期中课程报告

27组 组员：吕浩铭、谢俊勋、徐聪

一．

1 CIFAR100数据集

CIFAR100数据集包含100个类，每个类包含600张大小为32×32的RGB图像，其中500张为训练集，100张为测试集。整体数据集共包含60000张图像。下图为数据集官网给与的类别名称。

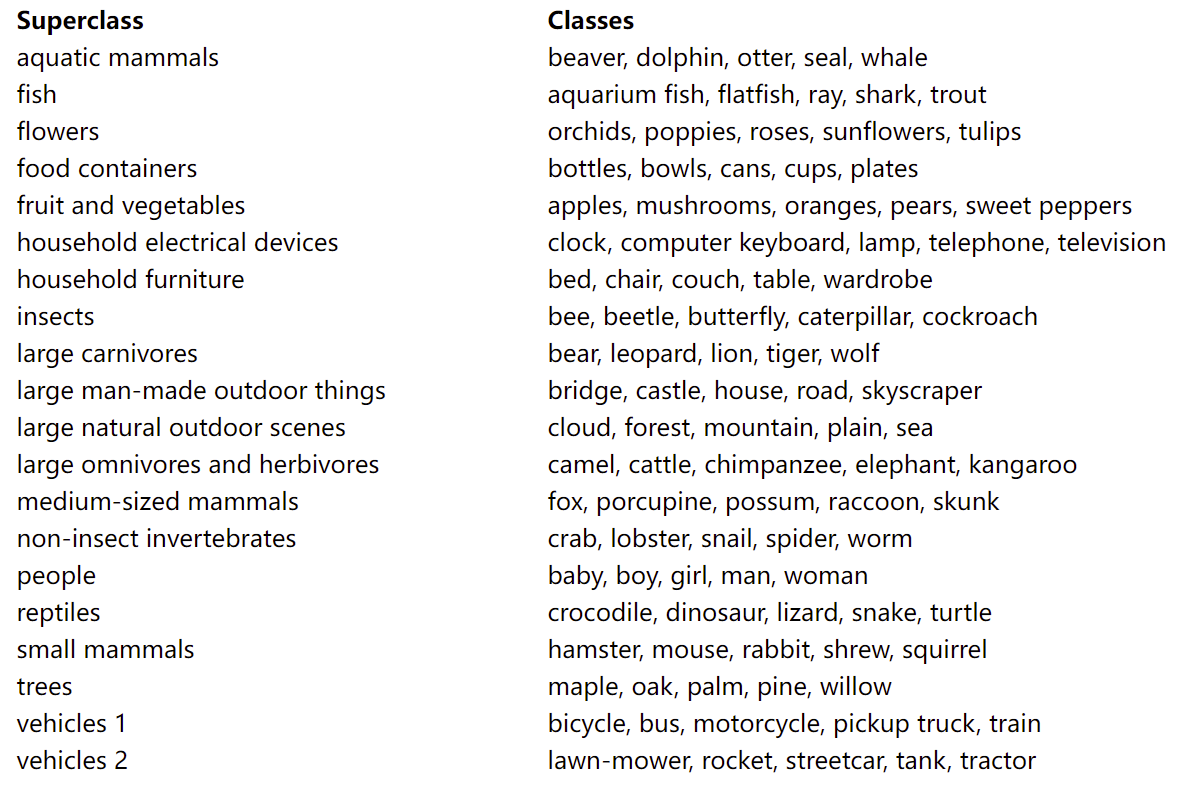


图1 CIFAR100 数据集

2 模型结构

2.1 残差网络

残差网络使用函数表示为 。残差网络结构可以避免当网络层数较深时引起的梯度消失现象。残差结构包含标准残差block以及bottle block，其中Resnet50采用bottle block作为基本block。

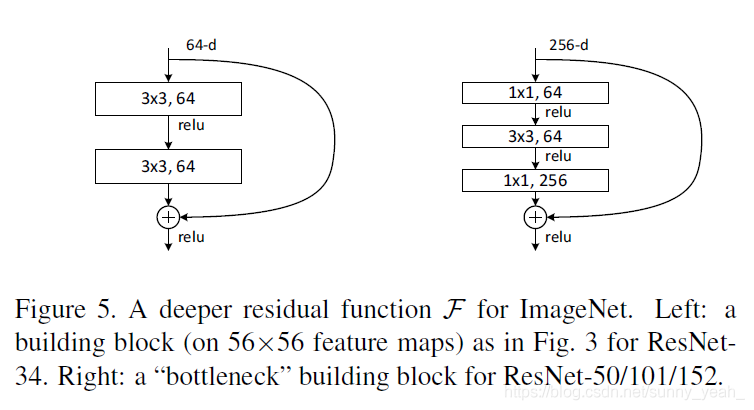


图 2 block(左) 和 bottle block(右)

