인공지능 심화 과정

YOLO v1
YOLO v2
YOLO v3
YOLO v4

Introduction

- This is a TECH REPORT.
- "I managed to make some improvements to YOLO. But, honestly, nothing like super interesting, just a bunch of small changes that make it better."
- Better, Not Faster, Stronger(?)
- The last sentence of the document: In closing, do not @me. (Because I finally quit Twitter).

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Anchor box는 YOLOv2와 동일

Direct location prediction

$$x = (t_x * w_a) - x_a$$
$$y = (t_y * h_a) - y_a$$

anchor box는 bounding box regressor 계수(coef ficient)를 통해 위의 공식과 같이 bounding box 의 위치를 조정합니다. 하지만 tx,tytx,ty와 같은 계수는 제한된 범위가 없기 때문에 anchor box는 이미지 내의 임의의 지점에 위치할 수 있다는 문제가 있습니다. 이로 인해 최적화된 값을 찾기 까지 오랜 시간이 걸려 모델은 초기에 불안정하게 됩니다.

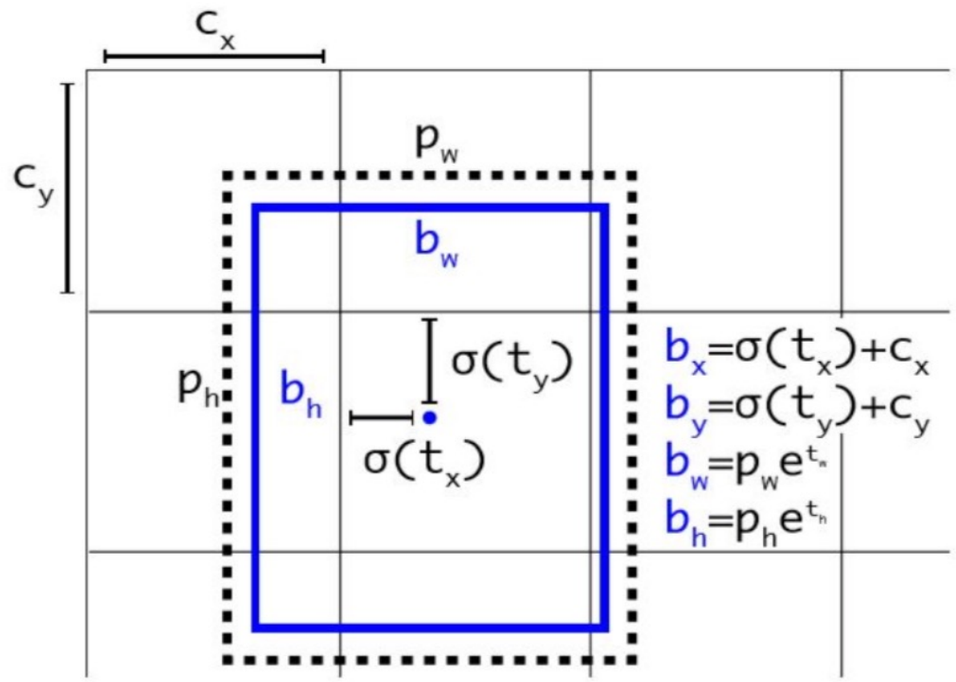


Figure 3: Bounding boxes with dimension priors and location prediction. We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function.

Anchor box는 YOLOv2와 동일

Direct location prediction

논문의 저자는 이러한 문제를 해결하기 위해 YOLO의 방식을 사용하여 grid cell에 상대적인 위치 좌표를 예측하는 방법을 선택했다. 이는 예측하는 bounding box의 좌표는 0~1 사이의 값을 가짐을 의미한다. 위의 그림에서 cx,cycx, cy는 grid cell의 좌상단 offset입니다. bounding box regression을 통해 얻은 txtx, tyty 값에 logistic regression 함수(σσ)를 적용하여 0~1 사이의 값을 가지도록 조정했다.

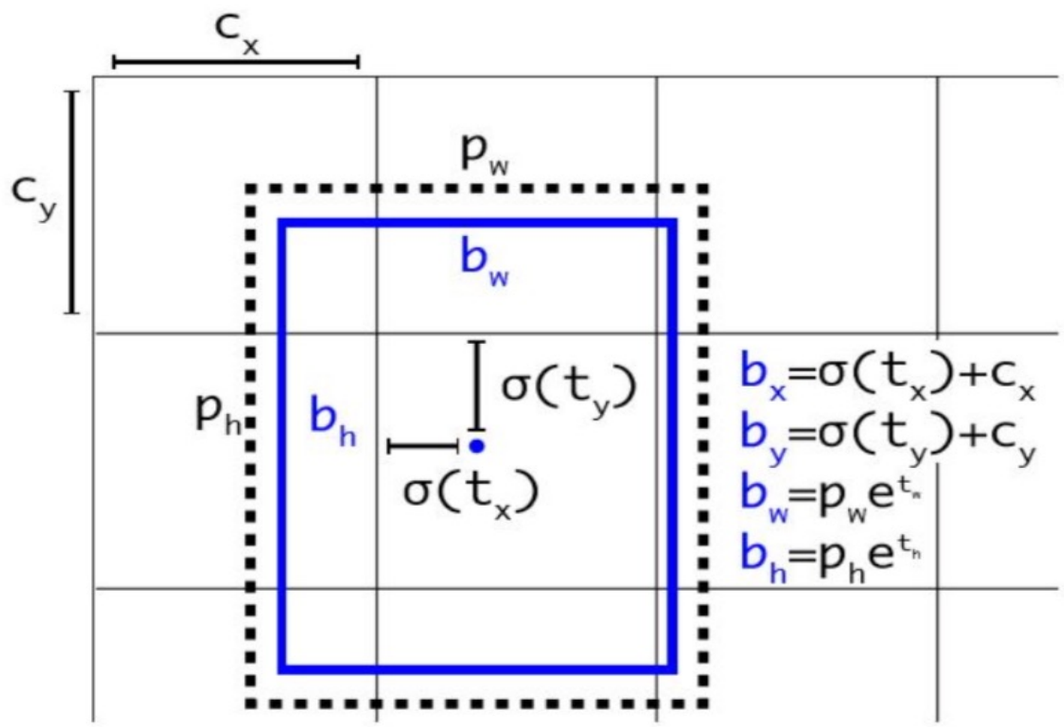


Figure 3: Bounding boxes with dimension priors and location prediction. We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function.

예측하는 위치의 범위가 정해짐으로써 네트워크는 안정적으로 학습을 진행하는 것이 가능해진다. Dimension clustering을 통해 최적의 prior를 선택하고, anchor box 중심부 좌표를 직접 예측함으로서 recall값이 5% 정도 향상된다고 한다.

