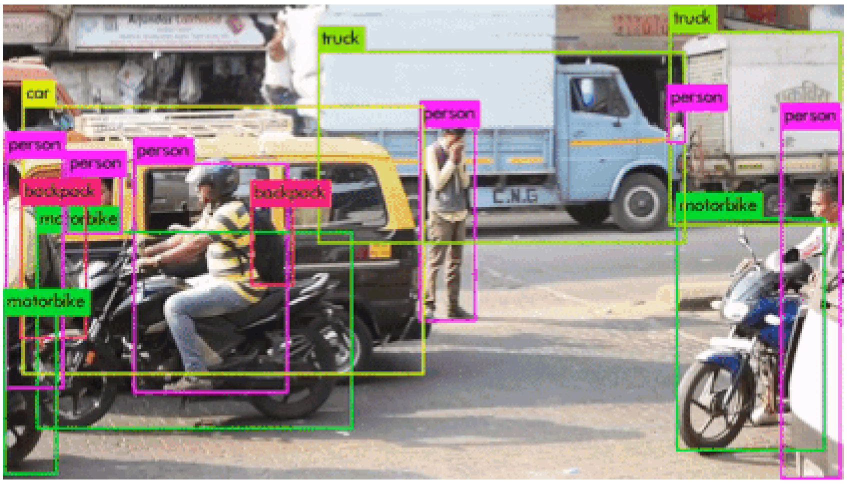
Review: YOLOv3 — You Only Look Once (Object Detection)

Improved YOLOv2, Comparable Performance with RetinaNet, 3.8× Faster!


SH Tsang Follow

Feb 8 · 5 min read



YOLOv3

In this story, **YOLOv3 (You Only Look Once v3)**, by **University of Washington**, is reviewed. YOLO is a very famous object detector. I think everybody must know it. Below is the demo by authors:

YOLOv3

As author was busy on Twitter and GAN, and also helped out with other people's research, YOLOv3 has few incremental improvements on YOLOv2. For example, a better feature extractor, **DarkNet-53** with shortcut connections as well as a better object detector with **feature map upsampling and concatenation**. And it is published as a **2018 arXiv** technical report with more than **200 citations**. (SH Tsang @ Medium)

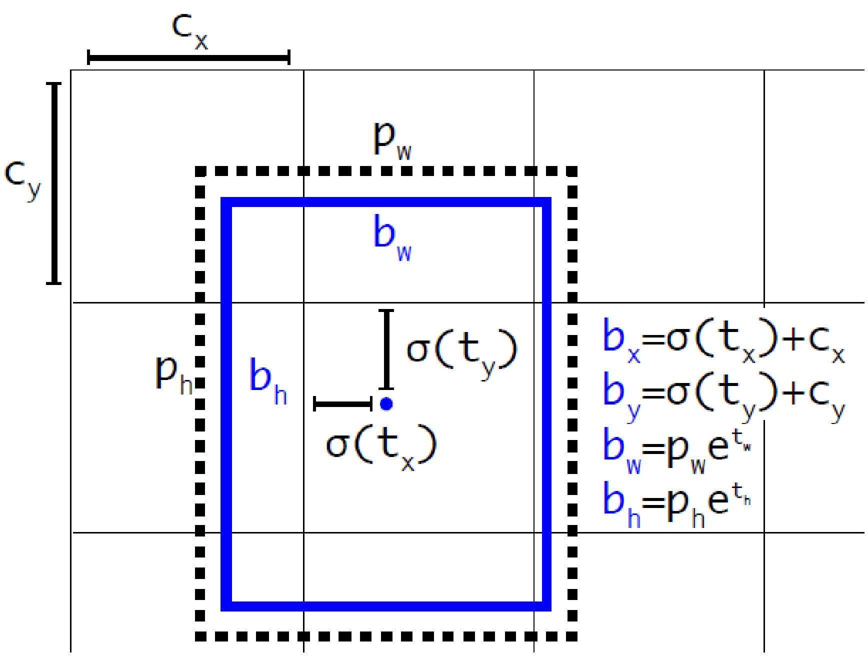
. . .

Outline

1. **Bounding Box Prediction**
2. **Class Prediction**
3. **Predictions Across Scales**
4. **Feature Extractor: Darknet-53**
5. **Results**

. . .

1. Bounding Box Prediction



Bounding Box Prediction, Predicted Box (Blue), Prior Box (Black Dotted)

- It is the same as YOLOv2.
- **tx, ty, tw, th are predicted.**
- During training, sum of squared error loss is used.
- And objectness score is predicted using logistic regression. It is 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior. Only one bounding box prior is assigned for each ground truth object.

· · ·

2. Class Prediction

- Softmax is not used.
- Instead, **independent logistic classifiers** are used and **binary cross-entropy loss** is used. Because there may be overlapping labels for multilabel classification such as if the YOLOv3 is moved to other more complex domain such as Open Images Dataset.

