

一、赛前准备

比较重要的几点：

- (1) 重要时间点。
- (2) 使用“计图”深度学习框架。
- (3) 除主办方提供的数据集和ImageNet预训练模型外，参赛选手不能使用任何其他标注数据。
- (4) **官方有计图教程和赛题baseline。**
- (5) A榜提交文件包含提交结果json文件以及代码文件，压缩包大小控制在50MB以内。
- (6) 交通标志在图片中占比较小，图像模糊，交通标志类别复杂，甚至某些交通标志含义与所在场景相关。

二、赛题说明

(1) 数据集：TT100K-2021 大小：18G。共包含100K张图片，其中有标注的图像有10592张，共包含232类交通标志。

(2) 目的：本赛题将提供若干张测试图片，参赛选手需要对测试图片中出现的交通标志进行检测与分类，输出每张图片中交通标志的包围框位置及其类别。（**目标检测**）

(3) 评测指标

本赛题将以mean Average Precision (mAP)作为交通标志检测结果的评测指标，其中mAP的计算方式参考自Pascal VOC 2010 Challenge。本次比赛按照包围框的面积计算 $[16^2, 32^2)$, $[32^2, 96^2)$, $[96^2, +\infty)$, $[16^2, +\infty)$ 四个不同范围的mAP，最终以 $[16^2, +\infty)$ 范围的mAP50作为排行榜指标，我们在下方提供了标准评测脚本。

部分名词解释：

1. IOU: 检测框和真实框的交集与两者并集的比值。
2. mAP50: 与真实框的IOU>0.5的检测框为True Positive, 否则为False Positive, 以此来计算每一类的Precision和Recall, 然后以VOC 2010 Challenge的计算方式来计算AP, 最终计算所有类的mAP。

(4) 提交结果是json文件。

(5) 交通标志图例可以使用？？？

(6) 交通类别分布不均。

(7) 交通标志缩略图。

(8) 提供代码段（可视化与验证集评估）。一会可以试一下。

问题：

- (1) 数据集标注图像仅占1/10，其他无标注图像是做什么的？不知道是否有论文介绍该数据集？
- (2) 交通标志类别数232，且分布不均，与HICODET相似？可以统计一下各类别对应的图像与实例数。
- (3) 验证集评估test_result.pkl格式与提交结果格式不一样？

三、baseline学习

问题思考：

- (1) 交通标志贴图的作用：augmentation.py数据增强，解决类别不平衡问题，例如粘贴较少类别。

水平翻转或者旋转可能会对一些交通标志存在影响，例如向左转翻转后变成向右转。

(2) baseline模型是Faster RCNN，backbone是ResNet50。

图片尺寸：2048*2048，太大了，对显存有影响，裁剪或者放缩？

batchsize：1，配合图片尺寸、学习率调节？

(3) targets为空怎么解决，图片中根本没有targets？避免loss为NAN。

(4) **PPT后三页**

(5) 多类问题，长尾分布，如何解决？分箱？扩充？组件学习？

分箱：比如带有数字的和不带有数字的，带有数字的可以OCR识别。我觉的也可以从语义上进行分箱。

扩充：贴图扩充。

组件学习：自行车右转标志=自行车标志+右转标志。

(6) 该任务的难度到底是在检测上还是识别上？前后景分类的精度如何？

(7) 亮度的影响？街景白天与夜晚？

(8) anchors重新聚类生成？

统计及可视化：

(1) 统计交通标志样本数目，统计训练集中出现多少类交通标志，因测试集中可能会出现训练集中不存在的交通标志，故训练时需要贴图增强。

(2) 统计单张图片中样本数目，判断是否存在无样本的情况。（最好研究一下baseline代码看其如何处理无样本图像）测试集中是否某张图片中根本无交通标志。

(3) 交通标志贴图。

(4) 某些交通标志是比较小的（小样本目标检测），可以统计一下检测框面积大小，统计图像大小。

jittor入门学习+多卡训练搭建

四、论文阅读

(1) 交通标志在图像中占据的大小仅1%-2%，能否先确定存在交通标志区域，然后再做。

(2) 场景信息是否能够辅助检测识别，有些交通标志可能仅会出现在某些固定场景下。

(3) 图像增强：贴图增强，然后放到没有交通标志的图里？加点噪声。

五、数据集研究

1、数据集结构说明

```
tt100k_2021
├─ annotations_all.json
├─ marks.jpg
├─ report.pdf
├─ test_result.pkl
├─ marks
│   └─ <mark name>.png
├─ re_marks
```

```
|   └─ <mark name>.jpg
├─ train
|   └─ <image id>.jpg
├─ test
|   └─ <image id>.jpg
└─ other
    └─ <image id>.jpg
```

(1) marks文件夹：全部是png格式交通标志图片，共128个不同类。有白底，需去除多余边缘。这部分的样本可能来自marks.jpg。

(2) re_marks文件夹：全部是jpg格式交通标志图片，共105个不同类。好像是从数据集中裁出来的，背景是实际场景，无多余边缘。这部分样本可能来自report.pdf。

