

# 课程论文题目

## YOLOv3: An Incremental Improvement

### 摘要

We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more accurate. It's still fast though, don't worry. At  $320 \times 320$  YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves 57.9 AP50 in 51 ms on a Titan X, compared to 57.5 AP50 in 198 ms by RetinaNet, similar performance but 3.8× faster.

**关键词:** YOLO; RetinaNet;

### 1 引言

从 R-CNN 到 Fast R-CNN 一直采用的思路是 proposal+分类 (proposal 提供位置信息, 分类提供类别信息) 精度已经很高, 但是速度还不行。YOLO 提供了另一种更为直接的思路: 直接在输出层回归 bounding box 的位置和 bounding box 所属的类别 (整张图作为网络的输入, 把 Object Detection 的问题转化成一个 Regression 问题)。

近年来, 深度学习模型逐渐取代传统机器视觉方法而成为目标检测领域的主流算法, 如何从图像中解析出可供计算机理解的信息, 是机器视觉的中心问题。YOLO 是一个最先进的实时物体检测系统。在 Pascal Titan X 上, 它以 30FPS 的速度处理图像, 并且在 COCO 测试开发中具有 57.9% 的 mAP。与其他探测器相比, YOLOv3 非常快速和准确。在 mAP 中, 在 0.5 IOU 下测量, YOLOv3 与焦点损失相当, 但快约 4 倍。此外, 只需更改模型的大小, 即可在速度和准确性之间轻松权衡, 无需重新训练!

### 2 目标检测之评价标准-mAP

#### 1)、AP&mAP

AP: PR 曲线下面积;

mAP: mean Average Precision, 即各类别 AP 的平均值。

#### 2)、TP FP FN TN

True Positive (TP):  $IoU > IoU_{threshold}$  (一般取 0.5) 的检测框数量 (同一 Ground Truth 只计算一次)

False Positive (FP):  $IoU \leq IoU_{threshold}$  的检测框数量, 或者是检测到同一个 GT 的多余检测框的数量

False Negative (FN): 没有检测到的 GT 的数量

True Negative (TN): 在 mAP 评价指标中不会使用到

### 3)、查准率、查全率

查准率 (Precision):  $TP / (TP + FP)$

查全率 (Recall):  $TP / (TP + FN)$

## 3 本文方法

### 3.1 工作原理概述

先前的检测系统会重新利用分类器或定位器来执行检测。他们将模型应用于多个位置和比例的图像。图像的高分区域被视为检测。

我们使用完全不同的方法。我们将单个神经网络应用于完整图像。该网络将图像划分为多个区域，并预测每个区域的边界框和概率。这些边界框由预测概率加权。与基于分类器的系统相比，我们的模型有几个优点。它在测试时查看整个图像，因此其预测由图像中的全局上下文提供信息。它还通过单个网络评估进行预测，这与 R-CNN 等系统不同，R-CNN 需要数千张图像。这使得它非常快，比 R-CNN 快 1000 倍以上，比快速 R-CNN 快 100 倍。

### 3.2 YOLO 的工作流程:

#### 1)、准备数据:

将图片缩放，划分为等分的网格，每个网格按跟 Ground Truth 的 IoU 分配到所要预测的样本。

这项挑战的目的是从许多现实场景中的视觉对象类（即未预先分割 对象）。从根本上说，这是一个监督学习问题 其中提供了标记图像的训练集，如图 1 所示。二十个对象，已选择的类包括：

- 人：人
- 动物：鸟、猫、牛、狗、马、羊
- 车辆：飞机、自行车、船、公共汽车、汽车、摩托车、火车
- 室内：瓶子、椅子、餐桌、盆栽植物、沙发、电视/显示器

1. 分类：对于二十个类中的每一个，预测测试图像中是否存在该类的示例。

2. 检测：预测边界框和标签测试图像中 20 个目标类别中的每个对象。

### The PASCAL Visual Object Classes Challenge

- **Person:** person
- **Animal:** bird, cat, cow, dog, horse, sheep
- **Vehicle:** aeroplane, bicycle, boat, bus, car, motorbike, train
- **Indoor:** bottle, chair, dining table, potted plant, sofa, tv/monitor

1. **Classification:** For each of the twenty classes, predicting presence/absence of an example of that class in the test image.

