ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC
THỊ GIÁC MÁY TÍNH NÂNG CAO

ĐỀ TÀI:
NHẬN DIỆN NGƯỜI ĐEO KHẨU TRANG VỚI YOLOv8

GIẢNG VIÊN HƯỚNG DẪN: TS Mai Tiến Dũng

SINH VIÊN THỰC HIỆN: Bùi Văn Thuận - 20521990

MÃ LỚP: CS331.N21.KHCL

TP. HỒ CHÍ MINH, 21 THÁNG 6 NĂM 2023

LỜI CẢM ƠN

Sau quá trình học tập và rèn luyện tại Trường Đại học Công Nghệ Thông Tin, chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Mai Tiến Dũng đã tận tâm hướng dẫn, truyền đạt những kiến thức cũng như kinh nghiệm cho chúng em trong suốt thời gian học tập môn Thị giác máy tính nâng cao.

Trong quá trình làm đồ án môn học, chúng em chắc chắn sẽ không tránh được những sai sót không đáng có. Mong nhận được sự góp ý cũng như kinh nghiệm quý báu của các các thầy để được hoàn thiện hơn và rút kinh nghiệm cho những môn học sau. Chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, tháng 6 năm 2023.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU

Trong thời điểm dịch Covid-19 diễn biến phức tạp ở nước ta và trên toàn thế giới, giãn cách xã hội và tránh tụ tập đông người là điều cần phải thực hiện trong mùa dịch này.

Tuy nhiên, nhu cầu đi lại là điều không thể tránh khỏi. Để giúp việc kiểm soát được tốt hơn tại các siêu thị, cửa hàng, nhà hàng và những nơi công cộng nên bài toán **"Nhận diện người đeo khẩu trang"** đã ra đời vì muốn việc theo dõi, giám sát các đối tượng không tuân thủ đúng quy tắc phòng ngừa dịch bệnh, giảm được phần nào lo ngại mà không làm lây nhiễm dịch bệnh trong cộng đồng. Điều này cũng sẽ giúp các cơ quan tiết kiệm được thời gian cũng như giảm được khoảng cách tiếp xúc trong việc kiểm soát dịch bệnh.

Điều này cũng sẽ giúp các cơ quan tiết kiệm được thời gian cũng như giảm được khoảng cách tiếp xúc trong việc kiểm soát dịch bệnh.

CHƯƠNG 2: ĐỊNH NGHĨA BÀI TOÁN

2.1. Mục tiêu bài toán

Xây dựng mô hình phát hiện người đeo khẩu trang đúng, đeo khẩu trang sai và không đeo khẩu trang tại các khu vực công cộng (nhà trường, siêu thị, xe bus,…)

2.2. Một số khái niệm

Khẩu trang là một loại vật dụng bảo vệ được sử dụng để bịt vùng mặt (mũi, miệng) để ngăn ngừa bảo vệ người đeo khỏi bị lây nhiễm các loại vi khuẩn, dịch bệnh, bụi bặm thông qua đường hô hấp.

Đeo khẩu trang đúng là đảm bảo rằng khẩu trang che phủ hoàn toàn từ mũi đến cằm và không để lộ phần mũi hoặc miệng.

Đeo khẩu trang sai là đeo khẩu trang nhưng để lộ mũi hoặc miệng

Không đeo khẩu trang là để lộ hoàn toàn mũi và miệng.

2.3. Xác định yêu cầu bài toán

Input: Ảnh có một hay nhiều khuôn mặt người

- Đảm bảo đuôi của hình ảnh được input là dạng của hình ảnh (.png, .jpg ,jpeg….).
- Hình ảnh phải sắc nét, rõ ràng không bị mờ.

Output: Ảnh của input chứa Bounding box tại vị trí mà mô hình cho rằng có chứa khuôn mặt người, nhãn và score cho từng Bounding box đó.

Bài toán thuộc dạng bài toán Object Detection, có tổng cộng 3 class:

- With_mask: đeo khẩu trang đúng
- Mask_weared_incorrect: đeo khẩu trang sai cách

– Without_mask: không đeo khẩu trang

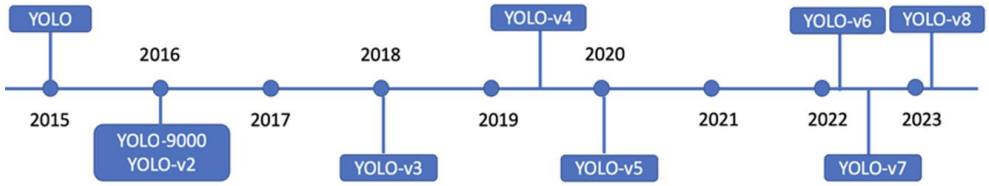
CHƯƠNG 3: PHƯƠNG PHÁP

3.1. Tổng quan về YOLO

Object Detection là một bài toán quan trọng trong lĩnh vực Computer Vision, 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 real-time

YOLO (You Only Look Once) là mô hình phát hiện đối tượng phổ biến được biết đến với tốc độ nhanh và độ chính xác cao. Mô hình này lần đầu tiên được giới thiệu bởi Joseph Redmon và cộng sự vào năm 2016. Kể từ đó đến nay, đã có nhiều phiên bản của YOLO, một trong những phiên bản gần đây nhất là YOLO v8.



Hình 1. YOLO timeline

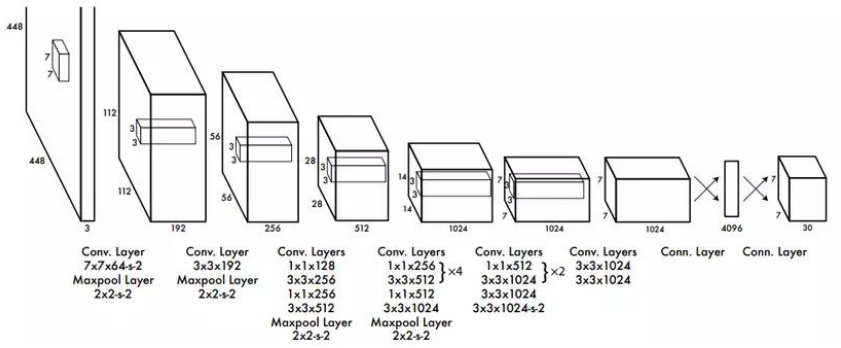
YOLO có thể không phải là thuật toán tốt nhất nhưng nó là thuật toán nhanh nhất trong các lớp mô hình object detection. Nó có thể đạt được tốc độ gần như real time mà độ chính xác không quá giảm so với các model thuộc top đầu

YOLO là thuật toán object detection nên mục tiêu của mô hình không chỉ là dự báo nhãn cho vật thể như các bài toán classification mà nó còn xác định vị trí của vật thể. Do đó YOLO có thể phát hiện được nhiều vật thể có nhãn khác nhau trong một bức ảnh thay vì chỉ phân loại duy nhất một nhãn cho một bức ảnh.

Một trong những ưu điểm mà YOLO đem lại đó là chỉ sử dựng thông tin toàn bộ bức ảnh một lần và dự đoán toàn bộ object box chứa các đối tượng, mô hình được xây dựng theo kiểu end-to-end nên được huấn luyện hoàn toàn bằng gradient descent.

3.2. Kiến trúc mạng YOLO

Yolo là một mô hình mạng CNN cho việc phát hiện, nhận dạng, phân loại đối tượng. Yolo được tạo ra từ việc kết hợp giữa các convolutional layers và connected layers.Trong đóp các convolutional layers sẽ trích xuất ra các feature của ảnh, còn full-connected layers sẽ dự đoán ra xác suất đó và tọa độ của đối tượng.



Hình 2. Kiến trúc mô hình YOLOv1

3.3. Nguyên lý hoạt động của YOLO

20 lớp tích chập đầu tiên của mô hình được đào tạo trước với ImageNet bằng cách cắm vào một lớp tổng hợp trung bình tạm thời (temporary average pooling) và lớp được kết nối đầy đủ (fully connected layer). Sau đó, mô hình đào tạo trước này được chuyển đổi để thực hiện phát hiện. Lớp được kết nối đầy đủ cuối cùng của YOLO dự đoán cả xác suất của lớp và tọa độ hộp giới hạn.

