

第 8 周教案：手写简答题识别

计算机视觉课程组

1 基本信息

周次	第 8 周
主题	手写简答题识别
学时	3 学时（160 分钟）
故事问题	能看懂学生写的答案吗？
OBE 目标	A3-特征识别：能识别手写文字
项目贡献	实现简答题手写识别模块

2 教学目标

1. 知识目标：

- 理解手写识别的挑战
- 了解端到端手写识别模型
- 掌握 TrOCR/PaddleOCR 手写识别

2. 能力目标：

- 能够使用手写识别 API
- 能够处理手写识别结果
- 能够优化识别准确率

3. 素养目标：

- 理解手写识别在实际应用中的价值
- 了解 AI 技术的局限性

3 教学重点与难点

教学重点

- 手写识别的挑战与特点
- TrOCR/PaddleOCR 手写模型使用
- 识别结果后处理

教学难点

- 手写字迹形变处理
- 低质量图像识别

4 教学过程设计

4.1 环节一：手写识别概述（40 分钟）

4.1.1 1.1 手写识别的挑战（15 分钟）

为什么手写识别比印刷识别难？

因素	印刷文字	手写文字
字体规范	统一标准字体	每人不同
字形稳定	字形固定	形状变化大
间距规范	固定间距	随意
倾斜角度	水平排列	可能倾斜
连笔情况	无连笔	常见连笔

手写识别的难点：

1. 字形差异大：不同人的书写风格完全不同
2. 连笔问题：字与字可能连在一起
3. 倾斜旋转：书写方向可能不正
4. 涂改修改：可能有涂改痕迹
5. 图像质量：拍照可能有模糊、阴影

阶段	技术	准确率
传统方法	特征工程 + SVM/HMM	60-70%
CNN 时代	LeNet, AlexNet 等 CNN	80-85%
RNN+Attention	RNN+Attention 机制	90-92%
Transformer	TrOCR, Donut	95%+
当前	端到端大模型	95%+

4.1.2 1.2 手写识别技术发展（15 分钟）

4.1.3 1.3 主流手写识别工具（10 分钟）

工具	优点	缺点	适用
TrOCR	准确率最高	速度慢，需 GPU	高精度场景
PaddleOCR	速度快，中文好	手写略弱于 TrOCR	实时应用
Tesseract	免费开源	中文手写差	英文手写
EasyOCR	多语言	中文一般	多语言混合

4.2 环节二：PaddleOCR 手写识别（40 分钟）

4.2.1 2.1 手写模型配置（10 分钟）

```

from paddleocr import PaddleOCR

# PaddleOCR支持手写识别
# 使用中文模型时，默认已包含手写识别能力

# 初始化手写识别OCR
ocr_handwrite = PaddleOCR(
    use_angle_cls=True,      # 启用方向分类（手写文字可能有倾斜）
    lang='ch',               # 中文
    det_model_dir=None,      # 使用默认检测模型
    rec_model_dir=None,      # 使用默认识别模型
    use_gpu=False,          # 是否使用GPU
    show_log=False
)

# 识别手写文字
result = ocr_handwrite.ocr('handwriting.jpg', cls=True)

```

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

4.2.2 2.2 手写图像预处理（15 分钟）

手写图像的特殊处理：

```
import cv2
import numpy as np

def preprocess_handwriting(image):
    """
    手写图像预处理

    参数：
        image: 输入图像

    返回：
        预处理后的图像
    """
    # 转灰度
    if len(image.shape) == 3:
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    else:
        gray = image.copy()

    # 去噪（手写可能有不规则噪声）
    denoised = cv2.fastNlMeansDenoising(gray, None, 10, 7, 21)

    # 对比度增强（手写可能对比度不足）
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    enhanced = clahe.apply(denoised)

    # 二值化
    _, binary = cv2.threshold(
        enhanced, 0, 255,
        cv2.THRESH_BINARY + cv2.THRESH_OTSU
```

```

)

# 形态学操作（去除小噪点）
kernel = np.ones((2, 2), np.uint8)
cleaned = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel)

return cleaned

# 使用
img = cv2.imread('handwriting.jpg')
processed = preprocess_handwriting(img)
cv2.imwrite('handwriting_processed.jpg', processed)

# 使用处理后的图像进行识别
result = ocr_handwrite.ocr('handwriting_processed.jpg', cls=True)

```

4.2.3 2.3 手写数字识别（15 分钟）

简答题常涉及数字答案：

```

def recognize_handwritten_digits(image_path):
    """
    识别手写数字

    适用于填空题、简答题中的数字答案
    """
    ocr = PaddleOCR(use_angle_cls=True, lang='ch', show_log=False)

    # 预处理
    img = cv2.imread(image_path)
    processed = preprocess_handwriting(img)

    # 识别
    result = ocr.ocr(processed, cls=True)

    # 提取数字
    import re
    digits = []

    for line in result:

```