NGHIÊN CỨU VÀ ỨNG DỤNG THUẬT TOÁN YOLOV7 ĐỂ PHÂN LOẠI CÀ CHUA

Nguyễn Văn Mạnh, Lê Văn An, Trần Đức Lương, Nguyễn Tuấn Tú
Nguyễn Thị Thanh Huyền, Ngô Thị Bích Thúy, Nguyễn Thái Cường*
Trường Đại học Công nghiệp Hà Nội

Tóm tắt

Hiện nay, trí tuệ nhân tạo và học máy đang ngày càng phát triển và được ứng dụng rộng rãi trong nhiều lĩnh vực trong đời sống. Trí tuệ nhân tạo có khả năng xử lý dữ liệu khổng lồ, học hỏi từ dữ liệu và đưa ra quyết định một cách tự động. Trong quá trình nghiên cứu, nhóm đã ứng dụng nhận diện cà chua bằng thị giác máy tính thông qua thuật toán YOLOv7. Từ đó đánh giá, phân tích dựa trên dữ liệu cà chua thu thập được. Mô hình nhận diện quả cà chua đã đạt được độ chính xác 93,3 % đối với quả bình thường và đạt 89,1 % với quả hỏng trên tập dữ liệu thử nghiệm. Từ kết quả nghiên cứu có thể phân loại được cà chua một cách chính xác nhất nhằm tăng sản lượng, chất lượng sản phẩm trong lĩnh vực nông nghiệp Việt Nam.

Từ khóa: YOLOv7; Học sâu; Nhận diện quả cà chua; Mạng nơ-ron tích chập (CNN).

Abstract

YOLOv7: A deep learning approach for tomato classification

Artificial intelligence and machine learning are expanding and being used extensively in many facets of life these days. Large-scale data processing, data-driven learning, and automated decision-making are all possible with artificial intelligence. The tomato recognition model achieved an accuracy of 93.3 % for normal tomatoes and 89.1 % for damaged tomatoes on the test dataset. Throughout the investigation, the group used the YOLOv7 algorithm for computer-visual tomato recognition. This allowed for evaluation and analysis based on the data gathered on tomatoes. To boost output and improve product quality in Vietnam's agricultural sector, the research's findings can be used to classify tomatoes in the most precise manner.

Keywords: YOLOv7; Deep learning; Tomato classification; Convolutional Neural Networks (CNN).

***Tác giả liên hệ, Email:** cuongnt@haui.edu.vn
DOI: https://doi.org/10.63064/khtnmt.2024.569

1. Giới thiệu

Phát hiện đối tượng theo thời gian thực là một chủ đề rất quan trọng trong thị giác máy tính, vì nó thường là một thành phần cần thiết trong hệ thống thị giác máy tính. Ví dụ: Theo dõi đa đối tượng, lái xe tự động, robot, phân tích hình ảnh y tế,... Thuật toán phát hiện đối tượng được chia thành hai loại chính như phát hiện một giai đoạn (Single-shot object detection) và phát hiện hai giai đoạn (Two-shot object detection). YOLO là mô hình phát hiện, phân loại đối tượng tiên tiến và hiện đại được biết đến với độ chính xác cao và tốc độ nhanh. Việc ứng dụng YOLOv7 trong nhận dạng cà chua là một ý tưởng mới mẻ trong thời đại phát triển của trí tuệ nhân tạo (AI).

Có nhiều giải thuật ứng dụng trong nhận dạng như HOG (Histogram

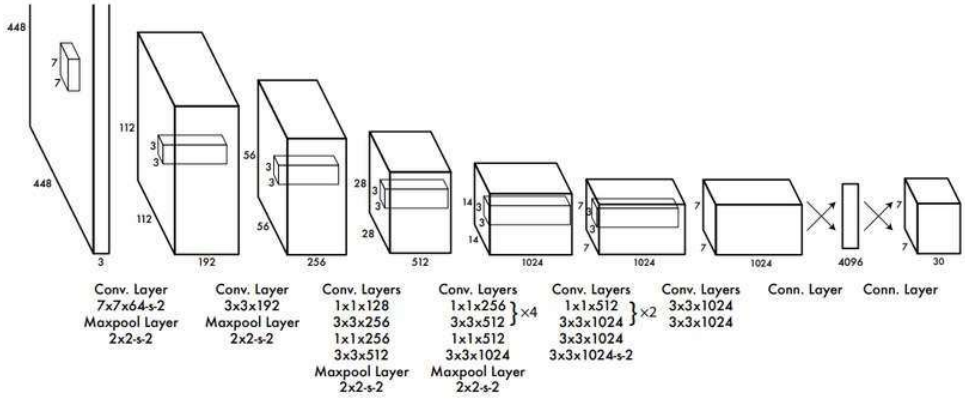
of Oriented Gradients), SIFT (Scale-Invariant Feature Transform), CNN (Convolutional Neural Networks), R-CNN (Region based Convolutional Neural Networks), SVM (Support Vector Machine). Sự ra đời của YOLO với phiên bản mới của mình là YOLOv7, có nhiều cải tiến so với các phiên bản ra đời trước đó. Một trong những cải tiến quan trọng trong YOLOv7 là việc áp dụng hàm loss mới gọi là "focal loss". Các phiên bản trước của YOLO sử dụng cross-entropy và loss function được biết là không chính xác trong các đối tượng nhỏ và quá nhỏ [1]. Để giải quyết vấn đề này "focal loss" đã giảm trọng số mất mát và tập trung vào các đối tượng khó phát hiện. YOLOv7 sử dụng cụ thể là 9 anchor box, cho phép YOLO phát hiện hình dạng và kích thước, phạm vi đối tượng rộng so với các phiên bản đi trước của mình, từ đó tăng tính chính xác và hiệu xuất của mô hình [2].