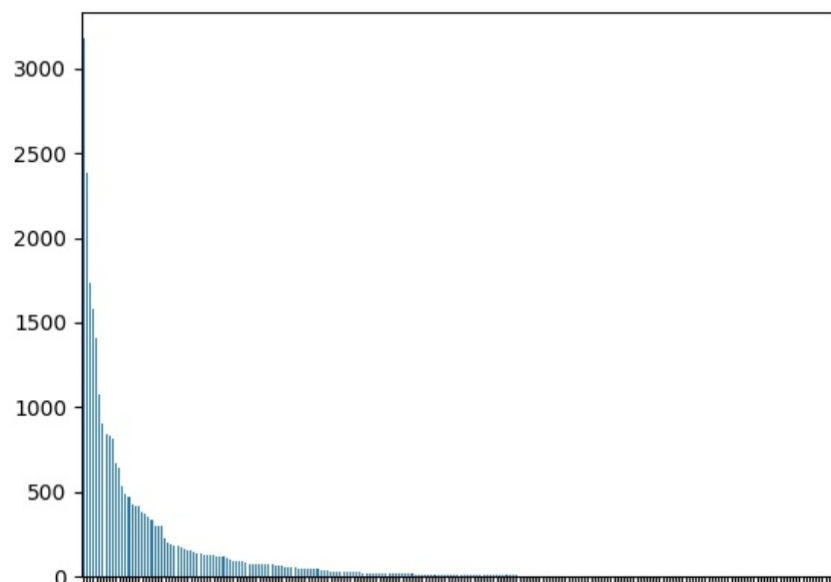
128+105=233>232，多出一个 p22 类，该类在marks文件夹中。

(3) annotations_all.json

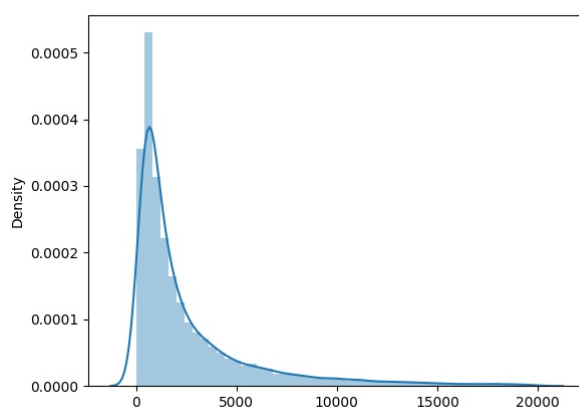
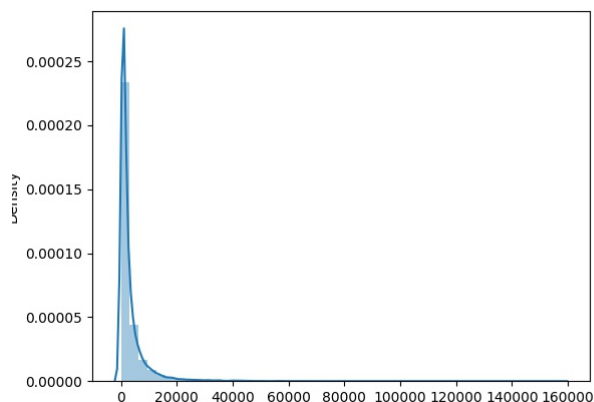
- types: 数据集交通标志的类型
- imgs: 字典格式，分别表示每一张图片的标注
 - path: 图片的相对路径
 - id: 图片的id (不包含后缀名)
 - objects: 列表类型，图片中所包含的交通标志
 - category: 交通标志的类型
 - bbox: 交通标志的包围盒(xmin,ymin),(xmax,ymax)分别代表框的左上角和右下角
 - xmin: 包围盒的左边缘
 - xmax: 包围盒的右边缘
 - ymin: 包围盒的上边缘
 - ymax: 包围盒的下边缘

有标注的图像共有10592，即 `len(imgs)`；交通标志共有232类，即 `len(types)`，但是数据集中包含的交通标志类为 $232-31=201$ 个。annotations中不包含空样本，即该10592张图片都存在标注，`len(objects)>0`。

annotations中的图片来自于train、test、other三个文件夹，前两者占比较多，后者较少，分别为 6034/6105，3016/3071，1542/7641。这也可能是在other中选择无标注图像的原因。



上图是对annotations中交通标志的样本进行统计，数目最多为3176，最少为0，样本数小于100的交通标志有187类，约占81%，长尾分布很严重。



对bbox面积的统计核密度图如上，数据集图像原始大小为 $2048 \times 2048 = 4194304$ ，但是bbox面积范围为 $24 \sim 157608$ ，占比最高为 3.7%，可见交通标志较小。而且从核密度图中可以发现大部分bbox面积较小，20000以下的bbox样本数为 26836，总样本数为 27346。

