

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KÌ
Nhận diện biến báo giao thông

Môn học:

Máy học

Giảng viên:

PGS. TS Lê Đình Duy
ThS. Phạm Nguyễn Trường An
CS114.N21.KHCL

Mã lớp:

Thành viên nhóm:

Ngô Trần Tuấn Anh – 21520567
Phan Thanh Quang – 21521350
Nguyễn Minh Hiếu – 21520853
Nguyễn Hoàng Nguyên - 21521203

Contents

Chương 1. Tổng quan đồ án.....	3
1.1. Ngữ cảnh ứng dụng	3
1.2. Input và Output.	3
Chương 2. Thu Thập bộ dữ liệu.	3
2.1. Thu Thập dữ liệu.....	3
2.1.1. Thông tin thu thập dữ liệu.....	3
2.1.2. Công cụ gán label cho dữ liệu.....	3
2.1.3. Kết quả thu thập dữ liệu.....	5
2.2. Xử lý dữ liệu.....	13
Chương 3. Training và đánh giá model.....	14
3.1. Training.....	14
3.1.1. Lựa chọn mô hình.....	14
3.1.2. Các bước tiến hành train.....	14
3.1.3. YOLOv5:	14
3.2. Đánh giá.....	14
3.2.1. mAP - mean Average Precision for Object Detection.....	14
3.2.2. Đánh giá model.....	17
Chương 4. Demo mô hình.....	28
Chương 5. Nguồn tham khảo.....	29

Link Github nhóm:

<https://github.com/imqp/CS114.N21.KHCL?fbclid=IwAR3-mg940UwD0A0rdlIzBsVbqxjsfpO9NmjwoVTUHS9Jui857BOnnUHYiIM>

Chương 1. Tổng quan đề án.

1.1. Ngữ cảnh ứng dụng.

- Hiện nay, xe tự hành đang phát triển rất nhanh, và để có thể vận hành được trên đường phố thì ngoài việc nhận diện được con người và các vận thể thì nó cần phải nhận diện đc các biển báo giao thông để có thể tham gia giao thông đúng luật hiện hành. Vì vậy, bài toán Nhận diện biển báo giao thông ở Việt Nam là rất cần thiết, nên chúng em đã chọn đề tài này.

1.2. Input và Output.

- Input:
 - Video được quay từ điện thoại.
- Output:
 - Video chứa các thông tin:
 - Khung bao quanh biển báo.
 - Tên biển báo.

Chương 2. Thu Thập bộ dữ liệu.

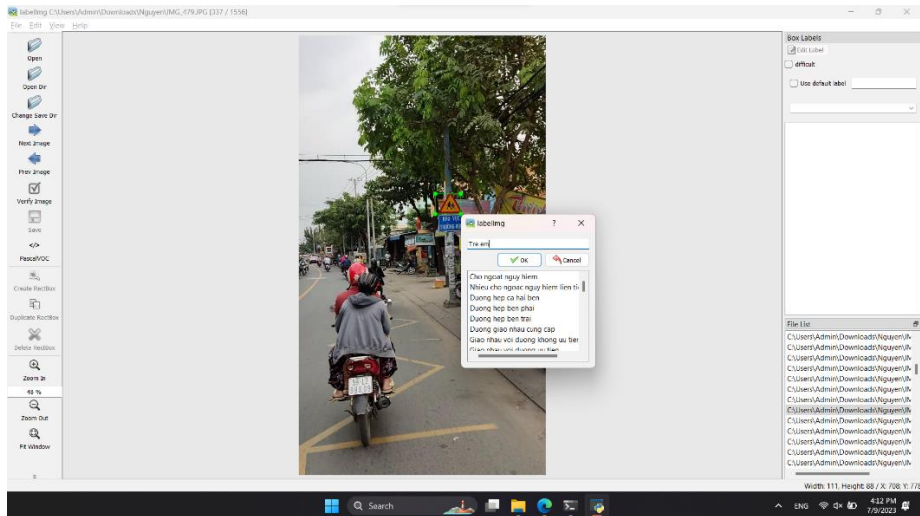
2.1. Thu Thập dữ liệu.

