ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

Bộ môn: MÁY HỌC – CS114.M11.KHCL

ĐỀ TÀI:

NHẬN DIỆN SẢN PHẨM Ở QUẦY THANH TOÁN CỦA CÁC CỬA HÀNG, SIÊU THỊ VỚI MÔ HÌNH YOLOv4-Tiny

Giảng viên hướng dẫn: *thầy* Phạm Nguyễn Trường An

Sinh viên thực hiện:

- Hoàng Ngọc Bá Thi (19522255@gm.uit.edu.vn)
- Nguyễn Quốc Đạt (19521338@gm.uit.edu.vn)
- Nguyễn Thanh Sang (19522124@gm.uit.edu.vn)

TP.HCM, tháng 1 năm 2022

LỜI MỞ ĐẦU

Công nghệ đang dần trở thành một yếu tố không thể thiếu trong mọi lĩnh vực đời sống, đặc biệt là sự góp mặt của các công nghệ hiện đại trong thương mại, mua sắm. Thời gian gần đây, nhờ ứng dụng công nghệ mà việc mua sắm và thanh toán ngày càng trở trên thuận tiện cho cả người mua sắm và các đơn vị bán hàng. Các cửa hàng, siêu thị sử dụng phương thức quét mã BarCode trên các sản phẩm giúp việc kiểm hàng và xuất hóa đơn thanh toán trở nên nhanh và chính xác hơn cách thủ công rất nhiều. Tuy nhiên, phương thức quét mã BarCode cũng có khuyết điểm là không thể quét nếu mã BarCode bị mờ, mất mã, bị che khuất, bị méo mó do bao bì của sản phẩm.

Với mong muốn tìm hiểu phương thức quét sản phẩm mới khác với quét mã BarCode, cũng như hướng đến sự tiện lợi hơn trong quá trình mua-bán sản phẩm, nhóm sinh viên chúng em đã chọn đề tài:"**Nhận diện sản phẩm ở quầy thanh toán của các cửa hàng, siêu thị với mô hình YOLOv4-Tiny**". Vì khả năng và kiến thức của chúng em vẫn còn nhiều điều cần cải thiện nên rất mong nhận được sự nhận xét, tư vấn và hướng dẫn thêm từ giảng viên.
Xin chân thành cảm ơn!

MỤC LỤC

I. GIỚI THIỆU BÀI TOÁN NHẬN DIỆN SẢN PHẨM…….....3

II. DỮ LIỆU DÙNG ĐỂ HUẤN LUYỆN VÀ KIỂM THỬ MÔ HÌNH…….4
1. Xây dựng bộ dữ liệu……...**5**
2. Mô tả bộ dữ liệu……..…**6**

III. LÝ THUYẾT ĐÁNH GIÁ MÔ HÌNH NHẬN DIỆN VẬT THỂ (OBJECT DETECTION MODEL)……7

IV. ÁP DỤNG MÔ HÌNH MÁY HỌC YOLOv4 ĐỂ GIẢI QUYẾT BÀI TOÁN
1. Giới thiệu về mô hình YOLO và YOLOv4-Tiny……...……**9**
2. Huấn luyện mô hình YOLOv4-Tiny……..**12**
3. Thực nghiệm và nhận xét……...**12**

