SSL | Smart Systems Lab.
YONSEI UNIVERSITY, SEOUL, KOREA

YOLOv4 : Optimal Speed and Accuracy of Object Detection

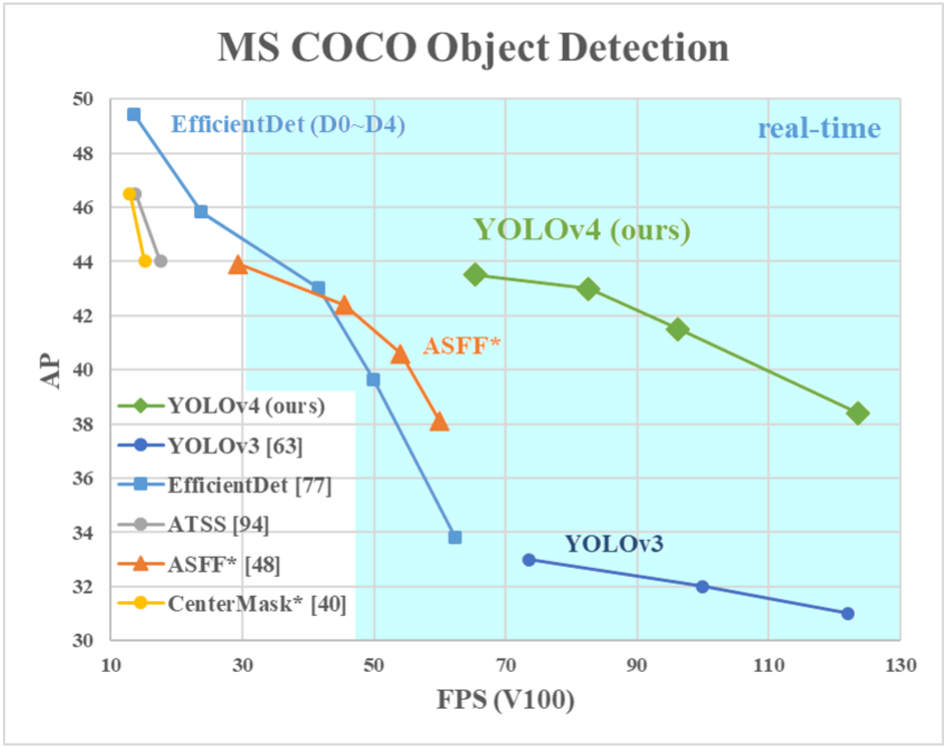
Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao
2020. 04. 23.

석사과정 1학기 박희성

목차

1.Yolo v4 Introduction

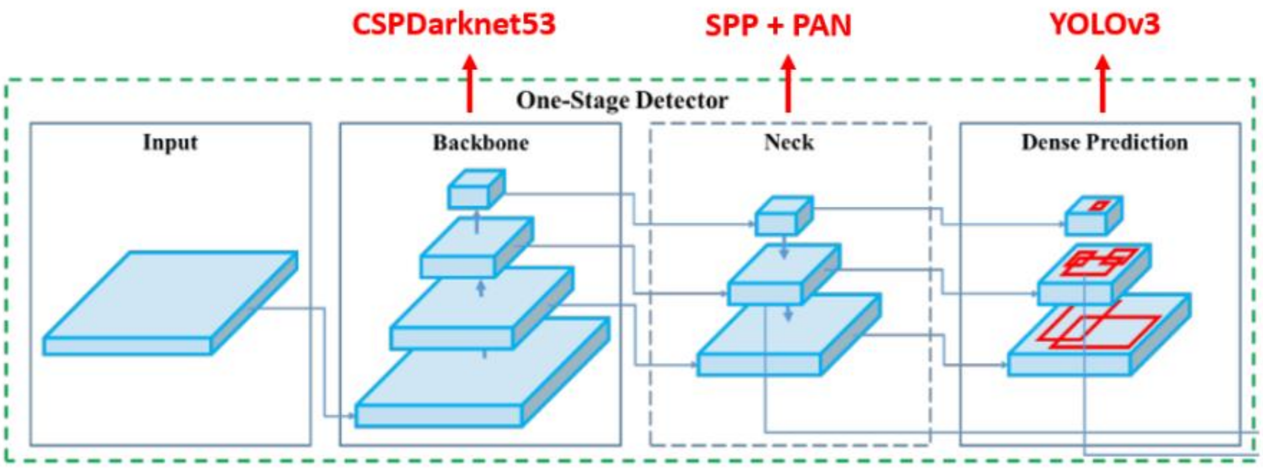
Smart Systems Lab.
YONSEI UNIVERSITY, SEOUL, KOREA



1. 일반적인 학습환경(1개의 GPU)에서 높은 정확도와 빠른 Object Detector를 설계
2. 학습 과정에서 최신 BoF, BoS 기법이 성능에 미치는 영향을 증명
3. CBN, PAN, SAM을 포함한 기법을 활용하여 single GPU training에 효과적

1.Yolo v4 Model Introduction

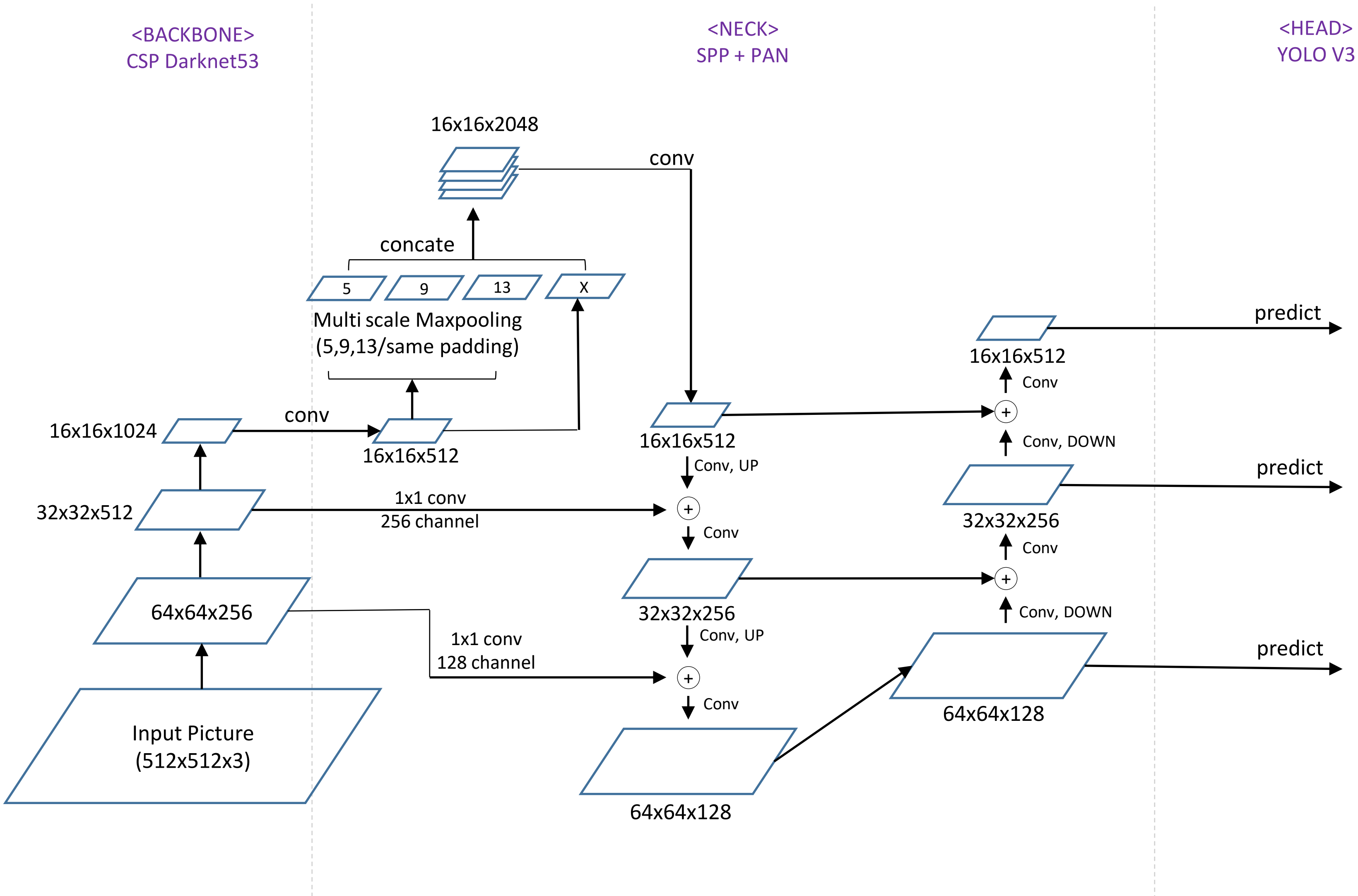
YOLOv4 = YOLOv3 + CSPDarknet53 + SPP + PAN + BoF + BoS



