



# 智能文字识别和文档处理项目

## Umi-OCR

2024-12

# 目录

## CONTENTS

01

项目介绍

02

分工、协作  
、管理

03

系统设计

04

模块分析

05

成果展示



01

## 项目介绍

## 项目介绍

---

在现代数字化时代，文字识别（OCR）技术变得越来越重要。随着办公自动化和数字化转型的需求增长，快速、准确地从图像、截图、文档中提取文字成为许多用户和工作场景的迫切需求。传统的OCR工具往往**操作复杂、效率低下**。

Umi-OCR的主要目的是开发一款开源、免费、高效的文字识别工具





# 02

成员分工、团队协作、  
项目管理

# 成员介绍

第2组	2022330300160	韩圣杰	系统架构师 测试研发工程师	文字识别
	2022331201045	雷心成	算法工程师 研发工程师	
	2022333500054	褚昊	研发工程师 UI设计师	
	2022334323029	张雅瑞	项目经理-组长 产品经理	

## 职责分工

8.2沟通内容记录

时间	主题	内容概要	参与人员	记录
12.11	确认选题	确认选题	全体成员	
12.13	分工设计	发布确认任务，开始相关学习	全体成员	
12.18	前端设计	设计和开始前端任务	全体成员	
12.18	后端设计	设计和开始后端	全体成员	
12.23	相关部分整合	将代码整合	全体成员	
12.24	测试	将测试工作开展	全体成员	
12.25	文档撰写	相关人员进行文档的撰写	全体成员	

## 沟通记录

姓名	褚昊	韩圣杰	雷心成
评分	98	97	97

表 10-1 团队成员评分表

## 组员评价

8. 2沟通内容记录

时间	主题	内容概要	参与人员	记录
12. 11	确认选题	确认选题	全体成员	
12. 13	分工设计	发布确认任务，开始相关学习	全体成员	
12. 18	前端设计	设计和开始前端任务	全体成员	
12. 18	后端设计	设计和开始后端	全体成员	
12. 23	相关部分整合	将代码整合	全体成员	
12. 24	测试	将测试工作开展	全体成员	
12. 25	文档撰写	相关人员进行文档的撰写	全体成员	

沟通记录

分工：

- A、B、C、D四个人
- A——报告整理、报告细节部分
- B、C——代码分析、模块分析
- D——PPT、汇报

提交内容：

- 代码
- PPT
- 报告
- 汇报

报告撰写：

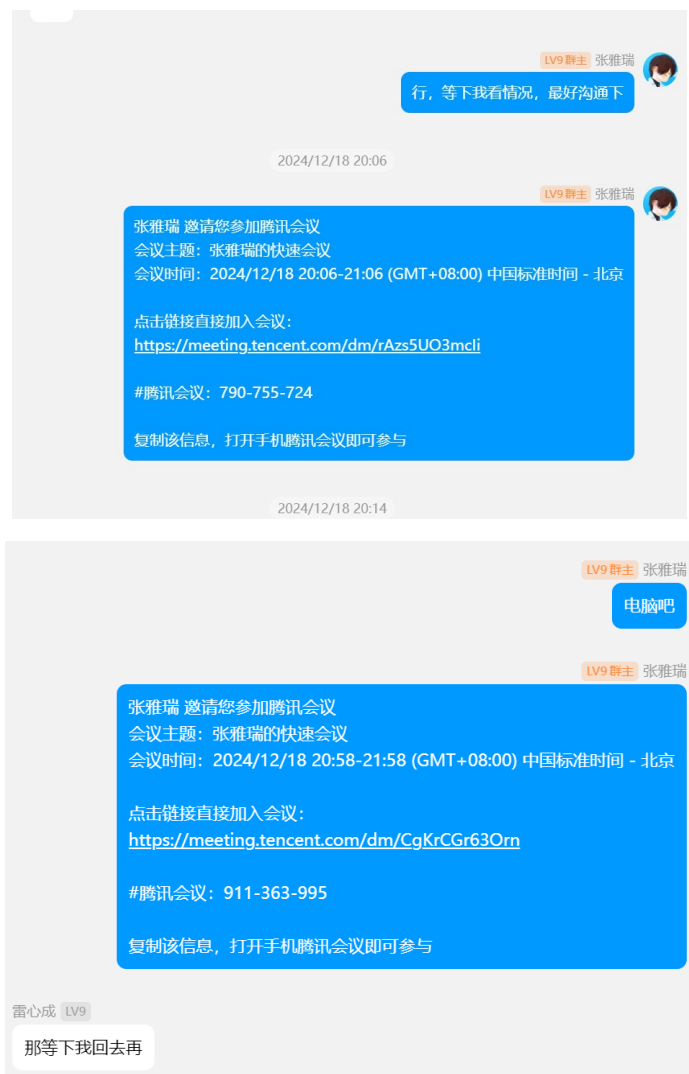
- 需求分析——写项目背景分析——A写，要画图
- 设计分析——分析前后端的技术栈——前端B、后端C
- 模块设计——分析每个模块的功能——前端模块B、后端模块C
- 引言、项目分工——A写
- 报告格式统一——A写

