**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

**•••••**🙠󠆸🕮🙢**•••••** 

**BÁO CÁO ĐỒ ÁN**

**NHẬP MÔN THỊ GIÁC MÁY TÍNH**

Đề tài: Nhận diện lon, chai nhựa trong bài toán Object Detection

Giảng viên hướng dẫn: Mai Tiến Dũng

Sinh viên thực hiện:

+ Phạm Thiện Bảo – 20521107

+ Nguyễn Anh Dũng – 20521209  
Lớp: CS231.M21.KHCL

🙡🙢 Năm học: 2021 - 2022 🙠🙣

**Mục lục**

[**I. Giới thiệu chung về đề tài** 3](#_heading=h.1fob9te)

[**1. Lý do chọn đề tài** 3](#_heading=h.3znysh7)

[**2. Xu hướng phát triển** 3](#_heading=h.2et92p0)

[**3. Bài toán object detection** 3](#_heading=h.tyjcwt)

[**4. Phương pháp giải quyết bài toán** 4](#_heading=h.3dy6vkm)

[**II. Các khái niệm trong bài toán Object Detection** 4](#_heading=h.1t3h5sf)

[**1. Bounding box** 4](#_heading=h.4d34og8)

[**2. IOU** 5](#_heading=h.2s8eyo1)

[**3. Confidence score** 5](#_heading=h.17dp8vu)

[**III.Phương pháp đánh giá mô hình** 5](#_heading=h.3rdcrjn)

[**1. Precision - Recall** 5](#_heading=h.26in1rg)

[**2. AP** 6](#_heading=h.lnxbz9)

[**3. mAP** 6](#_heading=h.35nkun2)

[**IV. Chuẩn bị dataset** 7](#_heading=h.1ksv4uv)

[**1. Xây dựng dataset** 7](#_heading=h.44sinio)

[**2. Yolo annotations** 9](#_heading=h.z337ya)

[**V. Giới thiệu về mô hình Yolo** 9](#_heading=h.3j2qqm3)

[**1. Tổng quát** 9](#_heading=h.1y810tw)

[**2. Yolov3** 11](#_heading=h.4i7ojhp)

[**3. Yolov5** 12](#_heading=h.2xcytpi)

[**VI. Thực nghiệm** 13](#_heading=h.1ci93xb)

[**1. Kết quả** 13](#_heading=h.3whwml4)

[**2. Kết luận** 15](#_heading=h.2bn6wsx)

[**VII. Danh mục tài liệu tham khảo** 16](#_heading=h.qsh70q)

## **I. Giới thiệu chung về đề tài**

### **1. Lý do chọn đề tài**

Cuộc sống đang ngày càng phát triển hiện đại, đời sống vật chất và tinh thần của người dân được cải thiện rõ rệt. Tuy nhiên, đối lập với nó, tình trạng ô nhiễm môi trường lại có những diễn biến phức tạp và là một vấn đề cấp bách cần phải giải quyết kịp thời. Trong năm 2021, thế giới thải ra 353 triệu tấn rác thải nhựa nhưng lượng rác được tái chế chỉ đạt 9%. Việt Nam nằm trong số 20 quốc gia có lượng rác thải lớn nhất và cao hơn mức trung bình của thế giới. Riêng tại Việt Nam, trung bình mỗi năm lượng rác thải nhựa khoảng 1,8 triệu tấn. Trên thế giới, chỉ có khoảng 19% số rác được tiêu hủy và chỉ gần 50% được chôn lấp tại các hố rác đủ tiêu chuẩn.

Từ việc nhận thức sai lầm rằng rác thải sẽ do bên đơn vị thu gom rác xử lý và không có nghĩa vụ phải phân loại chúng dẫn đến việc lượng rác thải gây hại cho môi trường ngày càng tăng, ảnh hưởng vô cùng lớn đến sức khoẻ của con người. Chính vì thế, cần phải có một công cụ hỗ trợ phân loại rác để để giảm bớt chi phí cho các nhà máy tái chế, giúp thu gom và phân loại rác từ lúc chúng vừa mới được vứt bỏ.

Áp dụng lĩnh vực thị giác máy tính, ta có thể tạo ra thiết bị hỗ trợ nhận diện và phân loại nhiều loại rác thải trong cuộc sống. Tuy nhiên, nhằm giới hạn phạm vi bài toán, chúng em chỉ tiếp cận nhóm rác thải tái chế, cụ thể là lon và chai nhựa.