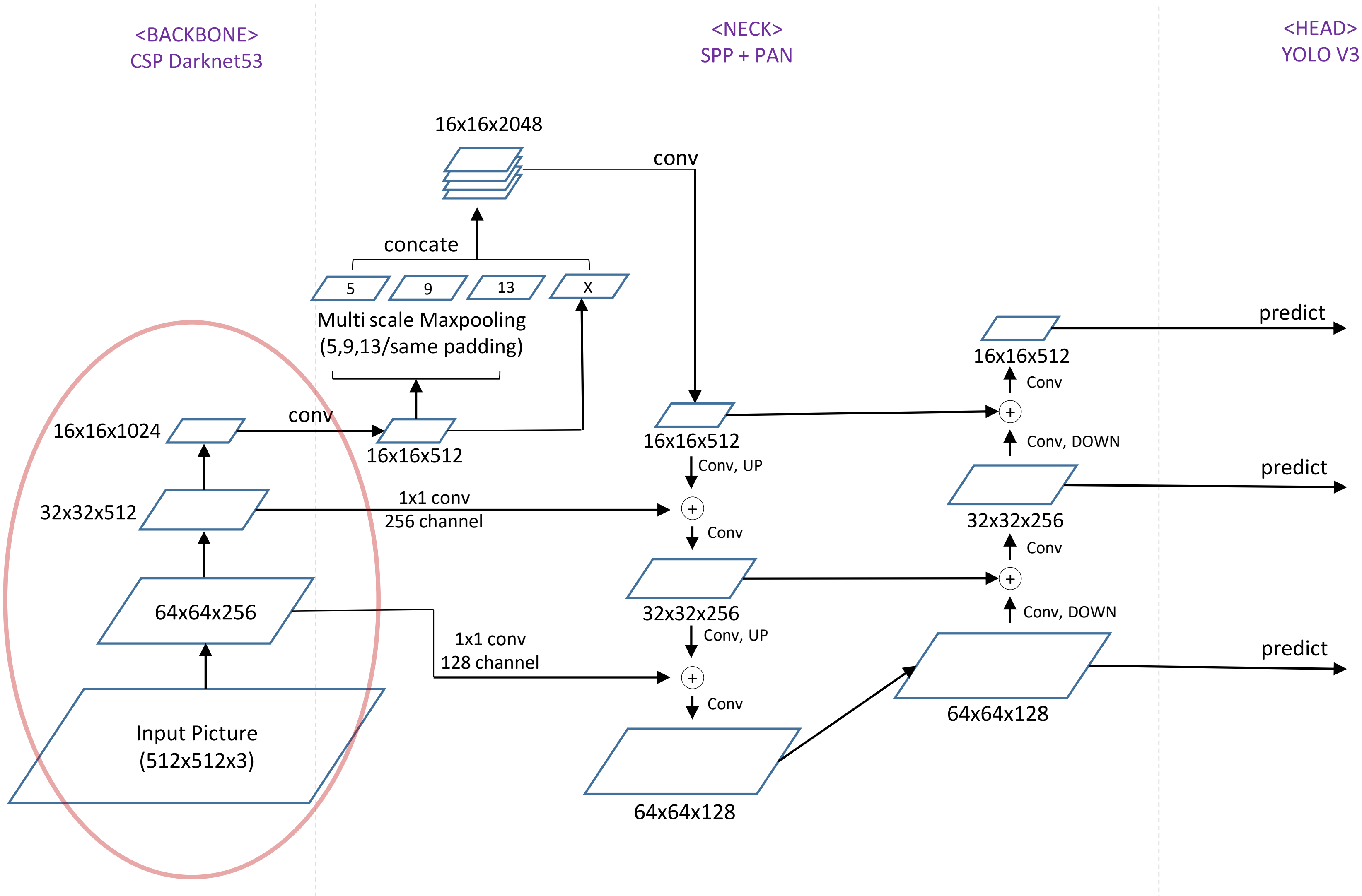
Input Image : 512 X 512 X 3
Backbone : Input Image를 Feature map으로 변형시켜주는 부분
Neck : Backbone에서 추출한 Feature map을 재구성하는 부분
Head(Dense Prediction) : Object detection 수행 (Yolo - One stage detection)

BoF(Bag of Freebies) : 학습에 영향을 주는 부분(train cost) 전처리, loss
BoS(Bag of Specials) : 모델의 Forward pass에 영향을 주는 부분(inference cost) 연산량 structure

1.Yolo v4 Model Introduction

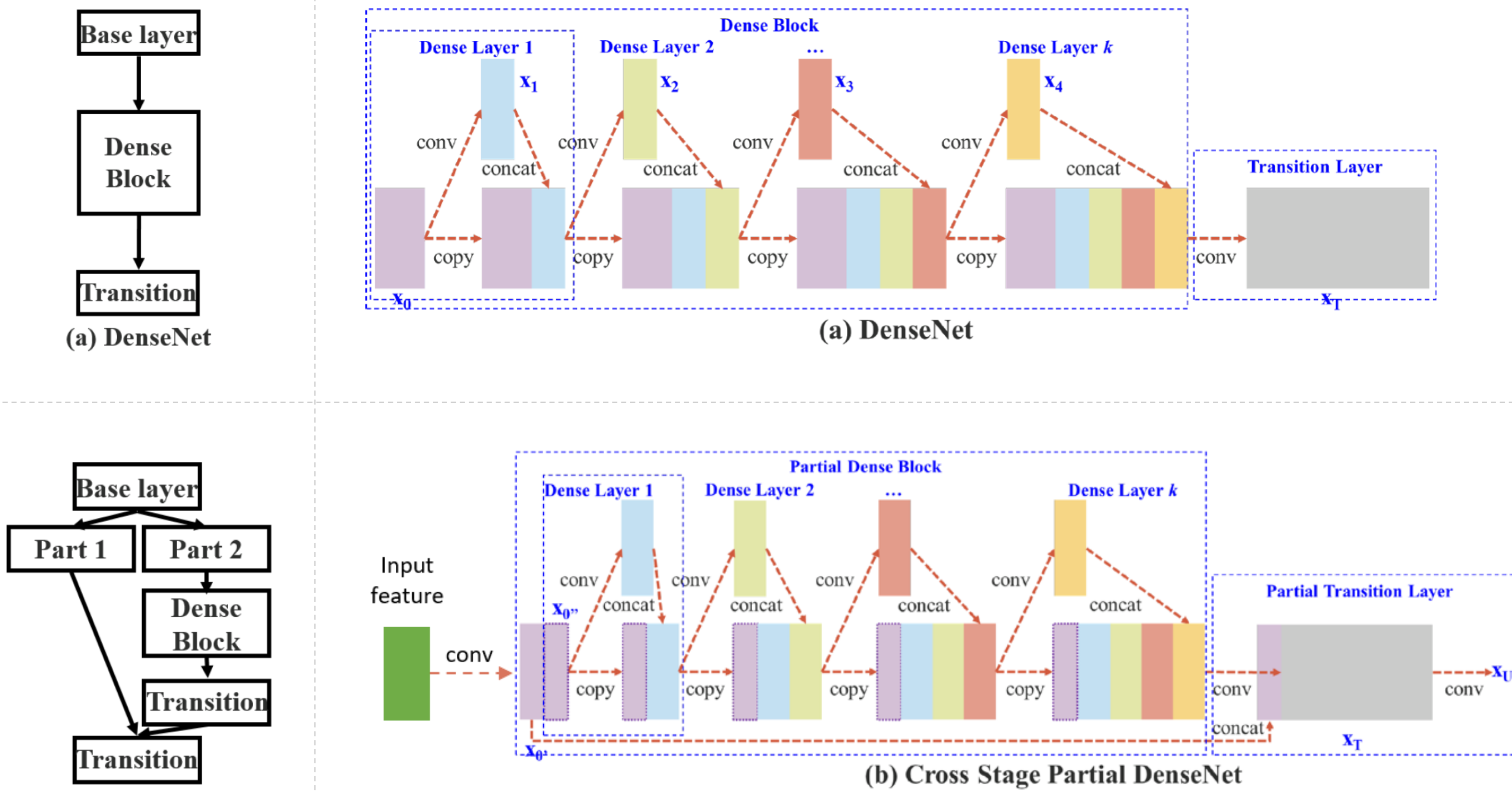


2.1 Backbone [CSP Darknet53]



2.1 Backbone [CSP Darknet53]

1) DenseNet과 CSP DenseNet의 차이



(a) DenseNet

(b) CSPDenseNet

(b) Cross Stage Partial DenseNet

2) CSP Darknet 53의 구조

	Type	Filters	Size	Output	
	Conv	32	3x3	512x512x32	
	Conv	64	3x3	256x256x64	
1x	Conv	32	1x1		concate
1x	Conv	64	3x3		
1x	Residual			256x256x64	
	Conv	128	3x3	128x128x128	
2x	Conv	64	1x1		concate
2x	Conv	128	3x3		
2x	Residual			128x128x128	
	Conv	256	3x3	64x64x256	
8x	Conv	128	1x1		concate
8x	Conv	256	3x3		
8x	Residual			**64x64x256**	
	Conv	512	3x3	32x32x512	
8x	Conv	256	1x1		concate
8x	Conv	512	3x3		
8x	Residual			**32x32x512**	
	Conv	1024	3x3	16x16x1024	
4x	Conv	512	1x1		concate
4x	Conv	1024	3x3		
4x	Residual			**16x16x1024**	

[Use of CSP]

반복되는 Convolutional, Residual Block
이전의 output을 이후에 concat 해줌

1) Strengthening learning ability of a CNN

2) Removing computational bottlenecks

3) Reducing memory costs

[Backbone Result]

Feature Map을 만드는 Backbone
에서는
64x64x256
32x32x512
16x16x1024
3가지 output이 Neck에 전달

2.1 Backbone [CSP Darknet53]

SSL Smart Systems Lab. YONSEI UNIVERSITY, SEOUL, KOREA

4) CSP Darknet 53의 효과

Table 1: Parameters of neural networks for image classification.

