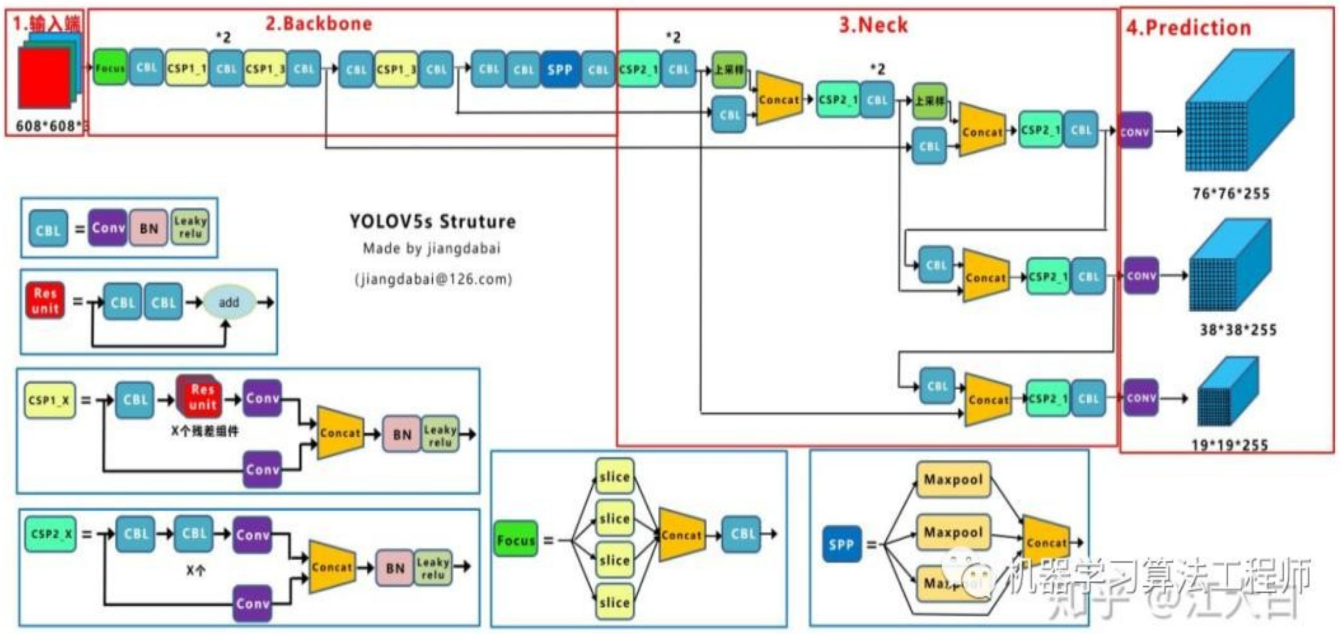
Name : Panneerselvam N
Role: Research Intern – Inspect
Date : 04/06/2021

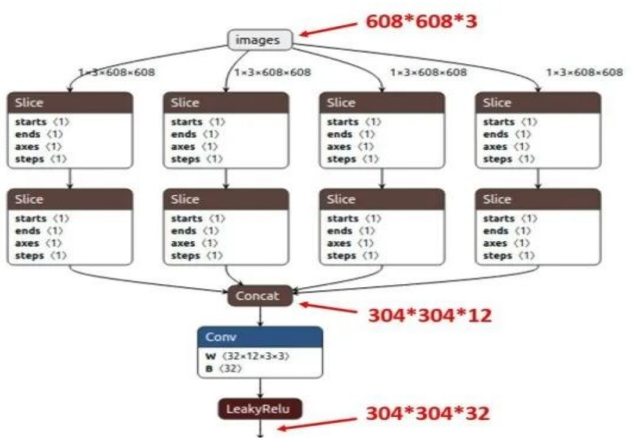
<u>You Only Look Once v5 (YOLOv5)</u>

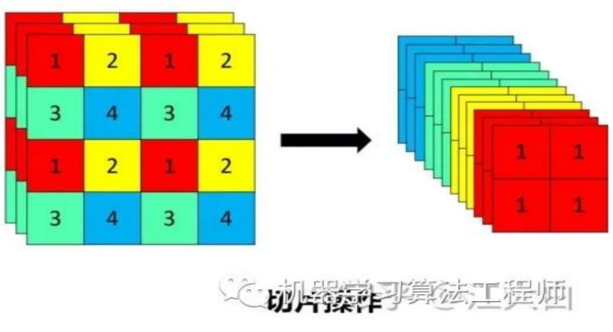
Yolo v5 is almost similar to yolo v4, but some modifications are done in architecture and optimization.
Yolo v5 is implemented in PyTorch framework.
Architecture :



Above figure shows the architecture of Yolo v5.
Backbone :
Cross Stage Partial Darknet(CSPDarknet) is used as Backbone for Yolo v5. Cross Stage Partial Net is Modified version DenseNet and DenseNet is advanced version of ResNet.

