YOLOv3: An Incremental Improvement

Zhang, Liqiang

August 3, 2018

Abstract

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.

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 [1]. 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:

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

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. Predictions Across Scale

YOLOv3 predicts boxes at 3 different scales. This system extracts features from those scales using a similar concept to feature pyramid networks. From this base feature extractor add several convolutional layers. The last of these predicts a 3-d tensor encoding bounding box, objectness, and class predictions. In this experiments with COCO it predict 3 boxes at each scale so the tensor is $N \times N \times [3 * (4 + 1 + 80)]$ for the 4 bounding box offsets, 1 objectness prediction, and 80 class predictions.

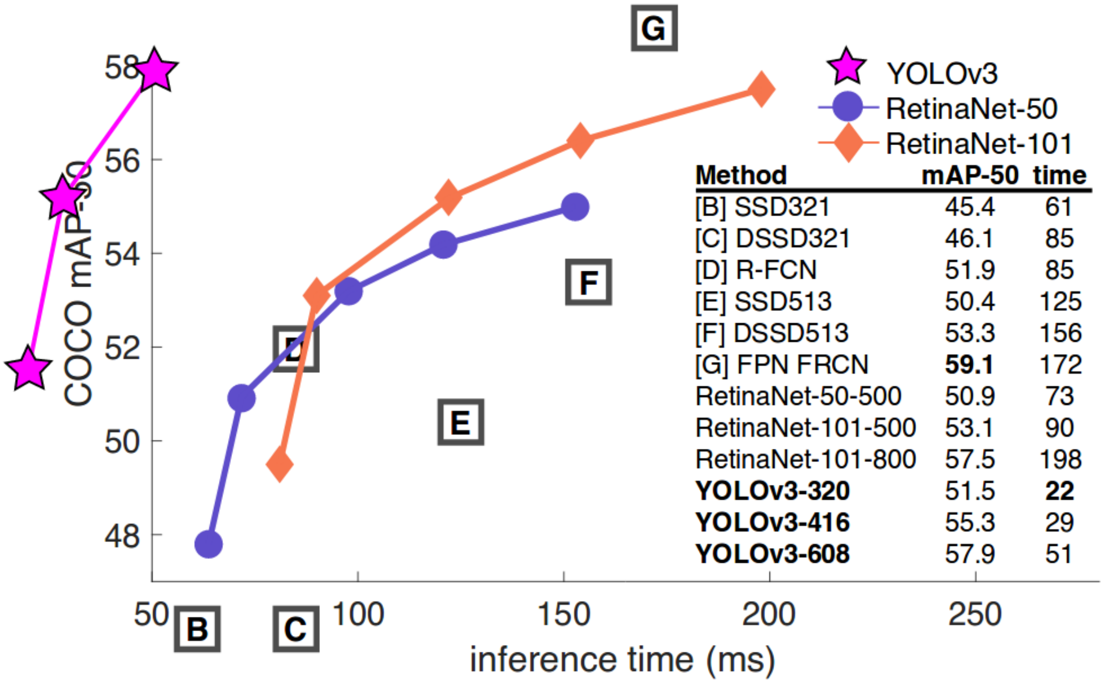
Next this team take the feature map from 2 layers previous and upsample it by 2×. They also take a feature map from earlier in the network and merge it with their upsampled features using concatenation. This method allows them to get more meaningful semantic information from the upsampled features and finer-grained information from the earlier feature map. They then add a few more convolutional layers to process this combined feature map, and eventually predict a similar tensor, although now twice the size [3]. It shows that the author predict the width and height of the box as offsets from cluster centroids. They predict the center coordinates of the box relative to the location of filter application using a sigmoid function.

3.1. 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.

4. Things that the author didn't do

Anchor box x, y offset predictions. The author tried using the normal anchor box prediction mechanism where you predict the x, y offset as a multiple of the box width or



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 1. Again adapted from the [2], this time displaying speed/accuracy tradeoff on the mAP at .5 IOU metri

height using a linear activation.

Linear x, y predictions instead of logistic. The author tried using a linear activation to directly predict the x, y offset instead of the logistic activation. This led to a couple point drop in mAP.

Focal loss. The author tried using focal loss. It dropped their mAP about 2 points. YOLOv3 may already be robust to the problem focal loss is trying to solve because it has separate objectness predictions and conditional class predictions. Thus for most examples there is no loss from the class predictions? Or something? They arent totally sure.

References

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

[2] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*. 2

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