· · ·

3. Prediction Across Scales

- **3 different scales** are used.
- Features are extracted from these scales like FPN.
- **Several convolutional layers are added to the base feature extractor Darknet-53** (which is mentioned in the next section).
- **The last of these layers predicts the bounding box, objectness and class predictions.**
- On COCO dataset, **3 boxes at each scales.** Therefore, the output tensor is $N \times N \times [3 \times (4+1+80)]$, i.e. **4 bounding box offsets, 1 objectness prediction, and 80 class predictions**.
- Next, the **feature map** is taken from 2 layers previous and is **upsampled by $2\times$**. A feature map is also taken from earlier in the network and merge it with our upsampled features using **concatenation**. This is actually the typical **encoder-decoder architecture**, just like SSD is evolved to DSSD.

- This method allows us to get **more meaningful semantic information** from the upsampled features and **finer-grained information** from the earlier feature map.
- Then, **a few more convolutional layers are added to process this combined feature map**, and eventually predict a similar tensor, although now twice the size.
- **k-means clustering** is used here as well to find **better bounding box prior**. Finally, on COCO dataset, **(10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116×90), (156×198), and (373×326)** are used.

. . .

4. Feature Extractor: 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			

Darknet-53

- Darknet-19 classification network is used in YOLOv2 for feature extraction.
- Now, in YOLOv3, **a much deeper network Darknet-53** is used, i.e. 53 convolutional layers.
- Both YOLOv2 and YOLOv3 also use Batch Normalization.
- **Shortcut connections** are also used as shown above.

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

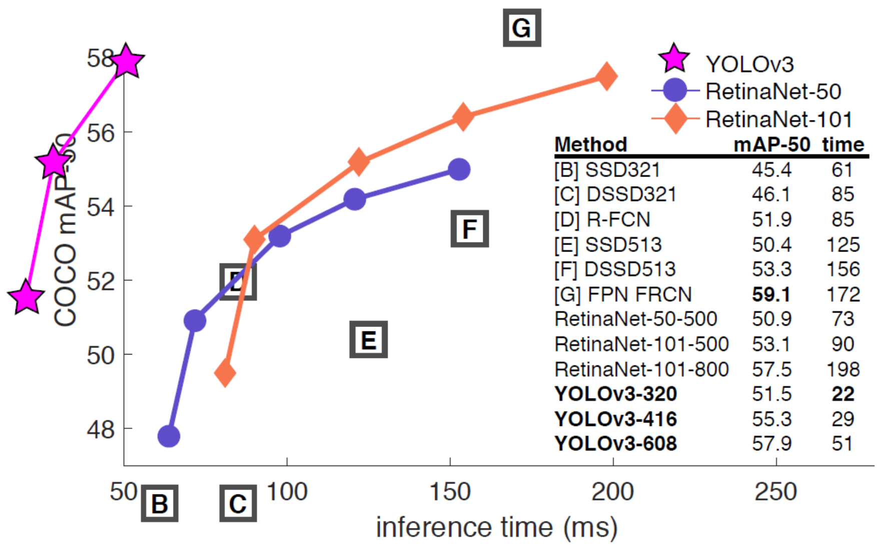
1000-Class ImageNet Comparison (Bn Ops: Billions of Operations, **BFLOP/s**: Billion Floating Point Operation Per Second, **FPS**: Frame Per Second**)**

- 1000-class ImageNet Top-1 and Top5 error rates are measured as above.
- Single Crop 256×256 image testing is used, on a Titan X GPU.
- Compared with ResNet-101, Darknet-53 has better performance (authors mentioned this in the paper) and it is 1.5× faster.
- Compared with ResNet-152, Darknet-53 has similar performance (authors mentioned this in the paper) and it is 2× faster.

· · ·

5. Results

5.1. COCO mAP@0.5

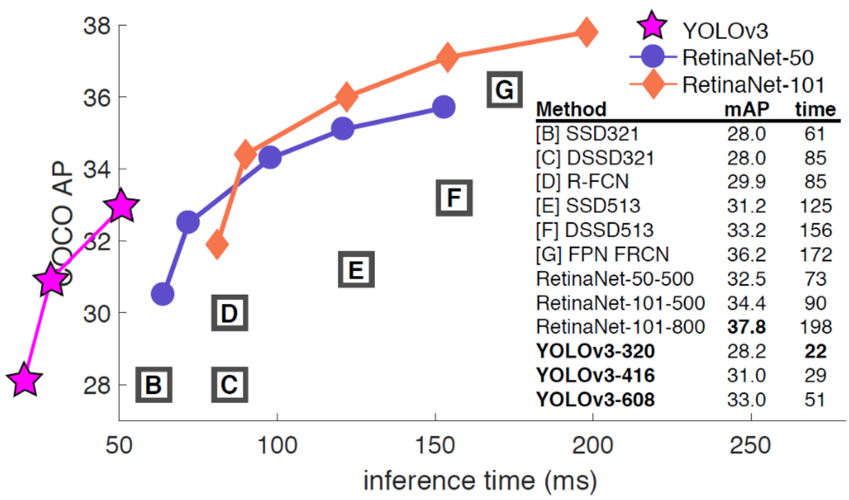


mAP@0.5

- As shown above, compared with RetinaNet, YOLOv3 got comparable mAP@0.5 with much faster inference time.