### **2. Xu hướng phát triển**

Đề tài có thể phát triển trong việc tạo ra một con robot đi gom rác kết hợp phân loại ở các nơi công cộng như công viên, trường học, khu dân cư hay cống rãnh, ven biển và cả dưới đại dương.

Ngoài ra, ta có thể tích hợp trong robot với hệ thống máy nghiền rác, ép lon chai để chứa được nhiều rác thải tái chế, giúp bỏ qua được các bước ép rác thủ công ở các vựa thu mua ve chai trước khi đưa về nhà máy tái chế, giúp tiết kiệm thời gian và chi phí cho các nhà máy xử lý rác khi phân loại rác…

### **3. Bài toán object detection**

Do bài toán của chúng em là nhận diện nhiều hơn một vật thể, cũng như muốn xác định được vị trí của chúng trong output trả về nên không thể dùng Classification để giải quyết bài toán. Vì thế, chúng em đã dùng phương pháp tiếp cận Object Detection.

A picture containing text, mammal

Description automatically generated

Tóm lại, bài toán object detection xác định được nhãn của đối tượng và định vị trí của các đối tượng thông qua các bounding box.

Icon

Description automatically generatedIcon

Description automatically generated

**Object Detection =** ![Shape

Description automatically generated with low confidence](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADAAAAAwCAMAAABg3Am1AAADAFBMVEUAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACzMPSIAAAAAXRSTlMAQObYZgAAAGFJREFUeF7tlkEKwCAMBI3//7NNlV7StWaFXmQHvKyZQ4RIrEDMT4th5755M1JoQOEJkVFjsEJCBgkZThBsMnQTmlcz9W4YV7/Rw/8C3zT/rAD9fAEJGSRkgBNHLydf688FXP8KV3eMyGMAAAAASUVORK5CYII=)

+ Tổng quát bài toán Object Detection:

* Input là một ảnh màu có một hoặc nhiều đối tượng.
* Output là một hoặc nhiều bounding box trong bức ảnh thể hiện nhãn và vị trí của đối tượng.

+ Trong đề tài:

* Input là một ảnh màu gồm có một hay nhiều đối tượng như lon, chai nhựa.
* Output là một hoặc nhiều bounding box trong bức ảnh thể hiện nhãn và vị trí của đối tượng lon, chai nhựa trong bức ảnh.

### **4. Phương pháp giải quyết bài toán**

Để giải quyết bài toán Object Detection, chúng ta cần phải chọn một model để thực hiện nó. Chúng em đã sử dụng mô hình Yolov3 và Yolov5 để nghiên cứu và thực nghiệm đề tài trên.

Chúng em chọn 2 phương pháp này là vì chúng em đã cố gắng thử trước đó những phương pháp hỗ trợ khác như RCNN, Fast RCNN, Faster RCNN,… nhưng không thành công. Lý do vì dataset ảnh chuẩn bị để phục vụ cho việc training mô hình đưa vào khá lớn cũng như thời gian training quá lâu, vượt quá thời gian chạy của Colab và RAM của máy tính dẫn đến việc tràn bộ nhớ trong quá trình học. Do đó, chúng em chuyển sang hướng tiếp cận mô hình Yolov3 và Yolov5 để phù hợp với khả năng của máy tính và dataset chuẩn bị.

## **II. Các khái niệm trong bài toán Object Detection**

### **1. Bounding box**

Trong object detection, chúng ta thường sử dụng bounding box để mô tả vị trí của đối tượng trong bức ảnh. Bounding box là hình chữ nhật, được xác định bởi giá trị tọa độ x của góc trên bên trái của hình chữ nhật và giá trị tọa độ y của góc dưới bên phải. Một biểu diễn hộp giới hạn thường được sử dụng khác là (x center ,y center) - trục tọa độ của tâm hộp giới hạn, chiều rộng và chiều cao của hộp.

+ Predicted bounding box: là bounding box sử dụng trong model detection, thể hiện dự đoán vật thể của model.

+ Ground Truth: là bounding box ban đầu do người dùng gán nhãn để thực hiện training.

