

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á

KHOA: CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH

Đề tài số 11: Xây dựng hệ thống phát hiện và nhận diện đối tượng trong video trong thời gian thực

Giảng viên hướng dẫn: Lương Thị Hồng Lan

STT	Mã sinh viên	Sinh viên thực hiện	Lớp
1	20210416	Nguyễn Huy Chiến	DCCNTT12.10.2
2	20210379	Trần Tùng Dương	DCCNTT12.10.2
3	20211910	Lê Đức Anh	DCCNTT12.10.2
4	20210358	Đỗ Thị Vân	DCCNTT12.10.2

Bắc Ninh, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á

KHOA: CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH

Đề tài số 11: Xây dựng hệ thống phát hiện và nhận diện đối tượng trong video trong thời gian thực

Giảng viên hướng dẫn: Lương Thị Hồng Lan

STT	Mã sinh viên	Sinh viên thực hiện	Lớp
1	20210416	Nguyễn Huy Chiến	DCCNTT12.10.2
2	20210379	Trần Tùng Dương	DCCNTT12.10.2
3	20211910	Lê Đức Anh	DCCNTT12.10.2
4	20210358	Đỗ Thị Vân	DCCNTT12.10.2

Bắc Ninh, năm 2024

PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN

Mã đề thi: 11

Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH

Lớp tín chỉ: XATGMT.03.K12.02.LH.C04.1_LT

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Huy Chiến	Trần Tùng Dương	Lê Đức Anh	Đỗ Thị Vân
			20210416	20210379	20211910	20210358
1	Nội dung báo cáo trên Word đầy đủ	3.5				
1.1	Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận).	0,5				
1.2	Nội dung phân tích rõ ràng, logic.	0,5				
1.3	Có dẫn chứng, số liệu minh họa đầy đủ.	0,5				
1.4	Ngôn ngữ và trình bày chuẩn, không lỗi chính tả.	0,5				
1.5	Có trích dẫn tài liệu tham khảo đúng quy cách.	0,5				
1.6	Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý).	0,5				

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Huy Chiến	Trần Tùng Dương	Lê Đức Anh	Đỗ Thị Vân
			20210416	20210379	20211910	20210358
1.7	Tài liệu đầy đủ, bám sát yêu cầu của đề bài.	0,5				
2	Nội dung thuyết trình đầy đủ	1.0				
2.1	Trình bày tự tin, phát âm rõ ràng, mạch lạc.	0,5				
2.2	Nội dung thuyết trình đúng trọng tâm, không lan man.	0,5				
3	Slides báo cáo đầy đủ nội dung + Hỏi đáp	3.0				
3.1	Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận).	0,5				
3.2	Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa).	0,5				
3.3	Nội dung trên slides ngắn gọn, dễ hiểu, súc tích.	0,5				
3.4	Nội dung slides phù hợp với nội dung báo cáo.	0,5				
3.5	Trả lời câu hỏi đầy đủ, chính xác.	0,5				
3.6	Trả lời câu hỏi tự tin, thuyết phục.	0,5				

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Huy Chiến	Trần Tùng Dương	Lê Đức Anh	Đỗ Thị Vân
			20210416	20210379	20211910	20210358
4	Code đầy đủ	2,5				
1.1	Code được trình bày rõ ràng, có chú thích đầy đủ.	0,5				
1.2	Code chạy đúng, không lỗi.	0,5				
1.3	Code tối ưu, không dư thừa.	0,5				
1.4	Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài.	0,5				
1.5	Có tính sáng tạo hoặc cải thiện so với yêu cầu.	0,5				
TỔNG ĐIỂM BẢNG SỐ:		10				
TỔNG ĐIỂM BẢNG CHỮ:		Mười tròn				

Cán bộ chấm thi 1

(Ký và ghi rõ họ tên)

Lương Thị Hồng Lan

Cán bộ chấm thi 2

(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ ngày càng phát triển, các ứng dụng của Xử lý ảnh và Thị giác máy tính đang dần trở nên phổ biến trong nhiều lĩnh vực như an ninh, giám sát, y tế, giao thông và giải trí. Với mục tiêu cung cấp các giải pháp thông minh và tự động hóa, việc xây dựng hệ thống theo dõi đối tượng trong video đã và đang thu hút sự quan tâm lớn từ cộng đồng nghiên cứu cũng như ứng dụng thực tế.

Bài tập lớn này được thực hiện nhằm vận dụng các kiến thức lý thuyết và kỹ thuật đã học trong môn Xử lý ảnh và Thị giác máy tính để xây dựng một hệ thống theo dõi đối tượng trong video. Đề tài này không chỉ giúp sinh viên nắm vững các phương pháp cơ bản như xử lý khung hình, phát hiện đối tượng, và theo dõi chuyển động mà còn mở ra cơ hội tiếp cận với các thuật toán hiện đại như YOLO, CNN, hoặc DeepSORT.

Hệ thống được xây dựng sẽ bao gồm các tính năng như: phát hiện đối tượng trong các khung hình video, theo dõi các đối tượng qua các khung hình liên tiếp, và trực quan hóa kết quả theo dõi trên giao diện hiển thị. Các công cụ và ngôn ngữ lập trình như Python, OpenCV, TensorFlow sẽ được sử dụng để triển khai hệ thống.

Với đề tài "Xây dựng hệ thống theo dõi đối tượng trong video", nhóm chúng em hy vọng không chỉ hoàn thành mục tiêu của bài tập lớn mà còn tích lũy được các kỹ năng và kinh nghiệm thực tế để áp dụng trong các dự án thực tế sau này.

LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến Bộ môn Xử lý ảnh và Thị giác máy tính, Khoa Công nghệ Thông tin, Trường Đại học Công Nghệ Đông Á, đã tạo điều kiện và cung cấp nền tảng kiến thức vững chắc để chúng em thực hiện bài tập lớn này.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc tới cô Lương Thị Hồng Lan vì đã tận tình giảng dạy, định hướng, và hỗ trợ chúng em trong suốt quá trình học tập và triển khai đề tài. Sự hướng dẫn tận tình của cô không chỉ giúp nhóm hiểu sâu hơn về lý thuyết mà còn ứng dụng hiệu quả vào bài tập thực tế.

Chúng em cũng xin gửi lời cảm ơn đến các thành viên trong nhóm, vì sự phối hợp nhịp nhàng và tinh thần làm việc trách nhiệm đã góp phần giúp hoàn thành bài tập lớn này.

Mặc dù đã nỗ lực hết sức, nhưng bài tập lớn không thể tránh khỏi những thiếu sót. Chúng em mong nhận được sự góp ý từ thầy cô và các bạn để có thể hoàn thiện hơn trong các dự án tương lai.

Trân trọng cảm ơn!

LỜI CAM ĐOAN

Chúng em, nhóm tác giả của bài tập lớn này, cam đoan rằng mọi nội dung được trình bày trong bài viết là hoàn toàn dựa trên nghiên cứu và hiểu biết của chúng em về chủ đề. Chúng em cam kết rằng không có phần nào của bài viết được sao chép hoặc tham khảo từ nguồn khác mà không được ghi rõ.

Chúng em cũng cam đoan rằng mọi thông tin, số liệu, và ý kiến cá nhân được trình bày trong bài viết là chân thực và không gian dối. Chúng em chịu trách nhiệm hoàn toàn về tính xác thực của nội dung đã được trình bày.

Cuối cùng, chúng em cam đoan rằng chúng em đã làm việc một cách tận tâm và nghiêm túc để hoàn thành bài tập lớn này với mục đích cung cấp thông tin hữu ích và chất lượng. Chúng em rất mong nhận được sự đánh giá và góp ý xây dựng từ các bạn đọc và giáo viên hướng dẫn để cải thiện và hoàn thiện bài viết.

MỤC LỤC

LỜI NÓI ĐẦU	4
LỜI CẢM ƠN.....	5
LỜI CAM ĐOAN	6
DANH MỤC HÌNH ẢNH.....	9
DANH MỤC BẢNG BIỂU.....	10
Chương 1: Tổng quan bài toán	11
1.1. Nhận dạng đối tượng	11
1.1.1. Cơ sở của nhận dạng đối tượng.....	11
1.1.2. Quy trình nhận dạng đối tượng	11
1.1.3. Khó khăn trong bài toán nhận dạng đối tượng	13
1.1.4. Ứng dụng của nhận dạng đối tượng	14
1.2. Phương pháp áp dụng cho bài toán.....	15
1.2.1. CNN	15
1.2.2. Mô hình thuật toán YOLOv3	16
1.3. Công nghệ sử dụng	19
1.3.1. Python	19
1.3.2. Thư viện OpenCV	19
1.3.4. PyQt6	20
Chương 2: Xây dựng hệ thống	22
2.1. Yêu cầu bài toán	22
2.1.1. Mô tả	22
2.1.2. Thành phần bài toán.....	22
2.2. Xây dựng hệ thống.....	23
Chương 3: Thực nghiệm.....	28
3.1. Dữ liệu	28
3.1.1. COCO Dataset (https://cocodataset.org/#download):.....	28
3.1.2. Tiền xử lý dữ liệu:.....	28
3.2. Độ đo.....	32

3.2.1. Precision (Độ chính xác).....	32
3.2.2. Recall (Độ bao phủ)	33
3.2.3. mAP (mean Average Precision).....	33
3.3. Kết quả thực nghiệm.....	35
3.3.1. Tổng quan về kết quả.....	35
3.3.2. Nhận xét tổng quát	36
3.4. Xây dựng giao diện chương trình	36
KẾT LUẬN	38
TÀI LIỆU THAM KHẢO	40

DANH MỤC HÌNH ẢNH

Hình 1.1: Minh họa tổng quan về quá trình nhận dạng đối tượng trong thị giác máy tính	12
Hình 1.2: Cấu trúc của CNN	15
Hình 1.3: Cấu trúc của mô hình thuật toán YOLOv3	17
Hình 1.4: So sánh tốc độ của các mô hình phát hiện đối tượng với YOLOv3 cùng thời điểm ra mắt	17
Hình 2.1: Hình ảnh mô tả bài toán	22
Hình 2.2: Hệ thống nhận dạng đối tượng với mô hình YOLO	23
Hình 2.3: Hàm mất mát (Loss Function) trong mô hình YOLOv3	25
Hình 2.4: Công thức tính IOU	26
Hình 2.5: Kết quả sau khi hệ thống xử lý và nhận dạng đối tượng với mô hình YOLO	27
Hình 3.1: Một số hình ảnh trong COCO Dataset	28
Hình 3.2: Kết quả minh họa kỹ thuật xoay ảnh.....	29
Hình 3.3: Kết quả minh họa kỹ thuật lật ảnh.....	30
Hình 3.4: Kết quả minh họa kỹ thuật thay đổi tương phản	31
Hình 3.5: Kết quả minh họa kỹ thuật thay đổi độ sáng	32
Hình 3.6: Hình ảnh kết quả quá trình training.....	35
Hình 3.7: Giao diện hệ thống nhận dạng đối tượng	36

DANH MỤC BẢNG BIỂU

Bảng 3.1: Kết quả quá trình huấn luyện mô hình.....	35
---	----

Chương 1: Tổng quan bài toán

1.1. Nhận dạng đối tượng

Nhận dạng đối tượng (Object Recognition) là một trong những lĩnh vực nghiên cứu quan trọng và cốt lõi của thị giác máy tính (Computer Vision). Đây là quá trình cho phép máy tính có khả năng phát hiện, nhận biết và phân loại các đối tượng xuất hiện trong ảnh hoặc video kỹ thuật số. Mục tiêu chính của nhận dạng đối tượng là mô phỏng khả năng nhận thức trực quan của con người, giúp máy tính có thể "nhìn" và "hiểu" thế giới xung quanh một cách tự động.

