

学习目标：理解赛题背景和赛题数据，完成赛题报名和数据下载，理解赛题的解题思路

首先贴出Datawhale开源项目链接和天池赛事报名链接：

[Datawhale开源项目链接](#)

[天池赛事报名链接](#)

该项目为Datawhale和天池合作举办的cv入门赛事街景字符识别开源项目，比较适合刚入门深度学习并且想要找项目练手的同学们。

1 赛题背景

谷歌街景，是谷歌地图的一项特色服务，是由专用街景车进行拍摄，然后把360度实景拍摄照片放在谷歌地图里供用户使用。

该赛题即来源于Google街景图片中的门牌号识别问题。

我们知道，在受约束环境下（如文档处理）的字符识别技术，比方说OCR技术已经被研究得很透。但是对照片中的多字符文本识别却是个难题，因为这些照片中的字体、颜色、样式、方向、排列不一，再加上光照、阴影、镜像、遮蔽等环境因素影响，还有图像本身的分辨率、焦点模糊、抖动等问题。而传统的图像文字识别，一般要经过字符定位、分割和识别三个步骤，处理效率不高。考虑Google街景的每天上传的图片量，以往的这些方法显然缺乏实用价值。

为此，Google开发了一套大型的神经网络来处理海量的Google街景图片中的门牌号识别问题。这个神经网络的代号是DistBelief，其学名是深度卷积神经网络。经过DistBelief训练的这个大型分布式神经网络，可以把定位、分割和识别三个步骤集成到一起，直接对每一个像素进行操作。其性能随着神经网络的深度增加而提高，在11层的时候达到最高。根据论文的数据，该系统对SVHN（街景门牌号数据集）中的门牌号识别率达96%，而单数字识别率达97.84%，对于Google街景图库的上千万门牌号的识别率也超过90%。

此处附上论文下载链接：https://pan.baidu.com/s/1Z3g1pps_pP_Ef849vslHWg 提取码：220B

2 赛题数据

赛题来自Google街景图像中的门牌号数据集SVHN（The Street View House Numbers Dataset），并根据一定方式采样得到比赛数据集。需要注意的是本赛题需要选手识别图片中所有的字符，为了降低比赛难度，我们提供了训练集、验证集和测试集中字符的位置框（如下图所示）。



数据集

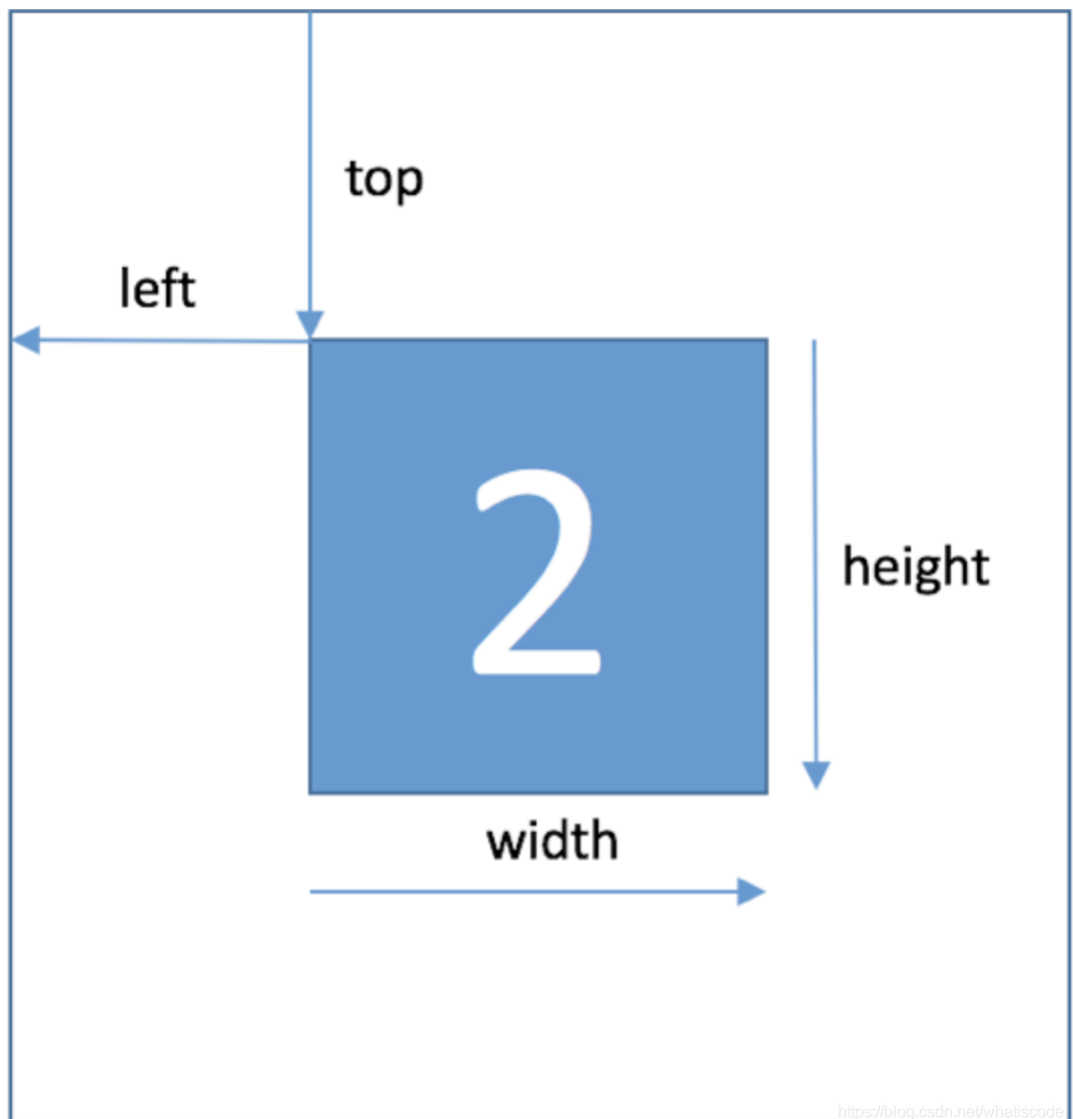
训练集数据包括3W张照片，验证集数据包括1W张照片，每张照片包括颜色图像和对应的编码类别和具体位置；为了保证比赛的公平性，测试集A包括4W张照片，测试集B包括4W张照片。

