# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

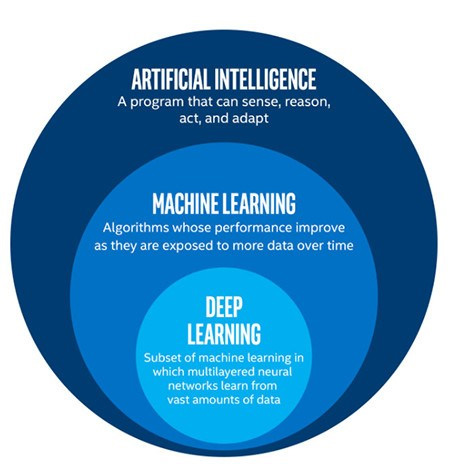
Ngày nay, Deep Learning được ứng dụng rất rộng rãi ở hầu hết các ngành bao gồm: công nghiệp, y học, giáo dục, nông nghiệp,... với các bài toán điển hình như nhận diện hình ảnh, giọng nói, xử lý ngôn ngữ tự nhiên. Trong đó, Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến được sử dụng phổ biến.

Chương này sẽ tìm hiểu tổng quan về mạng học sâu (Deep Learning), mô hình mạng nơ-ron trong nhận dạng đối tượng và các mạng phổ biến nhất hiện nay trong bài toán nhận dạng đối tượng.

## HỌC SÂU: DEEP LEARNING

Học sâu (viết tắt là DL – Deep Learning) là một nhánh của học máy (Machine Learning), được xem là một thuật toán dựa trên ý tưởng đến từ bộ não thông qua việc tiếp thu nhiều tầng biểu đạt hay trừu tượng để có thể làm rõ nghĩa của các loại dữ liệu. Những thuật toán của Deep Learning vô cùng đa dạng, mỗi thuật toán được ứng dụng tùy vào từng bài toán cụ thể.

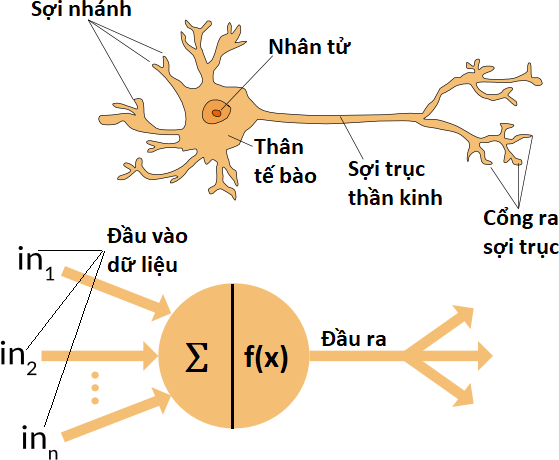
Deep Learning là thứ xuất hiện sau trí tuệ nhân tạo (Artificial Intelligence – AI) và Machine Learning và đang là thứ thúc đẩy sự bùng nổ AI ngày nay – là vòng tròn nhỏ nhất.



Hình 1.. Mối liên hệ giữa AI, Machine Learning và Deep Learning

Kỹ thuật học sâu được ứng dụng vào rất nhiều ngành công nghiệp bao gồm xe hơi tự hành, phát hiện các bệnh lý trong ứng dụng y học, công nghệ điện toán đám mây, nhận diện giọng nói và các ứng dụng trí tuệ. Cũng như các thuật toán học máy, học sâu cũng dựa trên các thuật toán phức tạp và kỹ thuật xác suất thống kê. Một số loại mạng nơ-ron nhân tạo phổ biến như mạng tích chập Convolutional Neural Networks (CNN) và mạng nơ-ron hồi qui Recurrent Neutral Netwrok (RNN) là những ứng dụng của kỹ thuật học sâu hiện đại – mạng lưới này mô phỏng tương tự như mạng lưới thần kinh của con người. Mỗi một loại mạng nơ-ron được sử dụng vào ứng dụng phức tạp khác nhau như: bài toán phân loại, hay bài toán dự đoán. Ví dụ, nhận diện hình ảnh và nhận diện khuôn mặt người ta sẽ sử dụng mạng tích chập CNN, trong khi xử lý ngôn ngữ tự nhiên – Natural Language Processing (NLP) thì người ta lại sử dụng mạng hồi qui RNN.

### Mạng nơ-ron nhân tạo

Mạng lưới thần kinh nhân tạo là các mô hình tính toán tuy được ra đời cách đây hơn 60 năm về trước nhưng sự khởi đầu mạnh mẽ của nó mới chỉ diễn ra vào những năm của đầu thế kỷ 20. Và mạng nơ-ron nhân tạo đã trở thành một trong những trụ cột căn bản nhất của ngành điện toán hiện đại chỉ với một số chứng minh của mô hình đơn giản ban đầu. Nơ-ron học máy tuy là một hàm nhưng nó lại có cấu tạo dựa trên mô hình cấu trúc như của nơ-ron sinh học.

Tương tự như trong mô hình sinh học, các xung thần kinh được truyền qua sợi trục thần kinh với một tỷ lệ nào đó, thì ở mô hình học máy khi mô phỏng lại cũng có một hàm phi tuyến được sử dụng để điều chỉnh tỉ lệ đầu ra sao cho càng giống dữ liệu đầu vào. Hàm đó được gọi là hàm kích hoạt (activation function), có rất nhiều loại hàm kích hoạt có thể kể đến như Sigmoid, ReLU, Tanh. Trong đó hàm Leaky ReLU là một trong các hàm kích hoạt được sử dụng phổ biến trong nhiều nghiên cứu những năm gần đây do hàm này khắc phục được hạn chế được “vùng chết” của hàm ReLU.

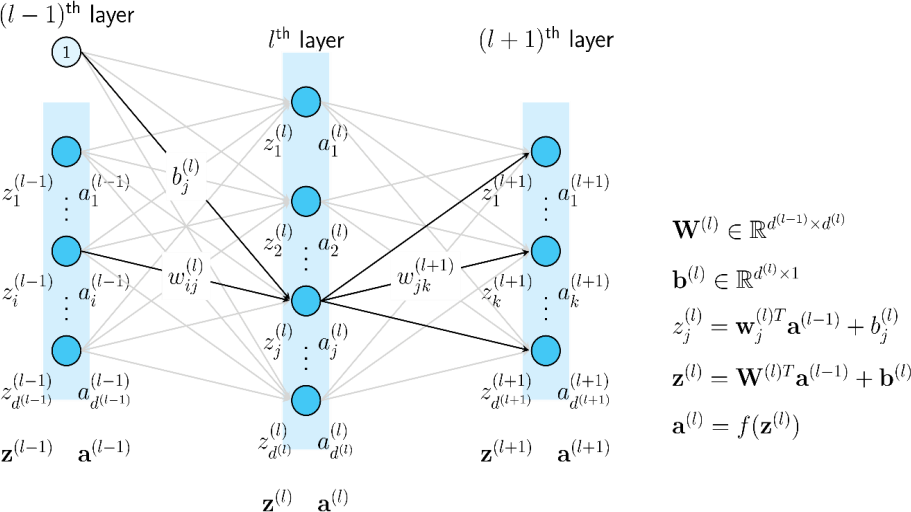
𝐿𝑒𝑎𝑘𝑦𝑅𝑒𝐿𝑈(𝑥) = max(0.01𝑥,𝑥) (1)

Đầu ra của một “đơn vị” nơ-ron được tính qua công thức (2) hàm tuyến tính

𝑦 = 𝜎(𝑤𝑖𝑥𝑖 + 𝑏) (2)

Trong đó là dữ liệu đầu vào thứ , là trọng số của nơ-ron, *b* là nhiễu bias, 𝜎 là hàm kích hoạt.

