.2 Mục tiêu nghiên cứu** 2](#_Toc198561708)

[**1.3 Phạm vi nghiên cứu** 3](#_Toc198561709)

[**1.4 Phương pháp nghiên cứu** 3](#_Toc198561710)

[**1.5 Cấu trúc đồ án.** 5](#_Toc198561711)

[**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT** 6](#_Toc198561712)

[**2.1. Tổng quan về Thị giác máy tính (Computer Vision)** 6](#_Toc198561713)

[**2.2. Phát hiện đối tượng (Object Detection)** 6](#_Toc198561714)

[**2.3. Tổng quan về mô hình YOLO (You Only Look Once)** 6](#_Toc198561715)

[**2.3.1 Các phiên bản YOLO** 6](#_Toc198561716)

[**2.3.2. Ưu điểm của YOLO** 7](#_Toc198561717)

[**2.4. Các phương pháp đánh giá mô hình** 7](#_Toc198561718)

[**2.4.1. Độ chính xác trung bình tại IoU = 0.5 (mAP@0.5)** 7](#_Toc198561719)

[**2.4.2. Độ chính xác trung bình tại IoU = 0.5 đến 0.95 (mAP@0.5:0.95)** 7](#_Toc198561720)

[**2.4.3. Precision (Độ chính xác dự đoán dương)** 8](#_Toc198561721)

[**2.4.4. F1 – Score** 8](#_Toc198561722)

[**2.4.5. Recall (Độ nhạy)** 8](#_Toc198561723)

[**2.4.6. Ma trận nhầm lẫn (Confusion Matrix)** 9](#_Toc198561724)

[**CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT** 10](#_Toc198561725)

[**3.1. Tổng quan mô hình YoLo** 10](#_Toc198561726)

[**3.2. Kiến trúc tổng thể mô hình đề xuất** 10](#_Toc198561727)

[**3.2.1. Giai đoạn 1 – Trích xuất đặc trưng (Backbone)** 12](#_Toc198561728)

[**3.3.2. Giai đoạn 2 – Tăng cường và hợp nhất đặc trưng (Neck)** 12](#_Toc198561729)

[**3.3.3. Giai đoạn 3 – Dự đoán và phân loại (Head)** 12](#_Toc198561730)

[**3.3. Ưu điểm của mô hình đề xuất** 13](#_Toc198561731)

[**CHƯƠNG 4. THỰC NGHIỆM** 14](#_Toc198561732)

[**4.1. Môi trường thực nghiệm** 14](#_Toc198561733)

[**4.1.1. Cấu hình phần cứng** 14](#_Toc198561734)

[**4.1.2. Cấu hình phần mềm** 14](#_Toc198561735)

[**4.1.3. Thiết lập môi trường** 14](#_Toc198561736)

[**4.1.4. Lý do chọn Kaggle** 15](#_Toc198561737)

[**4.2. Tập dữ liệu** 15](#_Toc198561738)

[**4.2.1. Nguồn dữ liệu** 15](#_Toc198561739)

[**4.2.2. Tổng quan về dữ liệu** 16](#_Toc198561740)

[**4.2.3. Số lượng dữ liệu** 16](#_Toc198561741)

[**4.2.4. Tiền xử lý dữ liệu** 17](#_Toc198561742)

[**4.2.5. Chia dữ liệu** 17](#_Toc198561743)

[**4.2.6. Lý do chọn tập dữ liệu** 17](#_Toc198561744)

[**4.3. Ứng dụng thực nghiệm** 18](#_Toc198561745)

[**4.3.1. Quy trình huấn luyện mô hình** 18](#_Toc198561746)

[**4.3.2. Cấu hình huấn luyện** 18](#_Toc198561747)

[**4.3.3. Kết quả thực nghiệm** 18](#_Toc198561748)

[**4.4. Đánh giá kết quả** 22](#_Toc198561749)

[**4.4.1 Ma trận nhầm lẫn (Confusion Matrix)** 22](#_Toc198561750)

[**4.4.2 Chỉ số mAP@0.5 (mean Average Precision)** 23](#_Toc198561751)

[**4.4.3 Precision và Recall theo từng lớp** 24](#_Toc198561752)

[**4.5. Xây dựng hệ thống bằng Tkiner** 25](#_Toc198561753)

[**4.5.1. Màn hình đăng nhập** 25](#_Toc198561754)

[**4.5.2. Màn hình chính** 26](#_Toc198561755)

[**4.6. Tổng kết** 28](#_Toc198561756)

[**CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 30](#_Toc198561757)

[**5.1. Kết luận** 30](#_Toc198561758)

[**5.2. Hạn chế** 30](#_Toc198561759)

[**5.3 Hướng phát triển** 31](#_Toc198561760)

**MỤC LỤC HÌNH ẢNH**

Hình 1. Kiến trúc mô hình YOLO. Hình ảnh bởi người dùng GitHub RangeKing 11

[Hình 2. Biểu đồ các chỉ số đánh giá mô hình Yolov8 19](#_Toc184148871)

[Hình 3. Bảng so sánh các giá trị mAP 20](#_Toc184148872)

Hình 4. Độ chính xác (Precision) theo từng lớp đối tượng 20

[Hình 5. Độ phủ (Recal) theo từng lớp đối tượng 21](#_Toc184148874)

Hình 6. mAP@0.5 theo từng lớp đối tượng 21

[Hình 7. Ma trận nhầm lẫn 22](#_Toc184148876)

[Hình 8. Màn hình đăng nhập hệ thống 26](#_Toc184148876)

[Hình 9. Màn hình chính của hệ thống 28](#_Toc184148876)

**MỤC LỤC BẢNG BIỂU**

[Bảng 1. Kết quả thực nghiệm mô hình Yolov8 19](#_Toc184207958)

**DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Nghĩa |
| **Pascal VOC** | Visual Object Classes | Là bộ dữ liệu phát hiện, phân đoạn và phân loại đối tượng nổi tiếng. Bộ dữ liệu này được thiết kế để khuyến khích nghiên cứu về nhiều loại đối tượng khác nhau và thường được sử dụng để đánh giá chuẩn các mô hình thị giác máy tính. |
| **YOLO** | You Only Look Once | Mô hình phát hiện đối tượng và phân đoạn hình ảnh phổ biến, được phát triển bởi Joseph Redmon và Ali Farhadi tại Đại học Washington. Ra mắt vào năm 2015, YOLO được ưa chuộng vì tốc độ và độ chính xác cao. |

# **CHƯƠNG 1.** **TỔNG QUAN**

## **1.1 Lý do chọn đề tài**

Trong những năm gần đây, trí tuệ nhân tạo (Artificial Intelligence – AI), đặc biệt là học sâu (Deep Learning), đã trở thành một trong những lĩnh vực công nghệ mũi nhọn, mang lại nhiều bước tiến quan trọng trong các ngành công nghiệp như y tế, giao thông, giáo dục, sản xuất, và đặc biệt là lĩnh vực xử lý ảnh và video. Một trong những nhánh quan trọng của AI là thị giác máy tính (Computer Vision), cho phép máy tính có khả năng “nhìn”, phân tích và hiểu được nội dung trong hình ảnh hoặc video tương tự như con người.

Một trong những bài toán trọng tâm của thị giác máy tính là phát hiện đối tượng (Object Detection), với mục tiêu xác định được loại đối tượng và vị trí của nó trong một ảnh đầu vào. Bài toán này đóng vai trò cốt lõi trong nhiều ứng dụng quan trọng như:

* Hệ thống giám sát an ninh và cảnh báo xâm nhập,
* Xe tự hành và hỗ trợ người lái,
* Robot công nghiệp và nông nghiệp,
* Phân tích hành vi trong bán lẻ và thương mại,
* Nhận diện khuôn mặt và kiểm soát truy cập, v.v.

