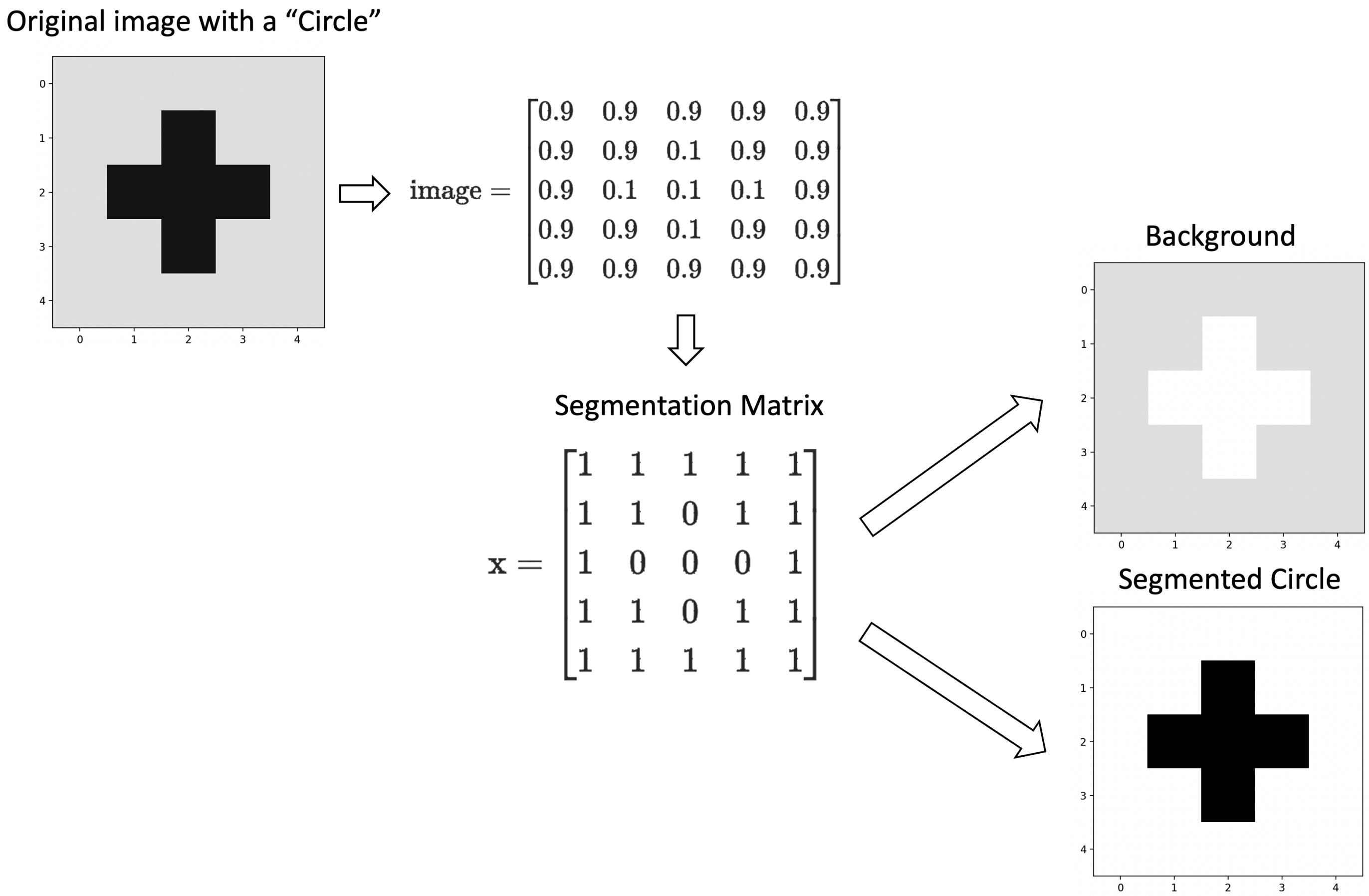
Convex Optimization for Image Segmentation

1. Introduction

Convex optimization is a powerful tool widely used in various fields such as finance, engineering, and machine learning. In this assignment, we explore its application in the domain of image processing, specifically for the task of image segmentation. Image segmentation involves dividing an image into meaningful parts which are easier to analyze and interpret.

2. Assignment Objective

Image segmentation involves dividing an image into segments to simplify or change the representation of an image into something that is more meaningful and easier to analyze. Specifically, you will segment a synthetic image consisting of a black circle on a white background.