Hình 1.3. Sơ đồ mạng nơ-ron lan truyền thẳng

Mạng truyền thẳng nhiều lớp là mô hình mạng lưới nơ-ron nhân tạo được tạo bởi một tập hợp của các nơ-ron. Mỗi nơ-ron sẽ chứa một hàm kích hoạt, tùy thuộc vào dữ liệu đầu vào mà mạng lưới sẽ tính toán ra một đầu ra bằng cách áp dụng hàm kết hợp (3). Kiến trúc của mạng nơ-ron bao gồm 3 loại lớp nơ-ron: lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer). Lớp đầu vào là nơi dữ liệu được đưa vào hệ thống, phần lớp ẩn là nơi dữ liệu được tính toán và được kết nối với nhau thông qua các lớp ẩn. Mỗi mạng nơ-ron có thể chứa nhiều lớp ẩn. Gọi lớp 𝑙 = 0 là lớp đầu vào, và lớp đầu ra là 𝑙 = 𝐿 − 1, gọi là trọng số kết nối từ node thứ i của lớp 𝑙 − 1 đến node thứ j của lớp 𝑙. Tương tự ta có hệ số bias của node thứ j trong lớp 𝑙 là , và ta có ma trận hệ số giữa hai lớp 𝑙 − 1 và lớp 𝑙, ma trận này có kích thước . Ta tính được tổng hàm tuyến tính của lớp *l* qua công thức (3) sau:

𝑗

𝑗𝑘

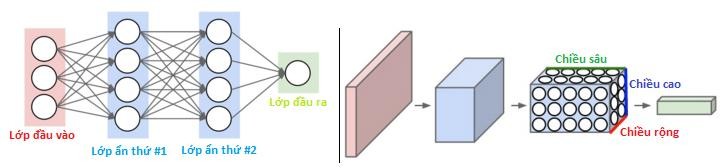
(3)

Rồi áp dụng hàm kích hoạt: (4)

Cuối cùng ta sẽ có hàm kích hoạt cho lớp l: (5)

### Mạng nơ-ron tích chập (Convolutional Neural Network)

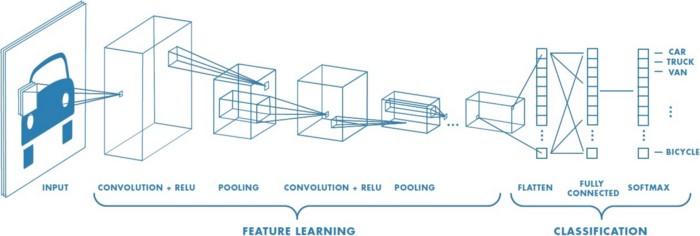
Mạng lưới thần kinh tích chập được tạo ra để giải quyết vấn đề về số lượng tham số rất lớn giữa các nốt của các lớp ẩn trong mạng lưới thần kinh kết nối đầy đủ. Trong mạng lưới thần kinh chuyển đổi, các lớp ẩn được thay đổi bằng các lớp tích chập, tại các lớp ẩn sẽ sử dụng ô tích chập để thực hiện phép tính tích chập lên toàn bộ dữ liệu bức ảnh đầu vào, hay nói cách khác là các pixel chia sẻ hệ số với nhau.



Hình 1.4. Cấu trúc của FCN và CNN

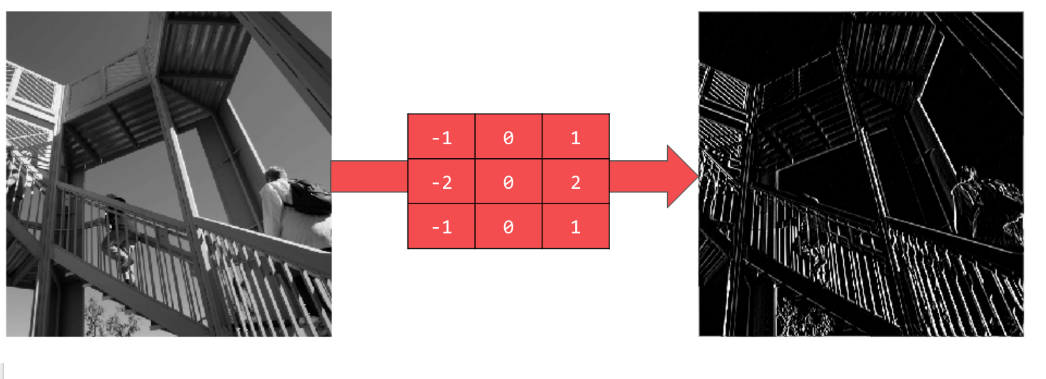
Nếu như trong mạng kết nối đầy đủ ta có lớp đầu vào, các lớp ẩn và lớp đầu ra. Ở trong mạng tích chập các nơ-ron được sắp xếp thành 3 chiều: chiều rộng, chiều cao, độ sâu. Đây là một thiết kế rất phù hợp cho bài toán phân loại dữ liệu đầu vào là dữ liệu ảnh có kích thước chiều cao x chiều rộng x 3 kênh màu (Đỏ-Lục-Lam). Các nơ-ron trong mạng tích chập không kết nối đầy đủ từng nốt mà chỉ kết nối một vùng nhỏ của lớp tích chập trước đó, liên kết này được gọi là vùng lĩnh hội.

Cấu trúc của mạng tích chập bao gồm đầu vào → lớp tích chập → hàm kích hoạt ReLU → Lớp đầu ra, được mô tả trong hình ví dụ dưới đây:



Hình 1.5. Ví dụ về bài toán phân loại ảnh sử dụng CNN

Ô tích chập có kích thước n x n sẽ được áp dụng lên toàn bộ dữ liệu ảnh, nó sẽ lần lượt dịch cửa sổ bộ lọc qua từng pixel lần lượt từ trái qua phải, từ trên xuống dưới với bước nhảy (stride) thường là 1, do kết quả của tích chập sẽ cho ra kích thước nhỏ hơn nên nhờ việc thêm padding = 1 sẽ thêm vào các cột và hàng xung quanh dữ liệu ảnh các giá trị bằng không để giúp đầu ra bảo toàn được kích cỡ. Sau mỗi lớp tích chập, ta sẽ thu được bản đồ đặc trưng trừu tượng (feature map), bản đồ đặc trưng trừu tượng sẽ cho ta biết thông tin đặc trưng của ảnh như viền của vật thể nếu như ta cho tích chập ảnh với cửa sổ bộ lọc viền, khi trượt qua ảnh đầu vào những pixel trung tâm sẽ có giá trị lớn nhất, và những pixel xung quanh sẽ tối đi khi đó nó sẽ để lại những pixel trung tâm, đó chính là viền của vật thể như trong ảnh 1.7 dưới đây.



