Altering Images with Python: Graphics Programming Assignment 00

Akarsh Kumar

Chosen Images for Altering

Switching Red and Green Channels

- Switch the red and green channels of image1 and save it to the output folder as **pic_1_a.png**.
- The red and green channels of image 1 were switched by saving all the red values of the image into a 3d array derived from the original 4d array. The green values of the image were saved into another array.
- The green channels of the image were replaced with data from the red array and the red channels were replaced with the green array.
- Code:

```
red = img1[:,:,2]
green = img1[:,:,1]
img1[:,:,1] = red
img1[:,:,2] = green
```

original

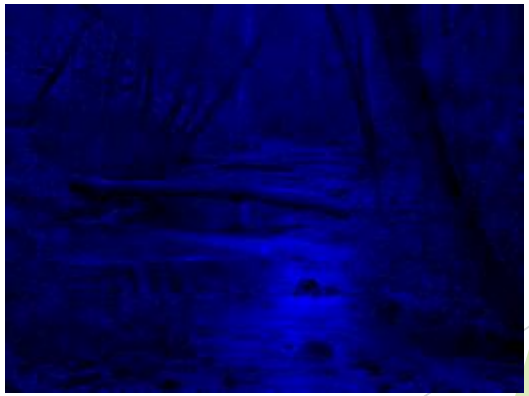
transformed

Monochromatic Blue Image

- The monochromatic blue image version of the original image was derived by setting the red and green channels of the original image to 0.
- Using simple python slicing syntax, this was done in two lines. Since cv2 uses BGR to save color values, the 1 and 2 index of the image pixels were set to zero.
- Code:

```
img2[:,:,1] = 0
img2[:,:,2] = 0
```

original



transformed

Inverting Intensity of Green

- The intensity of green was inverted in the image by first creating an array containing the inverted green values.
- The image's green values were then replaced with this new inverted data.
- Using python, this was done in three simple lines.
- Code:

```
green = img1[:,:,1]
green = 255-green
img1[:,:,1] = green
```

original

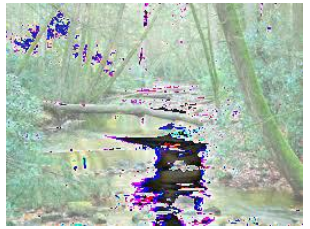
transformed

Brightening an Image

- 100 was added to all values of the pixels in the image. Every place where the value was greater than 255, it was clipped to 255. This method, however, did not work correctly because of how uint8 works.
- Code:

```
img2+=100
img2[img2>255] = 255
```

original



transformed

- Another method is available, however. Before adding 100, if you cap the pixels above 155 to 255, then the desired effect will be achieved.
- Code:

```
img2[img2>155] = 255
img2[img2<=155] +=100
```

transformed

Changing Center of Image

- To get the top left corner of the middle 100x100 pixel box of the image, the following code was used:

```
h,w = img1.shape[:2]
x = int(w/2-50)
y = int(h/2-50)
```

- Afterwards, the green channel of the central box was changed to the full 255 by simply assigning 255 to the green array data of the image:

```
img1[y:y+100,x:x+100,1] = 255
```

- The middle box of the other image was copied and pasted into the new image by replacing the data in the center of the image with data from the other image's central box:

```
img2[y2:y2+100,x2:x2+100] = img1[y1:y1+100,x1:x1+100]
```

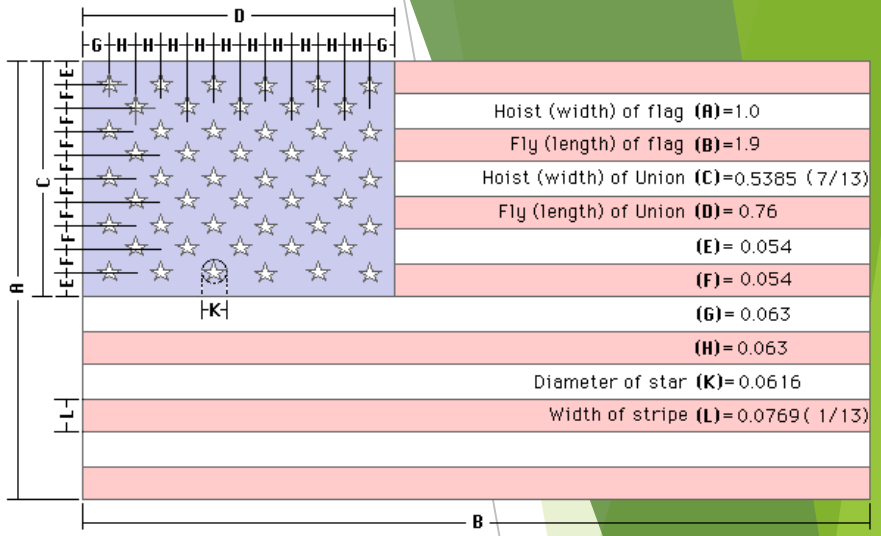
Statistics of Image 1

- The statistics of the number of pixels, as well as the min/max intensities, standard deviation of the intensities, and the average intensity was calculated using built in numpy functions on the image array.
- Code:

```
print("pixels: "+ str(math.size(img1ori)/3))
print("min intensity: "+ str(math.min(img1ori)))
print("max intensity: "+ str(math.max(img1ori)))
print("standard dev. of intensity: "+ str(math.std(img1ori)))
print("mean intensity: "+ str(math.mean(img1ori)))
```

- The resulting values were:
  - pixels: 120000.0
  - min intensity: 0
  - max intensity: 255
  - standard dev. of intensity: 66.3180418677
  - mean intensity: 210.039377778
- All of these values were expected given the diversity in color and the brightness of image 1. The standard deviation in this context means the average deviation from the 210 intensity mean of the color intensities.

Programming the US Flag



- By using the following dimensions online, I was able to recreate the US flag to near perfection.
- I found the color values of the US flag through a program online that let you see the color code at any given pixel.

Official US Flag

My version

Difference in the Flags

- The difference in the flags was calculated by subtracting off the original flag from my version and scaling those values from 0 to 255.
- By doing so, the following image was created.
- The near complete blackness of the image shows the accuracy of my flag's design and color. The stars on the top left are caused by a slight error in my star creation code that causes imperfect edges. Changing the size of the stars made no difference.