实验 1 分类标注结果转换

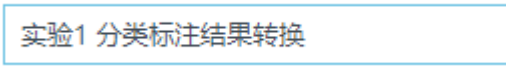
实验时长：4 小时

实验难度：一般

实验摘要：通过 labelme 进分类标注后，完成的标注数据用 json 文件保存。但是在不同的应用场景中，可能需要有不同的格式。在本任务将使用一个 csv 文件保存所有的图片文件名，和对应的类别标签。

实验建议：了解数据标注的基本格式

实验目标：能够解析 json 格式的标注文件，并保存结果。



1、分类标注结果转换

-

1.1、读入标注 json 文件

可以利用 python 完成一个小程序，自动把所有的 json 文件转换成一个 csv 文件。首先读取标注 json 文件，并解析出其中的 flags 字段数据。

需要调用处理文件的库 os，解析 json 的库 json 和生成 csv 的库 pandas。

```
import json
```

```
import os
```

```
import pandas as pd
```

因为 labelme 生成的 json 文件都保存在图片相同的目录下，因此直接去读取这个目录。设置 ROOT_PATH 指向目录名，并使用 os.listdir 获取目录下所有文件名。

```
#设定读取 json 文件的目录

ROOT_PATH = './flowers'

#读取所有文件名

files = os.listdir(ROOT_PATH)
```

对于所有的文件，根据后缀判断是否是 json 文件

```
for file in files:

    #判断是否是 json 文件，
```

```
    if not file.endswith('.json'):

        continue
```

如果是 json 文件，则打开改文件，并使用 json 库来加载文件内容到 annotation 变量中。

```
    #获取文件全路径

    filepath = os.path.join(ROOT_PATH, file)

    print(filepath)

    #读取文件，并用 json 解析

    with open(filepath, 'r') as f:

        annotation = json.load(f)
```

下一步

1、分类标注结果转换

- 1.1、读入标注 json 文件
- ---
- 1.2、解析 json 文件中的标注信息
- ---
- 因为是分类任务的标注结果，因此分类标签在 flags 对象中，同时 imagePath 中记录了对应的图像文件名。这些都是生成 csv 需要的标注信息。
- ---
- `#读取其中的 imagePath，获取图像路径`
- ---
- `imgpath = annotation.get("imagePath")`
- ---
- 然后再读取 flags 的值
- ---
- `#读取其中的 flags 作为分类标签`
- ---
- `flags = annotation.get("flags")`
- ---

读取到的 flags 是字典数据类型，其中 key 对应花朵的类型，value 对应 boolean 值标注图片是否是这种花朵类型。因为所有的花朵类型的顺序都和 flags.txt 中的一致。因此采用花朵类型的序列号作为类别的标签。下面把字典 flags 中的 boolean 值转换成数组类型。

- `#读取所有标签`
- `values = list(flags.values())`
-
- 获得的数组类似[False, False, True, False,False, False]。
- 遍历数组，将对应为 true 的序列号作为 label 保存。
-
- `label =0`
- `for i in range(len(values)):`
- `  #如果为 true，则取当前序号作为标签`
- `  if values[i]:`
- `    label = i`
- `print("label:",label)`
-

- 可以得到类似 label: 2

- 将解析出来的文件名称和标签组成一个数组。

-
-

```
#生成文件名和标签组成的数组
```

-

```
data = []
```

-

```
data.append(imgpath)
```

-

```
data.append(label)
```

- 得到类似['1185.jpg', 2]

- 然后将每个 json 文件解析出来的数组汇总成一个二维数组

-
-
-

```
#汇总到大数组中
```

-

```
alldata.append(data)
```

- 1.3、保存成 csv 文件格式

查看一下 alldata 数组的长度，应该是和标注的文件个数相同。

```
print(len(alldata))
```

为了方便通过 pandas 转成 csv 文件格式，先将数组类型的 alldata 转换成 pandas 的 dataframe 格式

```
#转换成 dataframe 格式
flowers_data = pd.DataFrame(alldata)
```

获得类似如下格式：

0 1

0 1185.jpg 2

1 1195.jpg 2

2 1223.jpg 1

3 1346.jpg 4

…

然后 pandas 可以方便的把 dataframe 类型的数据直接转成 csv 文件进行保存。不需要保存列名称和序列号，因此可以使用下面代码。

```
#转换成 csv 文件，不需要保存列名和序号
flowers_data.to_csv("flowers.csv",header=False,index=False)
```

可以查看获得 flowers.csv 文件，类似如下

1185.jpg,2

1195.jpg,2

1223.jpg,1

1346.jpg,4

1480.jpg,5

1527.jpg,5

2、完整代码

- 2.1、完整代码

- 完整代码如下：

```
import json
import os
import pandas as pd


#设定读取 json 文件的目录
ROOT_PATH = './flowers'
```

```
#读取所有文件名
files = os.listdir(ROOT_PATH)
alldata = []
for file in files:
    #判断是否是 json 文件，
    if not file.endswith('.json'):
        continue
    #获取文件全路径
    filepath = os.path.join(ROOT_PATH, file)
    print(filepath)
    #读取文件，并用 json 解析
    with open(filepath, 'r') as f:
        annotation = json.load(f)


    #读取其中的 imagePath，获取图像路径
```

```
    imgpath = annotation.get("imagePath")
    #读取其中的 flags 作为分类标签
    flags = annotation.get("flags")
    print(type(flags))

    #读取所有标签
    values = list(flags.values())
    print(values)


    label  =0
    for i in range(len(values)):
        #如果为 true，则取当前序号作为标签
        if values[i]:
            label = i
    print("label:",label)
```

```
    #生成文件名和标签组成的数组
    data = []
    data.append(imgpath)
    data.append(label)

    #汇总到大数组中
    alldata.append(data)

#print(len(alldata))
#print(alldata)

#转换成 dataframe 格式
flowers_data = pd.DataFrame(alldata)
print(flowers_data)
```

-
  ```
  #转换成 csv 文件，不需要保存列名和序号
  ```
-
  ```
  flowers_data.to_csv("flowers.csv",header=False,index=False)
  ```
-
-