

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 MÁY HỌC**

Đề tài: AGERATUM CONYZOIDES DETECTION

Sinh viên thực hiện:

Nguyễn Anh Dũng – 20521209

VÕ MINH TRÍ - 20520821

☞ Năm học: 2022 - 2023 ☜

I. Giới thiệu chung về đề tài

1. Lý do chọn đề tài

Đây là một bài toán nhận diện cây hoa cút lợn để dùng chữa bệnh trong y học cổ truyền như mụn nhọt, viêm họng, rong huyết, đau nhức xương khớp, phong thấp, viêm xoang và nhiều bệnh viện đã sử dụng các chế phẩm của cỏ cút lợn để điều trị viêm xoang mũi mãn tính và dị ứng.

Ngoài ra người ta còn dùng cỏ cút lợn phối hợp với bồ kết nấu nước gội đầu cho thơm, trơn tóc, sạch gàu.

Thậm chí khi tra trên google thì đến 90% nếu như không muốn nói hầu hết đang cho hai cái tên như kể trên chỉ một loại đó là cây cút lợn.

Do đó đây cũng là lý do chúng tôi làm về đề tài này.

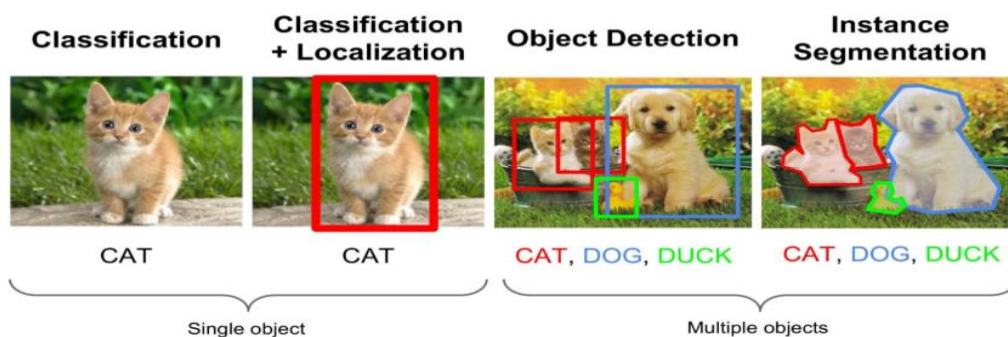
Do đó nhóm chúng em đã mong muốn sử dụng máy học để phân biệt loại cây này dựa trên vào hoa và lá trên cây

2. Xu hướng phát triển

Đề tài có thể phát triển tiếp để người dân phát triển tiếp nhận biết cây cút lợn và cây cỏ dại. bài toán này có thể mở rộng thành bài toán phát hiện giữa cây cỏ dại và cây thuốc cho các trẻ em và người thiếu kinh nghiệm nhìn loại cây để có thể dùng với các mục đích tốt nhất

Hiện nay chưa có nhiều bài báo hay nhiều nguồn trích dẫn để phân biệt cây cút lợn so với các loại cây khác như cây cỏ dại, cây ngũ sắc.

3. Bài toán object detection



Bài toán object detection xác định được nhãn của đối tượng và định vị trí của các đối tượng thông qua các bounding box.

+ 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 cần phát hiện.
- Output là một hoặc nhiều bounding box trong bức ảnh thể hiện nhãn và vị trí của đối tượng như cây chuối.

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 Yolov5 để giải quyết vấn đề này.

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.

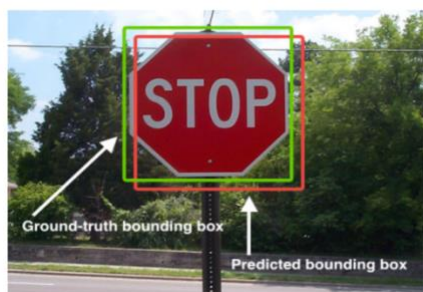
+ Predicted bounding box: là bounding box sử dụng trong model detection, thể hiện dự đoán vật thể của model.

+ 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 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.



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

3. 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, ...

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.