Hình 1.6. Ví dụ tích chập ảnh với bộ lọc

Sau khi có được bản đồ đặc trưng trừu tượng, áp dụng hàm kích hoạt (như Leaky ReLU, ReLU, Sigmoid) để có được các thông tin trừu tượng rõ ràng hơn cho các lớp tiếp theo. Tiếp theo, để giảm khối lượng tính toán và giúp cải thiện thời gian học, ta sử dụng lớp tổng hợp Max Pooling, ngoài ra nó còn giúp tránh vấn đề thừa ăn khớp (overfitting). Và cuối cùng lớp đầu ra sẽ sử dụng lớp kết nối đầy đủ, để biến đổi các đặc trưng trừu tượng 3 chiều thành một véc-tơ một chiều để đưa ra kết quả xác xuất tương ứng với mỗi đặc trưng của từng đối tượng cần phân loại.

## NHẬN DẠNG ĐỐI TƯỢNG: OBJECT DETECTION

Phát hiện đối tượng là làm nhiệm vụ xác định một hoặc nhiều đối tượng hiện diện trong cùng một hình ảnh. Việc phát hiện đối tượng bao gồm hai phần cụ thể là phân loại và xác định vị trí. Đây là một thách thức trong bài toán này vì phải phân loại đối tượng mà còn phải xác định chính xác vị trí tất cả các đối tượng có trong hình ảnh cùng với việc phát hiện sự hiện diện của chúng. Phát hiện đối tượng là một trong số ứng dụng quan trọng, bởi nhiều nhiệm vụ cần sự giám sát của con người có thể tự động hóa bằng cách phát hiện các đối tượng trong hình ảnh. Phát hiện đối tượng về cơ bản là phân loại và định vị các đối tượng, gắn nhãn cho chúng bằng các hộp hình chữ nhật để thể hiện điểm tin cậy của dự đoán. Mạng nơ-ron tích chập (CNN) kết hợp với sliding windows cửa sổ trượt để phân loại từng hình ảnh này có đối tượng hoặc không đối tượng, sau đó kết hợp các kết quả lại với nhau để thực hiện dự đoán.

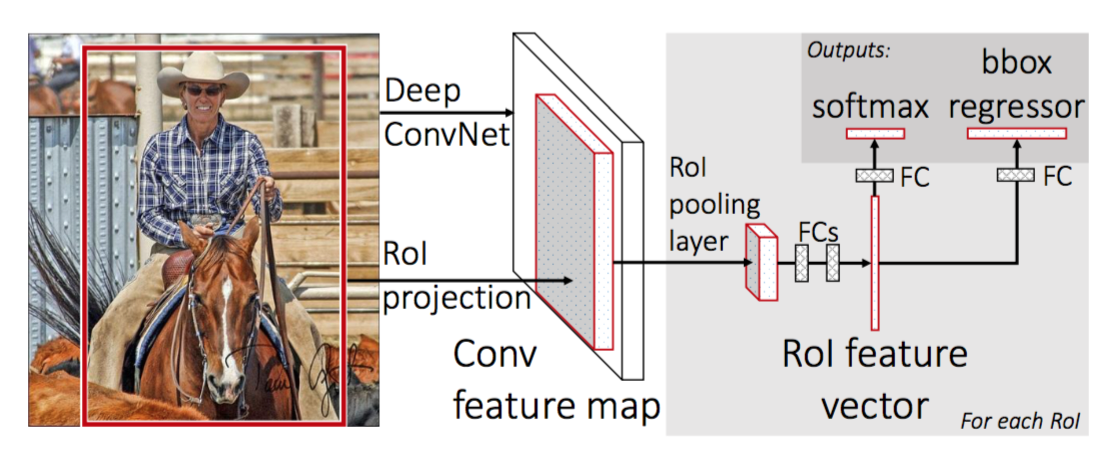
Trong những năm gần đây, rất nhiều mạng hiện đại phát hiện đối tượng đã được đề xuất bởi cộng đồng học sâu DL, như Faster R-CNN , YOLO , R-FCN và SSD . Mục tiêu chính của các thiết kế này là cải thiện độ chính xác phát hiện theo mAP và độ phức tạp trong tính toán của các mô hình giúp cải thiện thời gian xử lý, để có thể đạt được hiệu suất thời gian thực cho các nền tảng nhúng và di động. Các mô hình mạng phát hiện vật thể này có thể được chia thành hai loại dựa trên kiến trúc của chúng: 1) phương pháp tiếp cận một bước và 2) phương pháp hai bước (dựa trên khu vực – region based). Cách tiếp cận một bước có tốc độ chạy nhanh hơn và cho thấy hiệu quả bộ nhớ cao hơn trong khi cách tiếp cận hai bước tuy đạt được độ chính xác mAP tốt hơn nhưng lại tốn bộ nhớ hơn để chạy.

### Mạng phát hiện đối tượng theo khu vực

Mạng phát hiện đối tượng theo khu vực được chia làm hai bước. Bước thứ nhất là tạo các phân vùng trong ảnh mà khả năng đối tượng cần phát hiện sẽ nằm trong đó. Bước thứ hai sẽ thực hiện việc phát hiện đối tượng và phân loại đối tượng đó là vật gì dựa theo đầu vào phân vùng được tạo ở bước thứ nhất. Có thể kể đến một số mô hình nổi tiếng như R-CNN, Fast R-CNN, FPN và R-FCN.

#### Mạng Fast R-CNN

Để tăng tốc cho R-CNN, Girshick đã cải thiện quá trình huấn luyện bằng các hợp nhất 3 mô hình độc lập thành một và được đào tạo chung và tăng khả năng tính toán. Mô hình này gọi là Fast-RCNN. Thay vì trích xuất đặc trưng bằng CNN cho mỗi vùng ảnh, mô hình tổng hợp chúng thành một CNN forward trên toàn bộ hình ảnh. Chính vì điều này giúp cho mạng Fast R-CNN giảm được cả thời gian chạy và khối lượng tính toán. Ngoài ra Fast R-CNN còn đưa ra phương pháp cho phép các đầu ra phân loại và hồi quy được huấn luyện đồng thời như trong hình 1.8.