2. **Detection:** Predicting the bounding box and label of each object from the twenty target classes in the test image.

20 classes




图 1：数据集示意图

2)、卷积网络：由 GoogLeNet 更改而来，每个网格对每个类别预测一个条件概率值，并在网格基础上生成 B 个 box，每个 box 预测五个回归值，四个表征位置，第五个表征这个 box 含有物体（注意不是某一类物体）的概率和位置的准确程度（由 IoU 表示）。测试时，分数计算公式如图所示：

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

图 2：准确度计算公式

等式左边第一项由网格预测，后两项由每个 box 预测，以条件概率的方式得到每个 box 含有不同类别物体的分数。因而，卷积网络共输出的预测值个数为  $S \times S \times (B \times 5 + C)$ ，其中 S 为网格数，B 为每个网格生成 box 个数，C 为类别数。

3)、后处理：使用 NMS（Non-Maximum Suppression，非极大抑制）过滤得到最后的预测框

### 3.3 特征提取

使用 Darknet-53 进行特征提取，其性能与最先进的分类器不相上下，但浮点运算更少，速度更快，实现了每秒最高的测量浮点预算，网络结构更好地利用 GPU，使其评估更高效。结合残差思想，提取更深层次的语义信息。仍然使用连续的 3\*3 和 1\*1 的卷积层。通过上采样对三个不同尺寸做预测。采用了步长为 2 的卷积层代替 pooling 层。

	Type	Filters	Size	Output
1x	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	128 × 128
	Convolutional	64	3 × 3	
	Residual			
2x	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	64 × 64
	Convolutional	128	3 × 3	
	Residual			
8x	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	32 × 32
	Convolutional	256	3 × 3	
	Residual			
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	16 × 16
	Convolutional	512	3 × 3	
	Residual			
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	8 × 8
	Convolutional	1024	3 × 3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

图 3：Darknet-53 示意图

### 3.4 损失函数设计

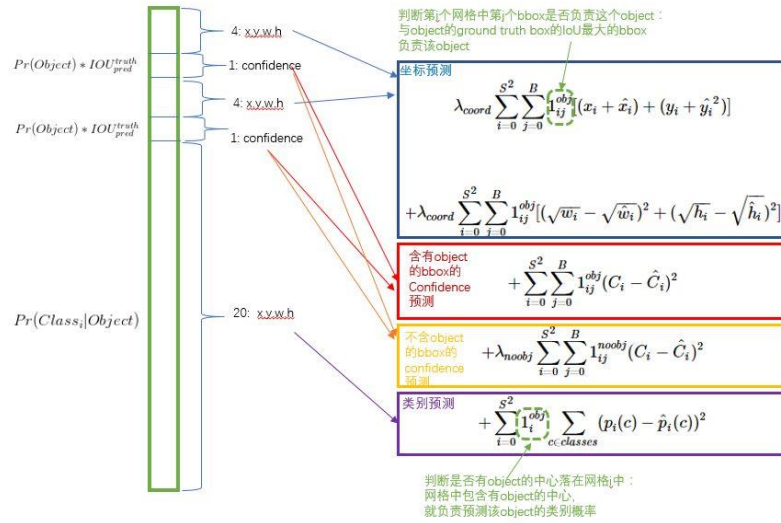


图 4：损失函数示意图

坐标误差、物体误差、类别误差。为了平衡类别不均衡和大小物体等带来的影响，损失函数中添加了权重并将长宽取根号。

## 4 复现细节

### 4.1 复现部分说明

检测图片

```
def detect_image(self, image, crop = False, count = False):
    image_shape = np.array(np.shape(image)[0:2])
    在这里将图像转换成 RGB 图像，防止灰度图在预测时报错。
    Image = vctColor(image)
    给图像增加灰条，实现不失真的 resize
    image_data = resize_image(image,
    (self.input_shape[1],self.input_shape[0]), self.letterbox_image)
    添加 batch_size 维度
    image_data =
    np.expand_dims(np.transpose(preprocess_input(np.array(image_data,
    dtype='float32'))), (2, 0, 1)), 0)
    将预测框进行堆叠，然后进行非极大抑制
    results = self.bbox_util.non_max_suppression(torch.cat(outputs,
    1), self.num_classes, self.input_shape, image_shape,
    self.letterbox_image, conf_thres = self.confidence, nms_thres =
    self.nms_iou)
```

