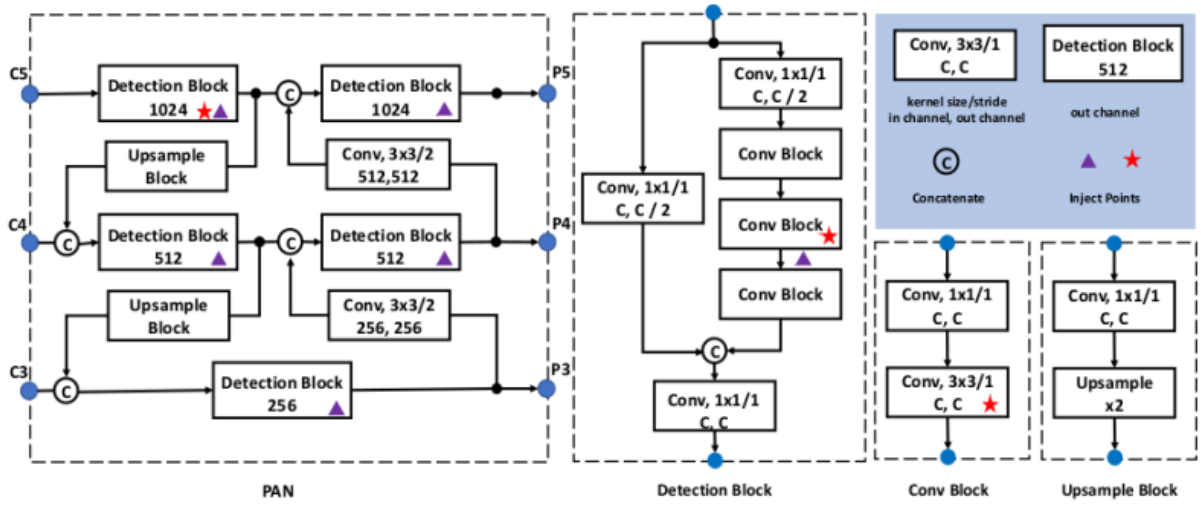
Task: 08

PP YOLO

Paddle paddle YOU ONLY LOOK ONCE -- V2

The PP YOLO v2 boosts performance from 45.9% - 49.5% mAP, runs on 68.9FPS at 640x640 input size & achieves better balance between speed & accuracy than YOLOv4 & v5.



Backbone :

1. In PP-Yolo v1, ResNet-50 is used as Backbone , but in PP-Yolo v2 ResNet101 is used as Backbone.
2. ResNet-101 is pre-trained with COCO dataset.

Neck :

1. Path Aggregation Network(PAN) is used as the Neck part of PP-Yolo v2. Path Aggregation Network is an improved version of PP-Yolov2.

Detection Part:

1. Detection part is designed in the structure of Residual Blocks which has 3x3 and 1x1 convolutional blocks to produce detection results.
2. The output channel of each final prediction is 3(K + 5), where K is the number of classes.

Optimizations :

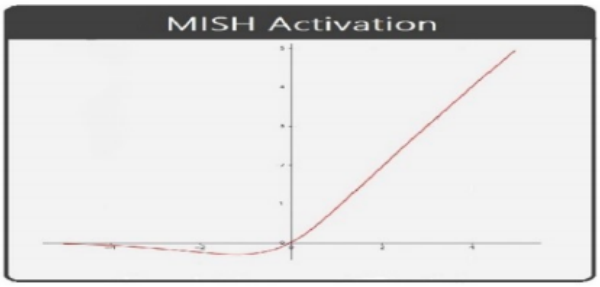
1. Apart from PP-Yolo v1 optimizations , v2 has introduced lots of optimization techniques.

Path Aggregation Network:

1. PP-Yolo v2 uses PAN as the Neck part of Architecture. PAN is an advanced version of Feature Pyramid Network.
2. Normal FPN merges the high level features to low level features in TopBottom operation. So the features of middle and low level are modified . This gives difficulties for detecting small objects.
3. To overcome the above problem, the PANet comes with an additional layer pyramid of Bottom-top , this gives very efficient extraction of low and middle level features.

Mish Activation Function:

In Yolo v4 and v5, Mish activation function gives good results in accuracy, so that PP-Yolo v2 also uses Mish Activation Function. The Mish Activation function is only used in Path Aggregation Network(PAN) in PP-Yolo v2.



$$f(x) = x \cdot tanh\big(softplus(x)\big)$$
$$= x \cdot tanh(\ln(1 + e^x))$$

Large Input Size:

1. Increasing input size is one of key techniques to improve model accuracy. But increasing image size will take up a large memory space .
2. Decreasing batch size is one of the solutions for the above problem, image size is increased from 608 to 768 and batch size is decreased from24 to 12.

IoU Aware Lose:

In PP-Yolo v2 modified IoU Loss is introduced for better result in bounding box prediction.

Optimization technique that didn't work:

Author tried some other optimization also, but that didn't improve performance. The are some of that techniques,

1. Cosine Learning Rate Decay
2. Backbone Parameter Freezing
3. SiLU activation function

Conclusion :

Finally PP-YOLOv2 runs at 68.9FPS at 640x640 input size. Paddle inference engine with TensorRT, FP16-precision and batch size = 1 further improves PP-YOLOv2’s infer speed, which achieves 106.5 FPS. MAP also gives as 45.5