

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

---□&□---



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN:

BÀI TOÁN NHẬN DIỆN BIỂN BÁO GIAO THÔNG

Giảng viên hướng dẫn:

PGS.TS Lê Đình Duy

ThS. Phạm Nguyễn Trường An

Sinh viên thực hiện :

| STT | Họ và Tên | MSSV |
|------------|---------------------|-------------|
| 1 | Đỗ Vũ Gia Cần | 19521271 |
| 2 | Nguyễn Thị Như Ý | 19522555 |
| 3 | Lê Dương Khánh Việt | 19522515 |

Thành phố Hồ Chí Minh – 01/2022

LỜI CẢM ƠN

Sau quá trình học tập và rèn luyện tại Trường Đại học Công Nghệ Thông Tin, chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Phạm Nguyễn Trường An và thầy Lê Đình Duy – Giảng viên phụ trách môn Máy Học đã tận tâm hướng dẫn, truyền đạt những kiến thức cũng như kinh nghiệm cho chúng em trong suốt thời gian học tập.

Trong quá trình làm đồ án môn học, mặc dù nhóm chúng em đã cố gắng nhưng chắc chắn sẽ không tránh được những sai sót không đáng có. Mong nhận được sự góp ý cũng như kinh nghiệm quý báu của các thầy và các bạn sinh viên để được hoàn thiện hơn và rút kinh nghiệm cho những môn học sau. Chúng em xin chân thành cảm ơn!

TP.Hồ Chí Minh, tháng 1, năm 2022

MỤC LỤC

| | |
|--|----|
| CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN | |
| 1.1 Lý do chọn đề tài | 4 |
| 1.2 Mục tiêu đề tài: | 4 |
| 1.3 Input và Output bài toán: | 4 |
| 1.4 Các bước thực hiện: | 5 |
| CHƯƠNG 2. XÂY DỰNG BỘ DỮ LIỆU | 5 |
| 2.1 Tiêu chí khi thu thập dữ liệu: | 5 |
| 2.2 Tổng quan về bộ dữ liệu: | 6 |
| CHƯƠNG 3. TỔNG QUAN VỀ MÔ HÌNH ĐƯỢC SỬ DỤNG | 10 |
| 3.1 Sơ lược về kiến trúc mạng YOLO v5: | 10 |
| 3.2 Chuẩn bị dữ liệu cho traininh model YOLO v5: | 11 |
| 3.3 Tiêu chí đánh giá kết quả thực nghiệm: | 14 |
| CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM | 16 |
| CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 19 |
| 5.1 Ưu điểm: | 19 |
| 5.2 Hạn chế: | 19 |
| 5.3 Hướng cải tiến: | 20 |
| CHƯƠNG 6. TÀI LIỆU THAM KHẢO | 21 |

CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN

1.1 Lý do chọn đề tài

Ngày nay, chúng tôi nhận thấy rằng những người tham gia giao thông đặc biệt là những người đi xe gắn máy, motor họ thường hay không quan tâm đến những biển báo giao thông bên đường hoặc một số người thậm chí còn không biết biển báo hiệu mà họ thấy là gì. Vì thế, để góp phần nhắc nhở cũng như nâng cao hiểu biết của người lái xe về các biển báo giao thông để tránh xảy ra những vấn đề đáng tiếc, nhóm chúng tôi đã cố gắng thử nghiệm việc vận dụng các kiến thức Machine Learning đã học được để giải quyết bài toán **Nhận diện biển báo giao thông**.

1.2 Mục tiêu đề tài

- Hiểu và vận dụng được những kiến thức của Machine Learning hay Deep Learning vào một vấn đề thực tế.
- Nắm được các bước trong quá trình xây dựng một mô hình Machine Learning.
- Nhận diện các biển báo giao thông thường thấy khi một người tham gia giao thông, đồng thời giúp người lái xe biết được các biển báo ít thông dụng.

1.3 Input và Output bài toán

- Input:
 - Một video ghi lại một đoạn đường có chứa các biển báo giao thông, thời lượng không quá 3 phút.
 - Độ phân giải tối thiểu là 480p
 - Video không có nhiễu và ánh sáng tốt.
- Output:

Video đầu vào sẽ được xuất ra với những thông tin sau:

 - Bounding box bao quanh biển báo nhận diện được trong video
 - Trên mỗi bounding box có ghi lại tên biển báo tương ứng và chỉ số biểu thị độ chính xác của biển báo nhận diện được.