YOLO chia hình ảnh đầu vào thành lưới S × S. Nếu tâm của một đối tượng rơi vào một ô lưới thì ô lưới đó có nhiệm vụ phát hiện đối tượng đó. Mỗi ô lưới dự đoán các hộp giới hạn B và điểm tin cậy cho các hộp đó. Các điểm tin cậy này phản ánh mức độ tin cậy của mô hình rằng hộp chứa một đối tượng và mức độ chính xác mà mô hình cho rằng hộp được dự đoán.

YOLO dự đoán nhiều hộp giới hạn trên mỗi ô lưới. Tại thời điểm đào tạo, ta chỉ muốn một bộ dự đoán hộp giới hạn thể hiện cho từng đối tượng. YOLO chỉ định bộ dự đoán dựa trên chỉ số IOU hiện tại cao nhất với thực tế. Điều này dẫn đến sự chuyên môn hóa giữa các bộ dự đoán hộp giới hạn. Mỗi công cụ dự đoán trở nên tốt hơn trong việc dự báo các kích thước, tỷ lệ khung hình hoặc loại đối tượng nhất định, cải thiện tổng thể recall score.

Một kỹ thuật quan trọng được sử dụng trong các mô hình YOLO là NMS (non-maximum suppression). NMS là một bước hậu xử lý được sử dụng để cải thiện độ chính xác và hiệu quả của việc phát hiện đối tượng. Trong phát hiện đối tượng, thông thường có nhiều hộp giới hạn được tạo cho một đối tượng trong một hình ảnh. Các hộp giới hạn này có thể chồng lên nhau hoặc nằm ở các vị trí khác nhau, nhưng tất cả chúng đều đại diện cho cùng một đối tượng. NMS được sử dụng để xác định và loại bỏ các hộp giới hạn dư thừa hoặc không chính xác và đề xuất một hộp giới hạn duy nhất cho từng đối tượng trong ảnh.

3.4. Điểm cải tiến của các phiên bản YOLO

- **YOLOv2**

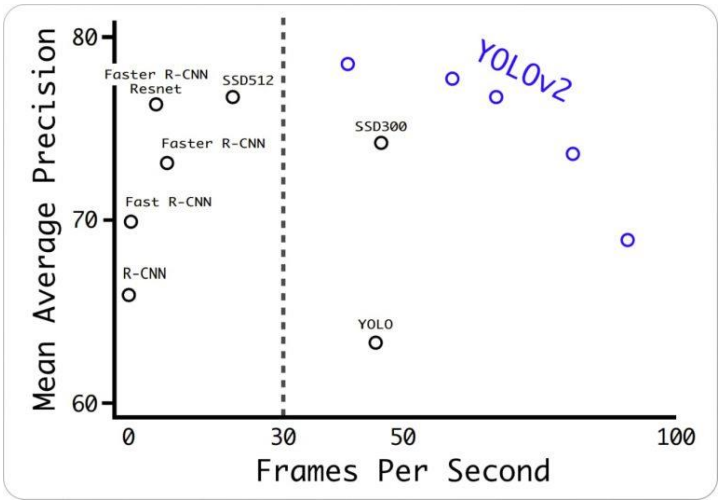
YOLO v2, còn được gọi là YOLO9000, được giới thiệu vào năm 2016 như một cải tiến so với thuật toán YOLO ban đầu. Nó được thiết kế để nhanh hơn và chính xác hơn YOLO và có thể phát hiện nhiều loại đối tượng hơn. Phiên bản cập nhật này cũng

sử dụng một xương sống CNN khác có tên là Darknet-19, một biến thể của kiến trúc VGGNet với các lớp progressive convolution và pooling layers đơn giản.

Một trong những cải tiến chính trong YOLO v2 là việc sử dụng các anchor boxes. Các anchor boxes là một tập hợp các hộp giới hạn được xác định trước với các tỷ lệ và tỷ lệ khung hình khác nhau. Khi dự đoán các hộp giới hạn, YOLO v2 sử dụng kết hợp các anchor boxes và độ lệch được dự đoán để xác định hộp giới hạn cuối cùng. Điều này cho phép thuật toán xử lý phạm vi kích thước và tỷ lệ khung hình rộng hơn của đối tượng.

Một cải tiến khác trong YOLO v2 là sử dụng chuẩn hóa hàng loạt, giúp cải thiện độ chính xác và ổn định của mô hình. YOLO v2 cũng sử dụng chiến lược đào tạo đa quy mô, bao gồm đào tạo mô hình trên hình ảnh ở nhiều tỷ lệ và sau đó lấy trung bình các dự đoán. Điều này giúp cải thiện hiệu suất phát hiện các đối tượng nhỏ.

YOLO v2 cũng giới thiệu một loss function phù hợp hơn với các tác vụ phát hiện đối tượng. Loss function dựa trên tổng các lỗi bình phương giữa các hộp giới hạn sự thật được dự đoán và xác suất của lớp.



Hình 3. Kết quả thu được từ YOLO v2 so với phiên bản gốc và các mô hình hiện đại khác

❖ YOLO v3

YOLO v3 là phiên bản thứ ba của thuật toán phát hiện đối tượng YOLO. Nó được giới thiệu vào năm 2018 như một cải tiến so với YOLO v2, nhằm tăng độ chính xác và tốc độ của thuật toán.

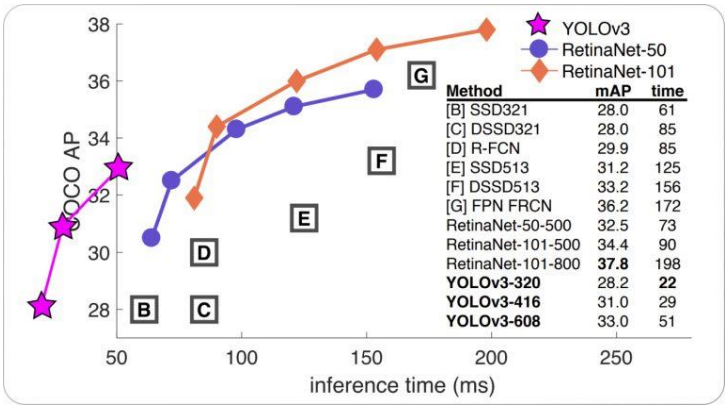
Một trong những cải tiến chính trong YOLO v3 là việc sử dụng kiến trúc CNN mới có tên là Darknet-53. Darknet-53 là một biến thể của kiến trúc ResNet và được thiết kế dành riêng cho các nhiệm vụ phát hiện đối tượng. Nó có 53 lớp tích chập và có thể đạt được kết quả tiên tiến trên nhiều tiêu chuẩn phát hiện đối tượng khác nhau.

Một cải tiến khác trong YOLO v3 là các anchor box với các tỷ lệ và tỷ lệ khung hình khác nhau. Trong YOLO v2, các anchor box đều có cùng kích thước, điều này đã hạn

chế khả năng phát hiện các đối tượng có kích thước và hình dạng khác nhau của thuật toán. Trong YOLO v3, các anchor box được chia tỷ lệ và tỷ lệ khung hình thay đổi để phù hợp hơn với kích thước và hình dạng của các đối tượng được phát hiện.

YOLO v3 cũng giới thiệu khái niệm "feature pyramid networks" (FPN). FPN là một kiến trúc CNN được sử dụng để phát hiện các đối tượng ở nhiều tỷ lệ. Nó xây dựng một kim tự tháp gồm các bản đồ đặc trưng, với mỗi cấp độ của kim tự tháp được sử dụng để phát hiện các đối tượng ở một tỷ lệ khác nhau. Điều này giúp cải thiện hiệu suất phát hiện trên các đối tượng nhỏ, vì mô hình có thể nhìn thấy các đối tượng ở nhiều tỷ lệ.

Ngoài những cải tiến này, YOLO v3 có thể xử lý nhiều kích thước đối tượng và tỷ lệ khung hình hơn. Nó cũng chính xác và ổn định hơn so với các phiên bản trước của YOLO.



