

第 7 周：OCR 基础与文字识别

怎么让机器“阅读”文字？

计算机视觉课程组

通选课

课程概览

什么是 OCR?

OCR: Optical Character Recognition

光学字符识别——将图像中的文字转换为计算机可编辑的文本

应用场景:

- 文档数字化
- 车牌识别
- 证件识别
- **试卷文字识别**

OCR 技术发展

| 阶段 | 技术 | 准确率 |
|-------------|-------------|--------|
| 传统方法 | 模板匹配、特征提取 | 60-70% |
| 机器学习 | SVM、HMM | 80-85% |
| 深度学习 | CNN、RNN | 90-92% |
| Transformer | TrOCR、Donut | 95%+ |

当前：端到端大模型时代

主流 OCR 工具对比

| 工具 | 优点 | 缺点 | 适用 |
|-----------|--------|---------|-------------|
| Tesseract | 开源免费 | 中文准确率一般 | 英文文档 |
| PaddleOCR | 中文准确率高 | 需要下载模型 | 中文场景 |
| EasyOCR | 多语言支持 | 速度较慢 | 多语言混合 |
| TrOCR | 准确率最高 | 资源占用大 | 高精度场景 |

本课程推荐： PaddleOCR（中文友好、易用）

PaddleOCR 安装

```
# 安装PaddlePaddle
pip install paddlepaddle

# 安装PaddleOCR
pip install paddleocr

# 安装其他依赖
pip install opencv-python pillow
```

注意

- 首次使用会自动下载模型
- 需要联网
- 首次运行较慢

基础使用

```
from paddleocr import PaddleOCR

# 初始化OCR
ocr = PaddleOCR(use_angle_cls=True, lang='ch')

# 识别图像
result = ocr.ocr('exam_paper.jpg', cls=True)

# 打印结果
for line in result:
    text = line[1][0]          # 文字内容
    confidence = line[1][1]    # 置信度
    print(f"{text} ({confidence:.4f})")
```