

# CS 4476 Spring 2022

## PS1

Devanshi Gupta  
dgupta94@gatech.edu  
903536448

**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.  
    """  
  
    arr = numpy.random.randint(1, 7, N)  
    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).  
    """  
  
    z = y.reshape(3,2)  
  
    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.

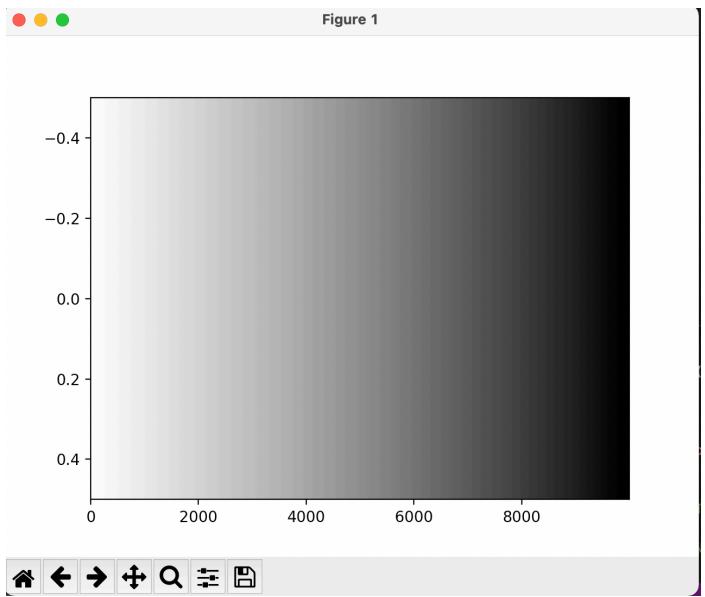
```
def prob_1_3(z):
    """
    Args: z: numpy array of shape (3,2).
    Returns: x: max value in z.
             r: row index of x.
             c: column index of x.
    """
    x = numpy.max(z)
    r = numpy.where(z == x)[0][0]
    c = numpy.where(z == x)[1][0]

    return (x, r, c)
```

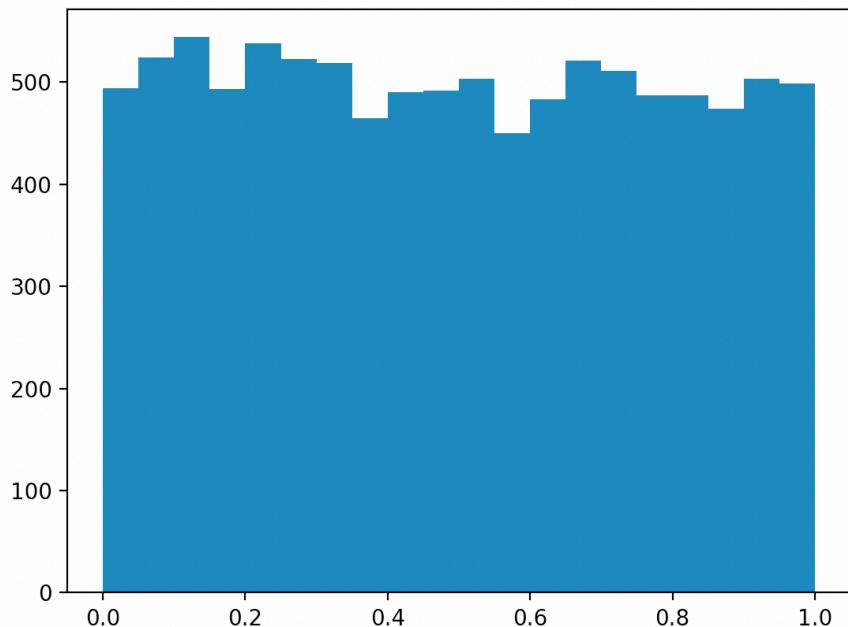
**1.4** Let `v` be the vector: `v = np.array([1, 4, 7, 1, 2, 6, 8, 1, 9])`. Set a new variable `x` to be the number of 1's in the vector `v`.

