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 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH

Giảng viên hướng dẫn: Lương Thị Hồng Lan

TT	Mã sinh viên	Sinh viên thực hiện	Lớp hành chính
1	20210797	Lương Thị Lan Anh	DCCNTT12.10.3
2	20210719	Nguyễn Thế Mạnh	DCCNTT12.10.3
3	20210720	Bùi Quang Trung	DCCNTT12.10.3
4	20210663	Trần Quang Minh	DCCNTT12.10.3
5	20210712	Hồ Anh Nam	DCCNTT12.10.3

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 14: XÂY DỰNG HỆ THỐNG XÁC NHẬN ĐỐI TƯỢNG VÀ ĐẾM ĐỐI TƯỢNG TRONG ẢNH

Giảng viên hướng dẫn: Lương Thị Hồng Lan

TT	Mã sinh viên	Sinh viên thực hiện	Lớp hành chính
1	20210797	Lương Thị Lan Anh	DCCNTT12.10.3
2	20210719	Nguyễn Thế Mạnh	DCCNTT12.10.3
3	20210720	Bùi Quang Trung	DCCNTT12.10.3
4	20210663	Trần Quang Minh	DCCNTT12.10.3
5	20210712	Hồ Anh Nam	DCCNTT12.10.3

Bắc Ninh, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á

KỲ THI KẾT THÚC HỌC PHẦN

HỌC KỲ 1, NĂM HỌC 2024 – 2025

KHOA CÔNG NGHỆ THÔNG TIN

PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN

Mã đề thi: 14

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

Lớp Tín chỉ: XATGMT.03.K12.03.LH.C04.1_LT

Cán bộ chấm thi 1

(Ký và ghi rõ họ tên)

Cán bộ chấm thi 2

(Ký và ghi rõ họ tên)

Lương Thị Hồng Lan

TT	TIÊU CHÍ	THANG ĐIỂM	Lương Thị Lan Anh	Nguyễn Thế Mạnh	Bùi Quang Trung	Trần Quang Minh	Hồ Anh Nam
			20210797	20210719	20210720	20210663	20210712
1	**Nội dung báo cáo trên Word đầy đủ**	**3.5**					

TT	TIÊU CHÍ	THANG ĐIỂM	Lương Thị Lan Anh	Nguyễn Thế Mạnh	Bùi Quang Trung	Trần Quang Minh	Hồ Anh Nam
			20210797	20210719	20210720	20210663	20210712
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					
1.7	Tài liệu đầy đủ, bám sát yêu cầu của đề bài.	0,5					
2	**Nội dung thuyết trình đầy đủ**	**1.0**					

TT	TIÊU CHÍ	THANG ĐIỂM	Lương Thị Lan Anh	Nguyễn Thế Mạnh	Bùi Quang Trung	Trần Quang Minh	Hồ Anh Nam
			20210797	20210719	20210720	20210663	20210712
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					

TT	TIÊU CHÍ	THANG ĐIỂM	Lương Thị Lan Anh	Nguyễn Thế Mạnh	Bùi Quang Trung	Trần Quang Minh	Hồ Anh Nam
			20210797	20210719	20210720	20210663	20210712
3.6	Trả lời câu hỏi tự tin, thuyết phục.	0,5					
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 BẰNG SỐ:		**10**					
TỔNG ĐIỂM BẰNG CHỮ:		Mười tròn					

LỜI NÓI ĐẦU

Xác định và đếm đối tượng trong ảnh là một bài toán phức tạp, đòi hỏi việc kết hợp nhiều kỹ thuật xử lý ảnh và thị giác máy tính. Các yếu tố như độ phân giải, ánh sáng, góc chụp, sự đa dạng của đối tượng và nền ảnh đều ảnh hưởng đến độ chính xác của kết quả. Trong những năm gần đây, sự phát triển của học sâu và các mô hình mạng nơron đã mang lại những tiến bộ đáng kể trong việc giải quyết bài toán này. Tuy nhiên, vẫn còn nhiều thách thức cần được vượt qua, đặc biệt khi đối mặt với các hình ảnh phức tạp và các đối tượng nhỏ, chồng lấp.

Bài tập lớn này nhằm mục tiêu xây dựng một hệ thống hiệu quả để xác định và đếm các đối tượng trong ảnh, đặc biệt tập trung vào việc cải thiện độ chính xác khi đối mặt với các hình ảnh phức tạp. Để đạt được mục tiêu này, bài tập sẽ sử dụng các kỹ thuật học sâu, bao gồm mạng nơron convolutional (CNN) và mô hình Yolo để thực hiện đề tài "**Xây dựng hệ thống xác nhận đối tượng và đếm đối tượng trong ảnh".**

MỤC LỤC

LỜI CẢM ƠN

Trước hết, em xin bày tỏ tình cảm và lòng biết tới cô Lương Thị Hồng Lan, người đã giảng dạy học phần "xử lý ảnh và thị giác máy tính". Cảm ơn tới các thành viên trong nhóm: Lương Thị Lan Anh, Nguyễn Thế Mạnh, Bùi Quang Trung, Trần Quang Minh đã xây dựng, đóng góp ý kiến và công sức vào để hoàn thiện bài tập này. Tuy có nhiều cố gắng trong quá trình học tập, cũng như trong quá trình làm bài, song không thể tránh khỏi những thiếu sót, chúng em rất mong nhận sự góp ý quý báu của tất cả các thầy cô giáo cũng như tất cả các bạn để kết quả của chúng em được hoàn thiện tốt hơn. Một lần nữa chúng em xin chân thành cảm ơn!

DANH MỤC TỪ VIẾT TẮT

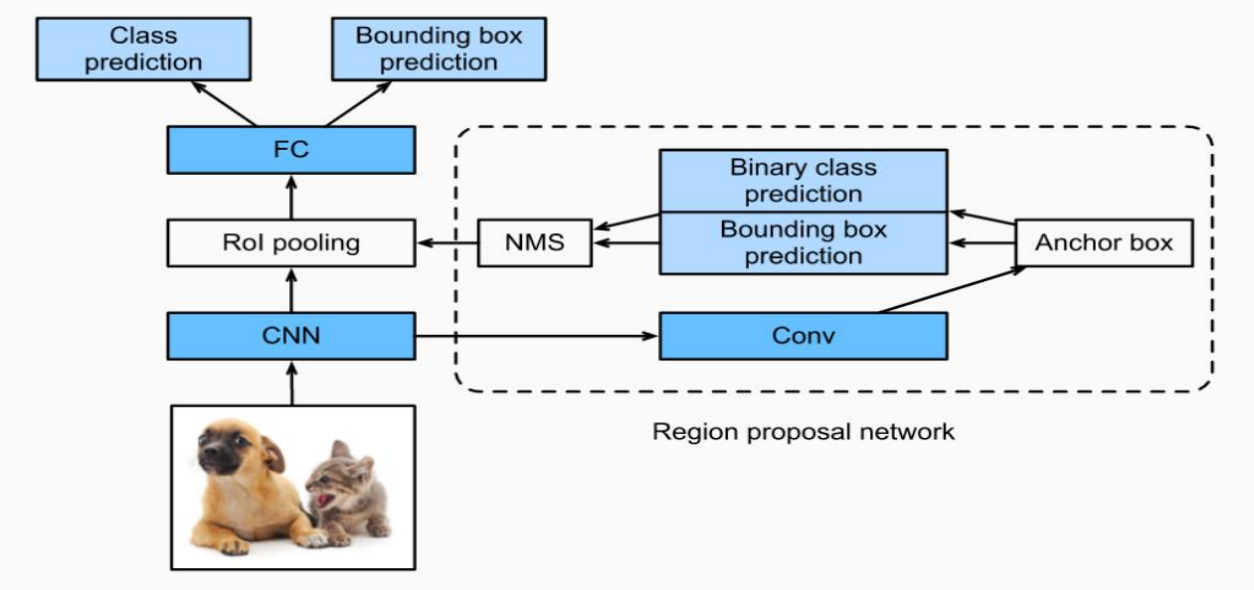
STT	Từ viết tắt	Giải thích
1	API	Application Programming Interface
2	CNN	Convolutional Neural Network
3	CNPM	Công nghệ phần mềm
4	COCO	Common Objects in Context
5	MHHM	Mô hình học máy
6	OpenCV	Open Source Computer Vision
7	Pillow (PIL)	Python Imaging Library
8	UI	Giao diện người dùng
9	UX	Trải nghiệm người dùng
10	Yolo	You Only Look Once

DANH MỤC HÌNH ẢNH

CHƯƠNG 1 CÁC KIẾN THỨC CƠ SỞ

1.1 Nhận dạng đối tượng

Bài toán nhận dạng là một lĩnh vực rộng lớn trong trí tuệ nhân tạo, bao gồm việc máy tính tự động phân loại, phân nhóm hoặc xác định các đối tượng, sự kiện, hoặc các mẫu trong dữ liệu [1]. Để hoàn thành một bài toán nhận dạng, chúng ta thường trải qua các bước sau:



Hình 1.1 1 Các giai đoạn cho bài toán nhận dạng

Bước 1. Thu thập và chuẩn bị dữ liệu

Thu thập: Tìm kiếm và thu thập một lượng lớn dữ liệu đại diện cho các đối tượng cần nhận dạng. Ví dụ: nếu muốn nhận dạng khuôn mặt, cần thu thập nhiều hình ảnh khuôn mặt khác nhau.

Làm sạch: Loại bỏ dữ liệu nhiễu, không chính xác hoặc trùng lặp.

Đánh nhãn: Gán nhãn cho từng dữ liệu để máy tính hiểu được nó thuộc loại nào. Ví dụ: trong nhận dạng khuôn mặt, mỗi hình ảnh sẽ được gán nhãn là "người A", "người B",...

Chia dữ liệu: Chia dữ liệu thành ba tập: tập huấn luyện (để huấn luyện mô hình), tập kiểm tra (để đánh giá mô hình trong quá trình huấn luyện), và tập kiểm định (để đánh giá mô hình cuối cùng).

Bước 2. Trích xuất đặc trưng

Chọn các đặc trưng: Lựa chọn những đặc trưng quan trọng nhất để phân biệt các đối tượng. Ví dụ: trong nhận dạng hình ảnh, các đặc trưng có thể là màu sắc, hình dạng…

Trích xuất: Sử dụng các thuật toán để trích xuất các đặc trưng từ dữ liệu thô.

Bước 3. Chọn mô hình

Lựa chọn: Chọn một mô hình phù hợp với bài toán. Các mô hình phổ biến bao gồm:

Mô hình dựa trên thống kê: Naive Bayes, SVM

Mạng thần kinh nhân tạo: CNN (Convolutional Neural Network), RNN (Recurrent Neural Network)

Mô hình dựa trên cây quyết định: Decision Tree, Random Forest

Huấn luyện: Dùng tập dữ liệu huấn luyện để huấn luyện mô hình. Trong quá trình này, mô hình sẽ học cách liên kết giữa các đặc trưng và nhãn.

Bước 4. Đánh giá mô hình:

Sử dụng tập kiểm tra: Đánh giá độ chính xác của mô hình trên tập kiểm tra.

Tính các chỉ số: Tính các chỉ số đánh giá như độ chính xác, độ nhạy, độ đặc hiệu,... để đánh giá hiệu suất của mô hình.

Bước 5. Điều chỉnh và tối ưu:

Điều chỉnh siêu tham số: Thay đổi các siêu tham số của mô hình để cải thiện hiệu suất.

Thay đổi mô hình: Nếu cần, có thể thay đổi mô hình hoặc phương pháp trích xuất đặc trưng.

Bước 6. Triển khai:

Tích hợp: Tích hợp mô hình vào ứng dụng thực tế.

Dự đoán: Sử dụng mô hình để dự đoán nhãn của dữ liệu mới.

Ví dụ: Nhận dạng hình ảnh

Thu thập: Thu thập hàng nghìn hình ảnh của các đối tượng khác nhau (mèo, chó, xe hơi).