1.4 Các bước thực hiện

- B1: Thu thập dữ liệu
 - + Dữ liệu thu thập sẽ bao gồm các ảnh chứa các biển báo giao thông được chụp trên đường và một vài video thỏa mãn yêu cầu của input.
- B2: Xử lý dữ liệu
 - + Tiến hành gán nhãn cho các ảnh thu thập được.
- B3: Chia các ảnh thành hai tập – tập train và tập validation, video đầu vào sẽ được xem như là tập test.
- B4: Lựa chọn model phù hợp để huấn luyện
- B5: Thử nghiệm mô hình đã huấn luyện được lên tập test và đánh giá độ hiệu quả của mô hình.

CHƯƠNG 2. XÂY DỰNG BỘ DỮ LIỆU

Để có thể tạo ra một mô hình Machine Learning thì thứ tiên quyết và quan trọng nhất chính là bộ dữ liệu. Bộ dữ liệu tốt hay xấu cũng ảnh hưởng rất nhiều đến chất lượng của mô hình vì thế, để kiểm soát được những yếu tố tác động lên dữ liệu như độ sáng, tối của ảnh, góc chụp hay số lượng các biển báo nhận diện cũng như cân bằng được số lượng các loại biển báo có trong bộ dữ liệu không chênh lệch nhau quá lớn, chúng tôi đã tiến hành thu thập dữ liệu thủ công bằng cách sử dụng camera cá nhân để chụp lại các biển báo trong đời thực.

Tuy nhiên, quá trình chụp ảnh thủ công vẫn gặp phải một số khó khăn như việc có một số biển báo mà số ảnh thu thập được rất ít dẫn đến mất cân bằng dữ liệu. Để khắc phục việc này, chúng tôi đã có tìm kiếm thêm trên mạng ảnh của các biển báo đó.

2.1 Tiêu chí khi thu thập dữ liệu

- Ảnh phải có chứa các biển báo giao thông, thuộc các loại biển báo mà chúng tôi cần nhận diện.
- Ảnh có nền sáng.
- Vị trí đặt camera cách biển báo không quá 10m.

2.2 Tổng quan về bộ dữ liệu

Sau khi tiến hành thu thập chúng tôi là có được một bộ dữ liệu gồm 3812 ảnh trong đó có 3512 ảnh được các thành viên trong nhóm chụp thủ công, và 300 ảnh được tìm kiếm trên mạng.

Trong 3812 ảnh đó, có chứa 61 loại biển báo khác nhau, và chúng tôi đã chia thành 61 classes khi áp dụng vào mô hình. Cụ thể như sau:

| | |
|---------------------------------------|--------------------------------|
| 0. CamXeTren10Tan | 30. CongTruong |
| 1. CamXeTren5Tan | 31. BatDauKhuDongDanCu |
| 2. CamxeTren2.5Tan | 32. HetKhuDongDanCu |
| 3. CamDiNguocChieu | 33. GiaoNhuChayVongXuyen |
| 4. CamReTrai | 34. HuongPhaiVongChuongNgaiVat |
| 5. CamRePhai | 35. HuongTraiVongChuongNgaiVat |
| 6. CamDungVaDoXe | 36. BenXeBuyt |
| 7. CamXeTaiTren5Tan | 37. Cho |
| 8. CamOToKhach | 38. NoiDoXe |
| 9. CamQuayXe | 39. TramXangDau |
| 10. ChoQuayXe | 40. CamDoXeNgayChan |
| 11. CamOToKhachVaOToTai | 41. TreEmDiNgang |
| 12. TocDoToiDa20km/h | 42. DuongNguoiDiBoCatNgang |
| 13. TocDoToiDa30km/h | 43. DuongNguoiDiBoSangNgang |
| 14. TocDoToiDa40km/h | 44. DuongHepHaiBen |
| 15. TocDoToiDa50km/h | 45. DuongGiaoNhuCungCap |
| 16. TocDoToiDa60km/h | 46. ReTrai |
| 17. TocDoToiDa80km/h | 47. RePhai |
| 18. GiaoNhuVoiDuongKhongUuTienBenTrai | 48. DuongLai |
| 19. GiaoNhuVoiDuongKhongUuTienBenPhai | 49. TocDoToiDa100km/h |
| 20. GiaoNhuVoiDuongKhongUuTienHaiBen | 50. DuongOGa |
| 21. GiaoNhuVoiDuongUuTien | 51. CamDoXeNgayLe |
| 22. GiaoNhuVoiDuongKhongUuTien | 52. CamXeTai |
| 23. GiaoNhuCoTinHieuDen | 53. CamXeTaiTren7Tan |
| 24. PhiaTruocNguyHiem | 54. HetHanCheTocDo40 |
| 25. NhieuChoNgoacNguyHiemLienTiep | 55. CamVuot |
| 26. ChoGoatNguyHiemVongBenTrai | 56. ChiDuocReTraiVaDiThang |
| 27. ChoGoatNguyHiemVongBenPhai | 57. HetHanCheTocDo50 |
| 28. DiCham | 58. HetHanCheTocDo60 |
| 29. DiThang | 59. CamOToTaiVuot |
| 30. CongTruong | 60. ChiDuocRePhaiVaDiThang |

- Tổng quan về bộ dữ liệu:

- Bộ dữ liệu gồm 3812 ảnh được chia thành hai tập: tập train gồm 3050 ảnh và tập validation gồm 762 ảnh và cùng với một bộ dữ liệu test mà mô hình hoàn toàn chưa tiếp xúc gồm 12 video được quay như điều kiện ràng buộc đã được đề cập ở phần input.

