

22336216_陶宇卓

实验

要求

- 编写一个全局阈值化程序，该程序使用Otsu方法自动估计阈值，程序的输出是一个二值分割图像。
- 使用阈值程序对image.png进行分割。

原理

Otsu方法是一种自适应阈值分割方法，通过最大化类间方差来确定图像的最佳阈值。该方法假设图像由前景和背景组成，并且这两部分的灰度直方图呈双峰分布。Otsu方法通过遍历所有可能的阈值，找到使类间方差最大的那个阈值，从而实现图像的二值化。

具体实现

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
font = {'family': 'Microsoft YaHei',
        'weight': 'bold',
        'size': 'larger'}
(plt.rc("font", family='Microsoft YaHei', weight="bold"))

def manual_otsu_thresholding(image_path):
    # 读取图像
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # 计算直方图
    hist, bins = np.histogram(image.ravel(), 256, [0, 256])

    # 总像素数
    total_pixels = image.size

    # 初始化变量
    current_max, threshold = 0, 0
    sum_total, sum_foreground = 0, 0
    weight_background, weight_foreground = 0, 0

    for i in range(256):
        sum_total += i * hist[i]

    for i in range(256):
        weight_background += hist[i]
        if weight_background == 0:
            continue
        weight_foreground = total_pixels - weight_background
        if weight_foreground == 0:
            break

        sum_foreground += i * hist[i]

        mean_background = sum_foreground / weight_background
        mean_foreground = (sum_total - sum_foreground) / weight_foreground

        # 计算类间方差
        between_class_variance = weight_background * weight_foreground * (mean_background - mean_foreground) ** 2

        # 找到最大类间方差对应的阈值
        if between_class_variance > current_max:
            current_max = between_class_variance

```

```

        threshold = i

    # 应用阈值
    _, binary_image = cv2.threshold(image, threshold, 255, cv2.THRESH_BINARY)

    return binary_image

# 使用手动实现的Otsu方法对image.png进行分割
binary_image = manual_otsu_thresholding('image.png')

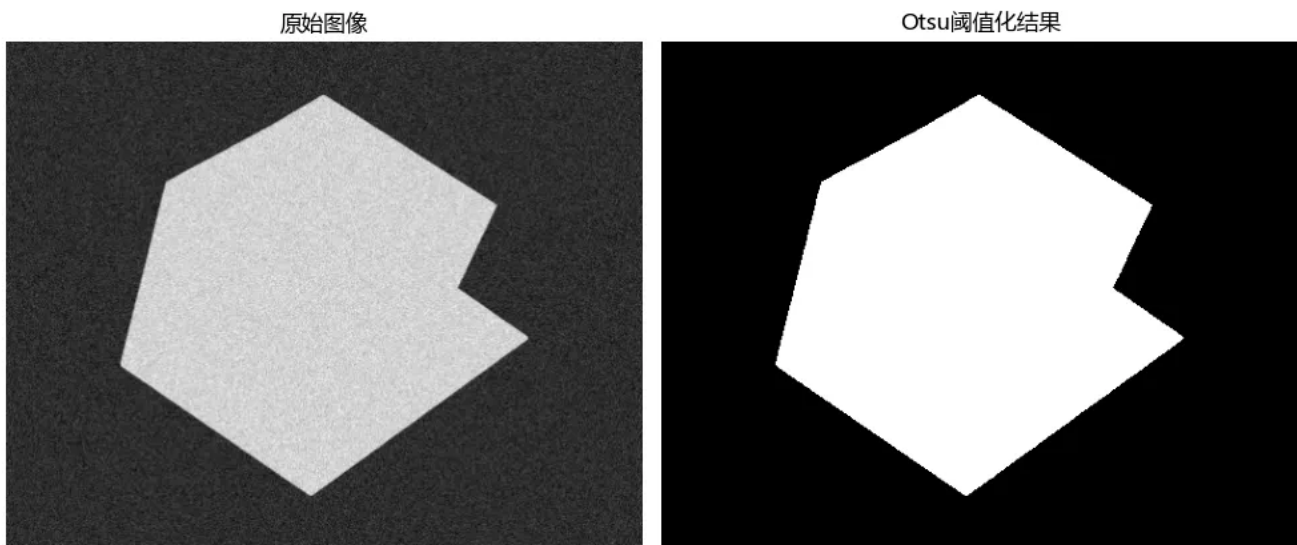
# 显示结果
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("原始图像")
plt.imshow(cv2.imread('image.png', cv2.IMREAD_GRAYSCALE), cmap='gray')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.title("Otsu阈值化结果")
plt.imshow(binary_image, cmap='gray')
plt.axis('off')

plt.tight_layout()
plt.show()

```

结果分析



实验结果图像对比

通过上述程序，我们成功地对image.png进行了Otsu阈值化处理。原始图像和二值化结果显示在同一窗口中。可以看到，Otsu方法能够有效地将图像分割为前景和背景，前景部分为白色，背景部分为黑色。

总结

本实验通过实现Otsu阈值化方法，成功地对图像进行了二值化处理。Otsu方法通过最大化类间方差，自动确定了最佳阈值，分割效果良好。该方法适用于灰度直方图呈双峰分布的图像，对于单峰或多峰分布的图像，可能需要其他分割方法。