Trích xuất: Sử dụng CNN để trích xuất các đặc trưng hình ảnh.

Huấn luyện: Huấn luyện mô hình để phân loại các hình ảnh.

Đánh giá: Đánh giá khả năng phân loại của mô hình trên tập kiểm tra.

Các ứng dụng của nhận dạng:

Xử lý ảnh và video: Nhận dạng khuôn mặt, vật thể, văn bản.

Xử lý ngôn ngữ tự nhiên: Nhận dạng giọng nói, dịch máy, phân loại văn bản.

Y tế: Phân tích hình ảnh y tế, dự đoán bệnh. [2]

1.2 Các kỹ thuật sử dụng trong bài toán nhận dạng

1.2.1 Học Có Giám sát (Supervised Learning)

Học có giám sát (Supervised Learning) là một nhánh quan trọng trong học máy, nơi mô hình học từ dữ liệu đã được gắn nhãn trước. Nghĩa là, mỗi dữ liệu đầu vào sẽ đi kèm với một nhãn tương ứng, cho biết dữ liệu đó thuộc lớp nào.

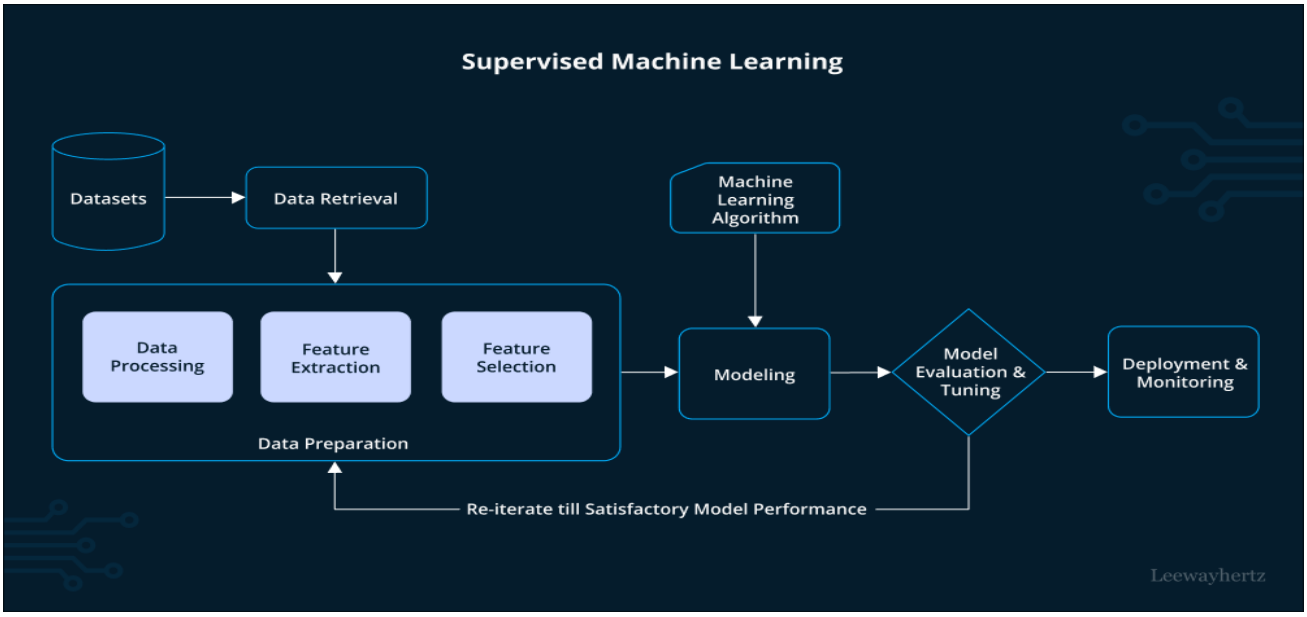
Tư tưởng cơ bản

Dữ liệu Đã Được Gắn Nhãn: Dữ liệu huấn luyện bao gồm các cặp (đầu vào, nhãn). Ví dụ, trong bài toán nhận dạng ảnh, đầu vào là một hình ảnh và nhãn là tên của đối tượng trong hình (mèo, chó, xe hơi).

Học Mối Liên Hệ: Mô hình học để tìm ra mối liên hệ giữa đầu vào và nhãn. Mối liên hệ này được biểu diễn dưới dạng một hàm toán học.

Dự Đoán: Sau khi được huấn luyện, mô hình có thể dự đoán nhãn của một dữ liệu đầu vào mới chưa từng gặp.

Hình mô tả kỹ thuật



Hình 1.2 1 Mô phỏng kỹ thuật học có giám sát

Các bước trong quá trình học có giám sát

1. Thu thập dữ liệu: Thu thập một lượng lớn dữ liệu đại diện cho bài toán cần giải quyết.
2. Đánh nhãn dữ liệu: Gán nhãn cho từng dữ liệu để mô hình biết được đâu là đầu vào và đâu là nhãn tương ứng.
3. Chọn mô hình: Lựa chọn một mô hình phù hợp với bài toán.
4. Huấn luyện mô hình: Dùng dữ liệu đã được đánh nhãn để huấn luyện mô hình. Trong quá trình này, mô hình sẽ điều chỉnh các tham số để giảm thiểu sai số giữa dự đoán và nhãn thực tế.
5. Đánh giá mô hình: Sử dụng một tập dữ liệu độc lập để đánh giá hiệu suất của mô hình.
6. Điều chỉnh mô hình (nếu cần): Nếu kết quả chưa đạt yêu cầu, có thể điều chỉnh các siêu tham số hoặc chọn mô hình khác.

Ưu điểm:

Độ chính xác cao khi có đủ dữ liệu chất lượng.

Có thể giải quyết nhiều loại bài toán khác nhau.

Nhược điểm:

Cần lượng lớn dữ liệu đã được đánh nhãn.

Khó áp dụng với các bài toán có ít dữ liệu hoặc dữ liệu không đồng nhất.

Các thuật toán phổ biến:

Hồi quy Logistic: Dùng để phân loại nhị phân.

Máy hỗ trợ vector (SVM): Hiệu quả với dữ liệu có chiều cao.

Mạng Neural: Đặc biệt là CNN (Convolutional Neural Networks) cho bài toán nhận dạng hình ảnh và RNN (Recurrent Neural Networks) cho bài toán xử lý ngôn ngữ tự nhiên. [3]

1.2.2 Học Không Giám sát (Unsupervised Learning)

Học không giám sát (Unsupervised Learning) là một nhánh của học máy nơi mô hình học từ dữ liệu không được gắn nhãn. Khác với học có giám sát, ở đây chúng ta không biết trước các nhãn tương ứng với mỗi dữ liệu. Thay vào đó, mô hình tự tìm kiếm các cấu trúc ẩn, các mẫu trong dữ liệu để đưa ra những kết luận nhất định.

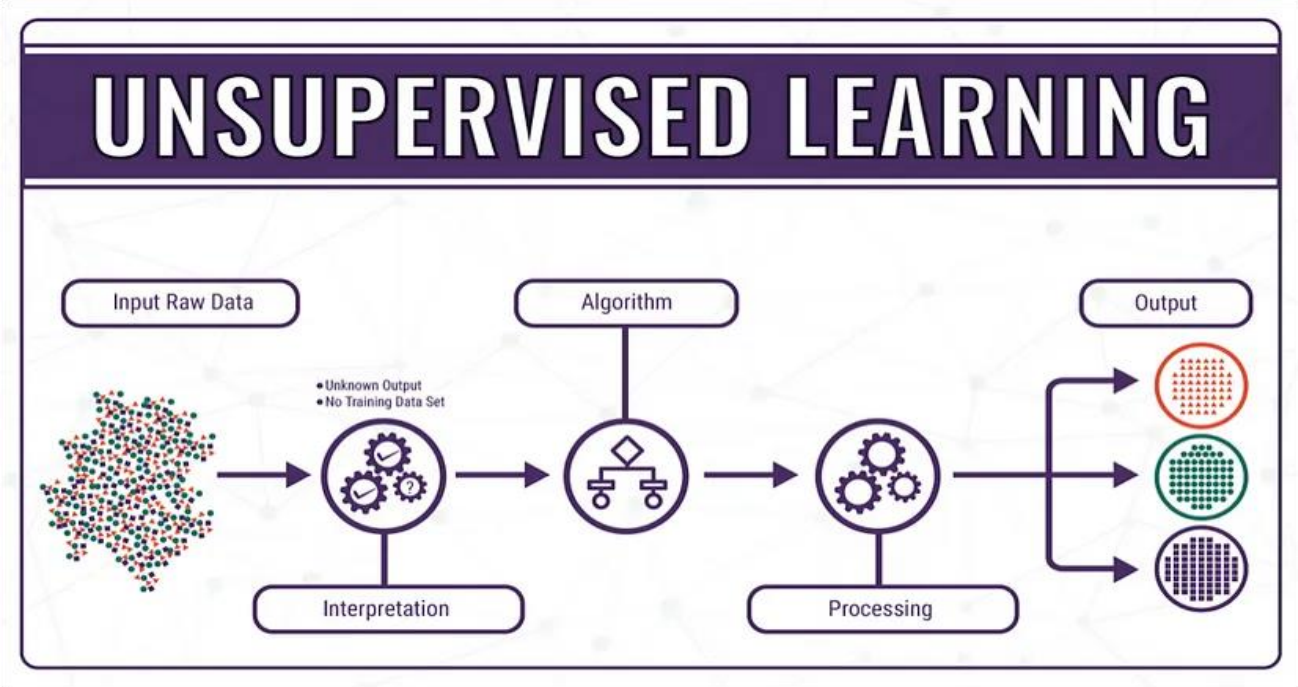
Tư tưởng cơ bản

Dữ liệu không gắn nhãn: Dữ liệu đầu vào không có nhãn tương ứng.

Tìm kiếm cấu trúc ẩn: Mô hình tự tìm kiếm các nhóm, các cụm hoặc các quy tắc ẩn trong dữ liệu.

Không có đáp án đúng: Không có một đáp án đúng duy nhất cho một bài toán học không giám sát, kết quả có thể phụ thuộc vào thuật toán và các tham số được chọn.

Hình mô tả kỹ thuật



Hình 1.2 2 Mô phỏng kỹ thuật học không giám sát

Các bước trong quá trình học không giám sát:

1. Thu thập dữ liệu: Thu thập một lượng lớn dữ liệu.
2. Chọn mô hình: Lựa chọn một mô hình phù hợp với bài toán.
3. Huấn luyện mô hình: Dùng dữ liệu để huấn luyện mô hình. Trong quá trình này, mô hình sẽ tìm kiếm các cấu trúc ẩn trong dữ liệu.
4. Đánh giá mô hình: Đánh giá kết quả dựa trên các tiêu chí như độ chặt chẽ của các cụm, khả năng tổng quát hóa.

Ưu điểm:

Không cần dữ liệu đánh nhãn.

Có thể phát hiện các mẫu ẩn trong dữ liệu.

Nhược điểm:

Kết quả khó diễn giải.

Khó đánh giá chất lượng mô hình.

Các thuật toán phổ biến:

Phân cụm (Clustering): K-means, Hierarchical Clustering

Giảm chiều (Dimensionality Reduction): PCA, t-SNE [4]

1.2.3 Học Tăng Cường (Reinforcement Learning)

Học tăng cường (Reinforcement Learning - RL) là một nhánh của học máy, mô phỏng quá trình học hỏi của con người thông qua thử và sai. Trong RL, một tác nhân (agent) sẽ tương tác với một môi trường (environment) và học cách thực hiện các hành động để tối đa hóa phần thưởng (reward) nhận được.

Tư tưởng cơ bản

Tác nhân (Agent): Đây là thực thể đưa ra quyết định, ví dụ: một robot, một chương trình máy tính chơi game.