Nghiên cứu này nhằm tìm hiểu, tối ưu và cải tiến thuật toán YOLOv7 bằng cách sử dụng tập dữ liệu cà chua, cải tiến kiến trúc mạng nơ-ron và tối ưu hóa thuật toán huấn luyện để nâng cao khả năng nhận dạng và định vị cà chua trong ảnh. Nhóm tác giả tiến hành phân tích chi tiết các thành phần quan trọng của thuật toán, xây dựng và sử dụng một tập dữ liệu đa dạng và đại diện chứa ảnh cà chua từ nhiều nguồn khác nhau để đảm bảo tính đáng tin cậy và hiệu quả của thuật toán. Kết quả của nghiên cứu này sẽ đóng góp vào việc cải thiện quá trình nhận dạng cà chua, giúp tăng cường năng suất và chất lượng trong ngành nông nghiệp, đồng thời mang lại lợi ích thiết thực cho người nông dân và ngành công, nông nghiệp.

2. Cơ sở lý luận

You Only Look Once hay còn gọi tắt là YOLO. YOLO thuộc về Object Detection trong lĩnh vực Computer Vision và là mô hình có những ưu điểm khiến nó trở thành một phương pháp rất hiệu quả trong những bài toán nhận diện đối tượng (object detection). Thuật toán Object Detecttion được chia làm 2 nhóm chính: Các mô hình RCNN, mô hình về YOLO. Mô hình được thiết kế để nhận diện các vật thể real-time.

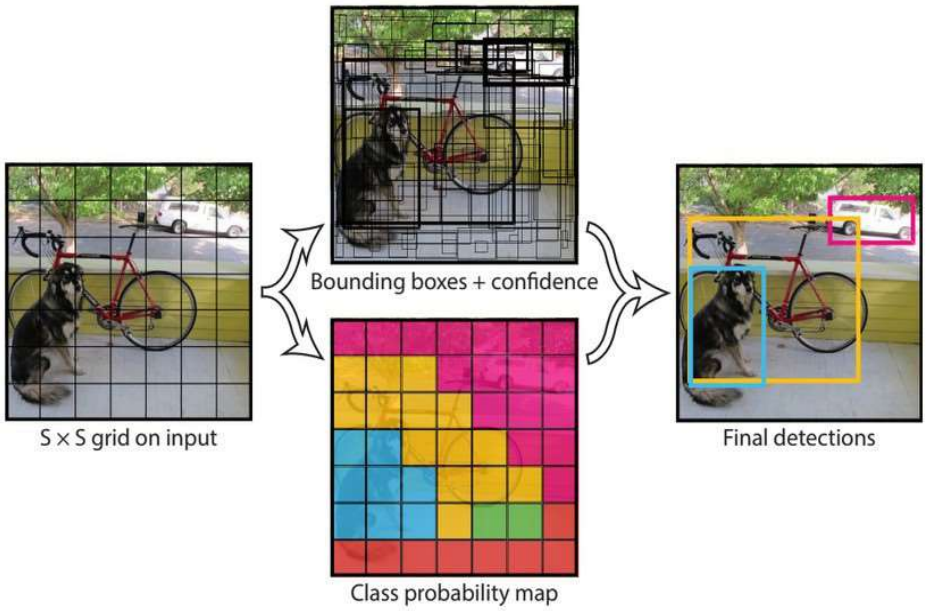
YOLO là mô hình mạng CNN sử dụng để phát hiện, phân loại đối tượng. YOLO là sự kết hợp giữa các convolutional layers và connected layers. Các convolutional layers sử dụng để lấy ra feature của ảnh, full-connected layers dự đoán xác suất và tọa độ của các đối tượng.



Hình 1: Mô hình YOLO

Cách mô hình YOLO hoạt động:

Mô hình đầu vào là ảnh và nhận dạng ảnh đó có những đối tượng nào, xác định tọa độ của các đối tượng đó. Ảnh đầu vào có thể là 3 × 3, 7 × 7 việc chia ô như này ảnh hưởng tới việc nhận dạng đối tượng của mô hình YOLO như sau:



Hình 2: Phương pháp nhận diện vật thể của YOLO

Input là 1 ảnh truyền vào, Output của mô hình là một ma trận 3 chiều với kích thước như sau: S × S × (5 × N + M) và tham số mỗi ô có số lượng (5 × N + M). Trong đó, N là số lượng Box và M Class ô cần phải dự đoán. Với ví dụ trên hình ảnh chia thành ô 7 × 7, mỗi ô cần mô hình dự đoán gồm 2 bounding box, 3 objects: Dog, car, bike. Vậy đầu ra là 7 × 7 × 13, có nghĩa là có 13 tham số cho mỗi ô, chúng ta có kết quả là: (7 × 7 × 2 = 98) bounding box.

Bounding box gồm 5 thành phần như sau: (x, y, w, h, prediction). Trong đó, x và y là tọa độ giá trị âm của bounding box, còn w và h là chiều rộng và chiều cao bounding box, prediction là Pr (Object).

