ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG
----------o0o----------

BÁO CÁO ĐỒ ÁN 2

Đề tài

HỆ THỐNG THEO DÕI CON NGƯỜI (HUMAN FOLLOWING SYSTEM)

GVHD: Nguyễn Khánh Lợi

SVTH: Dương Hoài Thương
MSSV: 2012177

Thành phố Hồ Chí Minh, tháng 5 năm 2024

LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn bộ môn Điện tử - Viễn thông trường Đại học Bách Khoa đã tạo điều kiện thuận lợi cho em thực hiện đồ án. Đặc biệt, em xin chân thành cảm ơn thầy Nguyễn Khánh Lợi đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện đồ án vừa qua.

Em cũng xin chân thành cảm ơn tất cả các thầy, các cô trong trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được đồ án này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của thầy về những vấn đề triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

Tp. Hồ Chí Minh, ngày 28 tháng 05 năm 2024 .

TÓM TẮT ĐỒ ÁN

Đồ án này trình bày bản thiết kế về hệ thống theo dõi con người, cụ thể là theo dõi chuyển động của con người và bám theo một người nhất định đã được cài đặt sẵn. Mục tiêu của hệ thống là khả năng bám được người và di chuyển hệ thống theo người chỉ định thường được ứng dụng trong thực tế để phục vụ mục đích vận chuyển hành lý hay dụng cụ y tế,… Tổng quan hệ thống sử dụng mô hình Yolov4 Tiny, DeepSORT và các mô hình nhận diện khuôn mặt, cử chỉ tay hỗ trợ cho quá trình điều khiển hệ thống và triển khai trên Raspberry Pi 4.

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH SÁCH TỪ VIẾT TẮT

ML	Machine Learning
AI	Artificial Intelligence
CNN	Convolutional Neural Network
YOLO	You Only Look Once
FPS	Frame Per Second
SOT	Single Object Tracking
MOT	Mutiple Object Tracking
SORT	Simple Online and Realtime Tracking
KF	Kalman Filter
IOU	Intersection Over Union

CHƯƠNG 1. **GIỚI THIỆU**

1.1 Đặt vấn đề

Sự phát triển của xã hội đã không ngừng đẩy nhanh sự tiến bộ của khoa học kỹ thuật. Việc ứng dụng các công nghệ về xử lí ảnh, AI hay thị giác máy ngày càng trở nên phổ biến và có nhiều đóng góp đáng kể nhằm phục vụ tốt hơn cho con người. Những ứng dụng mà chúng ta có thể thường thấy ngoài thực tế là việc ứng dụng công nghệ máy học, thị giác máy trong việc theo dõi chuyển động của con người thông qua camera và thực hiện các mục đích có ích như: đếm số lượng người, nhận diện những người đặc trưng, di chuyển hoặc làm việc theo những hành động của con người,…

Một trong những ứng dụng điển hình của việc theo dõi con người thông qua camera đó là phát triển một hệ thống có thể tự di chuyển và bám theo chuyển động của con người. Mô hình này được ứng dụng rộng rãi điển hình trong thực tế như robot giúp di chuyển hành lí cho con người tại sân bay hay robot vận chuyển dụng cụ y tế trong bệnh viện,…

Hình 1-1: Robot trí tuệ nhân tạo đẩy hành lý Care-E

Đề tài tập trung nghiên cứu và phát triển một hệ thống có thể phát hiện người và di chuyển theo chuyển động của con người đã được chỉ định sử dụng các mô hình máy học và thị giác máy.

1.2 Phạm vi nghiên cứu và phương pháp nghiên cứu

1.2.1 Phạm vi nghiên cứu

Đề tài nghiên cứu về cách vận hành của một hệ thống di chuyển bám theo con người dựa trên mô hình máy học quy mô nhỏ, mô phỏng cụ thể một hệ thống bằng việc sử dụng phần cứng và đánh giá khả năng hệ thống một cách trực quan và tương đối thông qua khả năng vận hành thực tế

1.2.2 Phương pháp nghiên cứu

Tham khảo thêm từ những Group học tập trên mạng xã hội, trang web học tập, những video hướng dẫn trên Youtube, các bài viết về dự án xe bám người.

Khảo sát một số model huấn luyện sẵn trên mạng internet, khảo sát các robot bám người hiện hành để chọn lựa phương án thiết kế sau này.

1.3 Các đóng góp của đồ án

- Cung cấp mô hình nghiên cứu thực tế về hệ thống xe bám người bằng camera sử dụng các mô hình máy học và thị giác máy
- Đưa ra những đánh giá trong việc sử dụng các light-weight model đối với hiệu suất bám đối tượng
- Nhận xét những khả năng đáp ứng của hệ thống trong những điều kiện làm việc khác nhau

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VỀ MÁY HỌC

2.1 Máy học

Máy học (ML) là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc sử dụng dữ liệu và thuật toán để bắt chước hành động của con người, dần dần cải thiện độ chính xác của nó. Nó còn là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được đào tạo để đưa ra các phân loại hoặc dự đoán và khám phá những thông tin chi tiết từ chính các dự án khai thác dữ liệu.

Thông qua các thông tin chi tiết có được để thúc đẩy việc đưa ra quyết định đối với các ứng dụng và doanh nghiệp, tác động mạnh đến các chỉ số tăng trưởng. Khi dữ liệu lớn tiếp tục nhu cầu mở rộng và phát triển đòi hỏi nhu cầu tuyển dụng các nhà khoa học dữ liệu sẽ tăng lên. Họ sẽ được yêu cầu giúp xác định các câu hỏi kinh doanh có liên quan nhất và dữ liệu để trả lời chúng.

Hình 2-1: Máy học

Một số kiểu học máy chính thường được ứng dụng như: học giám sát (Supervised Learning), học không giám sát (Unsupervised Learning), học tăng cường (Reinforcement Learning),... Bài toán của ML thường được chia làm hai loại là dự đoán (prediction), phân loại (classification), phân nhóm (clustering), v.v. Các bài toán

dự đoán thường là giá nhà, giá xe, v.v, còn các bài toán phân loại thường là nhận diện chữ viết tay, đồ vật, v.v.

2.1.1 Các thuật toán phổ biến của ML:

- Neural networks: Mô phỏng cách thức hoạt động của bộ não con người, với một số lượng khổng lồ các nút xử lý được liên kết. Neural networks là thuật toán được dùng trong việc nhận dạng các mẫu và đóng một vai trò quan trọng trong các ứng dụng bao gồm dịch ngôn ngữ tự nhiên, nhận dạng hình ảnh, nhận dạng giọng nói và tạo hình ảnh.
- Linear regression: Thuật toán này được sử dụng để dự đoán các giá trị số, dựa trên mối quan hệ tuyến tính giữa các giá trị khác nhau.
- Logistic regression: Thuật toán giúp đưa ra dự đoán cho các biến phản hồi phân loại, chẳng hạn như câu trả lời "có/không" cho các câu hỏi. Nó có thể được sử dụng cho các ứng dụng như phân loại thư rác và kiểm soát chất lượng trên dây chuyền sản xuất.
- Clustering: Các thuật toán phân cụm có thể xác định các mẫu trong dữ liệu để nó có thể được nhóm lại. Máy tính có thể giúp các nhà khoa học dữ liệu bằng cách xác định sự khác biệt giữa các mục dữ liệu mà con người đã bỏ qua.
- Decision trees: Là thuật toán được sử dụng để dự đoán giá trị số (hồi quy) và phân loại dữ liệu. Decision trees sử dụng một chuỗi phân nhánh của các quyết định được liên kết có thể được biểu diễn bằng sơ đồ cây. Một trong những ưu điểm của decision trees là chúng dễ xác thực và kiểm tra, không giống thuật toán neural networks.
- Random forests: Trong một khu rừng ngẫu nhiên, thuật toán máy học dự đoán một giá trị hoặc danh mục bằng cách kết hợp các kết quả từ một số cây quyết định.

2.1.2 Ứng dụng của ML

Với sự đa dạng trong cách học và thuật toán, máy học được ứng dụng trong nhiều lĩnh vực khác nhau như:

- Trong lĩnh vực kinh tế: dự đoán giá nhà, giá cổ phiếu, dự đoán sự biến động cung cầu, phân nhóm khách hàng,…

- Trong lĩnh vực y tế: phát hiện bệnh, dự đoán diễn biến bệnh, phân nhóm bệnh nhân,…
- Trong lĩnh vực quản lý: nhận diện người, giám sát hành vi, quản lý đối tượng,…
- Trong lĩnh vực kỹ thuật: tích hợp với các thiết bị để trở nên thông minh hơn, đáp ứng tốt hơn yêu cầu của người sử dụng,…

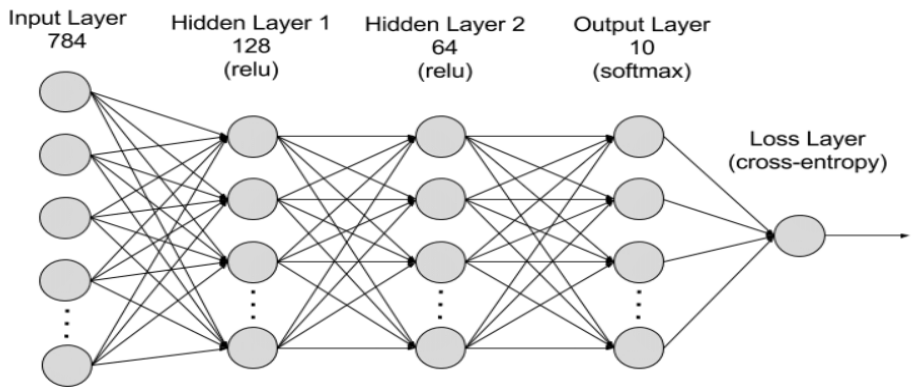
2.2 Mạng thần kinh nơ-ron và ứng dụng của CNN

2.2.1 Mạng thần kinh nơ-ron

Mạng thần kinh nơ-ron (Neural Network) là một phương thức trong lĩnh vực trí tuệ nhân tạo, được sử dụng để dạy máy tính xử lý dữ liệu theo cách được lấy cảm hứng từ bộ não con người. Đây là một loại quy trình máy học, được gọi là học sâu (Deep Learning), sử dụng các nút hoặc nơ-ron liên kết với nhau trong một cấu trúc phân lớp tương tự như bộ não con người. Phương thức này tạo ra một hệ thống thích ứng được máy tính sử dụng để học hỏi từ sai lầm của chúng và liên tục cải thiện.

Mạng thần kinh nơ-ron được ứng dụng nhiều trong thực tế tập trung ở bốn lĩnh vực chính như thị giác máy tính, nhận diện khuôn mặt, xử lí ngôn ngữ tự nhiên, công cụ đề xuất giải pháp,…

Một mạng nơ-ron thường có 3 lớp chính: lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer). Dựa vào phương thức truyền dữ liệu ta có thể chia mạng nơ-ron thành 3 kiểu chính: mạng nơ-ron truyền thẳng, mạng nơ-ron dùng thuật toán truyền ngược và mạng nơ-ron tích chập.



