

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



BÁO CÁO CUỐI KỲ
CS114 – MÁY HỌC

Đề tài: Trash Classification – Nhận diện và phân loại rác thải

Giảng viên hướng dẫn: **Ths. Phạm Nguyễn Trường An**
TS. Lê Đình Duy

Lớp: **CS114.N11.KHCL**

Nhóm sinh viên thực hiện:

- | | |
|-----------------------|----------|
| 1. Phạm Thiện Bảo | 20521107 |
| 2. Lê Văn Khoa | 20521467 |
| 3. Lê Nguyễn Tiến Đạt | 20521167 |

☞ Tp Hồ Chí Minh, Tháng 02/2023 ☞

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

This image shows a full page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

....., ngày.....tháng.....năm 20...

Người nhận xét

(Ký tên và ghi rõ họ tên)

Mục lục

| | |
|---|----|
| PHẦN I. TỔNG QUAN | 5 |
| 1. Giới thiệu đề tài | 5 |
| 2. Xu hướng phát triển..... | 6 |
| 3. Thách thức của bài toán..... | 6 |
| 4. Input – Output bài toán..... | 6 |
| 5. Nghiên cứu liên quan đến đề tài của tác giả khác | 7 |
| 1. Bài báo: “Using YOLOV5 for garbage classification” | 7 |
| 2. Bài báo của tác giả Yanwei Liu – Huawei trash detection YOLOv5 | 7 |
| PHẦN II. BỘ DỮ LIỆU | 9 |
| 1. Xây dựng bộ dữ liệu | 9 |
| 2. Tổng quát bộ dữ liệu..... | 11 |
| 3. Gán nhãn dữ liệu | 12 |
| PHẦN III. PHƯƠNG PHÁP ĐÁNH GIÁ | 14 |
| 1. Các khái niệm quan trọng..... | 14 |
| 2. IoU | 14 |
| 3. Precision – Recall | 15 |
| 4. AP | 17 |
| 5. mAP | 18 |
| PHẦN IV. GIỚI THIỆU VỀ YOLOv5 | 19 |
| 1. Tổng quát..... | 19 |
| 2. Cấu trúc của Yolov5 | 20 |
| 3. Input - Output của Yolov5..... | 21 |
| PHẦN V. THỰC NGHIỆM | 22 |
| 1. Cài đặt thực nghiệm..... | 22 |
| 2. Kết quả thực nghiệm | 22 |
| 3. Kết luận | 24 |
| 4. Hướng phát triển..... | 24 |
| PHẦN VI. SỬ DỤNG GRADIO TRIỂN KHAI MODEL | 25 |
| 1. Giới thiệu về Gradio | 25 |

| | |
|------------------------------------|----|
| 2. Code triển khai | 25 |
| 3. Giao diện kết quả..... | 25 |
| PHẦN VII. TÀI LIỆU THAM KHẢO | 26 |

PHẦN I. TỔNG QUAN

1. Giới thiệu 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. Theo các nghiên cứu, mỗi người trên thế giới trung bình sẽ phát sinh khoảng 1,2 tấn rác thải mỗi năm. Cụ thể hơn là chỉ trong năm 2021 thế giới thải ra khoảng 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.



Rác thải là một vấn đề môi trường rất quan trọng vì nó có thể gây tác động đến sức khỏe cộng đồng và môi trường xung quanh. Theo đó, rác được vứt bỏ từ những sinh hoạt trong cuộc sống con người hay trong quá trình sản xuất kinh doanh sẽ dần gây nên ô nhiễm trầm trọng nếu như không được thu gom cũng như xử lý kịp thời. Cho nên việc phân loại rác thải là một vấn đề nan giải trong sự phát triển bền vững và giữ gìn môi trường xanh sạch đẹp. Phân loại rác thải đúng có thể giúp giảm số lượng rác được đưa vào các nhà máy xử lý, tăng tỷ lệ chất thải có thể tái chế, giảm tác động xấu đến môi trường, giảm chi phí xử lý rác...