PPT：D做

汇报：D

分工安排

# 团队协作



沟通记录



进度安排



03

系统设计

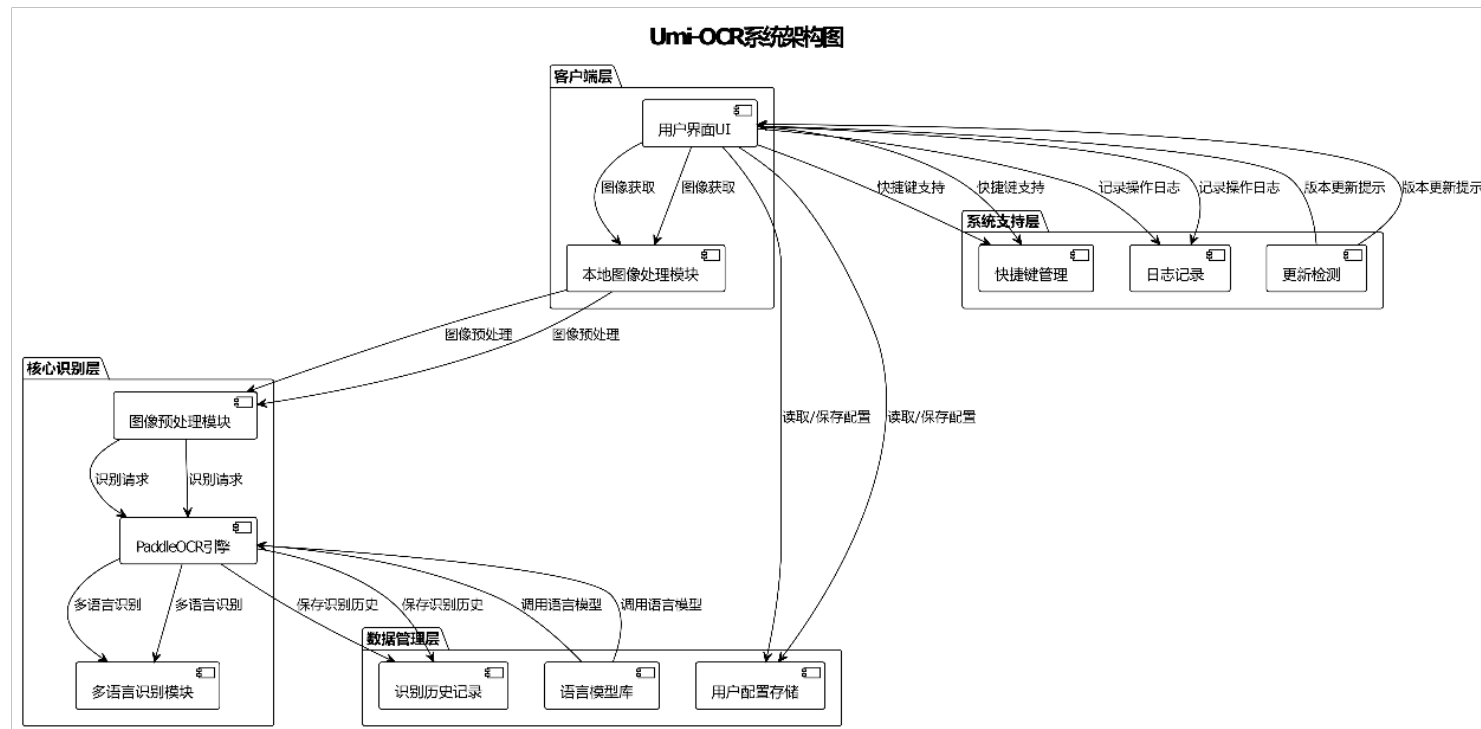
# 系统设计

前端设计:

- QtQuick (Python)