Hình 2-2: Kiến trúc mạng nơ-ron

2.2.2 CNN

Mạng nơ-ron tích chập (CNN) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

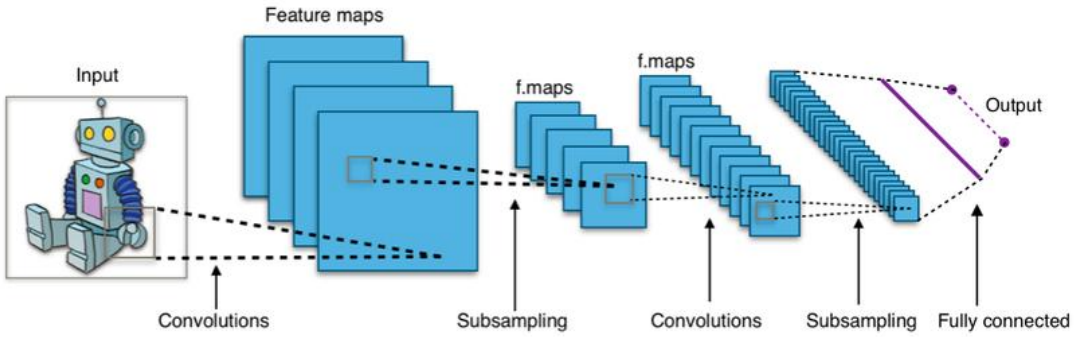
Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNN thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNN sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



Hình 2-3: Mạng CNN

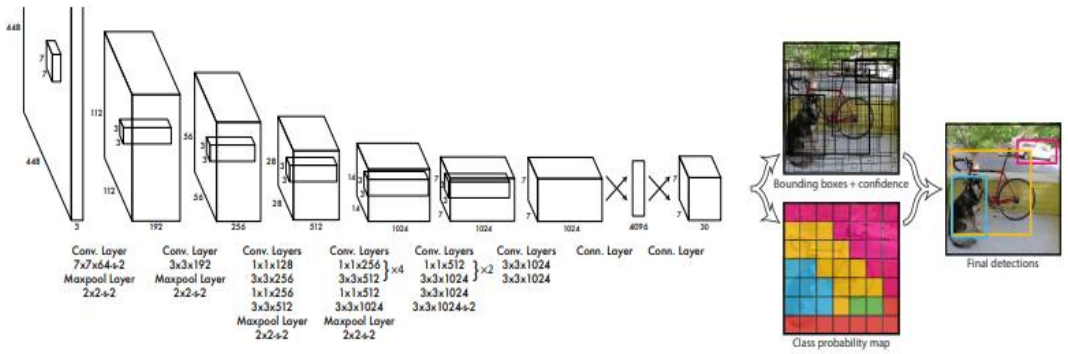
Trong mô hình CNN có 2 khía cạnh cần quan tâm là **tính bất biến** (Location Invariance) và **tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

Đó là lý do tại sao CNN cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

2.3 Mô hình chung YOLO

YOLO là thuật toán object detection nên mục tiêu của mô hình không chỉ là dự báo nhãn cho vật thể như các bài toán classification mà nó còn xác định location của vật thể. Do đó YOLO có thể phát hiện được nhiều vật thể có nhãn khác nhau trong một bức ảnh thay vì chỉ phân loại duy nhất một nhãn cho một bức ảnh.



Hình 2-4: Mô hình YOLO

Thuật toán YOLO lấy hình ảnh làm đầu vào, sau đó sử dụng mạng nơ-ron tích chập sâu đơn giản để phát hiện các đối tượng trong ảnh. Kiến trúc của mô hình CNN tạo thành xương sống của YOLO được hiển thị bên dưới.

20 lớp tích chập đầu tiên của mô hình được đào tạo trước với ImageNet bằng cách cắm vào một lớp tổng hợp trung bình tạm thời (temporary average pooling) và lớp được kết nối đầy đủ (fully connected layer). Sau đó, mô hình đào tạo trước này được chuyển đổi để thực hiện phát hiện. Lớp được kết nối đầy đủ cuối cùng của YOLO dự đoán cả xác suất của lớp và tọa độ hộp giới hạn.

YOLO chia hình ảnh đầu vào thành lưới S × S. Nếu tâm của một đối tượng rơi vào một ô lưới thì ô lưới đó có nhiệm vụ phát hiện đối tượng đó. Mỗi ô lưới dự đoán các hộp giới hạn B và điểm tin cậy cho các hộp đó. Các điểm tin cậy này phản ánh mức độ tin cậy của mô hình rằng hộp chứa một đối tượng và mức độ chính xác mà mô hình cho rằng hộp được dự đoán.

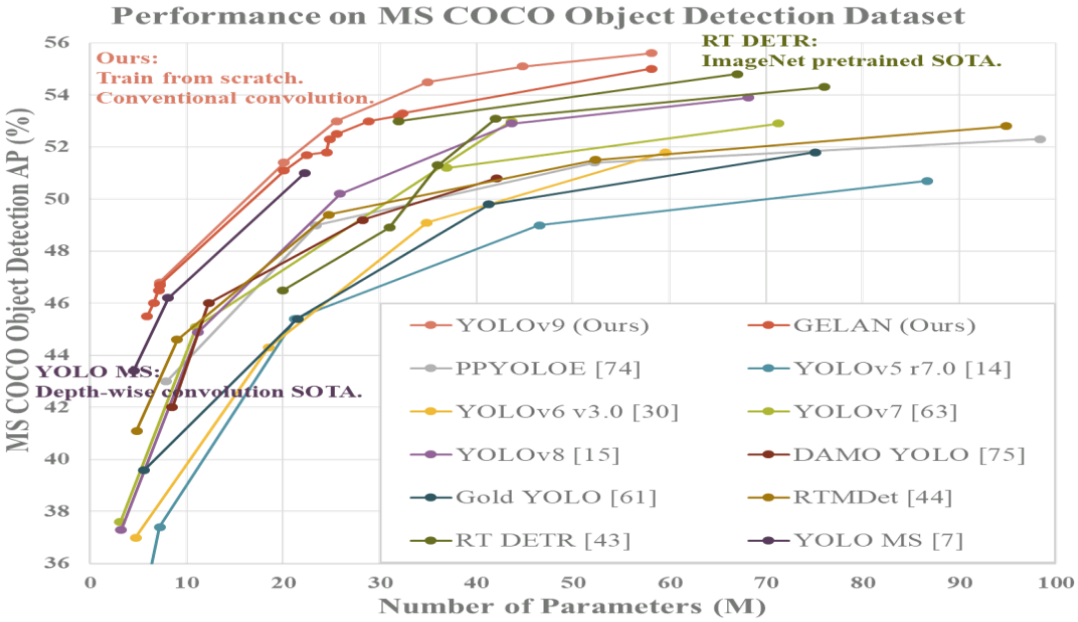
YOLO dự đoán nhiều hộp giới hạn trên mỗi ô lưới. Tại thời điểm đào tạo, ta chỉ muốn một bộ dự đoán hộp giới hạn thể hiện cho từng đối tượng. YOLO chỉ định bộ dự đoán dựa trên chỉ số IOU hiện tại cao nhất với thực tế. Điều này dẫn đến sự chuyên môn hóa giữa các bộ dự đoán hộp giới hạn. Mỗi công cụ dự đoán trở nên tốt hơn trong việc dự báo các kích thước, tỷ lệ khung hình hoặc loại đối tượng nhất định, cải thiện tổng thể recall score.

Một kỹ thuật quan trọng được sử dụng trong các mô hình YOLO là NMS (non-maximum suppression). NMS là một bước hậu xử lý được sử dụng để cải thiện độ chính xác và hiệu quả của việc phát hiện đối tượng. Trong phát hiện đối tượng, thông thường có nhiều hộp giới hạn được tạo cho một đối tượng trong một hình ảnh. Các hộp giới hạn này có thể chồng lên nhau hoặc nằm ở các vị trí khác nhau, nhưng tất cả

chúng đều đại diện cho cùng một đối tượng. NMS được sử dụng để xác định và loại bỏ các hộp giới hạn dư thừa hoặc không chính xác và đề xuất một hộp giới hạn duy nhất cho từng đối tượng trong ảnh.

2.4 Ứng dụng của YOLO và YOLO tiny trong bài toán object detection

Với những đặc điểm trên, YOLO là một mô hình nên được cân nhắc sử dụng cho các nhiệm vụ phát hiện vật thể. Nhóm thuật toán này có khả năng nhận diện và xử lí nhanh, hầu hết nhóm này được đánh giá cao về khả năng "realtime" trong nhận diện vật thể. Có thể xem như đây là nhóm thuật toán có ứng dụng cao trong các ứng dụng cần thời gian xử lí nhanh với yêu cầu độ chính xác ở mức "tương đối". Hiện nay các YOLO đã liên tục ra các phiên bản nâng cấp của để cải thiện về độ chính xác và tốc độ nhận diện. Phiê n bản mới nhất mà YOLO hiện có là YOLOv9



Hình 2-5: Độ chính xác của YOLO qua các phiên bản

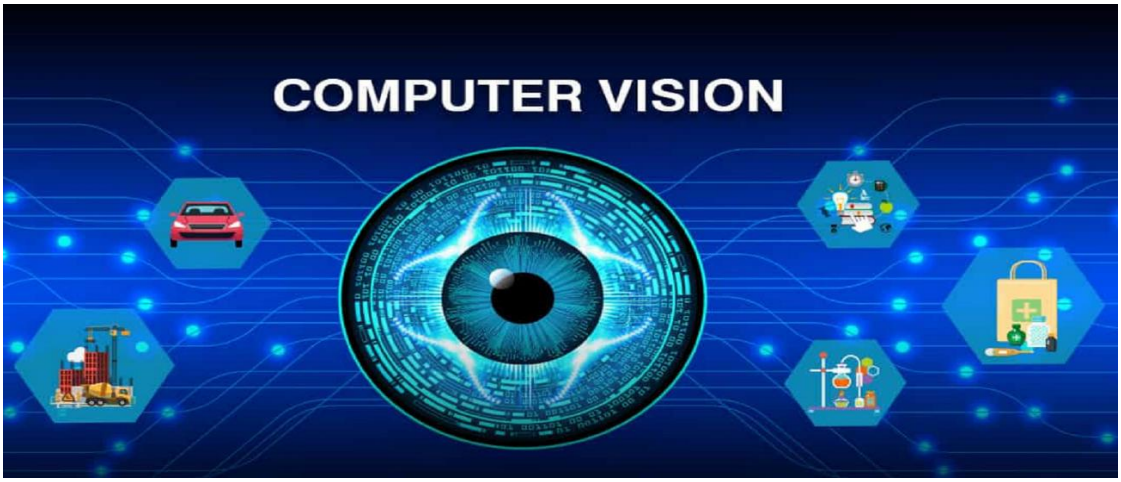
Một phiên bản nhỏ hơn của mô hình YOLO (hay còn được gọi là light-weight model) là YOLO tiny thường được ứng dụng trong các nhiệm vụ phát hiện và nhận diện vật thể với tốc độ xử lí (FPS) rất cao. Mặc dù được phát triển từ mô hình YOLO nhưng có kích thước mô hình nhỏ nên khả năng khoanh vùng chính xác đối tượng không cao như mô hình YOLO. Vì vậy, YOLO tiny thường được ứng dụng cho các nhiệm vụ phát hiện vật thể real-time nhanh và thường được ứng dụng vào các phần cứng cỡ nhỏ

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT VỀ THỊ GIÁC MÁY

3.1 Tổng quan về thị giác máy

Ngày nay, tiến bộ trong lĩnh vực này kết hợp với sự tăng cường đáng kể của sức mạnh điện toán đã cải thiện cả quy mô và độ chính xác của quy trình xử lý dữ liệu hình ảnh. Các hệ thống thị giác máy tính được hỗ trợ bởi tài nguyên điện toán đám mây hiện giờ trở nên dễ tiếp cận với tất cả mọi người. Bất kỳ tổ chức nào cũng có thể sử dụng công nghệ này để xác minh danh tính, kiểm duyệt nội dung, phân tích video phát trực tuyến, phát hiện lỗi và nhiều tính năng khác.

