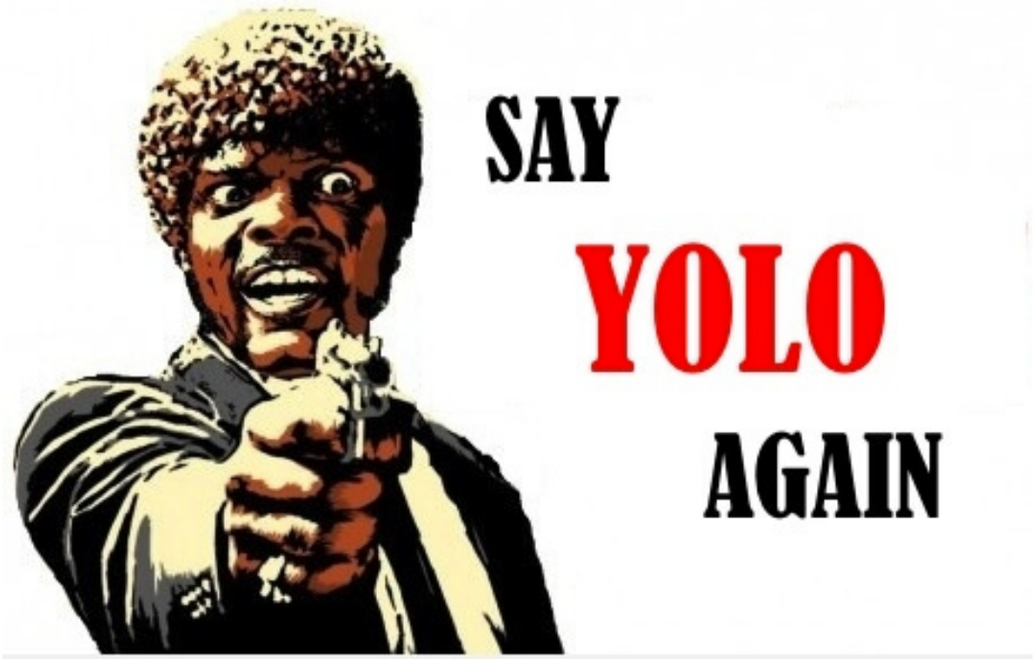
DIVE INTO CODE

YOLOv3: Incremental Improvement



Joseph Redmon Ali Farhadi University of Washington
https://arxiv.org/pdf/1804.02767.pdf

Summary

By Komiya moriyasu
2019/03/11

Summary

DIVE INTO CODE

What is this thesis for?

YOLO 9000からパワーアップ！

Where is an important point compared to previous researches?

SSDと同じくらいの正確さで3倍高速になった

Where are the key points of technology and method?

1. Softmax関数の禁止 ロジスティク回帰
2. 3つの異なるスケールでボックスを予測
3. 53層のモデルを使用

Is there discussions?

大きいオブジェクトでのパフォーマンス低下

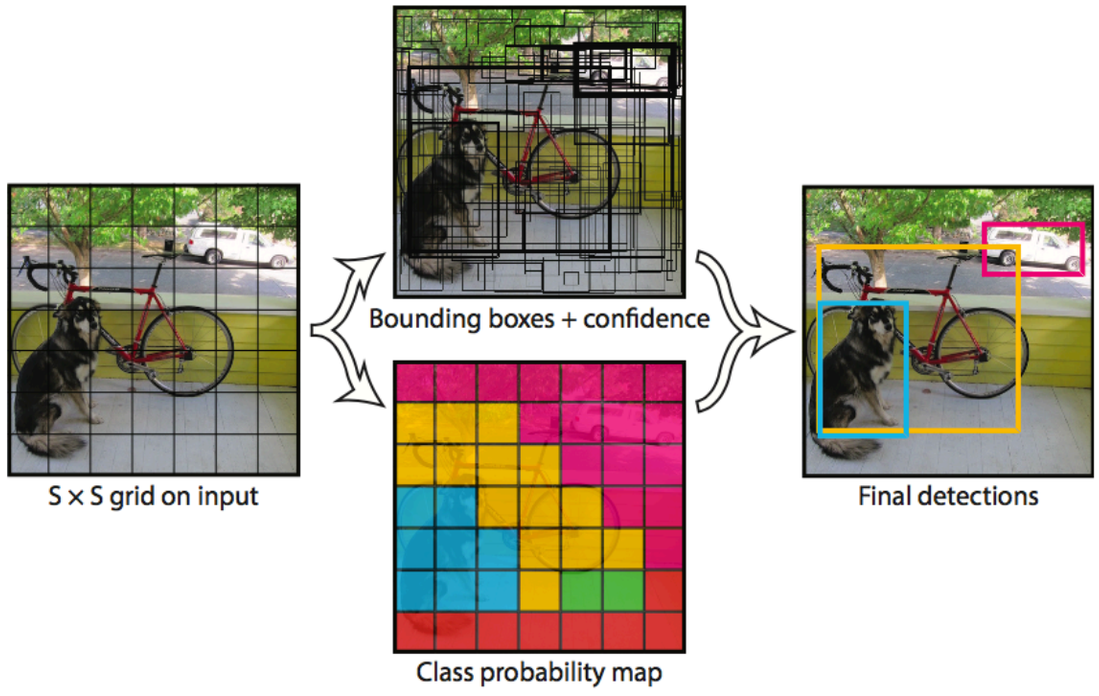
Which reserches should I read next?