Hình 1.7. Cấu trúc mạng Fast R-CNN

Mạng Fast-RCNN tối ưu tốc độ và tài nguyên hơn R-CNN. Hàm mất mát dễ huấn luyện hơn và tránh được vấn đề “Nổ” Gradient (Exploding Gradient). Nổ gradient là hiện tượng khi sai số gradient dần bị tích lũy và ngày càng tăng lên giá trị rất lớn khiến cho mạng học sâu trở nên bất ổn định. Không những thế giá trị của trọng số trở nên lớn hơn bất thường còn làm gia tăng các tham số tính toán, khiến cho tài nguyên bộ nhớ bị tràn.

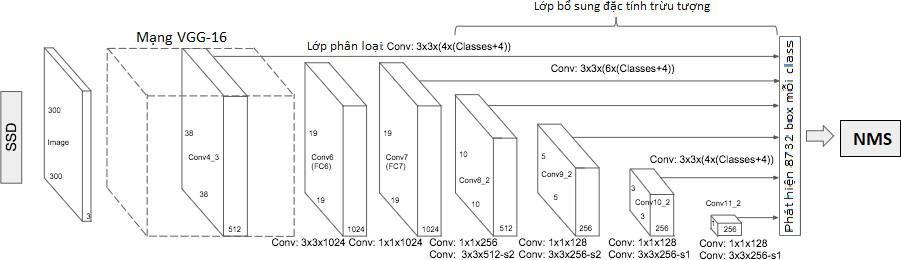
### Mạng phát nhận dạng một bước

Mạng phát hiện đối tượng một bước có kiến trúc giản đơn và thời gian xử lý của mạng nhanh hơn, với độ chính xác có thể chấp nhận được khi so sánh với các loại mạng phát hiện đối tượng hai bước dựa theo vùng. Các mạng phát hiện đối tượng một bước gần giống với mạng phát hiện đối tượng hai bước RPN, nhưng thay vì dự đoán vật hay không phải là vật, mạng phát hiện đối tượng một bước thì dự đoán phân loại vật thuộc lớp nào và cũng xác định vị trí tọa độ của vật trong ảnh. Một số mạng phát hiện đối tượng một bước hiện đại như **SSD** và **YOLO.**

#### Single Shot Multibox Detection (SSD)

SSD [4] là mạng nhận dạng đối tượng có đầu ra là một tập hợp nhận dạng cùng kích thước mà kết quả có giá trị lớn nhất sẽ được lựa chọn, và giá trị đó được coi là giá trị tượng trưng cho độ tin tưởng đối tượng được nhận diện đó thuộc lớp vật nào. Cấu trúc của mạng SSD ở những lớp đầu sử dụng cấu trúc từ các mạng bài toán phân loại như CNN, VGG-16, ResNet để làm mạng cơ sở, hay còn được biết đến như phần xương sống cho mạng học sâu nhận dạng đối tượng. Ở lớp cuối của phần mạng cơ sở đó, lớp kết nối đầy đủ sẽ được bỏ đi, cuối cùng kết quả ở phần xương sống sẽ là đặc tính trừu tượng của dữ liệu ảnh đầu vào.

Trong tài liệu [4] tác giả đã sử dụng mạng VGG-16 với một số chỉnh sửa để cấu trúc hoạt động nhanh hơn mà không làm giảm đáng kể động chính xác sau đó. Sau đó họ thêm một tập hợp các lớp mạng tích chập ở cuối mạng cơ sở để giảm dần kích thước ảnh, đồng thời cho phép trích xuất thêm đặc điểm đặc trưng ở các tỷ lệ kích thước khác như như trong hình 1.9. Mỗi bản đồ trích xuất đặc trưng (feature map) có kích thước m \* n \* p.



Hình 1.8. Cấu trúc mạng SSD sử dụng mạng cơ sở VGG-16

Trong đó m \* n là số lượng ô đặc trưng trong bản đồ và p là độ sâu của véc-tơ đặc trưng cho mỗi ô. Đặc tính trừu tượng của mỗi lớp được sử dụng vào hai việc: thứ nhất là đưa vào lớp tiếp theo, hai là được trực tiếp sử dụng để dự đoán vật thể. Đối với mỗi ô, người ta có thể xuất số lượng đầu ra dự đoán cố định (ví dụ 4 đầu ra) liên quan đến vị trí của ô trong ảnh, cho phép các hộp giới hạn có kích thước và vị trí khác nhau. Ma trận điểm ảnh ở các lớp ban đầu có kích thước m\*n rất lớn sẽ được chia ra thành nhiều phần nhỏ để dự đoán vật thể nhỏ hơn. Những cải tiến chính của SSD đã vượt qua YOLO là nó có khả năng tích chập hoàn toàn, có phát hiện ở nhiều kích thước tỷ lệ khác nhau. Với mạng CNN, các lớp kết nối đầy đủ là cực kỳ tốn kém vì chúng đòi hỏi một số lượng lớn các kết nối như vậy làm chậm đáng kể việc tính toán.

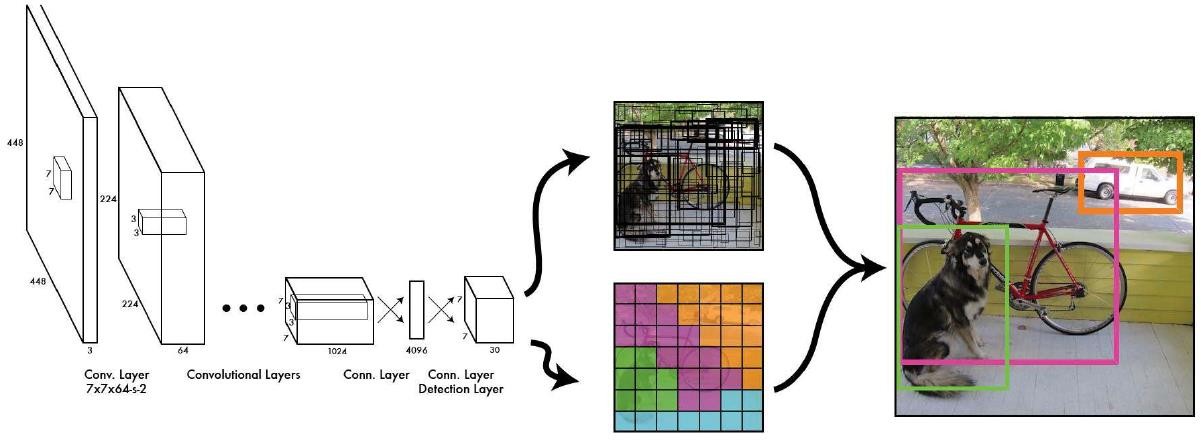
#### YOLO – You Only Look Once

YOLO là một phương pháp của Đại học Washington, giới thiệu một mạng lưới đơn sắc dự đoán các hộp giới hạn và phân loại của chúng trực tiếp từ một hình ảnh có kích thước đầy đủ. Mục tiêu của mạng là chỉ nhìn vào mọi phần của hình ảnh một lần, do đó có tên You Only Look Once – tạm dịch là **“Bạn chỉ nhìn một lần”**. Để nhận ra điều này, nhóm tác giả đã bắt đầu bằng việc chia hình ảnh thành một lưới 7x7 trong đó mỗi ô có trách nhiệm dự đoán. Đầu tiên, họ dự đoán một nhóm nhỏ các hộp giới hạn mà mỗi hộp có ô là trung tâm của chúng. Số lượng dự đoán cho mỗi ô là một hằng số nhỏ, trong bài báo được đặt giá trị bằng 2. Thứ hai, là nó sẽ dự đoán các giá trị độ tin cậy cho mỗi hộp giới hạn. Điều này phản ánh mức độ chắc chắn của mạng YOLO là hộp giới hạn đó bao phủ một số đối tượng.



