王建堯老師 - YOLOv9

出自

ECCV

題目

YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information

YOLOv8

1. anchor-free
2. C2f：CSP 的加強版，時間更短。會先透過 1 * 1 conv (第一個 conv) 將 input channel 壓縮成 c′, 接著 c′ 分成 c1′ 與 c2′ ， c1′ 直接是 residual 而 c2′ 則是比 CSP 更短的一串 NN (內涵第二個 conv) ，最後這兩個 concate 後再經過 fusion conv 做結合 (f)
3. neck：SPPF (SPP 的進化版)，不用多種 pooling 而是用多層 pooling ex：5 * 5 再 5 * 5 可以得到 9 * 9 pooling ，加快速度

YOLOv10

1. 用 Consistent Dual Assignments 取代 NMS ， 加快 inference
2. Lightweight Classification Heads：Spatial-Channel Decoupled Downsampling 減少下採樣過程中的信息損失
3. Rank-Guided Block Design：分析模型各阶段的内在冗余性，动态调整模块设计

YOLOv11：特徵提取

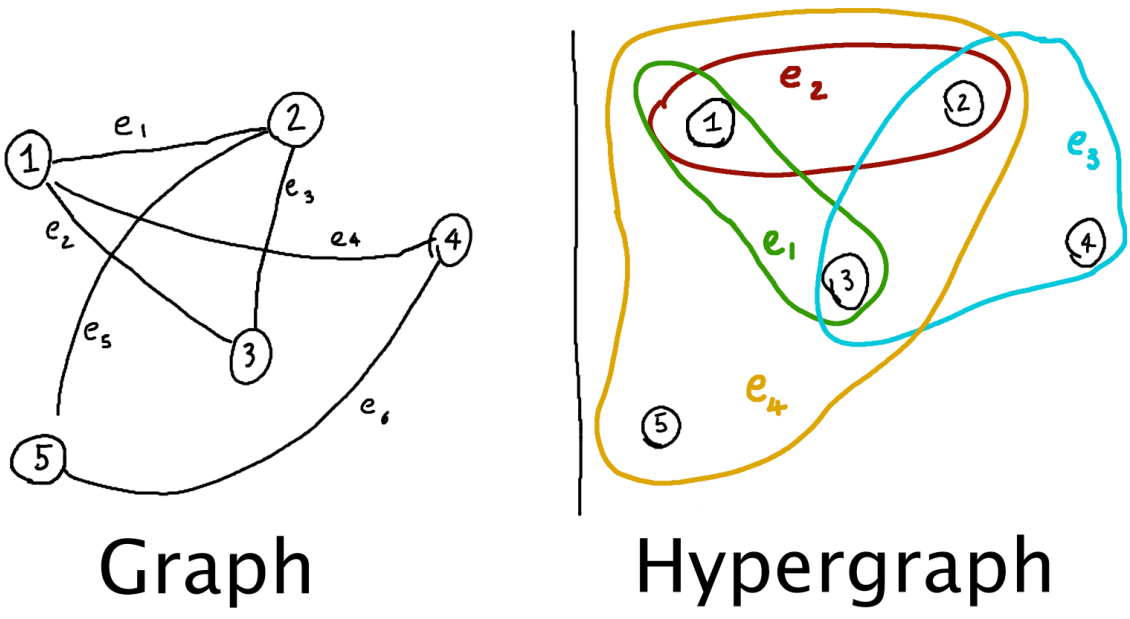
1. Cross-Stage Partial with kernel size 2：取代 C2f ， 提高速度
2. Convolutional block with Parallel Spatial Attention : 增加抗遮擋
3.

YOLOv12：

1. Area Attention : 將特徵圖劃分為多個區域，進行自注意力計算
2. R-ELAN backbone
3. FlashAttention

YOLOv13：

1. 超圖（Hypergraph）計算 : 捕捉場景中物體之間的高階關聯
   傳統圖 graph 上的 edge 只能兩個 vertex 連成一條 edge，現在 hyper edge 可以由多個 vertex 組成，因而代表更複雜的關係



2. 深度可分離卷積模塊 = depth-width + pointwise

Introduction

背景

動機

information bottleneck : input data 可能在 forward 中有 information loss

目的

提出 programmable gradient information (PGI) 來解決 infromation bottleneck 並透過 auxiliary reversible branch 來達成，他會用不同 semantic levels 的資訊，而不會像 deep supervision 一樣受到 multi-path features 的影響 (因為每一個 head 的 loss 導致資訊互相影響)。同時 auxiliary reversible branch 也是 Reversible 的，可以更加不受 information bottleneck

backbone 使用 GELAN

PGI 也可以用在淺層網路，不像 Deep Supervision 只能用在深層 network

相關研究

1. 現存解決 information bottleneck 的方法 :