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

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



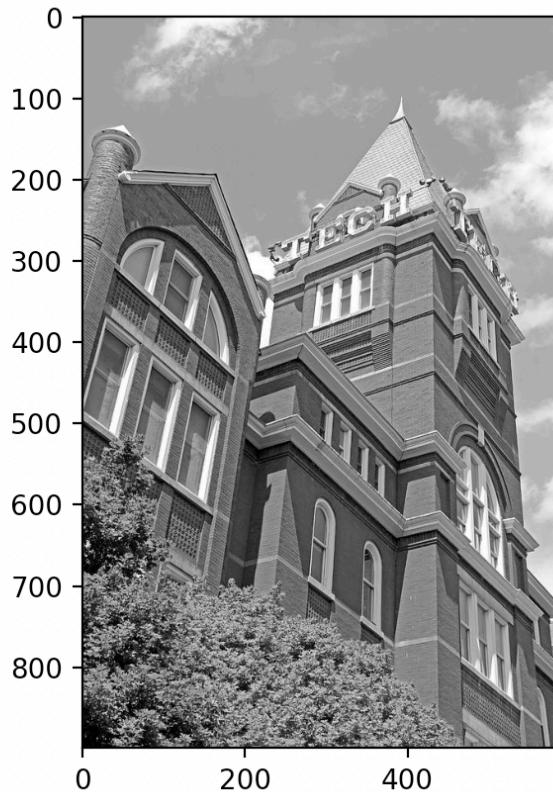
**2.2** Display a histogram of  $\mathbb{A}$ 's intensities with 20 bins. Again, we do not care about the 2D structure. Provide the histogram in your answer sheet.



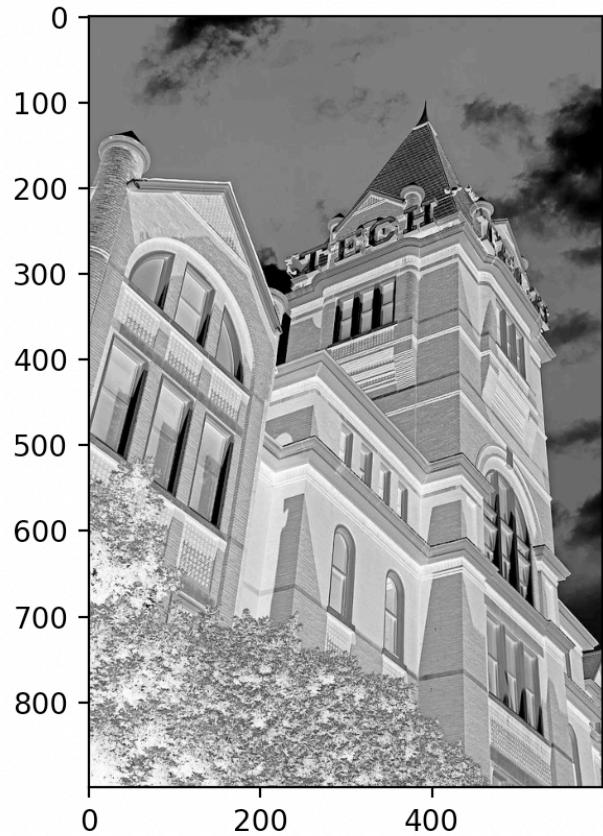
**3.1** Display the color channel swapped image.