### **2. IOU**

Intersection over Union (IOU) là một số liệu đánh giá được sử dụng để đo độ chính xác trong bài toán phát hiện đối tượng trên một tập dữ liệu cụ thể. Intersection over Union chỉ đơn giản là một thước đo đánh giá. Bất kỳ thuật toán nào cung cấp các hộp giới hạn dự đoán dưới dạng đầu ra đều có thể được đánh giá bằng IoU.

Tóm lại, nó sử dụng trong việc đánh giá xembounding box dự đoán đối tượng khớpvới ground truth thật của đối tượng hay không. Chỉ số IoU trong khoảng [0,1] và nếu IoU càng gần 1 thì bounding box dự đoán càng gần ground truth.

Graphical user interface, application

Description automatically generated

### **3. Confidence score**

Confidence score là xác xuất mà model object detection dự đoán vật thể đó. Giá trị nhằm xác định model có phát hiện chính xác vật thể hay không, cũng như biết được dự đoán của model có hiệu quả. Thông qua giá trị của confidence score, ta có thể điều chỉnh model training, căn chỉnh giá trị IOU phù hợp, chuẩn bị thêm dataset, …

## **III.Phương pháp đánh giá mô hình**

### **1. Precision - Recall**

Dựa vào một ngưỡng confidence score trong quá trình training (threshold) để xác định phát hiện đúng, phát hiện sai. Thường chọn là 0.5

**+ True Positive (TP)**: IoU lớn hơn hoặc bằng ngưỡng, là một correct detection.

**+ False Positive (FP)**: IoU bé hơn ngưỡng, là một wrong detection.

**+ False Negative (FN)**: trường hợp mà ground truth không có predicted bounding box.

+ Precision là tỷ lệ số dự đoán True positive (TP) trong tổng số dự đoán là positive.

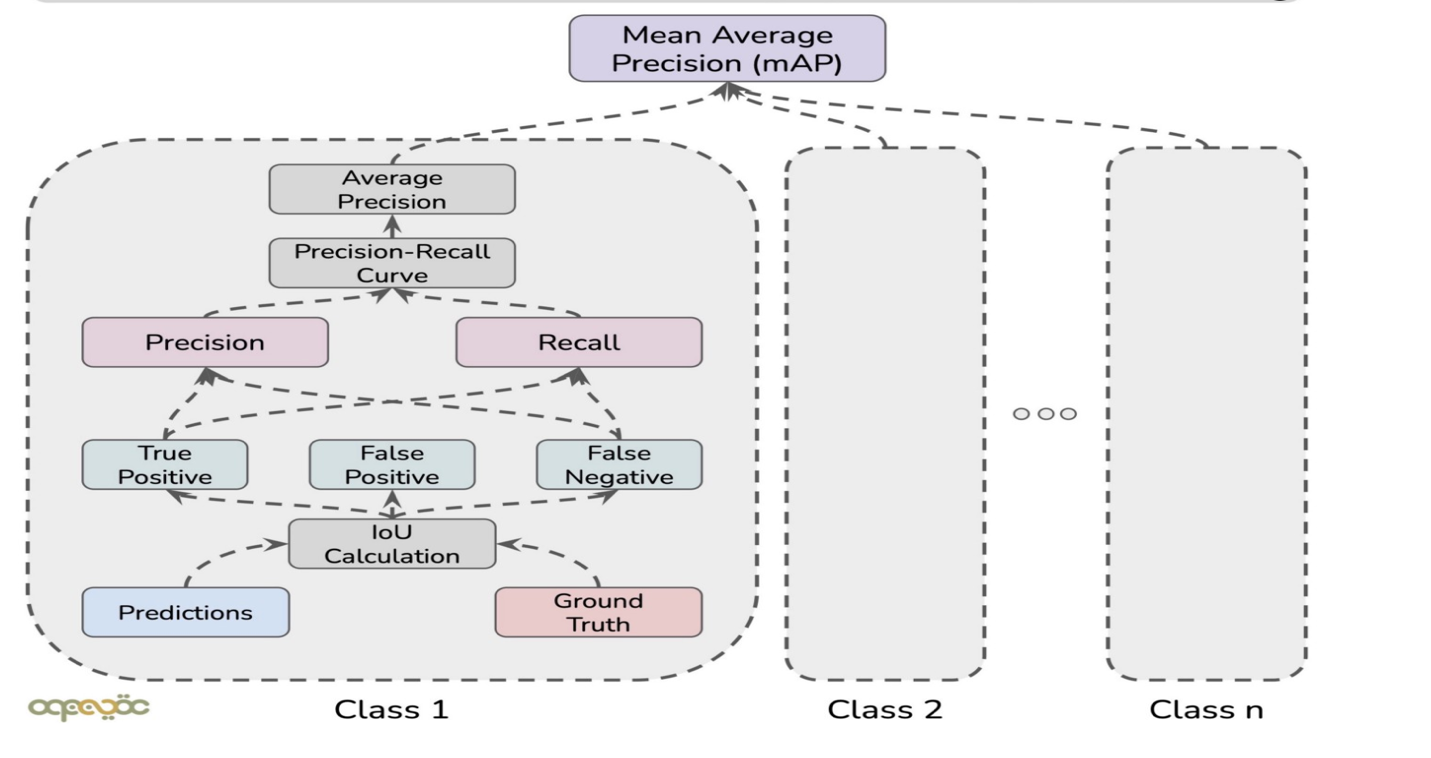
+ Recall là tỉ lệ số dự đoán True positive trong số những positive thực sự.