V. TỔNG KẾT……14

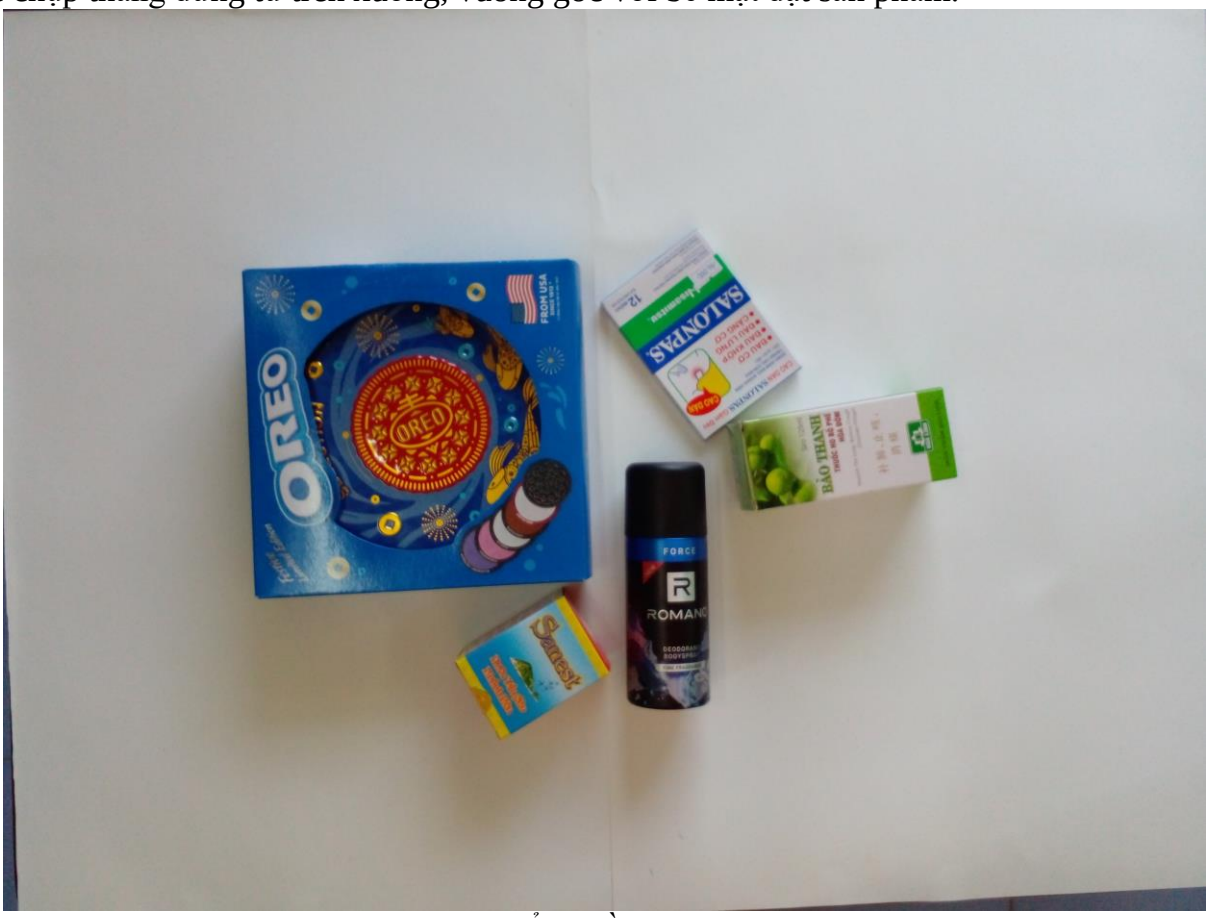
VI. TÀI LIỆU THAM KHẢO……..15

I. GIỚI THIỆU BÀI TOÁN NHẬN DIỆN SẢN PHẨM

Đồ án này hướng đến phương thức nhận diện hàng hóa như sau: chỉ cần đặt các sản phẩm vào khu vực thanh toán, hệ thống sẽ thông qua ảnh chụp từ camera để xuất ra thông tin sản phẩm có trong khu vực thanh toán, từ thông tin này, người bán chỉ cần xuất hóa đơn và thanh toán ngay hay thậm chí người mua có thể tự thanh toán mà không cần người bán/nhân viên bán hàng. Ở đồ án này, nhóm sử dụng một mặt phẳng màu trắng để mô phỏng khu vực thanh toán.

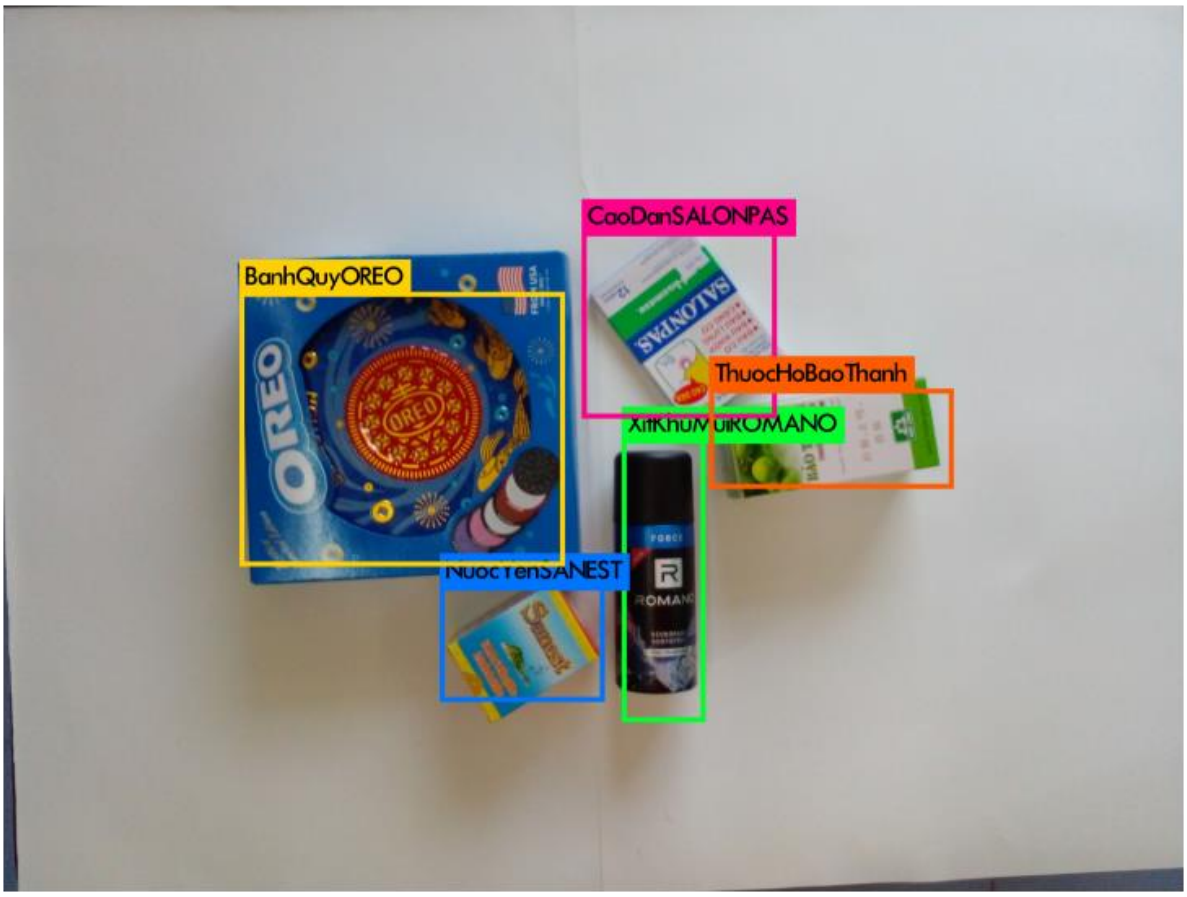
Bài toán đặt ra: Hệ thống cần nhận diện được sản phẩm trong khu vực thanh toán. Đây là bài toán nhận diện vật thể (Object Detection).

INPUT: Ảnh chụp khu vực thanh toán có chứa các mặt hàng của cửa hàng, siêu thị. Ảnh có góc chụp thẳng đứng từ trên xuống, vuông góc với bề mặt đặt sản phẩm.