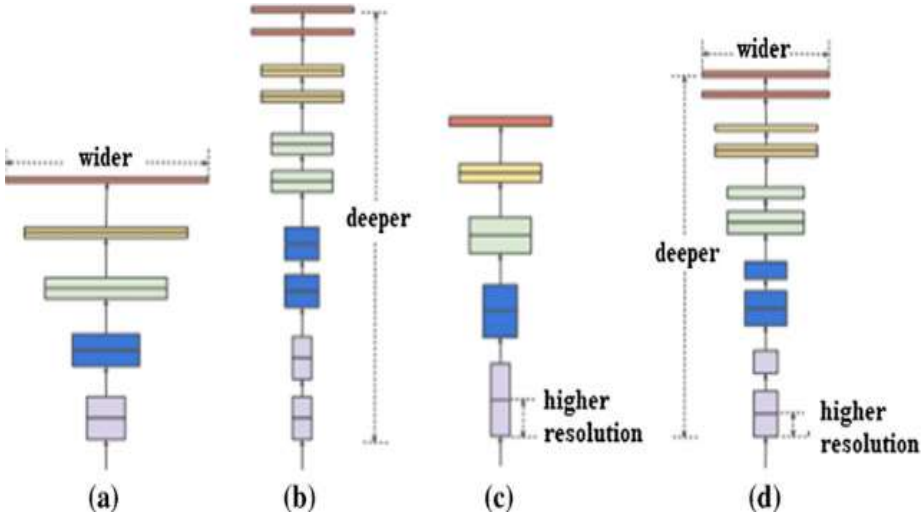
2.1. Những kiến thức cơ bản

Bag-of-freebies là lần đầu tiên xuất hiện và phát triển trong YOLOv4. BoF bao gồm các kỹ thuật Augmentations, hàm loss, label smoothing,... được thêm vào trong đào tạo có thể tăng độ chính xác mà không tăng thời gian xử lý.

Re-parameterization lần đầu được sử dụng ở phiên bản YOLOv5, nó thực hiện việc hợp nhất lớp Convolution (Conv) và lớp BatchNorm (BN) vào làm một lớp, khiến việc inference diễn ra nhanh hơn (từ 2 layers là Conv và BN sinh ra Conv). Quá trình hợp nhất diễn ra trong inference, còn training model thì vẫn hoạt động bình thường, có 2 lớp riêng biệt đó là Conv và BN.

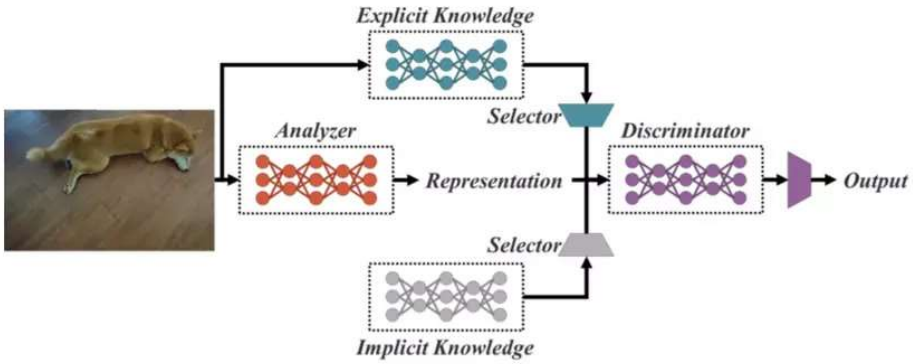
Khuếch đại độ model là một phương pháp quan trọng để tăng hiệu năng của model. Bằng cách sử dụng kỹ thuật tăng chiều tổng hợp ba chiều của mạng nơ-ron (chiều sâu, chiều rộng và chiều độ phân giải của ảnh đầu vào), model tăng chiều đã được phân tích lần đầu tiên trong EfficientNet.

Để khuếch đại độ lớn của model, đạt hiệu năng tốt hơn ta sử dụng Model scaling. Model scaling được phân tích kĩ lần đầu tiên trong EfficientNet sử dụng kỹ thuật scale cả 3 chiều của mạng nơ-ron là chiều rộng và chiều sâu, chiều độ phân giải của ảnh đầu vào.



Hình 3: Kĩ thuật compound scaling của EficientNet

Trong YOLO, Implicit knowledge (kiến thức tiềm ẩn) được giới thiệu và áp dụng. Ví dụ, con người có thể hiểu biết một sự vật hay hiện tượng bằng cách trải nghiệm trực tiếp và rút kinh nghiệm từ những trải nghiệm đó. YOLO muốn đưa kiến thức gián tiếp vào mạng nơ-ron. Trong mạng nơ-ron, kiến thức gián tiếp là kiến thức mà model học được thông qua sự tiếp xúc với các input, trong khi kiến thức gián tiếp là kiến thức mà model sẽ tự rút ra trong quá trình đào tạo, độc lập với các input. YOLO đã đưa ra 3 phương pháp để trình bày kiến thức implicit. Tuy nhiên, phương pháp đơn giản nhất, dưới dạng vector, được YOLO sử dụng và đạt được hiệu quả ổn định.