Theo báo cáo của Allied Market Research, thị trường thị giác máy tính toàn cầu được định giá khoảng 9,4 tỷ USD vào năm 2020 và dự kiến đạt hơn 41 tỷ USD vào năm 2030, với tốc độ tăng trưởng hàng năm hơn 16% [1].

Trong bối cảnh đó, YOLO (You Only Look Once) là một trong những mô hình phát hiện đối tượng nổi bật, với ưu điểm vượt trội về tốc độ xử lý và độ chính xác cao, phù hợp với các ứng dụng yêu cầu phản hồi thời gian thực. Không giống như các mô hình trước đây sử dụng hai giai đoạn (two-stage detection như R-CNN), YOLO sử dụng kiến trúc một giai đoạn (one-stage detection), giúp giảm thiểu độ trễ và tối ưu hiệu suất [2]. Các phiên bản cải tiến như YOLOv4, YOLOv5, và mới nhất là YOLOv8 đã liên tục cải thiện khả năng phát hiện trong điều kiện môi trường phức tạp.

Từ thực tiễn ứng dụng và xu hướng phát triển mạnh mẽ của công nghệ, nhóm chúng em lựa chọn đề tài: “Xây dựng hệ thống ứng dụng mô hình Deep Learning YOLO để phát hiện đối tượng trong ảnh” như một cơ hội để nghiên cứu sâu hơn về lĩnh vực thị giác máy tính, học sâu và áp dụng chúng vào một hệ thống thực tế.

Việc thực hiện đề tài không chỉ giúp nhóm chúng em củng cố kiến thức lý thuyết, rèn luyện kỹ năng lập trình, xử lý dữ liệu và đánh giá mô hình, mà còn giúp nhóm chúng em hiểu rõ hơn về quy trình triển khai một mô hình học sâu vào thực tiễn. Đồng thời, đề tài này có thể làm nền tảng cho các nghiên cứu chuyên sâu hơn sau này như: theo dõi đối tượng (Object Tracking), nhận diện hành vi, phân tích video thời gian thực,…

## **1.2 Mục tiêu nghiên cứu**

Mục tiêu chính của đề tài là xây dựng một hệ thống ứng dụng mô hình Deep Learning cụ thể là YOLO – để phát hiện các đối tượng trong ảnh một cách chính xác và nhanh chóng, đáp ứng yêu cầu của các bài toán xử lý thời gian thực trong thực tế.

Cụ thể, đề tài hướng đến việc đạt được các mục tiêu sau:

1. Tìm hiểu và hệ thống hóa kiến thức nền tảng về học sâu (Deep Learning), thị giác máy tính (Computer Vision) và các thuật toán phát hiện đối tượng, đặc biệt là các mô hình thuộc họ YOLO từ phiên bản đầu tiên đến phiên bản mới (YOLOv5, YOLOv8).
2. Thu thập và xử lý tập dữ liệu chuẩn (Pascal VOC), thực hiện các bước tiền xử lý dữ liệu như mã hóa nhãn, chuyển đổi định dạng, và chia tập huấn luyện/kiểm tra phù hợp để huấn luyện mô hình.
3. Xây dựng hệ thống phát hiện đối tượng sử dụng mô hình YOLO, bao gồm huấn luyện mô hình, đánh giá hiệu suất với các chỉ số như:

* Mean Average Precision (mAP@0.5)
* Precision
* Recall
* F1-score

1. Triển khai một hệ thống thử nghiệm để minh họa khả năng phát hiện đối tượng trong ảnh tĩnh. Hệ thống có thể chạy trên GPU hoặc CPU, phục vụ mục tiêu trình diễn thời gian thực.
2. Đánh giá, so sánh kết quả thực nghiệm và rút ra nhận xét về khả năng ứng dụng của mô hình trong thực tế, từ đó đề xuất hướng cải tiến hoặc mở rộng trong các nghiên cứu tiếp.

## **1.3 Phạm vi nghiên cứu**

Đối tượng nghiên cứu: Các hình ảnh kỹ thuật số thuộc tập dữ liệu Pascal VOC 2012, trong đó tập trung vào việc phát hiện và định vị các đối tượng như người, xe cộ, động vật và các vật thể thường gặp trong môi trường tự nhiên và đô thị. Mỗi ảnh đều được gán nhãn với tọa độ hộp giới hạn (bounding box) và lớp đối tượng tương ứng.

Phạm vi kỹ thuật: Đề tài sử dụng mô hình học sâu YOLO (You Only Look Once) để thực hiện bài toán phát hiện đối tượng. Phiên bản được nghiên cứu bao YOLOv8. Quá trình huấn luyện và đánh giá mô hình được thực hiện với sự hỗ trợ của ngôn ngữ lập trình Python cùng các thư viện phổ biến như PyTorch, OpenCV, Ultralytics YOLO.

Phạm vi kiểm tra: Mô hình được huấn luyện và kiểm tra trên tập dữ liệu Pascal VOC 2012, sử dụng phương pháp chia dữ liệu theo tỉ lệ tiêu chuẩn (80% huấn luyện, 20% kiểm tra). Việc đánh giá mô hình được thực hiện dựa trên các chỉ số:

* mAP@0.5 (mean Average Precision tại IoU = 0.5)
* mAP@0.5-0.95 (mean Average Precision tại IoU = 0.5 to 0.95)
* Precision
* Recall
* F1-score
* Confusion matrix (ma trận nhầm lẫn)

## **1.4 Phương pháp nghiên cứu**

Để thực hiện đề tài, nhóm sử dụng phương pháp nghiên cứu thực nghiệm kết hợp với nghiên cứu tài liệu. Cụ thể, các bước nghiên cứu được tiến hành như sau:

* **Nghiên cứu lý thuyết**
  + Tìm hiểu tổng quan về bài toán phát hiện đối tượng (Object Detection), các kỹ thuật trích xuất đặc trưng, và các phương pháp học sâu (Deep Learning) trong lĩnh vực thị giác máy tính (Computer Vision).
  + Nghiên cứu chi tiết về kiến trúc và nguyên lý hoạt động của các phiên bản mô hình YOLO, bao gồm YOLOv5 và YOLOv8.
  + Tìm hiểu các chỉ số đánh giá hiệu suất mô hình như: precision, recall, F1-score và mean Average Precision (mAP),..
* **Phân tích và xử lý dữ liệu**
  + Khai thác tập dữ liệu Pascal VOC 2012, bao gồm hơn 11.000 hình ảnh với hơn 20 lớp đối tượng.
  + Tiền xử lý dữ liệu: chuyển đổi định dạng ảnh, chuẩn hóa kích thước, và chuyển đổi định dạng nhãn về định dạng phù hợp với mô hình YOLO.
  + Chia tập dữ liệu thành hai phần: 80% dữ liệu để huấn luyện, 20% để kiểm tra mô hình.
* **Xây dựng mô hình**
  + Triển khai mô hình phát hiện đối tượng sử dụng YOLOv8 bằng thư viện Ultralytics YOLO với nền tảng PyTorch.
  + Tinh chỉnh các siêu tham số: learning rate, batch size, số epoch, và các thông số anchor box để tối ưu hóa mô hình.
  + Kết hợp các kỹ thuật như FPN (Feature Pyramid Network) để cải thiện khả năng phát hiện đối tượng nhỏ hoặc nhiều kích thước.
* **Huấn luyện và đánh giá mô hình**
  + Tiến hành huấn luyện mô hình trên tập huấn luyện, giám sát quá trình hội tụ và điều chỉnh siêu tham số nếu cần.
  + Đánh giá hiệu suất mô hình trên tập kiểm tra sử dụng các chỉ số: mAP@0.5, precision, recall, và F1-score, ma trận nhầm lẫn,..
  + Trực quan hóa kết quả dự đoán bằng hình ảnh để quan sát hiệu quả thực tế của mô hình.
* **So sánh và rút ra kết luận**
  + So sánh hiệu suất giữa phiên bản YOLO được sử dụng.
  + Đánh giá ưu điểm, hạn chế của từng mô hình và đưa ra kết luận về khả năng áp dụng thực tế.