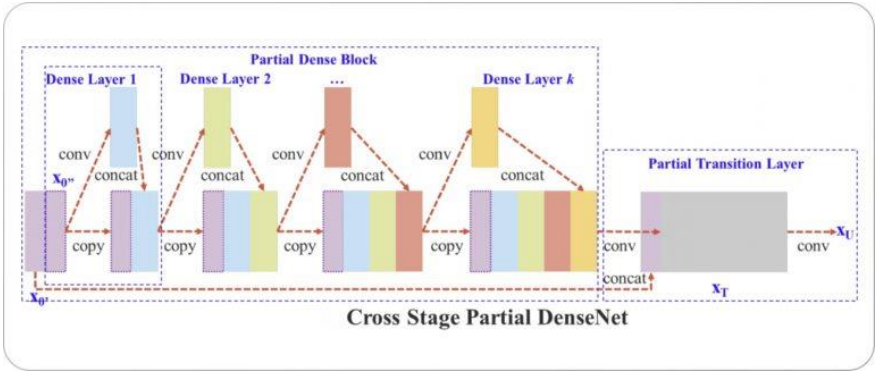
Hình 4. Kết quả mô hình YOLOv3

❖ YOLO v4

Lưu ý: Joseph Redmond, tác giả ban đầu của YOLO, đã rời khỏi cộng đồng AI vài năm trước, vì vậy YOLOv4 và các phiên bản khác trước đây không phải là tác phẩm chính thức của ông. Một số trong số chúng được duy trì bởi các đồng tác giả, nhưng không có bản phát hành nào trước đây của YOLOv3 được coi là YOLO "chính thức".

YOLO v4 là phiên bản thứ tư của thuật toán phát hiện đối tượng YOLO được giới thiệu vào năm 2020 bởi Bochkovskiy và cộng sự như một cải tiến so với YOLO v3.

Cải tiến chính trong YOLO v4 so với YOLO v3 là việc sử dụng kiến trúc CNN mới có tên là CSPNet (hiển thị bên dưới). CSPNet là viết tắt của "Cross Stage Partial Network" và là một biến thể của kiến trúc ResNet được thiết kế đặc biệt cho các nhiệm vụ phát hiện đối tượng. Nó có cấu trúc tương đối nông, chỉ có 54 lớp chập. Tuy nhiên, nó có thể đạt được kết quả tiên tiến trên các tiêu chuẩn phát hiện đối tượng khác nhau.



Hình 5. Kiến trúc của CSPNet

Cả YOLO v3 và YOLO v4 đều sử dụng các anchor box có tỷ lệ và tỷ lệ khung hình khác nhau để phù hợp hơn với kích thước và hình dạng của các đối tượng được phát hiện. YOLO v4 giới thiệu một phương pháp mới để tạo các anchor box, được gọi là "k-means clustering". Nó liên quan đến việc sử dụng thuật toán phân cụm để nhóm các hộp giới hạn thực tế thành các cụm và sau đó sử dụng trọng tâm của các cụm làm anchor box. Điều này cho phép các anchor box được căn chỉnh chặt chẽ hơn với kích thước và hình dạng của các đối tượng.

Mặc dù cả YOLO v3 và YOLO v4 đều sử dụng loss function tương tự để đào tạo mô hình, nhưng YOLO v4 giới thiệu một thuật ngữ mới gọi là "GHM loss". Đây là một biến thể của focal loss function và được thiết kế để cải thiện hiệu suất của mô hình trên các bộ dữ liệu mất cân bằng. YOLO v4 cũng cải thiện kiến trúc của FPN được sử dụng trong YOLO v3.

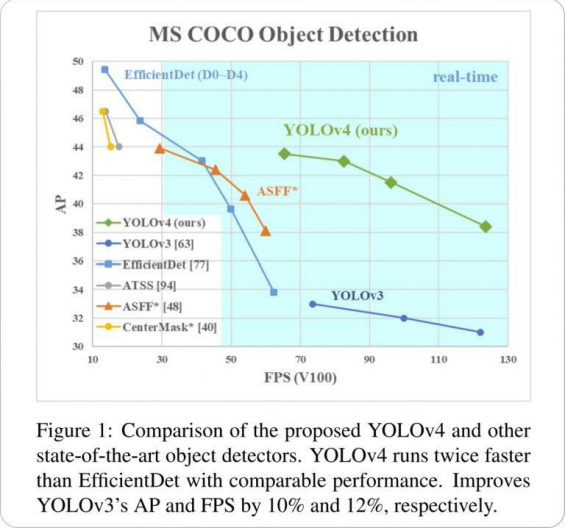


Figure 1: Comparison of the proposed YOLOv4 and other state-of-the-art object detectors. YOLOv4 runs twice faster than EfficientDet with comparable performance. Improves YOLOv3's AP and FPS by 10% and 12%, respectively.

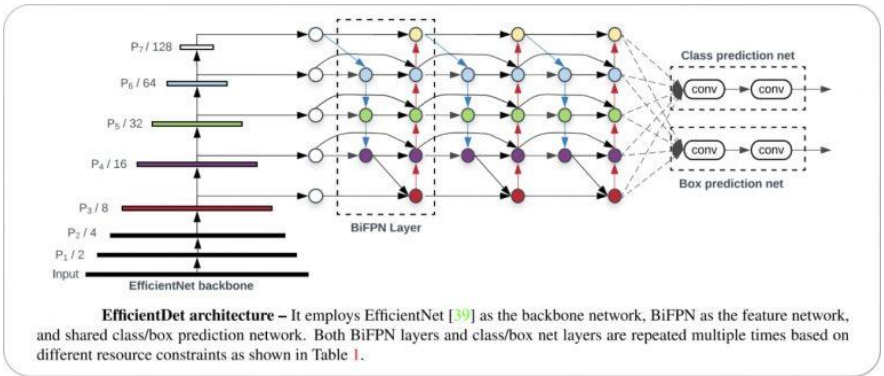
Hình 6. So sánh hiệu năng của YOLO v4

❖ YOLO v5

YOLO v5 được giới thiệu vào năm 2020 bởi cùng một nhóm đã phát triển thuật toán YOLO ban đầu dưới dạng một dự án mã nguồn mở và được duy trì bởi Ultralytics.

YOLO v5 được xây dựng dựa trên sự thành công của các phiên bản trước và bổ sung một số tính năng và cải tiến mới.

Không giống như YOLO, YOLO v5 sử dụng một kiến trúc phức tạp hơn gọi là EfficientDet (kiến trúc hiển thị bên dưới), dựa trên kiến trúc mạng EfficientNet. Việc sử dụng một kiến trúc phức tạp hơn trong YOLO v5 cho phép nó đạt được độ chính xác cao hơn và khả năng khái quát hóa tốt hơn cho nhiều loại đối tượng hơn.



EfficientDet architecture – It employs EfficientNet [39] as the backbone network, BiFPN as the feature network, and shared class/box prediction network. Both BiFPN layers and class/box net layers are repeated multiple times based on different resource constraints as shown in Table 1.

Hình 7. Cấu trúc của mô hình EfficientDet.

Một điểm khác biệt nữa giữa YOLO và YOLO v5 là dữ liệu đào tạo được sử dụng để học mô hình phát hiện đối tượng. YOLO được đào tạo trên bộ dữ liệu PASCAL VOC, bao gồm 20 danh mục đối tượng. Mặt khác, YOLO v5 được đào tạo trên tập dữ liệu lớn hơn và đa dạng hơn có tên là D5, bao gồm tổng cộng 600 danh mục đối tượng.

YOLO v5 sử dụng một phương pháp mới để tạo các anchor box, được gọi là "dynamic anchor boxes". Nó liên quan đến việc sử dụng thuật toán phân cụm để nhóm các hộp giới hạn thực tế thành các cụm và sau đó sử dụng trọng tâm của các cụm làm anchor box. Điều này cho phép các anchor box được căn chỉnh chặt chẽ hơn với kích thước và hình dạng của các đối tượng.

YOLO v5 cũng giới thiệu khái niệm "spatial pyramid pooling" (SPP), một loại lớp tổng hợp được sử dụng để giảm độ phân giải không gian của bản đồ đối tượng địa lý. SPP được sử dụng để cải thiện hiệu suất phát hiện trên các đối tượng nhỏ, vì nó cho phép mô hình nhìn thấy các đối tượng ở nhiều tỷ lệ. YOLO v4 cũng sử dụng SPP, nhưng YOLO v5 bao gồm một số cải tiến đối với kiến trúc SPP cho phép nó đạt được kết quả tốt hơn.