Focus Structure is a new concept implemented in Yolo v5.





Above image shows operation of Focus Structure.
Simply, Focus Structure means it slice the input image and transform into new size.
Ex:
If input image is 608*608*3 after slicing image will be 304*304*30

Neck :
In Yolo v5 , Feature Pyramid Network(FPN) and Path Aggregation Network(PAN) is used as Neck part.

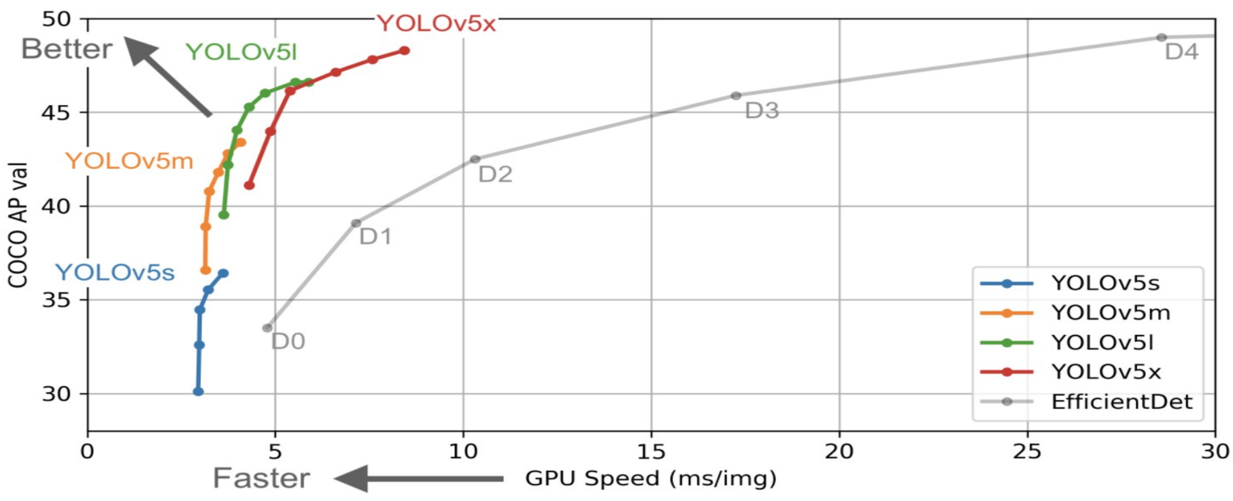
In the Neck structure of Yolov4, ordinary convolution operations are used. In the Neck structure of Yolov5, the CSP2 structure designed by CSPnet is adopted to strengthen the ability of network feature integration.

Output :
In detection part all are same as Yolo v4 , but bounding box loss function is changed as Generalized Intersection of Union(GIoU).

Optimization Techniques:

- Mosaic data enhancement
- Adaptive anchor frame calculation
- Adaptive image scaling

Performance :