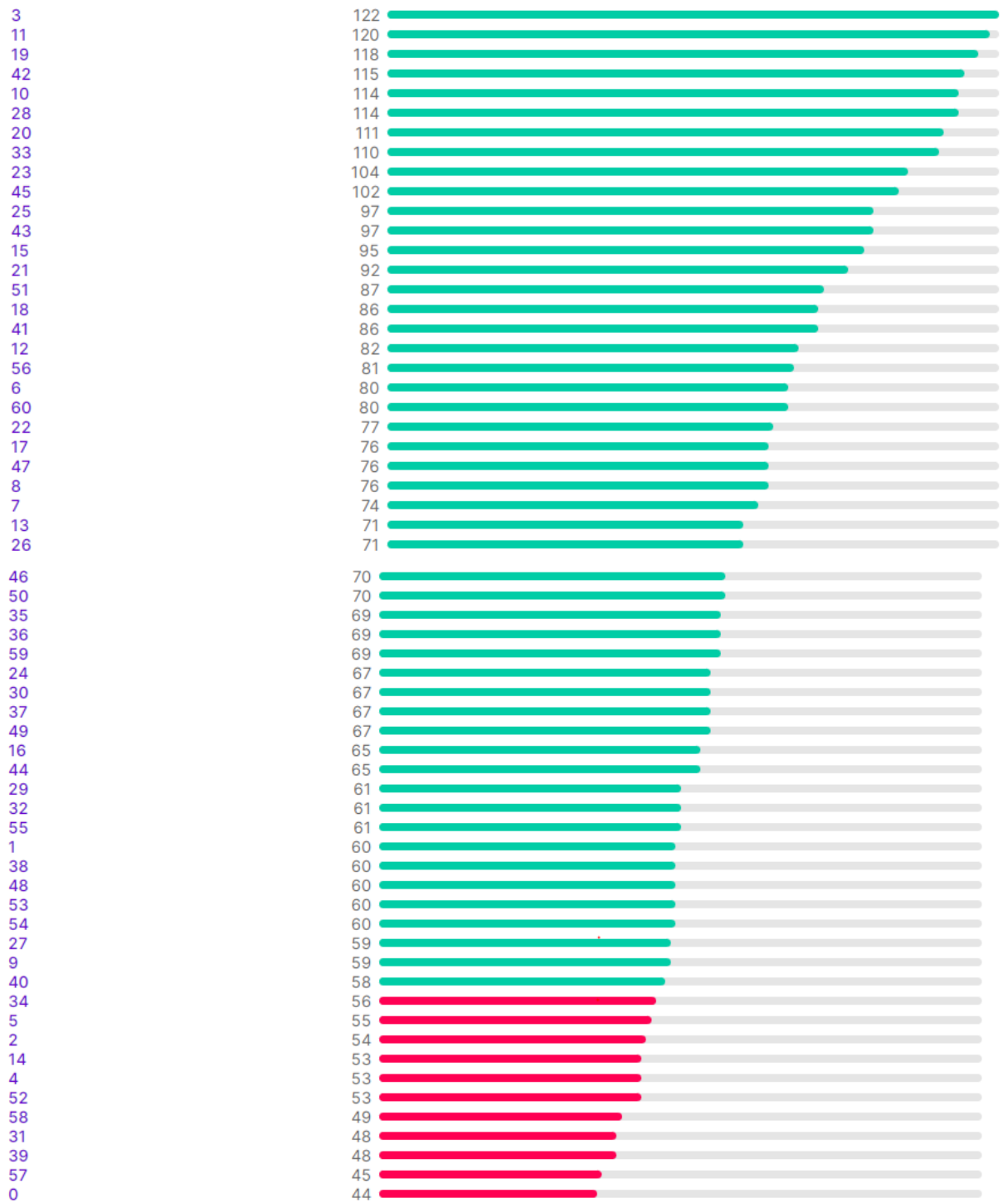


Hình 1: Một số biển báo giao thông trong bộ dữ liệu

- Những khó khăn trong quá trình thu thập dữ liệu:
 - + Một số biển báo giao thông ít được tìm thấy, dẫn đến việc phân lớp của biển báo đó bị ít đi số lượng ảnh.
 - + Số lượng ảnh nhiều, yêu cầu phải đa dạng về số phân lớp làm cho việc thu thập data tốn nhiều thời gian.
 - + Việc label ảnh cũng khá tốn thời gian và đôi khi không tránh khỏi những sai sót.