Chính vì những lý do trên, con người cần phải có một công cụ thông minh hỗ trợ phân loại rác 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 máy học thông qua sử dụng thuật toán YOLOv5, nhóm chúng em muốn tạo ra một thiết bị hỗ trợ nhận diện và phân loại rác thải sinh hoạt hằng ngày. Tuy nhiên, nhằm giới hạn phạm vi bài toán, nhóm chúng em chỉ tiếp cận và giải quyết bài toán phân loại 28 loại rác quen thuộc như lon, chai nhựa, hộp xốp, bao nylon, khẩu trang,...

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 công 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. Thách thức của bài toán

Phân loại quy mô lớn: Số lượng rác là vô cùng lớn có thể lên đến hàng chục nghìn loại dẫn đến vượt xa khả năng thông thường của máy dò đối tượng. Số lượng lớn cũng ảnh hưởng đến việc thu gom và tích trữ rác phân loại được.

Sự phức tạp của các loại rác thải: Có nhiều loại rác thải khác nhau với các đặc điểm vật lý, hình dạng và màu sắc khác nhau. Rác để lâu qua thời sẽ có những biến đổi, nó không còn hình dạng, kích thước hoặc màu sắc như ban đầu. Điều này có thể gây khó khăn cho mô hình phân loại rác bằng Machine Learning.

Nhiều loại rác bị biến dạng thì mô hình khó có thể đưa ra dự đoán chính xác cho đối tượng đó. Một số trường hợp các loại rác chất chồng lên nhau làm nhiều khả năng dự đoán của mô hình.

4. Input – Output bài toán

+ **Input**: Là một bức ảnh chụp từ trên chiếu xuống, cách mặt đất 1-2m với ánh sáng rõ ràng; trong đó bao gồm không hay nhiều đối tượng như lon, chai nhựa, hộp nhựa, túi rác,...

+ **Output**: Là bức ảnh từ input nhưng có không hoặc nhiều bounding box trong bức ảnh thể hiện nhãn và vị trí của đối tượng trong bức ảnh mà mô hình dự đoán được.



Bài báo được công bố khi tác giả tham gia cuộc thi Huawei Cloud competition 2020, Yanwei Liu đã sử dụng 3 model là Yolov5, Yolov3-tiny và Yolov4-tiny vào bộ dataset của mình.

| Tên dataset | TACO dataset | HUAWEI dataset |
|---------------|--------------|----------------|
| Train | 1189 images | 9410 images |
| Validation | 297 images | 2349 images |
| Total dataset | 1486 images | 11759 images |



Huawei dataset

b) Kết quả của bài báo

Tác giả đã sử dụng 3 model khác nhau là Yolov3-tiny, Yolov4-tiny và Yolov5 cho bài toán của mình và thu được kết quả như sau:

| Model | mAP@0.5 | AVG FPS |
|-------------|---------|---------|
| Yolov3-tiny | 46.98% | 6.2 |
| Yolov4-tiny | 58.03% | 12.3 |
| Yolov5 | 68% | 5.6 |

Về độ chính xác thì model Yolov5 cho ra kết quả lớn nhất với $68\% > \text{Yolov4 tiny}(58.03\%) > \text{Yolov5-tiny}(46.98\%)$

PHẦN II. BỘ DỮ LIỆU

1. Xây dựng bộ dữ liệu

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: 1500 tấm ảnh màu.
- Kích thước: 918 x 1224 \rightarrow 4000 x 3000
- Vật thể trong bức hình: gồm 28 loại rác khác nhau, quy định sử dụng tên tiếng Anh của rác.