   (1) Reversible architectures : 藉由 repeated input data 。 但是會需要額外的 layer 而增加 inference time，而且他的網路不能太深，沒辦法抽取到高階語意

   Reversible Module : 因為要可以 reversible ，所以她的 output 一定含有大量的 input 訊息
   RevCol : 把 reversible layer 擴展成多層，得到更高層的語意

   (2) Masked modeling : 遮住某塊，讓 model 預測被遮住的地方，目標是 min reconstruction loss 。缺點是這個 reconstruction loss 有可能會和 target task 的 loss 衝突

   (3) deep supervision (Auxiliary supervision) : 缺點是會有 error accumulation 的問題，如果前面幾層就不夠好了，那後面幾層更是完蛋

   (4) increase the width of the layer : 更大的 W 更有機會保存 input 資訊

2. DETR 系列 : backbone 是 CNN, head 使用 transformer 的 encoder 與 decoder 。因為計算成本昂貴，所以通常不會再用資料重 train 而是使用 pretrain model 所以泛用性沒有 yolo 高

3. Res2Net module

   在 **channel 維度**上切成 s 個 partitions x1 ~ xs ，經過一個一個 partition 的混和，然後再將 y1, y2 ... 合併

1. 第一個 partition x_1 → 直接送到 3x3 conv → y_1
2. 第二個 partition x_2 → 加上 y_1 再做 conv → y_2
3. 第三個 partition x_3 → 加上 y_2 再做 conv → y_3

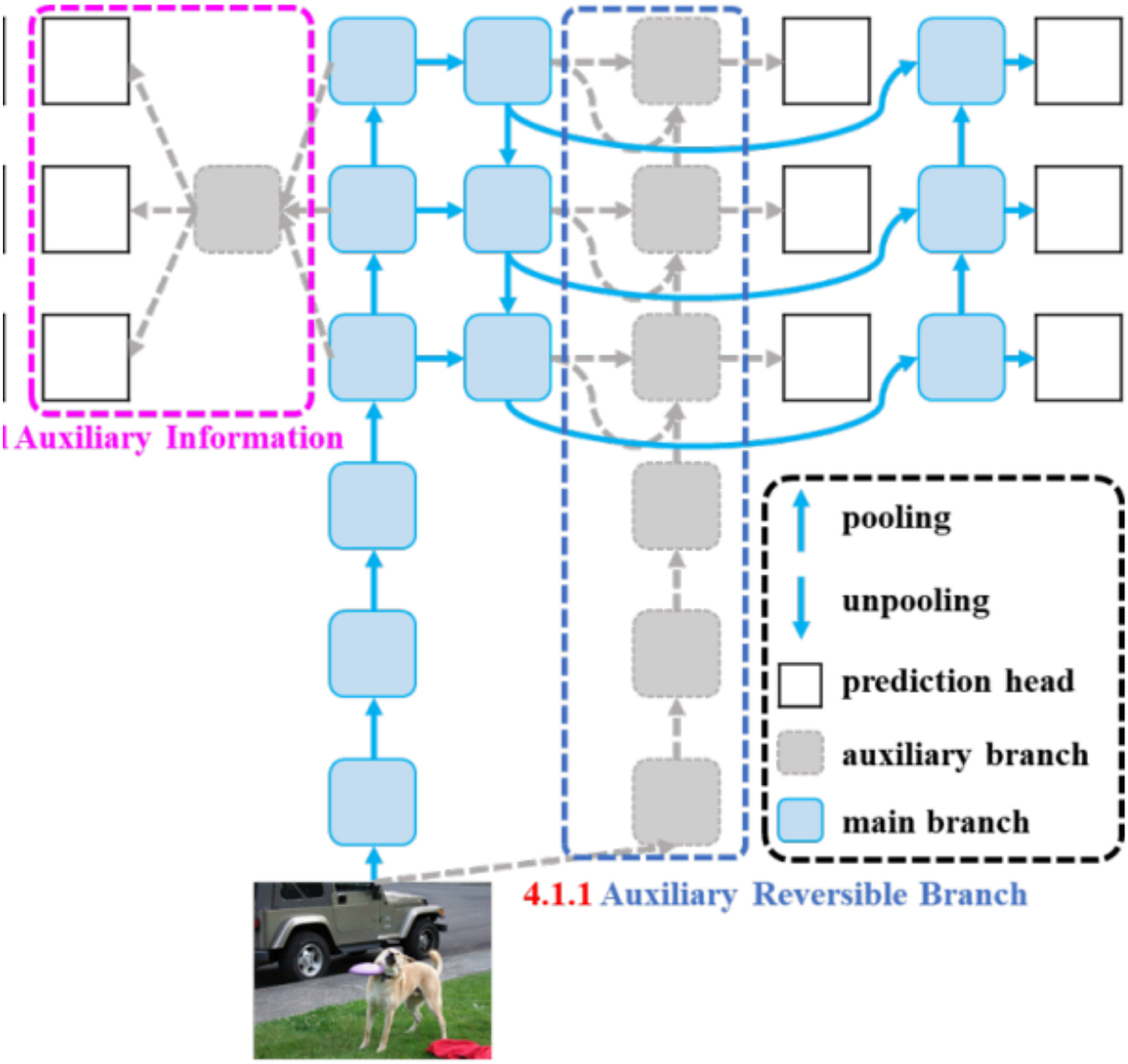
4. 現在的 object detection model 使用的 head，是 improved YOLOv3 head 或是 FCOS head

