PaddleOCR 检测+识别（det+rec）模型训练方案

# 1. 项目目标

- 使用在留卡图像进行字段级文本检测与识别训练  
- 目标部署环境：轻量设备 / 中低端 GPU / 本地服务器  
- 教师模型来源：DeepSeek-VL2 生成伪标注

# 2. 技术选型对比

| 项目 | PaddleOCR（det+rec） | DeepSeek-VL（大模型） |  
|------|----------------------|-------------------------|  
| 功能 | 文本框检测 + OCR识别 | 多模态推理（字段提取） |  
| 精度 | 中-高（依赖数据质量） | 高（zero-shot能力强） |  
| 推理速度 | 快（可部署边缘设备） | 慢（需大GPU，响应慢） |  
| 训练资源 | 中（消费级GPU可用） | 非常高（需A100/H100） |  
| 模型大小 | 5MB～100MB 可选 | 数十GB |  
| 开源许可 | Apache 2.0 | DeepSeek社群/非商用可用 |

# 3. 数据格式（det+rec）

每张图像配备标注格式如下：  
  
images/img\_001.jpg [  
 {"transcription": "氏名", "points": [[30, 40], [130, 40], [130, 65], [30, 65]]},  
 {"transcription": "ZHA JUNJIE", "points": [[140, 40], [300, 40], [300, 65], [140, 65]]}  
]

# 4. 性能表现参考

| 模型名 | 任务 | 精度 | 推理时间 | 备注 |  
|--------|------|------|----------|------|  
| PP-OCRv3-Mobile | det+rec | F-score > 85% | ~70ms | 轻量部署 |  
| PP-OCRv3-Server | det+rec | F-score > 90% | ~30ms | 高精度服务器端 |

# 5. 训练计划（Training Plan）

\*\*阶段1：数据准备\*\*  
- 使用 DeepSeek 输出生成字段识别结果  
- 自动转换为 label.txt 格式（包含文本与位置）  
  
\*\*阶段2：模型训练\*\*  
示例命令：  
python tools/train.py \  
 -c configs/ppocrv3/ch\_PP-OCRv3\_det\_rec.yml \  
 -o Global.save\_model\_dir=./output/zairyu\_ocr/ \  
 Train.dataset.data\_dir=./dataset/ \  
 Train.dataset.label\_file\_list=["./dataset/label.txt"]

# 6. 模型导出与部署

导出命令：  
python tools/export\_model.py \  
 -c configs/ppocrv3/ch\_PP-OCRv3\_det\_rec.yml \  
 -o Global.pretrained\_model=output/zairyu\_ocr/best\_accuracy \  
 Global.save\_inference\_dir=./inference/zairyu\_ocr/

# 7. 优势总结

| 指标 | 内容 |  
|------|------|  
| 成本 | 消费级 GPU 可训练 |  
| 效果 | 接近大型模型精度（经蒸馏） |  
| 速度 | 单图<80ms，适合实时系统 |  
| 部署 | 支持轻量级嵌入式平台 |  
| 兼容 | 可与DeepSeek联合训练 |

# 8. 后续建议

- 构建1000张真实在留卡伪标注数据集  
- 完成PaddleOCR第一轮训练与验证  
- 如需精度提升，进行半监督微调  
- 构建可上传图像 → 输出字段的演示系统