| Class | Tên Tiếng Anh | Tên Tiếng Việt |
|-------|----------------|-----------------|
| 0 | bottle | chai |
| 1 | can | lon |
| 2 | glass bottle | chai thủy tinh |
| 3 | plastic bag | bao nylon |
| 4 | paper | giấy |
| 5 | carton box | thùng giấy |
| 6 | foam box | hộp xốp |
| 7 | plastic glass | ly nhựa |
| 8 | drinking straw | ống hút |
| 9 | tobaco | bao thuốc lá |
| 10 | plastic spoon | muỗng nhựa |
| 11 | plastic plate | đĩa nhựa |
| 12 | plastic bowl | chén nhựa |
| 13 | plastic cap | nắp nhựa |
| 14 | plastic box | hộp nhựa |
| 15 | milk bottle | hộp sữa |
| 16 | zipper bag | túi zip |
| 17 | trash bag | túi rác hỗn tạp |

| | | |
|----|------------------------|---------------|
| 18 | mask | khẩu trang |
| 19 | sausage package | vỏ xúc xích |
| 20 | snack package | vỏ túi snack |
| 21 | clothes | vải (quần áo) |
| 22 | coconut | trái dừa |
| 23 | broom | chổi |
| 24 | instant noodle package | bao mì tôm |
| 25 | pillbox | vỏ thuốc |
| 26 | PP woven bag | bao dệt pp |
| 27 | shoes | giày dép |

- Độ sáng: Trời sáng, ánh sáng đủ để nhìn thấy rõ vật thể; không bị chói, bị nhòe.
- Background: Nền gạch đường, bãi lá khô, nền cỏ, ven mép đường, dưới gốc cây, quanh bãi rác...
- Góc chụp: hướng nhìn từ trên xuống, cách vật thể khoảng 1m, không quá 2m.

Link chi tiết dataset: <https://by.com.vn/WqTtu>



| | |
|------------------------|-----|
| plastic plate | 92 |
| plastic bowl | 45 |
| plastic cap | 67 |
| plastic box | 103 |
| milk bottle | 51 |
| zipper bag | 56 |
| trash bag | 427 |
| mask | 34 |
| sausage package | 25 |
| snack package | 67 |
| clothes | 44 |
| coconut | 13 |
| broom | 14 |
| instant noodle package | 18 |
| pillbox | 17 |
| PP woven bag | 57 |
| shoes | 14 |

3. Gán nhãn dữ liệu

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

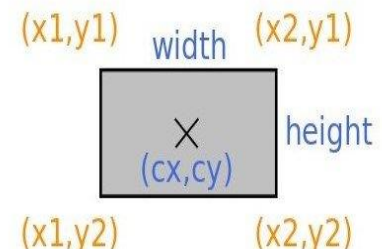
- + Mục đích: Yolo annotations giúp thể hiện được ground truth của các vật thể trong từng bức hình trước khi đưa vào training model.
- + Công cụ sử dụng: MakeSense.AI – một website hỗ trợ gán nhãn.
- + Nội dung của file ở định dạng txt, thể hiện các thông số:

<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.

| | | | | | | |
|--------|-------------|---|---------------|----------|-----------------------|----------|
| 4 bbox | 1 | 2 | 0.414500 | 0.574667 | 0.361000 | 0.578667 |
| | 2 | 2 | 0.517000 | 0.482667 | 0.284000 | 0.589333 |
| | 3 | 2 | 0.560000 | 0.409333 | 0.264000 | 0.536000 |
| | 4 | 2 | 0.630500 | 0.324000 | 0.303000 | 0.397333 |
| | class index | | center <x, y> | | scale <width, height> | |



The diagram shows a gray rectangle representing a bounding box. The top-left corner is labeled (x1,y1), the top-right is (x2,y1), the bottom-left is (x1,y2), and the bottom-right is (x2,y2). The center is marked with a cross and labeled (cx,cy). The horizontal dimension is labeled 'width' and the vertical dimension is labeled 'height'.

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.

PHẦN III. PHƯƠNG PHÁP ĐÁNH GIÁ

1. Các khái niệm quan trọng

+ Bounding box:

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.

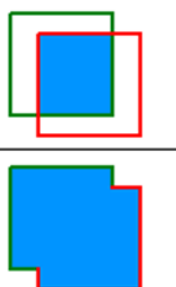
+ Confidence score:

Confidence score là xác suất 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, ...

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 xem bounding box dự đoán đối tượng khớp với 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.

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


3. Precision – Recall

Dựa vào một ngưỡng **confidence score** trong quá trình training 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: tỉ lệ số dự đoán True positive (TP) trong tổng số dự đoán là positive.