Hình 1.9. Cách thức nhận dạng của YOLO

Nếu ô không chứa bất kỳ đối tượng nào, nó vẫn sẽ đề xuất cùng một bộ các hộp giới hạn, nhưng giá trị độ tin cậy sẽ thấp hơn so với các hộp giới hạn ở ô có chứa đối tượng. Bằng cách kết hợp các kết quả từ mỗi ô 7x7, chúng ta sẽ có được thứ gì đó trông rất giống với dự đoán của khung giới hạn được mô tả trong các phương thức R-CNN, nhưng ít trùng lặp hộp giới hạn với nhau hơn. So với R-CNN có khoảng 2000 hộp giới hạn có thể phân loại riêng lẻ, YOLO hiện có 7x7x2 = 98 (2 hộp dự đoán cho mỗi ô trong lưới) với các giá trị độ tin cậy tương ứng. Tiếp theo, mạng YOLO sẽ cố gắng phân loại từng hộp giới hạn (anchor box). Điều này không được thực hiện bằng cách chạy riêng cho từng hộp, mà chạy tính trên cho từng ô. Bởi vì mọi hộp giới hạn được đặt ở giữa một trong các ô, chúng ta có thể giả định rằng đối tượng sẽ nằm trong hộp giới hạn giống như đối tượng được xác định cho ô. Đầu ra cuối cùng sau đó là tất cả các hộp giới hạn với các giá trị độ tin cậy và xác suất cho từng lớp tương ứng. Để tìm các hộp chính xác và các lớp của chúng, YOLO đặt ngưỡng cho các giá trị độ tin cậy để loại bỏ tất cả các hộp xác suất hay độ tin cậy thấp hơn. Quá trình YOLO được hiển thị trong hình 1.10.



Hình 1.10. Quá trình nhận dạng của YOLO

## Theo dõi đối tượng: Object Tracking

Object Tracking là bài toán theo dõi một hoặc nhiều đối tượng chuyển động theo thời gian trong một video. Hiểu một cách đơn giản nhất, nó là bài toán ở mức độ cao hơn so với object detection, khi đối tượng được xử lí không đơn giản là một hình ảnh mà là một chuỗi các hình ảnh hay video.

Object tracking và object detection có một mối quan hệ rất chặt chẽ. Object detection là việc mô hình xác định một hoặc nhiều đối tượng trong một khung hình nhất định còn object traking là theo dõi đối tượng trong toàn bộ video. Để theo dõi được một đối tượng trước hết mô hình cần phải phát hiện được đối tượng đó trong từng frame sau đó sẽ cho qua một thuật toán về tracking nên việc theo dõi đối tượng có chính xác hay không thì cần phải xem việc phát hiện đối tượng tốt đến đâu. Object detection sẽ phát hiện sự xuất hiện của những đối tượng trong hình ảnh tuy nhiên để xem trong video đó khi các đối tượng chuyển động thì cùng 1 đối tượng mô hình sẽ hiểu đó là cùng 1 đối tượng chứ không phải là 2 đối tượng để giữ nguyên màu sắc trên đối tượng thì đó chính là bài toán tracking.

### Ứng dụng của Object Tracking

Bài toán theo dõi đối tượng được ứng dụng vào rất nhiều lĩnh vực trong cuộc sống cụ thể như:

* Autonomous driving: dùng để truy vết cars và các vật cản khác ở trên đường nhằm lên kế hoạch cho một tuyến đường tốt nhất tránh gây tắc nghẽn.
* Traffic control: phát hiện những phương tiện đang di chuyển bao gồm những thông tin như loại xe, thương hiệu, biển số, tốc độ,…
* Digital forensic: theo dõi tội phạm trực tuyến (đơn và đa mục tiêu)
* Visual surveillance: dùng để xử lý các vấn đề về trộm cắp xe, quản lý đám động,…
* Monitoring social distancing: sử dụng trong đại dịch COVID-19
* Phát hiện các sản phẩm lỗi trong dây chuyền sản xuất, theo dõi vật nuôi
* Theo dõi hành vi, hoạt động, sức khoẻ của người cao tuổi

### Một số kỹ thuật Tracking Object phổ biến

#### ****Centroid based Object Tracking****

Centroid-based object tracking tận dụng Euclidean distance (khoảng cách euclid) giữa các Tâm của đối tượng được phát hiện giữa 2 frames liên tiếp trong một video.

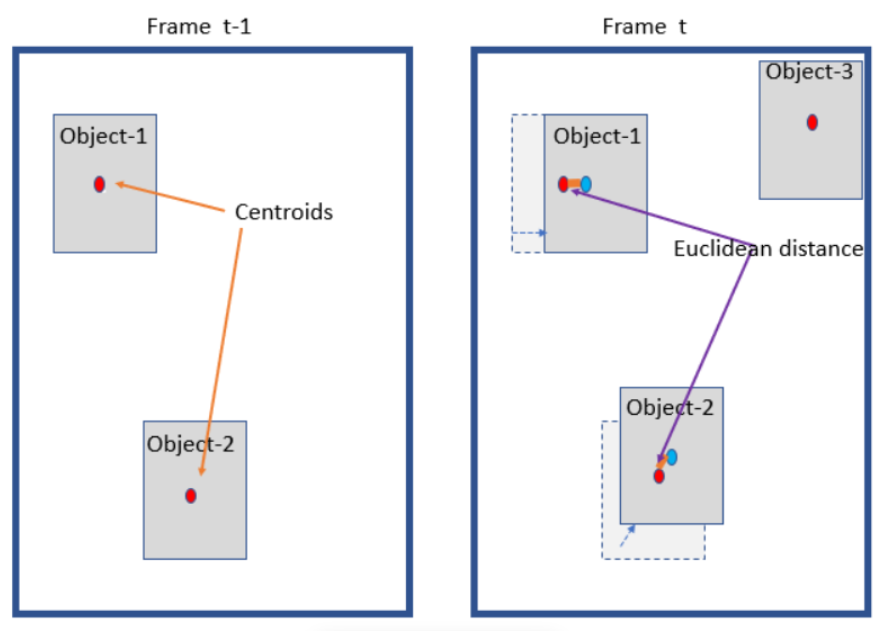
**Bước 1:**Tại thời gian t-1, Các Objects được phát hiện đóng khung trong các bounding boxes.