Tóm lại, ta có công thức:

Precision = =

Recall = =

### **2. AP**



Khi quá trình training kết thúc, ta sẽ có được các kết prediction của mỗi vật thể trong hình. Thông qua quá trình tính toán IOU để đo độ chính xác dự đoán, ta tính được giá trị TP, FP, FN. Từ đó dễ dàng tính được thông số của Precision và Recall. Hai giá trị này nhằm để vẽ được biểu đồ Precision – Recall Curve, áp dụng công thức tính để tìm được AP cho từng class.

### **3. mAP**

Bài toán Object Detection của chúng ta có một hoặc nhiều class, mỗi class ta sẽ tiến hành đo AP, sau đó lấy trung bình tất cả các giá trị AP của các class thì ta tìm được chỉ số mAP của mô hình. Do đó, mAP được hiểu là giá trị trung bình của các tất cả các class.

**+ mAP@.5:** có nghĩa là mAP trung bình khi chọn IoU = 0.5

Ví dụ: mAP@0.5 = 0.7 –> Tại IoU = 0.5, AP của mô hình là 70%.

**+ mAP@[.5:.95]** có nghĩa là mAP trung bình trên các ngưỡng IoU khác nhau, từ 0,5 đến 0,95 ,bước nhảy 0,05.

Người ta thường chọn khoảng IoU từ [.5:.95] bởi vì rất khó để predicted bounding box trùng khớp với ground truth thực sự của vật thể, dẫn tới việc kết quả luôn sai mặc dù mô hình đã dự đoán gần như tương đối chính xác vật thể.

## **IV. Chuẩn bị dataset**

### **1. Xây dựng dataset**

Để thực hiện đề tài, chúng em đã tự xây dựng bộ dataset riêng và có những ràng buộc, thông tin cụ thể là:

- Số lượng ảnh: 1267 tấm ảnh màu.

- Kích thước: 918 x 1224 → 4000 x 3000

- Vật thể trong bức hình:

+ Lon, chai của nhiều thương hiệu.

+ Hơn 1200 label chai, hơn 1500 label lon.

+ Khoảng 20% chai và khoảng 70% lon bị biến dạng nhẹ.

+ 2% lon, chai biến dạng nặng.

+ Lon, chai không xếp chồng lên nhau, nằm ngang hoặc đứng.

- Độ sáng: Trời sáng, ánh sáng đủ để nhìn thấy rõ vật thể, không bị chói.

- Background:

+ Nền gạch đường, bãi lá khô, nền cỏ, ven mép đường, dưới gốc cây…

+ Khoảng 30% ảnh có vật thể lạ: rác, bao thuốc lá, ly nhựa, hộp xốp, ống hút, bao nilon, khúc cây...

- Góc chụp: hướng nhìn từ trên xuống, cách vật thể khoảng 1m, không quá 2m.

**Train:** 1000 ảnh .

**Val:** 267 ảnh .

**Test:** 50 ảnh + 1 video (xây dựng riêng nhằm kiểm tra tính hiệu quả, chỉ đổi background còn vật thể và các ràng buộc không đổi).