Hình 4: Implicit Knowledge trong YOLO

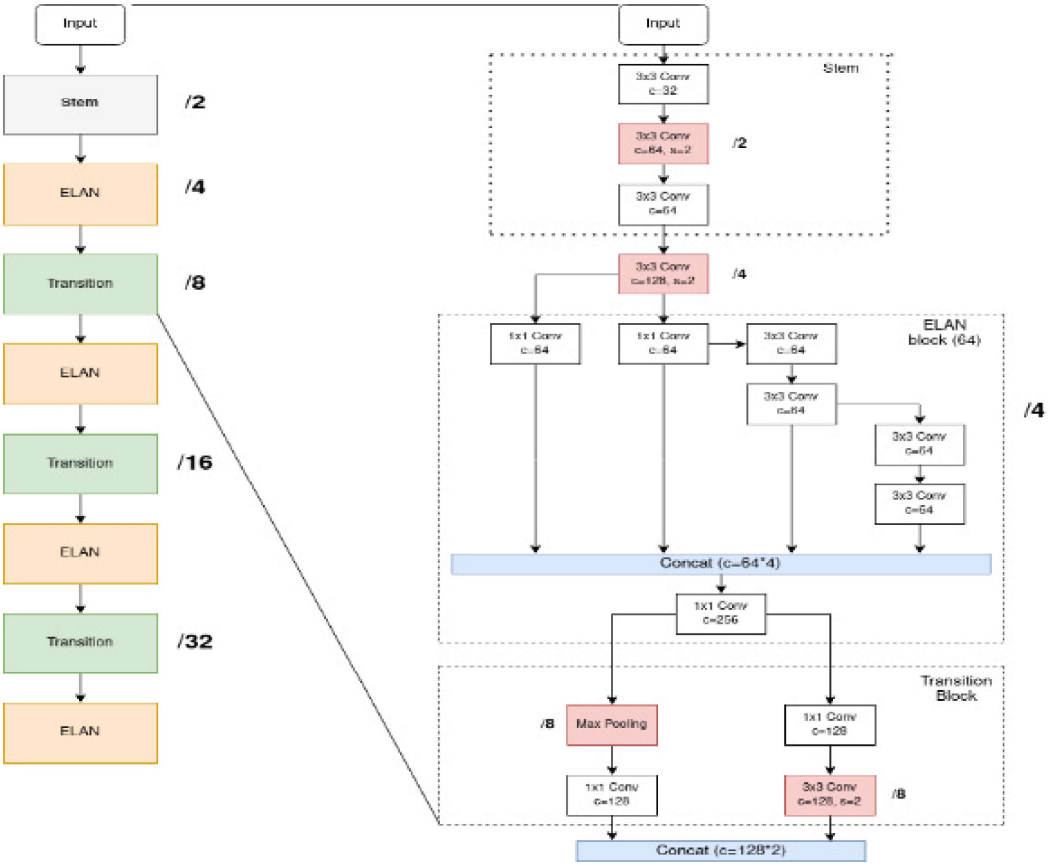
2.2. Kiến trúc mạng của YOLOv7

Cũng như các phiên bản tiền nhiệm của YOLO, YOLOv7 có kiến trúc 3 phần: Backbone, Neck và Head

a. Backbone

YOLOv7 sử dụng backbone là ELAN (Eficient Layer Aggregation Network). Một ELAN block gồm 3 phần chính:

Cross Stage Partial, Computation Block và phép PointWiseConv. Thiết kế của ELAN Block có sự kế thừa từ hai nghiên cứu trước đó là CSPNet và VoVNet [3, 4]. Ý tưởng về CSP hóa một block là việc tạo thêm một nhánh "cross stage partial" đã xuất hiện từ phiên bản YOLOv4. Các lớp Conv nằm trong block Computation Block tính toán và thông qua các 3 × 3 Conv sinh ra tính năng mới. Sau đó, tổng hợp các feature map lại ở cuối và sử dụng toán tử concatenate như VoVNet trên chiều channel, tiếp theo đưa qua PointWiseConv (1 × 1 Conv). Các ELAN Block được kết nối với nhau thông qua các Transition Block là một lần giảm 2 kích cỡ của feature maps. Input ảnh sẽ đi qua Stem Block trước khi tiến vào ELAN Block đầu tiên trong backbone. Vì vậy, một backbone hoàn chỉnh của YOLOv7 sẽ là tập hợp của các ELAN Block và các Transition block.



