Seeded Region Growing Image Segmentation using top-hat transform

Atharva Tambat 210070014
Deepasha 210070025
Aditya Yadav 210050006

April 23, 2023

Problem statement

Implement the seeded region growing algorithm for segmentation of an image. Use Top-hat transform to generate seed points.

Image segmentation

Seeded region growing is a common image segmentation technique used to group pixels or voxels in an image into distinct regions based on similarity criteria.
The basic formulation of the problem is as follows:

$$(a) \bigcup_{i=1}^{n} R_i = R.$$

$$(b)\ R_i \text{ is a connected region, } \mathrm{i} = 1,\ 2,\ \ldots, \mathrm{n}$$

$$(c)\ R_i \bigcap R_j = \varnothing, i \neq j$$

$$(d)\ P(R_i) = \text{TRUE for } i = 1, 2, \ldots, n.$$

$$(e)\ P(R_i \bigcup R_j) = \text{FALSE for any adjacent region } R_i \text{ and } R_j.$$

Region growing

There are a few fundamentally different approaches to the problem of image segmentation. Region-growing image segmentation methods rely mainly on the assumption that the neighboring pixels within one region have similar values. This approach to segmentation examines neighboring pixels of initial seed points and determines whether the pixel neighbors should be added to the region or not on the basis of a similarity criterion.

Initial seed points

The first step in the region growing is to select a set of seed points. According to the problem statement, this selection is to be done using top-hat transform. The regions are then grown from these seed points to adjacent points depending on a region membership criterion.

Top-hat transform

The top-hat transform is an operation that extracts small elements and details from given images. The top-hat transform returns an image, containing those "objects" or "elements" of an input image that:

- Are "smaller" than the structuring element, and
- are brighter than their surroundings.

Because of these properties, it is used to choose the initial seed points.

Solution

The algorithm followed is:

1. Initialize seed points using Top Hat Transform of the image
2. Add the seed points to a stack
3. Remove a point X, from the stack and get its 8-neighbor points
4. From the 8-neighbors of X, if a pixel Y is within a user-defined threshold of the X (i.e. $|gray\_level(X) - gray\_level(Y)| < Threshold$), then add Y to the same class as X and add Y to the stack, else leave Y unassigned.
5. Repeat the above two steps till all the pixels of the image are visited once
6. Make one pass over the entire image and classify the unassigned pixels into the class of their nearest neighboring pixel (i.e. $\min|gray\_level(X) - gray\_level(Y)|$)

Screenshot of GUI

1. Double click on the executable file named "img_segment" OR type ./img_segment on the terminal and wait for it to open
2. A window, like the one shown below will open.

