

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



ĐỒ ÁN MÔN: MÁY HỌC – CS112.N21

NHÓM: DuongGiaARap

**ĐỀ TÀI: MÔ HÌNH NHẬN DIỆN SẢN PHẨM HÀNG HÓA
ỨNG DỤNG VÀO HỆ THỐNG SIÊU THỊ KHÔNG THU NGÂN**

Giảng viên hướng dẫn: Phạm Nguyễn Trường An

Thành viên thực hiện:

1. Nguyễn Nhật Minh – 21521135
2. Lê Tiến Quyết – 21520428
3. Trương Tấn Sang – 20520736

Thành phố Hồ Chí Minh, 7/2023

TÓM TẮT ĐỒ ÁN

Tên đề tài: Mô hình nhận diện sản phẩm hàng hóa ứng dụng vào hệ thống siêu thị không thu ngân

1. Mô tả bài toán

- Input: một bức ảnh chứa sản phẩm trên quầy thanh toán.
- Output: thông tin về sản phẩm xuất hiện trong bức ảnh đó, vị trí của vật thể trong bức hình

2. Xây dựng bộ dữ liệu

- Cách thu thập dữ liệu: các thành viên trong nhóm sử dụng camera trên máy điện thoại của mình để chụp hình sản phẩm
- Cách gán nhãn: sử dụng tool labelImg để dán nhãn cho từng bức ảnh
- Số lượng: bao gồm 44 sản phẩm khác nhau, mỗi sản phẩm 100 tấm hình, tổng cộng là 4400 tấm hình khác biệt nhau

3. Tiền xử lý

- Do ảnh chụp từ nhiều máy điện thoại khác nhau nên có kích thước rất lớn, resize tất cả ảnh về kích thước 600x600 px
- Về điều kiện ánh sáng chụp, vì các bức ảnh đều được chụp chung với cùng một điều kiện ánh sáng đầy đủ nên không thực hiện tinh chỉnh độ sáng, độ tương phản,...
- Gán nhãn cho từng tấm ảnh, sử dụng tool labelImg.

4. Trích xuất đặc trưng

- Sử dụng mô hình CNN CSP-Darknet53 có được xây dựng sẵn trong YOLOv5 để trích xuất đặc trưng của bức ảnh thông qua các layers.

5. Sử dụng mô hình phân lớp

- Mô hình lựa chọn training bao gồm Yolov5n và Yolov5s.

6. Đánh giá

- Các độ đo: mAP, Precision, Recall, Accuracy, F1-score.

7. Tinh chỉnh siêu tham số

- Cho mô hình chạy với số img_size, batch, epoch, mô hình(yolo5n,5s,...) khác nhau

8. Xem xét vấn đề và hướng phát triển

- Xem xét vấn đề overfitting, vấn đề tốc độ xử lý, hướng phát triển ứng dụng realtime detection và multi-class detection.

Link đến dataset: [link dataset](#)

Link đến repo chứa mã nguồn mở và kết quả: [link mã nguồn và kết quả](#)

Mục lục

| | |
|---|-----------|
| I. Mô tả bài toán | 4 |
| 1. Đặt vấn đề, bối cảnh hiện nay | 4 |
| 2. Các nghiên cứu/dự án về bài toán này trước đó..... | 5 |
| 3. Phát biểu bài toán | 7 |
| II. Xây dựng bộ dữ liệu..... | 8 |
| 1. Thu thập dữ liệu | 8 |
| 2. Gán nhãn dữ liệu | 11 |
| 3. Tổng quan về bộ dữ liệu..... | 11 |
| III. Trích xuất đặc trưng | 13 |
| IV. Mô hình và huấn luyện mô hình YOLOv5..... | 13 |
| 1. Lý do chọn mô hình YOLOv5 | 13 |
| 2. Các thông tin về YOLOv5 mà nhóm tìm hiểu được | 15 |
| 3. Quá trình chuẩn bị dữ liệu cho model YOLOv5..... | 17 |
| 4. Quá trình huấn luyện mô hình..... | 17 |
| V. Đánh giá các mô hình | 18 |
| 1. Kết quả của mô hình khi đánh giá trên bộ dữ liệu đã xây dựng | 18 |
| 2. Đánh giá mô hình trong trường hợp mô hình phải nhận diện với nhiều sản phẩm cùng lúc | 21 |
| a. Tổng quan về dữ liệu cho việc đánh giá | 21 |
| b. Đánh giá mô hình với bộ dữ liệu đánh giá mới được thu thập | 22 |
| 3. Giải thích một số khái niệm có liên quan..... | 42 |
| VI. Deploy mô hình lên ứng dụng demo..... | 44 |
| VII. Ứng dụng và hướng phát triển | 45 |
| VIII. Mục tham khảo..... | 46 |
| VIII. Cập nhật sau khi vấn đáp | 47 |

I. Mô tả bài toán

1. Đặt vấn đề, bối cảnh hiện nay

Thế giới của chúng ta đã bước sang năm thứ 24 của cuộc CMCN 4.0. Với sự phát triển rất nhanh chóng của công nghệ thông tin, đặc biệt là trí tuệ nhân tạo, được ứng dụng rất rộng rãi cho mọi lĩnh vực trong đời sống của chúng ta.

Hiện nay, ngoài hình thức mua sắm qua các trang thương mại điện tử (trực tuyến), thì hình thức mua bán trực tiếp vẫn còn rất phổ biến. Là một nhóm nhỏ các người dùng đã từng trải nghiệm mua sắm tại nhiều siêu thị, chúng em nhận thấy việc thanh toán tại quầy thu ngân hiện nay còn nhiều hạn chế và thiếu hiệu quả:

- Cần lượng nhân lực lớn cho việc túc trực tại các quầy thu ngân
- Một số nhân viên thu ngân có thể gặp khó khăn trong việc tìm mã vạch/QR của sản phẩm
- Quầy thu ngân bị bỏ trống trong một số thời điểm để giảm chi phí nhân công
- Hiện tượng quá tải thu ngân trong thời điểm khách hàng quá đông (dịp lễ, cuối tuần) gây bất tiện và mất thời gian cho khách hàng

Đó là những bất cập và hạn chế đối với hình thức thanh toán và thu ngân truyền thống. Chính vì những lý do đó, hiện nay, trên thế giới đã tiến hành thử nghiệm và áp dụng nhiều mô hình cửa hàng tiện lợi không thu ngân khác nhau, ví dụ: các cửa hàng mô hình này ở Nhật Bản, trong mỗi xe đầy đựng hàng sẽ được lắp một camera và có đèn chiếu sáng xung quanh, ứng dụng Computer Vision để nhận diện loại sản phẩm có trong giỏ hàng; một ví dụ khác là hệ thống siêu thị Walmat, người ta lắp đến 700 camera giám sát cho diện tích 30.000 m² và khách hàng phải tự quét mã vạch để tính tổng tiền cho sản phẩm. Trong 2 ví dụ trên, có thể thấy chúng tốn rất nhiều chi phí cho cơ sở hạ tầng, việc bảo trì cũng sẽ mất rất nhiều chi phí (tưởng tượng một cửa hàng lớn, phải trang bị nhiều xe đầy thông minh thì phải đầu tư rất nhiều tiền cho việc mua sắm và bảo trì các thiết bị đó).



Hình 1. Hình ảnh khách hàng đang sử dụng xe đẩy thông minh ở Nhật Bản

Chính vì những lý do trên, nhóm em đã đưa ra một giải pháp, đó là: việc thanh toán sản phẩm sẽ diễn ra tại các vị trí thanh toán tự động nhất định, người dùng phải đặt tất cả các sản phẩm của mình vào băng chuyền của hệ thống thanh toán, để hệ thống cho ra tổng số tiền mà khách hàng phải trả, chỉ khi khách hàng thanh toán thì máy mới trả ra hàng hóa đã đóng gói cho khách hàng (đảm bảo an ninh). Những lợi ích mà nhà cung cấp và khách hàng có thể nhận được:

- Phù hợp với xu hướng thanh toán không tiền mặt (vẫn cung cấp tính năng thanh toán bằng tiền mặt qua khe nhận tiền)
- Tối ưu chi phí xây dựng cơ sở hạ tầng, chi phí bảo trì thiết bị cho nhà cung cấp
- Cắt giảm nhân lực cho công việc thu ngân
- Duy trì hệ thống thu ngân liên tục mà không cần phải cắt giảm nhân viên
- Không mất thời gian cho việc tìm mã vạch/QR khi thanh toán
- Dễ dàng, người dùng có thể tự mình thực hiện thanh toán sản phẩm nhanh chóng, tiết kiệm thời gian

Ý tưởng giải pháp này của chúng em tuy không phải là hoàn toàn mới và đã có những sản phẩm với ý tưởng tương tự, tuy nhiên mô hình này chưa có ghi nhận là được áp dụng vào thực tế. Trong nội dung của môn học cũng như giới hạn kiến thức của bản thân, chúng em chỉ xây dựng một mô hình có chức năng nhận diện đối tượng sản phẩm, và cung cấp những thông tin cần thiết về sản phẩm đó.

2. Các nghiên cứu/dự án về bài toán này trước đó

a. Retail Store Item Detection using YOLOv5 - Joseph Nelson – 18/07/2020 [\[1\]](#)

Xét tại thời điểm bài báo được phát hành, YOLOv5 là phiên bản mới nhất của thuật toán YOLO. Nó cho độ chính xác và tốc độ nhanh hơn các phiên bản trước đó, và cũng là phiên bản có kích thước model nhỏ nhất.

- Bộ dữ liệu: SKU110k với 8232 ảnh train, 587 ảnh validation, 2940 ảnh test
- Model được dùng: YOLOv5, epoch = 300
- Thời gian train: 4 giờ 37 phút khi sử dụng Tesla P100 16GB GPU (Google Colab Pro)
- Performance: mAP = 0.7 với IoU threshold = 0.5

Kết luận của bài báo: mô hình YOLOv5 đáp ứng tốt yêu cầu của bài toán, tuy nhiên việc train model cần cấu hình mạnh và nhiều thời gian. YOLOv5 có kích thước các weights nhỏ và mức khung hình tốt là lựa chọn tốt cho các hệ thống nhúng cho việc phát hiện đối tượng trong thời gian thực.

**b. Deep Learning for Retail Product Recognition: Challenges and Techniques –
Yuchen Wei, Son Tran, Shuxiang Xu, Byeong Kang, Matthew Springer –
12/11/2020 [2]**

- Giới thiệu bài toán: Mã vạch được sử dụng rất phổ biến trong công nghiệp và bán lẻ. Nhưng vị trí in mã vạch là không cố định với từng loại sản phẩm dẫn đến mất nhiều thời gian để tìm. Dựa trên khảo sát của Digimarc, 45% khách hàng phản nản về sự bất tiện của mã vạch trong một số trường hợp.
- Sử dụng bộ dữ liệu từ: RPC, GroZi-120, GroZi-3.2k, Freiburg Grocery, Cigarette, Grocery Store, GP181, Checkout Datasets, D2S
- Model được sử dụng: Faster-RCNN, SSD, YOLOv2
- Thách thức:
 - + Bài toán phân loại với quy mô lớn: số lượng các sản phẩm khác nhau có thể rất lớn, lên tới vài nghìn loại
 - + Giới hạn lượng dữ liệu: yêu cầu một lượng lớn dữ liệu được gán nhãn cho việc train
 - + Ánh sáng phản xạ từ sản phẩm: do mỗi sản phẩm khi có góc nhìn và điều kiện ánh sáng khác nhau thì sẽ thể hiện thông tin khác nhau
 - + Tính linh hoạt của sản phẩm: số lượng các sản phẩm mới tăng lên theo từng ngày, bù ngoài của sản phẩm thay đổi thường xuyên

- Các kĩ thuật được áp dụng để khắc phục các thách thức trên:
 - + CNN-Based Feature Descriptors giải quyết Bài toán phân loại với quy mô lớn. Gần đây, YOLO9000 ra đời có khả năng phát hiện 9000 loại đối tượng khác nhau, nhưng nó được train với hàng triệu các bức ảnh, do đó, trong không thể áp dụng trong trường hợp này vì chi phí gán nhãn quá cao.
 - + Data Augmentation giải quyết vấn đề Giới hạn lượng dữ liệu
 - + Fine-Grained Classification giải quyết vấn đề Ánh sáng phản xạ từ sản phẩm
 - + One-Shot Learning giải quyết vấn đề Giới hạn lượng dữ liệu, Tính linh hoạt của sản phẩm
- Hướng nghiên cứu của nhóm tác giả: (nguyên văn tiếng Anh)
 - + Generating data with deep neural networks
 - + Graph neural networks with deep learning
 - + Cross-domain recognition with transfer learning
 - + Joint feature learning from text information on packaging
 - + Incremental learning with the CNN
 - + The regression-based object detection methods for retail product recognition

Kết luận của bài báo: phương pháp nhận diện sản phẩm này sẽ trở nên càng phổ biến, điều quan trọng là phải giải quyết 4 vấn đề thách thức và phát triển nghiên cứu theo 6 định hướng ở trên.

3. Phát biểu bài toán

Trong đồ án này, nhóm chỉ xây dựng một mô hình nhận diện đối tượng sản phẩm xuất hiện trong khung hình và trả về cho khách hàng những thông tin cần thiết về các sản phẩm đó.

- Input: là một tấm hình có chứa sản phẩm cần thanh toán được chụp trong môi trường đầy đủ ánh sáng
- Output: thông tin về sản phẩm và bounding box chứa object đó

Ví dụ:

- Input: hình chụp một chai tương ớt Chinsu
- Output: “Tuongotchinsu” và bounding box chứa object đó

II. Xây dựng bộ dữ liệu

1. Thu thập dữ liệu

Dữ liệu được nhóm tự thu thập bằng cách chụp hình 44 sản phẩm khác nhau với quy định cụ thể để đảm bảo tính thực tế của dữ liệu cho bài toán. Giả sử với ý tưởng của nhóm, trong thực tế thì camera thu nhận hình ảnh của sản phẩm được gắn cố định vào một điểm nhất định, với góc quan sát nhất định, sản phẩm được đặt trong khay đựng hoặc trên băng chuyền (nghĩa là đặt trên nền tròn), với ánh sáng được cung cấp đầy đủ.

Lý do cần phải tự thu thập dữ liệu thủ công:

- Dữ liệu có sẵn trên Internet không đảm bảo yêu cầu về dữ liệu cho bài toán: số lượng hình ảnh không đáp ứng đủ, các hình ảnh được chèn thêm các đối tượng không liên quan để quảng cáo ảnh hưởng đến chất lượng dữ liệu, kích thước của mỗi bức ảnh là khác nhau (có cái lớn, có cái nhỏ), không có đủ các góc chụp sản phẩm,...
- Để đảm bảo yêu cầu về dữ liệu cho mô hình, chúng em phải tự thu thập với các yêu cầu cụ thể về góc chụp, ánh sáng, số lượng,...



