TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐIỆN TỬ - VIỄN THÔNG

-----OO🕮OO-----

BÁO CÁO BTL:

Phát hiện, theo dõi và xác định vận tốc phương tiện bằng camera

Giảng viên môn học: Ts. Võ Lê Cường

Nhóm sinh viên thực hiện:

Bùi Trung Kiên	20172638
Phạm Duy Khoa	20172631
Đào Thế Anh	20172409

Hà Nội, 1-2022

LỜI NÓI ĐẦU

Xử lý hình ảnh, đặc biệt là xử lý hình ảnh trong thời gian thực(real time) đang là lĩnh vực nóng trong sự phát triển khoa học công nghệ hiện nay. Nó hứa hẹn sẽ trở thành con mắt cho các robot để góp phần tăng khả năng tự động hóa trong sản xuất, từng bước giúp robot có khả năng thay thế con người trong các công việc hàng ngày.

Được sự giúp đỡ của thầy Võ Lê Cường đã tạo điều kiện cho bọn em là những sinh viên còn đang ngồi trên ghế nhà trường có cơ hội thực hiện đè tài cùng doanh nghiệp. Đây là một điều vô cùng quý giá đối với bọn em.

Qua trao đổi với các anh/ chị trong công ty MQ solutions, bọn em đã tiến tới thống nhất chọn đề tài "**Phát hiện, theo dõi và xác định vận tốc phương tiện bằng camera**". Báo cáo sẽ gồm 3 phần chính: Phát hiện đối tương, Theo dõi đối tượng và cuối cùng là xác định vận tốc.

Chúng em xin chân thành cảm ơn thầy và các anh/ chị đã nhiệt tình giúp đỡ trong quá trình bọn em thực hiện bài tập lớn.

MỤC LỤC

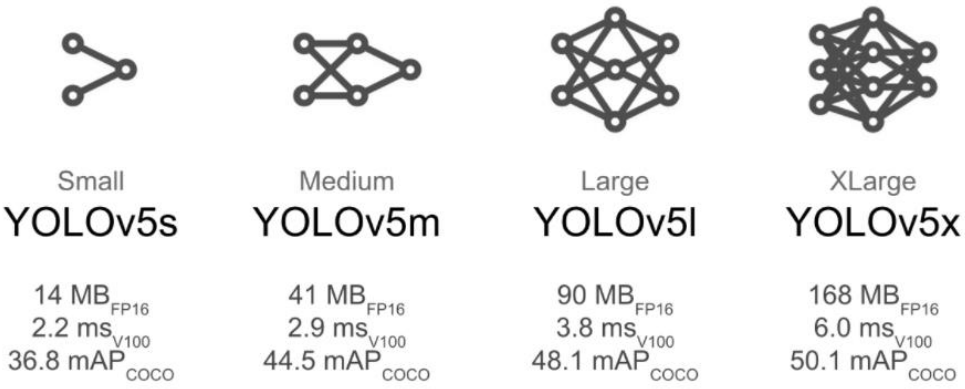
DANH SÁCH HÌNH ẢNH

CHƯƠNG 1. PHÁT HIỆN ĐỐI TƯỢNG VỚI YOLOv5 (OBJECT DETECTION)

1. Huấn luyện mô hình đào tạo

1.1. Tổng quan về YOLOv5

Object Detection đề cập đến khả năng của hệ thống máy tính và phần mềm để phát hiện các đối tượng trong một hình ảnh và xác định từng đối tượng. Trong phần này, ta sẽ thực hiện nhận dạng và phát hiện đối tượng, cụ thể ở đây là xe ô tô di chuyển trong một file video. Để thực hiện hóa ý tưởng này, nhóm em áp dụng mô hình YOLOV5. Khác với các bản YOLO trước, YOLOv5 cung cấp 4 phiên bản với kiến trúc mạng khác nhau:



Hình 1: Các mô hình huấn luyện

1.2. Cài đặt YOLOv5 trên Google Colab

Trước khi training model ta cần chuẩn bị một số thư viện cũng như môi trường như sau: sao chép kho lưu trữ, tải xuống tập dữ liệu hướng dẫn và cài đặt các thư viện bao gồm Python ≥ 3.8 và Pytorch ≥ 1.7 bằng các lệnh:

!git clone https://github.com/ultralytics/yolov5 # clone repo
!cd yolov5

!pip install -r requirements.txt # install

1.3. Đào tạo mô hình huấn luyện riêng

1.3.1. Tạo dataset.yaml

COCO128 là một tập dữ liệu hướng dẫn nhỏ bao gồm 128 hình ảnh đầu tiên trong COCO train2017. Đây là thư viện chuẩn mà YOLOV5 hay sử dụng để train mẫu. Các hình ảnh giống nhau này được sử dụng cho cả quá trình đào tạo và xác thực để xác minh quy trình đào tạo và phát hiện model có khả năng bị overfit. Trong báo cáo này, nhóm em sẽ điều chỉnh lại một số chi tiết để phù hợp với đề tài của dự án. Ta có ***coco128.yaml*** hiển thị ở bên dưới

```yaml
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017) by Ultralytics
# Example usage: python train.py --data coco128.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco128  ← downloads here


# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
path: ../datasets/coco128  # dataset root dir
train: /content/coco128  # train images (relative to 'path') 128 images
val: /content/coco128  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
nc: 1  # number of classes
names: ['car']  # class names
```

1.3.2. Gán nhãn

Tiếp theo, ta đến với bước tạo nhãn cho hình ảnh. Ở đây, nhóm em dùng công cụ ***makesense.ai*** được thể hiện ở hình bên dưới:

Hình 2: Gán nhãn

Sau đó, ta xuất nhãn ra định dạng YOLO với một tệp ***.txt*** cho mỗi hình ảnh. Cấu trúc của tệp ***.txt*** là:

- Một hàng cho mỗi đối tượng
- Mỗi hàng có định dạng: class x_center y_center width height
- Tọa độ hộp phải ở định dạng xywh chuẩn hóa (từ 0 - 1)
- Số lớp (số nhãn của đối tượng) được bắt đầu từ 0

Tệp nhãn tương ứng với hình trên chứa 5 xe ô tô (thuộc lớp 2). Ta có file định dạng txt có thông tin như sau:

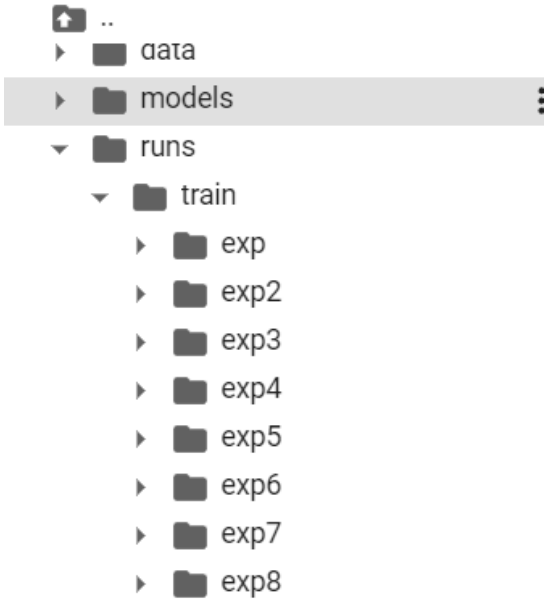
```
2 0.156095 0.812153 0.230100 0.332077
2 0.562811 0.540956 0.152985 0.281159
2 0.765547 0.404804 0.160448 0.283373
2 0.436567 0.267546 0.144279 0.185963
2 0.537935 0.110362 0.105721 0.159397
```

1.3.3. Huấn luyện cho mô hình phát hiện

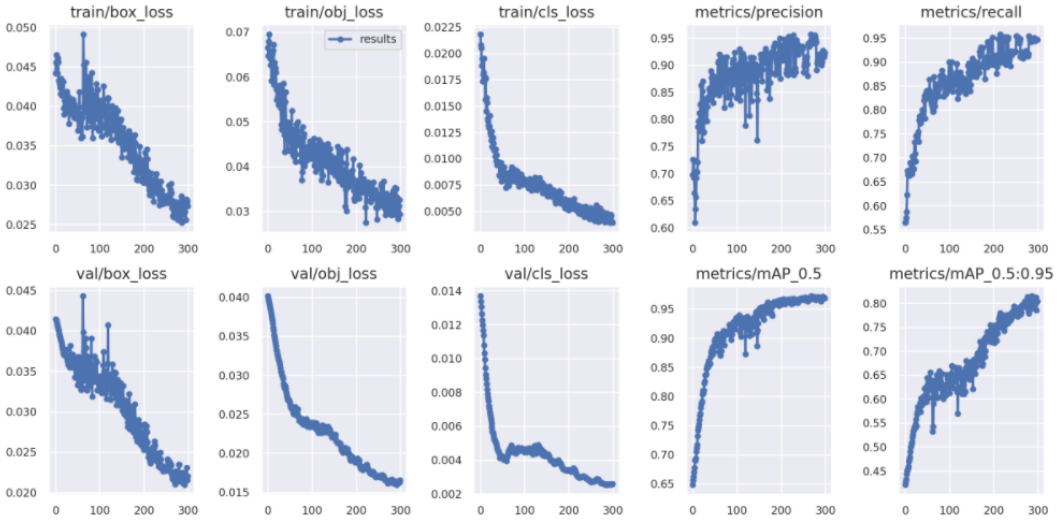
Đầu tiên, ta chọn mô hình đào tạo để train thư viện. Ở đây, nhóm em chọn YOLOv5s, bởi mô hình mạng này so với các mạng lớn hơn thì tốc độ đào tạo sẽ nhanh và inference sẽ cao hơn. Nhóm đã đào tạo mô hình YOLOv5s trên COCO128 bằng cách chỉ định tập dữ liệu, kích thước lô, kích thước hình ảnh. Ta nhập lệnh sau trên Google Colab để thực thi:

!python train.py --img 640 --batch 16 --epochs 300 --data coco128.yaml --weights yolov5s.pt

Ở đây ta sẽ huấn luyện với ảnh kích thước nhỏ hơn hoặc bằng 640x640 pixels, mỗi lần sẽ train 16 hình ảnh với số lần train là 300 lần, mô hình huấn luyện YOLOv5s. Sau đó, các file đào tạo sẽ được lưu trong thư mục ***runs/train*** với các thư mục tự động được đặt tăng dần cho mỗi lần chạy:



Kết quả được thể hiện như hình bên dưới:



Hình 3: Kết quả đào tạo

Dựa theo kết quả biểu đồ, các loss có xu hướng giảm, chỉ số mAP có xu hướng tăng lên. Điều này cho thấy rằng kết quả train đã gần đạt lý tưởng.

2. Phát hiện đối tượng

Sau khi hoàn thành các bước nêu trên, tiếp theo ta sẽ đến bước phát hiện đối tượng.

Mục tiêu của phát hiện đối tượng là có thể phát hiện và đóng khung các đối tượng cần phát hiện trong luồng video trực tiếp. Các tác giả trong [1] đã trình bày chi tiết cách để sử dụng mô hình YOLOv5s đã được đào tạo từ trước từ PyTorch hub và sử dụng để suy luận với 1 hình ảnh. Trong dự án này, chúng em sử dụng 'yolov5s' để phát hiện mục tiêu do nó là mô hình nhanh nhất và nhẹ nhất. Tất nhiên nó sẽ hi sinh về độ chính xác so với các mô hình còn lại.

```
import torch

# Model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')

# Image
img = 'https://ultralytics.com/images/zidane.jpg'

# Inference
results = model(img)

results.pandas().xyxy[0]
#      xmin    ymin    xmax   ymax  confidence  class    name
# 0  749.50   43.50  1148.0  704.5    0.874023      0  person
# 1  433.50  433.50   517.5  714.5    0.687988     27     tie
# 2  114.75  195.75  1095.0  708.0    0.624512      0  person
# 3  986.00  304.00  1028.0  420.0    0.286865     27     tie
```

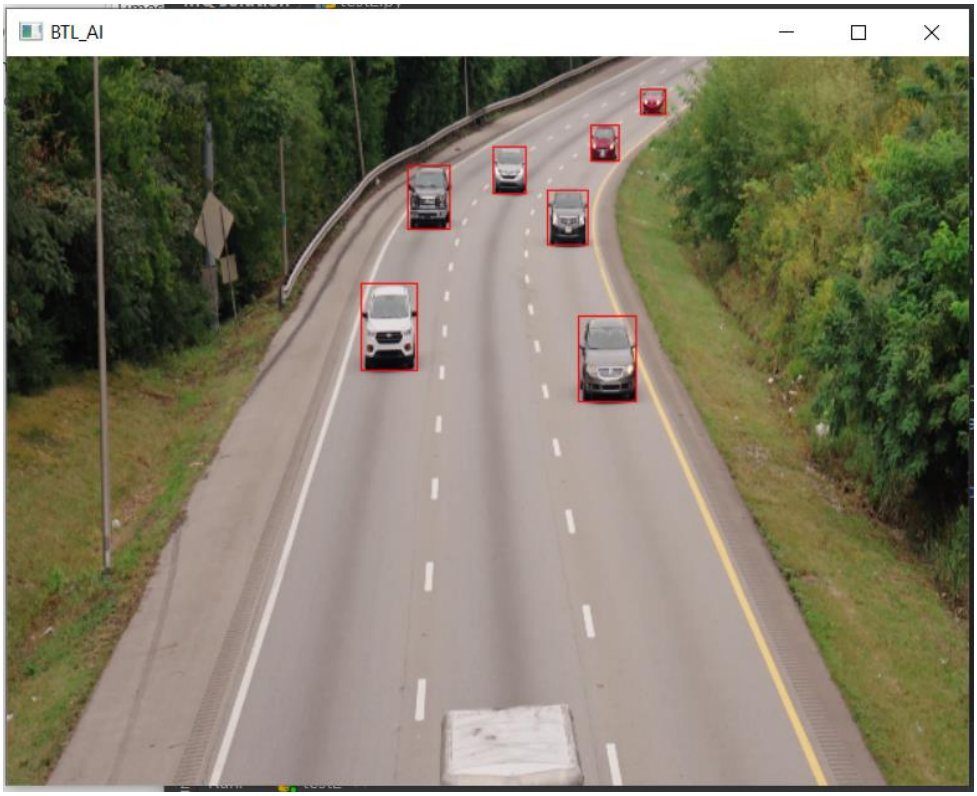
Hình 4: Mã triển khai suy luận của mô hình