5. gradient vanish ⇒ 已經因為 normalization 與 activation function 而修補好了

6. Perceiver : 優化後的 transformer，原本的 transformer 要 O(N^2)， Perceiver 會把 N 投影到 f 再做 attention

Train-from-scratch : 是從隨機化的參數開始訓練的，而不是從一個 preset 的參數下去開始的

方法



(d) Programmable Gradient Information

PGI 由三個部分組成 :

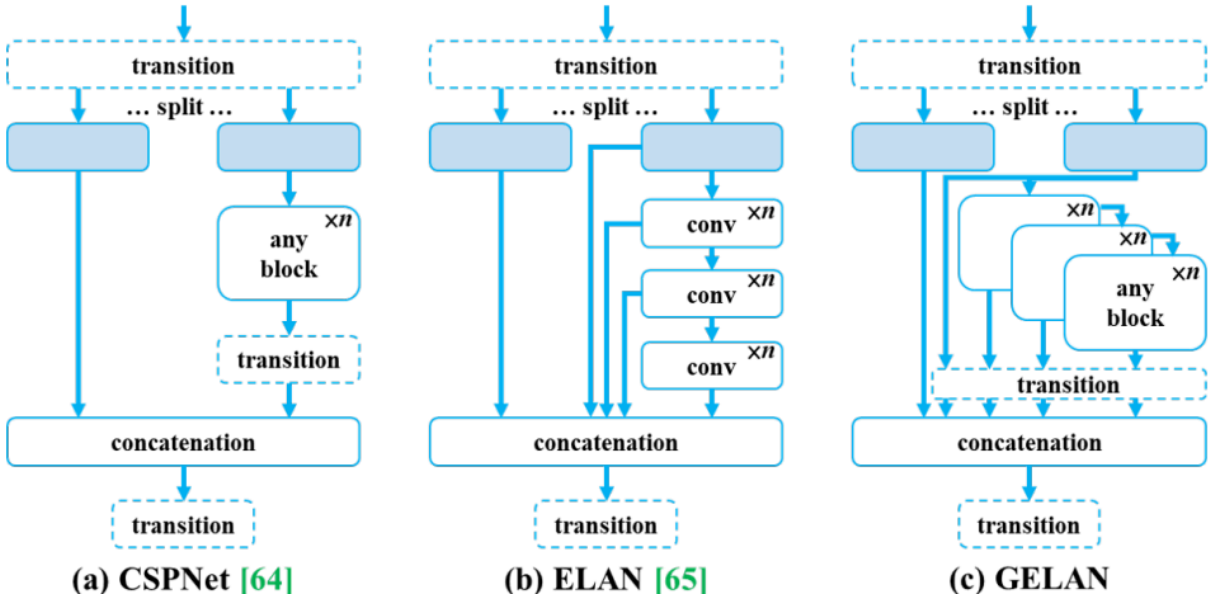
(1) main branch, (2) auxiliary reversible branch, and (3) multi-level auxiliary information

(2) auxiliary reversible branch : 用來解決 information bottleneck

(3) multi-level auxiliary information : 用來解決 deep supervision 的 error accumulation problem

原本的 deep supervision 會將其他 size 的 object 當成背景，因此會損失 information

GELAN = CSPNet + Elan



1. 前 3 個 epoch : linear warm-up
   接著開始 decay
   最後 15 個 epoch 關掉 mosaic data augmentation

Experiments

1. GELAN 裡面的 Module 塞 CSP blocks 最佳
2. multi-level auxiliary information : 使用 FPN or PAN
3. auxiliary reversible branch 為 ICN to use DHLC [34] linkage to obtain multi-level reversible information.， vs PFH(tranditional deep supervision)

可以學習的地方

問題