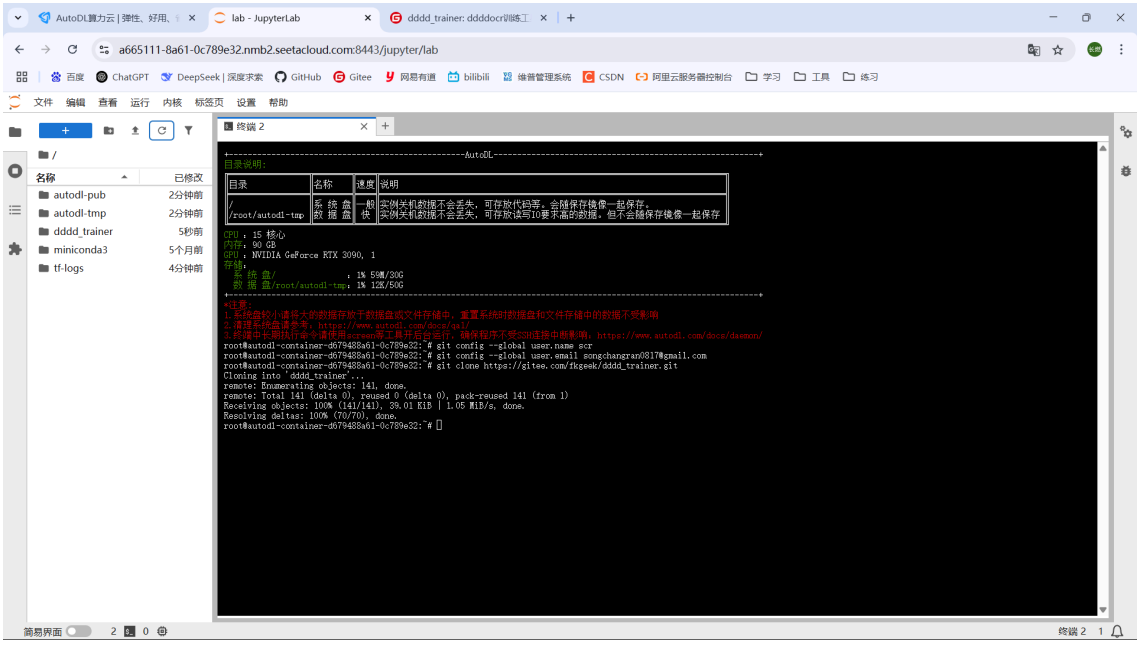
使用ddddocr训练字体提高字体识别成功率

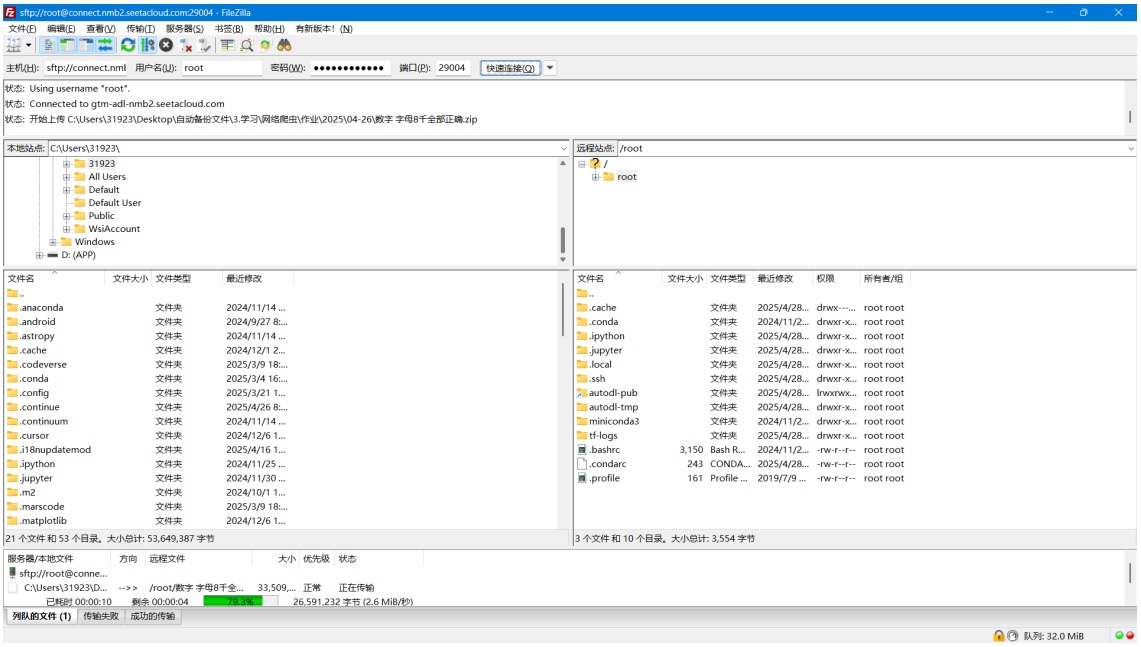
1 获取ddddocr训练工具

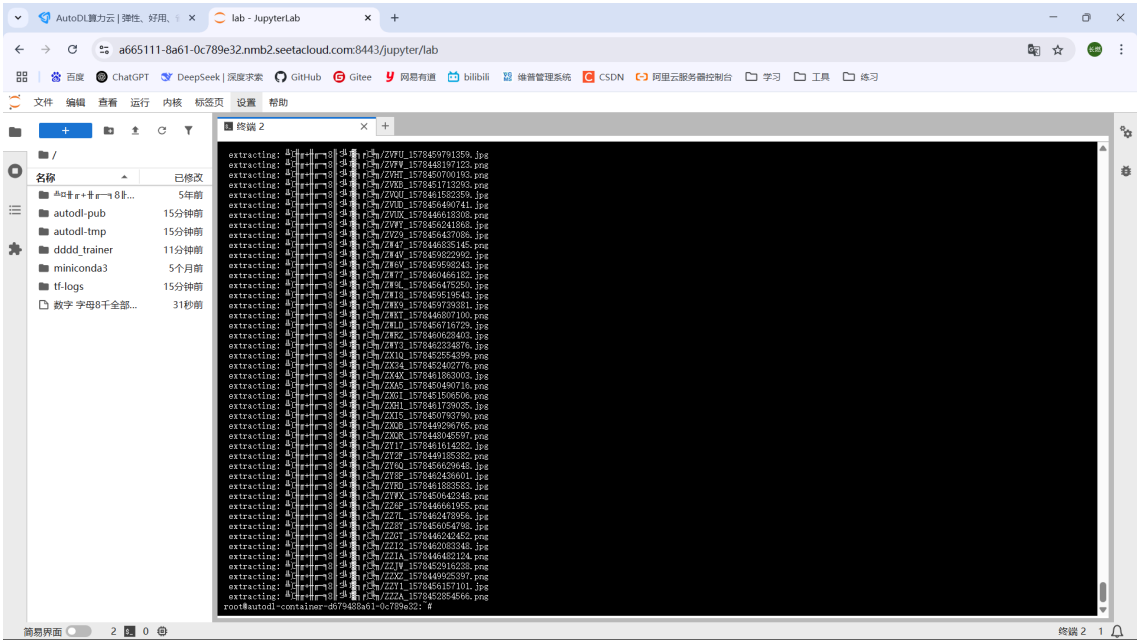
- git config --global user.name userName
- git config --global user.email userEmail
- git clone https://gitee.com/fkgeek/dddd_trainer.git