Bounding Box Prediction

- YOLOv3 predicts an objectness score for each bounding box using logistic regression.

- This should be 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior.

- They use the threshold of 0.5. Unlike Faster R-CNN, YOLOv3 only assigns one bounding box prior for each ground truth object.

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Class Prediction

- YOLO v3 now performs multilabel classification for objects detected in images.
- Softmaxing classes rests on the assumption that classes are mutually exclusive.
- However, when we have classes like **Person** and **Women** in a dataset, then the above assumption fails. This is the reason why the authors of YOLO have refrained from softmaxing the classes. Instead, each class score is predicted using logistic regression and a threshold is used to predict multiple labels for an object.

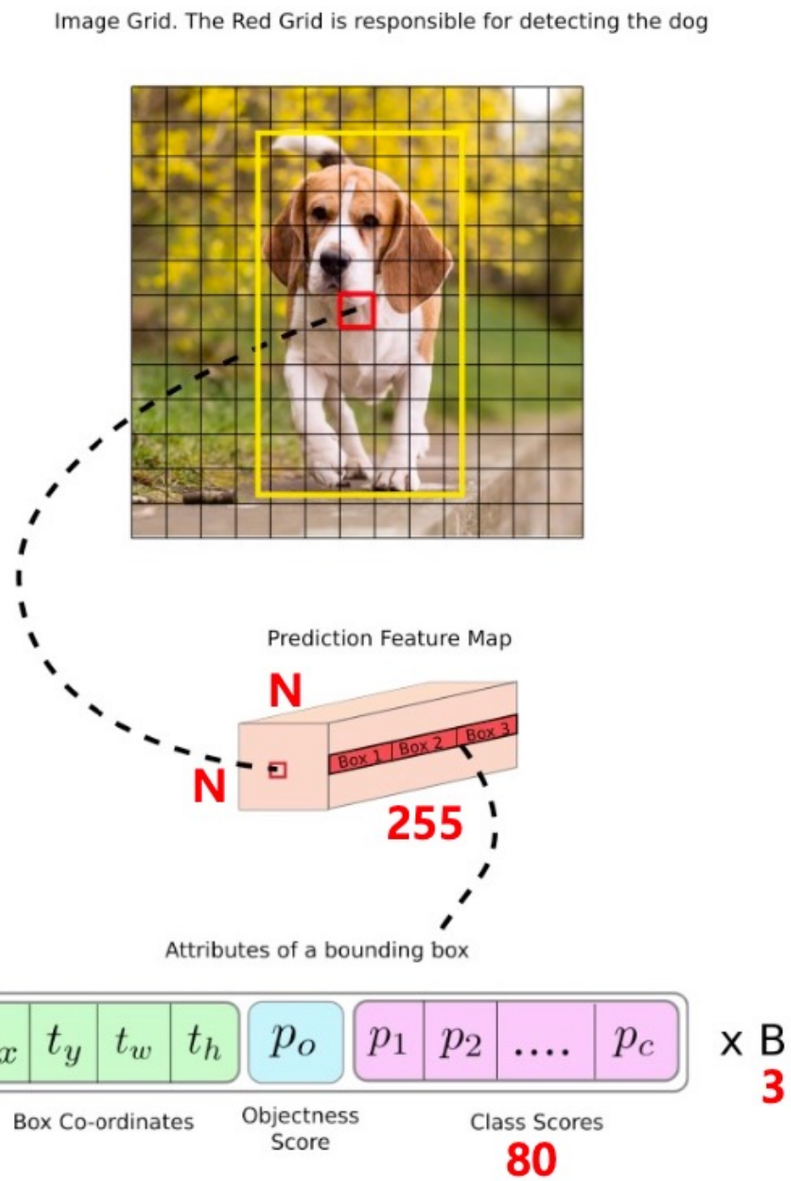
참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Predictions Across Scales

- The most salient feature of YOLOv3 is that it makes detections at three different scales
- YOLOv3 predicts boxes at 3 scales
- YOLOv3 predicts 3 boxes at each scale

→ in total 9 boxes

  - So the tensor is N x N x (3 x (4 + 1 + 80))



참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Anchor Boxes

- They still use k-means clustering to determine bounding box priors. They just sort of chose 9 clusters and 3 scales arbitrarily and then divide up the clusters evenly across scales.

- On the COCO dataset the 9 clusters were:
(10x13), (16x30), (33x23), (30x61), (62x45), (59x119), (116x90), (156x198), (373x326).

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

No. of Bounding Boxes

- YOLOv1 predicts 98 boxes (7x7 grid cells, 2 boxes per cell @448x448)
- YOLOv2 predicts 845 boxes (13x13 grid cells, 5 anchor boxes @416x416)
- YOLOv3 predicts 10,647 boxes (@416x416)
- YOLOv3 predicts more than 10x the number of boxes predicted by YOLOv2

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Feature Extraction

- Darknet-53

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

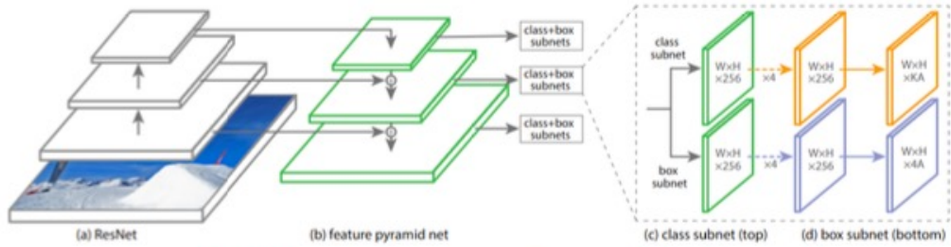
- Darknet-53 is better than ResNet-101 and 1.5x faster. Darknet-53 has similar performance to ResNet-152 and is 2x faster.
- Darknet-53 also achieves the highest measured floating point operations per second. This means the network structure better utilizes the GPU, making it more efficient to evaluate and thus faster.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

