Assignment 1: Basic Image Manipulation Techniques and Experimentation with Image Smoothing and Binarization Techniques.

KEJVI CUPA
Department of Computer Science and Engineering
University of South Florida, Tampa, Florida, USA

1. Introduction and overall description

This assignment focuses on practicing image manipulation techniques and experimentation with image smoothing and binarization techniques. Regions of Interest (ROI) are presented in this assignment and allow certain image manipulations to be performed in certain regions of the image rather than the whole image. Up to three regions of interest can be used on an image. Image processing will be applied to Grey level and color images. Overall, there will be five new functions that will be implemented: add, binarize, uniformsmooth, adptvsmooth, and mulcolorbright. A more detailed description of each function will be provided in this report.

2. Description of Algorithm

The first algorithm to be implemented is “add”. The purpose of this function is to change the brightness of the image by adjusting the pixel values by a given intensity value p1. Instead of adjusting the brightness on the entire image, you can specify up to three ROI(region of interest) by providing the initial row(x), initial columns(y), total number of pixels across the x axis(Sx) and total number of pixels across the y axis(Sy) to build the ROI. You can pass a different intensity value for each ROI. The ROIs will be implemented in all following functions so the parameters required will be the same for the other functions as well. Therefore, I will not mention the ROI component on every function because it will be the same.

The second algorithm to be implemented is “binarize”. In addition to the ROI parameters, the user provides the threshold parameter values. If pixel value is less than threshold, set pixel value to MINRGB else set to MAXRGB. A check function is in place to make sure the values are within the 0-255 range.

The third algorithm to be implemented is “uniformsmooth”. In addition to the ROI parameters, the user would need to provide the “window size” parameter value which represents the kernel size. The kernel size represents the number of all surrounding pixels to the current pixel, which will be summed and divided by the number of all pixels used (window size squared). The average value is then used to replace the current pixel value. In case the kernel size is large, and the current pixel location is such that it is in the edges, a condition has been implemented to not take those values into consideration and reflect their absence in the average calculation. So when a kernel pixel is supposed to be in a location where column or row is < 0 or > 255, it will be skipped. This is "mean smoothing" so it is uniform smoothing.

The fourth algorithm to be implemented is adaptive smoothing "adptvsmooth" which is the same as the third algorithm, but takes an extra parameter per ROI, threshold. If the change of the pixel value due to the uniform smoothing is larger than the threshold, then the value of that pixel will remain unchanged. This basically allows for a better smoothing which is uniform smoothing with threshold.

The fifth algorithm to be implemented is "mulcolorbright". This algorithm allows one to adjust the brightness of a color image by changing the pixel value of one of the channels (RED, GREEN, BLUE). In addition to the ROI parameters the user needs to pass floating parameter values (DR, DG, DB). These values are multiplied with the respective pixel's channel. This will allow color changes based on the channels to be very noticeable.

3. Results

Below are shown the results (images) with different combinations of the respective function parameters.