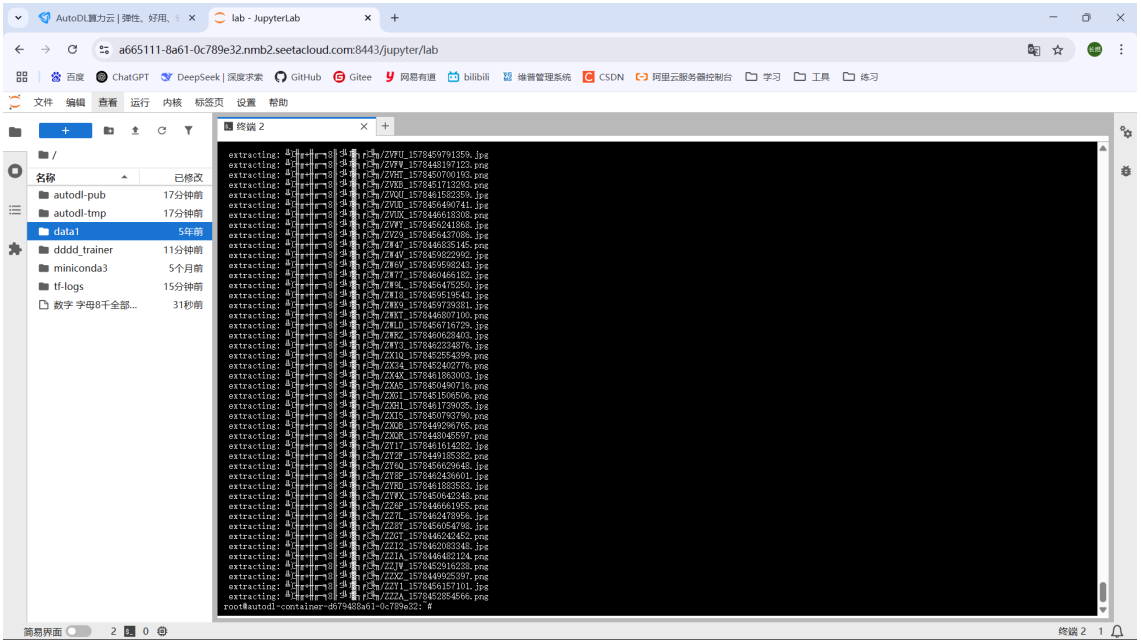


2 上传训练数据并且安装相关Python包

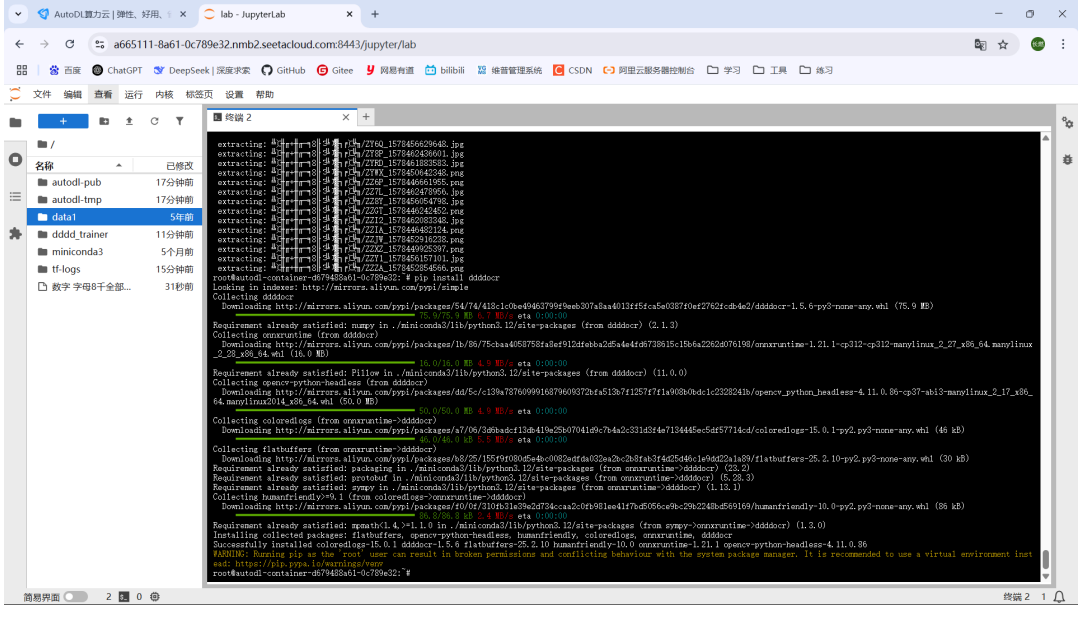
- 使用FZ上传训练数据并解压，将解压后的文件夹重命名（解压后出现乱码，防止出现错误）



