ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

Môn học: Phân Tích Dữ Liệu

MÔ HÌNH YOLOV5
Ứng dụng trong nhận diện đối tượng

Giảng viên hướng dẫn: Ts. Đỗ Trọng Hợp

Sinh viên thực hiện:		
STT	**Họ tên**	**MSSV**
1	Nguyễn Trung Hiếu	22550004
2	Ngô Trần Tuấn Phong	22550016
3	Đào Nhâm Phúc	22550017
4	Phạm Hoàng Sang	22550019

Thành phố Hồ Chí Minh – 02/2024

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày 21 tháng 02 năm 2024

NHẬN XÉT ĐỒ ÁN MÔN HỌC

Tên đồ án:

MÔ HÌNH YOLOV5

Ứng dụng trong nhận diện đối tượng

Nhóm SV thực hiện:

Họ và tên: Nguyễn Trung Hiếu

MSSV: 22550004

Họ và tên: Ngô Trần Tuấn Phong

MSSV: 22550016

Họ và tên: Đào Nhâm Phúc

MSSV: 22550017

Họ và tên: Phạm Hoàng Sang

MSSV: 22550019

Giảng viên phụ trách:

Ts. Đỗ Trọng Hợp

Đánh giá Đồ án:

1. Về cuốn báo cáo:

Số trang :	Số chương :
Số bảng số liệu :	Số hình vẽ :
Số tài liệu tham khảo :	Sản phẩm :

Một số nhận xét về hình thức cuốn báo cáo:

---

2. Về nội dung nghiên cứu:

---

3. Về thái độ làm việc của sinh viên:

Đánh giá chung:

Điểm sinh viên:

Nguyễn Trung Hiếu:……../10

Ngô Trần Tuấn Phong:……../10

Đào Nhâm Phúc:……../10

Phạm Hoàng Sang:……../10

Người nhận xét
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trải qua thời gian học tập và rèn luyện cùng Thầy và các bạn, bản thân chúng em đã tiếp thu được rất nhiều những kiến thức.

Chúng em xin chân thành cảm ơn Thầy Đỗ Trọng Hợp đã tận tình giảng dạy, hướng dẫn và giúp đỡ chúng em trong suốt quá trình học tập.

Cuối cùng, chúng em chúc Thầy công tác tốt và luôn dồi dào sức khỏe để có thể cống hiến hết mình cho những thế hệ sinh viên tiếp theo.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Trung Hiếu

Ngô Trần Tuấn Phong

Đào Nhâm Phúc

Phạm Hoàng Sang

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự Do - Hạnh Phúc

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỒ ÁN:

MÔ HÌNH YOLOV5 - Ứng dụng trong nhận diện đối tượng

Cán bộ hướng dẫn:
Đỗ Trọng Hợp

Thời gian thực hiện:

Từ ngày 01-01-2024 đến ngày 21-02-2024

Sinh viên thực hiện:

Nguyễn Trung Hiếu – 22550004

Ngô Trần Tuấn Phong – 22550016

Đào Nhâm Phúc – 22550017

Phạm Hoàng Sang – 22550019

Nội dung đồ án:

- Giới thiệu về Deep Learning
- Giới thiệu về thuật toán Object Detection
- Hướng dẫn sử dụng mô hình YOLO để phát hiện và nhận diện vật thể
- Demo
- Kết luận

Kế hoạch thực hiện:

Tìm hiểu và demo về nhận diện vật thể trong hình ảnh và video sử dụng mô hình YOLO

Xác nhận của Giảng viên (Ký tên và ghi rõ họ tên)	**TP. HCM, ngày 21 tháng 02 năm 2024** **Sinh viên** (Ký tên và ghi rõ họ tên) **Nguyễn Trung Hiếu, Ngô Trần Tuấn Phong, Đào Nhâm Phúc, Phạm Hoàng Sang**

MỤC LỤC

DANH MỤC HÌNH ẢNH

I. Giới thiệu về Deep Learning

1. Giới thiệu

Deep Learning (học sâu) là một chức năng của trí tuệ nhân tạo (AI), có thể được xem là một lĩnh vực con của Machine Learning (học máy) – ở đó các máy tính sẽ học và cải thiện chính nó thông qua các thuật toán. Deep Learning được xây dựng dựa trên các khái niệm phức tạp hơn rất nhiều, chủ yếu hoạt động với các mạng nơ-ron nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người để xử lí dữ liệu, tạo ra các mẫu để sử dụng cho việc đưa ra quyết định.

Thật ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã xuất hiện từ khoảng những năm 1960, tuy nhiên nó lại bị giới hạn bởi khả năng tính toán và số lượng dữ liệu lúc bấy giờ. Trong những năm gần đây, những tiến bộ trong phân tích dữ liệu lớn (Big Data) đã cho phép ta tận dụng được tối đa khả năng của mạng nơ-ron nhân tạo.

Mạng nơ-ron nhân tạo chính là động lực chính để phát triển Deep Learning. Các mạng nơ-ron sâu (DNN) bao gồm nhiều lớp nơ-ron khác nhau, có khả năng thực hiện các tính toán có độ phức tạp rất cao. Deep Learning hiện đang phát triển rất nhanh và được xem là một trong những bước đột phá lớn nhất trong Machine Learning.

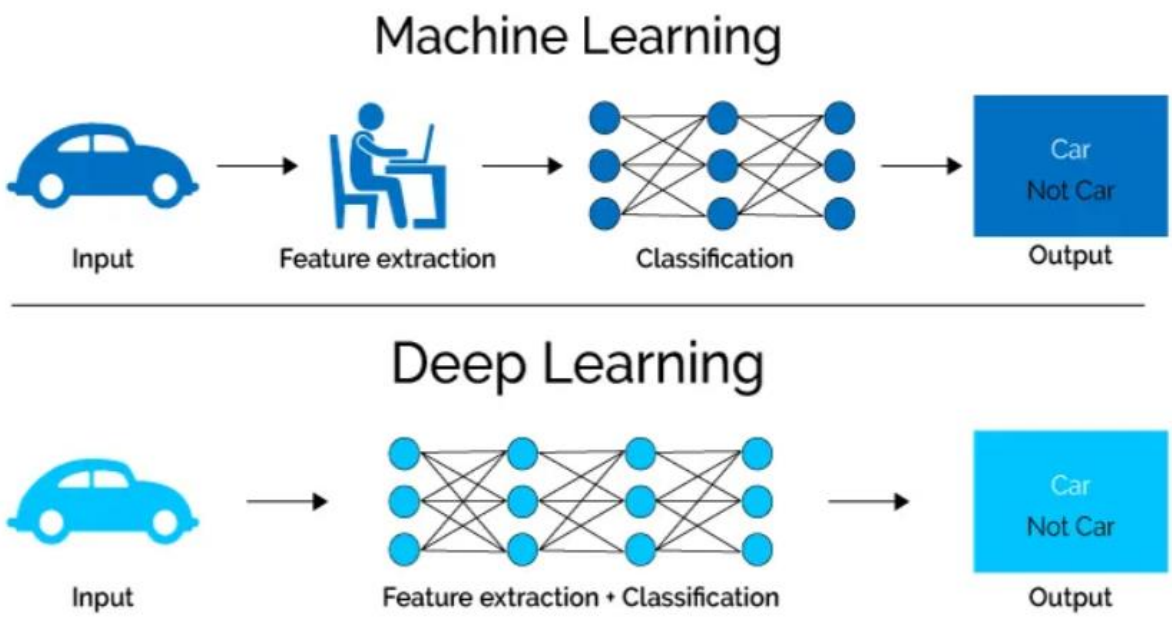
2. Cách hoạt động Deep Learning

Deep Learning là một phương pháp của Machine Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người.

Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, số lượng layer càng nhiều thì mạng sẽ càng "sâu". Có 3 loại layer chính của các nơ ron trong mạng nơ ron là: Input layer, Các hidden layer, Output layer.

Trong mỗi layer là các nút mạng (node) và được liên kết với những lớp liền kề khác. Mỗi kết nối giữa các node sẽ có một trọng số tương ứng, trọng số càng cao thì ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.

Mỗi nơ-ron sẽ có một hàm kích hoạt, về cơ bản thì có nhiệm vụ "chuẩn hoá" đầu ra từ nơ-ron này. Dữ liệu được người dùng đưa vào mạng nơ-ron sẽ đi qua tất cả layer và trả về kết quả ở layer cuối cùng, gọi là output layer.



Hình 1: Hoạt động của Deep Learning

Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm ra bộ giá trị của trọng số sao cho phán đoán là tốt nhất.

Các hệ thống Deep Learning yêu cầu phần cứng phải rất mạnh để có thể xử lý được lượng dữ liệu lớn và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất nhiều tuần, thậm chí nhiều tháng để triển khai trên những phần cứng tiên tiến nhất hiện nay.

3. Yêu cầu khi sử dụng Deep Learning

Mặc dù có hiệu năng và mức độ chính xác vượt trội nhờ nhờ nguồn dữ liệu lớn, mô hình phức tạp. Tuy nhiên, deep learning không phải lúc nào cũng là sự lựa chọn duy nhất cho các bài toán trong lĩnh vực trí tuệ nhân tạo và học máy.

Vậy nên, việc nên sử dụng deep learning lúc nào phụ thuộc vào các yếu tố sau đây:

➢ Mục tiêu và độ phức tạp của dự án: Lợi thế của deep learning là giải quyết các vấn đề phức tạp bằng cách đưa ra các phân tích trong mối quan hệ ẩn trong dữ liệu. Đặc biệt mô hình này phù hợp khi áp dụng vào việc xử lý dữ liệu ở nhiều dạng khác như ngôn ngữ, hình ảnh, nhận diện giọng nói, v.v.

➢ Tài nguyên: Một khối lượng lớn dữ liệu của doanh nghiệp cần được xử lý thông qua mô hình deep learning sẽ dễ dàng hơn. Tuy nhiên, quá trình xử lý vô cùng phức tạp và tốn kém do đó tùy vào khối lượng dữ liệu mà doanh nghiệp sẽ đưa ra quyết định nên lựa chọn mô hình deep learning hay machine learning.

➢ Số lượng lớn dữ liệu: Mô hình deep learning chỉ ra các mối quan hệ ẩn sâu bên trong bộ dữ liệu. Tuy nhiên, điều này cũng đồng nghĩa với việc dữ liệu đầu vào phải lớn hơn nhiều so với thuật toán của machine learning. Do đó, đối với lượng dữ liệu lớn việc sử dụng deep learning rất phù hợp.

4. Các kỹ thuật của Deep Learning

Có đa dạng kỹ thuật và thuật toán Deep Learning, từ những phương pháp đơn giản đến những hệ thống phức tạp, có thể áp dụng cho hầu hết các bài toán trong lĩnh vực trí tuệ nhân tạo hiện nay.

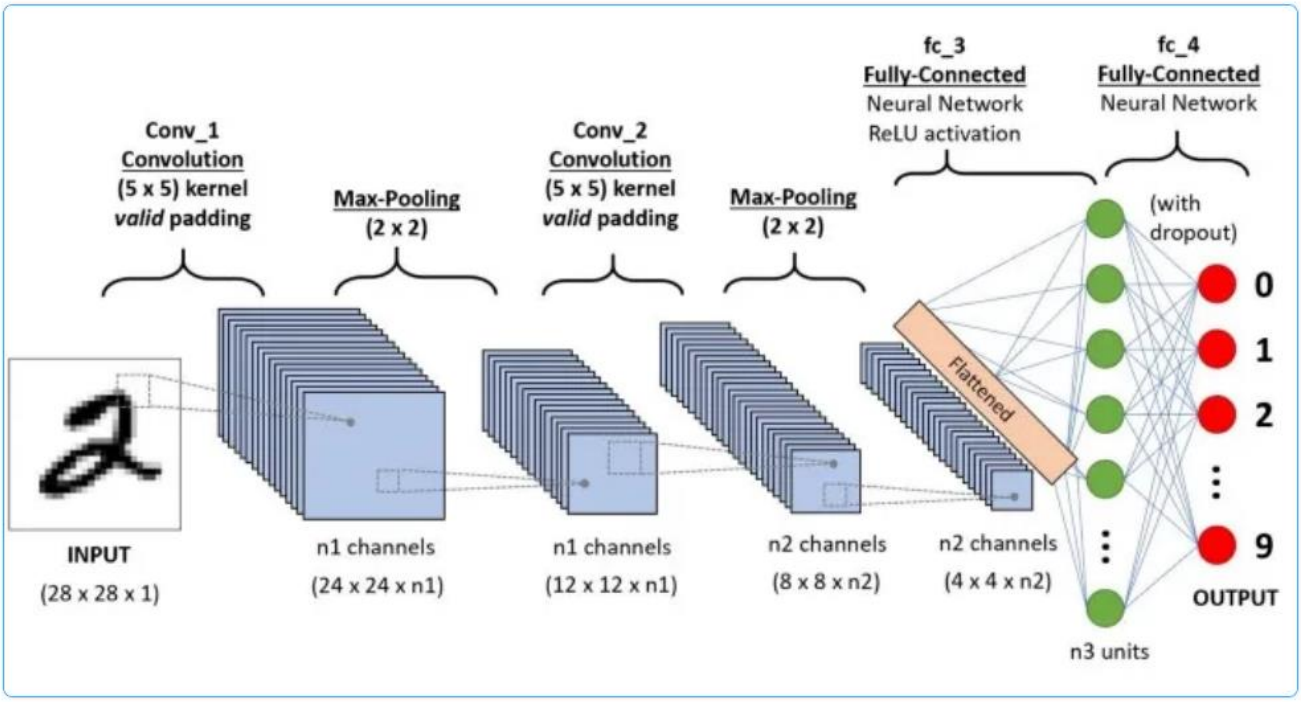
4.1 Mạng nơ-ron cổ điển

Kiến trúc cổ điển của mạng nơ-ron là mạng kết nối đầy đủ, thường được xác định bằng các perceptron đa lớp. (Perceptron là một thuật toán đơn giản, cho phép tìm một ranh giới siêu phẳng cho các bài toán phân lớp nhị phân). Có ba loại hàm thường được sử dụng trong mô hình này là:

➢ Hàm tuyến tính.

