深度学习方法与实践实验一

2020 年 2 月 26 日

年　级:	2020 级	学　　号:	2016012963
姓　名:	董佩杰	指导老师:	牛新

1　用 OpenCV 画飞机

1.1　要求

- 用 python 中的 opencv 库画 3 种飞机，并经过旋转扩充，标注关键点，形成飞机飞机分类与关键点预测数据集”AIRPLANES”。
- 飞机颜色要求正确：椭圆用蓝色，长方形用红色，三角形用黄色, 并且椭圆机身在最顶层，盖住机翼和尾翼。
- 按照要求打包和提交材料。

1.2　飞机的结果

以下是按照格式进行旋转的图片，分别对应的是三个类：“AIRBUS”、“FIGHTER” 和 “UAV”

图 1: AIRBUS 类别对应的图片和标签

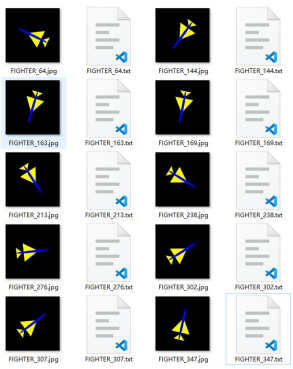

图 2: FIGHTER 类别对应的图片和标签

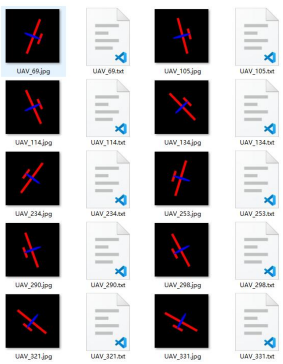

图 3: UAV 类别对应的图片和标签

最终文件内容为：

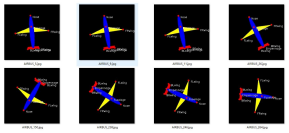

图 4: AIRBUS 类别对应的标注结果

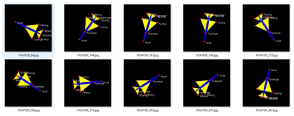

图 5: FIGHTER 类别对应的标注结果

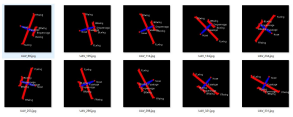

图 6: UAV 类别对应的标注结果

AIRPLANES > UAV > UAV_69.txt

```
1  Nose 87 112
2  Fuselage 126 127
3  Empennage 166 143
4  FLwing 95 209
5  FRwing 158 46
6  BLwing 158 166
7  BRwing 176 118
8
```

图 7: 以 UAV 为例的标注文件内容

1.3 思路

实验的要求是使用 opencv 来完善代码中没有构建完成的部分，并且提供了“AIRBUS”类别的样例。按照样例可以很快完成剩余两个类的构建。在构建过程中需要注意两个地方：

(1) 画图的顺序，椭圆代表的机身是需要最后画的。

(2) cv2.rectangle 的用法，这个方法需要注意的是使用的坐标是矩形左上角坐标和右下角坐标。并且由于 rectangle 有两个构造函数，其中一个不是坐标，而是 rect 参数，这时候最好指名使用的是哪个构造函数：

```
img = cv2.rectangle(
        img=img,
        pt1=tuple(np.array(LP_UAV['BLwing']) - np.array([0, WIDTH_WING])),
        pt2=tuple(LP_UAV['BRwing']),
        color=(0, 0, 255),
        thickness=-1)
```

三个构件图片的函数写完以后，开始完善 **rotate_anno** 函数，首先要搞清楚三个参数：savepath 代表保存的 txt 标注文件；angle 代表旋转的角度，p_sets 不是特别明显，但是可以通过下面的提示得到这是字典，所以就直接通过字典访问得到对应的坐标即可。

```
def rotate_anno(savepath, angle, p_sets):
        f_save = open(savepath, 'w')
        for p in p_sets:
                #-------------to do----------------------
                x = p_sets[p][0]
                y = p_sets[p][1]
                #-------------to do----------------------
                x, y = rotate_onepoint(x, y, angle)
                newline = str(p) + ' ' + str(x) + ' ' + str(y) + '\n'
                f_save.writelines(newline)
        f_save.close()
```

接下来就是 **make_data** 函数，在题目中已经有提示，只需要注意文件命名规则等细节即可完成。代码如下：

```
def make_data(PATH_DATA_SET, name_data, num):
        #-------------to do-----------------------
        PATH_DATA = osp.join(PATH_DATA_SET, name_data)
        if not osp.exists(PATH_DATA):
                os.makedirs(PATH_DATA)
        #-------------to do-----------------------
        if name_data == 'AIRBUS':
                img = draw_airbus()
                p_sets = LP_AIRBUS
        elif name_data == 'FIGHTER':
                img = draw_fighter()
                p_sets = LP_FIGHTER
        elif name_data == 'UAV':
                img = draw_uav()
                p_sets = LP_UAV
        else:
                print('wrong data')


        #-------------to do-----------------------
        for i in range(num):
                angle = random.randint(0, 359)
                rotate_anno(osp.join(PATH_DATA, name_data + "_%d.txt" % angle), angle, p_sets)
                matRotate = cv2.getRotationMatrix2D(
                (SIZE_IMG * 0.5, SIZE_IMG * 0.5), angle,1)
                img_r = cv2.warpAffine(img, matRotate, (SIZE_IMG, SIZE_IMG))
                cv2.imwrite(osp.join(PATH_DATA, name_data + "_%d.jpg" % angle), img_r)
```

2 拟合余弦函数

2.1 要求

假设有函数 y = cos(ax + b), 其中 a 为学号前两位，b 为学号最后两位。首先从此函数中以相同步长（点与点之间在 x 轴上距离相同），在 0<(ax+b)<2pi 范围内，采样出 2000 个点，然后利用采样的 2000 个点作为特征点进行三次函数拟合。

2.2 拟合结果

主要是通过 np.polyfit 函数进行三次拟合。代码如下：

```
# 学号： 2016012963
# 函数：y = cos(20x+63)
import math
import numpy as np
import matplotlib.pyplot as plt


def func(x):
        return math.cos(20 * x + 63)

if __name__ == "__main__":
        x = np.linspace(-63.0 / 20, float(2 * math.pi - 63) / 20, 2000)
```

```
13          y = [func(i) for i in x]
14
15          f1 = np.polyfit(x, y, 3)
16          p = np.poly1d(f1)
17
18          y2 = p(x)
19
20          plt.scatter(x, y)
21          plt.scatter(x, y2)
22          plt.show()
```

下面是可视化结果：

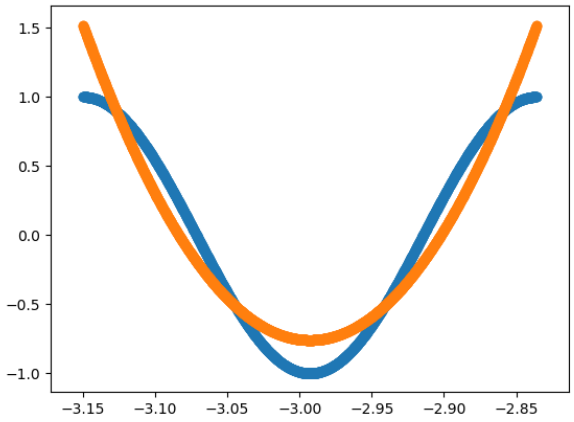

图 8: 用三次函数模拟余弦函数的结果