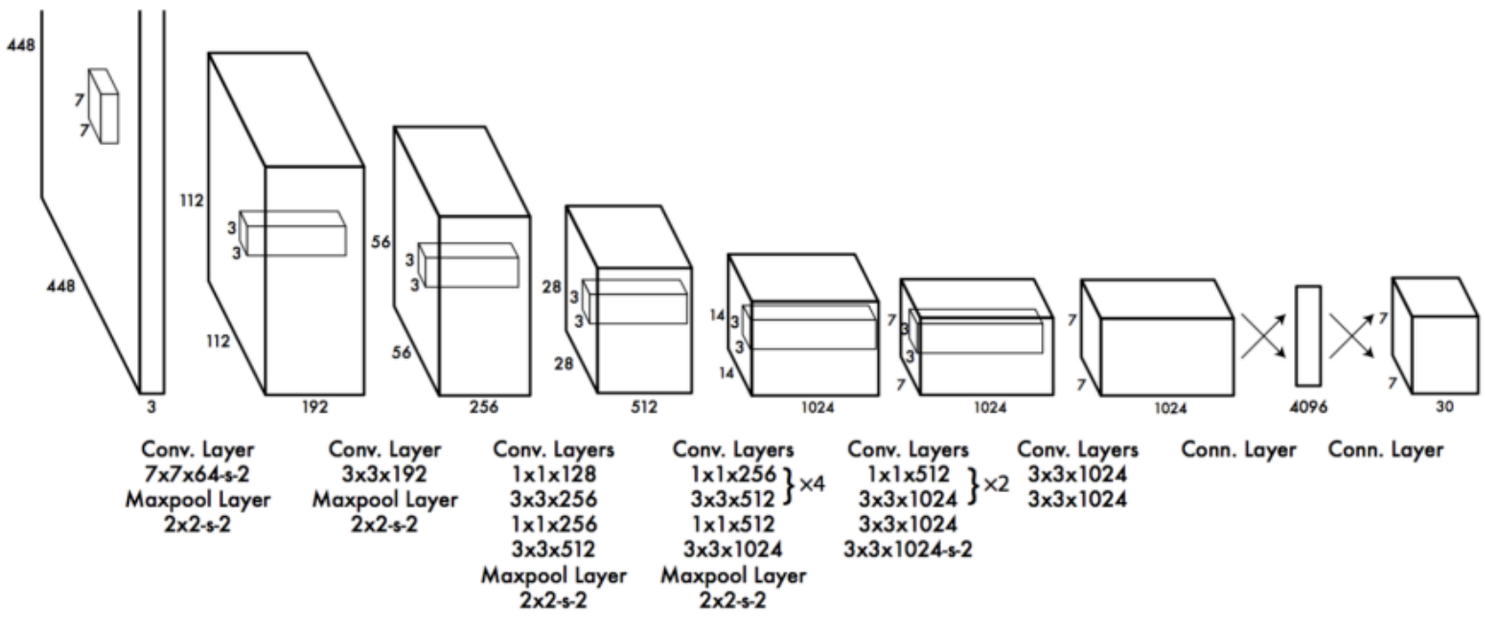
YOLO,
YOLO 9000

What is ~~...??:

そもそもYOLOとは → 物体検出をワンステージで行う

1． 入力画像をS×Sのグリッドセルと呼ばれる領域に分割
2. それぞれのグリッドセルについてB個のバウンディングボックス(boundhing box)と信頼度スコア(confidence score)を推測
それと同時にそれぞれのグリッドセルはC個の物体クラスそれぞれの条件付きクラス確率を推定
3.「条件付きクラス確率」と「個々のボックスの信頼度スコア」をそれぞれ掛け合わせる

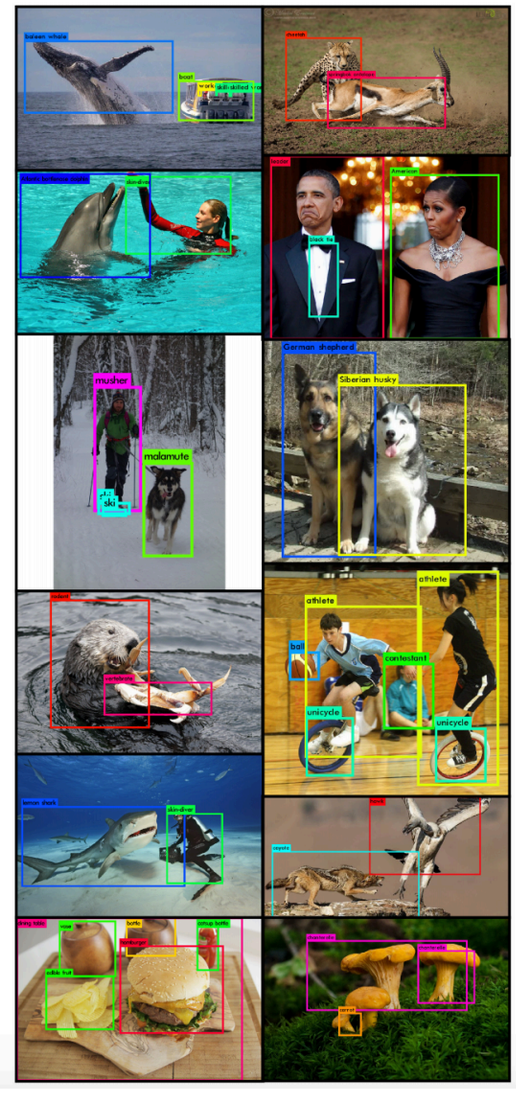




What is ~~...??:

YOLO v2

1．グリッドベース→アンカーボックス(kmeans)
Gridごとにbounding box及びconfidenceを予測
2.FCN(Fully Convolutional Networks)による特徴マップ抽出 (特徴マップの精確な位置情報を保持したまま最終層まで伝播)
3.bounding boxごとに条件付き確率を予測



Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Table 6: Darknet-19.

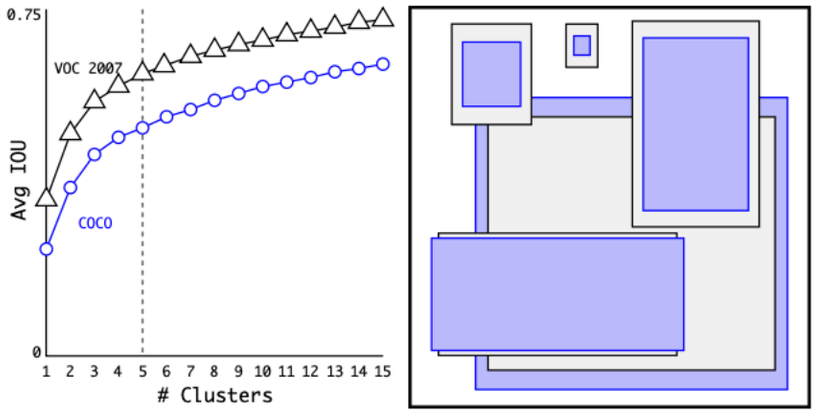


Figure 2: Clustering box dimensions on VOC and COCO. We run k-means clustering on the dimensions of bounding boxes to get good priors for our model. The left image shows the average IOU we get with various choices for k. We find that $k = 5$ gives a good tradeoff for recall vs. complexity of the model. The right image shows the relative centroids for VOC and COCO. Both sets of priors favor thinner, taller boxes while COCO has greater variation in size than VOC.