**Bước 2**: Tính toán các centroids đối với các đối tượng được phát hiện tại thời gian t-1.

**Bước 3**: Tại thời gian t, Các Objects được phát hiện đóng khung trong các bounding box . Gán các unique ID tới các objects.

**Bước 4**: Tính toán các centroids của các objects được phát hiện tại thời gian t.

**Bước 5**: Tính toán Euclidean distance giữa các centroids của các objects được phát hiện tại thời gian t-1 và t.



Hình 1.11. Kỹ thuật **Centroid based Object Tracking**

**Bước 6:** Nếu khoảng cách giữa centroids tại thời gian t-1 và t nhỏ hơn threshold 🡪 nó là cùng của một đối tượng chuyển động. Do đó sử dụng lại object ID hiện tại và cập nhật lại bounding box mới cho đối tượng.

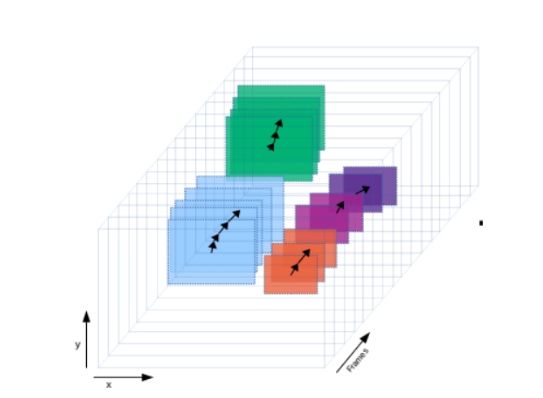
**Bước 7**: Nếu khoảng cách lớn hơn threshlod, thêm một object ID mới.

**Bước 8**: Khi các objects được phát hiện từ các frame phía trước không thể match với bất kỳ các objects đang tồn tại nào–> xóa object id ra khỏi quá trình truy vết

* + - 1. **IOU Object Tracker**

Intersection-over-Union là một kỹ thuật khác trong tracking liên kết các detections của các frames liên tiếp sử dụng sự chồng lấn không gian của nó để có thể truy vết.

Mỗi track sẽ lấy object được phát hiện với IOU cao nhất với một object đã được định danh gán ID trước đó. Sự liên kết này được giải quyết bằng một bài toán linear assignment – thuật toán Hungarian để tối đa hóa tổng của tất cả các IOUs trong từng frame.



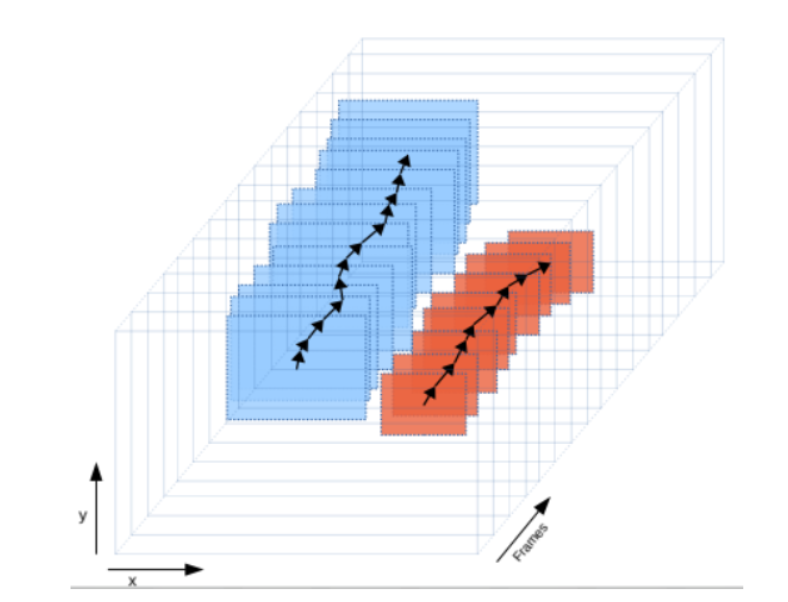
Hình 1.12. Kỹ thuật IOU Object Tracker

* + - 1. **Visual IOU Object Tracker**

Đây là một sự cải tiến đối với IOU-based Object Tracker bằng việc kết hợp với visual tracking. Visual IOU Object Tracker được thực hiện trong 2 hướng: visual forward và backward tracking nhằm giúp hợp nhất các tracks bị gián đoạn không liên tục.

* Khi không có object detection nào thỏa mãn IOU threshold đối với một object track, visual tracker sẽ được khởi tạo tại vị trí được biết đến ở frame trước đó của object và được sử dụng để truy vết object trong một vài frames nhất định.
* Nếu một object mới thỏa mãn IOU threshold được phát hiện chỉ trong một vài frames chỉ định kế tiếp, visual tracking sẽ dừng lại và IOT tracker sẽ tiếp tục. Nếu không có thì việc truy vết sẽ kết thúc.

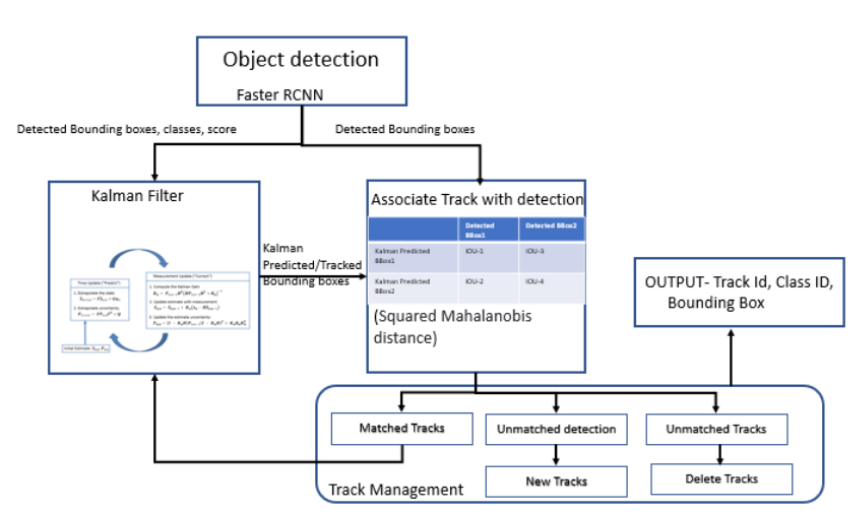
Hướng tiếp cận trên giúp giải quyết phần nào tình trạng mất dấu đối tượng.

****

Hình 1.13. Kỹ thuật Visual IOU Object Tracker

* + - 1. **SORT - Simple Online Realtime Object Tracking**

Thuật toán SORT dựa trên một giả định rằng chất lượng của tracking sẽ phụ thuộc vào performance của thuật toán object detection sử dụng.



Hình 1.14. Kỹ thuật SORT

Đầu tiện, SORT sử dụng mạng Faster Region-CNN(FrRCNN) dùng để detect các objects.

