YOLOv3: An Incremental Improvement

Zhang, Liqiang

July 27, 2018

Abstract

I continue to read this paper in this week. 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.

1. Class Prediction

In YOLOv3, each box predicts the classes the bounding box may contain using multilabel classification. It use a softmax as the author have found it is unnecessary for good performance, instead they simply use independent logistic classifiers.

This formulation helps when moving to more complex domains like the Open Images Dataset [2]. 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.

2. 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 [3] 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. In Fig. 1, 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.

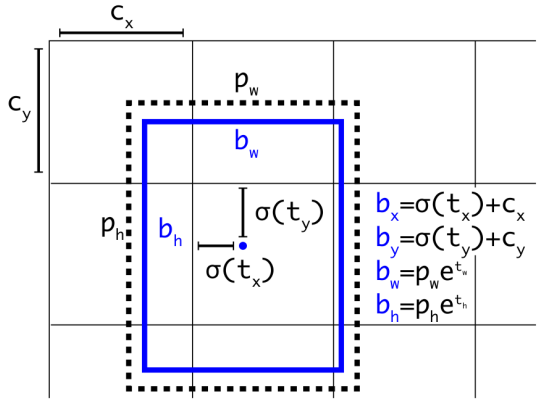

Figure 1. Bounding boxes with dimension priors and location prediction.

3. Feature Extractor

YOLOv3 use a new network for performing feature extraction. This new network is a hybrid approach between the network used in YOLOv2, Darknet-19, and that newfangled residual network stuff. The network uses successive 3 × 3 and 1 × 1 convolutional layers but now has some shortcut connections as well and is significantly larger. It has 53 convolutional layers so the author call it Darknet-53.

This new network is much more powerful than Darknet-19 but still more efficient than ResNet-101 or ResNet-152. Some ImageNet results shows in Table 1:

Each network is trained with identical settings and tested at 256 × 256, single crop accuracy. Run times are measured on a Titan X at 256 × 256. Thus Darknet-53 performs on par with state-of-the-art classifiers but with fewer floating

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [4]	74.1	91.8	7.29	1246	**171**
ResNet-101 [1]	77.1	93.7	19.7	1039	53
ResNet-152 [1]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

Table 1. **Comparison of backbones**. Accuracy, billions of operations, billion floating point operations per second, and FPS for various networks.

point operations and more speed. Darknet-53 is better than ResNet-101 and 1.5 $\times$ faster. Darknet-53 has similar performance to ResNet-152 and is 2 $\times$ 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. That's mostly because ResNets have just way too many layers and aren't very efficient.

4. Training

YOLOv3 still train on full images with no hard negative mining or any of that stuff. It use multi-scale training, lots of data augmentation, batch normalization, all the standard stuff and use the Darknet neural network framework for training and testing.

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. 1

[3] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

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