What is ~~...??:

バウンディングボックス

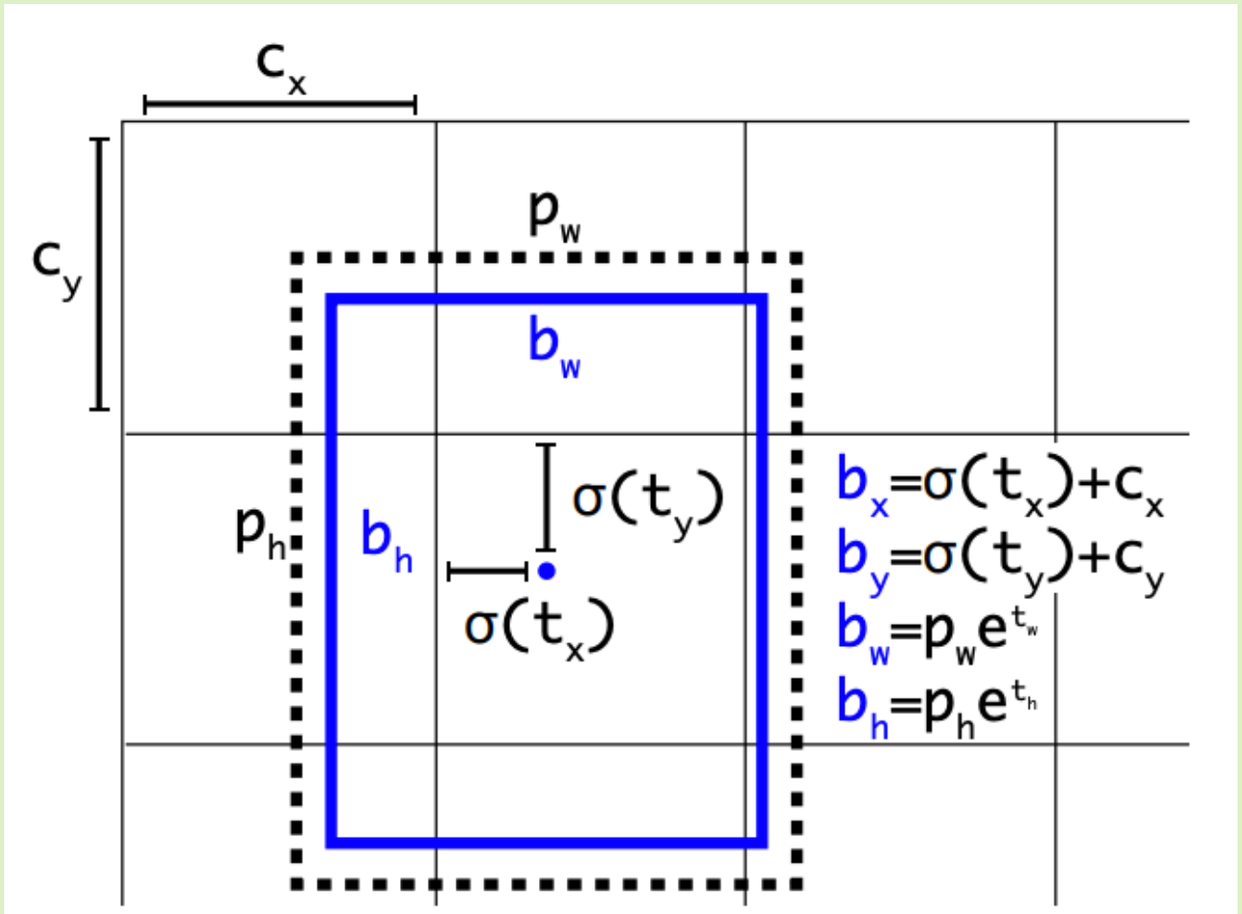


Figure 2. **Bounding boxes with dimension priors and location prediction.** We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function. This figure blatantly self-plagiarized from [15].

What is ~~...??:

Darknet-53

	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			

Table 1. **Darknet-53.**

v2の19層モデルに対してv3では
53層のNeural Networkモデルを使用

プーリング目的の畳み込み層と
複数の Residual Block を繰り返す構造となっている。
53 層の畳み込み層で構成される。

Body の部分を特徴抽出器として使用

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

Table 2. **Comparison of backbones.** Accuracy, billions of operations, billion floating point operations per second, and FPS for various networks.