- Mô tả thông số của bộ dữ liệu:

Class Balance



Hình 2: Thống kê số ảnh của từng class

*** Nhận xét:**

- Sự chênh lệch giữa một số phân lớp vẫn còn khá nhiều → dữ liệu chưa được đồng bộ
- Vẫn có một số phân lớp chứa ít hơn 50 ảnh → chưa đạt mục tiêu nhóm đã đề ra
- Không có quá trình tiền xử lý dữ liệu làm cho mô hình dễ bị overfitting.
- Cần thu thập thêm những biển báo có số lượng ít để đảm bảo cân bằng dữ liệu.

*** Một số loại biển báo ít ảnh và cần phải thu thập thêm:**



Hình 3: Biển báo Cấm xe trên 10 Tấn



Hình 4: Biển báo Trạm xăng ở phía trước



Hình 5: Biển báo Cấm xe trên 2.5 tấn



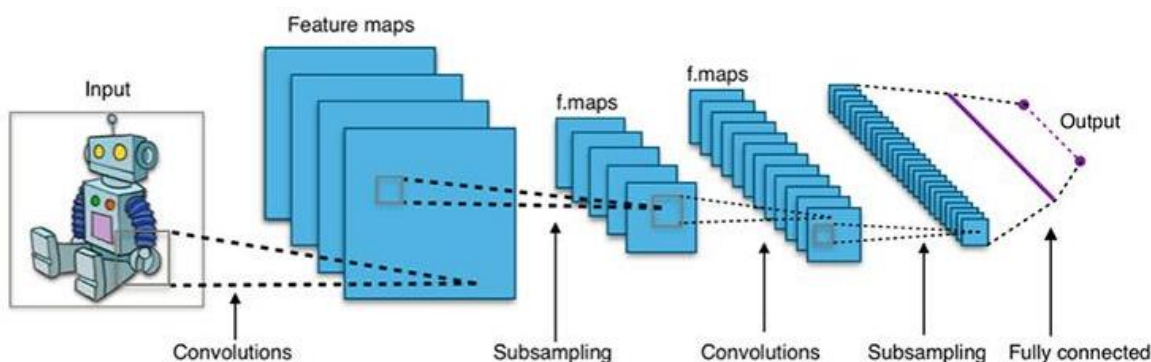
Hình 6: Biển báo Cấm rẽ phải

CHƯƠNG 3. TỔNG QUAN VỀ MÔ HÌNH ĐƯỢC SỬ DỤNG

Bài toán nhận diện biển báo giao thông thuộc lớp bài toán object detection và đã có rất nhiều giải pháp cũng như các mô hình được nghiên cứu trước đó được làm ra để xử lý bài toán này.

Một trong số đó là mô hình CNN (Convolution Neural Network), một mô hình được sử dụng rất rộng rãi ngày nay cho việc giải quyết các bài toán phân lớp dựa trên những yếu tố:

- Được huấn luyện tốt trên một bộ dữ liệu lớn trước đó.
- Thiết kế thông minh, có những thành phần được lặp đi lặp lại nên chia sẻ được các trọng số đã học được.
- Có kiến trúc phân tầng, học được từng đặc trưng từ các cấp độ khác nhau, đến lớp cuối cùng thì có thể học ra được các phần của một biển báo hoàn chỉnh.



Hình 7: Cấu trúc mạng CNN

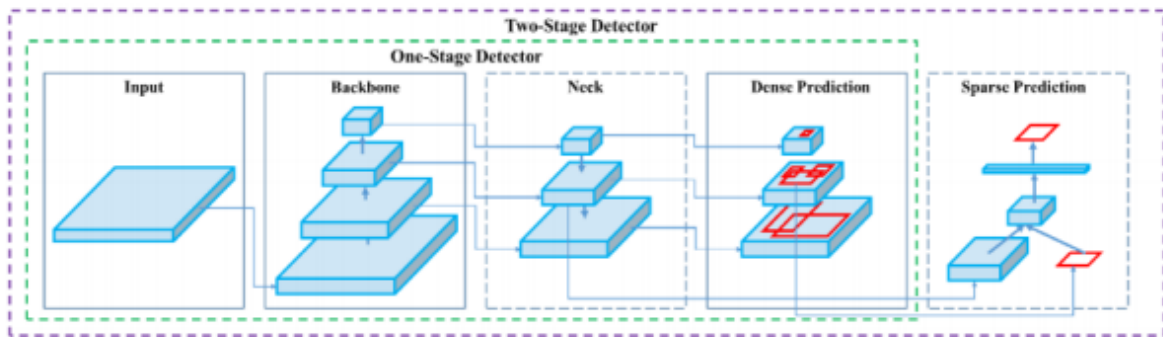
3.1 Sơ lược về kiến trúc mạng YOLO v5

Qua quá trình tìm hiểu, chúng tôi đánh giá rằng mô hình YOLO v5 là mô hình phù hợp nhất để giải quyết bài toán đã đặt ra và đã tiến hành áp dụng mô hình này trong phần thực nghiệm.