Thị giác máy hay thị giác máy tính (Computer Vision) là một lĩnh vực khoa học máy tính liên quan đến việc xử lý và hiểu thông tin từ hình ảnh và video. Thị giác máy tính sử dụng các thuật toán để trích xuất các đặc trưng từ dữ liệu hình ảnh, chẳng hạn như đường viền, góc và màu sắc. Các đặc trưng này sau đó được sử dụng để xác định các đối tượng, thực hiện các nhiệm vụ phân loại và theo dõi chuyển động..Thị giác máy là một công nghệ mà thiết bị sử dụng tự động nhận biết và mô tả hình ảnh một cách chính xác và hiệu quả. Các hệ thống máy tính có quyền truy cập vào khối lượng lớn hình ảnh và dữ liệu video bắt nguồn từ hoặc được tạo bằng điện thoại thông minh, camera giao thông, hệ thống bảo mật và các thiết bị khác.



Hình 3-1: Thị giác máy

Ứng dụng thị giác máy sử dụng trí tuệ nhân tạo và máy học (AI/ML) để xử lý những dữ liệu này một cách chính xác nhằm xác định đối tượng và nhận diện khuôn mặt, cũng như phân loại, đề xuất, giám sát và phát hiện.

Hệ thống thị giác máy tính sử dụng công nghệ trí tuệ nhân tạo (AI) để bắt chước khả năng của não người trong việc nhận biết đối tượng và phân loại đối tượng. Các nhà khoa học máy tính đào tạo máy tính nhận biết dữ liệu hình ảnh bằng cách nhập khối lượng lớn thông tin. Thuật toán máy học (ML) xác định các kiểu mẫu thông thường trong những hình ảnh hoặc video này và áp dụng kiến thức đó để xác định chính xác những hình ảnh chưa biết. Ví dụ: nếu máy tính xử lý hàng triệu hình ảnh ô tô, chúng sẽ bắt đầu xây dựng kiểu mẫu nhận dạng và có thể phát hiện chính xác phương tiện trong một hình ảnh. Thị giác máy tính sử dụng các công nghệ như được đưa ra dưới đây.

Khác với hướng phát triển của máy học, thị giác máy tính chủ yếu tập trung vào việc xử lí các hình ảnh, video và phân loại cụ thể cho từng đối tượng phát hiện được. Các nhiệm vụ chính mà thị giác máy tính hướng đến bao gồm: phân loại ảnh (Image Classification), phát hiện đối tượng (Object Detection), phân đoạn (Segmentation), theo dõi đối tượng (Object Tracking), nhận dạng kí tự quang học (Optical Character Recognition),…

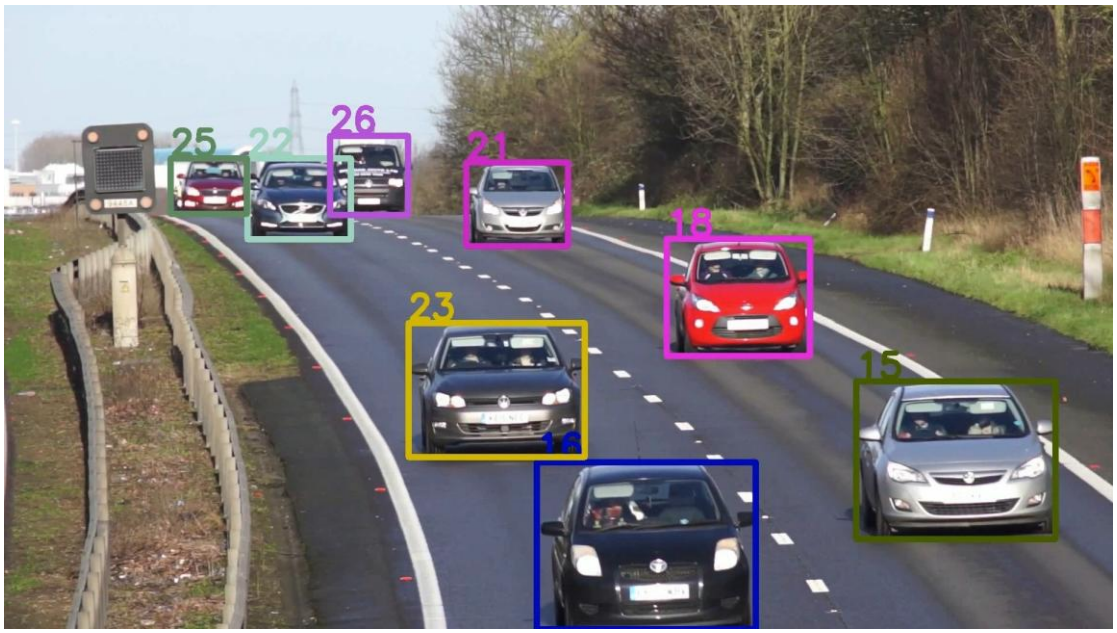
3.2 Thị giác máy tính và nhiệm vụ theo dõi đối tượng

3.2.1 Khái niệm về theo dõi đối tượng

Theo dõi đối tượng (Object Tracking) là một trong những nhiệm vụ điển hình của thị giác máy được phát triển nhằm theo dõi một hoặc nhiều đối tượng chuyển động theo thời gian trong một video. Khác với phát hiện đối tượng (Object Detection), theo dõi đối tượng là việc máy phải phát hiện được đối tượng từ lúc xuất hiện đến lúc rời khỏi khung hình và có định danh cho từng đối tượng đó ví dụ như việc ta sẽ sẽ theo dõi chiếc xe hơi đó từ lúc nó xuất hiện cho đến khi nó rời khỏi khung hình, duy trì danh tính của chiếc xe qua từng khung hình.

Để có thể theo dõi đối tượng trong video hoặc camera realtime, việc phát hiện và định vị đối tượng có trong khung hình cần được thực hiện trước. Điều này cần sự kết hợp của mô hình học sâu (Deep Learning) cùng các kỹ năng xử lí ảnh. Khi đã định vị được đối tượng, các thuật toán theo dõi đối tượng (Tracking Algorithm) sẽ được áp dụng để liên tục cập nhật sự thay đổi vị trí của đối tượng qua các khung hình và định danh cho đối tượng đó một giá trị nào đó (ID). Vì vậy, các bài toán theo dõi đối tượng

sẽ tập trung vào việc cố định giá trị ID của đối tượng không đổi qua các khung hình, vấn đề tái định danh đối tượng khi đối tượng biến mất khỏi khung hình hay phương pháp nâng cao khả năng theo dõi để đáp ứng được tính thời gian thực.



Hình 3-2: Theo dõi đối tượng

3.2.2 *Phân loại theo dõi đối tượng*

Dựa vào phương pháp tiếp cận của bài toán theo dõi đối tượng, ta có thể chia thành 2 loại chính:

- Theo dõi một đối tượng (SOT): tập trung vào việc theo dõi một đối tượng duy nhất trong toàn bộ video. Và tất nhiên, để biết được cần theo dõi đối tượng nào, việc cung cấp một bounding box từ ban đầu là việc bắt buộc phải có.
- Theo dõi nhiều đối tượng (MOT): hướng tới các ứng dụng có tính mở rộng cao hơn. Bài toán cố gắng phát hiện đồng thời theo dõi tất cả các đối tượng trong tầm nhìn, kể cả các đối tượng mới xuất hiện trong video. Vì điều này, MOT thường là những bài toán khó hơn SOT và nhận được rất nhiều sự quan tâm của giới nghiên cứu.

Ngoài ra, bài toán còn có thể chia theo phương pháp xử lí video như sau:

- Online Tracking: Khi xử lí video, Online Tracking chỉ sử dụng frame hiện tại và frame ngay trước đó để tracking. Cách xử lí này có thể sẽ làm giảm độ chính xác của thuật toán, tuy nhiên nó lại phản ảnh đúng cách vấn đề được xử lí trong thực tế, khi mà tính "online" là cần thiết
- Offline Tracking: Các phương pháp Offline thường sử dụng toàn bộ frame của video, do đó thường đạt được độ chính xác cao hơn nhiều so với Online Tracking

3.2.3 Các thách thức trong theo dõi đối tượng

Theo dõi đối tượng cũng là một bài toán mang nhiều thách thức đáng quan tâm và luôn không ngừng cải tiến để đem lại hiệu quả cao hơn. Các thách thức mà nó gặp phải bao gồm:

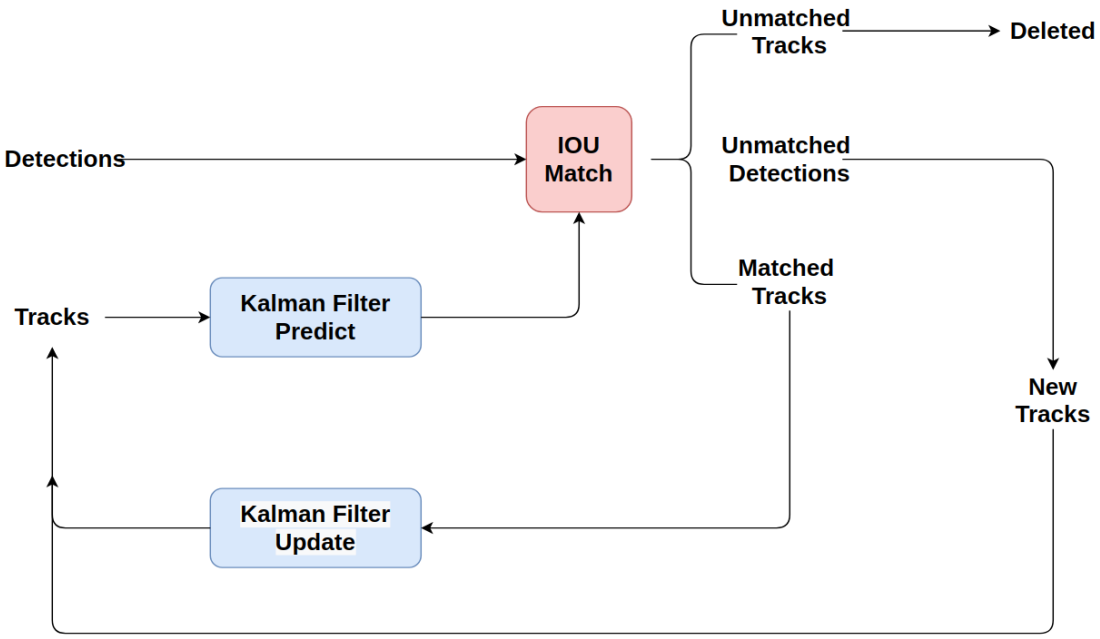
- Phát hiện được tất cả các đối tượng: Để có thể theo dõi tốt đối tượng qua các khung hình liên tục, việc phát hiện đối tượng ngay từ những khung hình đầu tiên là rất cần thiết. Điều này còn phụ thuộc vào khả năng của từng mô hình máy học và hoàn cảnh cụ thể của đối tượng
- Đối tượng bị che khuất một phần hoặc toàn bộ: Đây là một trong những vấn đề khó điển hình của bài toán theo dõi đối tượng. Bởi vì khi một giá trị ID được gán cho 1 đối tượng, ID đó cần đảm bảo nhất quán trong suốt video, tuy nhiên, khi một đối tượng bị che khuất, nếu chỉ dựa riêng vào object detection là không đủ để giải quyết vấn đề này mà cần phải có phương pháp xử lí phù hợp
- Đối tượng ra khỏi khung hình và xuất hiện trở lại: tương tự như trường hợp đối tượng bị che khuất, đây cũng là một trong những vấn đề nổi bật mà khi thực hiện theo dõi đối tượng phải luôn chú ý. Việc của chúng ta là phải làm giảm số ID và tăng tính nhất quán cho từng đối tượng bằng việc tái nhận dạng đối tượng khi đối tượng đó xuất hiện trở lại khung hình.
- Các đối tượng có quỹ đạo chuyển động giao nhau hoặc chồng chéo lên nhau: Việc các đối tượng có quỹ đạo chống chéo lên nhau cũng có thể dẫn đến hậu quả gán nhầm ID cho các đối tượng, đây cũng là vấn đề chúng ta cần chú ý xử lí khi làm việc với MOT

3.3 Ứng dụng SORT và DeepSORT trong object tracking

3.3.1 SORT

SORT là một giải thuật được triển khai theo dạng tracking-by-detection, nghĩa là thực hiện theo dõi bằng cách sử dụng các hộp giới hạn (bounding box) của nhiều đối tượng được phát hiện từ mỗi khung hình của chuỗi hình ảnh. Quá trình xử lí với mỗi khung hình của SORT như sau:

- Đầu tiên, việc phát hiện (detect) đối tượng được thực hiện trước để xác định vị trí của các đối tượng
- Sau khi có vị trí của đối tượng, SORT sẽ dự đoán (predict) vị trí mới của các đối tượng dựa vào các frame trước đó
- Cuối cùng, SORT sẽ liên kết (associate) các vị trí đã phát hiện với các vị trí dự đoán được để gán ID tương ứng



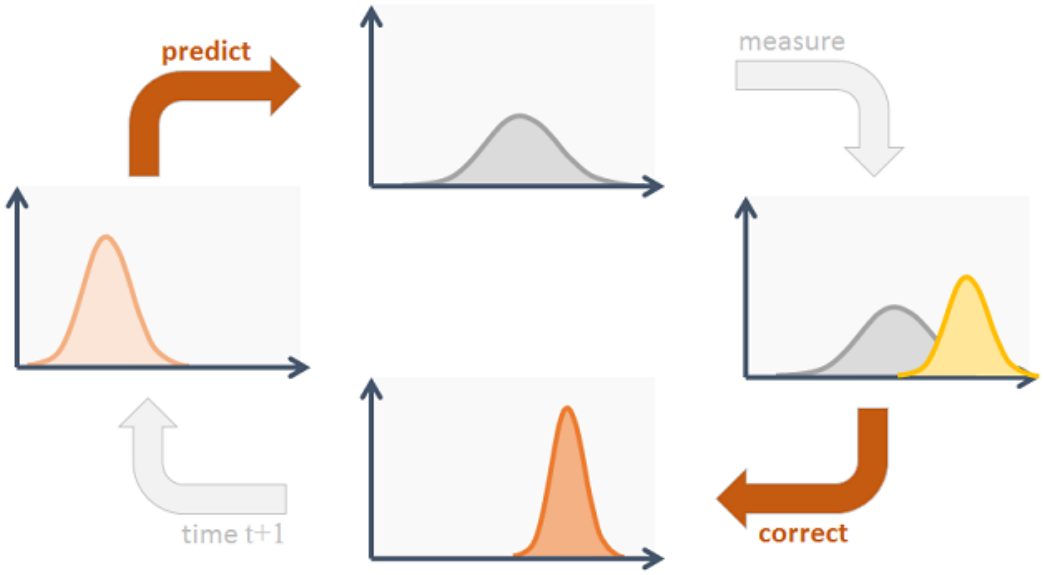
Hình 3-3: Lưu đồ giải thuật của SORT

3.3.1.1 Các thuật toán chính trong SORT

- Giải thuật Hungary (Hungary Algorithm): giải thuật này giải quyết bài toán gán nhãn tối ưu (optimal assignment) giữa các đối tượng được phát hiện (detections) và các đối tượng đang theo dõi (trackers). SORT xây dựng một ma trận chi phí (cost matrix) dựa trên khoảng cách giữa các

đối tượng được phát hiện mới và các đối tượng đang theo dõi hiện có. Nhiệm vụ chính của giải thuật Hungary trong SORT là để tìm ra cách kết hợp tốt nhất giữa các đối tượng được phát hiện mới và các trackers hiện có sao cho tổng chi phí (thường là khoảng cách hoặc sai số) là nhỏ nhất.

- Bộ lọc Kalman (KF): bộ lọc này đóng vai trò quan trọng trong việc dự đoán vị trí tiếp theo của đối tượng và cập nhật trạng thái của đối tượng dựa trên các quan sát mới. Đầu tiên, SORT sẽ sử dụng KF (thường là Linear KF) để dự đoán vị trí (x,y) và vận tốc (vx,vy) của đối tượng trong khung hình tiếp theo. Khi có phát hiện mới (bounding box mới), KF cập nhật trạng thái dự đoán bằng cách kết hợp thông tin từ quan sát mới và trạng thái dự đoán. Sau đó KF sẽ tính toán trọng số (weights) và ma trận hiệp phương sai (covariance matrix) để xác định độ tin cậy



Hình 3-4: Kalman Filter

3.3.1.2 Những vấn đề cần cải thiện của SORT

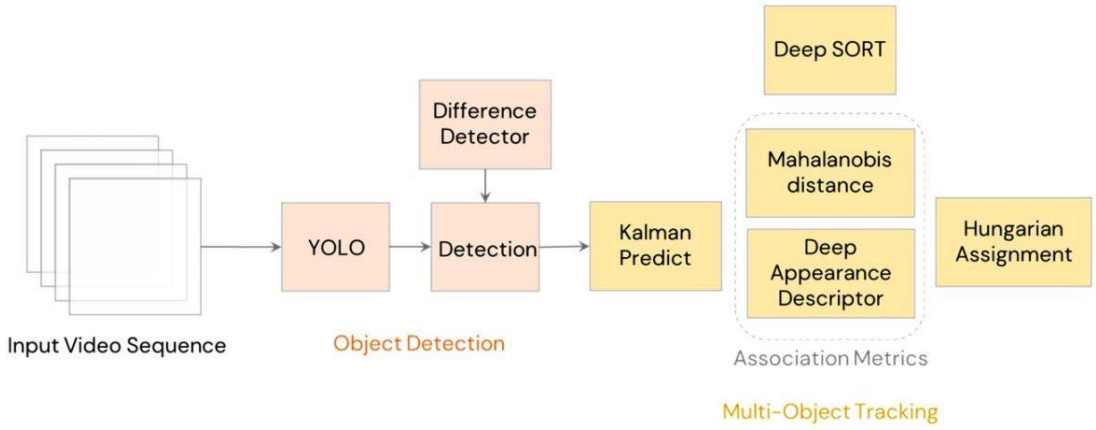
Việc sử dụng KF tuyến tính (Linear Kalman Filter) chưa thực sự hiệu quả, thay vào đó cần phải sử dụng các KF phức tạp hơn để tăng khả năng tracking như Extended Kalman filter, Unscented Kalman filter,…

Vấn đề ID Switches – xảy ra do việc có trackers không được phát hiện và gán ID để sau vài khung hình ID của đối tượng đó được xem là không tồn tại và bị xóa.

Điều đó có nghĩa là đối tượng theo dõi đó khi được cập nhật sẽ được tiếp tục gán ID mới dẫn đến việc chuyển đổi ID liên tục

3.3.2 DeepSORT

DeepSORT là phần mở rộng của thuật toán SORT, sử dụng bộ lọc Kalman để theo dõi đối tượng. DeepSORT kết hợp số liệu liên kết sâu dựa trên các đặc điểm bề ngoài được mạng nơ-ron tích chập sâu học được. Điều này cho phép thuật toán xử lý các tình huống trong đó các đối tượng có thể tạm thời biến mất hoặc bị che khuất trong luồng video. DeepSORT cũng kết hợp tính năng gán ID để theo dõi các đối tượng riêng lẻ trên nhiều khung, điều này rất quan trọng đối với các ứng dụng như giám sát, xe tự hành và tương tác giữa người với máy tính. Thuật toán thực hiện cách tiếp cận hai giai đoạn: đầu tiên tạo ra các phát hiện đối tượng và sau đó liên kết các phát hiện đó với các dấu vết hiện có. Nhìn chung, DeepSORT đã cho thấy những cải tiến đáng kể về độ chính xác và độ mạnh mẽ của việc theo dõi so với các thuật toán theo dõi truyền thống, khiến nó trở thành một công cụ có giá trị trong các ứng dụng thị giác máy tính và trí tuệ nhân tạo.



Hình 3-5: Kiến trúc của mô hình DeepSORT

3.3.2.1 Các đặc điểm của DeepSORT

DeepSORT sử dụng thêm 2 độ đo mới gồm: khoảng cách Mahalanobis và khoảng cách cosine để tăng khả năng theo dõi đối tượng. Khoảng cách Mahalanobis cung cấp các thông tin về vị trí đối tượng dựa trên chuyển động tức thời, đặc biệt hữu ích cho các dự đoán ngắn hạn, tập trung vào đo lường khoảng cách giữa track và detection và còn được dùng để loại trừ các liên kết không chắc chắn bằng cách lập ngưỡng khoảng cách Mahalanobis. Bên cạnh đó, khoảng cách cosine xem xét thông

tin về đặc trưng của đối tượng nhằm đảm bảo việc liên kết chuẩn xác dù đối tượng đã biến mất và sau đó xuất hiện trở lại trong khung hình, đặc biệt hữu ích cho các dự đoán dài hạn hoặc các đối tượng khó phân biệt.