Trong thực tế, con người có thể dễ dàng nhận biết và phân biệt các đối tượng khác nhau trong môi trường xung quanh một cách tự nhiên và gần như ngay lập tức. Tuy nhiên, đối với máy tính, đây là một thách thức phức tạp đòi hỏi nhiều bước xử lý và tính toán. Máy tính cần phải được "học" cách nhận biết các đặc điểm đặc trưng của đối tượng thông qua các thuật toán và mô hình toán học phức tạp.

1.1.1. Cơ sở của nhận dạng đối tượng

Nhận dạng đối tượng dựa trên nguyên lý cơ bản là mọi đối tượng đều có những đặc điểm riêng biệt có thể được mô tả thông qua các đặc trưng (features). Các đặc trưng này có thể bao gồm:

Thứ nhất, về hình dạng hình học, mỗi đối tượng sẽ có những đường nét, góc cạnh và tỷ lệ đặc trưng riêng. Ví dụ, một chiếc ô tô thường có hình dạng hộp chữ nhật với các đường cong nhất định, trong khi một con mèo có hình dạng hữu cơ với các đường cong mềm mại hơn.

Thứ hai, về màu sắc và kết cấu, các đối tượng thường có những mẫu màu và bề mặt đặc trưng. Chẳng hạn như lông của động vật có kết cấu khác biệt so với bề mặt kim loại của xe cộ.

Thứ ba, về cấu trúc không gian, các đối tượng thường có mối quan hệ không gian nhất định giữa các bộ phận cấu thành. Ví dụ như khuôn mặt người luôn có hai mắt đối xứng nhau qua trục dọc.

1.1.2. Quy trình nhận dạng đối tượng

Một hệ thống nhận dạng đối tượng hoàn chỉnh thường bao gồm các giai đoạn xử lý chính sau:

Giai đoạn đầu tiên là thu nhận dữ liệu, trong đó hệ thống tiếp nhận hình ảnh hoặc video từ các nguồn như camera, sensor hoặc tệp tin số. Chất lượng của dữ liệu đầu vào có ảnh hưởng quan trọng đến hiệu quả của quá trình nhận dạng.

Tiếp theo là giai đoạn tiền xử lý, nơi hình ảnh được chuẩn hóa và tối ưu hóa để phù hợp với các bước xử lý tiếp theo. Quá trình này có thể bao gồm việc điều chỉnh độ sáng, độ tương phản, loại bỏ nhiễu và chuẩn hóa kích thước.

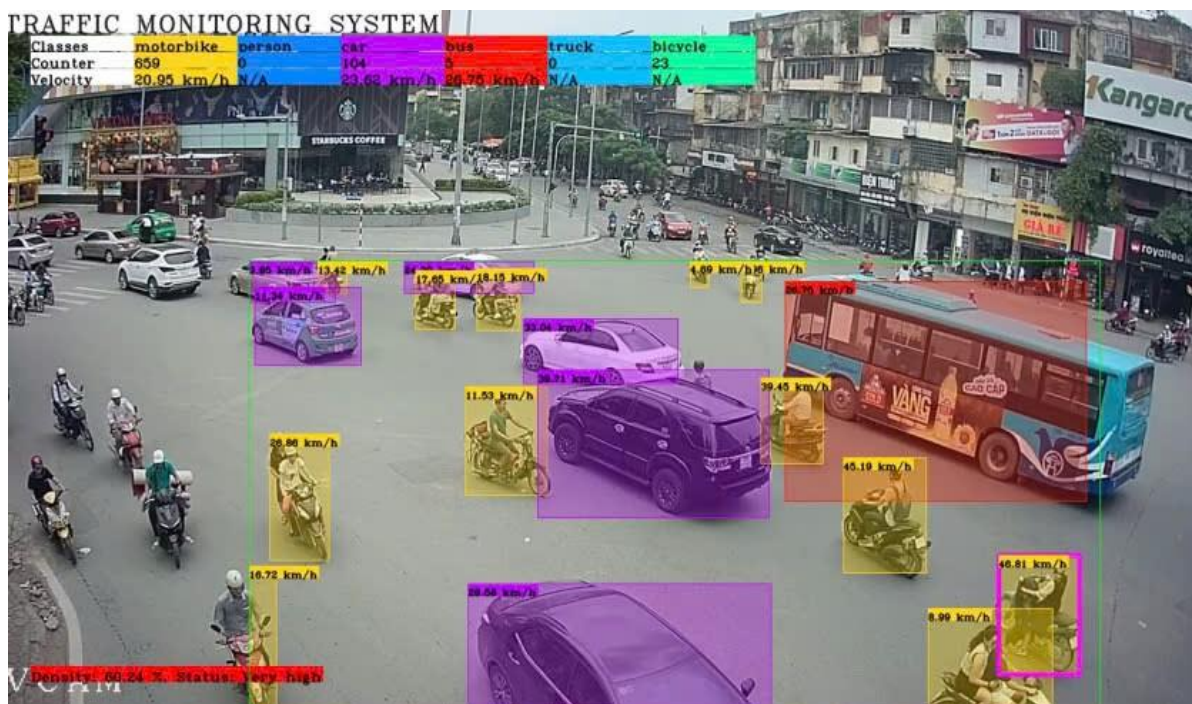
Giai đoạn then chốt là trích xuất đặc trưng, trong đó hệ thống phân tích và rút trích các thông tin quan trọng từ hình ảnh. Các đặc trưng này có thể là các điểm đặc biệt, cạnh, góc, hoặc các mẫu texture đặc trưng của đối tượng.

Cuối cùng là giai đoạn phân loại, nơi hệ thống sử dụng các mô hình học máy để so sánh các đặc trưng đã trích xuất với cơ sở dữ liệu đã được huấn luyện trước đó để đưa ra quyết định về danh tính của đối tượng.

1.1.3. Vai trò của học máy trong nhận dạng đối tượng

Học máy đóng vai trò quan trọng trong việc nâng cao hiệu quả của các hệ thống nhận dạng đối tượng. Thông qua quá trình học từ dữ liệu, các mô hình học máy có thể tự động phát hiện các mẫu và đặc trưng phức tạp mà con người khó có thể mô tả một cách tường minh.

Đặc biệt, với sự phát triển của học sâu (Deep Learning) và các mạng nơ-ron tích chập (Convolutional Neural Networks - CNN), khả năng nhận dạng đối tượng đã đạt được những tiến bộ vượt bậc. Các mô hình học sâu có thể tự động học các biểu diễn phân cấp của đặc trưng, từ các đặc trưng cơ bản như cạnh và góc đến các đặc trưng phức tạp hơn như khuôn mặt hay các bộ phận của đối tượng.



Hình 1.1: Minh họa tổng quan về quá trình nhận dạng đối tượng trong thị giác máy tính

Tóm lại, nhận dạng đối tượng là một lĩnh vực nghiên cứu phức tạp nhưng có tầm quan trọng to lớn trong thị giác máy tính. Sự kết hợp giữa các kỹ thuật xử lý ảnh truyền thống và các phương pháp học máy hiện đại đã tạo ra những hệ thống nhận dạng ngày càng thông minh và hiệu quả, mở ra nhiều ứng dụng quan trọng trong thực tế.

1.1.3. Khó khăn trong bài toán nhận dạng đối tượng

Nhận dạng đối tượng trong thị giác máy tính đối mặt với nhiều thách thức phức tạp, bao gồm:

Biến đổi về tư thế và góc nhìn:

Đối tượng có thể xuất hiện ở nhiều tư thế và góc nhìn khác nhau, dẫn đến sự thay đổi đáng kể trong hình ảnh thu được. Ví dụ, khuôn mặt người có thể được chụp từ phía trước, bên cạnh hoặc từ trên xuống, gây khó khăn cho việc nhận dạng chính xác.

Sự che khuất và thiếu hụt thông tin:

Đối tượng có thể bị che khuất một phần hoặc hoàn toàn bởi các vật thể khác, làm giảm lượng thông tin cần thiết cho việc nhận dạng. Chẳng hạn, một chiếc xe có thể bị che khuất bởi cây cối hoặc các phương tiện khác trong ảnh.

Biến dạng và thay đổi hình dạng:

Đối tượng có thể thay đổi hình dạng do các yếu tố như chuyển động, biến dạng vật lý hoặc do góc nhìn. Ví dụ, một chiếc ô tô có thể trông khác nhau khi nhìn từ phía trước so với từ phía sau.

Sự phức tạp của nền và nhiễu:

Hình nền phức tạp hoặc nhiễu trong ảnh có thể gây nhầm lẫn cho hệ thống nhận dạng, đặc biệt khi đối tượng và nền có màu sắc hoặc kết cấu tương tự nhau. Ví dụ, nhận dạng một con mèo màu xám trên nền bê tông xám có thể gặp khó khăn.

Điều kiện ánh sáng và chất lượng hình ảnh:

Sự thay đổi về ánh sáng, bóng đổ hoặc chất lượng hình ảnh kém (như độ phân giải thấp, mờ) có thể ảnh hưởng tiêu cực đến khả năng nhận dạng. Ví dụ, nhận dạng khuôn mặt trong điều kiện ánh sáng yếu hoặc ngược sáng thường gặp nhiều khó khăn.

Đa dạng về chủng loại và biến thể của đối tượng:

Đối tượng có thể có nhiều biến thể về màu sắc, kích thước, hình dạng hoặc kiểu dáng, đòi hỏi hệ thống phải có khả năng nhận dạng chính xác trong mọi trường hợp. Ví dụ, nhận dạng các loại xe ô tô với nhiều mẫu mã và màu sắc khác nhau.