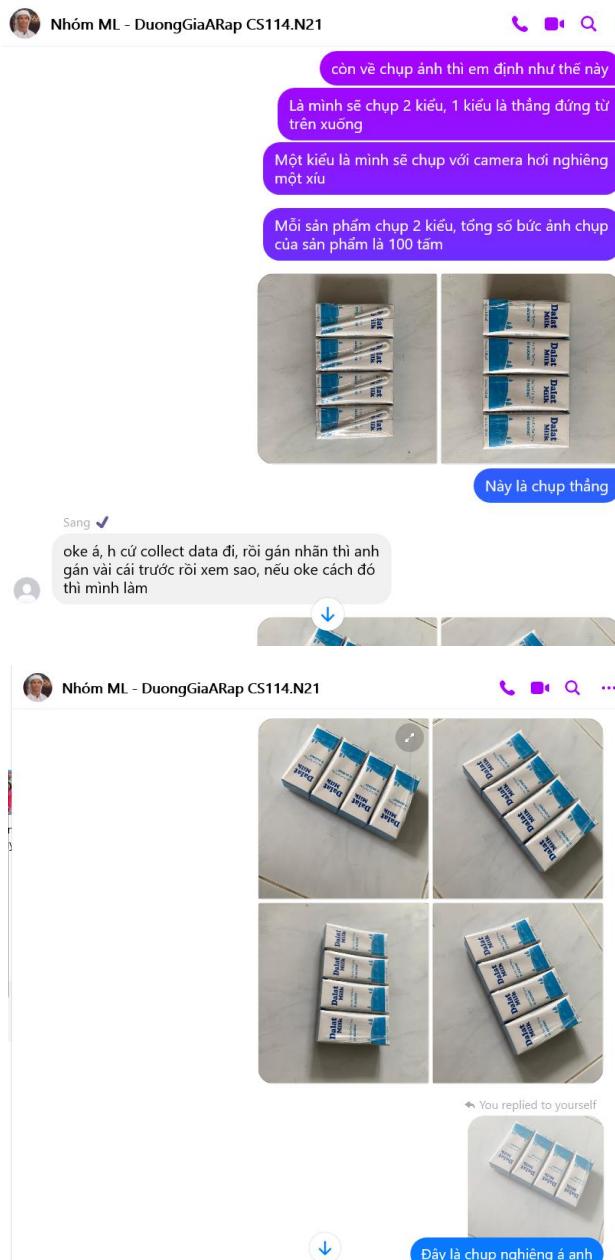
Hình 2. Một bức ảnh từ internet không đảm bảo yêu cầu thực tế của bài toán

Các yêu cầu khi thu thập dữ liệu:

- Đảm bảo ánh sáng của sản phẩm được chụp
- Sản phẩm phải được đặt trên nền sáng như: sàn nhà lát gạch sáng, bàn gỗ có màu sáng
- Vị trí của camera cách sản phẩm từ 20-30cm
- Đặt sản phẩm nằm khi chụp, phần bao bì chứa tên sản phẩm, thương hiệu, logo hướng lên trên
- Góc chụp: thẳng đứng và nghiêng góc khoảng 30 độ so với phương thẳng đứng chụp nhiều góc xung quanh sản phẩm

Cách thu thập dữ liệu:

- Các sản phẩm chụp được: các món đồ có sẵn ở trong phòng ở (khoảng 15 loại khác nhau), các sản phẩm được bày bán trong siêu thị Go Dĩ An (khoảng 29 loại khác nhau)
- Các sản phẩm được chụp bằng điện thoại cá nhân, sau đó được upload lên drive chung của nhóm



Hình 3. Nhóm thảo luận và thống nhất về cách chụp hình sản phẩm



Hình 4. Quá trình chụp hình sản phẩm



Hình 5. Hình ảnh sản phẩm được chụp thẳng đứng từ trên xuống



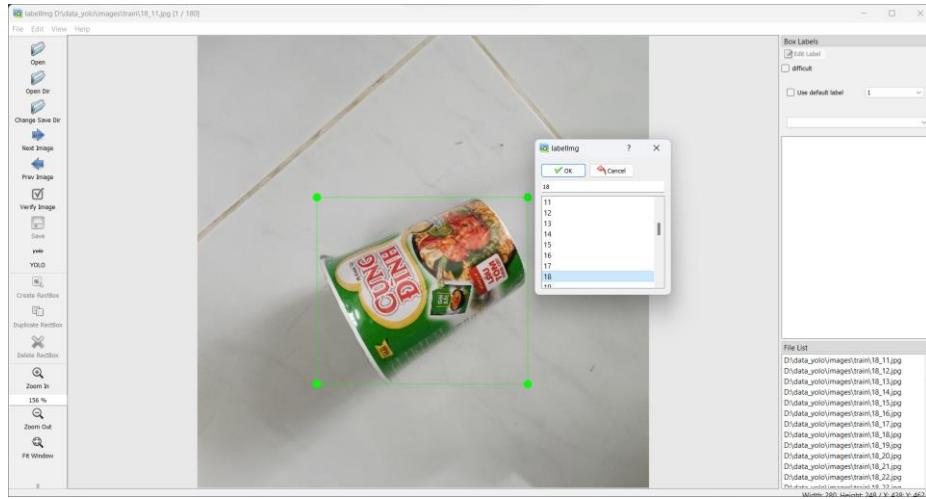
Hình 6. Hình ảnh sản phẩm được chụp hình ở góc nghiêng

Khó khăn nhóm gặp phải khi thu thập dữ liệu:

- Số lượng sản phẩm cần chụp lớn, yêu cầu nhiều không gian lưu trữ trong thiết bị di động (vì kích thước ảnh mà điện thoại di động chụp được có kích thước lớn, khoảng 2MB/1 tấm hình)
- Siêu thị không cho chụp hình sản phẩm của họ (rút kinh nghiệm từ anh chị đi trước), nhóm phải chụp lén trong thời gian ngắn và tránh sự kiểm tra của bảo vệ, nên có một số ít bức ảnh bị nhòe nhẹ.

2. Gán nhãn dữ liệu

Nhóm sử dụng công cụ LabelImg [3] để gán nhãn cho dữ liệu

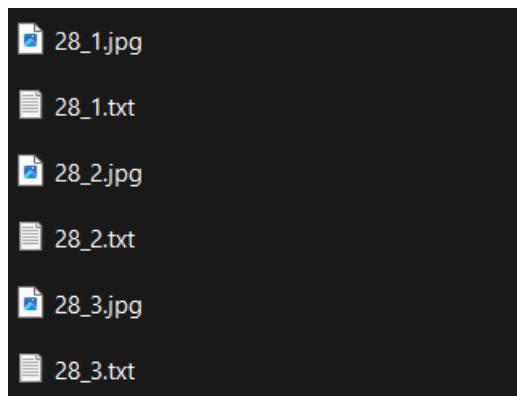


Hình 7. Thực hiện gán nhãn dữ liệu sử dụng công cụ labelImg

Nhóm mất hơn 1 ngày để chụp hình sản phẩm, 8 tiếng để gán nhãn cho toàn bộ 4400 bức ảnh. Ảnh được gán nhãn là ảnh đã được resize về kích thước 600x600.

Sau khi gán nhãn, bộ dữ liệu bao gồm:

- 4400 tệp hình ảnh được chụp đã resize về kích thước 600x600
- 4400 tệp *.txt là nhãn của từng tệp hình ảnh trùng tên tương ứng



Hình 8. Tệp hình ảnh và tệp nhãn .txt tương ứng

3. Tổng quan về bộ dữ liệu

- Bộ dữ liệu có tổng cộng 4400 tấm ảnh khác biệt được chụp của 44 loại sản phẩm khác nhau, mỗi loại chụp 100 tấm à Cân bằng dữ liệu.

| | |
|-----------------------------------|---|
| DauGoiDauClearMen : 100 | BanhQuyPhoMaiGery : 100 |
| KeoMamXoiVaDao : 100 | SuaChuaVinamilkCoDuong : 100 |
| ChaoGaGauDo : 100 | XucXichPonnieThitHeo : 100 |
| DauGoiDauHead&Shoulder : 100 | MiLyHaoHaoTomChuaCay : 100 |
| BanhQuyYenMachSocolaCosy : 100 | NuocCoGazPepsiCola : 100 |
| SuaTuoiaDalatMilk : 100 | HatNguCocMilo : 100 |
| BanhXopNabatiWafer : 100 | TraVietQuat&HoaAtisoDo : 100 |
| TuongOtChinSu : 100 | NuocC2TraXanhViChanh : 100 |
| SuaDauNanhFami1Lit : 100 | TrungVitKho : 100 |
| DauGoixmenForBoss : 100 | TrungGaAnLien : 100 |
| BanhQuyAFCViComNon : 100 | XucXichHeoCaoBoiPhoMaiBapBo : 100 |
| MiTomLyCungDinh : 100 | SnickersOats : 100 |
| HatVaHoaQuaBuasangDinhDuong : 100 | XucXichBapFiveStars : 100 |
| BlendHouseCaPheRangXay : 100 | CocaColaNguyenBan : 100 |
| MeXungGionThienHuong : 100 | KemDanhRangColgate : 100 |
| MiXaoHaoHaoTomXaoChuaNgot : 100 | GiaViNemSanCaKho : 100 |
| MiXaoHaoHaoTomHanh : 100 | TuongCaNamDuong : 100 |
| SuaTamLifebuoyVitaminDo : 100 | SotChamThitNuongHanQuoc : 100 |
| SuaYomostViCam : 100 | PhoMaiConBoCuoi : 100 |
| XaPhongRuaTayLifeBuoyXanh : 100 | Số lượng sản phẩm là 44 và tổng số lượng hình ảnh 4400. |
| TraThaiNguyenHuongNhai : 100 | |
| SuaCGHLViDau : 100 | |
| XylitolHuongLimeMint : 100 | |
| TokpokkiHanQuoc : 100 | |

Hình 9. Thống kê số lượng ảnh mỗi sản phẩm, và tổng số lượng ảnh

- Bộ dữ liệu được chia để train/validation/test theo tỉ lệ: 80/10/10
 - + Bộ dữ liệu train: 3520
 - + Bộ dữ liệu validation: 440
 - + Bộ dữ liệu test: 440
- Các class được đánh số thứ tự, một file csv lưu thông tin ảnh xạ từ số thứ tự thành tên sản phẩm.

Dữ liệu được nhóm hoàn toàn tự thu thập, các tấm ảnh được chụp với các quy định đã yêu cầu từ trước, nên đảm bảo dữ liệu tốt cho mô hình. Vì từ lúc tạo data đã đảm bảo dữ liệu không nhiễu nên chỉ cần resize kích thước ảnh cho phù hợp và gán nhãn.

Dataset được phân nhánh để phù hợp với mô hình theo sơ đồ

Train_data

```

  \_images
    \_train
    \_val
    \_test

  \_labels
    \_ train
    \_ val
    \_ test

```

Tất cả thông tin về classes, path được lưu trong file custom_data.yaml

III. Trích xuất đặc trưng

Để các object được nhận diện một cách chính xác, phân biệt với nhau ta nhận thấy các đối tượng trong bài toán thường là vật có dạng hình hộp, trụ, cầu. Mục tiêu của nhóm là nhận diện sản phẩm trên kệ thanh toán. Từ đó đặt trưng về hình dạng được quy định là hình chiểu vuông góc, hoặc gần vuông góc với vật đặt nằm ngang không xác định phương trên một mặt phẳng tròn. Từ đó trích xuất được đặc trưng của vật thể cần nhận diện là vật thể 2d có dạng hình chữ nhật hoặc hình tròn, ngoài ra còn có các đặc trưng riêng biệt của dữ liệu như có quai cầm, có vòi bơm,...

Đặc trưng về màu sắc của vật thể, đây là một đặc trưng quan trọng nên hệ màu của ảnh bắt buộc là hệ màu đa sắc. Đôi với yolov5 là hệ RGB.

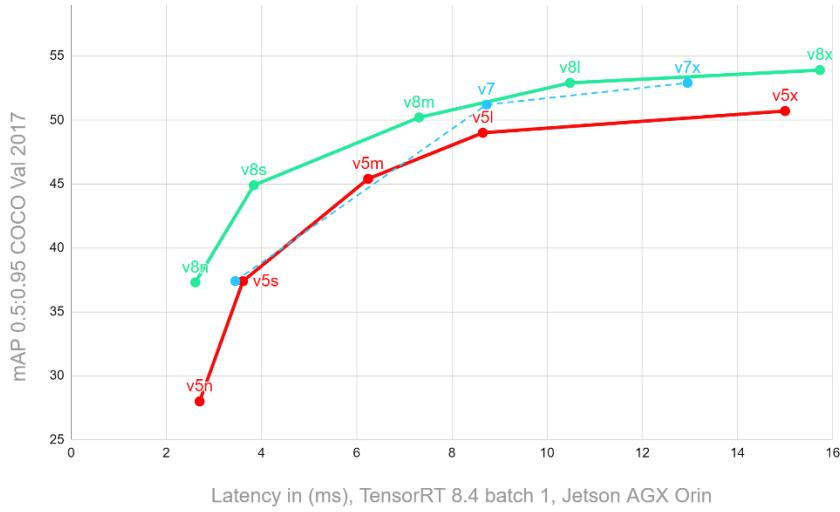
Với dataset của nhóm, dữ liệu được trích xuất bao gồm object và background, các đặc trưng của object được trích xuất thông qua mạng neural, quá trình trích xuất đặc trưng thủ công của bộ dữ liệu mà nhóm thu thập là việc phân tách giữa object và background bằng cách gán nhãn (labelling) cho mỗi object xuất hiện trong hình. Phần còn lại được mô hình tự động trích xuất và training theo cơ chế học transfer learning (mô hình được huấn luyện từ trước và có thể được huấn luyện lại với bộ dataset tùy chỉnh để thực hiện các nhiệm vụ riêng biệt).

IV. Mô hình và huấn luyện mô hình YOLOv5

1. Lý do chọn mô hình YOLOv5

Tính đến hiện nay, YOLO đã cho ra đời các phiên bản v6, v7, v8. Tuy nhiên nhóm vẫn chọn YOLOv5 làm mô hình cho bài toán của mình với các lý do sau:

- Khi so sánh YOLOv5 với YOLOv8, YOLOv8 cho ra kết quả tốt hơn YOLOv5 ở tốc độ phát hiện đối tượng, độ chính xác. Nhưng, YOLOv5 phù hợp hơn cho nhóm vì nhóm muốn deploy mô hình trên thiết bị laptop cá nhân với GPU (NVIDIA MX230) yếu hơn nhiều so với các loại GPU (NVIDIA Jetson AGX Orin 32GB, NVIDIA RTX 4070) mà các bài báo nghiên cứu sử dụng, trong khi đó, YOLOv8 thường được lựa chọn khi cần ưu tiên tốc độ và có sự hỗ trợ của GPU. Hơn nữa, YOLOv5 lại dễ sử dụng hơn. [\[4\]](#)



Hình 9. Đồ thị so sánh giữa các mô hình [5]

- Khi so sánh YOLOv5 với YOLOv6, YOLOv7, YOLOv5 vượt trội YOLOv6 về tốc độ xử lý khung hình trong video (kể cả trường hợp có hay không sự hỗ trợ của GPU), trong một số trường hợp có GPU, YOLOv7 có tốc độ xử lý nhanh hơn YOLOv5. Về độ chính xác, độ đo mAP của YOLOv5 không bằng YOLOv7 (chênh lệch ít hay nhiều còn tùy thuộc vào bộ dữ liệu), và tốt hơn YOLOv6 trong một số trường hợp [6]. Về tốc độ train, YOLOv5 nhanh hơn 2 phiên bản kia nhiều. Trong tốc độ nhận diện đối tượng, YOLOv5 tương đương YOLOv7, nhanh hơn YOLOv6. Các kết quả này còn phụ thuộc vào dữ liệu, số epochs, trong so sánh về độ chính xác, tốc độ train và detect nói trên sử dụng bộ dữ liệu nhỏ, với epochs = 50 [7].

