jkilver3

Assignment 2

COMPUTATIONAL PHOTOGRAPHY

JACOB KILVER

1. Question 1 – numberOfPixels

The number of pixels in an image is the area of the picture. As such, it is the product of the number of pixels along the length of the image and the number of pixels along the height of the image

2. Question 2 – averagePixel

Since numpy.mean and numpy.average could not be used, I summed all the values in the image array using numpy.sum. Then I used the result from the numberOfPixels method above to divide by the total number of pixels in the image. This result was then converted to the integer type.

3. Question 3 – convertToBlackAndWhite

I started by creating a numpy array the same size of as the original image that was filled with zeros. Then, all that was necessary was to determine what pixels were above 128. So I used a double nested for loop to iterate through all the pixels in image. If the value was above 128, I set the value of the new image to 255. The values at 128 and below were already set to zero, so they did not need changed. The resulting image is shown in Figure 2 while the input is in Figure 1. Notice that this operation has the effect of heightening the contrast of the image.

Figure 1: Image input for convertToBlackAndWhite

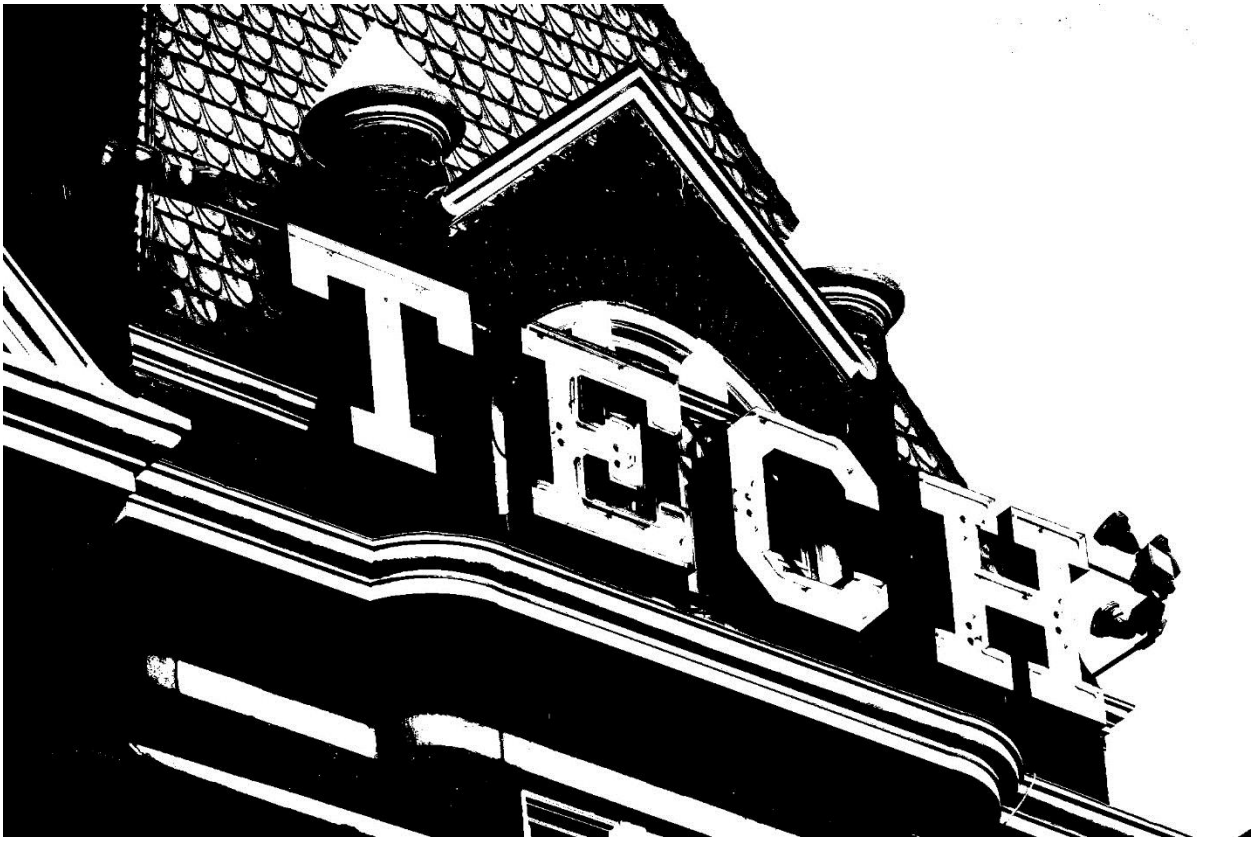

Figure 2: Binary image output

4. Question 4 – averageTwoImages

I followed the instructions given in instructions for this function. It says to add the two images on a per pixel basis, then divide by two. There are some potential problems with values being truncated if the sum is greater than 255, but the instructions say to add first, then divide. A safer approach would be to divide first, then add. This would preserve the information found in the two images better.