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

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

$$Recall = \frac{TP}{TP + FN} = \frac{\text{Tổng số lần dự đoán chính xác}}{\text{Tổng số lần dự đoán đúng có thể có}}$$

*Ví dụ về cách tính:



Hình bên trái là Predict, bên phải là GroundTruth của hình IMG1220 trong dataset
Sau khi detect, ta có kết quả dự đoán của mô hình trả về như sau:

| | xmin | ymin | xmax | ymax | confidence | class | name |
|---|-------------|-------------|-------------|-------------|------------|-------|---------------|
| 0 | 2113.780762 | 427.300049 | 3001.625244 | 1034.012817 | 0.910353 | 0 | bottle |
| 1 | 2186.543701 | 1776.300171 | 2914.270020 | 2336.337158 | 0.908171 | 2 | glass bottle |
| 2 | 833.275513 | 789.768616 | 1557.267700 | 1573.054077 | 0.867432 | 20 | snack package |
| 3 | 1072.071289 | 1946.171997 | 1915.006104 | 2580.460938 | 0.861454 | 3 | plastic bag |

Bước 1: Xác định tọa độ của Ground Truth và Bounding Box của hình IMG 1220

Ta có kích thước của ảnh là 4032 x 3024, vì vậy lần lượt chia các x cho 4032 và y cho 3024. Ta được **tọa độ bounding box dự đoán**:

| X1 | Y1 | X2 | Y2 | class |
|-------|-------|-------|-------|-------|
| 0.52 | 0.141 | 0.744 | 0.342 | 0 |
| 0.54 | 0.587 | 0.722 | 0.772 | 2 |
| 0.206 | 0.26 | 0.386 | 0.52 | 20 |
| 0.265 | 0.482 | 0.475 | 0.853 | 3 |

Tọa độ của ground truth khi gán nhãn có dạng (x_center, y_center, width, height)

```
20 0.289624 0.386404 0.193202 0.257603
0 0.630411 0.237030 0.230769 0.209302
3 0.368113 0.747764 0.216011 0.218247
2 0.631082 0.668157 0.186494 0.194991
```

Ta sẽ đổi về định dạng (x1, y1, x2, y2) như của bounding box dự đoán:

```
def yolo_to_pascal_voc(x_center, y_center, w, h, image_w, image_h):
    w = w * image_w
    h = h * image_h
    x1 = ((2 * x_center * image_w) - w)/2
    y1 = ((2 * y_center * image_h) - h)/2
    x2 = x1 + w
    y2 = y1 + h
    return [x1, y1, x2, y2]
print(yolo_to_pascal_voc(0.63, 0.237, 0.23, 0.209, 1, 1))
print(yolo_to_pascal_voc(0.63, 0.668, 0.186, 0.194, 1, 1))
print(yolo_to_pascal_voc(0.2896, 0.386, 0.193, 0.2576, 1, 1))
print(yolo_to_pascal_voc(0.368, 0.7477, 0.216, 0.218, 1, 1))

[0.515, 0.1325, 0.745, 0.3415]
[0.537, 0.5710000000000001, 0.7230000000000001, 0.7650000000000001]
[0.19310000000000002, 0.2572, 0.3861, 0.5147999999999999]
[0.26, 0.6387, 0.476, 0.8567]
```

Sau khi đổi thành định dạng (x1, y1, x2, y2), ta có được bảng sau thể hiện **tọa độ của groundtruth**:

| X1 | Y1 | X2 | Y2 | class |
|-------|--------|--------|--------|-------|
| 0.515 | 0.1325 | 0.745 | 0.3415 | 0 |
| 0.537 | 0.571 | 0.723 | 0.765 | 2 |
| 0.193 | 0.2572 | 0.3861 | 0.5148 | 20 |
| 0.26 | 0.6387 | 0.476 | 0.8567 | 3 |

Bước 2: Tính IOU cho từng class xuất hiện trong ảnh

Ta tính IoU cho mỗi class:

$$\text{IoU}_{\text{class0}} = 0.9321$$

$$\text{IoU}_{\text{class2}} = 0.8671$$

$$\text{IoU}_{\text{class20}} = 0.904$$

$$\text{IoU}_{\text{class3}} = 0.5625$$

Bước 3: So sánh IoU của mỗi class với threshold mà ta đã định trước

Ta đặt threshold = 0.4 và lần lượt so sánh từng IoU của các class với threshold

Nếu $\text{IoU} > \text{threshold}$ thì đó là 1 TP (True Positive) và FP (False Positive) nếu $\text{IoU} < \text{threshold}$. Sau đó ta sẽ tính Precision và Recall theo công thức đã trình bày ở trên.