So sánh YOLOv5 với một mô hình phổ biến khác, Faster R-CNN:

- Mô hình YOLOv5 tốt hơn trong việc nhận diện các đối tượng nhỏ hơn. Tốc độ xử lý của YOLOv5 cũng tốt hơn nhiều so với Faster R-CNN. Xét trong hoàn cảnh với đoạn video thời gian thực (real-time), có nhiều đối tượng nhỏ phân bố gần nhau, thì YOLOv5 ít xảy ra hiện tượng chồng lên nhau (overlapping) giữa các bounding box của các đối tượng. [8]

| | YOLO v5 | Faster RCNN |
|--|---------|-------------|
| Inference Speed | ✓ | |
| Detection of small or far away objects | ✓ | |
| Little to no overlapping boxes | ✓ | |
| Missed Objects | ✗ | ✗ |
| Detection of crowded objects | ✓ | ✓ |

Hình 10. Kết luận so sánh của tác giả Priya Dwivedi – trang towardsdatascience

- Đánh đổi về tốc độ xử lý, thì Faster R-CNN cho kết quả độ chính xác, độ đo mAP cao hơn YOLOv5

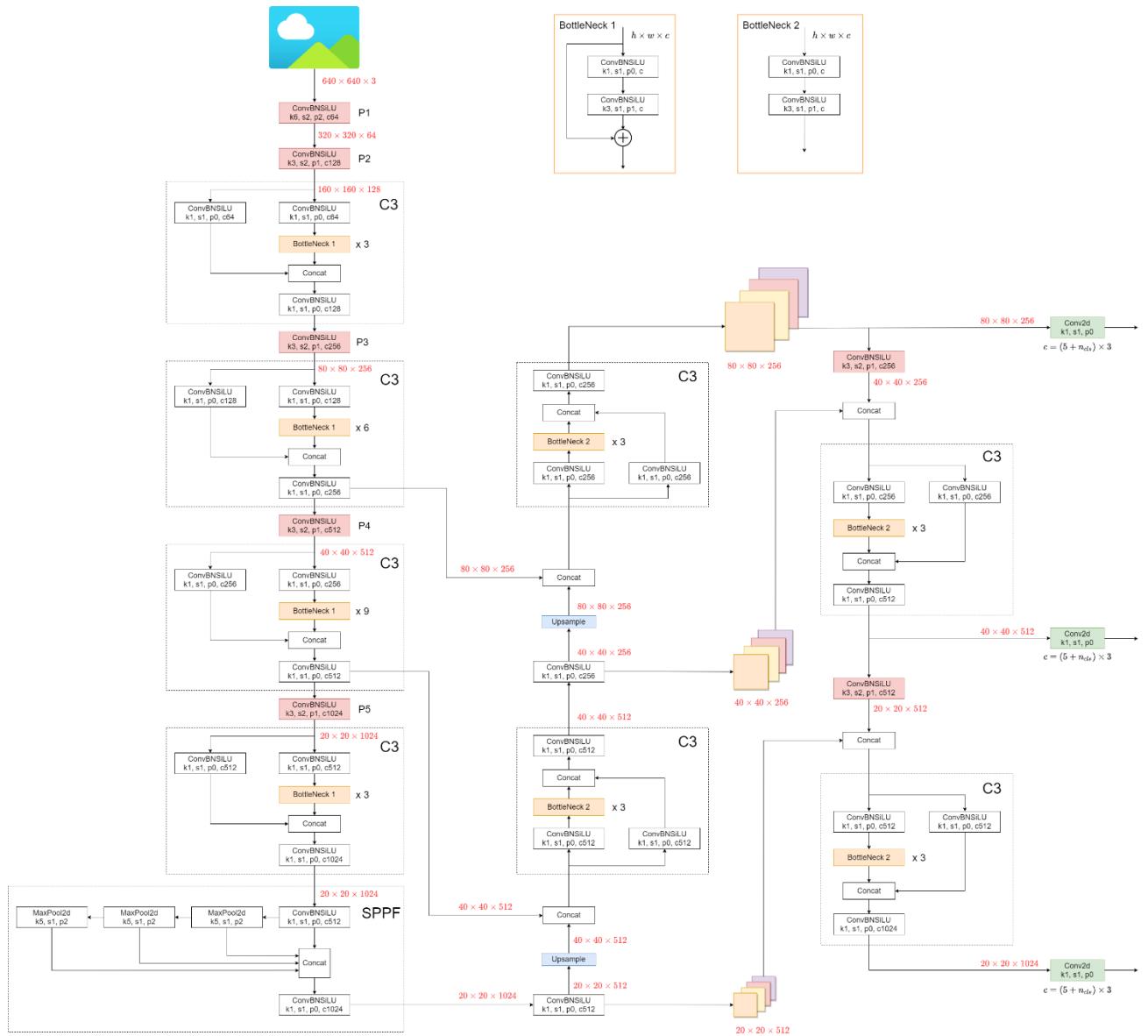
2. Các thông tin về YOLOv5 mà nhóm tìm hiểu được

Mạng YOLOv5 gồm 3 thành phần chính: backbone, neck và head: [9]

- Backbone là một mạng tích chập (CNN) tổng hợp và định dạng các đặc trưng của hình ảnh với các mức độ chi tiết khác nhau, chủ yếu của dụng một mạng CSP (BottleneckCSP được sử dụng để giảm số lượng tính toán và tăng tốc độ suy luận, trích xuất thông tin chi tiết từ các tầng gốc và cải thiện khả năng nhận dạng đối tượng) và SPP (SPP giúp cho mạng nơ-ron có khả năng xử lý các hình ảnh có kích thước khác nhau mà không cần thực hiện điều chỉnh kích thước ở bước tiền xử lý, nâng cao độ chính xác phát hiện) để trích xuất feature maps của những kích thước khác nhau từ đầu vào bằng nhiều tích chập và pooling. Backbone của YOLOv5 được thiết kế để sử dụng cấu trúc

New CSP - Darknet53

- Neck làm một mạng nơ-ron bao gồm nhiều lớp để kết hợp các đặc trưng của ảnh và chuyển tiếp chúng tới quá trình dự đoán. SPPF và New CSP-PAN là hai cấu trúc được sử dụng. Hai cấu trúc này cùng tăng cường đặc trưng được trích xuất từ các lớp mạng khác nhau trong Backbone, giúp cải thiện hơn nữa khả năng phát hiện của mô hình.
- Head là bước phát hiện cuối cùng, đầu ra của Head được sử dụng để dự đoán các mục tiêu có kích thước khác nhau trên feature maps.



Hình 11. Cấu trúc của YOLOv5 [\[10\]](#)

YOLOv5 bao gồm 10 phiên bản, trong đó, nhóm quan tâm đến 2 phiên bản là YOLOv5n và YOLOv5s. YOLOv5n là phiên bản có thời gian suy luận nhanh nhất, có số lượng params nhỏ nhất (1.9 triệu), tiếp theo đó, đến YOLOv5s có số lượng tham số nhỏ thứ hai (7.2 triệu). Đây là 2 phiên bản cho thời gian chạy trên nhanh nhất trong họ nhà YOLOv5. Với kích thước bộ dữ liệu của nhóm em, thì đây là phiên bản phù hợp vì dữ liệu của nhóm nhỏ, không quá đa dạng về chủng loại. [\[11\]](#)

| Model | size (pixels) | mAP ^{val} 50-95 | mAP ^{val} 50 | Speed CPU b1 (ms) | Speed V100 b1 (ms) | Speed V100 b32 (ms) | params (M) | FLOPs @640 (B) |
|-------------------|---------------|-----------------------------|--------------------------|-------------------------|--------------------------|---------------------------|---------------|-------------------|
| YOLOv5n | 640 | 28.0 | 45.7 | 45 | 6.3 | 0.6 | 1.9 | 4.5 |
| YOLOv5s | 640 | 37.4 | 56.8 | 98 | 6.4 | 0.9 | 7.2 | 16.5 |
| YOLOv5m | 640 | 45.4 | 64.1 | 224 | 8.2 | 1.7 | 21.2 | 49.0 |
| YOLOv5l | 640 | 49.0 | 67.3 | 430 | 10.1 | 2.7 | 46.5 | 109.1 |
| YOLOv5x | 640 | 50.7 | 68.9 | 766 | 12.1 | 4.8 | 86.7 | 205.7 |
| YOLOv5n6 | 1280 | 36.0 | 54.4 | 153 | 8.1 | 2.1 | 3.2 | 4.6 |
| YOLOv5s6 | 1280 | 44.8 | 63.7 | 385 | 8.2 | 3.6 | 12.6 | 16.8 |
| YOLOv5m6 | 1280 | 51.3 | 69.3 | 887 | 11.1 | 6.8 | 35.7 | 50.0 |
| YOLOv5l6 | 1280 | 53.7 | 71.3 | 1784 | 15.8 | 10.5 | 76.8 | 111.4 |
| YOLOv5x6 + TTA | 1280 1536 | 55.0 55.8 | 72.7 72.7 | 3136 - | 26.2 - | 19.4 - | 140.7 - | 209.8 - |

Hình 11. Bảng kết quả khi train với epochs = 300, bộ dữ liệu COCO 2017

3. Quá trình chuẩn bị dữ liệu cho model YOLOv5

Bộ dữ liệu:

- Bộ dữ liệu mà nhóm đã xây dựng, chi tiết cụ thể được trình bày ở phần II

Phân chia dữ liệu:

- Theo tỷ lệ 80% train, 10% validation, 10% test

Toàn bộ dữ liệu được upload lên drive trước khi train model.

4. Quá trình huấn luyện mô hình

Import bộ dữ liệu đã được upload lên drive từ trước đó và giải nén.

Đầu tiên, clone model YOLOv5 và cài đặt các yêu cầu ở trong file requirements.txt

Chuẩn bị các tài nguyên cần thiết cho việc huấn luyện mô hình:

- Chuẩn bị file chứa dữ liệu data.yaml, chứa đường dẫn đến dữ liệu train, test, validation và tên của các classes có trong bộ dữ liệu.

Lựa chọn các thông số cho mô hình (cả 2 phiên bản YOLOv5s và YOLOv5n đều được huấn luyện với các thông số này)

- --img 640: kích thước của network mà các bức ảnh sẽ được resize về trước khi training
- --batch 32: nếu dùng batch quá nhỏ thì việc cập nhật lại các tham số không được ổn định, ngược lại, nếu batch cao thì mất nhiều chi phí tính toán nên Google Colab bản base chỉ cho tối đa là 32.

- --epochs 50: nhóm chọn giá trị của epochs là 50. Vì nếu epochs quá lớn, với bộ dữ liệu nhỏ của nhóm thì có thể gây ra overfitting. Khi train với 50 epochs thì đã cho kết quả chính xác rất cao (đề cập trong phần V).
- --data data_yaml.yaml: cùng dẫn đến một file .yaml chứa đường dẫn đến dữ liệu train, validation, test và tên của các classes có trong bộ dữ liệu.

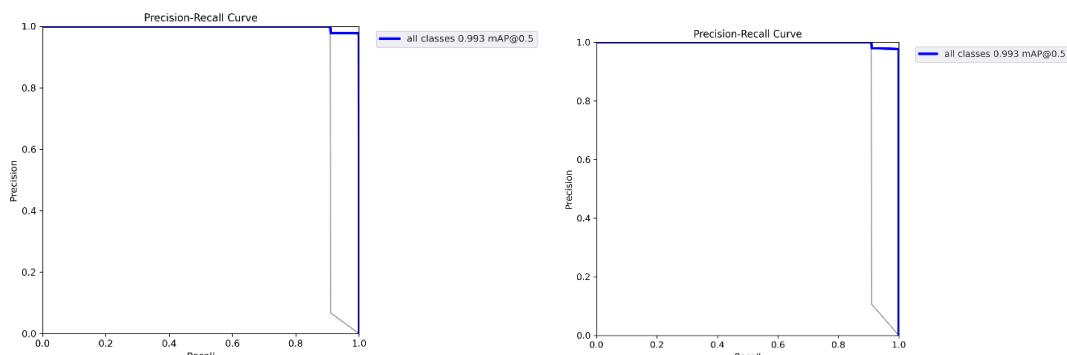
Thực hiện huấn luyện hai mô hình với cùng các cài đặt ở trên.

Thời gian huấn luyện của hai mô hình với các cài đặt ở trên:

- YOLOv5s: 2.223 giờ ~ 2 giờ 13 phút (thời gian trung bình cho mỗi epoch khoảng 2 phút 40 giây)
- YOLOv5n: 2.374 giờ ~ 2 giờ 23 phút (chạy epoch đầu tiên bị lỗi, mất 18 phút, tuy nhiên các epoch sau có thời gian trung bình cho mỗi epoch là khoảng 2 phút 34 giây)

V. Đánh giá các mô hình

1. Kết quả của mô hình khi đánh giá trên bộ dữ liệu đã xây dựng



Hình 12. PR-Curve của YOLOv5n (bên trái) và YOLOv5s (bên phải) với threshold = 0.5

Nhìn vào hình 13, diện tích phần nằm dưới đường màu xanh của hai phiên bản là 0.993 (~1).

Cho thấy mô hình dự đoán rất tốt trên tập val.

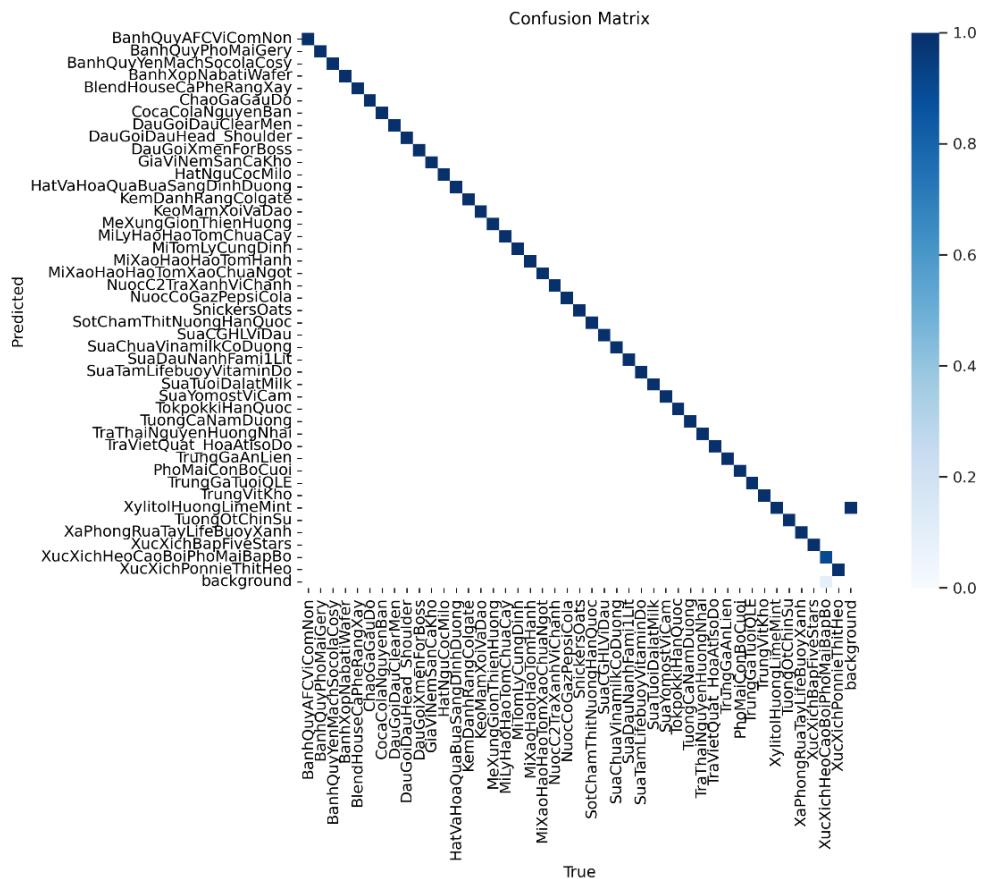
| Epoch | GPU_mem | box_loss | obj_loss | cls_loss | Instances | Size |
|-------|---------|----------|-----------|----------|-----------|--|
| 49/49 | 4.46G | 0.01741 | 0.00789 | 0.01306 | 68 | 640: 100% 110/110 [02:20<00:00, 1.28s/it] |
| | Class | Images | Instances | P | R | mAP50 mAP50-95: 100% 7/7 [00:09<00:00, 1.40s/it] |
| | all | 440 | 441 | 0.981 | 0.996 | 0.993 0.911 |
| | | | | | | |
| Epoch | GPU_mem | box_loss | obj_loss | cls_loss | Instances | Size |
| 49/49 | 8.56G | 0.01713 | 0.007292 | 0.009193 | 68 | 640: 100% 110/110 [02:31<00:00, 1.38s/it] |
| | Class | Images | Instances | P | R | mAP50 mAP50-95: 100% 7/7 [00:10<00:00, 1.54s/it] |
| | all | 440 | 441 | 0.988 | 0.994 | 0.993 0.931 |

Hình 13. Kết quả hai model YOLOv5n (bên trên) và YOLOv5s (bên dưới) ở epoch cuối cùng
Các giá trị: box_loss, obj_loss, cls_loss của YOLOv5s thấp hơn so với YOLOv5n. Cho thấy, mô hình YOLOv5s detect sản phẩm tốt hơn so với YOLOv5n.

Các chỉ số khác như: Precision, mAP50-95 của YOLOv5s khá tốt so với YOLOv5n.

Còn Recall và mAP50 thì chênh lệch thấp hoặc không chênh lệch.

