

凸优化 project 报告

王熠笑 522031910732

2024 年 4 月 27 日

1 Objective Function and Constraints

1.1 Objective Function

在本次 project 中, objective function 为:

$$\begin{aligned} \text{Minimize } Z = & \sum_{i,j} (x_{i,j} \times \text{image}[i,j])^2 \\ & + \sum_{i,j} ((1 - x_{i,j}) \times (1 - \text{image}[i,j]))^2 \end{aligned}$$

其中, $x_{i,j}$ 表示在图像中每个像素 (i, j) 是否属于黑色圆圈的二进制变量, 值为 0 时表示属于圆圈, 值为 1 时表示属于背景。

1.2 constraints

二进制变量: 每个像素 (i, j) 的分割变量 $x_{i,j}$ 必须是二进制的, 即它可以取值 0 或 1。这反映了每个像素的分割决策是二元的。

分割选择的反映: 约束条件强制每个像素要么完全属于黑色圆圈, 要么完全属于白色背景, 没有中间状态。

2 logic of the code implementation

图像生成: 首先, 定义函数 `create_image` 来合成图像。该图像是一个 100x100 像素的数组, 其中心位置有一个半径为 30 像素的黑色圆圈。圆圈的像素值被设置为 0.01, 而背景的像素值初始化为 0.99。生成图像后, 我们使用 `matplotlib` 库将其以灰度图的形式可视化, 并保存到指定路径。

优化问题构建：接下来，我们构建了优化问题。定义了一个布尔变量 x 来表示每个像素是否属于圆圈，其中 x 的值为 1 表示像素属于圆圈，为 0 则表示属于背景。之后，我们定义目标函数，它由两部分组成：一部分惩罚错误地将圆圈像素分类为背景的像素；另一部分惩罚错误地将背景像素分类为圆圈的像素。目标函数的目的是最小化这两种错误的总和。

优化问题求解：使用 `cvxpy` 库中的 `ECOS_BB` 求解器来求解优化问题。求解完成后，我们根据 `x.value` 反转像素的分类结果，将布尔变量 x 转换为二进制分割图像，并再次使用 `matplotlib` 进行可视化，同时保存到指定路径。

结果检查：最后，我们打印出优化问题的求解状态和最优值，以验证问题是否成功解决。

3 result of experiment

初始的圆形图像，优化过后的圆形图像以及初始的三角形图像，优化过后的三角形图像如附图所示，可见，本次实验成功的对其进行了凸优化（但是因为初始值较小，所以视觉效果不明显）。

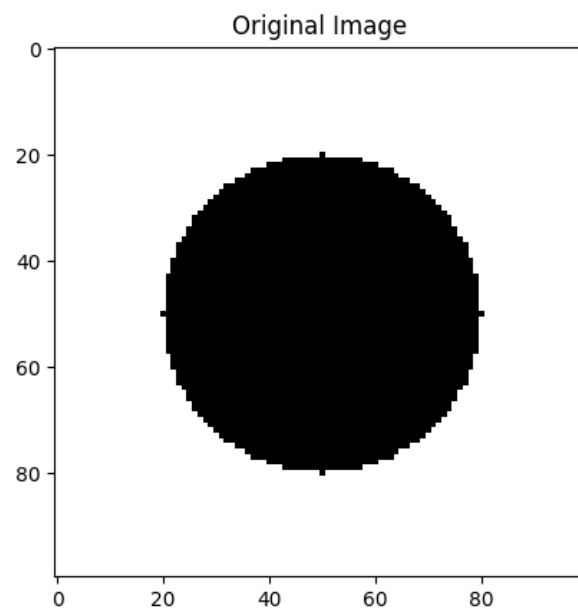


图 1: 原始的 circle 图片

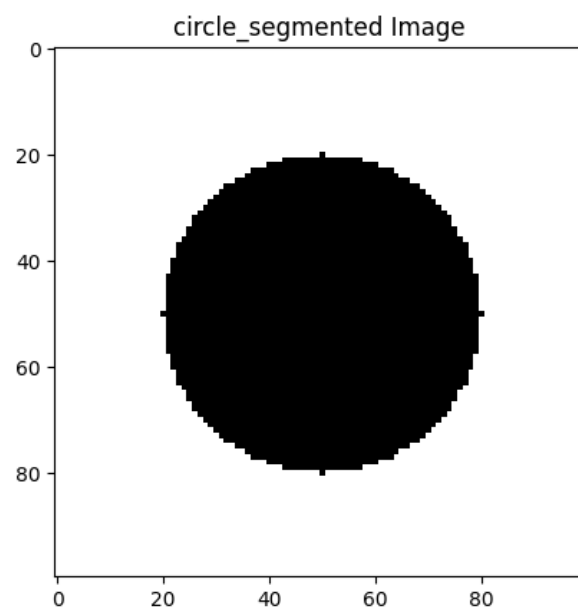


图 2: 处理后的 circle 图片

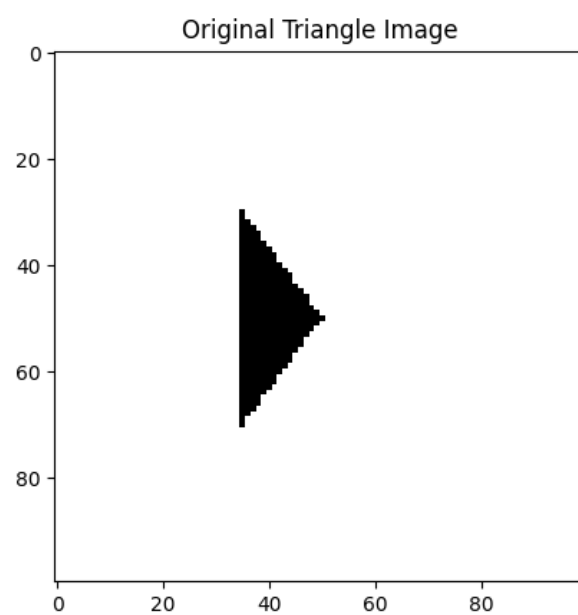


图 3: 原始的 triangle 图片

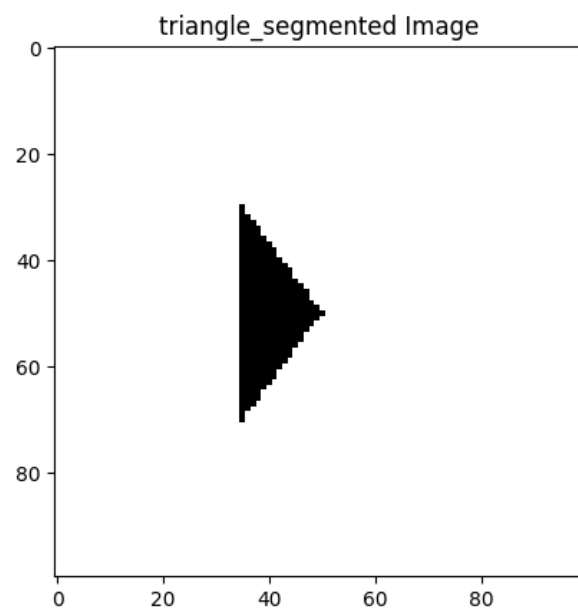


图 4: 处理后的 triangle 图片