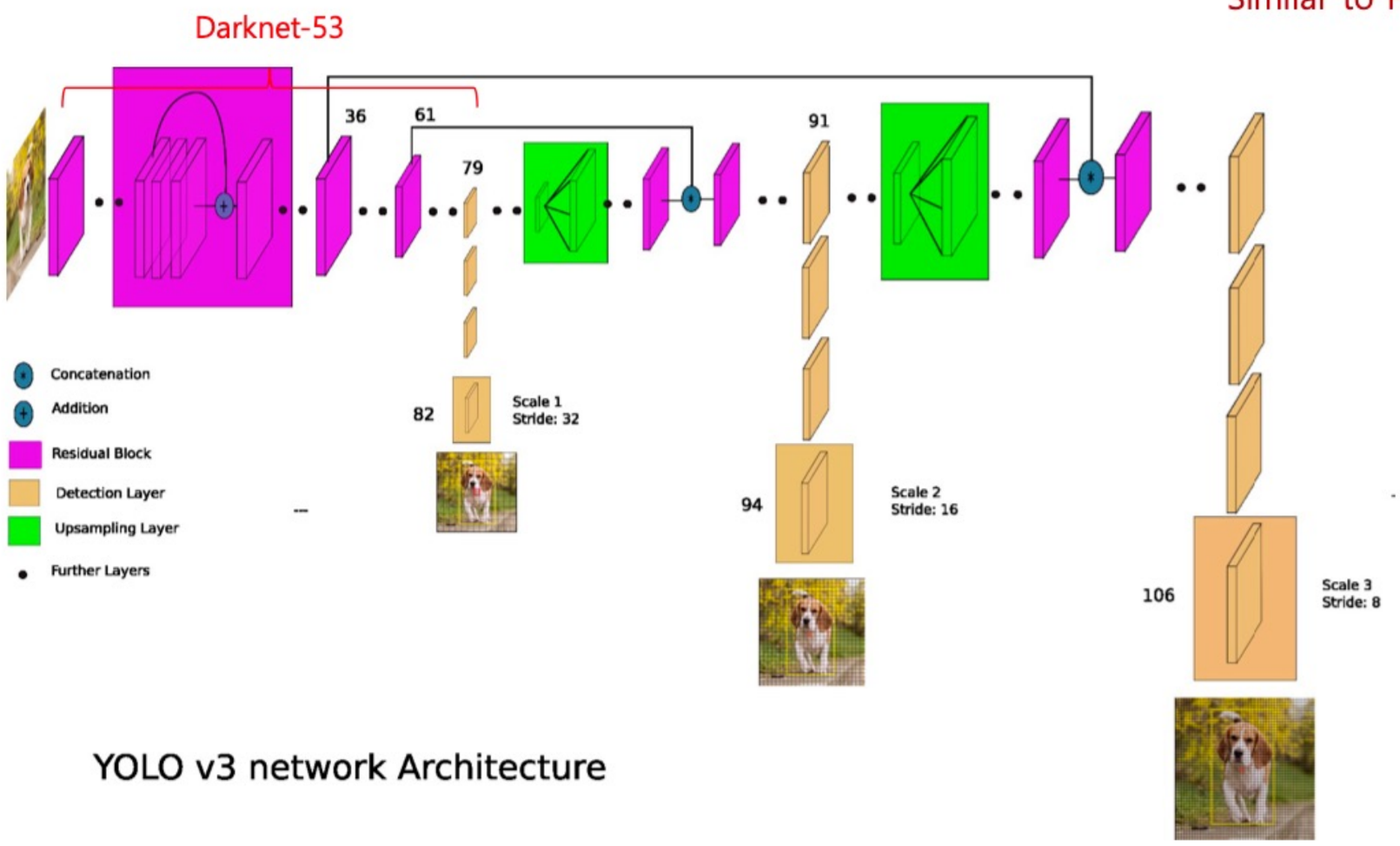
참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

YOLOv3 Architecture

YOLOv3 Architecture



Similar to Feature Pyramid Network



YOLO v3 network Architecture

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Training

- Authors still train on full images with no hard negative mining or any of that stuff.
- They use multi-scale training, lots of data augmentation, batch normalization, all the standard stuff.

오답 노트

Hard negative mining
https://m.blog.naver.com/sogangori/221073537958

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

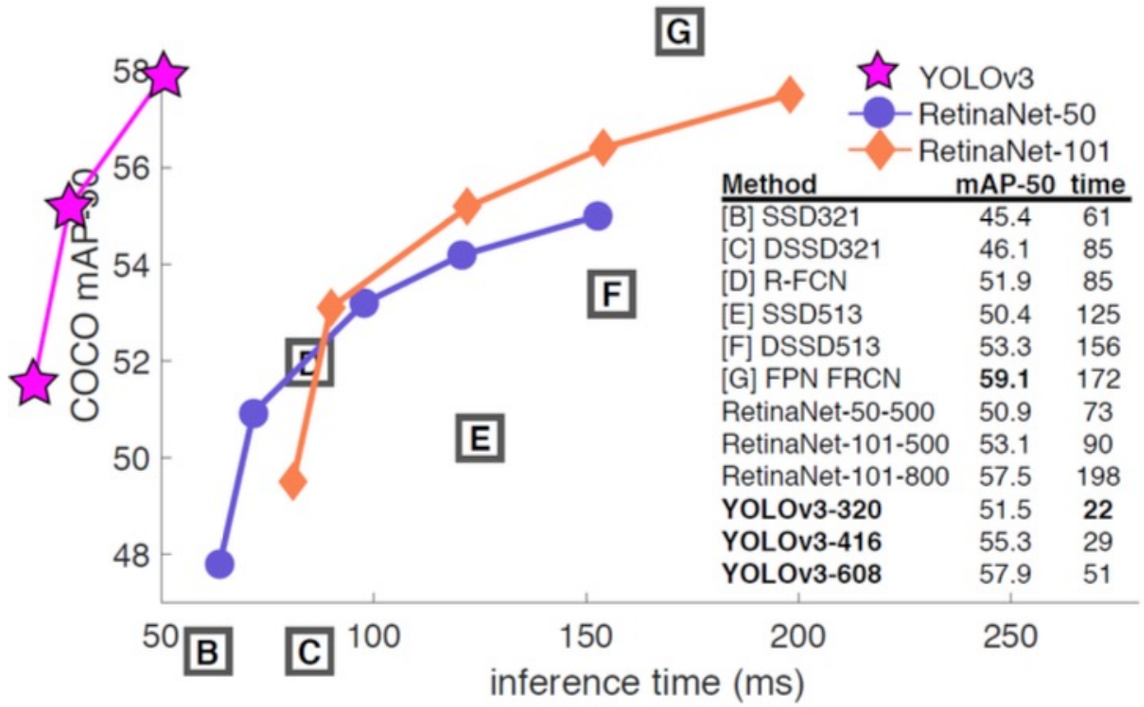
Results

- It is still quite a bit behind other models like RetinaNet in this metric though.
- However, when we look at the "old" detection metric of mAP at IOU= 0.5 (or AP50 in the chart) YOLOv3 is very strong.
- With the new multi-scale predictions YOLOv3 has relatively high APs performances.