Confusion Matrix của YOLOv5s



Hình 14. Confusion matrix của YOLOv5s trên tập val

Từ Confusion Matrix có thể thấy, có 2 trường hợp mà YOLOv5s gặp khó khăn khi thực hiện detect:

- Trường hợp với sản phẩm “XylitolHuongLimeMint”: mặc dù thực tế đó là background, nhưng mô hình lại đoán là sản phẩm XylitolHuongLimeMint



Hình 15. Sản phẩm XylitolHuongLimeMint

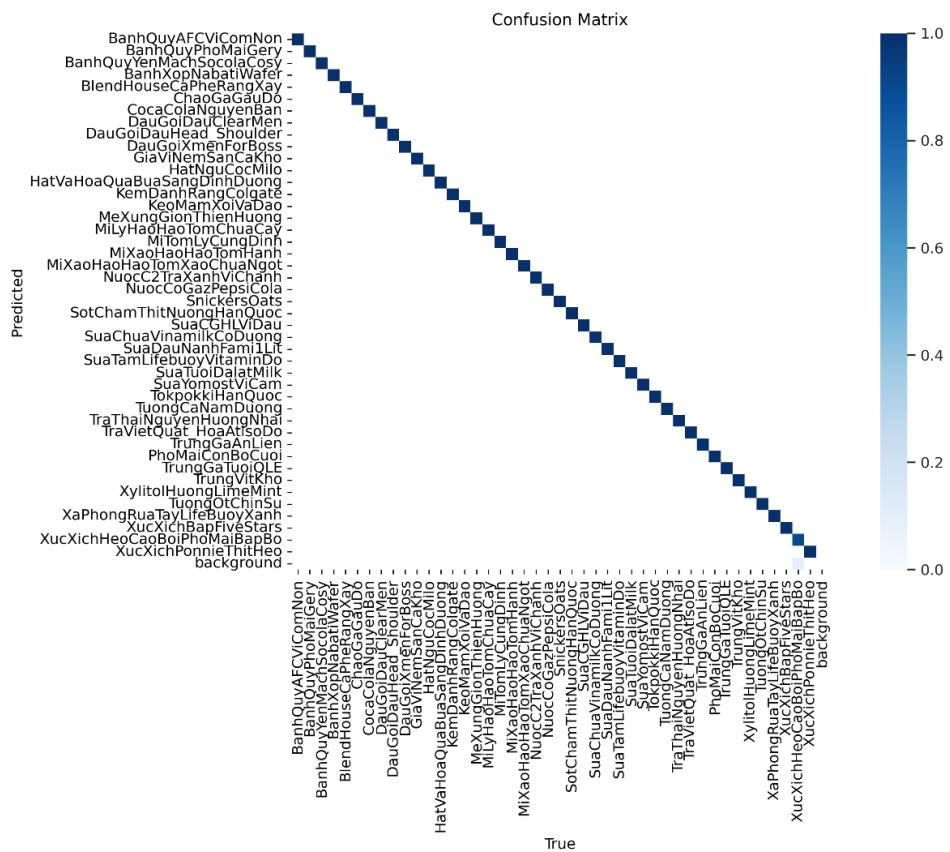
Nguyên nhân dự đoán: chưa tìm được nguyên nhân

- Trường hợp “XucXichHeoCaoBoiPhoMaiBapBo” có một lượng nhỏ tẩm hình mà mô hình không phát hiện ra đối tượng



Hình 16. Sản phẩm XucXichHeoCaoBoiPhoMaiBapBo

Confusion Matrix của YOLOv5n



Hình 17. Confusion matrix của YOLOv5n trên tập val

Từ Confusion Matrix ta có thể thấy, YOLOv5n không xảy ra hiện tượng đoán sai sản phẩm “XylitolHuongLimeMint” như YOLOv5s. Tuy nhiên, vẫn xảy ra trường hợp không phát hiện được sản phẩm “XucXichHeoCaoBoiPhoMaiBapBo”.

2. Đánh giá mô hình trong trường hợp mô hình phải nhận diện với nhiều sản phẩm cùng lúc

a. Tổng quan về dữ liệu cho việc đánh giá

Nhóm vẫn lựa chọn thực hiện phương án vào siêu thị chụp lén hình ảnh sản phẩm.

Cách chụp:

- Các sản phẩm được đặt nằm trên mặt bàn, với mặt thông tin của sản phẩm được hướng lên
- Để camera điện thoại cách vật thể khoảng 40 ± 5 cm
- Hướng camera điện thoại theo phương thẳng đứng, với chiều từ trên xuống dưới
- Đảm bảo sao cho các sản phẩm được nằm gọn trong khung hình của điện thoại
- Chụp các sản phẩm trong điều kiện áng sáng đầy đủ, được đặt trên nền mặt bàn trơn (do không có điều kiện để chụp sản phẩm trực tiếp trên khay thanh toán)

Cách bố trí sản phẩm:

- Nhóm thực hiện chụp nhóm các sản phẩm, với số lượng sản phẩm trong mỗi nhóm là 2,3,4,5 (để quan sát khả năng nhận diện của mô hình với số lượng các sản phẩm trong một khung hình khác nhau)
- Cách sắp xếp: bố trí các sản phẩm với các vị trí ngẫu nhiên



Hình 18. Một số hình ảnh được chụp

Tổng quan về bộ dữ liệu mới được thu thập:

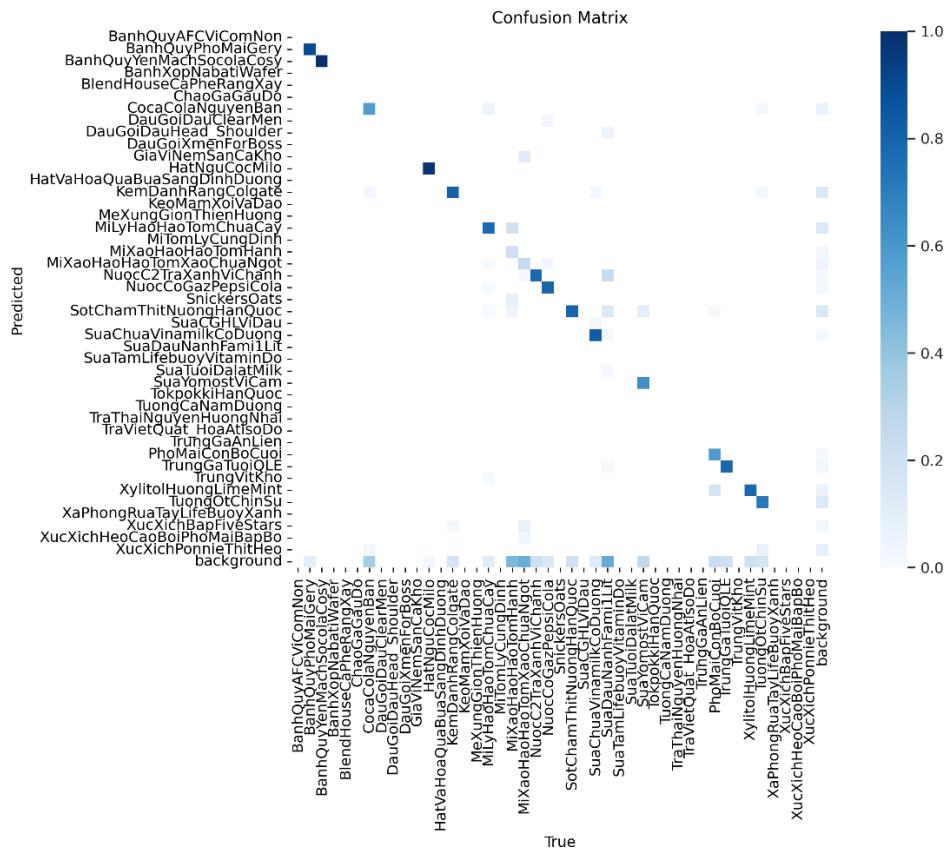
- Số lượng ảnh được chụp với 2 sản phẩm: 100 ảnh
 - Số lượng ảnh được chụp với 3 sản phẩm: 100 ảnh
 - Số lượng ảnh được chụp với 4 sản phẩm: 100 ảnh
 - Số lượng ảnh được chụp với 5 sản phẩm: 15 ảnh (nhóm chỉ chụp một vài tấm hình với 5 sản phẩm vì đây là số ảnh chụp bị dư ra, nên nhóm đưa vào bộ đánh giá để đa dạng hơn, nhưng vì số lượng ít nên chưa có nhiều ý nghĩa để kết luận với trường hợp này)

Dữ liệu được gán nhãn bằng tool labelImg tương tự như bộ dữ liệu đã thu thập để train.

b. Đánh giá mô hình với bộ dữ liệu đánh giá mới được thu thập

Đánh giá mô hình YOLOv5s

Danh sách trên toàn bộ dữ liệu mới thu thập



Hình 19. Confusion matrix của YOLOv5s khi đánh giá với toàn bộ bộ dữ liệu mới

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|-------|-------|----------------|
| all | 315 | 975 | 0.87 | 0.736 | 0.846 | 0.693 |
| BanhQuyPhoMaiGery | 315 | 10 | 0.977 | 1 | 0.995 | 0.851 |
| BanhQuyYenMachSocolaCosy | 315 | 12 | 0.944 | 1 | 0.995 | 0.936 |
| CocaColaNguyenBan | 315 | 31 | 0.679 | 0.71 | 0.714 | 0.611 |
| HatNguCocMilo | 315 | 39 | 1 | 0.994 | 0.995 | 0.879 |
| KemDanhRangColgate | 315 | 47 | 0.649 | 0.894 | 0.887 | 0.7 |
| MiLyHaoHaoTomChuaCay | 315 | 87 | 0.823 | 0.874 | 0.915 | 0.775 |
| MiXaoHaoHaoTomHanh | 315 | 46 | 1 | 0.375 | 0.739 | 0.611 |
| MiXaoHaoHaoTomXaoChuaNgot | 315 | 74 | 0.892 | 0.336 | 0.676 | 0.555 |
| NuocC2TraXanhViChanh | 315 | 46 | 0.541 | 0.783 | 0.765 | 0.636 |
| NuocCoGazPepsiCola | 315 | 34 | 0.934 | 0.831 | 0.857 | 0.673 |
| SotChamThitNuongHanQuoc | 315 | 69 | 0.579 | 0.841 | 0.867 | 0.705 |
| SuaChuaVinamilkCoDuong | 315 | 41 | 0.973 | 0.873 | 0.943 | 0.808 |
| SuaDauNanhFami1Lit | 315 | 78 | 1 | 0 | 0.333 | 0.293 |
| SuaYomostViCam | 315 | 60 | 1 | 0.695 | 0.989 | 0.917 |
| PhoMaiConBoCuoi | 315 | 69 | 1 | 0.66 | 0.942 | 0.691 |
| TrungGaTuoIQL | 315 | 75 | 0.921 | 0.773 | 0.906 | 0.58 |
| XylitolHuongLimeMint | 315 | 75 | 0.77 | 0.8 | 0.773 | 0.521 |
| TuongOtChinSu | 315 | 82 | 0.971 | 0.811 | 0.931 | 0.735 |

Hình 20. Kết quả các độ đo đánh giá mô hình YOLOv5s trên toàn bộ bộ dữ liệu mới



Hình 21. 16 trong toàn bộ 315 bức ảnh là kết quả dự đoán của mô hình YOLOv5s

Giá trị Recall chung là 0.736, điều này cho thấy rằng, có những sản phẩm xuất hiện trong khung hình nhưng mô hình không detect được. Nhìn vào phần độ đo của từng class, ta thấy sản phẩm SuaDauNanhFami1Lit thì mô hình đoán nhầm sai trong toàn bộ bộ dữ liệu đánh giá ($R = 0$, nguyên nhân có thể là do bộ dữ liệu train của sản phẩm này chưa tốt, chưa tổng quát) vì vậy, ở các phần sau, không cần đánh giá cho sản phẩm này. Ngoài ra còn có 2

sản phẩm MiXaoHaoHaoTomXaoChuaNgot và MiXaoHaoHaoTomHanh có $R < 0.5$ (nghĩa là mô hình chỉ detect các sản phẩm này chưa được một nửa số lần xuất hiện của chúng trong bộ dữ liệu). Có 2 sản phẩm được mô hình detect toàn bộ là BanhQuyPhoMaiGery và BanhQuyYenMachSocolaCosy.

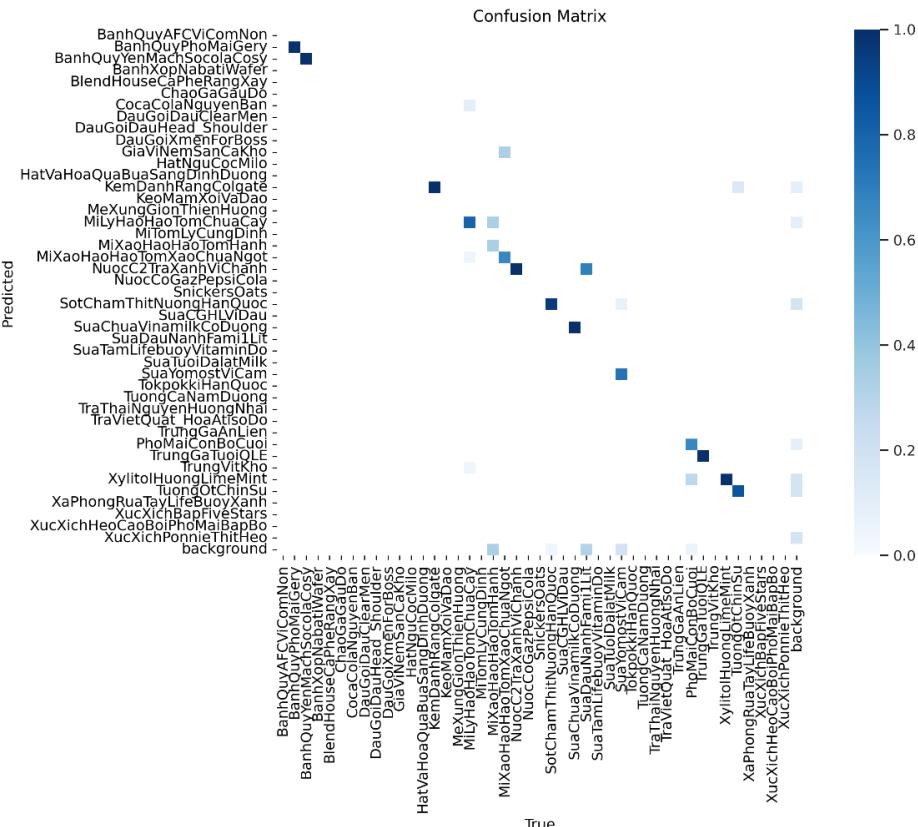
Giá trị Precision chung là 0.87, khá tốt (> 0.8). Xét Precision của từng class, có 5 sản phẩm được mô hình đoán nhẫn đúng hết tất cả các đối tượng mà mô hình detect được với $P = 1$ (HatNguCocMilo, MiXaoHaoHaoTomHanh, SuaDauNanhFami1Lit, SuaYomostViCam, PhoMaiConBoCuoi). Tuy nhiên, vẫn có những đối tượng mà mô hình đoán sai nhẫn, 2 nhẫn có tỷ lệ bị gán sai nhiều nhất là NuocC2TraXanhViChanh và SotChamThitNuongHanQuoc.

Giá trị mAP50-95 cũng không tốt ($0.693 < 0.8$).

Việc mô hình bỏ sót nhiều sản phẩm có xuất hiện trong khung hình và đoán sai nhẫn của sản phẩm là không chấp nhận được trong thực tế, vì điều này làm ảnh hưởng xấu đến lợi ích của người bán hàng.