**\*Một số hình ảnh:**

**+ Bình thường:**

Map

Description automatically generated

**+ Biến dạng nhẹ:**

A picture containing dirty, trash

Description automatically generated

**+ Biến dạng nặng:**

A picture containing pile

Description automatically generated

### **2. Yolo annotations**

Các phiên bản của YOLO từ v1 đến v5 khi training đều yêu cầu định dạng annotation riêng cho tập dataset.

+ Mục đích: Yolo annotations giúp thể hiện được ground truth của các vật thể trong từng bức hình trước khi đưa vào training model.

+ Công cụ sử dụng: MakeSense.AI – một website hỗ trợ gán nhãn.

+ Nội dung của file ở định dạng txt, thể hiện các thông số:

**<id-class> <center-x> <center-y> <width> <height>**

* **id-class**: Số nguyên từ 0 đến số lượng class - 1. Mỗi số nguyên tương ứng với 1 lớp.
* **center-x**: x center của bounding box.
* **center-y**: y center của bounding box.
* **width**: Chiều rộng của bounding box.
* **height**: Chiều cao của bounding box.

Các giá trị center-x, center-y, width, height đều được chuẩn hoá về khoảng giá trị [0, 1]. Mục đích của việc tạo ra các giá trị trên để giúp tỉ lệ hóa kích thước vật thể so với bức hình trước khi đưa vào model học.

Text

Description automatically generatedGraphical user interface

Description automatically generated with low confidence

## **V. Giới thiệu về mô hình Yolo**

### **1. Tổng quát**

Vài năm trở lại đây, object detection là một trong những đề tài quan trọng của deep learning bởi khả năng ứng dụng cao, dữ liệu dễ chuẩn bị và kết quả ứng dụng rất nhiều. Các thuật toán mới của object detection có thể thực hiện được các tác vụ dường như là real time, thậm chí là nhanh hơn so với con người mà độ chính xác không giảm. Trong đó, YOLO - You Only Look Once có thể không phải là thuật toán tốt nhất nhưng nó là thuật toán nhanh nhất trong các lớp mô hình object detection. Các phiên bản của mô hình này đều có những cải tiến rất đáng kể sau mỗi phiên bản.

Thuật toán Object Detection được chia thành 2 nhóm chính:

* Họ các mô hình RCNN ( Region-Based Convolutional Neural Networks) để giải quyết các bài toán về định vị và nhận diện vật thể.
* Họ các mô hình về YOLO (You Only Look Once) dùng để nhận dạng đối tượng được thiết kế để nhận diện các vật thể ở thời gian thực (real-time).

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.

\* YOLO thực hiện những bước sau:

+ ***Bước 1***: Phân chia tấm ảnh thành G×G ô lưới (grid cell).

+ ***Bước 2***: Với mỗi ô lưới, chạy một mạng CNN dự đoán các bounding box trong ô đó. Trọng tâm của vật thể sẽ được tìm trong các grid và nếu nó nằm trong ô lưới nào, thì ô lưới chứa trọng tâm của đối tượng sẽ chịu trách nhiệm tìm vật thể đó.

+ ***Bước 3***: Chạy thuật toán non-max suppression

**Các bước của non-max-suppression:**

* **Bước 1:** Đầu tiên chúng ta sẽ tìm cách giảm bớt số lượng các bounding box bằng cách lọc bỏ toàn bộ những bounding box có xác suất chứa vật thể nhỏ hơn một  ngưỡng (threshold) nào đó, thường chọn là 0.5.
* **Bước 2:** Đối với các bounding box giao nhau, non-max suppression sẽ lựa chọn ra  một bounding box có xác suất chứa vật thể là lớn nhất. Sau đó tính toán chỉ số giao thoa IoU với các bounding box còn lại. Nếu chỉ số này lớn hơn ngưỡng threshold thì điều đó chứng tỏ tỉ lệ 2 bounding boxes đang chồng lên nhau rất cao. Ta sẽ xóa các bounding có xác suất thấp hơn và giữ lại bounding box có xác suất cao nhất. Cuối cùng, ta thu được một bounding box duy nhất cho một vật thể.

A picture containing text, sky, road, way

Description automatically generated

### **2. Yolov3**

YOLOv3 dự đoán bounding box với 3 kích thước khác nhau. Hệ thống trích xuất đặc tính từ những kích thước này bằng cách sử dụng một cấu trúc tương tự mạng kim tự tháp. Từ các đặc tính cơ sở, nó thêm vài lớp convolutional.