Từ đó, ta thu được 1 bảng kết quả hoàn chỉnh:

| Class | IoU | TP | FP | cumTP | cumFP | All_detections | P | R |
|-------|--------|----|----|-------|-------|----------------|---|------|
| 0 | 0.9321 | 1 | 0 | 1 | 0 | 1 | 1 | 0.25 |
| 2 | 0.8671 | 1 | 0 | 2 | 0 | 2 | 1 | 0.5 |
| 20 | 0.904 | 1 | 0 | 3 | 0 | 3 | 1 | 0.75 |
| 3 | 0.5625 | 1 | 0 | 4 | 0 | 4 | 1 | 1 |

+ Chú thích:

1. cumTP : tổng số lượng các TP
2. cumFP : tổng số lượng các FP
3. all ground-truths = 4 tương ứng với 4 class

Kết luận:

Sau khi áp dụng YOLO thì trong hình có 4 class được dự đoán là bottle, glass bottle, snack package và plastic bag và có tổng cộng 4 bounding box được dự đoán tương ứng với 4 class lần lượt với confidence score là 0.91, 0.91, 0.87 và 0.86

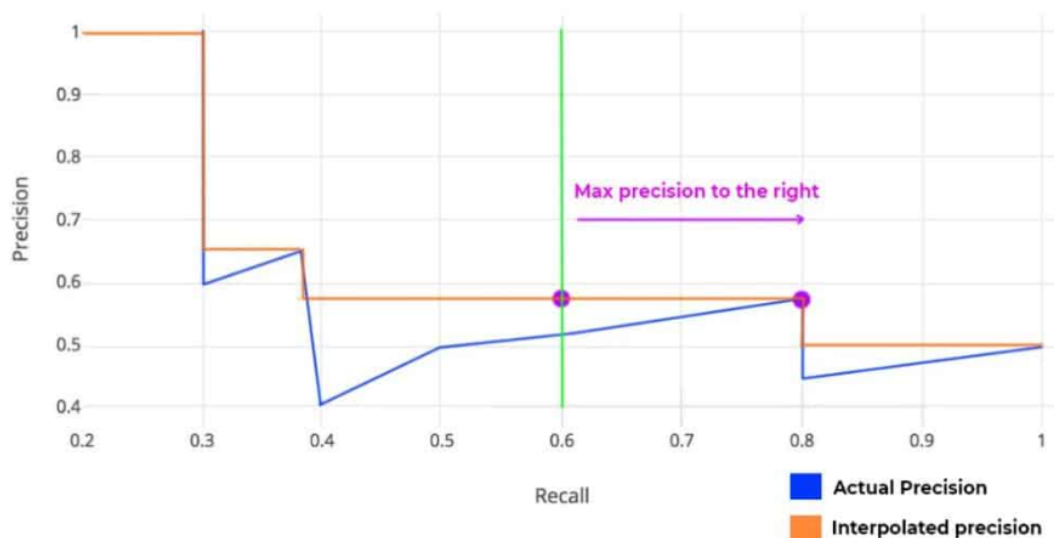
4. AP

AP: là chỉ số có quan hệ mật thiết với chỉ số Precision (phần trăm bounding box được dự đoán đúng) và Recall (tỉ lệ phần trăm các bounding box được đoán đều chính xác).

+ AP50: là độ chính xác với $\text{IoU} = 0.5$

+ AP75: là độ chính xác với $\text{IoU} = 0.75$

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.



5. mAP

Bài toán 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.