Dánh giá với từng số lượng sản phẩm trong khung hình

2 sản phẩm trong một khung hình



Hình 22. Confusion matrix của YOLOv5s khi đánh giá với các bức ảnh chứa 2 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|-------|-------|----------------|
| all | 100 | 200 | 0.861 | 0.87 | 0.927 | 0.802 |
| BanhQuyPhoMaiGery | 100 | 6 | 0.897 | 1 | 0.995 | 0.911 |
| BanhQuyYenMachSocolaCosy | 100 | 3 | 0.833 | 1 | 0.995 | 0.995 |
| KemDanhRangColgate | 100 | 2 | 0.373 | 1 | 0.995 | 0.92 |
| MiLyHaoHaoTomChuaCay | 100 | 20 | 0.947 | 0.9 | 0.971 | 0.879 |
| MiXaoHaoHaoTomHanh | 100 | 3 | 1 | 0.659 | 0.995 | 0.858 |
| MiXaoHaoHaoTomXaoChuaNgot | 100 | 3 | 0.747 | 0.99 | 0.746 | 0.689 |
| NuocC2TraXanhViChanh | 100 | 18 | 0.652 | 1 | 0.975 | 0.866 |
| SotChamThitNuongHanQuoc | 100 | 22 | 0.846 | 0.996 | 0.989 | 0.897 |
| SuaChuaVinamilkCoDuong | 100 | 14 | 0.952 | 1 | 0.995 | 0.898 |
| SuaDauNanhFami1Lit | 100 | 13 | 1 | 0 | 0.401 | 0.375 |
| SuaYomostViCam | 100 | 27 | 1 | 0.776 | 0.995 | 0.932 |
| PhoMaiConBoCuoi | 100 | 18 | 1 | 0.734 | 0.984 | 0.756 |
| TrungGaTuoqliQUE | 100 | 17 | 0.961 | 1 | 0.995 | 0.648 |
| XylitolHuongLimeMint | 100 | 20 | 0.723 | 1 | 0.876 | 0.594 |
| TuongOtChinSu | 100 | 14 | 0.978 | 1 | 0.995 | 0.811 |

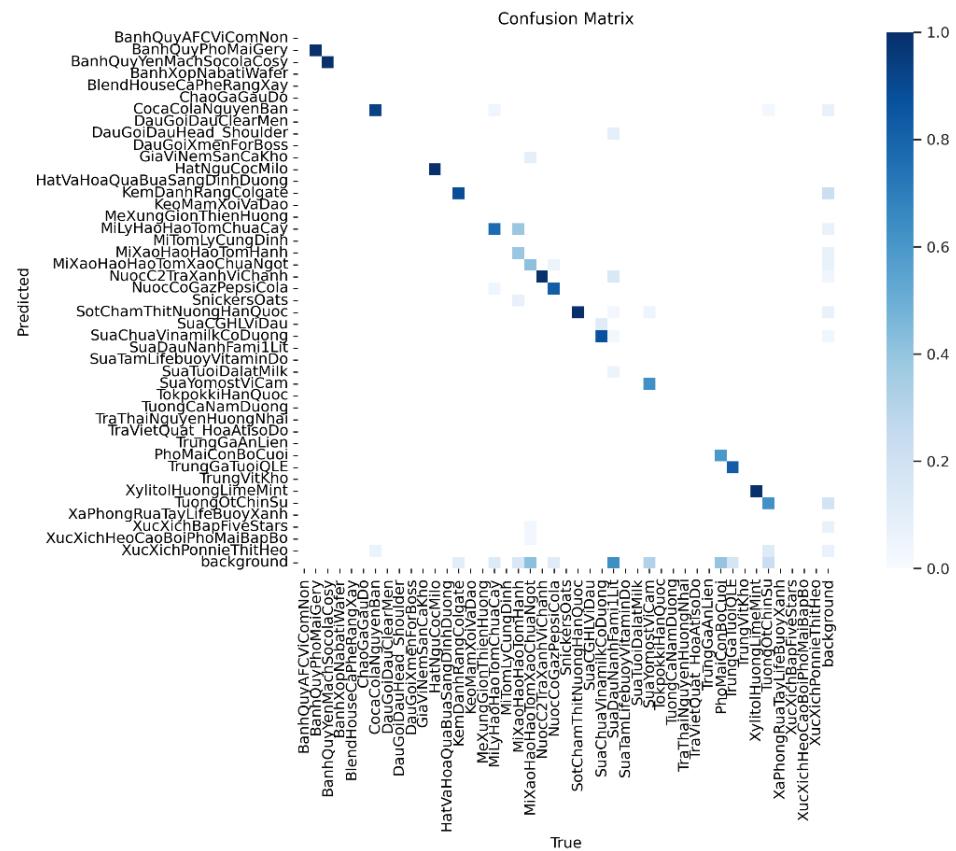
Hình 23. Kết quả các độ đo đánh giá mô hình YOLOv5s với các bức ảnh có 2 đối tượng



Hình 24. 16 trong 100 bức ảnh chứa 2 đối tượng mà YOLOv5s dự đoán

Với các bức ảnh có 2 đối tượng, độ đo Precision, Recall, mAP50, mAP50-95 đều khá tốt (đều trên > 0.8). Tuy nhiên, với trường hợp đánh giá này, vẫn xảy ra các hiện tượng mô hình không detect được/đoán sai nhãn của sản phẩm trong hình (tỷ lệ nhiều nhất là MiXaoHaoHaoTomHanh) và đoán nhãn bị sai (KemDanhRangColgate và NuocC2TraXanhViChanh là hai nhãn có tỷ lệ bị gán sai nhiều nhất).

3 sản phẩm trong một khung hình



Hình 25. Confusion matrix của YOLOv5s khi đánh giá với các bức ảnh chứa 3 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|-------|-------|----------------|
| all | 100 | 300 | 0.865 | 0.786 | 0.914 | 0.763 |
| BanhQuyPhoMaiGery | 100 | 2 | 0.839 | 1 | 0.995 | 0.945 |
| BanhQuyYenMachSocolaCosy | 100 | 9 | 0.955 | 1 | 0.995 | 0.914 |
| CocaColaNguyenBan | 100 | 15 | 0.881 | 0.986 | 0.988 | 0.865 |
| HatNguCocMilo | 100 | 17 | 0.99 | 1 | 0.995 | 0.95 |
| KemDanhRangColgate | 100 | 27 | 0.81 | 0.889 | 0.923 | 0.731 |
| MiLyHaoHaoTomChuaCay | 100 | 22 | 0.782 | 0.818 | 0.897 | 0.751 |
| MiXaoHaoHaoTomHanh | 100 | 13 | 1 | 0.471 | 0.979 | 0.865 |
| MiXaoHaoHaoTomXaoChuaNgot | 100 | 31 | 0.919 | 0.484 | 0.784 | 0.66 |
| NuocCoGazPepsiCola | 100 | 2 | 0.344 | 1 | 0.828 | 0.663 |
| NuocCoGazPepsiCola | 100 | 17 | 0.91 | 0.824 | 0.826 | 0.704 |
| SotChamThitNuongHanQuoc | 100 | 3 | 0.476 | 1 | 0.995 | 0.686 |
| SuaChuaVinamilkCoDuong | 100 | 8 | 0.859 | 1 | 0.995 | 0.941 |
| SuaDauNanhFami1Lit | 100 | 33 | 1 | 0 | 0.47 | 0.41 |
| SuaYomostViCam | 100 | 19 | 1 | 0.638 | 0.995 | 0.945 |
| PhoMaiConBoCuoi | 100 | 5 | 1 | 0.634 | 0.995 | 0.706 |
| TrungGaTuoQL | 100 | 29 | 0.912 | 0.713 | 0.921 | 0.637 |
| XylitolHuongLimeMint | 100 | 8 | 0.936 | 1 | 0.995 | 0.678 |
| TuongOtChinSu | 100 | 40 | 0.965 | 0.694 | 0.867 | 0.676 |

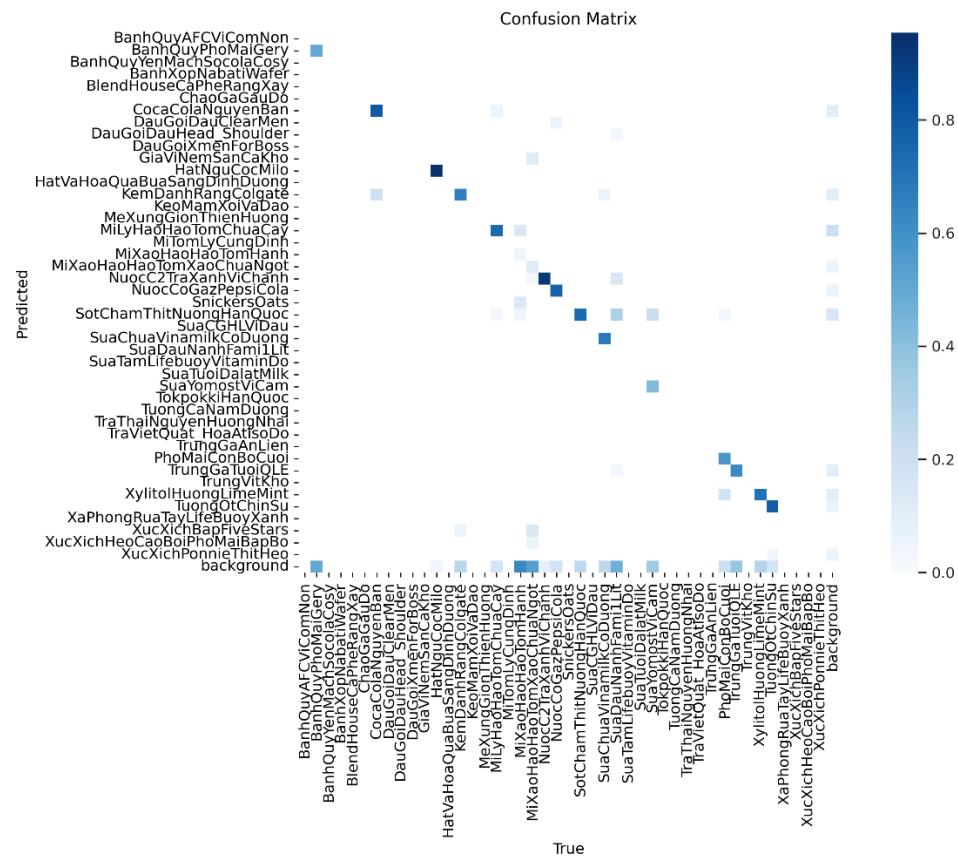
Hình 26. Kết quả các độ đo đánh giá mô hình YOLOv5s với các bức ảnh có 3 đối tượng



Hình 27. 16 trong số 100 bức ảnh chứa 3 đối tượng mà YOLOv5s dự đoán

Với các bức ảnh có 3 đối tượng, các độ đo Recall, mAP50 và mAP50-95 không tốt bằng trường hợp các bức ảnh có 2 đối tượng, trong đó Recall và mAP50-95 < 0.8 . Sản phẩm MiXaoHaoHaoTomHanh vẫn là sản phẩm có tỷ lệ bị đoán sai nhầm và không được phát hiện nhiều nhất. Tương tự với 2 sản phẩm trong 1 khung hình, NuocC2TraXanhViChanh và SotChamThitNuongHanQuoc là 2 nhãn có tỷ lệ bị gán sai nhiều nhất.

4 sản phẩm trong một khung hình



Hình 28. Confusion matrix của YOLOv5s khi đánh giá với các bức ảnh chứa 4 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|----------------------------|--------|-----------|-------|-------|-------|----------------|
| all | 100 | 400 | 0.855 | 0.684 | 0.814 | 0.629 |
| BanhQuyPhoMaiGery | 100 | 2 | 0.916 | 1 | 0.995 | 0.581 |
| CocaColaNguyenBan | 100 | 5 | 0.525 | 1 | 0.662 | 0.556 |
| HatNguCocMilo | 100 | 22 | 1 | 0.99 | 0.995 | 0.816 |
| KemDanhRangColgate | 100 | 18 | 0.688 | 0.856 | 0.845 | 0.656 |
| MiLyHaoHaoTomChuaCay | 100 | 35 | 0.825 | 0.857 | 0.893 | 0.747 |
| MiXiaoHaoHaoTomHanh | 100 | 22 | 1 | 0.112 | 0.672 | 0.515 |
| MiXiaoHaoHaoTomXaoChuaNgot | 100 | 35 | 1 | 0.169 | 0.702 | 0.555 |
| NuocC2TraXanhViChanh | 100 | 12 | 0.47 | 0.917 | 0.827 | 0.717 |
| NuocCoGazPepsiCola | 100 | 17 | 0.934 | 0.837 | 0.893 | 0.655 |
| SotChamThitNuongHanQuoc | 100 | 39 | 0.585 | 0.795 | 0.833 | 0.649 |
| SuaChuaVinamilkCoDuong | 100 | 19 | 1 | 0.726 | 0.874 | 0.684 |
| SuaDauNanhFami1Lit | 100 | 32 | 1 | 0 | 0.278 | 0.237 |
| SuaYomostViCam | 100 | 14 | 1 | 0.487 | 0.941 | 0.838 |
| PhoMaiConBoCuoi | 100 | 33 | 1 | 0.678 | 0.904 | 0.693 |
| TrungGaTuoIQLE | 100 | 29 | 0.863 | 0.654 | 0.818 | 0.495 |
| XylitolHuongLimeMint | 100 | 38 | 0.732 | 0.711 | 0.713 | 0.516 |
| TuongOtChinSu | 100 | 28 | 1 | 0.844 | 0.989 | 0.789 |

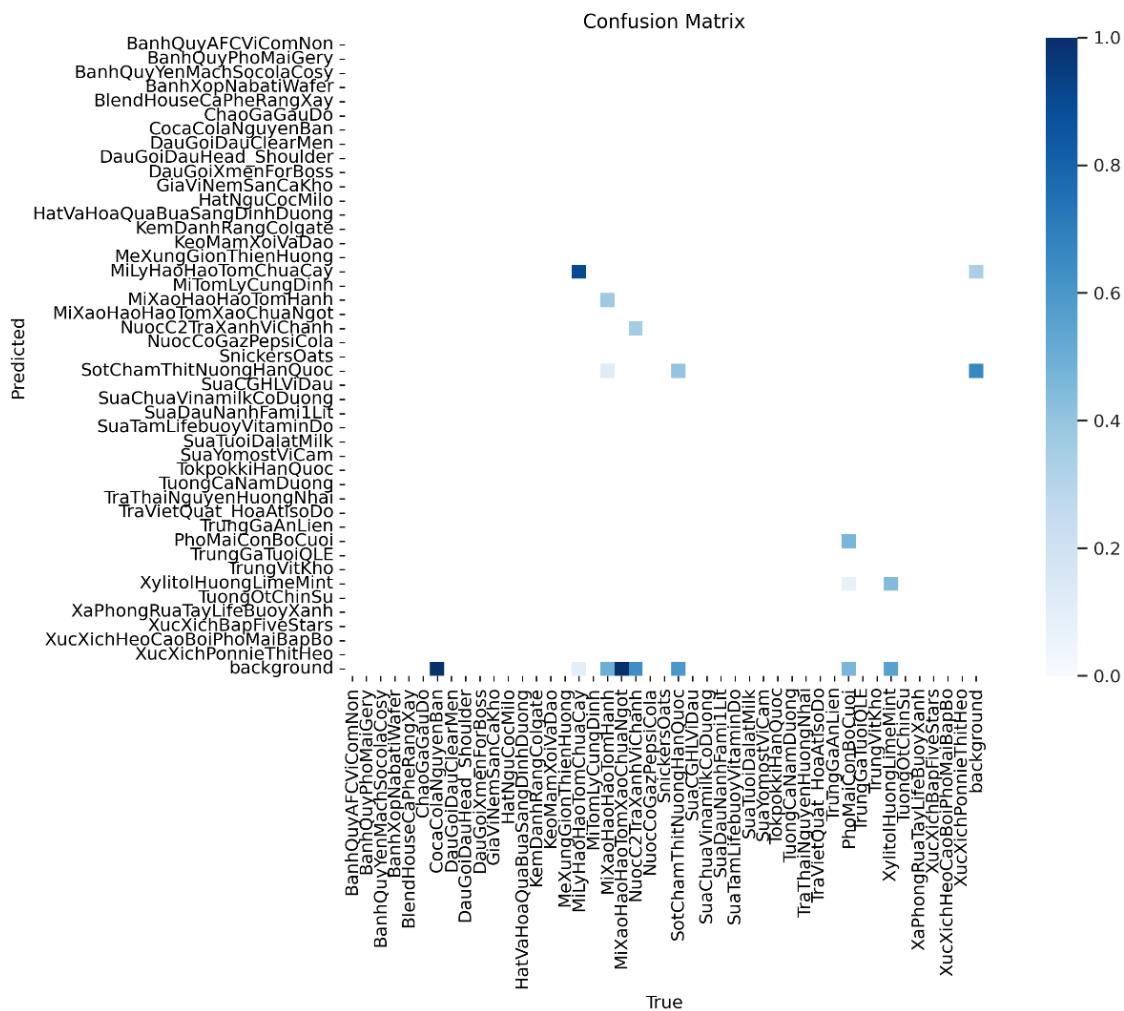
Hình 29. Kết quả các độ đo đánh giá mô hình YOLOv5s với các bức ảnh có 4 đối tượng



Hình 30. 16 trong số 100 bức ảnh chứa 4 đối tượng mà YOLOv5s dự đoán

Có thể thấy, với kết quả các độ đo trên, khi đánh giá mô hình trong trường hợp các bức ảnh có 4 đối tượng cho các chỉ số Presicion, Recall, mAP50 và mAP50-95 thấp hơn với trường hợp 2 và 3 đối tượng. Các sản phẩm như MiXaoHaoHaoTomHanh và MiXaoHaoHaoTomXaoChuaNgot có tỷ lệ không được phát hiện/bị đoán nhầm sai nhiều nhất (< 0.2) và SuaYomostViCam (< 0.5). Trong đó, nhãn NuocC2TraXanhViChanh vẫn là nhãn có tỷ lệ bị gán sai cho các sản phẩm khác nhiều nhất.