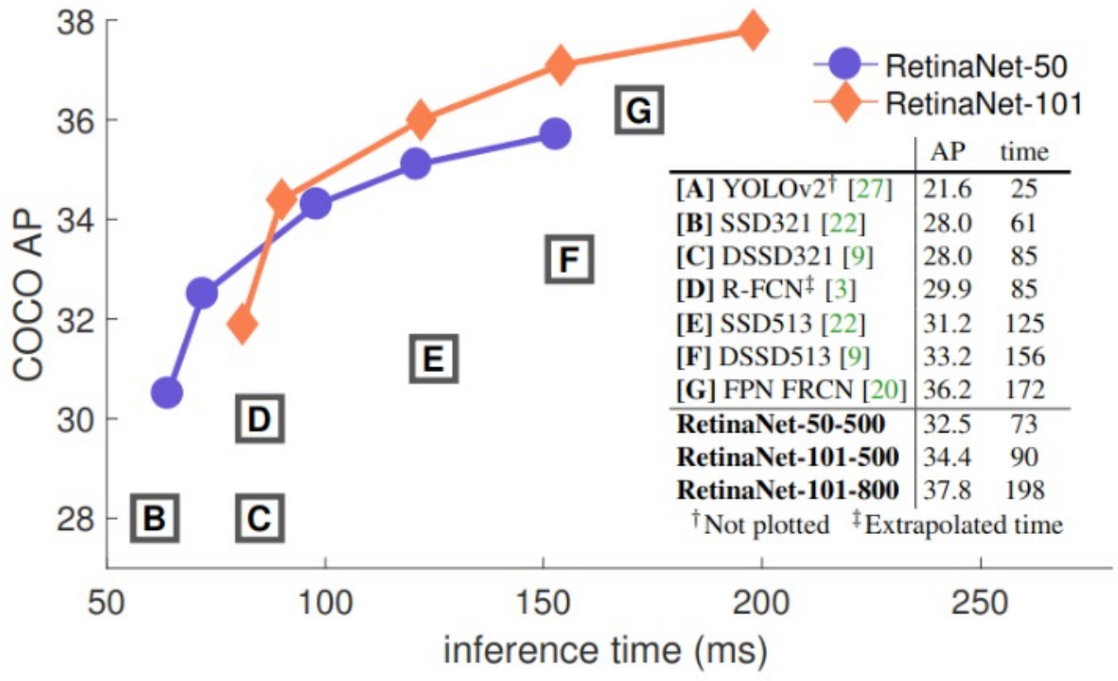
	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Results



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Things We Tried That Didn't Work

Things We Tried That Didn't Work

$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P)).$$

- Anchor box x, y offset predictions
  - Using the normal anchor box prediction mechanism where you predict the x, y offset as a multiple of the box width or height
- Linear x, y predictions instead of logistic
- Focal loss
  - Focal loss dropped YOLOv3's mAP about 2 points
- Dual IOU thresholds and truth assignment
  - Faster R-CNN uses two IOU thresholds during training. If a prediction overlaps the ground truth by 0.7 it is as a positive example, by [0.3-0.7] it is ignored, less than 0.3 for all ground truth objects it is a negative example.

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

What This All Means

- YOLOv3 is a good detector. It's fast, it's accurate. It's not as great on the COCO average AP between .5 and .95 IOU metric. But it's very good on the old detection metric of .5 IOU.

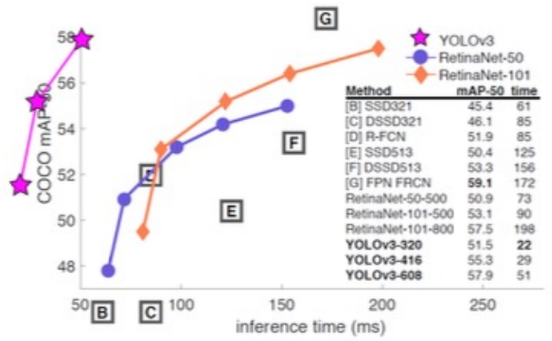
- Russakovsky et al report that that humans have a hard time distinguishing an IOU of .3 from .5!
  - Training humans to visually inspect a bounding box with IOU of 0.3 and distinguish it from one with IOU 0.5 is surprisingly difficult.

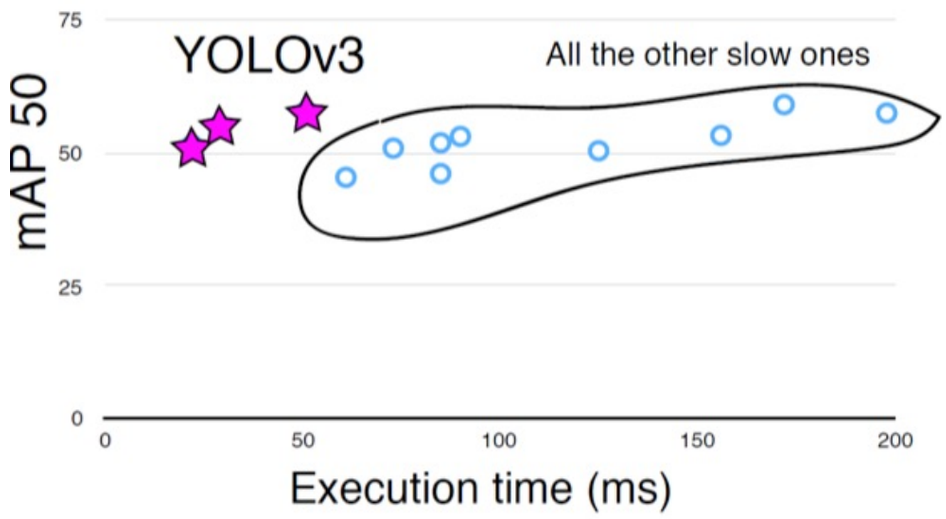
참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

