

CS 6476 Computer Vision

Assignment 1

Images as Functions

Kilver, Jacob R
8-25-2015

Beginning notes:

You should be able to run the Python file (ps1.py) without doing anything special. It should generate all the images in this document as well as the image statistics for question 4a. I assumed a directory structure where the folder with the Python file has two subfolders, one named input and one named output. The two input files are assumed to be in the directory names input. The images that are produced are saved to the folder named output.

1. Input Images

I put my two images both in the input directory (because that just seemed intuitive) and in the output directory as directed in the instructions.



Figure 1: ps1-1-a-1.png



Figure 2: ps1-1-a-2.png

2. Color planes

- a. Swap red and blue channels

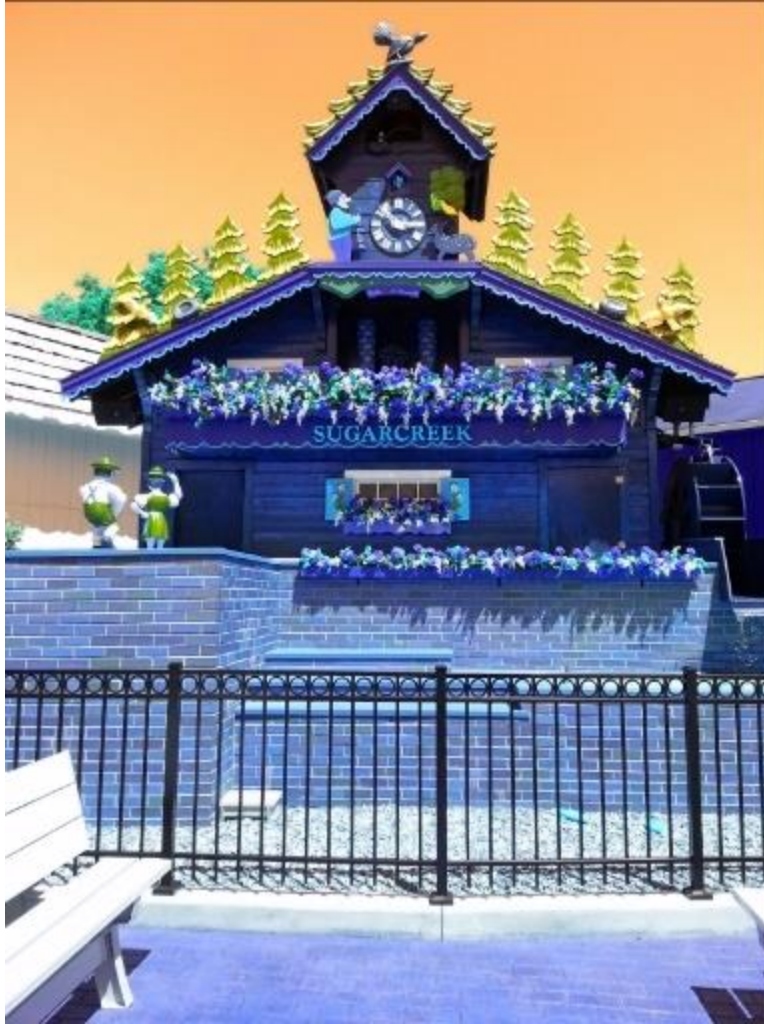


Figure 3: ps1-2-a-1.png

b. Green image channel



Figure 4: ps1-2-b-1.png

c. Red image channel



Figure 5: *ps1-2-c-1.png*

d. Questions

The green color channel image looks more like a monochrome image to me than the red channel image. The red image looks more washed out in the center of the picture. This indicates that the red pixels are close to saturation in these areas. The values for green, then, are not saturated. Since they vary more than the red values, the image looks more like a monochromatic image.

I would expect a computer vision algorithm to operate better on the green channel image better than the red channel image for the reasons above.

3. Replacement of pixels

a. Crop and replace



Figure 6: ps1-3-a-1.png

4. Arithmetic and Geometric operations

a. Min,max, mean, standard deviation

The table below shows the statistics computed from the green channel of image 1. These values were computed using the standard numpy functions. See the code excerpt below.