As shown in the figure above, we can denote the grayscale image as $\mathrm{image}[i, j]$, where $\mathrm{image}[i, j] \in [0, 1]$ represents the pixel intensity at location (i, j). Let the segmentation matrix x be a binary variable for each pixel in the image, where $x_{i,j}$ is 1 if the pixel belongs to the black circle and 0 otherwise.

Optimization Problem

To perform image segmentation, we need to find the optimal binary variables $x_{i,j}$ that minimize the difference between the segmented regions and the true areas. This can be formulated as an optimization problem: we want to find an optimal x that minimizes the sum of the squared differences between the segmented regions and their corresponding true areas.

Objective Function

Specifically, the objective function is defined as:

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

Here:

- The first term penalizes incorrectly assigning pixels of the circle as part of the background (where $x_{i,j} = 1$).
- The second term penalizes incorrectly assigning pixels of the background as part of the circle (where $x_{i,j} = 0$).

By minimizing this objective function, we aim to segment the circle from the background.

Constraints

- Each $x_{i,j}$ must be either 0 or 1, reflecting the segmentation choice at each pixel.

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

This constraint enforces the decision nature of the segmentation, making each pixel either part of the circle or the background, with no intermediate values allowed.

Such a special case of convex optimization, where the decision variables are integers, is known as Integer Programming. In `cvxpy`, we can simply set the boolean flag to `True` when defining the variables to indicate that they are binary integers (0 or 1).

Dataset

We generate a synthetic image for this segmentation with the provided Python function below. This function creates an image of specified size with a black circle of a given radius centered in the middle. The image is represented in a binary format where 0.99 indicates a white background and 0.01 indicates a black circle.

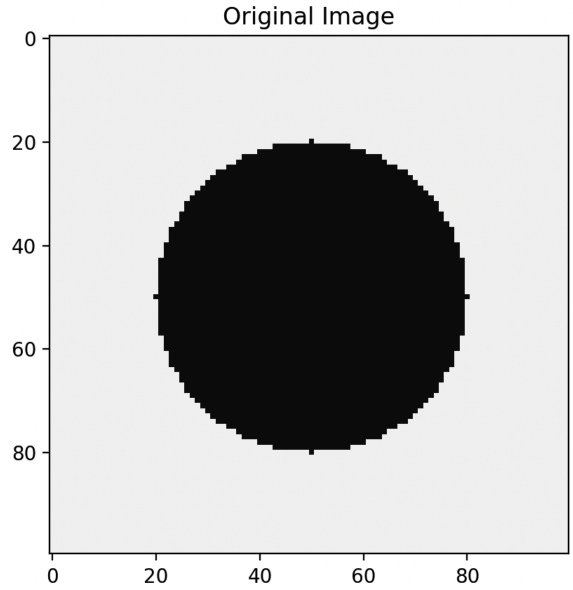
```
import numpy as np

def create_image(size, radius):
    image = 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
```

```
    image[mask] = 0.01
    return image

size = 100  # Image size
radius = 30  # Circle radius
image = create_image(size, radius)
```

The image below was generated using the above code, which you will also use to produce your data for the segmentation task.



3. Assignment Tasks

Task Description:

1. **Code Implementation (7 points)**:
    - Generate and visualize the synthetic image based on the provided function. (1 point)
    - Construct the optimization problem using CVXPY with correct objective function and constraints. (2 points)
    - Successfully solve the optimization problem with the appropriate solver. (2 points)
    - Visualize the segmented image. (1 point)
    - You may also: (pick one of the following options, 1 point)
        - Try other synthetic images (e.g. other shapes like squares, triangles, etc.) to test the segmentation algorithm.
        - Improve the algorithm and apply it to real-world images.
2. **Report Writing (3 points)**:
    - Describe the objective function and the constraints used in the optimization problem. (1 point)
    - Briefly introduce the logic of the code implementation. (1 point)
    - Include the results by providing the original and segmented images, discussing how effectively the segmentation was achieved. (1 point)
    - The report should be concise, clear, and well-structured. To save your time, it should be within 2 pages (excluding the images).

Hints

- To set up boolean variables in CVXPY, you can use `cp.Variable(..., boolean=True)` to define binary variables.
- You may use the `ECOS_BB` solver in CVXPY with `problem.solve(solver=cp.ECOS_BB)`, which supports Boolean variables.

4. Submission

- Submit a single zip file named `StudentID_StudentName.zip` that includes:
  1. The Python code file(s) in a separate folder named `code`.
  2. The PDF report (named as `StudentID_StudentName.pdf`).

```
StudentID_StudentName.zip
├── code
│   ├── main.py
│   └── ...
└── StudentID_StudentName.pdf
```

- You may get penalized with 1 point if the submission is not in the correct format.