

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



NGÀNH: CÔNG NGHỆ THÔNG TIN
MÔN: NHẬP MÔN XỬ LÝ ẢNH SỐ

Báo Cáo Khóa Luận

Đề tài:

**Phát hiện và đếm số lượng đối tượng trong ảnh
(người, xe, vật thể) sử dụng YOLOv5**

Nhóm 15:

Thành viên 1: Bùi Hữu Tài - 207CT58588

Thành viên 2: Lưu Võ Phương Mai - 2374802010299

Thành viên 3: Trần Nghiêm Nhật Thiện - 2174802010938

Thành viên 4: Nguyễn Hoàng Long - 2174802010937

GVHD: Tiến Sĩ. Đỗ Hữu Quân

TP. Hồ Chí Minh – năm 2025

LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Tiến Sĩ Đỗ Hữu Quân, giảng viên môn Xử lý ảnh số, người đã tận tình giảng dạy, định hướng và hỗ trợ chúng em trong suốt quá trình học tập và thực hiện báo cáo này.

Nhờ vào kiến thức quý báu mà Thầy đã truyền đạt, cùng với những góp ý chân thành trong từng buổi học, nhóm đã có nền tảng vững chắc để nghiên cứu, tìm hiểu và hoàn thành tốt đề tài. Thầy không chỉ giúp chúng em tiếp cận các khái niệm, thuật toán xử lý ảnh một cách khoa học mà còn khơi gợi niềm đam mê nghiên cứu và khả năng ứng dụng thực tiễn.

Nhóm xin chân thành cảm ơn Thầy và mong tiếp tục nhận được sự quan tâm, hướng dẫn của Thầy trong các môn học và dự án sau này.

TÓM TẮT

Trong thời đại công nghệ 4.0, việc áp dụng trí tuệ nhân tạo vào xử lý ảnh số ngày càng trở nên phổ biến, đặc biệt là trong các bài toán nhận diện và theo dõi đối tượng. Đề tài "Phát hiện và đếm số lượng đối tượng trong ảnh (người, xe, vật thể) sử dụng YOLOv5" nhằm mục tiêu xây dựng một hệ thống có khả năng nhận diện và đếm chính xác các đối tượng trong ảnh đầu vào thông qua mô hình học sâu YOLOv5.

Báo cáo trình bày từ cơ sở lý thuyết về xử lý ảnh, các phương pháp học sâu hiện đại, đến quá trình phân tích, thiết kế và triển khai hệ thống nhận diện đối tượng. Thông qua việc huấn luyện và kiểm thử trên tập dữ liệu thực tế, hệ thống đã cho thấy hiệu quả cao trong việc phát hiện nhanh và chính xác các đối tượng với độ chính xác (precision) và tốc độ xử lý tốt. Bên cạnh đó, báo cáo cũng so sánh kết quả thu được với các phương pháp truyền thống để làm nổi bật ưu điểm của YOLOv5.

Kết quả của đề tài có thể được ứng dụng trong nhiều lĩnh vực thực tiễn như giám sát an ninh, giao thông thông minh, phân tích hành vi người dùng,... đồng thời mở ra hướng phát triển tiếp theo như cải thiện độ chính xác cho từng loại đối tượng, tối ưu hóa tốc độ xử lý và áp dụng trên thiết bị di động.

TỪ KHÓA

- Xử lý ảnh số
- Phát hiện đối tượng
- Đếm đối tượng
- YOLOv5
- Học sâu (Deep Learning)
- Trí tuệ nhân tạo (AI)
- Python
- OpenCV
- Nhận diện hình ảnh (Detection)
- Thị giác máy tính (Computer Vision)
- Ultralytics

MỤC LỤC

| | |
|---|----|
| LỜI CẢM ƠN..... | 2 |
| TÓM TẮT..... | 3 |
| TỪ KHÓA..... | 4 |
| MỤC LỤC..... | 5 |
| CHƯƠNG I: GIỚI THIỆU | 7 |
| 1.1. Lý do chọn đề tài | 7 |
| 1.2. Mục tiêu nghiên cứu..... | 8 |
| 1.3. Phạm vi và giới hạn của đề tài | 8 |
| 1.4. Phương pháp nghiên cứu..... | 10 |
| 1.5. Cấu trúc báo cáo..... | 11 |
| CHƯƠNG II: CƠ SỞ LÝ THUYẾT | 12 |
| 2.1. Tổng quan về xử lý ảnh số..... | 12 |
| 2.2. Các khái niệm và thuật toán liên quan (ví dụ: lọc ảnh, phân đoạn, phát hiện cạnh,...)..... | 13 |
| 2.3. Mô hình hoặc phương pháp dự định áp dụng (ví dụ: CNN, Haar Cascade, Otsu, etc.) | 15 |
| 2.4. Công cụ và thư viện sử dụng (Python, OpenCV, Keras, MATLAB, v.v.)..... | 16 |
| CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG | 17 |
| 3.1. Quy trình xử lý tổng thể | 17 |
| 3.2. Kiến trúc hệ thống / sơ đồ luồng xử lý..... | 18 |
| 3.3. Thiết kế các bước xử lý ảnh cụ thể | 20 |
| 3.4. Mô tả dữ liệu đầu vào / đầu ra | 20 |
| CHƯƠNG IV: CÀI ĐẶT VÀ THỰC NGHIỆM | 22 |
| 4.1. Môi trường cài đặt (công cụ, ngôn ngữ, hệ điều hành) | 22 |
| 4.2. Mô tả quá trình cài đặt chương trình..... | 22 |
| 4.3. Kết quả thực nghiệm..... | 23 |
| 4.4. Đánh giá kết quả và so sánh với phương pháp khác (nếu có) | 27 |
| CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN..... | 29 |
| 5.1. Tóm tắt kết quả đạt được (gồm có link github dự án)..... | 29 |
| Kết quả đầu ra chính | 29 |

| | |
|--|----|
| Đánh giá nhanh kết quả | 29 |
| 5.2. Hạn chế của đề tài | 30 |
| Hạn chế kỹ thuật | 30 |
| Hạn chế trong ứng dụng thực tế | 30 |
| 5.3. Hướng phát triển trong tương lai | 30 |
| Gợi ý & hướng phát triển | 31 |
| TÀI LIỆU THAM KHẢO | 32 |

CHƯƠNG I: GIỚI THIỆU

1.1. Lý do chọn đề tài

Trong bối cảnh hiện nay, việc tự động hóa và ứng dụng trí tuệ nhân tạo (AI) vào xử lý ảnh ngày càng trở nên phổ biến và cần thiết trong nhiều lĩnh vực như giám sát an ninh, giao thông, y tế, công nghiệp và thương mại. Một trong những bài toán quan trọng và thực tế trong xử lý ảnh là phát hiện và đếm số lượng đối tượng trong ảnh.