Thời gian xử lý và yêu cầu tính toán:

Đối với ứng dụng thời gian thực, hệ thống phải xử lý nhanh chóng và hiệu quả, đòi hỏi tài nguyên tính toán mạnh mẽ và thuật toán tối ưu. Ví dụ, trong hệ thống giám sát an ninh, việc nhận dạng đối tượng phải diễn ra tức thì để kịp thời phản ứng.

1.1.4. Ứng dụng của nhận dạng đối tượng

Nhận dạng đối tượng trong thị giác máy tính có nhiều ứng dụng quan trọng và đa dạng trong cuộc sống hàng ngày cũng như trong các ngành công nghiệp khác nhau. Dưới đây là một số ứng dụng cốt lõi và tiêu biểu:

An ninh và giám sát: Nhận dạng đối tượng được sử dụng rộng rãi trong các hệ thống an ninh để phát hiện và theo dõi các đối tượng như con người, xe cộ, hoặc các vật thể khác. Ví dụ, camera giám sát có thể sử dụng nhận dạng khuôn mặt để xác định danh tính của một cá nhân hoặc phát hiện các hành vi bất thường.

Xe tự hành: Trong ngành công nghiệp xe hơi, nhận dạng đối tượng là một phần không thể thiếu của các hệ thống lái xe tự động. Các thuật toán nhận dạng đối tượng giúp xe tự hành nhận diện các đối tượng trên đường như xe khác, người đi bộ, biển báo giao thông, v.v., từ đó đưa ra các quyết định lái xe an toàn.

Y tế: Trong lĩnh vực y tế, nhận dạng đối tượng được áp dụng để phân tích hình ảnh y tế, giúp bác sĩ chẩn đoán bệnh một cách chính xác hơn. Ví dụ, các mô hình nhận dạng có thể phát hiện các khối u trong ảnh X-quang hoặc MRI, hỗ trợ đắc lực trong việc chẩn đoán và lên kế hoạch điều trị cho bệnh nhân.

Thương mại điện tử và bán lẻ: Nhận dạng đối tượng cũng được sử dụng trong thương mại điện tử và ngành bán lẻ để cải thiện trải nghiệm mua sắm của khách hàng. Ví dụ, các ứng dụng có thể nhận dạng sản phẩm trong ảnh và liên kết người dùng đến nơi họ có thể mua sản phẩm đó. Ngoài ra, trong các cửa hàng, công nghệ này có thể giúp theo dõi hàng tồn kho và phân tích hành vi mua hàng của khách.

Nông nghiệp: Trong nông nghiệp, nhận dạng đối tượng được sử dụng để phân tích các bức ảnh từ drone hoặc máy bay không người lái nhằm đánh giá sức khỏe của cây trồng, phát hiện sâu bệnh, và quản lý nông trại hiệu quả hơn.

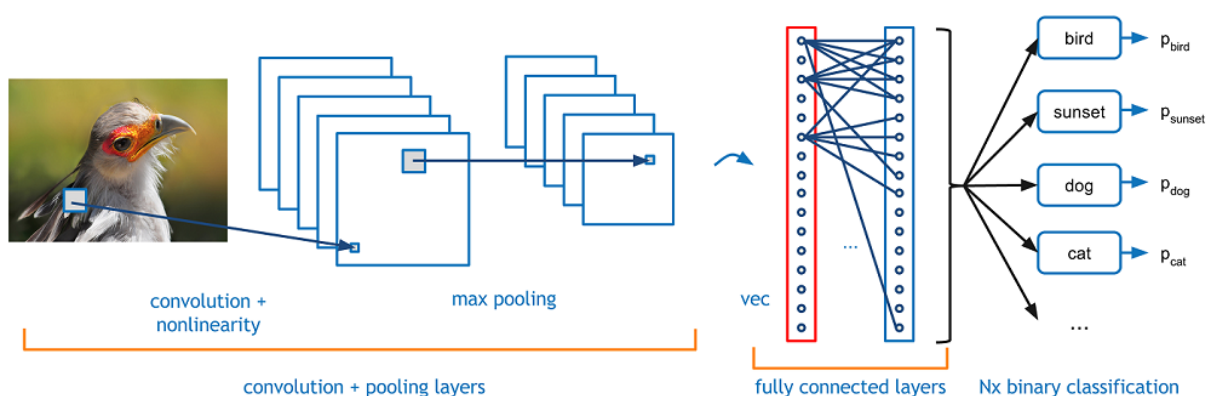
Những ứng dụng này chỉ là một phần nhỏ trong số rất nhiều khả năng của nhận dạng đối tượng trong thị giác máy tính. Sự phát triển của công nghệ này tiếp tục mở ra nhiều cơ hội mới để cải thiện cuộc sống và hiệu quả công việc trong nhiều lĩnh vực.

1.2. Phương pháp áp dụng cho bài toán

1.2.1. CNN

CNN (Convolutional Neural Network) là một loại mạng nơ-ron nhân tạo được thiết kế đặc biệt để xử lý dữ liệu dạng lưới, chẳng hạn như hình ảnh và video. CNN được sử dụng phổ biến trong các bài toán thị giác máy tính như phân loại ảnh, phát hiện đối tượng, nhận dạng khuôn mặt, và phân đoạn ảnh.

CNN hoạt động bằng cách trích xuất các đặc trưng từ dữ liệu đầu vào thông qua các lớp tích chập (convolutional layers), sau đó sử dụng các lớp kết nối đầy đủ (fully connected layers) để thực hiện dự đoán. Cơ chế này giúp CNN học cách phát hiện các đặc điểm của hình ảnh, từ đơn giản như đường nét, cạnh, đến phức tạp như hình dạng hoặc đối tượng cụ thể.



Hình 1.2: Cấu trúc của CNN

CNN hoạt động dựa trên kiến trúc:

Tích chập (Convolution):

- Là phép toán chính của CNN, trong đó một bộ lọc (filter) được trượt qua hình ảnh để trích xuất các đặc trưng quan trọng.

- Bộ lọc học cách phát hiện các đặc điểm như cạnh, góc, hoặc họa tiết cụ thể.

Kết hợp (Pooling):

- Kỹ thuật giảm kích thước không gian của đặc trưng để giảm số lượng tham số và tăng tính hiệu quả.

- Các loại pooling phổ biến: max pooling (chọn giá trị lớn nhất), average pooling (tính giá trị trung bình).

Kết nối đầy đủ (Fully Connected):

- Sau các lớp tích chập và pooling, đặc trưng đã được trích xuất sẽ được đưa vào các lớp kết nối đầy đủ để thực hiện dự đoán (phân loại, xác suất, v.v.).

Ưu điểm:

- Tự động học đặc trưng: Không cần thiết kế thủ công các đặc trưng như phương pháp truyền thống (SIFT, HOG). Các lớp tích chập học cách nhận diện đặc trưng từ cơ bản đến phức tạp.
- Khả năng tổng quát hóa tốt: CNN có thể hoạt động tốt trên nhiều loại dữ liệu hình ảnh khác nhau.
- Hiệu quả tính toán: Các phép tích chập có thể tận dụng tính song song của GPU, giúp tăng tốc độ huấn luyện.
- Khả năng mở rộng: CNN có thể được áp dụng cho nhiều ứng dụng khác nhau, từ phân loại ảnh đến nhận dạng khuôn mặt, phát hiện đối tượng.

Nhược điểm:

- Đòi hỏi dữ liệu lớn: CNN thường yêu cầu lượng dữ liệu huấn luyện lớn để tránh overfitting và đạt hiệu suất cao.
- Tiêu tốn tài nguyên tính toán: Các mô hình CNN lớn yêu cầu phần cứng mạnh mẽ (GPU, TPU) để huấn luyện.
- Nhạy cảm với nhiễu và biến dạng: CNN có thể gặp khó khăn khi dữ liệu đầu vào bị nhiễu hoặc biến dạng mà không có kỹ thuật tăng cường dữ liệu (data augmentation) phù hợp.
- Thiếu khả năng giải thích: Mặc dù CNN đạt độ chính xác cao, nó hoạt động như một "hộp đen", khó giải thích cách đưa ra quyết định.

1.2.2. Mô hình thuật toán YOLOv3

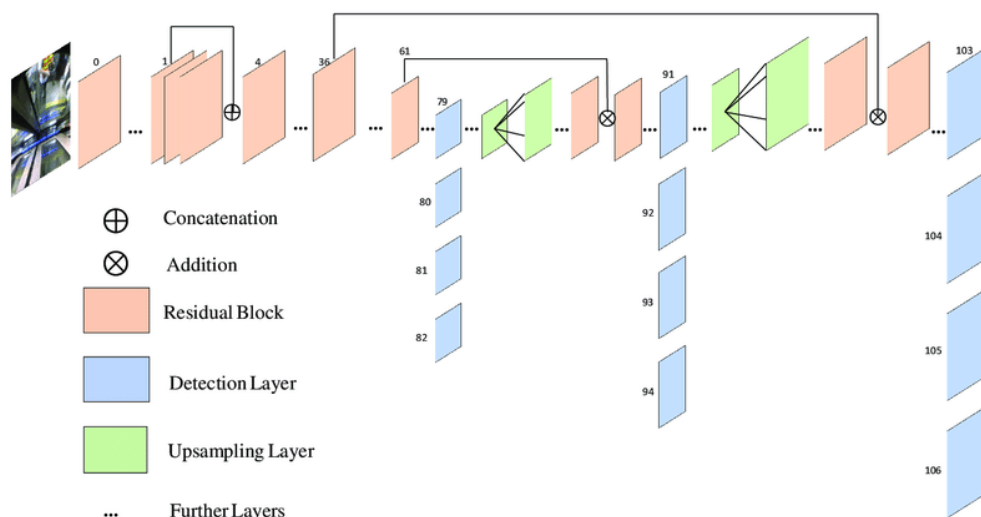
YOLO (You Only Look Once) là một thuật toán phát hiện đối tượng (object detection) nổi bật, được giới thiệu bởi Joseph Redmon vào năm 2016. Đây là một mô hình học sâu sử dụng mạng nơ-ron tích chập (CNN) để phát hiện và định vị các đối tượng trong hình ảnh. Điểm đặc trưng của YOLO là khả năng phát hiện đối tượng trong thời gian thực, giúp nó trở thành một lựa chọn lý tưởng cho các ứng dụng yêu cầu tốc độ cao.