## **1.5 Cấu trúc đồ án.**

Chương 1: Tổng quan

Chương 2: Cơ sở lý thuyết

Chương 3: Mô hình đề xuất

Chương 4: Thực nghiệm và đánh giá kết quả

Chương 5: Kết luận và hướng phát triển

Tài liệu tham khảo.

# **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

## **2.1. Tổng quan về Thị giác máy tính (Computer Vision)**

Thị giác máy tính là một lĩnh vực của trí tuệ nhân tạo tập trung vào việc xây dựng các hệ thống có khả năng hiểu và phân tích hình ảnh hoặc video như con người. Các bài toán tiêu biểu trong lĩnh vực này bao gồm: phân loại ảnh (image classification), phát hiện đối tượng (object detection), phân đoạn ảnh (image segmentation), theo dõi đối tượng (object tracking), v.v.

Thị giác máy tính đóng vai trò quan trọng trong nhiều ứng dụng thực tế như xe tự hành, giám sát an ninh, y tế, robot công nghiệp, và nhận diện khuôn mặt.

## **2.2. Phát hiện đối tượng (Object Detection)**

Phát hiện đối tượng là quá trình xác định vị trí (bounding box) và phân loại các đối tượng xuất hiện trong ảnh. Đây là một bài toán phức tạp hơn so với phân loại ảnh thông thường.

Có hai nhóm phương pháp phát hiện đối tượng chính:

* Two-stage detectors: như R-CNN, Fast R-CNN, Faster R-CNN – hoạt động theo hai giai đoạn: tạo vùng đề xuất (region proposal) và phân loại.
* One-stage detectors: như YOLO, SSD – thực hiện phát hiện đối tượng trong một bước duy nhất, nhanh hơn và phù hợp với các ứng dụng thời gian thực..

## **2.3. Tổng quan về mô hình YOLO (You Only Look Once)**

YOLO là một trong những mô hình phát hiện đối tượng nổi bật nhờ tốc độ xử lý nhanh và độ chính xác cao. YOLO chia ảnh đầu vào thành một lưới và dự đoán bounding boxes cùng với xác suất thuộc các lớp đối tượng cho mỗi ô lưới.

### **2.3.1 Các phiên bản YOLO**

* YOLOv1 - v3: Khởi đầu từ YOLOv1 [Redmon et al., 2016], mô hình có kiến trúc đơn giản, tốc độ cao nhưng độ chính xác còn hạn chế. YOLOv2 và YOLOv3 cải tiến về mặt kiến trúc và đa cấp độ đặc trưng.
* YOLOv4: Tăng hiệu suất đáng kể nhờ sử dụng CSPDarknet53, SPP, PANet và nhiều kỹ thuật tối ưu như Mish activation, Mosaic augmentation.
* YOLOv5: Được phát triển bởi Ultralytics (không chính thức từ tác giả ban đầu), dễ sử dụng, linh hoạt và phổ biến trong thực tế.
* YOLOv6/v7/v8: Các phiên bản cải tiến tiếp theo, trong đó YOLOv8 (2023) hỗ trợ cả detection và segmentation, tối ưu hóa về tốc độ, chính xác, và khả năng mở rộng. YOLOv8 loại bỏ cấu trúc anchor truyền thống, thay thế bằng anchor-free detection và sử dụng kiến trúc mới (C2f, BiPAFPN).

### **2.3.2. Ưu điểm của YOLO**

* Tốc độ xử lý cao, thích hợp cho ứng dụng thời gian thực.
* Dễ triển khai và tùy chỉnh.
* Khả năng phát hiện nhiều đối tượng trong một khung hình.

## **2.4. Các phương pháp đánh giá mô hình**

Hiệu suất của mô hình phát hiện đối tượng được đánh giá dựa trên tập kiểm tra thông qua các chỉ số quan trọng, bao gồm: mAP@0.5, Precision, Recall, F1-Score, và ma trận nhầm lẫn (Confusion Matrix). Các chỉ số này phản ánh khả năng mô hình nhận diện đúng, đầy đủ và chính xác các đối tượng trong ảnh.

### **2.4.1. Độ chính xác trung bình tại IoU = 0.5 (mAP@0.5)**

Chỉ số mAP@0.5 đo lường độ chính xác trung bình của mô hình tại ngưỡng IoU = 0.5 – nghĩa là vùng dự đoán phải trùng với vùng thực tế ít nhất 50% mới được tính là đúng. Đây là chỉ số quan trọng trong phát hiện đối tượng vì nó phản ánh mức độ khớp giữa khung dự đoán và khung thật.

[mAP@0.5](mailto:mAP@0.5) =

Trong đó:

* APi: là Average Precision của lớp thứ i
* N: là tổng số lớp

### **2.4.2. Độ chính xác trung bình tại IoU = 0.5 đến 0.95 (mAP@0.5:0.95)**

Chỉ số mAP@0.5:0.95 là thước đo tổng quát và nghiêm ngặt hơn mAP@0.5. Nó tính trung bình của các chỉ số Average Precision tại các ngưỡng Intersection over Union (IoU) khác nhau, từ 0.5 đến 0.95, cách nhau 0.05. Điều này giúp đánh giá mô hình không chỉ về khả năng phát hiện đúng mà còn về độ chính xác vị trí của khung dự đoán.

[mAP@0.5:0.95](mailto:mAP@0.5:0.95) =

### **2.4.3. Precision (Độ chính xác dự đoán dương)**

Đánh giá tỉ lệ dự đoán đúng trong tất cả các dự đoán mà mô hình cho là có đối tượng. Precision cao nghĩa là mô hình ít bị "báo động giả".

Precision =

Trong đó:

* TP: Dự đoán dương đúng (True Positives).
* FP: Dự đoán dương sai (False Positives).

### **2.4.4. F1 – Score**

F1-Score là trung bình điều hòa giữa Precision và Recall. F1-Score là chỉ số quan trọng trong các bài toán phân loại không cân bằng, khi mà cả Precision và Recall đều quan trọng. F1-Score giúp cân bằng giữa việc không bỏ sót các trường hợp các lớp và không nhầm lẫn quá nhiều giữa các lớp.

F1 =

### **2.4.5. Recall (Độ nhạy)**

Đánh giá khả năng mô hình phát hiện được tất cả các đối tượng thực sự có trong ảnh. Recall cao chứng tỏ mô hình ít bỏ sót đối tượng.

Recall =

Trong đó:

* TP: Dự đoán dương đúng (True Positives).
* FN: Số lượng trường hợp dương bị bỏ sót (False Negatives

### **2.4.6. Ma trận nhầm lẫn (Confusion Matrix)**

Ma trận nhầm lẫn thể hiện chi tiết số lượng dự đoán đúng và sai cho từng lớp. Trong bài toán phát hiện đối tượng, có thể sử dụng ma trận nhầm lẫn theo số lượng phát hiện đúng/sai trên từng loại đối tượng hoặc tổng hợp theo các ngưỡng IoU. Ma trận này giúp phân tích sâu các lỗi mà mô hình thường gặp, từ đó đưa ra hướng cải tiến phù hợp.