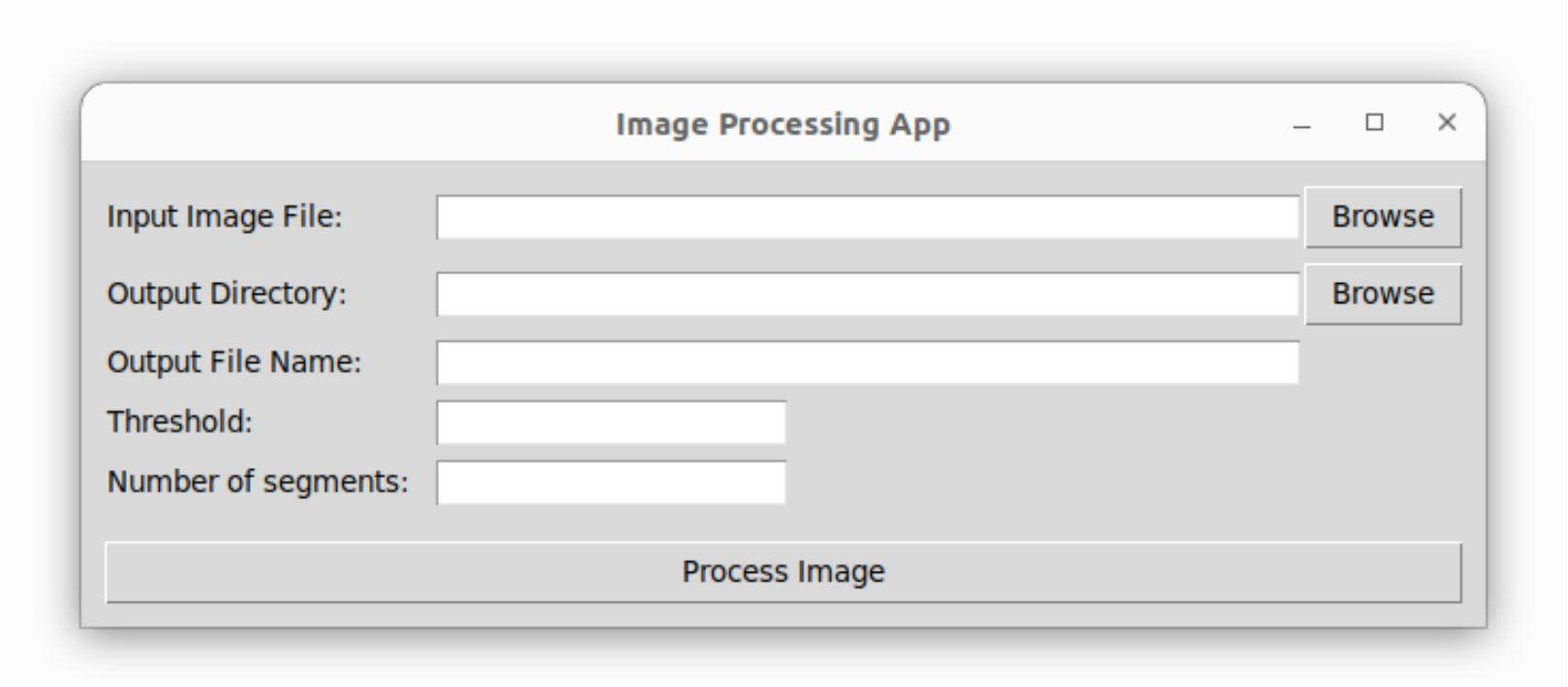

Figure: Application GUI

Screenshot of GUI

1. Choose the input file and the directory in which you would like your segmented image to be present
2. Type in the name of your segmented image file (with an image extension - .png, .jpg etc.)
3. Enter the threshold parameter
4. Enter the number of segments which you speculate your image to have
5. Press "Process Image"
6. A window (like the one below) with the grayscale version of the image with bright spots at the locations of the seed points calculated using TopHat Transform is displayed

Screenshot of GUI

Figure: Top Hat Transform of the Image

Screenshot of GUI

1. Press 'ESC' to continue
2. A dialog box, indicating that the segmentation was successful will be displayed if it was successful, or an error dialog box will be shown.
3. Click "OK" and repeat as many times as you want, before exiting the window by clicking "X" on the top right corner of the window.

