# 微信小程序后端

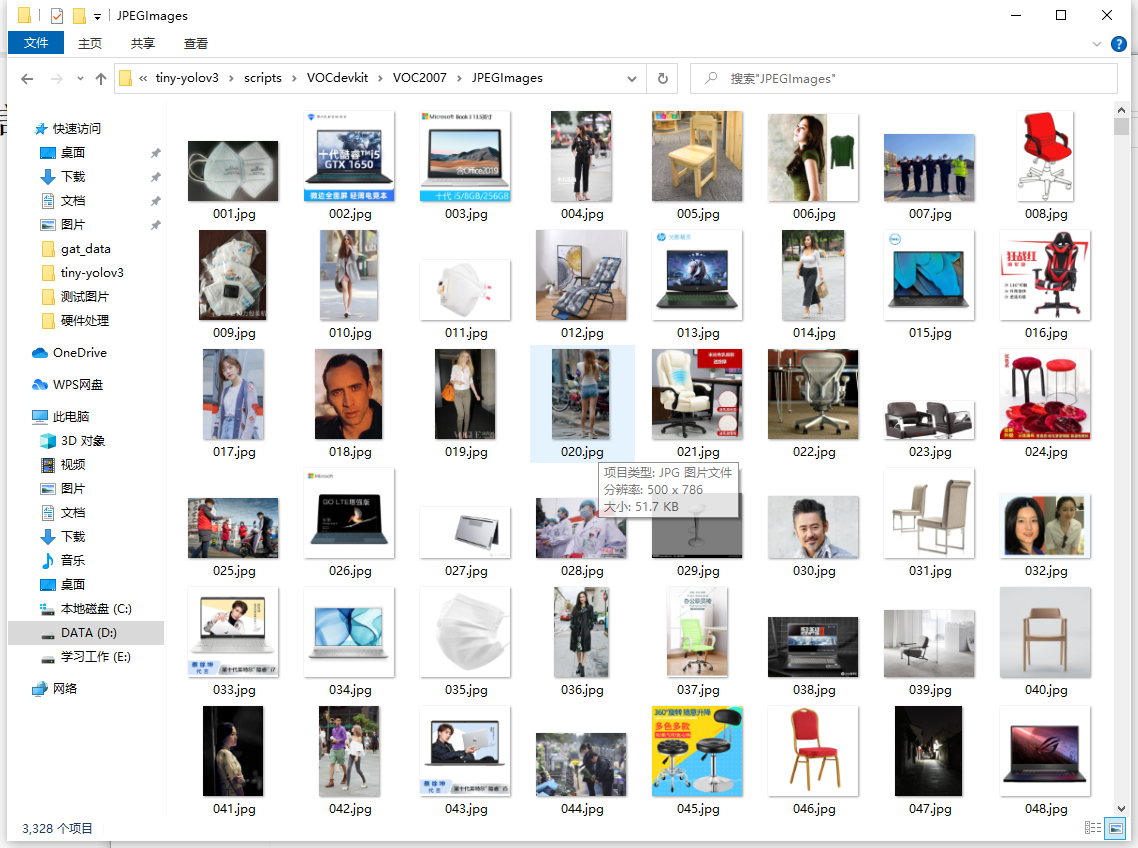
此项目yolo训练模型源代码放入网盘，有需求的可下载尝试，仅供学习，切勿商用

链接：https://pan.baidu.com/s/1xEgixEM4dlm4HF\_yfJtCJw

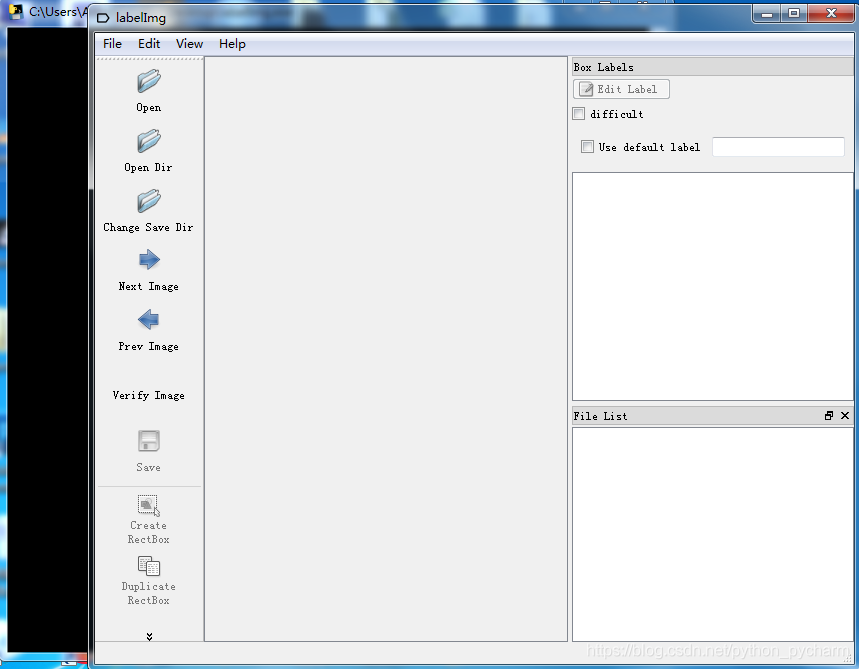
