

第 2 周：AI 辅助编程工具实战

怎么让 AI 帮我写代码？

王文通

北京石油化工学院
BEIJING INSTITUTE OF PETROCHEMICAL TECHNOLOGY



北京石油化工学院
人工智能研究院

2025-2026 学年

北京石油化工学院



AI 辅助编程工具实战



课程概览

本周内容：

- ① LLM 时代的编程新范式
- ② AI 编程工具全景
- ③ Prompt 工程精讲
- ④ CV 领域深度实战
- ⑤ 调试、重构与安全

核心技能：

- 结构化 Prompt 设计 (RTF)
- 思维链 (Chain-of-Thought)
- 少样本提示 (Few-shot)
- 多轮迭代优化
- 代码审查与验证

本周作业

用 AI 辅助实现 答题卡边界检测，提交完整的 AI 对话记录、代码和测试报告



教学安排

3 课时时间分配

时间	环节	内容
09:00-09:20	理论讲解	AI 编程范式与工具发展史
09:20-09:40	现场演示	Cursor IDE 实战演示
09:40-10:20	实践环节	Prompt 工程练习（小组协作）
10:20-10:30	课间休息	休息 10 分钟
10:30-11:00	小组讨论	案例分析与代码优化
11:00-11:15	难点突破	调试与重构技巧
11:15-11:30	成果分享	小组展示与总结



三种参与模式

观察者模式

适合: 编程基础较弱的同学

- 理解 AI 编程概念
- 观看教师演示
- 理解 Prompt 设计原理

目标: 能用自然语言描述需求

使用者模式

适合: 有一定编程基础的同学

- 使用 AI 工具生成代码
- 运行并测试代码
- 调整参数优化结果

目标: 能独立完成基础任务

创造者模式

适合: 编程基础较好的同学

- 设计高质量的 Prompt
- 修改和优化 AI 生成的代码
- 探索创新解决方案

目标: 能设计和优化系统

提示: 本周作业提供基础版/进阶版/挑战版, 可根据自身水平选择



小组协作安排

分组原则

- 每 4 人一组，不同专业背景混合
- 确保每组至少有一名有编程基础的同学
- 鼓励强弱搭配，互相帮助

角色分工：

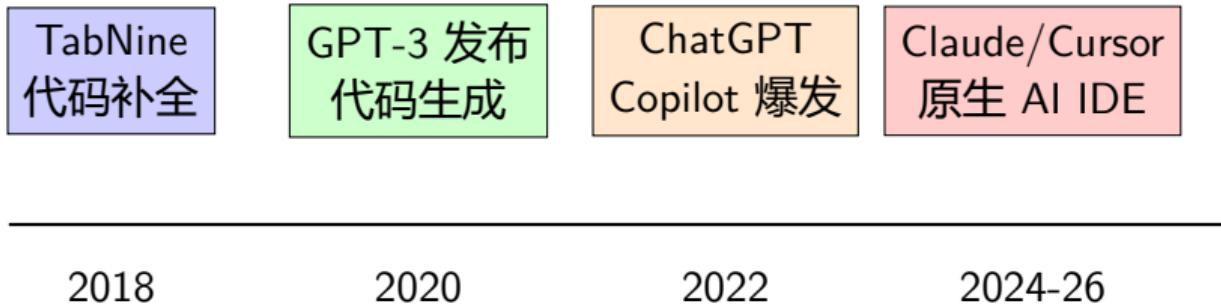
- ① **组长**: 协调进度，分配任务
- ② **技术负责人**: 把关代码质量，审核 AI 生成的代码
- ③ **开发者 A**: 负责 Prompt 设计
- ④ **开发者 B**: 负责代码测试与优化

协作建议：

- 观察者模式同学可担任组长
- 使用者模式同学担任开发者
- 创造者模式同学担任技术负责人
- 鼓励角色轮换，体验不同角色



AI 编程工具发展史：从传统 IDE 到 AI 助手



里程碑事件

- 2018: TabNine 首次将深度学习用于代码补全
- 2020: GPT-3 展示强大的代码生成能力
- 2022: ChatGPT 和 GitHub Copilot 引爆 AI 编程浪潮
- 2024-26: Claude、Cursor 等原生 AI IDE 重新定义编程体验



行业影响：编程效率提升 10 倍 +

GitHub Copilot 统计数据 (2024):

- 全球开发者超过 1300 万
- 代码接受率约 30-40%
- 编码速度提升 55%
- 开发者满意度 75%+

Cursor 用户反馈:

- 代码重构效率提升 10 倍
- 新功能开发周期缩短 50%
- 代码质量显著改善

对开发者的影响:

- ① 从“记忆”到“理解”
 - 不再需要记忆大量 API
 - 更需要理解原理和设计
- ② 从“编写”到“审查”
 - AI 生成初稿
 - 人审查、修改、优化
- ③ 从“实现”到“设计”
 - 更多时间用于架构设计
 - 更少时间用于编码细节