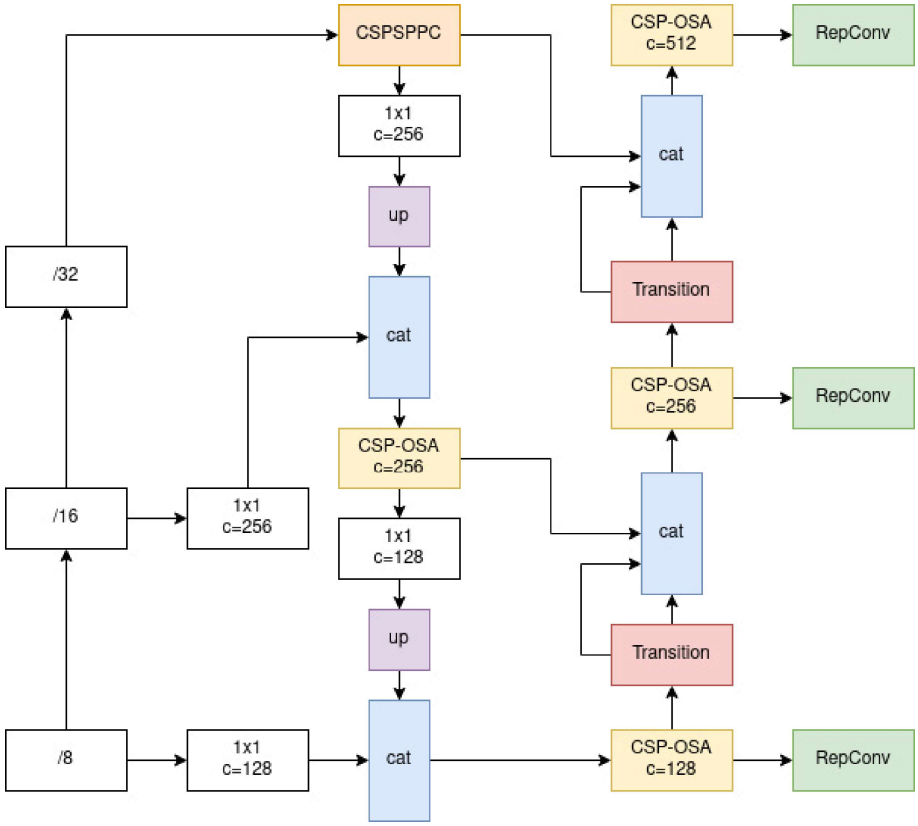
Hình 5: Backbone của YOLOv7

b. Neck

Spatial Pyramid Pooling (SPP) là một lớp trong mạng nơ-ron tích chập (CNN) giúp loại bỏ yêu cầu về ảnh đầu vào có kích thước cố định. SPP được thêm vào phía trên lớp chập cuối cùng, thực hiện tích chập các đặc trưng và tạo ra các đầu ra có độ dài cố định. SPP giải quyết vấn đề về kích thước đầu vào cố định mà không làm giảm hiệu suất tổng thể của mô hình. SPP duy trì thông tin không gian trong các vùng không gian cục bộ, với một số lượng và kích thước vùng cố định. Trong mỗi vùng không gian, các phản hồi của từng bộ lọc được pooling. SPP cho phép ảnh đầu vào có kích thước bất kỳ, điều này

cho phép tỷ lệ khung hình và tỷ lệ ảnh tùy ý. SPP được sử dụng trong việc nhận dạng đối tượng.

CSPSPP (Cross Stage Partial Spatial Pyramid Pooling) là phiên bản SPP được sử dụng trong YOLOv7. Việc sử dụng CSPSPP nhằm xử lý ảnh có kích thước khác nhau mà không cần thay đổi kích thước trước, điều này rất quan trọng cho các ứng dụng thực tế, nơi ảnh đầu vào có thể đến từ nhiều nguồn khác nhau. Bên cạnh đó, SPP còn có thể nắm bắt các đặc trưng ở nhiều tỷ lệ khác nhau đồng thời bảo toàn vị trí của chúng trong ảnh, việc này hỗ trợ rất lớn cho việc phát hiện đối tượng. So với SPP tiêu chuẩn, CSPSPP giảm số lượng phép tính cần thiết, từ đó giúp mô hình hoạt động hiệu quả hơn. Mặt khác việc sử dụng CSPSPP làm tăng độ phức tạp của mạng so với các phương pháp pooling đơn giản hơn.

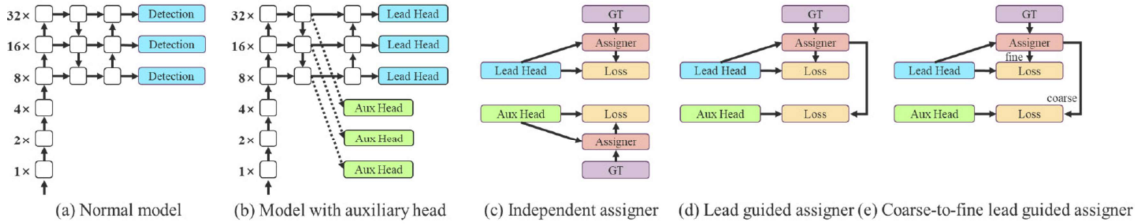


Hình 6: Neck của YOLOv7

c. Head

Ở phiên bản này, head của YOLOv7 không có nâng cấp gì mới, vẫn giữ nguyên việc sử dụng YOLO và auxiliary head như các phiên bản tiền nhiệm của dòng họ YOLO. Cấu trúc của head trong YOLOv7 gồm 3 lớp mạng nơ-ron hoàn toàn kết nối (fully-connected layers). Lớp thứ nhất đó là dự đoán các ô lưới (grid cells) chứa đối tượng. YOLOv7 chia ảnh đầu vào thành một lưới các ô vuông và lớp này dự đoán xem có đối tượng nào nằm trong mỗi ô lưới hay không. Lớp thứ hai là dự đoán các bounding box cho các đối tượng. Nếu lớp thứ nhất dự đoán một ô lưới chứa đối tượng, lớp thứ hai sẽ dự đoán kích thước và tỷ lệ khung hình (bounding box) của đối tượng đó. Lớp thứ ba là dự