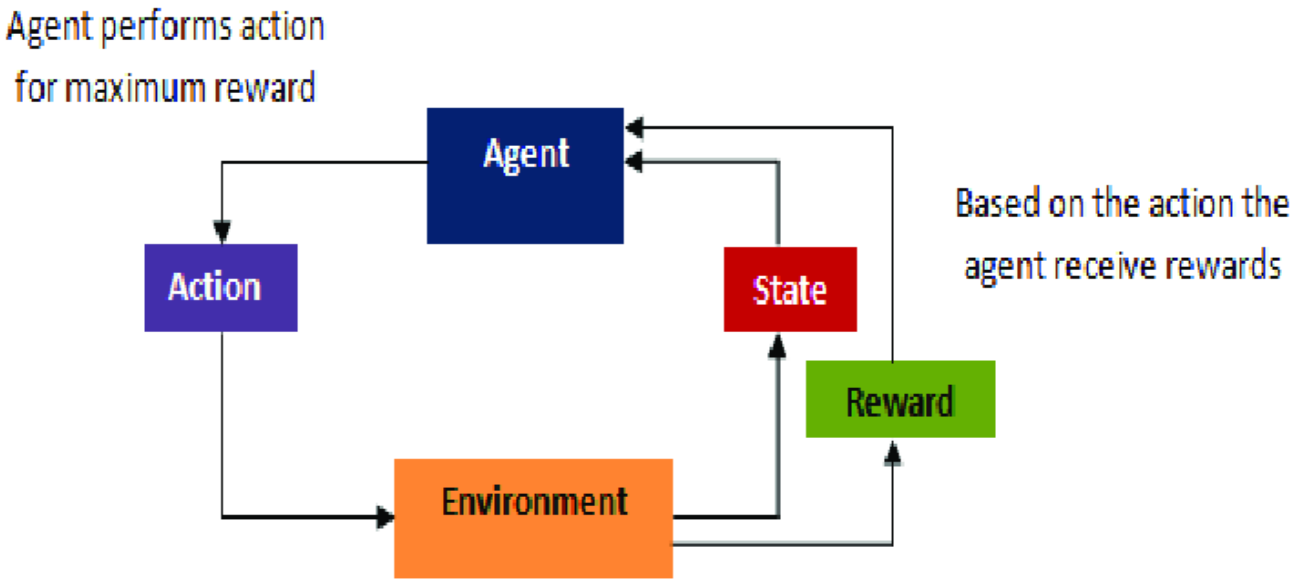
Môi trường (Environment): Là thế giới mà tác nhân tương tác, có thể là một trò chơi, một robot mô phỏng, hoặc một hệ thống thực tế.

Hành động (Action): Các lựa chọn mà tác nhân có thể thực hiện trong một trạng thái nhất định.

Trạng thái (State): Mô tả tình huống hiện tại của môi trường.

Phần thưởng (Reward): Một tín hiệu số cho biết hành động của tác nhân là tốt hay xấu. Mục tiêu của tác nhân là tối đa hóa tổng phần thưởng nhận được trong dài hạn.

Hình mô tả kỹ thuật



Hình 1.2 3 Mô phỏng kỹ thuật học tăng cường

Các bước trong quá trình học tăng cường

1. Khởi tạo: Tác nhân bắt đầu ở một trạng thái ban đầu.
2. Chọn hành động: Tác nhân chọn một hành động dựa trên chính sách hiện tại (policy).
3. Thực hiện hành động: Môi trường thay đổi trạng thái và trả về một phần thưởng.
4. Cập nhật chính sách: Tác nhân cập nhật chính sách dựa trên phần thưởng nhận được.
5. Lặp lại: Quá trình trên được lặp lại nhiều lần cho đến khi tác nhân đạt được hiệu suất mong muốn.

Ưu điểm:

Có thể học các nhiệm vụ phức tạp.

Không cần dữ liệu đánh nhãn.

Nhược điểm:

Cần thiết kế môi trường học tập phù hợp.

Quá trình học thường chậm và đòi hỏi nhiều tính toán. [5]

1.2.4 Học sâu (Deep Learning)

Học sâu là một nhánh của học máy, sử dụng các mạng thần kinh nhân tạo nhiều lớp để học trực tiếp từ dữ liệu, thường là dữ liệu không có cấu trúc như hình ảnh, âm thanh và văn bản. Nó được lấy cảm hứng từ cách hoạt động của não người, nơi thông tin được xử lý qua nhiều lớp neuron.

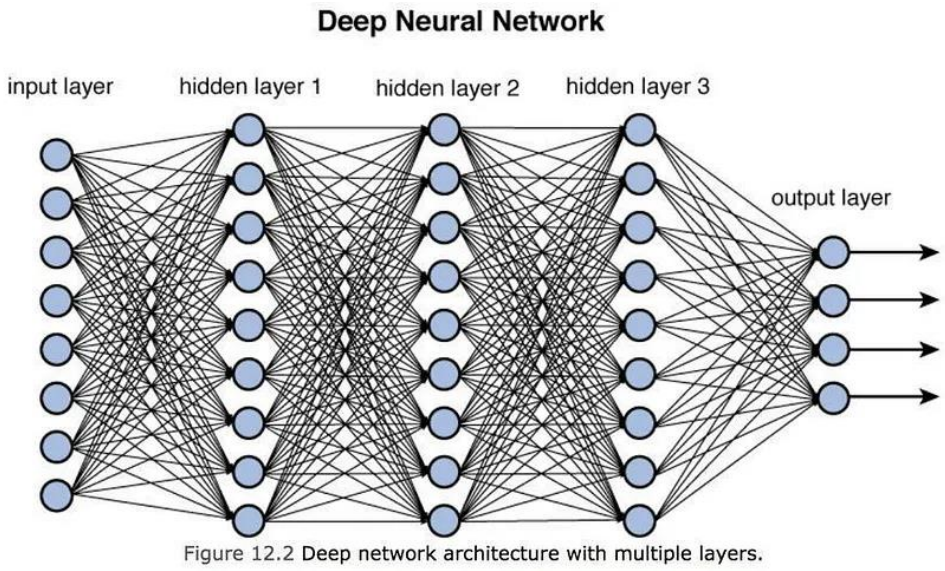
Tư tưởng cơ bản

Mạng thần kinh nhân tạo: Là một mô hình tính toán mô phỏng các neuron trong não người. Các neuron này được kết nối với nhau tạo thành các lớp, và thông tin được truyền đi qua các lớp này.

Học các đặc trưng: Thay vì người lập trình phải xác định thủ công các đặc trưng, mạng thần kinh học sâu tự động học các đặc trưng từ dữ liệu. Các lớp đầu tiên học các đặc trưng đơn giản, các lớp sau học các đặc trưng phức tạp hơn.

Học bằng cách backpropagation: Quá trình huấn luyện mạng thần kinh bao gồm việc so sánh kết quả dự đoán của mạng với nhãn thực tế, tính toán lỗi và điều chỉnh các trọng số của mạng để giảm thiểu lỗi. Quá trình này được lặp đi lặp lại nhiều lần.

Hình mô tả kỹ thuật



Hình 1.2 4 Mô phỏng kỹ thuật học sâu

Phân tích

- Các lớp (layers): Mỗi lớp gồm nhiều neuron. Thông tin được truyền từ lớp này sang lớp khác.
- Neuron: Mỗi neuron thực hiện một phép tính đơn giản và truyền kết quả đến các neuron ở lớp tiếp theo.
- Trọng số (weights): Các kết nối giữa các neuron có các trọng số. Trọng số này được điều chỉnh trong quá trình huấn luyện.
- Biến thiên (bias): Mỗi neuron có một biến thiên, giúp điều chỉnh đầu ra của neuron.
- Hàm kích hoạt: Mỗi neuron có một hàm kích hoạt, giúp giới hạn đầu ra của neuron.

Ưu Điểm

Hiệu suất cao: Học sâu đạt được độ chính xác cao trong nhiều bài toán, đặc biệt là các bài toán liên quan đến dữ liệu không có cấu trúc.

Tự động hóa: Học sâu tự động trích xuất các đặc trưng, giảm thiểu sự can thiệp của con người.

Ứng dụng rộng rãi: Học sâu được áp dụng trong nhiều lĩnh vực như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, thị giác máy tính, và nhiều lĩnh vực khác.

Nhược Điểm

Yêu cầu lượng dữ liệu lớn: Học sâu cần một lượng lớn dữ liệu để huấn luyện.

Thời gian huấn luyện lâu: Huấn luyện các mạng thần kinh sâu có thể mất nhiều thời gian và tài nguyên tính toán.

Hộp đen: Quá trình ra quyết định của các mạng thần kinh sâu thường khó giải thích, gây khó khăn trong việc kiểm chứng và tin tưởng vào kết quả.

Overfitting: Mô hình học sâu có thể quá khớp với dữ liệu huấn luyện, dẫn đến hiệu suất kém trên dữ liệu kiểm tra. [6]

1.2.5 Các kỹ thuật khác

Transfer Learning: Tận dụng kiến thức từ một mô hình đã được huấn luyện trên một nhiệm vụ để giải quyết một nhiệm vụ khác.

Ensemble Learning: Kết hợp nhiều mô hình để cải thiện hiệu suất. [7]

1.3 Ngôn ngữ lập trình và các thư viện 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. [8]

Những lợi ích của Python bao gồm:

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
- 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.
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
- Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix. [9]

1.3.2. Các thư viện

1.3.2.1 OpenCV (cv2)

Thư viện mạnh mẽ dành cho xử lý ảnh và video.

Dùng để đọc, xử lý, và phân tích hình ảnh/video, hỗ trợ tích hợp mô hình học sâu (Deep Learning).

Ví dụ trong mã nguồn: Dùng cv2.dnn.readNet để tải mô hình YOLOv3 với tệp cấu hình và trọng số. [10]

1.3.2.2 NumPy (np)

Thư viện cho xử lý mảng và ma trận, cung cấp các phép toán toán học cấp cao.

Vai trò: Hỗ trợ xử lý dữ liệu hình ảnh, đặc biệt là định dạng dữ liệu đầu vào của mô hình YOLOv3. [11]

1.3.2.3 Tkinter

Bộ công cụ giao diện người dùng (GUI) tiêu chuẩn của Python.

Dùng để xây dựng giao diện tương tác như chọn tệp hình ảnh hoặc hiển thị kết quả.

Các thành phần sử dụng:

filedialog: Hỗ trợ mở hộp thoại chọn tệp.

messagebox: Hiển thị thông báo hoặc cảnh báo tới người dùng. [12]

1.3.2.4 Pillow (PIL)

Thư viện mạnh mẽ dùng để xử lý hình ảnh, hỗ trợ chuyển đổi định dạng, chỉnh sửa kích thước và hiển thị hình ảnh trong ứng dụng GUI.

Ứng dụng trong mã:

Image.open: Mở tệp hình ảnh.

ImageTk.PhotoImage: Hiển thị ảnh trong ứng dụng Tkinter. [13]

1.3.2.5 YOLOv3 (Darknet)

YOLOv3 sử dụng kiến trúc Darknet-53 làm mạng nền tảng (backbone):

Darknet-53 bao gồm 53 lớp convolutional, được xây dựng trên các khối residual (Residual Blocks) tương tự như ResNet.

Kết hợp các kỹ thuật như skip connections giúp cải thiện khả năng huấn luyện và giảm gradient vanishing. Hỗ trợ dự đoán đối tượng ở 3 cấp độ khác nhau (multi-scale prediction), cho phép phát hiện tốt các đối tượng có kích thước khác nhau.[14]

CHƯƠNG 2 XÂY DỰNG HỆ THỐNG

2.1 Mô tả bài toán

Bài toán "Xây dựng hệ thống xác nhận đối tượng và đếm đối tượng trong ảnh" đặt ra mục tiêu xây dựng một hệ thống thông minh, có khả năng tự động nhận diện và đếm các đối tượng cụ thể trong một bức ảnh. Hệ thống này sẽ dựa trên mô hình học sâu YOLO (You Only Look Once), một trong những thuật toán phát hiện đối tượng nhanh và hiệu quả nhất hiện nay.

