IP - Experiment No. 1

- Name: Sachi Shah
- Roll No.: C094
- Batch: EB1
- Sap Id: 70321018081

Aim: To study the basic Python commands used in image processing to read write and display image

To Read, show using Skimage and matplotlib for gray scale image

```
from skimage import io
I = io.imread('cameraman.tif')
I #array of numbers
```

```
array([[156, 157, 160, ..., 152, 152, 152],
       [156, 157, 159, ..., 152, 152, 152],
       [158, 157, 156, ..., 152, 152, 152],
       ...,
       [121, 123, 126, ..., 121, 113, 111],
       [121, 123, 126, ..., 121, 113, 111],
       [121, 123, 126, ..., 121, 113, 111]], dtype=uint8)
```

```
#Size of image
I.shape
```

```
(512, 512)
```

```
#to check rows & columns seperately
rows = I.shape[0]
columns = I.shape[1]
print("Rows= ",rows)
print("Coulumns = ",columns)
```

```
Rows=  512
Coulumns =  512
```

Check Pixel Value

```
#to check/display pixel value
I[100,100] = 173 #to change the value of pixel to 173
I[100,100]       #to see the value of pixel located at 100,100
```

```
173
```

```
#to view the image
import matplotlib.pyplot as plt
io.imshow(I)
plt.axis('off')    #remove the x and y axis
plt.show()         #remove additional comments
```

```
#save array of numbers(I) as image
io.imsave('newI.jpg',I)   #Method 1
import cv2
cv2.imwrite('newimage.png',I)  #Method 2
```

```
True
```

To Read, show using Scikit image and matplotlib for color image

```
#Read image
IC = io.imread('cat.png')
IC #array of numbers
```

```
array([[[188, 201, 184],
        [221, 234, 216],
        [212, 226, 203],
        ...,
        [172, 191, 172],
        [177, 189, 175],
        [139, 156, 146]],

       [[185, 198, 181],
        [224, 237, 219],
        [211, 225, 202],
        ...,
        [168, 185, 167],
        [178, 189, 175],
        [132, 148, 138]],

       [[185, 198, 181],
        [220, 233, 215],
        [215, 229, 206],
        ...,
        [163, 176, 158],
        [173, 179, 167],
        [127, 140, 131]],

       ...,

       [[158, 145, 128],
        [155, 142, 125],
        [151, 138, 121],
        ...,
        [ 87,  68,  61],
        [101,  82,  75],
        [ 68,  59,  60]],

       [[156, 141, 122],
        [157, 142, 123],
        [151, 136, 117],
        ...,
        [ 91,  72,  65],
        [ 97,  78,  71],
        [ 67,  58,  59]],

       [[163, 148, 129],
        [166, 151, 132],
        [162, 147, 128],
        ...,
        [ 87,  68,  61],
        [ 97,  78,  71],
        [ 67,  58,  59]]], dtype=uint8)
```

```
#Size of image
IC.shape
#(rows,columns,planes)
```

```
(733, 490, 3)
```

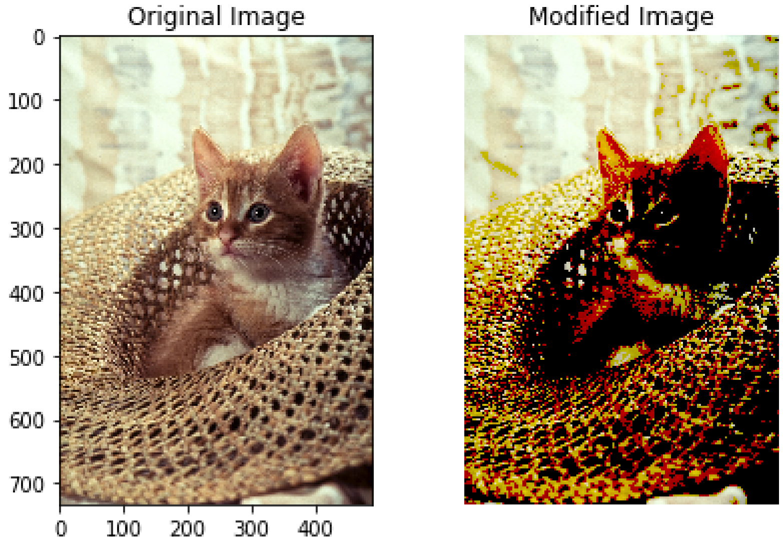
```
#To check the value of a pixel
IC[1,0]
```

```
array([185, 198, 181], dtype=uint8)
```