# **CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT**

## **3.1. Tổng quan mô hình YoLo**

YOLO (You Only Look Once) là một trong những mô hình nổi bật nhất trong lĩnh vực phát hiện đối tượng (object detection) nhờ tốc độ xử lý nhanh và độ chính xác cao. Khác với các phương pháp phát hiện đối tượng truyền thống như R-CNN hay Fast R-CNN, vốn tách riêng hai giai đoạn là trích xuất vùng đề xuất và phân loại đối tượng, YOLO thực hiện toàn bộ quy trình trong một lần dự đoán duy nhất.

Cốt lõi của YOLO là chia ảnh đầu vào thành một lưới (grid), và tại mỗi ô lưới, mô hình dự đoán bounding boxes kèm theo xác suất thuộc về các lớp đối tượng. Điều này giúp giảm thiểu thời gian xử lý và tối ưu hiệu suất trên các thiết bị thời gian thực.

Các phiên bản gần đây như YOLOv4, YOLOv5, YOLOv7, YOLOv8 đã được cải tiến đáng kể với các kiến trúc backbone mạnh mẽ (CSPDarknet53, EfficientNet), các thành phần tăng cường như PANet, SPP, cùng các kỹ thuật tối ưu hóa như Mosaic augmentatio.

Trong chương này, nhóm chúng em trình bày chi tiết mô hình được đề xuất sử dụng trong bài toán phát hiện đối tượng trong ảnh. Mô hình được xây dựng dựa trên kiến trúc YOLOv8 – một trong những phiên bản tiên tiến và hiệu quả nhất của dòng mô hình You Only Look Once (YOLO), nổi bật với khả năng phát hiện nhanh, chính xác và hiệu quả trong thời gian thực. Cấu trúc của mô hình được thiết kế nhằm tối ưu hóa hiệu suất nhận dạng đối tượng ở nhiều kích thước khác nhau, đồng thời giảm thiểu độ phức tạp tính toán và cải thiện khả năng tổng quát hóa của mô hình.n, CIoU loss, và auto-learning anchor.

## **3.2. Kiến trúc tổng thể mô hình đề xuất**

Đầu vào của mô hình là ảnh RGB đã qua bước tiền xử lý và chuẩn hóa, với kích thước cố định 640x640x3 nhằm đảm bảo mô hình nhận dạng hiệu quả và đồng nhất.

Đầu ra của mô hình là tập hợp các bounding box cùng với xác suất và nhãn phân loại các đối tượng được phát hiện trong ảnh. Đầu ra này cho phép mô hình vừa định vị chính xác vị trí, vừa phân loại đa dạng các đối tượng trong cùng một ảnh.

* Mô hình YOLOv8 được chia thành ba giai đoạn chính như sau:

A screenshot of a computer

AI-generated content may be incorrect.

**Hình 1:** Kiến trúc mô hình YOLO. Hình ảnh bởi người dùng GitHub RangeKing

### **3.2.1. Giai đoạn 1 – Trích xuất đặc trưng (Backbone)**

Giai đoạn đầu tiên trong kiến trúc YOLOv8 sử dụng CSPDarknet làm backbone để trích xuất các đặc trưng từ ảnh đầu vào. Kiến trúc này bao gồm:

* CSP (Cross Stage Partial): Kỹ thuật này chia luồng thông tin thành hai nhánh: một nhánh đi qua các lớp học sâu, một nhánh còn lại được giữ nguyên và sau đó hợp nhất lại. Điều này giúp giảm số lượng tham số, giảm chi phí tính toán nhưng vẫn giữ được hiệu quả học biểu diễn.
* Conv-BN-SiLU: Các lớp tích chập được kết hợp với chuẩn hóa theo lô (Batch Normalization) và hàm kích hoạt SiLU để tăng khả năng phi tuyến và tốc độ hội tụ.

Backbone thực hiện việc rút trích các đặc trưng từ cơ bản đến phức tạp, hình thành nên nền tảng cho việc tổng hợp và dự đoán ở các bước tiếp theo.

### **3.3.2. Giai đoạn 2 – Tăng cường và hợp nhất đặc trưng (Neck)**

Ở giai đoạn này, mô hình sử dụng kiến trúc PANet (Path Aggregation Network) làm phần “cổ” (neck) để kết hợp và tăng cường các đặc trưng từ các tầng khác nhau của backbone:

* FPN (Feature Pyramid Network): Hợp nhất đặc trưng từ các tầng khác nhau, từ thấp đến cao.
* PANet: Cải tiến từ FPN, bổ sung dòng thông tin từ dưới lên nhằm nâng cao khả năng định vị và phát hiện các đối tượng nhỏ.
* Biến đổi độ phân giải: Các đặc trưng ở nhiều tỉ lệ độ phân giải khác nhau (P3, P4, P5) được trích xuất, cho phép mô hình phát hiện được các đối tượng từ nhỏ đến lớn một cách hiệu quả.

Giai đoạn này đóng vai trò quan trọng trong việc tổng hợp thông tin không gian và ngữ nghĩa, giúp mô hình nhận diện đối tượng chính xác hơn.

### **3.3.3. Giai đoạn 3 – Dự đoán và phân loại (Head)**

Phần head của YOLOv8 sử dụng thiết kế anchor-free, khác biệt so với các phiên bản YOLO trước đây. Thay vì sử dụng các anchor box cố định, YOLOv8 trực tiếp dự đoán tọa độ của các đối tượng. Điều này giúp giảm độ phức tạp và tăng độ chính xác:

* Phát hiện đa cấp độ (multi-scale prediction): Dự đoán được thực hiện trên các đặc trưng P3, P4, P5 tương ứng với các tỉ lệ khác nhau, đảm bảo phát hiện được đối tượng ở nhiều kích cỡ khác nhau.
* Hộp giới hạn (bounding box): Bao gồm tọa độ trung tâm (x, y), chiều rộng và chiều cao (w, h).
* Nhãn phân loại (class label): Mỗi đối tượng được gán một xác suất tương ứng với các lớp trong tập dữ liệu.
* Confidence score: Xác suất tin cậy thể hiện mức độ chắc chắn của mô hình đối với một dự đoán.

Sau khi hoàn tất dự đoán, thuật toán Non-Maximum Suppression (NMS) sẽ được áp dụng để loại bỏ các dự đoán trùng lặp và giữ lại những hộp có độ tin cậy cao nhất.

## **3.3.** **Ưu điểm của mô hình đề xuất**

Mô hình YOLOv8 mang lại nhiều cải tiến so với các phiên bản trước và các mô hình phát hiện đối tượng khác:

* Anchor-free head: Đơn giản hóa việc huấn luyện và cải thiện độ chính xác.
* Phát hiện đối tượng nhỏ tốt hơn: Nhờ kết hợp FPN và PANet.
* Tốc độ xử lý cao: Thích hợp cho ứng dụng thời gian thực.
* Độ chính xác vượt trội: Cải thiện trên cả mAP và IoU.
* Dễ dàng tùy biến: Có các phiên bản YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x phù hợp với từng yêu cầu tài nguyên.