➢ Hàm phi tuyến: gồm có hàm sigmoid, hàm tanh và hàm ReLU (Rectified Linear Unit).

4.2 Mạng nơ-ron tích chập (CNN)



Hình 2: Mạng nơ-ron tích chập (CNN)

Mạng nơ-ron tích chập (Convolutional Neural Network – CNN) là một kiến trúc Neural Network nhân tạo nâng cao, được xây dựng để giải quyết các bài toán phức tạp, đặc biệt là liên quan đến xử lý hình ảnh.

Tích chập là một khái niệm trong xử lý tín hiệu số nhằm biến đổi thông tin đầu vào qua một phép tích chập với bộ lọc, nhằm trả về đầu ra là một tín hiệu mới. Tín hiệu này sẽ giảm bớt những đặc trưng mà bộ lọc không quan tâm, giữ lại những đặc trưng chính và quan trọng nhất.

Bên cạnh input layer và output layer, mô hình CNN còn có thêm một sampling layer để giới hạn số lượng nơ-ron tham gia vào các layer tương ứng. Việc xây dựng mô hình trải qua ba giai đoạn chính:

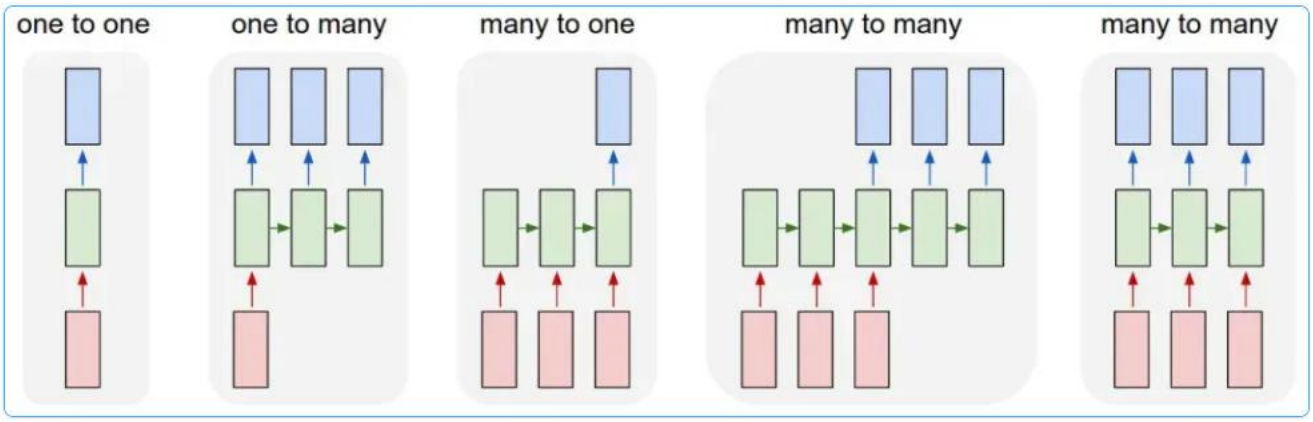
➢ Quá trình tích chập (convolution): Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một tầng mới. Quá trình này có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng kèm với hàm kích hoạt ReLU. Mục tiêu của tầng này là trích suất đặc trưng hai chiều.

➢ Quá trình tổng hợp (max pooling): Giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh.

➢ Quá trình kết nối hoàn toàn (fully connected): Sau khi đã giảm kích thước đến một mức độ hợp lý, ma trận cần được trải phẳng (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các tầng. Tầng kết nối hoàn toàn cuối cùng (fully connected layer) sẽ có số lượng đơn vị bằng với số lớp.

Dựa vào những đặc điểm của mình, các ứng dụng phổ biến nhất của mạng CNN gồm có: Nhận diện, phân tích và phân khúc hình ảnh, phân tích video, xử lý ngôn ngữ tự nhiên,…

4.3 Mạng nơ-ron hồi quy (RNN)



Hình 3: Mạng nơ-ron hồi quy (RNN)

Recurrent Neural Network (RNN) là một thuật toán nổi tiếng trong lĩnh vực xử lý ngôn ngữ tự nhiên. Trong các mô hình mạng nơ-ron truyền thống, đầu vào và đầu ra độc lập với nhau.

Tuy nhiên RNN thực hiện cùng một tác vụ cho tất cả phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Vì vậy mạng RNN có khả năng nhớ các thông tin được tính toán trước đó.

Có hai thiết kế chính của RNN:

➢ *LSTM (Long Short-Term Memory)*: Được dùng để dự đoán dữ liệu dạng chuỗi thời gian, có khả năng bỏ đi hoặc thêm các thông tin cần thiết, được điều chỉnh bởi các nhóm được gọi là cổng (gate): Input, Output và Forget.

➢ *Gated RNN*: Cũng là một thiết kế phổ biến trong lĩnh vực dự đoán dữ liệu của chuỗi thời gian, có hai cổng là Update và Reset.

Các dạng bài toán RNN:

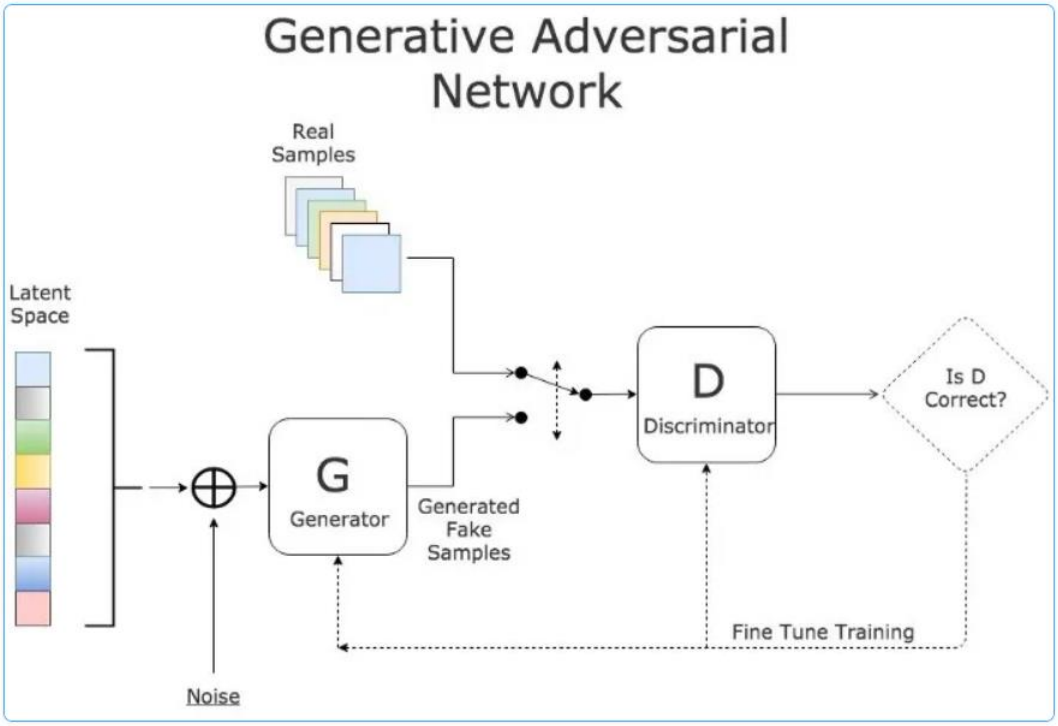
➢ *One to one*: Chỉ có một input kết nối với một output duy nhất, chẳng hạn như các bài toán phân loại hình ảnh.

➢ *One to many*: Một input liên kết với nhiều chuỗi output, phổ biến là các bài toán đặt caption cho ảnh.

➢ *Many to One*: Nhiều input nhưng chỉ có output, phổ biến là bài toán phân loại cảm xúc.

➢ *Many to many*: Nhiều input và nhiều output, chẳng hạn như phân loại video.

4.4 Mạng sinh đối nghịch (GAN)



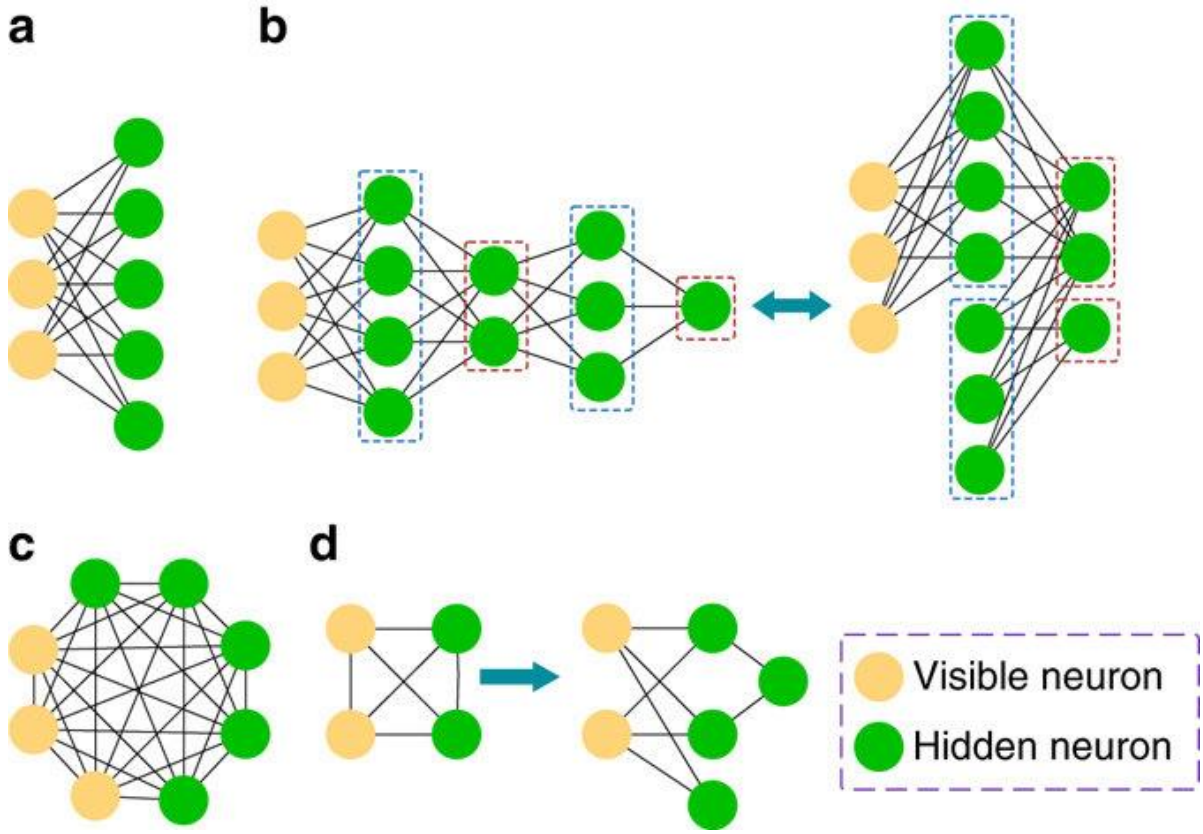
Hình 4: Mạng sinh đối nghịch (GAN)

Generative Adversarial Networks (GAN) là lớp mô hình có mục tiêu tạo ra dữ liệu giả giống với thật, tên của mạng được dựa trên kiến trúc gồm hai mạng có mục tiêu đối nghịch nhau: Generator và Discriminator.