YOLOv5 (You Only Look Once version 5) ra đời vào năm **2020**, là một trong những mô hình phát hiện đối tượng tiên tiến, nổi bật với khả năng xử lý nhanh, chính xác và hiệu quả cao trên cả ảnh tĩnh và video thời gian thực. Việc áp dụng YOLOv5 vào bài toán phát hiện và đếm số lượng đối tượng không chỉ giúp nâng cao hiệu suất công việc mà còn mở rộng tiềm năng ứng dụng trong thực tế, như:

- Kiểm đếm sản phẩm trong dây chuyền sản xuất.
- Giám sát đám đông, phát hiện người trong khu vực công cộng.
- Quản lý phương tiện trong giao thông thông minh.
- Phát hiện và phân loại vật thể trong nghiên cứu khoa học.

Bên cạnh đó, các nghiên cứu và ứng dụng liên quan đến YOLOv5 tại Việt Nam vẫn còn hạn chế và chưa được khai thác tối đa. Do đó, nhóm lựa chọn đề tài này với mong muốn tìm hiểu sâu hơn về thuật toán, cách huấn luyện mô hình, cũng như khả năng áp dụng vào các tình huống thực tế tại địa phương và trong công nghiệp.

Qua đề tài này, nhóm không chỉ củng cố kiến thức về xử lý ảnh và học sâu mà còn có cơ hội thực hành trên các công nghệ mới nhất, góp phần nâng cao kỹ năng nghiên cứu, lập trình và ứng dụng AI trong đời sống.

1.2. Mục tiêu nghiên cứu

Mục tiêu của đề tài là nghiên cứu và ứng dụng mô hình YOLOv5 trong bài toán phát hiện và đếm số lượng đối tượng trong ảnh, cụ thể:

- Tìm hiểu kiến thức nền tảng về xử lý ảnh, học sâu (deep learning) và các mô hình phát hiện đối tượng.
- Nghiên cứu kiến trúc và nguyên lý hoạt động của mô hình YOLOv5.
- Tiến hành huấn luyện và đánh giá hiệu quả của mô hình YOLOv5 trên một tập dữ liệu cụ thể.
- Đề xuất quy trình ứng dụng mô hình YOLOv5 để phát hiện và đếm số lượng đối tượng trong ảnh.
- Xây dựng mô hình thử nghiệm có khả năng áp dụng vào các tình huống thực tế như giám sát an ninh, kiểm đếm sản phẩm, giao thông thông minh,...
- Đánh giá khả năng ứng dụng thực tiễn của mô hình trong điều kiện tại địa phương hoặc trong môi trường công nghiệp.

1.3. Phạm vi và giới hạn của đề tài

Phạm vi của đề tài:

- Đề tài tập trung nghiên cứu mô hình **YOLOv5** – một trong những mô hình phát hiện đối tượng tiên tiến trong lĩnh vực học sâu và thị giác máy tính.

- Ứng dụng mô hình YOLOv5 để **phát hiện và đếm số lượng đối tượng trong ảnh tĩnh**, với dữ liệu đầu vào là hình ảnh được thu thập từ các nguồn như camera giám sát, hình ảnh sản xuất, ảnh giao thông,...
- Thực hiện huấn luyện và đánh giá mô hình trên **một hoặc một vài tập dữ liệu cụ thể**, chẳng hạn như người, phương tiện giao thông hoặc sản phẩm trên dây chuyền sản xuất.
- Triển khai mô hình thử nghiệm nhằm minh họa khả năng ứng dụng vào **các bài toán thực tiễn** như giám sát, kiểm đếm hoặc phân tích hình ảnh.

Giới hạn của đề tài:

- Mô hình được áp dụng chủ yếu trên **ảnh tĩnh**, chưa mở rộng sang video thời gian thực do giới hạn về thời gian và tài nguyên xử lý.
- Độ chính xác và hiệu quả của mô hình phụ thuộc vào **chất lượng và số lượng dữ liệu huấn luyện**, trong khi dữ liệu thu thập thực tế có thể còn hạn chế hoặc chưa đa dạng.
- Đề tài chỉ tập trung vào việc **ứng dụng mô hình đã có sẵn (YOLOv5)**, chưa đi sâu vào việc cải tiến hoặc phát triển kiến trúc mới.
- Khả năng ứng dụng thực tế có thể bị ảnh hưởng bởi **hạn chế về phần cứng**, như hiệu năng GPU trong huấn luyện và triển khai mô hình.

1.4. Phương pháp nghiên cứu

Để thực hiện đề tài này, nhóm thực hiện sử dụng các phương pháp nghiên cứu như sau:

Phương pháp nghiên cứu tài liệu:

- Tìm hiểu các kiến thức liên quan đến trí tuệ nhân tạo, học sâu (deep learning), xử lý ảnh và thị giác máy tính.
- Nghiên cứu mô hình YOLOv5: cấu trúc, cách hoạt động, các phiên bản, ưu điểm và nhược điểm.
- Tham khảo các công trình nghiên cứu, bài báo khoa học, tài liệu kỹ thuật và các nguồn học thuật liên quan.