2.1.1. Thông tin thu thập dữ liệu.

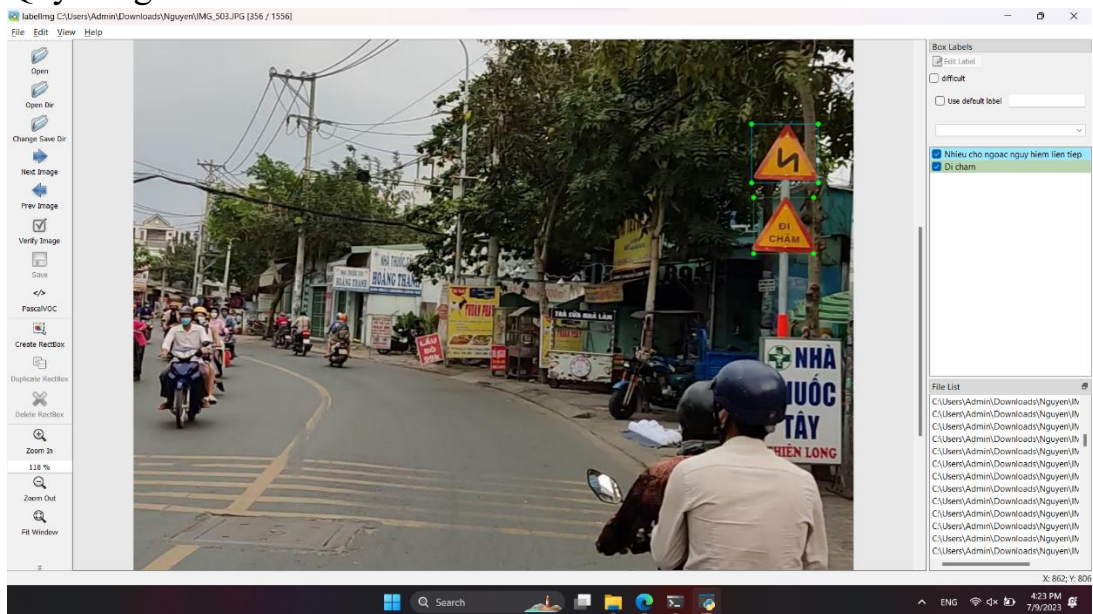
- Cách thức thu thập: Sử dụng điện thoại quay video trong quá trình chạy xe. Sau đó cắt những ảnh trong video ra để làm data.
- Ảnh được chụp từ thiết bị điện thoại.
- Video được quay vào ban ngày và buổi tối.

2.1.2. Công cụ gán label cho dữ liệu.

- Sử dụng công cụ LabelImg để gán label.
- Lý do: dễ sử dụng, Giao diện khá tốt với đầy đủ chức năng: open, load, autosave,... Định dạng của nhãn phù hợp để dùng YOLO train.



- Quy tắc gán label:



- Label bounding box ôm gọn biển báo, tránh label rộng hơn, hay không label hết phần biển báo.
- Label những biển báo cách vị trí xe từ 0- 20m. Vì khi nhận diện để thông báo cho người tham gia giao thông, ta cần nhận diện và thông báo trước khi đi qua biển báo đó, để người đi điều chỉnh tốc độ hay chú ý hơn.
- Đối với những biển báo bị mất hơn 40% diện tích, thì sẽ bỏ qua. Lí do là ảnh đó có thể đánh mất một số features quan trọng, có thể làm cho model học sai.

- Đối với những biển báo có thời gian, cần label phần biển báo và khoảng thời gian áp dụng lên biển đó.
- Đối với mỗi ảnh sau khi gán nhãn, sẽ tạo ra một file txt, gọi là annotation.







- Với w, h là khoảng cách từ tâm bounding box đến cạnh bên trái và vạnh bên trên của bức ảnh (như trong hình vẽ trên), các thành phần của 1 file annotaion như sau:
 - Mỗi dòng là thông tin của 1 bounding box:
 - id: Thứ tự của class do mình định nghĩa.
 - Center: Là tọa độ tâm của bounding box (X_{center} , Y_{center}). Tính như sau:

$$X_{center} = \frac{w}{width_{image}}, Y_{center} = \frac{h}{height_{image}}$$















2.1.3. Kết quả thu thập dữ liệu.

- Hơn 3000 ảnh, có 102 classes:

ID	Tên biển báo	Hình ảnh	ID	Tên biển báo	Hình ảnh
0	Cho ngoat nguy hiem		51	Het han che toc do toi da	
1	Nhieu cho ngoac nguy hiem lien tiep		52	Het tat ca cac len cam	

2	Duong hep ca hai ben		53	Cam su dung coi	
3	Duong hep ben phai		54	Cam di thang	
4	Duong hep ben trai		55	Cam re trai va re phai	
5	Duong giao nhau cung cap		56	Cam di thang va re trai	
6	Giao nhau voi duong khong uu tien		57	Cam di thang va re phai	
7	Giao nhau voi duong uu tien		58	Chi duoc di thang	
8	Duong khong bang phang		59	Chi duoc re trai	