YOLO là một thuật toán phát hiện đối tượng thời gian thực, sử dụng cách tiếp cận "single pass" để phân tích toàn bộ hình ảnh và dự đoán các bounding box (hộp giới hạn) cùng với nhãn lớp của đối tượng. Ý tưởng chính:

- Phát hiện đồng thời: Xử lý toàn bộ hình ảnh trong một lần duy nhất, thay vì chia nhỏ thành các vùng như các phương pháp truyền thống (R-CNN, Faster R-CNN).

- Mạng hồi quy: Biến phát hiện đối tượng thành bài toán hồi quy duy nhất, dự đoán vị trí bounding box, độ tin cậy, và xác suất lớp.

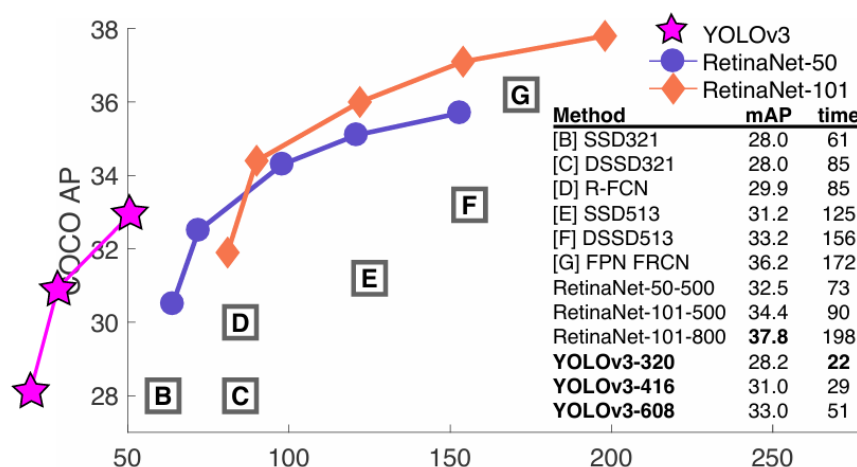
- Chia lưới hình ảnh: Hình ảnh được chia thành các ô lưới $S \times S$. Mỗi ô lưới dự đoán các bounding box nếu tâm đối tượng nằm trong ô đó.



Hình 1.3: Cấu trúc của mô hình thuật toán YOLOv3

Lý do lựa chọn mô hình YOLOv3

Nhóm em lựa chọn mô hình YOLOv3 vì đây là phiên bản cân bằng tốt giữa tốc độ và độ chính xác, đáp ứng hiệu quả yêu cầu của bài toán. YOLOv3 được thiết kế để hoạt động nhanh, chính xác, và tối ưu trên các thiết bị có tài nguyên hạn chế, đồng thời phù hợp với các ứng dụng thực tế trong nhiều lĩnh vực. So với các mô hình khác tại thời điểm ra mắt, YOLOv3 vượt trội ở khả năng phát hiện đối tượng đa scale, tính đơn giản trong triển khai và khả năng tối ưu hóa dễ dàng.



Hình 1.4: So sánh tốc độ của các mô hình phát hiện đối tượng với YOLOv3 cùng thời điểm ra mắt

So với SSD

Tốc độ: YOLOv3 nhanh hơn SSD ở tất cả các kích thước (320, 416, 608) với độ chính xác tương đương hoặc tốt hơn.

Độ chính xác: SSD chỉ đạt mAP 33.2 với SSD513, trong khi YOLOv3-608 đạt mAP 33.0 nhanh hơn đáng kể.

So với RetinaNet

Độ chính xác: RetinaNet đạt mAP cao hơn, đặc biệt ở RetinaNet-101-800 (37.8 mAP), nhưng tốc độ xử lý quá chậm (198ms).

Tốc độ: YOLOv3 nhanh gấp 2-3 lần RetinaNet ở cùng mức độ chính xác, điều này khiến YOLOv3 phù hợp hơn cho các ứng dụng cần xử lý nhanh như video streaming.

So với FPN FRCN

Độ chính xác: FPN Faster R-CNN đạt mAP cao (36.2), nhưng tốc độ xử lý chậm (172ms). Điều này khiến FPN FRCN không phù hợp với các ứng dụng thời gian thực.

Ưu thế của YOLOv3: Dù mAP thấp hơn, tốc độ của YOLOv3 vượt trội, đặc biệt với YOLOv3-608 đạt tốc độ xử lý nhanh gấp 3 lần so với FPN FRCN.

Ưu điểm và nhược điểm của mô hình YOLOv3:

Ưu điểm

Tốc độ cao: Phù hợp với các ứng dụng thời gian thực, có thể đạt hàng chục khung hình mỗi giây (FPS).

Hiệu quả trên thiết bị hạn chế tài nguyên: Có thể triển khai trên thiết bị như Raspberry Pi.

Phát hiện đa đối tượng: Khả năng phát hiện nhiều đối tượng trong cùng một hình ảnh.

Cấu trúc đơn giản: Toàn bộ quy trình phát hiện được xử lý trong một mạng duy nhất.

Dễ dàng triển khai: có nhiều phiên bản được tối ưu hóa, như YOLOv4, YOLOv5 hoặc YOLOv8, với các model được tinh chỉnh sẵn, dễ dàng áp dụng cho các bài toán thực tế.

Nhược điểm:

Hiệu suất kém hơn với đối tượng nhỏ: Đặc biệt trong các hình ảnh có độ phân giải cao.

Vùng giao nhau: Phát hiện kém với các đối tượng quá gần nhau.

Khả năng tổng quát hóa hạn chế: Yêu cầu huấn luyện với dữ liệu phong phú và đa dạng để hoạt động tốt trong các môi trường khác nhau.

Yêu cầu dữ liệu chất lượng cao: Dữ liệu huấn luyện cần được gắn nhãn chính xác và phản ánh đúng các tình huống trong thực tế.

1.3. Công nghệ sử dụng

1.3.1. Python

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Lợi ích khi lập trình bằng ngôn ngữ Python:

Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.

Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.

Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.

Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

1.3.2. Thư viện OpenCV

OpenCV (viết tắt của Open Source Computer Vision Library) là một thư viện mã nguồn mở chuyên dùng trong xử lý ảnh và thị giác máy tính. Công nghệ cung cấp các công cụ và thư viện để phân tích và xử lý ảnh, video từ việc xác định các đối tượng trong ảnh đến việc nhận diện khuôn mặt hoặc theo dõi chuyển động khác.

Ứng dụng của OpenCV

Nhận diện khuôn mặt: OpenCV cung cấp các thuật toán mạnh mẽ để nhận diện và phân tích khuôn mặt trong ảnh, video. Công nghệ sẽ được sử dụng trong các ứng dụng nhận diện khuôn mặt tự động, nhận dạng người dùng và các hệ thống an ninh.

Xác định đối tượng: OpenCV cung cấp các công cụ để xác định và phân tích các đối tượng trong ảnh, video. Công cụ đã trở thành một phần quan trọng của các ứng dụng như theo dõi vật thể, nhận diện biển số xe và quản lý hàng hóa.

Thị giác công nghệ: OpenCV có nhiều tiện ích dùng để xây dựng các hệ thống thị giác máy tính cho robot và xe tự động. Những tính năng thường gặp chính là: nhận diện đường, dự đoán chuyển động, tránh va chạm.

Xử lý video: OpenCV và thư viện liên quan được tích hợp để xử lý, phân tích video theo yêu cầu. Tiện ích có khả năng trích xuất thông tin từ video, phát hiện chuyển động và phân tích hành vi. Từ đó cho thấy OpenCV có rất nhiều ứng dụng trong lĩnh vực thị giác máy tính và xử lý ảnh.

1.3.3. PyTorch

PyTorch là một framework học máy dựa trên thư viện Torch, được sử dụng trong lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên, do Meta AI phát triển và ngày nay là một phần của Linux Foundation. Đây là phần mềm tự do nguồn mở phát hành dựa trên giấy phép BSD đã qua sửa đổi. Mặc dù được phát triển chính yếu ở dạng giao diện ngôn ngữ Python.

Lợi ích khi sử dụng PyTorch

Mã nguồn mở: như đã chia sẻ ở trên, nhờ sử dụng mã nguồn mở đã tạo nên một cộng đồng rất lớn với nguồn tài nguyên “chất lượng” và “số lượng”.

Khả năng xử lý đồ họa: như Numpy đồng thời có kiểm soát CPU & GPU rõ ràng.

Tập hợp nhiều Pythonic trong tự nhiên.

Dễ dàng xử lý khi gặp bug.

Có TouchScript được xem là một tập hợp con của Python. Tập hợp này giúp triển khai các ứng dụng vào quy mô sản xuất từ đó mở rộng quy mô. Đồng thời khi nói đến việc xây dựng các nguyên mẫu với tốc độ nhanh, sử dụng Pytorch được ưu tiên hơn so với Tensorflow vì nó nhẹ hơn.

Các hàm, cú pháp cơ bản trong Pytorch giúp xử lý các bài toán về AI nhanh chóng.

1.3.4. PyQt6

PyQt6 là một bộ công cụ giao diện người dùng (GUI) mạnh mẽ và phổ biến dành cho Python, được xây dựng trên Qt 6 – một framework phát triển ứng dụng đa nền tảng. PyQt6 cung cấp các thành phần cần thiết để tạo các ứng dụng desktop có giao diện đẹp, hiện đại và tương tác cao, chạy được trên các hệ điều hành như Windows, macOS và Linux.

Đặc điểm nổi bật của PyQt6:

Đa nền tảng: Cung cấp khả năng phát triển ứng dụng trên nhiều hệ điều hành (Windows, macOS, Linux) mà không cần thay đổi mã nguồn.

Hỗ trợ phong phú:

Hỗ trợ các thành phần GUI như nút bấm, hộp văn bản, menu, bảng, biểu đồ, canvas đồ họa, v.v.

Dễ dàng tạo giao diện phức tạp với các công cụ như Qt Designer (thiết kế GUI trực quan).

Tích hợp mạnh mẽ:

Kết hợp giữa Python và Qt, tận dụng sự dễ sử dụng của Python với hiệu năng cao và tính năng phong phú của Qt.

Hỗ trợ lập trình hướng đối tượng:

Sử dụng cú pháp Pythonic để viết mã GUI, giúp dễ đọc và bảo trì.

Tích hợp QtCore và QtGui:

QtCore: Xử lý logic ứng dụng, signal/slot, đa luồng, v.v.

QtGui: Tạo giao diện đồ họa bao gồm cửa sổ, nút, biểu mẫu, v.v.