5 sản phẩm trong một khung hình



Hình 31. Confusion matrix của YOLOv5s khi đánh giá với các bức ảnh chứa 5 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: | 100% |
|---------------------------|--------|-----------|-------|-------|-------|-----------|------|
| all | 15 | 75 | 0.751 | 0.582 | 0.725 | 0.511 | |
| CocaCola | 15 | 11 | 0.781 | 0.455 | 0.538 | 0.444 | |
| NguyenBan | | | | | | | |
| MiLyHaoHaoTomChuaCay | 15 | 10 | 0.653 | 1 | 0.936 | 0.774 | |
| MiXaoHaoHaoTomHanh | 15 | 8 | 1 | 0.492 | 0.832 | 0.688 | |
| MiXaoHaoHaoTomXaoChuaNgot | 15 | 5 | 1 | 0.375 | 0.703 | 0.55 | |
| NuocC2TraXanhViCanh | 15 | 14 | 0.751 | 0.5 | 0.681 | 0.422 | |
| SotChamThitNuongHanQuoc | 15 | 5 | 0.295 | 0.674 | 0.438 | 0.203 | |
| PhoMaiConBoCuoi | 15 | 13 | 1 | 0.606 | 0.937 | 0.59 | |
| XylitolHuongLimeMint | 15 | 9 | 0.526 | 0.556 | 0.732 | 0.417 | |

Hình 32. Kết quả các độ đo đánh giá mô hình YOLOv5s với các bức ảnh có 5 đối tượng



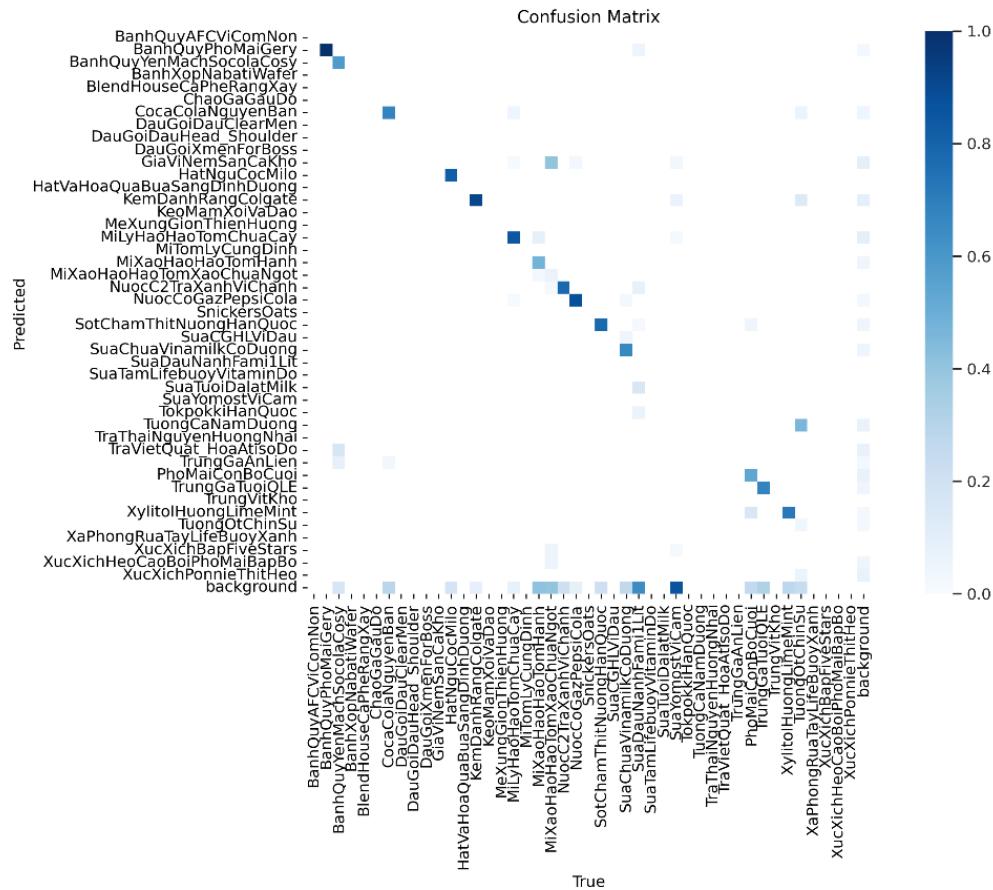
Hình 34. 15 bức ảnh chưa 5 đối tượng mà YOLOv5s dự đoán

Do số lượng các bức ảnh với trường hợp này rất ít (chỉ 15 bức ảnh). Tuy nhiên, có thể thấy chỉ số các độ đo Precision, Recall, mAP50 và mAP50-95 đối với số lượng dữ liệu hiện tại đều không tốt (<0.8). Nhìn vào Confusion matrix, có thể thấy tỷ lệ các đối tượng mà mô hình không phát hiện được là cao. Như đã nói ở trên, do số lượng không đảm bảo nên không có nhiều ý nghĩa để kết luận.

Kết luận với mô hình YOLOv5s: khi đánh giá mô hình với từng số lượng sản phẩm trong bức hình, ta thấy, kết quả cho ra với performance giảm dần như sau: $2 > 3 > 4 > 5$. Hiện tượng bỏ sót đối tượng, gán nhãn sai cho đối tượng vẫn xảy ra nhiều. Mặc dù các độ đo cho kết quả khá tốt (trường hợp 2 và 3 sản phẩm) nhưng nếu xét về mặt ứng dụng thực tế thì điều này tác động xấu đến người bán hàng, không cho kết quả tốt. Do đó, mô hình YOLOv5s của nhóm hiện tại là chưa thể đáp ứng được yêu cầu thực tế. Giải pháp đặt ra có thể cải thiện: train mô hình với các bức ảnh có nhiều đối tượng để tăng khả năng phát hiện đối tượng và thu thập bộ dữ liệu tổng quát hơn cho việc train mô hình.

Đánh giá mô hình YOLOv5n

Đánh giá trên toàn bộ dữ liệu mới thu thập



Hình 35. Confusion matrix của YOLOv5n khi đánh giá với toàn bộ bộ dữ liệu mới

| Class | Images | Instances | P | R | mAP50 | mAP50-95: |
|---------------------------|--------|-----------|-------|--------|-------|-----------|
| all | 315 | 975 | 0.883 | 0.613 | 0.808 | 0.649 |
| BanhQuyPhoMaiGery | 315 | 10 | 0.614 | 1 | 0.968 | 0.836 |
| BanhQuyYenMachSocolaCosy | 315 | 12 | 1 | 0.759 | 0.989 | 0.809 |
| CocaColaNguyenBan | 315 | 31 | 0.644 | 0.71 | 0.719 | 0.613 |
| HatNguCocMilo | 315 | 39 | 1 | 0.844 | 0.978 | 0.841 |
| KemDanhRangColgate | 315 | 47 | 0.577 | 0.851 | 0.817 | 0.681 |
| MiLyHaoHaoTomChuaCay | 315 | 87 | 0.93 | 0.897 | 0.95 | 0.77 |
| MiXaoHaoHaoTomHanh | 315 | 46 | 1 | 0.561 | 0.833 | 0.688 |
| MiXaoHaoHaoTomXaoChuaNgot | 315 | 74 | 0.853 | 0.0787 | 0.419 | 0.348 |
| NuocC2TraXanhViChanh | 315 | 46 | 0.818 | 0.804 | 0.857 | 0.667 |
| NuocCoGazPepsiCola | 315 | 34 | 0.825 | 0.882 | 0.932 | 0.713 |
| SotChamThitNuongHanQuoc | 315 | 69 | 0.896 | 0.797 | 0.868 | 0.695 |
| SuaChuaVinamilkCoDuong | 315 | 41 | 1 | 0.772 | 0.925 | 0.746 |
| SuaDauNanhFami1Lit | 315 | 78 | 1 | 0 | 0.526 | 0.458 |
| SuaYomostViCam | 315 | 60 | 1 | 0 | 0.429 | 0.382 |
| PhoMaiConBoCuoi | 315 | 69 | 1 | 0.606 | 0.984 | 0.799 |
| TrungGaTuoqli | 315 | 75 | 0.915 | 0.68 | 0.822 | 0.541 |
| XylitolHuongLimeMint | 315 | 75 | 0.813 | 0.733 | 0.827 | 0.568 |
| TuongOtChinSu | 315 | 82 | 1 | 0.0573 | 0.706 | 0.533 |

Hình 36. Kết quả các độ đo đánh giá mô hình YOLOv5n trên toàn bộ bộ dữ liệu mới



Hình 37. 16 trong toàn bộ 315 bức ảnh là kết quả dự đoán của mô hình YOLOv5n

Giá trị Recall chung là 0.613 (thấp hơn so với Recall = 0.736 của mô hình YOLOv5s) đã cho thấy có những sản phẩm xuất hiện ở trong khung hình nhưng mô hình không detect ra được. Ngoài SuaDauNanhFami1Lit như mô hình YOLOv5s thì ở YOLOv5n lại có thêm SuaYomostViCam có Recall = 0 thì mô hình đoán sai nhãn/không được mô hình detect ra đối tượng trong toàn bộ dữ liệu đánh giá, vì vậy, ở các phần sau, không cần đánh giá cho hai sản phẩm này. Đặc biệt với TuongOtChinSu thì Recall = 0.0573 (rất nhỏ) cho thấy sản phẩm này bị đoán sai rất nhiều. Các sản phẩm còn lại thì Recall hầu như đều trên 0.5.

Giá trị Precision chung là 0.883 (tốt hơn một chút so với Precision = 0.87 của YOLOv5s). Xét đến từng class, thì có đến 8 sản phẩm được mô hình đoán nhãn đúng hết cho tất cả các đối tượng mà mô hình detect được với Precision = 1 (BanhQuyYenMachSocolaCosy, HatNguCocMilo, MiXaoHaoHaoTomHanh, SuaChuaVinamilkCoDuong, SuaDauNanhFami1Lit, SuaYomostViCam, PhoMaiConBoCuoi, TuongOtChinSu). Dù vậy, vẫn còn những đối tượng mà mô hình đoán sai nhãn. 3 nhãn bị gán sai nhiều nhất là BanhQuyPhoMaiGery, CocaColaNguyenBan và

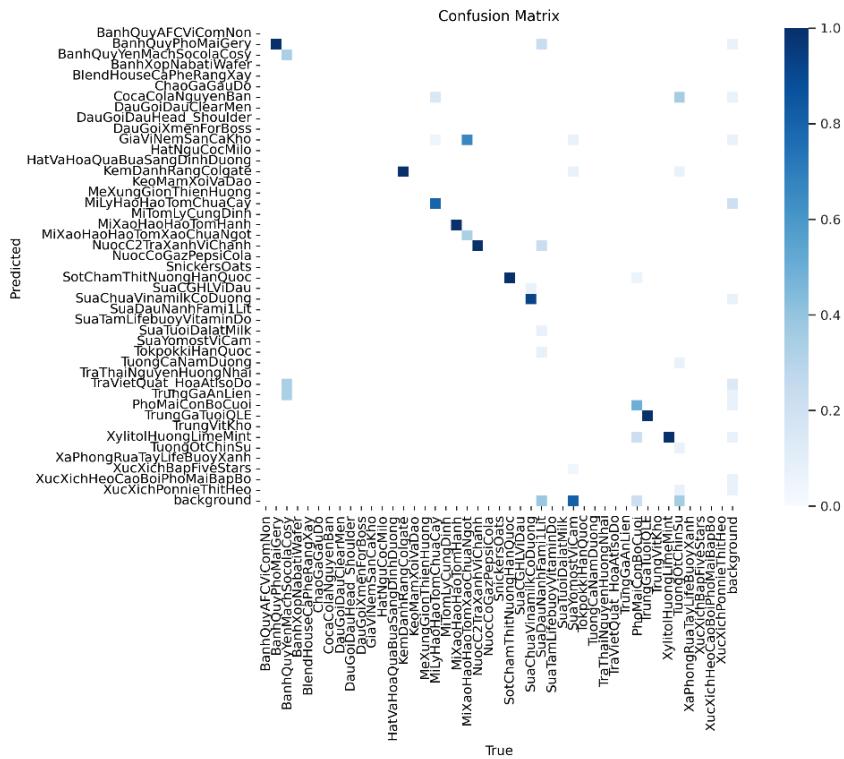
KemDanhRangColgate. Các đối tượng còn lại thì mô hình đoán nhãn khá tốt (Precision > 0.8).

Giá trị mAP50 trên 0.8 nhưng mAP50-95 không tốt ($0.649 < 0.8$):

Việc mô hình bỏ sót nhiều sản phẩm có xuất hiện trong khung hình và đoán sai nhãn của sản phẩm là không chấp nhận được trong thực tế, vì điều này làm ảnh hưởng xấu đến lợi ích của người bán hàng.