9	Tre em		60	Chỉ được rẽ phải	
10	Đường người đi bộ cắt ngang		61	Chỉ được đi thang và rẽ phải	
11	Đường người đi xe đạp cắt ngang		62	Chỉ được đi thang và rẽ trái	
12	Công trường		63	Đi vòng sang trái	
13	Đường cáp điện phía trên		64	Đi vòng sang phải	
14	Nơi giao nhau với đường sắt		65	Nơi giao nhau chạy theo vòng xuyến	

15	Đi chậm		66	Duong danh cho người đi bộ	
16	Chú ý xe đạp		67	Duong danh cho oto	
17	Duong cam		68	Duong danh cho oto va moto	
18	Cam đo xe		69	Huong đi theo vạch kẻ đường	
19	Cam dung va đo xe		70	Chỉ đường	
20	Cam đo xe ngay le		71	Cầu vượt qua đường	
21	Cam đo xe ngay chạn		72	Ben xe buýt	

22	Cam di nguoc chieu		73	Cho	
23	Cam oto		74	Bat dau duong uu tien	
24	Cam oto re phai		75	Het doan duong uu tien	
25	Cam oto re trai		76	Gioi han chieu cao 4m	
26	Cam re trai		77	Tu 21h den 6h	

27	Cam re phai			78	Cam xe ba banh va xe tho so bon banh	
28	Cam moto			79	Tu 6h den 22h	
29	Cam oto va moto			80	Han che trong luong xe 10 tan	
30	Cam xe tai			81	Cam xe tai tu 5 tan	
31	Cam xe tai tu 2.5 tan			82	Han che trong luong xe 5 tan	
32	Cam xe tai tu 10 tan			83	Duong mot chieu	

33	Cam xe tai tu 900kg	06a 	84	Toc do toi da 20kph	
34	Cam xe khach va xe tai		85	Gioi han chieu cao 1.8m	
35	Cam di xe dap		86	Gioi han chieu cao 2m	
36	Cam nguoi di bo		87	Han che trong luong xe 2.5 tan	
37	Cu ly toi thieu giua hai xe		88	Tu 6h den 9h va tu 16h den 19h	
38	Stop		89	Han che trong luong xe 25 tan	
39	Cam quay dau xe		90	Gioi han chieu cao 1.7m	

40	Cam oto quay dau xe		91	Gioi han chieu cao 2.4m	
41	Cam re trai va quay dau		92	Han che trong luong xe 30 tan	
42	Cam re phai va quay dau		93	Han che trong luong xe 13 tan	
43	Cam oto re trai va quay dau		94	Tu 6h den 20h	
44	Cam oto re phai va quay dau		95	Benh vien	
45	Toc do toi da 40kph		96	Cam xe hai banh va ba banh	

46	Tốc độ tối đa 50kph		97	Cấm xe tải từ 1 tấn	
47	Tốc độ tối đa 60kph		98	Cấm xe khách	
48	Tốc độ tối đa 70kph		99	Góp lan theo phương tiện	
49	Tốc độ tối đa 80kph		100	Đường đi bộ	
50	Tốc độ tối đa 90kph		101	Cấm xe tải từ 1.5 tấn	

2.2. Xử lý dữ liệu.

Datasets gồm 3000 bức ảnh, chia thành tập train, và validation theo tỉ lệ 4:1.

Chương 3. Training và đánh giá model.

3.1. Training.

3.1.1. Lựa chọn mô hình.

- Dựa vào đặc điểm của bài toán chúng em đặt ra, yếu tố tốc độ nhận diện có vai trò quan trọng hơn so với độ chính xác. Do đó chúng em quyết định dùng YOLO để thực hiện bài toán này. Hai phiên bản chúng em chọn là YOLOv5.

3.1.2. Các bước tiến hành train.

- Môi trường train và đánh giá:
Google colab là một virtual cloud machine được google cung cấp miễn phí cho các nhà nghiên cứu. Đây là môi trường lý tưởng để phát triển các mô hình vừa và nhỏ. Điểm tuyệt vời ở google colab đó là môi trường của nó đã cài sẵn các packages machine learning và frame works deep learning thông dụng nhất.
- Chúng em lưu trữ dữ liệu bài toán trên google drive, sau đó kết nối drive với colab.

3.1.3. YOLOv5:

- o Clone model từ github: <https://github.com/ultralytics/yolov5>
- o Lấy data từ Google drive của nhóm.
- o Train model yolov5s 32 batch 250 epoch.
- o Datasets gồm 3000 bức ảnh, chia thành tập train, và validation theo tỉ lệ 4:1.

3.2. Đánh giá.

