

## COMP2005 Laboratory Sheet 4: Thresholding & Binary Images

### 1. Binary Images

Thresholding commonly produces a binary image with pixel values of either 0 (black) or 255 (white). Binary images are often used in image processing for edge detection and segmentation. To obtain a binary image, thresholding must take place. Figure 2 shows the binarized image obtained from performing thresholding on Figure 1.



Figure 1: Original Image



Figure 2: Binarized Image

### 2. Thresholding

Thresholding classifies the pixels in an image to either foreground or background based on the intensity values of the pixels. The pixel values of the image are set to either 0 or 255, depending on if it is above or below the decided threshold value. Using the *some text.png* image from the Moodle page:

- Use the *threshold* function and the *THRESH\_BINARY* flag to perform thresholding on the image with a threshold value of your choice.
- Perform Otsu thresholding by adding the *THRESH\_OTSU* flag and compare the results with your selected threshold value. Has Otsu picked the best threshold value?

For adaptive thresholding, there are two ways that the OpenCV function *adaptiveThreshold* uses to decide the threshold value. The first is by using the mean of the neighbourhood values, and the other is by using the Gaussian-weighted sum of the neighbourhood values.

- Using the *ADAPTIVE\_THRESH\_MEAN\_C* flag, tweak the *blockSize* and *C* parameters to achieve the best result.
- Do the same using the *ADAPTIVE\_THRESH\_GAUSSIAN\_C* flag. What is the difference between the two methods?
- Compare the results of the four methods. Which method produced the best result?
- Repeat the steps with the *Tulip.jpeg* from the Moodle page. Which method produced the best result?

### 3. Expected Results

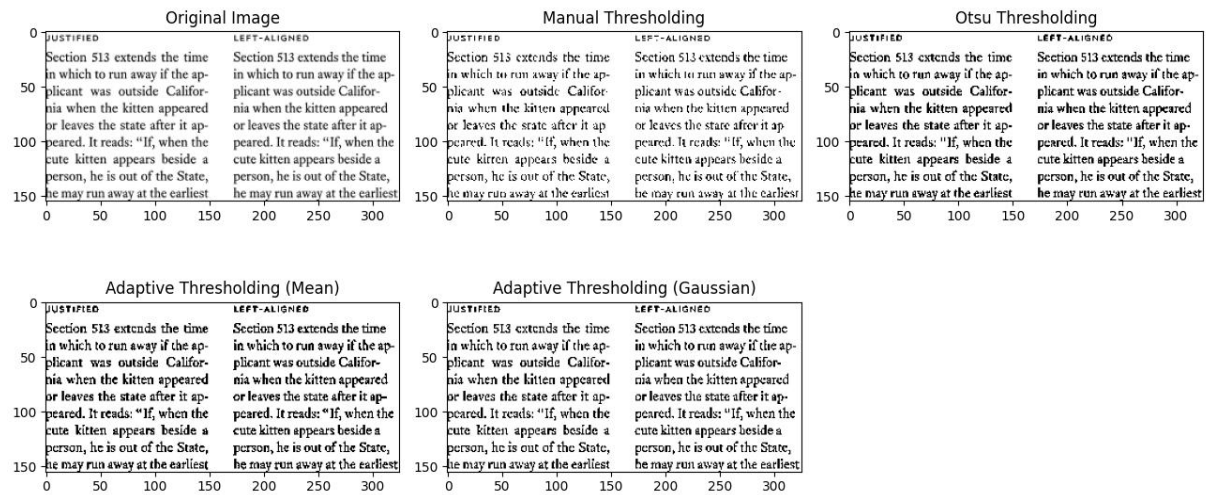


Figure 3: Thresholding using some text.png

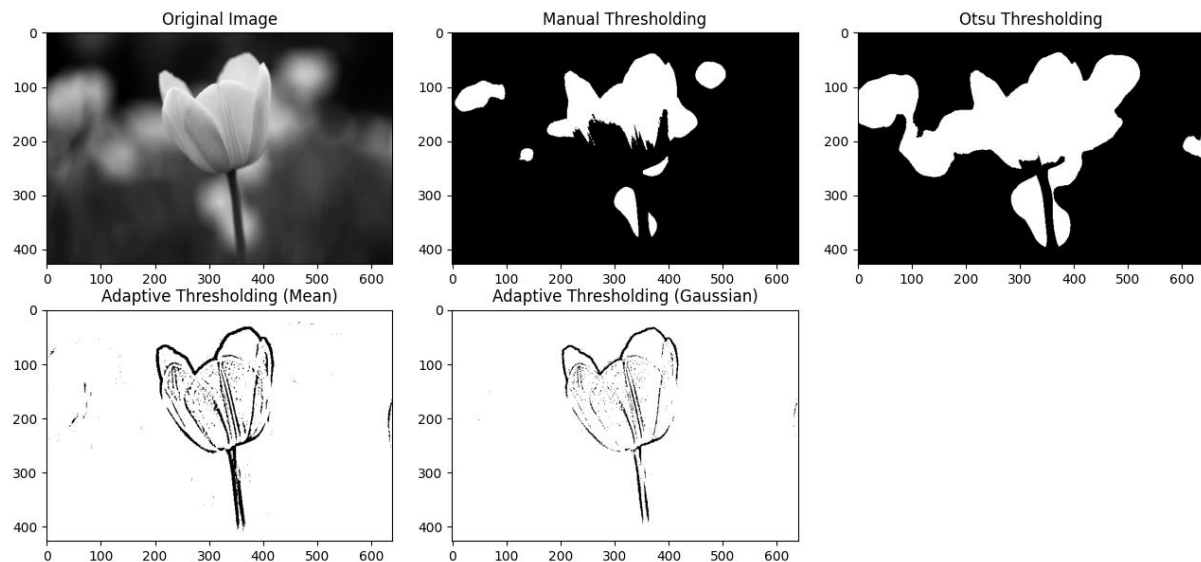


Figure 4: Thresholding using Tulip.jpeg