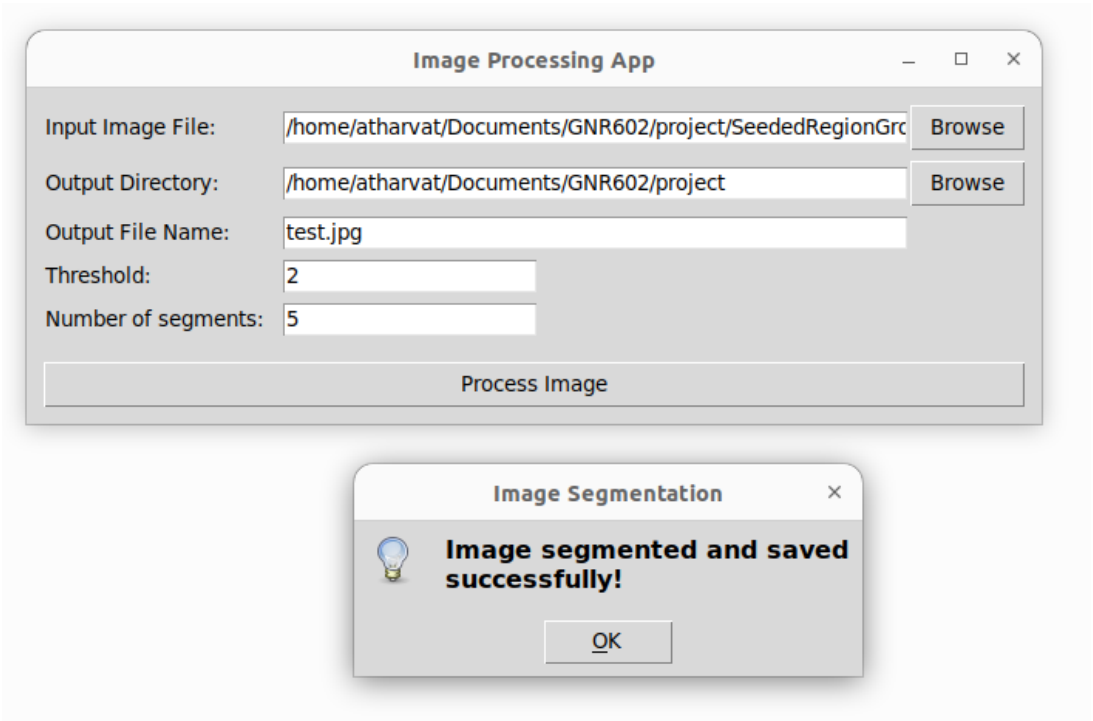

Figure: Successful image segmentation

Input data

The input dataset consists of aerial imagery of Dubai obtained by MBRSC satellites. Input data has been taken from: https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery

Input data

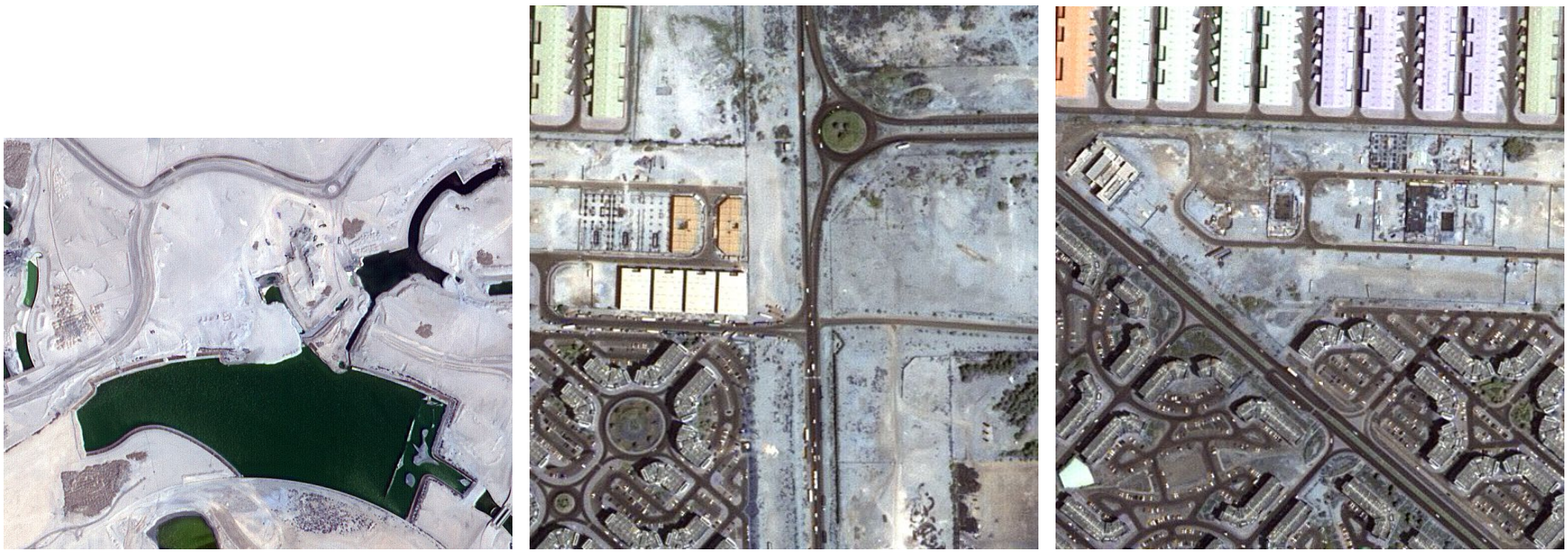

Figure: Some of the images used to produce the results

Results

Figure: Results obtained