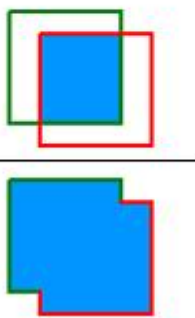
- Trong bài toán này, chúng em quyết định sử dụng mAP để đánh giá về độ chính xác model của mình

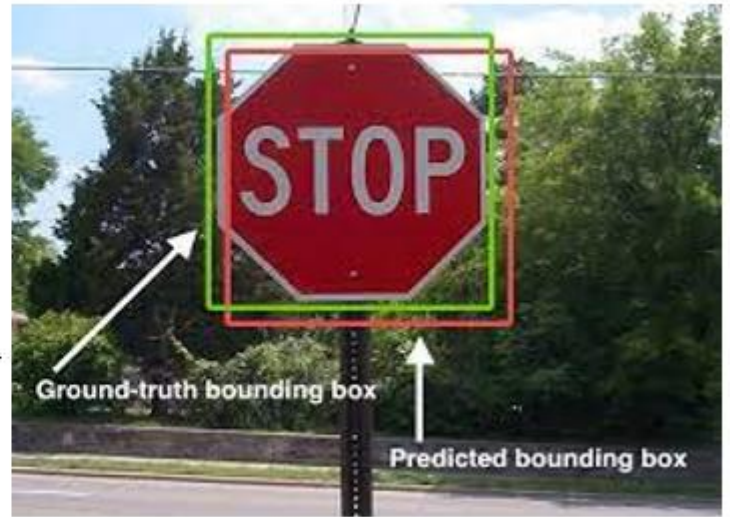
3.2.1. mAP - mean Average Precision for Object Detection.

- **mAP** (mean average precision) là độ đo phổ biến để đánh giá độ chính xác của bài toán object detection như Faster R-CNN, SSD... mAP chính là trung bình của các average precision của từng class. Trước khi tìm hiểu về mAP chúng ta cùng nhắc lại một số khái niệm như precision, recall trong bài toán phân loại.

3.2.1.1. IOU (Intersection over Union).

- IOU là hàm đánh giá độ chính xác của object detector trên tập dữ liệu, cụ

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$




- thể được xác định bởi phép chia:
- Trong đó:
 - o Area of overlap là diện tích phần giao giữa predicted bounding box với ground-truth bounding box.
 - o Area of Union là diện tích phần hợp giữa predicted bounding box với ground-truth bounding box.
 - o Với ground-truth bounding box là do ta xác định (trong lúc label data), predicted bounding box do model xác định.
 - Dựa vào thông số IOU này người ta sẽ xác định TP, FP, TN, FN như sau:
 - o TP (True positive): Khi IOU của predicted box vs gtbox \geq iou threshold
 - o FP (False positive): Khi IOU của predicted box vs gtbox $<$ iou threshold
 - o TN (True negative): Thông số này ta có thể hiểu nó như là background và ta sẽ không cần quan tâm thông số này.
 - o FN (False negative): Bouding box của đối tượng không được detect (detect sót)
 - Và sau đó precision và recall vẫn được tính theo công thức:

$$precision = \frac{TP}{TP + FP} = \frac{TP}{\text{Tổng số dự đoán}}$$

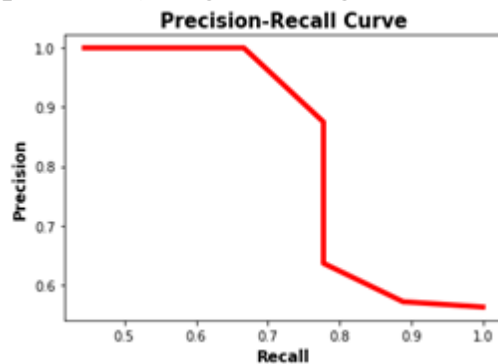
$$recall = \frac{TP}{TP + FN} = \frac{TP}{\text{Tổng số gtbox}}$$

Trong đó:

- Precision - độ tin cậy của model, cho biết bao nhiêu % dự đoán Positive là True Positive
- Recall - độ nhạy của model cho biết model có thể đoán đúng được bao nhiêu Positive trong dữ liệu được cho.

3.2.1.2. Precision Recall Curve và Average precision (AP).

- Precision và Recall thay đổi với mỗi Confidence threshold. Để quan sát tất cả các precision và recall tương ứng các threshold ta sử dụng Precision Recall Curve – đường đi qua tất cả các điểm với giá trị (recall, precision) ứng với từng threshold.