字段表

所有的数据（训练集、验证集和测试集）的标签（label）使用JSON格式，并使用文件名进行索引。如果一个文件中包括多个字符，则使用列表将字段进行组合。

Field	Description
top	左上角坐标X
height	字符高度
left	左上角最表Y
width	字符宽度
label	字符编码

字符的具体坐标如下图：



<https://blog.csdn.net/whaliscode>

在比赛数据（训练集和验证集）中，同一张图片中可能包括一个或者多个字符，因此在比赛数据的JSON标注中，会有两个字符的边框信息：

原始图片	图片JSON
	<pre>: train_json['0 executed in 6ms, fin : {'height': [2 'label': [1, 'left': [246 'top': [77, 'width': [81</pre>

3 评测指标

选手提交结果与实际图片的编码进行对比，以编码整体识别准确率为评价指标。任何一个字符错误都为错误，最终评测指标结果越大越好，具体计算公式如下：

$$score = \frac{\text{编码识别正确的数量}}{\text{测试集图片数量}}$$

4 结果提交

提交前请确保预测结果的格式与sample_submit.csv中的格式一致，以及提交文件后缀名为csv。

```
file_name, file_code
0010000.jpg,451
0010001.jpg,232
0010002.jpg,45
0010003.jpg,67
0010004.jpg,191
0010005.jpg,892
```

5 数据读取

JSON中标签的读取方式：

```
# 读取图片中位置框里面的字符
import json
train_json = json.load(open('../input/train.json'))

# 数据标注处理
def parse_json(d):
    arr = np.array([
        d['top'], d['height'], d['left'], d['width'], d['label']
    ])
    arr = arr.astype(int)
    return arr

img = cv2.imread('../input/train/000000.png')
arr = parse_json(train_json['000000.png'])

plt.figure(figsize=(10, 10))
plt.subplot(1, arr.shape[1]+1, 1)
plt.imshow(img)
plt.xticks([]); plt.yticks([])

for idx in range(arr.shape[1]):
    plt.subplot(1, arr.shape[1]+1, idx+2)
    plt.imshow(img[arr[0, idx]:arr[0, idx]+arr[1, idx], arr[2, idx]:arr[2, idx]+arr[3, idx]])
    plt.title(arr[4, idx])
    plt.xticks([]); plt.yticks([])
```

6 解题思路

赛题思路分析：赛题本质是分类问题，需要对图片的字符进行识别。但赛题给定的数据图片中不同图片中包含的字符数量不等。比如说在本文中的第一张图里面，有的图片的字符个数为2，有的图片字符个数为3，有的图片字符个数为4。

因此本次赛题的难点是需要对不定长的字符进行识别，与传统的图像分类任务有所不同。为了降低参赛难度，大佬们提供了以下解题思路：

(1)简单入门思路：定长字符识别

可以将赛题抽象为一个定长字符识别问题，在赛题数据集中大部分图像中字符个数为2-4个，最多的字符个数为6个。因此可以对于所有的图像都抽象为6个字符的识别问题，字符23填充为23XXXX，字符231填充为231XXX。

经过填充之后，原始的赛题可以简化了6个字符的分类问题。在每个字符的分类中会进行11个类别的分类，假如分类为填充字符，则表明该字符为空。

(2)专业字符识别思路：不定长字符识别

在字符识别研究中，有特定的方法来解决此种不定长的字符识别问题，比较典型的有CRNN字符识别模型。在本次赛题中给定的图像数据都比较规整，可以视为一个单词或者一个句子。

(3)专业分类思路：检测再识别

在赛题数据中已经给出了训练集、验证集中所有图片中字符的位置，因此可以首先将字符的位置进行识别，利用物体检测的思路完成。

此种思路需要参赛选手构建字符检测模型，对测试集中的字符进行识别。选手可以参考物体检测模型SSD或者YOLO来完成。

7 总结

首先很感谢Datawhale和天池能够提供一个这么好的开源项目，让我们这些对深度学习感兴趣的小白们也有机会参与到实践当中，并且认识了一群志同道合的小伙伴们。项目中不仅提供了每一步的具体流程和代码，还会定期有大佬直播分享经验，并且还能和队友们一起讨论，互相监督。

经过这几天的学习，认知到自己的不足，尤其是实践方面；所以在这里定下一些小目标，希望在这个项目结束后能够有所收获：

1. 能够基本学会PyTorch框架
2. 阅读5篇以上相关论文
3. 每天按时汇报情况并归纳笔记