下次尝试不加载预训练模型从头开始训练

**：利用 softer nms 实现相同类的位置修正**

**平均情况下608训练模型最佳 320预测效果最佳，但最佳状态不一定，训练到多尺度第二个608时效果达到最优**

# 服务器开始训练时间20181031

**对应GPU：双1080Ti GPU**

训练文件：train\_mutiscale\_mutiGPU20181031

重新生成数据：共训练29个类，train:6275 ,val: 946 , test: 612

**修改数据增强模块，根据输出的图像尺寸调整了一些参数**

Batchsize：第一阶段 32 第二阶段：32 16 8

Epoch 第一阶段：**0/40 数据不增强**

第二阶段：muti\_scales = [[(416, 416), **True**,(40,70)], [(608, 608), **True**,(45,75)],[(320,320), **False**,(50,80)], [(416, 416), **True**,(55,85)],[(320,320), **True**,(60,90)],[(608,608), **False**,(65,125)]]

object\_classes：Data/ object\_classes2018\_10\_29.txt

最大mAP： 到 平均mAP：

# 服务器开始训练时间20181029

**对应GPU：双1080Ti GPU**

训练文件：train\_mutiscale\_mutiGPU20181029

重新生成数据：共训练29个类，train:6275 ,val: 946 , test: 612

**不加载预训练模型**

Batchsize：第一阶段 16 第二阶段：32 16 8

Epoch 第一阶段：**0/40 数据不增强**

第二阶段：muti\_scales = [[(416, 416), **True**,(40,70)], [(608, 608), **True**,(45,75)],[(320,320), **False**,(50,80)], [(416, 416), **True**,(55,85)],[(320,320), **True**,(60,90)],[(608,608), **False**,(65,125)]]

object\_classes：Data/ object\_classes2018\_10\_29.txt

最大mAP： 到 平均mAP：

# 服务器开始训练时间20181028

**对应GPU：双1080Ti GPU**

训练文件：train\_mutiscale\_mutiGPU20181026

重新生成数据：共训练29个类，train:6159 ,val: 929 , test: 600

**预加载：logs/20181026/yolov3\_20181026\_0.h5**

Batchsize：第一阶段 32 第二阶段：32 16 8

Epoch 第一阶段：**35/35（利用之前结果直接训练）** 第二阶段：50/80

muti\_scales=[ (416,416),(608,608),(320,320),(416,416),(608,608),(320,320)，（608,608）]

object\_classes：Data/ object\_classes2018\_10\_24.txt

最大mAP：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| yolov3\_20181028\_6.h5 | 320 | 0.2 | 0.05 | FALSE | 99.235 |

到 平均mAP：97.2177143

**开启数据增强：速度分析：**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **608** | **416** | **320** |
| **不增强** | **12.56** | **6.11** | **4** |
| **增强** | **18.44** | **7.83** | **5.33** |
| **速度变慢** | **1.46** | **1.28** | **1.33** |

|  |
| --- |
| **20181028数据增强后****train比 val比0.8%； val比test好0.007%个点** |

**20181026数据不增强** **train比 val比好2.6% ；val比test好0.05%个点**

**增强比不增强 test好6%，val好6% train好4.3**

精度分析具体见《20181026-20181028数据增强不增强，训练集、测试集、验证集情况汇总.xlsx》

# 服务器开始训练时间20181026

**对应GPU：双1080Ti GPU**

训练文件：train\_mutiscale\_mutiGPU20181026

重新生成数据：共训练29个类，train:6159 ,val: 929 , test: 600

**预加载：logs/20181024/****yolov3\_20181024\_5.h5**

Batchsize：第一阶段 32 第二阶段：32 16 8

Epoch 第一阶段：**0/35（减少）** 第二阶段：50/80

muti\_scales=[ (416,416),(608,608),(320,320),(416,416),(608,608),(320,320)，（608,608）]

object\_classes：Data/ object\_classes2018\_10\_24.txt

修改损失函数：**将分类的损失系数还为1.5**

最大mAP：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| yolov3\_20181026\_3.h5 | 416 | 0.2 | 0.03 | FALSE | 99.313 |

到 平均mAP：91.11

**关闭数据增强：发现关闭这个选项后训练速度大大变快，说明CPU准备数据的时间严重拖慢了训练时间，但训练结果却不晓得 第一阶段速度为3.7倍 第二阶段：416阶段 速度为1.5 618阶段提速：1.28**

训练结果：

# 服务器开始训练时间20181024

**对应GPU：双1080Ti GPU**

训练文件：train\_mutiscale\_mutiGPU20181024

重新生成数据：共训练25个类，train:5157 ,val: 780 , test: 502

Batchsize：第一阶段 32 第二阶段：32 16 8

Epoch 第一阶段：0/50 第二阶段：50/80

muti\_scales=[ (416,416),(608,608),(320,320),(416,416),(608,608),(320,320)，（608,608）]