DeepSORT sử dụng một giải thuật một chiến lược đối sánh theo tầng (Matching Cascade) để tối ưu cho việc gán kết các tracker một cách hiệu quả. Chiến lược đối sánh theo tầng tiến hành lấy lần lượt từng track ở các frame trước đó, để tiến hành xây dựng ma trận chi phí và giải bài toán phân công theo từng tầng. Việc liên kết được tiến hành theo nguyên tắc ưu tiên gán ID cho các tracker tồn tại qua ít khung hình nhất rồi mới đến các tracker xuất hiện nhiều. Điều này giúp mô hình ít gán nhầm ID và giúp các đối tượng có cơ hội được theo dõi ở khung hình tiếp theo cao hơn

Listing 1 Matching Cascade

Input: Track indices $\mathcal{T} = \{1, \ldots, N\}$, Detection indices $\mathcal{D} = \{1, \ldots, M\}$, Maximum age $A_{\max}$

Compute cost matrix $C = [c_{i,j}]$
Compute gate matrix $B = [b_{i,j}]$
Initialize set of matches $\mathcal{M} \leftarrow \emptyset$
Initialize set of unmatched detections $\mathcal{U} \leftarrow \mathcal{D}$
for $n \in \{1, \ldots, A_{\max}\}$ **do**
  Select tracks by age $\mathcal{T}_n \leftarrow \{i \in \mathcal{T} \mid a_i = n\}$
  $[x_{i,j}] \leftarrow$ min cost matching $(C, \mathcal{T}_n, \mathcal{U})$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{(i,j) \mid b_{i,j} \cdot x_{i,j} > 0\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{j \mid \sum_i b_{i,j} \cdot x_{i,j} > 0\}$
end for
return $\mathcal{M}, \mathcal{U}$

Hình 3-6: Chiến lược đối sánh theo tầng

3.3.2.2 Ưu thế của DeepSORT

So với SORT, DeepSORT là phiên bản được cải thiện nhằm tối ưu hóa hơn cho việc theo dõi vật thể. Tương tự như SORT, DeepSORT cũng sẽ thực hiện việc liên kết giữa detect và track dựa trên IOU. Ngoài ra DeepSORT còn sử dụng các yếu tố khoảng cách khác như: khoảng cách Mahalanobis (khoảng cách của detection và track mà xét theo tính tương quan trong không gian vector) và khoảng cách cosine

giữa 2 vector đặc trưng được trích xuất từ detection và track - 2 vector đặc trưng của cùng 1 đối tượng sẽ giống nhau hơn là đặc trưng của 2 đối tượng khác nhau. Việc sử dụng nhiều yếu tố sẽ giúp xác định đối tượng dễ dàng và giữ được đối tượng tốt hơn qua các khung hình tiếp theo.

DeepSORT cải thiện hiệu suất SORT bằng cách tích hợp thông tin về giao diện. Điều này có nghĩa là DeepSORT sử dụng mô hình học sâu để đưa ra các đặc điểm xuất hiện cho các vùng hộp giới hạn được phát hiện bởi mô hình phát hiện đối tượng. Các tính năng xuất hiện từ mô hình học sâu được chuyển đổi thành ma trận chi phí và ma trận chi phí được sử dụng để thực hiện liên kết, phát hiện và theo dõi dữ liệu. Kết quả là số lượng chuyển đổi ID do tắc nghẽn, vốn là một vấn đề của SORT, đã giảm đi một cách hiệu quả.

CHƯƠNG 4. **HỆ THỐNG THEO DÕI CON NGƯỜI**

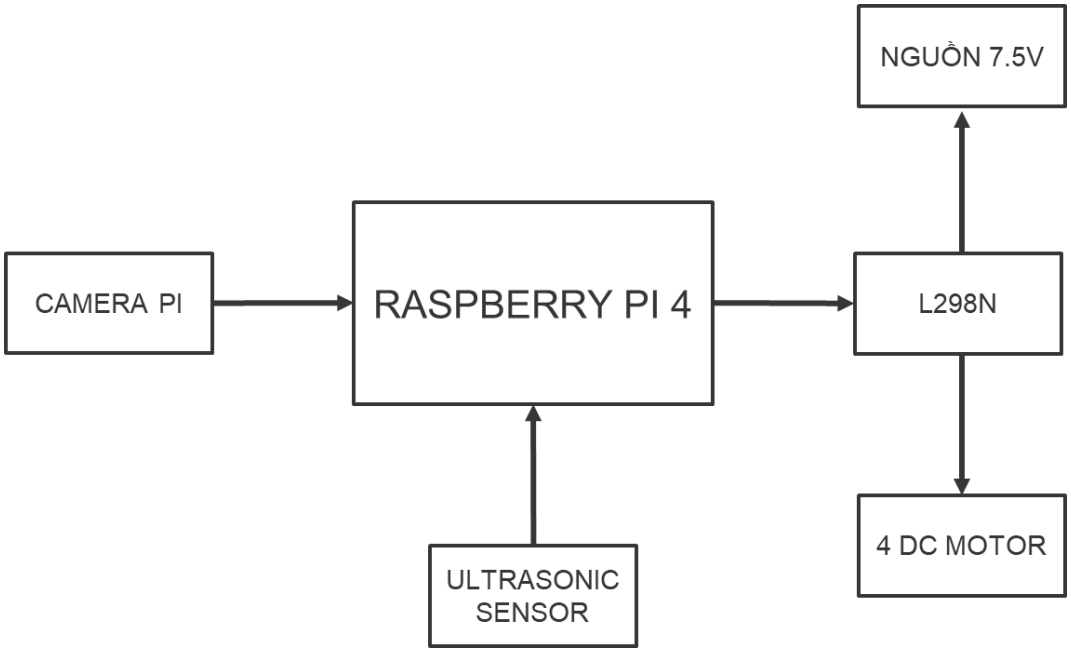
4.1 Phương pháp tiếp cận

Hệ thống theo dõi con người (hay hệ thống bám người) là một hệ thống công nghệ được thiết kế để phát hiện, theo dõi và giám sát sự di chuyển của con người trong không gian. Hệ thống này sử dụng các cảm biến, thiết bị ghi hình, và các thuật toán phân tích để nhận diện và theo dõi vị trí cũng như hoạt động của các đối tượng con người qua nhiều khung hình video hoặc trong môi trường thực tế. Hệ thống thường sử dụng các ứng dụng về máy học, thị giác máy và các kỹ thuật xử lí ảnh, kết hợp cảm biến và có thể điều khiển động cơ để vận hành hệ thống di chuyển và bám theo chuyển động của con người, cụ thể là một người đã được chỉ định từ trước.

Đề tài tập trung xây dựng hệ thống theo dõi con người bằng camera được điều khiển bởi Raspberry Pi, sử dụng mô hình YOLO và DeepSORT kết hợp với hệ thống khung động cơ để bám theo chuyển động con người. Mô hình chung của hệ thống được xây dựng theo hướng dưới đây:

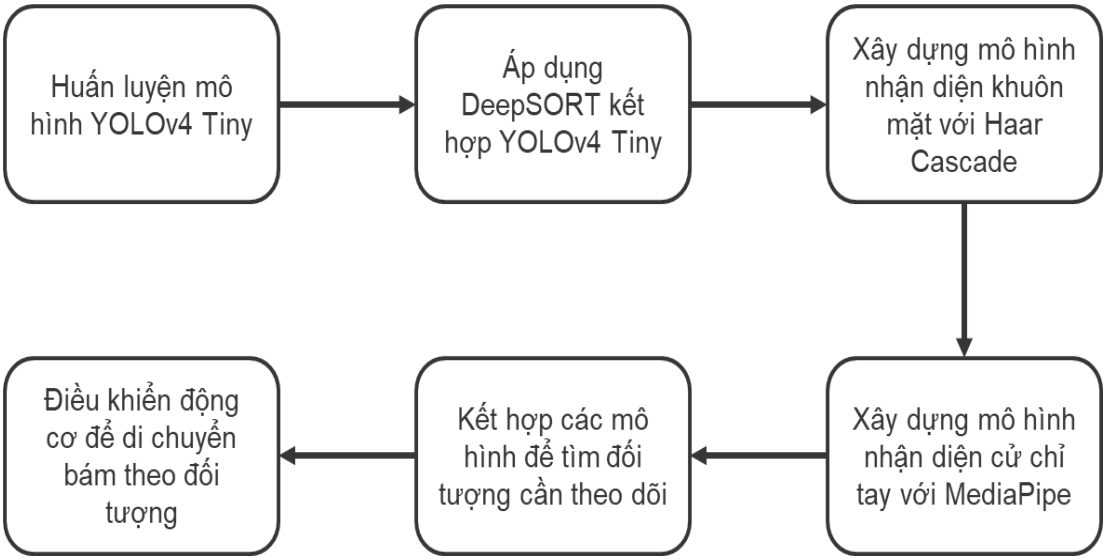
- Đầu tiên, ta cần huấn luyện một mô hình để lấy hộp giới hạn (bounding box) của đối tượng. Mô hình được sử dụng là YOLOv4 Tiny
- Sau khi định vị được đối tượng, ta áp dụng DeepSORT kết hợp với mô hình YOLOv4 Tiny để tiến hành theo dõi đối tượng
- Ta sử dụng thêm các mô hình nhận diện mặt và tay để hỗ trợ cho việc điều khiển hệ thống
- Điều chỉnh động cơ để hệ thống có thể di chuyển bám theo đối tượng

Từ định hướng về phương pháp thực hiện trên, ta thiết kế được tổng quan về phần cứng của hệ thống và mô hình hóa thông qua sơ đồ sau:



Hình 4-1: Sơ đồ phần cứng hệ thống

Với thiết kế hệ thống như trên, ta xây dựng được một quy trình (pipeline) cần thiết để thực hiện hệ thống bao gồm:



4.2 Chuẩn bị dữ liệu đầu vào (Dataset)

Để thực hiện đề tài cần xây dựng 3 mô hình gồm mô hình phát hiện con người YOLOv4 Tiny, mô hình nhận diện khuôn mặt để phát hiện đối tượng chỉ định và một mô hình nhận diện cử chỉ tay để điều khiển hoạt động. Việc chuẩn bị dữ liệu về 3 mô hình được thực hiện như sau:

- YOLOv4 Tiny: Mô hình được huấn luyện từ bộ dữ liệu đầu vào là ảnh người với các nhiều tư thế và khung nền khác nhau. Các ảnh được làm giàu (data augmentation) hoặc thay đổi tính chất để tăng khả năng nhận diện mô hình dưới những điều kiện đông người. Nguồn dataset được lấy từ Roboflow với kích thước 640 x 640 và tiến hành gán nhãn đối tượng để phù hợp hơn với mô hình thông qua một bộ công cụ là labelImg được cung cấp bởi Python. Bộ dữ liệu gồm 2384 ảnh được dùng cho huấn luyện (train) và 565 ảnh dùng cho đánh giá mô hình (validation)
- Face Recognition: Mô hình nhận diện khuôn mặt sử dụng ảnh đầu vào gồm 150 khuôn mặt ở gần và xa camera của đối tượng cần theo dõi để làm dữ liệu cho mô hình Haar Cascade
- Gesture Recognition: Mô hình nhận diện cử chỉ tay sử dụng dữ liệu đầu vào gồm các dạng ảnh về cử chỉ tay mỗi dạng gồm 200 ảnh.