- AUC (Area Under the Curve) - diện tích nằm dưới Curve giúp đánh giá model. Với Precision Recall Curve, Area Under the Curve (AUC) còn được gọi là Average precision (AP). AP được xác định bởi công thức:

$$AP = \sum_{k=1}^n (R_k - R_{k-1}) * P_k$$

- R_k, P_k lần lượt là Recall và Precision ứng với threshold thứ k
- n : số threshold

- AP lớn nếu vùng AUC này lớn, suy ra đường cong có xu hướng gần góc trên bên phải và có nghĩa là tại các threshold khác nhau thì Precision và Recall đều khá cao. Từ đó suy ra model tốt.
- AP nhỏ thì cả Precision và Recall đều khá thấp và model không tốt.

3.2.1.3. mAP.

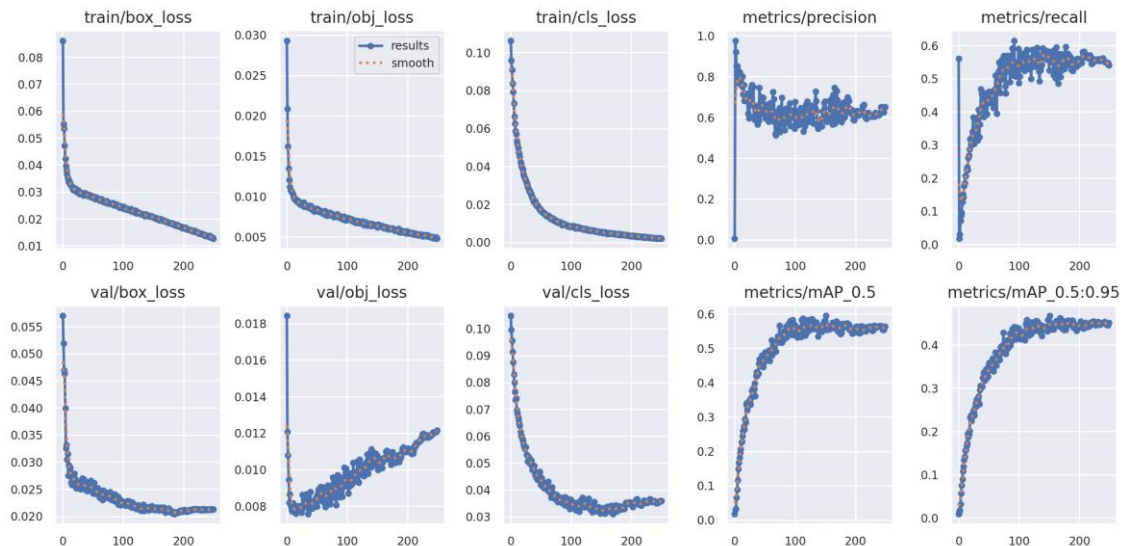
- Trong bài toán Object Detection nói chung hay YOLO nói riêng thì mAP được định nghĩa là trung bình cộng giá trị AP của tất cả các class. Trong đó:

$$mAP = \frac{1}{n} \sum_{i \in C} AP(i)$$

- C là tập hợp tất cả các class
- n là số class

3.2.2. Đánh giá model.

- YOLOv5: sau khi train 250 epochs, chúng em được kết quả:



- Phân tích cụ thể trên YOLOv5:

Validating runs/train/exp7/weights/best.pt...
Fusing layers...