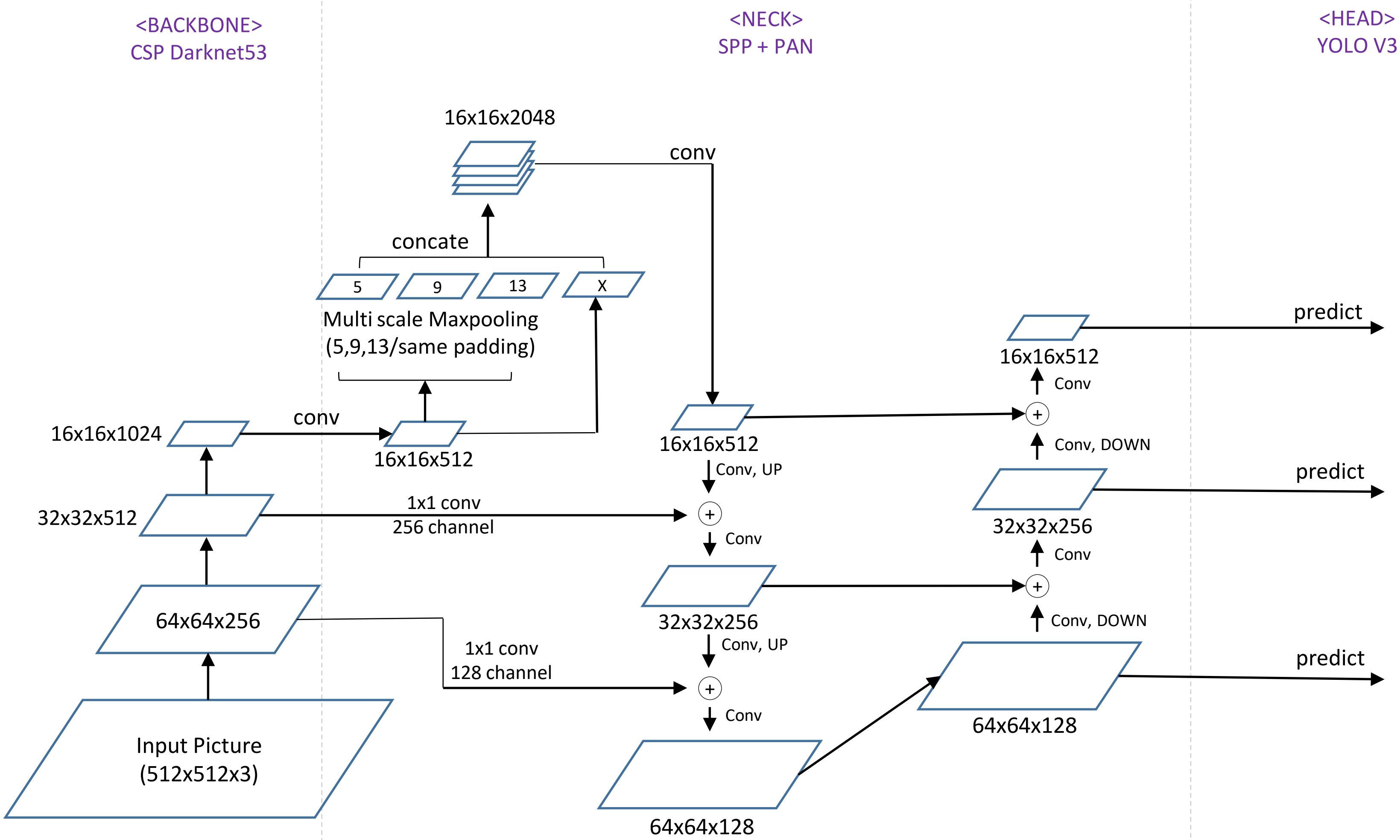
Backbone model	Input network resolution	Receptive field size	Parameters	Average size of layer output (WxHxC)	BFLOPs (512x512 network resolution)	FPS (GPU RTX 2070)
CSPResNext50	512x512	425x425	20.6 M	**1058 K**	31 (15.5 FMA)	62
CSPDarknet53	512x512	725x725	**27.6 M**	950 K	52 (26.0 FMA)	**66**
EfficientNet-B3 (ours)	512x512	**1311x1311**	12.0 M	668 K	11 (5.5 FMA)	26

1. **Higher input network size** (resolution)
2. **More layers** : for a higher receptive field
3. **More parameters** : for greater capacity of a model to detect multiple objects of different sizes

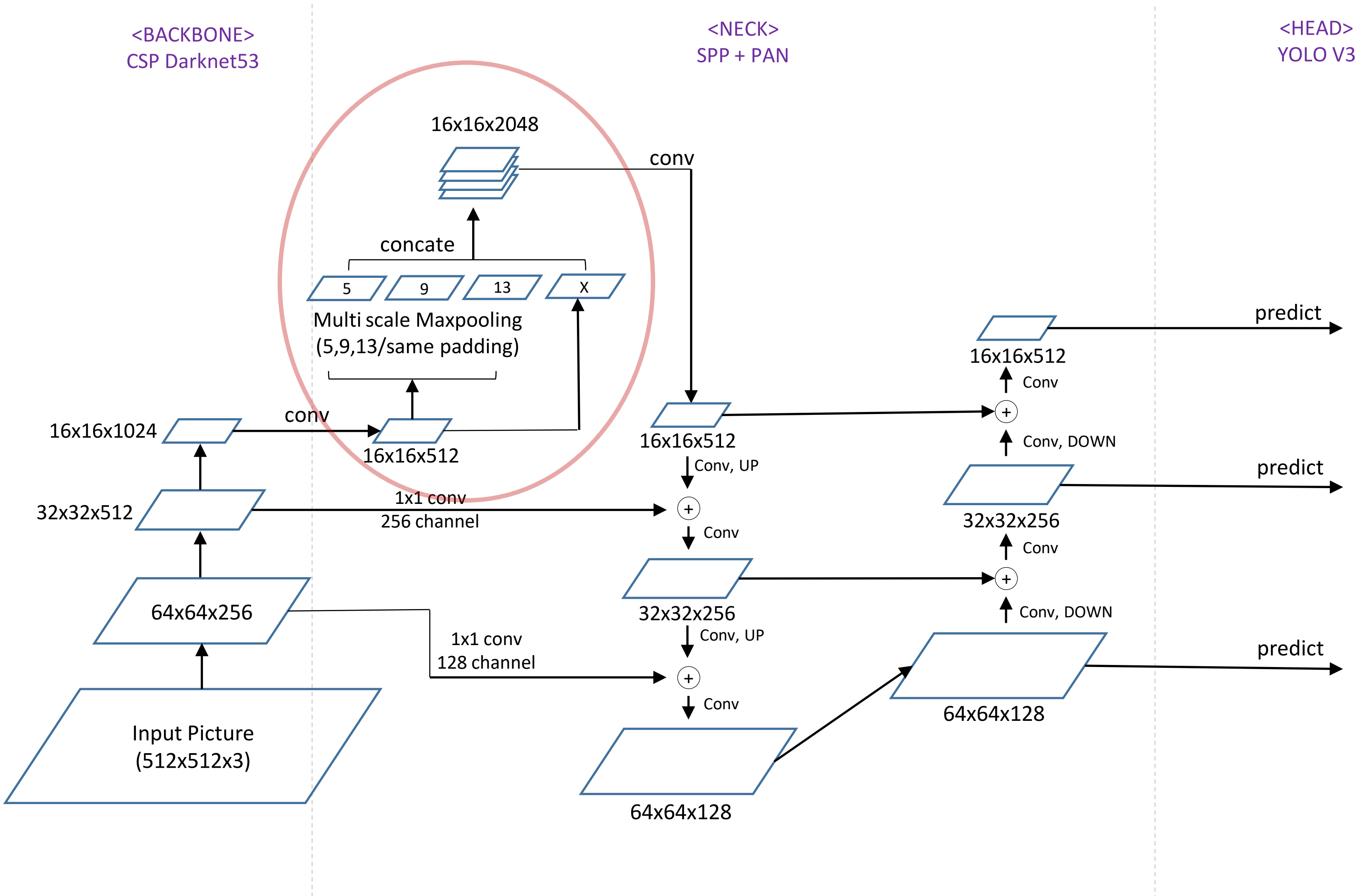
"Cross Stage Partial DenseNet"을 적용한 CSP Darknet53을 Backbone으로 활용