Khi đưa đối tượng (là 1 hình ảnh) qua model, giá trị trả về gồm:

- Các giá trị tọa độ của bounding box
- Tỷ lệ chính xác khi nhận diện của đối tượng
- Lớp của đối tượng
- Tên lớp của đối tượng

Với yêu cầu phát hiện của dự án là nhận dạng trong miền video liên tục cùng với các đầu ra như đã kể trên, phương án giải quyết đơn giản là áp dụng mô hình cho 1 vòng

lặp vô hạn (giả sử video là liên tục và không dừng). Ta sẽ có kết quả nhận được như hình:



Hình 5: Nhận dạng đối tượng trong luồng video liên tục với mô hình huấn luyện YOLOv5s

Do là mô hình đào tạo trước, do đó, độ chính xác của quá trình phát hiện phụ thuộc rất nhiều vào quá trình huấn luyện trước đó. Nếu phải nhận dạng một đối tượng đặc biệt, không có sẵn trong tệp huấn luyện mặc định, khi đó mới nên đào tạo một mô hình mới. Nếu đối tượng cần phát hiện có sẵn trong mô hình đào tạo mặc định của các tác giả trong [1], ta nên dùng luôn thay vì đào tạo một mô hình mới kém chính xác hơn.

Lý do, vì khi đào tạo 1 đối tượng mới, bạn nên đào tạo thêm 1 vài đối tượng ở môi trường xung quanh để so sánh. Khi đó, kết quả phát hiện không chỉ có kết quả phát hiện đối tượng đó mà còn bao gồm kết quả so sánh với đối tượng khác nhưng gần giống về các đặc điểm nhận dạng. Một ví dụ là xe hơi và xe bán tải, xe cắm trại,…

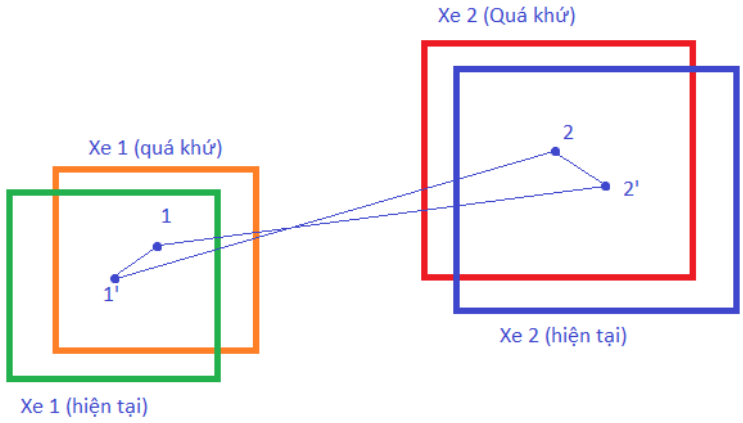
CHƯƠNG 2: THEO DÕI PHƯƠNG TIỆN

Trong phần này, mục tiêu tiếp theo là theo dõi được các đối tượng. Nọi dung sẽ gồm 2 phần chính là: Ý tưởng thực hiện và triển khai.

2.1. Ý tưởng thực hiện

Với yêu cầu theo dõi các phương tiện di chuyển trên đường, nhiệm vụ cần hoàn thành là gán 1 id duy nhất cho từng phương tiện và id đó sẽ đại diện cho phương tiện đó.

Với đặc thù là các phương tiện sẽ di chuyển một quảng đường nhất định trong khoảng thời gian giữa các khung hình và khoảng cách này chính là mấu chốt để xác định xem xe số 1 của khung hình trước là xe nào trong khung hình sau để gán nhãn. Và vì vậy, chúng ta cần xác định điều này.