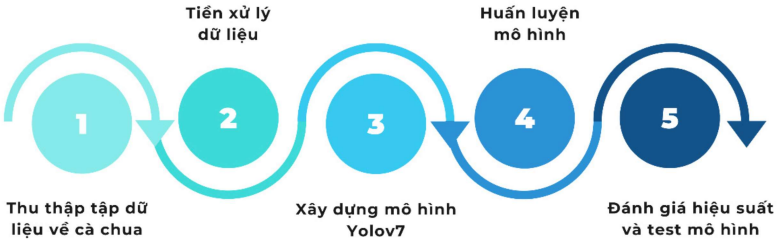
đoán lớp (class) cho mỗi đối tượng được phát hiện. Lớp này sẽ dự đoán đối tượng thuộc lớp nào trong số các lớp đã được xác định trước. Điểm khác biệt so với các phiên bản YOLO trước là YOLOv7 có thể sử dụng một hoặc nhiều "head" tùy thuộc vào phiên bản cụ thể. Một số cải tiến trong Head của YOLOv7 đó là GIOU (Generalized Intersection over Union) để huấn luyện "head", giúp cải thiện khả năng dự đoán bounding box chính xác hơn. Thêm vào đó, năng lực học tập nhiều kích thước (Multi-scale training), trong quá trình huấn luyện, YOLOv7 sử dụng ảnh đầu vào với nhiều kích thước khác nhau, giúp "head" học cách phát hiện đối tượng ở các kích thước đa dạng. Trong kiến trúc của YOLOv7, "head" chịu trách nhiệm cho đầu ra cuối được gọi là "lead head" và "head" được sử dụng để hỗ trợ việc huấn luyện được gọi là "auxiliary head". YOLOv7 sử dụng "lead head" làm hướng dẫn để tạo ra các nhãn phân cấp từ thô đến mịn. Các nhãn này lần lượt được sử dụng để huấn luyện "auxiliary head" và "lead head" [7].



Hình 7: Gán nhãn thô cho auxiliary head và gán nhãn mịn cho lead head

3. Kết quả nghiên cứu

Để xây dựng chương trình phân loại cà chua dựa trên thuật toán YOLOv7 cần phải trải qua các bước trong quy trình sau:



Hình 8: Quy trình thực hiện bài toán

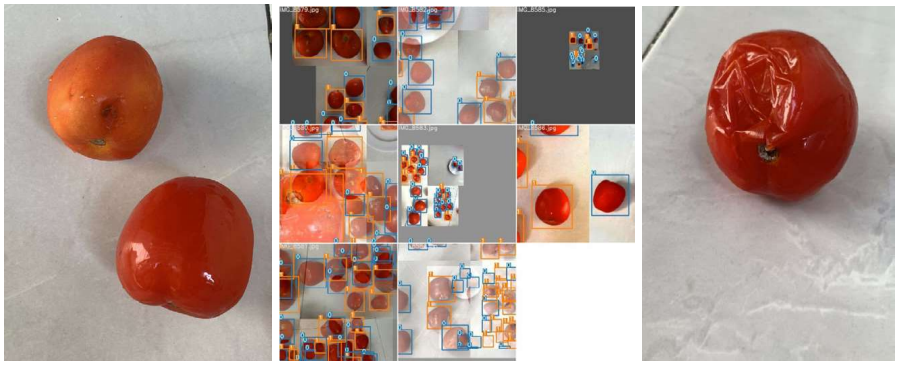
3.1. Thu thập dữ liệu

Tổng cộng 1.029 ảnh trong đó có 887 ảnh được huấn luyện (training) và 142 ảnh được kiểm tra (test), đã được thu thập từ bộ dữ liệu của Roboflow và được chụp bổ sung thêm bằng điện thoại (Redmi Note 8), mỗi ảnh có kích thước là 1.024 × 631 pixels, một bit màu sâu 24 và độ phân giải 96 dpi. Để duy trì tính nhất quán và độ tin cậy, một giao thức camera tỉ mỉ đã được tuân thủ. Hình ảnh thu được từ khoảng cách tiêu chuẩn khoảng 30 cm, đảm bảo độ biến dạng tối thiểu và duy trì tỷ lệ vật thể nhất quán. Máy ảnh được đặt vuông góc với mặt phẳng của mặt đất để giảm thiểu mọi sai lệch về góc độ. Để tăng độ mạnh mẽ của mô hình đào tạo, hình ảnh được thu thập từ các góc độ

khác nhau về điều kiện chiếu sáng và mức độ tương phản khác nhau giữa mục tiêu (quả cà chua) và nền (sàn gạch). Các hình ảnh được chuẩn hóa thành kích thước 640 × 640 pixel để phù hợp với mô hình YOLOv7 và tăng tính nhất quán của các mẫu huấn luyện. Trong ứng dụng thực tế, một camera chuyên nghiệp có thể được sử dụng để chụp ảnh thay vì điện thoại thủ công. Do đó, khoảng cách và góc quay của hình ảnh có thể bị thay đổi. Các mô hình dựa trên YOLO của CNN, chẳng hạn như nhiều thuật toán phát hiện đối tượng khác, rất nhạy cảm với những thay đổi về khoảng cách và góc quay của hình ảnh. Các mô hình YOLO được thiết kế để phát hiện các đối tượng trong hình ảnh bằng cách chia hình ảnh thành một lưới và dự đoán các hộp giới hạn cũng như xác suất lớp cho mỗi ô lưới. Thiết kế này cho phép YOLO xác định các đối tượng ở các vị trí khác nhau trong một hình ảnh.

3.2. Tiền xử lý dữ liệu

Ảnh sau khi được thu thập sẽ được tiến hành gán nhãn (Data labeling and Data annotations). Ở đây, nhóm tác giả sử dụng công cụ Labeling để thực hiện việc gán nhãn. Kết quả thu được sau khi xử lý thu được YOLO annotations là <object-class> <x> <y> <width> <height>. Trong đó object-class gồm 2 giá trị là "0" và "1", tương ứng với tình trạng hỏng và không hỏng của quả cà chua. "x" và "y" lần lượt là tọa độ trục hoành và trục tung cho center box của bounding box. "Width" và "Height" là chiều dài và rộng của bouding box. Tất cả các chỉ số đều được chuẩn hóa về khoảng giá trị [0,1].