Results

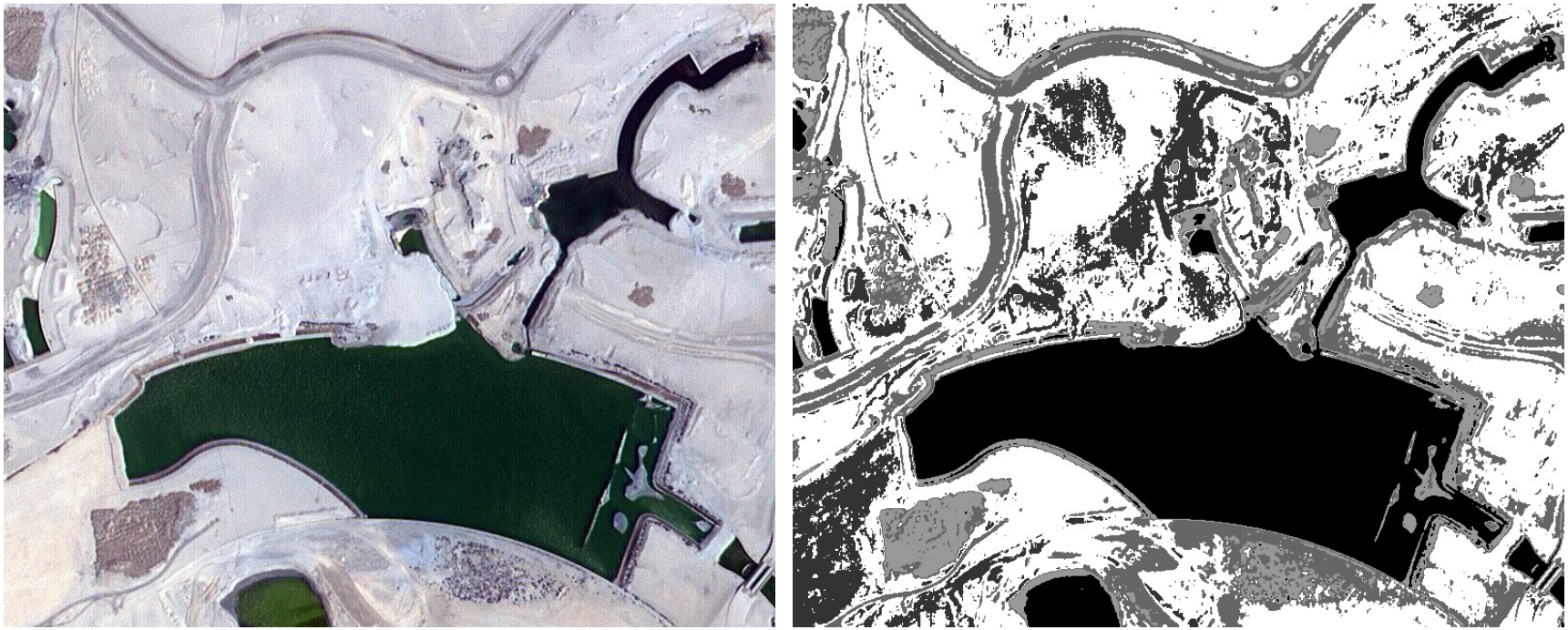

Figure: Results obtained

Results

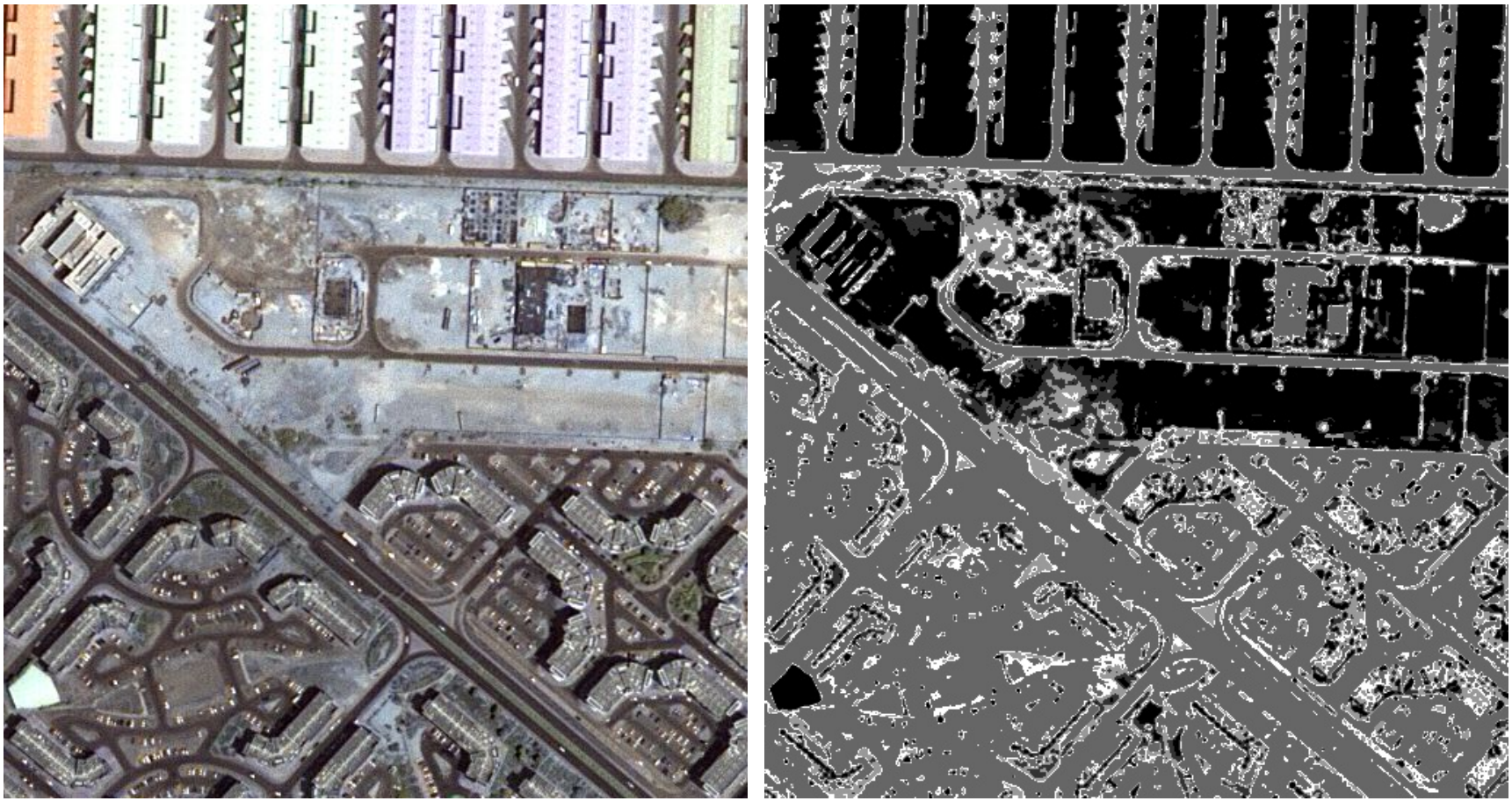

Figure: Results obtained