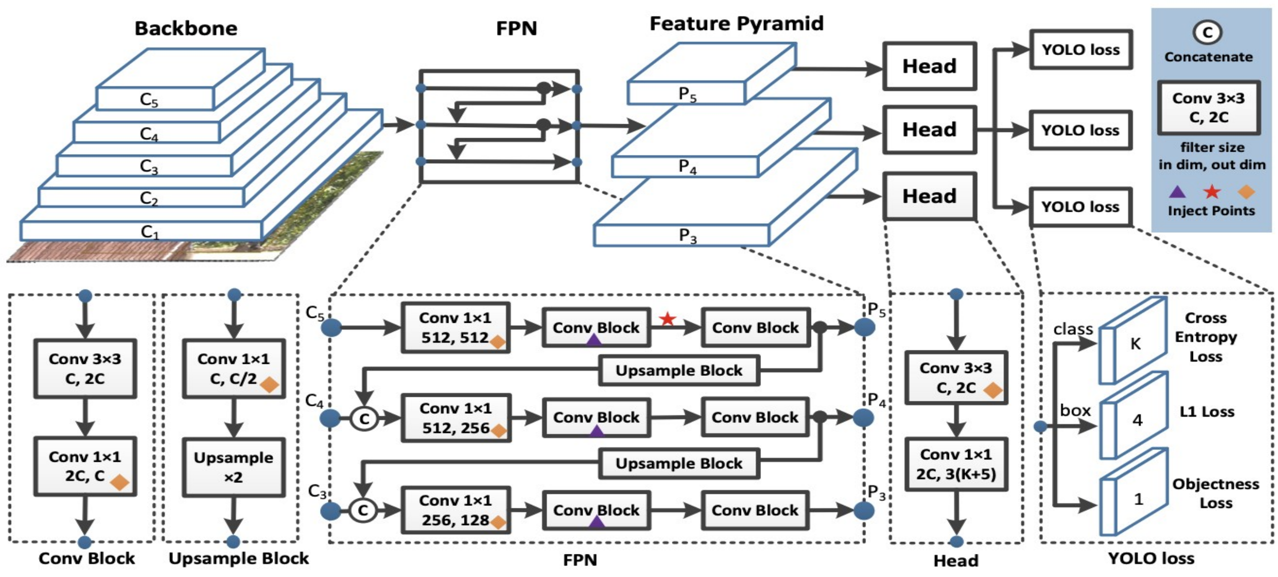


Speed and Accuracy of Yolo v5 is higher than Yolo v4. Size is also very less. Yolo v5 is best suit for Embedded device applications.

Paddle Paddle You Only Look Once(PP-YOLO v1)

PP-Yolo is created from the inspiration of Yolo v3. PP-Yolo gives more accuracy and speed than YOLO v4.

Architecture:



Backbone:

- ResNet-50-vd-dcn is used as Backbone for PP-Yolo. It is combination of ResNet-50 and Deformable Convolution.

- Deformable Convolution is only used in last stage of ResNet. Because DFN takes some amount Computations which affects model performance in FPS.

Neck:

- Feature Pyramid Network(FPN) is used as Detection neck for PP-Yolo.
- Feature maps C 3 , C 4 , C 5 are input to the FPN module.

Head:

- Head of PP-Yolo is as like normal Yolo-v3.

- A 3 × 3 convolutional followed by an 1 × 1 convolutional layer is adopt to get the final predictions.

- The output channel of each final prediction is 3(K + 5), where K is number of classes.

Optimization Tricks:
Author proposed many optimization techniques for Yolo-v3. Optimized Yolo-v3 is named as PP-Yolo.

Larger Batch Size:
In PP-Yolo Batch size 192 is used instead of 64 in model training.

Exponential Moving Average (EMA):
On the time of training EMA of updated weights is calculated respectively, because sometimes EMA of weights give more accuracy than normal weights.

DropBlock:
DropBlock is one of regularization technique which is used to avoid overfitting.

In PP-Yolo Backbine doesn’t contain any Dropblock, Only FPN contains Dropblock.

IoU Loss:
IoU loss is used as Loss fuction in Bounding Box Regression.

IoU Aware:
IoU Aware Loss is added in IoU prediction branch. Which is very helpful to accurately calculate confidence score.

Grid Sensitive:
Output of Yolo for center of bounding box coordinates is related to grid of image. some calculation is required to convert for original image. Grid sensitive is one of the optimal method,

$$x = s \cdot (g_x + \alpha \cdot \sigma(p_x) - (\alpha - 1)/2),$$
$$y = s \cdot (g_y + \alpha \cdot \sigma(p_y) - (\alpha - 1)/2),$$

