# YOLOv3: An Incremental Improvement

Zhang, Liqiang

July 20, 2018

## Abstract

*It's my honour to take part in underwater robot picking contest with classmates. My task is to collocation environment for YOLOv3, so I read a paper this week about YOLOv3. YOLOv3 is still fast but more accurate. At* 320 × 320 *YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. YOLOv3 is quite good for the old .5 IOU mAP detection metric. It achieves 57.9* $AP_{50}$ *in 51 ms on a Titan X, compared to 57.5* $AP_{50}$ *in 198 ms by RetinaNet, similar performance but 3.8* × *faster. All the code is online at* `https://pjreddie.com/yolo/`*.*

## 1. Introduction

Under the background of GANS, the author has made some improvements to the previous YOLO algorithm.

In this paper, first The author tell us what the deal is with YOLOv3. Then author tell us how they do. They also tell about some things they tried that didnt work. Finally They will contemplate what this all means.

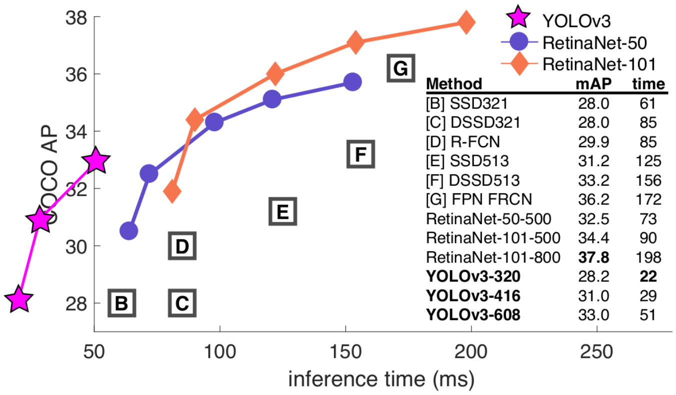
## 2. Bounding Box Prediction

Heres the deal with YOLOv3: The author trained a new classifier network thats better than the other ones. Fig. 1 can show the speed of YOLOv3.

### 2.1. Bounding Box Prediction

Following YOLO9000 the our system predicts bounding boxes using dimension clusters as anchor boxes. The network predicts 4 coordinates for each bounding box, $t_x, t_y, t_w, t_h$. If the cell is offset from the top left corner of the image by $(c_x, c_y)$ and the bounding box prior has width and height $p_w, p_h$, then the predictions correspond to the Eq. 1:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{1}$$



| 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 |

Figure 1. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

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. If the bounding box prior is not the best but does overlap a ground truth object by more than some threshold the author ignore the prediction, following [3]

### 2.2. Class Prediction

Each box predicts the classes the bounding box may contain using multilabel classification. The author do not use a softmax as they have found it is unnecessary for good performance, instead they simply use independent logistic classifiers. During training they use binary cross-entropy loss for the class predictions.

In this dataset there are many overlapping labels. Using a softmax imposes the assumption that each box has exactly one class which is often not the case. A multilabel approach better models the data.

|  | backbone | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|
| *Two-stage methods* |  |  |  |  |  |  |  |
| Faster R-CNN+++ [1] | ResNet-101-C4 | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster R-CNN w FPH [2] | ResNet-101-FPH | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| Faster R-CNN by G-RMI | Inception-ResNet-v2 | 34.7 | 55.5 | 36.7 | 13.5 | 38.1 | **52.0** |
| *One-stage methods* |  |  |  |  |  |  |  |
| YOLOv2 [3] | DarkNet-19 | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD513 | ResNet-101-SSD | 31.2 | 50.4 | 33.3 | 10.2 | 34.5 | 49.8 |
| DSSD513 | ResNet-101-DSSD | 33.2 | 53.3 | 35.2 | 13.0 | 35.4 | 51.1 |
| RetinaNet | 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 |

Table 1. YOLOv3 is much better than SSD variants and comparable to state-of the-art models on the $AP_{50}$ metric.

## 3. How to do

The Table 1 shows the performance of YOLOv3. In terms of COCOs weird average mean AP metric it is on par with the SSD variants but is 3× faster. It is still quite a bit behind other models like RetinaNet in this metric though.

In the past YOLO struggled with small objects. However, now the author see a reversal in that trend. With the new multi-scale predictions they see YOLOv3 has relatively high $AP_S$ performance. However, it has comparatively worse performance on medium and larger size objects. More investigation is needed to get to the bottom of this.

## References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2015. 2

[2] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2016. 2

[3] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *CVPR*, pages 6517–6525, 2017. 1, 2