Ảnh đầu vào

OUTPUT: Thông tin của các sản phẩm (có thể là: mã sản phẩm, tên sản phẩm,…) có trong ảnh chụp khu vực thanh toán.



Ảnh đầu ra (chữ trong ảnh: CaoDanSALONPAS, BanhQuyOREO, ThuocHoBAOTHANH, XitKhuMuiROMANO, NuocYenSANEST)

II. DỮ LIỆU DÙNG ĐỂ HUẤN LUYỆN VÀ KIỂM THỬ MÔ HÌNH

1. Xây dựng bộ dữ liệu

a) <u>Thu thập dữ liệu:</u>

Nhóm đã sử dụng điện thoại để chụp ảnh 10 sản phẩm bao gồm:

- Nước Yến SANNEST (NuocYenSANEST)
- Nhang muỗi RAID (NhangMuoiRAID)
- Mì xào Hảo Hảo (MiXaoHAOHAO)
- Bánh quy bơ GOODY (BanhQuyBoGOODY)
- Thuốc ho Bảo Thanh (ThuocHoBAOTHANH)
- Cao dán Salonpas (CaoDanSALONPAS)
- Bánh quy OREO (BanhQuyOREO)
- Nước tương Maggie (NuocTuongMAGIE)
- Nước ngọt 7-UP (NuocNgot7UP)
- Xịt khử mùi Romano (XitKhuMuiROMANO)

*** Yêu cầu về hình ảnh của các sản phẩm như sau:**

- Bề mặt đặt sản phẩm (mô phỏng khu vực thanh toán) có màu trắng và có kích thước (DxR): 80cm X 60cm.

- Góc chụp từ trên xuống, vuông góc với bề mặt đặt sản phẩm.

- Chất lượng ảnh rõ hình sản phẩm, không bị mờ, nhòe.

- Ánh sáng vừa đủ, không quá tối, không quá sáng, không bị chói lóa sản phẩm.

- Mỗi sản phẩm phải có đủ hình ảnh ở cả mặt trước và sau với sản phẩm có 2 mặt in nhãn mác; Với sản phẩm chỉ có 1 mặt in nhãn mác thì chỉ chụp mặt có nhãn mác; Với sản phẩm như Lon, Chai, Lọ thì chụp rõ phần nhãn mác chính.

- Mỗi bề mặt của mỗi sản phẩm phải có đủ hình ảnh ở 8 hướng xoay khác nhau: 0h, 1h30, 3h, 4h30, 6h, 7h30, 9h, 10h30.

- Kết hợp từ 2-5 sản phẩm, có cùng loại và khác loại: Mỗi bộ kết hợp có tối thiểu 2 cách sắp xếp khác nhau.

- Ảnh lưu dưới định dạng JPEG (.jpg).

b) Gắn nhãn dữ liệu:

Nhóm sử dụng công cụ có tên "**LabelImg**"[1], đây là công cụ mã nguồn mở giúp gắn nhãn hình ảnh, từ dữ liệu về nhãn đã gắn, chúng ta có thể tạo ra bộ dữ liệu dùng cho việc huấn luyện và kiểm thử các mô hình máy học. **LabelImg** hỗ trợ 2 định dạng dữ liệu gồm: file XML cho Pascal VOC và file TXT cho YOLO.

Nguồn tải công cụ và hướng dẫn cài đặt, hướng dẫn sử dụng có tại trang web: https://github.com/tzutalin/labelImg

c) Tiền xử lý và Tăng cường bộ dữ liệu với Roboflow:

Roboflow[2] là đội ngũ chuyên cung cấp các giải pháp ứng dụng Máy học phục vụ cho cá nhân, doanh nghiệp. Ngoài ra Roboflow còn cung cấp nền tảng Web hỗ trợ việc xây dựng các dự án Máy học miễn phí và có tính phí, trong đó có việc xây dựng bộ dữ liệu và huấn luyện các mô hình Máy học.

****Tiền xử lý dữ liệu:***

Nhóm đã sử dụng chức năng "Preprocessing" để điều chỉnh kích thước của ảnh trong bộ dữ liệu thành 416x416 pixel.

****Tăng cường bộ dữ liệu:***

Nhóm sử dụng chức năng "Augmentation" để áp dụng những thông số chỉnh sửa ảnh, nhằm tăng sự đa dạng cho bộ dữ liệu, những thông số chỉnh sửa bao gồm:

- Xoay ảnh theo góc 90 độ (90 Degree Rotate).

- Tăng và giảm 13% độ bão hòa màu sắc (Saturation).

- Tăng và giảm 12% độ sáng (Brightness).

2. **Mô tả bộ dữ liệu**

- Bộ dữ liệu thu thập được có 257 ảnh của 10 sản phẩm (đã nêu ở phần II.1), bao gồm ảnh chụp từ 1-5 sản phẩm trong cùng 1 khung hình.

- Kèm theo mỗi ảnh là 1 file TXT lưu lại thông tin gắn nhãn cho ảnh đó, dùng để huấn luyện mô hình YOLO. File TXT có dạng [mã Nhãn/Lớp] [tọa độ tâm của Bounding Box] [Chiều rộng Bounding Box] [Chiều dài Bounding Box]
ví dụ: 1 0.520683 0.481561 0.151757 0.206977.

- File classes.txt lưu tên của các Nhãn, mã Nhãn chính là thứ tự của các Nhãn trong file này.