Qt Quick:

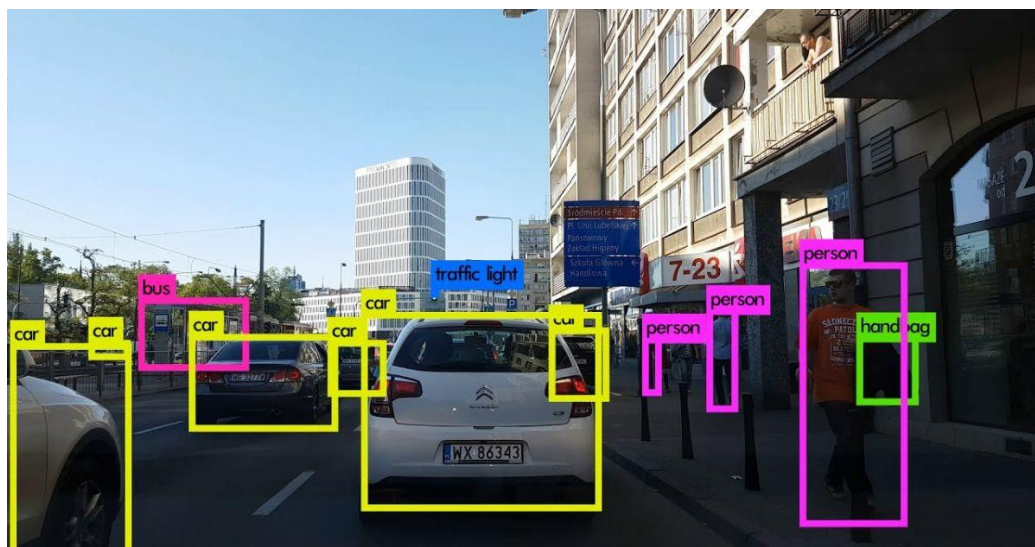
Hỗ trợ phát triển giao diện hiện đại bằng QML (Qt Modeling Language)

Chương 2: Xây dựng hệ thống

2.1. Yêu cầu bài toán

2.1.1. Mô tả

Hệ thống phát hiện và nhận diện đối tượng trong video thời gian thực nhằm mục đích phát hiện và phân loại các đối tượng (như người, xe, động vật, v.v.) trong từng khung hình của video trực tiếp. Hệ thống cần đáp ứng yêu cầu về tốc độ xử lý cao và độ chính xác cao khi nhận diện các đối tượng.



Hình 2.1: Hình ảnh mô tả bài toán

2.1.2. Thành phần bài toán

Đầu vào:

Luồng video thời gian thực từ camera hoặc video được cung cấp trước.

Dữ liệu đầu vào là các khung hình liên tiếp (frame) được trích xuất từ video hoặc camera.

Mỗi khung hình là một ảnh RGB có độ phân giải cụ thể ($H \times W$).

Yêu cầu đầu ra:

Bounding boxes: Vị trí của các đối tượng được phát hiện trong khung hình.

Nhãn lớp: Tên hoặc loại đối tượng được phát hiện (ví dụ: người, ô tô, mèo, chó, v.v.).

Confidence scores: Xác suất hoặc độ tin cậy của dự đoán.

Theo dõi đối tượng: Liên kết các đối tượng qua các khung hình liên tiếp, duy trì ID duy nhất cho mỗi đối tượng.

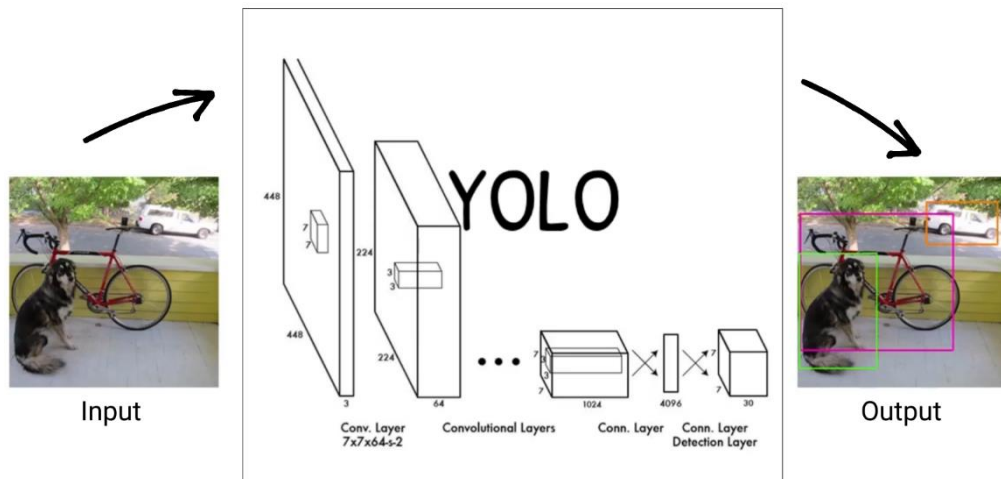
Mục tiêu:

Tốc độ xử lý nhanh (đảm bảo ≥ 30 FPS để không gây độ trễ cho video).

Độ chính xác cao trong việc phát hiện và gán nhãn đối tượng.

Khả năng xử lý đa đối tượng, bao gồm các đối tượng nhỏ, lớn và bị che khuất

2.2. Xây dựng hệ thống



Hình 2.2: Hệ thống nhận dạng đối tượng với mô hình YOLO

Hệ thống nhận dạng đối tượng sẽ có quy trình hoạt động như sau:

Tiền xử lý ảnh đầu vào:

Đầu vào

Nếu đầu vào là video, sử dụng thư viện (như OpenCV) để đọc file video và trích xuất từng khung hình (frame).

Nếu đầu vào là camera, sử dụng OpenCV để kết nối với camera và lấy luồng video thời gian thực (cv2.VideoCapture).

Chuẩn hóa và chia tỷ lệ

Hình ảnh được chuẩn hóa và chia tỷ lệ: Hình ảnh đầu vào được điều chỉnh kích thước về một kích thước cố định để phù hợp với đầu vào của mạng.

Chia lưới hình ảnh: Hình ảnh được chia thành lưới kích thước $S \times S$. Mỗi ô lưới chịu trách nhiệm dự đoán bounding boxes cho các đối tượng mà tâm nằm trong ô đó.

Trích xuất đặc trưng:

YOLO sử dụng một mạng tích chập (convolutional neural network) để trích xuất đặc trưng từ hình ảnh đầu vào.

Trong YOLOv3, mạng Darknet-53 được sử dụng, gồm 53 lớp tích chập với các shortcut connections (kết nối tắt).

Các đặc trưng trích xuất bao gồm thông tin về cạnh, hình dạng, và các đối tượng trong hình ảnh.

Dự đoán bounding boxes:

YOLO sử dụng các anchor boxes làm tham chiếu để dự đoán bounding boxes. Anchor boxes là các kích thước hộp đã được xác định trước, phù hợp với các đối tượng phổ biến.

Tọa độ của bounding box (bx, by, bw, bh) được tính toán dựa trên:

Tọa độ tâm (bx, by): Được dự đoán tương đối so với vị trí ô lưới.

Chiều rộng và chiều cao (bw, bh): Dự đoán dưới dạng tỷ lệ dựa trên anchor box.

Công thức:

$$bx = \sigma(tx) + cx, \quad by = \sigma(ty) + cy$$

$$bw = p_w e^{tw}, \quad bh = p_h e^{th}$$

Trong đó:

$\sigma(tx)$, $\sigma(ty)$: Kết quả dự đoán từ mạng (sigmoid activation).

cx , cy : Tọa độ ô lưới.

p_w , p_h : Kích thước anchor box.

Mỗi bounding box được dự đoán kèm một độ tin cậy (objectness score), thể hiện xác suất có đối tượng trong box và mức độ chính xác của dự đoán

Phân loại đối tượng:

Mỗi bounding box dự đoán một tập xác suất cho tất cả các lớp (ví dụ: ô tô, chó, mèo, v.v.) thông qua phương pháp hồi quy logistic. Hồi quy logistic là một phương pháp phân tích thống kê được sử dụng để dự đoán giá trị dữ liệu dựa trên các quan sát trước đó của tập dữ liệu, thuộc thuật toán học có giám sát, đây là thuật toán đơn giản nhưng lại rất hiệu quả trong bài toán phân loại (Classification)

Xác suất lớp được tính riêng biệt, không sử dụng softmax (giảm độ phụ thuộc giữa các lớp).

Kết hợp và tối ưu

Kết hợp các dự đoán từ nhiều scale:

+ YOLO dự đoán trên 3 thang đo khác nhau để xử lý các đối tượng lớn, trung bình, và nhỏ.

+ Kỹ thuật Feature Pyramid Networks (FPN) được sử dụng để trộn thông tin từ các tầng sâu và nông của mạng.

Hàm mất mát (Loss Function):

Hàm mất mát YOLO kết hợp các thành phần:

- + Loss vị trí: Chênh lệch giữa tọa độ dự đoán và ground truth.
- + Loss độ tin cậy: Độ chính xác của objectness score.
- + Loss phân loại: Chênh lệch giữa dự đoán lớp và ground truth.

$$\begin{aligned} \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} & \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} & \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} & (C_i - \hat{C}_i)^2 \\ + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} & (C_i - \hat{C}_i)^2 \\ + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} & (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

Hình 2.3: Hàm mất mát (Loss Function) trong mô hình YOLOv3

Trong đó:

λ_{coord} , $\lambda_{\text{noobject}}$: Là trọng số thành phần trong paper gốc tác giả lấy giá trị lần lượt là 5 và 0.5.

$K \times K$: Là tỷ lệ ô lưới tại ảnh đó (với một ảnh kích thước 416 thì có 3 tỷ lệ là 13×13 , 26×26 và 52×52).

M: Số bounding box cần dự đoán tại một ô, ở YOLOv3 là bằng 3.

$\mathbb{1}_{ij}^{\text{obj}} = 1$: Nếu box thứ j của ô thứ i có chứa object. Vì huấn luyện cần các image với ground-truth (vị trí của các objects) nên YOLO biết điểm trung tâm của từng object rơi vào ô nào trong grid $K \times K$.

$I_{ij}^{noobj} = 1$: Nếu box thứ j của ô thứ i không chứa object.

$(2 - w_i \times h_i)$: Hệ số tỷ lệ để tăng tổn thất của hộp nhỏ thứ i

(x_i, y_i) : Là tọa độ tâm của ground-truth bounding box thứ i và (\hat{x}_i, \hat{y}_i) là tọa độ tâm của predicted bounding box thứ i .

(w_i, h_i) : Là kích thước của ground-truth bounding box thứ i và (\hat{w}_i, \hat{h}_i) là kích thước của predicted bounding box thứ i .

C_i : Điểm tin cậy của ô i . Đối với các hộp j của ô i nơi object tồn tại C_i luôn = 1.

\hat{C}_i : Điểm tin cậy dự đoán của ô i .