Phương pháp thực nghiệm:

- Thu thập và tiền xử lý dữ liệu hình ảnh phù hợp với bài toán đếm đối tượng.
- Tiến hành huấn luyện mô hình YOLOv5 trên tập dữ liệu đã chọn.
- Tinh chỉnh các siêu tham số để cải thiện hiệu suất của mô hình.
- Kiểm tra và đánh giá độ chính xác, tốc độ xử lý và hiệu quả phát hiện đối tượng.

Phương pháp phân tích – đánh giá:

- Đánh giá kết quả mô hình dựa trên các chỉ số như độ chính xác (precision), độ bao phủ (recall), mAP (mean Average Precision).
- So sánh kết quả thu được với các nghiên cứu hoặc mô hình khác (nếu có).
- Phân tích khả năng ứng dụng thực tế của mô hình trong các tình huống cụ thể tại địa phương hoặc doanh nghiệp.

1.5. Cấu trúc báo cáo

Báo cáo được chia thành 5 chương chính, mỗi chương đảm nhận một vai trò cụ thể trong quá trình nghiên cứu và triển khai đề tài:

- **Chương I: Giới thiệu**

Trình bày lý do chọn đề tài, mục tiêu nghiên cứu, phạm vi và giới hạn, phương pháp nghiên cứu và cấu trúc tổng thể của báo cáo.

- **Chương II: Cơ sở lý thuyết**

Cung cấp kiến thức nền tảng về xử lý ảnh số, các khái niệm và thuật toán liên quan, mô hình YOLOv5 và các công cụ, thư viện hỗ trợ quá trình phát triển hệ thống.

- **Chương III: Phân tích và thiết kế hệ thống**

Mô tả quy trình xử lý tổng thể, kiến trúc hệ thống, các bước xử lý ảnh và cách mô hình nhận diện – đếm đối tượng hoạt động.

- **Chương IV: Cài đặt và thực nghiệm**

Trình bày chi tiết quá trình cài đặt môi trường, triển khai hệ thống, thực hiện các thí nghiệm và đánh giá kết quả thực tế khi áp dụng mô hình.

- **Chương V: Kết luận và hướng phát triển**

Tổng kết các kết quả đạt được, chỉ ra các hạn chế còn tồn tại và đề xuất những hướng nghiên cứu, phát triển tiếp theo cho mô hình trong thực tế.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về xử lý ảnh số

Trước hết, một ảnh số là biểu diễn của một hình ảnh hai chiều dưới dạng một tập hợp hữu hạn các giá trị số, được gọi là các phần tử ảnh hoặc pixel. Xử lý ảnh số tập trung vào hai nhiệm vụ chính:

- Cải thiện thông tin hình ảnh để con người dễ dàng nhận diện và diễn giải.
- Xử lý dữ liệu hình ảnh để phục vụ lưu trữ, truyền tải và biểu diễn cho việc nhận thức tự động của máy móc.

Có một số tranh luận về ranh giới giữa xử lý ảnh và các lĩnh vực như phân tích ảnh và thị giác máy tính.

Quá trình liên tục từ xử lý ảnh đến thị giác máy tính có thể được chia thành các cấp độ: thấp, trung bình và cao.

Xử lý ảnh số đóng vai trò nền tảng trong nhiều lĩnh vực công nghệ hiện đại, đặc biệt là trong trí tuệ nhân tạo và thị giác máy tính.

Các nhiệm vụ chính của xử lý ảnh số gồm:

1. **Cải thiện hình ảnh (Enhancement):** Làm rõ chi tiết, tăng độ tương phản, giảm nhiễu để phục vụ cho việc quan sát bằng mắt người.
2. **Khôi phục ảnh (Restoration):** Loại bỏ lỗi, biến dạng hoặc khôi phục ảnh bị mờ, nhiễu.
3. **Xử lý và nén ảnh (Compression & Storage):** Giảm kích thước dữ liệu ảnh để lưu trữ và truyền tải hiệu quả hơn.
4. **Phân tích và nhận diện ảnh (Analysis & Recognition):** Trích xuất đặc trưng, phân vùng, nhận dạng đối tượng nhằm phục vụ cho các hệ thống tự động như trí tuệ nhân tạo, thị giác máy tính.
5. **Biến đổi và biểu diễn ảnh:** Chuyển đổi ảnh từ không gian này sang không gian khác để dễ dàng xử lý, ví dụ: từ không gian RGB sang Grayscale.

2.2. Các khái niệm và thuật toán liên quan (ví dụ: lọc ảnh, phân đoạn, phát hiện cạnh,...)

Tăng cường ảnh miền không gian (Spatial domain image enhancement)

Tăng cường ảnh miền không gian là quá trình cải thiện chất lượng hình ảnh bằng cách điều chỉnh các giá trị pixel trong không gian hình ảnh. Các phương pháp này thường tập trung vào việc làm nổi bật các đặc điểm quan trọng trong ảnh, như độ tương phản và độ sáng.

Xử lý histogram (Histogram processing)

Xử lý histogram là một kỹ thuật tăng cường ảnh bằng cách điều chỉnh phân phối cường độ pixel. Các phương pháp phổ biến bao gồm:

- Cân bằng histogram: Tăng cường độ tương phản bằng cách phân phối lại các giá trị pixel.
- Cắt histogram: Giới hạn các giá trị pixel trong một khoảng nhất định để làm nổi bật các chi tiết trong ảnh.

Xử lý điểm (Point processing)

Xử lý điểm là kỹ thuật điều chỉnh giá trị của từng pixel độc lập. Một số phương pháp bao gồm:

- Biến đổi logarit: Tăng cường các chi tiết trong vùng tối của ảnh.
- Biến đổi lũy thừa: Tăng cường các chi tiết trong vùng sáng của ảnh.

Tiền xử lý ảnh và tăng cường ảnh miền tần số (Image pre-processing and frequency domain image enhancement)

Tiền xử lý ảnh là bước quan trọng trong xử lý ảnh, giúp cải thiện chất lượng ảnh trước khi áp dụng các thuật toán phân tích. Tăng cường ảnh miền tần số sử dụng các biến đổi Fourier để xử lý ảnh trong miền tần số, cho phép loại bỏ nhiễu và cải thiện độ rõ nét.