2.2 Neck [SPP + PAN]

SSL Smart Systems Lab.
YONSEI UNIVERSITY, SEOUL, KOREA



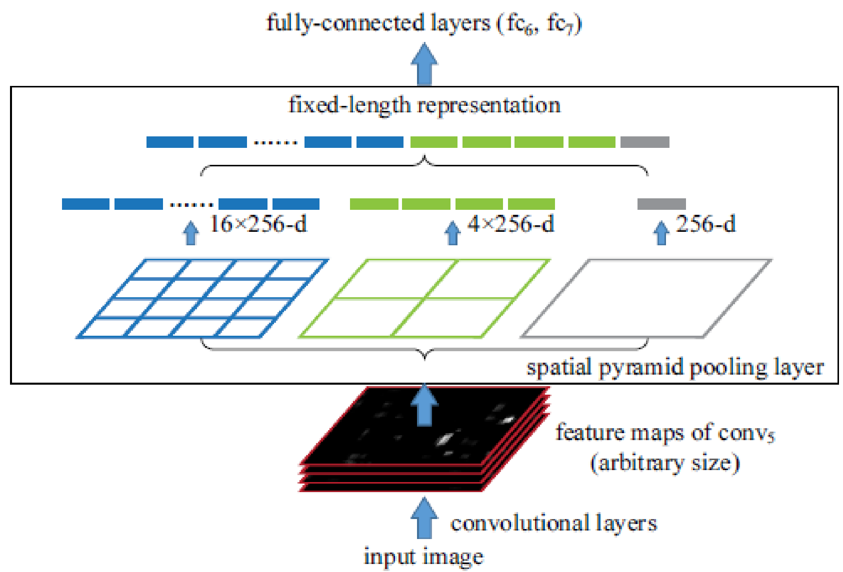
2.2 Neck [SPP + PAN]



2.2 Neck [SPP + PAN]

Smart Systems Lab. YONSEI UNIVERSITY, SEOUL, KOREA

1) 기존 Spatial Pyramid Polling Layer의 concept



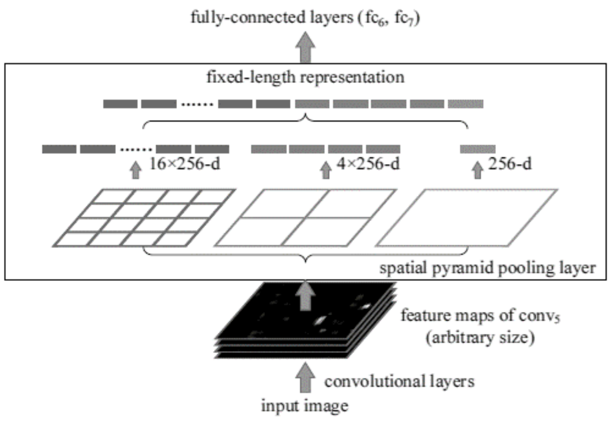
기존의 Spatial Pyramid Polling 방식은

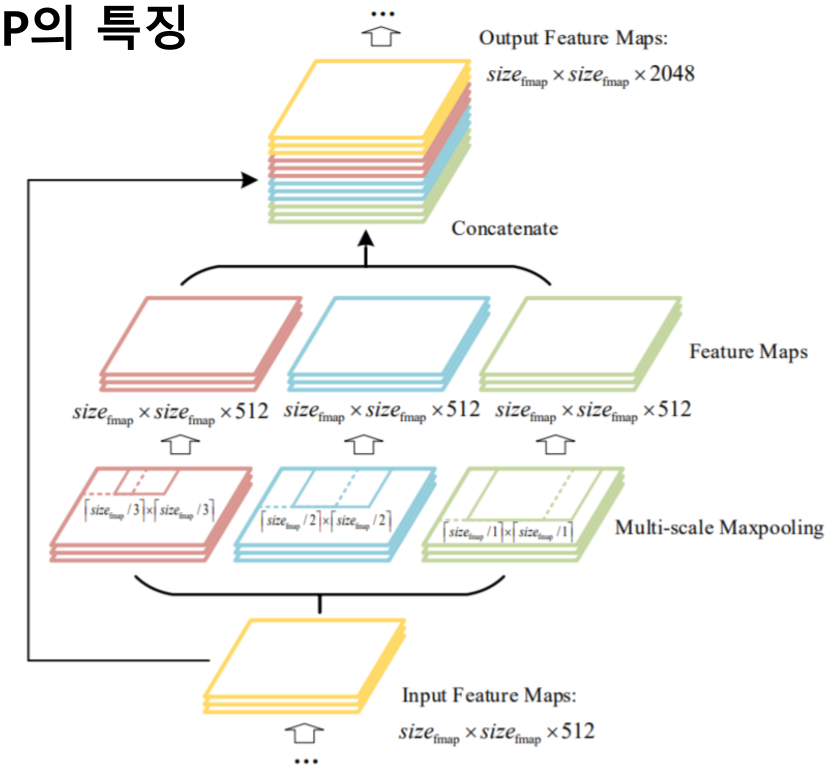
4x4, 2x2, 1x1로 input image를 나누어서

"Input Image에 상관없이, Fixed-Length를 가진 Output 생성"

SSL Smart Systems Lab. YONSEI UNIVERSITY, SEOUL, KOREA

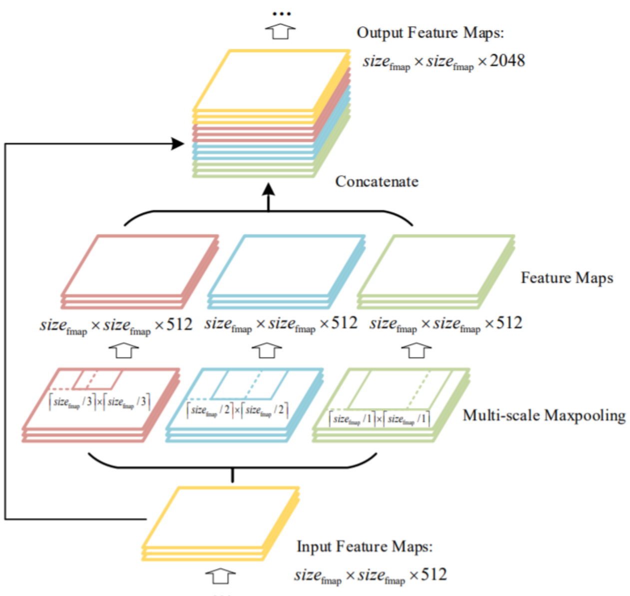
2) Yolo v4의 Neck 부분에 사용된 SPP의 특징

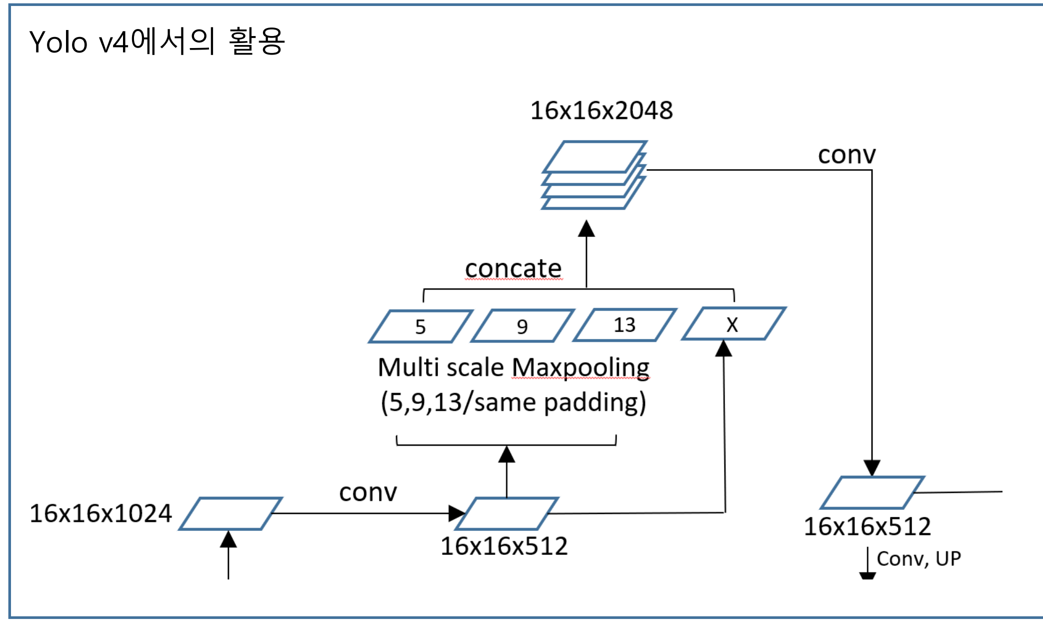




1. Input Feature의 크기를 미리 정하기 때문에 Same padding을 사용하여 pooling 이후 output의 크기를 맞추고 concate 진행
2. Neck 부분의 Receptive field를 증가시키는 효과 있었음
3. 모델의 정확도 증가와 inference time이 약간 감소

