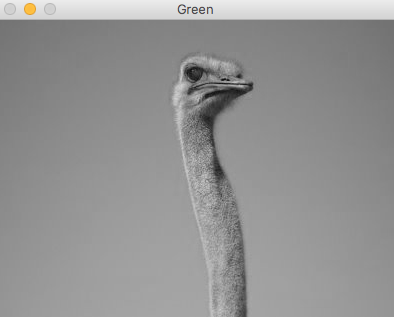
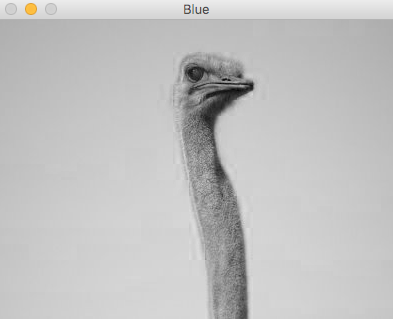
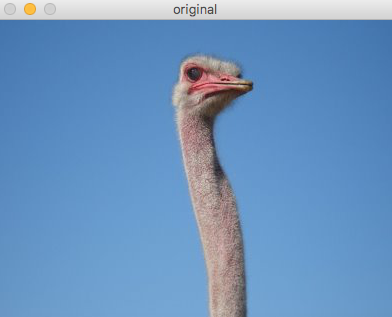
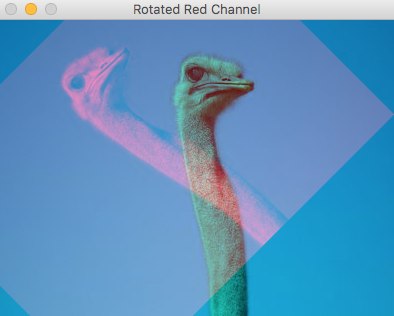
Peter Victoratos

CSC 411

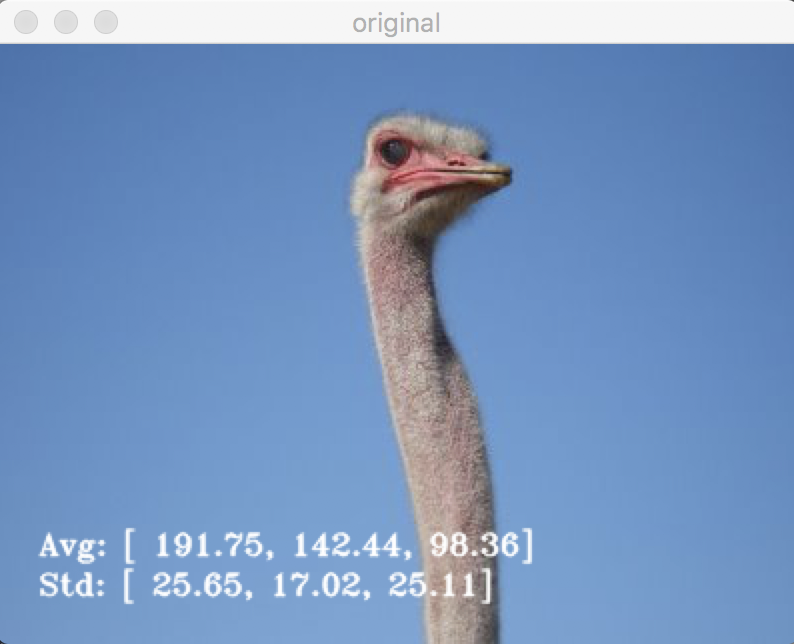
Lab03 Color

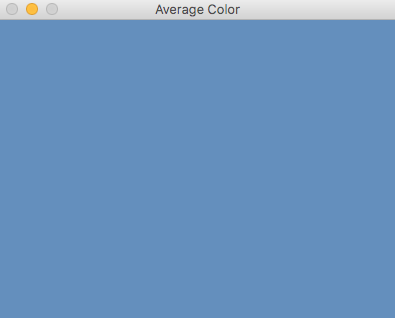
1. Separate Color Channels: My original image of an ostrich was split into three color channels, each becoming an image of its own. All of them looked pretty similar since these images only represent the value of each color to its respective channel. Before merging, I offset the red channel by 45 degrees which gave the ostrich a ghost of itself to hang around with.

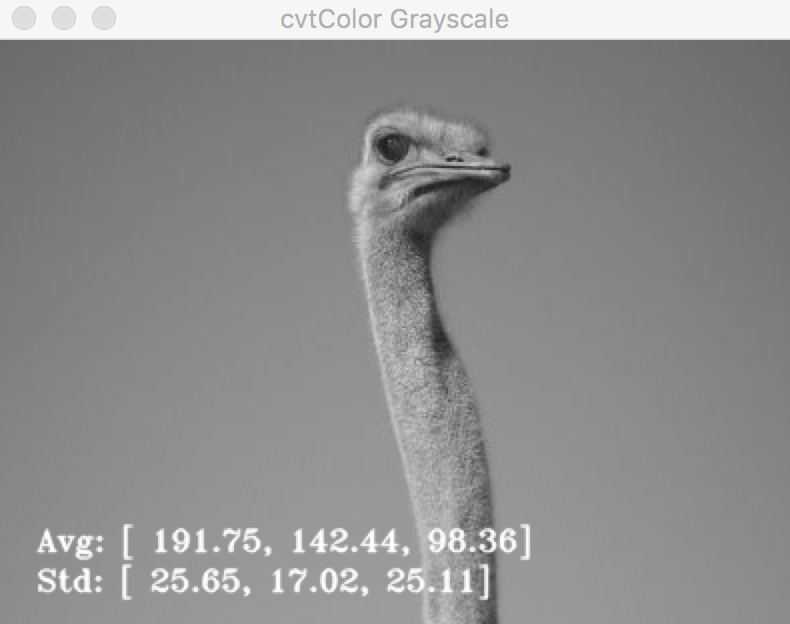




2. Average color: Using the same image as before, I was able to calculate the average value and standard deviation of each color channel. With this value I was able to create a new image of the average color of the original image. My new image was a pretty nice blue hue, very similar to the color of parts of the sky (which take up a majority of the image). I also added some text with the average hues and standard deviations on top of the original image.





3. Grayscale: This was the part of the lab where I ran into some issues. I was successfully able to call the cv2.COLOR\_BGR2GRAY statement which worked like a charm. Unfortunately beyond that I was not able to figure out how to verify if the image was in color or convert it with my own function. This is what I had in mind:

**def convert\_gray (img):**

**grayscale = img.astype(np.float)**

**if img.shape > [0, 0, 0]**

**return img**

**grayscale[cols, rows] = (b+g+r)/3**

**grayscale = grayscale.astype(np.uint8)**

**return grayscale**

**cv2.imshow("My First Grayscale", convert\_gray(img))**