Lọc nhiễu (Noise filtering)

Lọc nhiễu là quá trình loại bỏ các tín hiệu không mong muốn (nhiễu) khỏi ảnh. Một số phương pháp lọc phổ biến bao gồm:

- Lọc trung bình: Làm mịn ảnh bằng cách thay thế giá trị pixel bằng giá trị trung bình của các pixel lân cận.
- Lọc Gaussian: Sử dụng hàm Gaussian để làm mịn ảnh, giúp giảm nhiễu mà vẫn giữ lại các chi tiết quan trọng.
- Lọc trung vị: Thay thế giá trị pixel bằng giá trị trung vị của các pixel lân cận, hiệu quả trong việc loại bỏ nhiễu muối và tiêu.
- Lọc Wiener: Một phương pháp lọc thích ứng, điều chỉnh theo mức độ nhiễu trong ảnh.

Tăng cường ảnh miền tần số (Frequency domain image enhancement)

Tăng cường ảnh trong miền tần số sử dụng các biến đổi Fourier để phân tích và điều chỉnh các thành phần tần số của ảnh. Các phương pháp bao gồm:

- Bộ lọc tần số thấp: Giúp làm mịn ảnh bằng cách loại bỏ các tần số cao.
- Bộ lọc tần số cao: Giúp làm nổi bật các chi tiết và cạnh trong ảnh bằng cách loại bỏ các tần số thấp.

Khôi phục và phân đoạn ảnh số (Digital image restoration and segmentation)

Khôi phục ảnh số là quá trình cải thiện chất lượng ảnh bị hư hại hoặc nhiễu. Phân đoạn ảnh là bước quan trọng trong việc tách biệt các đối tượng trong ảnh.

- Khôi phục ảnh số (Digital image restoration): Sử dụng các phương pháp như lọc Wiener hoặc lọc tối ưu để khôi phục ảnh từ các tín hiệu bị nhiễu.

- Phân đoạn ảnh theo Điểm - Đường - Góc (Image segmentation by Point - Line - Corner): Phân đoạn dựa trên các đặc điểm hình học của ảnh, giúp xác định các đối tượng trong ảnh.
- Phân đoạn ảnh theo ngưỡng (Image segmentation by thresholding): Sử dụng ngưỡng để tách biệt các đối tượng trong ảnh. Phương pháp này có thể là đơn giản (ngưỡng đơn) hoặc phức tạp hơn (ngưỡng đa)

2.3. Mô hình hoặc phương pháp dự định áp dụng (ví dụ: CNN, Haar Cascade, Otsu, etc.)

Trong đề tài này, nhóm lựa chọn áp dụng mô hình *YOLOv5* (*You Only Look Once version 5*) – một trong những kiến trúc mạng nơ-ron tiên tiến và phổ biến nhất hiện nay trong lĩnh vực phát hiện đối tượng thời gian thực. Mô hình này thuộc họ *CNN* (*Convolutional Neural Network*) và hoạt động theo nguyên tắc phát hiện đối tượng chỉ qua một lần quét ảnh (one-stage detector), giúp tăng tốc độ xử lý đáng kể.

Các phương pháp và khái niệm chính được áp dụng trong đề tài gồm:

- **CNN (Mạng nơ-ron tích chập)**: Là nền tảng của YOLOv5, giúp trích xuất đặc trưng ảnh ở nhiều cấp độ khác nhau.
- **YOLOv5**: Mô hình phát hiện đối tượng thời gian thực có độ chính xác cao và tốc độ xử lý nhanh. YOLOv5 phân chia ảnh đầu vào thành các lưới và dự đoán các bounding boxes cùng xác suất của đối tượng trong từng lưới.
- **Transfer Learning (Học chuyển giao)**: Sử dụng trọng số từ mô hình YOLOv5 đã được huấn luyện sẵn trên tập dữ liệu COCO để tinh chỉnh với tập dữ liệu cụ thể của đề tài, giúp tiết kiệm thời gian huấn luyện và cải thiện độ chính xác.
- **Anchor Boxes**: Các hộp giới hạn định sẵn được sử dụng trong YOLOv5 để hỗ trợ mô hình xác định kích thước và vị trí đối tượng chính xác hơn.
- **NMS (Non-Maximum Suppression)**: Kỹ thuật loại bỏ các dự đoán trùng lặp để giữ lại khung giới hạn có xác suất cao nhất cho mỗi đối tượng.

2.4. Công cụ và thư viện sử dụng (Python, OpenCV, Keras, MATLAB, v.v.)

Trong quá trình xây dựng và triển khai hệ thống phát hiện và đếm số lượng đối tượng trong ảnh bằng YOLOv5, nhóm đã sử dụng các công cụ và thư viện sau:

- **Python:** Là ngôn ngữ lập trình chính được sử dụng để xây dựng toàn bộ hệ thống nhờ vào tính linh hoạt, dễ đọc, dễ bảo trì và có hệ sinh thái mạnh mẽ trong lĩnh vực học máy và xử lý ảnh.
- **YOLOv5:** Mô hình nhận diện đối tượng mạnh mẽ, được viết bằng PyTorch, hỗ trợ huấn luyện và phát hiện đối tượng hiệu quả trong thời gian thực.
- **PyTorch:** Framework học sâu mã nguồn mở, hỗ trợ xây dựng, huấn luyện và kiểm thử mô hình YOLOv5.
- **OpenCV:** Thư viện xử lý ảnh mã nguồn mở, hỗ trợ các thao tác như đọc ảnh, chuyển đổi định dạng, vẽ khung phát hiện, hiển thị kết quả,...
- **Pandas và NumPy:** Dùng để xử lý dữ liệu đầu vào, phân tích dữ liệu, xử lý mảng và bảng dữ liệu thuận tiện hơn.
- **Matplotlib / Seaborn:** Dùng để trực quan hóa kết quả, biểu đồ phân tích dữ liệu và biểu diễn quá trình huấn luyện mô hình.
- **Google Colab / Jupyter Notebook:** Môi trường lập trình trực tuyến hỗ trợ GPU, thuận tiện cho việc huấn luyện và thử nghiệm mô hình.
- **LabelImg:** Công cụ dùng để gán nhãn (label) dữ liệu huấn luyện theo định dạng YOLO.

CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Quy trình xử lý tổng thể

Quy trình xử lý tổng thể trong đề tài được thiết kế theo chuỗi các bước chính nhằm đảm bảo khả năng nhận diện và đếm số lượng đối tượng trong ảnh một cách hiệu quả, gồm các bước như sau:

3.1.1 Thu thập và chuẩn bị dữ liệu

- Thu thập ảnh từ nhiều nguồn như camera giám sát, hình ảnh giao thông, dây chuyền sản xuất,...
- Tiền xử lý ảnh: chuyển đổi kích thước, định dạng, chuẩn hóa ảnh.
- Gán nhãn đối tượng (labeling) sử dụng công cụ như LabelImg, theo định dạng YOLO.

3.1.2 Huấn luyện mô hình YOLOv5

- Chọn phiên bản YOLOv5 phù hợp (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) tùy theo yêu cầu về tốc độ và độ chính xác.
- Cấu hình tệp huấn luyện (train.yaml, data.yaml).
- Tiến hành huấn luyện trên tập dữ liệu đã chuẩn bị.
- Tinh chỉnh siêu tham số (learning rate, batch size, epochs,...) để tối ưu hiệu suất mô hình.

3.1.3 Dự đoán và phát hiện đối tượng

- Đưa ảnh đầu vào vào hệ thống.
- YOLOv5 thực hiện phát hiện các đối tượng, vẽ bounding box quanh từng đối tượng và phân loại (người, xe, vật thể,...).
- Lưu kết quả bao gồm hình ảnh đã nhận diện và thông tin số lượng từng loại đối tượng.

3.1.4 Đếm số lượng đối tượng

- Dựa vào kết quả phát hiện, hệ thống đếm số lượng mỗi loại đối tượng (sử dụng nhãn lớp của YOLOv5).
- Hiển thị kết quả trên giao diện hoặc lưu dưới dạng file CSV/JSON để phục vụ phân tích.

3.1.5 Đánh giá và cải tiến mô hình

- Đánh giá mô hình bằng các chỉ số như: Precision, Recall, mAP.
- So sánh với kết quả từ các mô hình truyền thống (nếu có).
- Cải thiện mô hình bằng cách tăng dữ liệu huấn luyện, điều chỉnh siêu tham số hoặc thử các phiên bản YOLO khác.

3.2. Kiến trúc hệ thống / sơ đồ luồng xử lý

| <i>Thành phần</i> | <i>Vai trò chính</i> |
|-------------------|---|
| Backbone | Trích xuất đặc trưng từ ảnh |
| Neck | Tổng hợp đặc trưng từ nhiều cấp độ |
| Head | Dự đoán tọa độ box và phân loại đối tượng |
| Post-processing | Lọc kết quả và loại bỏ box dư thừa (NMS) |

[Ảnh đầu vào]



[Tiền xử lý ảnh]

(resize, normalize)



[Backbone: CSPDarknet]

(trích xuất đặc trưng)



[Neck: PANet]

(tổng hợp đa cấp độ)



[Head]

(dự đoán box + class)

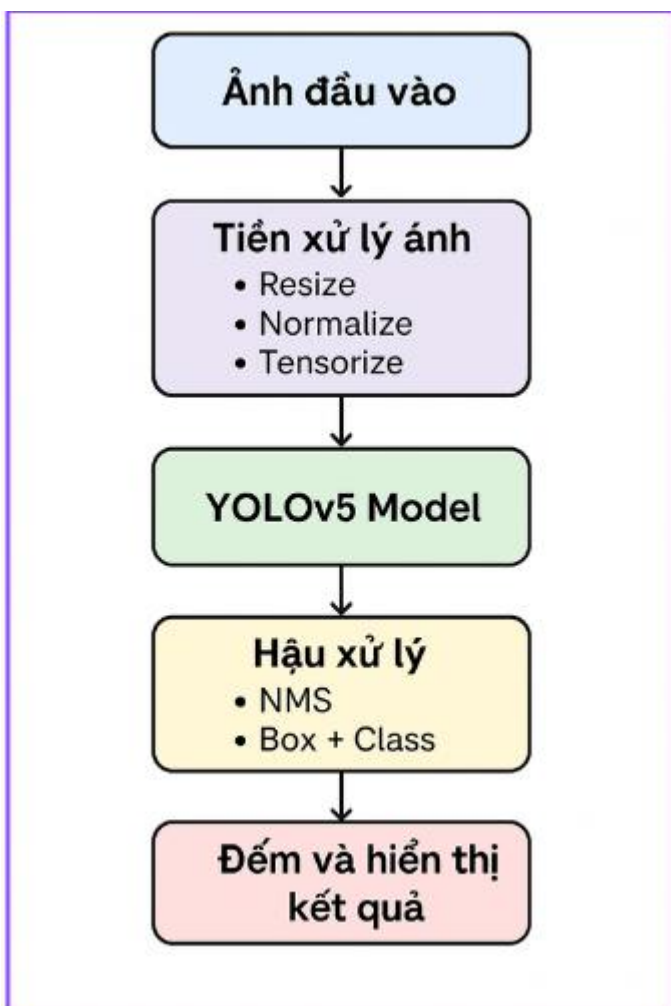


[Hậu xử lý (NMS)]



[Kết quả: Bounding Boxes + Nhãn + Xác suất]