YOLO v4 và YOLO v5 sử dụng loss function tương tự để huấn luyện mô hình. Tuy nhiên, YOLO v5 giới thiệu một thuật ngữ mới gọi là "CIoU loss", đây là một biến thể của IoU loss function được thiết kế để cải thiện hiệu suất của mô hình trên các bộ dữ liệu mất cân bằng.

- **YOLO v6**

YOLO v6 được đề xuất vào năm 2022 bởi Li và cộng sự như một cải tiến so với các phiên bản trước. Một trong những điểm khác biệt chính giữa YOLO v5 và YOLO v6 là kiến trúc CNN được sử dụng. YOLO v6 sử dụng một biến thể của kiến trúc

EfficientNet có tên là EfficientNet-L2. Đó là một kiến trúc hiệu quả hơn so với EfficientDet được sử dụng trong YOLO v5, với ít tham số hơn và hiệu quả tính toán cao hơn. Nó có thể đạt được kết quả tiên tiến trên các điểm chuẩn phát hiện đối tượng khác nhau. Framework của mô hình YOLO v6 được hiển thị bên dưới.

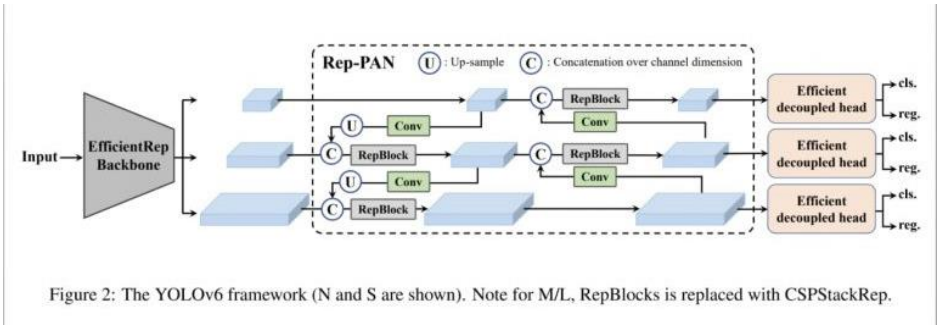
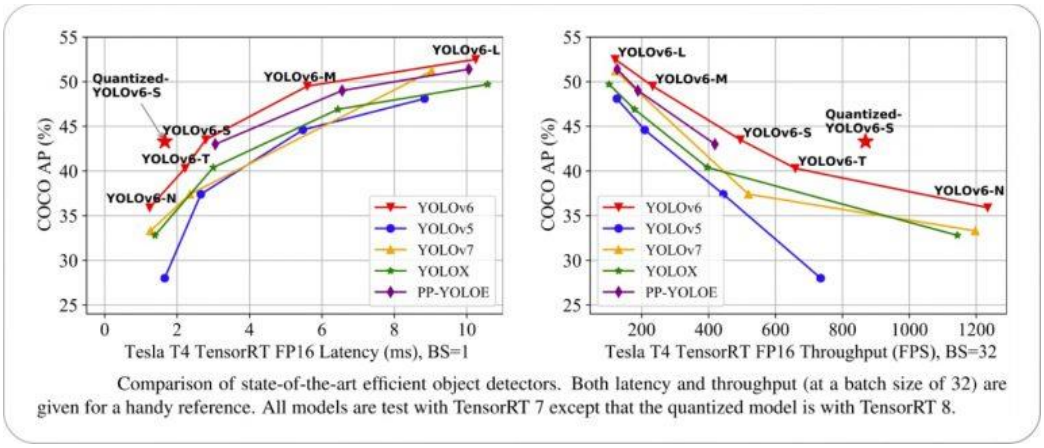


Figure 2: The YOLOv6 framework (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep.

Hình 8. Tổng quan về YOLO v6.

YOLO v6 cũng giới thiệu một phương pháp mới để tạo các anchor box, được gọi là "dense anchor boxes".



Comparison of state-of-the-art efficient object detectors. Both latency and throughput (at a batch size of 32) are given for a handy reference. All models are test with TensorRT 7 except that the quantized model is with TensorRT 8.

Hình 9. Kết quả thu được từ YOLO v6 so với các phương pháp hiện đại khác

❖ YOLOv7

Một trong những cải tiến chính là việc sử dụng các anchor box.

Các anchor box là một tập hợp các hộp được xác định trước với các tỷ lệ khung hình khác nhau được sử dụng để phát hiện các đối tượng có hình dạng khác nhau. YOLO v7 sử dụng chín anchor box, cho phép YOLO phát hiện phạm vi hình dạng và kích thước đối tượng rộng hơn so với các phiên bản trước, do đó giúp giảm số lượng xác định sai.

Một cải tiến quan trọng trong YOLO v7 là việc sử dụng một loss function mới gọi là "focal loss". Các phiên bản trước của YOLO đã sử dụng cross-entropy loss function tiêu chuẩn, được biết là kém hiệu quả hơn trong việc phát hiện các đối tượng nhỏ. Focal loss giải quyết vấn đề này bằng cách giảm trọng số mất mát cho các ví dụ được phân loại tốt và tập trung vào các ví dụ khó—các đối tượng khó phát hiện.

YOLO v7 cũng có độ phân giải cao hơn so với các phiên bản trước. Nó xử lý hình ảnh ở độ phân giải 608 x 608 pixel, cao hơn độ phân giải 416 x 416 được sử dụng trong YOLO v3. Độ phân giải cao hơn này cho phép YOLO v7 phát hiện các đối tượng nhỏ hơn và có độ chính xác tổng thể cao hơn.

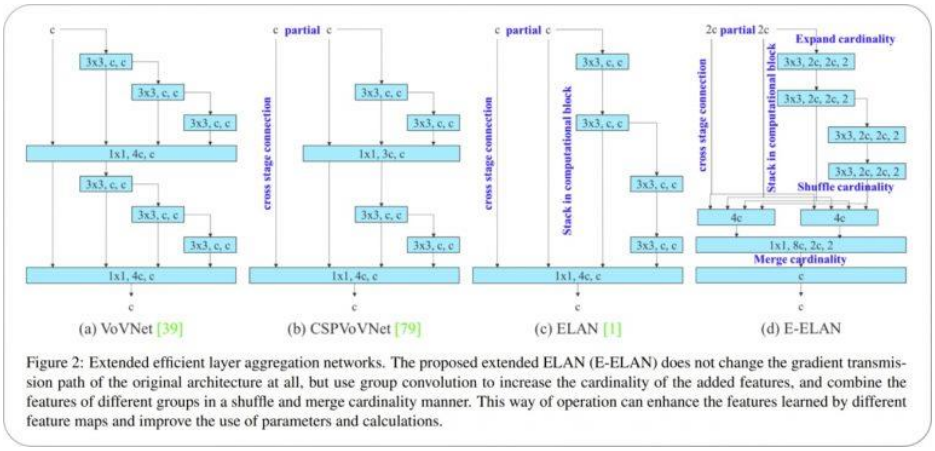
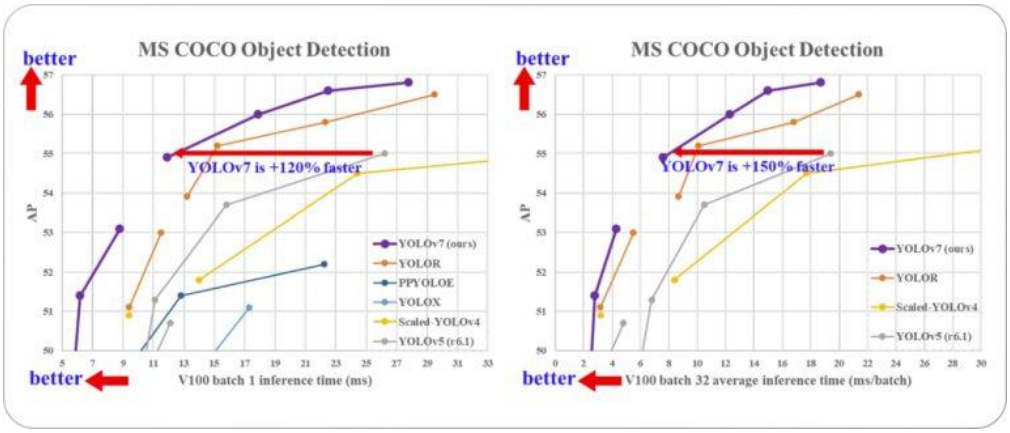


Figure 2: Extended efficient layer aggregation networks. The proposed extended ELAN (E-ELAN) does not change the gradient transmission path of the original architecture at all, but use group convolution to increase the cardinality of the added features, and combine the features of different groups in a shuffle and merge cardinality manner. This way of operation can enhance the features learned by different feature maps and improve the use of parameters and calculations.