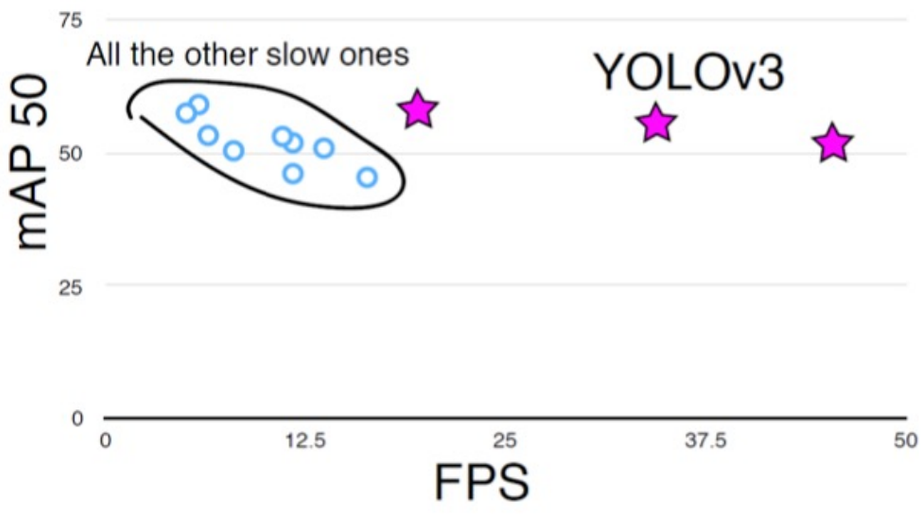
Rebuttal

Rebuttal

- Graphs have not one but two non-zero origins







참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

Rebuttal

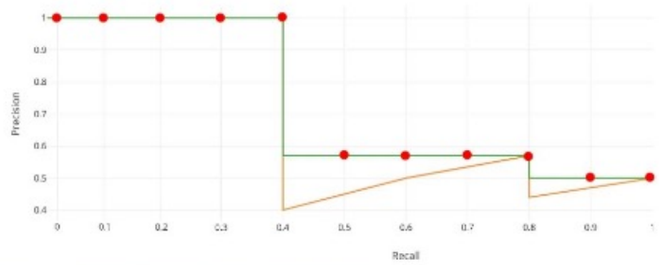
- For PASCAL VOC, the IOU threshold was "set deliberately low to account for inaccuracies in bounding boxes in the ground truth data"
- COCO can have better labelling than VOC since COCO has segmentation masks. But there was the lack of justification for updating mAP.
- Emphasis must mean it de-emphasizes something else, in this case classification accuracy. A miss-classified example is much more obvious than a bounding box that is slightly shifted.

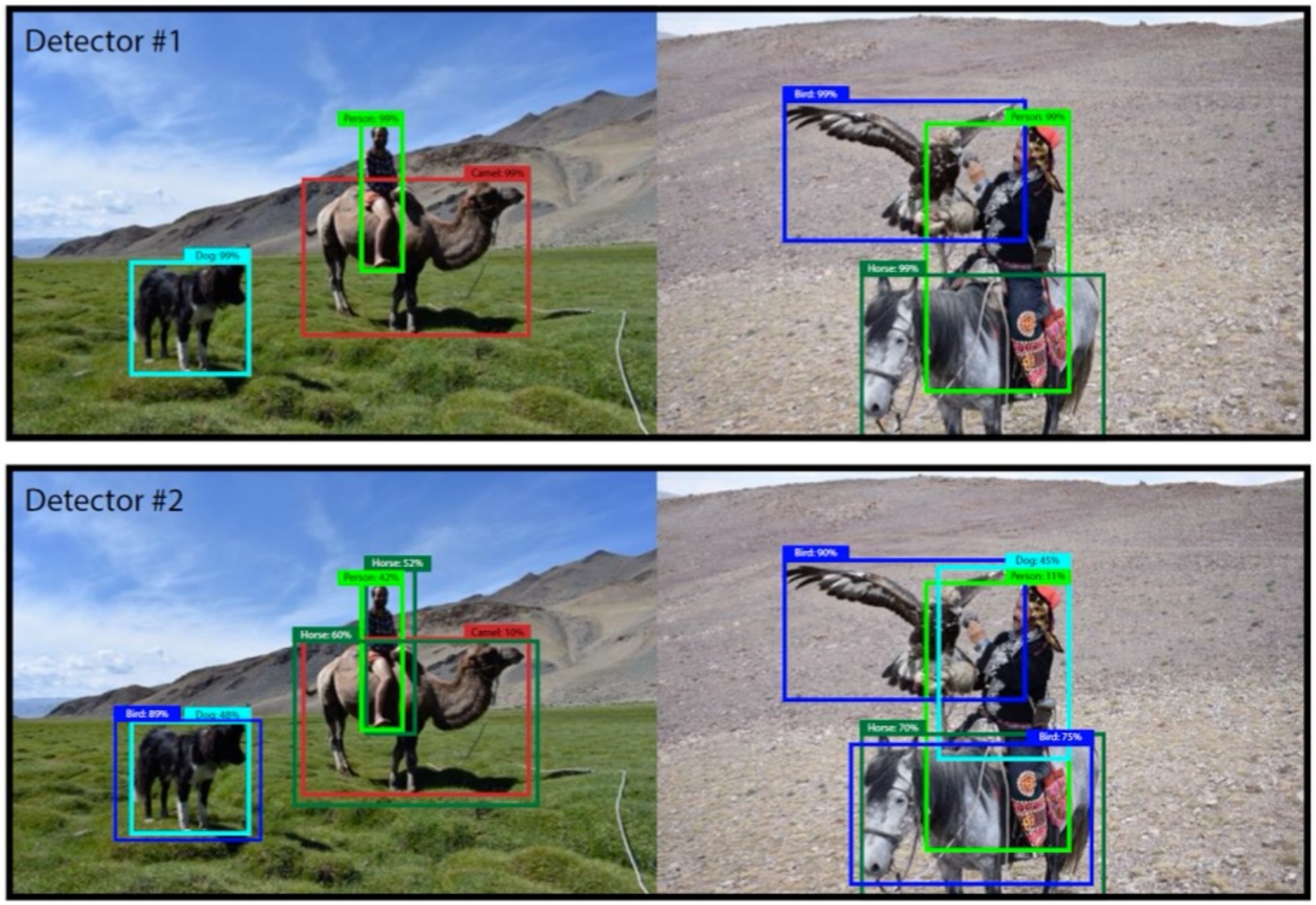
참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

mAP's Problems

mAP's Problems

- They are both perfect! (mAP = 1.0)





https://a292run.tistory.com/entry/mean-Average-PrecisionmAP-계산하기-1

참조 : https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

인공지능 심화 과정

YOLO v1
YOLO v2
YOLO v3
YOLO v4

Introduction

The majority of Object Detectors are largely applicable **only for** recommendation systems

- Searching for free parking space → it's okay to be slow → more accurate
- Car collision warning → need to fast → inaccurate

→ Need to design a fast and accurate object detector for production systems

Main Contributions

- Develop an efficient and powerful object detection models. It makes everyone can use just single GPU (1080 Ti or 2080 Ti)
- Verify the influence of SOTA Bag-of-Freebies and Bag-of-Specials methods
- Modify SOTA methods and make them more efficient and suitable for single GPU training

