-i <https://pypi.tuna.tsinghua.edu.cn/simple> 下载源加速用

Labelme的安装

1.安装好Anaconda 环境

2.开始安装labelme

进入 anaconda prompt 命令窗口

这是官网github提供的指令；

# 一、python2的输入下面的指令进行labelme的安装：

conda create --name=labelme python=2.7

#source activate labelme，官方说用这个，但是我提示有误，要用下面这一句才行

conda activate labelme

# conda install -c conda-forge pyside2

conda install pyqt

pip install labelme

# if you'd like to use the latest version. run below:

# pip install git+https://github.com/wkentaro/labelme.git

# 二、python3的输入下面的指令进行labelme的安装：

conda create --name=labelme python=3.6

#source activate labelme，官方说用这个，但是我提示有误，要用下面这一句才行

conda activate labelme

# conda install -c conda-forge pyside2

# conda install pyqt

pip install pyqt5 # pyqt5 can be installed via pip on python3

pip install labelme

启动lablelme 方法： 命令行输入 labelme

用labelme软件标注图片后，会得到一个.json文件

如何转换json

Activate label 先切换到需要进行转换的路径下面

命令：labelme\_json\_to\_dataset <文件名>.json

Ex: labelme\_json\_to\_dataset H0002.json

批量转换json

D:\programFiles\anaconda\envs\labelme\Lib\site-packages\labelme\cli

一、打开abelme安装目录

在lableme安装目录下有G:\Anaconda\Lib\site-packages\labelme\cli目录，可以看到json\_to\_dataset.py文件

这里面提供将json转成dataset的代码，所以我们只需要在这个基础上更改即可。

主要是加了一个for 循环 ，

修改完毕后，

找到labelme\_json\_to\_dataset.exe的路径D:\programFiles\anaconda\envs\labelme\Scripts>

在cmd命令 cd D:\programFiles\anaconda\envs\labelme\Scripts>

再输入：labelme\_json\_to\_dataset.exe D:\GitDownLoad\PaddleSeg\data\test

修改代码如下（标红部分为与源代码不同的地方）：

import argparse  
import base64  
import json  
import os  
import os.path as osp  
  
import imgviz  
import PIL.Image  
  
from labelme.logger import logger  
from labelme import utils  
  
  
def main():  
 logger.warning(  
 "This script is aimed to demonstrate how to convert the "  
 "JSON file to a single image dataset."  
 )  
 logger.warning(  
 "It won't handle multiple JSON files to generate a "  
 "real-use dataset."  
 )  
  
 parser = argparse.ArgumentParser()  
 parser.add\_argument("json\_file")  
 parser.add\_argument("-o", "--out", default=None)  
 args = parser.parse\_args()  
  
 json\_file = args.json\_file  
  
 if args.out is None:  
 out\_dir = osp.basename(json\_file).replace(".", "\_")  
 out\_dir = osp.join(osp.dirname(json\_file), out\_dir)  
 else:  
 out\_dir = args.out  
 if not osp.exists(out\_dir):  
 os.mkdir(out\_dir)  
 count = os.listdir(json\_file)  
 for i in range(0, len(count)):  
 path = os.path.join(json\_file, count[i])  
 if os.path.isfile(path):  
 data = json.load(open(path))  
 imageData = data.get("imageData")  
  
 if not imageData:  
 imagePath = os.path.join(os.path.dirname(path), data["imagePath"])  
 with open(imagePath, "rb") as f:  
 imageData = f.read()  
 imageData = base64.b64encode(imageData).decode("utf-8")  
 img = utils.img\_b64\_to\_arr(imageData)  
  
 label\_name\_to\_value = {"\_background\_": 0}  
 for shape in sorted(data["shapes"], key=lambda x: x["label"]):  
 label\_name = shape["label"]  
 if label\_name in label\_name\_to\_value:  
 label\_value = label\_name\_to\_value[label\_name]  
 else:  
 label\_value = len(label\_name\_to\_value)  
 label\_name\_to\_value[label\_name] = label\_value  
 lbl, \_ = utils.shapes\_to\_label(  
 img.shape, data["shapes"], label\_name\_to\_value  
 )  
  
 label\_names = [None] \* (max(label\_name\_to\_value.values()) + 1)  
 for name, value in label\_name\_to\_value.items():  
 label\_names[value] = name  
  
 lbl\_viz = imgviz.label2rgb(  
 label=lbl, img=imgviz.asgray(img), label\_names=label\_names, loc="rb"  
 )  
  
 PIL.Image.fromarray(img).save(osp.join(out\_dir, "img.png"))  
 utils.lblsave(osp.join(out\_dir, "label.png"), lbl)  
 PIL.Image.fromarray(lbl\_viz).save(osp.join(out\_dir, "label\_viz.png"))  
  
 with open(osp.join(out\_dir, "label\_names.txt"), "w") as f:  
 for lbl\_name in label\_names:  
 f.write(lbl\_name + "\n")  
  
 logger.info("Saved to: {}".format(out\_dir))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()