提取码：rxrv

1. 模型训练

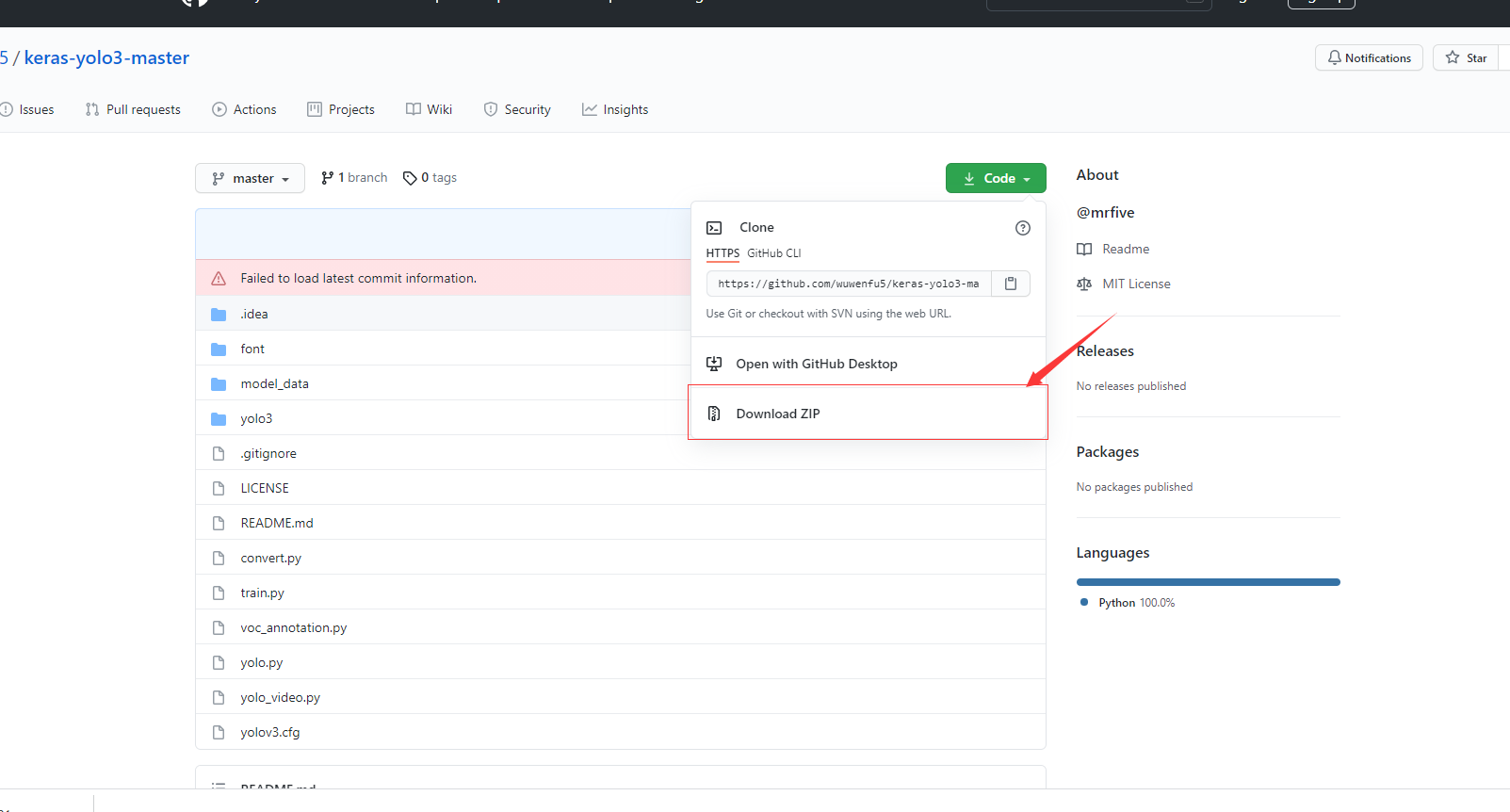
收集数据集



对数据集进行标注  
使用labeling对数据集一一进行标注

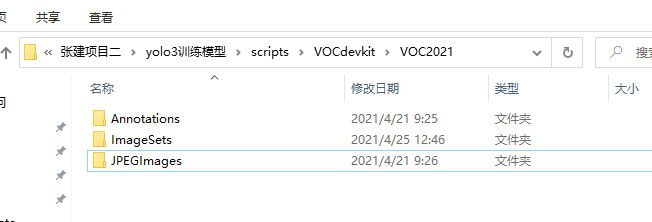


下载keras-yolo3源代码