YOLO v5 về cơ bản cũng kế thừa các phương pháp xử lý của YOLO nhưng nó áp dụng một số các thuật toán giúp hỗ trợ cho việc phát hiện vật thể nhanh hơn đồng thời tối ưu hóa các phép toán thực hiện và giảm thời gian huấn luyện một cách tối ưu, hay có thể hiểu rằng YOLO v5 là một phiên bản cải tiến mới nhất của YOLO.

Cấu trúc nhận diện vật thể của YOLO v5 bao gồm 3 phần:

- **Backbone:** là 1 mô hình pretrain của một mô hình transfer learning khác với nhiệm vụ là giúp cho mô hình học được các đặc trưng và vị trí của vật thể. Các mô hình transfer learning thường là: VGG16, ResNet-50,...
- **Head:** được sử dụng để tăng khả năng phân lớp các đối tượng của mô hình. Ở phần này có thể sử dụng tầng 1 hoặc tầng 2, trong đó:
 - Tầng 1: dự đoán trên toàn bộ mô hình với các mô hình RPN, YOLO, SSD,...
 - Tầng 2: dự đoán theo từng mảng được dự đoán là có vật thể với các mô hình như R-CNN series,...
- **Neck:** Neck là phần nằm ở giữa Backbone và Head, có chức năng làm giàu thêm thông tin vật thể cho mô hình nhận diện bằng cách kết hợp thông tin giữa quá trình bottom-up và quá trình top-down.



Hình 8: Cấu trúc nhận diện vật thể của YOLO v5

Ở đây, kiến trúc backbone là phần quan trọng nhất. Nó kết hợp giữa Darknet53 và chiến lược CPSNet để trích xuất đặc trưng. Sau đó, chuyển đổi các đặc trưng thành các feature map rồi đẩy vào các layers để dự đoán labels cũng như bounding box vật thể.

3.2 Chuẩn bị dữ liệu cho việc huấn luyện mô hình

B1: Đưa tất cả 3812 ảnh thu thập được về kích thước 640x640

B2: Sử dụng tool makesense.ai để gán nhãn cho dữ liệu, dữ liệu cơ bản bao gồm

- 3812 hình ảnh và 3812 file label.
- Tên mỗi hình ảnh sẽ được đặt theo thứ 0.img, 1.img,...và mỗi file sẽ được đặt theo tên ảnh tương ứng là 0.txt, 1.txt,...

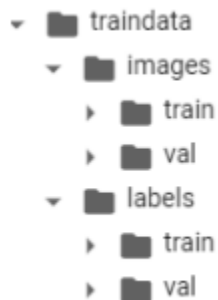


img2.txt - Notepad
File Edit Format View Help
21 0.856187 0.219064 0.060201 0.066890

Hình 9: Gán nhãn dữ liệu

B3: Chia dữ liệu và file gán nhãn tương ứng thành hai tập: tập train và tập val theo tỷ lệ 80:20

B4: Đưa dữ liệu về thành cấu trúc thư mục như sau:



Hình 10: Cấu trúc thư mục

B5: Giải nén thư mục này lại rồi đưa tất cả lên Drive

B6: Clone source code của mô hình YOLO v5 về

B7: Viết một files hình mô hình cho custom object detector. Files có đuôi là .yaml và có chứa những thông tin bao gồm:

- i) đường dẫn đến thư mục tập train trên Drive
- ii) đường dẫn đến thư mục tập test trên Drive
- iii) số classes

