CS 4476/6476 Spring 2020 PS1

Tarun Pasumarthi
tpasumarthi3@gatech.edu
903085537

Short answer problems

1.1 Use `numpy.random.rand` to return the roll of a six-sided die over N trials.

```
def prob_1_1(N):
    """
    Args: N: the number of trials.
    Returns: arr: array of rolls.
    """

    ### START CODE HERE ###
    arr=np.random.rand(N)
    arr=np.ceil(arr*6)
    ### END CODE HERE ###

    return arr
```

1.2 Let `y` be the vector: `y = np.array([11, 22, 33, 44, 55, 66])`. Use the reshape command to form a new matrix `z` that looks like this:

`[[11,22],[33,44],[55,66]]`

```
def prob_1_2(y):
    """
    Args: y: numpy array.
    Returns: z: numpy array of shape
    (new_size,2).
    """

    ### START CODE HERE ###
    rows=int(y.shape[0]/2)
    z=y.reshape((rows, 2))
    ### END CODE HERE ###

    return z
```

1.3 Use the `numpy.max` and `numpy.where` functions to set `x` to the maximum value that occurs in `z` (above), and set `r` to the row number (0-indexed) it occurs in and `c` to the column number (0-indexed) it occurs in.

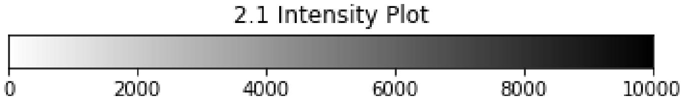
```
def prob_1_3(z):
    """
    Args: z: numpy array of shape (3,2).
    Returns: x: max value in z.
    r: row index of x.
    c: column index of x.
    """

    ### START CODE HERE ###
    x=np.max(z)
    g=np.where(z == x)
    r,c=g
    r=r[0]
    c=c[0]
    ### END CODE HERE ###

    return (x, r, c)
```

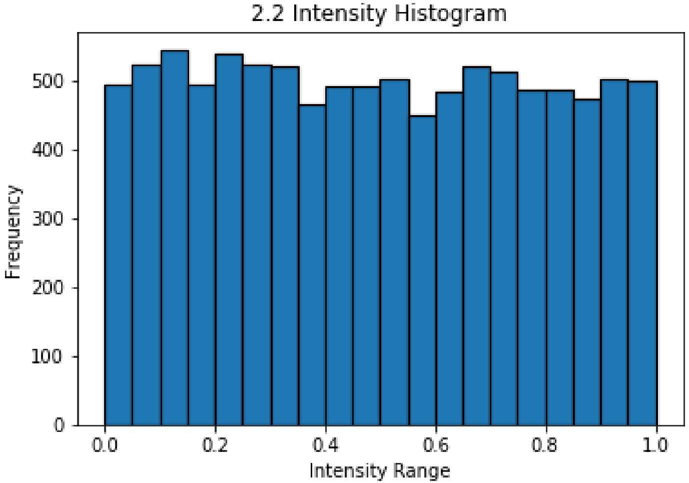
1.4 Let `v` be the vector: `v = np.array([1, 4, 7, 1, 2, 6, 8, 1, 9])`. Set a new variable `x` to be the number of 1's in the vector `v`.

```
def prob_1_4(v):
    """
    Args: v: numpy array.
    Returns: x: number of 1's in v.
    """

    ### START CODE HERE ###
    x=list(v).count(1)
    ### END CODE HERE ###

    return x
```

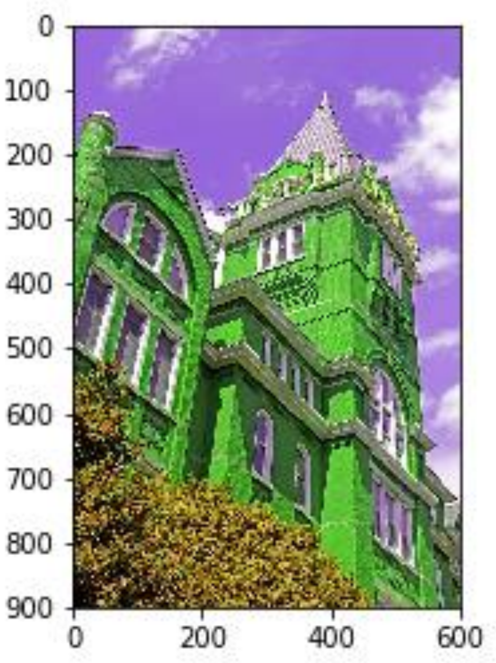
2.1 Plot all the intensities in `A`, sorted in decreasing value. Provide the plot in your answer sheet. (Note, in this case we don't care about the 2D structure of `A`, we only want to sort the list of all intensities.)