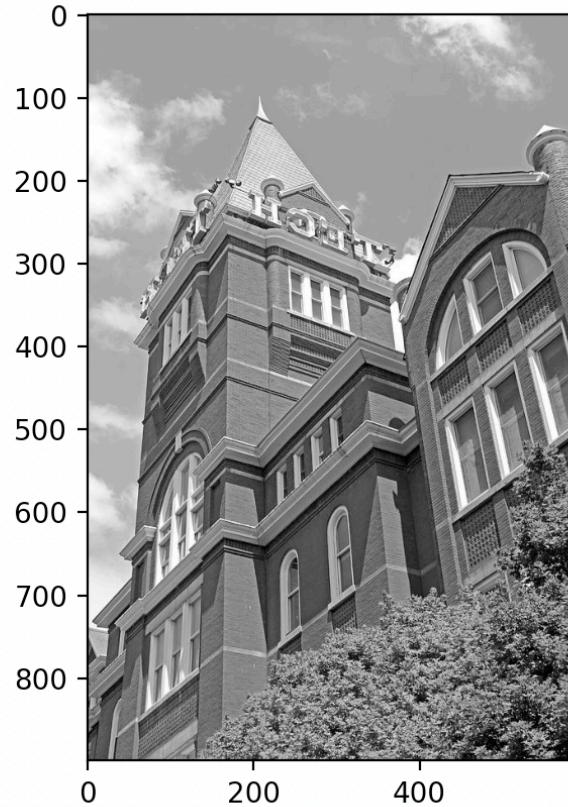
**3.2.** Display the grayscale image.



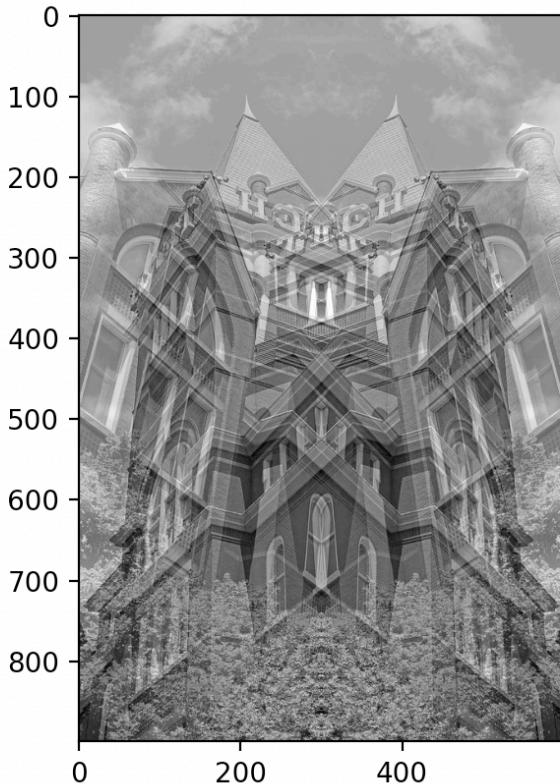
**3.3** Display the negative image.



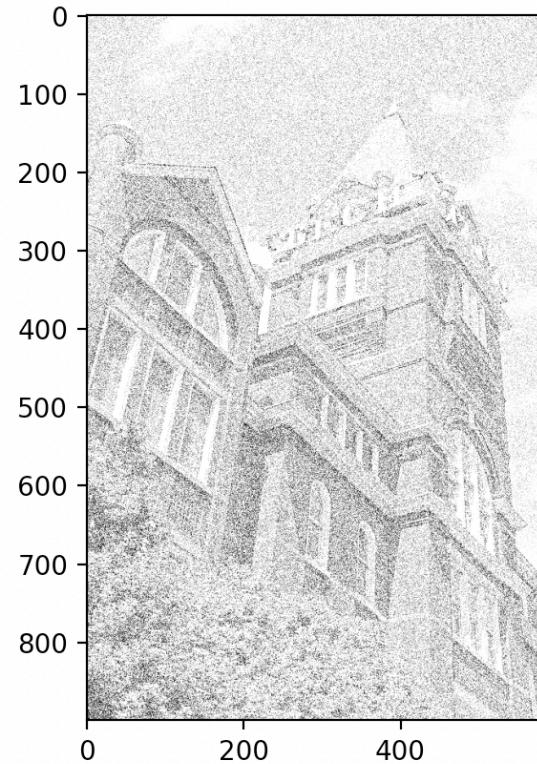
**3.4** Display the mirror image.