六、代码研读

1、数据增强

贴图过程：

- crop mask，去除多余边缘。cv2.IMREAD_UNCHANGED读取alpha透明度通道。
- select empty files，在other文件夹中挑选没有出现在annotation中的image。
- 选出要进行贴图的交通标志类别need_aug：样本数少于100。
- 随机选择空image，再随机生成不同大小的框，对应着随机的need_aug交通类别。aug_num为每个交通标志增强的样本数目。
- 贴图，添加噪声。
- 增强数据集合并到原始数据集。

```
def build_transforms(min_size=2048,
                    max_size=2048,
                    flip_horizontal_prob=0.5,
                    mean=[102.9801, 115.9465, 122.7717],
                    std = [1.,1.,1.],
                    to_bgr255=True):
```

```

transform = Compose([
    Resize(min_size, max_size),
    RandomHorizontalFlip(flip_horizontal_prob),
    ToTensor(),
    Normalize(mean=mean, std=std, to_bgr255=to_bgr255),
])
return transform

```

这部分处理存在问题，`RandomHorizontalFlip(flip_horizontal_prob)` 水平翻转处理可能会改变交通标志类型。**此处应该去除该处理。**

jittor的argmax返回两个元素，第一个是max inds，第二个是max value。

jt.where

七、学长

efficientdet + resnet。

裁剪。OK 数据增强 （不能缩，效果太差了）

fpn.py mmdetection

下采样可以改一下 stride

彩图转灰度图

新的思考：

(1) 对marks标准图像进行仿射变换、亮度调整、模糊处理会不会好些？

(2) train test other分开是为啥？猜想train ,test可以分开担任训练集与验证集，other可以用于生成贴图用于训练。问题：train test类别分布？不一致是不可以担任训练集与验证集的

train test前者类别数都不能包含后者。。且后者只有154类，所以应该不能担任训练集与验证集，如果适当的分配other的样本或许可以。

(3) baseline训练时只用了229类？会不会对测试的结果有影响

(4) proposal的min_size? min_size=16, 这是针对输入图像大小的，对应到下采样16倍的特征图上即1个特征点。这种设置会不会太小？如果该stride的话，这里是不是也要改？

这可能与评估时 $[16 \times 16, +\infty]$ 有关

(5) n_test_post_nms?

(6) FPN的问题：1.scales会不会安排的有些大；2.roi align求k时由于很多roi比较小，所以提取特征大部分都是在较浅层上，要不要改一下除的224。

目前修改：scales由 $[32, 64, 128, 256]$ --> $[16, 32, 64, 128]$

roi align被除数224-->50, k_0 由4-->3.

(7) 放大了有帮助吗？

(8) 代码中贴图是否会产生交通标志之间有所覆盖的现象？

(9) 额外分类器的学习：抛弃faster rcnn原本的rpn之后的classifier head，使用roi在原图中直接扣，然后放到一个性能优越的分类器（全卷积最好了，输入可以使任意尺寸）中进行分类，将结果用于计算faster rcnn的分类损失。

八、知乎学习

比赛的难点：**小目标检测，长尾目标检测。**

1、小目标检测

提高图像采集的分辨率。对内存与显存要求较高。

提高模型的输入分辨率。对内存与显存要求较高。

对图像进行Tiling，即将图像切割后形成batch。训练与推理tile要一致。

数据增强。随机裁剪、随机旋转、马赛克增强等。这个可以参考官方论文。

自动学习模型Anchors。

过滤掉无关类别。

多次复制小目标（单图）。我感觉这个在数据增强的mixup中是有实现的。

nms-->softnms 不太清楚数据集是否有遮挡的问题。

多尺度方法汇总。。见知乎

2、长尾目标检测

常用方法：

- 对少样本类别的物体进行过采样(over sampling),以保证类别平衡
 - 对少样本类别的物体的损失权重进行修改,使得少样本类别物体具有更大的loss Re-weight
- 分箱：按照数量分箱；按照语义分箱

论文：

<https://zhuanlan.zhihu.com/p/352303740>

repeat factor sampler, class balanced sampler

第二次组会：

cutmix cutout

分类：二分类，多分类

接下来优先做的实验：

(1) 级联RCNN的改进

(2) backbone替换 resnet50-->resnet101

```
正负样本选取，现在是随机选的，可以考虑用Focal Loss  
Roi Align 可以考虑从 7*7 变成 14*14  
anchor_scales=[8, 16, 32] 变为 anchor_scales=[4, 8, 16] base_size也要改  
添加SE Block，在ResNet中或者Roi Align之后  
Roi Head 中 in_channels=self.backbone.out_channels太大了，4096也有点太大了，可以考虑在FC加dropout防止过拟合  
RPN中的min_size=16有点大？
```

除focal loss的改进使用之后，效果提高了0.005个点。

focal loss。centernet2。

新的数据增强：（师兄原话）

我的意思就是你现在就是读到一张图片了吗？如果他上面已经有它上面有框的话，你就可以考虑拿一些别的框来替换掉他，然后这个被替换掉的，别的框就可以是从那些项目类别少的交通标志中采样得到的，然后替换的时候嗯，大小基本上是相同的，你可以考虑给它放大或者缩小一点点，但是不能差太多，就是大小应该是差不多的，然后替换完之后把那个图像存下来，就它就相当于是一张新的图像。