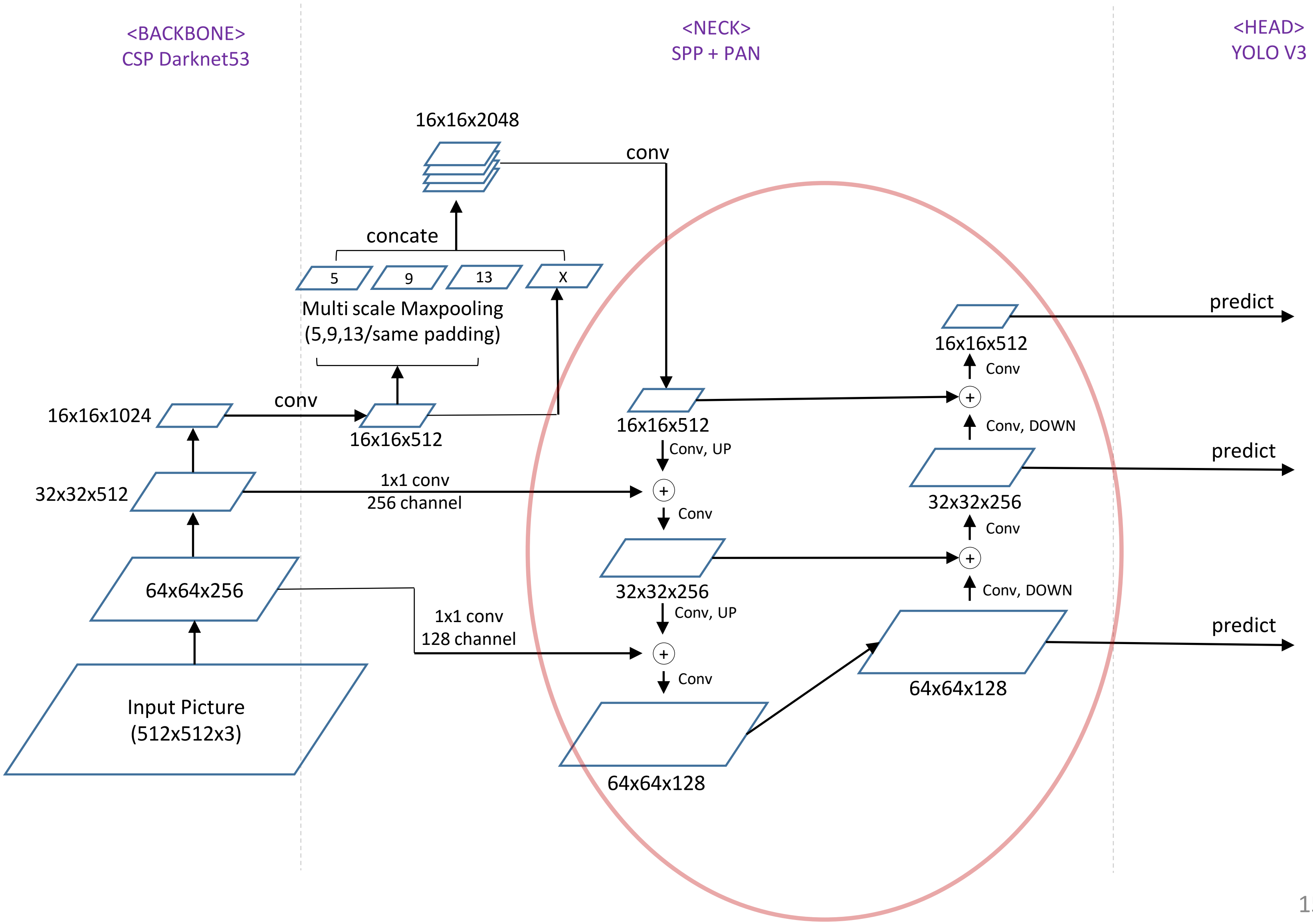
3) Yolo v4에서의 SPP 구조





2.2 Neck [SPP + PAN]

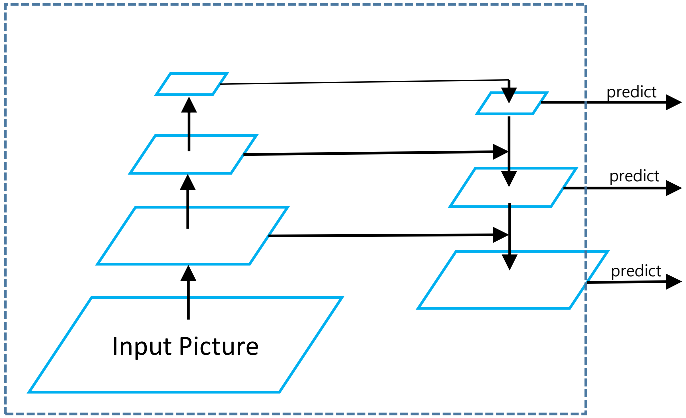
Smart Systems Lab. YONSEI UNIVERSITY, SEOUL, KOREA