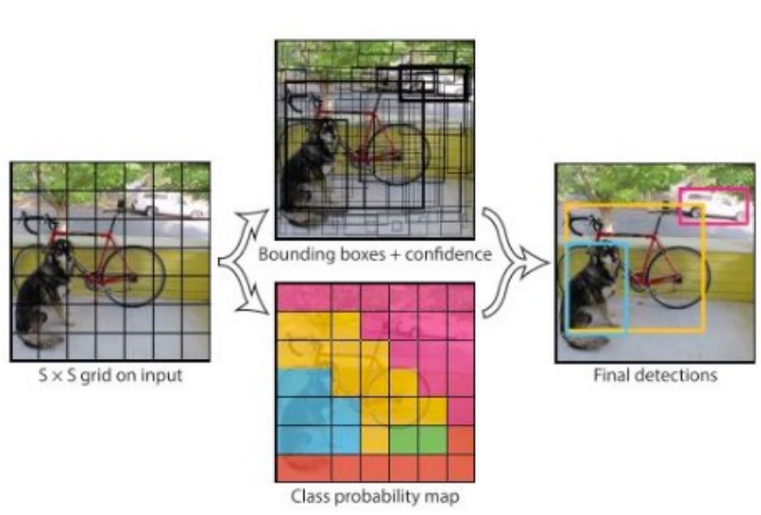
참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Introduction

YOLO v1 ~ v3 quick review: YOLO v2

- YOLO v1 + many algorithms

YOLO v1



Better

Batch Normalization
High resolution classifier
Anchor boxes
Dimension clusters
Direct location prediction
Fine-grained features
Multi-scale training

Faster

Darknet-19
Transfer learning

Stronger

Hierarchical classification
Dataset combination with Word-tree
Joint classification and detection

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Introduction

YOLO v1 ~ v3 quick review: YOLO v3

- YOLO v2 + many algorithms (**YOLOv3: An Incremental Improvement**)

YOLO v2

+

Better
Batch Normalization
High resolution classifier
Anchor boxes
Dimension clusters
Direct location prediction
Fine-grained features
Multi-scale training

Faster
Darknet-19
Transfer learning

Stronger
Hierarchical classification
Dataset combination with Word-tree
Joint classification and detection

+

Bounding box prediction → sum of squared loss

Class prediction → Multilabel classification

Predictions across scales

Darknet-53

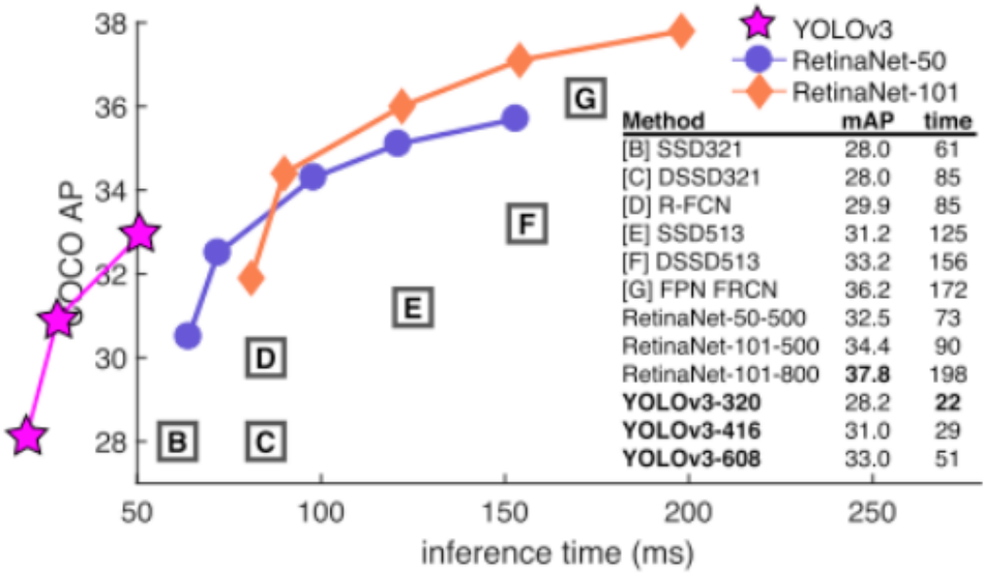
참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Introduction

YOLOv4: Optimal Speed and Accuracy of Object Detection

- YOLO 3 + many algorithms

YOLO v3



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

+

Bag of Freebies

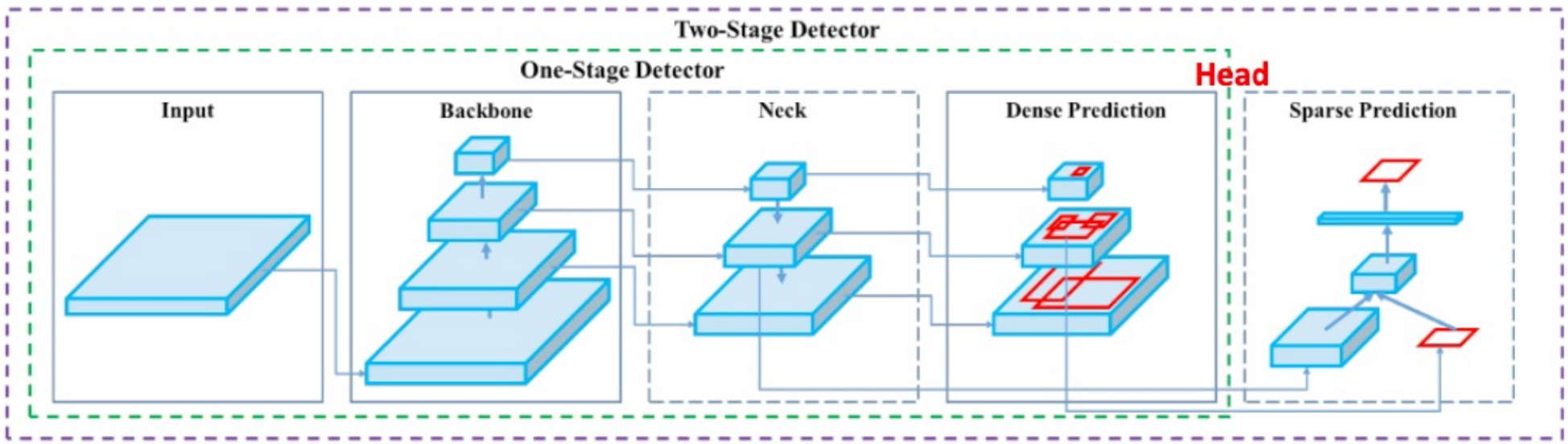
Bag of Specials

CSPDarknet53 + SPP, PAN

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Related Work

Object Detection Models



Input: { Image, Patches, Image Pyramid, ... }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], ... } EfficientNet-B0/B7 [75], CSPResNeXt50 [81], CSPDarknet53 [81]

Neck: { FPN [44], PANet [49], Bi-FPN [77], ... } NAS-FPN [17], ASFF [48], SFAM [98] SPP [25], ASPP [5], RFB [47], SAM [85]

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], ... } CornerNet [37], CenterNet [13], MatrixNet [60]

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], ... } Mask R-CNN [23] RepPoints [87]

Figure 2: Object detector.