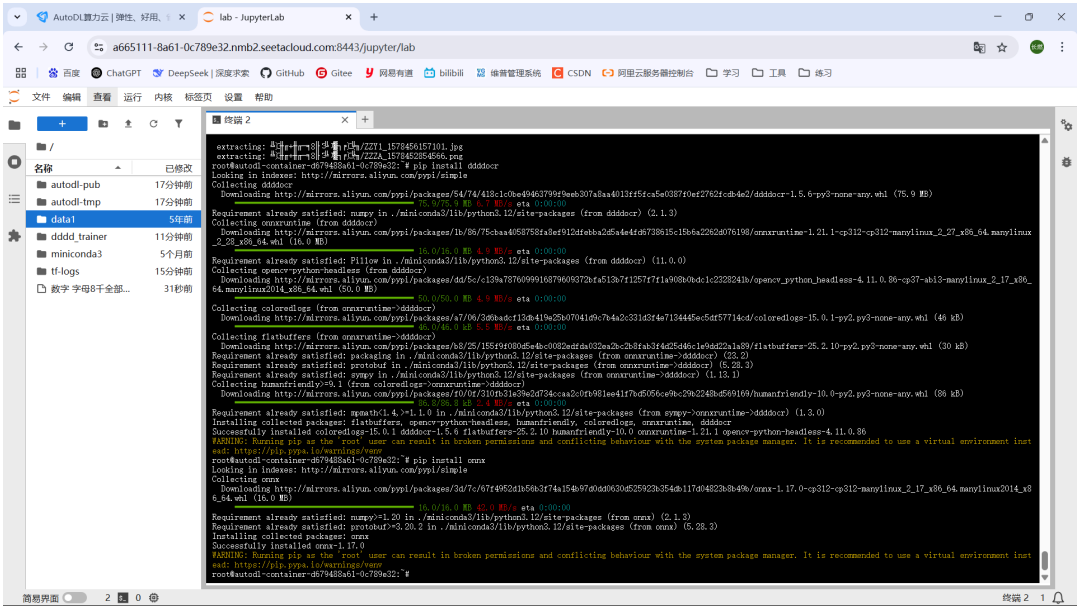




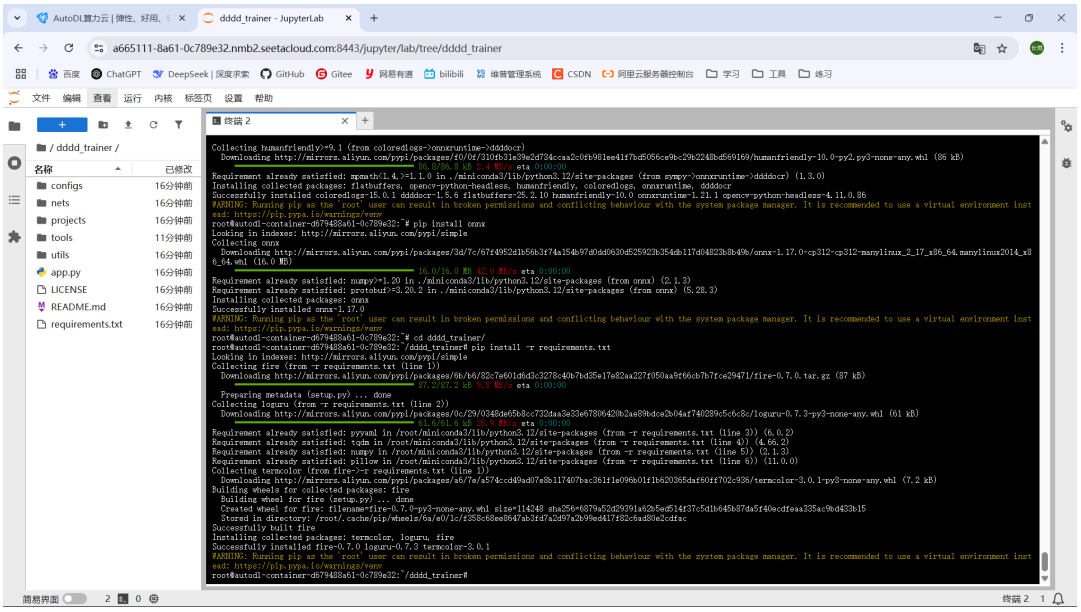
- 安装的部分包在`requirements.txt`文件里：pip install -r requirements.txt
  - pip install ddddocr