Hình 9: Hình ảnh quả hỏng và không hỏng trước và sau khi gán nhãn

3.3. Huấn luyện model

Để thực hiện việc huấn luyện model, ta cần cài đặt một số phần mềm sau: CUDA 11.3, cuDNN 8.2.1, Python 3.7.3 trở lên. Sau đó chúng ta tiến hành cài đặt những gói (package) cần thiết cho quá trình huấn luyện từ tập tin "requirements. txt". Sử dụng file pretrain có sẵn trong tài liệu của YOLOv7 và nhân bản (clone) để huấn luyện model theo các bước. Quá trình thực hiện sẽ được thực hiện trên Google Colab (Colaboratory), một công cụ miễn phí của Google cho phép mượn cấu hình (CPU và GPU) để thực hiện việc thực thi, huấn luyện dữ liệu bằng ngôn ngữ Python.

```
[ ] %cd /content/drive/MyDrive/Train_Tomato_yolov7/yolov7
    !python train.py --batch-size 8 --cfg cfg/training/yolov7.yaml --epochs 300  --data data/mydataset.yaml --weights 'pretrain/yolov7.pt'

    %cd /content/drive/MyDrive/Train_Tomato_yolov7/yolov7
    !python train.py --batch 8 --cfg cfg/training/yolov7.yaml --epochs 300  --data data/mydataset.yaml --weights 'pretrain/yolov7.pt' --device 0 --name yolov7_tomato --hyp data/hyp.scratch.p6.yaml --resume

     297/299     8.45G   0.01239    0.0104 0.0006509   0.02344        85       640: 100% 111/111 [01:19<00:00,  1.39it/s]
                 Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 5/5 [00:01<00:00,  2.94it/s]
                   all          71         315       0.973       0.984       0.982       0.783

       Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     298/299     8.45G   0.01238   0.01026 0.0006771   0.02331        55       640: 100% 111/111 [01:18<00:00,  1.42it/s]
                 Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 5/5 [00:03<00:00,  1.63it/s]
                   all          71         315       0.982        0.98       0.981       0.784

       Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     299/299     8.45G    0.0123   0.01026 0.0006632   0.02322       145       640: 100% 111/111 [01:19<00:00,  1.39it/s]
                 Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 5/5 [00:03<00:00,  1.51it/s]
                   all          71         315       0.982       0.981       0.982       0.783
                   Yes          71         203        0.99       0.985        0.99       0.797
                    No          71         112       0.973       0.977       0.974       0.768
130 epochs completed in 3.018 hours.
```

Hình 10: Quá trình huấn luyện

3.4. Triển khai mô hình

Sau khi đã hoàn thiện việc huấn luyện và thử nghiệm các mô hình của các bài toán con, nhóm tác giả sẽ thực hiện chạy hoàn chỉnh trên hệ điều hành Windows. Cấu hình phần cứng cơ bản là một bộ CPU (4 nhân trở lên), GPU có khả năng xử lý tính toán tốt nhiều VRAM (từ 2GB trở lên) và một camera HD. Ảnh sẽ được đưa từ thư mục để nhận dạng hoặc có thể sử dụng camera để nhận dạng trực tiếp. Kết quả được hiển thị dưới dạng thông số trong Bouding box như phần 3.2 ở trên.

3.5. Đánh giá hiệu suất và test mô hình

Để thực hiện việc test mô hình, ta sử dụng câu lệnh "detect.py" kết hợp cùng model đã "train" và đầu vào là một ảnh sau:

Hình 11: Hình minh họa cà chua trước khi test

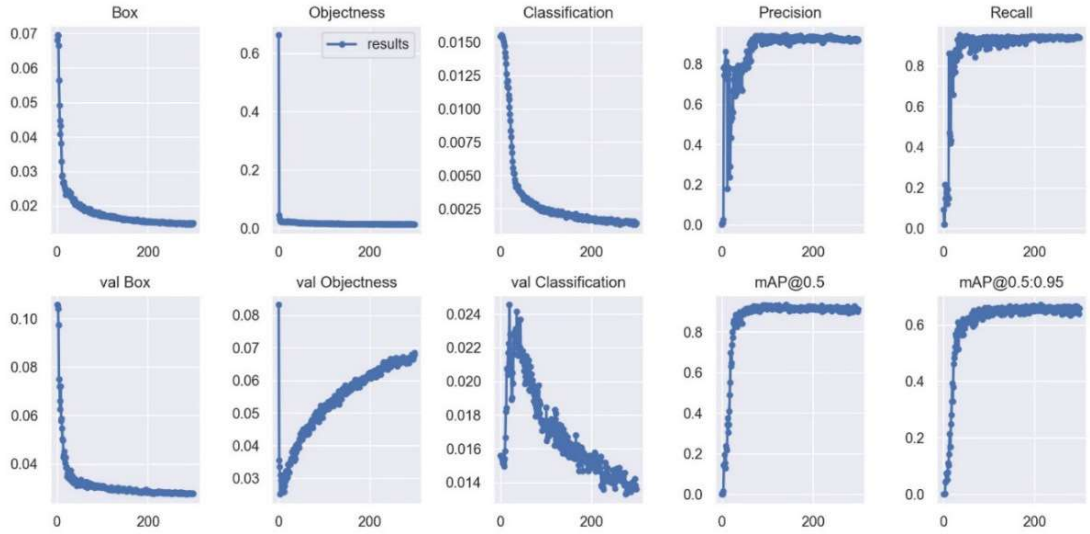
Hình 12: Hình minh họa kết quả

Số lượng quả cà chua thực tế trong Hình 12 là 14 quả cà chua bao gồm 7 quả tốt và 7 quả hỏng. Việc áp dụng model sau khi "train" để nhận diện và phân loại ta thu được: 6 quả tốt (OK), 8 quả không tốt (NG). Vậy hiệu suất model đạt độ chính xác khá tốt trong trường hợp trên.