# **CHƯƠNG 4. THỰC NGHIỆM**

## **4.1. Môi trường thực nghiệm**

Để đảm bảo quá trình thực nghiệm diễn ra hiệu quả, dự án được triển khai trên nền tảng Google colab với cấu hình phần cứng và phần mềm như sau:

### **4.1.1. Cấu hình phần cứng**

* Nền tảng: Google Colab
* Bộ xử lý đồ họa (GPU): NVIDIA Tesla T4 (16GB VRAM)
* RAM: 12.72 GB
* CPU: 2 nhân Intel Xeon (ảo hóa)
* Bộ nhớ ổ đĩa: 100GB dung lượng tạm thời
* Kết nối Google Drive: để lưu trữ dữ liệu và kết quả huấn luyện lâu dài

### **4.1.2. Cấu hình phần mềm**

* Hệ điều hành: Môi trường mặc định của Google Colab (Ubuntu)
* Ngôn ngữ lập trình: Python 3.10
* Thư viện chính sử dụng:
* NumPy và Pandas: Xử lý dữ liệu và phân tích.
* Matplotlib/Seaborn: Trực quan hóa dữ liệu.
* Scikit-learn: Cung cấp các công cụ bổ trợ như chia tách dữ liệu, tính toán các chỉ số đánh giá mô hình.
* Ultralytics YOLOv8

### **4.1.3. Thiết lập môi trường**

Toàn bộ các thí nghiệm được thực hiện trên nền tảng Kaggle, tận dụng các môi trường cài đặt sẵn, giúp tiết kiệm thời gian cài đặt và tối ưu hóa tài nguyên.

Các thiết lập khác bao gồm:

* Bước 1: Kết nối Google Drive để lưu và truy cập dữ liệu.
* Bước 2: Cài đặt thư viện YOLOv8.
* Bước 3: Tải dữ liệu từ Google Drive.
* Bước 4: Cấu trúc thư mục

### **4.1.4. Lý do chọn Kaggle**

Google colab được chọn làm môi trường thực nghiệm vì nhiều lý do vượt trội:

1. Miễn phí GPU và TPU: Google Colab cung cấp quyền truy cập miễn phí (hoặc chi phí thấp với bản Pro) đến các phần cứng tăng tốc như GPU (NVIDIA Tesla T4, P100, V100) và TPU. Điều này giúp rút ngắn đáng kể thời gian huấn luyện các mô hình học sâu như YOLOv8 mà không cần đầu tư phần cứng đắt tiền.
2. Dễ sử dụng và truy cập mọi nơi: Colab là một nền tảng dựa trên đám mây, người dùng chỉ cần trình duyệt web và tài khoản Google là có thể lập trình, lưu trữ và chạy notebook mà không cần cài đặt môi trường cục bộ phức tạp.
3. Tích hợp tốt với Google Drive: Dễ dàng lưu trữ và tải dữ liệu, mô hình, kết quả huấn luyện trực tiếp từ/đến Google Drive. Điều này giúp đồng bộ hóa công việc và đảm bảo không mất dữ liệu trong quá trình làm việc.
4. Hỗ trợ đầy đủ thư viện và công cụ: Google Colab hỗ trợ cài đặt các thư viện máy học/phân tích dữ liệu phổ biến như: PyTorch, TensorFlow, Ultralytics, OpenCV, scikit-learn, matplotlib, v.v. Điều này giúp triển khai mô hình YOLOv8 nhanh chóng và thuận tiện.
5. Hỗ trợ chia sẻ và cộng tác: Do nền tảng Colab tương tự như Google Docs, nên dễ dàng chia sẻ với giảng viên, bạn bè, hoặc nhóm nghiên cứu để cùng chỉnh sửa, kiểm tra, và đánh giá kết quả.
6. Tiết kiệm tài nguyên và chi phí: Việc sử dụng Google Colab giúp giảm thiểu chi phí đầu tư máy tính cấu hình cao, đồng thời tránh được các lỗi tương thích khi thiết lập môi trường cục bộ.

## **4.2. Tập dữ liệu**

### **4.2.1. Nguồn dữ liệu**

Pascal VOC (Visual Object Classes) là một trong những tập dữ liệu chuẩn và phổ biến nhất trong lĩnh vực thị giác máy tính, đặc biệt là cho các bài toán như phát hiện đối tượng (Object Detection), phân đoạn ảnh (Segmentation), và phân loại (Classification). Tập VOC 2012 được công bố trong cuộc thi PASCAL VOC Challenge 2012.

### **4.2.2. Tổng quan về dữ liệu**

Nghiên cứu sử dụng bộ dữ liệu Pascal VOC 2012 có các thông tin như sau:

* Nguồn gốc: Được phát triển trong khuôn khổ cuộc thi PASCAL VOC Challenge.
* Phiên bản: 2012 là phiên bản nâng cấp từ các phiên bản trước với dữ liệu phong phú và nhãn chính xác hơn.
* Loại dữ liệu: Hình ảnh và nhãn kèm theo.
* Số lượng hình ảnh: Hơn 17.000 ảnh.
* Định dạng ảnh: JPEG.
* Chú thích: gồm các nhãn phân loại (classification labels), hộp giới hạn (bounding boxes) cho phát hiện đối tượng và ảnh phân đoạn đối tượng (segmentation masks).

### **4.2.3. Số lượng dữ liệu**

Tập dữ liệu ban đầu có tổng cộng 17.125 ảnh được phân bổ cho 20 lớp như sau:

1. pottedplant: 1202 ảnh
2. person: 17401 ảnh
3. bird: 1271 ảnh
4. chair: 3056 ảnh
5. car: 2492 ảnh
6. horse: 803 ảnh
7. bicycle: 837 ảnh
8. diningtable: 800 ảnh
9. cat: 1277 ảnh
10. sofa: 841 ảnh
11. train: 704 ảnh
12. boat: 1059 ảnh
13. motorbike: 801 ảnh
14. bottle: 1561 ảnh
15. dog: 1598 ảnh
16. tvmonitor: 893 ảnh
17. aeroplane: 1002 ảnh
18. bus: 685 ảnh
19. cow: 771 ảnh
20. sheep: 1084 ảnh

### **4.2.4. Tiền xử lý dữ liệu**

Để chuẩn bị cho các bước tiếp theo thì ta sẽ tiến hành tiền sử lý dữ liệu trước, ở đây ta sẽ mã hóa nhãn, chuyển định dạng nhãn .xml sang .txt theo định dạng YOLO. Chia dữ liệu thành tập train, val, test. Gắn nhãn theo định dạng [class\_id, x\_center, y\_center, width, height] (normalized).

### **4.2.5. Chia dữ liệu**

Tập dữ liệu sau khi được tăng cường sẽ được chia thành 3 phần chính:

* Tập huấn luyện( training set): 80% tập dữ liệu.
* Tập kiểm định(validation set): 20% tập dữ liệu.

Quá trình chia dữ liệu được thực hiện ngẫu nhiên nhưng đảm bảo duy trì tỷ lệ cân đối giữa hai nhóm ảnh trong mỗi tập.

### **4.2.6. Lý do chọn tập dữ liệu**

Tập dữ liệu từ Pascal VOC 2012 được chọn vì:

* Dữ liệu phong phú và đa dạng: Bộ dữ liệu bao gồm nhiều lớp đối tượng khác nhau, từ con người đến các loại vật thể.
* Chất lượng nhãn cao: Chú thích chi tiết và chính xác giúp cải thiện hiệu quả huấn luyện mô hình.
* Tiêu chuẩn công nghiệp: Được sử dụng rộng rãi trong các nghiên cứu và ứng dụng thị giác máy tính, phù hợp để so sánh hiệu suất với các nghiên cứu khác.
* Hỗ trợ các bài toán liên quan: Pascal VOC 2012 hỗ trợ tốt cho cả phân loại, phát hiện, và phân đoạn, đáp ứng yêu cầu của đồ án.
* Chuẩn nghiên cứu: VOC là tập dữ liệu kinh điển, được nhiều nghiên cứu sử dụng để đánh giá mô hình, do đó dễ dàng so sánh hiệu suất.
* Có sẵn nhãn bounding box: thuận tiện cho việc huấn luyện các mô hình phát hiện đối tượng như YOLO.