**YOLOv3 có các cải tiến đó bao gồm:**

* **Logistic regression cho confidence score:** dự đoán predict độ tự tin của bounding box sử dụng logistic regression.
* **Backbone Darknet-53:** Backbone được thiết kế lại với việc thêm các residual blocks (kiến trúc sử dụng trong ResNet). Mạng này gồm 53 lớp tích chập kết nối liên tiếp, mỗi layer được theo sau bởi một batch normalization và một activation Leaky Relu. Để giảm kích thước của output sau mỗi convolution layer, tác giả down sample bằng các filter => có tác dụng giảm thiểu số lượng tham số cho mô hình.
* **Multi-scale prediction:** YOLOv3 sử dụng kiến trúc Feature Pyramid Networks (FPN) để đưa ra các dự đoán từ nhiều scale khác nhau của feature map.
* **Skip-layer concatenation:** YOLOv3 cũng thêm các liên kết giữa các lớp dự đoán. Mô hình upsample các lớp dự đoán ở các tầng sau và sau đó concatenate với các lớp dự đoán ở các tầng trước đó. Phương pháp này giúp tăng độ chính xác khi predict các object nhỏ.
* **Anchor box**: Anchor box là những khung hình có kích thước xác định trước, có tác dụng dự đoán bounding box. Và cùng với ground truth để tính ra sai sót giữa ground truth và anchor box bằng cách điều chỉnh các kích thước bounding box, từ đó học được các đặc điểm của vật thể.

YOLOv3 dự báo trên nhiều feature map. Những feature map ban đầu có kích thước nhỏ giúp dự báo được các object kích thước lớn. Những feature map sau có kích  thước lớn hơn trong khi anchor box được giữ cố định kích thước nên sẽ giúp dự báo các  vật thể kích thước nhỏ.

A picture containing text, dog, cage, tennis

Description automatically generated

Các feature maps của mạng YOLOv3 với input shape là 416x416, output  là 3 feature maps có kích thước lần lượt là 13x13, 26x26 và 52x52.

Trên mỗi một cell của các feature map chúng ta sẽ áp dụng 3 anchor box để dự  đoán vật thể. Như vậy số lượng các anchor box khác nhau trong một mô hình YOLO sẽ là 9 (3 featuer map x 3 anchor box).

Đồng thời trên một feature map hình vuông S x S, mô hình YOLOv3 sinh ra một  số lượng anchor box là: S x S x 3. Như vậy số lượng anchor boxes trên một bức ảnh sẽ là:

(13×13 + 26×26 + 52×52) × 3 = 10647 (anchor boxes)

Đây là một số lượng rất lớn và là nguyên nhân khiến quá trình huấn luyện mô hình YOLOv3 vô cùng chậm bởi chúng ta cần dự báo đồng thời nhãn và bounding box trên đồng thời 10647 bounding boxes.

### **3. Yolov5**

Yolov5 là sản phẩm của tác giả Glenn Jocher - nhà nghiên cứu, giám đốc điều hành của Ultralystic. Đây là tổ chức hướng tới việc giúp cho người học AI nói chung và những người đam mê công nghệ nói riêng có thể tiếp cận với các mô hình máy học một cách đơn giản, hiệu quả và trực quan hơn.

Mô hình cũng dựa trên kiến ​​trúc Yolo và sử dụng chiến lược tối ưu hóa thuật toán trong mạng nơ-ron tích chập, chẳng hạn như tùy chỉnh kích thước anchor box với từng loại dataset, sử dụng mosaic data augmentation (mỗi input image là sự kết hợp của 4 ảnh giúp context của ảnh phong phú hơn), CSPNet (giữ được một phần thông tin từ các layer trước, vừa giảm độ phức tạp của mô hình),..

Ngoài ra, mô hình sử dụng PyTorch và được ra mắt trên GitHub. Ban đầu khi vừa mới ra mắt, nó đã gây tranh cãi do không có cải thiện gì nổi bật so với Yolov4 dẫn tới việc không có bài báo khoa học chính thức cho mô hình này. Tuy nhiên, cộng đồng người sử dụng Pytorch lớn mạnh hơn Darknet dẫn tới việc sử dụng framework này giúp dễ dàng cài đặt và tích hợp trên các thiết bị IoT. Chính vì thế, nó được mọi người đón nhận cho tới ngày nay.

