

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

MÔN CS114 – MÁY HỌC

Đề tài: NHẬN BIẾT SẢN PHẨM BÁN LẺ

Giảng viên : **Phạm Nguyễn Trường An**
Lê đình Duy

Nhóm thực hiện:

Ngô Xuân Cường-21520663

Thi Vĩnh Huy-21522165

Trần Thanh Hà-21521749

Phi Quang Đạt-21520711

TP. HỒ CHÍ MINH, Tháng 6 Năm 2023

MỤC LỤC

1. Tổng quan về đề án	4
2. Các nghiên cứu đi trước	5
3. Xây dựng dataset.....	6
4. Thông số dữ liệu	8
5. Training và đánh giá model.....	10
6. Hướng cải thiện mô hình	20
7. Cập nhật sau khi vấn đáp.....	21
TÀI LIỆU THAM KHẢO	25

Link đến data:

https://drive.google.com/drive/folders/1ZdgNJcMONdbsrKfKnNCIHC_d5NJDA2Oqp?fbclid=IwAR3K-7QXGO9sD67-dINumPEsJ0QKJsx6saf8X9LbZcbX3Yv2C97aT9xO0c

Link đến Google Colab:

https://colab.research.google.com/drive/13fSAHJYmh092DAXCgm_mbWPyVtweP7dk?usp=sharing&fbclid=IwAR2GtQy4-gFb_m1PJmW5vS2ccnJs9Uwv08ldcrEY2htYRcMzsLNSv73QIG4

Link đến github:

<https://github.com/thivinhhuu/CS114.N21.KHCL?fbclid=IwAR1jgzJUxcRhxy8efjwnUVilAdYQhiydvZ9tiP7cas3CifqoqvzY3x8yoc>

[illegible]

TPHCM, Ngày ... Tháng ... Năm ...

Người nhận xét

(Ký tên)

1. Tổng quan về đề án

a) Tóm tắt đề án

- Mô hình siêu thị không thu ngân trong bối cảnh thanh toán điện tử phát triển mạnh là một xu hướng mới trong ngành bán lẻ và dịch vụ. Trong mô hình này, khách hàng có thể tự thanh toán sản phẩm mà họ chọn mua thông qua các phương thức thanh toán điện tử như ví điện tử, thẻ tín dụng, hoặc ứng dụng di động. Với bài đề án cuối kì này, chúng em muốn thử sức để thực hiện hoá lại các bước cần thiết của mô hình Mô Hình Siêu Thị Không Thu Ngân - Bài Toán Nhận Diện Sản Phẩm Thương Mại
- Mô hình siêu thị không thu ngân được nhóm chúng em thực hiện mang lại những lợi ích sau:
 - Tăng tốc độ thanh toán: Đẩy nhanh quy trình thanh toán tại cửa hàng và giảm thời gian chờ đợi xếp hàng cho khách hàng.
 - Giảm chi phí nhân lực: Không cần có nhân viên thu ngân tại mỗi điểm thu ngân, chúng ta sử dụng công nghệ tự động để thực hiện quá trình thanh toán. Điều này giúp tối ưu hóa tài nguyên nhân lực và giảm chi phí vận hành cho cửa hàng.
 - Nâng cao trải nghiệm người dùng: Tạo ra một trải nghiệm mua sắm tiện lợi và linh hoạt cho khách hàng. Họ có thể tự do lựa chọn sản phẩm, kiểm tra giá và thanh toán mà không cần phải xếp hàng hoặc chờ đợi nhân viên thu ngân.
 - Đáng tin cậy và chính xác: Tự động nhận diện sản phẩm và quá trình thanh toán, giúp giảm thiểu sai sót trong việc tính toán tổng giá trị đơn hàng và ghi nhận sản phẩm. Hệ thống tự động đảm bảo tính chính xác và đáng tin cậy của quá trình thanh toán.

b) Mô tả Input và Output của bài toán

- INPUT:

- Video thời lượng tối đa 1 phút các sản phẩm thương mại
- Các sản phẩm trong clip, nếu tính cùng một vật thể thì chỉ xuất hiện đúng 1 lần (giả sử trong video test nếu sản phẩm đó xuất hiện lần 2 thì được tính là 2 sản phẩm khác nhau)
- Độ phân giải tối thiểu 480p
- Video được quay trực diện sản phẩm, cách sản phẩm từ 5 -15cm
- Tất cả các sản phẩm được đặt nằm ngang, hoặc đứng tùy thuộc hình dạng sản phẩm (ưu tiên đặt các sản phẩm nằm ngang)
- Video không có noises, ánh sáng tốt (ISO camera > 100)

- OUTPUT:

- Video đầu vào với các thông tin được hiển thị trên video cụ thể như sau:
- Bounding box bao quanh các sản phẩm được nhận diện đi kèm với tên sản phẩm+chỉ số biểu thị độ chính xác của sản phẩm nhận diện
- Tên sản phẩm, tổng số lượng, tổng giá sản phẩm được nhận diện.
- Bill thanh toán format file text, có tổng giá khách hàng phải thanh toán.

2. Các nghiên cứu đi trước

a) Retail Store Item Detection using YOLOv5

- Theo bài báo:
 - Bộ dữ liệu sử dụng: SKU110k images dataset, bao gồm 2940 ảnh trong tập test, 8232 ảnh trong tập train và 587 ảnh trong tập validation
 - Model Training: YOLOv5
 - Performance: mAP là 0.7 với IoU là 0.5

b) Deep Learning for Retail Product Recognition: Challenges and Techniques:

- Theo bài báo:
 - Những nhà nghiên cứu đã áp dụng nhiều kỹ thuật vào những bộ dataset khác nhau để đánh giá hiệu quả của chúng. Những bộ dữ liệu GroZi-120, Grozi-320k, COCO, Freiburg Grocery Dataset, Cigarette Dataset, Grocery Store Dataset, GP181 Dataset, D2S Dataset và RPC Dataset sẽ được dùng để train và đánh giá qua các kỹ thuật khác nhau gồm: Mask R-CNN, FCIS, Faster R-CNN, RetinaNet. Bảng kết quả thu được như sau:

Approaches	mAP (%)	mAP50 (%)	mAP75 (%)
Mask R-CNN	78.3	89.8	84.9
FCIS	68.3	88.5	80.9
Faster R-CNN	78.0	90.3	84.8
RetinaNet	80.1	89.6	84.5