**3.5** Display the averaged image.



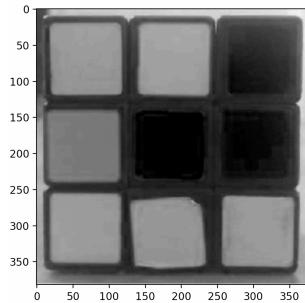
**3.6.** Display the clipped image.



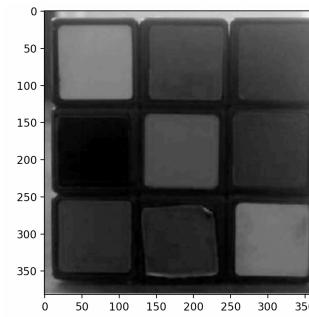
**4.1.** Load the images and plot their R, G, B channels separately as grayscale images using `plt.imshow()`

Indoor

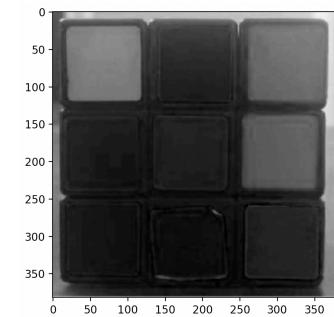
Red



Green

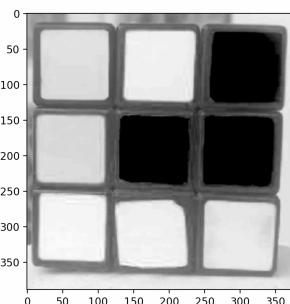


Blue

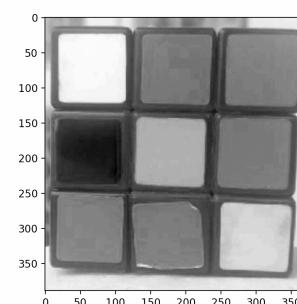


Outdoor

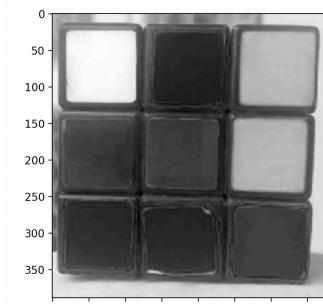
Red



Green



Blue



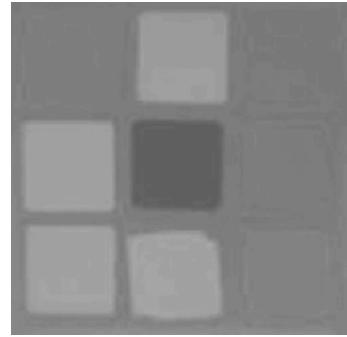
**4.1.(contd)** Then convert them into LAB color space and plot the three channels again.

Indoor

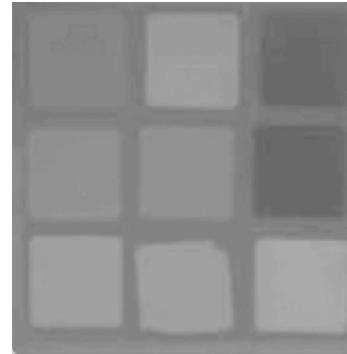
L



A



B

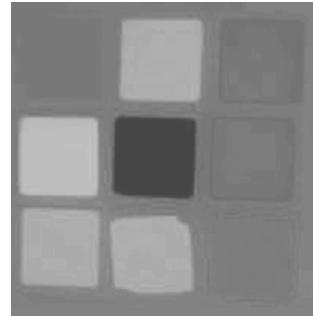


Outdoor

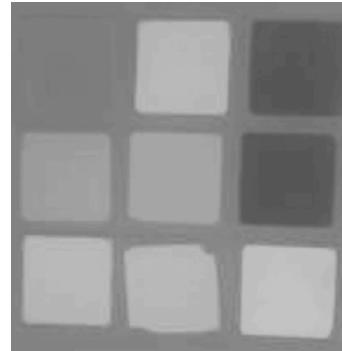
L



A



B



**4.2.** Convert the input image from RGB to HSV.