**\*Cấu trúc của Yolov5:**

Gồm 4 phần chính: Input, Backbone, Neck, Head

+ ***Input****:* chủ yếu chứa quá trình xử lý trước dữ liệu, bao gồm cả tăng cường dữ liệu khảm (mosaic data augmentation).

+ ***Backbone****:* CSPDarknet53 => Cấu trúc mới này giúp tăng khả năng học của mạng CNN, giảm khối lượng tính toán và giảm chi phí bộ nhớ.

+***Neck****:* tránh mất mát thông tin trong quá trình bottom-up, top- down ở quá trình xây dựng lại các layer

+ ***Head****:* Sử dụng **Transformer encoder block,** tăng khả năng phân biệt các đặc trưng, dự đoán class và bounding box. Gồm 2 tầng:

* Dense Prediction: giúp dự đoán trên toàn bộ mô hình, định vị được các bounding box => tìm được vùng có khả năng là vật thể
* Sparse Prediction: những vùng có khả năng là vật thể tiếp tục được dự đoán => trả về kết quả dự đoán cuối cùng

Chọn một mô hình đào tạo trước (Pretrained models) để bắt đầu đào tạo sẽ giúp việc đào tạo nhanh hơn, mô hình sau đó có thể được huấn luyện thêm để phù hợp với bộ dữ liệu thực tế hoặc sử dụng trực tiếp trong bài toán học máy.

Diagram

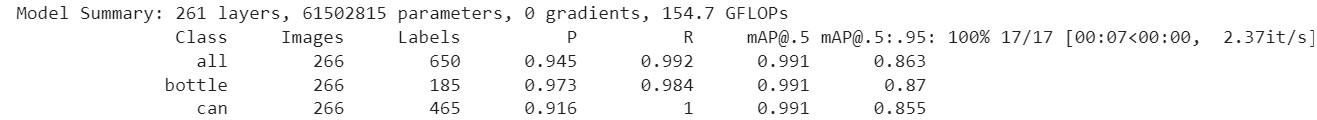
Description automatically generated

## **VI. Thực nghiệm**

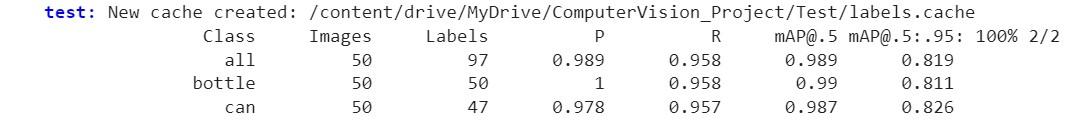
### **1. Kết quả**

**+ Yolov3:**

**Kết quả trên tập Validate:**



**Kết quả trên tập Test custom:**



**+ Yolov5:**

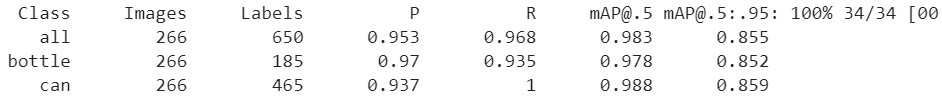
**Kết quả trên tập Validate:**

+ Áp dụng Pretrain Yolov5s:

A screenshot of a computer

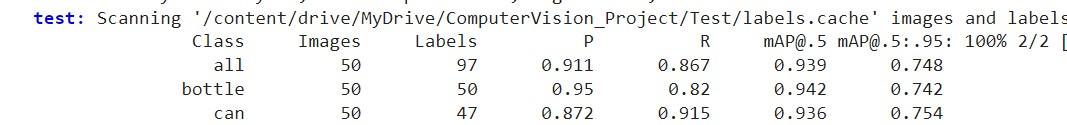
Description automatically generated with low confidence

+ Áp dụng Pretrain Yolov5m:



**Kết quả trên tập Test custom:**

+ Áp dụng Pretrain Yolov5s:



+ Áp dụng Pretrain Yolov5m:

Table

Description automatically generated

### **2. Kết luận**

Với bộ dataset chuẩn bị và việc áp dụng 2 mô hình Yolov3 và Yolov5, chúng em nhận thấy giá trị mAP khi training và kết quả của video thực nghiệm (đính kèm trong thư mục source báo cáo) của Yolov3 có kết quả chính xác **cao hơn** Yolov5.

