

# Homework 0: Image thresholding given a specific value

Christopher Kantor

Department of Computer Science and Engineering

University of South Florida

## 1. Introduction and description

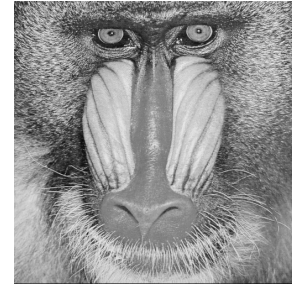
This assignment focuses on teaching students about fundamental image manipulation functions. These functions include adding intensity, binarization, and scaling. Students also had to implement a function that reduces a given pixel's intensity if that pixel had an intensity less than a given threshold. Through this assignment, students also learned about the .pgm format, and how to work with it.

## 2. Description of algorithms

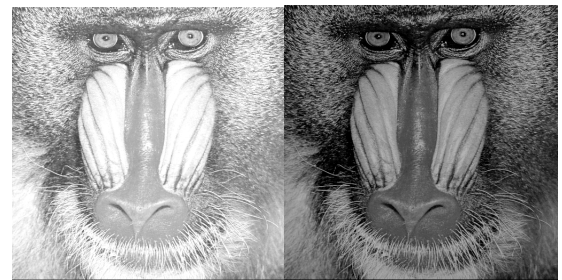
There are 4 algorithms in this project which will be described in the following sections. All the algorithms have a time complexity equal to the size of the image, as the algorithms process every pixel.

### 2.1. Intensity

This function takes a value as an input and adds it to each pixel's intensity. After this occurs, we check if the new intensity is between 0 and 255, setting it to whichever is closer in the case that it is not. We can use this function to brighten or darken any given image.



**Figure 1: Original image used for this project.**



**Figure 2: Images that have been manipulated using the intensity function. The left one had 50 intensity added to each pixel, while the right had 50 intensity removed from each pixel.**

### 2.2. Binarization

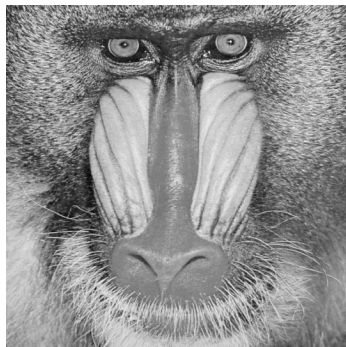
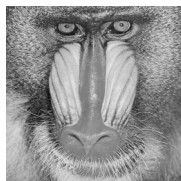
Given a threshold, set all pixels with intensity less than the threshold to 0, and all pixels with intensity greater than the threshold to 255. Useful for quickly seeing which pixels fall outside of a certain intensity range.



**Figure 3: Images that were binarized. The left one had a threshold of 50, and the right one had a threshold of 125.**

### 2.3. Scaling

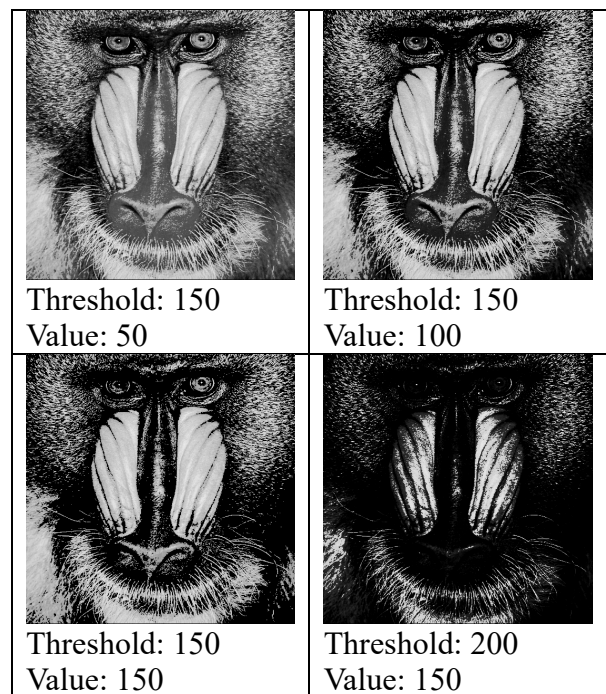
Reduces or expands the height and width of the image by either a factor of 2 or 0.5. If we are increasing the size of the image, each pixel in the original image is mapped to 4 in the new image, and we just copy the values from the original image. If we are shrinking the image, every 4 pixels in the original image are mapped to 1 pixel in the new image. In this case, we average the intensities of the 4 pixels in the original to get the value we will put in the new image.



**Figure 4: Images that have been resized using the scaling function.**

### 2.4. Threshold with a value

This function takes 2 parameters, a threshold, and a value. If a pixel has an intensity less than the threshold, we reduce it by the value, checking if it is within the range of 0 – 255. This is like the binarization function, but instead of setting each pixel to either 0 or 255, we change the pixels intensity by the value. We can use this function to change the intensity of pixels that lie below a certain range.



**Figure 5: Images that had the threshold value function applied to them. As the value and threshold increase, more of the image appears darker, as more pixels are affected by the function.**

## 3. Description of implementation

The threshold with value function builds upon the code provided by the professor. We use the built in image class, which contains the necessary data and functions to store and manipulate images for our task. The utilities

class is where most of our changes are made. We added our new function in this file and made some slight modifications to read in the extra command line argument needed for this function.

#### **4. Conclusions**

In this assignment, students were tasked with implementing a value thresholding function that would work with the source code provided. During this process, we learned how some basic image manipulation functions worked, and how to process, store, and view images in the .pgm format. These skills will be used during the rest of the semester as we tackle more and more complex ways to process digital images.