* Ở stage thứ nhất, FrRCNN sẽ trích xuất đặc trưng và đề xuất những vùng có khả năng chứa các objects cho stage thứ hai.
* Ở stage thứ hai, FrRCNN sẽ phân loại các objects trong ở các vùng được đề xuất.

FrRCNN chia sẻ các parameters giữa 2 stages, tạo ra một model hiệu quả trong việc detection.

Những track mới được khởi tạo khi các detections không thể liên kết với các tracks đã tồn tại trước đó. Khi một object detection với overlap nhỏ hơn IOU, báo hiệu rằng có sự tồn tại của một untracked object.

Những tracks mà vượt quá maximun age đã được định ra từ trước thì xem như đã ra khỏi ngữ cảnh và track sẽ bị xóa. Các tracks được chấm dứt nếu như chúng không được detected trong T(lost) frames, và T(Lost) frames thường được thiết lập bằng 1.

**Thuật toán SORT giúp giảm tình trạng mất dấu khi các mục tiêu bị che khuất, ID switches để hoạt động tốt khi chuyển động của các đối tượng tương đối chậm. SORT có thể gặp khó khăn trong trường hợp ngữ cảnh động đúc hoặc các đối tượng chuyển động nhanh.**

* + - 1. **DeepSort**

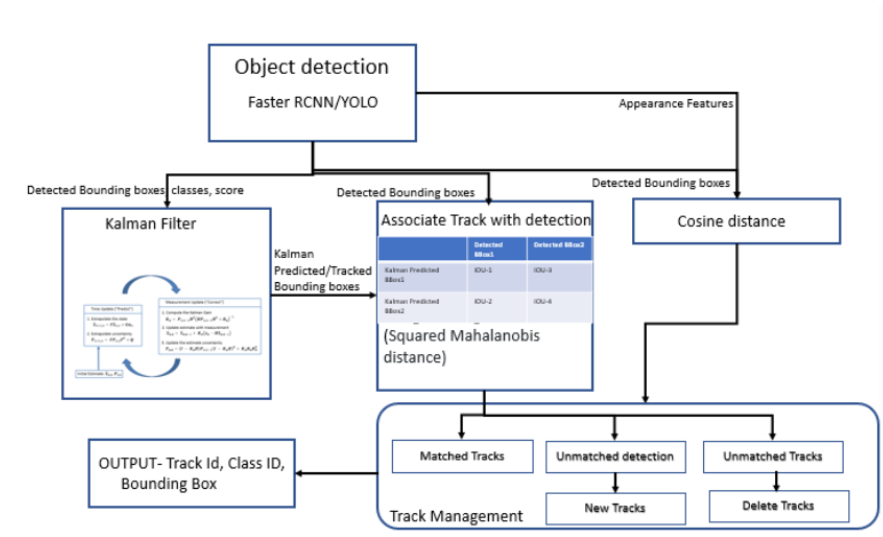
**Deep SORT và SORT là những phương pháp tracking theo cascade-style (kiểu thác nước).** Đầu tiên chúng thực hiện dự đoán những bounding boxes đối với những đối tượng được phát hiện và sau đó trích xuất các features từ chúng để ước tính các re-ID features tương ứng.

**Deep SORT cho phép thực hiện truy vết đối với những đối tượng bị chồng lấn trong khoảng thời gian dài hơn, đơn giản để implement và chạy trong real-time.**

Deep SORT cũng tuân theo một phương pháp tracking sử dụng**recursive Kalman filering và liên kết dữ liệu frame-by-frame sử dụng thuật toán Hungarian.**

Kalman filter dự đoán quỹ đạo có thể có của đối tượng trong tương lai dựa trên vị trí hiện tại. Nó lấy tọa độ bounding box tại vị trí quan sát hiện tại của đối tượng để cung cấp một ước tính sơ lược về vị trí di chuyển đến kế tiếp của đối tượng.

Để mỗi bounding boxes được đựa vào danh sách tracked của Kalman Filter, chúng ta phải liên kết những detections ở frame mới nhất với những predictions mới nhất của Kalman. **Quá trình liên kết đó được giải quyết sử dụng thuât toán Hungarian.**



Hình 1.15. Kỹ thuật DEEPSORT

**Deep SORT giúp giảm tình trạng ID Switches và occlusions (che lấp), từ đó giảm tỷ lệ False Positives.**

# CHƯƠNG II: THEO DÕI PHƯƠNG TIỆN GIAO THÔNG SỬ DỤNG YOLO VÀ SORT

## SORT - Simple Online Realtime Object Tracking

### Giải thuật Hungary

Giải thuật Hungary được phát triển và công bố vào năm 1955, đề xuất để giải bài toán phân công công việc (assignment problem).

Tương tự, với bài toán phân công công việc, object tracking cần giải quyết như sau Có n detection (i = 1, 2, …, n) và n track predicted (j = 1, 2, … n). Để liên kết một detection i với một track j giả sử dựa vào 1 độ đo D - D là khoảng cách giữa i và j trong không gian vector . Bài toán đặt ra là cần liên kết mỗi detection với mỗi track tương ứng sao cho sai số của việc liên kết là nhỏ nhất.

Với: -

-

-

Các thỏa mãn các điều kiện trên gọi là một phương án phân công, một phương án đạt cực tiểu của z được gọi là một phương án tối ưu hay lời giải của bài toán.

#### Các bước giải thuật toán Hungary

* **Bước 1:** (Bước chuẩn bị). Trừ các phần tử trên mỗi hàng của C cho phần tử nhỏ nhất trên hàng đó, tiếp theo trừ các phần tử trên mỗi cột cho phần tử nhỏ nhất trên cột đó. Kết quả ta nhận được ma trận C' có tính chất: trên mỗi hàng, cột có ít nhất một phần tử 0 và bài toán giao việc với ma trận C' có cùng lời giải như bài toán với ma trận C.
* **Bước 2**: Vẽ một số tối thiểu các đường thẳng trên dòng và cột để đảm bảo mọi phần tử 0 đều được đi qua.
* **Bước 3:** Nếu có n đường thẳng được vẽ, kết thúc thuật toán và tiến hành phân công công việc. Nếu số đường thẳng được vẽ nhỏ hơn n, vẫn chưa tìm được phương án phân công tối ưu, tiến hành bước tiếp theo.
* **Bước 4:** Mỗi hàng (hoặc cột) có đường thẳng vẽ qua, ta gọi các hàng (cột) đó là các hàng (cột) thiết yếu. Các hàng (cột) còn lại là các hàng (cột) không thiết yếu. Tìm phần tử nhỏ nhất không nằm trong các hàng (cột) thiết yếu, tiến hành trừ mỗi hàng không thiết yếu cho phần từ nhỏ nhất ấy và cộng giá trị nhỏ nhất ấy cho cột thiết yếu. Ta được ma trận C’’ có cùng lời giải với ma trận C’. Sau đó quay lại Bước 2
  + - 1. **Ví dụ**