Lý do:

+ Thứ nhất, Yolov5 có nhiều cải tiến hơn Yolov3, nó có khả năng phát hiện chính xác các vật thể nhỏ, quan tâm nhiều đến các tiểu tiết trong bức hình để trích xuất đặc trưng tốt hơn. Chính vì thế, trong bức hình xuất hiện nhiều vật thể lạ như rác thải, nắp ly, hộp xốp,… khiến Yolov5 phát hiện dễ sai hơn.

+ Thứ hai, theo như lời tác giả khuyến khích khi xây dựng dataset để sử dụng mô hình Yolov5 thì cần phải thêm một số hình ảnh không được có vật thể nhãn(background) để tăng độ chính xác, chiếm khoảng 5-10% trong bộ dataset.

+ Thứ ba, khi muốn đánh giá so sánh, người ta thường sử dụng mô hình Yolov5l – phù hợp training trên các thiết bị đám mây. Tuy nhiên, khi sử dụng Yolov5l chúng em xuất hiện hiện tượng tràn bộ nhớ nên không thể tiếp tục đánh giá trên mô hình này. Có thể nhận thấy, khi chúng em training sử dụng mô hình Yolov5s thì cho kết quả mAP dưới 0.8. Tuy nhiên, khi chuyển qua Yolov5m thì kết quả tăng lên khá cao, trên mức 0.8. Do đó, mô hình Pretrain phức tạp sẽ cho kết quả tốt hơn.

YOLO là một thuật toán rất phức tạp và bên trong nó có rất nhiều bước xử lý tính toán không đơn giản và áp dụng nhiều thuật toán máy học. Đối với đồ án này, về cơ bản đã thực hiện được mục tiêu đề ra ban đầu là cố gắng đọc hiểu và tóm tắt thuật toán, sử dụng mô hình YOLO để training và cuối cùng là đánh giá kết quả thực nghiệm. Tuy còn nhiều thiếu sót nhưng đây là nỗ lực rất lớn trong khi thời gian môn học còn hạn chế.

Như đã trình bày, đề tài không chỉ dừng lại vào việc phát hiện và phân loại một loại rác thải, nó còn có thể phát triển thêm nữa. Trong tương lai, chúng em sẽ cố gắng tối ưu các bước xử lý dữ liệu dataset cũng như mở rộng bài toán bằng các phương pháp máy học khác.

## **VII. Danh mục tài liệu tham khảo**

**[1] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. Computer Science, arXiv: 1804.02767.** <http://arxiv.org/abs/1804.02767>

**[2]** [**YOLO you only look once real time object detection explained - Manish Chablani**](https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006)

**[3]** [**YOLO object detection YOLO - forum machine learning cơ bản**](https://forum.machinelearningcoban.com/t/object-detection-yolo/503)

**[4]** [**YOLO, YOLOv2 - Jonathan hui**](https://medium.com/%40jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088)

**[5]** [**You Only Look Once: Unified, Real-Time Object Detection - Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi**](https://arxiv.org/pdf/1506.02640.pdf)

**[6]** [**YOLO9000: Better, Faster, Stronger - Joseph Redmon, Ali Farhadi**](https://arxiv.org/pdf/1612.08242.pdf)

**[7] YOLOv5 Improved YOLOv5 Based\_on Transformer Prediction Head for Object - ICCVW\_2021**

<https://openaccess.thecvf.com/content/ICCV2021W/VisDrone/papers/Zhu_TPH-YOLOv5_Improved_YOLOv5_Based_on_Transformer_Prediction_Head_for_Object_ICCVW_2021_paper.pdf?fbclid=IwAR1xVO_v_m57tgToewuQ7F33NE3rhiPIVT7JPbMoEcdy40Ol0JMzP1oTHGE>

**[8] Stanford University: Cheatsheet convolutional neural networks**

<https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-convolutional-neural-networks#:~:text=T%E1%BA%A7ng%20t%C3%ADch%20ch%E1%BA%ADp%20(CONV)%20T%E1%BA%A7ng,feature%20map%20hay%20activation%20map>.

**[9] A DEEP LEARNING OBJECT DETECTION METHOD FOR AN EFFICIENT CLUSTERS INITIALIZATION**

<https://arxiv.org/pdf/2104.13634.pdf>