Các bước thực hiện bài toán

Thu thập và chuẩn bị dữ liệu:

- Thu thập ảnh: Tập trung thu thập một lượng lớn ảnh đa dạng, bao gồm nhiều góc độ, điều kiện ánh sáng khác nhau và các đối tượng cần nhận diện xuất hiện ở nhiều vị trí và kích thước.
- Đánh dấu dữ liệu (annotation): Mỗi đối tượng trong ảnh cần được đánh dấu bằng bounding box (khung bao) và gán nhãn tương ứng. Việc này giúp mô hình học cách liên kết các đặc trưng hình ảnh với các đối tượng cụ thể.
- Chia tập dữ liệu: Chia tập dữ liệu thành ba phần: tập huấn luyện, tập kiểm định và tập thử nghiệm để đánh giá hiệu suất của mô hình.

Xây dựng mô hình YOLO

- Chọn kiến trúc: Lựa chọn phiên bản YOLO phù hợp (YOLOv5, YOLOv8,...) dựa trên độ phức tạp của bài toán, yêu cầu về tốc độ và độ chính xác.
- Cấu hình mô hình: Điều chỉnh các hyperparameter như số lớp, kích thước ảnh đầu vào, learning rate,... để tối ưu hóa hiệu suất.

Huấn luyện mô hình

- Đưa dữ liệu vào mô hình: Cho mô hình học từ tập dữ liệu đã đánh dấu.
- Tối ưu hóa: Sử dụng thuật toán tối ưu hóa (ví dụ: Stochastic Gradient Descent) để tìm các tham số tốt nhất cho mô hình.

- Đánh giá: Đánh giá hiệu suất của mô hình trên tập kiểm định bằng các chỉ số như mAP (mean Average Precision), độ chính xác, độ nhạy,...

Triển khai hệ thống

- Tích hợp mô hình vào ứng dụng: Nhúng mô hình đã huấn luyện vào một ứng dụng (ví dụ: ứng dụng di động, web app) để người dùng có thể sử dụng.
- Xử lý ảnh đầu vào: Tiền xử lý ảnh đầu vào (resize, normalization) để phù hợp với yêu cầu của mô hình.
- Phát hiện và đếm đối tượng: Cho mô hình dự đoán các bounding box và lớp của các đối tượng trong ảnh.
- Hiển thị kết quả: Hiển thị các bounding box và nhãn của các đối tượng lên ảnh gốc.

2.2 Xây dựng hệ thống

2.2.1 Mô hình Yolo

YOLO (You Only Look Once) là một trong những mô hình hàng đầu cho bài toán nhận dạng và phát hiện đối tượng. Nó tích hợp nhiều kỹ thuật hiệu quả để tăng tốc độ và độ chính xác. Dưới đây là các kỹ thuật chính thường sử dụng trong mô hình YOLO, bao gồm tư tưởng kỹ thuật, cùng ưu nhược điểm. [15]

2.2.1.1 Grid-based Detection

Tư tưởng kỹ thuật: YOLO chia ảnh thành một lưới (grid) S x S. Mỗi ô lưới chịu trách nhiệm dự đoán các bounding boxes và xác suất tồn tại của đối tượng trong vùng đó.

Ưu điểm:

Phát hiện nhanh do chỉ cần xử lý ảnh một lần qua mạng.

Hệ thống duy nhất thực hiện cả phát hiện và phân loại đối tượng.

Nhược điểm:

Hạn chế trong việc xử lý các đối tượng nhỏ, đặc biệt khi nằm ở giữa các ô lưới.

2.2.1.2 Anchor Boxes

Tư tưởng kỹ thuật: YOLO sử dụng các anchor boxes (hộp neo) có kích thước và tỷ lệ khác nhau để phát hiện các đối tượng có hình dạng đa dạng. Các bounding boxes dự đoán sẽ được căn chỉnh với các hộp neo này.

Ưu điểm:

Hỗ trợ phát hiện các đối tượng có kích thước và tỷ lệ khác nhau.

Tăng khả năng phát hiện các đối tượng chồng chéo.

Nhược điểm:

Có thể tăng độ phức tạp tính toán nếu số lượng anchor boxes lớn.

2.2.1.3 Non-Maximum Suppression (NMS)

Tư tưởng kỹ thuật: Sau khi phát hiện nhiều bounding boxes cho cùng một đối tượng, YOLO sử dụng NMS để loại bỏ các bounding boxes không tối ưu, chỉ giữ lại bounding box có độ tin cậy cao nhất.

Ưu điểm:

Loại bỏ trùng lặp, đảm bảo mỗi đối tượng chỉ được phát hiện một lần.

Tăng độ chính xác đầu ra.

Nhược điểm:

Hiệu quả phụ thuộc vào ngưỡng confidence và IoU (Intersection over Union).

2.2.1.4 Feature Pyramid Network (FPN)

Tư tưởng kỹ thuật: FPN giúp YOLO phát hiện các đối tượng có kích thước khác nhau bằng cách kết hợp thông tin từ các tầng đặc trưng khác nhau trong mạng học sâu.

Ưu điểm:

Cải thiện khả năng phát hiện các đối tượng nhỏ và lớn.

Tăng cường hiệu quả cho các ảnh có đối tượng đa kích thước.

Nhược điểm:

Tăng thêm độ phức tạp của mô hình và thời gian xử lý.

2.2.1.5 Objectness Score

Tư tưởng kỹ thuật: YOLO dự đoán một objectness score cho mỗi bounding box, đại diện cho mức độ tin cậy rằng hộp này chứa một đối tượng.

Ưu điểm:

Giảm số lượng dự đoán sai.

Đơn giản hóa quá trình lọc bounding box không phù hợp.

Nhược điểm:

Độ chính xác phụ thuộc vào việc định nghĩa ngưỡng score phù hợp.

2.2.1.6 Intersection over Union (IoU)

Tư tưởng kỹ thuật: IoU đo lường mức độ trùng khớp giữa bounding box được dự đoán và bounding box thật. Giá trị IoU càng cao thì dự đoán càng chính xác.

Ưu điểm:

Đảm bảo đánh giá chính xác độ khớp giữa dự đoán và nhãn thực tế.

Hỗ trợ NMS hiệu quả hơn.

Nhược điểm:

Có thể không tốt cho các đối tượng rất nhỏ hoặc quá chồng chéo.

2.2.1.7 Data Augmentation

Tư tưởng kỹ thuật: Kỹ thuật tăng cường dữ liệu bằng cách xoay, cắt, phóng to, thu nhỏ, hoặc thay đổi màu sắc để tăng tính đa dạng của dữ liệu huấn luyện.

Ưu điểm:

Tăng tính tổng quát của mô hình.

Giảm nguy cơ overfitting.

Nhược điểm:

Nếu dữ liệu tăng cường không hợp lý, có thể làm giảm hiệu quả mô hình.

2.2.1.8 Loss Function

Tư tưởng kỹ thuật: YOLO sử dụng hàm mất mát tổng hợp gồm ba thành phần:

Localization Loss: Đánh giá sai lệch giữa bounding box dự đoán và bounding box thật.

Confidence Loss: Đánh giá độ chính xác của objectness score.

Class Prediction Loss: Đánh giá độ chính xác của dự đoán nhãn lớp.

Ưu điểm:

Kết hợp đa mục tiêu (vị trí, độ tin cậy, phân loại) để tối ưu hóa toàn diện.

Giảm mất mát cho các bounding boxes không chứa đối tượng (negative samples).

Nhược điểm:

Việc cân bằng trọng số giữa các loại mất mát cần điều chỉnh cẩn thận.

2.2.2 Các bước thực hiện bài toán với mô hình Yolo

2.2.2.1 Thu thập dữ liệu

Đánh giá bộ dữ liệu

Nhóm sử dụng bộ dữ liệu được lấy từ Common Objects in Context, click vào link để xem https://cocodataset.org/. Bộ dữ liệu COCO (Common Objects in Context) là bộ dữ liệu phát hiện, phân đoạn và chú thích đối tượng quy mô lớn. 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**.**

Gắn nhãn (Labeling)

Mỗi ảnh trong bộ dữ liệu cần được gắn nhãn chính xác để mô hình có thể học được các đặc điểm của các đối tượng trong ảnh. Quá trình gắn nhãn yêu cầu phải tạo ra các bounding boxes (hộp bao quanh) cho các đối tượng, xác định vị trí của chúng trong ảnh và phân loại các đối tượng đó.

Các công cụ phổ biến để gắn nhãn ảnh bao gồm:

LabelImg: Là một công cụ mã nguồn mở, dễ sử dụng để gắn nhãn cho ảnh.

VGG Image Annotator (VIA): Một công cụ gắn nhãn trực tuyến cho phép chúng ta dễ dàng tạo tệp nhãn cho ảnh.

2.2.2.2 Tiền xử lý dữ liệu

Chuyển đổi dữ liệu:

Sau khi thu thập và gắn nhãn dữ liệu, cần chuẩn bị dữ liệu để huấn luyện mô hình YOLO. Các bước tiền xử lý bao gồm:

Thay đổi kích thước ảnh: YOLO yêu cầu ảnh đầu vào có kích thước cố định (ví dụ: 416x416 hoặc 608x608). Điều này giúp mô hình dễ dàng xử lý ảnh và giảm thiểu sự phức tạp trong quá trình huấn luyện.

Chuyển đổi nhãn thành định dạng YOLO: Mỗi ảnh sẽ có một tệp nhãn riêng, trong đó chứa thông tin về các lớp đối tượng và vị trí bounding box. Định dạng nhãn YOLO bao gồm lớp đối tượng và các tọa độ của bounding box dưới dạng tỉ lệ so với kích thước ảnh (x_center, y_center, width, height).

Tăng cường dữ liệu (Data Augmentation)

Tăng cường dữ liệu là một kỹ thuật quan trọng giúp cải thiện khả năng tổng quát của mô hình và giảm thiểu overfitting. Các kỹ thuật tăng cường dữ liệu phổ biến cho nhận diện đối tượng bao gồm:

Lật ảnh (Flipping): Lật ảnh theo chiều ngang hoặc dọc.

Xoay ảnh (Rotation): Xoay ảnh một góc ngẫu nhiên.

Thay đổi độ sáng và độ tương phản: Điều chỉnh độ sáng, độ tương phản của ảnh.

Thêm nhiễu: Thêm nhiễu ngẫu nhiên vào ảnh để mô phỏng các tình huống thực tế.

Các thư viện như OpenCV và Albumentations có thể giúp ta thực hiện các bước tăng cường dữ liệu này một cách dễ dàng.

2.2.2.3 Xây dựng mô hình và huấn luyện

Cài đặt và cấu hình mô hình YOLO

Trước khi huấn luyện, cần phải cài đặt và cấu hình mô hình YOLO. Các bản YOLO phổ biến có thể sử dụng bao gồm:

Darknet: Đây là bản YOLO gốc được phát triển bởi Joseph Redmon. Darknet là một framework nhẹ và nhanh.

Ultralytics YOLOv3: Đây là bản YOLO dễ sử dụng hơn và hỗ trợ nhiều tính năng hữu ích.

Cấu hình các tham số mô hình

Cấu hình tệp YOLO: Tệp cấu hình chứa các tham số cần thiết như số lớp đối tượng (classes), số anchor boxes, kích thước ảnh đầu vào, và các tham số khác.

Tạo tệp nhãn: Tệp nhãn chứa thông tin về các đối tượng trong mỗi ảnh, bao gồm lớp và tọa độ của bounding box.

Huấn luyện mô hình

Chọn hàm mất mát (Loss function): YOLO sử dụng một hàm mất mát kết hợp, bao gồm:

- Mất mát cho vị trí bounding box.
- Mất mát cho sự xác suất của đối tượng.
- Mất mát cho phân loại đối tượng.