AI 编程辅助的优势

1. 快速生成代码框架

- 根据描述生成完整函数
- 自动生成类结构
- 提供多种实现方案

3. 提供优化建议

- 性能优化方案
- 代码重构建议
- 最佳实践推荐

2. 解释错误原因

- 分析报错信息
- 定位问题根源
- 提供修复建议

4. 降低学习门槛

- 即时答疑解惑
- 提供学习路径
- 降低技术门槛



回顾：第 1 周的挑战

还记得这些痛点吗？

- cv2.imread() 返回 None，找不到原因
- 图像显示颜色异常
- 数组越界、维度不匹配
- API 参数太多，记不住

典型错误代码

```
1 # 死循环！
2 while img is not None:
3     cv2.imshow('image', img)
4     if cv2.waitKey(1) == 'q':
5         break
```

传统解决方式：

- ① 翻文档 (docs.opencv.org)
- ② 搜索 StackOverflow
- ③ 问同学/老师
- ④ 试错（耗费大量时间）

AI 辅助的新方式

- 直接问 AI
- 解释错误原因
- 给出修改建议
- 降低学习门槛！



大模型写代码的本质

Token 预测

+

模式匹配

为什么 AI 擅长语法但不一定懂逻辑？

- **擅长：**基于海量代码库的模式复刻
- **不擅长：**理解你的具体业务逻辑



幻觉专题

什么是 AI 幻觉？

AI 自信满满地生成看似合理但实际上错误或不存在的信息。

案例：AI 发明了一个不存在的 OpenCV 函数

AI 的“创意”代码

```
1 import cv2
2
3 img = cv2.imread('exam.jpg')
4
5 # 不存在的函数！
6 fixed = cv2.auto_fix_exposure(img)
```

如何识别幻觉？

查官方文档 (docs.opencv.org)



AI 辅助编程工具实战

编程范式的演进

1.0 阶段：查阅文档/StackOverflow

- 手动搜索
- 阅读文档
- 试错调试

效率：两星

2.0 阶段：AI 辅助

- 从“写作者”转变为“审查者”
- AI 生成初稿，人审查修改
- 快速原型验证

效率：四星

3.0 阶段：AI 原生？

- AI 直接理解需求生成完整系统
- 人主要负责需求定义和架构设计
- 可能的未来方向

效率：五星？

核心结论：程序员的核心竞争力转变

从“记忆语法”转向“问题定义”与“代码审查”



不同专业背景的学习路径

理工科背景

优势：编程基础较好

- 直接进入使用者/创造者模式
- 重点学习 Prompt 工程技巧
- 探索复杂场景的 AI 协作

文科/经管背景

优势：表达与逻辑能力

- 从观察者模式开始
- 发挥自然语言优势设计 Prompt
- 逐步掌握基础代码理解

艺术/设计背景

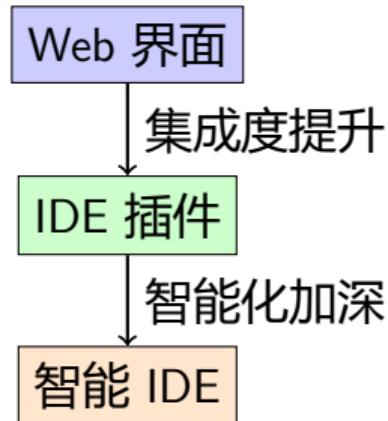
优势：视觉敏感度高

- 适合图像处理任务
- 关注可视化结果
- AI 辅助实现创意想法

AI 辅助编程降低了技术门槛，不同专业都能找到适合自己的切入点



AI 编程工具全景图



选择建议

- **学习理解:** ChatGPT/Claude (对话深入)
- **实时编码:** GitHub Copilot/Cursor (IDE 集成)
- **国内使用:** 通义千问/DeepSeek/CodeGeeX

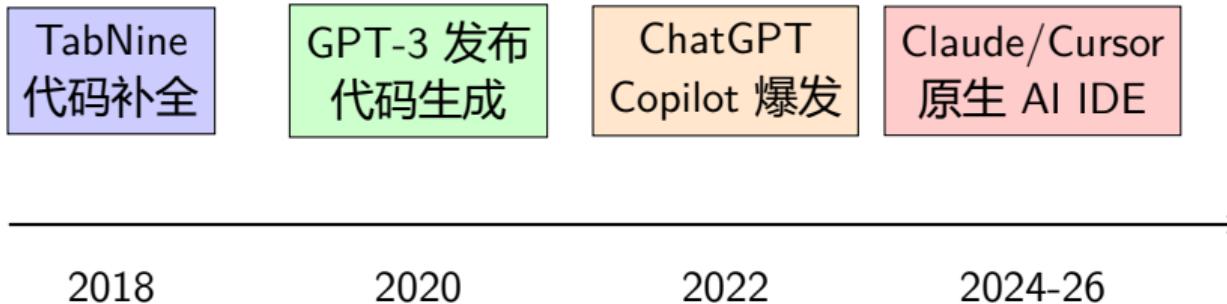


主流工具详细对比

工具	特点	适用场景	推荐指数
ChatGPT	对话能力强，代码生成准确	学习、调试、解释	[star][star][star][star][star]
Claude	代码分析深入	代码审查、架构设计	[star][star][star][star][star]
GitHub Copilot	IDE 集成，实时补全	日常编码、快速开发	[star][star][star][star][star-empty]
Cursor 通义千问	AI 原生 IDE 中文友好，国内可用	项目开发、重构 中文问题咨询	[star][star][star][star][star] [star][star][star][star][star-empty]
DeepSeek	编程能力强，开源	代码生成、算法实现	[star][star][star][star][star]