2.2 Display a histogram of A's intensities with 20 bins. Again, we do not care about the 2D structure. Provide the histogram in your answer sheet.

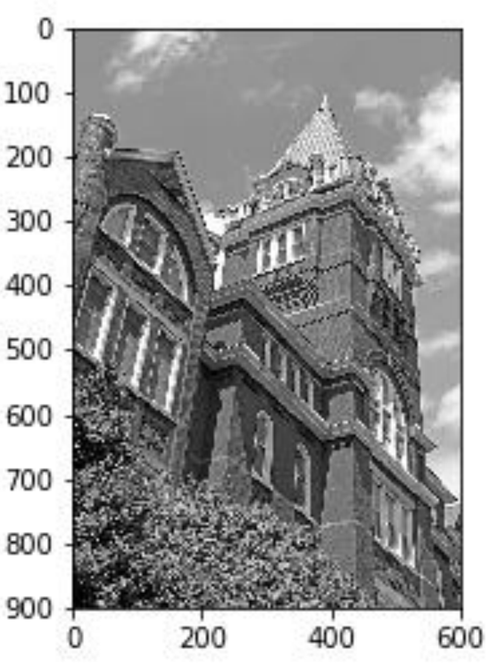




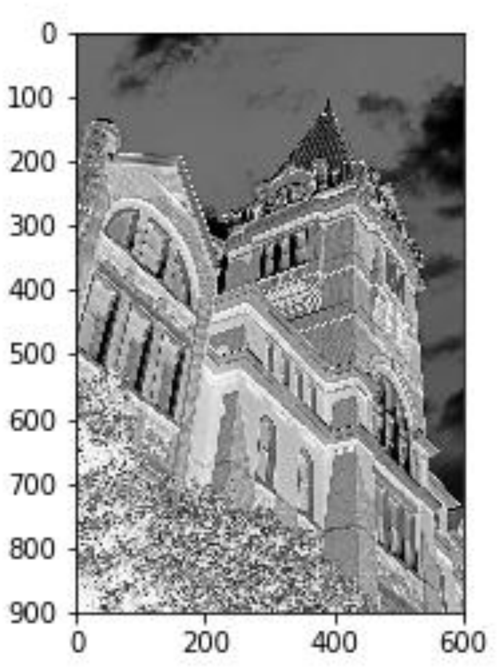
3.1 Display the color channel swapped image.



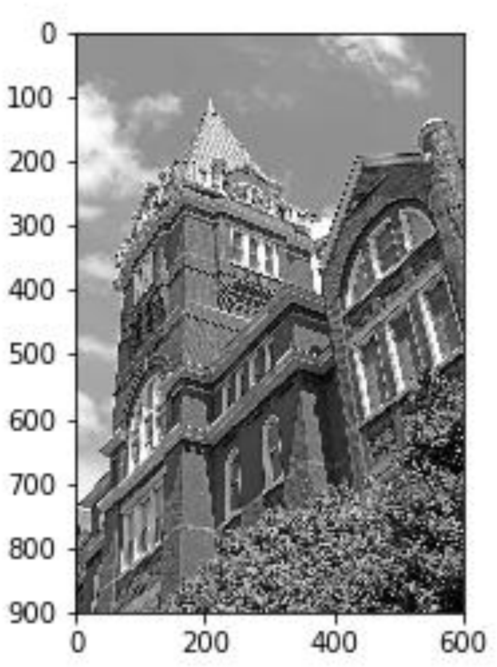
3.2. Display the grayscale image.



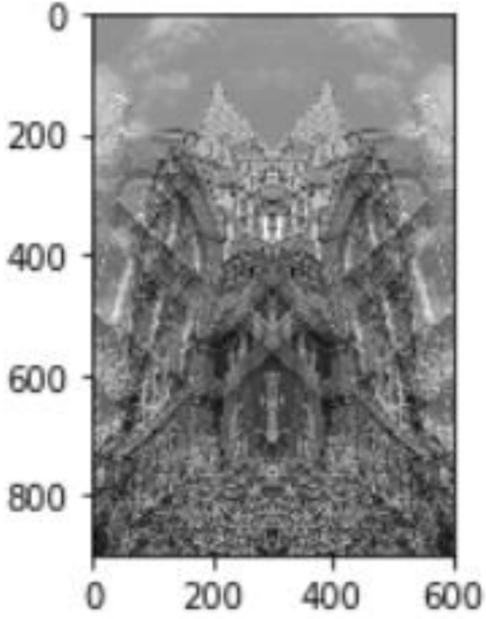
3.3 Display the negative image.