$\hat{C}_i = P(\text{contain object}) * IoU(\text{predict bbox}, \text{ground truth bbox})$.

$p_i(c)$: Xác suất có điều kiện, có hay không ô i có chứa một đối tượng của lớp $c \in \text{classes}$. Chú ý $p_i(c)$ luôn = 1 nếu đúng lớp c với ground - truth, ngược lại thì $p_i(c)$ luôn = 0.

$\hat{p}_i(c)$: Xác suất có điều kiện dự đoán của ô i .

classes: Các lớp đối tượng cần được nhận dạng, ví dụ chó, mèo, oto...

Theo dõi đối tượng trong luồng video (Object Tracking):


Sử dụng thuật toán DeepSORT hoặc ByteTrack kết hợp với YOLO. Điều này giúp duy trì ID cố định cho mỗi đối tượng trong suốt luồng video.

Hậu xử lý và xuất kết quả

Non-Maximum Suppression (NMS):

+ YOLO thường dự đoán nhiều bounding boxes chồng lấn nhau cho cùng một đối tượng.

+ NMS giữ lại bounding box có độ tin cậy cao nhất và loại bỏ các box còn lại dựa trên ngưỡng IOU (Intersection over Union). IOU là hàm đánh giá độ chính xác của object detector trên tập dữ liệu cụ thể. IoU được tính bằng:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Hình 2.4: Công thức tính IOU

Trong đó Area of Overlap là diện tích phần giao nhau giữa predicted bounding box với growth truth box, còn Area of Union là diện tích phần hợp giữa predicted bounding box với hộp growth truth. Nếu $IOU > 0.5$ thì prediction được đánh giá là tốt.

Xuất kết quả:

Hiển thị bounding boxes, nhãn lớp, và độ tin cậy trên từng frame của video hoặc luồng camera.

Sử dụng OpenCV để vẽ các bounding boxes và nhãn trực tiếp lên frame.



Hình 2.5: Kết quả sau khi hệ thống xử lý và nhận dạng đối tượng với mô hình YOLO

Chương 3: Thực nghiệm

3.1. Dữ liệu

Mô hình mà hệ thống sử dụng đã được huấn luyện sẵn từ bộ dữ liệu COCO (Common Objects in Context), một trong những bộ dữ liệu nổi bật trong lĩnh vực nhận diện đối tượng. Bộ dữ liệu này chứa hơn 120.000 hình ảnh và 80 lớp đối tượng khác nhau bao gồm các đối tượng như người, xe, chó, mèo, động vật, v.v.

3.1.1. COCO Dataset (<https://cocodataset.org/#download>):



Hình 3.1: Một số hình ảnh trong COCO Dataset

Bao gồm 80 class đối tượng, ví dụ như: người, xe đạp, chó, mèo, xe hơi, v.v.

Hơn 120.000 hình ảnh trong tập huấn luyện (training set) và 5.000 hình ảnh trong tập kiểm tra.

Dữ liệu kèm theo annotations với bounding boxes (hộp bao quanh) để đánh dấu vị trí đối tượng trong mỗi bức ảnh, cùng với các thông tin nhãn của từng lớp đối tượng.

3.1.2. Tiền xử lý dữ liệu:

Chuẩn hóa kích thước: Chuẩn hóa kích thước là quá trình thay đổi kích thước (resize) hình ảnh nhưng vẫn giữ nguyên thông tin quan trọng, không làm biến dạng tỷ lệ hoặc mất mát thông tin quan trọng. Hàm chuẩn hóa kích thước giúp đưa tất cả hình ảnh về cùng kích thước cố định (ví dụ: 416x416 pixel) để phù hợp với mô hình YOLO. Mỗi pixel tại vị trí (x, y) trong ảnh gốc sẽ được ánh xạ sang vị trí (x', y') trong ảnh mới:

$$x' = \frac{x}{W} \times W_t, \quad y' = \frac{y}{H} \times H_t$$

Trong đó:

W_t, H_t : Kích thước chiều dài, chiều rộng của mục tiêu.

W, H : Kích thước chiều dài, chiều rộng gốc của ảnh.

Chuẩn hóa pixel: Chuẩn hóa pixel là quá trình đưa giá trị của từng pixel về khoảng $[0, 1]$, thay vì $[0, 255]$ như ban đầu. Điều này giúp mô hình học sâu hội tụ nhanh hơn và giảm nguy cơ mất ổn định trong quá trình huấn luyện. Mỗi giá trị pixel p được chuẩn hóa thành giá trị p' trong khoảng $[0, 1]$:

$$p' = \frac{p}{255}$$

Trong đó:

p : Giá trị pixel gốc (nguyên trong khoảng $[0, 255]$).

p' : Giá trị pixel sau chuẩn hóa (float trong khoảng $[0, 1]$).

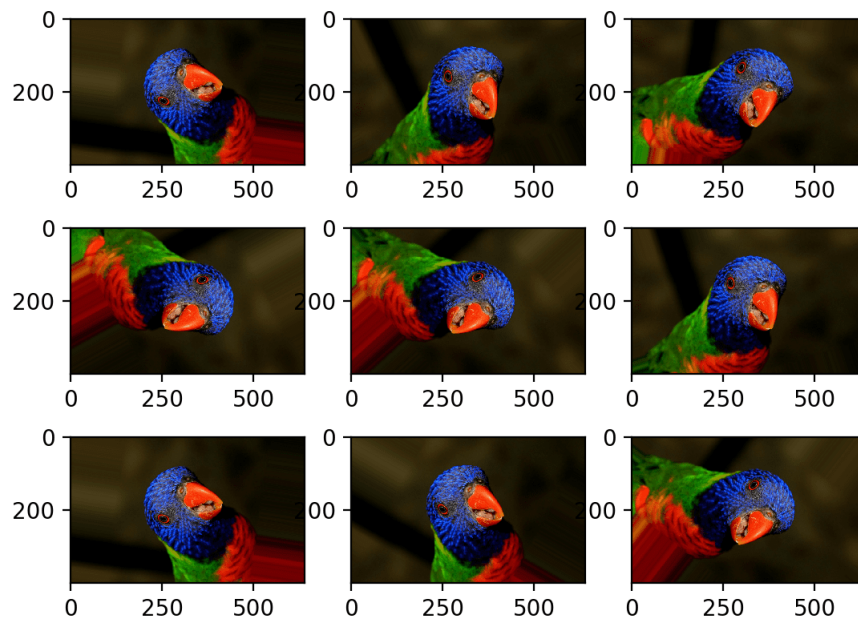
Tăng cường dữ liệu: Tăng cường dữ liệu là quá trình tạo ra các biến thể của dữ liệu gốc bằng cách áp dụng các phép biến đổi nhằm tăng tính đa dạng của bộ dữ liệu và cải thiện khả năng tổng quát hóa của mô hình. Các kỹ thuật phổ biến bao gồm:

Xoay ảnh:

Mục đích: Giúp mô hình học cách nhận diện đối tượng ở các góc độ khác nhau.

Phép toán: Mỗi pixel (x, y) được xoay quanh một tâm cố định (x_c, y_c) với góc θ :

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix}$$



Hình 3.2: Kết quả minh họa kỹ thuật xoay ảnh

Lật ảnh:

Mục đích: Tăng khả năng nhận diện các đối tượng đối xứng

Phép toán:

Lật ngang:

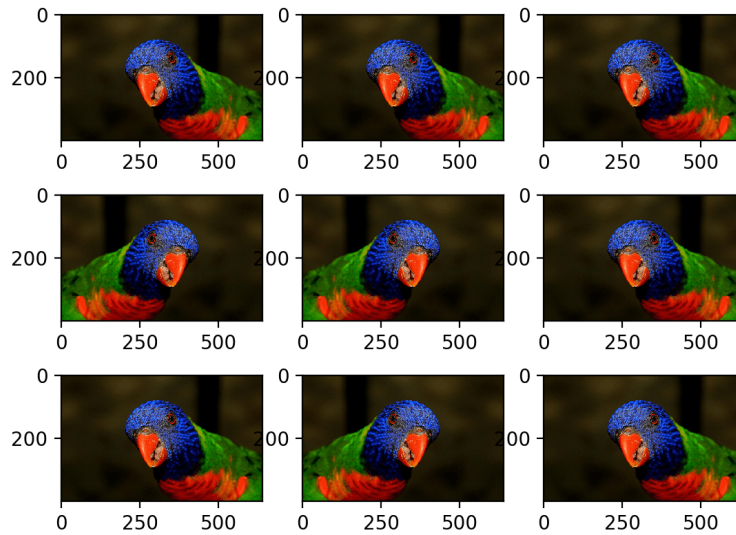
$$\hat{x} = x, \quad \hat{y} = H - 1 - y$$

Lật dọc:

$$\hat{x} = W - 1 - x, \quad \hat{y} = y$$

Lật cả hai chiều:

$$\hat{x} = W - 1 - x, \quad \hat{y} = H - 1 - y$$



Hình 3.3: Kết quả minh họa kỹ thuật lật ảnh

Thay đổi tương phản

Mục đích: Giúp mô hình hoạt động tốt hơn khi đối tượng trong ảnh có độ tương phản khác nhau.

Phép toán:

Nhân mỗi pixel với một hệ số α và cộng thêm β nếu cần:

$$I'(x, y) = \alpha \cdot I(x, y) + \beta$$

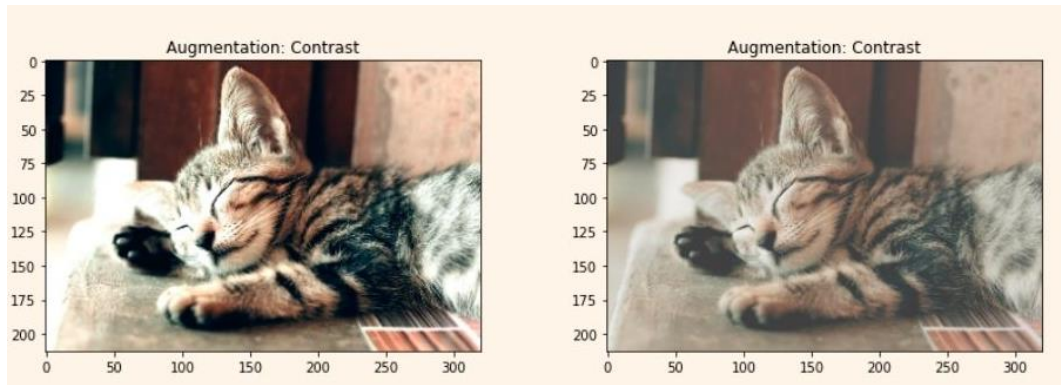
Trong đó:

$\alpha > 1$: Tăng tương phản

$0 < \alpha < 1$: Giảm tương phản

Giới hạn giá trị:

$$I'(x, y) = \min(\max(\alpha \cdot I(x, y) + \beta, 0), 255)$$



Hình 3.4: Kết quả minh họa kỹ thuật thay đổi tương phản

Thay đổi độ sáng:

Mục đích: Giúp mô hình nhận diện đối tượng trong các điều kiện ánh sáng khác nhau.