## **4.3. Ứng dụng thực nghiệm**

### **4.3.1. Quy trình huấn luyện mô hình**

Quy trình huấn luyện bao gồm các bước:

* Tiền xử lý từ tập ban đầu.
* Xây dựng mô hình.
* Huấn luyện mô hình trên tập dữ liệu đã được chuẩn bị.
* Đánh giá mô hình

### **4.3.2. Cấu hình huấn luyện**

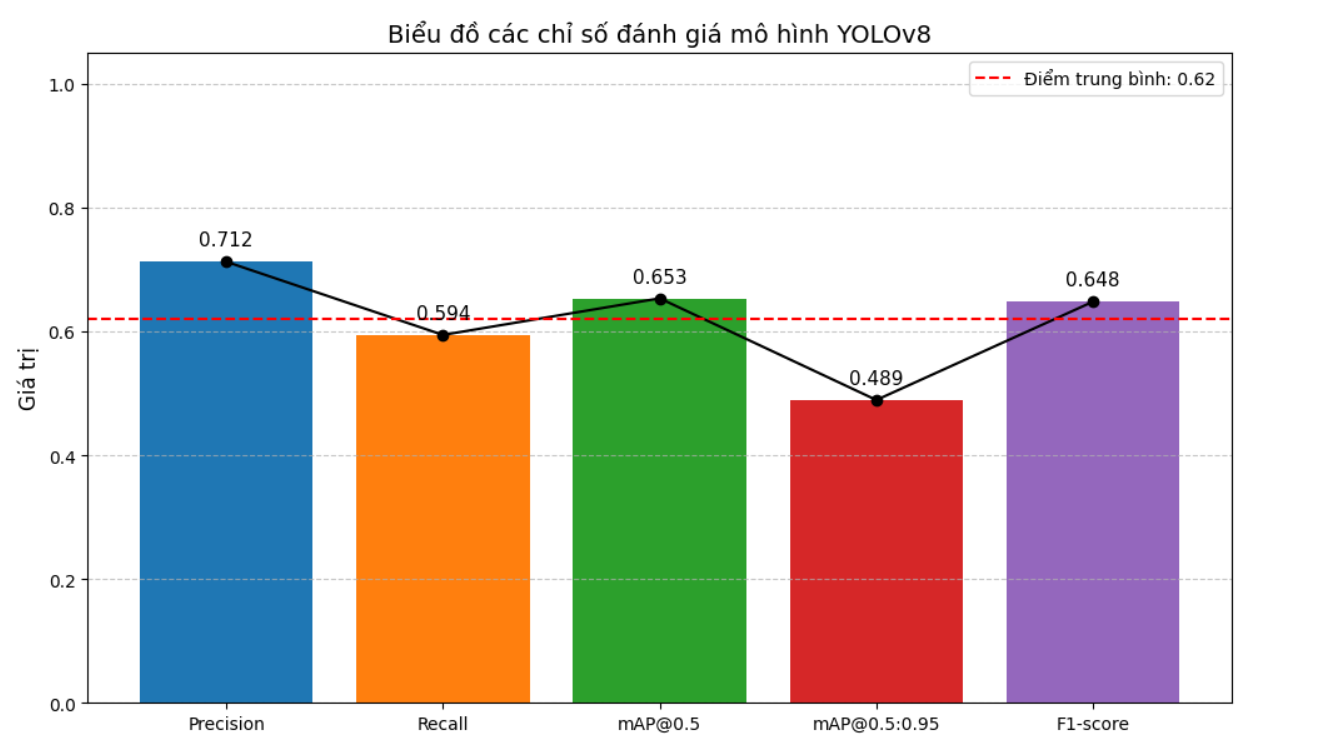
Mô hình được huấn luyện với các tham số sau:

* Epochs: 5
* Batch size: 32
* Image size: 320x320
* Optimizer: SGD (default YOLOv8)
* Early stopping: bật
* Pretrained weights: yolov8n.pt

### **4.3.3. Kết quả thực nghiệm**

Sau khi training với 5 epoch, mỗi epoch mất khoảng 3.540 - 3.780 giây với tổng thời gian đào tạo là 22254.33 giây (tương đương 6 giờ 18 phút) và thực hiện trên tập xác thực ta có được kết quả sau:

|  |  |
| --- | --- |
| **Chỉ số** | **Giá trị** |
| Độ chính xác (Precision) | 0.7122 |
| Độ phủ (Recal) | 0.5940 |
| [mAP@0.5](mailto:mAP@0.5) | 0.6531 |
| mAP@0.5:0.95 | 0.4890 |
| F1 – score | 0.6478 |

**Bảng 1:** Kết quả thực nghiệm mô hình Yolov8

**Hình 2:** Biểu đồ các chỉ số đánh giá mô hình Yolov8

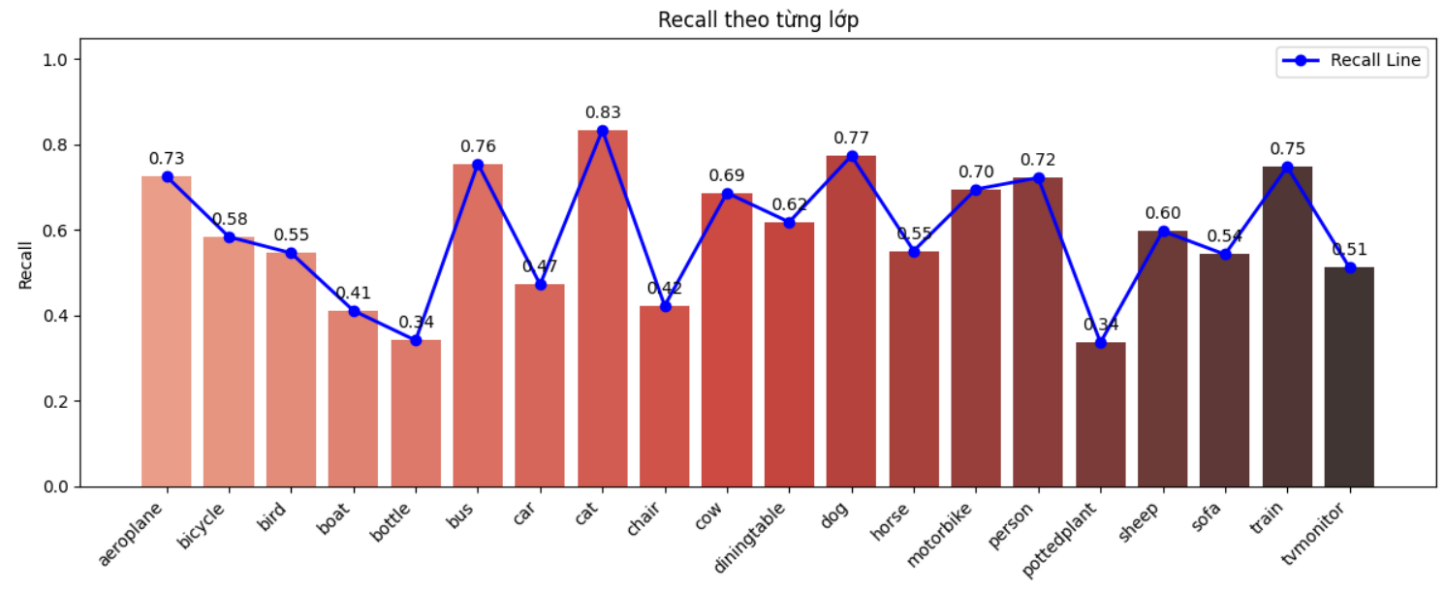
**A graph with a line pointing at the same color

AI-generated content may be incorrect.**

**A graph with a red line

AI-generated content may be incorrect.Hình 3:** Bảng so sánh các giá trị mAP

**Hình 4:** Độ chính xác (Precision) theo từng lớp đối tượng

****

**Hình 5:** Độ phủ (Recal) theo từng lớp đối tượng

A graph with green and yellow lines

AI-generated content may be incorrect.