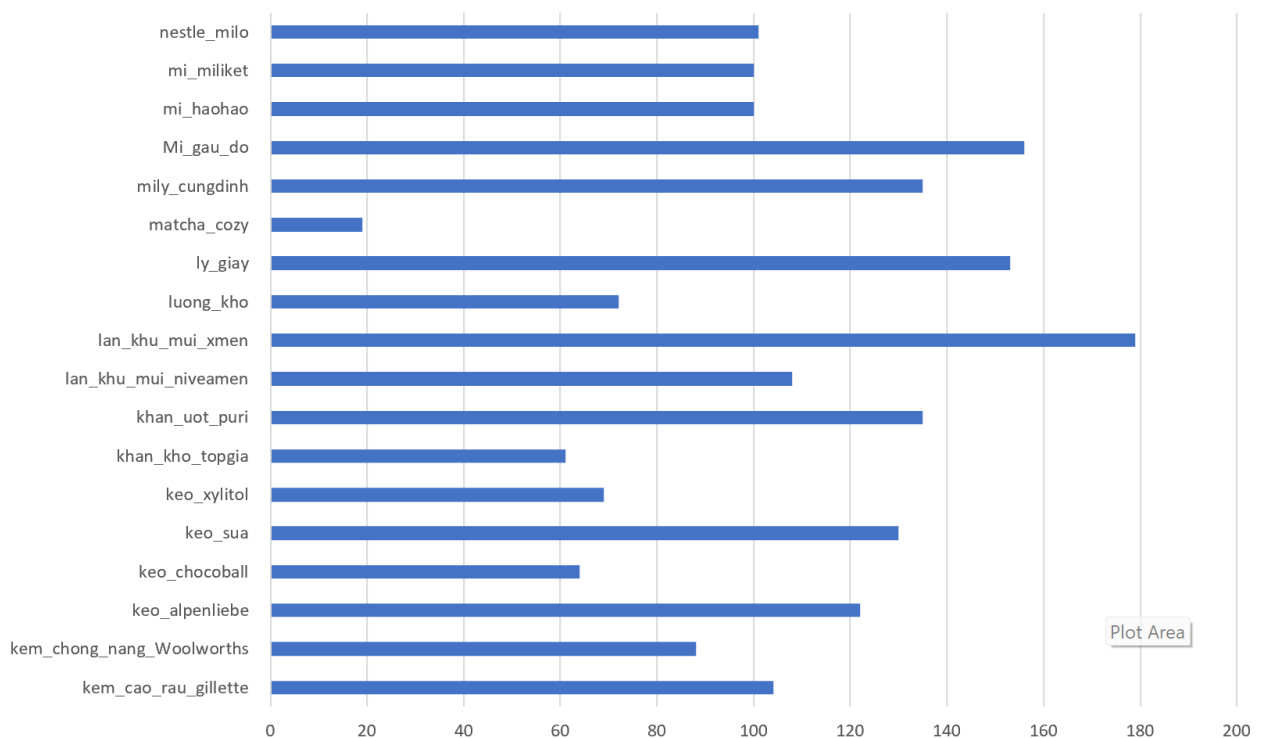
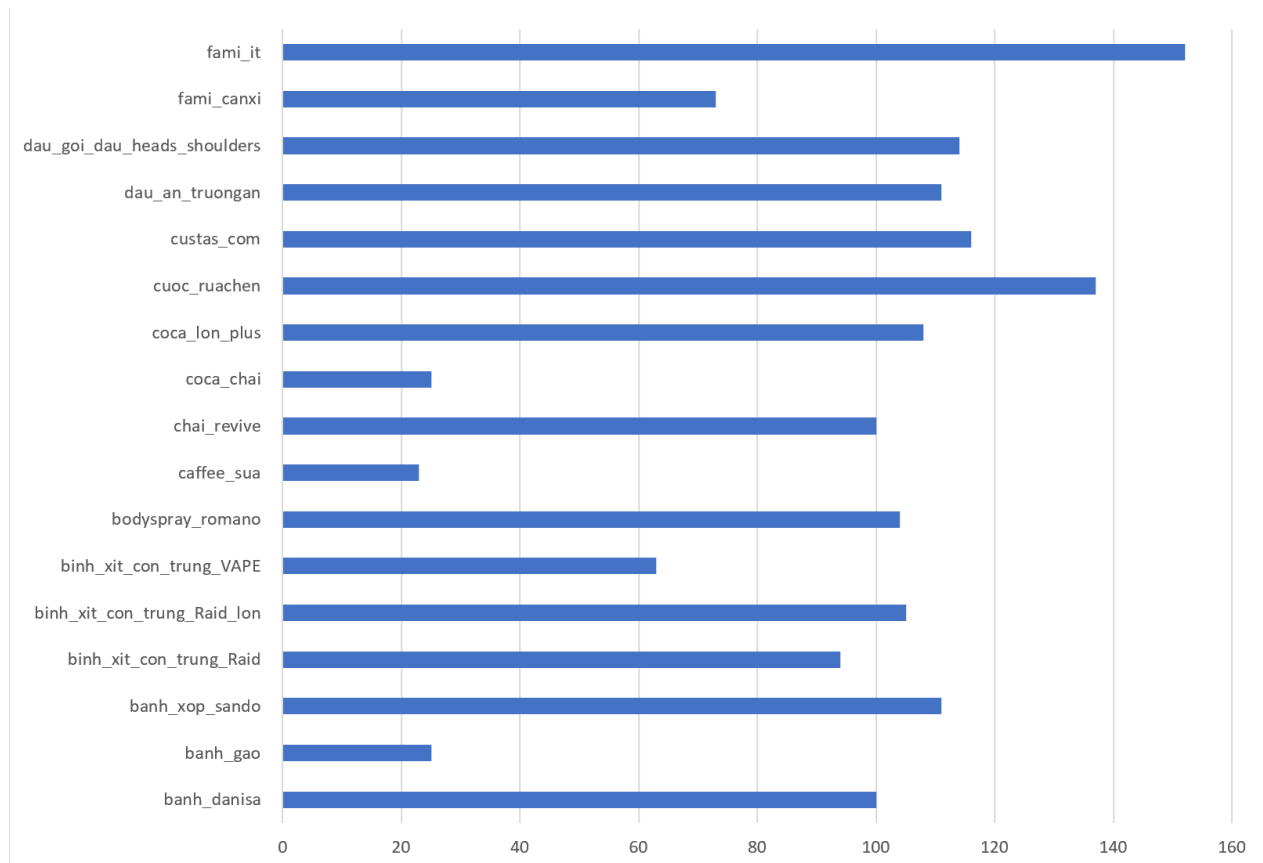
- Tổng kết của bài báo nghiên cứu:
 - Thách thức:
 - Phân loại bài toán với quy mô lớn
 - Hạn chế về dữ liệu
 - Biến thể nội thủy tinh (các sản phẩm có thủy tinh có sự phản xạ từ môi trường xung quanh)
 - Tính linh hoạt
 - Các kỹ thuật đã sử dụng trong nghiên cứu:
 - Generating data with deep neural 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
- Nhận xét từ bài báo: Bài báo có đề cập đến việc YOLO9000 đã cung cấp một phương pháp có thể phát hiện 9000 lớp đối tượng, nhưng không khả thi trong trường hợp phát hiện sản phẩm vì chi phí chú thích cao.