2.2 Resnet50

模型采用Resnet-50作为baseline，由于训练样本图片大小为32×32，因此将Resnet-50第一层kernel\_size = 7, stride = 2的卷积层删除，并移除之后的3×3 max pool层。使用kernel\_size = 3, stride = 1的卷积层代替。

经过卷积神经网络获得的特征图展平之后输入到全连接层预测最终的标签。

3 数据增强

3.1 CutMix

CutMix操作将一个batchsize中的原始图像进行随机裁剪，并将随机裁剪的空缺使用另一张图片进行补全。其中真实标签的混合同裁剪补丁的大小成正比。

和y表示一张训练图片以及其对应的标签。CutMix是将两张图片以及进行混合操作。可以表示为：

其中为mask矩阵。

3.2 CutOut

CutOut操作直接将训练样本剪切掉一个区域，即模拟遮挡。

在论文中，作者实验发现CutOut操作剪切的形状对实验结果影响不大，因此采用正方形的剪切方式。剪切的尺寸大小为原图像的一半。

3.3 Mixup

Mixup操作将两张图像按比例融合。同样假设和y表示一张训练图片以及其对应的标签。融合两张图像表示为：

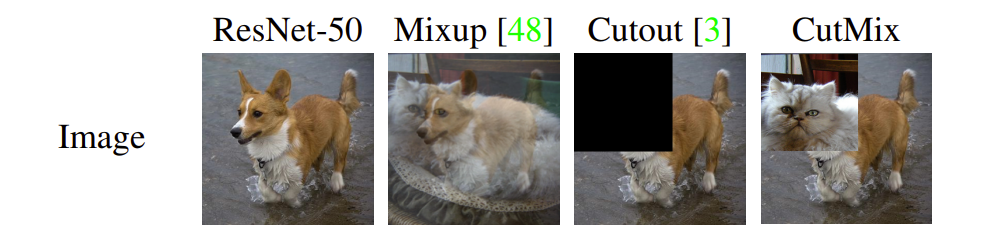


图 3 数据增强效果展示

4 训练设置

设备：RTX3090 ×1

Batchsize：256

Learning-rate：0.2

学习率衰减：第60,120,160个epoch时学习率设置为上个阶段的1/5

优化器：SGD

Epoch：200

损失函数采用十字交叉熵损失函数，计算方式为：

针对使用不同数据增强的模型训练，例如CutMix和Mixup，计算损失时需要根据lam值进行加权计算。

5 实验结果

5.1 数据增强可视化

针对三种数据增强方式，在训练集上可视化数据增强效果。

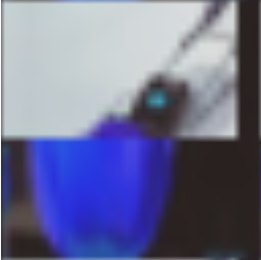


图4 CutMix可视化



图5 CutOut可视化



图5 Mixup可视化

5.2 实验结果

分别训练不使用数据增强和三种数据增强方式的模型，训练结果如下表所示。当使用数据增强时，模型性能均有所提升。

|  |  |  |  |
| --- | --- | --- | --- |
| Data\_Augment | Backbone | Accuracy |  |
| None | Resnet50 | 77.83 | - |
| CutMix | Resnet50 | 80.36 | + 2.53 |
| CutOut | Resnet50 | 78.47 | + 0.64 |
| Mixup | Resnet50 | 79.26 | + 1.43 |