Results

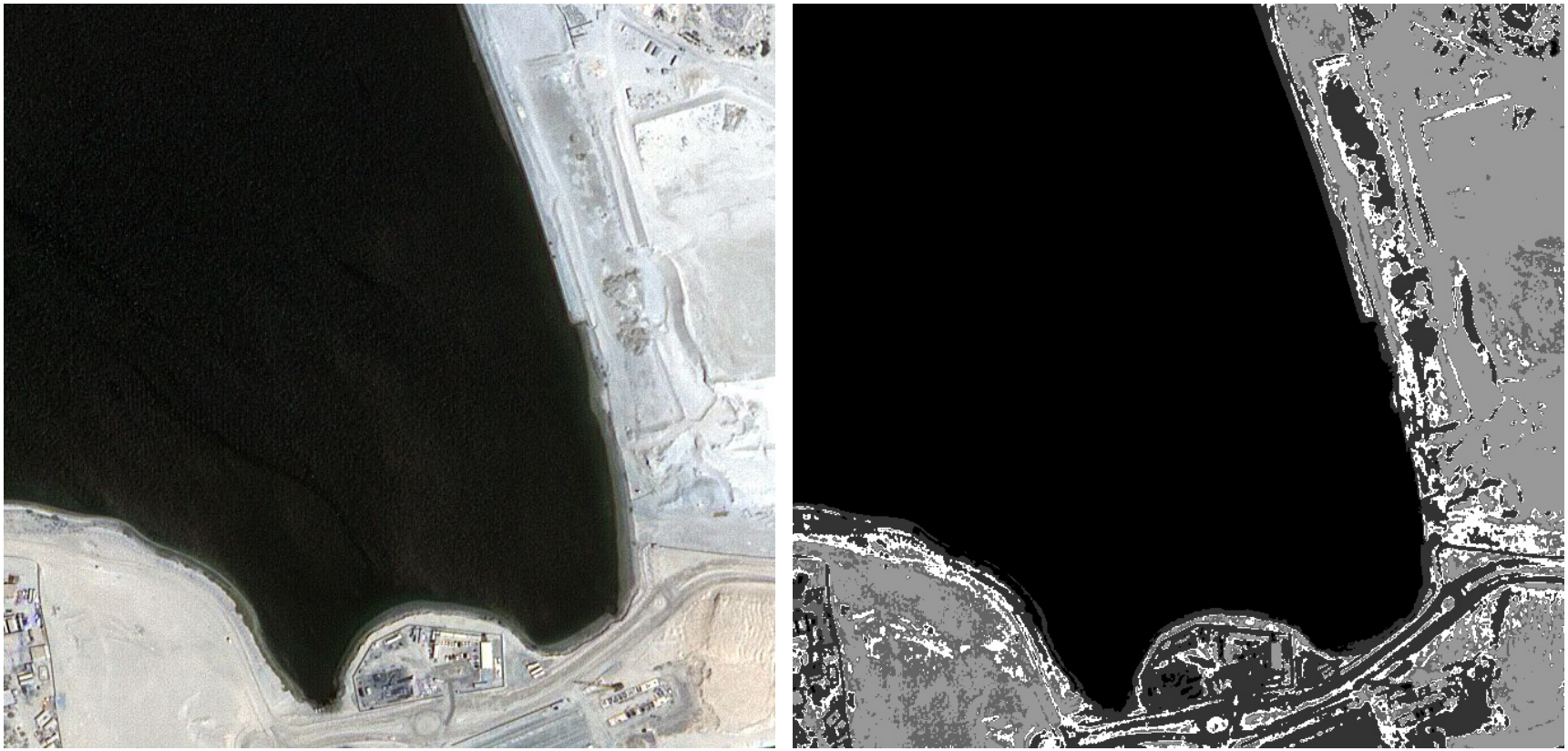

Figure: Results obtained

Results

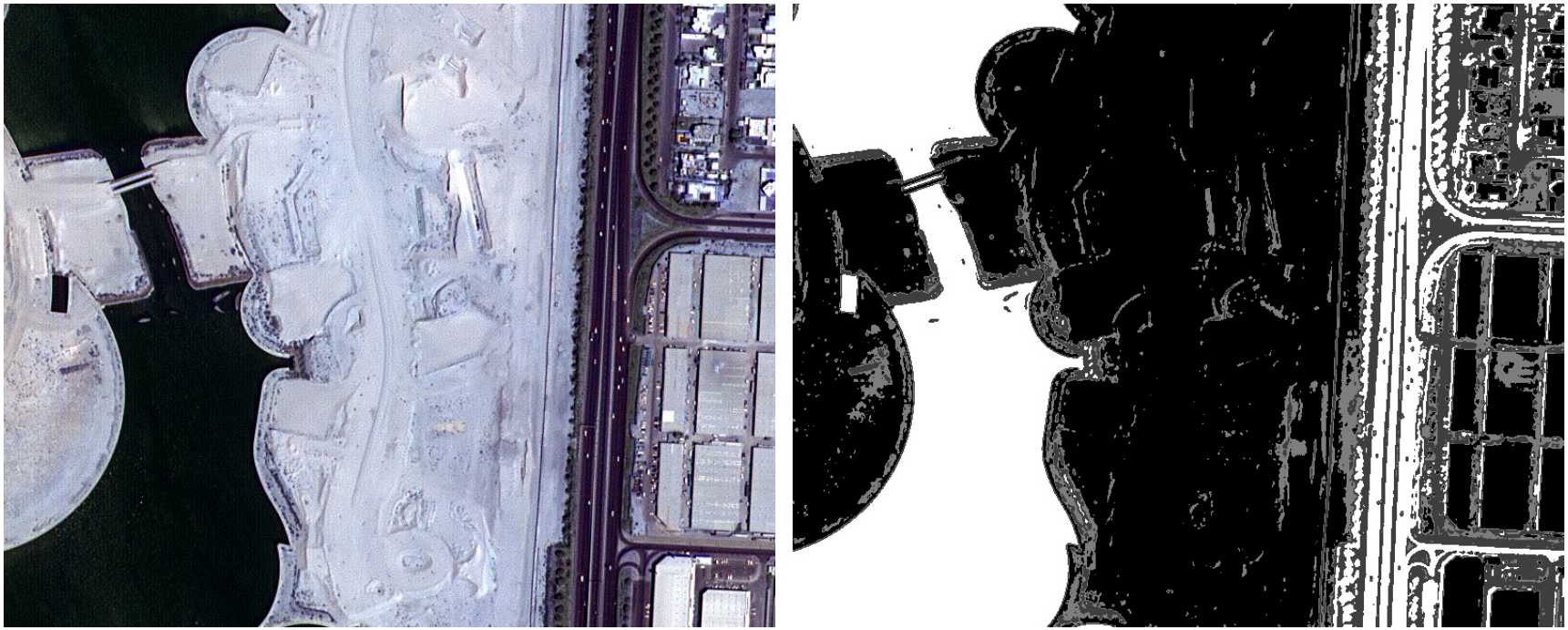