Minimum	0
Maximum	255
Mean	139.2
Standard Deviation	70.87

```
max1 = np.max(green1)
min1 = np.min(green1)
mean1 = np.mean(green1)
std1 = np.std(green1)
```

b. Image manipulation

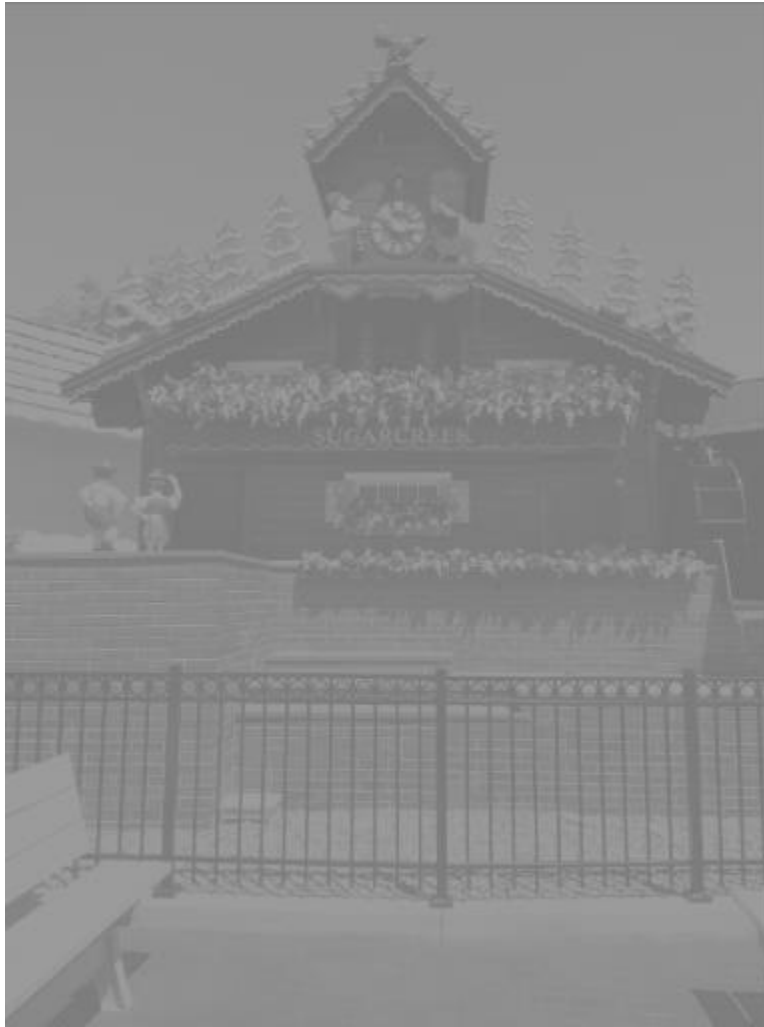


Figure 7: ps1-4-b-1.png

c. Shift image left 2 pixels



Figure 8: *ps1-4-c-1.png*

d. Image difference

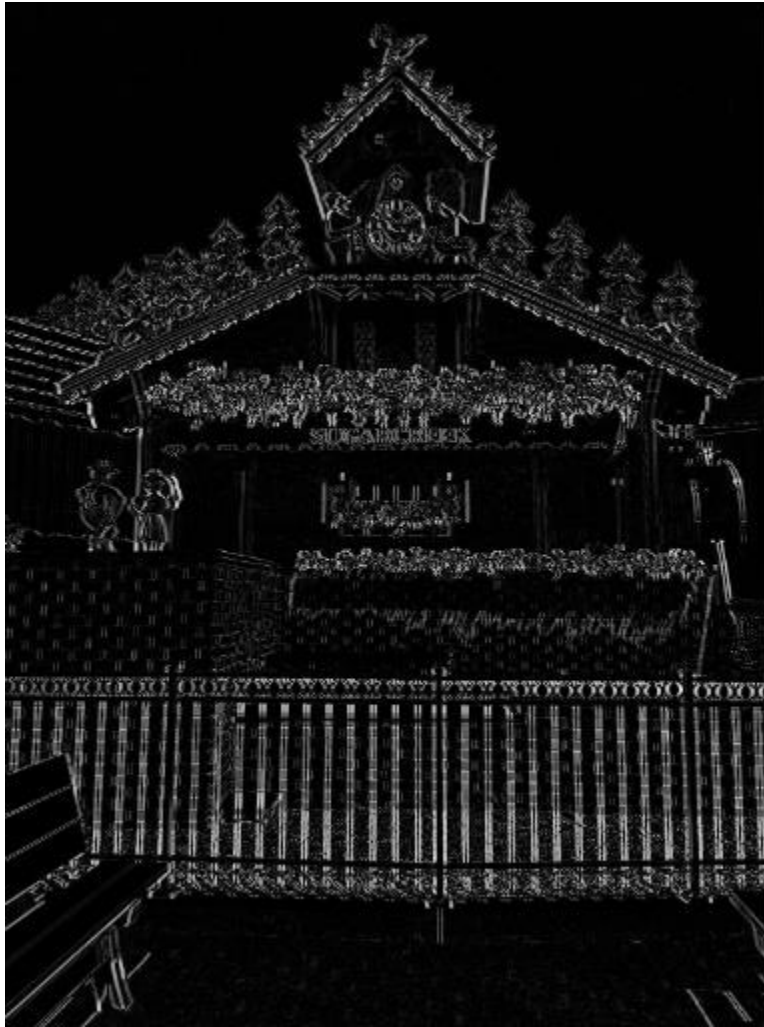


Figure 9: *ps1-4-d-1.png*

No major image format supports negative pixel values, so negative values are truncated to 0 (black).