表1 实验结果对比

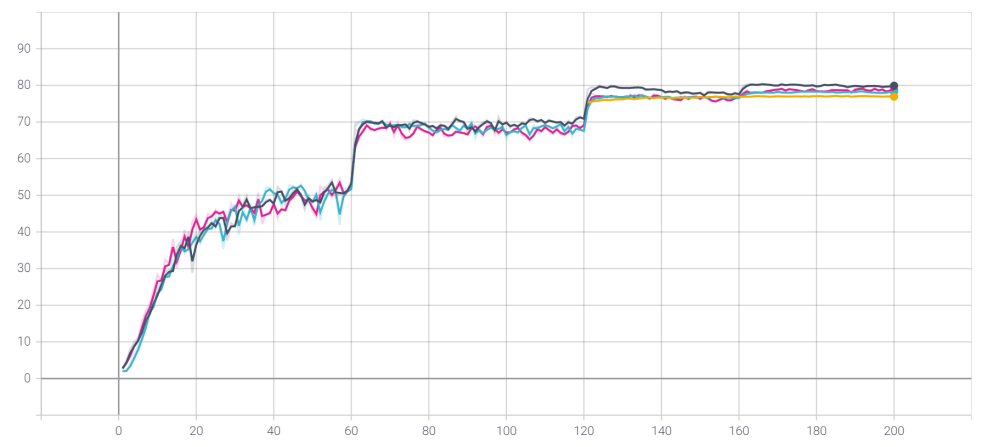


图7 模型Acc曲线 base(橙色) cutmix(黑色) cutout(蓝色) mixup(粉色)

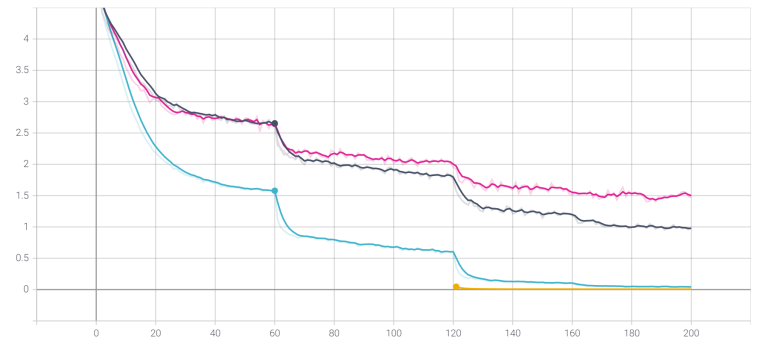


图8 train loss

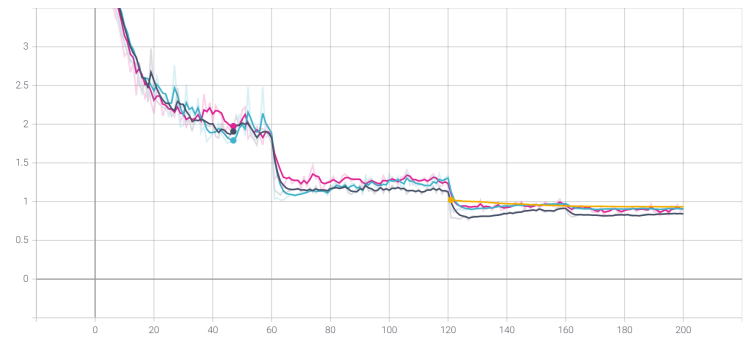


图9 val loss

二．

1 VOC数据集

VOC2007中包含9963张标注过的图片， 由train/val/test三部分组成， 共标注出24,640个物体。对于检测任务，VOC2012的trainval/test包含08-11年的所有对应图片。 trainval有11540张图片共27450个物体。对于分割任务， VOC2012的trainval包含07-11年的所有对应图片， test只包含08-11。trainval有 2913张图片共6929个物体。

PASCAL VOC2007数据集总共分 4 个大类：vehicle、household、animal、person，总共 20 个小类（加背景 21 类），预测的时候是只输出下图中黑色粗体的类别：

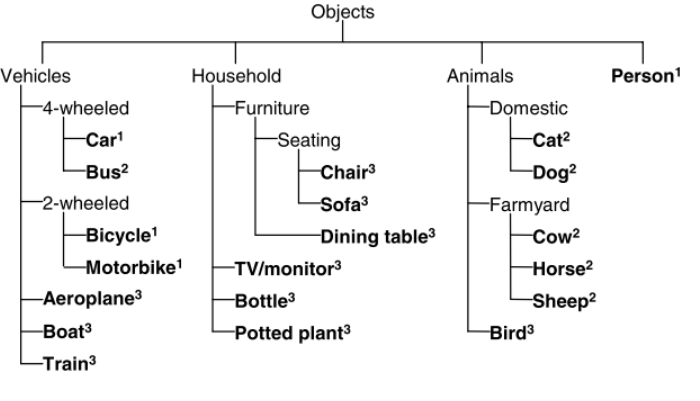


图9

2 模型结构

2.1 Faster-RCNN

2.1.1 残差网络

残差网络使用函数表示为 。残差网络结构可以避免当网络层数较深时引起的梯度消失现象。残差结构包含标准残差block以及bottle block，其中Resnet50采用bottle block作为基本block。