Trong đó Generator học cách sinh dữ liệu giả để lừa mô hình Discriminator, còn Discriminator lại học cách phân biệt giữa dữ liệu giả và dữ liệu thật. Thông qua quá trình huấn luyện thì cả hai mô hình này đều cùng cải thiện được khả năng của mình.

Một số ứng dụng phổ biến của GAN là: Tạo khuôn mặt người, thay đổi độ tuổi khuôn mặt, sinh ảnh vật thể, tạo nhân vật hoạt hình,…

4.5 Boltzmann machine

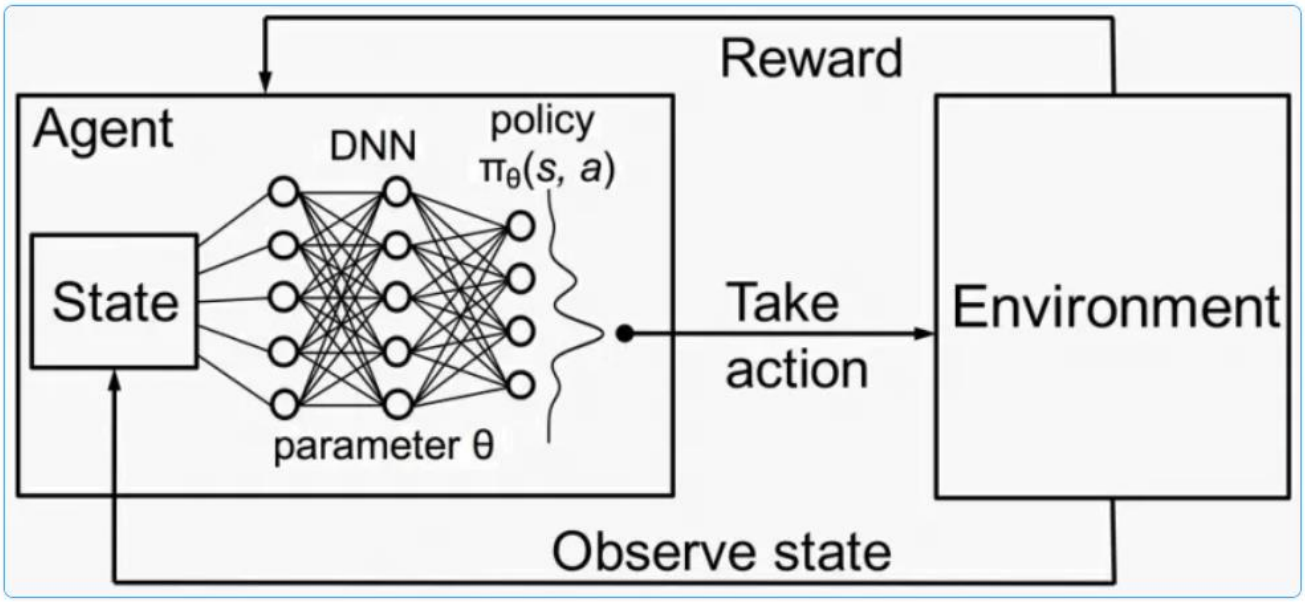


Hình 5: Boltzmann machine

Đây là một mô hình mạng không có hướng xác định, vì vậy các node của mạng này được liên kết với nhau thành một hình tròn. Dựa vào kiến trúc này, máy Boltzmann (Boltzmann machine) thường được sử dụng để tạo ra các tham số cho mô hình.

Các ứng dụng phổ biến nhất của mô hình là: giám sát hệ thống, xây dựng hệ thống khuyến nghị nhị phân,…

4.6 Deep Reinforcement Learning



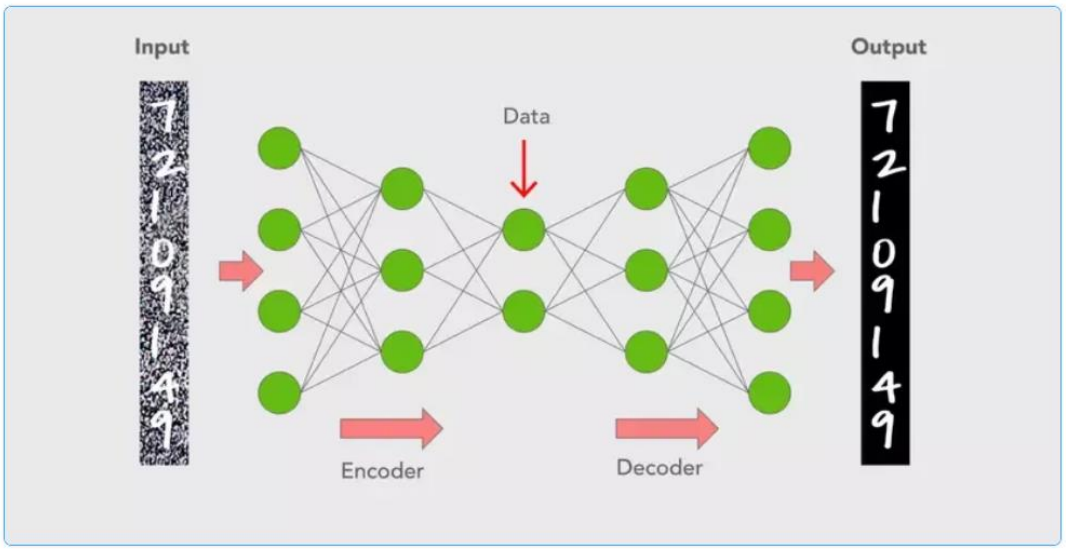
Hình 6: Deep Reinforcement Learning

Deep Reinforcement Learning (Học tăng cường sâu) là quá trình mà các tác tử (agent) tương tác với môi trường để thay đổi trạng thái của chính nó. Các tác tử có thể quan sát và thực hiện những hành động phù hợp, từ đó giúp mạng đạt được mục tiêu.

Mô hình mạng này gồm một input layer, output layer và nhiều hidden layer khác, trong đó trạng thái của môi trường chính là input layer. Mô hình sẽ huấn luyện liên tục để dự đoán điểm đạt được sau mỗi hành động được thực hiện trong từng trạng thái nhất định.

Mô hình học tăng cường sâu được ứng dụng chủ yếu trong các game cờ vua, poker, xe tự lái, robot,…

4.7 Autoencoder



Hình 7: Autoencoder

Autoencoder là một trong những kỹ thuật Deep Learning phổ biến nhất hiện nay, có khả năng học các biểu diễn của dữ liệu đầu vào mà không cần nhãn, hay nói cách khác thì mạng này có khả năng học không giám sát (unsupervised learning).

Một số loại autoencoder là:

➢ *Sparse (thưa):* Số lượng hidden layer lớn hơn số lượng input layer nhằm hạn chế hiện tượng quá khớp (overfitting). Phương pháp này giới hạn hàm mất mát và ngăn không cho autoencoder lạm dụng tất cả các node có trong mạng.

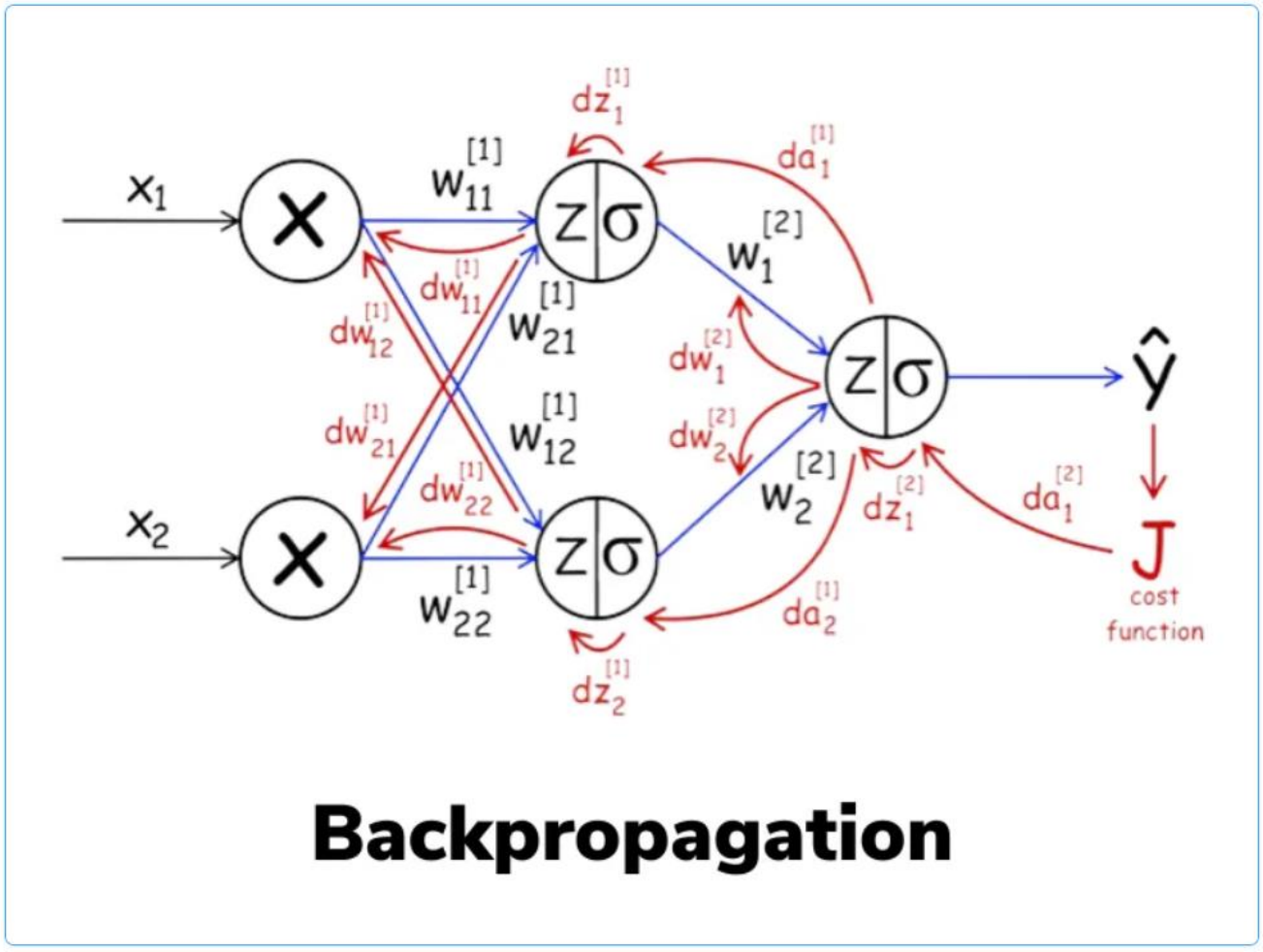
➢ *Denoising (lọc nhiễu)*: Một phiên bản input được chuyển thành 0 ngẫu nhiên.

➢ *Contractive*: Bổ sung hệ số phạt vào hàm mất mát để hạn chế overfitting trong trường hợp số lượng hidden layer lớn hơn input layer.

➢ *Stacked*: Xếp chồng nhiều hidden layer lên nhau để tạo thành một mạng autoencoder.