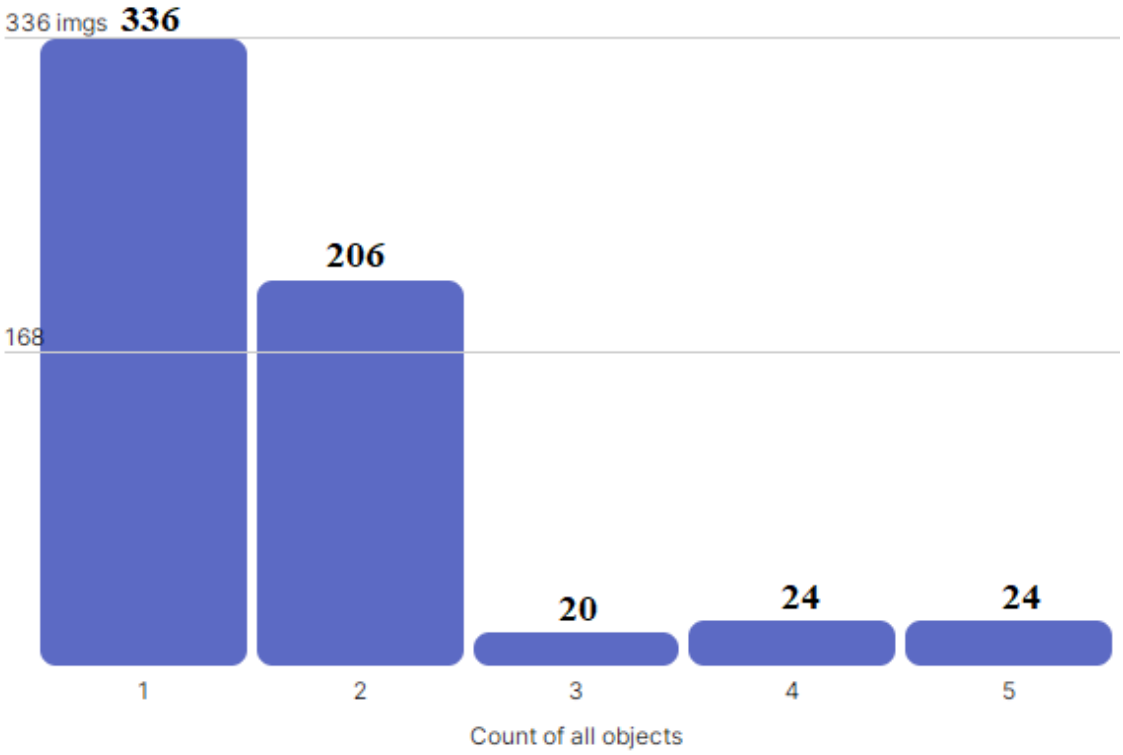
- Sau khi tiến hành tiền xử lý và tăng cường bộ dữ liệu với Roboflow, thì Nhóm đã dây dựng được bộ dữ liệu hoàn chỉnh gồm 608 ảnh kèm file TXT gắn nhãn của 10 sản phẩm khác nhau.

- Dữ liệu được chia thành 3 tập **train, validation và test** với tỉ lệ **7:2:1**.

- Nhìn chung thì số lần xuất hiện của các sản phẩm là nhiều và tương đối đều nhau, chỉ có sản phẩm "BanhQuyOREO", "NuocNgot7UP", "XitKhuMuiROMANO" là ít hơn so với các sản phẩm còn lại khoảng 20-30 lần xuất hiện.

Thống kê số lượng ảnh mà các sản phẩm có xuất hiện

- Về phân bổ số lượng ảnh trên số sản phẩm xuất hiện, thì số ảnh có chứa 1 và 2 vật thể là nhiều nhất, chiếm khoảng 89% tổng số ảnh. Với các ảnh có chứa 3, 4, 5 vật thể thì số lượng lần lượt là 20, 24, 24.



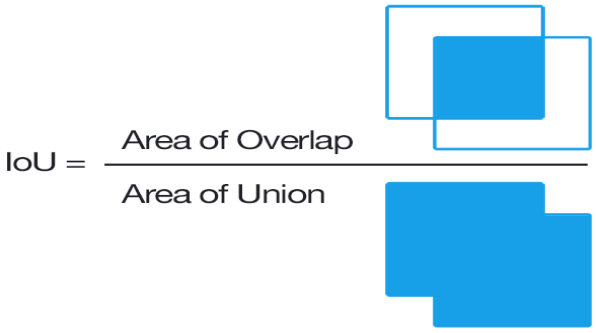
Biểu đồ thống kê số lượng ảnh có chứa từ 1 đến 5 sản phẩm

III. LÝ THUYẾT ĐÁNH GIÁ MÔ HÌNH NHẬN DIỆN VẬT THỂ (OBJECT DETECTION MODEL)

Trước khi đi đến phần huấn luyện một mô hình nhận diện vật thể, chúng ta cần nhắc lại một vài khái niệm cơ bản trong việc đánh giá mô hình, điều này giúp chúng ta nhận biết được mô hình đã huấn luyện có tốt hay không.

*** IOU – Intersection Over Union**

IOU là chỉ số đánh giá được sử dụng để đo độ chính xác của Object detector trên tập dữ liệu cụ thể, cụ thể ở đây là phần giao nhau (Area of Orverlap) giữa Bounding Box mà Object Detector dự đoán được (Predicted bounding box) và Bounding Box thực sự trên bộ dữ liệu (Ground-Truth bounding box), xét trên khu vực 2 thành phần trên hợp với nhau (Area of Union).



Cách tính IOU

*** True-Positive (TP), False-Positive (FP), False-Nagative (FN)**

Chỉ số IOU được dùng làm ngưỡng để chúng ta xác định TP, FP và FN. Thường thì ngưỡng IOU là **0.5**:

- Nếu IOU > 0.5: xác định đây là đối tượng đã được nhận diện đúng <=> TP.
- Nếu IOU < 0.5: xác định đây là đối tượng nhận diện sai <=> FP.