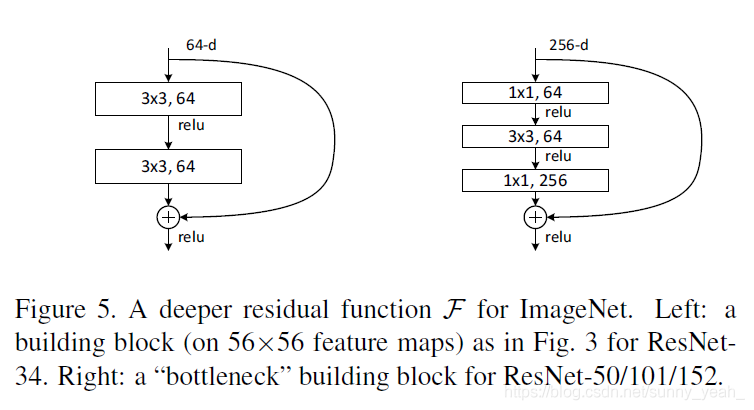


图 10 block(左) 和 bottle block(右)

2.1.2 Backbone-Resnet50

模型采用Resnet-50作为baseline，由于训练样本图片大小为32×32，因此将Resnet-50第一层kernel\_size = 7, stride = 2的卷积层删除，并移除之后的3×3 max pool层。使用kernel\_size = 3, stride = 1的卷积层代替。

经过卷积神经网络获得的特征图展平之后输入到全连接层预测最终的标签。

2.1.3 RPN网络

Faster RCNN中最复杂的一部分. 将通过ResNet50得到的图片特征喂入RPN(Region Proposal Network)网络，在原文中使用的是ZF model中，其Conv Layers中最后的conv5层num\_output=256，对应生成256张特征图，所以相当于feature map每个点都是256-dimensions。

在conv5之后，做了rpn\_conv/3x3卷积且num\_output=256，相当于每个点又融合了周围3x3的空间信息，同时256-d不变。

假设在conv5 feature map中每个点上有k个anchor（默认k=9），而每个anhcor要分positive和negative，所以每个点由256d feature转化为cls=2•k scores；而每个anchor都有(x, y, w, h)对应4个偏移量，所以reg=4•k coordinates。

然而全部anchors拿去训练太多了，训练程序会在合适的anchors中随机选取128个postive anchors+128个negative anchors进行训练得到proposals RPN网络结构如下图11所示:

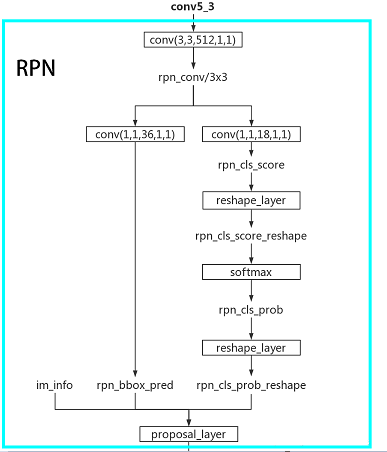


图11 RPN网络

## 2.1.4 通过RoI Pooling层得到综合的proposals特征

将 ROI 和 conv5\_3的输出 喂给ROI Pool 得到 ‘pool\_5’ , 关于ROI Pool的输入,尺寸是任意大小的,但输出的特征都是长度固定的, 如下图12所示：

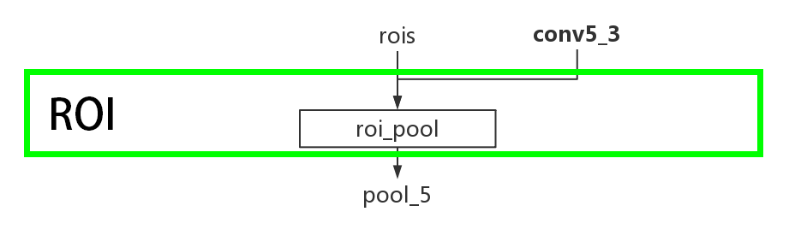


图12

## 2.1.5 根据poposals特征进行框回归和物体分类

大致步骤如下图13，具体即使当前的feature map经过两个全连接(FC)层，之后再分别进入两个不同的全连接层，进行分类和bounding box的预测:

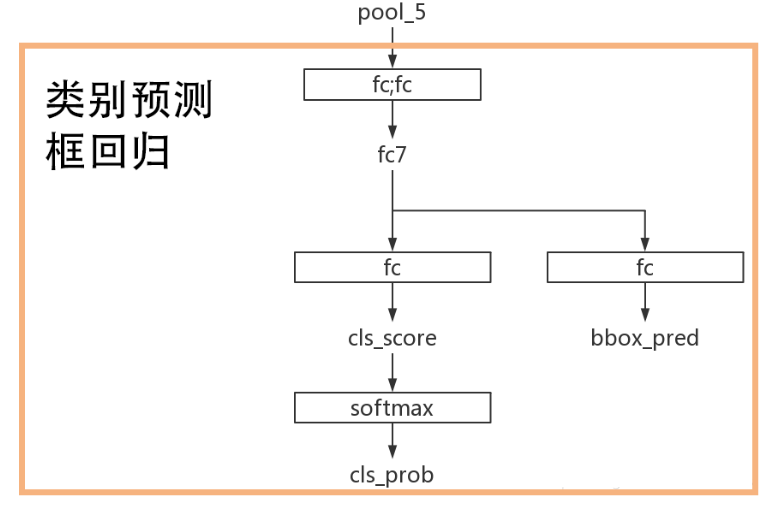


图13

**2.2 YOLOv3**

2.2.1 Backbone- darknet-53

darknet-53借用了resnet的思想，在网络中加入了残差模块，这样有利于解决深层次网络的梯度问题，每个残差模块由两个卷积层和一个shortcut connections，1,2,8,8,4代表有几个重复的残差模块，整个v3结构里面，没有池化层和全连接层，网络的下采样是通过设置卷积的stride为2来达到的，每当通过这个卷积层之后图像的尺寸就会减小到一半。而每个卷积层的实现又是包含卷积+BN+Leaky relu,每个残差模块之后又要加上一个zero padding,具体实现可以参考图14。

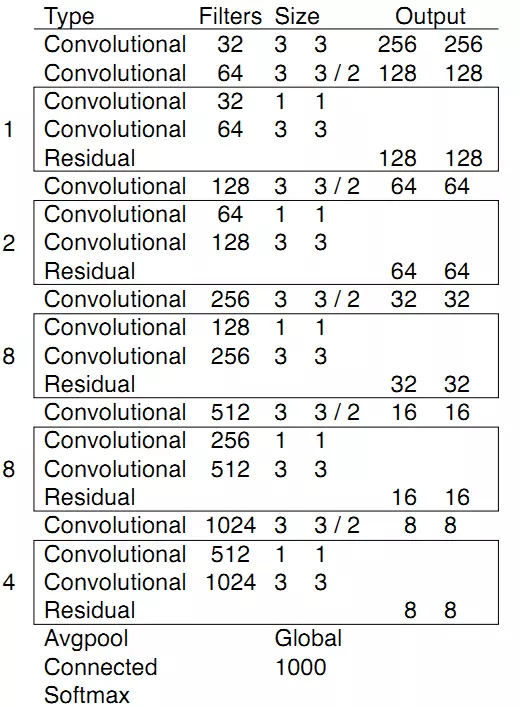


图14：darknet-53网络结构

**2.2.2 输入与输出**

输入416\*416\*3的图像，通过darknet网络得到三种不同尺度的预测结果，每个尺度都对应N个通道，包含着预测的信息：每个网格每个尺寸的anchors的预测结果。YOLOv3共有13\*13\*3 +26\*26\*3+52\*52\*3个预测。每个预测对应85维，分别是4（坐标值）、1（置信度分数）、80（coco类别数）。