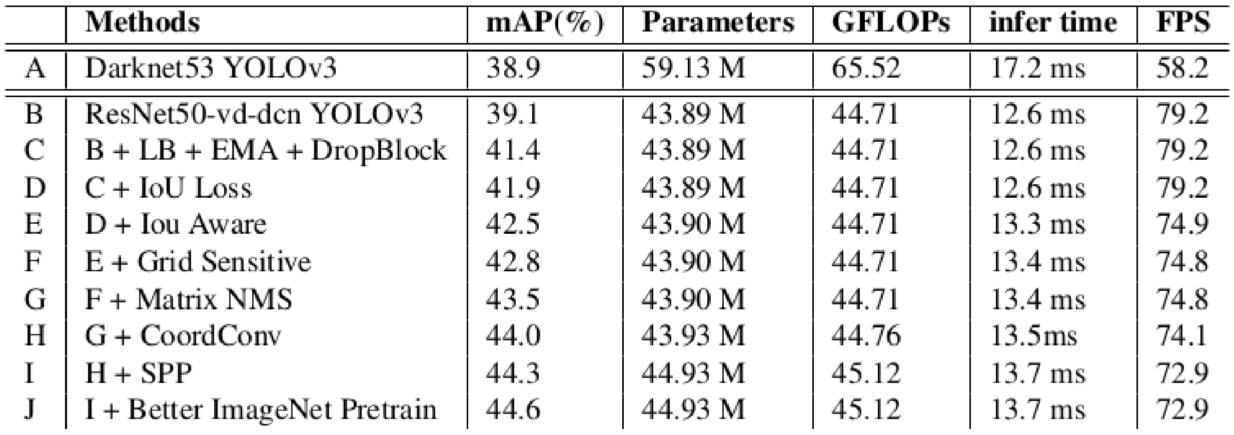
CoordConv :
CNN is sometimes lag on Cartesian Coordinate System operations, so CoordCov is a method to overcome that problem.

CoordConv will increase FLOPs, so it will used in only some layers which refers “Diamond” symbol in architecture.

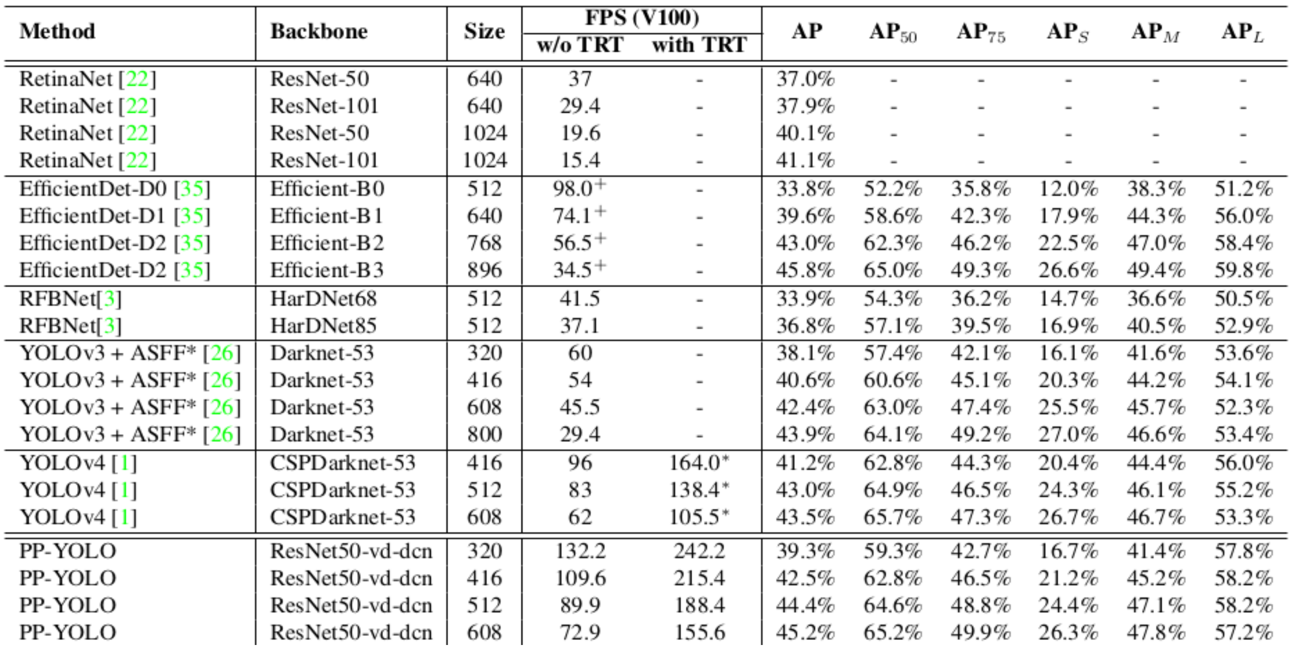
Spatial Pyramid Pooling(SPP):
Modified SPP is used in Yolo v4 for Low level feature extraction. So,same thing is used in PP-Yolo also.