- Nếu IOU = 0.5: tùy vào cách xác định của người thiết kế mô hình mà đưa vào dạng TP hay FP.
- FN cho trường hợp không nhận diện được đối tượng nào nhưng dữ liệu mẫu lại có chứa đối tượng cần nhận diện.
- Ngoài ra còn có True-Negative (TN) cho trường hợp dự đoán là "Không" và khớp với dự liệu mẫu cũng ghi "Không". Tuy nhiên, vì bài toán nhận diện vật thể chỉ tìm và xác định vật thể trong ảnh là gì chứ không trả lời câu hỏi Có-Không nên ở đây TN không được sử dụng.

*** Precision và Recall**
- Precision: là chỉ số đo độ chính xác của dự đoán, tức là tỉ lệ số lần dự đoán đúng trên tổng số lần dự đoán. Precision đại diện cho độ tin cậy của mô hình nhận diện.
- Recall: là chỉ số đánh giá khả năng tìm ra được tất cả các Ground-truth bounding box, được tính bằng cách lấy tỉ lệ Số lần dự dự đoán chính xác trên Số lần nhận dạng đúng có thể có. Cò thể hiểu là Recall đại diện cho độ nhạy của model.

$$Precision = \frac{TP}{TP + FP} = \frac{Số\ dự\ đoán\ chính\ xác}{Tổng\ số\ lần\ dự\ đoán}$$

$$Recall = \frac{TP}{TP + FN} = \frac{Số\ lần\ dự\ đoán\ chính\ xác}{Số\ lần\ nhận\ dạng\ đúng\ có\ thể\ có}$$

Công thức tính Precision và Recall

*** AP (Average Precision) và mAP (mean Average Precision)**
Từ precision và recall đã được định nghĩa ở trên chúng ta cũng có thể đánh giá mô hình dựa trên việc thay đổi một ngưỡng và quan sát giá trị của Precision và Recall. Khái niệm *Area Under the Curve* (AUC) cũng được định nghĩa tương tự. Với *Precision-Recall Curve*, AUC còn có một tên khác là **Average Precision** (AP).

Giả sử có N ngưỡng để tính Precision và Recall, với mỗi ngưỡng cho một cặp giá trị Precision, Recall là $Pn,$,=1,2,…,N. Precision-Recall curve được vẽ bằng cách vẽ từng điểm có toạ độ ($Pn,$) trên trục toạ độ và nối chúng với nhau. AP được xác định bằng:

$$AP=\sum(Rn-Rn-1)Pnn$$

AP tính theo ngưỡng IOU được ký hiệu là: AP@<ngưỡng IOU>. Ví dụ: AP@0.5 là tính AP với ngưỡng IOU = 0.5.

Và **mAP** là trung bình của AP được tính cho tất cả các lớp. Mối quan hệ giữa precision – recall giúp **mAP** đánh giá được **độ chính xác** của mô hình nhận diện. Vì precision – recall thay đổi khi ngưỡng IOU thay đổi nên tại một giá trị IOU xác định, chúng ta có thể tính toán được độ chính xác của các mô hình (ví dụ: mAP@0.5 = 80% có nghĩa là = 0.5, mAP của mô hình là 80%).

IV. ÁP DỤNG MÔ HÌNH MÁY HỌC YOLOv4-Tiny ĐỂ GIẢI QUYẾT BÀI TOÁN

1. Giới thiệu về mô hình YOLO và YOLOv4-Tiny

a) Tổng quan về YOLO

- YOLO là viết tắt của câu "You Only Look Once" (tạm dịch:"Bạn chỉ nhìn một lần"). Hiện tại có 4 phiên bản chính của YOLO bao gồm: YOLOv1, YOLOv2, YOLOv3, và YOLOv4. Hiện tại, đã có nhiều thông tin và phần cài đặt của YOLOv5 với tác giả là Glenn Jocher, tuy nhiên do chưa có bài viết nghiên cứu chính thức nào được công bố, và hiểu biết của các thành viên đều chưa nhiều nên nhóm quyết định chỉ nhắc đến như một sự tham khảo, chưa xác định đây là một trong những phiên bản chính của YOLO.

- Tác giả của 3 phiên bản đầu tiên là ông Joseph Redmon, sau phiên bản YOLOv3 thì ông Joseph không tiếp tục phát triển YOLO nữa. Do đó phiên bản YOLOv4 và YOLOv5 được những người khác có tài năng và đam mê với YOLO xây dựng nên:

  + Năm 2015, ông Joseph Redmon cùng cộng sự lần đầu công bố bài viết nghiên cứu có tên "*You Only Look Once: Unified, Real-Time Object Detection*"[3] (tạm dịch:"*Bạn chỉ nhìn một lần: Phát hiện đối tượng theo thời gian thực*"). Đây là phiên bản đầu tiên: YOLOv1.

