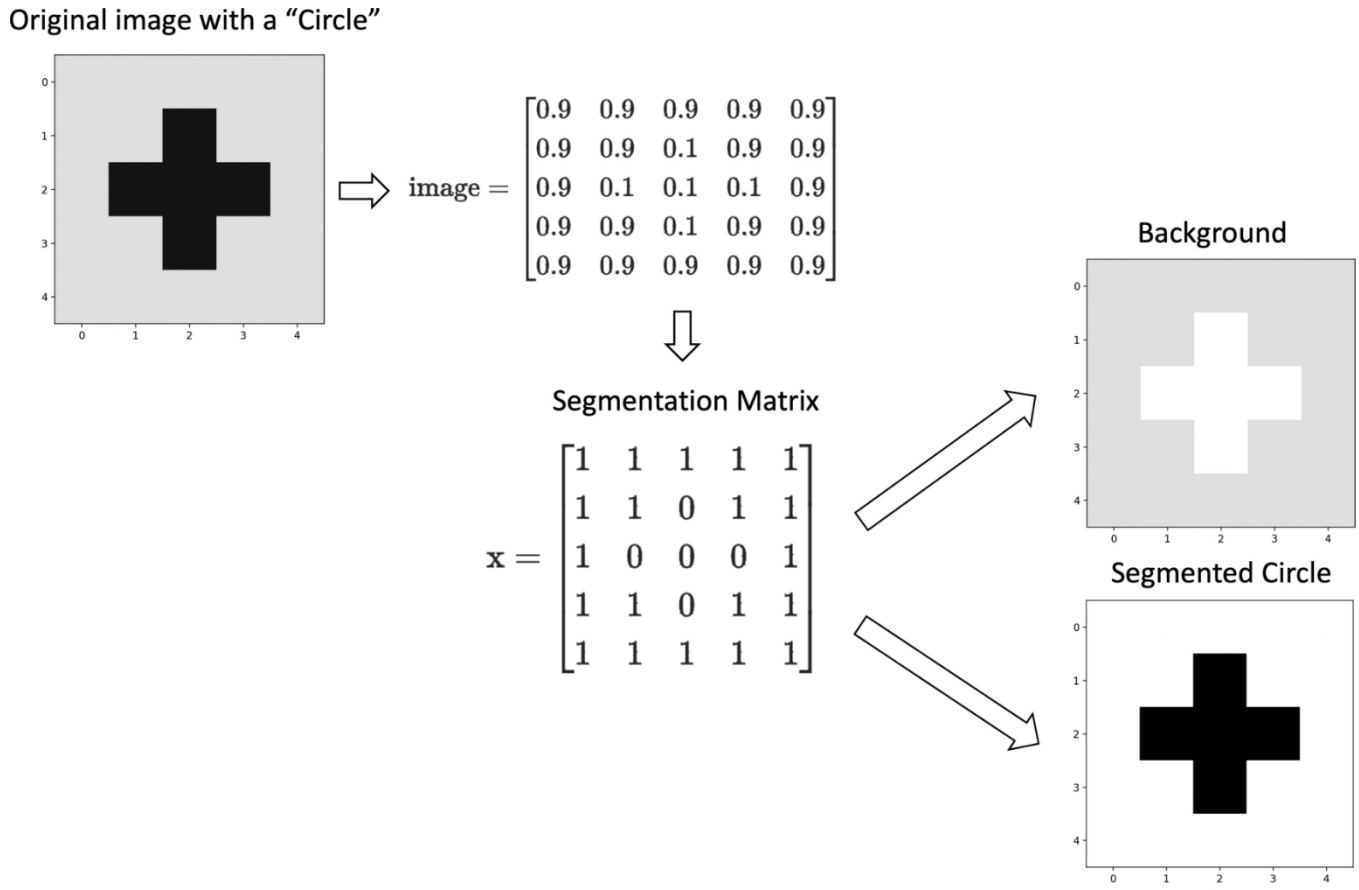
图像分割的凸优化

1.介绍 n

凸优化是一种强大的工具，广泛应用于金融、工程和机器学习等各个领域。在本作业中，我们将探讨其在图像处理领域的应用，特别是在图像分割任务中的应用。图像分割包括将图像分割成更易于分析和解释的有意义的部分。

2.作业 Objectiv e

图像分割是指将图像分割成若干段，以简化或改变图像的表示形式，使其更有意义、更易于分析。具体来说，您将分割一幅由白底黑字圆圈组成的合成图像。



如上图所示，我们可以将灰度图像表示为$\text{image}[i,j]$，其中 $\text{image}[i,j]\in$ 表示(i,j) 处的像素强度。让分割$[0,1]$矩阵x 成为图像中每个像素的二进制变量，其中 $x_{i,j}$ 为 1，否则为 0。

优化问题

要进行图像分割，我们需要找到最优二元变量$x_{i,j}$，使分割区域与真实区域之间的差异最小。这可以表述为一个优化问题：我们希望找到一个最优 x，使分割区域与相应真实区域的平方差之和最小。