AI 编程工具发展史：从传统 IDE 到 AI 助手



里程碑事件

- 2018: TabNine 首次将深度学习用于代码补全
- 2020: GPT-3 展示强大的代码生成能力
- 2022: ChatGPT 和 GitHub Copilot 引爆 AI 编程浪潮
- 2024-26: Claude、Cursor 等原生 AI IDE 重新定义编程体验



Cursor = VS Code + AI 原生集成

<https://cursor.sh>

核心快捷键：

- Ctrl+K: 原地编辑代码
- Ctrl+L: 上下文对话
- @Files: 引用项目文件
- @Code: 引用代码片段
- @Docs: 引用文档

强大功能：

- **代码生成**: 根据注释直接生成代码
- **代码解释**: 选中代码，一键解释
- **代码重构**: 智能重命名、提取函数
- **错误修复**: 自动检测并修复 bug
- **项目理解**: 全局搜索、跨文件分析



Cursor 实战演示

场景 1：原地编辑 (Ctrl+K)

- ① 选中要修改的代码
- ② 按 Ctrl+K
- ③ 输入指令：“添加异常处理”
- ④ AI 原地修改代码

场景 2：上下文对话 (Ctrl+L)

- ① 按 Ctrl+L 打开聊天面板
- ② 输入：“解释这段代码的作用”
- ③ AI 结合上下文给出解释

实战技巧

- 多轮对话细化需求
- 结合 QFiles 提供上下文



本地大模型与隐私保护

真实场景

企业/学校要求：代码不能上传到外部服务器！

解决方案：本地部署

1. Ollama (推荐)

- 一行命令部署本地模型
- 支持 Llama、DeepSeek、Qwen 等
- 完全离线运行

2. LM Studio

- 图形界面，易于使用
- 支持多种模型格式
- 内置聊天界面

3. vLLM

- 高性能推理引擎
- 适合企业级部署
- 支持多卡并行



不同模型的对比测试

测试任务：用 Python 和 OpenCV 实现答题卡边界检测

模型	代码质量	中文理解	运行成功率
GPT-4o	[star][star][star][star][star][star]	[star][star][star][star][star][star] empty]	95%
Claude 3.5	[star][star][star][star][star][star]	[star][star][star][star][star][star] empty]	93%
DeepSeek-Coder	[star][star][star][star][star][star]	[star][star][star][star][star][star] empty]	90%
通义千问 2.5	[star][star][star][star][star][star]	[star][star][star][star][star][star] empty]	88%
GPT-3.5	[star][star][star][star]- empty][star- empty]	[star][star][star][star]- empty][star- empty]	75%



ChatGPT 使用教程 (一): 账号与界面

1. 账号注册与登录

- 访问 <https://chat.openai.com>
- 使用邮箱/Google/Microsoft 账号注册
- 免费版使用 GPT-4o mini, Plus 版使用 GPT-4o

2. 界面介绍

左侧边栏

- 新建对话
- 历史记录
- GPTs 商店

输入区域

- 文字输入框
- 文件上传
- 语音输入



ChatGPT 使用教程 (二): 编程技巧

核心功能:

- ① **代码生成**: 描述功能需求, ChatGPT 生成代码
- ② **代码解释**: 粘贴代码, 逐行解释逻辑
- ③ **错误调试**: 粘贴错误信息, 分析原因
- ④ **代码优化**: 要求优化性能/可读性

高级功能 (Plus 用户):

- **GPTs**: 创建专用助手 (如"OpenCV 专家")
- **文件分析**: 上传 PDF/代码文件分析
- **联网搜索**: 获取最新文档
- **数据分析**: 上传 Excel/CSV, 生成分析代码



ChatGPT 使用教程 (三): 实战示例

示例: 用 ChatGPT 实现人脸检测

Prompt 模板:

推荐 Prompt

你是一位有 5 年 OpenCV 经验的工程师。请用 Python 和 OpenCV 实现人脸检测：

功能: 从图片中检测所有人脸，并用绿色矩形框标注

输入: 图片文件路径 (字符串)

输出: 显示标注后的图像，打印检测到的人脸数量

要求:

- 使用 Haar 级联分类器
- 处理中文路径问题
- 代码有详细中文注释



Claude 使用教程（一）：核心优势

Claude vs ChatGPT：代码分析更深入

Claude 独特优势：

- **长文本处理**: 支持 200K token
- **代码审查**: 分析完整项目
- **安全意识**: 避免不安全代码
- **诚实性**: 不知道时会说明

适用场景：

- 架构设计咨询
- 复杂代码重构
- 代码审查与优化
- 长文档分析

访问方式

- 网页版: <https://claude.ai>
- Claude Code: 命令行工具
- API: 开发者集成



Claude 使用教程 (二): Artifacts 功能

Artifacts: Claude 的杀手级功能

什么是 Artifacts?