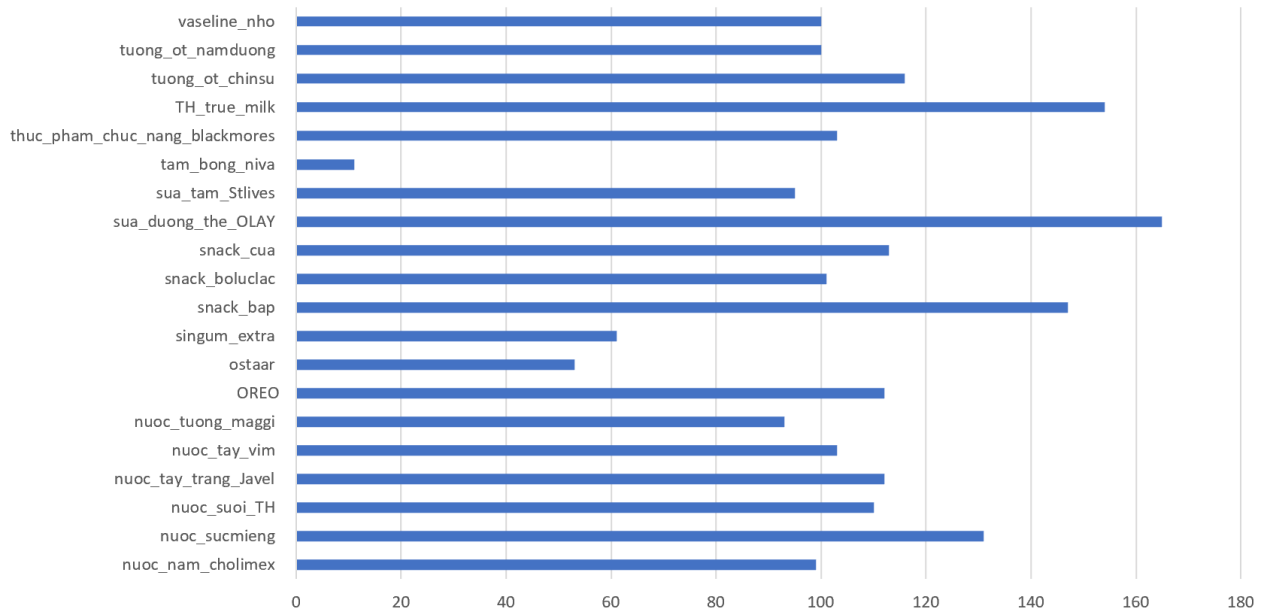
3. Xây dựng dataset

- Dataset được nhóm xây dựng thủ công.
- Nguyên nhân:
 - Những bộ dữ liệu có sẵn chưa phù hợp với yêu cầu và mục tiêu của bài toán là nhận diện sản phẩm. Nguyên nhân là do các sản phẩm được bán phổ biến ở các tạp hóa và siêu thị ở Việt Nam không được xuất hiện nhiều và cũng như điều kiện background của sản phẩm chưa phù hợp với bài toán.
 - Dữ liệu chụp thủ công dễ kiểm soát theo yêu cầu của nhóm. Việc sản phẩm được chụp quy định của bài toán sẽ giúp model dễ huấn luyện và cho những kết quả cao hơn.
- a) Những tiêu chí đặt ra về ảnh chụp sản phẩm:
 - Sản phẩm nằm trong hình, ưu tiên đặt sản phẩm giữa hình.
 - Background là nền sáng.
 - Góc máy trực diện từ trên xuống hoặc nghiêng một góc $< 45^0$.
 - Một sản phẩm sẽ được đặt và chụp theo nhiều góc độ khác nhau để tăng khả năng nhận diện:
 - Với những sản phẩm hình trụ thì ưu tiên chụp mặt có nhãn trước, sản phẩm sẽ được chụp đủ 6 mặt và sẽ ưu tiên những mặt có diện tích lớn hơn.
 - Với sản phẩm gói: Ưu tiên chụp mặt có nhãn. Sản phẩm sẽ được lật sang mặt sau hoặc quay để chụp.
- b) Quá trình thu thập dữ liệu:
 - Chọn địa điểm tiếp cận sản phẩm: sản phẩm chủ yếu được lấy từ siêu thị Coopmart, CoopSmile và Familymart cũng như những tạp hóa và cửa hàng tiện lợi khác.
 - Ảnh sẽ được chụp bằng điện thoại cá nhân của từng thành viên để tăng tốc độ, cũng như số lượng của việc thu thập dữ liệu.
 - Ảnh sau đó sẽ được tải lên google drive riêng của nhóm và để qua các bước sau.
- c) Những vấn đề gặp phải khi thu thập dữ liệu:
 - Gặp phải sự cản trở của siêu thị trong quá trình thu thập hình ảnh sản phẩm trực tiếp. Do đó, sản phẩm sẽ được mua về để chụp, nhưng số lượng sản phẩm giảm đáng kể so với dự tính ban đầu.
 - Số lượng ảnh yêu cầu khá lớn nên việc thu thập tốn khá nhiều thời gian. Cần thời gian là 7 ngày để có thể hoàn thành được việc thu thập dữ liệu.
 - Có khá nhiều ảnh có chứa những vật không liên quan hoặc nhiễu, nên phải tốn thời gian để lọc bớt ảnh.
- d) Về dataset:
 - Bộ dữ liệu tổng bao gồm 5678 tấm ảnh chụp 55 class khác nhau, có sự chênh lệch về ảnh trong mỗi class.
 - Bộ dữ liệu được chia train test và validation theo tỉ lệ 5397: 217; 58. Ngoài ra, model còn được test thông qua một số video dài từ 1 đến 2 phút của nhóm.
- e) Nhận xét:
 - Data được làm thủ công nên khá 'clean' → Giảm bớt thời gian tiền xử lý dữ liệu

- Data có kích thước lớn 14GB, do đó lúc train thường xảy ra quá trình tràn hay crash trên colab.