+ $\text{mAP}@.5$: có nghĩa là mAP trung bình khi chọn $\text{IoU} = 0.5$

Ví dụ: $\text{mAP}@0.5 = 0.7 \rightarrow$ Tại $\text{IoU} = 0.5$, AP của mô hình là 70%.

+ $\text{mAP}@[.5:.95]$ 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 chính xác vật thể.

PHẦN IV. GIỚI THIỆU VỀ YOLOv5

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 $G \times G$ 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.

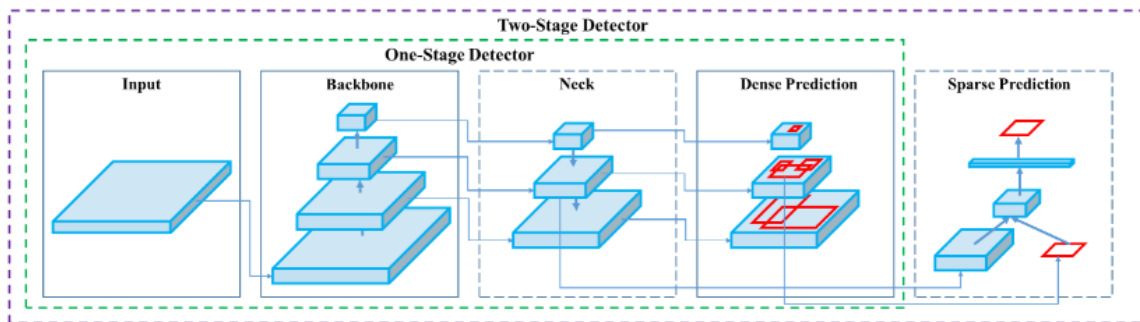


2. Cấu trúc của Yolov5

Bao gồm 3 phần chính:

- Backbone: Backbone là 1 mô hình pre-train của 1 mô hình học chuyên (transfer learning) khác để học các đặc trưng và vị trí của vật thể. Các mô hình học chuyên thường là VGG16, ResNet-50,...
- Head: Phần head được sử dụng để tăng khả năng phân biệt đặc trưng để dự đoán class và bounding-box. Ở phần head có thể áp dụng 1 tầng hoặc 2 tầng:
 - ♦ Tầng 1: Dense Prediction, dự đoán trên toàn bộ hình với các mô hình RPN, YOLO, SSD,...
 - ♦ Tầng 2: Sparse Prediction dự đoán với từng mảng được dự đoán có vật thể với các mô hình R-CNN series,...
- Neck: Ở phần giữa Backbone và Head, thường có thêm một phần Neck. Neck thường được dùng để làm giàu thông tin bằng cách kết hợp thông tin giữa quá trình bottom-up và quá trình top-down (do có một số thông tin quá nhỏ khi đi

qua quá trình bottom-up bị mất mát nên quá trình top-down không tái tạo lại được).



Cấu trúc của YOLOv5

3. Input - Output của YOLOv5

Input: Đầu vào của mô hình là một bức ảnh màu.

Output: là một véc tơ sẽ bao gồm các thành phần:

$$y^T = [\rho_0, \underbrace{\langle t_x, t_y, t_w, t_h \rangle}_{\text{boundingbox}}, \underbrace{\langle p_1, p_2, \dots, p_c \rangle}_{\text{score of } c \text{ classes}}]$$

Trong đó:

ρ_0 : là xác suất dự báo vật thể xuất hiện trong bounding box.

$\langle t_x, t_y, t_w, t_h \rangle$: giúp xác định bounding box. Trong đó t_x, t_y là tọa độ tâm và t_w, t_h là kích thước rộng, dài của bounding box.

$\langle p_1, p_2, \dots, p_c \rangle$: là véc tơ phân phối xác suất dự báo của các classes.
score of c classes

PHẦN V. THỰC NGHIỆM

1. Cài đặt thực nghiệm

- Sử dụng Google Colab và ngôn ngữ Python để tiến hành thực nghiệm bài toán.
- Khi sử dụng mô hình Yolov5, chúng em sử dụng pretrained model Yolov5s có sẵn. Lý do vì đây là mô hình vừa phải để train, sử dụng mô hình pretrained phức tạp hơn sẽ khiến xảy ra hiện tượng tràn bộ khi sử dụng bộ nhớ GPU hạn chế của Google Colab.
- Vì bộ nhớ có hạn, chúng em thực hiện resize ảnh về kích thước 640*640 để tránh trường hợp tràn bộ nhớ.