Model summary: 157 layers, 7285219 parameters, 0 gradients, 16.6 GFLOPs

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%		11/11 [00:08<00:00, 1.24it/s]
all	662	1024	0.61	0.57	0.595	0.467		
Cho ngoat nguy hiem	662	33	0.707	0.803	0.858	0.709		
Nhieu cho ngoat nguy hiem lien tiep	662	12	0.301	0.833	0.691	0.545		
Duong hep ca hai ben	662	3	1	0	0	0		
Duong hep ben phai	662	6	0.674	0.5	0.762	0.566		
Duong hep ben trai	662	7	0.223	0.143	0.237	0.204		
Duong giao nhau cung cap	662	20	0.358	0.446	0.341	0.248		
Giao nhau voi duong khong uu tien	662	28	0.577	0.893	0.874	0.607		
Giao nhau voi duong uu tien	662	15	1	0.447	0.638	0.454		
Tre em	662	15	0.365	0.733	0.526	0.402		
Duong nguoi di bo cat ngang	662	29	0.504	0.897	0.738	0.608		
Duong nguoi di xe dap cat ngang	662	3	1	0	0	0		
Duong cap dien phia tren	662	11	0.79	0.636	0.722	0.553		
Di cham	662	8	0.404	0.625	0.43	0.345		
Cam do xe	662	83	0.559	0.964	0.905	0.764		
Cam dung va do xe	662	96	0.883	0.729	0.816	0.635		
Cam do xe ngay le	662	54	0.851	0.637	0.888	0.611		
Cam do xe ngay chan	662	11	0.162	0.636	0.306	0.227		
Cam di nguoc chieu	662	48	0.715	0.896	0.865	0.632		
Cam oto	662	25	0.692	0.36	0.531	0.356		
Cam oto re phai	662	4	0.676	0.75	0.757	0.648		
Cam oto re trai	662	11	0.295	0.455	0.5	0.437		
Cam re trai	662	9	0.588	1	0.973	0.864		
Cam re phai	662	9	0.527	1	0.907	0.749		
Cam moto	662	14	0.663	0.282	0.406	0.305		
Cam xe tai	662	33	0.808	0.697	0.836	0.679		
Cam xe tai tu 2.5 tan	662	2	0	0	0.0262	0.0236		
Cam xe khach va xe tai	662	18	0.431	1	0.911	0.776		
Cam nguoi di bo	662	6	1	0.407	0.955	0.86		
Cam quay dau xe	662	23	0.823	0.87	0.911	0.712		
Toc do toi da 40kph	662	24	0.564	0.833	0.819	0.712		
Toc do toi da 50kph	662	48	0.395	0.125	0.0814	0.0616		
Toc do toi da 60kph	662	20	0.668	1	0.973	0.792		
Toc do toi da 80kph	662	18	0.75	1	0.995	0.824		
Het tat ca cac len cam	662	7	0.927	0.571	0.885	0.757		
Chi duoc re trai	662	9	0.542	0.111	0.17	0.139		
Chi duoc re phai	662	3	0.293	0.667	0.373	0.282		
Di vong sang trai	662	3	0	0	0.0119	0.0108		
Di vong sang phai	662	27	0.443	0.778	0.738	0.578		
Noi giao nhau chay theo vong xuyen	662	14	0.745	0.786	0.856	0.685		
Huong di theo vach ke duong	662	33	0.566	0.395	0.532	0.383		
Chi duong	662	10	0.0532	0.1	0.114	0.101		
Ben xe buyt	662	25	0.8	0.6	0.618	0.451		
Gioi han chieu cao 4m	662	9	1	0	0.00591	0.00472		
Cam xe ba banh va xe tho so bon banh	662	1	1	0	0	0		
Tu 6h den 22h	662	31	0.0948	0.226	0.103	0.0673		
Han che trong luong xe 10 tan	662	15	0.557	0.838	0.861	0.737		
Cam xe tai tu 5 tan	662	3	1	0	0.0378	0.0345		
Duong mot chieu	662	16	0.782	0.875	0.938	0.804		
Han che trong luong xe 30 tan	662	4	1	0	0.796	0.725		
Tu 6h den 20h	662	1	0.375	1	0.995	0.298		
Cam xe hai banh va ba banh	662	29	0.445	1	0.593	0.478		
Cam xe khach	662	15	0.896	1	0.987	0.759		
Gop lan theo phuong tien	662	20	0.486	0.568	0.674	0.539		
Duong di bo	662	3	0.997	0.667	0.666	0.5		

- Ảnh biển báo “Chi duong” bị sai, đúng thì nó phải là biển “Duong mot chieu”. Lí do là vì 2 biển này đều có hình vuông nền màu xanh dương, và có mũi tên nên bị nhầm lẫn, dẫn đến kết quả bị sai.



Hình: Nhầm lẫn giữa biển “Chi duong” và biển “Duong mot chieu”

- 2 Biển này đều có hình vuông và phông xanh kèm mũi tên:



- Không nhận ra được biển báo trong hình. Do biển “Cam oto re trai” bị mất khá nhiều chi tiết, còn biển “Tu 6h den 22h” bị méo và do ít data về hình này nên cũng không dự đoán được.



Hình: Không dự đoán được biển trong hình.

- Nhận diện đúng biển “Cam dung va do xe”, “Huong di theo vach ke duong”, “Chi duong 0.34”.



Hình: Nhận diện đúng.

- Tuy ảnh rất nhỏ và mờ nhưng vẫn nhận diện đúng biển “Cam dung va do xe” vì đã train rất nhiều ảnh có class “Cam dung va do xe”



Hình: Nhận diện đúng biển “Cam dung va do xe”.

- Nhận diện nhầm giữa biển “Tre em” và biển “Nhiều cho goats nguy hiem lien tiep”



Hình: Nhận diện nhầm biển “Tre em”.