What is ~~...??:

softmaxやめました

softmaxを使用すると、各ボックスに厳密に1つのクラスがあるという仮定 ❌

→ 独立したロジスティック分類器を使用

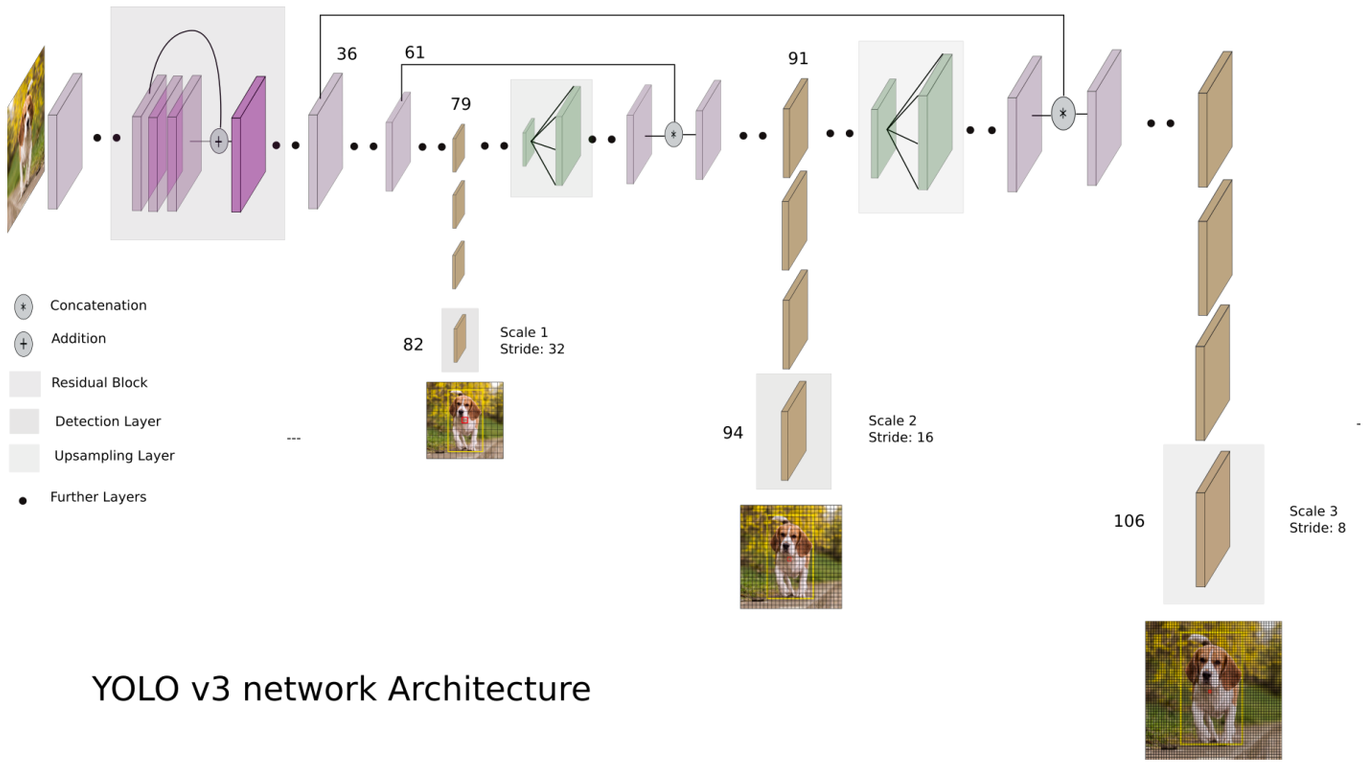
DIVE INTO CODE

What is ~~...??:

3つの異なるスケール&アップサンプリング

・ YOLOv3は3つの異なるスケールでボックスを予測する
この3つの異なるスケールから特徴量を抽出し、Feature mapを作成する

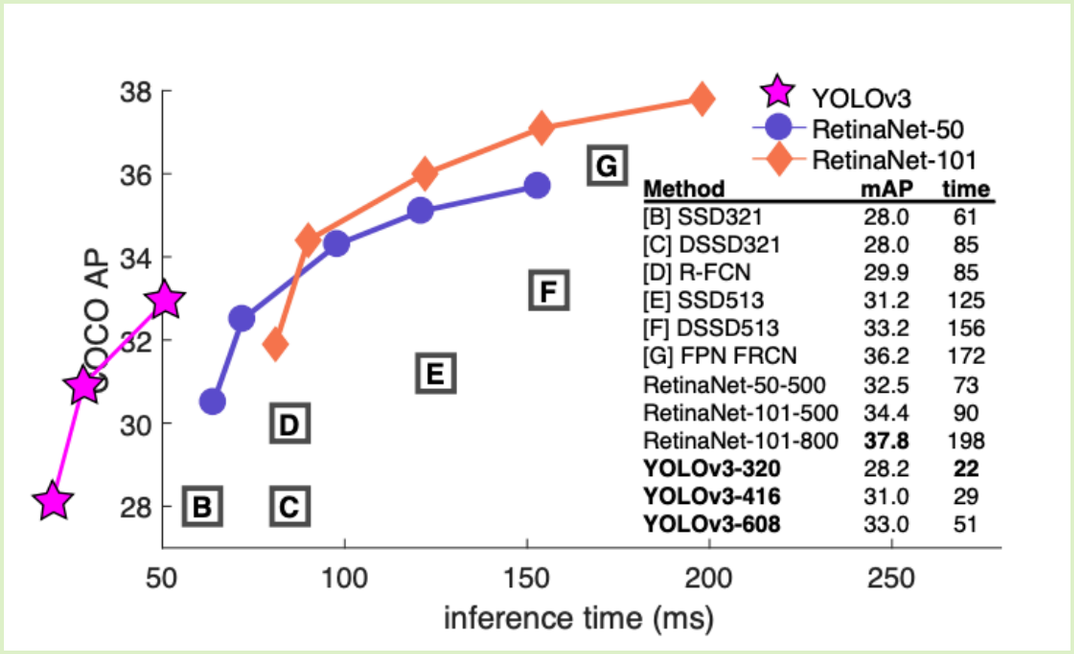
・アップサンプリング
直前の2つのレイヤー層からFeature mapを取得し、それを2倍にアップサンプリングする
また、ネットワークの最初のLayerからFeature mapを取得し、要素別の追加機能を使用して前述のmapとマージ。
この方法では、Object Boxのより意味のある情報と細かい情報を取得するのに役立つ



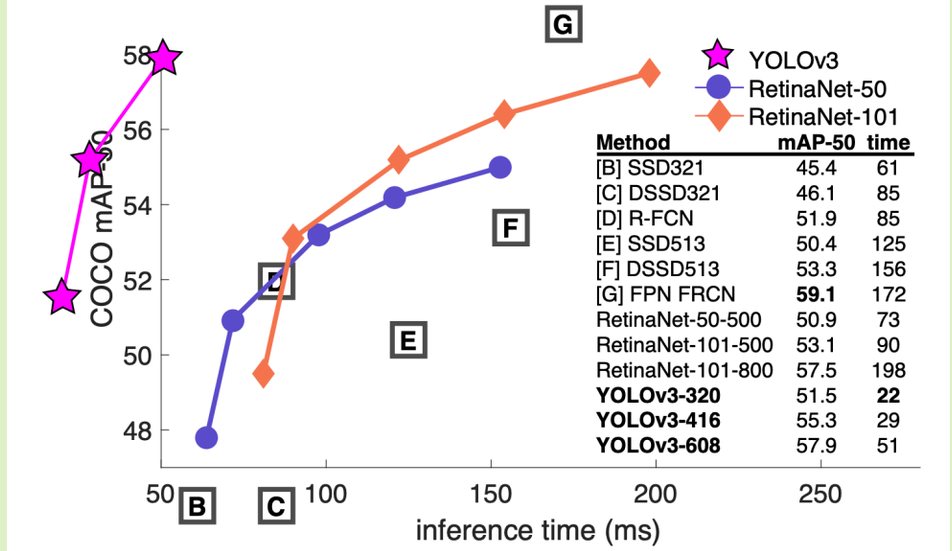
YOLO v3 network Architecture

What is ~~...??:

mAP



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



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

Figure 3. Again adapted from the [9], this time displaying speed/accuracy tradeoff on the mAP at .5 IOU metric. You can tell YOLOv3 is good because it's very high and far to the left. Can you cite your own paper? Guess who's going to try, this guy → [16]. Oh, I forgot, we also fix a data loading bug in YOLOv2, that helped by like 2 mAP. Just sneaking this in here to not throw off layout.

AP50だとYOLO v3が圧勝！

IOUしきい値が大きくなるにつれてパフォーマンスは大幅に低下。これは、YOLOv3がボックスをオブジェクトと完全に位置合わせするのに苦労していることを示してる