Chọn optimizer: Thường sử dụng Adam hoặc SGD.Huấn luyện mô hình: Quá trình huấn luyện mô hình bao gồm việc truyền dữ liệu qua mạng, tính toán độ mất mát, và cập nhật các trọng số của mô hình thông qua thuật toán lan truyền ngược (backpropagation).

Lưu mô hình

Sau khi huấn luyện xong, mô hình và các trọng số sẽ được lưu vào một tệp để có thể sử dụng lại trong các ứng dụng nhận diện đối tượng sau này.

2.2.2.4 Triển khai và sử dụng mô hình

Tải mô hình đã huấn luyện

Sau khi hoàn tất quá trình huấn luyện, mô hình YOLO sẽ được lưu lại dưới dạng một tệp trọng số (weights file). Cần tải tệp này vào bộ nhớ để sử dụng cho các dự đoán.

Dự đoán các đối tượng

Mô hình YOLO sẽ nhận đầu vào là một bức ảnh và phân tích toàn bộ ảnh để xác định vị trí của các đối tượng. Đầu ra của mô hình sẽ là một tập hợp các bounding boxes, mỗi bounding box bao quanh một đối tượng được phát hiện, cùng với nhãn và xác suất tương ứng.

Non-Maximum Suppression (NMS): Đây là một kỹ thuật hậu xử lý được sử dụng để loại bỏ các bounding boxes trùng lặp. Nếu nhiều bounding boxes dự đoán cho cùng một đối tượng, NMS sẽ giữ lại hộp bao quanh chính xác nhất.

Hiển thị kết quả

Sau khi nhận diện, bạn có thể vẽ các bounding boxes lên ảnh và hiển thị kết quả nhận diện hoặc lưu lại ảnh đã được nhận diện.

CHƯƠNG 3 KẾT QUẢ THỰC NGHIỆM

3.1 Dữ liệu

3.1.1 Mô tả bộ dữ liệu

Nhóm sử dụng bộ dữ liệu được lấy từ Common Objects in Context, click vào link để xem https://cocodataset.org/. Bộ dữ liệu COCO (Common Objects in Context) là bộ dữ liệu phát hiện, phân đoạn và chú thích đối tượng quy mô lớn. 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**.** Một số dữ kiện liên quan tới bộ dữ liệu COCO:

Tệp "coco.names"

Chứa danh sách các lớp đối tượng mà mô hình YOLO có thể nhận diện.

Các lớp này bao gồm những đối tượng thường gặp như: người (person), xe hơi (car), chó (dog), bàn (table), v.v.

File này được tải vào chương trình và lưu trong biến classes.Tệp "yolov3.weights" và "yolov3.cfg":

yolov3.weights: File chứa trọng số đã được huấn luyện của mô hình YOLOv3. Trọng số này được huấn luyện trên bộ dữ liệu COCO.

yolov3.cfg: File cấu hình kiến trúc mạng YOLOv3, xác định số lượng lớp, kích thước ảnh đầu vào, số lớp cuối, v.v.

Dữ liệu gốc:

Bộ dữ liệu COCO bao gồm hơn 330.000 ảnh với hơn 80 danh mục đối tượng và hàng triệu annotations (đánh dấu vị trí đối tượng). Bộ dữ liệu này là chuẩn mực cho các mô hình thị giác máy tính như YOLO.

Phạm vi nhận diện:

Với file coco.names, mô hình chỉ nhận diện được 80 loại đối tượng được định nghĩa sẵn trong bộ dữ liệu COCO.

Bộ dữ liệu COCO (Common Objects in Context). Đây là một bộ dữ liệu phổ biến cho các bài toán nhận diện và phân loại đối tượng. Thông tin chi tiết như sau:

PASCAL VOC: Một bộ dữ liệu tiêu chuẩn khác trong lĩnh vực nhận diện đối tượng, bao gồm 20 lớp đối tượng, từ người, động vật đến các vật thể hàng ngày như xe, máy bay, và ghế.

Bộ dữ liệu tùy chỉnh: Nếu yêu cầu nhận diện các đối tượng đặc biệt không có trong các bộ dữ liệu chuẩn, ta có thể tự tạo bộ dữ liệu riêng. Việc này đòi hỏi phải thu thập ảnh từ các nguồn khác nhau và gắn nhãn thủ công cho từng đối tượng trong ảnh.

3.1.2 Xử lý dữ liệu

Tiền xử lý dữ liệu nếu huấn luyện lại YOLOv3, nếu muốn huấn luyện lại YOLOv3 trên tập dữ liệu riêng, cần thực hiện các bước tiền xử lý dữ liệu sau:

Bước 1: Chuyển đổi định dạng nhãn

YOLO yêu cầu nhãn dữ liệu phải ở định dạng .txt, trong đó mỗi dòng có cấu trúc:

<class_id> <x_center> <y_center> <width> <height>

Giải thích:

<class_id>: ID của lớp đối tượng (số nguyên).

<x_center> và <y_center>: Tọa độ tâm của bounding box, được chuẩn hóa (0-1).

<width> và <height>: Chiều rộng và chiều cao của bounding box, cũng được chuẩn hóa.

Bước 2: Kỹ thuật tăng cường dữ liệu (Data Augmentation)

Tăng cường dữ liệu giúp cải thiện khả năng tổng quát hóa của mô hình. Các kỹ thuật bao gồm:

Thay đổi độ sáng, độ tương phản, và màu sắc: Làm tăng tính đa dạng của tập dữ liệu để mô hình hoạt động tốt trong các điều kiện ánh sáng khác nhau.

Xoay, lật, và cắt ảnh (Rotation, Flip, Crop): Giúp mô hình không bị lệ thuộc vào hướng hoặc vị trí cố định của đối tượng.

Thêm nhiễu (Noise) hoặc làm mờ ảnh (Blur): Làm mô hình mạnh mẽ hơn khi gặp dữ liệu thật có chất lượng kém.

Bước 3: Chuyển đổi kích thước ảnh

Tất cả ảnh đầu vào cần được chuyển về kích thước cố định (thường là 416x416 pixels) để đảm bảo tương thích với kiến trúc YOLOv3.

Việc resize ảnh có thể được thực hiện bằng thư viện xử lý ảnh như OpenCV hoặc Pillow (PIL).

Kết quả sau tiền xử lý

Tập dữ liệu đã chuẩn hóa và tăng cường sẽ sẵn sàng để chia thành tập train và test cho quá trình huấn luyện mô hình.

Cách lấy dữ liệu vào hệ thống

Dữ liệu đầu vào:

Ảnh được chọn từ giao diện.

Mô hình YOLO (yolov3.weights và yolov3.cfg) và danh sách các nhãn (coco.names).

Chuẩn bị ảnh đầu vào:

Ảnh được chuyển đổi thành blob (thông qua cv2.dnn.blobFromImage) để làm đầu vào cho mô hình YOLO.

Truyền qua mạng nơ-ron:

Blob được truyền vào mô hình (net.setInput(blob)).

Mạng dự đoán kết quả tại các lớp đầu ra không kết nối (net.forward (output_layers_names)).

Xử lý kết quả đầu ra:

Với mỗi kết quả từ lớp đầu ra, xác định các thông tin:

Confidence (độ tự tin): Giá trị dự đoán của mô hình về độ chắc chắn rằng một vùng là đối tượng.

Bounding box (khung chứa): Tọa độ và kích thước của khung bao đối tượng trong ảnh.

Class ID (mã lớp): Xác định đối tượng thuộc lớp nào trong danh sách coco.names.

Loại bỏ các phát hiện không cần thiết:

Chỉ giữ lại các phát hiện với confidence > 0.2.

Sử dụng Non-Maximum Suppression (NMS) để loại bỏ các khung bao chồng chéo.

Cách đo lường đối tượng

Tổng số đối tượng được đếm: a trên số lượng khung bao còn lại sau NMS (len(indexes.flatten())).

Độ tin cậy của từng đối tượng: Hiển thị confidence của từng đối tượng được phát hiện.

Hiển thị kết quả:

Vẽ khung bao và tên đối tượng lên ảnh (cv2.rectangle, cv2.putText).

Cập nhật danh sách đối tượng và tổng số lượng đối tượng trên giao diện.

3.1.3 Chia train-test của bộ dữ liệu

Chia dữ liệu Train và Test: Để huấn luyện mô hình từ đầu hoặc fine-tune mô hình YOLOv3, cần chia dữ liệu thành các phần:

Tỷ lệ chia:

80% cho tập huấn luyện (train).

20% cho tập kiểm thử (test).

Nếu có tập validation, có thể chia theo tỷ lệ 70% (train) - 15% (validation) - 15% (test).

Cách chia:

Ngẫu nhiên (Random Split): Phân phối đều các lớp đối tượng giữa các tập.

Stratified Split: Đảm bảo tỷ lệ giữa các lớp trong tập train/test tương ứng với tỷ lệ của chúng trong toàn bộ dữ liệu.

Kiểm tra dữ liệu:

Đảm bảo không có ảnh nào trong tập test bị trùng với tập train.

Xem xét cân bằng giữa các lớp để tránh mô hình bị bias về một lớp cụ thể.

3.2 Độ đo đánh giá

3.2.1 Cách tính và đo lường trong đoạn mã

1. Phát hiện đối tượng bằng mô hình Deep Learning

Quy trình chính:

Ảnh đầu vào được tiền xử lý thông qua hàm cv2.dnn.blobFromImage để chuẩn hóa.

Mạng nơ-ron (ví dụ YOLO) thực hiện forward pass qua net.forward(output_layers_names).

Kết quả đầu ra chứa các hộp giới hạn (bounding boxes), class scores và độ tin cậy (confidence) cho mỗi đối tượng phát hiện.

2. Lọc kết quả phát hiện

Điều kiện lọc:

confidence > 0.2: Chỉ giữ lại các dự đoán có độ tin cậy cao hơn 0.2.

(center_x, center_y): tọa độ trung tâm của đối tượng.

(w, h): chiều rộng và chiều cao của bounding box.

(x, y): góc trên trái của bounding box, được tính bằng:

$$x = center_x - \frac{w}{2}, \quad y = center_y - \frac{h}{2}$$

Độ đo và ngưỡng (Metrics):

Ngưỡng độ tin cậy (Confidence Threshold):

Chỉ các dự đoán có confidence > 0.2 mới được xem xét.

Ngưỡng này giúp loại bỏ các dự đoán không chắc chắn.

Ngưỡng NMS (IoU Threshold):

ỡng trùng lặp giữa các hộp (IoU - Intersection over Union).

IoU là tỷ lệ giữa diện tích giao nhau và diện tích hợp nhất của hai bounding boxes:

Ý nghĩa: Hộp nào có IoU > 0.4 với một hộp có độ tin cậy cao hơn sẽ bị loại bỏ.

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

Tổng kết quy trình

Ảnh đầu vào được tiền xử lý.

Dự đoán đối tượng qua mô hình và thu được bounding boxes.

Lọc kết quả qua confidence threshold và NMS.

Đếm số lượng đối tượng còn lại sau khi lọc.

Hiển thị bounding boxes và thông tin đối tượng lên ảnh.