- For example, YOLOv3–608 got 57.9% mAP in 51ms while RetinaNet-101–800 only got 57.5% mAP in 198ms, which is 3.8× faster.

5.2. COCO Overall mAP



Overall mAP

- For overall mAP, YOLOv3 performance is dropped significantly.
- Nevertheless, YOLOv3–608 got 33.0% mAP in 51ms inference time while RetinaNet-101–50–500 only got 32.5% mAP in 73ms inference time.
- And YOLOv3 is on par with SSD variants with 3× faster.

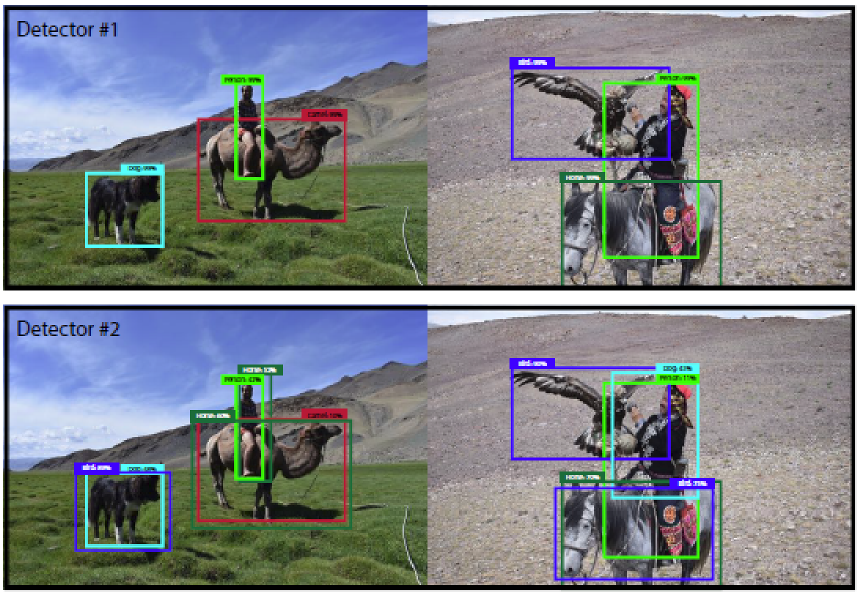
5.3. Details

	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

More Details

- YOLOv3 is much better than SSD and has similar performance as DSSD.
- And it is found that YOLOv3 has relatively good performance on AP_S but relatively bad performance on AP_M and AP_L.
- YOLOv3 has even better AP_S than two-stage Faster R-CNN variants using ResNet, FPN, G-RMI, and TDM.

5.4. Qualitative Results



Nearly Exactly The Same Between Predicted Boxes and Ground-Truth Boxes

. . .

Actually, there are not much details on YOLOv3 in the technical report. Thus, I can only briefly review about it. It is recommended to be back and forth between YOLOv2 and YOLOv3 when reading YOLOv3. (And there are passages talking about the measurement of overall mAP. "Is it really reflecting the actual detection accuracy?" If interested, please visit the paper.)

. . .

Reference

[2018 arXiv] [YOLOv3]
YOLOv3: An Incremental Improvement

My Previous Reviews

Image Classification
[LeNet] [AlexNet] [ZFNet] [VGGNet] [SPPNet] [PReLU-Net] [STN] [DeepImage] [GoogLeNet / Inception-v1] [BN-Inception / Inception-v2] [Inception-v3] [Inception-v4] [Xception] [MobileNetV1] [ResNet] [Pre-Activation ResNet] [RiR] [RoR] [Stochastic Depth] [WRN] [FractalNet] [Trimps-Soushen] [PolyNet] [ResNeXt] [DenseNet] [PyramidNet]

Object Detection
[OverFeat] [R-CNN] [Fast R-CNN] [Faster R-CNN] [DeepID-Net] [R-FCN] [ION] [MultiPathNet] [NoC] [G-RMI] [TDM] [SSD] [DSSD] [YOLOv1] [YOLOv2 / YOLO9000] [FPN] [RetinaNet] [DCN]

Semantic Segmentation
[FCN] [DeconvNet] [DeepLabv1 & DeepLabv2] [ParseNet] [DilatedNet] [PSPNet] [DeepLabv3]

Biomedical Image Segmentation
[CUMedVision1] [CUMedVision2 / DCAN] [U-Net] [CFS-FCN] [U-Net+ResNet]

Instance Segmentation
[DeepMask] [SharpMask] [MultiPathNet] [MNC] [InstanceFCN] [FCIS]

Super Resolution
[SRCNN] [FSRCNN] [VDSR] [ESPCN] [RED-Net] [DRCN] [DRRN] [LapSRN & MS-LapSRN]