- 侧边栏预览生成内容
- 支持 HTML/React/Mermaid
- 实时编辑与交互

实战应用:

- 生成可视化图表
- 创建交互式文档
- 快速原型开发

示例

让 Claude 生成:

- 数据处理流程图
- 算法架构图
- API 文档页面



Claude 使用教程 (三): 代码审查实战

用 Claude 进行代码审查

步骤 1: 提供上下文

输入

”这是我的人脸检测代码，请审查其质量和安全性”

步骤 2: Claude 分析

- 指出潜在的安全漏洞
- 建议性能优化方案
- 检查边界情况处理
- 评估代码可读性

步骤 3: 迭代改进

输出

北京石油化工学院



通义千问使用教程（一）：国内优势

为什么选择通义千问？

核心优势

- **中文友好**: 理解中文术语
- **国内可用**: 无需特殊网络
- **免费额度**: 每日百万 token
- **多模态**: 支持图像理解

适用场景

- 中文技术问题咨询
- 代码中文注释生成
- 中文文档撰写
- 图像识别与分析

访问方式：

- 网页版: <https://tongyi.aliyun.com>
- 通义灵码: IDE 插件 (类似 Copilot)
- API: Python SDK 集成

北京石油化工学院



通义千问使用教程 (二): 灵码插件

通义灵码: 阿里推出的智能编码助手

安装与配置:

- ① VS Code / JetBrains 搜索 "Tongyi Lingma"
- ② 使用阿里云账号登录
- ③ 开始使用自动补全

核心功能:

- **智能补全:** 根据上下文预测代码
- **注释生成:** 一键添加中文注释
- **代码解释:** 选中代码查看解释
- **单元测试:** 自动生成测试用例

免费额度

个人用户每月 20 万次补全, 完全免费!

通义千问使用教程 (三): Python SDK

在代码中直接调用通义千问

安装 SDK:

```
^^! pip install dashscope  
^^!
```

代码示例:

```
1 import dashscope  
2 from dashscope import Generation  
3  
4 # 设置 API Key  
5 dashscope.api_key = "your-api-key"  
6  
7 # 调用模型  
8 response = Generation.call(  
9     model='qwen-turbo',  
10    prompt='用 Python 和 OpenCV 实现图像灰度化',  
11    max_tokens=1500  
12 )  
13  
14 print(response.output.text)  
15
```



结构化 Prompt 框架：RTF 模式



RTF 模板示例

Role (角色): 你是一个资深计算机视觉算法专家，精通 OpenCV 和图像处理。

Task (任务): 实现试卷图像的二值化，要求能自适应处理光照不均的情况。

Format (格式):

- 返回带有详细 Docstring 的 Python 函数
- 包含输入输出示例
- 列出关键参数的调优建议



RTF 框架实战对比

[] *Prompt*

帮我写个代码处理图像。

问题分析：

- 没有定义角色
- 任务模糊
- 没有格式要求

[] 好的 Prompt (RTF)

Role: 你是一位有 10 年经验的计算机视觉工程师，精通 OpenCV。

Task: 请编写一个 Python 函数，实现答题卡图像的自适应二值化。

Format:

- 提供完整的、可运行的代码
- 包含详细的函数文档
- 解释关键参数的设置依据



进阶技巧：Chain-of-Thought (思维链)

什么是思维链？

让 AI “一步一步思考” (Step by Step)，而不是直接给出答案。

[×]

Prompt：用 OpenCV 实现透视变换矫正倾斜的试卷。

问题：

- AI 直接扔出代码
- 学生不理解原理

[✓] 思维链引导

Prompt：请一步一步教我如何用 OpenCV 实现透视变换：

- ① 首先解释什么是透视变换
- ② 然后说明需要哪些输入参数
- ③ 接着给出实现步骤
- ④ 最后提供完整代码



不同类型 Prompt 对比：代码生成

功能描述：

- 清晰说明要实现什么功能
- 描述输入和输出
- 指定技术栈

代码要求：

- 代码风格规范
- 错误处理机制
- 性能要求

代码生成 Prompt 示例

角色：你是一位 Python 开发者。

任务：请帮我写一个读取 CSV 文件的 Python 函数。

输入/输出：

- 输入：CSV 文件路径
- 输出：pandas DataFrame

要求：

- 使用 pandas 库
- 处理文件不存在的情况
- 返回 DataFrame 对象



不同类型 Prompt 对比：代码解释与调试

代码解释 Prompt：

- 粘贴代码
- 逐行解释要求
- 关键参数说明
- 可能的错误用法

调试求助 Prompt：

- 粘贴代码
- 错误信息
- 预期行为
- 上下文说明

代码解释 Prompt 示例

请解释以下代码的作用：

```
1 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
2 blur = cv2.GaussianBlur(gray, (5, 5), 0)
3 edges = cv2.Canny(blur, 50, 150)
4 ^~I~~I~~I
```

