

# CS 484, Fall 2019

## Take-home Quiz 1

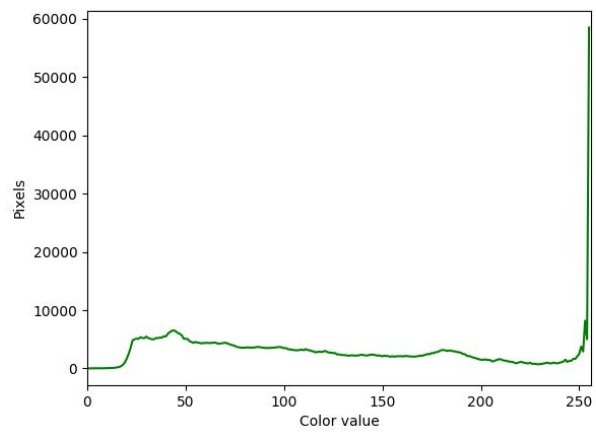
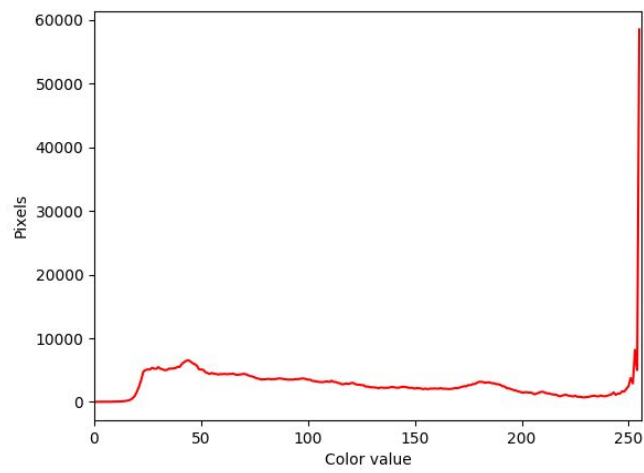
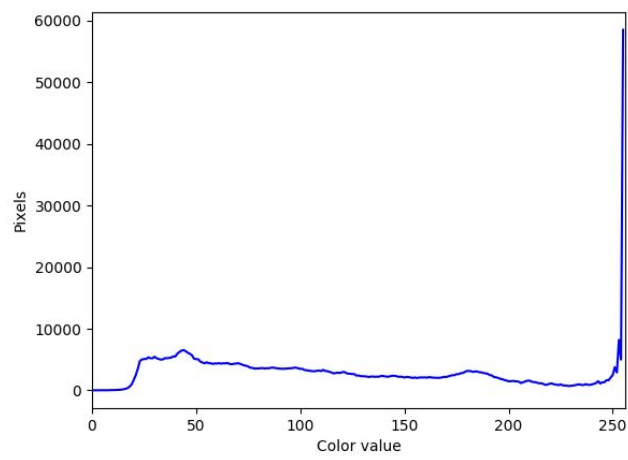
Zeynep Nur Öztürk  
21501472

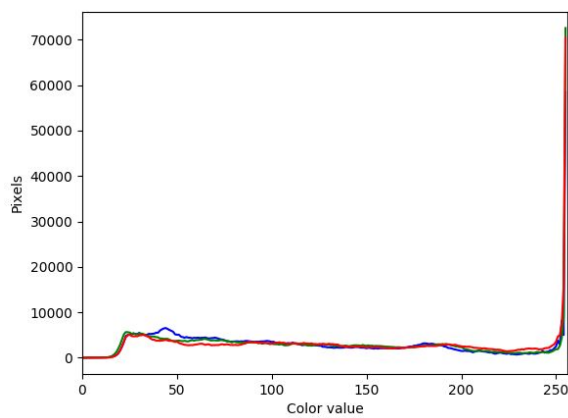
### Personal Picture:



### For Color Histograms:

Because I Have Ubuntu system in my computer I decided to use linux terminal and text editor to do this steps. I used Python and Matplotlib to obtain a good result. First I resize my picture as 1000x1000 pixel. Then based on BGR coloring I obtain three color histogram. And I used Opencv's calcHist function[1,2]. The results are given below:





Based on a reason that I don't understand the color value was parallel to each other.

## Detecting the Face and Skins:

First of all, I begin with give a range to obtain the skin color[3]. Then I convert our image to HSV (hue, saturation, value), then make a bitwise and operation to get the specific color that I want like below:

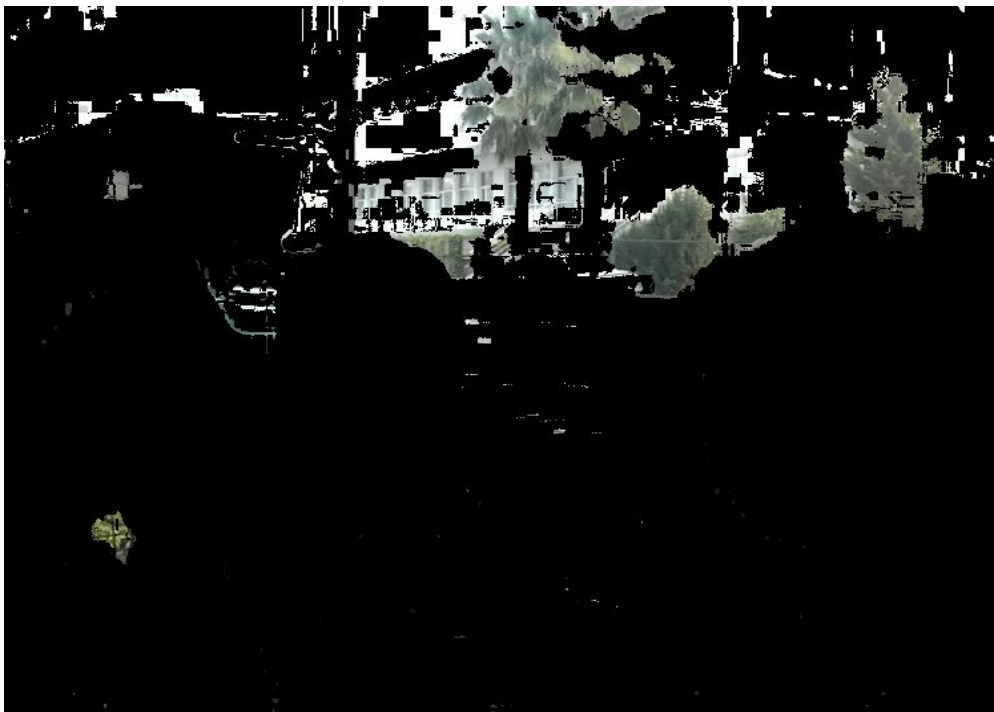


Then, I convert this to gray scale, and make it binary image as below:



### **Detecting the Trees:**

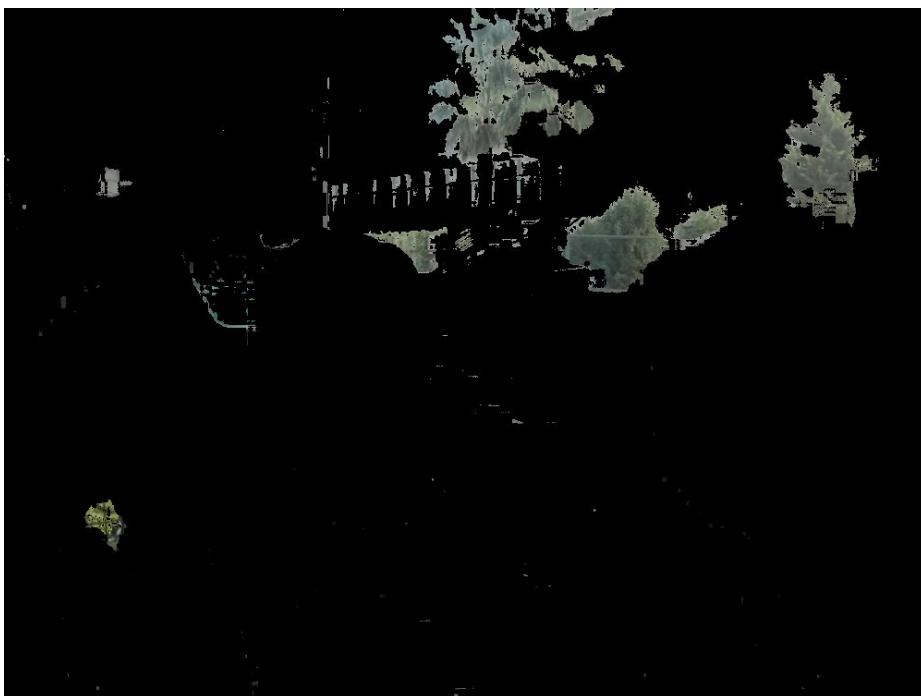
First of all, I begin with give a range to obtain the tree color. Than I convert our image to HSV (hue, saturation, value), then make a bitwise and operation to get the specific color that I want like below:



Then, I want to delete some of the white pixels. To do this, I arrange a white color range and imply it to the beginning HSV picture as below:



After bitwise operation, I subtract it from the above image and this result came out:



Then revert it to a binary image:



Even though I subtract the white area some of the windows of the building stayed in the picture.

### **Detecting the Building:**

To get the building image I used the same steps above like arrange color range, bitwise and operation, etc. Then I realized the white range get some parts of the skin. Therefore, I subtract skin image from the white color ranged image.







Then I tried to delete the road and the car in front of the building by subtracting the image.



**Short Discussion:** Provide a short discussion of the results (e.g., which objects were easy and which were more difficult, what was possible and what was not, etc.):

First of all, I am more convenient with python. It requires more code than matlab in any way but it was easy because I knew it.

Then for histograms, it was the easiest part, because example codes were available in the internet as I mentioned in references.

To make the second part, I spend so much time while finding the correct range for color. It was hard to obtain that color. Therefore, I spend many times until I found it. After finding the color it was easy to work with it. Because of we only looking at the color it was hard to distinguish between same colored object. For example I couldn't erase the building's windows from the trees.

## References for code:

- [1] "Resize",  
[https://medium.com/@manivannan\\_data/resize-image-using-opencv-python-d2cdbbc480f0](https://medium.com/@manivannan_data/resize-image-using-opencv-python-d2cdbbc480f0)
- [2] "Histogram Calculation",  
<https://www.pyimagesearch.com/2014/01/22/clever-girl-a-guide-to-utilizing-color-histograms-for-computer-vision-and-image-search-engines/>
- [3] "Skin Detection" <https://nalinc.github.io/blog/2018/skin-detection-python-opencv/>
- [4] "Convert Gray scale"  
<https://stackoverflow.com/questions/12201577/how-can-i-convert-an-rgb-image-into-grayscale-in-python>
- [5] "Detect Green Objects"  
<https://stackoverflow.com/questions/47483951/how-to-define-a-threshold-value-to-detect-only-green-colour-objects-in-an-image/47483966#47483966>
- [6] "Subtracting images"  
<https://stackoverflow.com/questions/21425992/how-to-subtract-two-images-using-python-opencv2-to-get-the-foreground-object>