  - pip install onnx

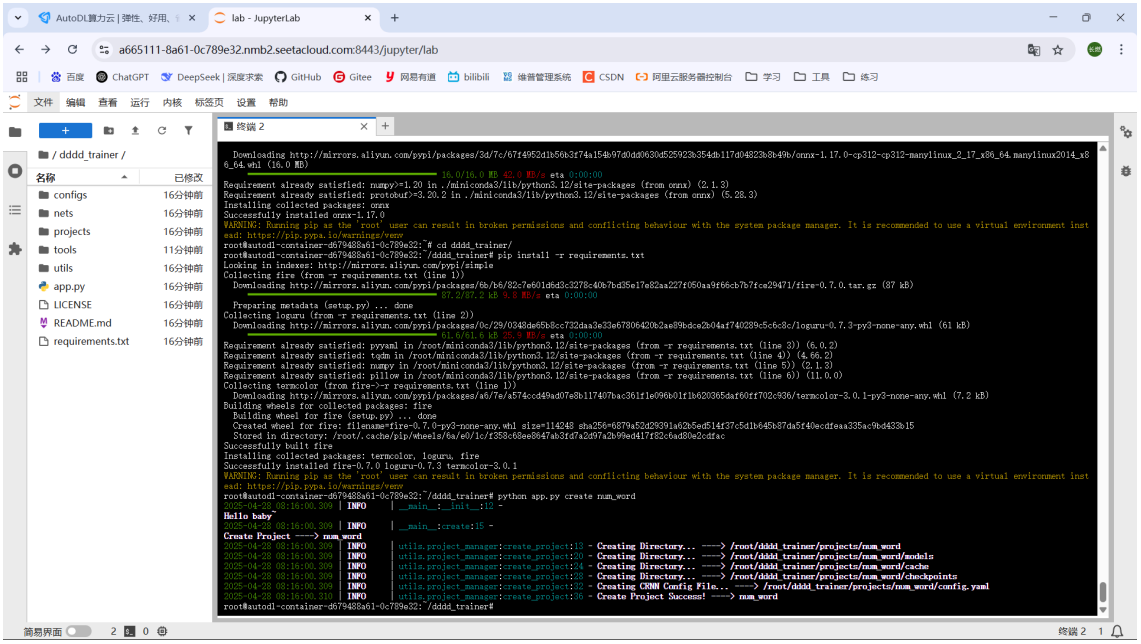


- pip install -r requirements.txt（注意：需要在`requirements.txt`文件的同级目录下执行）

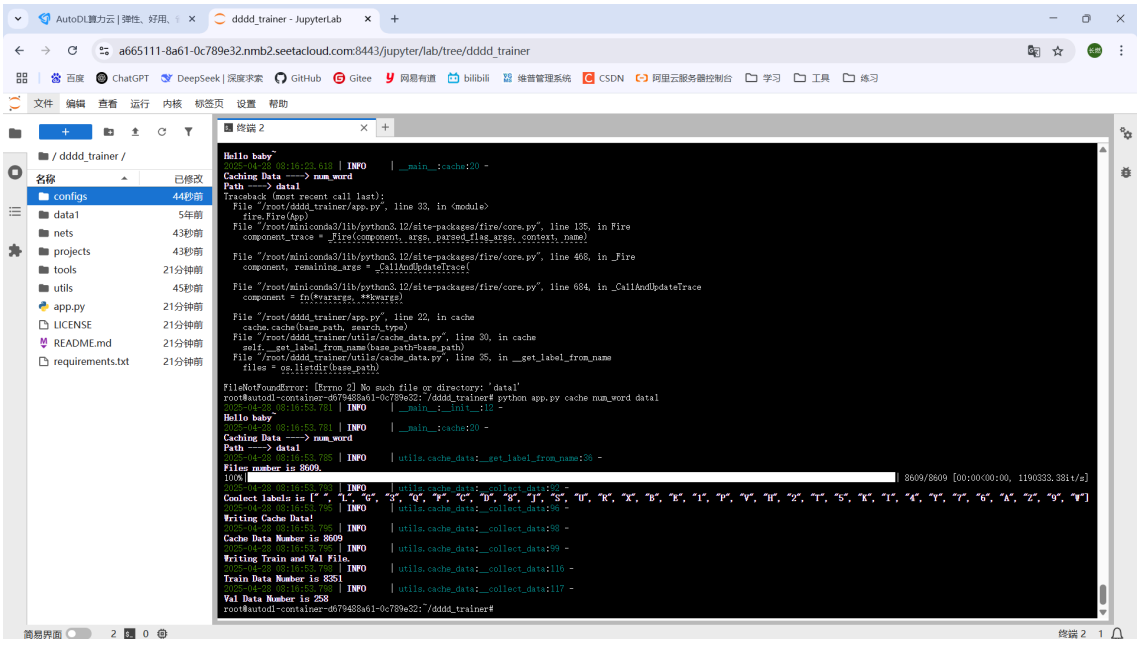


3 在训练工具里按顺序执行下述操作

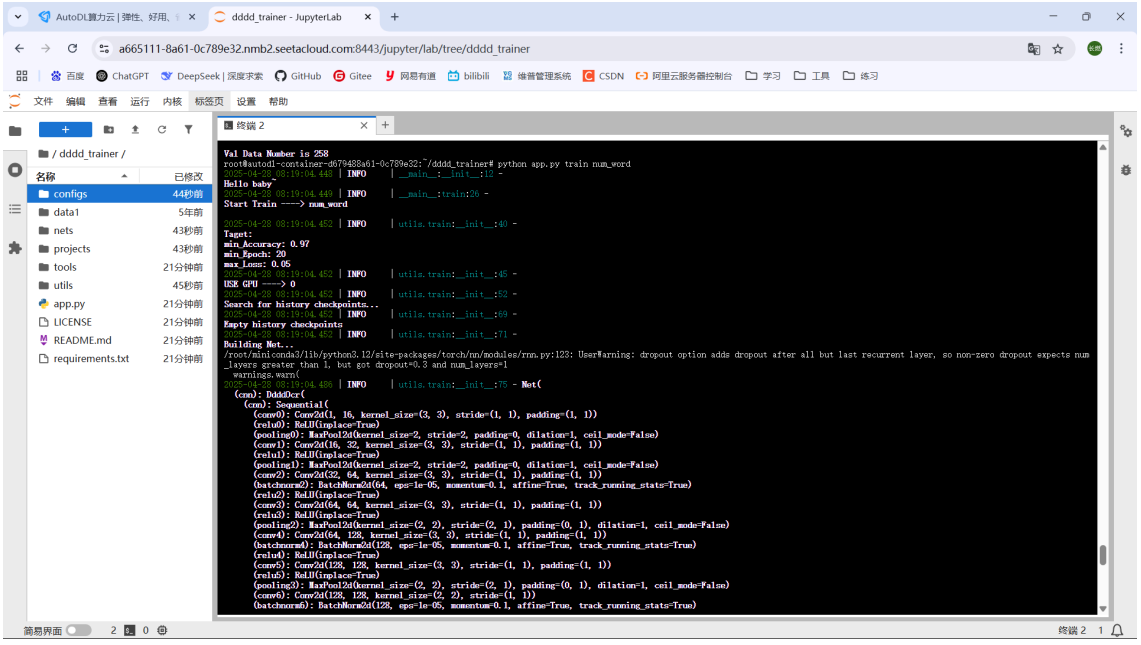
- 创建项目：python app.py create num_word

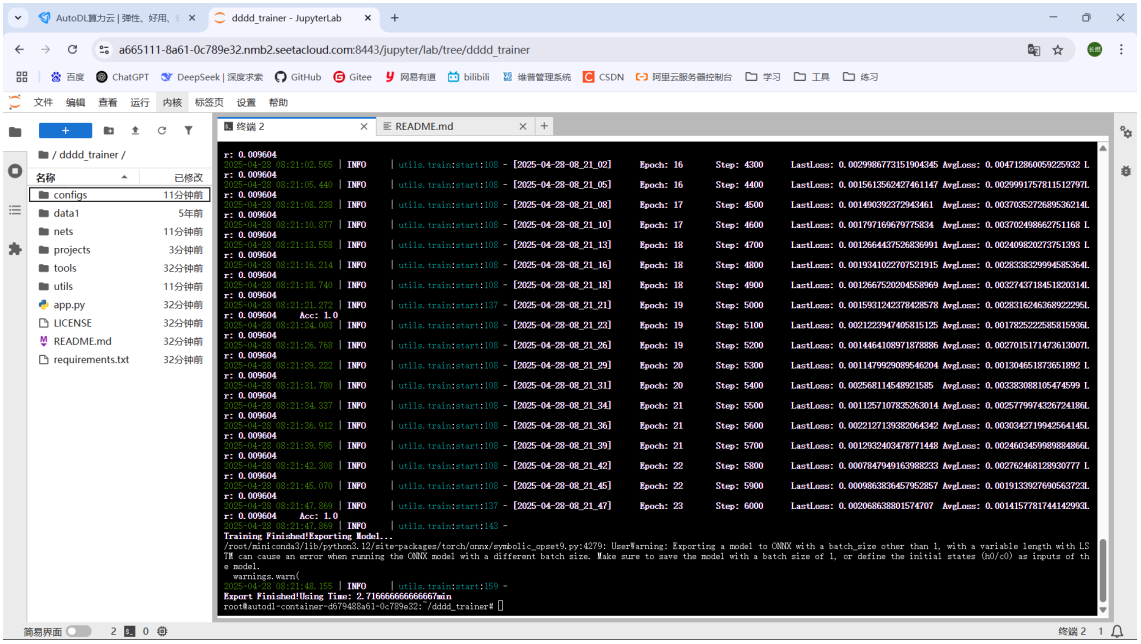


- 绑定训练资料：python app.py cache num_word data1（将解压后的训练数据移动到训练工具内，或者../data1）