iv) tên mỗi class tương ứng

```
C: > Users > User > Downloads > ! custom.yaml
1
2 # 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, ..]
3
4 train: ../train_data/images/train/ # train images (relative to 'path') 128 images
5 val: ../train_data/images/val/ # val images (relative to 'path') 128 images
6
7 # Classes
8 nc: 60 # number of classes
9 names: ['CamXeTren10Tan', 'CamXeTren5Tan', 'CamXeTren2.5Tan', 'CamDiNguocChieu', 'CamReTrai', 'CamRePhai', 'CamDungVaDoXe', 'CamXeTaiTren5Tan', '
10 'ChoQuayXe', 'CamOToKhachVaOToTai', 'TocDoToiDa20km/h', 'TocDoToiDa30km/h', 'TocDoToiDa40km/h', 'TocDoToiDa50km/h', 'TocDoToiDa60km/h', '
11 'GiaoNhuVoiDuongKhongUuTienBenHaiBen', 'GiaoNhuVoiDuongUuTien', 'GiaoNhuVoiDuongKhongUuTien', 'GiaoNhuCoTinHieuDen', 'PhiaTruocNguyHi
12 'CongTruong', 'BatDauKhuDongDanCu', 'HetKhuDongDanCu', 'GiaoNhuChayVongXuyen', 'HuongPhaiVongChuongNgaiVat', 'HuongTraiVongChuongNgaiVat
13 'NoiDoXe', 'TramXangDau', 'CamDoXeNgayChan', 'TreEmDiNgang', 'DuongNguoiDiBoCatNgang', 'DuongNguoiDiBoSangNgang', 'DuongHepHaiBen', 'Duon
14 'DungLai', 'TocDoToiDa100km/h', 'DuongOGa/h', 'CamDoXeNgayLe', 'CamXeTai', 'CamXeTren7Tan', 'HetHanCheTocDo40', 'CamVuot', 'ChiDuocReTraiVaDiT
15
16
```

Hình 11: File cấu hình cho Custom Object Detector

B8: Lựa chọn **Base Model** nhỏ nhất, nhưng đảm bảo quá trình huấn luyện nhanh nhất. Base model mà chúng tôi lựa chọn đó YOLOv5s.pt

| Model | size (pixels) | mAP ^{val} 0.5:0.95 | mAP ^{test} 0.5:0.95 | mAP ^{val} 0.5 | Speed V100 (ms) | params (M) | FLOPs 640 (B) |
|---------|------------------|--------------------------------|---------------------------------|---------------------------|--------------------|---------------|------------------|
| YOLOv5s | 640 | 36.7 | 36.7 | 55.4 | 2.0 | 7.3 | 17.0 |

Hình 12: Base model được sử dụng

B9: Điều chỉnh các thông số huấn luyện

```
[ ] !python /content/drive/MyDrive/yolov5/train.py --img 640 --batch 16 --epochs 150 --data /content/drive/MyDrive/yolov5/data/custom.yaml --weights yolov5s.pt --cache
wandb: (1) Create a W&B account
wandb: (2) Use an existing W&B account
wandb: (3) Don't visualize my results
wandb: Enter your choice: (30 second timeout) 28
wandb: WARNING Invalid choice
wandb: Enter your choice: (30 second timeout) 30
```

Hình 13: Các thông số huấn luyện

Trong đó:

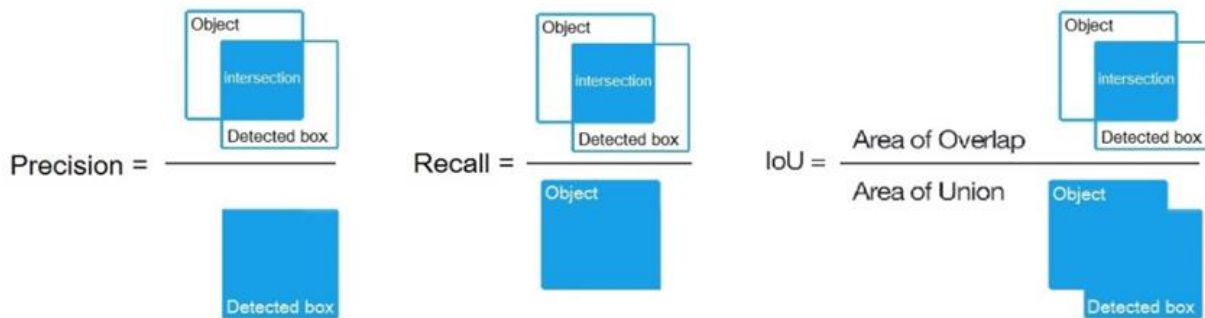
- i) img : kích thước của ảnh đầu vào
- ii) batch: số ảnh đưa vào quá trình huấn luyện một lần
- iii) epochs: số lần mô hình học dữ liệu đầu vào