Hình 6: Ý tưởng triển khai theo dõi

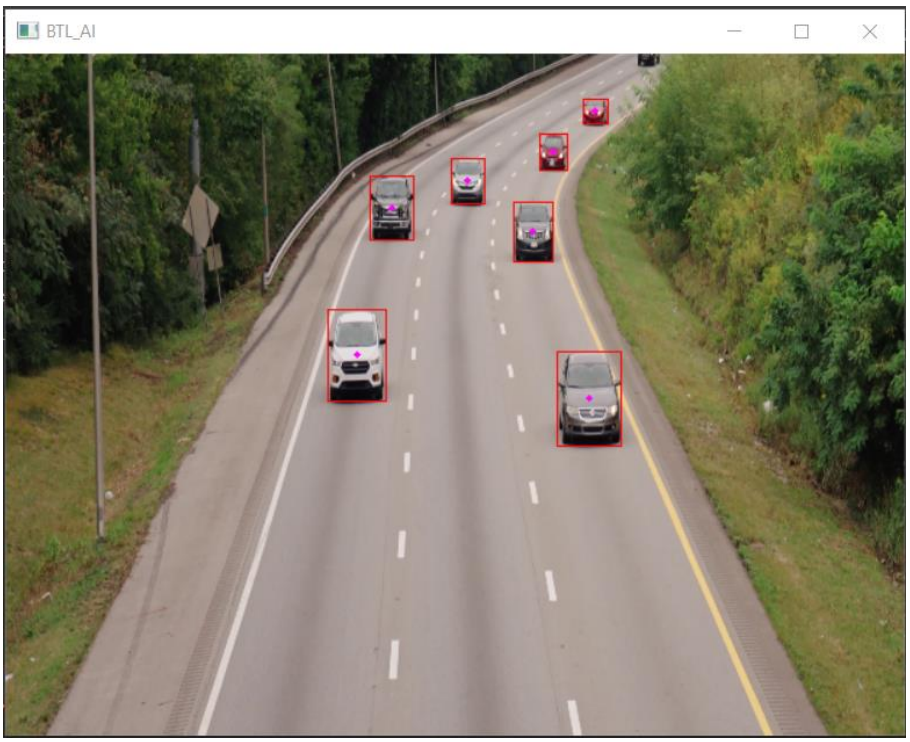
Trong hình trên, ta có thể thấy ý tưởng triển khai 1 tracking với 2 đối tượng có bounding box như hình.

Việc đầu tiên là điểm xét, chúng ta có thể lựa chọn điểm tâm hoặc bất cứ điểm nào (ví dụ như 4 điểm góc) để làm căn cứ. Có thể thấy, tại thời điểm quá khứ, 2 xe có các điểm tâm 1 và 2. Ta có thể lưu nó vào mảng để thuận tiện cho việc sử lý. Tiếp theo, tại thời điểm hiện tại, 2 xe có các tọa độ điểm tâm là 1’ và 2’. Và có thể dễ dàng thấy rằng, khoảng cách giữa các điểm tâm hiện tại và quá khứ sẽ chứng minh xem xe nào là xe 1 và xe nào là xe 2.

Việc tiếp theo đơn giản chỉ là tạo ra 1 list danh sách các xe có nội dung là ID và tọa độ điểm trung tâm của xe ứng với từng ID đó.

2.2. Triển khai thử nghiệm

Trong phần code phát hiện với mô hình huấn luyện trước YOLOv5, ta có kết quả đầu ra là gồm các tọa độ của bounding box, tỷ lệ cùng class. Từ đó, ta dễ dàng xác định được các tọa độ trung tâm của đối tượng.



Hình 7: Xác định tọa độ tâm điểm

Sau khi đã xác định được tọa độ điểm tâm, điều tiếp theo cần thực hiện là lưu các giá trị phù hợp vào các mảng lưu trữ các giá trị này. Giả sử ta lấy các giá trị tâm tại thời điểm hiện tại lưu vào mảng center_point_cur và các giá trị tâm trước đó 1 khung hình lưu vào mảng center_point_pre. Ta sẽ có kết quả như hình dưới:

```
Center point current:
[(406, 250), (375, 70), (317, 91), (365, 128), (239, 217), (403, 41), (264, 109)]
Center point present:
[(406, 250), (375, 70), (317, 91), (365, 128), (239, 217), (403, 41), (264, 109)]
=================
Center point current:
[(408, 255), (316, 93), (374, 71), (365, 129), (402, 43), (264, 111), (240, 221)]
Center point present:
[(406, 250), (375, 70), (317, 91), (365, 128), (239, 217), (403, 41), (264, 109)]
=================
Center point current:
[(411, 260), (316, 94), (374, 73), (365, 131), (239, 225), (400, 44), (264, 113)]
Center point present:
[(411, 260), (316, 94), (374, 73), (365, 131), (239, 225), (400, 44), (264, 113)]
=================
Center point current:
[(412, 265), (373, 73), (366, 133), (263, 114), (315, 95), (238, 229), (400, 44)]
Center point present:
[(411, 260), (316, 94), (374, 73), (365, 131), (239, 225), (400, 44), (264, 113)]
=================
Center point current:
[(412, 265), (373, 73), (366, 133), (263, 114), (315, 95), (238, 229), (400, 44)]
Center point present:
[(412, 265), (373, 73), (366, 133), (263, 114), (315, 95), (238, 229), (400, 44)]
=================
```