2. Kết quả thực nghiệm

Số lần training với epoch = 100, thì kết quả của mô hình như sau:

| Độ đo | Kết quả |
|---------------------|---------|
| mAP (IoU[0.5]) | 0.83 |
| mAP (IoU[0.5:0.95]) | 0.56 |

Một số kết quả trên tập test:





3. Kết luận

Với bộ dataset chuẩn bị và việc áp dụng mô hình Yolov5, chúng em nhận thấy kết quả đánh giá mAP (IoU[0.5:0.95]) là 0.56, thấp hơn với các kết quả của các nghiên cứu khác.

Lý do:

+ Thứ nhất, Yolov5 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 bộ dataset của chúng em có xuất hiện nhiều loại rác thải bị biến dạng, xếp chồng lên nhau khiến Yolov5 khó khăn hơn trong việc detect. Trong khi đó, các vật thể rác trong bộ dataset mà các tác giả trước nghiên cứu hầu như không bị biến dạng và được đặt ở background không có nhiều vật thể nhiễu như sỏi đá, lá cây... Do đó, kết quả trả về của chúng em sẽ có phần thấp 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 nhân (background) để tăng độ chính xác, chiếm khoảng 5-10% trong bộ dataset. Trong bộ dataset của chúng em, tất cả các hình đều có một hoặc nhiều object khác nhau và không có tấm nào là background. Chính vì vậy có thể đã làm ảnh hưởng một phần đến kết quả.

+ Thứ ba, khi muốn đánh giá để có kết quả tốt nhất, người ta thường sử dụng mô hình pretrained có kích thước lớn như Yolov5l, Yolov5n – 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.

4. Hướng phát triển

+ Về data:

- Xây dựng dataset có thêm nhiều loại rác khác
- Tìm hiểu và nghiên cứu thêm về các quy tắc xây dựng dataset để nâng cao chất lượng bộ dữ liệu, gom các loại rác liên quan vào cùng 1 nhóm nhằm giảm thiểu số lớp để học, thuận tiện cho việc xử lý.

+ Về model:

- Tiến hành cài đặt và thử nghiệm trên các đời Yolo mới hơn như v7, v8
- Sử dụng pretrained model phức tạp hơn để train.

PHẦN VI. SỬ DỤNG GRADIO TRIỂN KHAI MODEL

1. Giới thiệu về Gradio

Gradio là một thư viện Python được sử dụng để xây dựng các giao diện người dùng đơn giản cho các mô hình machine learning. Nó cung cấp cho người dùng các công cụ để tạo ra các giao diện đẹp mắt và dễ sử dụng cho các mô hình machine learning, cho phép người dùng tương tác với mô hình và thấy kết quả ngay lập tức.

Để sử dụng Gradio, người dùng chỉ cần định nghĩa một hàm Python chấp nhận các đối tượng dữ liệu đầu vào và trả về kết quả đầu ra, sau đó sử dụng Gradio để tạo ra giao diện cho hàm đó. Gradio cung cấp nhiều thành phần giao diện khác nhau, cho phép người dùng chọn giá trị từ một danh sách, tải lên hình ảnh hoặc văn bản, hay sử dụng một thanh trượt để điều chỉnh các tham số đầu vào...

