

MMCS Project1——Image Segmentation

522031910213 朱涵

April 25, 2024

1 生成测试图形

如下图，使用提供的代码成功生成了测试图形并进行了可视化。

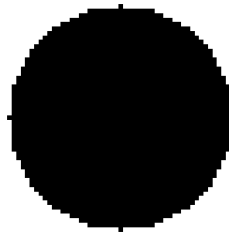


Figure 1: 测试图形（圆形）

2 优化问题的解决

代码见下图。先设置变量为一个布尔型的矩阵，然后定义目标函数，第一项是未分割出灰度较低部分的惩罚，第二项是未分割出灰度较高部分的惩罚，两者之和即目标函数。最后设置问题然后求解，返回最优解矩阵。由于约束条件（变量为布尔型）在变量定义时就已经进行限制，因此不需要显式地设置。

```
44 # 构建优化问题
45 def solve_optimization_problem(original_image):
46     # 定义分割变量
47     segmented_image = cp.Variable(original_image.shape, boolean=True)
48
49     # 定义目标函数
50     obj_1 = cp.sum(cp.square(cp.multiply(segmented_image, original_image)))
51     obj_2 = cp.sum(cp.square(cp.multiply((1 - segmented_image), (1 - original_image))))
52     objective = cp.Minimize(obj_1 + obj_2)
53
54     # 定义约束条件
55     constraints = []
56
57     # 求解优化问题
58     problem = cp.Problem(objective, constraints)
59     problem.solve(solver=cp.ECOS_BB)
60
61     return segmented_image.value
62
```

Figure 2: 优化部分代码

3 测试结果的可视化

代码见下图。使用matlab的plot库进行绘制，在这里我简单的直接根据灰度绘制了图象。

```
64 # 可视化分割结果
65 def visualize_segmentation(original_image, segmented_image):
66
67     # 背景图像
68     plt.subplot(1, 2, 1)
69     plt.imshow(original_image, cmap="gray")
70     plt.title("Original Image")
71     plt.axis("off")
72
73     # 分割结果
74     plt.subplot(1, 2, 2)
75     plt.imshow(segmented_image, cmap="gray")
76     plt.title("Segmentation Result")
77     plt.axis("off")
78
79     plt.show()
```

Figure 3: 可视化部分代码

可视化结果见下图。容易看到测试图形被很好的分割了出来。

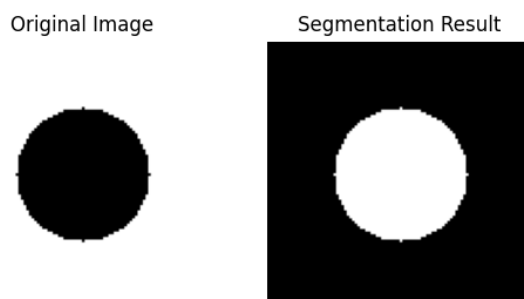
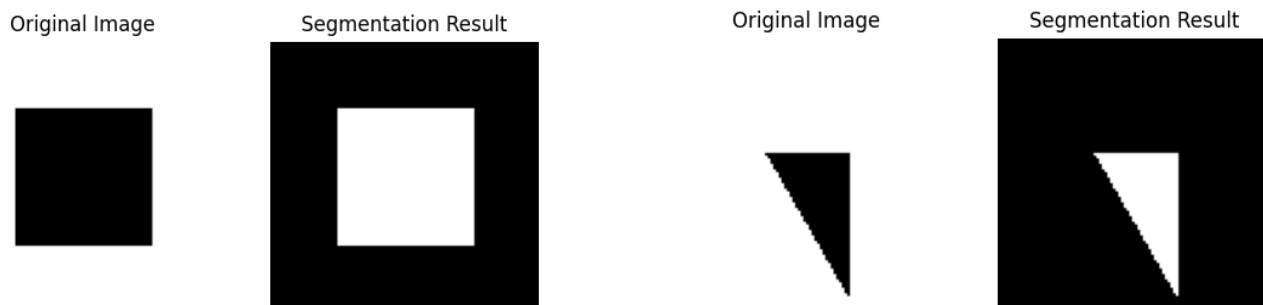


Figure 4: 可视化结果

4 实验内容的补充

由于篇幅有限，我选择了较为简单的一项补充：尝试对更多形状的图形进行测试分割。我简单的修改了生成测试图形的代码，使其可以生成正方形和三角形的图像。测试结果如下，可以看到分割结果都很不错，说明我们的优化函数是合理的。



(a) 补充：正方形图像分割结果

(b) 补充：三角形图像分割结果