Dánh giá với từng số lượng sản phẩm trong khung hình

2 sản phẩm trong một khung hình



Hình 38. Confusion matrix của YOLOv5n khi đánh giá với các bức ảnh chứa 2 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|--------|-------|----------------|
| all | 100 | 200 | 0.878 | 0.707 | 0.89 | 0.761 |
| BanhQuyPhoMaiGery | 100 | 6 | 0.597 | 1 | 0.995 | 0.933 |
| BanhQuyYenMachSocolaCosy | 100 | 3 | 1 | 0.501 | 0.995 | 0.886 |
| KemDanhRangColgate | 100 | 2 | 0.344 | 1 | 0.995 | 0.945 |
| MiLyHaoHaoTomChuaCay | 100 | 20 | 0.968 | 0.95 | 0.973 | 0.871 |
| MiXaoHaoHaoTomHanh | 100 | 3 | 0.828 | 1 | 0.995 | 0.88 |
| MiXaoHaoHaoTomXaoChuaNgot | 100 | 3 | 1 | 0.52 | 0.705 | 0.701 |
| NuocC2TraXanhViChanh | 100 | 18 | 0.821 | 1 | 0.984 | 0.843 |
| SotChamThitNuongHanQuoc | 100 | 22 | 0.931 | 1 | 0.995 | 0.882 |
| SuaChuaVinamilkCoDuong | 100 | 14 | 0.965 | 1 | 0.995 | 0.844 |
| SuaDauNanhFami1Lit | 100 | 13 | 1 | 0 | 0.494 | 0.446 |
| SuaYomostViCam | 100 | 27 | 1 | 0 | 0.549 | 0.492 |
| PhoMaiConBoCuoI | 100 | 18 | 1 | 0.598 | 0.995 | 0.795 |
| TrungGaTuoqliE | 100 | 17 | 0.937 | 0.941 | 0.955 | 0.65 |
| XylitolHuongLimeMint | 100 | 20 | 0.786 | 1 | 0.963 | 0.662 |
| TuongOtChinSu | 100 | 14 | 1 | 0.0909 | 0.757 | 0.578 |

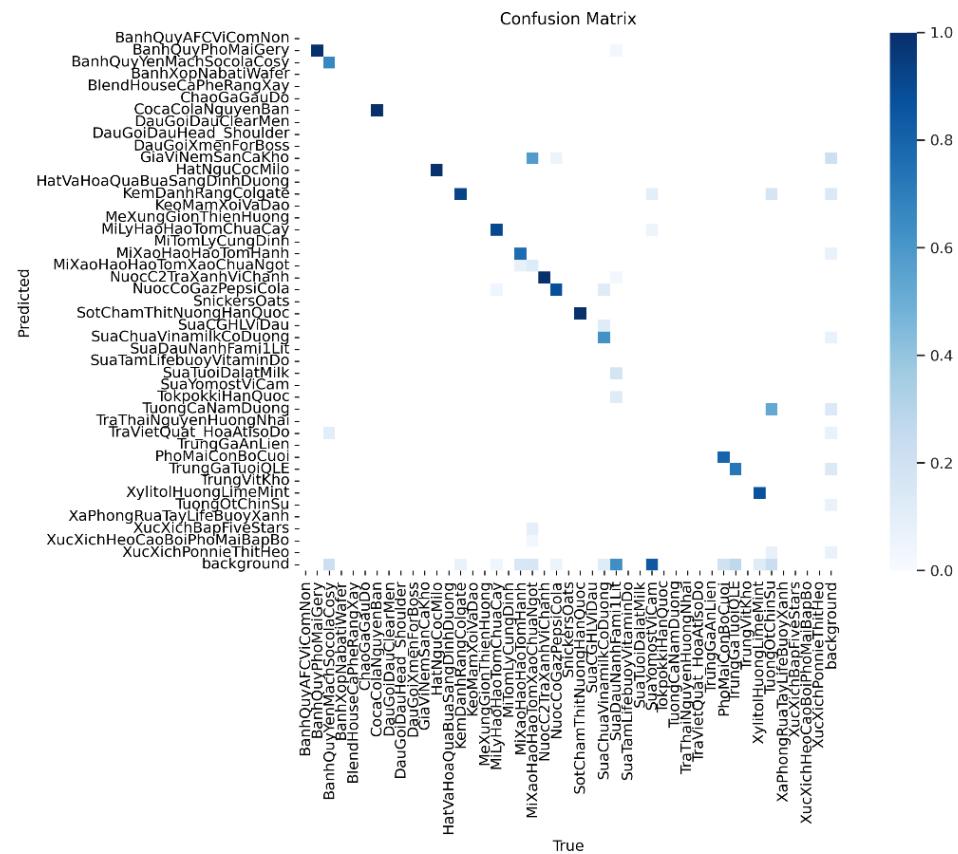
Hình 39. Kết quả các độ đo đánh giá mô hình YOLOv5n với các bức ảnh có 2 đối tượng



Hình 40. 16 trong 100 bức ảnh chứa 2 đối tượng mà YOLOv5n dự đoán

Với các bức ảnh có 2 đối tượng, các độ đo Precision, Recall, mAP50 và mAP50-95 cũng không được tốt (vì có Recall và mAP50-95 < 0.7). Với trường hợp của SuaDauNanhFami1Lit và SuaYomostViCam thì mô hình không detect/đoán đúng được trường hợp nào. Trường hợp của TuongOtChinSu thì mô hình detect/đoán đúng được rất ít. KemDanhRangColgate là nhãn có tỷ lệ bị gán sai nhiều nhất.

3 sản phẩm trong một khung hình



Hình 41. Confusion matrix của YOLOv5n khi đánh giá với các bức ảnh chứa 3 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|--------|-------|----------------|
| all | 100 | 300 | 0.908 | 0.7 | 0.885 | 0.733 |
| BanhQuyPhoMaiGery | 100 | 2 | 0.681 | 1 | 0.995 | 0.945 |
| BanhQuyYenMachSocolaCosy | 100 | 9 | 1 | 0.754 | 0.995 | 0.806 |
| CocaColaNguyenBan | 100 | 15 | 0.957 | 1 | 0.995 | 0.866 |
| HatNguCocMilo | 100 | 17 | 0.968 | 1 | 0.995 | 0.958 |
| KemDanhRangColgate | 100 | 27 | 0.671 | 0.83 | 0.82 | 0.695 |
| MiLyHaoHaoTomChuaCay | 100 | 22 | 0.961 | 0.909 | 0.986 | 0.786 |
| MiXaoHaoHaoTomHanh | 100 | 13 | 1 | 0.852 | 0.99 | 0.849 |
| MiXaoHaoHaoTomXaoChuaNgot | 100 | 31 | 0.813 | 0.142 | 0.572 | 0.49 |
| NuocC2TraXanhViChanh | 100 | 2 | 0.575 | 1 | 0.995 | 0.721 |
| NuocCoGazPepsiCola | 100 | 17 | 0.961 | 0.882 | 0.891 | 0.771 |
| SotChamThitNuongHanQuoc | 100 | 3 | 0.838 | 1 | 0.995 | 0.896 |
| SuaChuaVinamilkCoDuong | 100 | 8 | 1 | 0.814 | 0.995 | 0.871 |
| SuaDauNanhFami1Lit | 100 | 33 | 1 | 0 | 0.667 | 0.587 |
| SuaYomostViCam | 100 | 19 | 1 | 0 | 0.506 | 0.46 |
| PhoMaiConBoCuoi | 100 | 5 | 1 | 0.803 | 0.995 | 0.69 |
| TrungGaTuoIQUE | 100 | 29 | 0.913 | 0.69 | 0.833 | 0.585 |
| XylitolHuongLimeMint | 100 | 8 | 1 | 0.9 | 0.995 | 0.679 |
| TuongOtChinSu | 100 | 40 | 1 | 0.0291 | 0.718 | 0.538 |

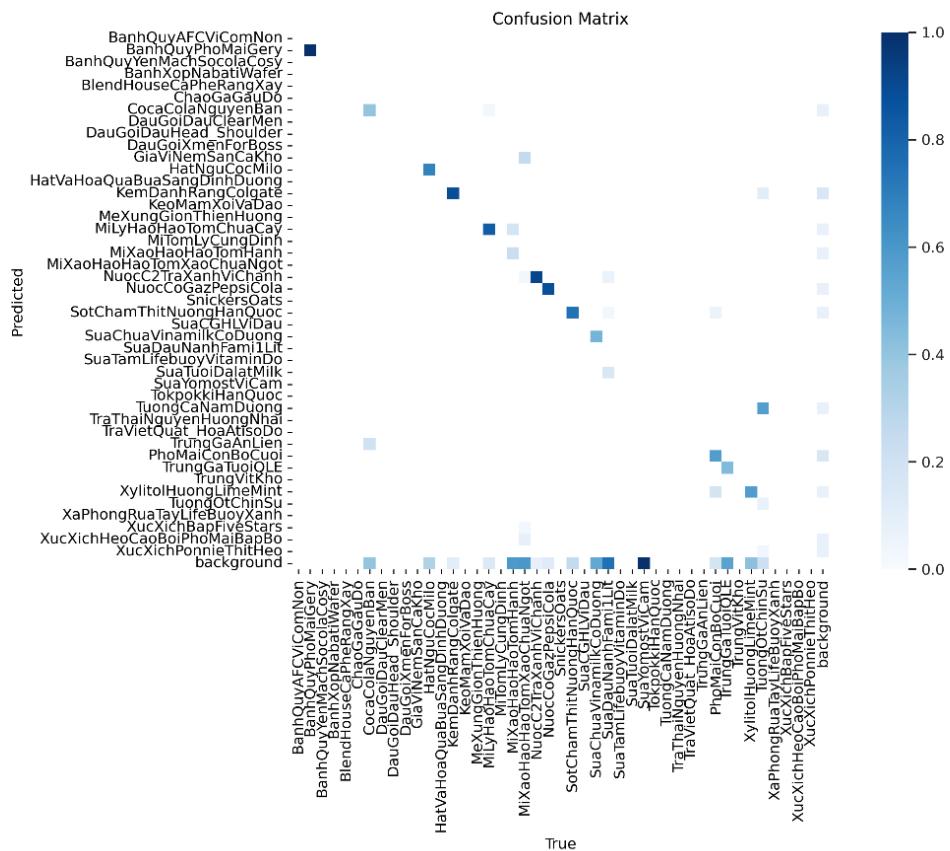
Hình 42. Kết quả các độ đo đánh giá mô hình YOLOv5n với các bức ảnh có 3 đối tượng



Hình 43. 16 trong 100 bức ảnh chứa 3 đối tượng mà YOLOv5n dự đoán

Với các bức ảnh có 3 đối tượng, các độ đo Precision, Recall, mAP50 và mAP50-95 tương đối tốt (≥ 0.7). Cũng giống như trường hợp bức ảnh có 2 đối tượng, với trường hợp của SuaDauNanhFami1Lit và SuaYomostViCam thì mô hình không detect/đoán đúng được trường hợp nào. Trường hợp của TuongOtChinSu thì mô hình detect/đoán đúng được rất ít trường hợp. NuocC2ViTraChanh, KemDanhRangColgate và BanhQuyPhoMaiGery là các nhãn có tỷ lệ bị đoán sai nhiều nhất. Các đối tượng còn lại thì đoán nhãn khá tốt ($P \geq 0.8$).

4 sản phẩm trong một khung hình



Hình 44. Confusion matrix của YOLOv5n khi đánh giá với các bức ảnh chứa 4 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|--------|-------|----------------|
| all | 100 | 400 | 0.843 | 0.602 | 0.783 | 0.586 |
| BanhQuyPhoMaiGery | 100 | 2 | 0.739 | 1 | 0.995 | 0.425 |
| CocaColaNguyenBan | 100 | 5 | 0.553 | 0.8 | 0.773 | 0.646 |
| HatNguCocMilo | 100 | 22 | 1 | 0.931 | 0.968 | 0.729 |
| KemDanhRangColgate | 100 | 18 | 0.589 | 0.833 | 0.825 | 0.659 |
| MiLyHaoHaoTomChuaCay | 100 | 35 | 0.817 | 0.857 | 0.913 | 0.706 |
| MiXaoHaoHaoTomHanh | 100 | 22 | 1 | 0.421 | 0.756 | 0.634 |
| MiXaoHaoHaoTomXaoChuaNgot | 100 | 35 | 0.747 | 0.0286 | 0.389 | 0.291 |
| NuocC2TraXanhViChanh | 100 | 12 | 0.71 | 0.917 | 0.917 | 0.727 |
| NuocCoGazPepsiCola | 100 | 17 | 0.796 | 0.916 | 0.972 | 0.669 |
| SotChamThitNuongHanQuoc | 100 | 39 | 0.772 | 0.795 | 0.861 | 0.649 |
| SuaChuaVinamilkCoDuong | 100 | 19 | 0.926 | 0.684 | 0.838 | 0.623 |
| SuaDauNanhFami1Lit | 100 | 32 | 1 | 0 | 0.528 | 0.441 |
| SuaYomostViCam | 100 | 14 | 1 | 0 | 0.364 | 0.319 |
| PhoMaiConBoCuoi | 100 | 33 | 1 | 0.709 | 0.975 | 0.84 |
| TrungGaTuoqliE | 100 | 29 | 0.944 | 0.579 | 0.737 | 0.467 |
| XylitolHuongLimeMint | 100 | 38 | 0.734 | 0.654 | 0.756 | 0.551 |
| TuongOtChinSu | 100 | 28 | 1 | 0.104 | 0.745 | 0.577 |

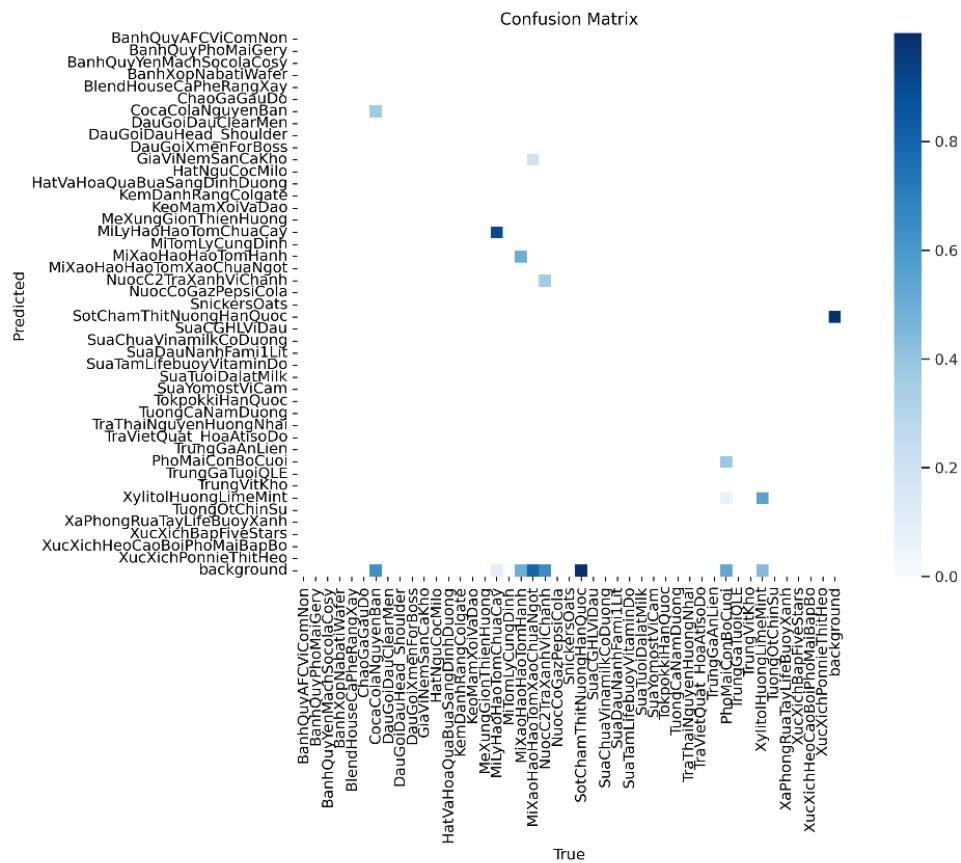
Hình 45. Kết quả các độ đo đánh giá mô hình YOLOv5n với các bức ảnh có 4 đối tượng



Hình 46. 16 trong 100 bức ảnh chứa 4 đối tượng mà YOLOv5n dự đoán

Với các bức ảnh có 4 đối tượng, các độ đo Precision, Recall, mAP50 và mAP50-95 đã thấp hơn so với bức ảnh chứa 3 đối tượng (≥ 0.6). Trường hợp của SuaDauNanhFami1Lit và SuaYomostViCam thì mô hình không detect/đoán đúng được trường hợp nào. Trường hợp của TuongOtChinSu và MiXaoHaoHaoTomChuaNgot thì mô hình detect/đoán đúng được rất ít trường hợp. KemDanhRangColgate và CocaColaViNguyenBan là các nhãn có tỷ lệ bị đoán sai nhiều nhất. Các đối tượng còn lại thì đoán nhãn khá tốt ($P \geq 0.7$).