Hình 8: Kết quả thu được

Khi đã có các giá trị tại tâm, bước tiếp theo, ta cần tính toán khoảng cách giữa các điểm trung tâm ở hiện tại và quá khứ. Sau đó, ta sẽ gán id cho các phương tiện. Ta cần 1 danh sách để lưu trữ điều này. Khi thực hiện thành công, kết quả thu được sẽ như sau:

```
tracking list:
{0: (325, 71), 1: (384, 188), 2: (247, 168), 3: (360, 101), 4: (386, 53), 5: (418, 27), 6: (273, 87)}
=================
```

Hình 9: Tracking list thu được

Hình 10: Kết quả tracking tại 2 thời điểm khác nhau

CHƯƠNG 3: XÁC ĐỊNH VẬN TỐC PHƯƠNG TIỆN

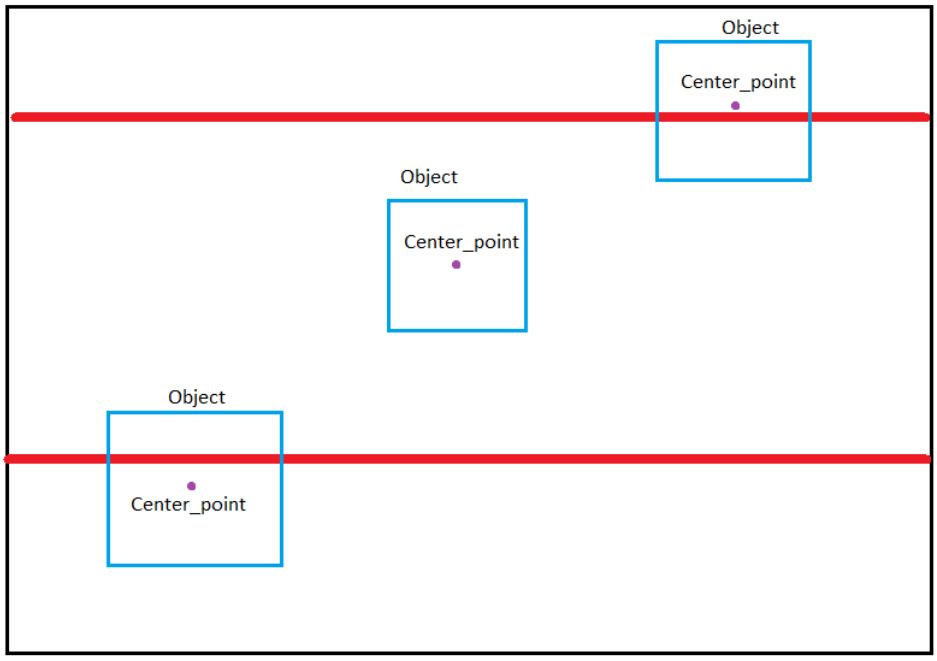
Sau khi đã theo dõi và đánh dấu được các đối tượng, việc tiếp theo cần thực hiện là xác định vận tốc của đối tượng. Các bước cần thực hiện sẽ tương tự phần trên với 2 bước: lên ý tưởng và triển khai thử nghiệm.

3.1. Lên ý tưởng

Xuất phát từ công thức cơ bản là: v = S/t với S là quảng đường đi được trong 1 khoảng thời gian t. Nhiệm vụ cần hoàn thành là xác định quảng đường và thời gian.

Cụ thể trong bài toán này, ta hoàn toàn có thể cố định 1 quãng đường S cho trước với tất cả các phương tiện di chuyển.

Sau khi đã có được quãng đường, vấn đề tiếp theo là xác định thời gian t. Ta có thể có 1 ý tưởng đơn giản như sau:

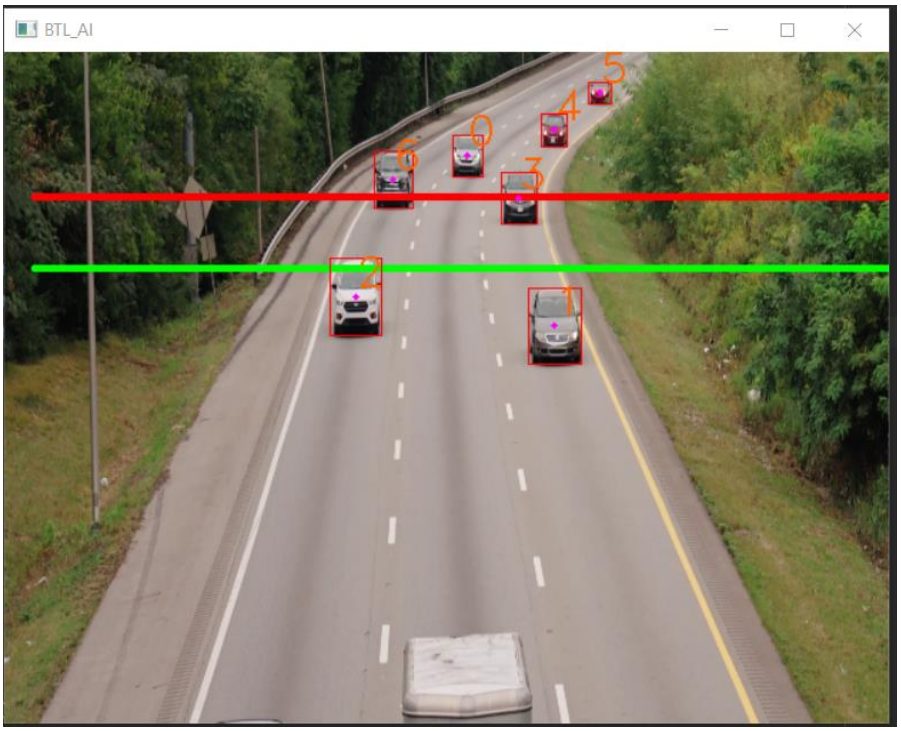


Hình 11: Ý tưởng thực hiện

Qua hình vẽ mô tả trên, ta có thể sử dụng lại tracking_list đã có từ phần trước để xác định xem thời điểm xe bắt đầu quãng đường và thời điểm xe kết thúc quãng đường. Từ đó, dễ dàng xác định thời gian xe đi hết quãng đường S cho trước.

3.2. Triển khai thử nghiệm

Đầu tiên, cần xác định quảng đường S cho trước. Công việc này khá đơn giản, Chỉ cần vẽ 2 đường thẳng căng ngang tượng trưng cho điểm đầu và điểm cuối của con đường. Trong phần này, chúng ta nên chọn khu vực có độ tin tưởng cao vì nó sẽ góp phần làm tăng độ chính xác. Và hình dưới đây là kết quả thu được:



Hình 12: Xác định quãng đường S

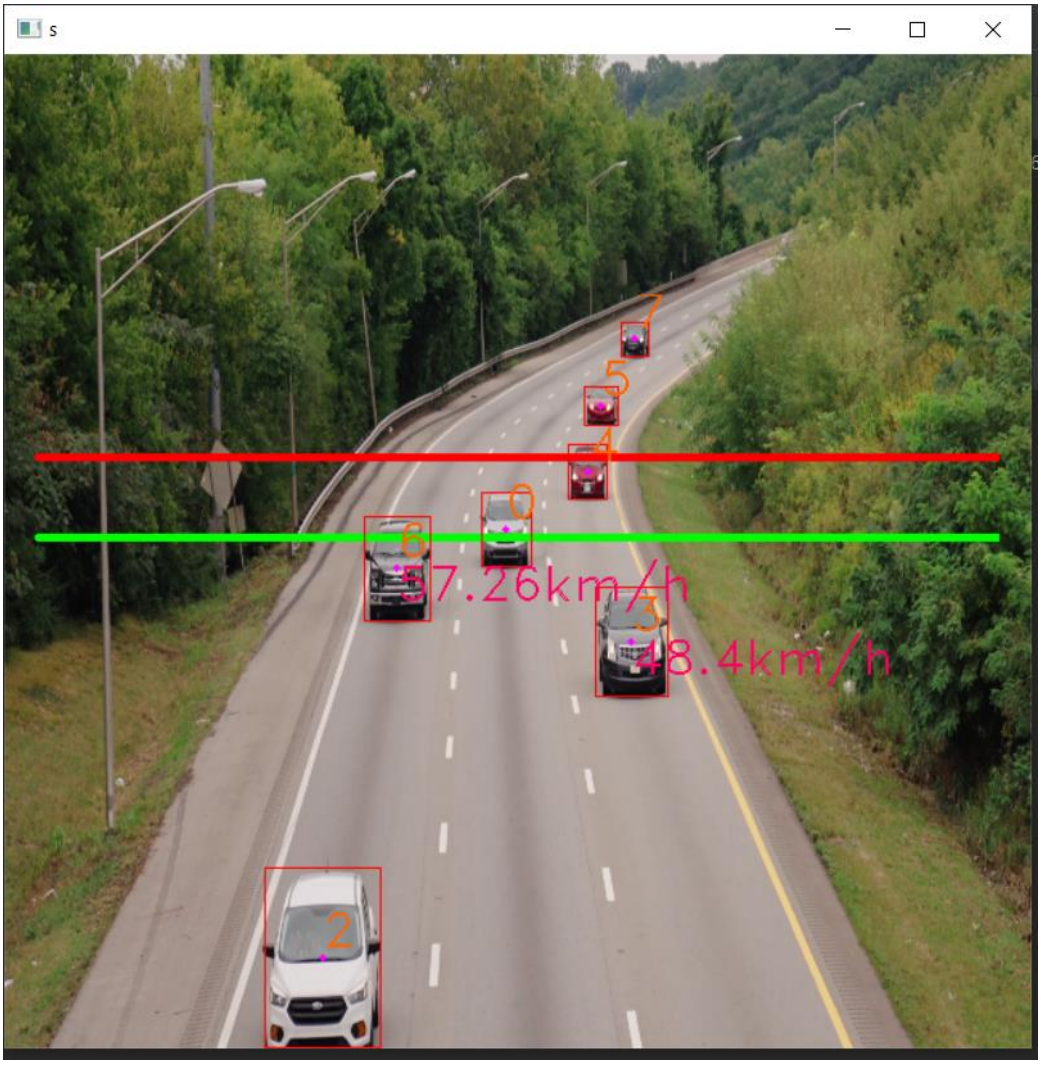
Tiếp theo là nhiệm vụ xác định thời gian t để phương tiện đi hết quãng đường trên. Để làm điều này, ta cần xác định thời gian lúc xe tới điểm đầu và thời gian xe tới điểm cuối và lưu nó vào 1 danh sách. Công việc này theo 1 khía cạnh nào đó khá giống với việc tracking. Vì vậy, tương tự như khi xử lý với tracking, ta sẽ có kết quả sau:

```
time start:
{3: 1641306813.3787856, 6: 1641306816.5356553, 0: 1641306821.4587862, 4: 1641306828.987536}
time end:
{3: 1641306822.9245987, 6: 1641306827.8157473}
-----
{3: 9.545813083648682, 6: 11.280092000961304}
```