“Star” symbol in architecture refers block of SPP used in PP-Yolo.

Performance:
Author tried many of above technique and gives as Ablation table:



	Methods	mAP(%)	Parameters	GFLOPs	infer time	FPS
A	Darknet53 YOLOv3	38.9	59.13 M	65.52	17.2 ms	58.2
B	ResNet50-vd-dcn YOLOv3	39.1	43.89 M	44.71	12.6 ms	79.2
C	B + LB + EMA + DropBlock	41.4	43.89 M	44.71	12.6 ms	79.2
D	C + IoU Loss	41.9	43.89 M	44.71	12.6 ms	79.2
E	D + Iou Aware	42.5	43.90 M	44.71	13.3 ms	74.9
F	E + Grid Sensitive	42.8	43.90 M	44.71	13.4 ms	74.8
G	F + Matrix NMS	43.5	43.90 M	44.71	13.4 ms	74.8
H	G + CoordConv	44.0	43.93 M	44.76	13.5ms	74.1
I	H + SPP	44.3	44.93 M	45.12	13.7 ms	72.9
J	I + Better ImageNet Pretrain	44.6	44.93 M	45.12	13.7 ms	72.9



Method	Backbone	Size	FPS (V100)		AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
			w/o TRT	with TRT						
RetinaNet [22]	ResNet-50	640	37	-	37.0%	-	-	-	-	-
RetinaNet [22]	ResNet-101	640	29.4	-	37.9%	-	-	-	-	-
RetinaNet [22]	ResNet-50	1024	19.6	-	40.1%	-	-	-	-	-
RetinaNet [22]	ResNet-101	1024	15.4	-	41.1%	-	-	-	-	-
EfficientDet-D0 [35]	Efficient-B0	512	98.0^+	-	33.8%	52.2%	35.8%	12.0%	38.3%	51.2%
EfficientDet-D1 [35]	Efficient-B1	640	74.1^+	-	39.6%	58.6%	42.3%	17.9%	44.3%	56.0%
EfficientDet-D2 [35]	Efficient-B2	768	56.5^+	-	43.0%	62.3%	46.2%	22.5%	47.0%	58.4%
EfficientDet-D2 [35]	Efficient-B3	896	34.5^+	-	45.8%	65.0%	49.3%	26.6%	49.4%	59.8%
RFBNet[3]	HarDNet68	512	41.5	-	33.9%	54.3%	36.2%	14.7%	36.6%	50.5%
RFBNet[3]	HarDNet85	512	37.1	-	36.8%	57.1%	39.5%	16.9%	40.5%	52.9%
YOLOv3 + ASFF* [26]	Darknet-53	320	60	-	38.1%	57.4%	42.1%	16.1%	41.6%	53.6%
YOLOv3 + ASFF* [26]	Darknet-53	416	54	-	40.6%	60.6%	45.1%	20.3%	44.2%	54.1%
YOLOv3 + ASFF* [26]	Darknet-53	608	45.5	-	42.4%	63.0%	47.4%	25.5%	45.7%	52.3%
YOLOv3 + ASFF* [26]	Darknet-53	800	29.4	-	43.9%	64.1%	49.2%	27.0%	46.6%	53.4%
YOLOv4 [1]	CSPDarknet-53	416	96	164.0^*	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4 [1]	CSPDarknet-53	512	83	138.4^*	43.0%	64.9%	46.5%	24.3%	46.1%	55.2%
YOLOv4 [1]	CSPDarknet-53	608	62	105.5^*	43.5%	65.7%	47.3%	26.7%	46.7%	53.3%
PP-YOLO	ResNet50-vd-dcn	320	132.2	242.2	39.3%	59.3%	42.7%	16.7%	41.4%	57.8%
PP-YOLO	ResNet50-vd-dcn	416	109.6	215.4	42.5%	62.8%	46.5%	21.2%	45.2%	58.2%
PP-YOLO	ResNet50-vd-dcn	512	89.9	188.4	44.4%	64.6%	48.8%	24.4%	47.1%	58.2%
PP-YOLO	ResNet50-vd-dcn	608	72.9	155.6	45.2%	65.2%	49.9%	26.3%	47.8%	57.2%

Above fig shows comparison of PP-Yolo with various architectures.

PP-Yolo achieves mAP of 45.2% and FPS of 72.9. Which is great performance against Yolo v3 and v4.