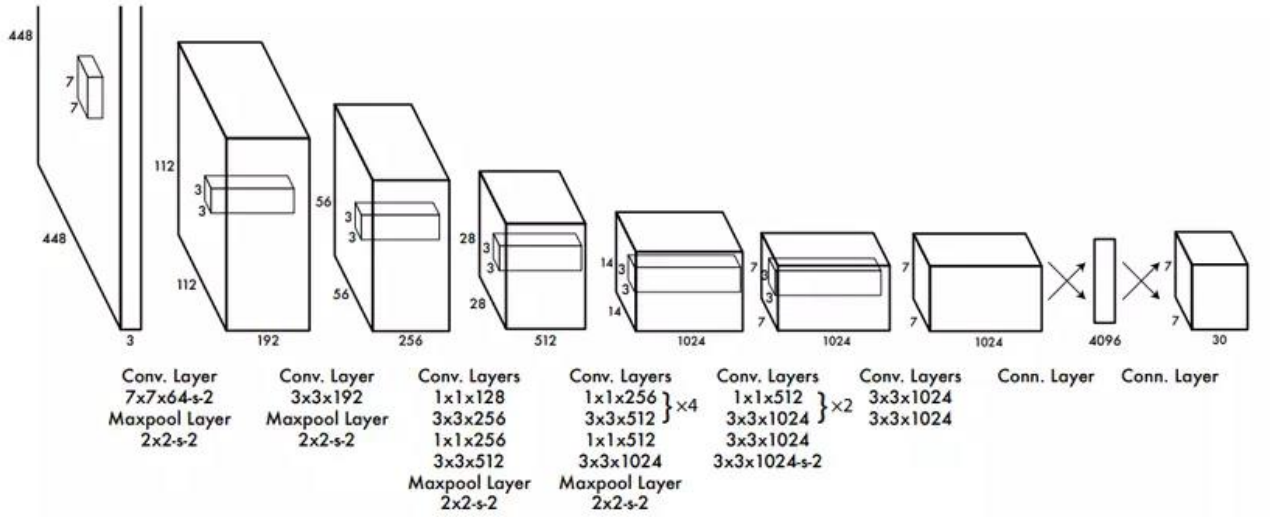
  + Năm 2016, phiên bản YOLOv2 được ông Joseph Redmon cùng công sự là ông Ali Farhadi giới thiệu trong bài nghiên cứu "*YOLO9000: Better, Faster, Stronger*"[4] (tạm dịch: "*YOLO9000: Tốt hơn, Nhanh hơn, Mạnh mẽ hơn*"). YOLOv2 được gọi là YOLO9000 vì mô hình này có khả năng nhận dạng 9000 loại đối tượng khác nhau.

  + Năm 2018, ông Joseph Redmon và ông Ali Farhadi công bố bài nghiên cứu có tên "*YOLOv3: An Incremental Improvement*"[5] (tạm dịch: "*YOLOv3: Một sự cải tiến gia tăng*"), đây là phiên bản YOLOv3 với những cải tiến từ YOLOv2 như: Darknet-53 với cách kết nối ngắn gọn, tính năng lấy mẫu cũng như phát hiện đối tượng và nguồn cấp dữ liệu trực tiếp hoặc hình ảnh tốt hơn, triển khai và cập nhật bằng cách sử dụng thư viện học sâu Keras hoặc OpenCV.

  + Năm 2020, vì trước đó Joseph Redmon đã tuyên bố dừng việc phát triển YOLO nên đã xuất hiện đội ngũ mới gồm có Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao tiếp tục nghiên cứu và công bố phiên bản YOLOv4 trong bài nghiên cứu "*YOLOv4: Optimal Speed and Accuracy of Object Detection*"[6] (tạm dịch: "*YOLOv4: Tối ưu hóa tốc độ và độ chính xác cho việc nhận diện vật thể*"). YOLOv4 có nhiều sự cải tiến đặc biệt giúp tăng độ chính xác lên cao hơn và tốc độ nhanh hơn so với phiên bản YOLOv3 trên cùng tập dataset COCO và trên GPU V100.

- Yolo là một mô hình mạng CNN (Convolutional Neural Network) dùng cho việc phát hiện, nhận dạng, phân loại đối tượng. Kiến trúc YOLO bao gồm Base network là các mạng convolution làm nhiệm vụ trích xuất đặc trưng. Phần phía sau là những Extra Layers được áp dụng để phát hiện vật thể trên feature map của base network. Thành phần Darknet Architechture được gọi là base network có tác dụng trích suất đặc trưng. Output của Base network là một feature map có kích thước 7x7x1024, sẽ được sử dụng làm input cho các extra layers có tác dụng dự đoán nhãn và tọa độ bounding box của vật thể. Base network của YOLO sử dụng chủ

yếu là các convolutional layer. Trong đó các convolutional layers sẽ trích xuất ra các đặc trưng (feature) của ảnh, còn fully-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.



Sơ đồ mô tả cấu trúc mạng CNN được dùng trong mô hình YOLO

*** Loss Function**

Loss Function của YOLO là sự tổng hợp của 3 Loss Function con.

- ***Classifycation loss***: Độ lỗi của việc dự đoán loại nhãn của object:

$$L_{classification} = \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{c \in class} (p_i(c) - \hat{p}_i(c))^2$$

Trong đó:

$\mathbb{I}_i^{obj}$: bằng 1 nếu ô vuông đang xét có object ngược lại bằng 0

$\hat{p}_i(c)$: là xác xuất có điều của lớp c tại ô vuông tương ứng mà mô hình dự đoán

- ***Localization loss***: Độ lỗi của dự đoán tọa độ tâm, chiều dài, rộng của bounding box (x, y ,w, h). Chúng ta sẽ tính Loss dự đoán tọa độ tâm (x,y) của Predicted Bounding Box và tọa độ tâm (x̂, ŷ) của Ground-Truth Bounding Box. Sau đó tính Loss dự đoán kích thước (w,h) của Predicted Bounding Box và kích thước (w^,y^) của Ground-Truth Bounding Box

Công thức tính Loss tọa độ tâm (x,y) so với (x̂, ŷ) :

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$$

Công thức tính Loss kích thước bounding box (w,h) và (w^, h^):

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$$

- ***Confidence loss***: Độ lỗi của việc dự đoán bounding box đó chứa vật thể so với nhãn thực tế tại bounding box đó.