要求：

- 逐行解释每行代码的作用
- 说明关键参数的含义
- 解释为什么要按这个顺序处理



Prompt 优化技巧：少样本提示 (Few-Shot)

什么是少样本提示？

提供 1-3 个示例，让 AI 通过模仿学习来完成任务。

[×] Zero – Shot

Prompt: 请将以下中文翻译成英文。

输入：计算机视觉

问题：

- AI 可能输出多种翻译



[√] 少样本 (Few-Shot)

Prompt: 请将以下中文翻译成英文。参考示例：

示例 1：

- 输入：人工智能
- 输出：Artificial Intelligence

示例 2：

- 输入：深度学习

Prompt 优化技巧：思维链 (Chain-of-Thought)

[×]

Prompt: 请帮我实现试卷透视变换矫正。

问题:

- 可能缺少关键步骤
- 换个场景就不会了

[✓] 思维链引导

Prompt:

请帮我实现试卷透视变换矫正。请按以下步骤思考：

Step 1: 透视变换的数学原理是什么？需要哪些参数？

Step 2: 如何从图像中自动找到试卷的四个角点？

Step 3: 请写出完整的 Python 实现代码



思维链实战：准确率对比

测试任务：用 OpenCV 实现自适应阈值处理

Prompt 类型	直接提问	思维链引导
代码完整度	70%	95%
参数解释清晰度	一般	详细
边界情况处理	很少提及	全面考虑
实际运行成功率	60%	90%

思维链 Prompt 模板

请帮我解决 [问题]。请按以下步骤思考：

Step 1：分析问题的核心要点是什么？

Step 2：有哪些可能的解决方案？各自的优缺点？

Step 3：请给出推荐的实现代码



进阶技巧：Few-shot (少样本提示)

什么是 Few-shot?

给 AI 几个 **正确的例子**，让它模仿你的风格或格式处理新任务。

示例场景： OpenCV 图像转换

Few-shot Prompt

请模仿以下示例的代码风格和注释规范：

示例 1 - 灰度化：

```
# 读取图像并转换为灰度图  
# 参数: image_path: 图像文件路径  
# 返回: gray_image: 灰度图像数组  
def load_and_gray(image_path):
```

待处理任务：

继续 Prompt

示例 2 - 高斯模糊：

```
# 对图像进行高斯模糊  
# 参数: image: 输入图像  
# 返回: blurred: 模糊后的图像  
def gaussian_smooth(image):  
    blurred =  
        cv2.GaussianBlur(image, (5,5), 0)
```



Few-shot 效果对比

Prompt 类型	无 Few-shot	有 Few-shot
代码风格一致性	随机, 不稳定	与示例高度一致
注释完整度	较简略	详细, 符合示例规范
参数说明	缺失或不清晰	完整的 Docstring
错误处理	经常遗漏	按示例模式添加
代码可读性	一般	优秀

Few-shot 使用技巧

- ① **示例数量:** 2-3 个示例通常足够
- ② **示例质量:** 确保示例是正确的、高质量的
- ③ **格式一致:** 示例之间保持风格一致
- ④ **明确指令:** 告诉 AI “请按照示例的风格”

上下文窗口管理

问题：为什么代码太长了 AI 就会“忘掉”前面的内容？

- 大模型有 **上下文窗口限制** (通常 4K-128K tokens)
- 超出限制后，模型会“遗忘”最早的内容
- 导致前后文不一致、回答质量下降

各模型上下文限制：

模型	上下文
GPT-4	8K/32K
GPT-4o	128K
Claude 3.5	200K
DeepSeek	64K
通义千问	128K

精简 Prompt 的技巧：

- ① **只提供必要代码**: 不要贴整个文件
- ② **使用摘要**: “前面我们讨论了 X, 现在要解决 Y”
- ③ **分段处理**: 长任务拆分成多个短任务
- ④ **定期总结**: 让 AI 总结当前进度



任务背景：试卷图像的噪声处理

故事问题

手机拍摄的试卷照片总是有噪点、光照不均，如何自动处理这些问题？

常见图像质量问题：

- **噪声**: 手机传感器产生的随机噪点
- **光照不均**: 拍摄环境光线不均匀
- **模糊**: 手抖或对焦不准
- **透视变形**: 拍摄角度倾斜
- **阴影**: 遮挡造成的暗区

北京石油化工学院



对阅卷系统的影响：

- 二值化效果差 → 填涂识别错误
- 边缘检测不准 → 答题卡定位失败
- 文字识别率低 → 主观题评分困难

本周目标

用 AI 辅助实现：

- ① 图像去噪 (高斯/中值滤波)
- ② 透视矫正 (透视变换)