4.3 Tiền xử lí dữ liệu (Preprocessing)

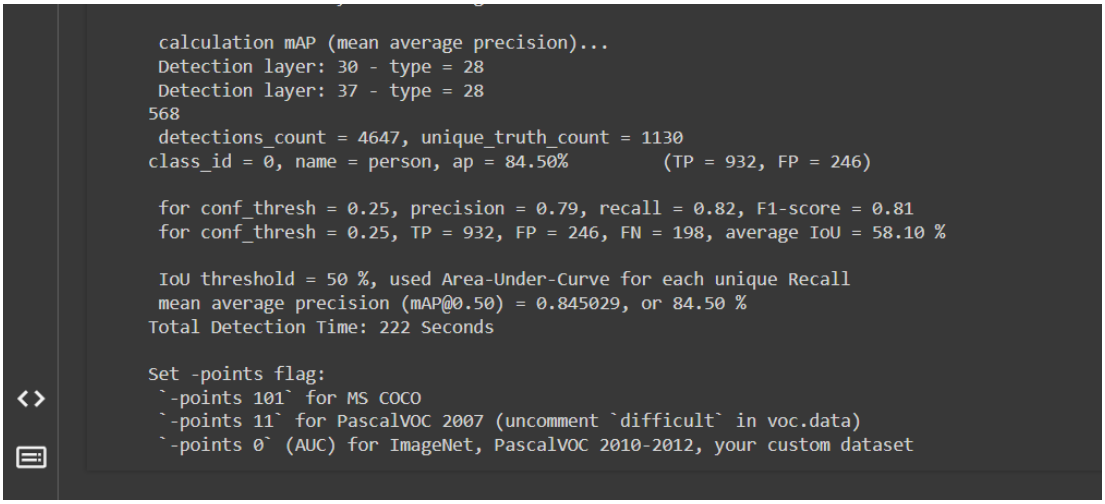
Sau khi đã chuẩn bị bộ dữ liệu đầu vào cho các mô hình, việc xử lí và đưa dữ liệu về đúng dạng đầu vào của mô hình là rất cần thiết. Chẳng hạn, việc chia dữ liệu thành train và validation để phù hợp cho mô hình YOLOv4 Tiny cần khởi tạo thêm 2 file train.txt và valid.txt để chứa đường dẫn đến các ảnh huấn luyện và đánh giá. Việc này được thực hiện thông qua các kỹ năng xử lí lấy tên và phân chia dữ liệu bằng code Python.

4.4 Huấn luyện mô hình

4.4.1 Mô hình Tiny YOLOv4

Mô hình được thực hiện theo dựa trên hướng dẫn từ nguồn github của AlexeyAB về huấn luyện các mô hình YOLO darknet, trong đó có YOLOv4 Tiny. Việc sử dụng mô hình nhẹ (light-weight model) này nhằm mục đích tăng khả năng phát hiện đối tượng của mô hình và phù hợp hơn với phần cứng xử lí thấp như Raspberry Pi

Quá trình huấn luyện được thực hiện trên Colab với 3248 vòng (epochs) và cho độ chính xác tương đối tốt (khoảng 84.5%). Kết quả này dựa trên một công cụ đánh giá độ chính xác (mAP50) được cung cấp như hình sau:



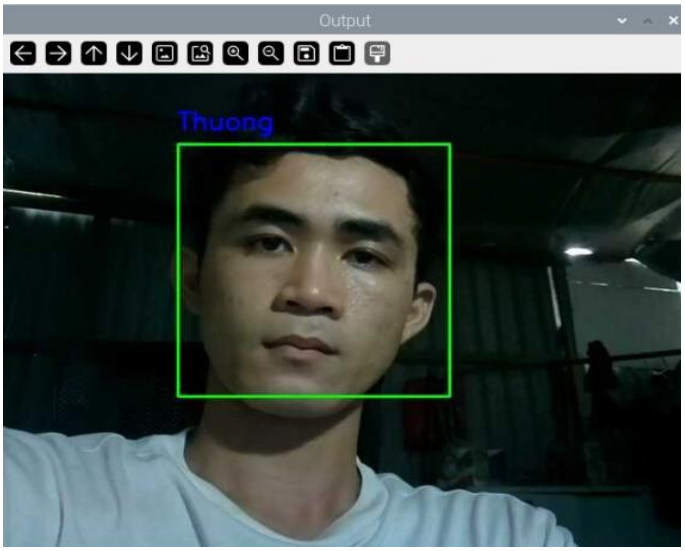
Hình 4-2: Đánh giá mô hình YOLOv4 Tiny

4.4.2 Mô hình nhận diện khuôn mặt với Haar Cascade

Phương pháp nhận diện khuôn mặt với Haar Cascade là một cách hiệu quả để nhận diện khuôn mặt trong thời gian thực. Mặc dù nó không mạnh mẽ và chính xác như các phương pháp học sâu hiện đại như CNN hoặc các mô hình dựa trên deep learning, Haar Cascade vẫn là một lựa chọn tốt cho các ứng dụng cần hiệu suất cao và yêu cầu tính toán thấp. Nó thực hiện nhận diện dựa trên các đặc trưng Haar và một thuật toán học máy gọi là Cascade Classifier

Các đặc trưng Haar là các mẫu đơn giản được sử dụng để tìm các đặc điểm bao gồm các cạnh, các đường thẳng, các góc, và các vùng khác nhau của đối tượng trong một hình ảnh. Mỗi đặc trưng Haar là sự khác biệt của tổng giá trị điểm ảnh trong các vùng đen và trắng. Cascade Classifier là một chuỗi các bộ phân loại đơn giản và hiệu quả, được xếp chồng lên nhau theo từng giai đoạn. Mỗi giai đoạn là một bộ phân loại yếu, nhưng được tối ưu hóa để nhanh chóng loại bỏ các vùng không chứa đối tượng. Tuy nhiên, quá trình huấn luyện một mô hình khá phức tạp nên ta thường sử dụng một mô hình đã được huấn luyện sẵn và lưu trữ dưới dạng file xml có tên "haarcascade_frontalface_default.xml" được cung cấp trên github của OpenCV.

Mô hình được huấn luyện cho đề tài sử dụng 150 ảnh là khuôn mặt của đối tượng chỉ định cần theo dõi. Khi kết thúc quá trình huấn luyện, dữ liệu khuôn mặt của đối tượng sẽ được mã hóa và lưu trữ trong file "encodings.pickle" và được sử dụng khi nhận diện khuôn mặt.

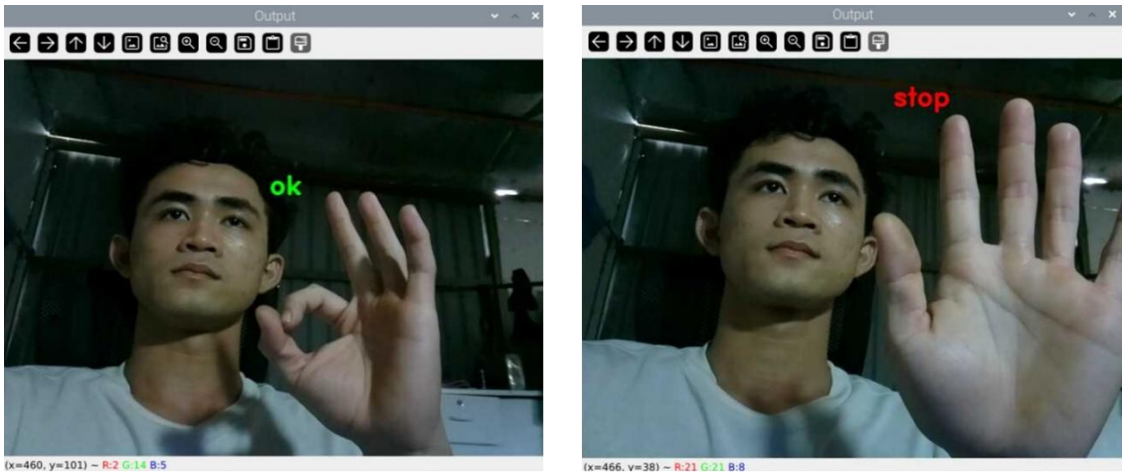


Hình 4-3: Nhận diện khuôn mặt

4.4.3 Mô hình nhận diện cử chỉ tay với Mediapipe

Với những nhiệm vụ nhận diện khuôn mặt, cử chỉ tay hay tư thế con người dưới dạng các điểm tư thế (landmark), việc sử dụng Mediapipe là một trong những lợi thế. Mediapipe cung cấp các thuật toán để hỗ trợ việc theo dõi các landmark và có thể sử dụng mô hình máy học để huấn luyện trạng thái để nhận diện được các tư thế hay cử chỉ tay mong muốn.

Đề tài xây dựng mô hình nhận diện cử chỉ tay với 2 trạng thái “ok” và “palm” để điều khiển việc tiếp tục hay dừng theo dõi đối tượng của hệ thống. Việc huấn luyện dựa trên các hướng dẫn được cung cấp bởi MediaPipe để tạo ra file mô hình “gesture_recognizer.task”.