Phép toán:

Cộng thêm một giá trị β vào mỗi pixel:

$$I'(x, y) = I(x, y) + \beta$$

Trong đó:

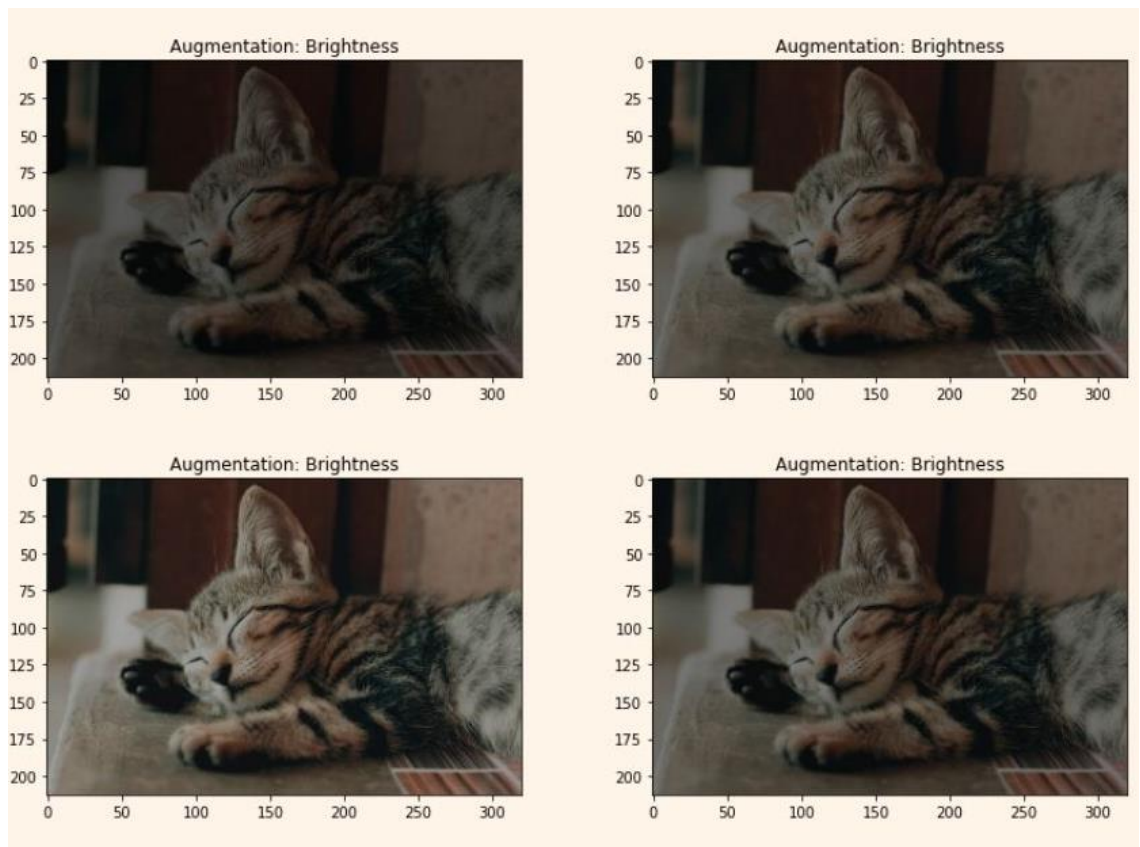
$\beta > 0$: Tăng sáng

$0 < \beta < 1$: Giảm sáng

Giới hạn giá trị:

$$I'(x, y) = \min(\max(I(x, y) + \beta, 0), 255)$$

Chia tách dữ liệu: Dữ liệu được chia thành ba phần: huấn luyện, kiểm tra và kiểm tra cuối cùng để đánh giá hiệu suất của mô hình.



Hình 3.5: Kết quả minh họa kỹ thuật thay đổi độ sáng

3.2. Độ đo

Trong quá trình đánh giá mô hình YOLOv3, sử dụng ba chỉ số chính để đo lường hiệu quả của mô hình trong việc phát hiện và nhận diện các đối tượng trong ảnh: Precision, Recall và mAP (mean Average Precision).

3.2.1. Precision (Độ chính xác)

Precision đo lường tỷ lệ các dự đoán đúng trên tổng số dự đoán của mô hình. Nó phản ánh mức độ chính xác của các đối tượng mà mô hình dự đoán là tồn tại.

$$Precision = \frac{TP}{TP + FP}$$

Ý nghĩa:

TP (True Positives): Số lượng dự đoán đúng (đối tượng thực sự tồn tại và được dự đoán đúng).

FP (False Positives): Số lượng dự đoán sai (mô hình dự đoán có đối tượng nhưng thực tế không có).

3.2.2. Recall (Độ bao phủ)

Recall đo lường tỷ lệ phát hiện đúng các đối tượng thực tế trong tổng số đối tượng cần phát hiện. Đây là chỉ số phản ánh khả năng của mô hình trong việc nhận diện tất cả các đối tượng có trong ảnh.

$$Recall = \frac{TP}{TP + FN}$$

Ý nghĩa:

FN (False Negatives): Số lượng đối tượng thực sự có nhưng mô hình không phát hiện được.

Recall cao nghĩa là mô hình phát hiện được hầu hết các đối tượng có thực.

3.2.3. mAP (mean Average Precision)

mAP (mean Average Precision) là trung bình của các giá trị AP (Average Precision) trên tất cả các lớp đối tượng trong bài toán phát hiện đối tượng hoặc phân loại. AP đo lường diện tích dưới đường cong Precision-Recall (PR curve) cho một lớp cụ thể. PR curve là biểu đồ biểu diễn mối quan hệ giữa Precision và Recall khi thay đổi ngưỡng (threshold) dự đoán.

Cách xây dựng PR Curve

1. Sắp xếp các dự đoán theo confidence score từ cao đến thấp.
2. Với mỗi điểm dự đoán:
 - Xác định True Positive (TP), False Positive (FP), và False Negative (FN).
 - Tính Precision và Recall tại điểm đó.
3. Nối các điểm (Recall, Precision) để tạo PR Curve.

Tính AP:

AP được tính bằng cách lấy trung bình giá trị Precision tại các điểm Recall khác nhau (thường chia đều từ 0 đến 1):

$$AP = \sum_{n=1}^N P_n \cdot \Delta R_n$$

Trong đó:

P_n : Precision tại điểm n

ΔR_n : Sự thay đổi Recall tại điểm n .

Tính mAP:

$$\text{mAP} = \frac{1}{C} \sum_{i=1}^C \text{AP}_i$$

Trong đó:

C: Số lượng lớp.

AP_i: Giá trị AP cho lớp i.

Quy trình tính mAP

Bước 1: Tính AP cho mỗi lớp

- + Với mỗi lớp, lấy danh sách tất cả các dự đoán (bounding box và confidence score).
- + So khớp các bounding box với các đối tượng thực tế dựa trên IoU (Intersection over Union):

Một dự đoán được coi là **True Positive (TP)** nếu:

$$\text{IoU} = \frac{\text{Diện tích giao nhau}}{\text{Diện tích hợp}} \geq \text{Ngưỡng IoU}$$

Ngưỡng IoU (Ngưỡng IoU phổ biến là 0.5 hoặc 0.5:0.95 đối với các mô hình hiện đại).

Nếu không, dự đoán được coi là **False Positive (FP)**.

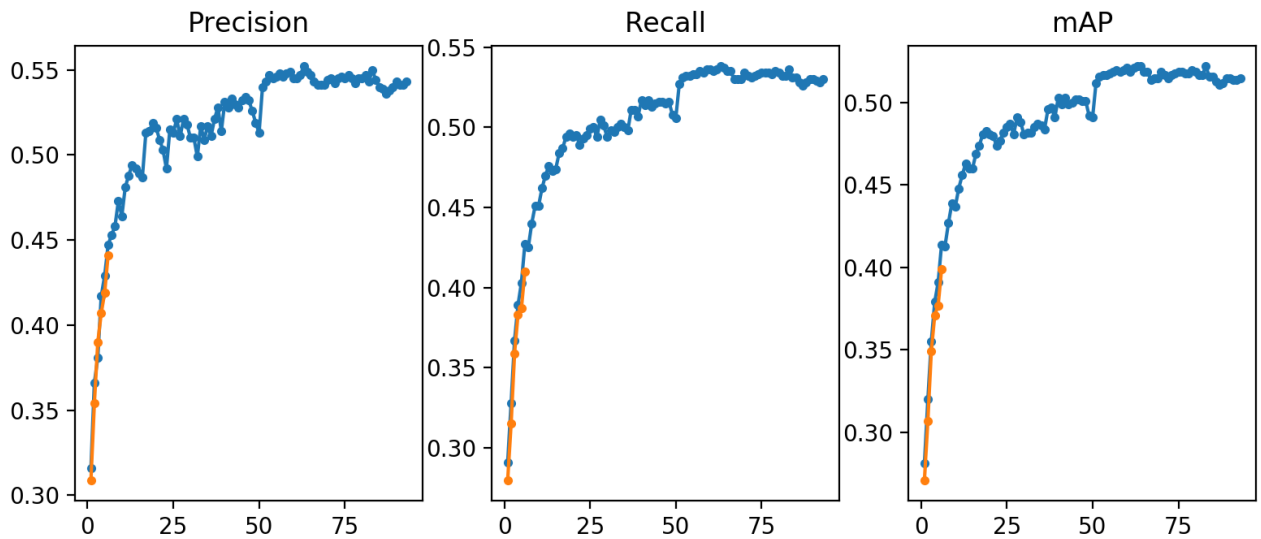
- + Tính Precision và Recall tại các ngưỡng khác nhau của confidence score.
- + Vẽ PR Curve và tính AP (diện tích dưới PR Curve).

Bước 2: Tính mAP

1. Lặp lại bước trên cho tất cả các lớp.
2. Lấy trung bình các giá trị AP để tính mAP.

3.3. Kết quả thực nghiệm

Sau quá trình huấn luyện, mô hình YOLOv3 đã được đánh giá trên các chỉ số Precision, Recall và mAP. Các kết quả thu được từ các chỉ số này giúp hiểu rõ hơn về khả năng phát hiện đối tượng của mô hình.