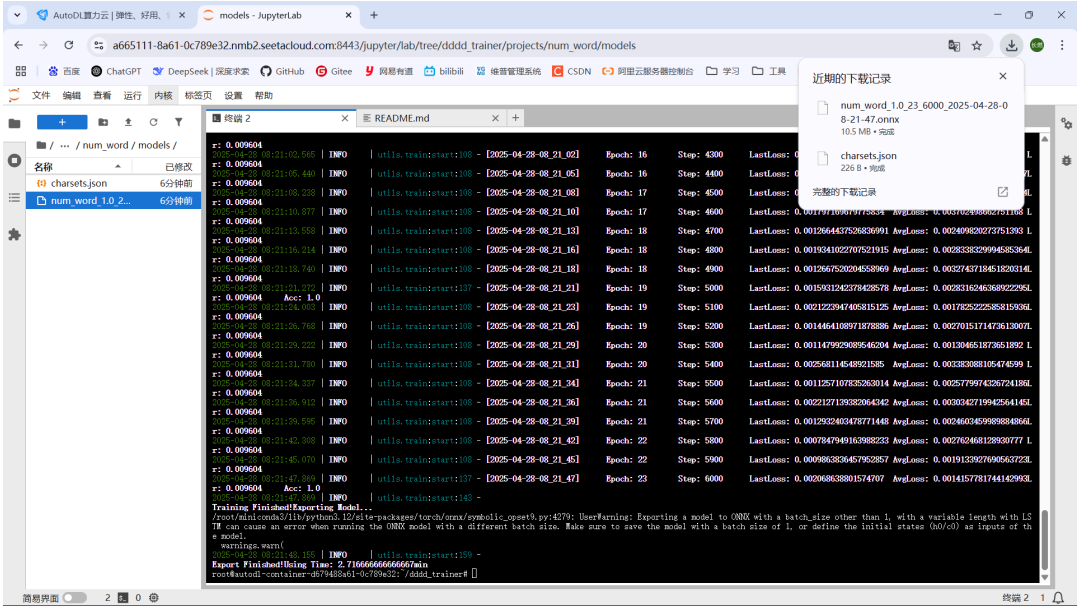
- 开启训练：python app.py train num_word（模型准确率达到95%时自动停止）





4 下载训练模型并完成准确率测试

- 下载训练模型（模型所在位置：dddd_trainer/projects/num_word/models/）
  - charsets.json
  - num_word_1.0_23_6000_2025-04-28-08-21-47.onnx



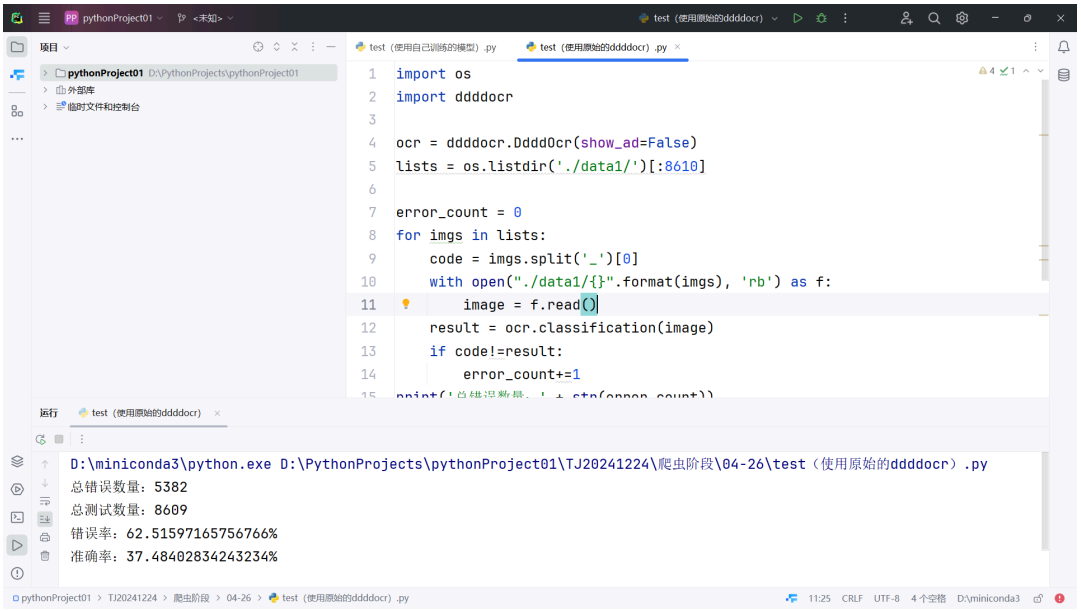
- 修改本地代码完成准确率测试（将训练模型移动到本地项目中）
  - 不使用训练模型准确率

```
import os
import ddddocr

ocr = ddddocr.DdddOcr(show_ad=False)
lists = os.listdir('./data1/')[:8610]

error_count = 0
for imgs in lists:
```

```
    code = imgs.split('_')[0]
    with open("./data1/{}".format(imgs), 'rb') as f:
        image = f.read()
    result = ocr.classification(image)
    if code!=result:
        error_count+=1
print('总错误数量：' + str(error_count))
print('总测试数量：8609')
print('错误率：' + str((error_count/8609)*100) + '%')
print('准确率：' + str(((8609-error_count)/8609)*100) + '%')
```