3.3. Thiết kế các bước xử lý ảnh cụ thể



3.4. Mô tả dữ liệu đầu vào / đầu ra

Tham số **--source** trong YOLOv5 nhận nhiều loại dữ liệu đầu vào khác nhau:

| Loại dữ liệu | Mô tả | Ví dụ |
|--------------|---------------------------------|---|
| Hình ảnh | File ảnh đơn lẻ (jpg, png, ...) | <code>--source path/to/image.jpg</code> |
| Thư mục ảnh | Thư mục chứa nhiều ảnh | <code>--source path/to/images/</code> |

| | | |
|-----------------------|-------------------------------------|---|
| Video | File video MP4, AVI,... | <code>--source path/to/video.mp4</code> |
| Webcam | Số index camera (0, 1, 2,...) | <code>--source 0</code> |
| Stream URL | Đường dẫn video stream RTSP/HTTP | <code>--source rtsp://...</code> |

CHƯƠNG IV: CÀI ĐẶT VÀ THỰC NGHIỆM

4.1. Môi trường cài đặt (công cụ, ngôn ngữ, hệ điều hành)

Để có thể “**Phát hiện và đếm số lượng đối tượng trong ảnh (người, xe, vật thể) sử dụng YOLOv5**” trước hết ta phải cài đặt các công cụ(tools) cũng như ngôn ngữ và hệ điều hành phù hợp với nhu cầu làm việc, YOLOv5 cung cấp

- **YOLOv5** buộc bạn phải cài đặt môi trường Python phiên bản $\geq 3.8.0$ và **PyTorch** phiên bản ≥ 1.8
- “**Ultralytics cung cấp nhiều môi trường dựng sẵn**, đã được cài đặt sẵn các phần mềm và thư viện cần thiết như **CUDA, CuDNN, Python và PyTorch** để bạn có thể bắt đầu dự án ngay lập tức mà không cần tự cấu hình phức tạp.” - trích từ Docs của ultralytics/yolov5
- **YOLOv5 là một ứng dụng cụ thể được xây dựng dựa trên PyTorch**. Nếu không có PyTorch, YOLOv5 sẽ không chạy được.
- Bạn cũng có thể tải **Git/Anaconda** để làm một môi trường phân nhánh chứa detection project của bạn

4.2. Mô tả quá trình cài đặt chương trình

Trước hết, bạn sẽ cài sẵn Python ver 3.8 trở lên và thực hiện các bước sau

- Mở môi trường làm việc (ở đây nhóm mình sử dụng **Visual Studio Code** và **Git**) và tạo một thư mục cho project, sau đó bật terminal lên và gõ lệnh như hình sau

=> Giải thích cho câu lệnh trên nằm na là chúng ta sẽ tạo một *clone*(bản sao) sao chép mã nguồn của **YOLOv5** về và *cd*(di chuyển) đến thư mục sau khi tạo, cài đặt các *dependencies*(thành phần/thư viện/công cụ) để **YOLOv5** có thể hoạt động

```
git clone https://github.com/ultralytics/yolov5 # clone repository
cd yolov5
pip install -r requirements.txt # install dependencies
```

- Sau khi cài đặt bước trên xong, bạn có thể kiểm tra xem YOLOv5 đã hoạt động được chưa bằng câu lệnh mẫu sau

```
import torch

# Model loading
model = torch.hub.load("ultralytics/yolov5", "yolov5s") # Can be 'yolov5n' - 'yolov5x6', or 'custom'

# Inference on images
img = "https://ultralytics.com/images/zidane.jpg" # Can be a file, Path, PIL, OpenCV, numpy, or list of images

# Run inference
results = model(img)

# Display results
results.show()
results.print() # Other options: .show(), .save(), .crop(), .pandas(), etc. Explore these in the Predict mode documentation.
```

=> Giải thích : ở dòng **img** = ‘ ‘ bạn có thể thay đổi tùy theo hình bạn muốn detect. ‘**results.show()**’ in ra hình kết quả sau khi detect , còn có nhiều câu lệnh khác như trong hình (hoặc có thể tham khảo ở mục Predict mode trên docs Ultralytics)

4.3. Kết quả thực nghiệm

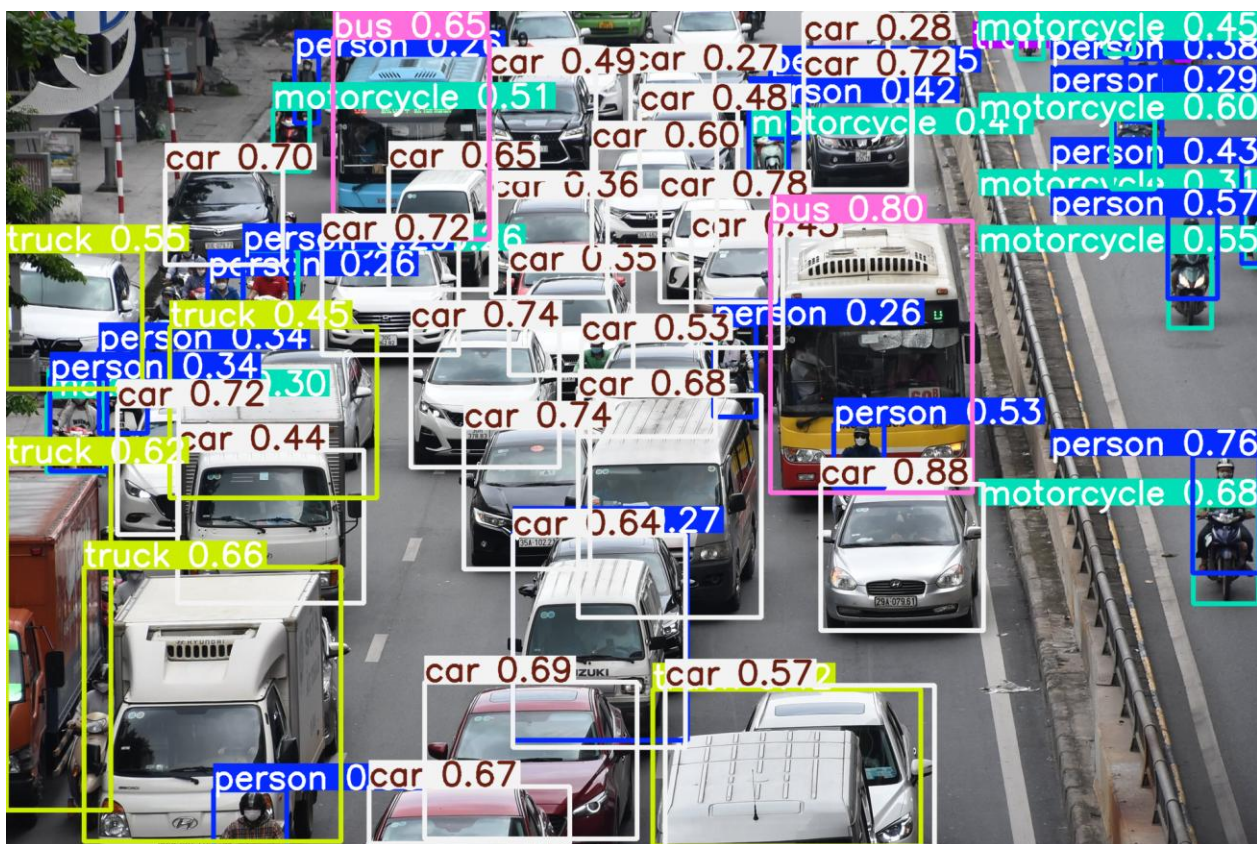
```
Bin@DESKTOP-5760EEA MINGW64 ~/yolov5 (master)
$ python detect.py --weights yolov5x.pt --source ~/Desktop/hoc/Xulyanh/project/xel.jpg --view-img
```