Hình 10. Thay đổi sơ đồ tổng hợp lớp của YOLO v7.

Một trong những ưu điểm chính của YOLO v7 là tốc độ. Nó có thể xử lý hình ảnh với tốc độ 155 khung hình mỗi giây, nhanh hơn nhiều so với các thuật toán phát hiện đối tượng hiện đại khác. Ngay cả mô hình YOLO cơ bản ban đầu cũng có khả năng xử lý ở tốc độ tối đa 45 khung hình mỗi giây. Điều này làm cho nó phù hợp với các ứng dụng thời gian thực nhạy cảm như giám sát và ô tô tự lái, trong đó tốc độ xử lý cao hơn là rất quan trọng.



Hình 11. So sánh hiệu suất và tốc độ suy luận của YOLO v7 với các công cụ phát hiện đối tượng thời gian thực hiện đại nhất.

Về độ chính xác, YOLO v7 thể hiện tốt so với các thuật toán phát hiện đối tượng khác. Nó đạt được độ chính xác trung bình là 37,2% ở ngưỡng IoU (giao điểm trên hợp nhất) là 0,5 trên bộ dữ liệu COCO phổ biến, có thể so sánh với các thuật toán phát hiện đối tượng hiện đại khác. So sánh định lượng của hiệu suất được hiển thị dưới đây.

Model	#Param.	FLOPs	Size	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	**51.2%**	**69.7%**	**55.5%**	**35.2%**	**56.0%**	**66.7%**
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X [81]	96.9M	226.8G	640	52.7%	**71.3%**	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	**52.9%**	71.1%	**57.5%**	**36.9%**	**57.7%**	**68.6%**
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	**35.2%**	**52.8%**	**37.3%**	**15.7%**	**38.0%**	**53.4%**
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	**10.9%**	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	**30.8%**	**47.3%**	**32.2%**	10.0%	**31.9%**	**52.2%**
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	**60.4%**	69.2%
YOLOv7-E6	97.2M	515.2G	1280	**55.9%**	**73.5%**	**61.1%**	**40.6%**	60.3%	**70.0%**
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	**42.4%**	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	**56.8%**	**74.4%**	**62.1%**	40.8%	**62.1%**	**70.6%**
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

Hình 12. Đánh giá kết quả dựa trên tập dữ liệu COCO

Tuy nhiên, cần lưu ý rằng YOLO v7 kém chính xác hơn so với các công cụ phát hiện hai giai đoạn như Faster R-CNN và Mask R-CNN, những công cụ này có xu hướng đạt được độ chính xác trung bình cao hơn trên tập dữ liệu COCO nhưng cũng yêu cầu thời gian suy luận lâu hơn.

YOLO v7 là một thuật toán phát hiện đối tượng mạnh mẽ và hiệu quả, nhưng nó có một số hạn chế.

- YOLO v7, giống như nhiều thuật toán phát hiện đối tượng, gặp khó khăn trong việc phát hiện các đối tượng nhỏ. Nó có thể không phát hiện chính xác các đối tượng trong các cảnh đông đúc hoặc khi các đối tượng ở xa máy ảnh.
- YOLO v7 cũng không hoàn hảo trong việc phát hiện các đối tượng ở các tỷ lệ khác nhau. Điều này có thể gây khó khăn cho việc phát hiện các đối tượng rất lớn hoặc rất nhỏ so với các đối tượng khác trong cảnh.
- YOLO v7 có thể nhạy cảm với những thay đổi về ánh sáng hoặc các điều kiện môi trường khác, vì vậy có thể bất tiện khi sử dụng trong các ứng dụng thực, nơi điều kiện ánh sáng có thể thay đổi.
- YOLO v7 có thể đòi hỏi nhiều tính toán, điều này gây khó khăn khi chạy trong thời gian thực trên các thiết bị hạn chế về tài nguyên như điện thoại thông minh hoặc các thiết bị biên khác.

3.5. Mô hình sử dụng – YOLOv8

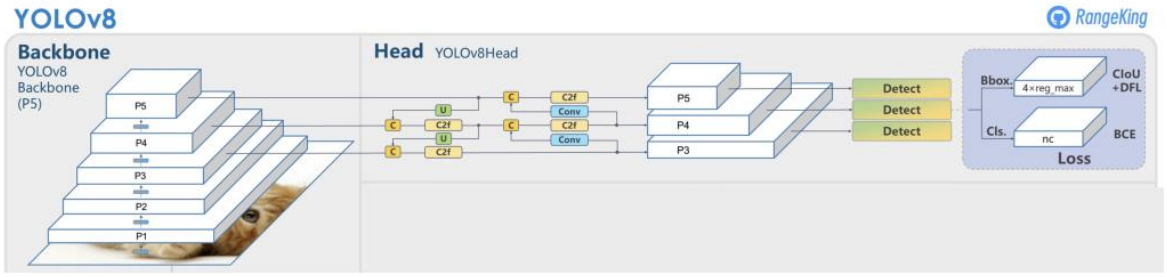
YOLOv8 là mô hình YOLO tiên tiến nhất có thể được sử dụng cho các tác vụ detection, segmentation, pose estimation, tracking và classification.

YOLOv8 được phát triển bởi Ultralytics, cũng chính là nhóm đã tạo ra mô hình YOLOv5 đã đạt được những thành công nhất định trước đây.

YOLOv8 bao gồm nhiều thay đổi và cải tiến về kiến trúc và trải nghiệm người dùng so với YOLOv5.

YOLOv8 đang tiếp tục được phát triển một cách tích cực bởi vì Ultralytics đang lắng nghe phản hồi từ cộng đồng để phát triển thêm những tính năng mới.