2.2 Neck [SPP + PAN]

-Yolo v3 : Backbone + Neck

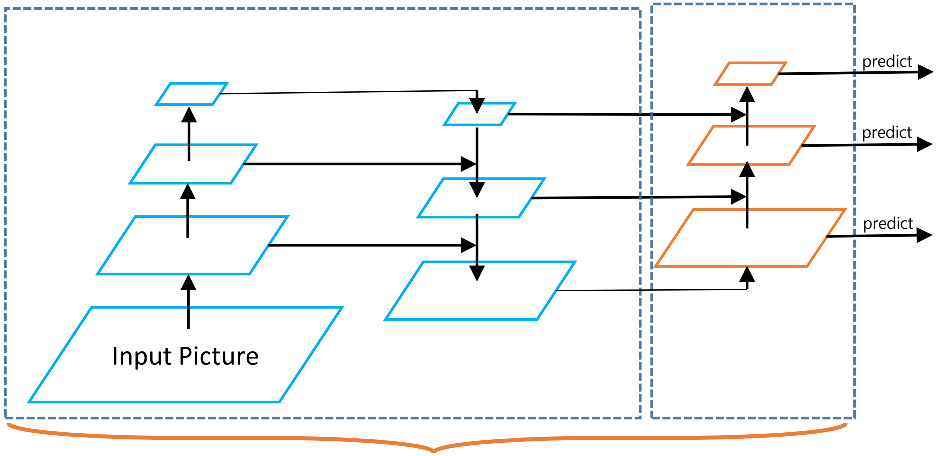
(a) Feature Pyramid Network Backbone



-Yolo v4 : Backbone + Neck

(a) Feature Pyramid Network Backbone

(b) Bottom-up path augmentation

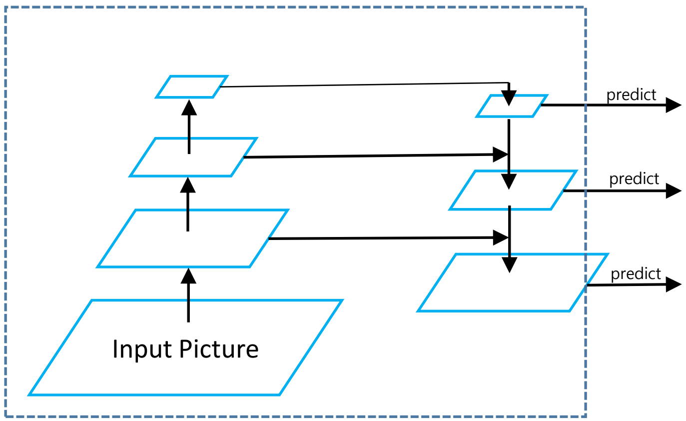


(a) + (b) = Path Aggregation Network

2.2 Neck [SPP + PAN]

-Yolo v3 : Backbone + Neck

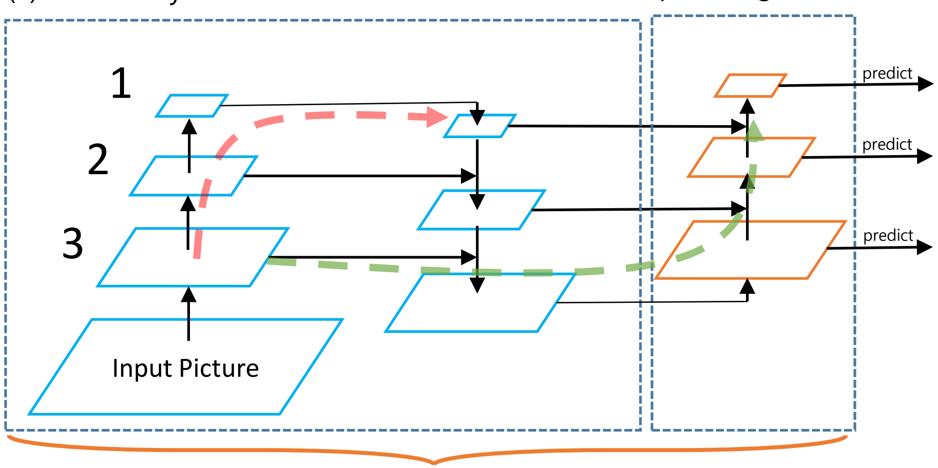
(a) Feature Pyramid Network Backbone



-Yolo v4 : Backbone + Neck

(a) Feature Pyramid Network Backbone

(b) Bottom-up path augmentation



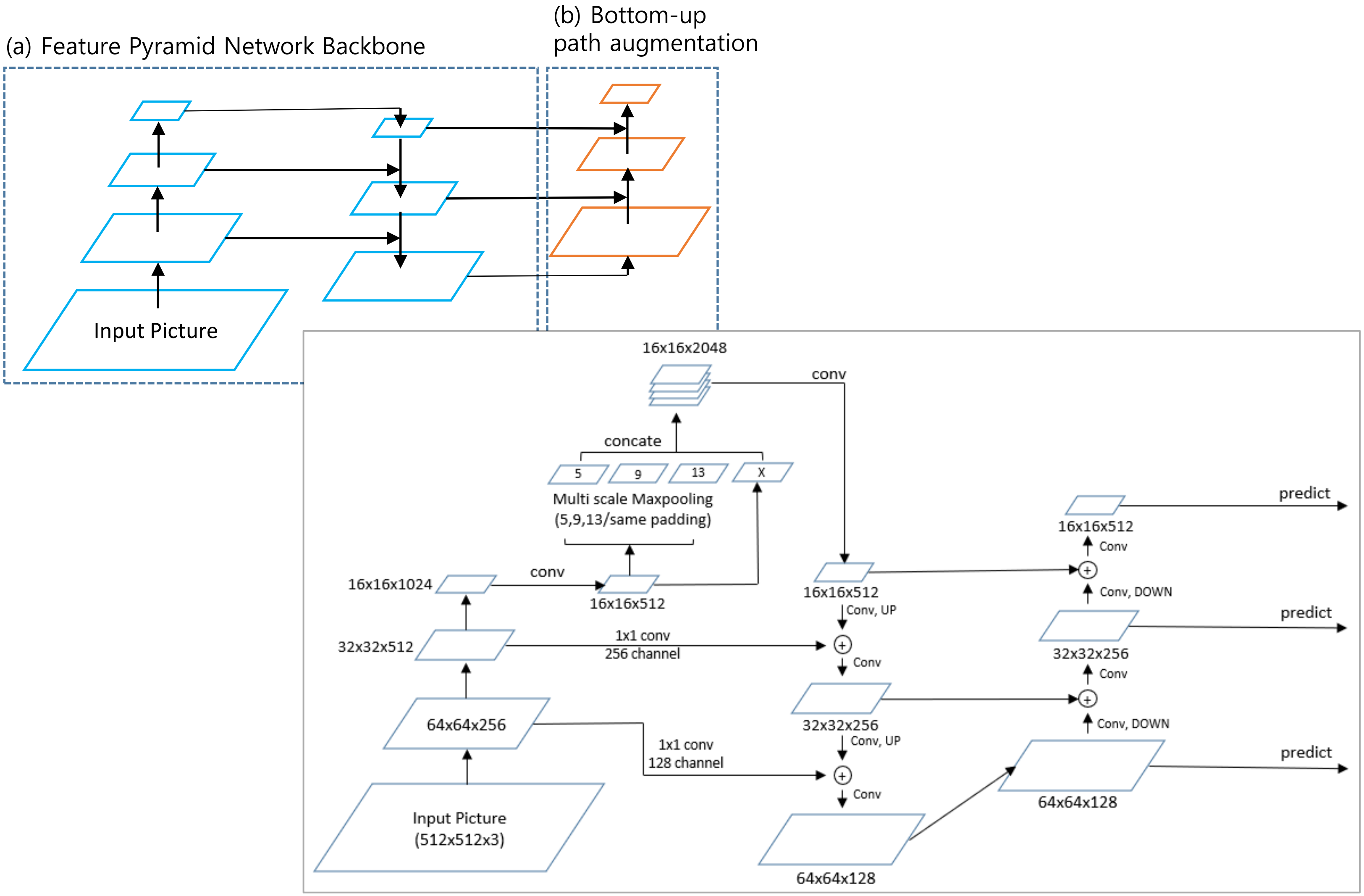
(a) + (b) = Path Aggregation Network

Prediction에 사용되는 작은 size의 feature map(16x16x512)에도

64x64x256에 있는 작은 object의 특징을 잘 포함시킬 수 있음

2.2 Neck [SPP + PAN]

-Yolo v4 : Backbone + Neck



2.2 Neck [SPP + PAN]

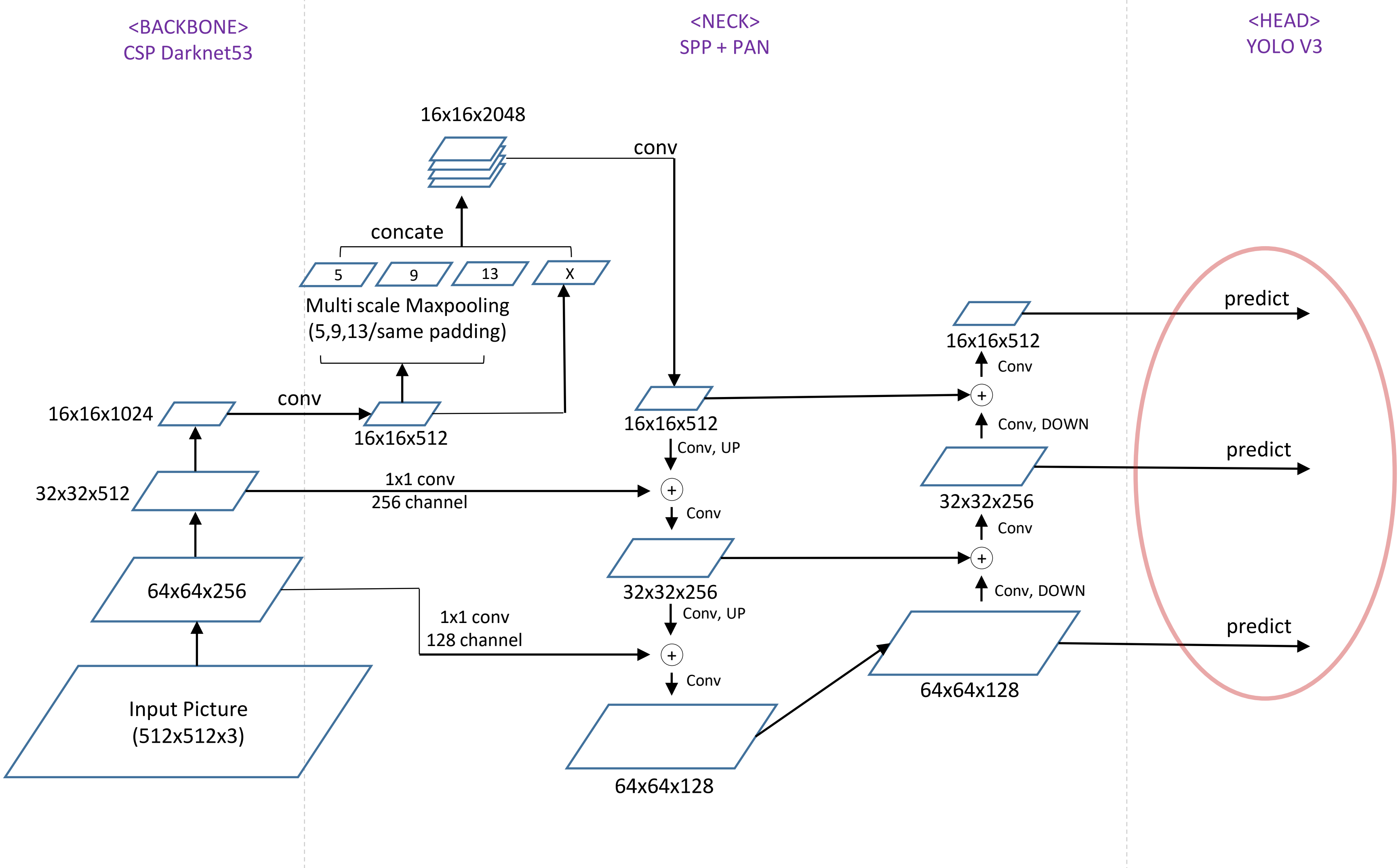
Table 5: Ablation Studies of Bag-of-Specials. (Size 512x512).

Model	AP	AP_{50}	AP_{75}
CSPResNeXt50-PANet-SPP	42.4%	64.4%	45.9%
CSPResNeXt50-PANet-SPP-RFB	41.8%	62.7%	45.1%
CSPResNeXt50-PANet-SPP-SAM	**42.7%**	**64.6%**	**46.3%**
CSPResNeXt50-PANet-SPP-SAM-G	41.6%	62.7%	45.0%
CSPResNeXt50-PANet-SPP-ASFF-RFB	41.1%	62.6%	44.4%

Table 6: Using different classifier pre-trained weightings for detector training (all other training parameters are similar in all models) .