Input IMG:



Output IMG:

image 1/1: 1283x1920 16 persons, 24 cars, 10 motorcycles, 2 buss, 5 trucks, 1 traffic light



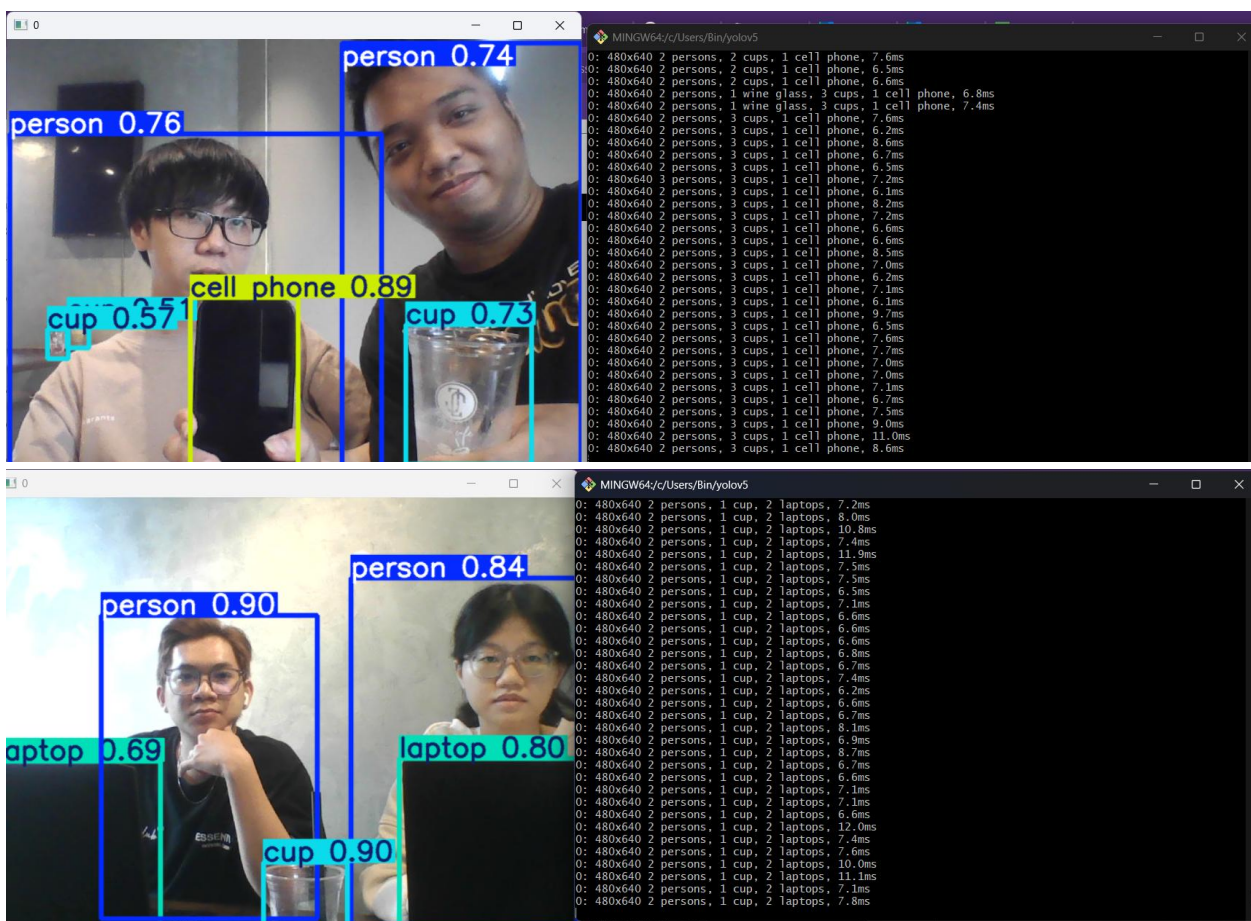
Detect file mp4:



Detect camer



a:Output:2 examples



4.4. Đánh giá kết quả và so sánh với phương pháp khác (nếu có)

Ưu điểm YOLOv5:

Tốc độ suy luận nhanh, khả năng chạy trên nhiều thiết bị.

Dễ cài đặt, thân thiện với sinh viên.