Change image to darker color

```
Inew = IC.copy() #copy the array
Inew[IC<150]=0 #Changing all the pixels less than 150 to 0
#Option: Inew[IC<150]=Inew[IC<150]-50    #Changing all the pixels less than 150

plt.subplot(1,2,1)  #rows,columns,positions
plt.title("Original Image")
io.imshow(IC)

plt.subplot(1,2,2)
plt.title("Modified Image")
```

```
io.imshow(Inew)

plt.axis('off')     #remove the x and y axis
plt.show()          #remove additional comments
```



To convert a color (RGB) image to Grayscale

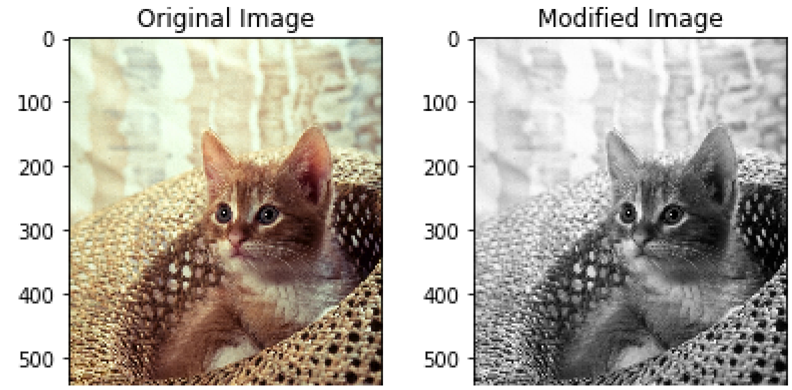
This example converts an image with RGB channels into an image with a single grayscale channel.

Y = 0.2125 R + 0.7154 G + 0.0721

These weights are used by CRT phosphors as they better represent human perception of red, green and blue than equal weights

```
from skimage import io
from skimage.color import rgb2gray
import matplotlib.pyplot as plt

img  = io.imread('cat.png')

#convet color to grayscale
I  = rgb2gray(img)

plt.subplot(1,2,1)  #rows,columns,positions
plt.title("Original Image")
io.imshow(img)

plt.subplot(1,2,2)
plt.title("Modified Image")
io.imshow(I)
```

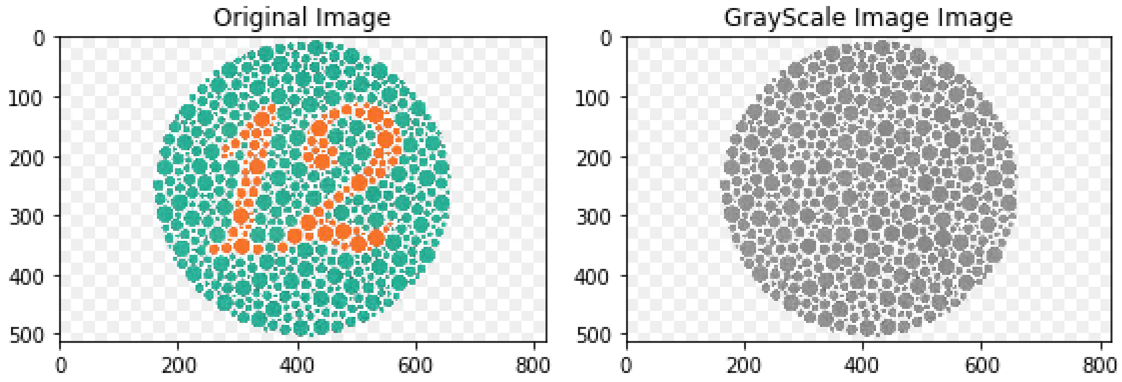
```
<matplotlib.image.AxesImage at 0x7f7cbda793d0>
```