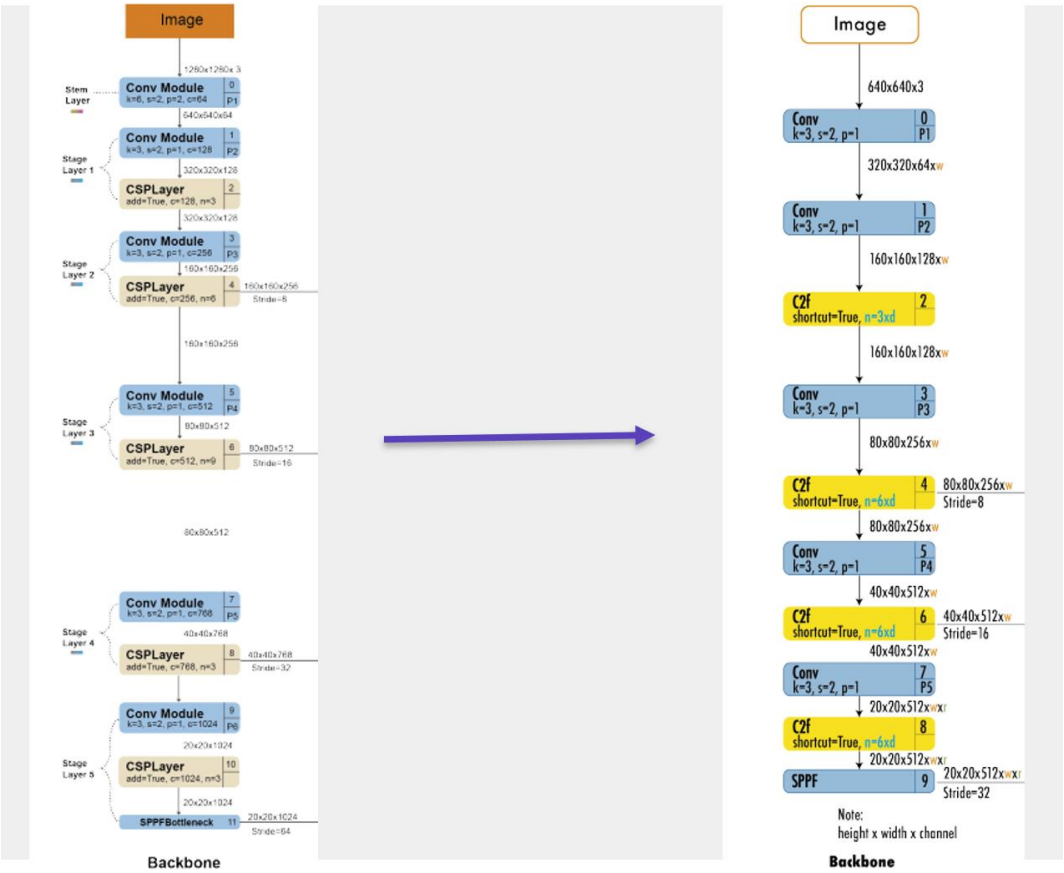
❖ Kiến trúc YOLOv8



Hình 13. Kiến trúc YOLOv8

❖ New Convolutions

YOLOv8 sử dụng một kiến trúc backbone tương tự như YOLOv5 với một số thay đổi trên CSPLayer(Cross-Stage Partial Layer), hiện được gọi là C2f module (cross-stage partial bottleneck with two convolutions).



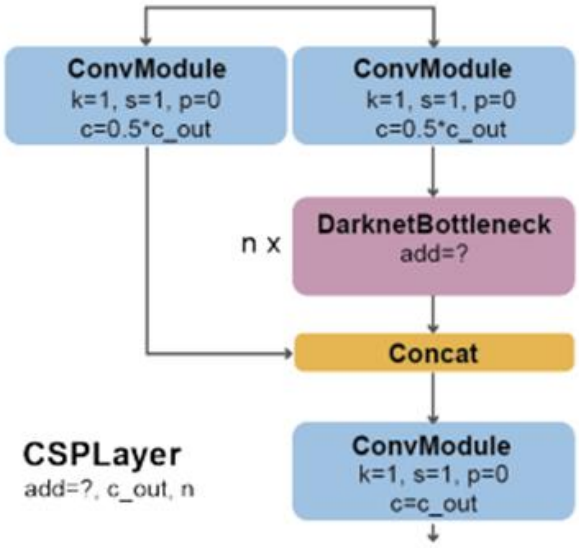
Hình 14. So sánh Backbone của YOLOv5 (trái) và YOLOv8 (phải)

C2f module là một sự cải tiến của CSPLayer (Cross Stage Partial Layer) được sử dụng trong phiên bản trước đó là YOLOv5.

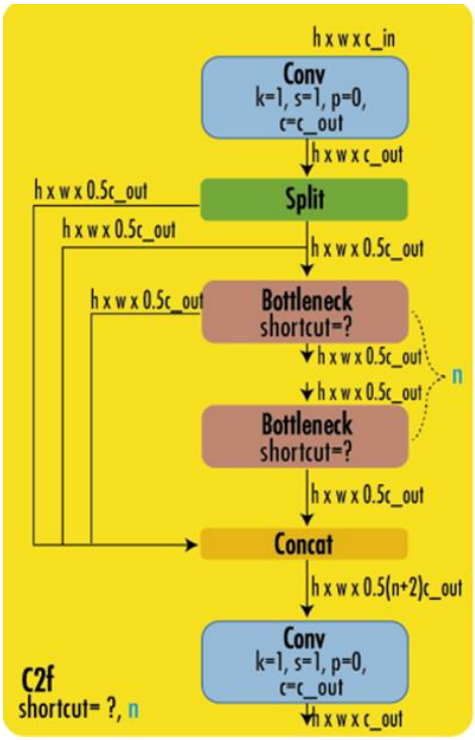
Mô-đun này kết hợp hai phép tích chập giúp kết hợp thông tin từ các tầng thấp đến các tầng cao hơn và cung cấp thông tin ngữ cảnh để cải thiện khả năng phát hiện vật thể của mô hình.

Trong C2f, tất cả các đầu ra từ Bottleneck được nối lại. Trong khi đó, trong CSPLayer chỉ sử dụng đầu ra của Bottleneck cuối cùng.

- Bottleneck giúp tạo ra các biểu diễn đặc trưng sâu hơn và cải thiện khả năng học tập của mô hình.
- Sử dụng để trích xuất thông tin từ đặc trưng đầu vào và giữ lại các thông tin quan trọng trong quá trình huấn luyện mô hình



Hình 15. Kiến trúc CSPLayer

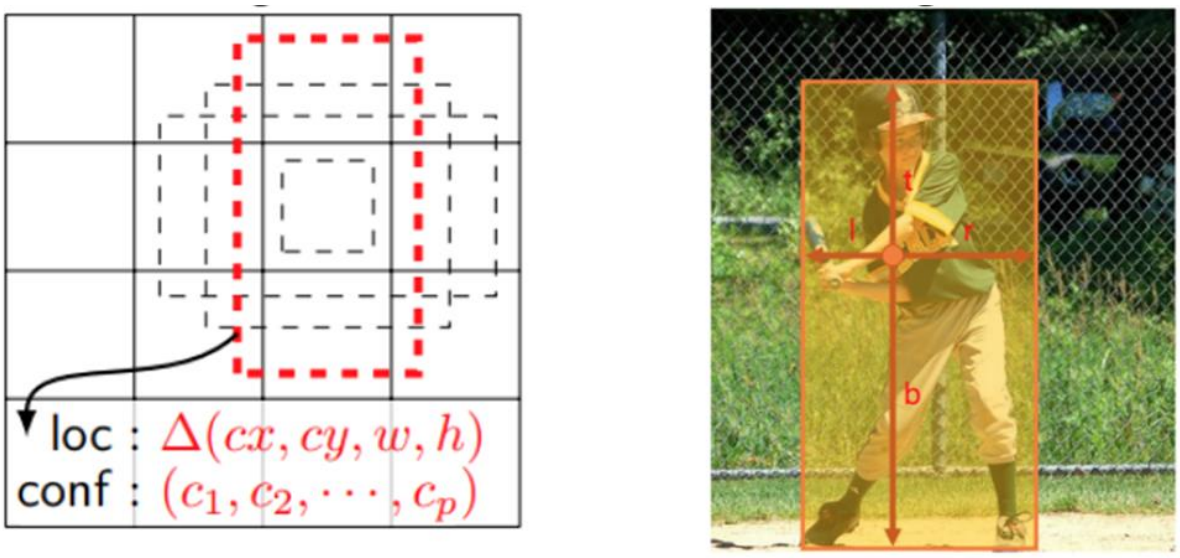


Hình 16. Kiến trúc C2f Module

❖ Anchor Free Detection

YOLOv8 không sử dụng các small, medium và large anchor boxes như các phiên bản trước đó. Thay vào đó, nó sử dụng anchor-free detection để trực tiếp dự đoán tâm của đối tượng thay vì dịch chuyển từ một anchor box đã biết.

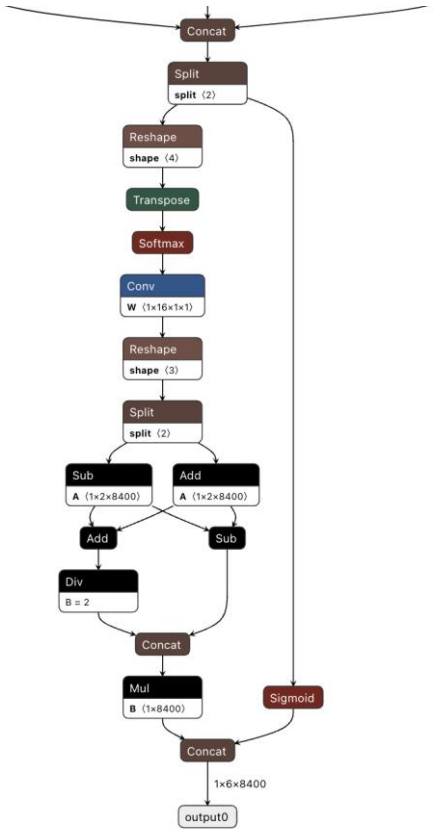
Điều này giúp giảm số lượng dự đoán bounding box, cải thiện tốc độ xử lý và chính xác của mô hình.