4. Thông số dữ liệu

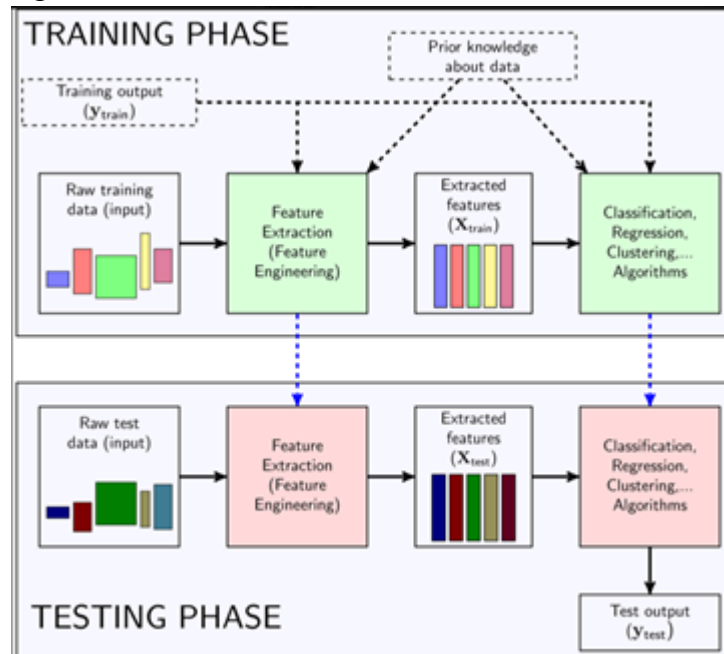




- Nhận xét thông số:
 - Bộ dữ liệu có sự chênh lệch giữa các class, tuy nhiên, chênh lệch là không lớn vì lúc chụp , nhóm đã quy định số lượng ảnh của mỗi class không quá 150.
 - Model train từ bộ dữ liệu này dự đoán sẽ không bị bias, tuy nhiên, vẫn có thể xảy ra trường hợp bị overfitting.
- Bộ dữ liệu train và test:
 - Bộ dữ liệu train và test có kích thước lần lượt là 12.8GB và 1GB, tập dữ liệu sẽ bao gồm file ảnh chụp từ định dạng .jpg và .png các sản phẩm và files label gán nhãn cho từng ảnh. Đối với file label, annotation sẽ có dạng file .txt (1 file ảnh sẽ có 1 file label trùng tên tương đương).
- Các trường hợp khó xử lý:
 - Sản phẩm nằm quá sát biên hoặc nằm quá sát nhau.
 - Các sản phẩm cùng màu sắc và hình dáng.
 - Các sản phẩm cùng hình dáng, khác tông màu.
 - Các sản phẩm khác hình dáng, cùng tông màu.

5. Training và đánh giá model

a) Quá trình training model



- Từ tập dataset đã có từ trước, dữ liệu đó sẽ được tiền xử lý trước khi đưa vào training.
- Sau đó, chúng sẽ được trích xuất các đặc trưng tạo ra các feature map tương đương.
- Những feature map này cuối cùng được đưa vào những thuật toán máy học để phân loại.

b) Về quá trình trích xuất đặc trưng (feature engineering):

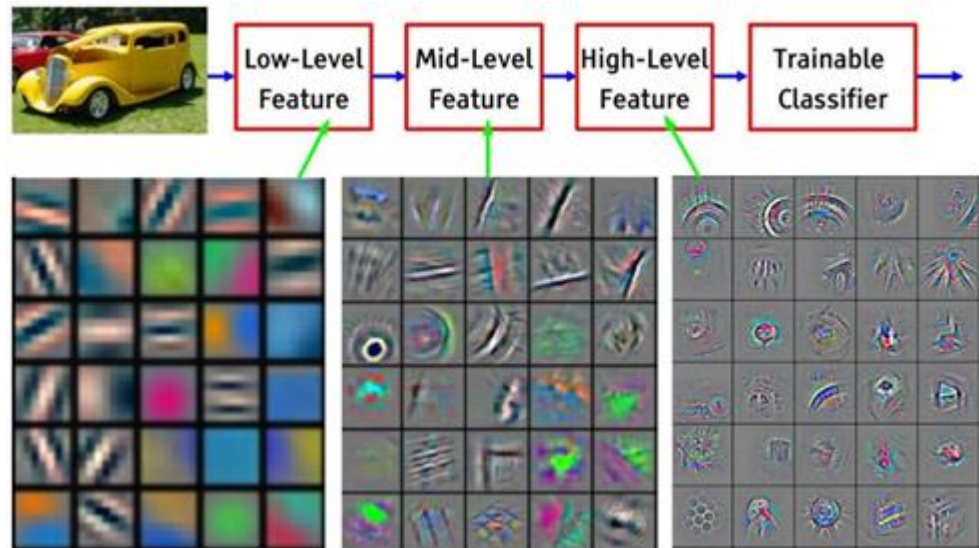
- Là quá trình chuyển những raw data và đặc điểm của chúng về dạng mà máy tính có thể hiểu để có thể hiểu được. Ví dụ: khi ta đưa hình một sản phẩm như "Hộp bánh OREO", máy không thể tự biết lấy những đặc điểm cần thiết như màu hộp, hình dạng hộp,... để "học" và nhận biết những lon hộp bánh khác về sau. Do đó, việc chuyển những đặc điểm đó về những đặc trưng mà máy tính học được là rất cần thiết.

c) Hướng tiếp cận và quá trình chọn model của nhóm:

- Tiếp cận theo hướng transfer learning (tận dụng những kinh nghiệm kiến thức đã học được từ vấn đề này để giải quyết một vấn đề khác có liên quan). Nguyên nhân:
 - Dataset quá nhỏ để có thể xây dựng một mạng lưới chiết xuất đặc trưng hiệu quả.
 - Chưa có kỹ năng chọn đặc trưng tốt.
- Sử dụng nguồn tài nguyên về các bài toán liên quan đến object detection có sẵn để giải quyết vấn đề.
- Sử dụng mô hình CNN (convolutional neural network) để trích xuất đặc trưng qua các layer. Nguyên nhân là vì CNN hoàn toàn phù hợp với yêu cầu bài toán nhóm đặt ra:
 - Kiến trúc phân tầng, học đặc trưng từ các cấp độ khác nhau, mức độ chi tiết đặc điểm của object cần detect trong hình ảnh cũng tăng lên thông qua các

Feature Map (ví dụ như màu sắc, đặc điểm logo,...) đến layer cuối cùng sẽ thu về được feature cho sản phẩm hoàn chỉnh.