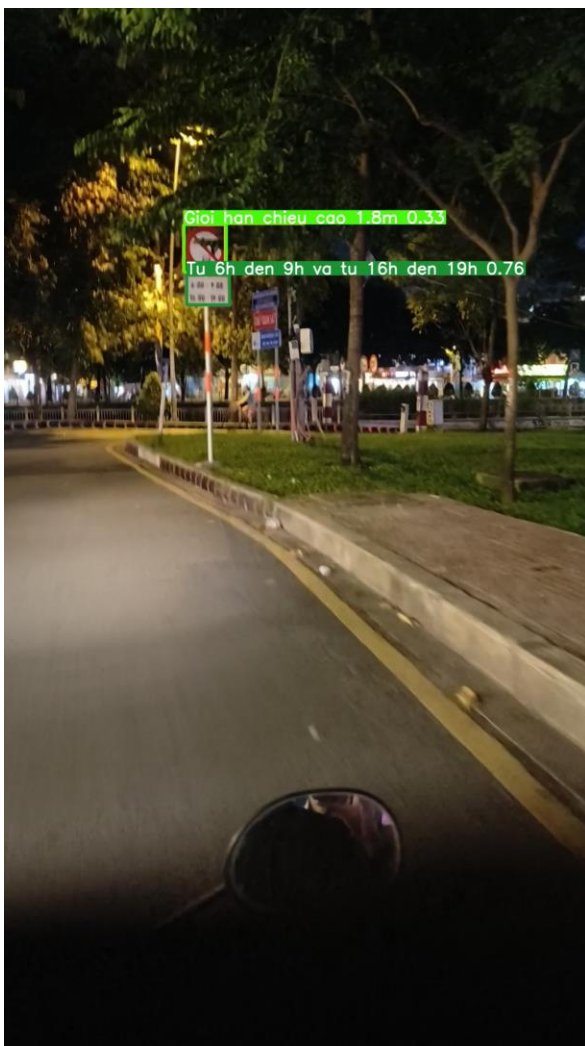
Do 2 biển này có sự giống nhau, đều là biển tam giác nền vàng biển đỏ, đặc biệt sự uốn lượn nét hình gần giống nhau nên bị nhầm lẫn.

- Nhận diện nhầm biển “Duong nguoi di bo cat ngang” với biển “Di cham” do ảnh bị mờ.



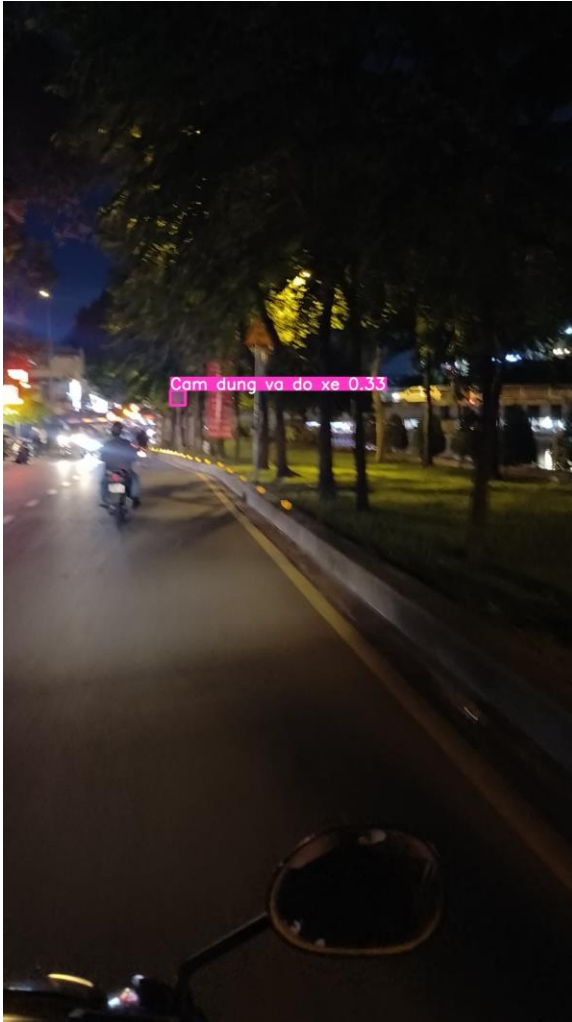
Hình: Nhận diện nhầm biển “Di cham”.

- Nhận diện nhầm biển ”Cam oto re trai” thành “Gioi han chieu cao 1.8m” do ảnh bị cây che khuất.



Hình: Nhận diện sai biển “Cam oto re trai”.

- Nhận diện đúng biển “Cam dung va do xe”.