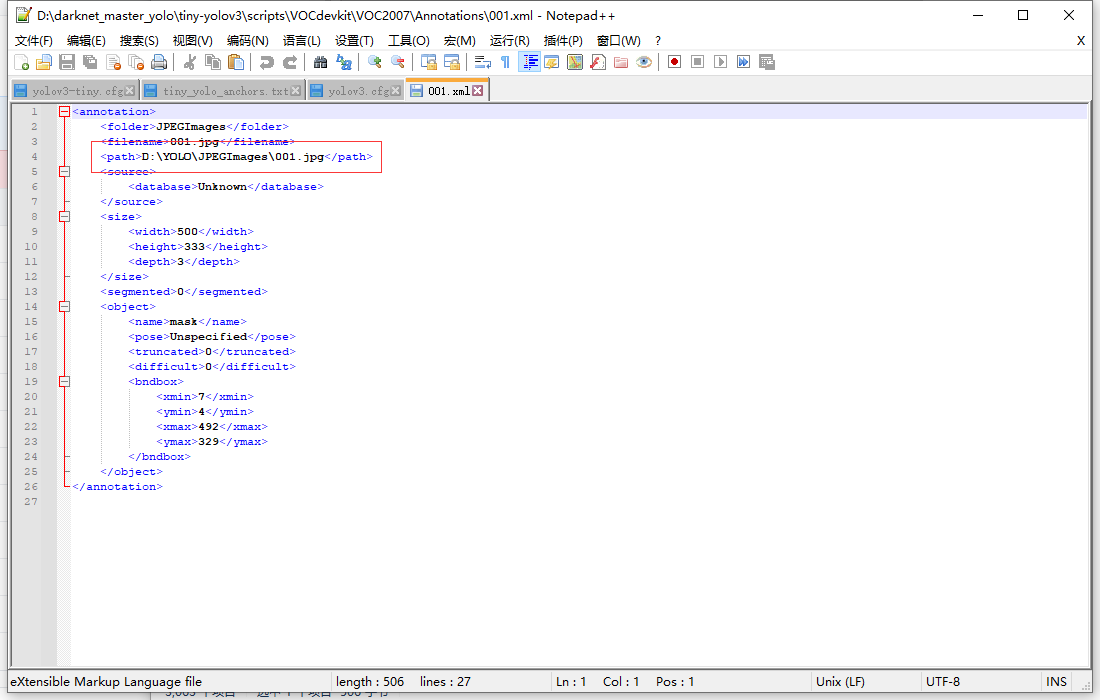
并解压

新建文件夹VOCdevkit，进入并新建VOC2021，进入并新建Annotations，ImageSets，JPEGImages

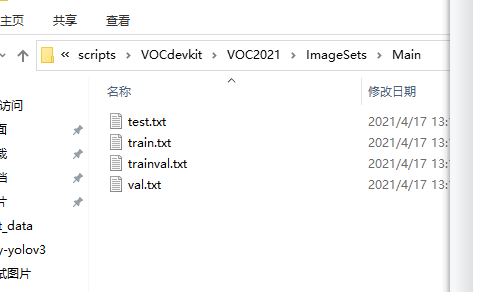


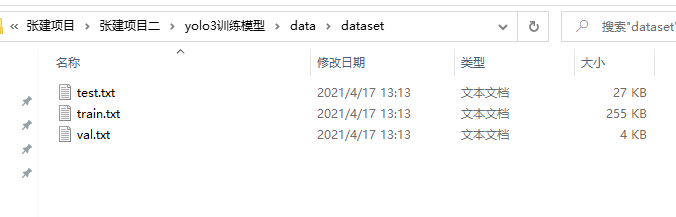
将数据集放入JPEGImages中，将标注文件放入Annotations

注：标注文件内的路径要对应数据集的路径



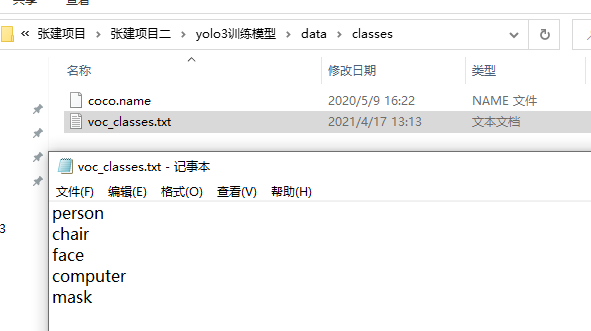
运行voc\_annotations.py后运行yolo\_annotations.py将数据写成目录



