

第 10 周：核心开发与调试

让系统真正跑起来

计算机视觉课程组

通选课

课程概览

主程序结构

```
class AutoGradingSystem:
    """自动阅卷系统"""

    def __init__(self):
        # 初始化各模块
        self.preprocessor = Preprocessor()
        self.layout_analyzer = LayoutAnalyzer()
        self.choice_recognizer = ChoiceRecognizer()
        self.judge_recognizer = JudgeRecognizer()
        self.essay_recognizer = EssayRecognizer()
        self.grading_module = GradingModule()

    def process(self, image_path):
        """处理试卷"""
        # 1. 预处理
        preprocessed = self.preprocessor.process(image)

        # 2. 版面分析
        layout = self.layout_analyzer.analyze(image)

        # 3. 识别
        choices = self.choice_recognizer.recognize_all(
```

集成步骤

① 搭建框架

- 创建项目结构
- 配置开发环境
- 编写基础代码

② 模块开发

- 从选择题开始
- 编写单元测试
- 调试优化

③ 模块集成

- 整合各模块
- 测试整体流程
- 修复接口问题

④ 测试优化

- 使用测试集验证
- 优化识别准确率

测试集准备

测试数据：

- 基础选择题：单独测试选择题识别
- 基础判断题：单独测试判断题识别
- 完整试卷：测试整体流程

测试目标：

- 选择题准确率 $>80\%$
- 判断题准确率 $>80\%$
- 简答题能提取文字

单元测试示例

```
class TestChoiceRecognizer(unittest.TestCase): def
    test_calculate_density(self): """ """test_img = create_test_image(density = 0.5)density =
    self.recognizer.calculate_density(test_img)self.assertAlmostEqual(density, 0.5, places = 1)
def test_recognize_single(self): """ """result =
    self.recognizer.recognize_question(img, positions)self.assertIn(result, [
    'A', 'B', 'C', 'D', None])
if __name__ == '__main__': unittest.main()
```

课堂思考题

问题 1：集成调试

- 模块集成时常见问题有哪些？
- 如何快速定位错误来源？

问题 2：性能优化

- 如何提升识别准确率？
- 如何优化代码执行速度？

开发检查点

检查点 1：选择题模块完成

- 能检测填涂
- 能识别 A/B/C/D
- 输出格式正确

检查点 2：判断题模块完成

- 能检测 ✓/×
- 能区分正误
- 输出格式正确

检查点 3：模块集成完成

- 主程序能调用各模块
- 数据能正确流转
- 无致命错误

常见问题与解决

问题	解决方案
填涂识别不准	调整阈值, 检查二值化
符号识别失败	检查轮廓, 用模板匹配后备
OCR 结果乱码	检查预处理, 调整分辨率
模块导入错误	检查 <code>sys.path</code> , 确保 <code>__init__.py</code>
数组越界	使用 <code>safe_crop</code> , 检查边界

本周任务

核心任务

- ① 完成所有模块开发
- ② 完成模块集成
- ③ 通过基本功能测试
- ④ 准备演示材料

时间紧迫，务必确保系统基本可用！

第 11 周：成果展示与总结

每组 5 分钟演示 + 2 分钟答辩

准备好你们的展示！

加油！

最后一周冲刺