目标检测在图像上bbox绘制与添加类别（解决中文字体添加）

在 目标检测中，得到结果之后往往需要在图像上添加bbox和预测类别与概率可视化显示，使用opencv可以简单实现这一功能（遇到中文label会出现乱码，下文解决）

1. opencv绘制bbox并且添加类别概率

```
# 采用随即生成所有类别的边框与字体的颜色
colors = [[random.randint(0, 255) for _ in range(3)] for _ in range(num_classes)]

# cv2.rectangle(img, (x,y), (x+w,y+h), (B,G,R), Thickness)
cv2.rectangle(img, (x1,y1), (x2, y2), colors[category_id], 2)
text = str(category_id)
Font = cv2.FONT_HERSHEY_SIMPLEX
# cv2.putText(img, text, (x,y), Font, Size, (B,G,R), Thickness)
cv2.putText(img, text, (x1,y1-10), Font, 1.5, colors[category_id], 2)
```

上边的方法所采用的类别的方式为数字表示，可视化看起来不直观，如果直接采用中文label，则会出现乱码，需要进行额外的处理

2. 利用PIL解决中文字体显示的问题

在终端中输入如下的语句，查看在/usr/share/fonts中有什么字体。

```
locate *.ttc
```

然后采用如下的代码：

```
from PIL import Image, ImageDraw, ImageFont
def drawOneBox(img, bbox, color, label):
    '''对于给定的图像与给定的与类别标注信息，在图片上绘制出bbox并且标注上指定的类别
    '''
    img_PIL = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    # locate *.ttc
    font = ImageFont.truetype("NotoSansCJK-Bold.ttc", 20, encoding='utf-8')

    x1, y1, x2, y2 = bbox
    draw = ImageDraw.Draw(img_PIL)
    position = (x1, y1 - 30)
    draw.text(position, label, tuple(color), font=font)
    img = cv2.cvtColor(np.asarray(img_PIL),cv2.COLOR_RGB2BGR)
    cv2.rectangle(img, (x1,y1), (x2, y2), colors[category_id], 2)
    return img
```

亲测可用，这里记录一下，防止随后有需要的时候再到处找方法。