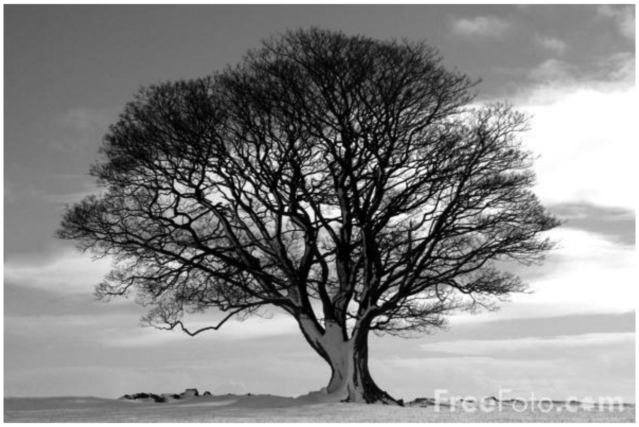

Figure 1: Original Image

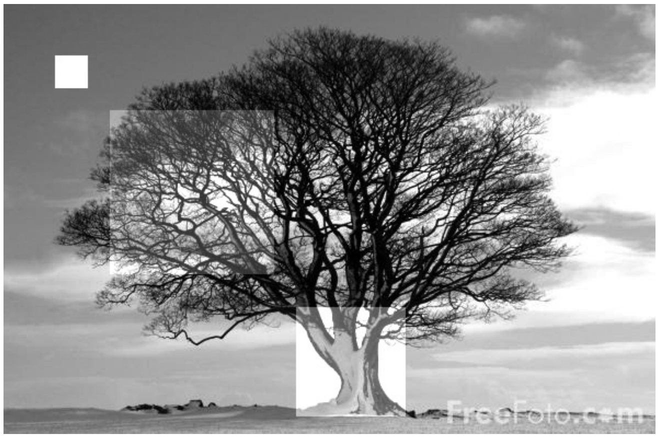

Figure 2: “add” function. Changing image brightness/intensity on the 3 used ROIs. Different intensity values were used for each ROI, and you can notice those changes based on the output. The used parameter configuration is: #roi = 3, x1= 100, y1= 100, Sx1= 150, Sy1= 150, Intensity1= 50, x2 = 50, y2 = 50, Sx2 = 30, Sy2 = 30, Intensity2 = 150, x3 = 280, y3 = 270, Sx3 = 100, Sy3 = 100, Intensity 3 = 90

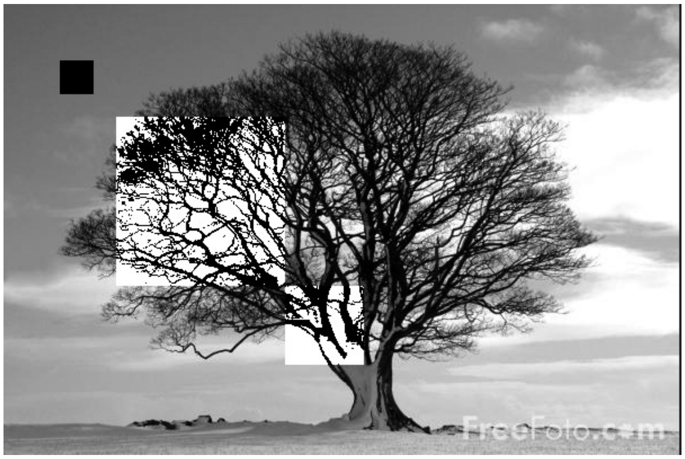

Figure 3: “binarize” function. Changing image pixel value based on the 3 used ROIs based on the provided thresholds. Different threshold values were used for each ROI, and you can notice the difference based on the output. The used parameter configuration is: #roi = 3, x1= 100, y1= 100, Sx1= 150, Sy1= 150, Threshold1= 50, x2 = 50, y2 = 50, Sx2 = 30, Sy2 = 30, Threshhold2= 150, x3 = 250, y3 = 250, Sx3 = 70, Sy3 = 70, Threshold3 = 40

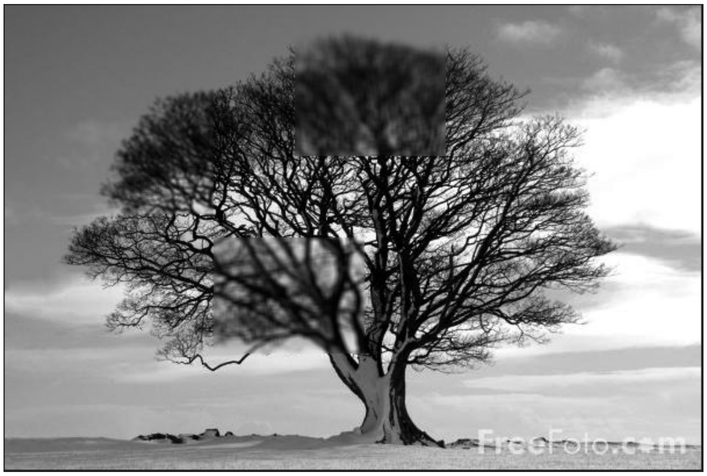

Figure 4: “uniformsmooth” function. Uniform smoothing is performed on the ROI using a window size (kernel size) passed by the user for each ROI. The average of the pixels in the kernel is used to replace the current pixel value. Different window size values were used for each ROI, and you can notice the difference based on the output. The used parameter configuration is: #roi = 3, x1= 0, y1= 0, Sx1= 180, Sy1= 180, WindowSize1= 3, x2 = 200, y2 = 180, Sx2 = 100, Sy2 = 130, WindowSize2= 5, x3 = 0, y3 = 250, Sx3 = 130, Sy3 = 130, WindowSize3 = 5