- Mô hình đã được huấn luyện trên một datasets lớn từ trước, tức là nó đã được học cách tự điều chỉnh những features mà nó cần trích xuất để phù hợp với tasks tương ứng. Vì vậy, nhóm nghĩ nó sẽ phù hợp với dataset nhỏ của mình.



- Sau khi lựa chọn, nhóm quyết định sẽ sử dụng mô hình YOLOv8, được xem là mô hình object detection tốt nhất hiện nay để huấn luyện và giải quyết bài toán.

d) Model YOLOv8:

- YOLOv8 của Ultralytics là một mô hình "Cutting-edge" (cạnh tranh mạnh), "State-of-the-art" (SOTA) xây dựng dựa trên sự thành công của các phiên bản YOLO trước đó và đưa ra các tính năng và cải tiến mới nhằm nâng cao hiệu suất và tính linh hoạt. YOLOv8 được thiết kế để nhanh chóng, chính xác và dễ sử dụng, là lựa chọn tuyệt vời cho nhiều nhiệm vụ như phát hiện và theo dõi đối tượng, phân đoạn theo thể hiện, phân loại hình ảnh và ước tính tư thế.

Trong C2f, output từ lớp Bottleneck đều được tổng hợp trong khi C3 thì chỉ output từ lớp Bottleneck cuối cùng.

- YOLOv8 loại bỏ 2 lớp convolution.
- Lớp Bottleneck của YOLOv8 giống của YOLOv5 nhưng lớp kernel của convolution đầu tiên được chỉnh từ 1x1 lên 3x3.

e) Để train mô hình:

- Sử dụng file pretrained model với pretrained weights yolov8n. Nguyên nhân:
 - Sử dụng file Pretrained Weights giúp tiết kiệm thời gian train lại toàn bộ model từ đầu.
 - Dataset của nhóm khá nhỏ, nhưng có những đặc điểm tương đồng với dataset dùng để train model YOLOv8 nên để giảm thiểu trường hợp bị overfitting, nhóm chọn giữ nguyên những tham số đã train của model và chỉ train thêm với dữ liệu mới.

f) Quá trình chuẩn bị dữ liệu:

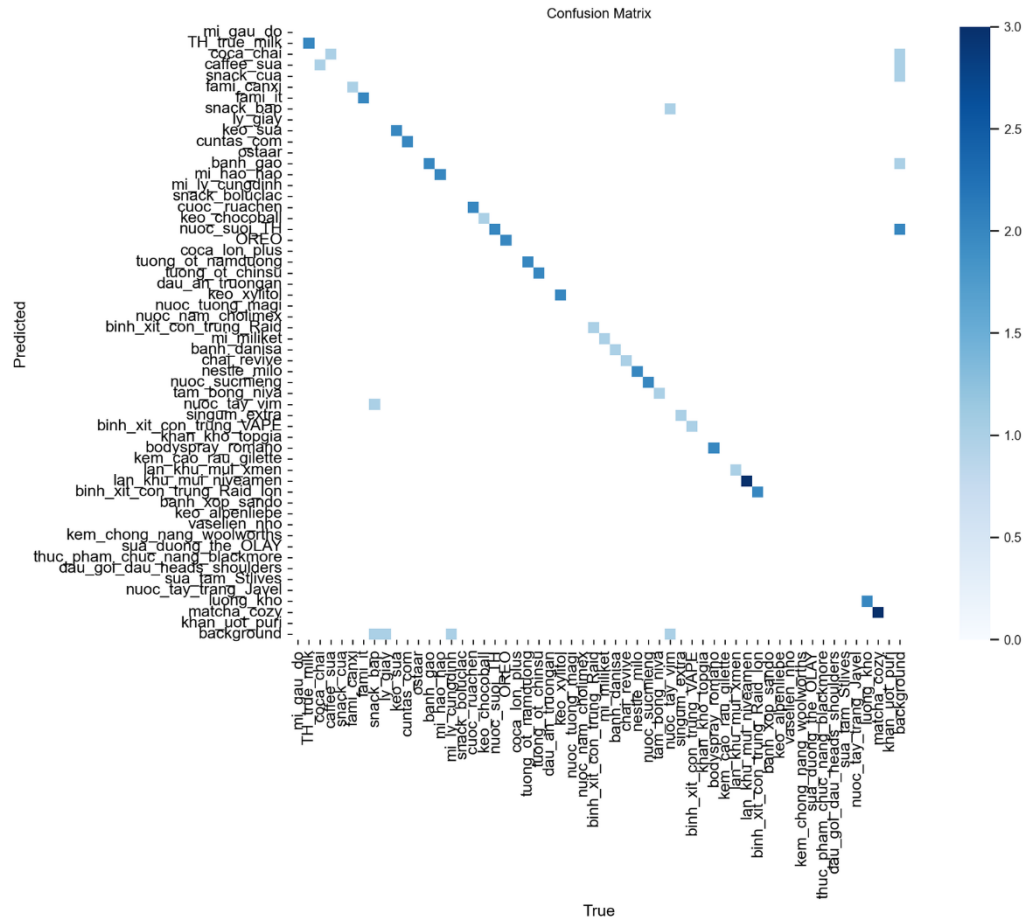
- Dữ liệu chụp thủ công 5678 ảnh.
- Sử dụng tool gán nhãn labelImg để tạo label cho tập ảnh.
- Kết quả sau khi dán nhãn:
 - 5678 file hình ảnh.
 - 5679 file text label (bao gồm 1 file classes.txt chứa tên 55 classes thành từng dòng, được đánh số bắt đầu từ 0). Một file ảnh sẽ có 1 file label tương ứng cùng tên.
 - Mỗi file labels text sẽ chứa số thứ tự của nhãn được gán trong file classes.txt, 4 con số thập phân liên qua đến thông tin của bounding box.
 - Dữ liệu được chia thành 12.8GB train, 2MB validation và 1 GB test.

g) Quá trình train model:

- Upload bộ dữ liệu thu thập được lên google drive.
- Tải các thư viện và framework cần thiết để train model.
- Chuẩn bị lại các file cần thiết để chuẩn bị cho việc training:
 - Chuẩn bị file 'name' chứa tên các classes sẽ được detect trong bộ datasets
 - Tải về các file data train, test, và validation.
 - Các thông số train:
 - Batch: nếu hoàn thành đủ số lượng batch được set up trước thì tính là hoàn thành 1 iterations. Batch size đặt bằng 8
 - Class = 55.
 - Filters cố định trong suốt quá trình train.
 - Epoch = 10.
- Khó khăn trong quá trình train:
 - Quá trình train diễn ra rất chậm. Tốn hơn 1 tiếng cho 1 epoch. Nguyên nhân:
 - Dữ liệu khá lớn.
 - Số ảnh được xử lý mỗi lần chạy chỉ là 2, nên thời gian train lâu.