object\_classes：Data/ object\_classes2018\_10\_24.txt

修改损失函数：**将分类的损失系数还为1.5**

anchor保持不变：model\_data: yolov3\_anchors\_416\_20181020.txt

最大mAP：98.827 平均mAP：

训练结果： mAP测试结果20181020-服务器

其中台式机：最佳mAP：97.445

# 服务器开始训练时间20181022

对应GPU：1080Ti

训练文件：train\_mutiscale\_\_20181022服务器

重新生成数据：共训练21个类，train:4225 ,val: 637 , test: 408

Batchsize：第一阶段 64 第二阶段：24 12 6

Epoch 第一阶段：0/60 第二阶段：50/80

muti\_scales=[ (416,416),(608,608),(320,320),(416,416),(608,608),(320,320)，（608,608）]

object\_classes：Data/ object\_classes2018\_10\_20.txt

修改损失函数：**将分类的损失系数由1改为1.5**

anchor保持不变：model\_data: yolov3\_anchors\_416\_20181020.txt

最大mAP：98.6 到 平均mAP：94.9

训练时间： mAP测试结果20181020-服务器

**改进策略2：利用soft nms 实现非最大值抑制**

# 服务器开始训练时间20181020

对应GPU：1080Ti

训练文件：train\_mutiscale\_1080Ti\_20181020

重新生成数据：共训练21个类，train:4821 ,val: 732 , test: 469

Batchsize：第一阶段 64 第二阶段：24 12 6

Epoch 第一阶段：0/60 第二阶段：50/80（改回来了，感觉大了容易过拟合）

muti\_scales=[ (416,416),(608,608),(320,320),(416,416),(608,608),(288,288)]

object\_classes：Data/ object\_classes2018\_10\_20.txt

修改了数据增强模块：减少了尺度变换范围从0.25到2 变为0.5到1.3，增加高斯模糊

自己生成anchor：model\_data: yolov3\_anchors\_416\_20181020.txt

最大mAP：92.986 到96.326 平均mAP：85.7-90.3（修改了mAP计算公式）训练时间： 26h

mAP测试结果20181020-服务器正确

# 台式机开始训练时间20181020

对应GPU：1060

训练文件：train\_mutiscale\_1080Ti\_20181020

重新生成数据：共训练21个类，3975 train 596验证 397测试

Batchsize：第一阶段 64 第二阶段：24 12 6

Epoch 第一阶段：0/60 第二阶段：50/80

muti\_scales=[ (416,416),(608,608),(320,320),(416,416),(608,608),(288,288)]

object\_classes：Data/ object\_classes2018\_10\_20.txt

修改了数据增强模块：减少了尺度变换范围从0.25到2 变为0.5到1.3，增加高斯模糊

自己生成anchor：model\_data: yolov3\_tiny\_anchors\_416\_20181020.txt

最大mAP：91.424 平均mAP：75.78

训练时间： mAP测试结果20181020- tiny

# 服务器开始训练时间20181018

对应GPU：1080Ti

训练文件：train\_mutiscale\_1080Ti\_20181018

重新生成数据：共训练21个类，3975 train 596验证 397测试

Batchsize：第一阶段 64 第二阶段：24 12 6

Epoch 第一阶段：0/60 第二阶段：50/100

muti\_scales=[(288,288),(416,416),(608,608),(256,256),(288,288),(416,416),(608,608),(288,288)]

object\_classes：Data/ object\_classes2018\_10\_18.txt

最大mAP：92 平均mAP：84.46

训练时间： mAP测试结果20181018-服务器

# 台式机开始训练时间20181018（tiny）

对应GPU：1060

训练文件：train\_tiny\_mutiscale20181018

重新生成数据：共训练21个类，3975 train 596验证 397测试

Batchsize：第一阶段 32 第二阶段：24 12 6

Epoch 第一阶段：0/60 第二阶段：50/80 共训练4个size(288,288),(416,416),(608,608),(288,288) 第四个只训练了一半

最大mAP：91.92 平均mAP：83

object\_classes：20类 Data/object\_classes.txt

训练结果：mAP测试结果20181018\_台式机tiny.xls

# 台式机开始训练时间20181016

对应GPU：1060

Batchsize：16 8 4 2

Epoch 第一阶段：48/50 第二阶段：50/80 共训练6个size 288 416 608 288 416 608

最大mAP：96.98

平均mAP：87.3

训练结果：mAP测试结果20181016\_台式机.xls

# 服务器开始训练时间20181016

对应GPU：1080Ti

Batchsize：第一阶段 32 第二阶段：24 12 6

Epoch 第一阶段：0/50 第二阶段：50/80 共训练6个size同上

最大mAP：93.02

平均mAP：86.2

训练结果：mAP测试结果20181016\_服务器.xls

1、2对比初始化不一样 batchsize不一样，最体来说台式机训练的效果要好一些

# 台式机开始训练时间20181017

对应GPU：1060

训练文件：train\_mutiscale\_1060\_20181017

重新生成数据：共训练21个类，3975 train 596验证 397测试

Batchsize：第一阶段 16 第二阶段：8 4 2

Epoch 第一阶段：0/60 第二阶段：50/80 共训练6个size288 416 608 288 416 608

最大mAP：结果还可以

训练结果：mAP测试结果20181016\_服务器.xls

训练时间：3.9 2.5 5.2

1、2对比初始化不一样 batchsize不一样

下次改进思路：第二阶段的epoch可以设大点，这样使得大多数训练自动停下来，使得在这个尺度训练效果最优