Có phiên bản hỗ trợ phân vùng (YOLOv5-seg)

So sánh các phương pháp khác :

| Tiêu chí | YOLOv5 | Faster R-CNN | Mask R-CNN |
|------------------------------|-------------------|---------------------|-----------------------|
| Tốc độ | Rất nhanh | Chậm hơn | Trung bình |
| Dễ triển khai | Rất dễ | Cần cấu hình nhiều | Cần huấn luyện kỹ hơn |
| Khả năng segmentation | Có qua YOLOv5-seg | Không có | Có – mạnh mẽ |

CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Tóm tắt kết quả đạt được (gồm có link github dự án)

Kết quả đầu ra chính

| Kết quả | Mô tả |
|-------------------------|--|
| Bounding Box | Vị trí bao quanh các đối tượng được nhận diện trong ảnh |
| Nhãn (Label) | Tên lớp đối tượng (ví dụ: "person", "car", "dog"...) |
| Độ tin cậy (Confidence) | Mức độ chắc chắn của mô hình về mỗi dự đoán (0.00 → 1.00) |
| Ảnh kết quả | Ảnh được vẽ box + nhãn, lưu tại thư mục runs/detect/exp/ |
| File TXT (tùy chọn) | Tọa độ box, nhãn, độ tin cậy → xuất ra file .txt để xử lý tiếp |

Đánh giá nhanh kết quả

- **Tốc độ xử lý:** Nhanh, trung bình vài chục FPS trên GPU.
- **Độ chính xác:** Với mô hình YOLOv5x, có thể đạt mAP > 75% trên COCO dataset.

- **Tổ chức kết quả:** Mỗi ảnh đầu vào được lưu thành một ảnh đầu ra kèm box và nhãn.
- **Hiển thị trực tiếp:** Nếu dùng `--view-img`, ảnh kết quả sẽ hiện lên ngay sau khi xử lý.

5.2. Hạn chế của đề tài

Hạn chế kỹ thuật

- **Phụ thuộc vào dữ liệu huấn luyện:** Nếu dữ liệu không đủ đa dạng hoặc không được annotate kỹ, kết quả detect sẽ sai lệch.
- **Khó nhận diện đối tượng nhỏ:** Với ảnh có nhiều chi tiết hoặc đối tượng quá nhỏ, YOLOv5 thường bỏ sót hoặc nhận diện kém.
- **Không xử lý tốt các vùng bị che khuất:** Khi đối tượng bị che một phần, mô hình dễ nhầm lẫn hoặc bỏ qua.
- **Giới hạn phân loại:** YOLOv5 chỉ nhận diện các lớp đã được học; không thể phát hiện những lớp mới không có trong bộ dữ liệu gốc.
- **Yêu cầu tài nguyên tính toán:** Các phiên bản lớn như YOLOv5x cần GPU mạnh để chạy thời gian thực.

Hạn chế trong ứng dụng thực tế

- **Không tối ưu cho video dài:** Nếu áp dụng vào video liên tục, cần thêm các kỹ thuật tracking để duy trì thông tin đối tượng qua nhiều khung hình.
- **Khó tích hợp vào thiết bị IoT thấp cấp:** Các thiết bị như Raspberry Pi hoặc ESP32 khó chạy YOLOv5 hiệu quả mà không tối ưu lại mô hình.
- **Không có khả năng phân đoạn (segmentation):** YOLOv5 chỉ vẽ hộp bao quanh, không thể tạo mặt nạ phân đoạn như các mô hình như Mask R-CNN.

5.3. Hướng phát triển trong tương lai

Hướng phát triển

Trong tương lai, nhóm định hướng phát triển đề tài theo các hướng sau:

1. **Tối ưu hóa mô hình:** Nghiên cứu và áp dụng các phương pháp tối ưu để giảm thiểu kích thước mô hình, tăng tốc độ xử lý nhưng vẫn đảm bảo độ chính xác cao.
2. **Phát hiện trong môi trường phức tạp:** Mở rộng khả năng phát hiện và đếm đối tượng trong các điều kiện ánh sáng yếu, nhiều vật thể bị che khuất hoặc các cảnh có độ phức tạp cao.
3. **Ứng dụng thực tế:** Xây dựng các ứng dụng triển khai YOLOv5 vào hệ thống giám sát giao thông, kiểm kê hàng hóa, kiểm soát an ninh hoặc hỗ trợ nông nghiệp thông minh.
4. **Kết hợp với các công nghệ khác:** Kết hợp YOLOv5 với các công nghệ khác như phân tích video thời gian thực, AIoT (AI + IoT) để phát triển các hệ thống thông minh tự động hóa toàn diện.
5. **Huấn luyện mô hình với dữ liệu bản địa:** Thu thập và huấn luyện mô hình trên các tập dữ liệu phù hợp với môi trường Việt Nam nhằm nâng cao độ chính xác khi áp dụng trong nước.

Thông qua những hướng phát triển trên, đề tài không chỉ dừng lại ở mức độ nghiên cứu học thuật mà còn có tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực thực tiễn.

Gợi ý & hướng phát triển

- Kết hợp YOLOv5 với Deep SORT hoặc ByteTrack để cải thiện theo dõi đối tượng qua nhiều khung hình.

- Chuyển đổi mô hình sang định dạng nhẹ hơn (ONNX, TensorRT) để triển khai trên thiết bị hạn chế.
- Nâng cấp mô hình sang YOLOv8 hoặc dùng kết hợp với SAM (Segment Anything Model) nếu cần phân đoạn đối tượng.

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo thêm :

- [All about YOLOv5 Docs](<https://docs.ultralytics.com/vi/yolov5/>)
- [Github Ultralytics YOLOv5](<https://github.com/ultralytics/yolov5>)
- [Youtube using YOLOv5 for train custom datasets ver VIE](<https://www.youtube.com/watch?v=eSS0EnCX1A0>)
- [Train YOLOv5 custom dataset](<https://viblo.asia/p/su-dung-colab-train-yolov5-voi-custom-dataset-phat-hien-cac-doi-tuong-dac-thu-Az45bqv6lxY>)
- [Google](<https://www.google.com/>)
- Slide bài giảng Nhập môn Xử lý ảnh số - Văn Lang University