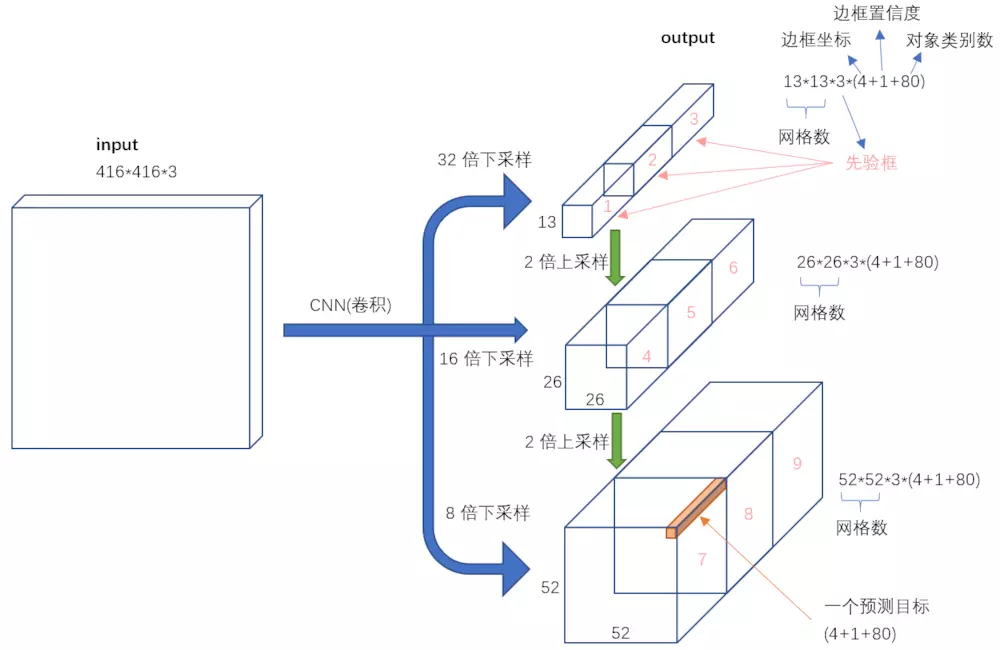


图 15 输入与输出展示图

**2.2.3 多尺度检测**

YOLOv3采用了3个不同尺度的特征图来进行对象检测。能够检测的到更加细粒度的特征。在第79层之后经过几个卷积操作得到的是1/32 （13\*13） 的预测结果，下采样倍数高，这里特征图的感受野比较大，因此适合检测图像中尺寸比较大的对象。然后这个结果通过上采样与第61层的结果进行concat,再经过几个卷积操作得到1/16的预测结果；它具有中等尺度的感受野，适合检测中等尺度的对象。91层的结果经过上采样之后在于第36层的结果进行concat，经过几个卷积操作之后得到的是1/8的结果，它的感受野最小，适合检测小尺寸的对象。

**2.2.4 loss function**

对于回归预测的部分是采用多个mse均方差相加来进行的，对于分类部分和置信度是采用K.binary\_crossentropy来进行的，最后把两种Loss相加得出最终的loss。