目标函数

具体来说，目标函数定义如下

$$\text{Minimize} \quad \sum_{i,j}(x_{i,j} \times \text{image}[i,j])^2 + \sum_{i,j}((1 - x_{i,j}) \times (1 - \text{image}[i,j]))^2$$

给你

- 第一项是对错误地将圆的像素指定为背景的一部分进行惩罚（其中 $x_{i,j}=1$）。
- 第二项对错误地将背景像素指定为圆的一部分进行惩罚（$x_{i,j}=0$）。

通过最小化这个目标函数，我们的目的是将圆从背景中分割出来。

制约因素

- 每个$x_{i,j}$必须为 0 或 1，反映每个像素的分割选择。

$$x_{i,j} \in \{0,1\} \quad \text{for all pixels} \quad (i,j)$$

这一约束强制了分割的决定性，使每个像素要么是圆的一部分，要么是背景的一部分，不允许有中间值。

这种决策变量为整数的凸优化特例被称为整数优化。

编程。在 `cvxpy` 中，我们可以简单地将布尔标志设置为二进 正确 在定义变量时

制整数（0 或 1）。

数据集

我们使用下面提供的 Python 函数生成合成图像进行分割。该函数创建了一幅指定大小的图像，中间有一个指定半径的黑色圆圈。图像以二进制格式表示，其中 0.99 表示白色背景，0.01 表示黑色圆圈。

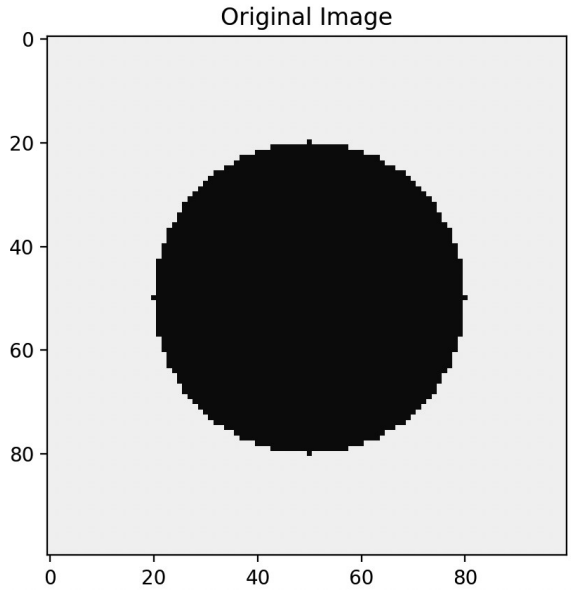
```
将 numpy 导入 np

def create_image(size, radius):
    图像 = np.ones((size, size))* 0.99 center
    = size // 2
    Y、X = np.ogrid[:size, :size]
    dist_from_center = np.sqrt((X - center)**2 + (Y - center)**2)
    mask = dist_from_center <= radius
```

```
    图像[掩码] = 0.01
    返回图像

size = 100 # 图像大小
半径 = 30 # 圆半径
图像 = create image（尺寸、半径）
```

下图是使用上述代码生成的，您也将使用它来生成分割任务的数据。



3.分配任务 s

任务描述：

1. 代码执行（**7** 分）：

    ○根据提供的函数生成合成图像并将其可视化。(1 分）

    ○使用 CVXPY 构建目标函数和约束条件正确的优化问题。(2 分）

    ○使用适当的求解器成功解决优化问题。(2 分）○ 可视化分割图像。(1 分）

    ○您还可以：（从以下选项中任选一项，1 分）

        ■尝试其他合成图像（如正方形、三角形等其他形状）来测试分割算法。

        ■ 改进算法，并将其应用于真实世界的图像。

2. 撰写报告（**3** 分）：

    ○描述优化问题中使用的目标函数和约束条件。(1 分)○ 简要介绍代码实现的逻辑。(1 分）

    ○通过提供原始图像和分割后的图像来说明结果，并讨论如何有效地进行分割。(1 分）

    ○报告应简洁、清晰、结构合理。为节省时间，报告应在 2 页以内（不包括图片）。

提示

- 要在 CVXPY 中设置布尔变量，可以使用二进制变量。`cp.Variable(..., boolean=True)` 以确定
- 您可以使用 CVXPY 中的求解器 `problem.solve(求解器=cp.ECOS_BB)`，其中 `ECOS_BB` 支持布尔变量。

4.提交 n

- 提交一个名为 `StudentID_StudentName.zip` 的压缩文件，其中包括
  1. 将 Python 代码文件放在名为 `code` .NET 的单独文件夹中。
  2. PDF 报告（命名为 `StudentID_StudentName.pdf` ）。

```
StudentID_StudentName.zip
├── 代码
    ├── main.py
└── ...
└── StudentID_StudentName.pdf
```

- 如果提交的格式不正确，您可能会被扣 1 分。