Hình 17. Anchor boxes và Anchor Free Detection

Trong lớp đầu ra của YOLOv8, họ sử dụng hàm sigmoid làm hàm kích hoạt cho điểm số đối tượng (objectness score), biểu thị xác suất rằng bouding box chứa một đối tượng.

Ngoài ra, họ sử dụng hàm softmax cho xác suất phân loại, biểu thị xác suất của các đối tượng thuộc về mỗi lớp khả thi.



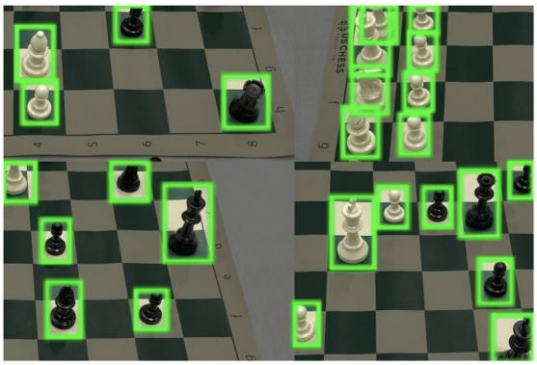
Hình 18. The detection head for YOLOv8

❖ Mosaic Augmentation

Mosaic Augmentation là quá trình kết hợp bốn hình ảnh để buộc mô hình học cách nhận dạng đối tượng ở các vị trí mới, một phần bị che khuất.

Tuy nhiên, đã được chứng minh rằng việc sử dụng Mosaic Augmentation trong toàn bộ quá trình huấn luyện có thể làm giảm độ chính xác của việc dự đoán. Vì vậy, YOLOv8 có thể dừng quá trình này trong các epoch cuối cùng của quá trình huấn luyện.

Điều này cho phép chạy mẫu huấn luyện tối ưu mà không phải áp dụng Mosaic Augmentation cho toàn bộ quá trình huấn luyện. Nhờ điều này, mô hình có thể tuân thủ một mẫu huấn luyện tối ưu và đạt được độ chính xác tốt hơn.

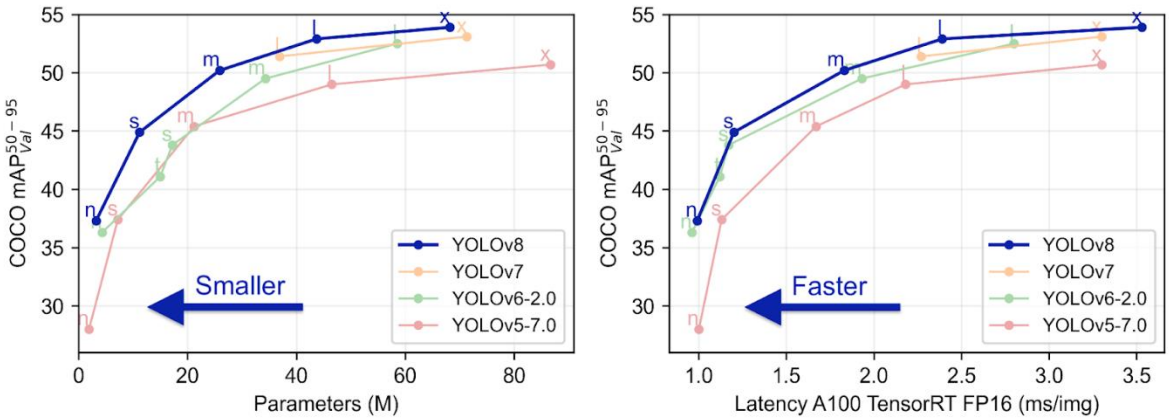


Hình 19. Mosaic augmentation of chess board photos

❖ YOLOv8 Accuracy Improvements

YOLOv8 có độ chính xác cao hơn và nhanh hơn so với các mô hình trước đây (được đánh giá bằng COCO và Roboflow 100).

YOLOv8 đạt được độ chính xác cao trên COCO. Lấy ví dụ, phiên bản YOLOv8m chỉ là phiên bản kích cỡ trung bình nhưng cũng đạt được 50,2% mAP khi đánh giá trên COCO (so với YOLOv5m đạt 45,4% mAP). Khi được đánh giá trên Roboflow 100, YOLOv8 đạt điểm cao hơn đáng kể so với YOLOv5.



Hình 20. YOLOv8 COCO evaluation

❖ YOLOv8 CLI và Python Package

YOLOv8 đi kèm với rất nhiều tính năng thuận tiện cho nhà phát triển, từ CLI dễ sử dụng đến gói Python được cấu trúc tốt.

YOLOv8 may be used directly in the Command Line Interface (CLI) with a `yolo` command:

```
yolo predict model=yolov8n.pt source='https://ultralytics.com/images/bus.jpg'
```

Hình 21. YOLOv8 CLI

YOLOv8 may also be used directly in a Python environment, and accepts the same arguments as in the CLI example above:

```python
from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n.yaml")  # build a new model from scratch
model = YOLO("yolov8n.pt")  # load a pretrained model (recommended for training)

# Use the model
model.train(data="coco128.yaml", epochs=3)  # train the model
metrics = model.val()  # evaluate model performance on the validation set
results = model("https://ultralytics.com/images/bus.jpg")  # predict on an image
path = model.export(format="onnx")  # export the model to ONNX format
```

Hình 22. YOLOv8 Python Package

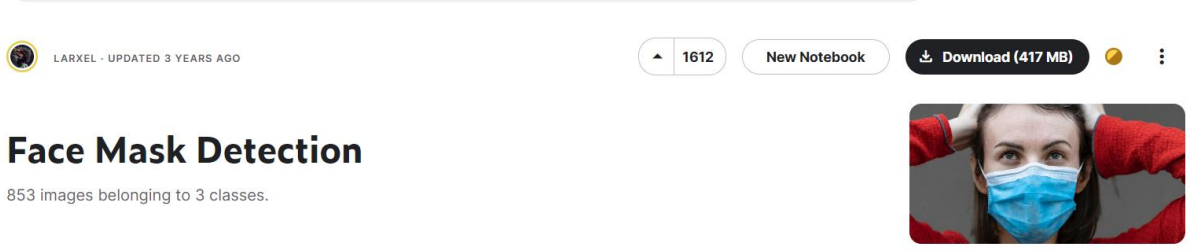
Chính vì những lí do đó mà em quyết định sử dụng mô hình YOLOv8 để áp dụng cho bài toán lần này.

CHƯƠNG 4: THỰC NGHIỆM

4.1. Dataset

Bộ dataset được lấy từ Kaggle: face-mask-detection

Gồm 853 ảnh với 3 class: With mask;Without mask; Mask weared incorrect



Hình 23. Face Mask Detection dataset

❖ **Nhận xét:**

- Các bộ data có sẵn không đáp ứng về độ phong phú và đa dạng (trời nắng, trời mưa, khẩu trang có kiểu dáng đặc biệt, …).
- Mất cân bằng dữ liệu (imbalanced data): số lượng bounding boxes của nhãn có sự chênh lệch lớn
- Việc mất cân bằng dữ liệu gây ảnh hưởng đến việc huấn luyện và đánh giá mô hình sau này.

4.2. Cải tiến Dataset

Chính vì những khó khăn đó nên em quyết định cải tiến Dataset bằng cách: Tăng cường thêm dữ liệu. Cụ thể:

- Tăng thêm 100 ảnh với nhiều dáng khẩu trang khác nhau được thu thập từ Internet
- Sử dụng công cụ Roboflow để gán nhãn dữ liệu

Hình 24. Một số ảnh tăng cường

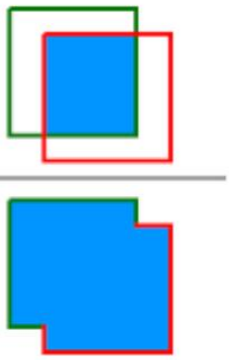
4.3. Đánh giá mô hình

Nếu như trong bài toán classification thì ta có thể đơn giản dùng precision để đánh giá model bằng cách lấy tổng số sample được phân loại đúng trên tổng số sample thực hiện phân loại, hay có thể sử dụng F1 score để tính,… thì trong object detection để đánh giá hiệu suất của model thì người ta sử dụng mAP như là một metric phổ biến để đánh giá hiệu suất của model.