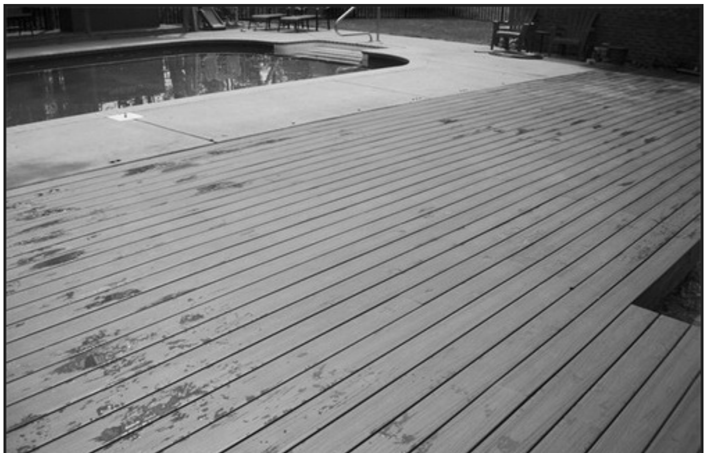

Figure 5: Original Image

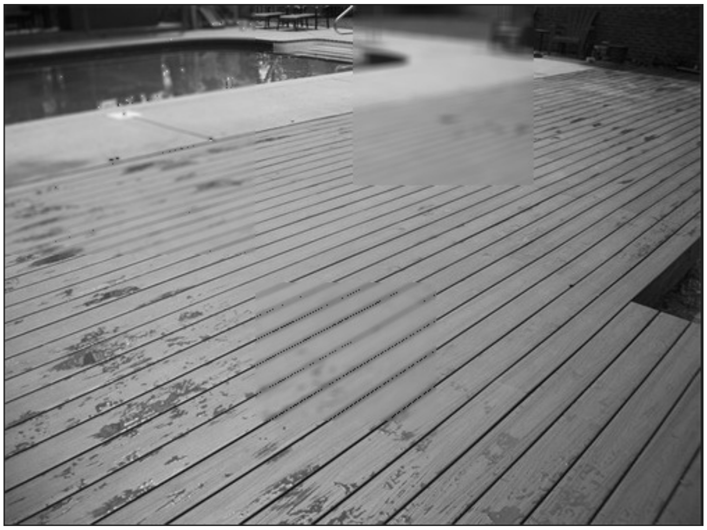

Figure 6: “adptvmooth” function. Uniform smoothing is performed on the ROI using a window size (kernel size) passed by the user for each ROI. The average of the pixels in the kernel is used to replace the current pixel value. A new parameter “threshold” is passed by the user. If the pixel value change due to the uniform smoothing is greater than the threshold, the pixel value will remain unchanged. Different window size and threshold values were used for each ROI, and you can notice the difference based on the output. The used parameter configuration is: #roi = 3, x1= 0, y1= 0, Sx1= 180, Sy1= 180, WindowSize1= 3, Threshold1 = 50, x2 = 200, y2 = 180, Sx2 = 100, Sy2 = 130, WindowSize2= 5, Threshold2 = 80, x3 = 0, y3 = 250, Sx3 = 130, Sy3 = 130, WindowSize3 = 7, Threshold3 = 150

