

深度学习快速入门

01Pytorch 加载数据

POET

2024 年 2 月 10 日

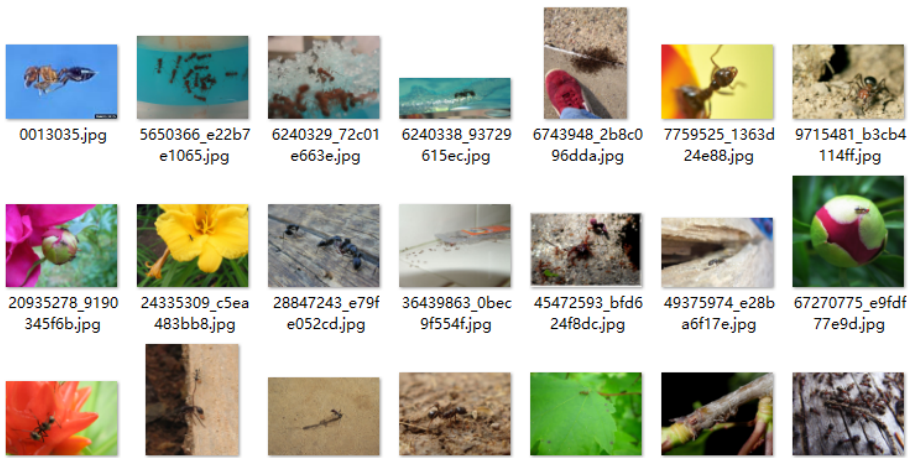
1 Pytorch 加载数据

- 1. Dataset: 提供一种方式去获取每个数据及其对应的 label, 告诉我们总共有多少个数据。
- 2. Dataloader: 为后面的网络提供不同的数据形式, 它将一批一批数据进行一个打包。

2 常见 Dataset 形式

```
from torch.utils.data import Dataset
help(Dataset)
```

- 常用的第一种数据形式, 文件夹的名称是它的 label



- 常用的第二种形式, label 为文本格式, 文本名称为图片名称, 文本中的内容为对应的 label

名称	修改日期	类型	大小
ants_image	2019/12/7 20:02	文件夹	
ants_label	2019/12/7 21:04	文件夹	
bees_image	2019/12/7 20:02	文件夹	
bees_label	2019/12/7 21:04	文件夹	

3 路径直接加载数据

```
from PIL import Image

img_path = "C:\\Users\\Asabopp\\Desktop\\Learn_pytorch\\Test\\dataset\\train\\ants_image\\0013035.jpg"

img = Image.open(img_path)
img.show()
```

4 Dataset 加载数据

```
from torch.utils.data import Dataset
from PIL import Image
import os

class MyData(Dataset):
    def __init__(self, root_dir, label_dir): # 该魔术方法当创建一个事例对象时，会自动调用该函数
        self.root_dir = root_dir # self.root_dir 相当于类中的全局变量
        self.label_dir = label_dir
        self.path = os.path.join(self.root_dir, self.label_dir) # 字符串拼接，根据是Windows或Linux系统情况进行拼接
        self.img_path = os.listdir(self.path) # 获得路径下所有文件或文件夹的名称

    def __getitem__(self, idx):
        img_name = self.img_path[idx]
```

```
        img_item_path = os.path.join(self.root_dir,
                                       self.label_dir, img_name)
        img = Image.open(img_item_path)
        label = self.label_dir
        return img, label

    def __len__(self):
        return len(self.img_path)

root_dir = "dataset/train"
ants_label_dir = "ants_image"
bees_label_dir = "bees_image"
ants_dataset = MyData(root_dir, ants_label_dir)
bees_dataset = MyData(root_dir, bees_label_dir)
print(len(ants_dataset))
print(len(bees_dataset))
train_dataset = ants_dataset + bees_dataset #
    train_dataset 就是两个数据集的集合了
print(len(train_dataset))

img, label = train_dataset[200]
print("label: ", label)
img.show()
```

5 补充代码：根据文件名称生成内容为 label 的 txt 文件

```
import os

root_dir = 'dataset/train'
```

```
target_dir = 'ants_image'
img_path = os.listdir(os.path.join(root_dir,
                                     target_dir))
label = target_dir.split('_')[0]
out_dir = 'ants_label'
for i in img_path:
    file_name = i.split('.jpg')[0]
    with open(os.path.join(root_dir, out_dir, "{}.
                    txt".format(file_name)), 'w') as f:
        f.write(label)
```