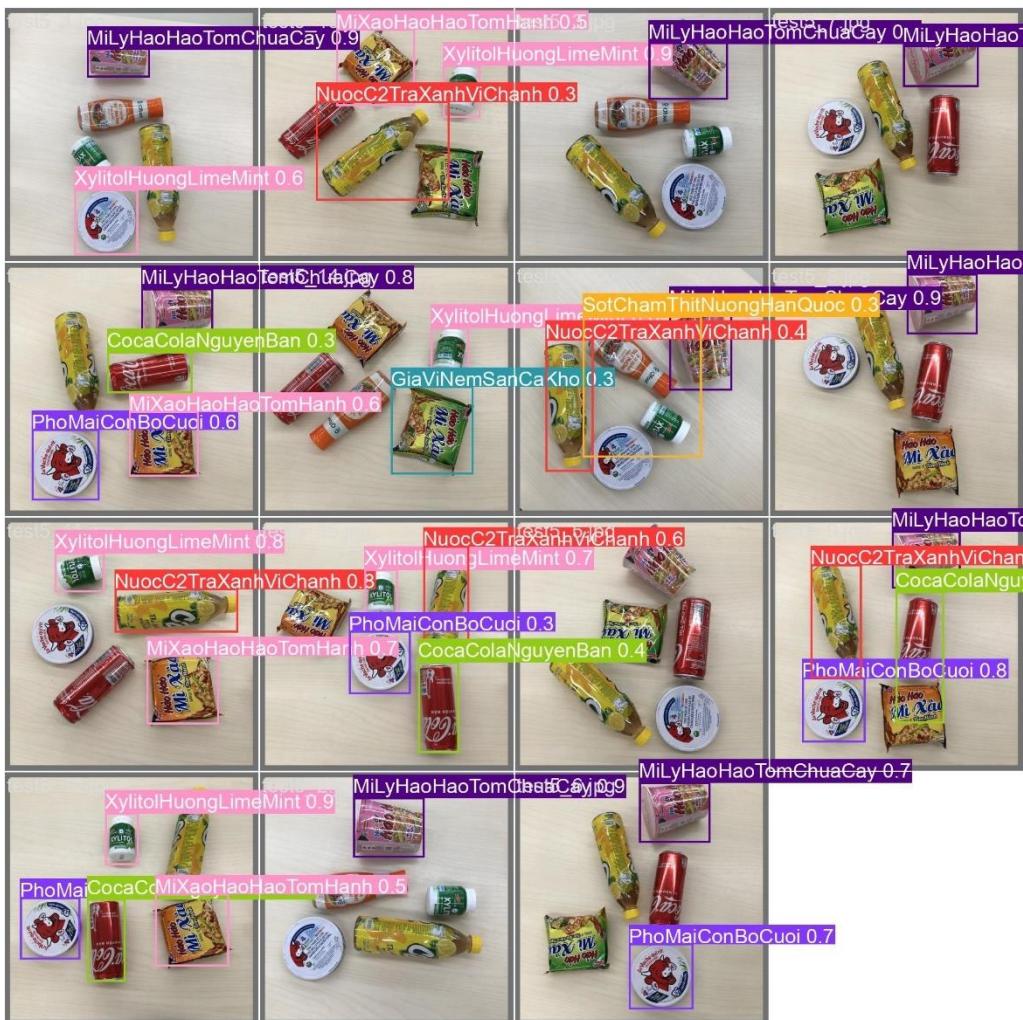
5 sản phẩm trong một khung hình



Hình 47. Confusion matrix của YOLOv5n khi đánh giá với các bức ảnh chứa 5 đối tượng

| Class | Images | Instances | P | R | mAP50 | mAP50-95: 100% |
|---------------------------|--------|-----------|-------|-------|-------|----------------|
| all | 15 | 75 | 0.754 | 0.502 | 0.643 | 0.476 |
| CocaColaNguyenBan | 15 | 11 | 0.757 | 0.364 | 0.488 | 0.393 |
| MilyHaoHaoTomChuaCay | 15 | 10 | 0.828 | 0.9 | 0.978 | 0.799 |
| MiXaoHaoHaoTomHanh | 15 | 8 | 1 | 0.747 | 0.863 | 0.655 |
| MiXaoHaoHaoTomXaoChuaNgot | 15 | 5 | 1 | 0 | 0.245 | 0.18 |
| NuocC2TraXanhViChanh | 15 | 14 | 0.581 | 0.596 | 0.764 | 0.472 |
| SotChamThitNuongHanQuoc | 15 | 5 | 0.153 | 0.2 | 0.105 | 0.0323 |
| PhoMaiConBoCuoi | 15 | 13 | 1 | 0.657 | 0.968 | 0.779 |
| XylitolHuongLimeMint | 15 | 9 | 0.717 | 0.556 | 0.734 | 0.5 |

Hình 48. Kết quả các độ đo đánh giá mô hình YOLOv5n với các bức ảnh có 5 đối tượng



Hình 49. 15 bức ảnh chứa 5 đối tượng mà YOLOv5n dự đoán

Do số lượng các bức ảnh với trường hợp này rất ít (chỉ 15 bức ảnh). Tuy nhiên, có thể thấy chỉ số các độ đo Precision, Recall, mAP50 và mAP50-95 đối với số lượng dữ liệu hiện tại đều không tốt (< 0.8), thậm chí độ đo mAP50-95 còn < 0.5 . Nhìn vào Confusion matrix, có thể thấy tỷ lệ các đối tượng mà mô hình không phát hiện được là cao. Như đã nói ở trên, do số lượng không đảm bảo nên không có nhiều ý nghĩa để kết luận.

Kết luận với mô hình YOLOv5n: Tương tự như mô hình YOLOv5s, khi đánh giá mô hình với từng số lượng sản phẩm khác nhau trong bức hình, kết quả cho ra với performance giảm dần như sau: $2 > 3 > 4 > 5$. Hiện tượng bỏ sót đối tượng, gán nhãn sai cho đối tượng vẫn xảy ra nhiều. Mặc dù các độ đo cho kết quả khá tốt (trường hợp 2 và 3 sản phẩm) nhưng nếu xét về mặt ứng dụng thực tế thì điều này tác động xấu đến người bán hàng, không cho kết quả tốt. Do đó, mô hình YOLOv5n của nhóm hiện tại là cũng chưa thể đáp ứng được yêu cầu thực tế.

Nhận xét 2 mô hình khi đánh giá với bộ dữ liệu mới thu

Nhìn chung, khi đánh giá trên toàn bộ bộ dữ liệu hay khi đánh giá với trường hợp tốt nhất của 2 mô hình (là trường hợp có 2 sản phẩm trong một khung hình) thì YOLOv5s có phần tốt hơn YOLOv5n (tốt hơn ở các độ đo Recall, mAP50, mAP50-95). Cả hai mô hình đều thể hiện không tốt khi để xảy ra hiện tượng đoán nhãn sai/không phát hiện được đối tượng trong bức hình (điều này xảy ra nhiều hơn ở mô hình YOLOv5n). Giải pháp có thể cải thiện độ chính xác của mô hình mà nhóm đưa ra: thu thập và sử dụng một bộ dữ liệu đa dạng và tổng quát hơn, huấn luyện mô hình với các trường hợp nhiều sản phẩm trong bức hình. Điểm chung của cả 2 mô hình là càng ít sản phẩm xuất hiện thì độ chính xác tốt hơn, điều này có thể do cùng đặt khoảng cách camera cách sản phẩm 40 ± 5 cm, nên khi ít sản phẩm thì khoảng cách các sản phẩm trong bức hình xa nhau hơn, dẫn đến nhận diện tốt hơn.

3. Giải thích một số khái niệm có liên quan

Bounding box [12]

Bounding box là một hình chữ nhật được vẽ bao quanh đối tượng nhằm xác định đối tượng. Các thông tin quan trọng để xác định vị trí của bounding box trong hình ảnh được gán nhãn: Tọa độ tâm (x,y) và chiều rộng, chiều cao (w, h) của bounding box. Các thông số này thường được scale về một miền giá trị từ 0 đến 1.

IoU (Intersection over Union)

Là chỉ số đánh giá được sử dụng để đo độ chính xác của một Object detector trên tập dữ liệu cụ thể. Được tính toán dựa trên vùng trùng nhau giữa ground truth bounding box (là bounding box mà được chúng em xác định trước trong quá trình gán nhãn) và predicted bounding box (là bounding box mà mô hình đoán là có chứa đối tượng trong đó).

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

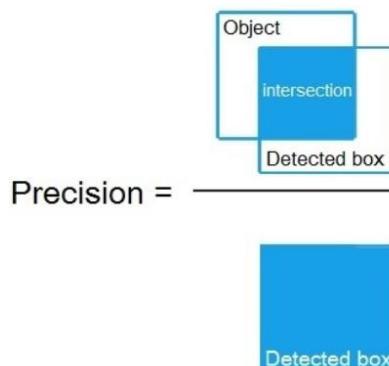
Hình 50. Công thức tính IoU [13]

Nguồn của IoU, thông số này quyết định xem với IoU đạt tối thiểu là bao nhiêu thì Bounding box được dự đoán được xem là một True Positive (nghĩa là Bounding box thực sự có chứa đối tượng/một phần đối tượng)

AP [13]

Trước tiên, chúng ta nói đến Precision và Recall

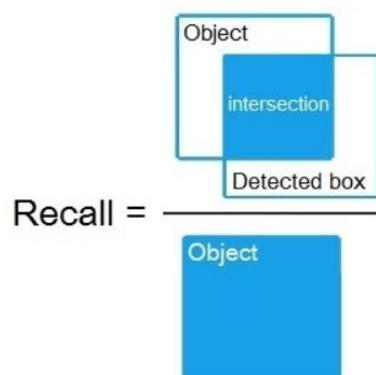
Cách tính Precision



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

Hình 51. Cách tính Precision

Cách tính Recall



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

Hình 52. Cách tính Recall

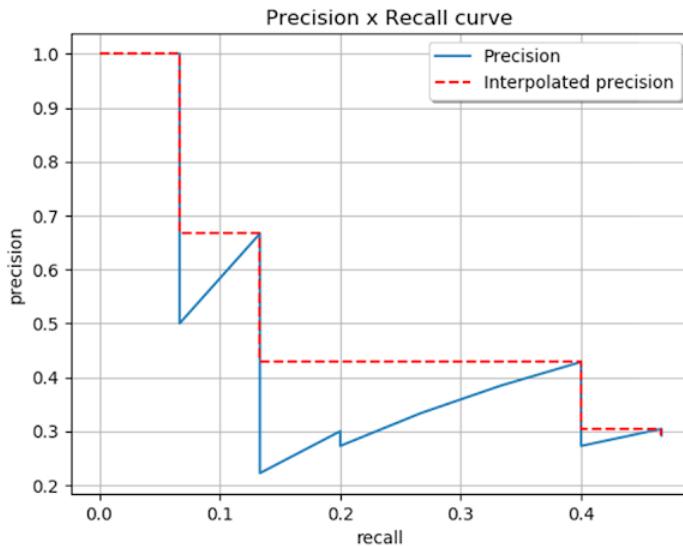
Chú thích:

- True Positive (TP): Khi $\text{IoU} \geq \text{IoU Threshold}$
- False Positive (FP): Khi $\text{IoU} < \text{IoU Threshold}$
- False Negative (FN): Bounding box của đối tượng không được mô hình detect

Average Precision (AP)

Từ Precision và Recall tìm được ở phía trên, ta có thể vẽ được PR Curve (biểu diễn Precision theo Recall) cho mỗi class riêng biệt.

AP chính là phần diện tích dưới đường PR Curve. Việc tính TP và FP còn phụ thuộc vào ngưỡng của IoU (IoU Threshold). Người ta thường ký hiệu AP@ a với a là giá trị IoU Threshold.



Hình 53. Minh họa PR Curve

mAP [13]

Sau khi tính được AP của từng class, ta tính được mAP. mAP là trung bình cộng AP của tất cả các classes trên.

Các ký hiệu: mAP@ a (hoặc mAP a) với a là giá trị của IoU Threshold. mAP@ $a:b$ (hoặc mAP $a:b$) với các giá trị trong miền $a:b$ là IoU Threshold, mAP@ $a:b$ là giá trị trung bình của các mAP của các ngưỡng từ a đến b .

VI. Deploy mô hình lên ứng dụng demo

Nhóm sử dụng thư viện streamlit để deploy mô hình, Streamlit là một thư viện Python mã nguồn mở giúp dễ dàng tạo và chia sẻ các ứng dụng web đẹp, tùy chỉnh cho máy học và khoa học dữ liệu. Nhóm sử dụng máy tính cá nhân làm server và ứng dụng được xây dựng trên môi trường website cho phép nhiều client truy cập và sử dụng. Do ứng dụng chạy trên môi trường local nên chỉ cho phép các thiết bị cùng mạng Lan truy cập. Ứng dụng có thể được truy cập bằng port 8501 của Ipv4 address của máy chủ.

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://172.16.0.2:8501>

Các bước deploy ứng dụng:

1. Đầu tiên training mô hình trên Colab. Sau khi thu được các file trọng số (weight) (best.pt/last.pt) ta tiến hành tải file weight về máy.
2. Xây dựng hàm predict từ model đã train. Nhóm sử dụng thư viện torch 2.0.1+cu117 và torchvision 0.15.2+cu117 để load mô hình bằng hàm torch.hub.load() để load file trọng số đã train vào model.
3. Dùng streamlit.file_upload() cho việc upload file ảnh cần detect vào ứng dụng và lưu vào folder data/uploads
4. Sau khi detect, file output được lưu vào folder data/outputs và hiển thị đồng thời lên ứng dụng.

Do sử dụng 2 model yolov5s và yolov5n training, nên ứng dụng sử dụng cả hai model để cùng detect, kết quả dùng để so sánh độ tin cậy khi phân loại và thời gian chạy của 2 model.

VII. Ứng dụng và hướng phát triển

Ứng dụng:

- Ứng dụng có thể dùng trong việc detect sản phẩm trên kệ thanh toán, mở ra một hướng phát triển cho công nghệ siêu thị không thu ngân.

Hướng phát triển:

- Dữ liệu của nhóm vẫn còn nhỏ và hạn chế về số lượng sản phẩm. Nhóm cần phải thu thập dữ liệu và cập nhật liên tục để phù hợp với các trung tâm thương mại.
- Quá trình thu thập dữ liệu phải được thực hiện sao cho dữ liệu clean nhất có thể.
- Model của nhóm hiện chỉ nhận dạng được các vật thể được chụp riêng lẻ và chỉ nhận diện được số ít trường hợp mà nhiều vật thể đê gần nhau. Do đó, nhóm cần phải cho model training với các trường hợp đặc biệt như: nhiều sản phẩm đê gần nhau, các sản phẩm đè lên nhau, ...
- Khi dữ liệu là đủ lớn thì cần phải thay đổi mô hình sang các phiên bản phù hợp hơn nhằm đảm bảo về độ chính xác và thời gian trong việc nhận dạng sản phẩm.
- Model của nhóm hiện chỉ cho biết tên sản phẩm nên trong tương lai sẽ mở rộng để cho biết thêm giá cả của sản phẩm.

VIII. Mục tham khảo

| | |
|------|---|
| [1] | https://blog.roboflow.com/retail-store-item-detection-using-yolov5/ |
| [2] | https://www.researchgate.net/publication/346856840 Deep Learning for Retail Product Recognition Challenges and Techniques |
| [3] | https://github.com/heartexlabs/labelImg |
| [4] | https://www.stereolabs.com/blog/performance-of-yolo-v5-v7-and-v8/ |
| [5] | https://www.augmentedstartups.com/blog/yolov8-vs-yolov5-choosing-the-best-object-detection-model#:~:text=YOLOv5%20is%20easier%20to%20use,specific%20needs%20of%20your%20application |
| [6] | https://learnopencv.com/performance-comparison-of-yolo-models/ |
| [7] | https://www.learnwitharobot.com/p/yolov5-vs-yolov6-vs-yolov7 |
| [8] | https://towardsdatascience.com/yolov5-compared-to-faster-rcnn-who-wins-a771cd6c9fb4 |
| [9] | https://www.researchgate.net/publication/363824867 A comparative study of YOLOv5 models performance for image localization and classification |
| [10] | https://docs.ultralytics.com/yolov5/tutorials/architecture_description/?fbclid=IwAR1u7LvoXQxZJVdV2Zu1lQoJpVQsLi-7jqugdTZiHMj3O3iPr2roQvJok5c#2-data-augmentation-techniques |
| [11] | https://github.com/ultralytics/yolov5 |
| [12] | https://phamdinhkhanh.github.io/2019/09/29/OverviewObjectDetection.html |
| [13] | https://blogcuabuicaodoanh.wordpress.com/2020/02/22/mean-average-precision-map-trong-bai-toan-object-detection/ |

VIII. Cập nhật sau khi vấn đáp

1. Sửa lại tiêu đề con, xóa nội dung cũ và sửa lại thành nội dung mới [tại đây](#)

Tiêu đề con cũ: Đánh giá mô hình với các trường hợp không có trong bộ dữ liệu thu thập của nhóm. Được sửa thành tiêu đề mới: Đánh giá mô hình trong trường hợp mô hình phải nhận diện với nhiều sản phẩm cùng lúc.

Sau khi được góp ý, nhóm làm lại toàn bộ nội dung đánh giá mô hình với trường hợp phải nhận diện nhiều sản phẩm cùng lúc.

2. Bổ sung thêm phần Giải thích một số khái niệm liên quan [tại đây](#)