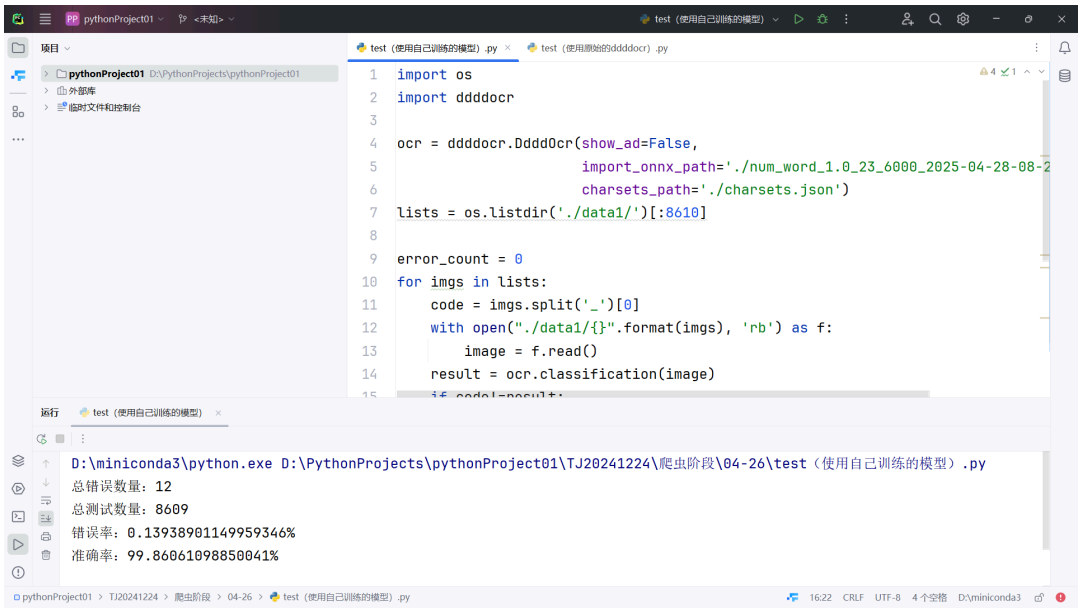


- 使用训练模型准确率：

```
import os
import ddddocr

ocr = ddddocr.DddOcr(show_ad=False,
                     import_onnx_path='./num_word_1.0_23_6000_2025-04-28-08-21-47.onnx',
                     charsets_path='./charsets.json')
lists = os.listdir('./data1/')[:8610]

error_count = 0
for imgs in lists:
    code = imgs.split('_')[0]
    with open("./data1/{}".format(imgs), 'rb') as f:
        image = f.read()
    result = ocr.classification(image)
    if code!=result:
        error_count+=1
print('总错误数量：' + str(error_count))
print('总测试数量：8609')
print('错误率：' + str((error_count/8609)*100) + '%')
print('准确率：' + str(((8609-error_count)/8609)*100) + '%')
```


pythonProject01
<未知>
test（使用自己训练的模型）
项目
pythonProject01 D:\PythonProjects\pythonProject01
外部库
临时文件和控制台
test（使用自己训练的模型）.py
test（使用原始的ddddocr）.py
import os
import ddddocr
ocr = ddddocr.DdddOcr(show_ad=False,
import_onnx_path='./num_word_1.0_23_6000_2025-04-28-08-2
charsets_path='./charsets.json')
lists = os.listdir('./data1/')[:8610]
error_count = 0
for imgs in lists:
code = imgs.split('_')[0]
with open("./data1/{}".format(imgs), 'rb') as f:
image = f.read()
result = ocr.classification(image)
运行
test（使用自己训练的模型）
D:\miniconda3\python.exe D:\PythonProjects\pythonProject01\TJ20241224\爬虫阶段\04-26\test（使用自己训练的模型）.py
总错误数量：12
总测试数量：8609
错误率：0.13938901149959346%
准确率：99.86061098850041%
pythonProject01 › TJ20241224 › 爬虫阶段 › 04-26 › test（使用自己训练的模型）.py
16:22
CRLF
UTF-8
4 个空格
D:\miniconda3