Các ứng dụng phổ biến: Phát hiện đặc trưng, xây dựng hệ thống khuyến nghị, bổ sung đặc trưng cho tập dữ liệu,…

4.8 Backpropagation

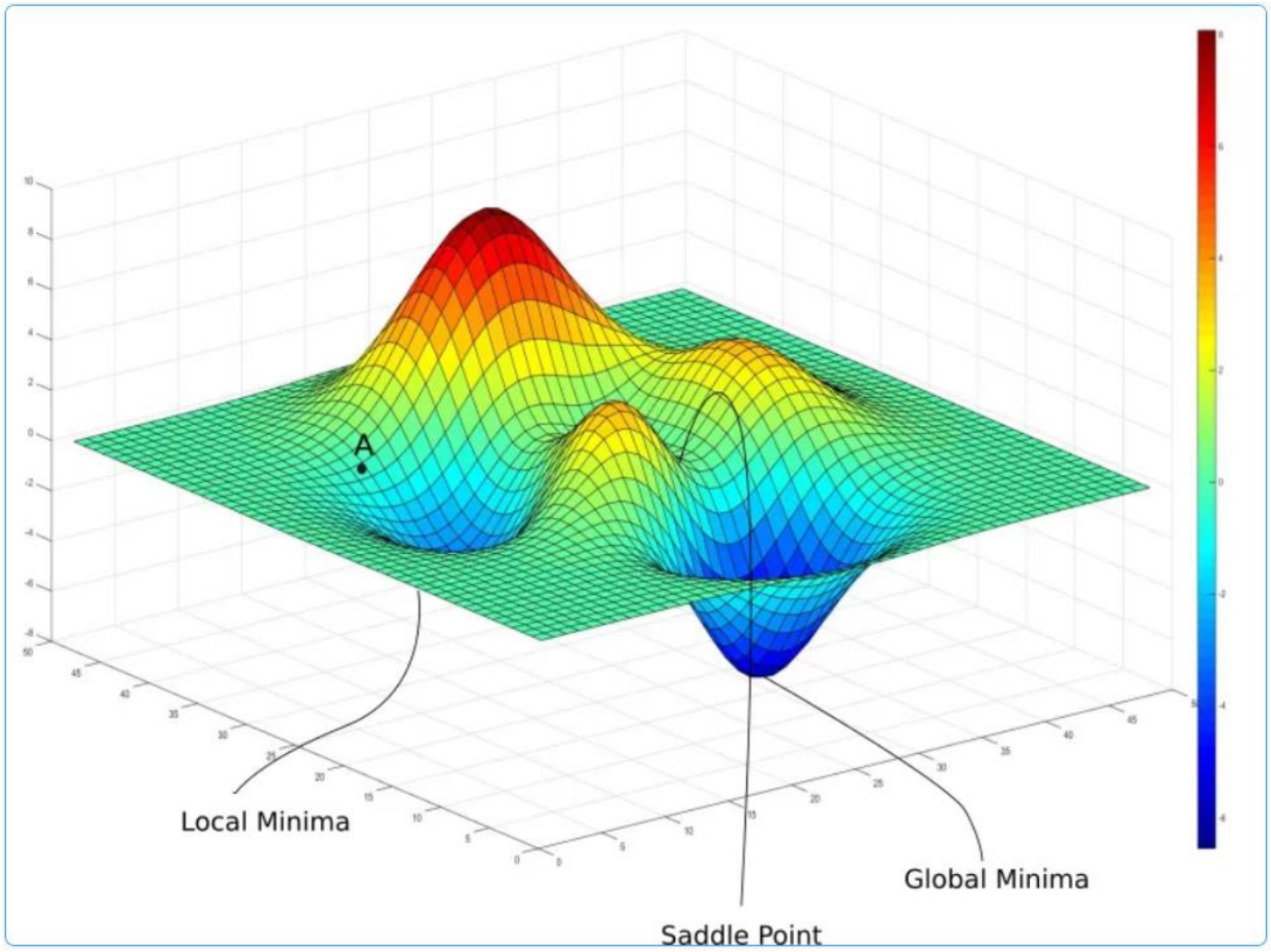


Hình 8: Backpropagation

Backpropagation (Lan truyền ngược) là một trong những kỹ thuật quan trọng nhất của mạng nơ-ron. Về cơ bản thì đây là phương pháp giúp tính gradient ngược từ layer cuối cùng đến layer đầu tiên của mạng.

Trước hết, mạng sẽ phân tích các tham số rồi điều chỉnh thông qua hàm mất mát. Tiếp theo, giá trị lỗi được tính toán sẽ lan truyền ngược lại để điều chỉnh các tham số cho phù hợp.

4.9 Gradient Descent



Hình 9: Gradient Descent

Trong Deep Learning và tối ưu hoá, ta thường phải tìm giá trị nhỏ nhất (hoặc lớn nhất) của một hàm số nào đó. Tuy nhiên việc tìm các điểm tối ưu toàn cục của hàm mất mát thường rất phức tạp, đôi khi là bất khả thi. Do đó ta có thể cố gắng tìm những điểm cực tiểu địa phương và có thể xem là nghiệm cần tìm của bài toán.

Các điểm cực tiểu địa phương về mặt toán học là nghiệm học phương trình đạo hàm bằng 0, tuy nhiên việc giải phương trình đạo hàm bằng 0 gần như là không thể trong Machine Learning hay Deep Learning. Một cách tiếp cận phổ biến là xuất phát từ một điểm mà ta coi là gần với

nghiệm của bài toán, sau đó dùng một phép lặp để tiến dần đến điểm cần tìm. Phương pháp này được gọi là hạ gradient và được sử dụng vô cùng phổ biến trong tối ưu.

Với các mạng nơ-ron hiện đại, nhờ vào thuật toán lan truyền ngược mà gradient descent có thể nhanh hơn hàng triệu lần so với cách truyền thống.

5. Ưu nhược điểm của Deep Learning

5.1 Ưu điểm

Deep Learning là một bước ngoặt to lớn trong lĩnh vực trí tuệ nhân tạo, cho phép khác nhà khoa học dữ liệu xây dựng nhiều mô hình có độ chính xác rất cao trong lĩnh vực nhận dạng ảnh, xử lý ngôn ngữ tự nhiên, xử lý giọng nói và nhiều lĩnh vực khác.

Một số ưu điểm vượt trội của Deep Learning:

➢Kiến trúc mạng nơ-ron linh hoạt, có thể dễ dàng thay đổi để phù hợp với nhiều vấn đề khác nhau.

- Có khả năng giải quyết nhiều bài toán phức tạp với độ chính xác rất cao.
- Tính tự động hoá cao, có khả năng tự điều chỉnh và tự tối ưu.
- Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý được lượng dữ liệu lớn.

5.2 Nhược điểm

Không chỉ có những ưu điểm vượt trội mà Deep Learning vẫn còn nhiều khó khăn và hạn chế, chẳng hạn như:

- Cần có khối lượng dữ liệu rất lớn để tận dụng tối đa khả năng của Deep Learning.
- Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
- Chưa có nền tảng lý thuyết mạnh mẽ để lựa chọn các công cụ tối ưu cho Deep Learning.

6. Ứng dụng của Deep Learning vào thực tế

Kiến trúc mạng nơ-ron trong Deep Learning được ứng dụng trong các công việc yêu cầu sức mạnh tính toán cao, xử lý nhiều dữ liệu và độ phức tạp lớn.

Tiếp theo cùng tìm hiểu 5 ứng dụng thân thuộc nhất của Deep Learning trong đời sống hàng ngày:

6.1 Xe tự lái

Một trong những công nghệ mới và hấp dẫn nhất hiện nay là xe tự động lái, nó được xây dựng dựa trên các mạng nơ-ron cấp cao. Nói một cách đơn giản, các mô hình Deep Learning sẽ nhận diện các đối tượng ở môi trường xung quanh xe, tính toán khoảng cách giữa xe và các phương tiện khác, xác định vị trí làn đường, tín hiệu giao thông,… từ đó đưa ra được các quyết định tối ưu và nhanh chóng nhất. Một trong những hãng xe tiên phong trong việc sản xuất xe tự lái hiện nay là Tesla.

6.2 Phân tích cảm xúc

Là lĩnh vực nghiên cứu về việc đánh giá cảm xúc của con người thông qua xử lý ngôn ngữ tự nhiên, phân tích văn bản và thống kê. Các doanh nghiệp có thể áp dụng Deep Learning để hiểu và dự đoán cảm xúc của khách hàng dựa trên đánh giá, bình luận,…và các nguồn thông tin khác, từ đó phát triển chiến lược kinh doanh và tiếp thị phù hợp với từng đối tượng khách hàng.

6.3 Trợ lý ảo

Là ứng dụng được sử dụng rất nhiều trong đời sống hàng ngày, trong đó phổ biến gồm có chatbot, giảng viên online, Google Assistant, Siri, Cortana,… Các trợ lý ảo được xây dựng dựa trên Deep Learning với các thuật toán nhận diện văn bản, xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói,…

6.4 Mạng xã hội

Một số nền tảng mạng xã hội lớn như Twitter, Instagram, Facebook cũng ứng dụng các thuật toán Deep Learning để cải thiện các dịch vụ của mình. Cụ thể, những trang này sẽ phân tích một lượng lớn dữ liệu thông qua mạng nơ-ron nhân tạo để tìm hiểu về một số các vấn đề như:

- Các tuỳ chọn của người dùng.
- Các hành vi bạo lực trên không gian mạng, chặn các bình luận vi phạm, …
- Gợi ý trang, bạn bè, dịch vụ, nhân diện khuôn mặt,…

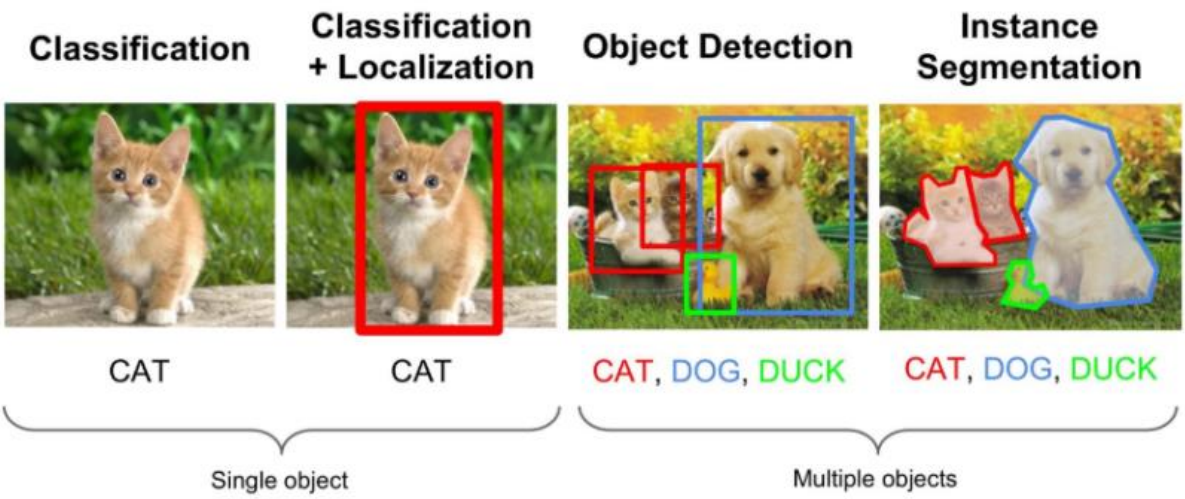
6.5 Chăm sóc sức khoẻ

Deep Learning cũng có đóng góp không nhỏ vào lĩnh vực y tế, trong đó phổ biến gồm có các mô hình dự đoán tình trạng bệnh, chẩn đoán ung thư, phân tích kết quả MRI, X-ray,…

II. Tổng quan thuật toán Object Detection

1. Giới thiệu về thuật toán Object Detection

1.1 Giới thiệu



Hình 10: Object Detection

Object Detection là một trong những lĩnh vực quan trọng của Trí tuệ nhân tạo (Artificial Intelligence) là thị giác máy (Computer Vision). Computer Vision là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh, phát hiện các đối tượng, tạo ảnh, siêu phân giải hình ảnh và nhiều hơn vậy. Object Detection có lẽ là khía cạnh sâu sắc nhất của thị giác máy do số lần sử dụng trong thực tế.

Đây là một nhiệm vụ thị giác máy quan trọng được sử dụng để phát hiện các phiên bản của đối tượng trực quan thuộc một số lớp nhất định (ví dụ: con người, động vật, ô tô hoặc tòa nhà) trong hình ảnh kỹ thuật số như ảnh hoặc khung video. Mục tiêu của Object Detection là phát triển các mô hình tính toán cung cấp thông tin cơ bản nhất mà các ứng dụng thị giác máy cần: " Đối tượng nào ở đâu? "

Các nhiệm vụ cơ bản của một Computer Vision:

- *Phân loại hình ảnh (image classification):* liên quan đến việc gán nhãn cho hình ảnh.
- *Định vị vật thể (object localization):* liên quan đến việc vẽ một hộp giới hạn (bounding box) xung quanh một hoặc nhiều đối tượng trong hình ảnh nhằm khoanh vùng đối tượng.
- *Phát hiện đối tượng (object detection):* Là nhiệm vụ khó khăn hơn và là sự kết hợp của cả hai nhiệm vụ trên: Vẽ một bounding box xung quanh từng đối tượng quan tâm trong ảnh và gán cho chúng một nhãn. Kết hợp cùng nhau, tất cả các vấn đề này được gọi là object recognition hoặc object detection.