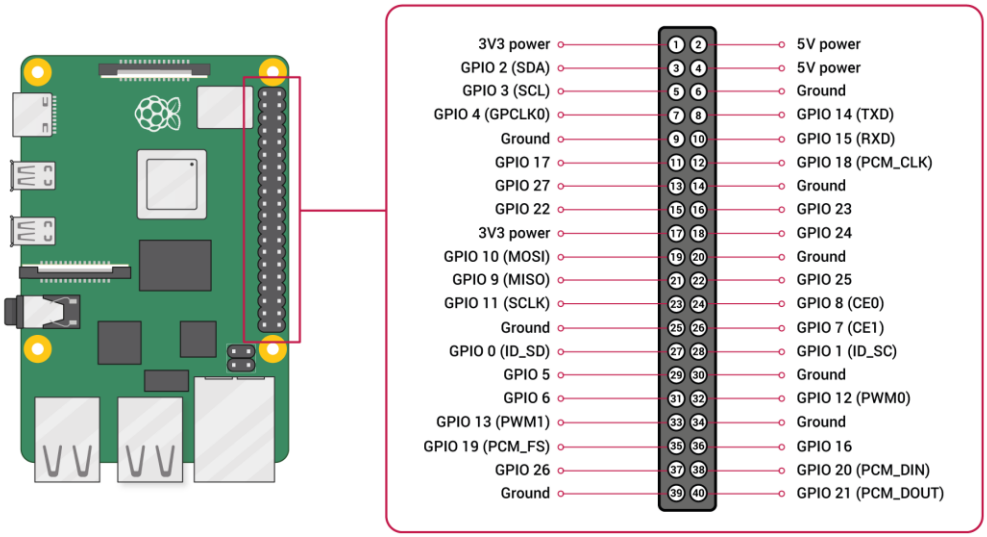
Hình 4-4: Nhận diện cử chỉ tay

4.5 Triển khai mô hình trên hệ thống

4.5.1 Thiết kế phần cứng hệ thống

Các mô hình được kết hợp và triển khai trên một hệ thống gồm Raspberry được gắn với khung xe động cơ. Pi sẽ truyền tín hiệu từ các GPIO đến mạch lái động cơ L298N để tiến hành băm xung PWM và điều khiển 4 động cơ di chuyển theo đối tượng. Module L298N sẽ sử dụng nguồn pin gồm 5 pin AA với tổng mức điện áp là 7.5 V để cấp nguồn và điều khiển cho 4 động cơ DC. Raspberry Pi sẽ được cấp nguồn từ sạc dự phòng 5V- 3A.

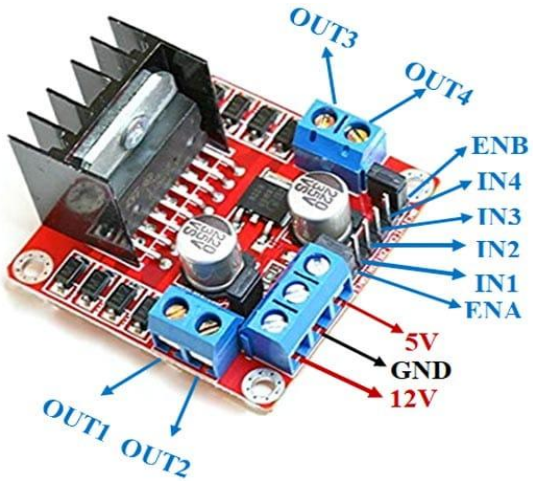
Phần điều khiển trung tâm của hệ thống là Raspberry Pi 4, sử dụng 6 chân GPIO 23, 24, 25, 8, 7, 1 để điều khiển lần lượt các chân ENA, IN1, IN2, IN3, IN4 trên module L298N. Ngoài ra, để tạo xung phát và nhận tín hiệu từ cảm biến siêu âm HC – SR04, Raspberry Pi cần cấp thêm 2 chân GPIO 26 để kết nối với chân Trigger và GPIO 19 để kết nối với chân Echo trên cảm biến.



Hình 4-5: Các chân GPIO của Raspberry Pi 4

Module điều khiển động cơ L298N với điện áp cấp nguồn được cho phép trong khoảng 5 – 30 VDC và nó có thể được sử dụng để điều khiển tối đa 4 động cơ. Các chân ENA, ENB được sử dụng để bật tắt động cơ, băm xung PWM để điều chỉnh tốc độ quay của động cơ và các chân IN1, IN2, IN3, IN4 để điều khiển 4 động cơ với hướng quay thuận (tiến lên) hoặc hướng quay ngược (lùi lại). Module có 4 cổng

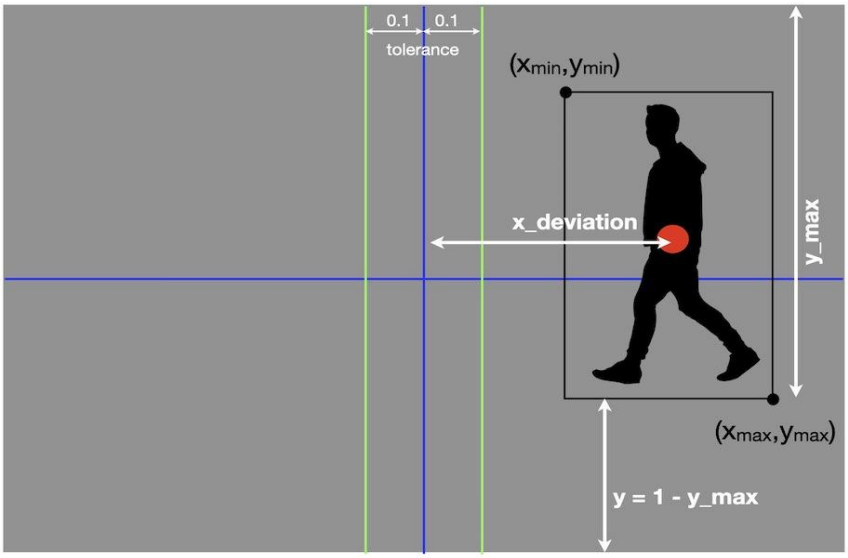
OUT1, OUT2, OUT3, OUT4 để điều khiển động cơ. Hệ thống sẽ sử dụng 2 cổng OUT1, OUT2 để điều khiển chung cho 2 động cơ bên trái và OUT3, OUT4 để điều khiển 2 động cơ còn lại.



Hình 4-6: Module điều khiển động cơ L298N

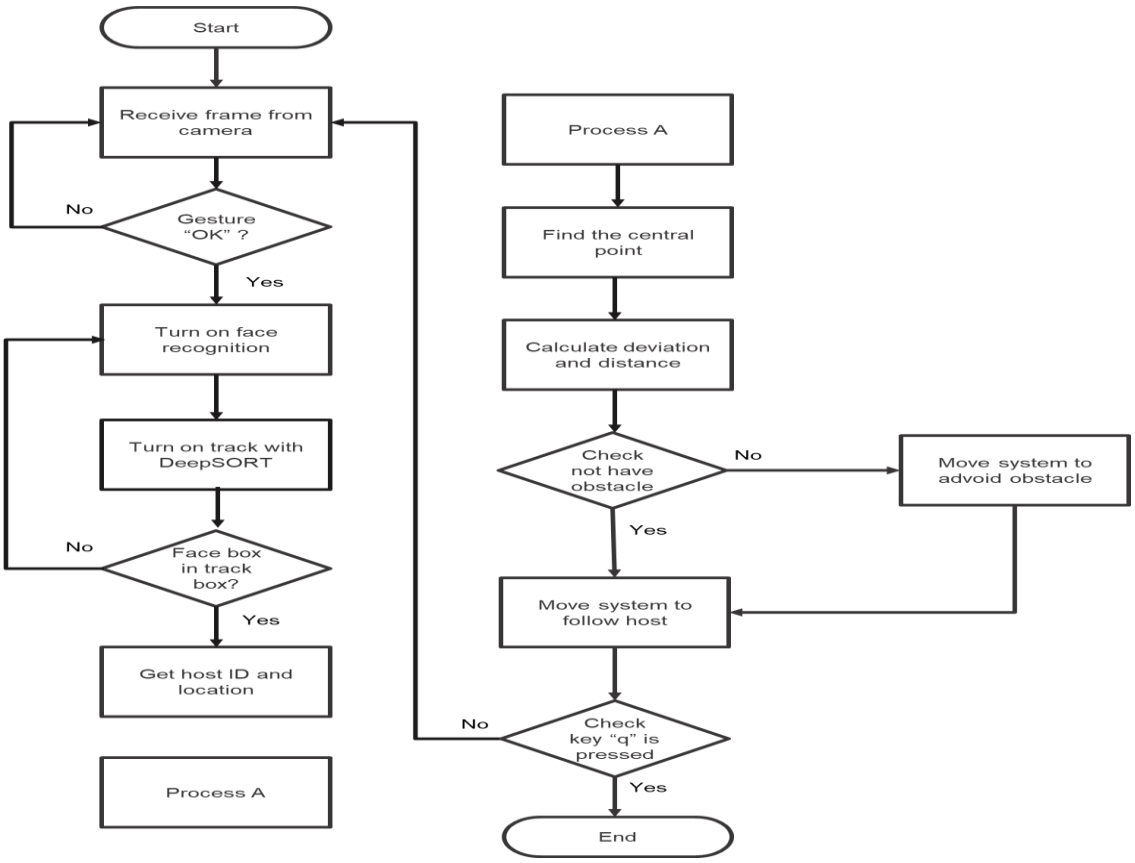
4.5.2 Triển khai phần mềm

Hệ thống hoạt động dựa trên sự kết hợp của các mô hình tracking DeepSORT, các mô hình nhận diện khuôn mặt, cử chỉ tay và điều khiển động cơ. Đầu tiên, hệ thống sẽ luôn bắt đầu với trạng thái tìm host bằng cách bật cả 3 mô hình để phát hiện người, tìm khuôn mặt và nhận diện cử chỉ tay. Nếu trong quá trình thực thi, hệ thống phát hiện được khuôn mặt chỉ định nằm trong phần bounding box của người nào đó đã có thì nó sẽ gán cho đối tượng đó là "host". Hệ thống sẽ luôn chờ cử chỉ tay từ host để xác định việc tiếp tục tiến hành hoặc dừng lại. Khi đã định vị được host, mô hình nhận diện khuôn mặt sẽ được tắt đi và giữ lại 2 mô hình còn lại để theo dõi đối tượng host. Vì có được các bounding box và ID của host, ta tiến hành tính được điểm ở trung tâm bounding box của host, từ đó xác định các sai lệch (deviation) theo trục x và khoảng cách (distance) từ camera đến người dựa trên khung hình. Cuối cùng dựa theo các thông số về độ lệch và khoảng cách mà hệ thống đưa ra quyết định (đi thẳng, rẽ trái, rẽ phải, dừng lại,…). Trong suốt quá trình di chuyển, nếu gặp phải vật cản, hệ thống sẽ ưu tiên việc tránh vật cản rồi mới tiếp tục di chuyển theo host.