Since I used numpy arrays, I could just add the two images, shown in Figure 3 and Figure 4. I then multiplied the sum by 0.5 to make sure that the ratios between the two images were the same. The resulting image can be seen in Figure 5.

Figure 3: Input image 1

Figure 4: Input image 2

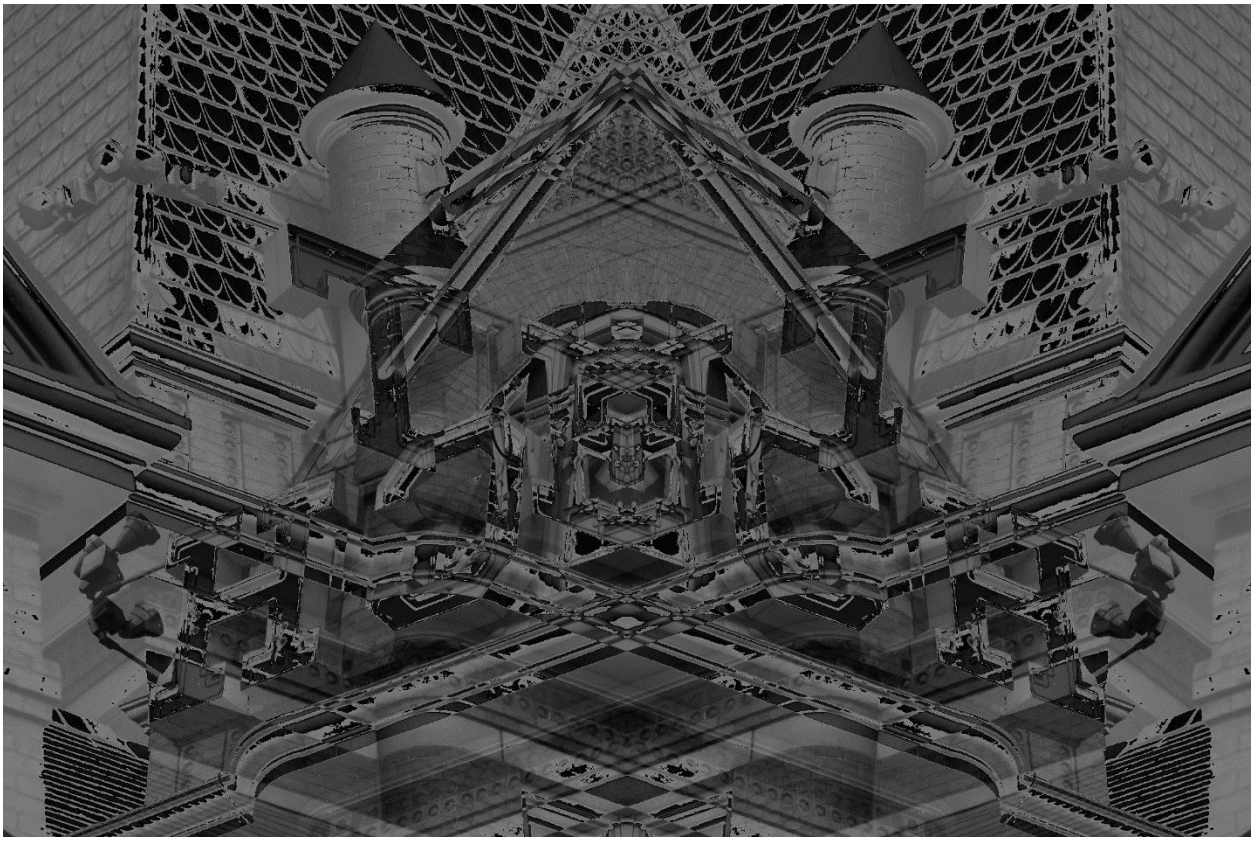

Figure 5: Average of two images

5. Question 5 - flipHorizontal

Since the input image was a numpy array, I took advantage of Python's negative indexing. A negative index in a Python array is counted from the end of the array. So I created an array of all zeros to hold the flipped image that was the same size as the original image. Then I iterated through each column and put it at the opposite end from where it started simply using the negation of the current index. This flipped the image along the horizontal axis. You can see the input image in Figure 6 and the output in Figure 7.

Figure 6: Input image for flipHorizontal

Figure 7: Horizontal flip of image