MS COCO

## Microsoft COCO: Common Objects in Context

Tsung-Yi Lin   Michael Maire   Serge Belongie   Lubomir Bourdev   Ross Girshick
James Hays   Pietro Perona   Deva Ramanan   C. Lawrence Zitnick   Piotr Dollár

MS COCO是一个非常大型且常用的数据集，
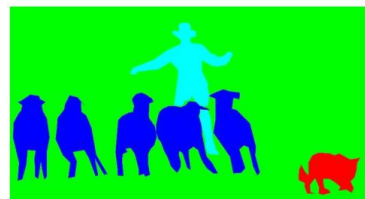其中包括了目标检测，分割，图像描述等。
其主要特性如下：

- **Object segmentation**: 目标级分割
- **Recognition in context**: 图像情景识别
- **Superpixel stuff segmentation**: 超像素分割
- **330K images (>200K labeled)**: 超过33万张图像，标注过的图像超过20万张
- **1.5 million object instances**: 150万个对象实例
- **80 object categories**: 80个目标类别
- **91 stuff categories**: 91个材料类别
- **5 captions per image**: 每张图像有5段情景描述
- **250,000 people with keypoints**: 对25万个人进行了关键点标注
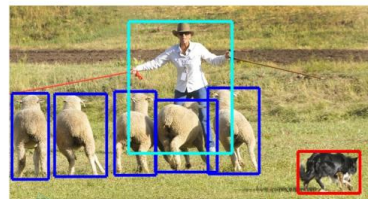
论文下载地址：https://arxiv.org/pdf/1405.0312.pdf
推荐博文：https://blog.csdn.net/qq_37541097/article/details/113247318

(a) Image classification

(b) Object localization

(c) Semantic segmentation

(d) This work

MS COCO

(a) Image classification
person, sheep, dog

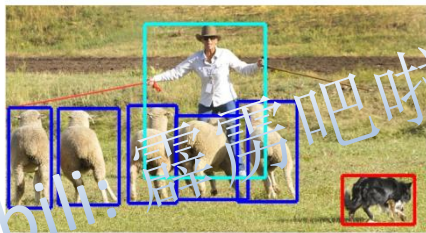(b) Object localization

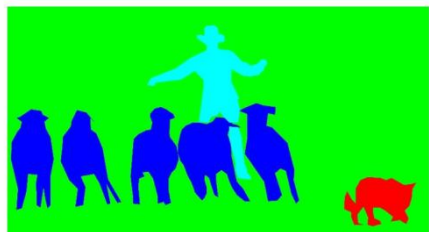(c) Semantic segmentation

(d) This work

这里需要注意的一个点是"什么是stuff类别"，在官方的介绍论文中是这么定义的：
where "stuff" categories include materials and objects with no clear boundaries (sky, street, grass)
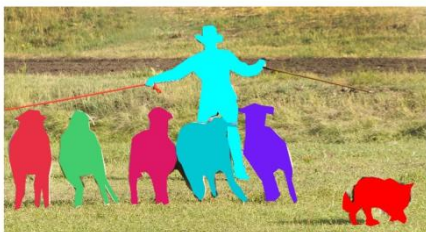**stuff中包含没有明确边界的材料和对象。**