实战 1：引导 AI 写出滤波代码

任务：对比高斯滤波、中值滤波对试卷扫描件的效果

第一轮 Prompt：

基础 Prompt

Role: 你是一位图像处理专家

Task: 请用 Python 和 OpenCV 实现图像去噪

要求:

- 分别实现高斯滤波和中值滤波
- 对比两种方法的效果
- 适用于试卷扫描件的去噪

第二轮 Prompt (优化)：

优化 Prompt

上面的代码很好，但还需要：

- ① 添加详细的函数文档 (Docstring)
- ② 解释两种滤波器的适用场景
- ③ 添加可视化对比图
- ④ 处理可能的错误 (如文件不存在)



实战 2：引导 AI 解决透视变换

挑战

透视变换 (Warp Perspective) 的数学原理复杂，涉及齐次坐标、变换矩阵等概念。
如何通过 AI 快速实现，而不需要深入数学细节？

第一轮：概念解释

概念 Prompt

请解释 OpenCV 中的透视变换原理：

- ① 什么是透视变换？
- ② getPerspectiveTransform 做了什么？
- ③ warpPerspective 做了什么？

第二轮：代码实现

实现 Prompt

请根据上面的解释，实现一个函数：

- 输入：任意角度拍摄的试卷图像
- 输出：矫正为正面视角的图像
- 自动或手动确定试卷的四个角点
- 使用 OpenCV 实现透视矫正



实战 3：自动识别答题卡边缘（迭代优化）

目标：通过 3 轮迭代 Prompt，让 AI 从写出简单 Canny 算子到写出鲁棒的轮廓提取逻辑。

第一轮：基础实现

基础 Prompt

请用 OpenCV 实现答题卡边缘检测。

AI 输出：

- 简单的 Canny 边缘检测

北京石油化工学院

王文通

第二轮：优化改进

优化 Prompt

上面的代码检测效果不好，请改进：

- ① 先进行图像预处理（去噪）
- ② 使用自适应阈值
- ③ 筛选出最大的四边形
- ④ 处理检测失败的情况

第三轮：鲁棒完善

完善 Prompt

请进一步优化，使其更鲁棒：

- ① 处理极端拍摄角度
- ② 处理光照不均和阴影
- ③ 添加透视变换矫正
- ④ 添加可视化调试信息
- ⑤ 封装成可复用的类

实战任务：用 AI 辅助实现人脸检测

Prompt 示例：

人脸检测 Prompt

请用 Python 和 OpenCV 实现一个人脸检测程序：

功能：从图片中检测所有人脸，并用矩形框标注

输入：图片文件路径

输出：标注了人脸框的图片

要求：

- 使用 OpenCV 的 Haar 级联分类器
- 在每个人脸周围绘制绿色矩形框
- 显示检测到的人脸数量
- 代码有详细中文注释

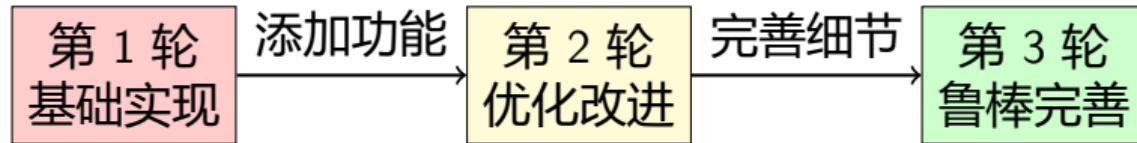


AI 生成的人脸检测代码示例

```
1 import cv2
2
3 def detect_faces(image_path):
4     """
5         检测图像中的人脸并标注
6
7     参数:
8         image_path: 图像文件路径
9     返回:
10        img: 标注了人脸框的图像
11    """
12    # 加载预训练的 Haar 级联分类器
13    face_cascade = cv2.CascadeClassifier(
14        cv2.data.haarcascades +
15        'haarcascade_frontalface_default.xml'
16    )
17
18    # 读取图像
19    img = cv2.imread(image_path)
20    if img is None:
21        raise ValueError(f"无法读取图像: {image_path}")
```



迭代优化演示：从简单到完善



第1轮：基础

- 简单 Canny
- 查找轮廓
- 基本功能

问题：

- 噪点干扰
- 参数固定
- 不完整边缘

第2轮：优化

- 高斯模糊
- 自适应阈值
- 面积筛选

改进：

- 抗噪性增强
- 自动调参
- 鲁棒性提升

第3轮：完善

- 完整类封装
- CLAHE 预处理
- 透视变换

特性：

- 可复用性强
- 调试可视化
- 生产级可用



AI 辅助调试 (Debug) 三部曲

第 1 步

提供 Traceback

第 2 步

说明环境

第 3 步

贴出数据

第 1 步：提供 Traceback

完整复制报错信息：

- 错误类型
- 错误位置（行号）
- 完整的堆栈跟踪

示例：

cv2.error: ...