Độ chính xác của kết quả phụ thuộc vào:

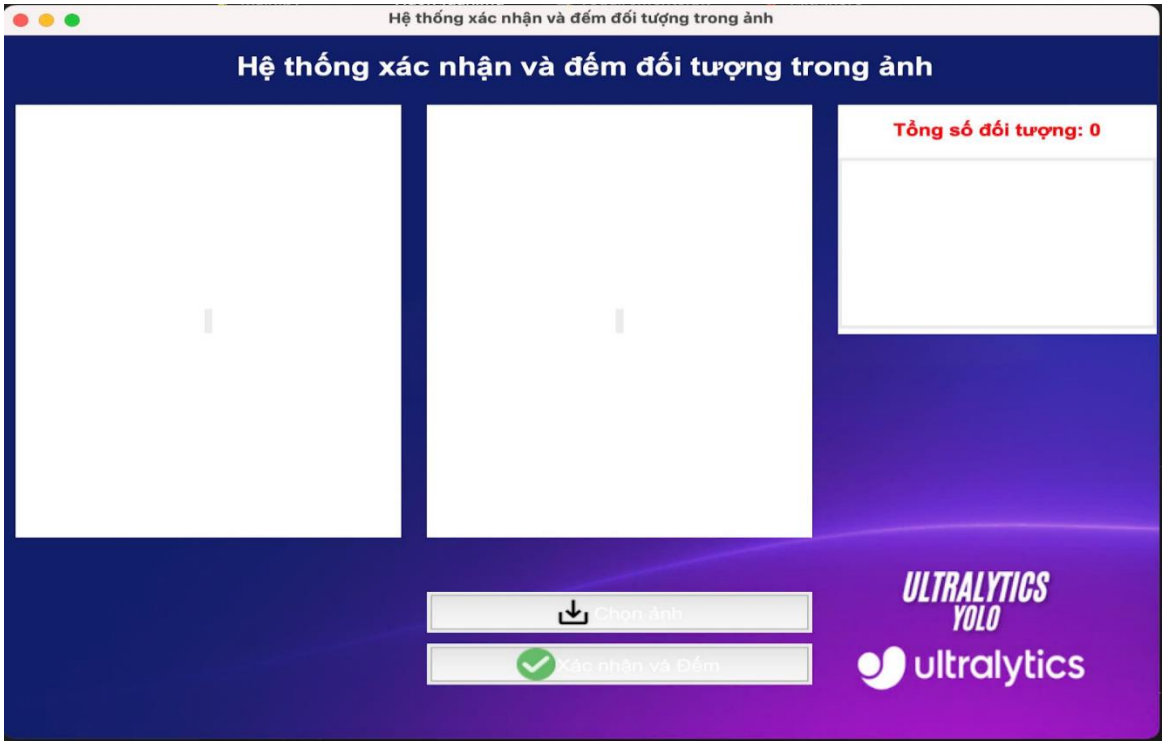
Ngưỡng confidence và IoU.

Hiệu suất của mô hình (YOLO, SSD, Faster R-CNN).

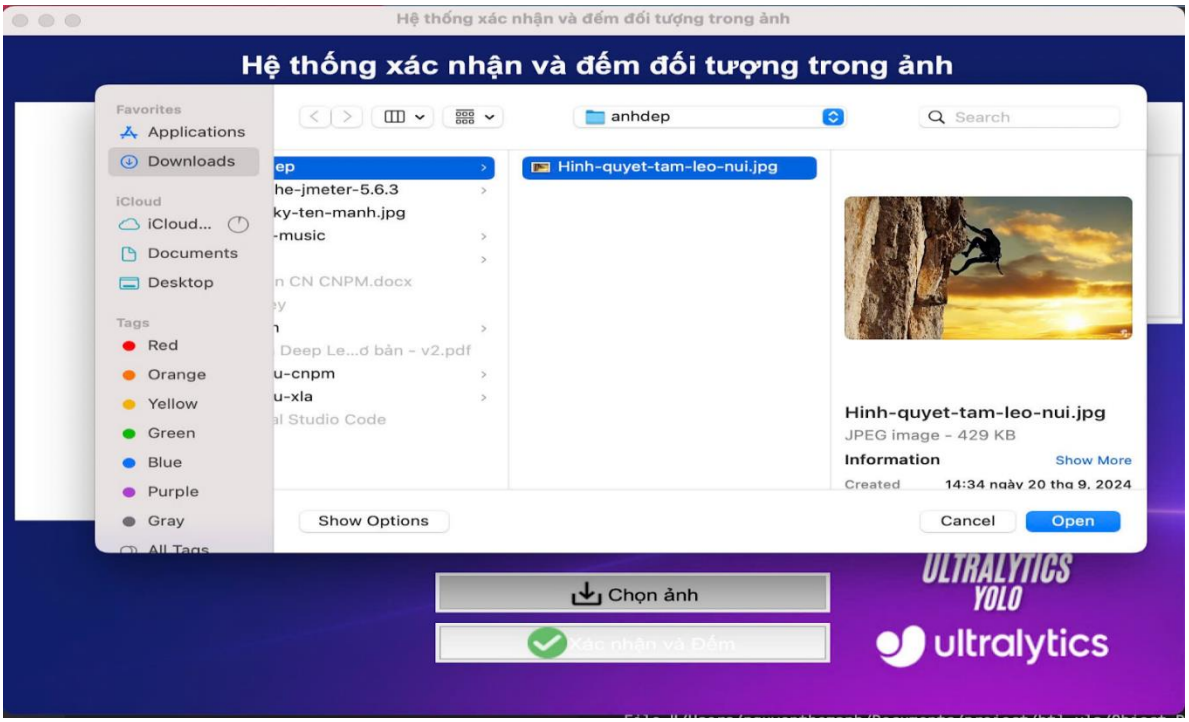
Kích thước ảnh đầu vào (ví dụ: 320x320 cho YOLO).

3.2.2 Cách thực hiện lấy ảnh để đo

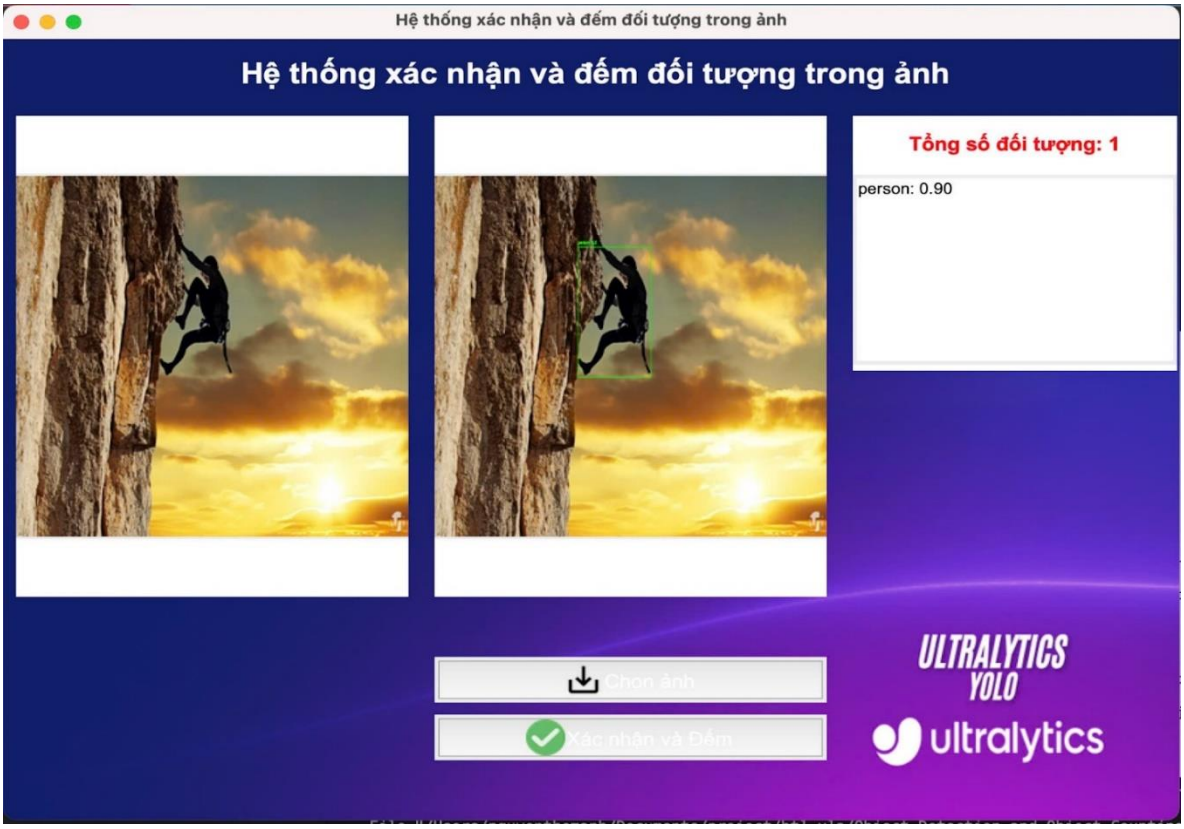
Nhấn vào button tải ảnh để tải ảnh cần xử lý nhận diện và đếm đối tượng lên.



Ấn vào xác nhận và đếm để xử lý và đưa ra kết quả đo được.



Kết quả nhận diện được hình ảnh 1 người với độ đo chính xác:



3.3 Thực nghiệm sản phẩm

3.3.1 Môi trường thực nghiệm

Với bài toán nhận dạng đối tượng trong ảnh bằng mô hình YOLO, môi trường thực nghiệm là không gian và điều kiện được thiết lập để phát triển, huấn luyện, kiểm thử, và đánh giá hiệu quả của mô hình. Cụ thể:

1. Môi trường thực nghiệm phần mềm

Hệ thống máy tính: Máy tính hoặc máy chủ có đủ tài nguyên (GPU mạnh, RAM cao) để huấn luyện và chạy mô hình YOLO.

Framework: Sử dụng các thư viện và công cụ:

YOLOv3 (Darknet), Pillow (PIL), Tkinter, NumPy (np), OpenCV (cv2)

Bộ dữ liệu:

Dataset chứa các hình ảnh gắn nhãn (bộ dữ liệu COCO) để huấn luyện và kiểm tra.

Các tệp nhãn phải đúng định dạng mà YOLO yêu cầu

Công cụ gắn nhãn ảnh: Các phần mềm như LabelImg, Roboflow dùng để chuẩn bị dữ liệu.

2. Môi trường thực nghiệm phần cứng

Thiết bị: Máy tính có GPU mạnh

Cấu hình tối thiểu: Bộ nhớ: Đủ để lưu trữ dữ liệu và mô hình.

3. Các yếu tố khác trong môi trường thực nghiệm

Dữ liệu đầu vào:

Hình ảnh được sử dụng để kiểm thử (độ phân giải, định dạng ảnh phù hợp).

Mô phỏng các điều kiện thực tế: ánh sáng kém, góc chụp nghiêng, hoặc đối tượng bị che khuất.

Mô hình YOLO cụ thể:

Lựa chọn phiên bản YOLO (YOLOv3) phù hợp với bài toán.

Tùy chỉnh tham số mô hình (kích thước ảnh đầu vào, batch size, learning rate).