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Related Work

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**

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Related Work

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”

Attention Module	Normalization	Post Processing
• Squeeze-and-Excitation (SE) • Spatial Attention Module (SAM)	• Batch Norm (BN) • Cross-GPU Batch Norm (CGBN or SyncBN) • Filter Response Normalization (FRN) • Cross-Iteration Batch Norm (CBN)	• NMS • Soft NMS • DIoU NMS

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Architecture

Selection of architecture

- Higher input network size (resolution) – for detecting multiple small-sized objects
- More layers – for a higher receptive field to cover the increased size of input network
- More parameters – for greater capacity of a model to detect multiple objects of different sizes in a single image

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

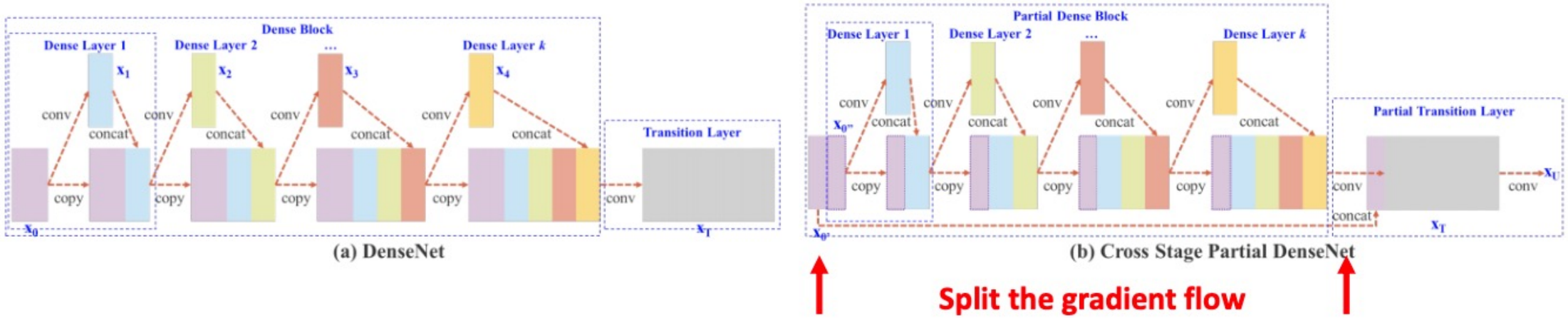
better!

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

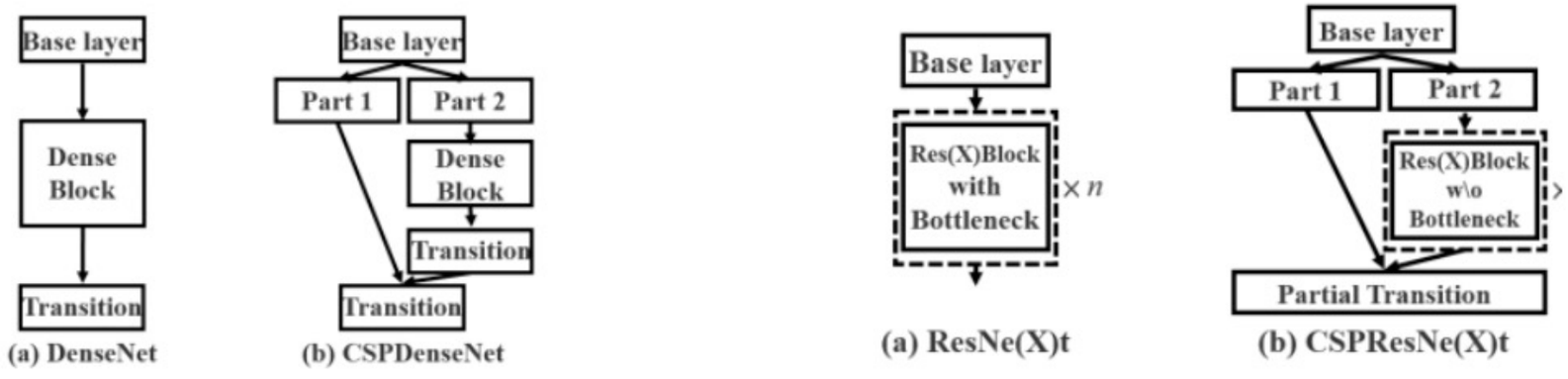
Architecture

CSPNet: A New Backbone that can Enhance Learning Capability of CNN, 2020 CVPRW

- Propose Cross Stage Partial Network to mitigate heavy inference computations
- Partition feature map of the base layer into two parts and the merge them