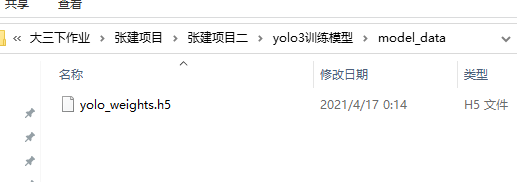


打开data/classes/voc.classes.txt

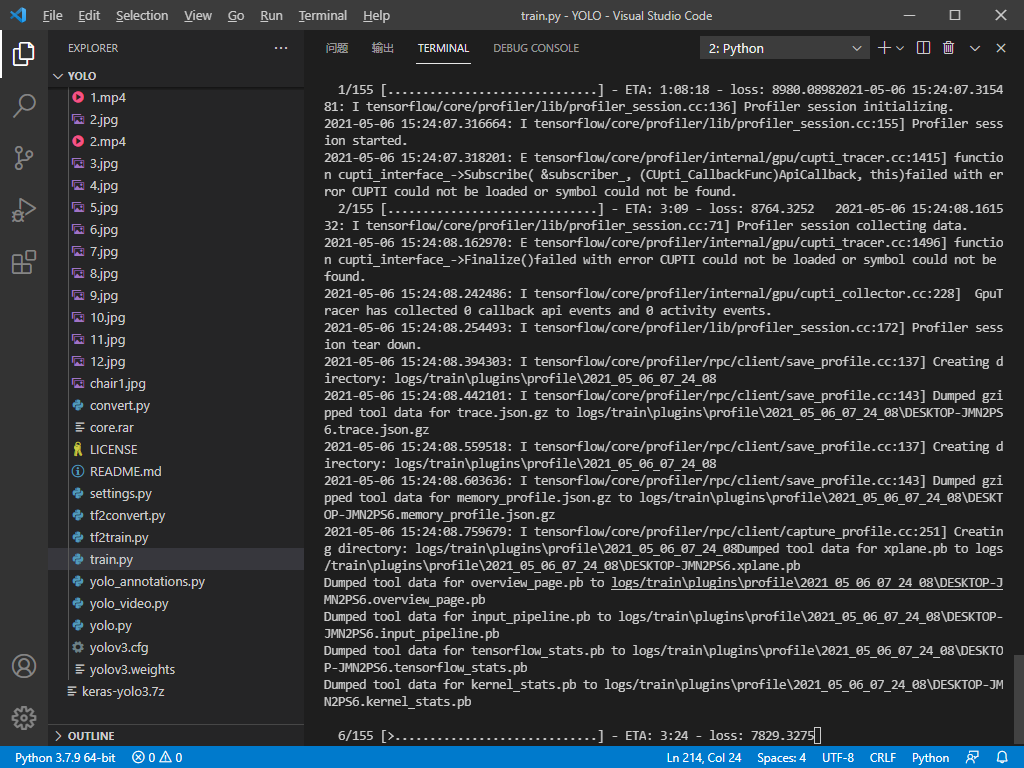
将里面的改成自己的标签



下载yolo.h5权重文件放入model\_data中



运行train.py进行训练

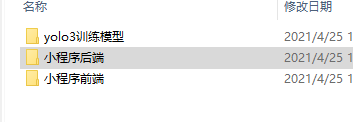


得到最后的权重文件

到目前为止模型训练完成

接下来是后端的搭建

创建文件夹——小程序后端



将训练中模型必要的文件复制到“小程序后端”的文件夹中



新建houduan.py，common\_util.py两个文件

Common\_util.py代码如下：

import numpy as np

import cv2

import base64

RESULTS = {

    200: {

        'code': 200,

        'msg': 'success'

    },

    400: {

        'code': 400,

        'msg': 'require paremeter'

    }

}

def get\_result\_model():

    return {'code': '', 'msg': '', 'data': {}, 'cost\_time': 0}

def base64\_2\_array(base64\_data):

    im\_data = base64.b64decode(base64\_data)

    im\_array = np.frombuffer(im\_data, np.uint8)

    im\_array = cv2.imdecode(im\_array, cv2.COLOR\_BAYER\_BG2BGR)

    return im\_array

def base64\_1\_array(base64\_data):

    im\_data = base64.b64encode(base64\_data)

    im\_array = np.frombuffer(im\_data, np.uint8)

    im\_array = cv2.imdecode(im\_array, cv2.COLOR\_BAYER\_BG2BGR)

    return im\_array

def result\_data(data, cost\_time=0):

    result = get\_result\_model()

    result['msg'] = RESULTS[200]['msg']

    result['code'] = RESULTS[200]['code']

    result['data'] = data

    result['cost\_time'] = cost\_time

    return result

houduan.py代码如下

from flask import Flask, request, jsonify

import json

from common\_util import \*

from yolo import \*

import os

# BASE\_DIR = os.path.dirname(os.path.dirname(\_\_file\_\_))

# static\_dir = os.path.join(BASE\_DIR, 'static')

app = Flask(\_\_name\_\_, static\_folder='static')

@app.route('/shabiyouxuan', methods=["GET", "POST"])

def get\_imagedata():

    yolo = YOLO()

    data = request.files

    file = data.get("file")

    file.save("./data/1.jpg")

    print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

    img = './data/1.jpg'

    # image = open(img, 'rb').read()

    image = Image.open(img)

    a = yolo.detect\_image(image)

    a.save('result.png')

    image\_data = open('result.png', "rb").read()

    img\_stream = base64.b64encode(image\_data)

    s = img\_stream.decode()

    return jsonify({'ok': s})

@app.route('/shabihouzi', methods=["GET", "POST"])

def get\_videodata():

    data = request.files

    file = data.get("file")

    file.save("./1.mp4")

    img = './1.mp4'

    yolo = YOLO()

    detect\_video(yolo, img, 'static/result.mp4')

    print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

    video = 'static/result.mp4'

    return json.dumps({'data': video})

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host='0.0.0.0', port='8808')

将“小程序后端”文件夹传输到服务器上并运行，后台搭建完成