

# 第 3 周：图像预处理与增强

## 试卷拍照模糊怎么办？

计算机视觉课程组

通选课

# 课程概览

1 预处理概述

2 图像去噪

3 图像二值化

4 透视矫正

5 课后作业

# 为什么需要预处理?

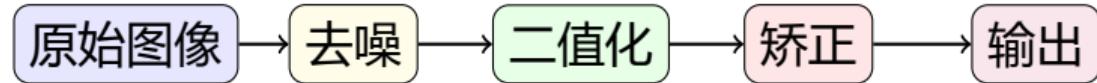
## 现实问题:

- 拍摄角度不正
- 光照不均匀
- 纸张有折痕
- 背景有杂物

## 预处理目标:

- ① 去除噪声干扰
- ② 增强目标特征
- ③ 规范图像格式
- ④ 提升识别准确率

# 预处理流水线



**关键：**预处理质量直接决定后续识别效果！

# 常见噪声类型

噪声类型	特征	典型场景
高斯噪声	随机分布的亮度变化	传感器噪声、低光拍摄
椒盐噪声	随机的黑点或白点	传输错误、老化传感器
周期噪声	规则的干扰条纹	电气干扰

## 试卷扫描常见问题

- 拍照时有噪点
- 扫描时有灰尘
- 压缩失真

# 滤波方法对比

## 1. 高斯滤波 - 适合高斯噪声

```
blur = cv2.GaussianBlur(img, (5, 5), 0)
```

## 2. 中值滤波 - 适合椒盐噪声，保边

```
median = cv2.medianBlur(img, 5)
```

## 3. 双边滤波 - 保边去噪

```
bilateral = cv2.bilateralFilter(img, 9, 75, 75)
```

建议：试卷预处理使用中值滤波

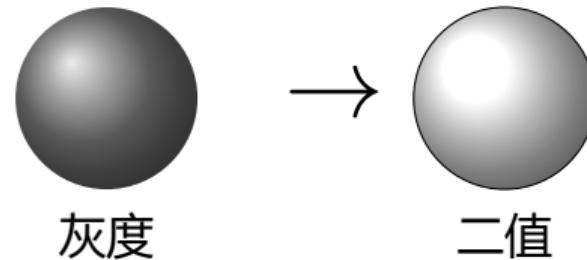
# 什么是二值化？

## 定义

将灰度图像转换为只有黑白两种颜色的图像

## 转换规则：

- 像素值  $>$  阈值  $\rightarrow$  白色 (255)
- 像素值  $\leq$  阈值  $\rightarrow$  黑色 (0)



# 二值化方法

## 1. 全局阈值

```
ret, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
```

## 2. Otsu 自动阈值

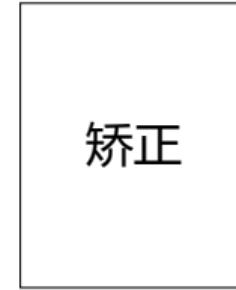
```
ret, otsu = cv2.threshold(gray, 0, 255,  
cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

## 3. 自适应阈值 - 推荐用于光照不均

```
adaptive = cv2.adaptiveThreshold(gray, 255,  
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,  
cv2.THRESH_BINARY, 11, 2)
```

# 透视矫正

问题：拍照时试卷角度不正



解决：透视变换（Perspective Transform）

# 透视变换实现

```
import numpy as np

# 四个角点坐标（实际需要检测）
pts = np.float32([[100, 150], [450, 120], [480, 380], [80, 400]])

# 目标矩形
width, height = 400, 300
dst = np.float32([[0, 0], [width-1, 0],
                  [width-1, height-1], [0, height-1]])

# 计算变换矩阵
M = cv2.getPerspectiveTransform(pts, dst)

# 应用变换
warped = cv2.warpPerspective(img, M, (width, height))
```

**关键：**需要先检测试卷的四个角点（下周内容）

## 问题 1：图像预处理

- 什么情况下使用高斯滤波？什么情况下使用中值滤波？
- 为什么二值化时要用反色 (THRESH\_BINARY\_INV)？

## 问题 2：透视变换

- 如何自动检测试卷的四个角点？
- 如果四个角点检测不准确，会影响什么？

# 课后作业

## 题目

实现试卷图像预处理完整流程

### 要求:

- ① 对试卷图像进行去噪处理
- ② 实现二值化（至少两种方法对比）
- ③ 提交处理前后对比图
- ④ 分析不同参数对结果的影响

### 评分标准:

- 代码实现: 40 分
- 效果对比: 30 分
- 参数分析: 20 分
- 代码规范: 10 分

## 第 4 周：试卷版面分析

故事问题：怎么知道选择题、简答题在哪里？

你将学会：

- 边缘检测与轮廓查找
- 区域定位与分割
- 版面结构分析

# 谢谢！

有问题随时交流