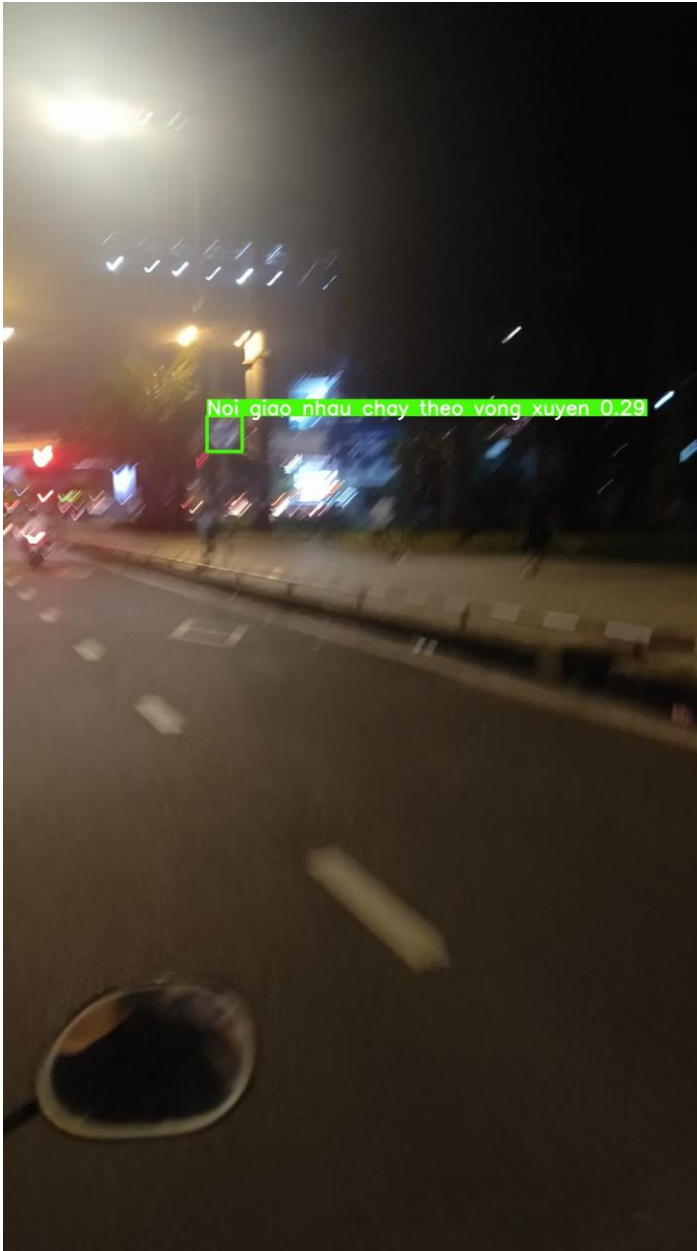
Hình: Nhận diện đúng biển “Cam dung va do xe”.

- Nhận diện đúng biển “Ben xe buyt”, dù bị mờ nhưng vẫn nhận diện đúng do ảnh này khá đặc trưng và có nhiều dữ liệu để train.



Hình: Nhận diện đúng biển “Ben xe buyt”

- Nhận diện đúng biển “Nơi giao nhau chạy theo vòng xuyến”, tuy ảnh hơi mờ nhưng do ảnh khá đặc trưng và train nhiều nên đã nhận diện đúng.



Hình: Nhận diện đúng biển “Nơi giao nhau chạy theo vòng xuyến”.

- Không nhận diện được biển “Cam oto rẽ trái”, do ảnh quá xa, mờ và bị cây che khuất.



Hình: Không nhận diện được biển “Cam oto re trai”.

➔ Nguyên nhân và cách khắc phục:

- Nguyên nhân: là do có nhiều class nên ảnh bị phân ra nhiều, do đó mỗi class có ít ảnh, có class chỉ có 1 ảnh.
- Khắc phục: Thu thập thêm nhiều data ứng với mỗi class để mô hình nhận ra được đặc trưng và học được tốt hơn, từ đó đưa ra kết quả tốt hơn.

Chương 4. Demo mô hình.

Link colab:

<https://colab.research.google.com/drive/10N-GM1NKw1bkk4m6zKoNjuBm-Kg1R-T#scrollTo=Mrmr-bv947bj>

Chương 5. Nguồn tham khảo.

- **Tìm hiểu về yolov5**

[1]<https://joaotavionf007.medium.com/orange-trees-detection-with-yolo-v5-in-uav-imagery-22ec29db922e>

[2]<https://github.com/ultralytics/yolov5>

- **Tìm hiểu về thang đo đánh giá:**

[1]https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html

[2]https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

[3]<https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/>

- **Tìm hiểu về biển báo giao thông:**

[1] <https://anycar.vn/cac-loai-bien-bao-giao-thong-duong-bo-ban-can-biet-t58043.html>

[2] <https://motogo.vn/bien-bao-giao-thong/>