4.3.1. IoU

IoU (Intersection over Union) chỉ ra độ khớp giữa bounding box được mô hình dự đoán và ground truth box do con người gán sẵn. Với mỗi ground truth box ta sẽ tiến hành tính IoU với tất cả các bounding box mà mô hình dự đoán

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



Hình 25. Minh họa cách tính IoU

Dựa vào thông số IOU này người ta sẽ xác định TP, FP, TN, FN như sau:

- TP (True positive): Khi IOU của predicted box vs gtbox ≥ iou threshold
- FP (False positive): Khi IOU của predicted box vs gtbox < iou threshold
- TN (True negative): Thông số này ta có thể hiểu nó như là background và ta sẽ không cần quan tâm thông số này.
- FN (False negative): Bouding box của đối tượng không được detect (detect sót)

4.3.2. Precision

Precision là tỷ lệ trường hợp bounding box có IoU ≥ threshold trong các bounding box được dự đoán

$$precision = \frac{TP}{TP + FP} = \frac{TP}{\text{Tổng số dự đoán}}$$

Hình 26. Công thức tính Precision

4.3.3. Recall

Recall là tỷ lệ bounding box được dự đoán có IoU ≥ threshold trên tổng số ground truth

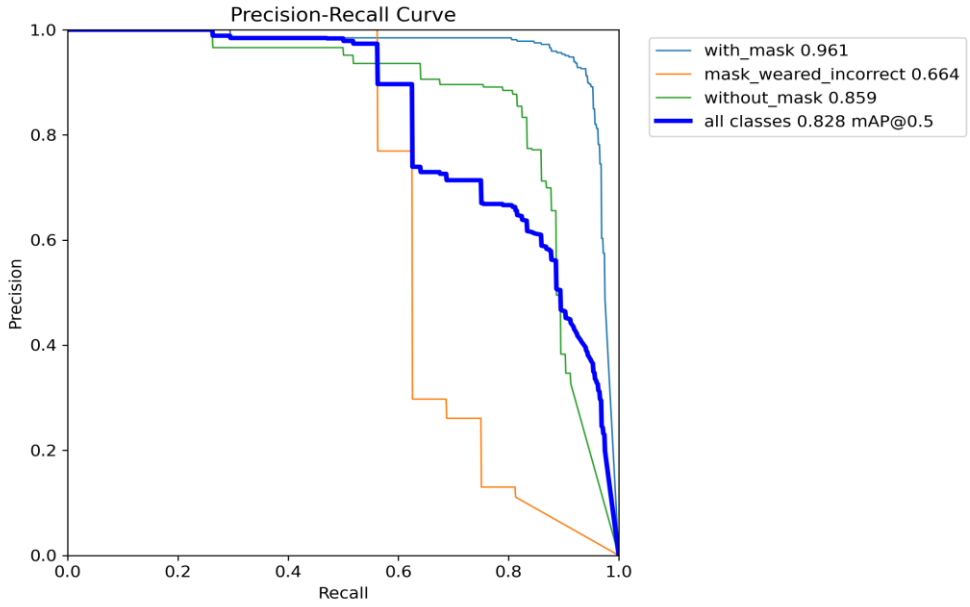
$$recall = \frac{TP}{TP + FN} = \frac{TP}{\text{Tổng số gtbox}}$$

Hình 27. Công thức tính Recall

4.3.4. PR Curve (Precision – Recall Curve) và AP (Average Precision)

Đường cong Precision Recall cho biết sự cân bằng giữa Precision và Recall đối với các giá trị confidence khác nhau.

AP(average precision) chính là phần diện tích phía dưới đường cong PR curve



Hình 28. Minh họa Đường cong Precision Recall

4.3.5. MAP

mAP sẽ là trung bình cộng AP của tất cả các class.

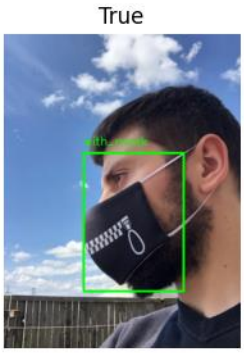
Bên dưới là bảng kết quả đánh giá mô hình trên tập Valid trước và sau khi tăng cường dữ liệu

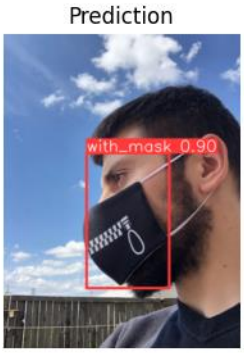
	Trước	Sau
Precision	85.5	91.3
Recall	77.1	77.5
mAP	82.7	87.2

Hình 29. Bảng kết quả Precision, Recall, mAP của mô hình trước và sau khi tăng cường thêm dữ liệu

4.4. Kết quả thực nghiệm

4.4.1. Một số kết quả từ tập Test





4.4.2. Một số ảnh từ Internet

Image

Prediction

Image

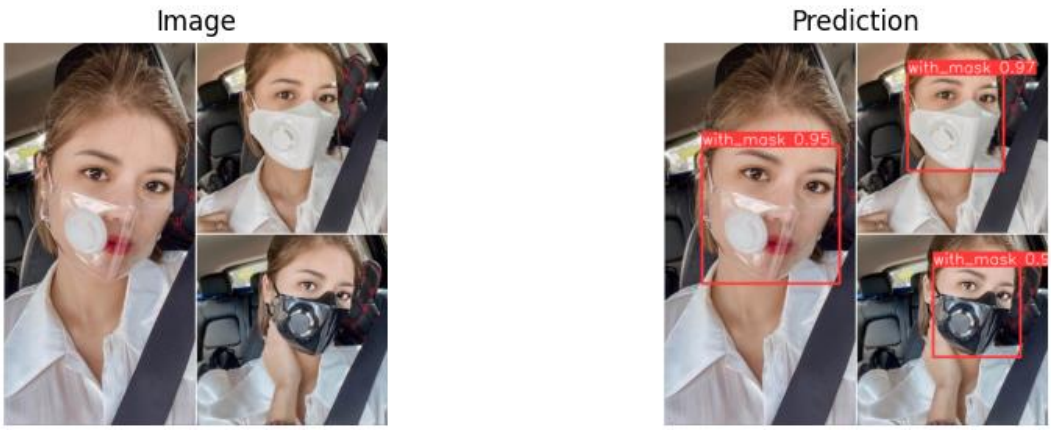
Prediction

Image

Prediction

Image

Prediction



CHƯƠNG 5: ỨNG DỤNG VÀ HƯỚNG PHÁT TRIỂN

- Nghiên cứu cải tiến phương pháp để giải quyết các trường hợp khó nhận diện như khuôn mặt người khi ở xa, khi quay trái hoặc quay phải, khi bị che khuất một phần, ảnh bị mờ…
- Đồng thời nghiên cứu tìm ra giải pháp để nâng cao độ chính xác của hệ thống.
- Áp dụng bài toán vào thực tế

TÀI LIỆU THAM KHẢO

[1] "What Is YOLOv8? The Ultimate Guide." Accessed June 21, 2023. https://blog.roboflow.com/whats-new-in-yolov8/#the-yolov8-annotation-format.

[2] "MAP TRONG OBJECT DETECTION." Accessed June 21, 2023. https://viblo.asia/p/map-trong-object-detection-38X4E55j4N2.

[3] "[2304.00501] A Comprehensive Review of YOLO: From YOLOv1 and Beyond." Accessed June 21, 2023. https://arxiv.org/abs/2304.00501.

[4] "YOLO V7: Thuật Toán Phát Hiện Đối Tượng Có Gì Mới? - Công Ty Cổ Phần VinBigData." Accessed June 21, 2023. https://vinbigdata.com/kham-pha/yolo-v7-thuat-toan-phat-hien-doi-tuong-co-gi-moi.html#YOLO_hoat_dong_nhu_the_nao_Kien_truc_YOLO.

[5] "Ultralytics/Ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > CoreML > TFLite." Accessed June 21, 2023. https://github.com/ultralytics/ultralytics.