1.2 Phát hiện người

Phát hiện người là một biến thể của phát hiện đối tượng được sử dụng để phát hiện "người" lớp chính trong hình ảnh hoặc khung hình video. Phát hiện người trong luồng video là một nhiệm vụ quan trọng trong các hệ thống giám sát video hiện đại. Các thuật toán học sâu gần đây cung cấp kết quả phát hiện người mạnh mẽ. Hầu hết các kỹ thuật phát hiện người hiện đại đều được đào tạo về các góc nhìn chính diện và không đối xứng.

Tuy nhiên, các mô hình deep learning như YOLO được đào tạo để phát hiện người trên tập dữ liệu nhìn trực diện vẫn mang lại kết quả tốt khi áp dụng cho tính năng đếm người nhìn từ trên cao (TPR là 95%, FPR lên tới 0,2%).

Hình 11: Phát hiện người

1.3 Tại sao Object Detection lại quan trọng?

Phát hiện đối tượng là một trong những vấn đề cơ bản của thị giác máy tính. Nó tạo thành nền tảng của nhiều tác vụ thị giác máy tính xuôi dòng khác, chẳng hạn như phân đoạn phiên bản và hình ảnh , chú thích hình ảnh, theo dõi đối tượng , v.v. Các ứng dụng phát hiện đối tượng cụ thể bao gồm phát hiện người đi bộ, phát hiện động vật, phát hiện phương tiện, đếm người , phát hiện khuôn mặt , phát hiện văn bản , phát hiện tư thế hoặc nhận dạng biển số .

Khi thị giác máy tính trưởng thành, nó đã chuyển từ "Slope of Enlightenment" sang "Plateau of Productivity" trong Gartner Hype Cycle. Với những tiến bộ trong công nghệ, việc áp dụng ngày càng tăng và ứng dụng thực tế trong các ngành công nghiệp, thị giác máy tính, bao gồm cả phát hiện đối tượng, đang bước vào giai đoạn ổn định và tích hợp rộng rãi. Trọng tâm hiện chuyển từ các giai đoạn thử nghiệm sang tinh chỉnh và tối ưu hóa các ứng dụng hiện có, đánh

dấu một bước quan trọng hướng tới việc hiện thực hóa và tác động đầy đủ đến các doanh nghiệp trong các lĩnh vực khác nhau.

1.4 Object Detection & Deep Learning

Trong vài năm gần đây, những tiến bộ nhanh chóng trong kỹ thuật học sâu đã thúc đẩy đáng kể động lực của công nghệ phát hiện đối tượng. Với mạng học sâu và sức mạnh tính toán của GPU, hiệu suất của máy dò và theo dõi đối tượng đã được cải thiện đáng kể, đạt được những đột phá đáng kể trong việc phát hiện đối tượng.

Học máy (ML) là một nhánh của trí tuệ nhân tạo (AI) và về cơ bản nó liên quan đến các mẫu học từ các ví dụ hoặc dữ liệu mẫu khi máy truy cập dữ liệu và có khả năng học từ dữ liệu đó (học có giám sát trên hình ảnh có chú thích). Deep Learning là một hình thức học máy chuyên biệt bao gồm việc học ở các giai đoạn khác nhau.

1.5 Những tiến bộ công nghệ mới nhất trong thị giác máy tính

Phát hiện và theo dõi đối tượng Deep Learning là nền tảng cơ bản của một loạt các ứng dụng thị giác máy tính hiện đại . Ví dụ: việc phát hiện vật thể cho phép giám sát chăm sóc sức khỏe thông minh, lái xe tự động, giám sát video thông minh, phát hiện bất thường, thị giác robot, v.v. Mỗi ứng dụng thị giác AI thường yêu cầu kết hợp nhiều thuật toán khác nhau tạo thành một luồng (đường ống) gồm nhiều bước xử lý.

Công nghệ hình ảnh AI đã phát triển vượt bậc trong những năm gần đây. Có thể sử dụng nhiều loại camera, bao gồm cả camera an ninh thương mại và camera quan sát. Bằng cách sử dụng nền tảng phần mềm AI tương thích, bạn không cần phải mua máy ảnh AI có khả năng nhận dạng hình ảnh tích hợp vì luồng video kỹ thuật số của bất kỳ máy quay video nào, về cơ bản đều có thể được phân tích bằng mô hình phát hiện đối tượng.

Nhờ đó, các ứng dụng trở nên linh hoạt hơn vì chúng không còn phụ thuộc vào các cảm biến tùy chỉnh, hệ thống lắp đặt đắt tiền và hệ thống phần cứng nhúng phải thay thế sau mỗi 3-5 năm.

Trong khi đó, sức mạnh tính toán đã tăng lên đáng kể và ngày càng trở nên hiệu quả hơn. Trong những năm qua, các nền tảng điện toán đã chuyển sang song song hóa thông qua xử lý đa lõi, bộ xử lý đồ họa (GPU) và bộ tăng tốc AI như bộ xử lý tensor (TPU)

Phần cứng như vậy cho phép áp dụng thị giác máy tính để phát hiện và theo dõi đối tượng trong môi trường gần thời gian thực. Do đó, sự phát triển nhanh chóng của mạng nơ-ron tích chập sâu (CNN) và sức mạnh tính toán nâng cao của GPU là động lực chính thúc đẩy sự tiến bộ vượt bậc của việc phát hiện đối tượng dựa trên thị giác máy tính.

Những tiến bộ đó đã tạo nên một khái niệm kiến trúc quan trọng được gọi là Edge AI . Khái niệm này còn được gọi là Intelligent Edge hoặc Distributed Edge. Nó di chuyển khối lượng công việc AI nặng từ Cloud đến gần nguồn dữ liệu hơn. Điều này dẫn đến các hệ thống phân tán, có thể mở rộng và hiệu quả hơn nhiều, cho phép sử dụng thị giác máy tính trong các hệ thống kinh doanh và nhiệm vụ quan trọng.

Edge AI liên quan đến IoT hoặc AIoT , học máy trên thiết bị với Thiết bị Edge và yêu cầu cơ sở hạ tầng phức tạp.

1.6 Nhược điểm và ưu điểm của phát hiện đối tượng

Máy dò đối tượng cực kỳ linh hoạt và có thể được đào tạo để thực hiện nhiều nhiệm vụ cũng như các ứng dụng tùy chỉnh, có mục đích đặc biệt. Việc tự động nhận dạng đồ vật, người và cảnh có thể cung cấp thông tin hữu ích để tự động hóa các tác vụ (đếm, kiểm tra, xác minh, v.v.) trên chuỗi giá trị của doanh nghiệp.

Tuy nhiên, nhược điểm chính của máy dò đối tượng là chúng có chi phí tính toán rất cao và đòi hỏi sức mạnh xử lý đáng kể. Đặc biệt, khi các mô hình phát hiện đối tượng được triển khai trên quy mô lớn, chi phí vận hành có thể tăng nhanh và thách thức khả năng tồn tại về mặt kinh tế của các trường hợp sử dụng trong kinh doanh.

2. Cách phát hiện đối tượng hoạt động

Việc phát hiện đối tượng có thể được thực hiện bằng cách sử dụng kỹ thuật xử lý hình ảnh truyền thống hoặc mạng học sâu hiện đại.

Các kỹ thuật xử lý hình ảnh thường không yêu cầu dữ liệu lịch sử để đào tạo và về bản chất là không được giám sát. OpenCV là một công cụ phổ biến cho các tác vụ xử lý ảnh.

- Ưu điểm: Do đó, những tác vụ đó không yêu cầu hình ảnh có chú thích , trong đó con người gắn nhãn dữ liệu theo cách thủ công (để đào tạo có giám sát).
- Nhược điểm: Những kỹ thuật này bị hạn chế ở nhiều yếu tố, chẳng hạn như các tình huống phức tạp (không có nền đơn sắc), sự che khuất (các vật thể bị ẩn một phần), độ chiếu sáng và bóng tối cũng như hiệu ứng lộn xộn.

Các phương pháp Deep Learning thường phụ thuộc vào việc học có giám sát hoặc không giám sát, với các phương pháp được giám sát là tiêu chuẩn trong các nhiệm vụ thị giác máy tính. Hiệu suất bị hạn chế bởi sức mạnh tính toán của GPU, vốn đang tăng nhanh qua từng năm.

- Ưu điểm: Khả năng phát hiện đối tượng deep learning mạnh mẽ hơn đáng kể đối với tình trạng tắc nghẽn, cảnh phức tạp và ánh sáng khó khăn.

➢ Nhược điểm: Cần một lượng lớn dữ liệu đào tạo; quá trình chú thích hình ảnh tốn nhiều công sức và tốn kém. Ví dụ: gắn nhãn 500.000 hình ảnh để huấn luyện thuật toán phát hiện đối tượng DL tùy chỉnh được coi là một tập dữ liệu nhỏ. Tuy nhiên, nhiều bộ dữ liệu điểm chuẩn (MS COCO , Caltech, KITTI, PASCAL VOC, V5) cung cấp dữ liệu được dán nhãn sẵn có.

Ngày nay, việc phát hiện đối tượng Deep Learning được các nhà nghiên cứu chấp nhận rộng rãi và được các công ty thị giác máy tính áp dụng để xây dựng các sản phẩm thương mại.

Hình 12: Phát hiện đối tượng hoạt động

3. Các trường hợp sử dụng và ứng dụng Object Detection

Các trường hợp sử dụng liên quan đến phát hiện đối tượng rất đa dạng; có những cách gần như không giới hạn để làm cho máy tính hoạt động giống con người nhằm tự động hóa các tác vụ thủ công hoặc tạo ra các sản phẩm và dịch vụ mới được hỗ trợ bởi AI. Nó đã được triển khai trong các chương trình thị giác máy tính được sử dụng cho nhiều ứng dụng, từ sản xuất thể thao đến phân tích năng suất.

Ngày nay, nhận dạng đối tượng là cốt lõi của hầu hết các chương trình và phần mềm AI dựa trên thị giác. Phát hiện đối tượng đóng một vai trò quan trọng trong việc hiểu cảnh, vốn phổ biến trong các trường hợp sử dụng an ninh , xây dựng , giao thông, y tế và quân sự.

➢ Phát hiện đối tượng trong Bán lẻ. Hệ thống đếm người được bố trí một cách chiến lược trong nhiều cửa hàng bán lẻ được sử dụng để thu thập thông tin về cách khách hàng sử dụng thời gian và lượng khách hàng đến. Phân tích khách hàng dựa trên AI để phát hiện và theo dõi khách hàng bằng camera giúp hiểu rõ hơn về tương tác và trải nghiệm của khách hàng, tối ưu

hóa cách bố trí cửa hàng và giúp hoạt động hiệu quả hơn. Trường hợp sử dụng phổ biến là phát hiện hàng đợi để giảm thời gian chờ đợi trong các cửa hàng bán lẻ.

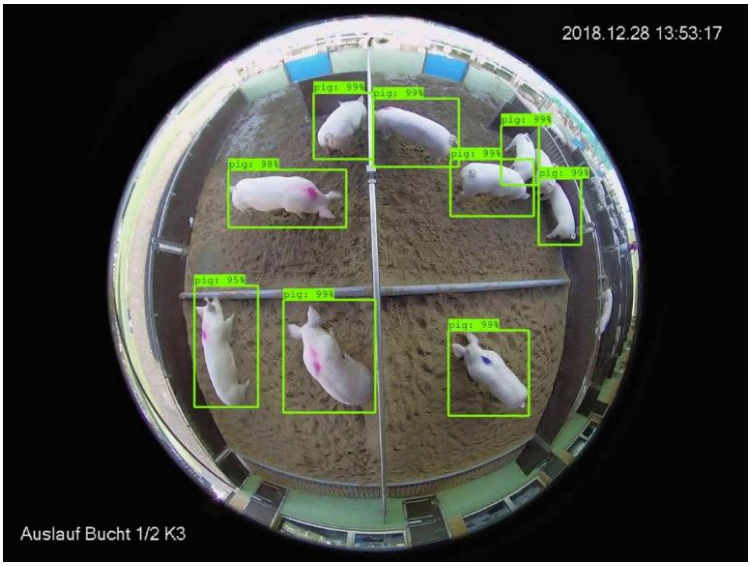
➢ Lái xe tự động. Xe tự lái phụ thuộc vào tính năng phát hiện vật thể để nhận dạng người đi bộ, biển báo giao thông, các phương tiện khác, v.v. Ví dụ: Autopilot AI của Tesla sử dụng rất nhiều tính năng phát hiện vật thể để nhận biết các mối đe dọa về môi trường và xung quanh, chẳng hạn như các phương tiện hoặc chướng ngại vật đang tới.

➢ Phát hiện động vật trong nông nghiệp. Phát hiện đối tượng được sử dụng trong nông nghiệp cho các nhiệm vụ như đếm, theo dõi động vật và đánh giá chất lượng nông sản. Sản phẩm bị hư hỏng có thể được phát hiện trong khi đang được xử lý bằng thuật toán học máy.