B10: Bắt đầu quá trình huấn luyện

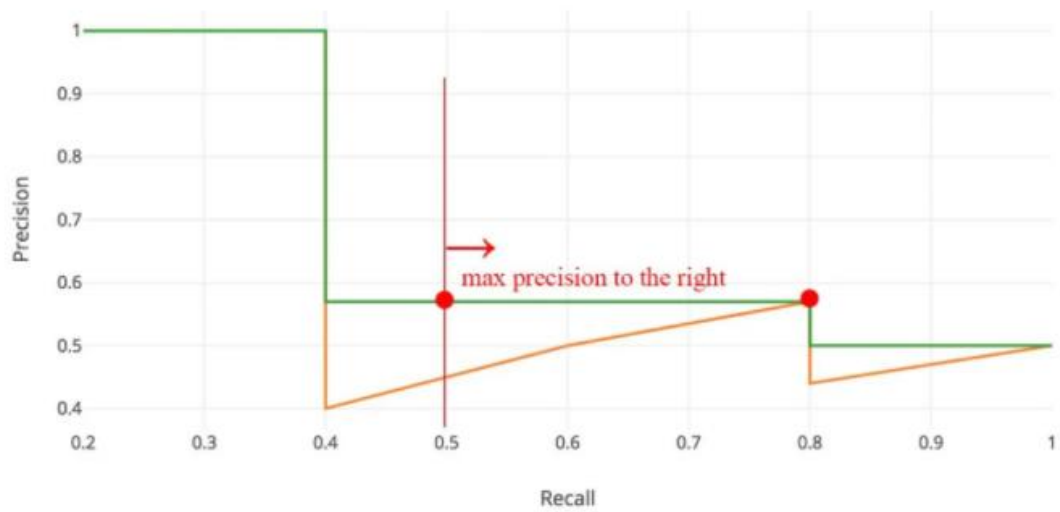
3.3 Tiêu chí đánh giá kết quả thực nghiệm

Hiện nay, trong các bài toán dạng object detection thì thước đo đánh giá phổ biến nhất đó là chỉ số mAP. Khi đề cập đến chỉ số này thì trước hết phải biết đến các khái niệm quan trọng như:

- **IoU**: là tỷ lệ giữa phần giao và phần hợp của bounding box vùng dự đoán và bounding box vùng đối tượng thật.
- ⇒ Đối với IoU chúng ta sẽ lựa chọn một ngưỡng cho trước (0.5, 0.55, 0.7,... tùy vào ngữ cảnh từng bài toán). Nếu IoU lớn hơn hoặc bằng ngưỡng này thì dự đoán đó được xem là true detection và ngược lại là false detection. Kí hiệu: IoU70 được hiểu là chỉ số IoU với ngưỡng là 0.7
- **Precision**: phần trăm các bounding box được dự đoán là true detection
 - **Recall**: phần trăm các bounding box được dự đoán đều chính xác
 - **AP**: là chỉ số đánh giá quan hệ giữa Precision và Recall
 - **mAP**: là trung bình cộng giá trị AP của các class khác nhau.



Hình 14: Minh họa cách tính IoU, Precision và Recall



$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

Hình 15: Minh họa cách tính AP thông qua Precision và Recall

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM

Chúng tôi đã tiến hành thực nghiệm dựa trên mô hình YOLO v5 với các thông số gồm: **batch= 16; img= 640** và **epochs = 200**. Kết quả đạt được đã cho ra kết quả [mAP@0.5:0.95](#) (mAP được lấy trung bình từ mAP của các ngưỡng từ 0.5 đến 0.95, bước nhảy là 0.05) là **0.76316** và [mAP@0.5](#) là **0.95795**, theo chúng tôi thì kết quả này cũng tương đối tốt.

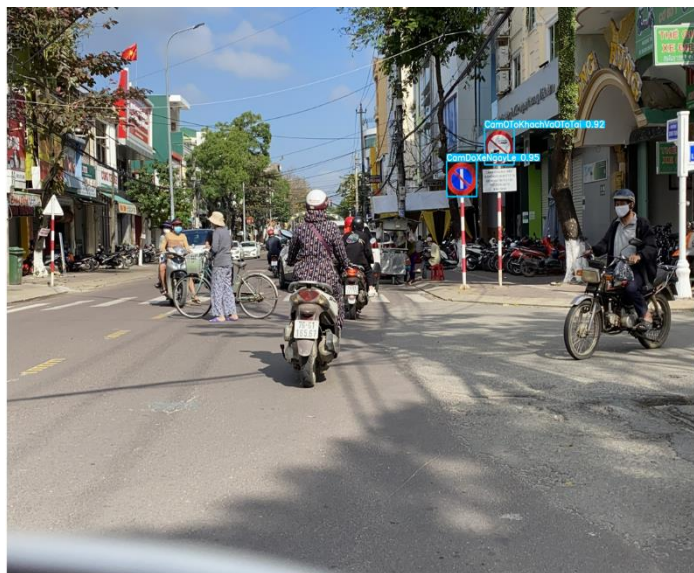
```
Run summary:
      best/epoch 148
      best/mAP_0.5 0.95795
    best/mAP_0.5:0.95 0.76316
      best/precision 0.88352
      best/recall 0.97539
    metrics/mAP_0.5 0.95795
  metrics/mAP_0.5:0.95 0.76324
    metrics/precision 0.88366
    metrics/recall 0.97524
    train/box_loss 0.01742
    train/cls_loss 0.00426
    train/obj_loss 0.00545
      val/box_loss 0.01792
      val/cls_loss 0.00449
      val/obj_loss 0.00322
        x/lr0 0.001
        x/lr1 0.001
        x/lr2 0.001
```