5. Noise

- a. Add Gaussian Noise to green channel



Figure 10: ps1-5-a-1.png

b. Add noise to blue channel



Figure 11: ps1-5-b-1.png

c. Analysis

The blue image looks better, i.e., it seems that the noise affected it less. The reason is that the blue pixel values were already saturated, while the green pixel values were not. This was tested using a histogram filter. See the figures below. For the green channel, the values are fairly well spaced out, but for the blue channel, the values are heavily weighted to the upper limit of the pixel values. Thus, when noise was added to the blue pixel values, some of the noise was simply truncated. This was not the case for the green pixel values. The noise affected the pixel values in both directions, so it was more evident.

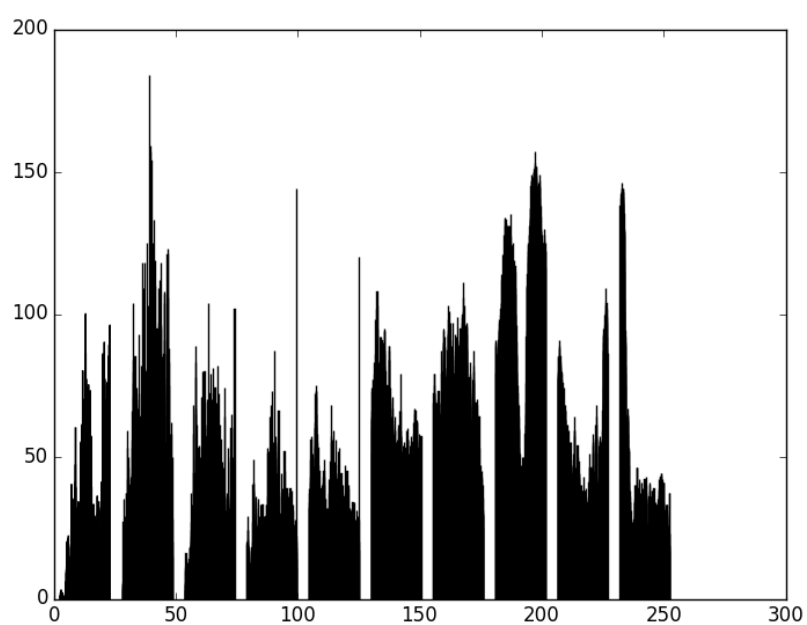


Figure 12: Histogram of green channel for image 1

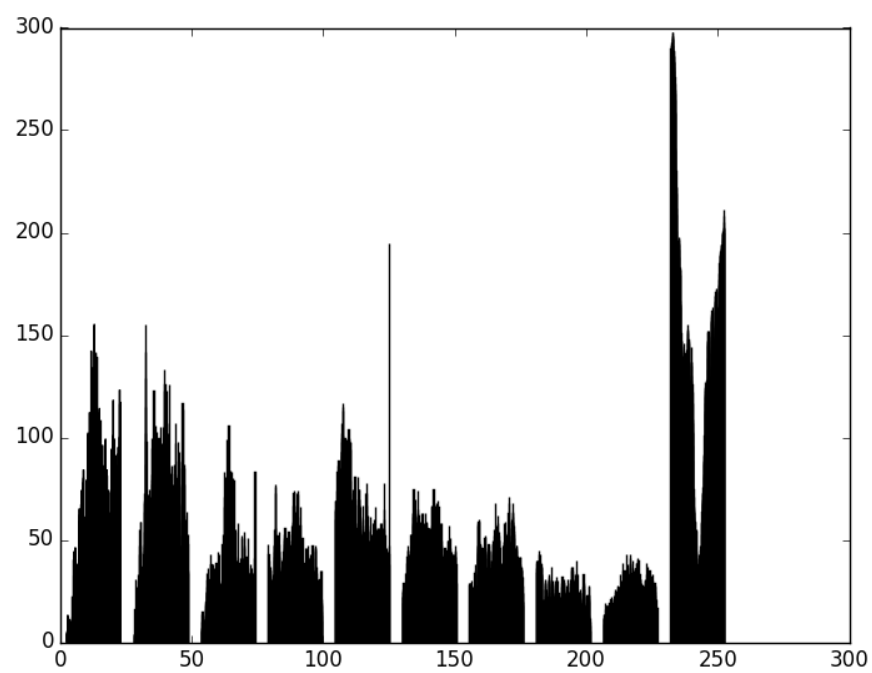


Figure 13: Histogram of blue channel for image 1