Invalid number of
channels

第 2 步：说明环境

提供运行环境信息：

- 操作系统
- Python 版本
- OpenCV 版本
- 其他相关库版本

示例：

Windows 11

Python 3.9.7

OpenCV 4.8.0

第 3 步：贴出输入数据

提供输入数据信息：

- 数据类型和格式
- 数据的维度/大小
- 期望的输出格式
- 实际的输出（如果有错误）

示例：

图像尺寸：1920x1080

通道：3 (RGB)



调试 Prompt 模板

完整的调试 Prompt 示例

我的 OpenCV 代码报错了，请帮我分析：

错误信息：

cv2.error: OpenCV(4.8.0) ...

Invalid number of channels in input image

运行环境：

Windows 11, Python 3.9.7, OpenCV 4.8.0

相关代码：

```
img = cv2.imread('test.jpg')
```

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

输入数据：

图像文件 test.jpg, 尺寸 1920x1080

重构：把 AI 当成导师

重构的意义

代码能运行只是第一步，优雅的代码才能长期维护。

重构 Prompt 模板：

重构请求

请帮我重构这段代码，提高代码质量：

[粘贴代码]

要求：

- ① 提高运行效率
- ② 添加类型提示
- ③ 改进命名规范

北京石油化工学院



AI 可能的重构建议：

① 性能优化：

- 向量化运算替代循环
- 避免重复计算
- 使用生成器替代列表

② 可读性提升：

- 变量名更有意义
- 函数拆分更合理
- 添加类型提示

安全与伦理：红线警告

[!] 重要提醒

使用 AI 辅助编程时，必须注意以下安全与伦理问题！

1. 数据安全

- [×] 不要上传：

- API Key、密码
- 个人隐私数据
- 商业机密代码
- 学生个人信息

- [√] 安全做法：

- 使用脱敏示例数据



王文通（

2. 版权与责任

- 代码归属：

- AI 生成代码的版权复杂
- 检查所用工具的许可协议
- 商业项目需谨慎

- 责任界定：

- AI 生成的代码可能有错误
- 最终责任人是你自己
- 务必测试验证

3. 学术诚信



优秀 Prompt 案例分析：图像处理 Prompt 优化

[] Prompt

帮我写个代码处理图像。

问题分析：

- 没有定义角色
- 任务模糊
- 没有格式要求
- 没有约束条件

[] 优化后的 Prompt (具体)

角色：你是一位有 10 年经验的计算机视觉工程师，精通 OpenCV。

任务：请编写一个 Python 函数，实现答题卡图像的自适应二值化。

格式：

- 提供完整的、可运行的代码
- 包含详细的函数文档



优秀 Prompt 案例分析：算法实现 Prompt 优化

[×] *Prompt*

实现一个排序算法。

问题分析：

- 没有指定算法类型
- 没有数据规模要求
- 没有性能约束
- 缺少示例

[√] 优化后的 Prompt (有示例)

角色：你是一位算法工程师。

任务：实现一个快速排序算法。

约束：

- 时间复杂度 $O(n \log n)$
- 空间复杂度 $O(\log n)$

示例：

- 输入: [3, 6, 8, 10, 1, 2, 1]
- 输出: [1, 1, 2, 3, 6, 8, 10]



常见错误分析与解决：Prompt 太模糊

[×] *Prompt*

帮我写代码。

分析原因：

- 缺少角色设定
- 缺少任务描述
- 缺少约束条件
- AI 无法确定具体需求

[√] **解决方案**

添加角色：你是一位 Python 开发者。

明确任务：请帮我写一个读取 CSV 文件的 Python 函数。



常见错误分析与解决：Prompt 太复杂

[×] Prompt

你是一位有 5 年 OpenCV 经验、精通 Python 和 C++、熟悉深度学习和机器学习的工程师，请用 Python 和 OpenCV 实现一个完整的智能阅卷系统，包括图像采集、预处理、版面分析、OCR 识别、评分输出等所有模块，要求代码有详细注释、性能优化、异常处理、单元测试、文档生成...

分析原因：

- Prompt 过长
- 信息过载
- AI 容易混淆
- 一次性要求太多

[√] 解决方案

北京石油化工学院

王文通



AI 辅助编程工具实战

2025-2026 学年



55 / 61

实时互动环节

互动方式建议

- **问卷星实时投票**: 使用手机扫码参与, 实时显示统计结果
- **代码拼图游戏**: 将打乱的代码片段重新排序, 小组竞赛
- **错误找茬挑战**: 展示有问题的代码, 集体找出 bug

问题 1: 好的 Prompt 应该包含哪些要素?

- A 角色设定
- B 任务描述
- C 约束条件
- D 以上都是 [√]

问题 2: 以下哪种 Prompt 更适合调试?

王文通 (AI 辅助编程工具实战)

问题 3: AI 生成的代码一定正确吗?

- A 一定正确
- B 不一定正确, 需要验证和测试 [√]

问题 4: 以下哪种情况说明 Prompt 太模糊?