$$L_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobject} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

b) Mô hình YOLOv4 và YOLOv4-Tiny

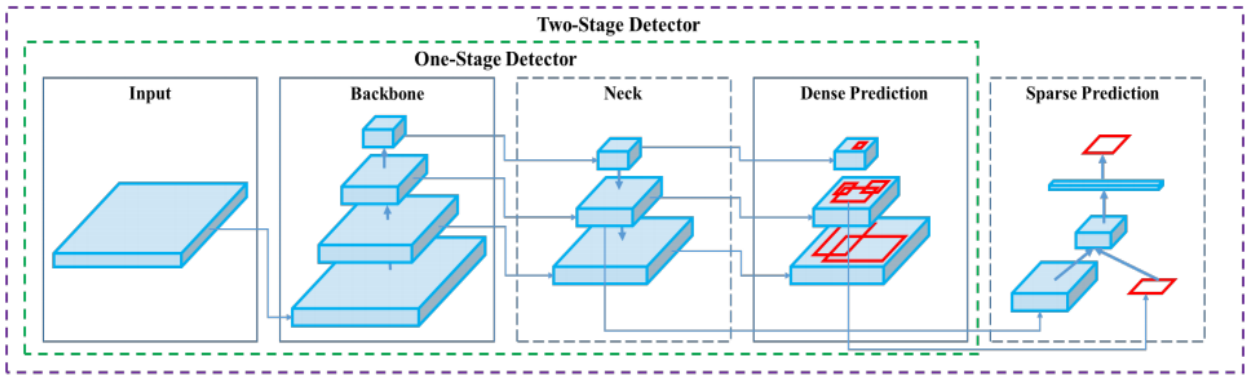
Cấu trúc của YOLOv4 gồm có 4 phần:

- Backbone (xương sống): Ở phiên bản này, tác giả chọn CSPDarknet53 làm Backbone.

- Neck (cổ): Tác giả cho phép tùy biến với các lựa chọn gồm: FPN, PAN, NAS-FPN, BiFPN, ASFF, SFAM, SSP.

- Dense prediction (dự đoán dày đặc): sử dụng các one-stage-detection như YOLO hoặc SSD.

- Sparse Prediction (dự đoán thưa thớt): sử dụng các two-stage-detection như RCNN.



Input: { Image, Patches, Image Pyramid, ... }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], ... }

Neck: { FPN [44], PANet [49], Bi-FPN [77], ... }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], ... }

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], ... }

Cấu trúc của YOLOv4

*** Phiên bản YOLOv4-Tiny:**

YOLOv4-Tiny là một phiên bản thu gọn của YOLOv4. Phiên bản này có cấu trúc dựa trên YOLOv4 nhưng đơn giản hơn và số lượng tham số (parameter) cũng được giảm bớt để có thể cài đặt trên các thiết bị đi động hoặc hệ thống nhúng. Theo thông tin trong bài nghiên cứu "*Real-time object detection method for embedded devices*"[7],

YOLOv4-Tiny có thời gian dự đoán thấp hơn, tốc độ dự đoán nhanh hơn, độ chính xác bằng khoảng 2/3 so với YOLOv4.

Nhóm chúng em chọn YOLOv4-Tiny vì sự gọn nhẹ và tốc độ dự đoán nhanh của phiên bản này sẽ phù hợp hơn với tình huống thực tế của bài toán nhóm đã đặt ra, đó là ở quầy thanh toán thường sử dụng các thiết bị gọn nhẹ và có cấu hình máy thấp hoặc trung bình mì mục đích tối ưu hóa không gian và chi phí.