3 训练设置

3.1 Faster-RCNN

设备：RTX3090 ×1

Batchsize：16

Learning-rate：0.001

学习率衰减：每100个epoch时学习率设置为上个阶段的1/10

优化器：SGD

Epoch：1000

NMS Thresh:0.5

损失函数为分类损失与回归损失的总和，计算方式为：



3.2 YOLO-V3

设备：RTX3090 ×1

Batchsize：64

Learning-rate：0.001

Momentum：0,9

Decay：0.0005

学习率衰减：在第400和500个epoch时学习率设置为上个阶段的1/10

优化器：SGD

Anchor: 9个

Truth Thresh：0.5

Epoch：600

4 实验结果

4.1 Faster-RCNN

使用tensorboard可视化验证集图：



图17

Train Loss曲线：

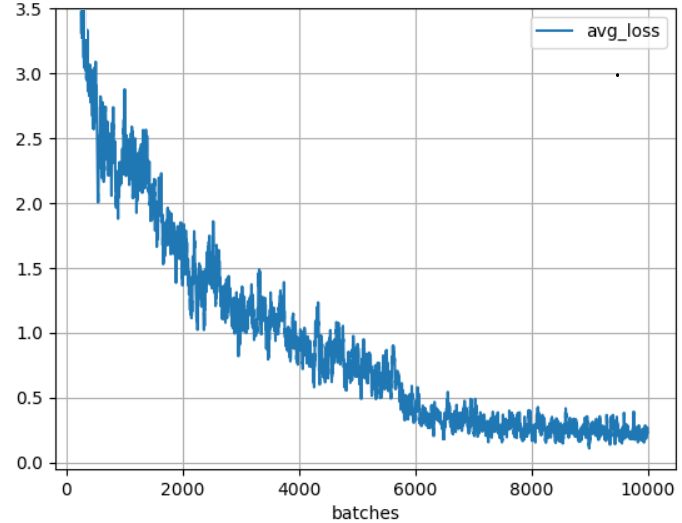


图18

四张测试图像上可视化Faster R-CNN第一阶段的proposal box如下图所示：

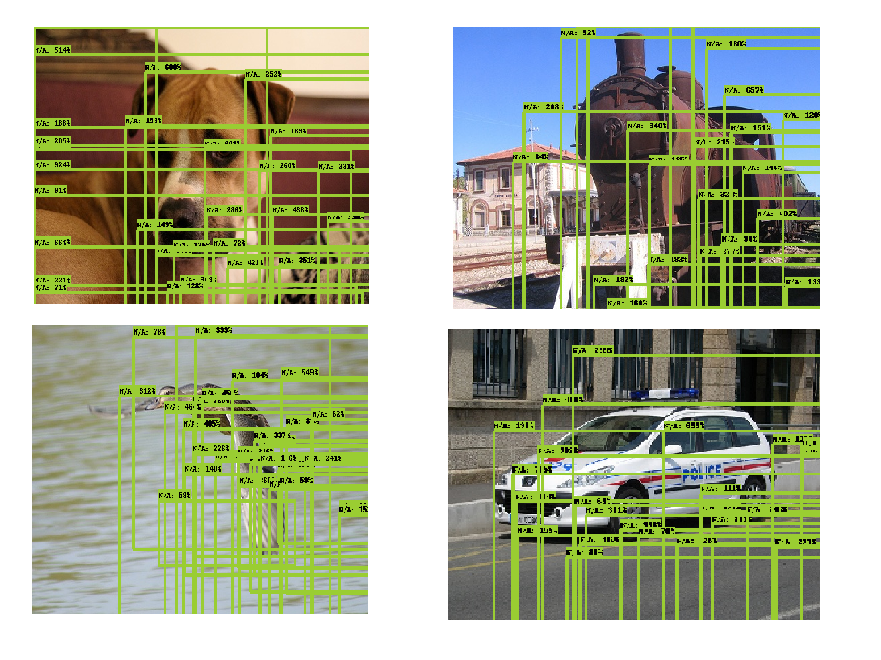


图19

三张非VOC上图像demo结果：

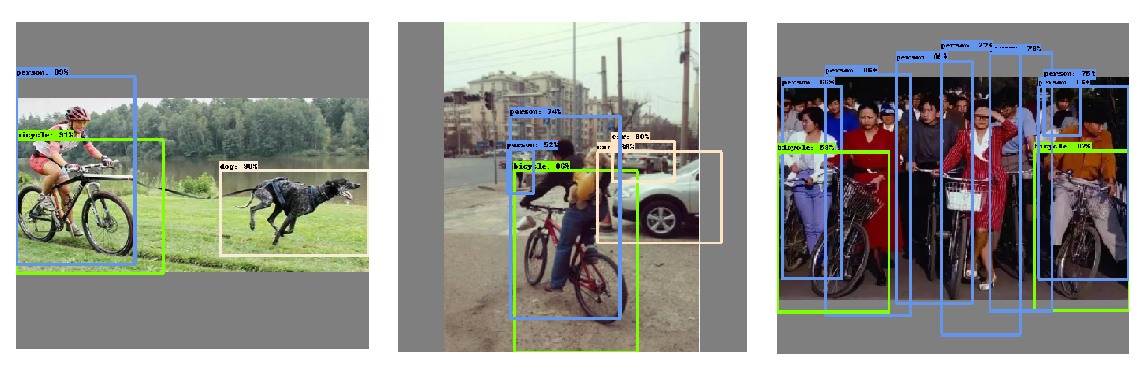


图20

4.2 YOLO-V3

三张非VOC上图像demo结果：

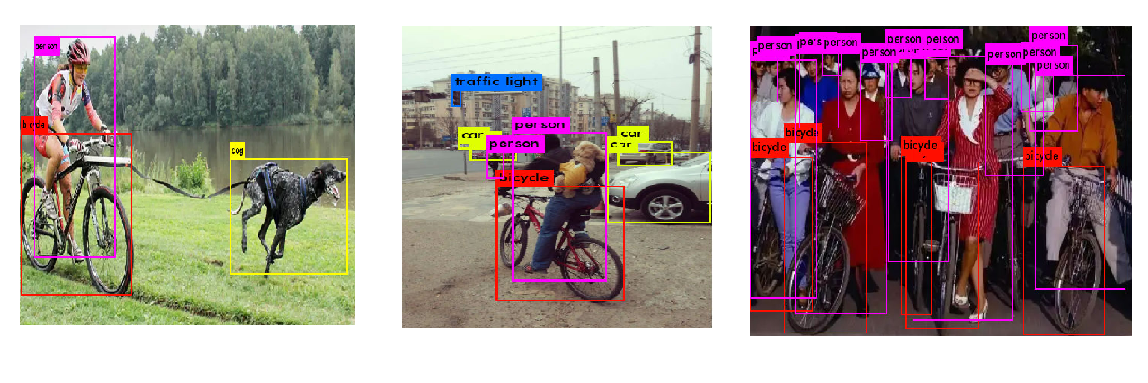


图21

YOLO-V3与Faster RCNN对比结果来看，总的来说YOLO-V3效果更佳。从视觉角度看，YOLOV3任何分辨率的图像，输出仍然填充了整张图像，且anchor不会超出图像范围；其次在速度方面，YOLOV3达到了26 FPS，略高于Faster RCNN的20 FPS；但是从实验结果来说，Faster R-CNN 的 mAP 值比 YOLOv3 的 mAP 值高 0.68%，略高于YOLOV3。