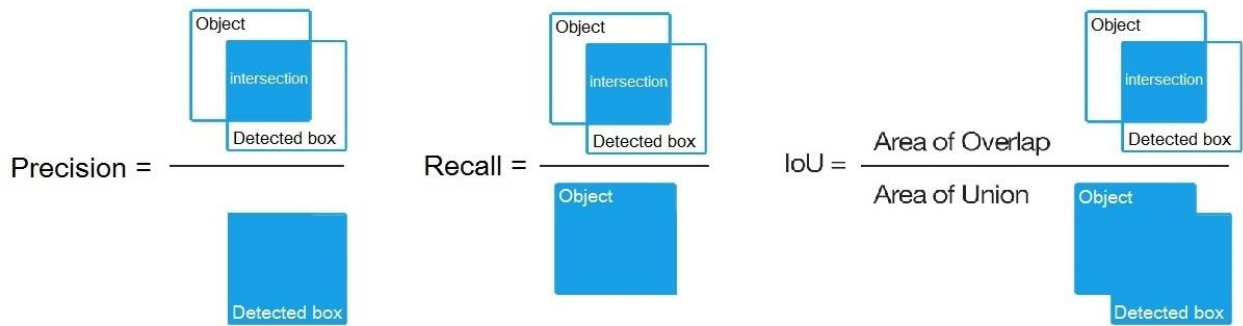
- Trong quá trình train bị out ra (ở epoch thứ 4) vì hết thời gian xử lý của google colab.
 - Sau khi điều chỉnh thông số nhiều lần, cũng như chạy thử trên máy local thì model vẫn được train đầy đủ 10 epoch.
- h) Kết quả và đánh giá:
- Kết quả train:

Class	Images	Instances	Box(P	R	mAP50	mAP50-95): 100
all	275	270	0.886	0.898	0.934	0.757
mi_gau_do	275	5	0.905	1	0.995	0.895
TH_true_milk	275	5	0.917	1	0.995	0.735
coca_chai	275	5	0.839	1	0.995	0.675
caffee_sua	275	5	0.586	1	0.898	0.657
snack_cua	275	5	0.889	1	0.995	0.935
fami_canxi	275	5	0.93	1	0.995	0.89
fami_it	275	5	0.93	1	0.995	0.912
snack_bap	275	5	1	0	0.0151	0.01
ly_giay	275	5	1	0.292	0.843	0.335
keo_sua	275	5	1	0.585	0.995	0.617
cuntas_com	275	5	0.943	1	0.995	0.836
ostaar	275	5	0.917	1	0.995	0.872
banh_gao	275	5	0.71	1	0.995	0.817
mi_hao_hao	275	5	1	0.789	0.995	0.572
mi_ly_cungdinh	275	5	0	0	0	0
snack_boluciac	275	5	1	0	0.88	0.655
cuoc_ruachen	275	5	1	0.917	0.995	0.895
keo_chocoball	275	5	0.92	1	0.995	0.818
nuoc_suoi_TH	275	5	0.847	1	0.995	0.84
OREO	275	5	0.902	1	0.995	0.902
coca_lon_plus	275	5	0.925	1	0.995	0.812
tuong_ot_namduong	275	5	0.897	1	0.995	0.892
tuong_ot_chinsu	275	5	0.923	1	0.995	0.916
dau_an_truongan	275	5	0.916	1	0.995	0.883
keo_xylitol	275	5	0.924	1	0.995	0.895
nuoc_tuong_magi	275	5	0.983	1	0.995	0.713
binh_xit_con_trung_Raid	275	5	0.924	1	0.995	0.892
mi_miliket	275	5	0.908	1	0.995	0.697
banh_danisa	275	5	0.929	1	0.995	0.916
chai_revive	275	5	0.923	1	0.995	0.824
nestle_milo	275	5	0.928	1	0.995	0.813
nuoc_sucmieng	275	5	0.903	1	0.995	0.906
tam_bong_niva	275	5	0.832	1	0.995	0.723
nuoc_tay_vim	275	5	0	0	0.0227	0.0163
singum_extra	275	5	0.999	1	0.995	0.876
binh_xit_con_trung_VAPE	275	5	0.925	1	0.995	0.723
khan_kho_topgia	275	5	0.924	1	0.995	0.951



- Đánh giá model Yolov8:

- Model sẽ được đánh giá dựa trên 2 độ đo là mAP và loss. Chỉ số mAP được nhóm sử dụng là AP50, AP50-95
- Nguyên nhân đánh giá model bằng mAP:
 - IoU: độ đo overlap giữa các bounding box, cụ thể là giữa groudth truth bounding box - bao quanh chính xác bounding box của vật thể - là bounding box mà chúng em đã lable với bounding box mà mô hình dự đoán.
 - AP: là chỉ số có quan hệ mật thiết với chỉ số precision(phần trăm các bounding box được dự đoán là đúng) và recall (tỉ lệ phần trăm các bounding box được dự đoán đều chính xác).



- AP50: là độ chính xác với $\text{IoU} = 0.5$
- AP50 - 95: là độ chính xác với $\text{IoU} = \text{từ } 0.5 \text{ đến } 0.95$
- IoU có ý nghĩa quan trọng đối với chỉ số mAP và việc lựa chọn giá trị của IoU sẽ ảnh hưởng đến kết quả đánh giá của model. Trong các bài toán nhận diện vật thể, chúng ta tính toán chỉ số precision và recall với một ngưỡng IoU cho trước, ví dụ đơn giản nhất là nếu ta cho ngưỡng IoU bằng 0.4 và chỉ số IoU sau khi tính toán trên bounding box được dự đoán là 0.5 thì ta tính rằng bounding box được dự đoán đó là đúng, tuy nhiên nếu đặt ngưỡng IoU bằng 0.6 thì với chỉ số IoU sau khi tính toán trên bounding box được dự đoán là 0.5 thì bounding box được dự đoán đó là sai.
- Khó khăn khi đánh giá model:
 - Mặc định ban đầu, do chính sách của colab notebook, các session khi train liên tục bị crashed giữa chừng, làm ảnh hưởng đến quá trình nắm bắt độ chính xác của model. Nhưng sau khi điều chỉnh lại thì phần khó khăn đã được khắc phục.
- Model detect trên dữ liệu test:
 - Model dự đoán khá chính xác các sản phẩm trong bộ test.