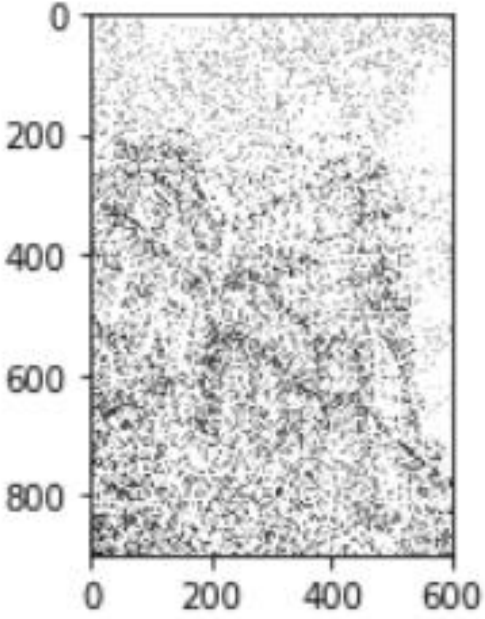
3.4 Display the mirror image.



3.5 Display the averaged image.

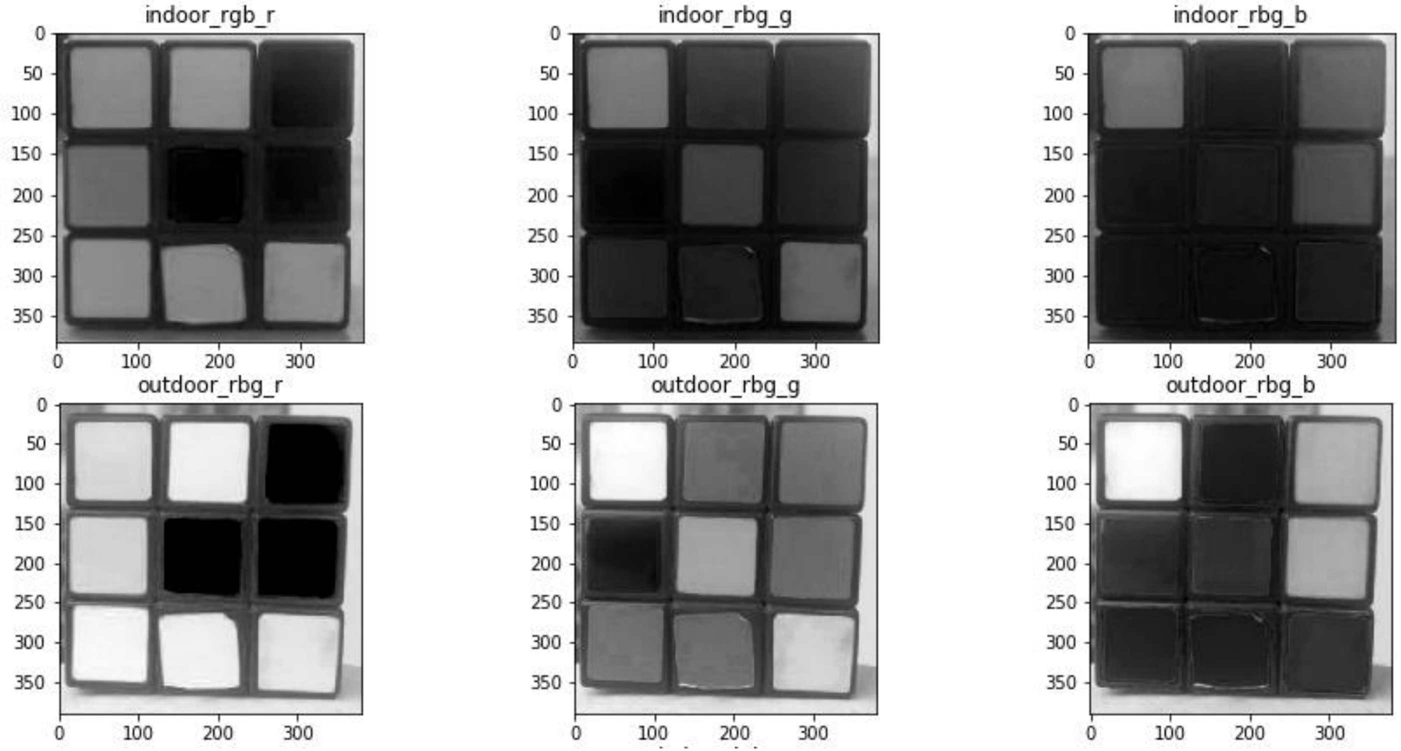


3.6. Display the clipped image.

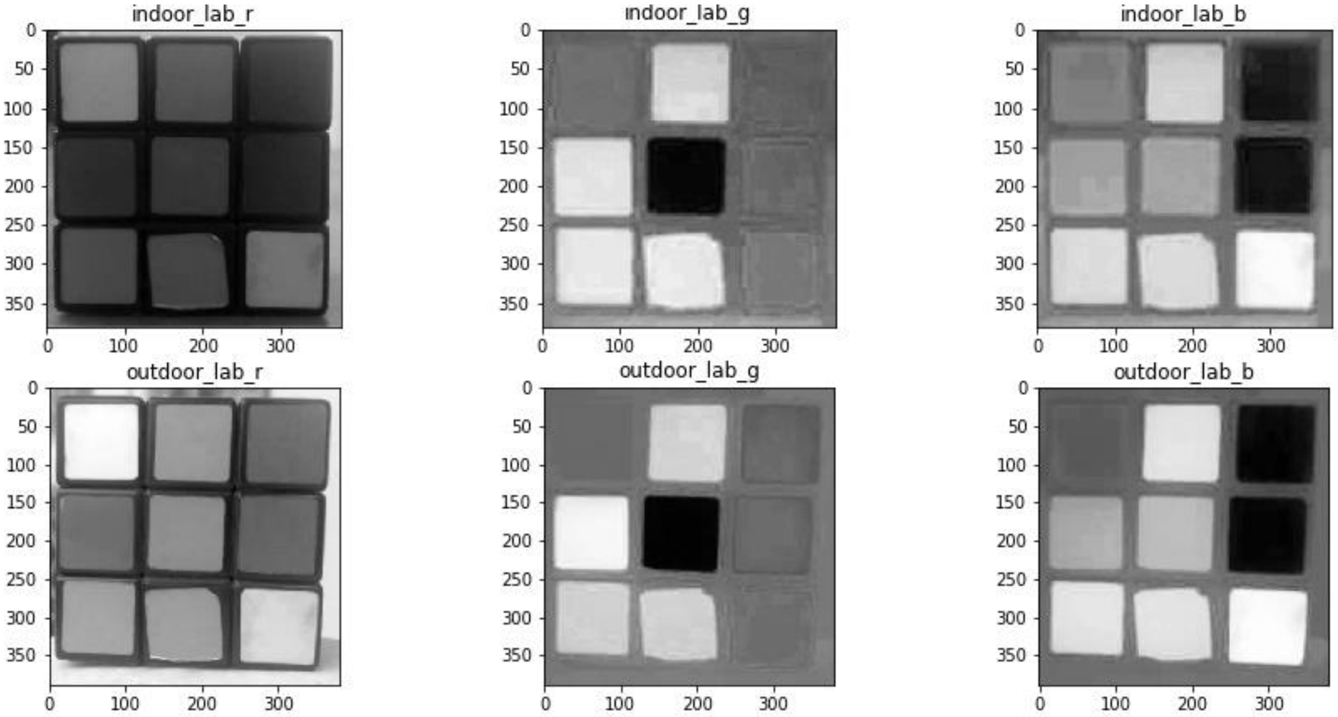


Understanding Color

4.1. Load the images and plot their R, G, B channels separately as grayscale images using `plt.imshow()` (beware of normalization).



4.1.(contd) Then convert them into LAB color space using `cv2.cvtColor()` and plot the three channels again.



4.2. How do you know the illuminance change is better separated in LAB color space?

Illuminance is better separated in the LAB color space since the brightness value is solely represented by the value of L and the color values are represented by A and B. In the RGB space, the brightness and the color values are intertwined with the values of R,G, and B. This is evident from our previous two slides as the r, g, b values have different intensities between their respective indoor vs outdoor pictures. Compared to that, the l, a, b values have the same intensities for a and b since they represent the color of the object without the influence of brightness, and only the l values are different since the two pictures are of the same object illuminated differently.

4.3. Convert the input image from RGB to HSV.