YOLOv8

* Hiện tại tui đang học và tìm hiểu các phiên bản, block và cách tính toán từng block thoi nên tui chưa hiểu sâu về các thuật toán của Yolo cho lắm thì sau đây là những gì tôi đã học được:

YOLO là một mô hình nhận diện đối tượng trong ảnh với đặc điểm nhận diện nhanh chóng và hiệu quả. Mỗi phiên bản của YOLO đã được cải tiến để tối ưu hóa tốc độ và độ chính xác. Dưới đây là các phiên bản YOLO chính:

1. **YOLOv1**

* **Cấu trúc**: chia ảnh đầu vào thành lưới (grid) và mỗi ô lưới dự đoán các **bounding box** và **class**.
* **Điểm mạnh**: Nhanh, đơn giản, và có thể nhận diện nhiều đối tượng trong ảnh.
* **Điểm yếu**: Khó nhận diện đối tượng nhỏ và gặp vấn đề khi các đối tượng nằm gần nhau.

**2. YOLOv2**

* **Cải tiến**: Sử dụng **Darknet-19** làm backbone và **anchor boxes** để cải thiện khả năng dự đoán các đối tượng có kích thước khác nhau.
* **Điểm mạnh**: Tốc độ nhanh, cải thiện nhận diện đối tượng nhỏ hơn.
* **Điểm yếu**: Khó khăn khi gặp đối tượng với tỷ lệ cực kỳ nhỏ.

**Note**: Darknet-19 là mạng trích xuất đặt trưng có 19 lớp (là một kiến trúc CNN)

Anchor boxes là mô hình nhận diện đối tượng như YoLo

**3. YOLOv3**

* **Cải tiến**: **Darknet-53** làm backbone, cải thiện khả năng phát hiện đối tượng nhỏ.
* **Điểm mạnh**: Chính xác hơn và hỗ trợ nhiều kích thước đối tượng.
* **Điểm yếu**: Cần nhiều tài nguyên tính toán hơn.

**4. YOLOv4**

* **Cải tiến**: **CSPDarknet53** làm backbone, sử dụng **mosaic augmentation**, **Mish activation**, và các kỹ thuật như **SAT** để cải thiện độ chính xác và tốc độ.
* **Điểm mạnh**: Đạt hiệu suất cao, tối ưu với cả tốc độ và độ chính xác.
* **Điểm yếu**: Cần tài nguyên tính toán cao.

**Note:** Mosaic Augmentation là một kỹ thuật tăng cường dữ liệu được sử dụng trong các mô hình học sâu

Mish Activation là một hàm kích hoạt có tác dụng giúp mạng học các đặc trưng tốt hơn.

**5. YOLOv5**

* **Cải tiến**: Được phát triển bởi **Ultralytics**, sử dụng **PyTorch** và dễ triển khai hơn.
* **Điểm mạnh**: Đơn giản hóa, dễ dàng sử dụng và có nhiều mô hình khác nhau.
* **Điểm yếu**: Không phải là phiên bản chính thức của YOLO.

**6. YOLOv6**

* **Cải tiến**: Tối ưu cho các thiết bị di động và edge devices.
* **Điểm mạnh**: Phù hợp với môi trường triển khai trên thiết bị di động.
* **Điểm yếu**: Không phổ biến như YOLOv5.

**Note**: edge devices là thiết bị xử lý các dữ liệu gần và nơi dư liệu được tạo ra

**7. YOLOv7**

* **Cải tiến**: **Trainable transformer** để cải thiện khả năng phân loại và dự đoán.
* **Điểm mạnh**: Độ chính xác cao trong các bài toán khó.
* **Điểm yếu**: Cần tài nguyên tính toán cao.

**Note**: : Trainable transformer là

**8. YOLOv8**

* **Cải tiến**: Sử dụng **PANet** và **CSPDarknet53** để cải thiện độ chính xác và tốc độ.
* **Điểm mạnh**: Tối ưu cho các tác vụ nhận diện đối tượng trong thời gian thực.
* **Điểm yếu**: Cần tài nguyên tính toán cao.

**Các block trong mô hình YOLO và cách tính toán:**

1. **Input Layer**: Hình ảnh được chuẩn hóa (chia cho 255).
2. **Backbone**:
   * **Convolutional Layers**: Trích xuất đặc trưng từ ảnh.
   * **ReLU Activation**: Tạo tính phi tuyến.
   * **Pooling**: Giảm kích thước, giữ lại thông tin quan trọng.
3. **Head**:
   * **Bounding Box Prediction**: Dự đoán các tọa độ của bounding box.
   * **Class Prediction**: Dự đoán lớp của đối tượng.
   * **Confidence Score**: Dự đoán mức độ xác suất có đối tượng trong bounding box.
4. **Loss Function**:
   * **Localization Loss**: Lỗi về tọa độ của bounding box (MSE).
   * **Confidence Loss**: Lỗi dự đoán có đối tượng (Binary Cross-Entropy).
   * **Classification Loss**: Lỗi dự đoán lớp đối tượng (Categorical Cross-Entropy).
5. **NMS (Non-Maximum Suppression)**: Chọn bounding box có độ tin cậy cao nhất và loại bỏ các dự đoán dư thừa.

Còn **FPN** và **PANet** thì em chưa hiểu cho lắm vì nó chúng liên quan đến việc xử lý và kết hợp các đặc trưng từ nhiều cấp độ khác nhau trong mạng neural.