□□□paddleocr□□

用 我 所 能　为 你 而 +



软件项目集成人工智能-OCR识别

武汉微品致远信息科技有限公司

AI框架技术选项

1 AI框架

PaddlePaddle	百度开源，文档较少，商用时候可以付费联系百度支持，推荐
PyTorch	Facebook开源，社区活跃，很多基于PyTorch的模型算法，没有付费支持，需要硬实力。

2 AI模型

YOLO	基于PyTorch框架训练的目标检测模型，当前覆盖面最全，检测速度最快，推荐
PaddleOCR	基于Paddle框架训练的OCR识别模型，速度快，覆盖面券，推荐

3 开发环境

编程语言	Python，AI行业事实上的主流编程语言
IDE	vscode，借助github copilot插件智能提示，让你如虎添翼
RESTful API	fastAPI，轻量、简洁的高效RESTful API框架，承载模型运算推理结果

PaddleOCR

1. **step 1**

   安装paddlepaddle框架

2. **step 2**

   加载paddleocr 模型

3. **step 3**

   RESTful API方式发布为OCR识别服务

参考网址

https://www.paddlepaddle.org.cn/
https://github.com/PaddlePaddle/PaddleOCR
https://fastapi.tiangolo.com/

□□□□□

- paddle-ocr.zip

□□□

- https://www.paddlepaddle.org.cn/

- https://github.com/PaddlePaddle/PaddleOCR
- https://fastapi.tiangolo.com/

□□paddle□paddle ocr□□

```
# □□paddlepaddle
# https://www.paddlepaddle.org.cn/
pip install paddlepaddle==2.6.1 -i https://pypi.tuna.tsinghua.edu.cn/simple

# □□paddleocr
# □□□□□□2024□5□2□□□□□python□□□3.11.□□□□□□□□□□□□□□□□
# □□□□□□2024□5□2□□paddleocr==2.7.5□bug
# □□□□□□□□
#  C:\Python311\Lib\site-packages\paddleocr\paddleocr.py
# □54□□□
# from ppstructure import predict_system



pip install paddleocr==2.7.5 -i https://mirror.baidu.com/pypi/simple



# □□□□□□□□□□□□□□□□□□□□□□□□,□□pip□□
pip install --upgrade pip wheel setuptools
```

□□fast api□□□□

```
# □□fastAPI
pip install fastapi -i https://pypi.tuna.tsinghua.edu.cn/simple

# □□web□□
pip install "uvicorn[standard]" -i https://pypi.tuna.tsinghua.edu.cn/simple

# □□from□□□□
pip install python-multipart -i https://pypi.tuna.tsinghua.edu.cn/simple
```

□□□□□□

```
import paddle as paddle
from paddleocr import PaddleOCR, draw_ocr
```

```
from PIL import Image
from fastapi import FastAPI, File, UploadFile
from fastapi.responses import HTMLResponse
from fastapi.staticfiles import StaticFiles
import time
import uvicorn

# □□paddle□□
print(paddle.__version__)

app = FastAPI()



def get_image_ocr(img_path: str):
    # □□□□OCR□□
    ocr = PaddleOCR(
        use_angle_cls=True, lang="ch"
    )  # need to run only once to download and load model into memory
    # img_path = "./images/test.png"
    # □□PaddleOCR□□□□□□□□□
    ocr_result = ocr.ocr(img_path, cls=True)
    # □□□□
    for line in ocr_result:
        print(line)

    # □□□□□□□□□□□□□
    result = ocr_result[0]
    image = Image.open(img_path).convert("RGB")
    boxes = [elements[0] for elements in result]
    # pairs = [elements[1] for elements in ocr_result]
    # txts = [pair[0] for pair in pairs]
    # scores = [pair[1] for pair in pairs]
    im_show = draw_ocr(image, boxes)
    im_show = Image.fromarray(im_show)
    im_show.save(f"{img_path}")
    # im_show.save("./images/result.jpg")
    return {"image": img_path, "result": result}



app.mount("/static", StaticFiles(directory="./static"), name="static")
```

```
@app.get("/")
def read_root():
    html_content = """
    <html>
        <head>
            <title>OCR</title>
        </head>
        <body>
            <h1>OCR</h1>
            <p>Upload an image to extract text from it.</p>
        </body>
    </html>
    """
    return HTMLResponse(content=html_content, status_code=200)


sequence_counter = 1


@app.post("/ocr/")
async def ocr_image(file: UploadFile | None = File(None)):
    # □□□□□□□□
    if file is None:
        return {"code": 400, "message": "No file found", data: None}

    # □□□□□□□□□□□□□□□□□
    current_time = int(time.time() * 10000)
    unique_sequnce = current_time + sequence_counter
    new_fileName = f"./static/{unique_sequnce}--{file.filename}.jpg"
    # □□□□□□□
    with open(f"{new_fileName}", "wb+") as fsio:
        fsio.write(await file.read())
    pass
    # □□OCR□□
    ocr_result = get_image_ocr(new_fileName)

    return {"code": 200, "message": "ok", "data": ocr_result}


# □□□□
```

```
uvicorn.run(app, host="0.0.0.0", port=8000)
```

□□□ #2
□ □□□ □□□ 7 □□ 2024 15:58:12
□ □□□ □□□ 18 □□ 2025 03:45:38