使用tensorboard可视化验证集图：

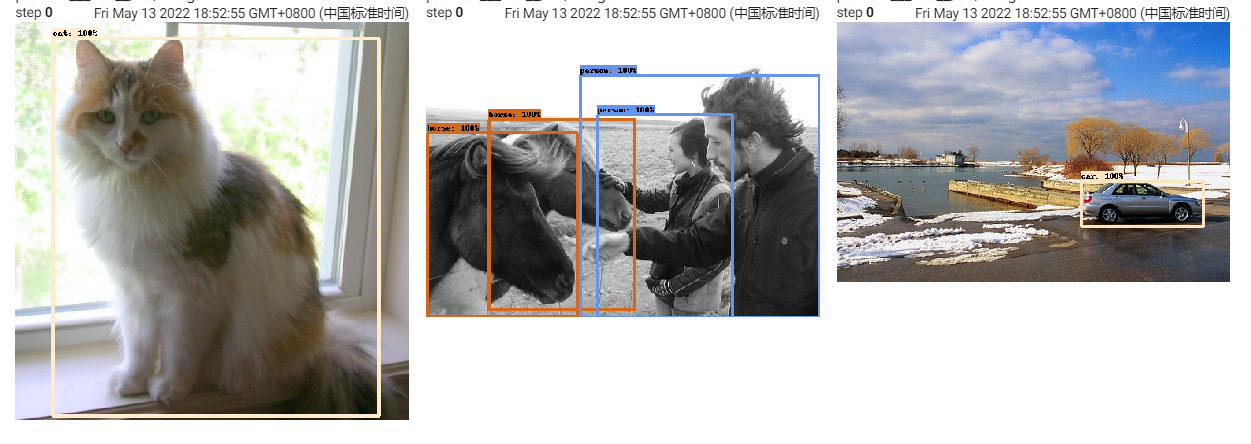


图22

YOLO-V3 Train Loss曲线：

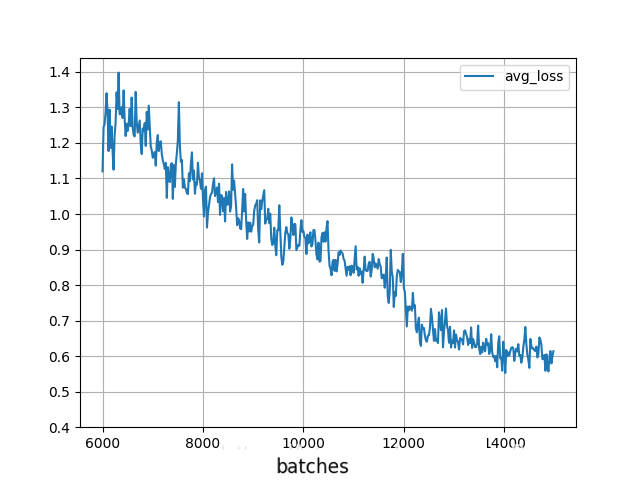


图21

**参考文献**

1. [Sangdoo Yun](https://arxiv.org/search/cs?searchtype=author&query=Yun,+S), [Dongyoon Han](https://arxiv.org/search/cs?searchtype=author&query=Han,+D), “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features”, 2019
2. [Terrance DeVries](https://arxiv.org/search/cs?searchtype=author&query=DeVries,+T), [Graham W. Taylor](https://arxiv.org/search/cs?searchtype=author&query=Taylor,+G+W), ”Improved Regularization of Convolutional Neural Networks with Cutout”, 2017
3. [Hongyi Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang,+H), [Moustapha Cisse](https://arxiv.org/search/cs?searchtype=author&query=Cisse,+M), “mixup: Beyond Empirical Risk Minimization”, 2017
4. [Kaiming He](https://arxiv.org/search/cs?searchtype=author&query=He,+K), [Xiangyu Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang,+X), “Deep Residual Learning for Image Recognition”, 2015
5. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.
6. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), pages 91–99, 2015.
7. Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.