Hình 16: Đánh giá mô hình

Dưới đây là một số biển báo được mô hình nhận diện lần lượt qua ảnh sau khi train, ảnh và video:



Hình 17: Kết quả nhận diện trên ảnh sau khi train



Hình 18: Kết quả nhận diện trên ảnh



Video 1: Kết quả nhận diện video

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Ưu điểm

Qua quá trình tìm hiểu và cùng nhau đóng góp xây dựng đề án thì về cơ bản, nhóm đã được một số điểm tích cực như sau:

- Nắm được toàn bộ quá trình để huấn luyện một mô hình máy học.
- Hiểu được những khó khăn trong quá trình thu thập và xây dựng bộ dữ liệu.
- Khi huấn luyện mô hình trên bộ dữ liệu thì kết quả đạt được tương đối tốt. Kết quả này phụ thuộc nhiều vào độ đa dạng của dữ liệu, phân lớp dữ liệu.

5.2 Hạn chế

Bên cạnh những điểm tích cực đạt được, thì nhóm không thể tránh khỏi những thiếu sót trong lần đầu tiên thực hiện xây dựng một mô hình máy học:

- Số phân lớp tuy nhiều nhưng vẫn chưa đầy đủ hết tất cả các loại biển báo
- Sự mất cân bằng dữ liệu vẫn còn lớn.

- Một số ảnh đã bị chụp sai kiểu so với yêu cầu đặt ra.
- Quá trình huấn luyện dựa trên một mô hình đã được xây dựng từ trước nên nhóm vẫn chưa tìm hiểu đầy đủ về cách thức mô hình học và huấn luyện như thế nào.
- Quy mô chưa đủ lớn để có thể đem ra áp dụng vào thực tiễn.

5.3 Hướng cải tiến

Dựa vào sự phát triển công nghệ mạnh mẽ như hiện nay thì tương lai có thể sử dụng các phiên bản YOLO mới nhất và nhiều mạng khác như Single Shot Detector, RentiaNet, CenterNet,...để có thể huấn luyện và nhận dạng được nhiều nhóm biển báo giao thông một cách chính xác và nhanh gọn hơn nhằm:

- Cải tiến chất lượng bộ huấn luyện phát hiện ảnh biển báo .
- Mở rộng cơ sở dữ liệu biển báo giao thông.
- Cải tiến phương pháp giải quyết trường hợp các biển báo bị hư hỏng hoặc bị chồng lấp.
- Nâng cấp và hoàn thiện khả năng của hệ thống trở thành một hệ thống nhận dạng và đưa ra cảnh báo tức thời cho người tham gia giao thông trong một chương trình hoàn chỉnh

CHƯƠNG 6. TÀI LIỆU THAM KHẢO

- Link github của nhóm chứa dataset, code, video, ảnh và báo cáo:

[1] <https://github.com/ynguyenntc/CS114.M11>

- Nguồn lấy hình ảnh

[1] <https://itnavi.com.vn/blog/cnn-la-gi>

[2] <https://dothanhblog.wordpress.com/2020/04/24/map-mean-average-precision/?fbclid=IwAR2cdVtFDOJsXyEANqs6SEQGvkEXA9FhtG33NSeoojCrQYBlkmdm7BxUwxc>

- Tìm hiểu về mô hình YOLO v5

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

[2] <https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb>

- Tìm hiểu về chỉ số đánh giá mAP

[1] <https://blogcuabuicaodoanh.wordpress.com/2020/02/22/mean-average-precision-map-trong-bai-toan-object-detection/>

[2] <https://miai.vn/2021/10/14/thu-tim-hieu-ve-map-do-luong-object-detection-model/>

[3] <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>

- Tìm hiểu về quá trình xây dựng mô hình

[1] <https://www.youtube.com/watch?v=I7F6pr9vhVo&list=WL&index=3&t=1076s>

[2] <https://www.youtube.com/watch?v=g8BXbj44KLI>

[3] Yi-Qi Huang, Jia-Chun Zheng, Shi-Dan Sun, Cheng-Yi Chen, Optimized YOLOv3 Algorithm and Its Application in Traffic Flow Detections, 2020.

[4] Vũ Hữu Tiệp, “Machine Learning cơ bản”, Nhà xuất bản khoa học và kỹ thuật, 2016.

- Các tool hỗ trợ

[1] <https://roboflow.com/>

[2] <https://www.makesense.ai/>