- Có những sản phẩm mà model không dự đoán được hoặc dự đoán sai:



- Nguyên nhân dự đoán: sản phẩm bị khuất một phần, hoặc nằm ở vị trí góc quay không có trong tập train.
- Hình chai nước bị dự đoán lặp tới 3 lần. Nguyên nhân dự đoán: phản chiếu của nước trong chai khiến mô hình sự đoán sai.



- Model đánh giá trên video:
 - Model thường đánh giá sai ở các trường hợp nằm ở góc, hoặc sản phẩm nằm quá gần nhau.



- Với những sản có hình trụ thường không được chính xác lắm. Nguyên nhân dự đoán: Hình bị khuất hoặc đứng gần nhau sẽ được xác định là một khối vật thể.

→ Kết luận chung:

- Mô hình dự đoán khá tốt trên những bức hình rõ ràng, có logo với độ sáng tốt.
- Tuy nhiên mô hình bị overfit do bộ dữ liệu chưa đủ lớn cùng với việc chuẩn bị dữ liệu chưa được tốt.
- Mô hình sẽ dễ sai trên những sản phẩm quá dài và bị trượt mất khỏi khung hình, cũng như những sản phẩm có phản chiếu như ly thủy tinh hay chai nước.
- Đôi khi mô hình sẽ dự đoán sai những vật dụng có màu sắc gần giống nhau cũng như có kích thước khối giống nhau.
- Những góc độ khác nhau cũng ảnh hưởng tới chất lượng dự đoán sản phẩm.
- Những vật phẩm nằm gần nhau cũng sẽ không được dự đoán tốt do bị che khuất.
- Trường hợp detect đúng:
 - Do màu sắc của những sản phẩm đặc trưng và khác biệt.
 - Số lượng ảnh được huấn luyện cũng tương đối so với những class khác.
 - Góc độ và ánh sáng giống với trong ảnh train.

6. Hướng cải thiện mô hình

- Về data:
 - Tăng sự đa dạng và số lượng, loại sản phẩm nhiều hơn, ngoài ra phải thường xuyên thu thập data vì mẫu mã sản phẩm, giá cả liên tục thay đổi, ngoài ra cũng phải đáp ứng được các chương trình khuyến mãi của các chuỗi cửa hàng,...
 - Quy trình thu thập data phải diễn ra chặt chẽ, có sự đầu tư về phần cứng thiết bị, ví dụ như set up một phòng lab riêng biệt chỉ dành cho việc chụp hình các sản phẩm, các camera set up có giá đỡ cố định ở các góc, ánh sáng điều kiện, phù hợp với bối cảnh thực tế, chú ý đến vấn đề tạo nên một balanced data,... => Clean Data
 - Tăng cường dữ liệu, sử dụng các kỹ thuật như Data Augmentation (blur, gray scale, cutout, cutmix, rotate,...),...việc lựa chọn các bước tăng cường dữ liệu rất quan trọng, không phải phương pháp nào cũng tốt cho tập dữ liệu.
- Về model:
 - Model bị overfit, vì vậy việc tăng cường độ đa dạng của dữ liệu sẽ giúp cho mô hình dự đoán tốt hơn.
 - Nếu có thêm thời gian, có thể điều chỉnh thêm các tham số của model để xem xét tính hiệu quả hơn.


7. Cập nhật sau khi vấn đáp

a) mAP và những khái niệm liên quan

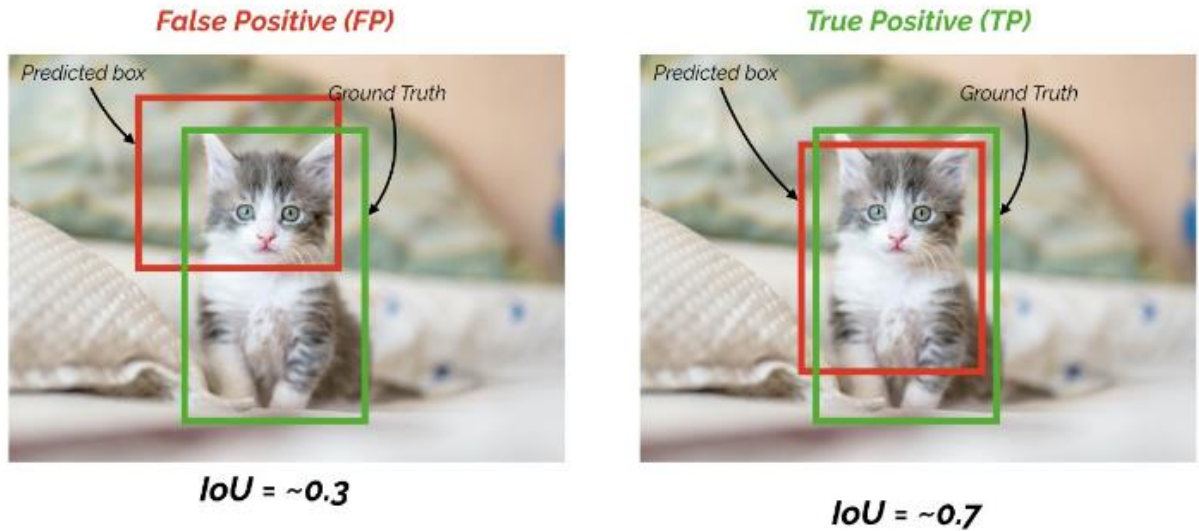
- Confusion matrix: là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- True positive (TP): Mô hình dự đoán đúng nhãn của vật có trong ảnh.
- True negative (TN): Mô hình không dự đoán có nhãn của vật đó trong ảnh, và thực sự trong ảnh không hề có vật đó trong ảnh.
- False positive (FP): Mô hình dự đoán nhãn của vật đó, nhưng không hề có vật đó trong ảnh.
- False negative (FN): Mô hình không dự đoán được có vật trong ảnh.
- Intersect over union (IoU): là tỉ lệ giữa giao và hợp của bounding box dự đoán và bounding box ground truth của vật trong hình. Công thức tính IoU như sau:

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


- Nhờ có IoU mà ta tính được các chỉ số TP, FP, FN, TN theo ngưỡng của IoU, ví dụ: nếu ngưỡng của IoU là 50 thì những bounding box có IoU lớn hơn 0.5 sẽ được tính là TP, bé hơn 0.5 sẽ được tính là FP.