object的80类与stuff中的91类的区别在哪？在官方的介绍论文中有如下说明：
Note that we have limited the 2014 release to a subset of 80 categories. We did not collect segmentations for the following 11 categories: hat, shoe, eyeglasses (too many instances), mirror, window, door, street sign (ambiguous and difficult to label), plate, desk (due to confusion with bowl and dining table, respectively) and blender, hair brush (too few instances).
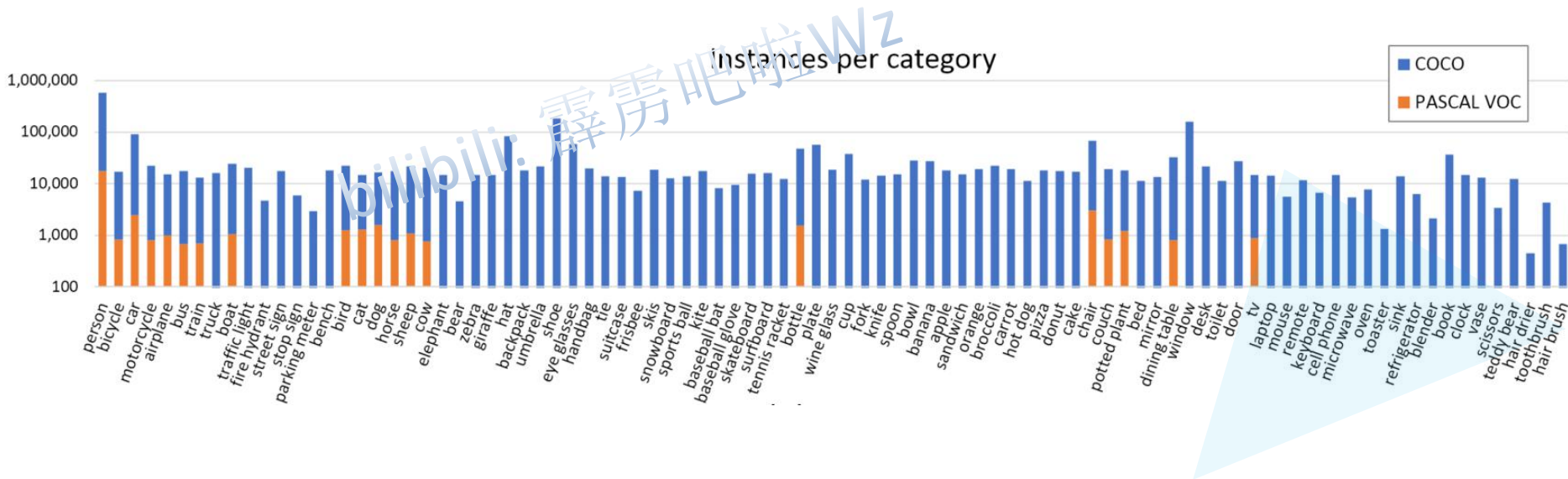简单的理解就是**object80类是stuff91类的子集。如果仅仅是做目标检测，基本只用object80类即可。**

MS COCO

与PASCAL VOC进行对比

预训练效果更好
但更费时

6x2x5/24=2.5 (单块GPU)
6x2x5x30/24=75 (CPU)

Instances per category

MS COCO

官网地址：

https://cocodataset.org/

MS COCO

对于目标检测主要下载三个文件：

☐ 2017 Train images [118K/18GB]：训练过程中使用到的所有图像文件

☐ 2017 Val images [5K/1GB]：验证过程中使用到的所有图像文件

☐ 2017 Train/Val annotations [241MB]：对应训练集和验证集的标注json文件

MS COCO

都解压到coco2017文件夹下，可得到如下文件结构：

```
├── coco2017：数据集根目录
     ├── train2017：所有训练图像文件夹(118287张)
     ├── val2017：所有验证图像文件夹(5000张)
     └── annotations：对应标注文件夹
              ├── instances_train2017.json：对应目标检测、分割任务的训练集标注文件
              ├── instances_val2017.json：对应目标检测、分割任务的验证集标注文件
              ├── captions_train2017.json：对应图像描述的训练集标注文件
              ├── captions_val2017.json：对应图像描述的验证集标注文件
              ├── person_keypoints_train2017.json：对应人体关键点检测的训练集标注文件
              └── person_keypoints_val2017.json：对应人体关键点检测的验证集标注文件夹
```

MS COCO

官网有给出一个关于标注文件的格式说明，可以通过以下链接查看：
https://cocodataset.org/#format-data

MS COCO

```
 1  [
 2      {
 3          "image_id": 139,
 4          "category_id": 62,
 5          "bbox": [
 6              295.11,
 7              219.6,
 8              59.11,
 9              100.73
10          ],
11          "score": 0.988
12      },
13      {
14          "image_id": 139,
15          "category_id": 72,
16          "bbox": [
17              8.5,
18              168.27,
19              151.78,
20              97.95
21          ],
22          "score": 0.981
23      },
24      {
25          "image_id": 139,
26          "category_id": 62,
27          "bbox": [
28              375.69,
29              217.23,
30              68.45,
31              95.52
```

根据官方文档给的预测结果格式可以看到，我们需要以列表的形式保存结果，列表中的每个元素对应一个检测目标（每个元素都是字典类型），每个目标记录了四个信息：

➢ image_id记录该目标所属图像的id（int类型）

➢ category_id记录预测该目标的类别索引．注意这里索引是对应stuff中91个类别的索引信息（int类型）

➢ bbox记录预测该目标的边界框信息，注意对应目标的[xmin, ymin, width, height]（list[float]类型）

➢ score记录预测该目标的概率（float类型）