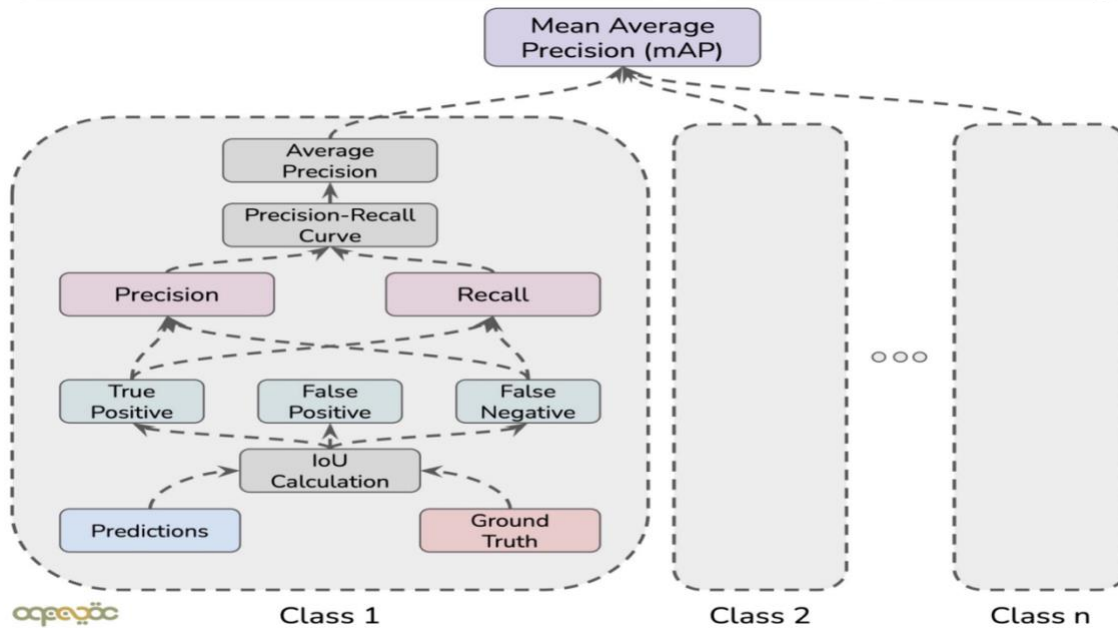
+ Recall là tỉ lệ số dự đoán True positive trong số những positive thực sự.

Tóm lại, ta có công thức:

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

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

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 $\text{IoU} = 0.5$

Ví dụ: $\text{mAP}@0.5 = 0.7 \rightarrow$ Tại $\text{IoU} = 0.5$, 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: 1025 tấm ảnh màu.
- Vật thể trong bức hình:
 - Gồm 1 hay nhiều cây cứt lợn
- Độ sáng: Trời sáng, ánh sáng đủ để nhìn thấy rõ vật thể, không bị chói.
- Background:
 - +Sau vườn gồm những loại cây khác
- Góc chụp: hướng nhìn từ trên xuống, cách vật thể khoảng 1m, không quá 2m.

Train: 918 ảnh .

Val: 90 ảnh .

***Một số hình ảnh:**



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.

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 đường 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.



3. YOLOv5

YOLOv5 là sản phẩm của tác giả Glenn Jocher - nhà nghiên cứu, giám đốc điều hành của Ultralytics. 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).

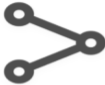

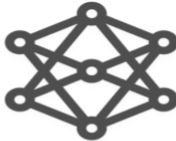
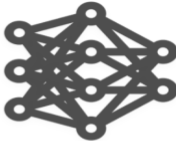
+ **Backbone**: CSPDarknet53 => Cấu trúc mới này giúp tăng khả năng học của mạng CNN, giảm khối lượng tính toán và giảm chi phí bộ nhớ.

+ **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.

			
Small YOLOv5s	Medium YOLOv5m	Large YOLOv5l	XLarge YOLOv5x
14 MB _{FP16} 2.2 ms _{V100} 36.8 mAP _{COCO}	41 MB _{FP16} 2.9 ms _{V100} 44.5 mAP _{COCO}	90 MB _{FP16} 3.8 ms _{V100} 48.1 mAP _{COCO}	168 MB _{FP16} 6.0 ms _{V100} 50.1 mAP _{COCO}

VI. Thực nghiệm

1. Kết quả

+ **Yolov5**:

```
test: Scanning /content/drive/.shortcut-targets-by-id/1vt4qnPC6V5S00w69_hQBAGyvnGBASBdE/CS114/data_test/labels.cache... 108 images, 0 backgrounds,
      Class  Images  Instances    P      R   mAP50  mAP50-95: 100% 4/4 [00:41<00:00, 10.41s/it]
      all     108      108      1      1    0.995    0.827
Speed: 0.2ms pre-process, 11.5ms inference, 3.1ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs/val/exp
```

2. Kết luận

Với bộ dataset chuẩn bị và việc áp dụng mô hình Yolov5, chúng tôi nhận thấy kết quả dự đoán khá cao, chúng tôi đã thử những tấm ảnh lấy từ nguồn trên mạng kết quả nó

dự đoán được. Tuy nhiên khi có quá nhiều cây trong tấm ảnh thì có thể nó sẽ ko dự đoán được.

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 đề 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 cây cứt lợn, 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

[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_m57tgToewuQ7F33NE3rhiPIVT7JPbMoEcdy40O10JMzP1oTHGE

[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.](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>