### 4.2 实验环境搭建

1)、环境配置：(30 系显卡) windows 下

scipy==1.7.1

numpy==1.21.2

matplotlib==3.4.3

```
opencv_python==4.5.3.56
torch==1.7.1
torchvision==0.8.2
tqdm==4.62.2
Pillow==8.3.2
h5py==2.10.0
```

- 2)、Anaconda 安装
- 3)、Cudnn 和 CUDA 的下载和安装
- 4)、配置 pytorch-gpu 环境
- 5)、安装 VSCODE

## 5 实验结果分析

实验中物体的检测置信度，关系到算法的监测正确率与召回率。图中对象的置信度在输出 25 位中占固定一位，由 sigmoid 函数解码即可，解码之后数值区间在  $[0, 1]$  中。会给全部对象打标签函数以及损失函数的计算。推理时，选取一个置信度阈值，过滤掉低阈值 box，再经过非极大值抑制，就可以输出整个网络的预测结果了。实验结果如图 5 所示。

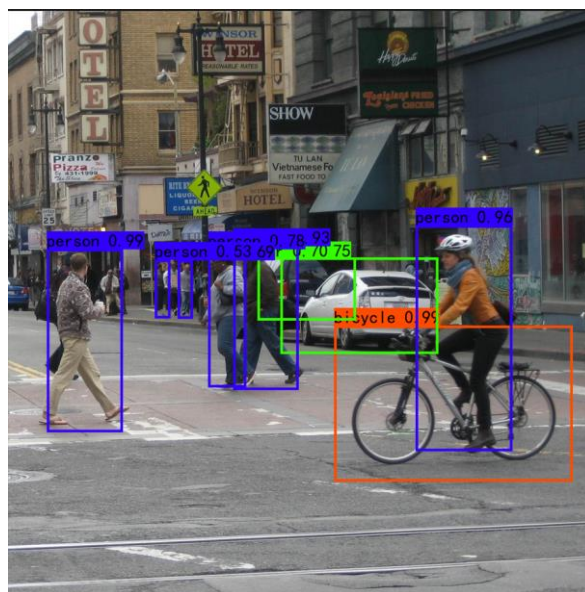


图 5：实验结果示意图

## 6 总结与展望

YOLOv3 包括识别给定照片中的一个或多个目标的存在、位置和类别，相比较 YOLOv2，第一个改进是网络结构的改变，DarkNet 模型搭建之后，使用预先训练好的权重文件来进行预测，在复现过程中，尝试过训练自己的目标检测模型，需要注意的是，训练前仔细检查自己的格式是否满足要求，该库要求数据集格式为 VOC 格式，需要准备好的内容有输入图片和标签等；损失值的大小用于判断是否收敛，比较重要的是有收敛的趋势，即验证集损失不断下降，如果验证集损失基本上不改变的话，模型基本上就收敛了。损失值的具体大小并没有什么意义，大和小只在损失的计算方式，并不是接近于 0 才好。

通过本次实验，熟悉课题相关的内容，很好地了解了实验基线，流程，为以

后科研生涯开启了初始探索之旅，感谢老师的指导，希望以后能有更多的科研交流机会。

## 参考文献

- [1] Analogy. *Wikipedia*, Mar 2018. 1
- [2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6
- [3] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 3
- [4] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. Iqa: Visual question answering in interactive environments. *arXiv preprint arXiv:1712.03316*, 2017. 1
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3
- [6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. 3
- [7] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from <https://github.com/openimages>*, 2017. 2
- [8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 2, 3
- [9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. ‘ Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 1, 3, 4
- [10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ‘ mon objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.- Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3
- [12] I. Newton. *Philosophiae naturalis principia mathematica*. William Dawson & Sons Ltd., London, 1687. 1
- [13] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein. Animal population censusing at scale with citizen science and photographic identification. 2017. 4
- [14] J. Redmon. Darknet: Open source neural networks in c. <http://pjreddie.com/darknet/>, 2013–2016. 3
- [15] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6517–6525. IEEE, 2017. 1, 2, 3
- [16] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 4
- [17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2
- [18] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2121–2131, 2015. 4
- [19] M. Scott. Smart camera gimbal bot scanlime:027, Dec 2017.4

- [20] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016. 3
- [21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017. 3