Model (with optimal setting)	Size	AP	AP_{50}	AP_{75}
CSPResNeXt50-PANet-SPP	512x512	42.4	64.4	45.9
CSPResNeXt50-PANet-SPP (BoF-backbone)	512x512	42.3	64.3	45.7
CSPResNeXt50-PANet-SPP (BoF-backbone + Mish)	512x512	42.3	64.2	45.8
CSPDarknet53-PANet-SPP (BoF-backbone)	512x512	42.4	64.5	46.0
CSPDarknet53-PANet-SPP (BoF-backbone + Mish)	512x512	43.0	64.9	46.5

2.3 Head [Yolo v3]

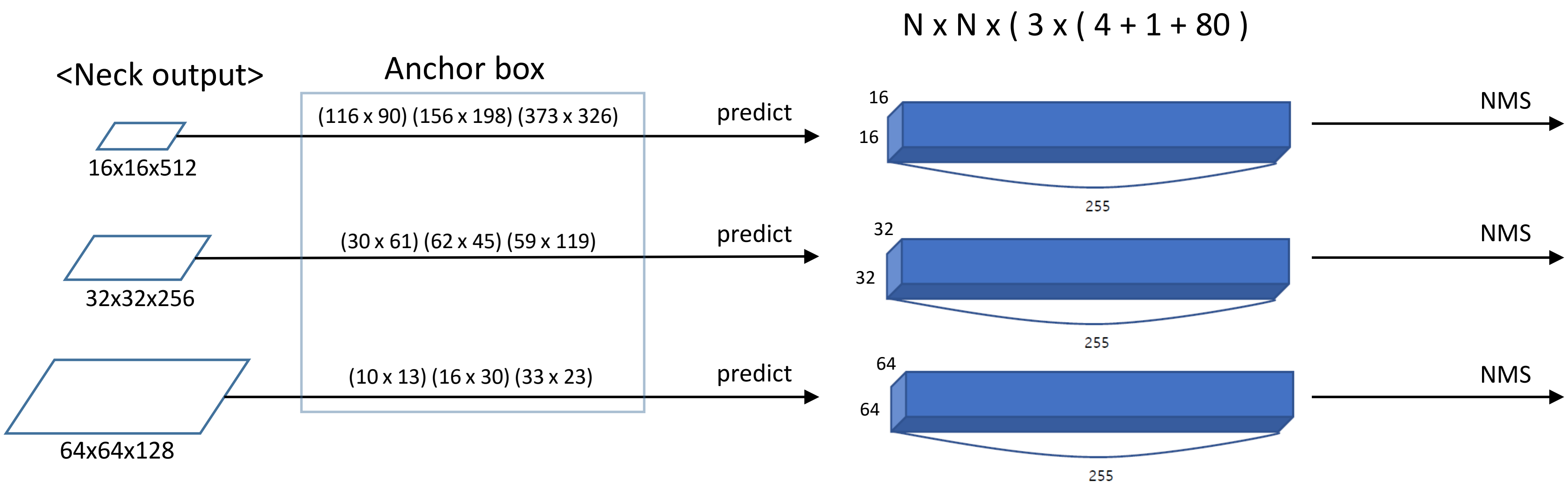


2.3 Head [Yolo v3]

<K-means clustering results>

(about size of ground truth)

(10 x 13) (16 x 30) (33 x 23) / (30 x 61) (62 x 45) (59 x 119) / (116 x 90) (156 x 198) (373 x 326)



SSL Smart Systems Lab. YONSEI UNIVERSITY, SEOUL, KOREA

YOLOv4 = YOLOv3 + CSPDarknet53 + SPP + PAN + $\boxed{\text{BoF}}$ + BoS

Bag of Freebies (pre-processing + training strategy)

Training Phase

- Call methods that only change the **training** strategy or only increase the **training cost** as "BoF"

Data Augmentation	Regularization	Loss Function
• Random erase	• DropOut	• MSE
• CutOut	• DropPath	• IoU
• MixUp	• Spatial DropOut	• GIoU **Generalized**
• CutMix	• DropBlock	• CIoU **Complete**
• Style transfer GAN		• DIoU **Distance**

[Bag of Freebies]

2.4 Selection of BoF and BoS

YOLOv4 = YOLOv3 + CSPDarknet53 + SPP + PAN + BoF + $\boxed{\text{BoS}}$

architecture related

Bag of Specials (plugin modules + post-processing)

Inference Phase

- Call methods that only increase the **inference cost** but can improve the accuracy as "BoS"

Enhancement of receptive field

- Spatial Pyramid Pooling
- ASPP (dilated conv)
- Receptive Field Block (RFB)

Feature Integration

- Skip-connection
- Feature Pyramid Network
- SFAM (Scale-wise Feature Aggregation Module)
- ASFF (adaptively spatial feature fusion)
- BiFPN

Activation function

- ReLU
- Leaky ReLU
- Parametric ReLU
- ReLU6
- Swish
- Mish

Attention Module

- Squeeze-and-Excitation (SE)
- Spatial Attention Module (SAM)

Normalization

- Batch Norm (BN)
- Cross-GPU Batch Norm (CGBN or SyncBN)
- Filter Response Normalization (FRN)
- Cross-Iteration Batch Norm (CBN)

Post Processing

- NMS
- Soft NMS
- DIoU NMS

2.4 Selection of BoF and BoS

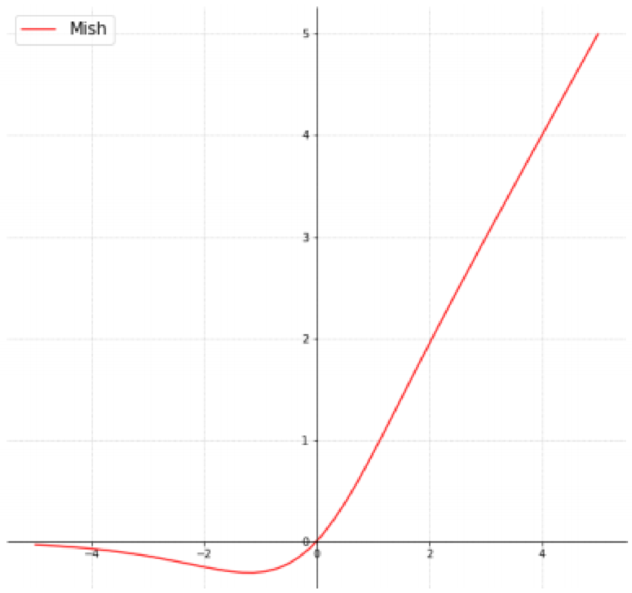
Smart Systems Lab.
YONSEI UNIVERSITY, SEOUL, KOREA

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU

2.4 Selection of BoF and BoS

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU

f(x) = x * tanh(softpuls(x))
// softplus(x) = ln(1 + e^x)



2.4 Selection of BoF and BoS

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU

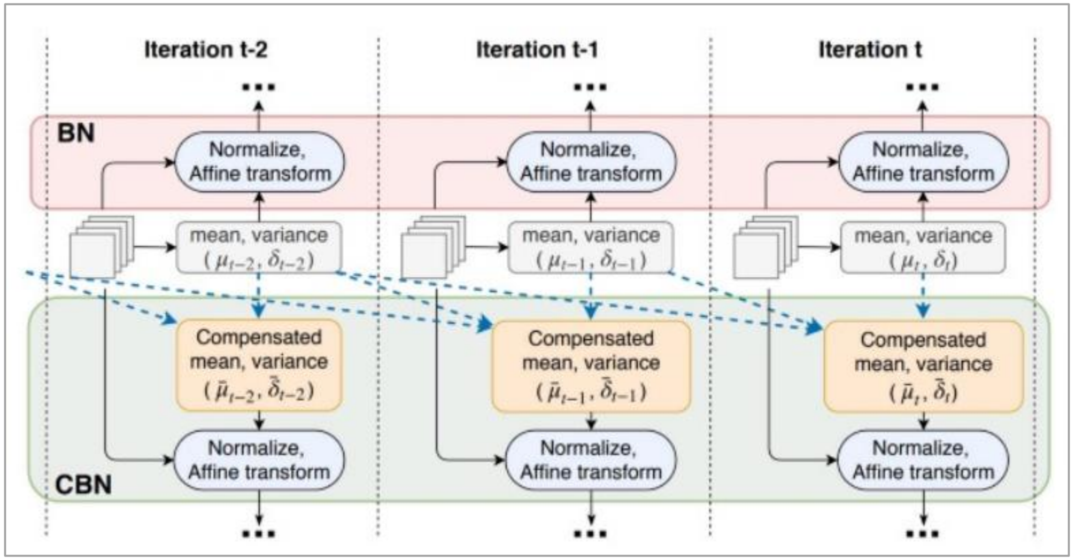
Dog 1.0

Dog 0.5
Cat 0.5

Dog 0.6
Cat 0.4

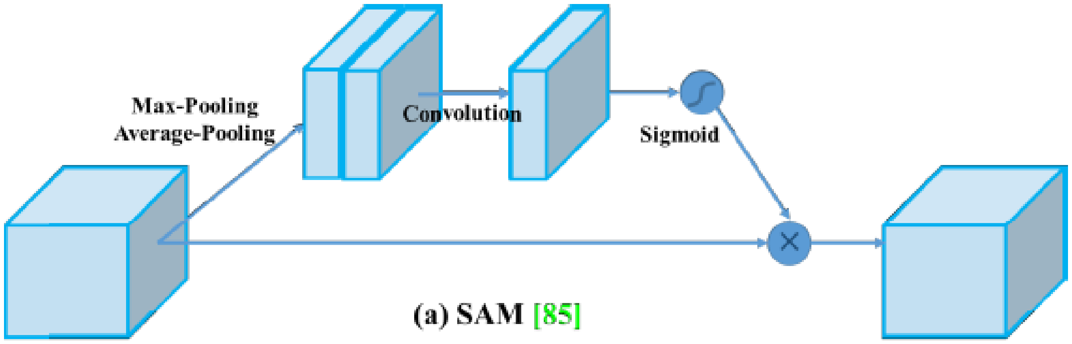
2.4 Selection of BoF and BoS

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU



2.4 Selection of BoF and BoS

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU



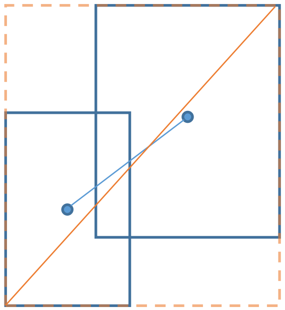
(a) SAM [85]