Hình 3.6: Hình ảnh kết quả quá trình training

3.3.1. Tổng quan về kết quả

Bảng 3.1: Kết quả quá trình huấn luyện mô hình

Chỉ số đánh giá	Kết quả	Đánh giá
Precision	Giá trị ban đầu: 0.30 Giá trị cuối: 0.55 Tăng trưởng: 83.3%	Precision tăng đều và ổn định, cho thấy model giảm thiểu được false positive
Recall	Giá trị ban đầu: 0.30 Giá trị cuối: 0.52 Tăng trưởng: 73.3%	Recall tăng nhanh trong 25 epoch đầu, sau đó tăng chậm và ổn định
mAP	Epoch 12: 0.45 (conf_thresh 0.30) Epoch 17: 0.48 (conf_thresh 0.30) Epoch 45: 0.50 (conf_thresh 0.30) Epoch 62: 0.522 (conf_thresh 0.30, img_size 416)	mAP tăng đều qua các epoch Tốc độ tăng chậm dần khi đạt ngưỡng 0.50 Đạt giá trị cao nhất 0.522 ở epoch 62 Không có dấu hiệu overfitting

3.3.2. Nhận xét tổng quát

Điểm mạnh:

Tất cả ba chỉ số (Precision, Recall, mAP) đều tăng ổn định, cho thấy mô hình đã học tốt và cải thiện khả năng phát hiện đối tượng qua từng epoch.

Không có dấu hiệu overfitting, mô hình có khả năng tổng quát tốt.

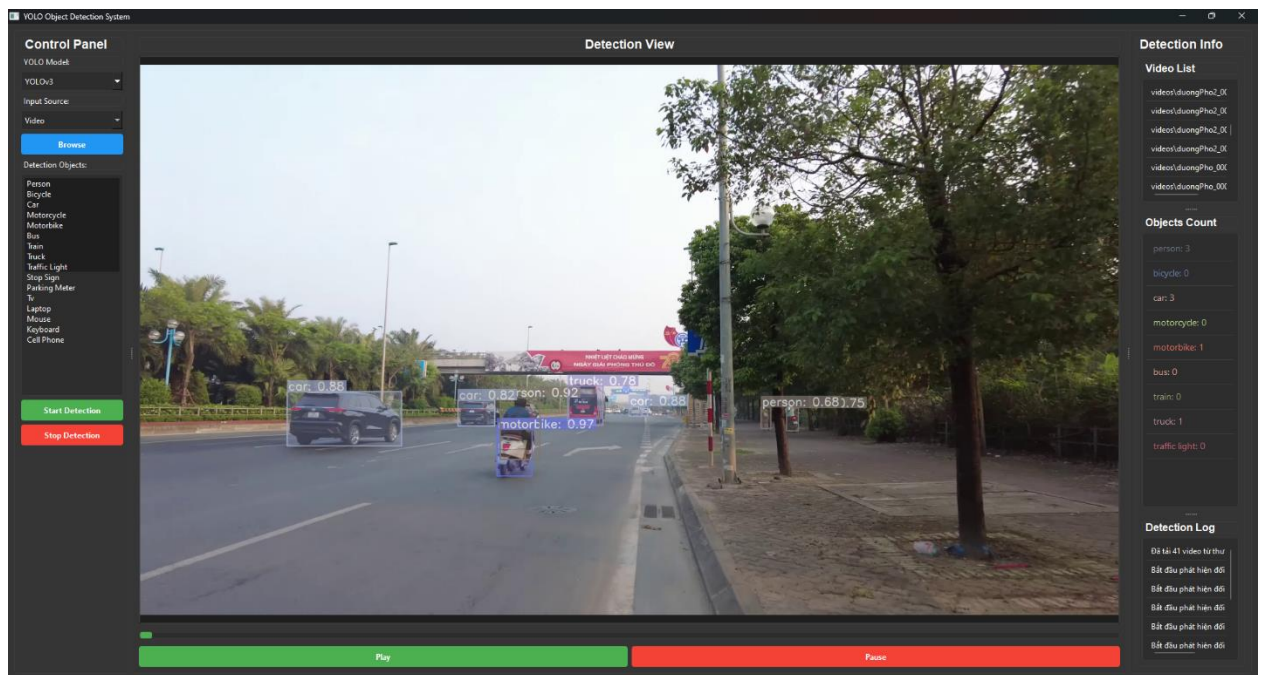
mAP đạt 0.522 là kết quả khá tốt, cho thấy mô hình có thể ứng dụng hiệu quả trong các bài toán nhận diện đối tượng.

Điểm cần cải thiện:

Precision và Recall có thể cải thiện thêm để đạt mức độ tối ưu.

Tốc độ hội tụ của mô hình có phần chậm sau epoch 45, điều này cần cải thiện trong các thử nghiệm sau.

3.4. Xây dựng giao diện chương trình



Hình 3.7: Giao diện hệ thống nhận dạng đối tượng

Giao diện chương trình sẽ gồm 3 phần chính:

- Control Panel (Bên trái):

Dropdown chọn YOLO Model (YOLOv3)

Dropdown chọn Input Source (Video)

Nút Browse

Danh sách các đối tượng có thể phát hiện (Person, Bicycle, Car, vv...)

Nút Start/Stop Detection màu xanh/đỏ

- Detection View (Phần giữa):

Hiển thị video/hình ảnh đang được phân tích

Các bounding box xung quanh đối tượng được phát hiện

Nhãn và độ tin cậy của mỗi đối tượng (ví dụ: car: 0.88, person: 0.95)

Thanh điều khiển Play/Pause ở dưới cùng

- Detection Info (Bên phải):

Video List hiển thị danh sách video

Objects Count đếm số lượng từng loại đối tượng được phát hiện

Detection Log ghi lại các sự kiện phát hiện

KẾT LUẬN

Kết quả đạt được

Trong bài tập lớn này, nhóm đã xây dựng thành công hệ thống theo dõi đối tượng trong video thời gian thực dựa trên thuật toán YOLOv3. Qua quá trình nghiên cứu và thực nghiệm, hệ thống đã đạt được các kết quả đáng ghi nhận:

- + **Hiệu suất nhận diện cao:** Với chỉ số mAP đạt 0.522, hệ thống chứng minh khả năng phát hiện và phân loại đối tượng hiệu quả. Precision và Recall cũng cho thấy sự cải thiện ổn định qua các vòng lặp huấn luyện.
- + **Tính thực tế:** Hệ thống xử lý dữ liệu thời gian thực với tốc độ ≥ 30 FPS, phù hợp cho các ứng dụng như giám sát an ninh, nhận diện đối tượng trong giao thông, và nhiều lĩnh vực khác.
- + **Khả năng tổng quát hóa tốt:** Hệ thống không có dấu hiệu quá khớp dữ liệu (overfitting), đảm bảo hiệu quả trên nhiều loại dữ liệu thử nghiệm khác nhau.

Tuy nhiên, hệ thống vẫn còn một số hạn chế:

- + Độ chính xác khi phát hiện các đối tượng nhỏ hoặc bị che khuất còn hạn chế.
- + Tốc độ hội tụ của mô hình chậm dần sau một số vòng lặp huấn luyện, điều này có thể được cải thiện bằng cách tối ưu siêu tham số hoặc sử dụng các mô hình tiên tiến hơn như YOLOv10 hoặc YOLOv11 ở hiện tại.

Hướng phát triển

Để nâng cao hơn nữa chất lượng và ứng dụng của hệ thống, các hướng phát triển sau đây được đề xuất:

- + **Sử dụng các mô hình mới hơn:** Áp dụng các phiên bản cải tiến như YOLOv10, YOLO v11 hoặc kết hợp với các mô hình tiên tiến khác như Transformer-based Models để mang đến sự đa năng và cải thiện độ chính xác và tốc độ.
- + **Tối ưu hóa hiệu suất:** Sử dụng kỹ thuật tăng tốc như TensorRT hoặc tích hợp phần cứng chuyên dụng (GPU, TPU) để cải thiện tốc độ xử lý thời gian thực.
- + **Cải thiện phát hiện đối tượng nhỏ:** Áp dụng các kỹ thuật như tăng cường dữ liệu (data augmentation) hoặc thay đổi kích thước lưới trong mô hình để phát hiện các đối tượng nhỏ hiệu quả hơn.
- + **Mở rộng ứng dụng:** Tích hợp hệ thống vào các lĩnh vực cụ thể như giao thông thông minh, theo dõi hành vi trong không gian công cộng, hoặc quản lý chuỗi cung ứng.

- + **Tăng cường dữ liệu huấn luyện:** Sử dụng các bộ dữ liệu phong phú và đa dạng hơn để cải thiện khả năng tổng quát của mô hình.
- + **Phát triển giao diện người dùng:** Hoàn thiện và tối ưu giao diện để người dùng dễ dàng thao tác và sử dụng hệ thống.

TÀI LIỆU THAM KHẢO

- [1] Analogy. Wikipedia, Mar 2018. 1 [2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303-338, 2010. 6 [3] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 3
- [2] (N.d.-b). Amazon.com. Retrieved December 7, 2024, from <https://aws.amazon.com/vi/what-is/python/#:~:text=Python%C3%A0%20m%E1%BB%99t%20ng%C3%B4n%20ng%E1%BB%AF,nhi%E1%BB%81u%20n%E1%BB%81n%20t%E1%BA%A3ng%20kh%C3%A1c%20nhau>.
- [3] Kim K. (2023, October 24). OpenCV là gì? Cách sử dụng OpenCV như thế nào. TEKLY - Học viện sáng tạo công nghệ; Tekly Academy. <https://teky.edu.vn/blog/opencv-la-gi/>
- [4] Tuấn H. (2024, July 4). Giới thiệu về thư viện PyQt6. Tìm ở đây | chia sẻ kiến thức IT; Tìm ở đây. <https://timoday.edu.vn/gioi-thieu-ve-thu-vien-pyqt6/>
- [5] Wikipedia contributors. (n.d.). PyTorch. Wikipedia, The Free Encyclopedia. <https://vi.wikipedia.org/w/index.php?title=PyTorch&oldid=71030073>
- [6] Blog, T. (2019, August 8). Thuật toán CNN là gì? Cấu trúc mạng Convolutional Neural Network. TopDev. <https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network>.
- [7] (N.d.-c). Amazon.com. Retrieved December 7, 2024, from <https://aws.amazon.com/vi/what-is/logistic-regression>.
- [8] Hàm mất mát (loss function). (n.d.). Gitbook.Io. Retrieved December 7, 2024, from <https://khanh-personal.gitbook.io/ml-book-vn/chapter1/ham-mat-mat>.