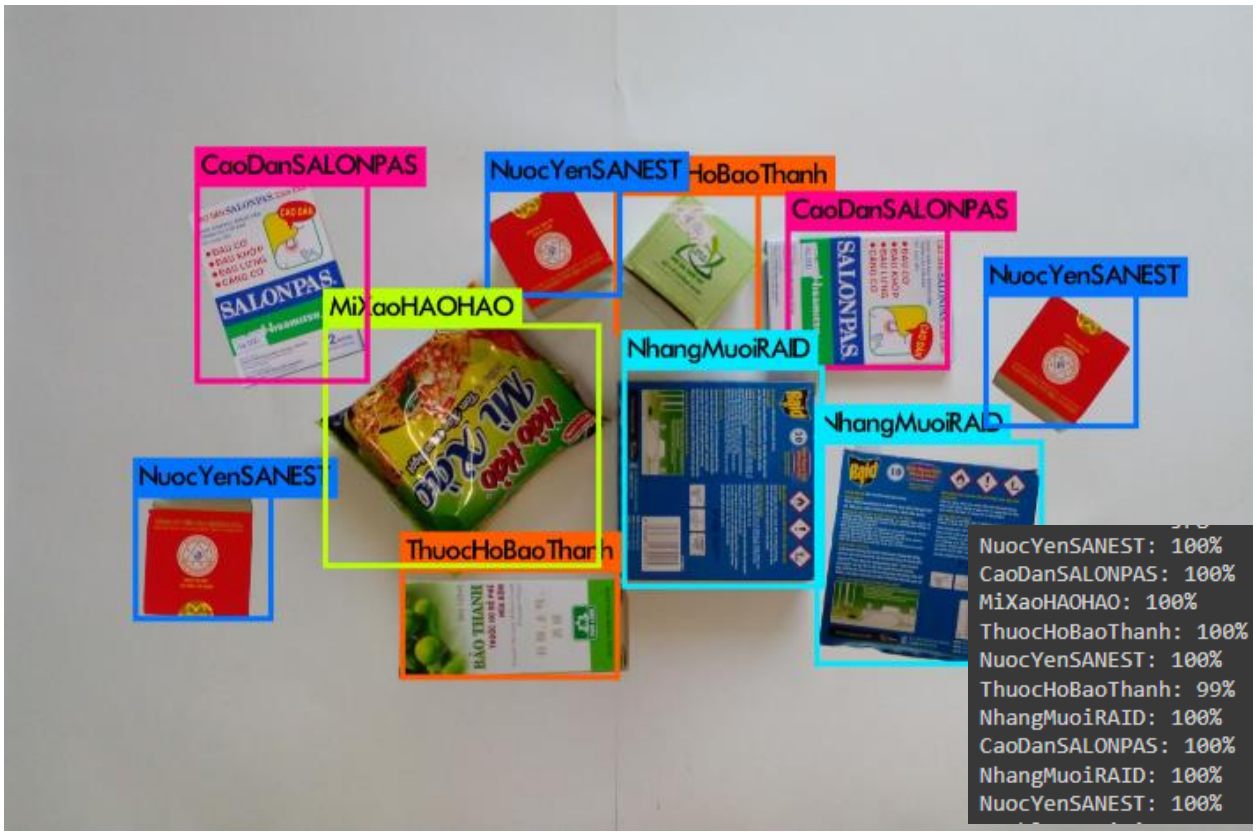
2. Huấn luyện mô hình YOLOv4-Tiny

Nhóm sử dụng mẫu **Colab Notebook**[8] được thiết kế sẵn bởi đội ngũ **Roboflow** và tiến hành huấn luyện trên bộ dữ liệu riêng của nhóm với khoảng 12000 vòng lặp (iteration).

Link Colab Notebook huấn luyện mô hình YOLOv4-Tiny của nhóm: https://colab.research.google.com/drive/1LbJ_mBykP_uydF3LwzjOrna1LBIObV9n?usp=sharing

Kết quả: Mô hình nhóm đã huấn luyện có **mAP@0.5 = 93,18%**, **loss = 0.037017**.

3. Một vài hình ảnh thực nghiệm và nhận xét



10 sản phẩm, thời gian dự đoán: 15.558000 milli-seconds

Nhận xét: hệ thống đã nhận diện chính xác tất cả 9 sản phẩm với Độ tự tin gần như là 100%

Tất cả 10 sản phẩm khác nhau, thời gian dự đoán: 15.751000 milli-seconds

Nhận xét: Hệ thống đã nhận diện được tất cả sản 10 phẩm khác nhau với Độ tự tin hầu hết là 99-100%, tuy nhiên với sản phẩm *NuocNgot7UP* thì Độ tự tin thấp hơn với 91%, có thể do đây là sản phẩm có ít dữ liệu hình ảnh hơn.

Trường hợp đè lên nhau, xếp cạnh nhau, thời gian dự đoán: 15.536000 milli-seconds

Nhận xét: 2 sản phẩm CaoDanSALONPAS tuy giống nhau và 1 sản phẩm bị che khuất vẫn có thể được dự đoán chính xác. Tuy nhiên, 2 sản phẩm XitKhuMuiROMANO xếp sát cạnh nhau lại chỉ có thể dự đoán là 1 sản phẩm.

V. TỔNG KẾT

- Với mAP@0.5 = 93.18%, Loss = 0.037017, khả năng nhận diện tất cả 10 sản phẩm có trong bộ dữ liệu cùng lúc và có thể nhận diện sản phẩm ở hầu hết các hướng xoay, mô hình YOLOv4-Tiny đã thực hiện khá tốt việc nhận diện sản phẩm theo yêu cầu của nhóm đó là: Độ chính xác trên 90%, Khả năng nhận diện tối thiểu 3 sản phẩm trong 1 hình ảnh, Nhận diện được sản phẩm ở 8 hướng xoay chính.

- Tuy nhiên, với trường hợp sản phẩm đặt chồng lên nhau hay là để sát cạnh nhau, việc nhận diện trở nên kém hơn với Độ tin cậy không cao. Do đó, khi sử dụng mô hình này, cần tránh việc đặt chồng sản phẩm, che khuất sản phẩm, đặt sản phẩm quá gần nhau.

- Nhìn chung, thông qua quá trình thử nghiệm với bộ dữ liệu nhỏ, nhóm nhận thấy việc áp dụng phương thức nhận diện sản phẩm với YOLOv4-Tiny trong quá trình mua bán tại các cửa hàng, siêu thị có độ khả thi nhất định.

- Ngoài ra, nhóm nhận thấy một vấn đề mới đó là: việc cập nhật sản phẩm ở các cửa hàng, siêu thị diễn ra thường xuyên, trong khi mô hình nhận diện sau khi huấn luyện đã bị giới hạn trong những sản phẩm của bộ dữ liệu trước đó, việc huấn luyện lại cho mỗi sản phẩm mới sẽ tốn thời gian và chi phí. Ý tưởng của nhóm để giải quyết vấn đề này đó là chỉ huấn luyện mô hình trên tập dữ liệu mới thêm, sau đó gộp chung bộ trọng số với bộ dữ liệu tổng. Nhóm hy vọng rằng ý tưởng này sẽ là hướng đi tiếp theo cho đồ án này trong tương lai.

VI. TÀI LIỆU THAM KHẢO

[1] LabelImg – URL: https://github.com/tzutalin/labelImg
[2] Roboflow – URL: https://roboflow.com/
[3] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi | "*You Only Look Once: Unified, Real-Time Object Detection*" – URL: https://arxiv.org/abs/1506.02640
[4] Joseph Redmon, Ali Farhadi | "*YOLO9000: Better, Faster, Stronger*" – URL: https://arxiv.org/abs/1612.08242
[5] Joseph Redmon, Ali Farhadi | "*YOLOv3: An Incremental Improvement*" – URL: https://arxiv.org/abs/1804.02767
[6] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao | "*YOLOv4: Optimal Speed and Accuracy of Object Detection*" – URL: https://arxiv.org/abs/2004.10934
[7] Zicong Jiang, Liquan Zhao, Shuaiyang Li, Yanfei Jia | "*Real-time object detection method for embedded devices*" – URL: https://arxiv.org/ftp/arxiv/papers/2011/2011.04244.pdf
[8] Roboflow | "YOLOv4-tiny" – URL: https://models.roboflow.com/object-detection/yolov4-tiny-darknet

HẾT