2.4 Selection of BoF and BoS

SSL Smart Systems Lab. YONSEI UNIVERSITY, SEOUL, KOREA

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU

$$\boxed{\mathcal{R}_{DIoU} = \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}}$$



ρ : Euclidean distance

b, b^{gt} : Central points of Bounding box

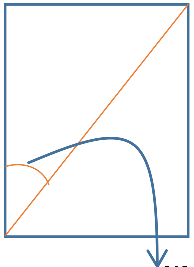
c : Diagonal length of the smallest enclosing box covering the two boxes

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}.$$

2.4 Selection of BoF and BoS

- **Activations** : Mish
- **Data augmentation** : CutOut, MixUp, CutMix
- **Normalization of the network activations by their mean and variance** : CmBN
- **Attention Module** : Spatial Attention Module
- **Post processing** : DIoU NMS
- **Bounding box regression loss** : CIoU

$$\boxed{\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v.}$$



$$\alpha = \frac{v}{(1 - IoU) + v}$$

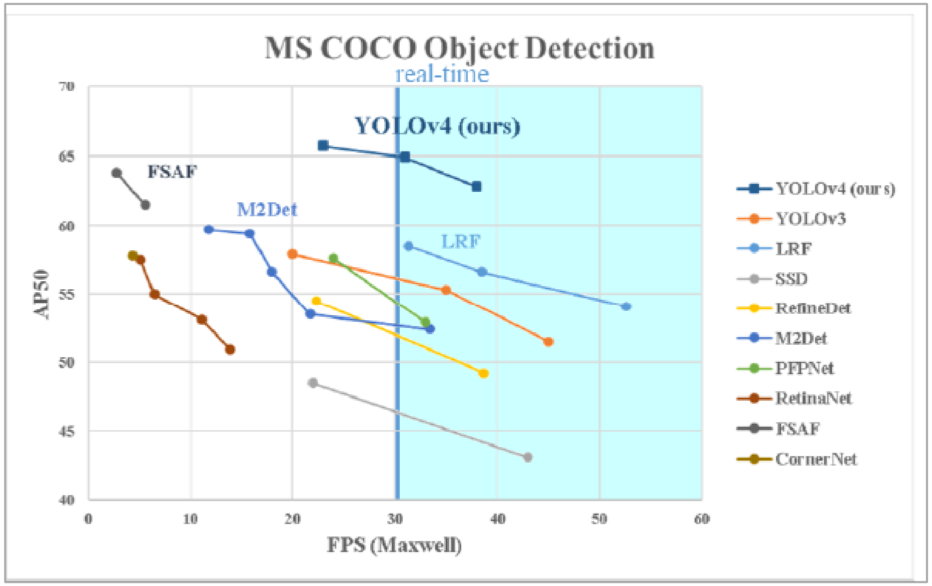
$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2}.$$

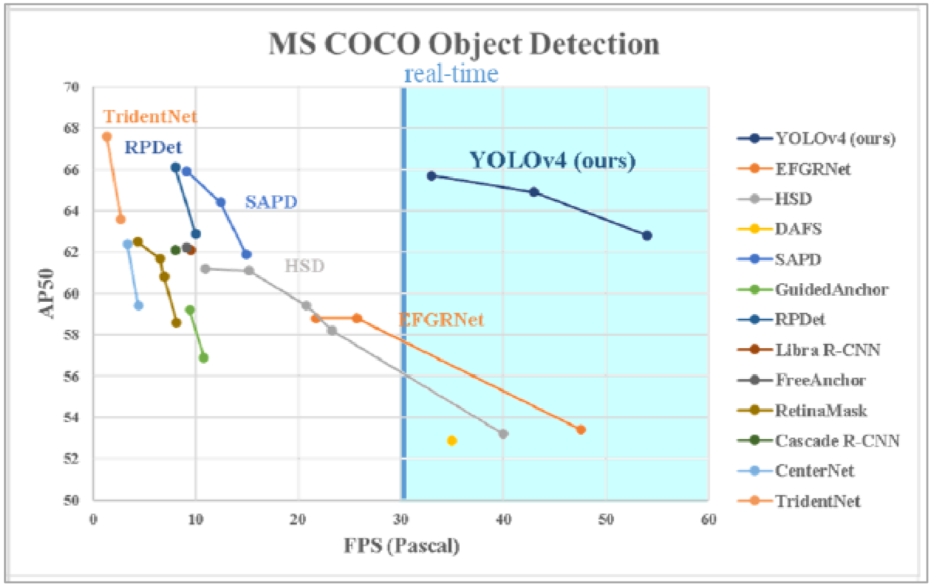
$$v = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2.$$

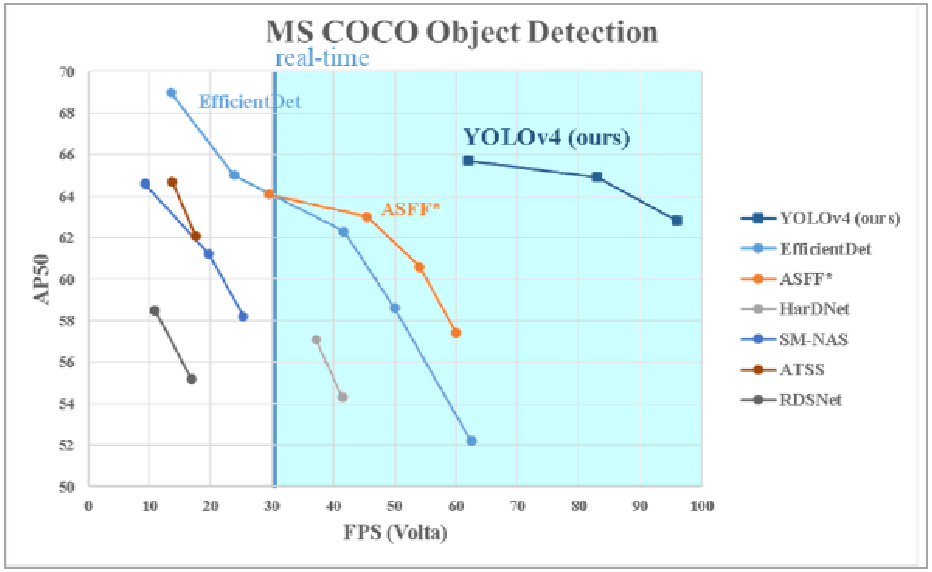
Table 10: Comparison of the speed and accuracy of different object detectors on the MS COCO dataset (test-dev 2017). (Real-time detectors with FPS 30 or higher are highlighted here. We compare the results with batch=1 without using tensorRT.)

Method	Backbone	Size	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv4: Optimal Speed and Accuracy of Object Detection									
YOLOv4	CSPDarknet-53	416	96 (V)	41.2%	62.8%	44.3%	20.4%	44.4%	56.0%
YOLOv4	CSPDarknet-53	512	83 (V)	43.0%	64.9%	46.5%	24.3%	46.1%	55.2%
YOLOv4	CSPDarknet-53	608	62 (V)	**43.5%**	**65.7%**	47.3%	**26.7%**	46.7%	53.3%
EfficientDet: Scalable and Efficient Object Detection [77]									
EfficientDet-D0	Efficient-B0	512	62.5 (V)	33.8%	52.2%	35.8%	12.0%	38.3%	51.2%
EfficientDet-D1	Efficient-B1	640	50.0 (V)	39.6%	58.6%	42.3%	17.9%	44.3%	56.0%
EfficientDet-D2	Efficient-B2	768	41.7 (V)	43.0%	62.3%	46.2%	22.5%	**47.0%**	**58.4%**
EfficientDet-D3	Efficient-B3	896	23.8 (V)	45.8%	65.0%	49.3%	26.6%	49.4%	59.8%
Learning Spatial Fusion for Single-Shot Object Detection [48]									
YOLOv3 + ASFF*	Darknet-53	320	60 (V)	38.1%	57.4%	42.1%	16.1%	41.6%	53.6%
YOLOv3 + ASFF*	Darknet-53	416	54 (V)	40.6%	60.6%	45.1%	20.3%	44.2%	54.1%
YOLOv3 + ASFF*	Darknet-53	608×	45.5 (V)	42.4%	63.0%	**47.4%**	25.5%	45.7%	52.3%
YOLOv3 + ASFF*	Darknet-53	800×	29.4 (V)	43.9%	64.1%	49.2%	27.0%	46.6%	53.4%
HarDNet: A Low Memory Traffic Network [4]									
RFBNet	HarDNet68	512	41.5 (V)	33.9%	54.3%	36.2%	14.7%	36.6%	50.5%
RFBNet	HarDNet85	512	37.1 (V)	36.8%	57.1%	39.5%	16.9%	40.5%	52.9%
Focal Loss for Dense Object Detection [45]									
RetinaNet	ResNet-50	640	37 (V)	37.0%	-	-	-	-	-
RetinaNet	ResNet-101	640	29.4 (V)	37.9%	-	-	-	-	-
RetinaNet	ResNet-50	1024	19.6 (V)	40.1%	-	-	-	-	-
RetinaNet	ResNet-101	1024	15.4 (V)	41.1%	-	-	-	-	-

3. Conclusion







감사합니다!