3.3.2 Mẫu dữ liệu đầu vào

Cho các đối tượng trong bộ dữ liệu COCO được huấn luyện, hệ thống nhận diện được các đối tượng sau:

Personbicycle	Catdog	Skissnowboard	Bananaapple
Carmotorbike	Horsesheep	Sports ballkite	Sandwichorange
Aeroplanebus	Cow	Spoonbowl	Broccolicarrot
Traintruck	Elephantbear	Skateboardsurfboard	Hot dogpizza
Boat	Zebragiraffe	Tennis racketbottle	Donutcake
Benchbird	Backpackumbrella	Wine glasscup	Chairsofa

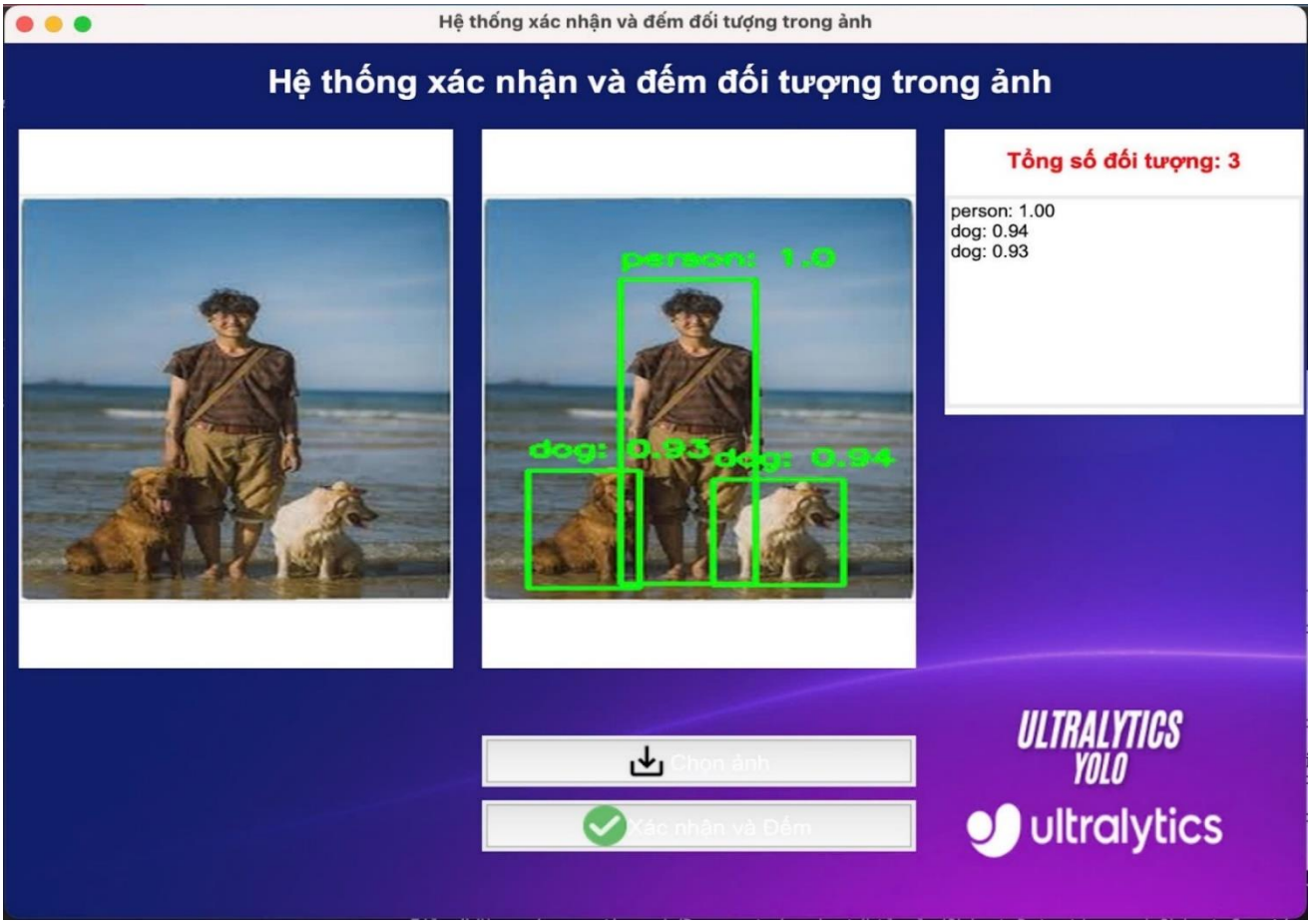
Traffic lightfire hydrant	Handbagtie	Forkknife	Pottedplantbed
Stop signparking meter	Suitcasefrisbee	Baseball batbaseball glove	Diningtabletoilet
Tvmonitorlaptop	Keyboardcell phone	Toastersink	Clockvase
Mouseremote	Microwaveoven	Refrigeratorbook	Scissorsteddy bear

3.3.3 Kết quả thực nghiệm

Hình 3.3.3 1 Kết quả thực nghiệm với dữ liệu đầu vào 1

Nhận diện được 23 đối tượng gồm với độ đo sau:

Person: 0.98	Person: 0.80	Person: 0.38
Person: 0.96	Traffic light: 0.80	Traffic light: 0.36
Truck: 0.96	Motorbike: 0.79	Motorbike: 0.30
Person: 0.93	Backpack: 0.46	Traffic light: 0.29
Motorbike: 0.87	Truck: 0.46	Bicycle: 0.26
Person: 0.86	Car: 0.45	Person: 0.24
Motorbike: 0.84	Motorbike: 0.45	Car: 0.24
Person: 0.82	Car: 0.43	



Hình 3.3.3 2 Kết quả thực nghiệm với dữ liệu đầu vào 2

Kết quả nhận diện được hình ảnh với 1 người và 2 chó với độ chính xác sau:

Person: 1.00

Dog: 0.94

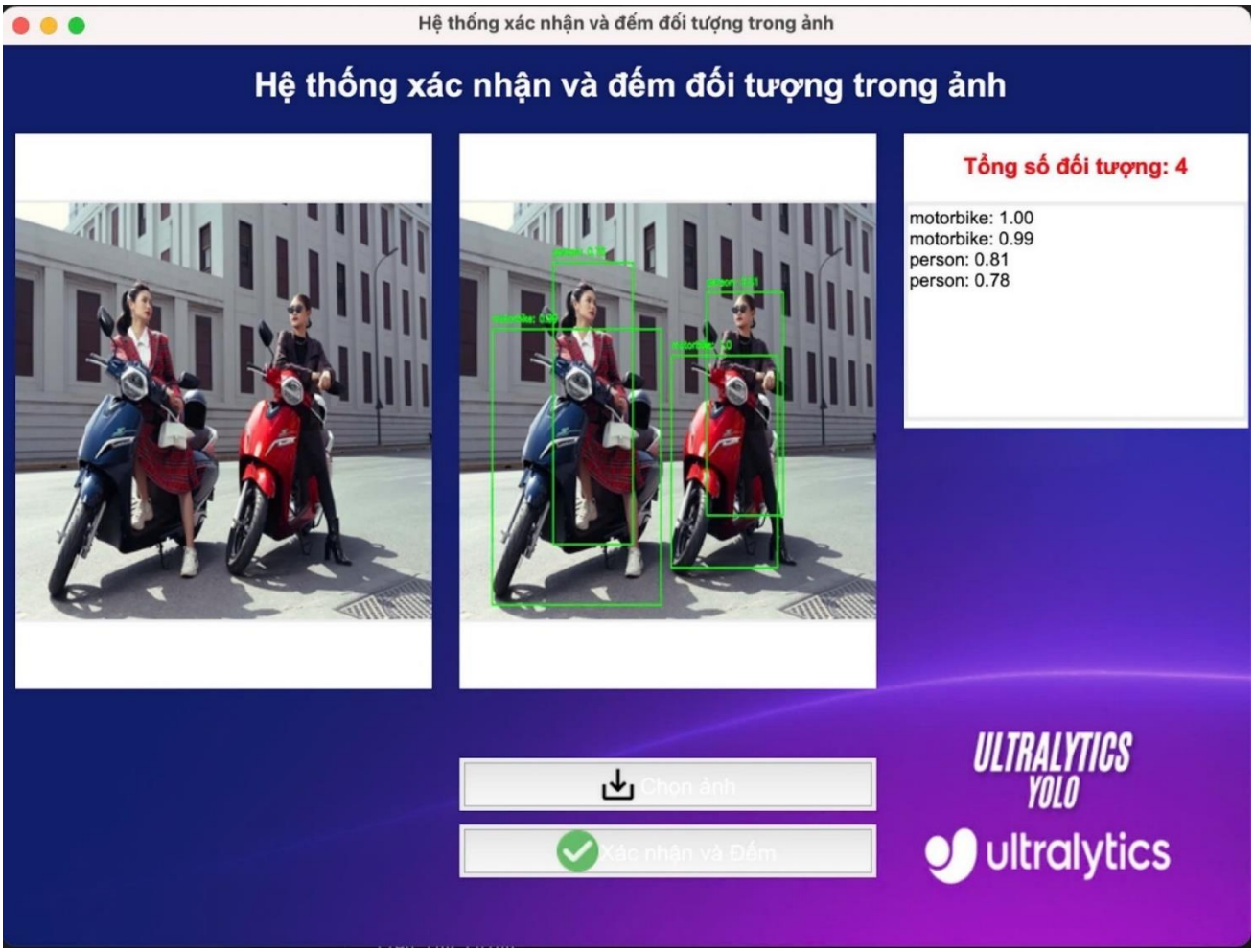
Dog: 0.93

Hình 3.3.3 3 Kết quả thực nghiệm với dữ liệu đầu vào 3

Phát hiện 2 đối tượng là 2 con mèo với độ chính xác sau:

Cat: 1.00

Cat: 1.00



Hình 3.3.3 4 Kết quả thực nghiệm với dữ liệu đầu vào 4

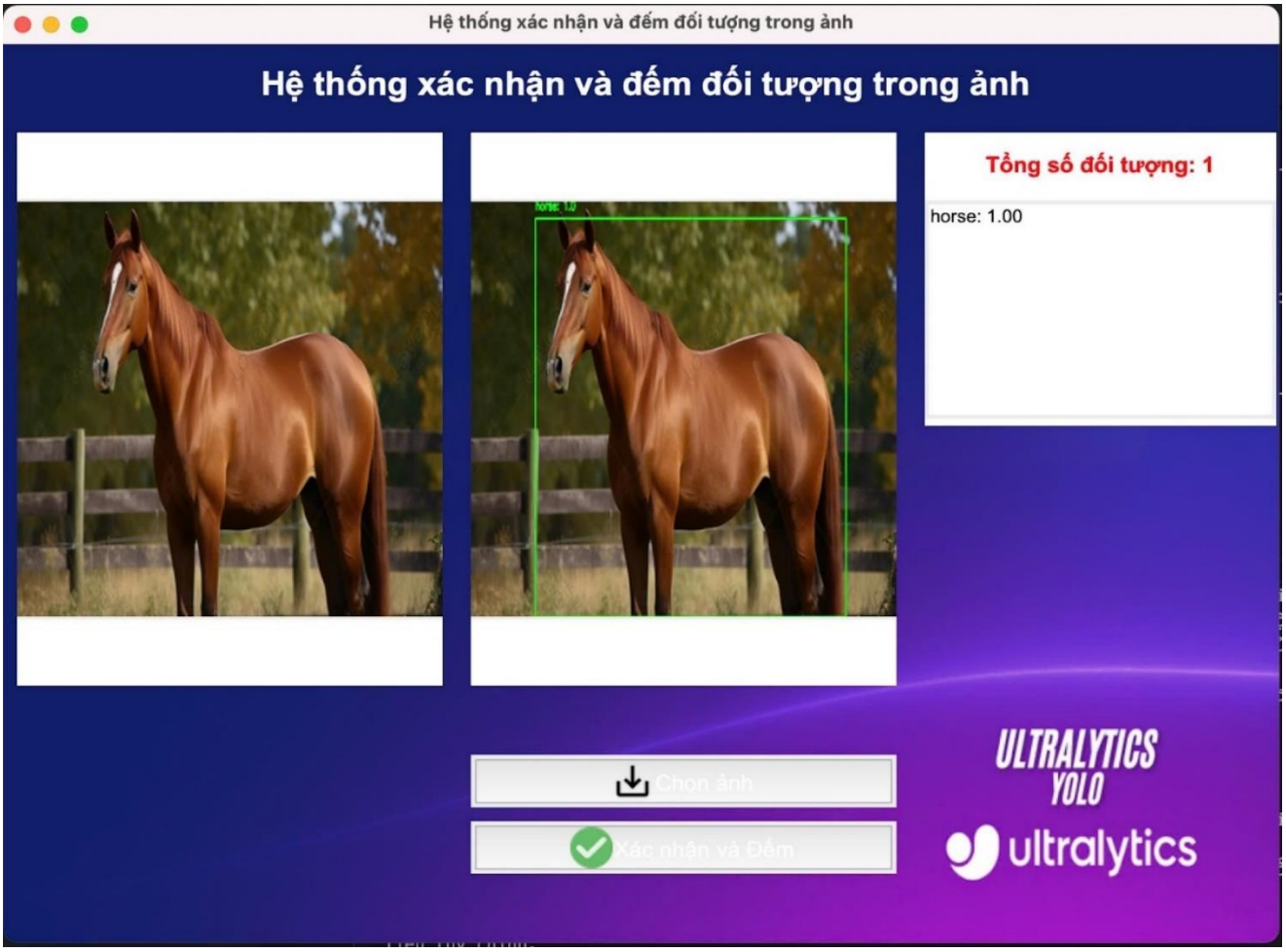
Nhận diện 4 đối tượng, phát hiện 2 người và 2 xe máy với độ chính xác sau:

Motobike: 1.00

Motobike: 0.99

Person: 0.81

Person: 0.78



Hình 3.3.3 5 Kết quả thực nghiệm với dữ liệu đầu vào 5

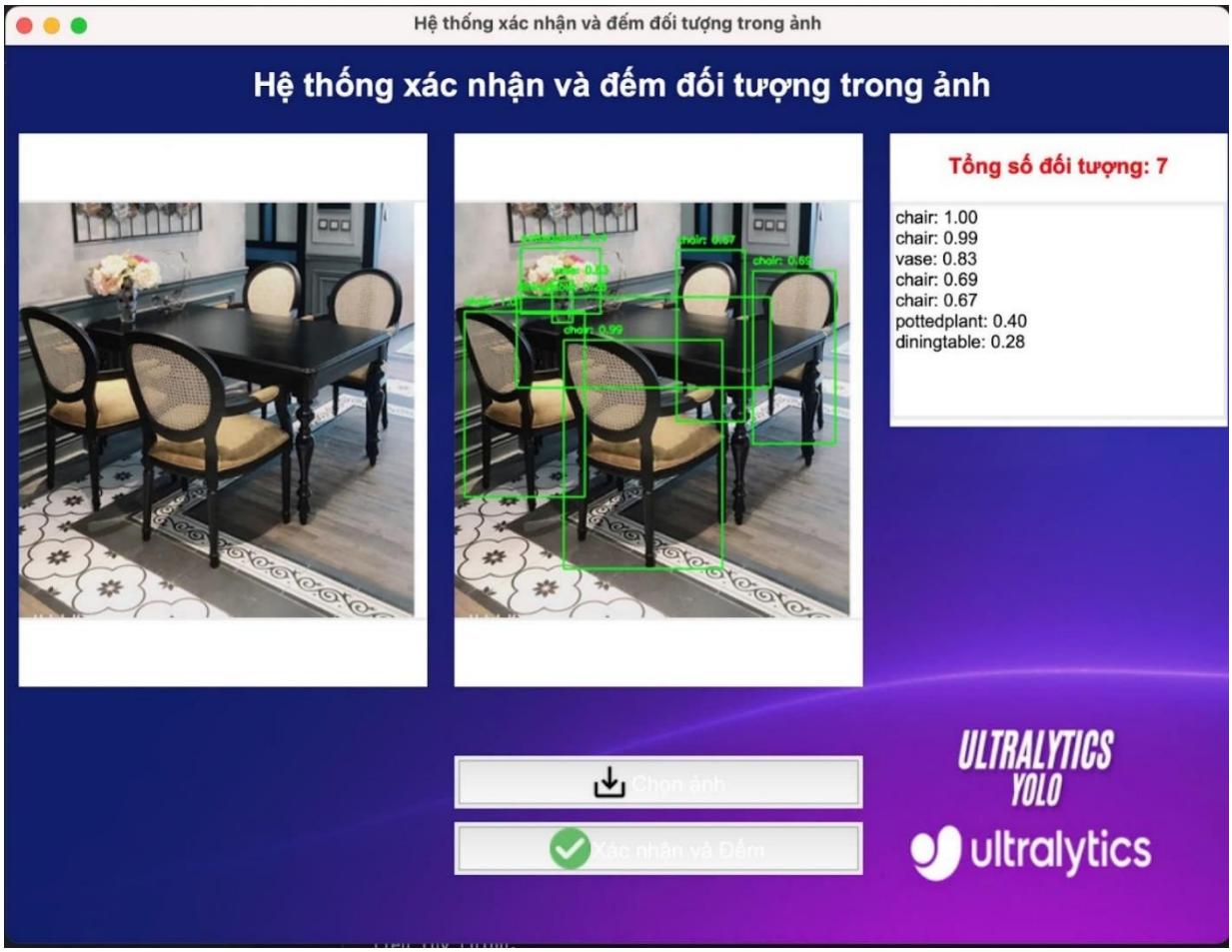
Nhận diện được 1 con ngựa với độ chính xác sau: Horse: 1.00

Hình 3.3.3 6 Kết quả thực nghiệm với dữ liệu đầu vào 6

Nhận diện được 2 con chim với độ chính xác sau:

Bird: 0.78

Bird: 0.68



Hình 3.3.3 7 Kết quả thực nghiệm với dữ liệu đầu vào 7

7

Nhận diện được 7 đối tượng gồm 4 ghế, 1 hoa, 1 cắm lọ, 1 bàn ăn với độ chính xác sau:

Chair: 1.00

Chair: 0.99

Chair: 0.69

Chair: 0.67

Vase: 0.83

Pottedplant: 0.40

Diningtable: 0.28

Hình 3.3.3 8 Kết quả thực nghiệm với dữ liệu đầu vào 8

Nhận diện được đối tượng con voi, gồm 2 đối tượng với độ chính xác sau:

Elephant: 0.93

Elephant: 0.84

Hình 3.3.3 9 Kết quả thực nghiệm với dữ liệu đầu vào 9

Kết quả nhận diện động vật chó có số lượng 5 con với độ chính xác sau:

Dog: 1.00

Dog: 0.98

Dog: 0.95

Dog: 0.93

Dog: 0.71

Hình 3.3.3 10 Kết quả thực nghiệm với dữ liệu đầu vào 10

Nhận diện được 14 người với độ chính xác sau:

Person: 1.00

Person: 1.00

Person: 0.99

Person: 0.99

Person: 0.99

Person: 0.99

Person: 0.99

Person: 0.98

Person: 0.98

Person: 0.97

KẾT LUẬN

Kết quả đạt được

Sản phẩm đã được phát triển và hoàn thiện theo đúng yêu cầu của bài toán đặt ra, với khả năng nhận dạng và đếm đối tượng trong ảnh một cách hiệu quả. Bằng cách áp dụng mô hình học máy YOLO (You Only Look Once) – một trong những kiến trúc mạng tiên tiến và tối ưu trong việc phát hiện đối tượng, phần mềm có thể nhận diện được khoảng 40 loại đối tượng khác nhau. Kết quả đếm số lượng các vật thể trong ảnh đầu vào được thực hiện với độ chính xác cao, đáp ứng tốt các tiêu chí của bài toán.

Phần mềm không chỉ dừng lại ở khả năng xử lý ảnh mà còn được thiết kế dưới dạng desktop application với giao diện thân thiện và trực quan, nhằm hỗ trợ người dùng dễ dàng thao tác và sử dụng các chức năng. Giao diện của ứng dụng bao gồm các nút chức năng như:

- Button "Tải ảnh": cho phép người dùng nhập dữ liệu đầu vào là ảnh từ máy tính.
- Button "Kiểm tra kết quả": thực thi quá trình xử lý và hiển thị kết quả trực tiếp.

Màn hình kết quả được thiết kế để hiển thị song song giữa ảnh gốc và ảnh đã qua xử lý. Trong ảnh đã xử lý, các đối tượng được khoanh vùng bằng khung giới hạn (bounding box), kèm theo nhãn nhận diện (label) chỉ rõ tên của từng đối tượng. Đồng thời, phần mềm còn liệt kê số lượng chính xác của từng loại đối tượng ngay trên giao diện, giúp người dùng dễ dàng quan sát và kiểm tra.

Phần mềm được xây dựng với mục tiêu hỗ trợ người dùng nhanh chóng thực hiện các thao tác nhận diện mà không đòi hỏi kiến thức chuyên sâu về công nghệ. Từ quá trình huấn luyện mô hình YOLO, tích hợp vào ứng dụng, đến việc thiết kế giao diện và tối ưu hóa quy trình xử lý, sản phẩm đã hoàn thiện đầy đủ các chức năng từ nhập dữ liệu, xử lý, và hiển thị kết quả một cách mượt mà. Nhờ đó, sản phẩm không chỉ đáp ứng được yêu cầu của bài toán mà còn có tiềm năng ứng dụng thực tế cao trong các lĩnh vực như kiểm kê hàng hóa, quản lý tài sản, hoặc hỗ trợ trong lĩnh vực an ninh giám sát.

Hướng phát triển

Trong tương lai, sản phẩm sẽ được phát triển theo nhiều hướng nhằm nâng cao hiệu quả và tính ứng dụng. Trước tiên, sẽ mở rộng khả năng nhận diện bằng cách huấn luyện mô hình với tập dữ liệu lớn và đa dạng hơn, cho phép nhận diện nhiều loại đối tượng, bao gồm cả các vật thể chuyên biệt như công cụ sản xuất, thiết bị y tế hoặc sản phẩm tiêu dùng. Bên cạnh đó, sản phẩm sẽ được tích hợp tính năng xử lý video theo thời gian thực, giúp nhận diện và đếm đối tượng trong các cảnh quay động, phù hợp cho các ứng dụng giám sát hoặc phân tích hành vi. Ngoài ra, việc cải thiện giao diện người dùng cũng được chú trọng, nhằm mang lại trải nghiệm trực quan và thân thiện hơn, chẳng hạn như bổ sung báo cáo chi tiết kết quả hoặc hỗ trợ ngôn ngữ đa dạng. Sản phẩm cũng có thể được triển khai trên nền tảng web hoặc thiết bị di động để tăng khả năng tiếp cận và ứng dụng thực tiễn.

TÀI LIỆU THAM KHẢO

[1] “Bài toán nhận dạng”

https://thigiacmaytinh.com/deep-learning/, ngày 29/11/2024

[2] “Các bước thực hiện bài toán nhận dạng”

Tham khảo: https://chatgpt.com/, ngày 29/11/2024

[3] “Học Có Giám sát (Supervised Learning)”

https://blog.vinbigdata.org/supervised-learning-va-unsupervised-learning-khac-biet-la-gi/, ngày 30/11/2024

[4] “Học Không Giám sát (Unsupervised Learning)”

https://blog.vinbigdata.org/supervised-learning-va-unsupervised-learning-khac-biet-la-gi/, ngày 30/11/2024

[5] “Học Tăng Cường (Reinforcement Learning)”

https://aws.amazon.com/vi/what-is/reinforcement-learning/, ngày 30/11/2024

[6] “Học sâu (Deep Learning)”

https://aws.amazon.com/vi/what-is/deep-learning/, ngày 30/11/2024

[7] “Các kỹ thuật khác”

https://aws.amazon.com/vi/what-is/deep-learning/, ngày 30/11/2024

[8] “Python là gì”

https://aws.amazon.com/vi/what-is/python/, ngày 30/11/2024

[9] “Lợi ích của python”

https://aws.amazon.com/vi/what-is/python/, ngày 30/11/2024

[10] “OpenCV (cv2)”

https://teky.edu.vn/blog/opencv-la-gi/, ngày 01/12/2024

[11] “NumPy (np)”

https://codelearn.io/sharing/tim-hieu-thu-vien-numpy-trong-python, ngày 01/12/2024

[12] “Tkinter”

https://www.icantech.vn/kham-pha/tkinter, ngày 01/12/2024

[13] “Pillow (PIL)”

https://viblo.asia/p/huong-dan-su-dung-thu-vien-pillow-de-xu-ly-hinh-anh-trong-python-cho-nguoi-moi-bat-dau-3Q75wm4MZWb, ngày 01/12/2024

[14] “YOLOv3 (Darknet)”

https://viblo.asia/p/yolov3-tinh-chinh-de-tro-nen-tot-hon-E1XVO3kPVMz, ngày 01/12/2024

[15] “Mô hình YOLO và các kỹ thuật chính thường sử dụng trong mô hình YOLO”

https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P, ngày 30/1112024