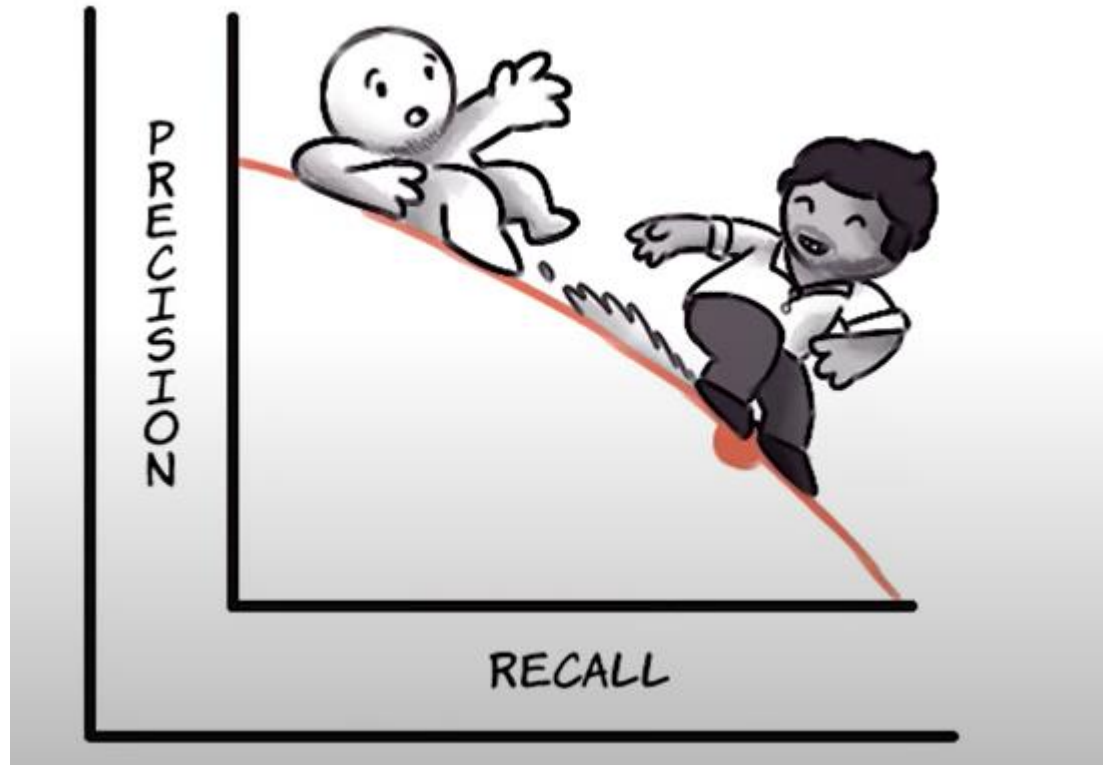


- Với các chỉ số của Confussion matrix, ta tính được recall và precision của mỗi class theo một ngưỡng IoU theo công thức sau:

$$\text{Precision} = \frac{\sum TP}{\sum TP + FP}$$

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN}$$

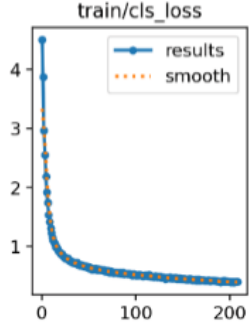
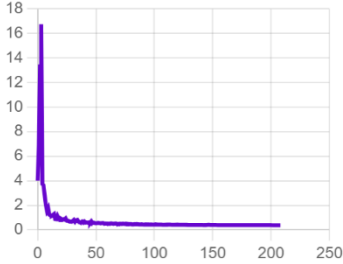
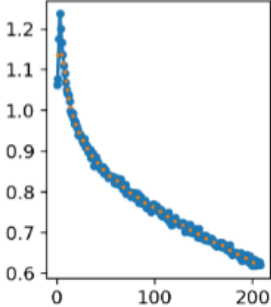
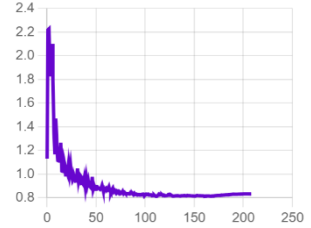
- Với mỗi ngưỡng IoU, ta tính được 2 chỉ số Precision và Recall khác nhau. Từ đó, ta sẽ có bảng tương qua của precision và recall:



- Average Precision (AP): là chỉ số Precision trung bình của nhiều IoU threshold của mỗi class.
- mAP: Là AP trung bình của nhiều class trong ảnh.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Tiêu chí	Yolov8	Faster R-CNN sử dụng Detectron2
Nơi train	Trên google colab với pretrained model default	Trên web Roboflow với pretrained model từ nhiều nguồn khác nhau
Định dạng	Custom dataset	COCO

Class loss		
Box loss		
mAP50	Khoảng 98%	Khoảng 99%
Kết quả	Tốc độ nhận diện chậm hơn Faster R-CNN, khi có nhiều sản phẩm xuất hiện trong khung hình thì dễ xảy ra sai.	Tốc độ nhận diện nhanh hơn Yolov8, khi có nhiều sản phẩm xuất hiện trong khung hình thì detect tốt hơn hẳn.

TÀI LIỆU THAM KHẢO

- [1] <https://docs.ultralytics.com/>
- [2] <https://towardsdatascience.com/enhanced-object-detection-how-to-effectively-implement-yolov8-afd1bf6132ae>
- [3] <https://github.com/ultralytics/ultralytics>
- [4] <https://blog.roboflow.com/retail-store-item-detection-using-yolov5/>
- [5] <https://www.hindawi.com/journals/cin/2020/8875910/>
- [6] <https://phamdinhhkhanh.github.io/2020/05/31/CNNHistory.html>
- [7] <https://viblo.asia/p/tim-hieu-ve-convolutional-neural-network-va-lam-mot-vi-du-nho-ve-phan-loai-anh-aWj53Wxo56m>
- [8] <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>
- [9] <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/>