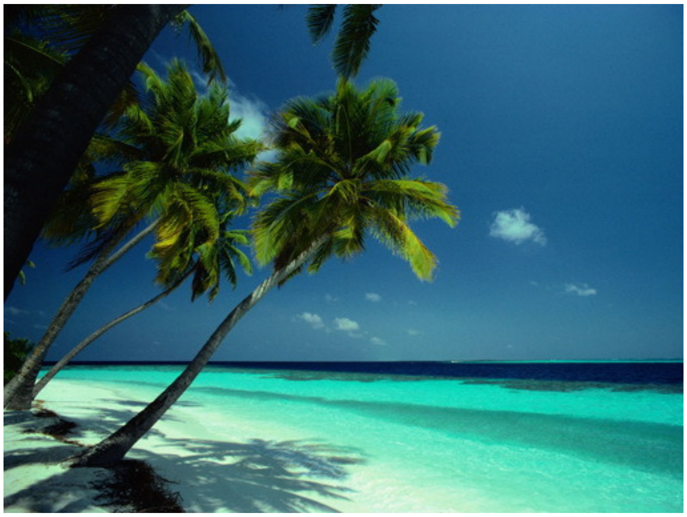

Figure 7: Original Color Image

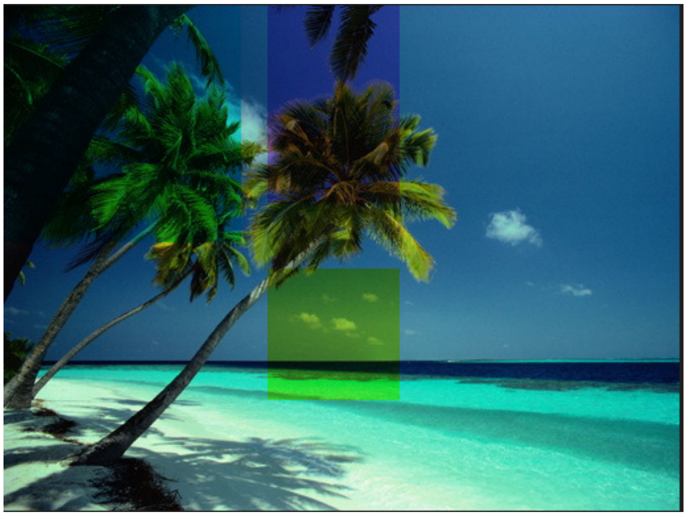

Figure 8: “mulcolorbright” function. Changing image brightness/intensity on the 3 used ROIs. Since this is a Color image, there are three channels (RED, GREEN, BLUE) and their respective channel values need to be changed. Different intensity values were used for each ROI, and you can notice those changes based on the output. The intensity values were float values and are multiplied with the current pixel value. The used parameter configuration is: #roi = 3, x1= 0, y1= 0, Sx1= 180, Sy1= 180, DR= 0.2, x2 = 0, y2 = 200, Sx2 = 150, Sy2 = 100, DG = 0.6, x3 = 200, y3 = 200, Sx3 = 100, Sy3 = 100, DB = 0.3

4. Discussion of Results

The use of the regions of interest (ROI) is very useful. Not only are we able to only perform manipulation on a specific region of the image, but it also allows us to better visualize the difference between the original image and the processed part. We are able to better understand visually the effects of the manipulation. In addition to that, we noticed how different parameter values provided different image output results, as expected. While there are many different combinations you can use, it is important to use values that are within the specifications of the algorithm and the constraints of the image. Furthermore, using different ROIs is very beneficial and cuts time consumption and costs in trying to manipulate and optimize an image. Instead of using resources to change the entire image, you can do that for a specific region and evaluate the result. Then you can optimize further from there. Based on the results, all algorithms worked as intended. The first four algorithms were used to manipulate Grey level images, and the fifth one was used to manipulate Color Images. Color image manipulation is more complicated than Grey level due to the presence of three channels (RED, GREEN, BLUE) instead of only one. Nevertheless, the results of that algorithm were as expected.

5. Conclusion

Overall, in this assignment we were able to implement and test different image manipulation techniques and had the chance to experiment with smoothing and binarization. Furthermore, we started performing image processing on Color images. The implemented algorithms are very useful in image manipulations and allow a vast combination of parameters to provide different image manipulation results. Due to the implementation of up to three ROIs the functions would take many parameters hence making the functions a bit more complicated, but we already discussed how ROI can be extremely useful in Image Processing. Ultimately, this assignment provided a productive way to further improve our knowledge in Image Processing.