Hình 13 gồm các chỉ số sau:

Precision Curve (P_curve.png): Biểu diễn các giá trị chính xác ở các ngưỡng khác nhau. Biểu đồ thể hiện độ chính xác thay đổi khi ngưỡng thay đổi. Recall Curve (R_curve.png): Tương ứng, biểu đồ này biểu diễn cách thay đổi qua các ngưỡng khác nhau của các giá trị recall [5]. Box/val Box: Mất hộp giới hạn của tập dữ liệu huấn luyện hoặc tập dữ liệu xác thực, hộp càng nhỏ thì càng chính xác. Objectness/val Objectness: Là tổn thất trung bình của việc phát hiện mục

tiêu và việc phát hiện mục tiêu càng nhỏ thì việc phát hiện càng chính xác. Classification/val classification: Đào tạo hoặc xác nhận được suy đoán là giá trị trung bình của mất phân loại và phân loại càng nhỏ thì càng chính xác. mAP50: Độ chính xác trung bình được xác định với giá trị 0,50 ở ngưỡng giao nhau trên giao nhau (IoU). Đó là thước đo độ chính xác của mô hình chỉ tính đến các kết quả "dễ dàng". mAP50-95: Trung bình của độ chính được tính ở các ngưỡng IoU khác nhau, dao động từ 0,50 đến 0,95 [6]. Nó cung cấp cái nhìn tổng thể về hiệu suất mô hình qua các mức độ và trường hợp khó phát hiện khác nhau.



Hình 13: Hình các loại đường cong hiệu suất

4. Thảo luận

Việc huấn luyện mô hình trong vài lần đầu có độ chính xác chưa cao. Tuy nhiên, sau khi thực hiện "training" nhiều lần bằng cách thêm dữ liệu và lặp lại việc huấn luyện lại các bức ảnh mà model chưa học kỹ, mô hình nhận diện quả cà chua đã đạt được độ chính xác 93,3 % đối với quả bình thường và đạt 89,1 % với quả hỏng trên tập dữ liệu thử nghiệm. Mô hình thử nghiệm này vẫn cần cải thiện. Độ chính xác có thể tăng thêm nếu có thêm dữ liệu hơn.

5. Kết luận

Trong nghiên cứu này, nhóm tác giả đã phân tích, huấn luyện model dựa trên thuật toán YOLOv7 trong việc nhận diện cà chua hỏng và cà chua không hỏng. Kết quả cho thấy, model sau khi train và test đạt được hiệu suất cao và đáng tin cậy trong việc phân loại cà chua.

Đề tài đã vượt qua những hạn chế của các phương pháp truyền thống trong việc nhận diện và phân loại cà chua. Nó cho phép chúng ta xử lý nhanh chóng các hình ảnh có kích thước lớn và đồng thời đảm bảo độ chính xác cao. Kết quả của nghiên cứu này có thể được áp dụng vào các quy trình sản xuất và kiểm tra chất lượng cà chua trong ngành công nghiệp thực phẩm. Các nhà sản xuất và nhà nghiên cứu có thể sử dụng để tự động hóa quá trình phân loại cà chua theo tình trạng hỏng và tốt, từ đó giảm thiểu sự lãng phí và tăng cường hiệu suất sản xuất. Tuy nhiên, để nâng cao hiệu quả và ứng dụng thực tế, cần tiếp tục

nghiên cứu và phát triển. Các nỗ lực có thể tập trung vào việc tăng cường tốc độ xử lý và độ chính xác của model, cũng như mở rộng phạm vi ứng dụng đến các loại trái cây và thực phẩm khác.

TÀI LIỆU THAM KHẢO

[1]. Redmon, J., & Farhadi, A., (2018). *YOLOv3: An incremental improvement*. arXiv preprint arXiv:1804.02767.

[2]. Wang, C. Y., Bochkovskiy, A., Hong-Yuan, Y., (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.* arXiv preprint arXiv:2207.02485.

[3]. Wang, C. Y., Bochkovskiy, A., (2020). *CSPNet: A new Backbone that can enhance the performance of visual recognition models*. arXiv preprint arXiv:1911.11929.

[4]. Wu, H., Zhang, Y., Xu, Y., Bao, J., Guo, Z., (2021). *VoVNet: Vision transformer with variable input resolution for object detection.* arXiv preprint arXiv:2106.14833.

[5]. Hastie, T., Tibshirani, R., Friedman, J., (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media.

[6]. Brachmann, M., Rother, C., (2014). *Measuring the object detection accuracy using intersection over union*. Proceedings of the IEEE conference on computer vision and pattern recognition (p. 2.980-2.987).

[7]. Nguyen Mai (2023). *Series [Paper Explain] YOLOv7: Su dung cac "trainable bag-off-freebies" dua YOLO len mot tam cao moi*. Kênh Youtube Nguyen Mai.

BBT nhận bài: 18/3/2024; Phản biện xong: 25/3/2024; Chấp nhận đăng: 28/3/2024