- A AI 生成了完全符合需求的代码



2025-2026 学年

56 / 61

代码找错挑战

找出以下 Prompt 中的 3 个问题

1 帮我写个代码。
2
3 实现一个系统。
4
5 要完整的。
6
7 快点。
8 ^^I^^I

答案揭晓

① 问题 1：缺少角色设定

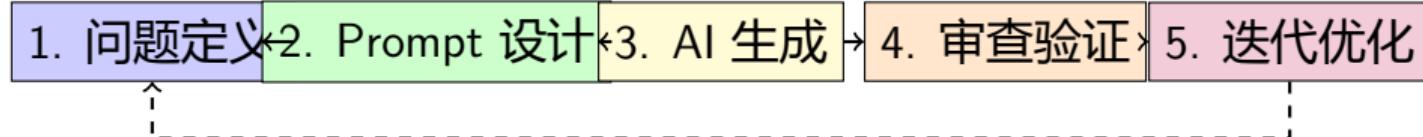


建立自己的"AI 协作 SOP"

什么是 SOP?

Standard Operating Procedure (标准操作流程)

一套可重复、可优化的工作流程



1. 问题定义

- 明确要解决的问题
- 确定输入输出

3. AI 生成

- 运行 Prompt
- 获取初始输出
- 记录生成时间

4. 审查验证



知识点总结：Prompt 工程核心要点

1. Prompt 工程三要素

① Role (角色)

- 明确 AI 的身份
- 例如：资深 CV 工程师

② Task (任务)

- 清晰说明要做什么
- 具体、明确、可执行

③ Format (格式)

- 指定输出格式和要求
- 代码、文档、表格等

2. Prompt 优化技巧

① 少样本提示 (Few-Shot)

- 提供 1-3 个示例
- 模仿学习

② 思维链 (Chain-of-Thought)

- 分步思考
- 推理过程

③ 角色扮演 (Role-Playing)

- 设定专业身份
- 专业术语



知识点总结：AI 工具使用规范

[√] 允许

- 用 AI 解释概念
- 调试错误
- 优化代码
- 学习新技术

[+] 鼓励

- 生成对比实验
- 可视化结果
- 探索不同方案
- 提高学习效率

[\times] 禁止

- 直接复制完整代码
- 用 AI 完成全部作业
- 不思考直接照搬
- 欺骗性使用

核心理念：理解原理 > 复制代码

AI 是助手，不是替代者



延伸学习资源

官方文档

① ChatGPT 官方文档

- <https://platform.openai.com/docs>
- API 使用指南
- 最佳实践

② Claude 官方文档

- <https://docs.anthropic.com>
- Claude API 文档
- 提示工程指南

③ 通义千问官方文档

推荐教程与资源

④ Prompt 工程教程

- 《Prompt Engineering Guide》
- <https://www.promptingguide.ai>
- 中文版：提示工程指南

⑤ GitHub Copilot 使用指南

- <https://github.com/features/copilot>
- 官方文档和教程
- 最佳实践分享

⑥ Cursor 官方教程

- <https://cursor.sh/tutorials>
- 视频教程



本周作业：答题卡边界检测

作业题目

用 AI 辅助实现 答题卡边界检测程序

项目关联

这是 AI 阅卷助手的第一步：

- ① 图像采集与预处理
 - ② 答题卡定位 (当前任务)
 - ③ 填涂检测与识别
 - ④ 手写文字 OCR
 - ⑤ 成绩统计与输出

分层任务说明

基础版 检测边界 + 透视

AI 对话要求

至少 3 轮交互：

- ① 第1轮：基础实现
 - ② 第2轮：优化改进
 - ③ 第3轮：完善功能

不同参与模式的建议

- 观察者：完成基础版，重点理解 Prompt 设计



作业提交内容与评分标准

提交内容

① AI 对话记录

- 截图或文本
- 标注关键 Prompt
- 说明迭代过程

② 最终代码

- 完整可运行
- 必要注释

③ 测试结果

- 测试图片
- 处理结果

④ 反思报告

- Prompt 设计思路



详细评分标准 (100 分)

分项	评分要点
Prompt 设计 (30 分)	结构清晰、迭代合理 RTF 框架应用恰当
代码完整性 (30 分)	功能完整、可运行 注释清晰、结构合理
测试覆盖 (20 分)	多场景测试 边界情况考虑
反思深度 (20 分)	思考深入 有独特见解



分层任务详解

基础版

要求:

- 读取答题卡图像
- Canny 边缘检测
- 查找轮廓
- 绘制边界框

适合:

- 观察者模式
- 编程基础较弱

目标: 理解基本流程

进阶版

要求:

- 完成基础版所有功能
- 添加预处理（去噪）
- 自适应阈值
- 透视变换矫正

适合:

- 使用者模式
- 有一定编程基础

目标: 独立完成实用功能

挑战版

要求:

- 完成进阶版所有功能
- 封装成类
- 处理极端情况
- 可视化调试信息

适合:

- 创造者模式
- 编程基础较好

目标: 生产级可用代码



谢谢！

第 3 周预告：图像预处理与增强

故事问题：试卷拍照模糊怎么办？

- 图像去噪（高斯/中值滤波）
- 图像二值化（全局/Otsu/自适应）
- 透视矫正（透视变换）