(a) DenseNet (b) Cross Stage Partial DenseNet

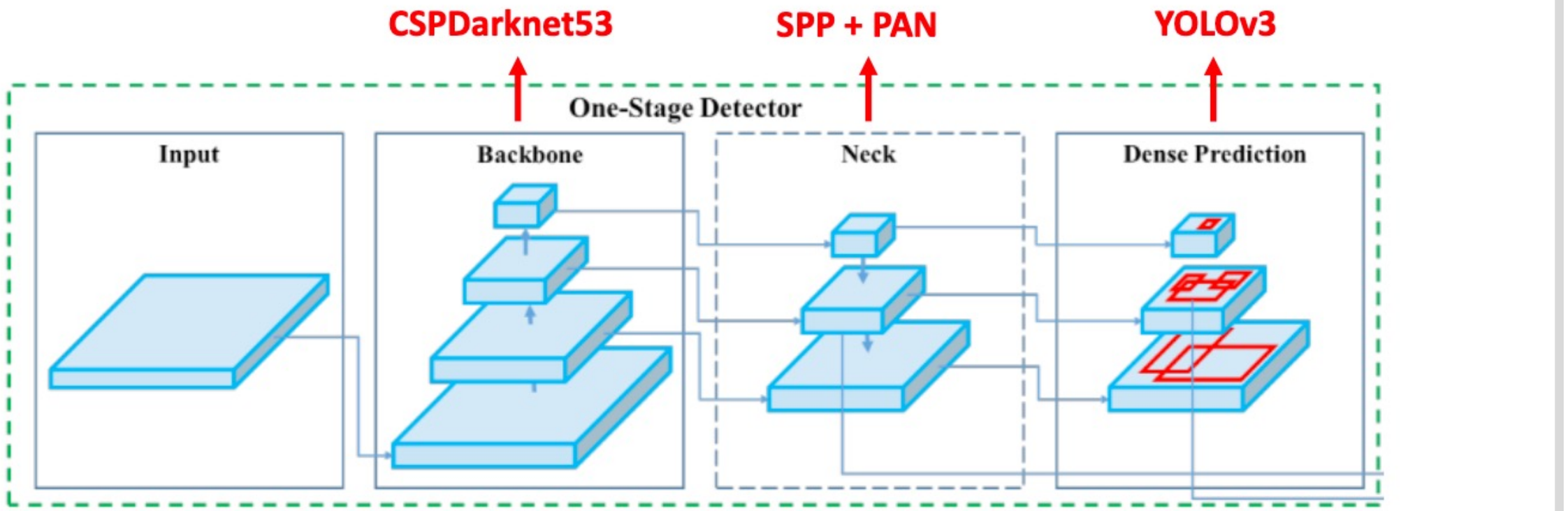


(a) DenseNet (b) CSPDenseNet (a) ResNe(X)t (b) CSPResNe(X)t

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Architecture

Selection of architecture



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

↓

Path Aggregation Network

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Selection BoF and BoS

- **Activations**: ReLU, leaky-ReLU, ~~parametric-ReLU, ReLU6, SELU,~~ Swish, or Mish

PReLU, SELU → difficult to train
ReLU6 → designed for quantization network

- **Bounding box regression loss**: MSE, IoU, GIoU, CIoU, DIoU

- **Data augmentation**: CutOut, MixUp, CutMix

- **Regularization method**: ~~DropOut, DropPath [36], Spatial DropOut [79],~~ or DropBlock

DropBlock's author have compared their method with other method and has won a lot

- **Normalization of the network activations by their mean and variance**: Batch Normalization (BN) [32], ~~Cross-GPU Batch Normalization (CGBN or SyncBN)~~ [93], Filter Response Normalization (FRN) [70], or Cross-Iteration Batch Normalization (CBN) [89]

Only use single GPU → SyncBN is not considered

- **Skip-connections**: Residual connections, Weighted residual connections, Multi-input weighted residual connections, or Cross stage partial connections (CSP)

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Experiments

Influence of **BoF and Mish** on classifier training

- CutMix, Mosaic data augmentation, Label smoothing → improved!
- Mish activation → Good!

(a) Crop, Rotation, Flip, Hue, Saturation, Exposure, Aspect.

(b) MixUp

(c) CutMix

(d) Mosaic

(e) Blur

$$f(x) = x \cdot tanh(\varsigma(x)) \quad (1)$$

where, $\varsigma(x) = \ln(1 + e^x)$ is the softplus activation [10] function. The graph of Mish is shown in Figure 1.

Table 2: Influence of BoF and Mish on the CSPResNeXt-50 classifier accuracy.

MixUp	CutMix	Mosaic	Bluring	Label Smoothing	Swish	Mish	Top-1	Top-5
							77.9%	94.0%
✓							77.2%	**94.0%**
	✓						**78.0%**	**94.3%**
		✓					**78.1%**	**94.5%**
			✓				77.5%	93.8%
				✓			**78.1%**	**94.4%**
					✓		64.5%	86.0%
						✓	**78.9%**	**94.5%**
✓	✓			✓			**78.5%**	**94.8%**
✓	✓			✓		✓	**79.8%**	**95.2%**

Table 3: Influence of BoF and Mish on the CSPDarknet-53 classifier accuracy.

MixUp	CutMix	Mosaic	Bluring	Label Smoothing	Swish	Mish	Top-1	Top-5
							77.2%	93.6%
✓	✓			✓			**77.8%**	**94.4%**
✓	✓			✓		✓	**78.7%**	**94.8%**

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Experiments

Influence of **BoF and Mish** on object detector training

- S: Eliminate grid sensitivity → worse performance
- M: Mosaic data augmentation
- IT: IoU threshold using multiple anchors for a single GT → worse performance
- GA: Genetic algorithms for selecting the optimal hyperparameters on the first 10% of time periods
- LS: Class label smoothing → worse performance
- CBN: Cross mini-Batch Normalization
- CA: Cosine annealing scheduler
- DM: Dynamic mini-batch size → worse performance
- OA: Optimized Anchors

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Experiments

Influence of **BoF and Mish** on object detector training

- PAN + SPP + SAM → better performance!
- RFP, Gaussian YOLO(G), ASFF → worse performance

Table 4: Ablation Studies of Bag-of-Freebies. (CSPResNeXt50-PANet-SPP, 512x512).

S	M	IT	GA	LS	CBN	CA	DM	OA	loss	AP	AP_{50}	AP_{75}
									MSE	38.0%	60.0%	40.8%
✓									MSE	37.7%	59.9%	40.5%
	✓								MSE	**39.1%**	**61.8%**	**42.0%**
		✓							MSE	36.9%	59.7%	39.4%
			✓						MSE	**38.9%**	**61.7%**	**41.9%**
				✓					MSE	33.0%	55.4%	35.4%
					✓				MSE	**38.4%**	**60.7%**	**41.3%**
						✓			MSE	**38.7%**	**60.7%**	**41.9%**
							✓		MSE	35.3%	57.2%	38.0%
✓									GIoU	**39.4%**	59.4%	**42.5%**
✓									DIoU	**39.1%**	58.8%	**42.1%**
✓									CIoU	**39.6%**	59.2%	**42.6%**
✓	✓	✓	✓						CIoU	**41.5%**	**64.0%**	**44.8%**
	✓		✓					✓	CIoU	36.1%	56.5%	38.4%
✓	✓	✓	✓					✓	MSE	**40.3%**	**64.0%**	**43.1%**
✓	✓	✓	✓					✓	GIoU	**42.4%**	**64.4%**	**45.9%**
✓	✓	✓	✓					✓	CIoU	**42.4%**	**64.4%**	**45.9%**

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%

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review

Experiments

Influence of **different backbones** on object detector training

- Although classification accuracy of CSPResNeXt is higher compared to CSPDarknet, CSPDarknet model shows higher accuracy in terms of object detection

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

참조 : https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review