2. Code triển khai

```
import gradio as gr
import torch
import cv2
from PIL import Image
# Load YOLOv5 model
model = torch.hub.load('ultralytics/yolov5', 'custom', path= '/content/drive/MyDrive/CS114/yolov5/runs/train/exp3/weights/best.pt')

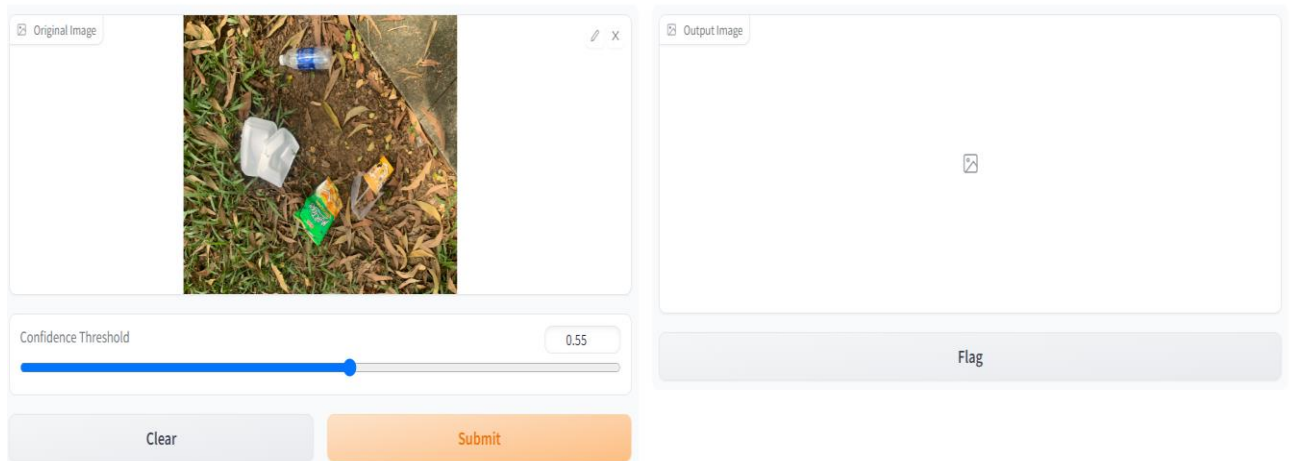
# Define input and output interfaces
def detect_objects(input_image, confidence_threshold):
    size=640
    input_image = input_image.resize((int(x * g) for x in input_image.size), Image.ANTIALIAS)
    output_image = model(input_image, confidence=confidence_threshold)
    output_image.render()
    return Image.fromarray(output_image.imgs[0])

input_img = gr.inputs.Image(type='pil', label="Original Image")
confidence_threshold = gr.inputs.Slider(minimum=0.0, maximum=1.0, default=0.45, label="Confidence Threshold")
output_interface = gr.outputs.Image(type='pil', label="Output Image")
gr.Interface(fn = detect_objects, inputs=[input_img, confidence_threshold], outputs=output_interface, title="Trash Classification").launch()
```

3. Giao diện kết quả

Sau khi load hình lên, ta có thể thay đổi threshold tùy vào ý muốn trước khi đưa vào mô hình để detect.

Trash Classification



PHẦN VII. TÀI LIỆU THAM KHẢO

- [1] Dai, Yuan, et al. "YOLO-Former: Marrying YOLO and Transformer for Foreign Object Detection." *IEEE Transactions on Instrumentation and Measurement* 71 (2022): 1-14.
- [2] Wu, Ziliang, et al. "Using YOLOv5 for garbage classification." *2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. IEEE, 2021.
- [3] Dong-e, Zhao, et al. "Research on garbage classification and recognition based on hyperspectral imaging technology." *Spectroscopy and Spectral Analysis* 39.3 (2019): 917-922.
- [4] Stanford University: Cheatsheet convolutional Neural Networks
[https://stanford.edu/~shervine/1/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](https://stanford.edu/~shervine/1/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).
- [5] Couturier, Raphaël, et al. "A deep learning object detection method for an efficient clusters initialization." *arXiv preprint arXiv:2104.13634* (2021).
- [6] Lv, Zhaohao, Huiyan Li, and Yeming Liu. "Garbage detection and classification method based on YoloV5 algorithm." *Fourteenth International Conference on Machine Vision (ICMV 2021)*. Vol. 12084. SPIE, 2022.
- [7] Yan, Xiaobo, et al. "A Garbage Classification Method Based on Improved YOLOv5." *2022 International Conference on Networks, Communications and Information Technology (CNCIT)*. IEEE, 2022.