**Hình 6:** mAP@0.5 theo từng lớp đối tượng

**HìnhA screenshot of a computer

AI-generated content may be incorrect. 7:** Ma trận nhầm lẫn

## **4.4. Đánh giá kết quả**

Sau quá trình huấn luyện mô hình phát hiện đối tượng trên tập dữ liệu Pascal VOC 2012, nhóm đã tiến hành đánh giá hiệu suất của mô hình bằng nhiều chỉ số quan trọng trong bài toán Object Detection. Các chỉ số được sử dụng gồm: ma trận nhầm lẫn (confusion matrix), độ chính xác trung bình mAP@0.5, Precision, Recall, và phân tích chi tiết theo từng lớp. Việc đánh giá giúp làm rõ khả năng học và tổng quát hóa của mô hình đối với các đối tượng khác nhau trong tập dữ liệu.

### **4.4.1 Ma trận nhầm lẫn (Confusion Matrix)**

Ma trận nhầm lẫn cung cấp cái nhìn trực quan về hiệu quả của mô hình trong việc phân loại các đối tượng. Kết quả cho thấy phần lớn các dự đoán đúng tập trung trên đường chéo chính của ma trận, điều này phản ánh rằng mô hình học được sự khác biệt giữa các lớp.

Cụ thể:

* Lớp person có số lượng mẫu lớn và mô hình nhận diện rất tốt, thể hiện qua ô tương ứng có giá trị cao.
* Lớp dog, cat, và car cũng được nhận diện chính xác với tỷ lệ nhầm lẫn thấp.
* Một số lớp như chair, pottedplant, bottle có mức độ nhầm lẫn cao hơn, thường bị mô hình nhầm với các đối tượng cùng bối cảnh như sofa hoặc diningtable.

Việc đánh giá qua ma trận nhầm lẫn còn giúp xác định những cặp lớp dễ nhầm lẫn với nhau để từ đó có thể đề xuất các kỹ thuật cải thiện mô hình trong các phần sau.

### **4.4.2 Chỉ số mAP@0.5 (mean Average Precision)**

Chỉ số mAP@0.5 là một trong những tiêu chuẩn phổ biến nhất để đánh giá mô hình phát hiện đối tượng. Đây là chỉ số đo lường độ chính xác trung bình của mô hình với ngưỡng IoU (Intersection over Union) bằng 0.5.

* Giá trị trung bình của mAP@0.5 trên toàn bộ tập test đạt 0.66, cho thấy mô hình có hiệu suất nhận diện khá tốt trong bối cảnh đa lớp.
* Một số lớp đạt mAP cao:
  + cat: 0.88
  + aeroplane: 0.81
  + car: 0.80

Những lớp này có đặc điểm hình dạng rõ ràng, dễ phân biệt, ít bị che khuất hoặc trùng lặp với các lớp khác trong ảnh.

* Một số lớp có mAP thấp:
  + bottle: 0.39
  + pottedplant: 0.40
  + chair: 0.49
* Nguyên nhân có thể đến từ:
  + Kích thước nhỏ, dễ bị mất mát thông tin trong quá trình downsample.
  + Vị trí khuất hoặc xuất hiện ở các góc ảnh.
  + Hình dạng và màu sắc không đặc trưng.

Kết quả mAP phản ánh sự ảnh hưởng của độ đa dạng dữ liệu, tỷ lệ xuất hiện của từng lớp, và mức độ khó trong việc phát hiện đối tượng.

### **4.4.3 Precision và Recall theo từng lớp**

Precision và Recall là hai chỉ số bổ sung giúp đánh giá chi tiết hơn khả năng phát hiện đúng và đầy đủ các đối tượng của mô hình.

* Precision: Đo tỷ lệ dự đoán đúng trên tổng số dự đoán. Precision cao nghĩa là ít dự đoán sai
  + aeroplane: 0.88
  + car: 0.86
  + train: 0.84

Điều này cho thấy mô hình rất “chắc chắn” khi đưa ra dự đoán cho những lớp này.

* Tuy nhiên, precision thấp ở một số lớp như:
  + chair: 0.53
  + cow: 0.59

Điều này cho thấy mô hình thường nhầm lẫn các đối tượng thuộc các lớp này với lớp khác.

* Recall: Đo tỷ lệ dự đoán đúng trên tổng số đối tượng thực tế. Recall cao nghĩa là ít bỏ sót đối tượng.
  + cat: 0.83
  + bus: 0.77
  + dog: 0.74

Mô hình đã nhận diện phần lớn đối tượng thuộc các lớp này trong ảnh.

* Các lớp có recall thấp bao gồm:
  + pottedplant: 0.35
  + bottle: 0.39

Nguyên nhân là do đối tượng nhỏ, nằm rải rác và ít nổi bật trong ảnh.

## **4.5. Xây dựng hệ thống bằng Tkiner**

Sau khi huấn luyện mô hình thành công để nâng cao trải nghiệm người dùng và tạo điều kiện thuận lợi trong việc nhận diện đối tượng, nhóm chúng em tiến hành xây dựng hệ thống nhận diện đối tượng hoàn chỉnh, kết hợp giữa mô hình YOLOv8 và giao diện người dùng bằng Tkinter. Giao diện này bao gồm hai phần chính: màn hình đăng nhập và màn hình chính của hệ thống.

### **4.5.1. Màn hình đăng nhập**

Giao diện đăng nhập là lớp bảo mật đầu tiên giúp hạn chế truy cập trái phép vào hệ thống. Màn hình bao gồm các thành phần sau:

* Tiêu đề: "HỆ THỐNG NHẬN DIỆN ĐỐI TƯỢNG TRONG ẢNH/WEBCAM", được căn giữa và in đậm.
* Khung đăng nhập: hiển thị hai trường thông tin:
* Tên đăng nhập
* Mật khẩu
* Nút Đăng nhập: Khi nhấn vào nút này, hệ thống sẽ kiểm tra thông tin xác thực. Nếu đúng, người dùng sẽ được chuyển đến màn hình chính của hệ thống; nếu sai, hệ thống sẽ hiển thị cảnh báo.

A screenshot of a computer

AI-generated content may be incorrect.Ý nghĩa: Việc bổ sung màn hình đăng nhập không chỉ giúp tăng tính bảo mật mà còn phân quyền sử dụng nếu hệ thống được triển khai trên quy mô lớn.

**Hình 8:** Màn hình đăng nhập hệ thống

### **4.5.2. Màn hình chính**

Sau khi đăng nhập thành công, người dùng được chuyển đến màn hình chính, nơi tất cả các chức năng chính của hệ thống được triển khai. Các thành phần chính bao gồm:

1. Khu vực điều khiển (Thanh công cụ chức năng)
   * Thanh điều khiển nằm ở đầu giao diện và bao gồm các nút lệnh giúp người dùng tương tác trực tiếp với hệ thống:
   * Chọn ảnh: Cho phép người dùng tải ảnh tĩnh từ máy tính để thực hiện phát hiện đối tượng.
   * Mở webcam / Tắt webcam: Kích hoạt hoặc ngắt nguồn cấp hình ảnh từ webcam, giúp người dùng thực hiện nhận diện đối tượng trong thời gian thực.
   * Lưu CSV: Cho phép xuất dữ liệu kết quả nhận dạng đối tượng (ảnh, tên đối tượng, xác suất, vị trí) ra file .csv để lưu trữ hoặc phân tích sau.
   * Xóa kết quả: Xóa toàn bộ dữ liệu và hình ảnh đang hiển thị, giúp làm mới giao diện để tiếp tục phiên nhận diện mới.
   * Dark Mode: Thay đổi giao diện sang chế độ nền tối, giúp giảm mỏi mắt khi sử dụng lâu và phù hợp với môi trường ánh sáng yếu.
2. Khu vực hiển thị ảnh/Webcam
   * Phần trung tâm giao diện là vùng lớn dùng để hiển thị hình ảnh đầu vào từ webcam hoặc ảnh người dùng chọn. Sau khi xử lý, ảnh sẽ được vẽ bounding box và nhãn tương ứng với từng đối tượng được mô hình YOLOv8 phát hiện.
3. Khu vực hiển thị kết quả
   * Bên dưới giao diện là bảng “Kết quả phát hiện”, hiển thị thông tin chi tiết về các đối tượng được phát hiện dưới dạng bảng gồm các cột:
   * Ảnh/Webcam: Nguồn ảnh sử dụng để phát hiện (file ảnh hay webcam).
   * Tên đối tượng: Nhãn lớp (label) mà mô hình YOLOv8 nhận dạng được (ví dụ: person, car, dog,...).
   * Xác suất: Độ tin cậy của mô hình đối với đối tượng được phát hiện (theo dạng phần trăm).
   * Vị trí: Tọa độ khung bao (bounding box) của đối tượng trong ảnh (gồm thông tin như x, y, width, height hoặc tương đương).

Tính năng nổi bật: Hệ thống có thể xử lý cả ảnh tĩnh và luồng video thời gian thực nhờ tích hợp với YOLOv8, giúp ứng dụng linh hoạt trong nhiều tình huống khác nhau.

A screenshot of a computer

AI-generated content may be incorrect.

**Hình 9:** Màn hình chính của hệ thống

## **4.6. Tổng kết**

Trong Chương 4, nhóm chúng em đã tiến hành thực nghiệm hệ thống phát hiện đối tượng trong ảnh dựa trên mô hình YOLOv8, với tập dữ liệu Pascal VOC 2012. Quá trình thực nghiệm bao gồm các bước chính: chuẩn bị và tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá hiệu suất, và trực quan hóa kết quả dự đoán.

Mô hình YOLOv8 cho kết quả khả quan với chỉ số mAP@0.5 đạt 66%, trong đó các lớp phổ biến như aeroplane, car, và cat đạt độ chính xác cao. Các ảnh kết quả cho thấy hệ thống có thể nhận diện tốt trong những tình huống có ánh sáng và bố cục đa dạng.

Bên cạnh những kết quả đạt được, quá trình thực nghiệm cũng giúp nhận diện một số hạn chế như: hiệu suất chưa đồng đều giữa các lớp, ảnh hưởng từ mất cân bằng dữ liệu, và chưa ứng dụng nhiều kỹ thuật tăng cường dữ liệu tiên tiến.

Việc xây dựng giao diện bằng Tkinter đã giúp hệ thống thân thiện và dễ sử dụng hơn đối với người dùng cuối. Giao diện cho phép người dùng linh hoạt lựa chọn giữa ảnh tĩnh và webcam, đồng thời hiển thị kết quả trực quan và rõ ràng.

Kết quả thực nghiệm là cơ sở để đánh giá khách quan hiệu quả của mô hình, đồng thời làm tiền đề cho việc đề xuất các hướng cải tiến và mở rộng trong chương tiếp theo.

# **CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **5.1. Kết luận**

Trong khuôn khổ đề tài, nhóm đã nghiên cứu và triển khai thành công hệ thống phát hiện đối tượng trong ảnh sử dụng mô hình YOLO (You Only Look Once) trên tập dữ liệu Pascal VOC 2012. Quá trình thực nghiệm đã được tiến hành đầy đủ qua các bước:

* Tiền xử lý dữ liệu: Chuẩn hóa, phân chia tập train/val/test, trực quan hóa dữ liệu gốc và bounding box.
* Huấn luyện mô hình YOLOv8 kết hợp FPN: Sử dụng mô hình YOLO hiện đại có độ chính xác cao và khả năng nhận diện thời gian thực.
* Đánh giá mô hình: Sử dụng các chỉ số đánh giá phổ biến như mAP@0.5, Precision, Recall, và Confusion Matrix. Mô hình đạt mAP@0.5 trung bình 0.66, trong đó các lớp như cat, car, aeroplane đạt kết quả rất tốt.
* Phân tích kết quả: Rút ra được những ưu nhược điểm của mô hình trong từng tình huống cụ thể, qua đó đề xuất các hướng cải tiến mô hình.

Kết quả cho thấy mô hình có khả năng nhận diện tốt các đối tượng trong ảnh với tốc độ nhanh, độ chính xác cao, phù hợp để triển khai trong các ứng dụng thực tế như giám sát giao thông, kiểm tra sản phẩm công nghiệp, và các hệ thống an ninh tự động.

Việc sử dụng mô hình YOLO mang lại lợi thế lớn nhờ kiến trúc một bước (one-stage detector), giúp cân bằng tốt giữa tốc độ và độ chính xác, đặc biệt hữu ích trong các bài toán yêu cầu xử lý thời gian thực.

## **5.2. Hạn chế**

Tuy đạt được những kết quả khả quan, hệ thống vẫn tồn tại một số hạn chế:

* Hiệu suất chưa đồng đều giữa các lớp: Một số lớp có mAP thấp do kích thước nhỏ, bị che khuất, hoặc có đặc điểm hình dạng không rõ ràng (ví dụ: bottle, pottedplant, chair).
* Chưa xử lý mất cân bằng lớp triệt để: Các lớp có số lượng mẫu ít ảnh hưởng đến quá trình huấn luyện và hiệu suất tổng thể.
* Chưa áp dụng nhiều kỹ thuật tối ưu: Như Focal Loss, Mosaic augmentation, hoặc các kiến trúc nâng cao như YOLOv8-seg, YOLOv8x.

## **5.3 Hướng phát triển**

Trong tương lai, nhóm đề xuất một số hướng phát triển nhằm cải thiện hiệu suất và khả năng ứng dụng của hệ thống:

1. Tối ưu hóa mô hình

* Thử nghiệm các backbone mạnh hơn như ResNet-101, EfficientNet, CSPDarknet.
* Áp dụng tối ưu siêu tham số tự động bằng các kỹ thuật như Grid Search hoặc Bayesian Optimization.

1. Tăng cường dữ liệu và xử lý mất cân bằng

* Áp dụng kỹ thuật Mosaic Augmentation, CutMix, Random Crop để tạo sự đa dạng.
* Sử dụng Focal Loss để giải quyết bài toán mất cân bằng lớp.
* Thêm dữ liệu huấn luyện hoặc khai thác từ các tập mở rộng như COCO, Open Images Dataset.

1. Giảm chi phí tính toán

* Tối ưu hóa mô hình để hoạt động hiệu quả trên các thiết bị phần cứng có tài nguyên hạn chế, ví dụ như thiết bị di động hoặc các máy tính tại các khu vực y tế thiếu thốn.

1. Mở rộng bài toán

* Nâng cấp mô hình sang YOLOv8-Segmentation để phát hiện đồng thời vị trí và phân vùng đối tượng.
* Áp dụng vào các bài toán thực tế như:
  + Phát hiện vi phạm giao thông.
  + Nhận diện đối tượng trong nông nghiệp (cây trồng, vật nuôi).
  + Theo dõi đối tượng trong hệ thống CCTV.

**TÀI LIỆU THAM KHẢO**

[1] Allied Market Research, “Computer Vision Market by Component, Product Type, Application, and End User: Global Opportunity Analysis and Industry Forecast, 2021–2030,” [Online]. Available: https://www.alliedmarketresearch.com/computer-vision-market

[2] 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 (CVPR), 2016, pp. 779–788.

[3] Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.