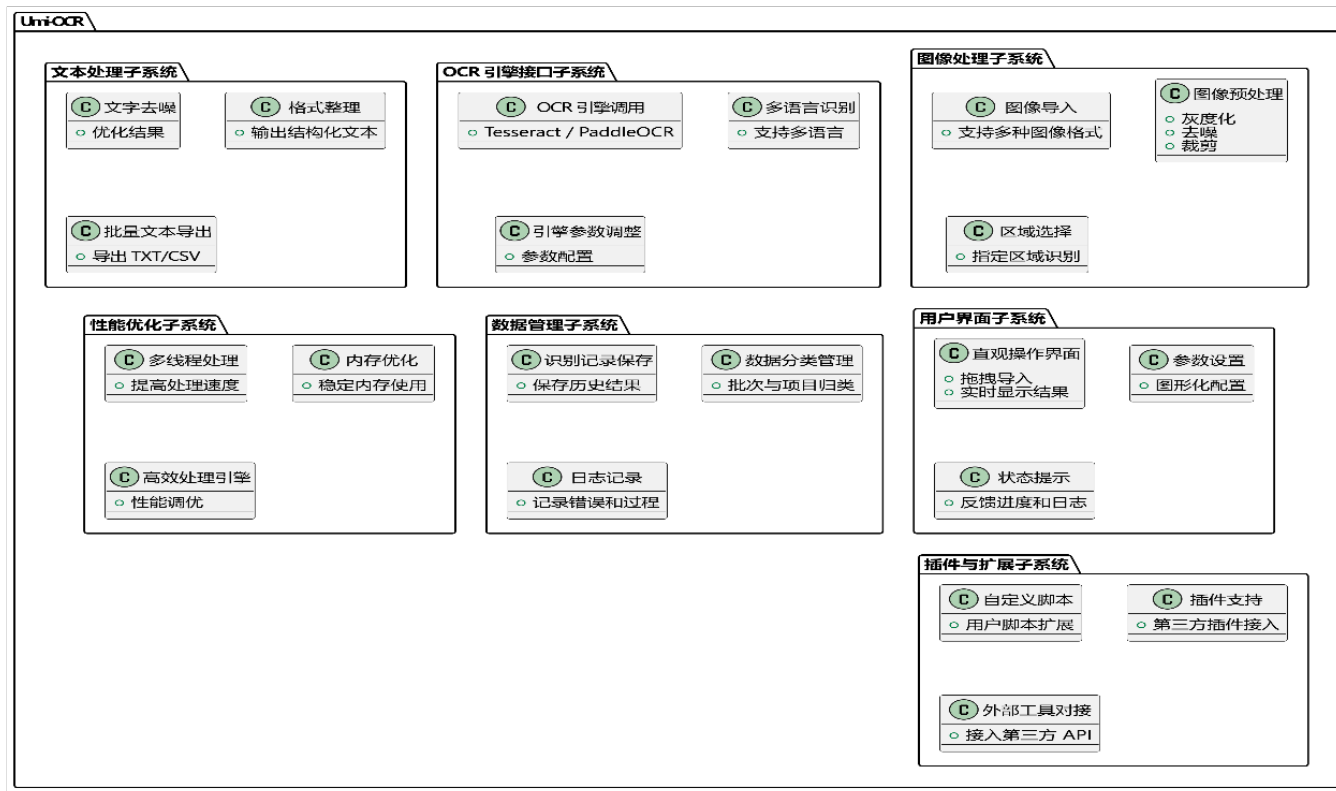
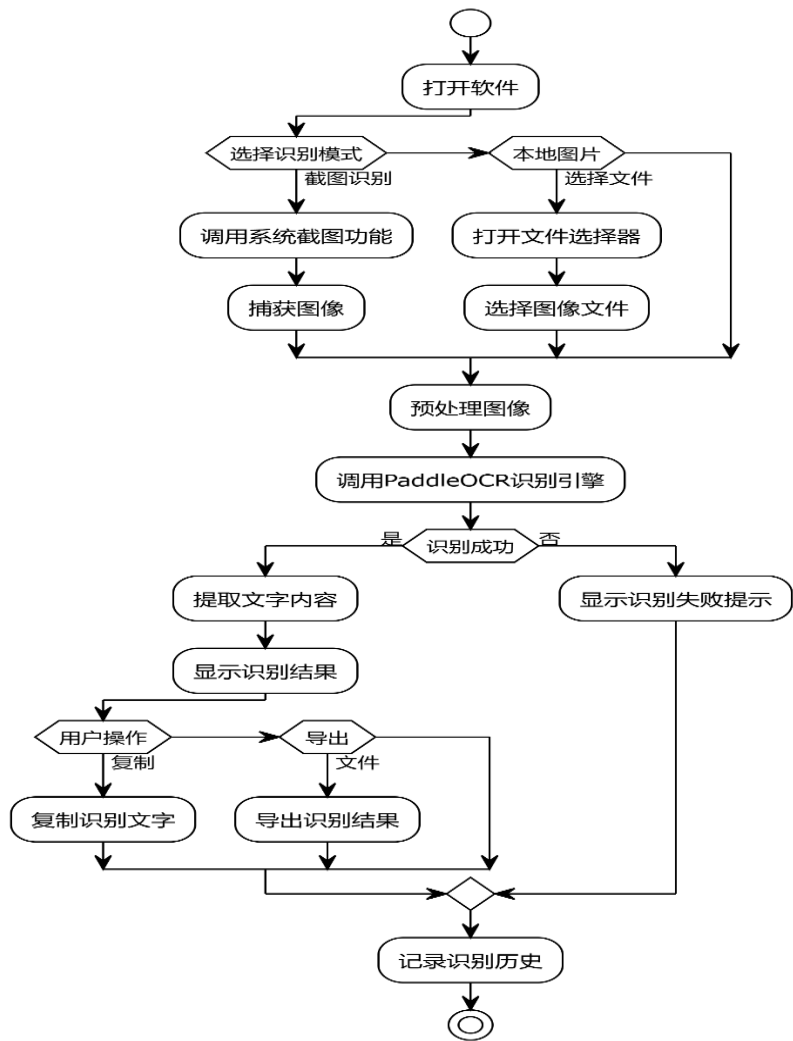
后端设计:

- Bottle框架  
(Python Web)
- PaddleOCR (OCR api )





Umi-OCR软件流程图



模块框架图



04

模块设计

# OCR识别

用户可通过截图、拖拽或粘贴图片进行OCR识别，管理批量任务，显示进度，并支持暂停、恢复和停止。

## 1. 截图、拖入、粘贴图片处理:

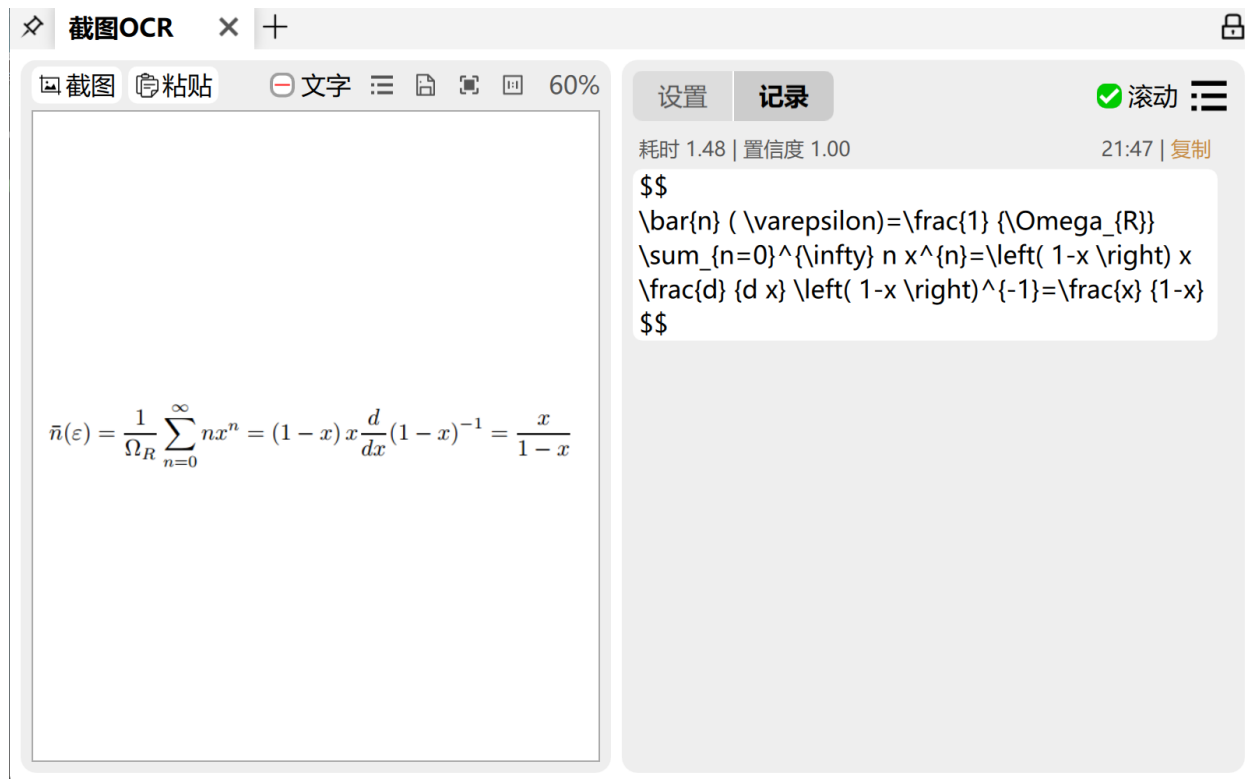
$$\text{ImageCapture} = \begin{cases} \text{screenshot}() \Rightarrow \text{handleScreenshot}(\text{clipID}) \\ \text{paste}() \Rightarrow \text{handlePaste}(\text{res}) \\ \text{DropArea\_} \Rightarrow \text{tabPage.ocrPaths}(\text{paths}) \end{cases} \quad (1)$$

## 2. 图片预览和编辑:

$$\text{ImageProcessing} = \begin{cases} \text{previewImage}(\text{imageID}) \\ \text{cropImage}(\text{x}, \text{y}, \text{width}, \text{height}) \Rightarrow \text{handleCrop}(\text{croppedImageID}) \end{cases} \quad (2)$$



# 公式识别



基于 Pix2Text 引擎，支持从图像中精准识别数学公式（输出为 LaTeX 格式）和混合文本，通过模块化设计与标准化接口，实现高效的公式提取与文本整合。

# 二维码识别



```
# 预处理图像
if "argd" in msnInfo:
    img = self.preprocessing(img, msnInfo["argd"])

# 调用二维码解析
try:
    codes = zxingcpp.read_barcode(img)
except Exception as e:
    return {"code": 204, "data": f"【Error】二维码解析失败。 {e}"}

# 转换解析结果为统一格式
return self._zxingcpp2dict(codes)
```

二维码识别模块通过解析图片中的二维码，支持路径、PIL对象和Base64数据作为输入，利用 zxingcpp 解析二维码内容与位置信息，提供多任务管理和图像预处理功能。



05

成果展示



## ○ 成果展示

---





感谢您的观看