➢ Phát hiện người trong Bảo mật. Một loạt các ứng dụng bảo mật trong giám sát video dựa trên việc phát hiện đối tượng, chẳng hạn như để phát hiện người ở khu vực hạn chế hoặc nguy hiểm, ngăn ngừa tự sát hoặc tự động hóa các nhiệm vụ kiểm tra ở những địa điểm xa bằng thị giác máy tính.

➢ Phát hiện phương tiện bằng AI trong Giao thông vận tải. Nhận dạng đối tượng được sử dụng để phát hiện và đếm phương tiện để phân tích giao thông hoặc phát hiện ô tô dừng ở khu vực nguy hiểm, ví dụ như trên đường giao nhau hoặc đường cao tốc.

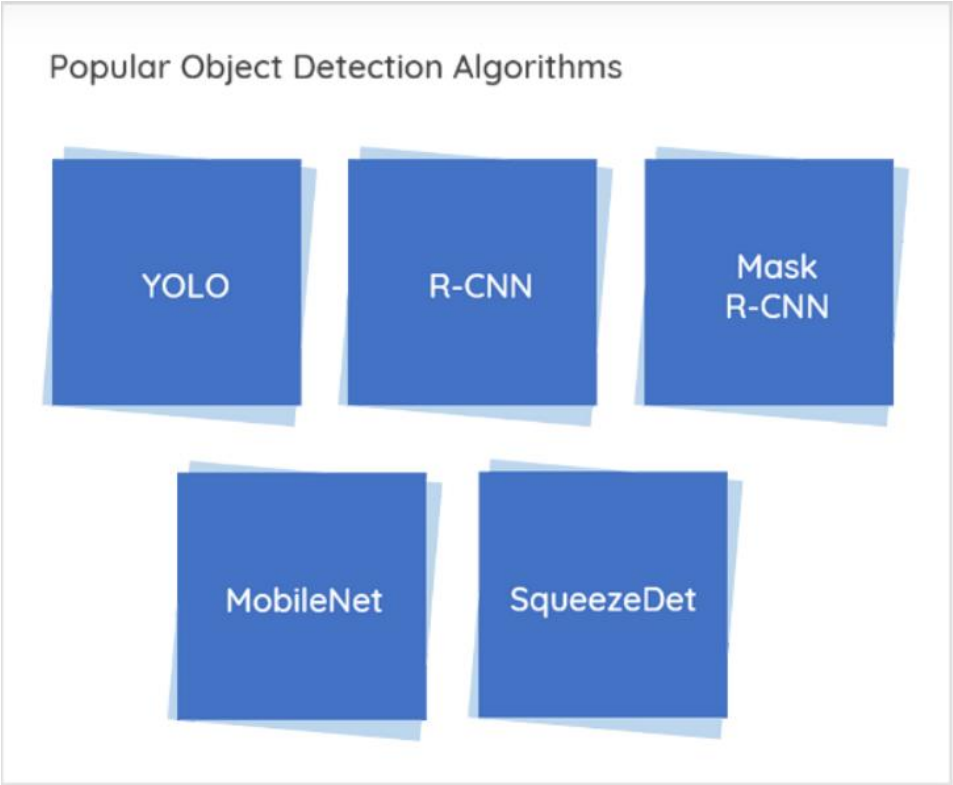
➢ Phát hiện tính năng y tế trong Chăm sóc sức khỏe. Phát hiện vật thể đã mang lại nhiều đột phá trong cộng đồng y tế. Bởi vì chẩn đoán y tế chủ yếu dựa vào nghiên cứu hình ảnh, bản quét và ảnh chụp, nên việc phát hiện đối tượng liên quan đến quét CT và MRI đã trở nên cực kỳ hữu ích để chẩn đoán bệnh, chẳng hạn như với thuật toán ML để phát hiện khối u.



Hình 13: Ứng dụng Deep Learning thương mại phát hiện đối tượng trong giám sát động vật

4. Thuật toán phát hiện đối tượng phổ biến nhất

Các thuật toán phổ biến được sử dụng để thực hiện phát hiện đối tượng bao gồm mạng thần kinh tích chập (R-CNN, Region-Based Convolutional Neural Networks),Fast R-CNN và YOLO (You Only Look Once) . R-CNN's thuộc họ R-CNN, trong khi YOLO là một phần của họ máy dò phát một lần. Sau đây, chúng tôi sẽ cung cấp cái nhìn tổng quan và sự khác biệt giữa các thuật toán phát hiện đối tượng phổ biến.



Hình 14: Tổng quan về phát hiện đối tượng của các thuật toán phổ biến

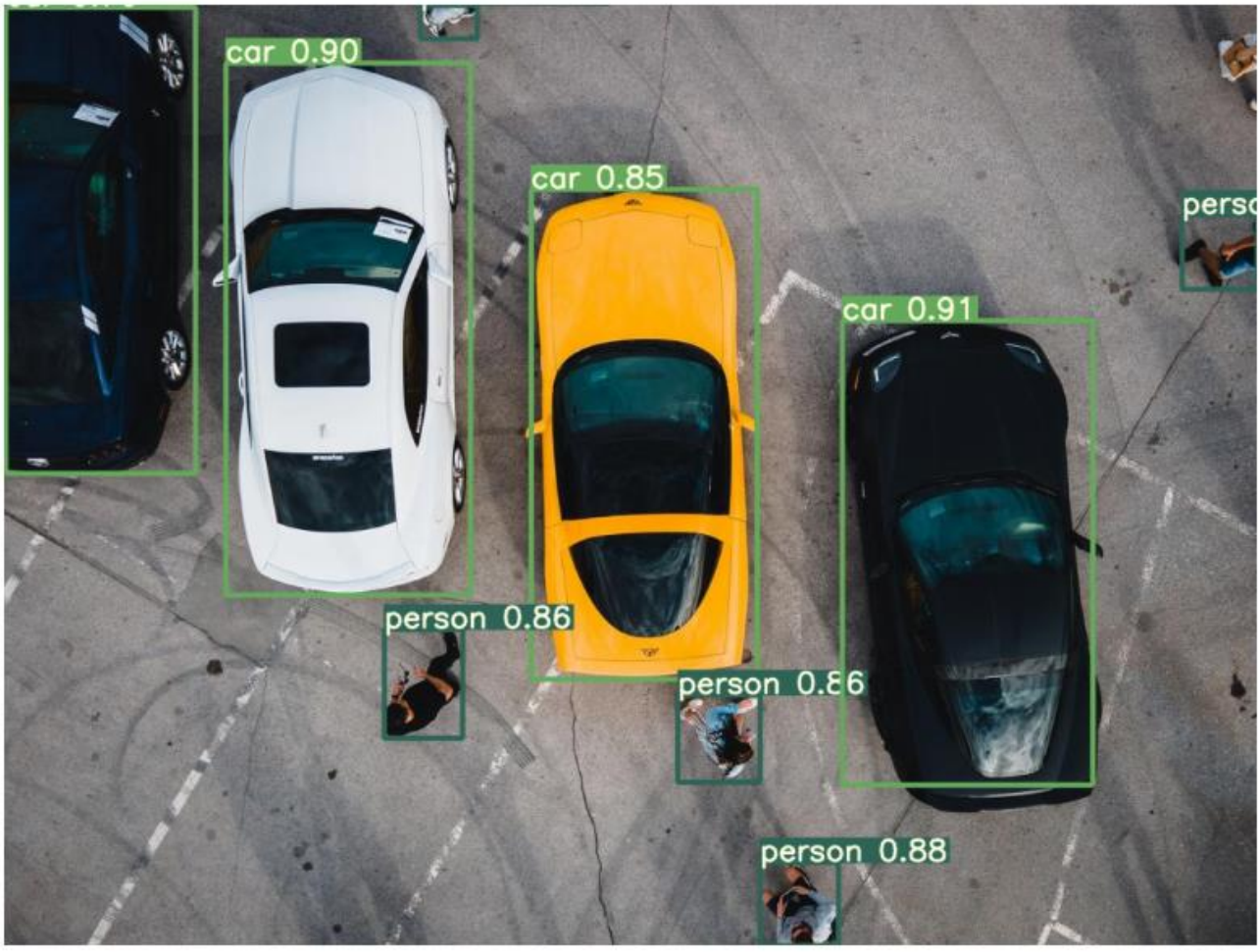
4.1 YOLO – You Only Look Once

YOLO là viết tắt của "You Only Look Once", đây là một loại thuật toán phát hiện đối tượng theo thời gian thực phổ biến được sử dụng trong nhiều sản phẩm thương mại của các công ty công nghệ lớn nhất sử dụng thị giác máy tính. Trình phát hiện đối tượng YOLO ban đầu được phát hành lần đầu tiên vào năm 2016 và kiến trúc mới nhanh hơn đáng kể so với bất kỳ trình phát hiện đối tượng nào khác.

Kể từ đó, nhiều phiên bản và biến thể của YOLO đã được phát hành, mỗi phiên bản đều mang lại sự gia tăng đáng kể về hiệu suất và hiệu quả. YOLOv4 là phiên bản cải tiến của YOLOv3 chính thức . Các nhóm nghiên cứu đã phát hành phiên bản YOLO của riêng họ, ví dụ: YOLOv5 , YOLOv7 hoặc YOLOv8 . Những cải tiến chính là nâng cao dữ liệu khảm, đào tạo tự đối nghịch và chuẩn hóa nhiều đợt nhỏ.

YOLOv7 là một trong những mô hình phát hiện đối tượng theo thời gian thực nhanh nhất và chính xác nhất cho các tác vụ thị giác máy tính. Bài báo YOLOv7 chính thức được phát hành vào tháng 7 năm 2022 bởi Chien-Yao Wang, Alexey Bochkovskiy và Hong-Yuan Mark Liao.

Một mô hình nổi bật khác là YOLOv8 được phát triển bởi Ultralytics. Nó được thiết kế để hoạt động nhanh, chính xác và dễ sử dụng, khiến nó trở thành sự lựa chọn tuyệt vời cho nhiều nhiệm vụ phát hiện và theo dõi đối tượng, phân đoạn đối tượng, phân loại hình ảnh và ước tính tư thế.



Hình 15: Phát hiện xe và người dựa trên camera với YOLOv7

4.2 R-CNN – Region-based Convolutional Neural Networks

Các mạng thần kinh tích chập dựa trên vùng hoặc các vùng có tính năng CNN (R-CNN) là những phương pháp tiên phong áp dụng các mô hình sâu để phát hiện đối tượng. Các mô hình R-CNN trước tiên chọn một số vùng được đề xuất từ một hình ảnh (ví dụ: hộp neo là một loại phương pháp lựa chọn) và sau đó gắn nhãn cho danh mục và hộp giới hạn của chúng (ví dụ: độ lệch). Các nhãn này được tạo dựa trên các lớp được xác định trước cho chương trình. Sau đó, họ sử dụng mạng nơ ron tích chập (CNN) để thực hiện tính toán chuyển tiếp nhằm trích xuất các đặc điểm từ từng khu vực được đề xuất.

Trong R-CNN, hình ảnh đầu vào trước tiên được chia thành gần hai nghìn phần vùng, sau đó mạng nơ ron tích chập được áp dụng tương ứng cho từng vùng. Kích thước của các vùng được tính toán và vùng chính xác sẽ được chèn vào mạng lưới thần kinh. Có thể suy ra rằng một phương pháp chi tiết như vậy có thể tạo ra những hạn chế về thời gian. Thời gian đào tạo lớn hơn đáng kể so với YOLO vì nó phân loại và tạo các hộp giới hạn riêng lẻ và mạng lưới thần kinh được áp dụng cho một vùng tại một thời điểm.

Vào năm 2015, Fast R-CNN được phát triển để giảm đáng kể thời gian chạy tàu. Trong khi R-CNN ban đầu tính toán độc lập các tính năng của mạng thần kinh trên mỗi vùng trong số khoảng hai nghìn vùng quan tâm thì Fast R-CNN chạy mạng thần kinh một lần trên toàn bộ hình ảnh. Điều này rất giống với kiến trúc của YOLO, nhưng YOLO vẫn là sự thay thế nhanh hơn cho Fast R-CNN vì tính đơn giản của mã.

Ở cuối mạng là một phương pháp mới được gọi là Nhóm vùng quan tâm (ROI), phương pháp này sẽ tách từng Vùng quan tâm khỏi tenxơ đầu ra của mạng, định hình lại và phân loại nó (Phân loại hình ảnh). Điều này làm cho Fast R-CNN chính xác hơn R-CNN ban đầu. Tuy nhiên, do kỹ thuật nhận dạng này nên cần ít dữ liệu đầu vào hơn để huấn luyện các máy dò R-CNN và Fast R-CNN.