Figure: Results obtained

Results

Figure: Results obtained

Results

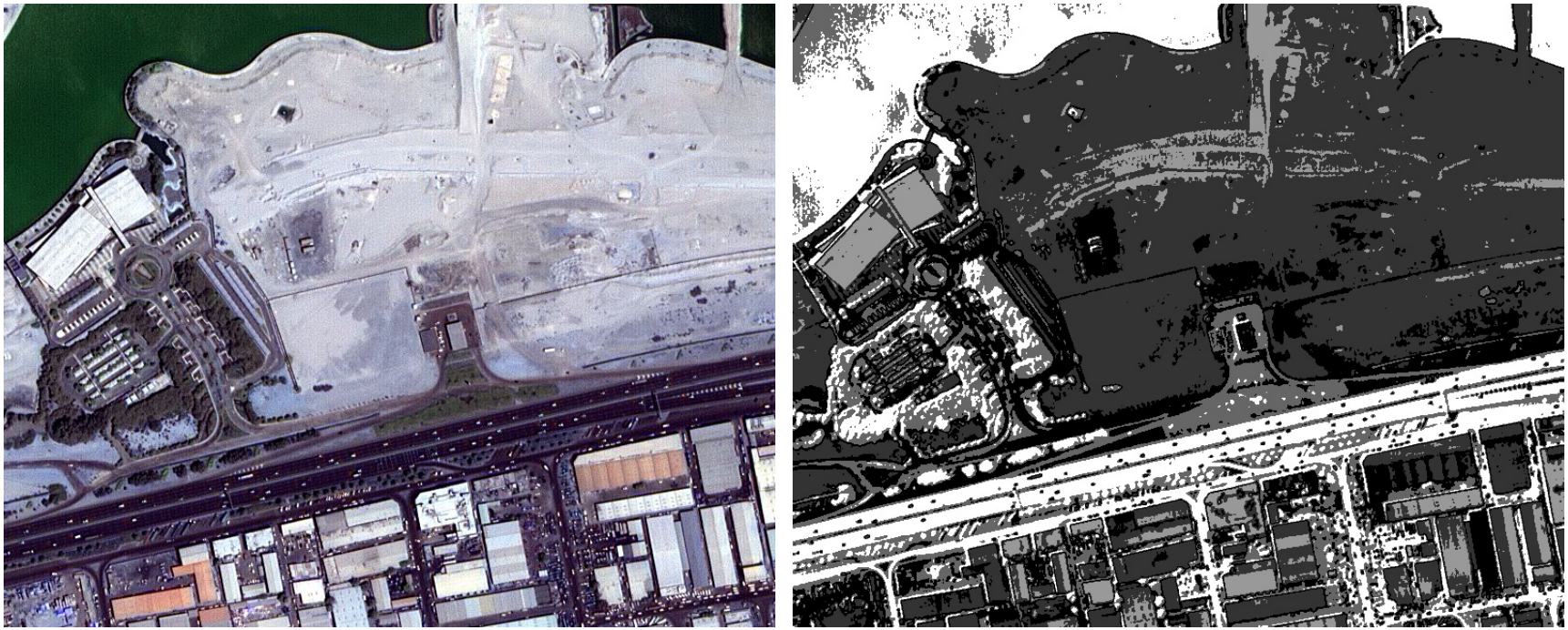

Figure: Results obtained

Thank You!