Hình 13: Các mảng lưu trữ thời gian

Có thể thấy rằng, các xe 3,6,0,4 đã đi qua điểm xuất phát và thời gian chũng đi qua đã được lưu lại vào mảng thời gian xuất phát (tất nhiên là giá trị trong quá khứ). Và ngay tại lúc này, chỉ có xe 3 và xe 6 là đã đi qua điểm cuối của đoạn đường S và chúng mất lần lượt là 9.5s và 11.28s cho 2 xe. Các xe 0 và 4 vẫn còn trong hành trình nên chưa có giá trị thời gian t cho chúng.

Và sau khi đã có giá trị S và t, việc còn lại chỉ là duyệt danh sách lưu trữ giá trị thời gian, áp dụng công thức tính vận tốc và in ra màn hình kết quả tính toán được. Nếu không có bất kỳ lỗi gì, kết quả sẽ tương tự như hình dưới:



Hình 14: Bài toán xác định vận tốc

CHƯƠNG IV: TỔNG KẾT

Qua quá trình thực hiện đề tài, bọn em đã có một số gặt hái như sau:

- Tiếp cận với các công nghệ, công cụ tiên tiến trong xử lý ảnh.
- Làm quen với môi trường lập trình xử lý ảnh.
- Có khả năng tự huấn luyện một mô hình phát hiện cho riêng mình.
- Thực hiện mô phỏng và chạy thử nghiệm thành công đề tài với môi trường python.

Trong quá trình thực hiện đề tài, do hạn chế về hiểu biết cũng như kỹ năng nên bọn em đã gặp phải một số vấn đề khó giải quyết. Rất may mắn có sự giúp đỡ tận tình của các anh chị trong công ty giúp bọn em giải quyết vấn đề. Qua đó, chúng em cũng hiểu hơn về văn hóa làm việc ở các doanh nghiệp, trau dồi thêm về kỹ năng mềm và kỹ năng chuyên môn.

TÀI LIỆU THAM KHẢO

[1]. https://github.com/ultralytics/yolov5

[2]. https://www.kaggle.com/learn/computer-vision

[3]. https://realpython.com/tutorials/computer-vision/

[4]. Real Time Object Detection with Audio Feedback using Yolo - Mansi Mahendru; Sanjay Kumar Dubey - https://ieeexplore.ieee.org/document/9377064

[5]. Vehicle Tracking based on an Improved DeepSORT Algorithm and the YOLOv4 Framework - Imalie Perera; Shehan Senavirathna; Aseni Jayarathne; Shamendra Egodawela; Roshan Godaliyadda; Parakrama Ekanayake; Janaka Wijayakulasooriya; Vijitha Herath; Sathindra Sathyaprasad (https://ieeexplore.ieee.org/document/9606052)

[6]. Object Tracking in a Zone using DeepSORT, YOLOv4 and TensorFlow - Rajni Jindal; Aditya Panwar; Nishant Sharma; Aman Rai

(https://ieeexplore.ieee.org/document/9456443)