Hình 4-7: Định hướng di chuyển của hệ thống

Lưu đồ giải thuật của hệ thống được mô tả thông qua hình sau:

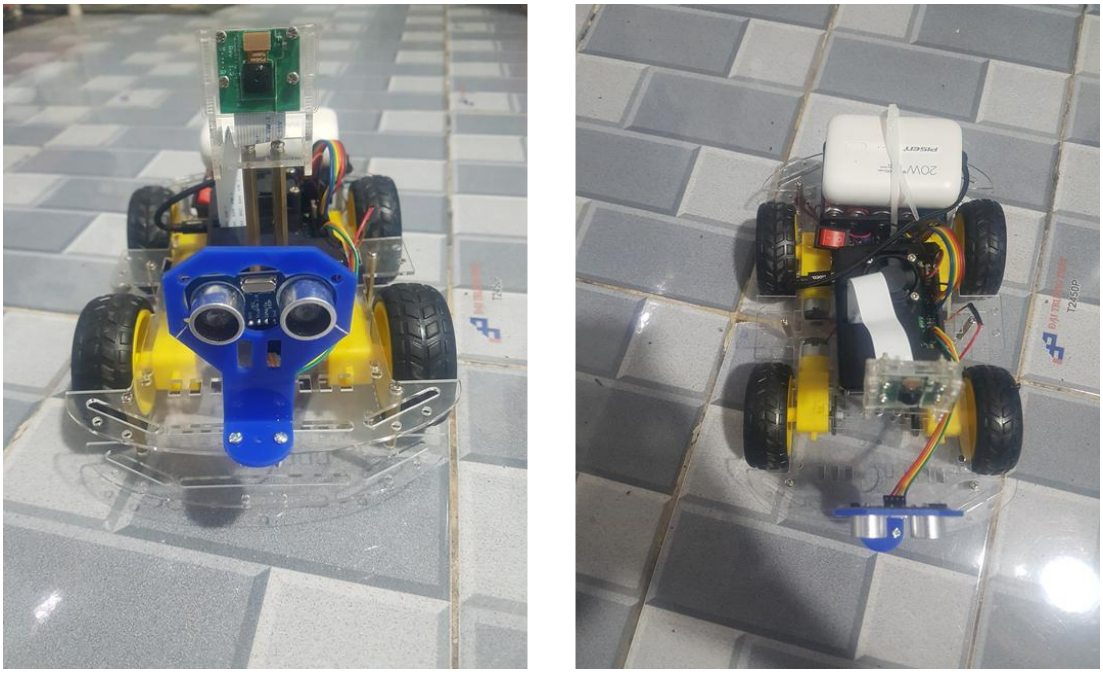


Hình 4-8: Lưu đồ giải thuật của hệ thống

CHƯƠNG 5. **KẾT QUẢ VÀ ĐÁNH GIÁ**

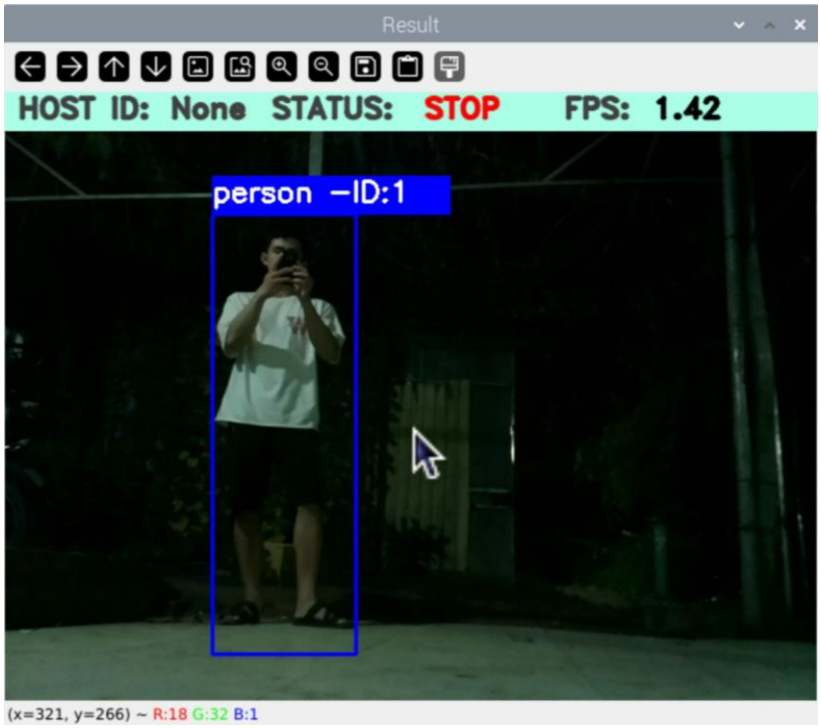
5.1 Kết quả thực hiện

Mô hình được triển khai thành công trên Rapsberry Pi 4 với tốc độ khung hình khoảng 1 – 1.5 FPS. Mô hình phần cứng thực tế của hệ thống như sau:



Hình 5-1: Mô hình hệ thống thực tế

Lúc đầu hệ thống không phát hiện đối tượng hoặc đối tượng phát hiện được không phải host, điều này được gây ra bởi việc không tìm thấy face của host trong các đối tượng đã phát hiện được. Hệ thống sẽ đánh dấu các đối tượng đó bằng một bounding box màu xanh dương.



Hình 5-2: Đối tượng tracking không là host

Sau khi đã phát hiện được host, hệ thống sẽ đánh dấu đối tượng host bằng bounding box màu đỏ để phân biệt các đối tượng còn lại. Ta sẽ có được tọa độ điểm trung tâm của host (điểm màu vàng) để sử dụng cho điều khiển động cơ.

Hình 5-3: Phát hiện đối tượng host và tracking

Dựa vào điểm trung tâm và tọa độ ymax của bounding box host ta có thể xác định hướng và khoảng cách mà động cơ cần di chuyển.

Hình 5-4: Hướng động cơ sang trái theo đối tượng host

5.2 Đánh giá hệ thống

- Về khả năng phát hiện đối tượng: Hệ thống sử dụng mô hình YOLOv4 Tiny đem lại khả năng bắt đối tượng nhanh và tương đối chính xác. Hệ thống phát hiện được dễ dàng các đối tượng con người ở các trạng thái khác nhau hoặc trong hoàn cảnh có nhiều người. Mặc dù có thể phát hiện nhanh nhưng việc sử dụng mô hình light-weight khả năng định vị chưa thật sự tốt dẫn đến các bounding box của đối tượng có thể không ổn định hoặc mất vị trí đối tượng
- Về khả năng theo dõi đối tượng: Hệ thống có thể phát hiện và theo dõi được nhiều đối tượng và khả năng phân biệt host tương đối tốt trong điều kiện đủ sáng. Tuy nhiên, việc đối tượng host di chuyển quá nhanh hoặc đột ngột thay đổi hướng chuyển động sẽ gây ảnh hưởng đến việc theo dõi đối tượng.

- Về khả năng điều khiển động cơ bám đối tượng: Hệ thống về cơ bản có thể dựa vào thông tin tracking của đối tượng host để tiến hành bám theo. Bởi vì tốc độ xử lí chưa cao nên quá trình tracking đối tượng và điều khiển động cơ có sự trễ nhất định.
- Về thời gian hoạt động của hệ thống: Hệ thống sử dụng nguồn sạc dự phòng 10000mAh để cấp nguồn cho Raspberry Pi và nguồn 7.5V để cấp cho module L298N vận hành 4 động cơ và cảm biến siêu âm. Nhìn chung, thời gian thời gian hoạt động cũng tương đối tốt và đáp ứng yêu cầu sử dụng.
- Về khả năng tránh vật cản: Hệ thống có sử dụng module HC-SR04 để tránh vật cản thông qua sóng siêu âm và có thể tránh một số vật thể đơn giản. Tuy nhiên, vì những sai số nhất định việc tránh vật cản chỉ tốt đối với những vật cản nhỏ.

CHƯƠNG 6. **KẾT LUẬN**

6.1 Tóm tắt và kết luận chung

Đề tài xây dựng được cơ bản một hệ thống theo dõi con người dựa trên các ứng dụng của mô hình máy học và thị giác máy, vận dụng kết hợp các kỹ năng xử lí dữ liệu và kết hợp mô hình để có một hệ thống hoàn thiện. Bên cạnh đó là việc tìm hiểu các cơ sở lý thuyết và giải thuật phù hợp cho hệ thống để có thể vận hành trên các phần cứng nhỏ như Raspberry Pi

Với những tìm hiểu và nghiên cứu, bài báo cáo đã chỉ ra được phương pháp để xây dựng một mô hình bám người dựa trên camera. Bằng việc sử dụng mô hình phát hiện cỡ nhỏ và phương pháp theo dõi DeepSORT đã đem lại hiệu quả trong phát hiện và bám theo chuyển động của đối tượng chỉ định.

Mặc dù hệ thống có khả năng theo dõi và bám đối tượng chỉ định nhưng việc chạy mô hình trên Raspberry Pi còn chưa thực sự tốt. Việc sử dụng liên tiếp các mô hình làm giảm đi khá nhiều FPS, làm giảm đi tính real-time của hệ thống. Việc theo dõi qua camera cũng còn hạn chế về tầm nhìn và chủ yếu theo dõi tốt trong các trường hợp đơn giản. Cuối cùng, việc xử lí chậm kéo theo các tín hiệu điều khiển động cơ không được liên tục

6.2 Hướng phát triển

- Cải thiện hệ thống với khả năng bám người tốt hơn thông qua việc tối ưu các mô hình hoặc sử dụng các mô hình phù hợp hơn
- Sử dụng phần cứng có khả năng xử lí mạnh mẽ hơn như Jetson Nano, Intel Nuc,…
- Có thể thêm các giải thuật hoặc cảm biến để hỗ trợ hệ thống

TÀI LIỆU THAM KHẢO

[1]. Bùi Trung Nghĩa (26/12/2022), Nguyễn Văn Nam, *Human following and collision avoidance control of mobile robots, by vision-based deep neural network*

[2]. Bui Tien Tung (16/12/2020), *SORT - Deep SORT : Một góc nhìn về Object Tracking (phần 1),* https://viblo.asia/p/sort-deep-sort-mot-goc-nhin-ve-object-tracking-phan-1-Az45bPooZxY

[3]. Đào Văn Hậu (13/03/2019), *Nhận diện khuôn mặt với OpenCV trên Raspberry Pi,* https://pivietnam.com.vn/nhan-dien-khuon-mat-voi-opencv-tren-raspberry-pi-phan-1-pivietnam-com-vn.html

[4]. Fan Zhang, Valentin Bazarevsky, (18 Jun 2020), *MediaPipe Hands: On-device Real-time Hand Tracking*, https://arxiv.org/abs/2006.10214

[5]. Isaac, *GPIO: tất cả về kết nối Raspberry Pi 4 và 3*, https://www.hwlibre.com/vi/gpio-m%C3%A2m-x%C3%B4i-pi/

[6]. Spark, *AI Robot - Human Following Robot using TensorFlow Lite on Raspberry Pi,* https://helloworld.co.in/article/ai-robot-human-following-robot-using-tensorflow-lite-raspberry-pi

[7]. Tim (16 February 2023), *Hand Recognition and Finger Identification with Raspberry Pi and OpenCV,* https://core-electronics.com.au/guides/hand-identification-raspberry-pi/

[8]. Việt Hoàng (16/11/2019), Tìm hiểu về YOLO trong bài toán real-time object detection, https://viblo.asia/p/tim-hieu-ve-yolo-trong-bai-toan-real-time-object-detection-yMnKMdvr57P

[9]. AlexeyAB, *YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet)*, https://github.com/AlexeyAB/darknet

[10]. theAIGuysCode, *Object tracking implemented with YOLOv4, DeepSort, and TensorFlow,* https://github.com/theAIGuysCode/yolov4-deepsort

PHỤ LỤC

Code của hệ thống được lưu trữ tại github sau: https://github.com/duongthuongkk/Human-following-system.git