Bài toán: Phân công cho 5 nhân viên đi địa điểm nào với thời gian là thấp nhất

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nhân Viên  / Địa điểm | I | II | III | IV | V |
| Nhân | 54 | 40 | 25 | 42 | 52 |
| Lễ | 50 | 59 | 18 | 40 | 37 |
| Nghĩa | 61 | 45 | 32 | 40 | 45 |
| Trí | 33 | 35 | 29 | 45 | 15 |
| Tín | 35 | 30 | 20 | 22 | 40 |

Bước 1: Trừ các phần tử trên mỗi hàng của C cho phần tử nhỏ nhất trên hàng đó, tiếp theo trừ các phần tử trên mỗi cột cho phần tử nhỏ nhất trên cột đó

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 29 | 15 | 0 | 17 | 27 |
| 32 | 41 | 0 | 22 | 19 |
| 29 | 13 | 0 | 8 | 13 |
| 18 | 20 | 14 | 30 | 0 |
| 15 | 10 | 0 | 2 | 20 |
| 14 | 5 | 0 | 15 | 27 |
| 17 | 31 | 0 | 20 | 19 |
| 14 | 3 | 0 | 6 | 13 |
| 3 | 10 | 14 | 28 | 0 |
| 0 | 0 | 0 | 0 | 20 |

Bước 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 14 | 5 | 0 | 15 | 27 |
| 17 | 31 | 0 | 20 | 19 |
| 14 | 3 | 0 | 6 | 13 |
| 3 | 10 | 14 | 28 | 0 |
| 0 | 0 | 0 | 0 | 20 |

Bước 3+4:

3 đường thẳng đã được vẽ < n=5

Chọn số 3 là phần tử nhỏ nhất của cột(hàng) chưa tối ưu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | 2 | 0 | 12 | 27 |
| 14 | 28 | 0 | 17 | 19 |
| 11 | 0 | 0 | 3 | 13 |
| 0 | 7 | 14 | 25 | 0 |
| 0 | 0 | 3 | 0 | 23 |

4 đường đã được vẽ < n = 5 tiếp tục thực hiện (chọn 3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 8 | 2 | 0 | 9 | 24 |
| 11 | 28 | 0 | 14 | 16 |
| 8 | 0 | 0 | 0 | 10 |
| 0 | 10 | 17 | 25 | 0 |
| 0 | 3 | 6 | 0 | 23 |

4 đường đã được vẽ < n = 5 vẫn tiếp tục thực hiện (chọn 2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 0 | 0 | 7 | 22 |
| 9 | 26 | 0 | 12 | 14 |
| 8 | 0 | 2 | 0 | 10 |
| 0 | 10 | 19 | 25 | 0 |
| 0 | 3 | 8 | 0 | 23 |

5 đường thẳng đã được vẽ = n=5 kết thúc thuật toán

Kết luận: - Nhân đi địa điểm II với thời gian là 40phut

- Lễ đi địa điểm III với thời gian là 18p

- Nghĩa đi địa điểm IV với thời gian là 40p

- Trí đi địa điểm V với thời gian là 15p

- Tín đi địa điểm I với thời gian là 35p

- Tổng thời gian thấp nhất là 148p

### Bộ lọc Kalman

Bộ lọc Kalman (Kalman Filter) là một mô hình Linear-Gaussian State Space Model, được giới thiệu lần đầu năm 1960 bởi Rudolf (Rudy) E. Kálmán, là thuật toán sử dụng chuỗi các giá trị đo lường, bị ảnh hưởng bởi nhiễu hoặc sai số, để ước đoán biến số nhằm tăng độ chính xác so với việc sử dụng duy nhất một giái trị đo lường. Bộ lọc Kalman thực hiện phương pháp truy hồi đối với chuỗi các giá trị đầu vào bị nhiễu, nhằm tối ưu hoá giái trị ước đoán trạng thái của hệ thống. Bộ lọc Kalman được ứng dụng trong rất nhiều lĩnh vực khác nhau: Xe tự lái, thực tế ảo, kinh tế lượng, tracking, điều khiển tối ưu, ...

Trong object tracking, Kalman filter được biết đến nhiều nhất với vai trò dự đoán các trạng thái của đối tượng hiện tại dựa vào các track trong quá khứ và update lại các detection sau khi đã được liên kết với các track trước đó.

Quá trình cần xử lí là 1 quá trình ngẫu nhiên với các mô hình đã được định nghĩa từ trước:

Với:

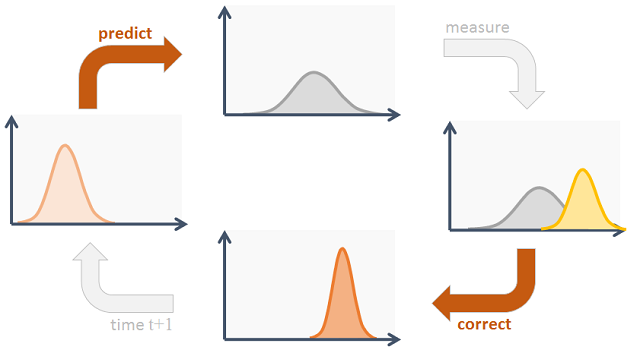
* là giá trị biến trạng thái của quá trình, thường là các giá trị ẩn, không thể quan sát được.
* là giá trị đo được, quan sát được của quá trình.
* là các mô hình định nghĩa từ trước.
* lần lượt là nhiễu của quá trình và nhiễu trong lúc đo đạc.

Áp dụng thuật toán Linear Kalman Filter (ngoài ra còn có Extended Kalman Filter, Unscented Kalman Filter, …) vào bài toán.

Linear Kalman Filter giả định các mô hình của quá trình đều là các mô hình tuyến tính. Khi đó:

Với:

* ~ N (: p(=
* là các ma trận n\*n (state transistion matrix) và m\*n (measurement function)
* ~ N(0,Q), ~ N(0,R)



Các bước xử lý tiếp theo của Kalman Filter có thể chia làm 2 phần chính (cách tiếp cận dựa trên xác xuất):

**Dự đoạn (Prediction):**

Để dự đoán các giá trị trạng thái của quá trình ngẫu nhiên, ta dự đoán các giá trị mean và covariance Σ. Theo tính chất kỳ vọng và ma trận phương sai với vector ngẫu nhiên, ta có:

**Hiệu chỉnh (Update):**

Áp dụng định lý Bayes cho Linear Gaussian System:

Tiếp tục áp dụng đồng nhất thức ma trận Woodbury:

“Đồng nhất thức ma trận Woodbury (Matrix Inversion Lemma)

Có 4 ma trận A(n\*n), U(n\*k), C(k\*k), V(k\*n).

Khi đó (A+UCV)-1 = A-1 – A-1U(C-1+VA-1U)-1VA-1”

Đặt với được gọi là hệ số Kalman tại thời điểm k, khi đó:

)

|  |  |
| --- | --- |
| Dự đoán (Prediction) | Hiệu chỉnh (Update) |
|  | * ) |