4.3 Mask R-CNN

Mặt nạ R-CNN là một cải tiến của Fast R-CNN. Sự khác biệt giữa hai loại này là Mask R-CNN đã thêm một nhánh để dự đoán mặt nạ đối tượng song song với nhánh hiện có để nhận dạng hộp giới hạn. Mask R-CNN dễ huấn luyện và chỉ tốn thêm một khoản chi phí nhỏ cho Faster R-CNN; nó có thể chạy ở tốc độ 5 khung hình/giây.

Hình 16: Ví dụ về Mask R-CNN với phân đoạn hình ảnh và phát hiện đối tượng hình ảnh

4.4 SqueezeDet

SqueezeDet là tên của mạng lưới thần kinh sâu dành cho thị giác máy tính được phát hành vào năm 2016. SqueezeDet được phát triển đặc biệt cho xe tự lái, nơi nó thực hiện phát hiện đối tượng bằng kỹ thuật thị giác máy tính. Giống như YOLO, nó là thuật toán phát hiện một lần.

Trong SqueezeDet, các lớp tích chập chỉ được sử dụng để trích xuất các bản đồ đặc trưng mà còn là lớp đầu ra để tính toán các hộp giới hạn và xác suất của lớp. Đường dẫn phát hiện của các mô hình SqueezeDet chỉ chứa các mạng thần kinh chuyển tiếp đơn lẻ, cho phép chúng hoạt động cực kỳ nhanh.

4.5 MobileNet

MobileNet là mạng phát hiện nhiều hộp một lần chụp được sử dụng để chạy các tác vụ phát hiện đối tượng. Mô hình này được triển khai bằng cách sử dụng khung Caffe. Đầu ra của mô hình là một vectơ điển hình chứa dữ liệu đối tượng được theo dõi , như được mô tả trước đây.

III. Sử dụng mô hình YOLOv5 để nhận diện đối tượng

1. Giới thiệu mô hình YOLOv5

YOLOv5 là một mô hình Object Detection thuộc họ mô hình YOLO, 3 phiên bản YOLO đầu tiên được phát triển bởi Joseph Redmon. Sau đó, Alexey Bochkovskiy cho ra mắt YOLOv4 với sự cải thiện cả về tốc độ cũng như độ chính xác. Và rồi YOLOv5 được công bố gần đây với những so sánh ban đầu cho thấy độ chính xác tương đương YOLOv4 và có tốc độ nhanh hơn khi thực hiện dự đoán (tuy nhiên vẫn có rất nhiều hoài nghi về độ tin cậy của những so sánh này vì YOLOv5 mới được ra mắt trên GitHub chứ chưa có bài báo chính thức nào cả).

Khác với những phiên bản tiền nhiệm, YOLOv5 được phát triển dựa trên PyTorch thay vì DarkNet. Đây là một điểm cộng không nhỏ cho YOLOv5 vì PyTorch phổ biến hơn rất nhiều, điều này đồng nghĩa với việc sẽ có nhiều tài liệu và hướng dẫn tham khảo về mô hình này.

2. Cài đặt nhận diện hình ảnh và video bằng YOLOv5

2.1 Chuẩn bị dữ liệu

Trước hết chúng ta cần phải có dữ liệu ảnh và nhãn (images & labels). Dữ liệu đối với mô hình cũng giống những viên gạch cho việc xây nhà, không có dữ liệu thì không xây dựng được mô hình nào cả.

2.2 Tập hợp ảnh

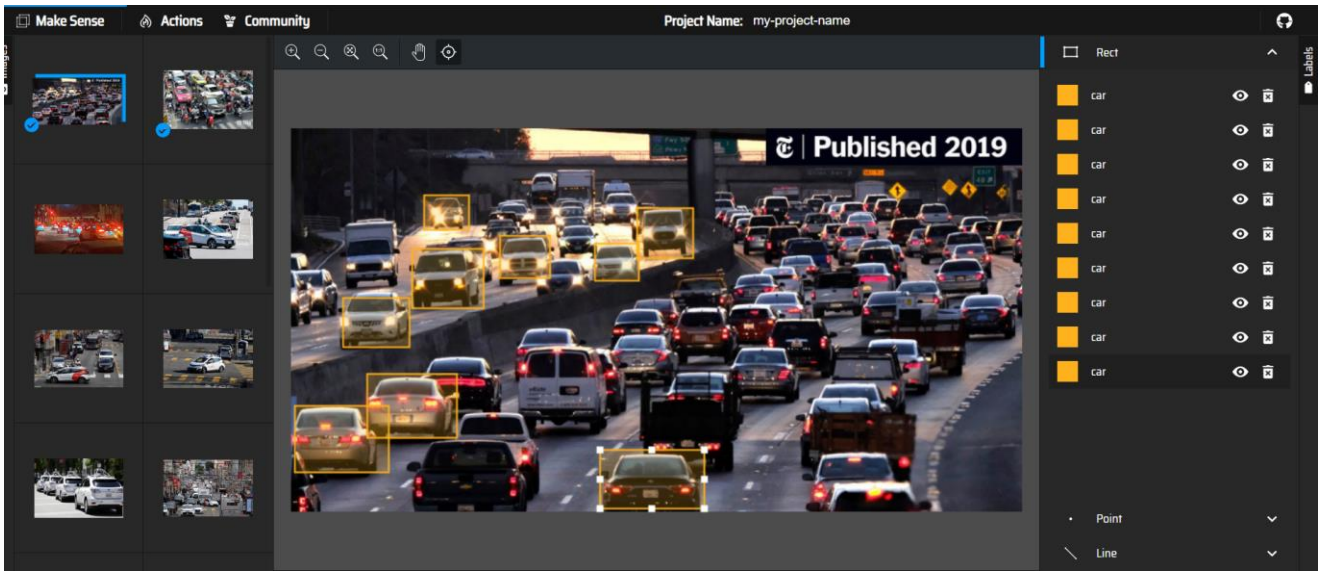
Thu thập ảnh cho việc huấn luyện mô hình:

- Tìm trên Open Images Dataset, tập hợp hơn 9 triệu ảnh với 6000 nhóm khác nhau

- Tìm với công cụ tìm kiếm (Google, Bing, …) với lưu ý cần kiểm tra kỹ về bản quyền sử dụng ảnh.
- Trích xuất các khung hình từ Videos
- Tự chụp ảnh

2.3 Gán nhãn

Để phát hiện đối tượng nào đó, bước đầu tiên chúng ta cần thu thập thật nhiều ảnh của chúng, sau đó gán nhãn cho chúng rồi chia thành các tập train/val hoặc train/val/test.

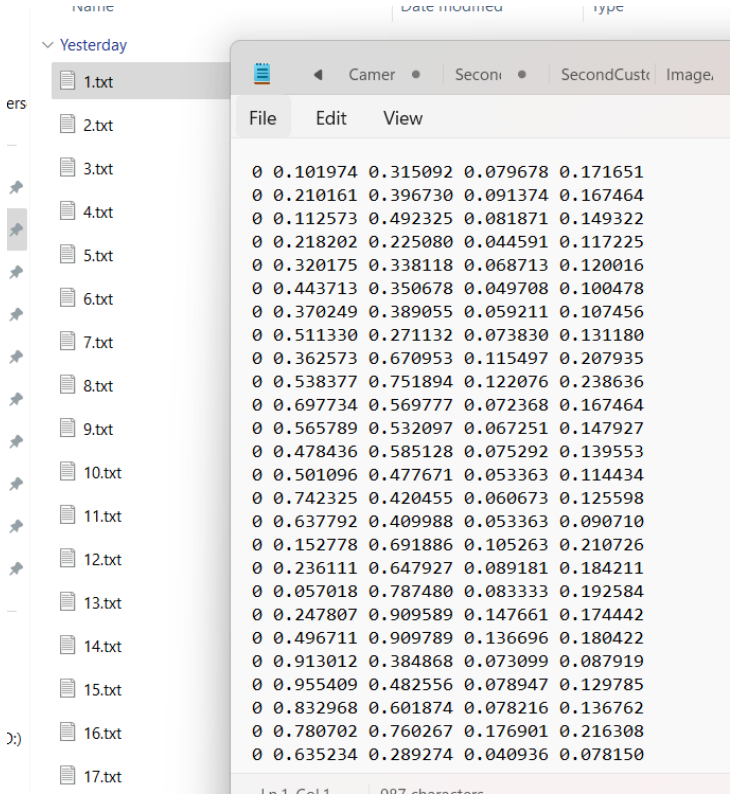


Hình 17: Gán nhãn cho đối tượng

Mỗi ảnh sẽ có một nhãn (file txt) tương ứng để chứa thông tin về đường bao của đối tượng:

- Mỗi dòng chứa thông tin của một đối tượng
- Mỗi dòng có 5 giá trị chứa thông tin về đối tượng và toạ độ đường bao: đối tượng, toạ độ trung tâm x, toạ độ trung tâm y, chiều rộng, chiều dài.
- Lưu ý là toạ độ đường bao đã được chuẩn hoá về khoảng (0, 1) (đây là yêu cầu của mô hình YOLO)

Dưới đây là ví dụ về nhãn và file chứa thông tin về đối tượng:



Hình 18: Nhãn và file chứa tên các nhóm đối tượng

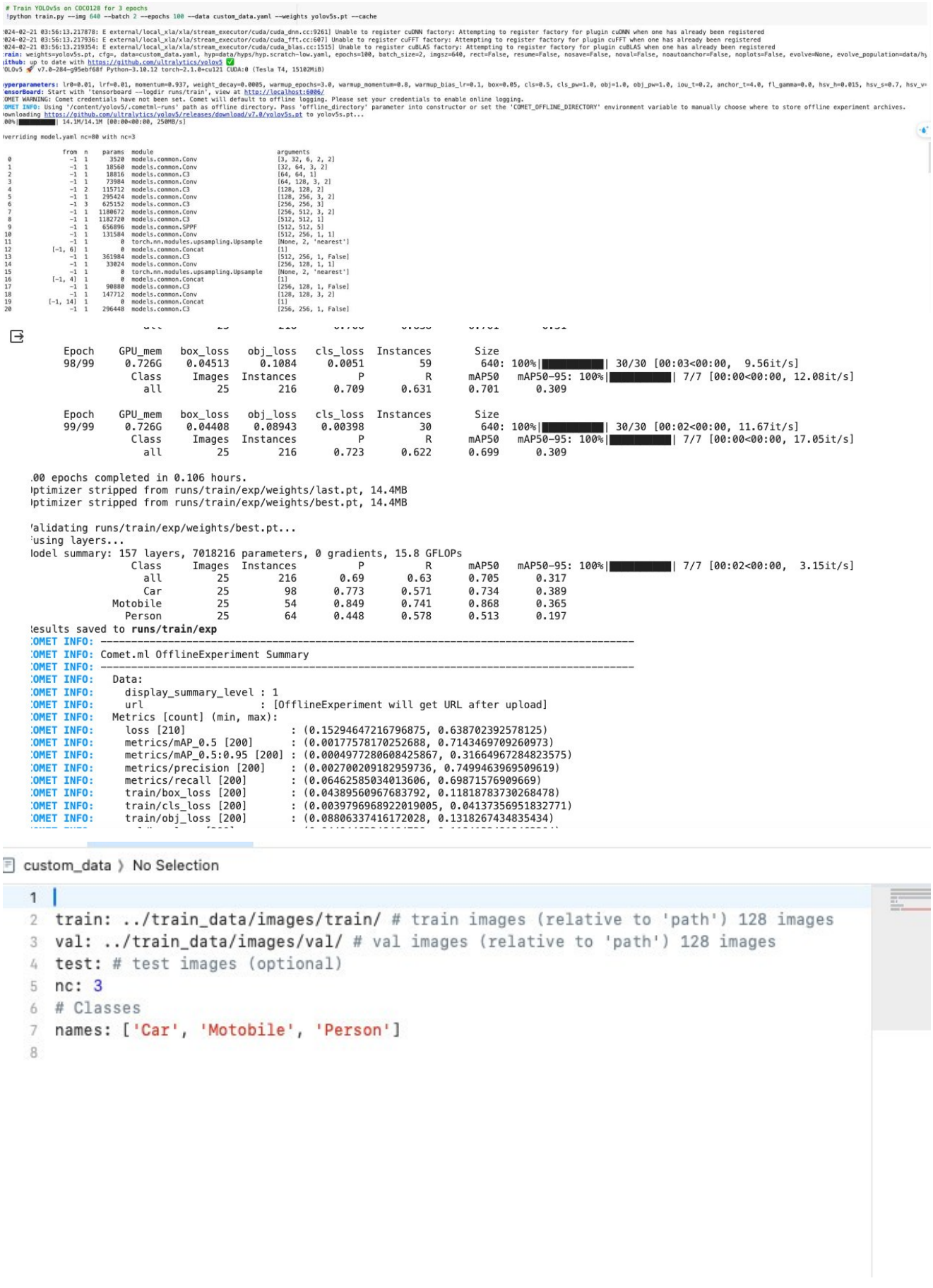
2.4 Huấn luyện mô hình

Tải YOLOv5 từ GitHub và các cài đặt các thư viện liên quan

Hình 19: Setup thư viện liên quan

Lưu ý là chúng ta cần YOLOv5 ở cùng thư mục với custom_dataset ở trên

Tiến hành huấn luyện:



Hình 20: Huấn luyện

- img: kích thước ảnh (độ phân giải)

- batch: số ảnh dùng để huấn luyện trong mỗi lượt
- epochs: số lượt huấn luyện cho tất cả các ảnh trong tập dữ liệu train
- data: đường dẫn đến file cấu hình của tập dữ liệu
- weights: đường dẫn đến file weight chứa độ liên kết giữa các neuron (để " là để huấn luyện từ đầu)
- cache: dùng bộ nhớ đệm để huấn luyện nhanh hơn

IV. Demo quá trình nhận diện đối tượng

Khi quá trình huấn luyện kết thúc, có 2 files weights mô hình được lưu lại:

- Mô hình tốt nhất: best.pt (tốt nhất dựa trên chỉ số Average Precision trên tập kiểm định)
- Mô hình cuối cùng: last.pt

```
python detect.py --source 0  # webcam
                         img.jpg  # image
                         vid.mp4  # video
                         screen  # screenshot
                         path/  # directory
                        'path/*.jpg'  # glob
                        'https://youtu.be/LNwODJXcvt4'  # YouTube
                        'rtsp://example.com/media.mp4'  # RTSP, RTMP, HTTP stream
```

```
[ ]  1  Bắt đầu lập trình hoặc tạo mã bằng trí tuệ nhân tạo (AI).
```

```
[ ]  1  !python detect.py --weights runs/train/exp2/weights/last.pt --img 640 --conf 0.25 --source ../car1.jpg
     2  # display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

Hình 21: Tiến hành nhận diện

1. Nhận diện đối tượng trong hình ảnh

```
1  !python detect.py --weights runs/train/exp2/weights/last.pt --img 640 --conf 0.25 --source ../car1.jpg
2  # display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

Hình 21: Tiến hành nhận diện

- weights: file weight dùng để dự đoán
- img: kích thước ảnh (độ phân giải)
- conf: độ tin cậy
- source: đường dẫn đến thư mục ảnh

Kết quả dự đoán được lưu tại thư mục /yolov5/runs/detect/exp1/ (hoặc exp2, 3, … tuỳ vào số lần chạy dự đoán với detect.py).

Hình 22: Địa chỉ lưu ảnh kết quả

Hình ảnh trước khi nhận diện:

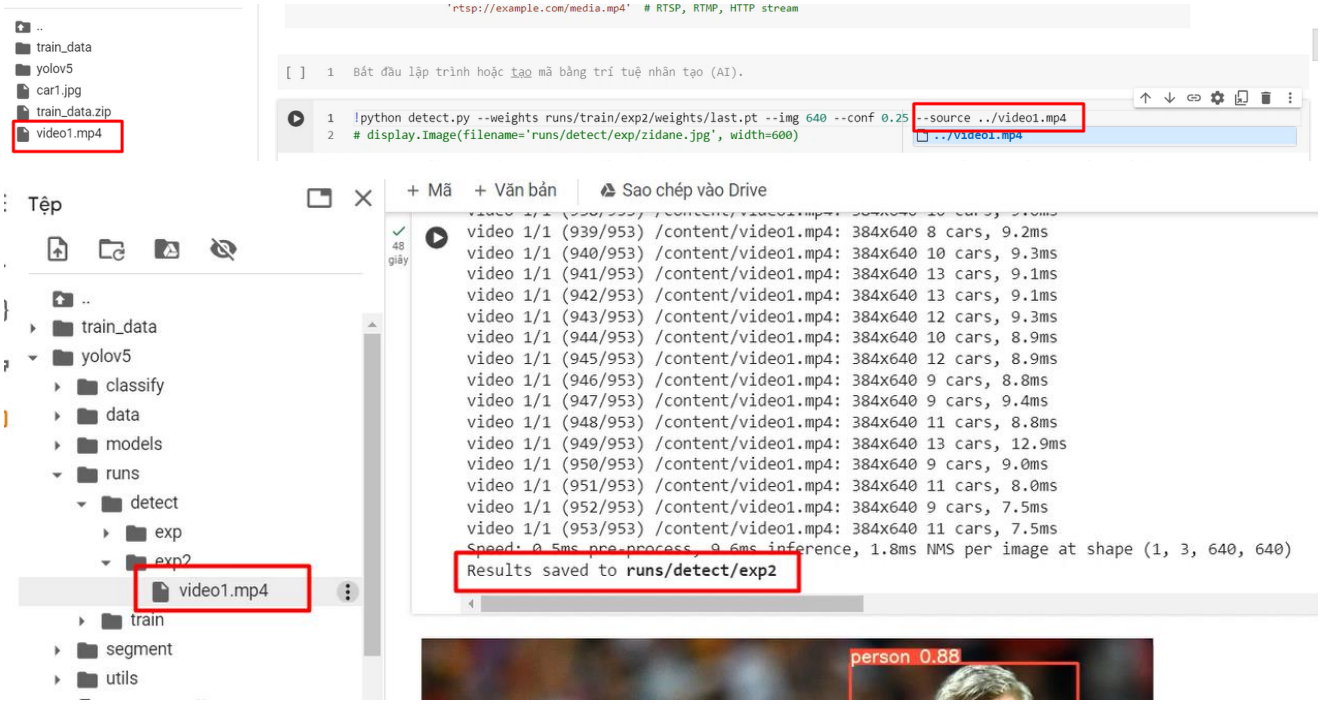
Hình 23: Hình ảnh trước khi nhận diện

Hình ảnh sau khi nhận diện:

Hình 24: Hình ảnh sau khi nhận diện

2. Nhận diện đối tượng trong video

Kết quả dự đoán được lưu tại thư mục /yolov5/runs/detect/exp1/ (hoặc exp2, 3, … tuỳ vào số lần chạy dự đoán với detect.py).

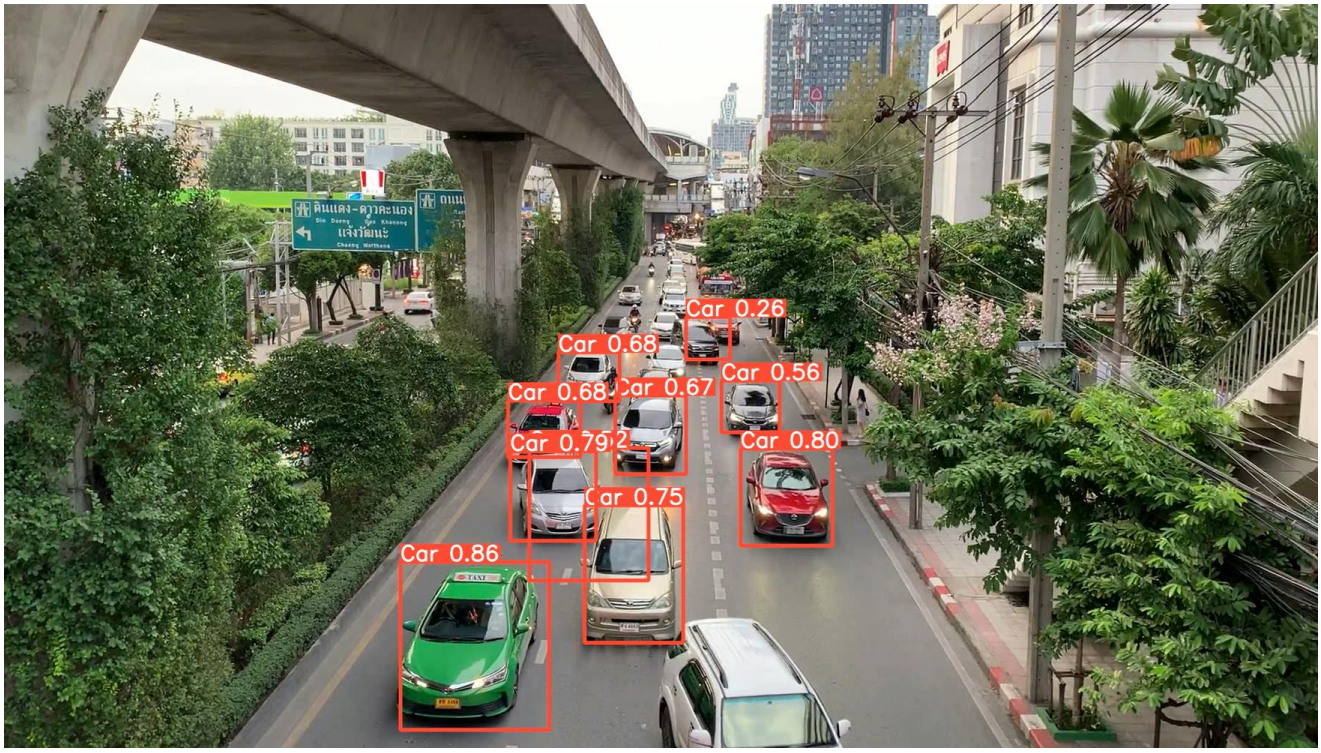


Hình 25: Địa chỉ lưu video kết quả

Video trước khi nhận diện:

Hình 26: Video trước khi nhận diện

Video sau khi nhận diện:



Hình 27: Video sau khi nhận diện

V. Kết luận

Trong quá trình nghiên cứu về nhận diện ảnh và video thông qua Object Detection, chúng tôi đã trải qua một hành trình sâu sắc để hiểu rõ về ứng dụng của công nghệ này và cách nó có thể cung cấp giải pháp cho nhiều vấn đề thực tế. Việc sử dụng thuật toán YOLO đã giúp tôi tiếp cận một cách linh hoạt và hiệu quả với các mô hình tiên tiến trong lĩnh vực nhận diện đối tượng.

Không chỉ dừng lại ở việc triển khai mô hình, tôi còn tập trung vào việc tối ưu hóa và cải thiện độ chính xác của mô hình. Việc điều chỉnh siêu tham số và tối ưu hóa thông số đầu vào đã đóng một vai trò quan trọng trong việc nâng cao khả năng dự đoán và giảm thiểu sai số.

Tôi cũng đã mở rộng phạm vi nghiên cứu của mình từ việc chỉ nhận diện đối tượng sang việc đối mặt với thách thức của việc nhận diện trong video. Việc này đã yêu cầu sự linh hoạt trong cách tiếp cận vấn đề và tìm kiếm giải pháp đáp ứng.

Tổng cộng, nghiên cứu này không chỉ mang lại sự hiểu biết sâu sắc về lĩnh vực nhận diện đối tượng, mà còn cung cấp những kinh nghiệm quý báu về quá trình triển khai và tối ưu hóa mô hình. Điều này mở ra nhiều cơ hội để ứng dụng các kiến thức này trong nhiều lĩnh vực, từ an ninh đến công nghiệp giải trí.

Cuối cùng, qua quá trình nghiên cứu này, tôi tin rằng sự hiểu biết và kỹ năng thu được sẽ tạo ra những đóng góp tích cực và đáng giá trong cả môi trường học thuật và thực tế.

TÀI LIỆU THAM KHẢO

[1] Object Detection (nhận diện vật thể) chỉ với 10 dòng code sử dụng ImageAI. Link: https://viblo.asia/p/computer-vision-object-detection-nhan-dien-vat-the-chi-voi-10-dong-code-su-dung-imageai-naQZRbdjZvx (Ngày truy cập 01/01/2024)

[2] Official English Documentation for ImageAI! Link: https://imageai.readthedocs.io/en/latest/ (Ngày truy cập 01/01/2024)

[3] What Is Object Detection? Link: https://www.mathworks.com/discovery/object-detection.html#:~:text=Object%20detection%20is%20a%20computer,learning%20to%20produce%20meaningful%20results. (Ngày truy cập 01/01/2024)

[4] What is Object Detection? Link: https://viso.ai/deep-learning/object-detection/