While converting a color to gray image you can loose some information Ishihara plates are used to check color blindness

```
from skimage import io
from skimage.color import rgb2gray
import matplotlib.pyplot as plt

img2  = io.imread('ishihara.jpg')
#convert color to grayscale
I2  = rgb2gray(img2)

plt.figure(figsize=(8,8))

plt.subplot(1,2,1)
io.imshow(img2)
plt.title('Original Image')

plt.subplot(1,2,2)
io.imshow(I2)
plt.title('GrayScale Image Image')

```

```
Text(0.5, 1.0, 'GrayScale Image Image')
```



To convert image formats

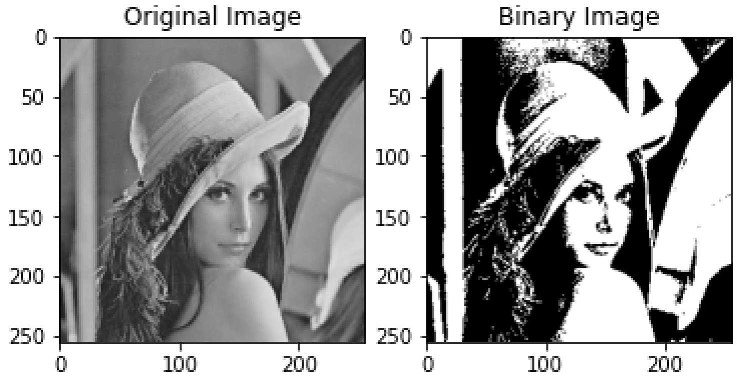
```
from skimage.color.colorconv import rgb2gray
img = io.imread('ishihara.jpg')
```

```python
img_gray = rgb2gray(img)
```

Converting an gray scale image to a binary image (in exp 2)

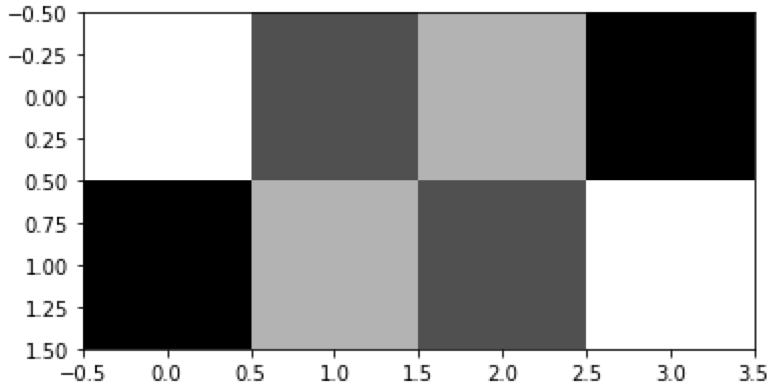
```python

```

```
Text(0.5, 1.0, 'Binary Image')
```



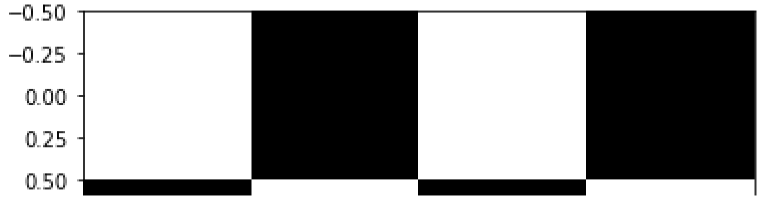
To observe the effect of different gray level images on their corresponding image matrix

```python
data=[[255,80,180,0],[0,180,80,255]]
plt.imshow(data,cmap='gray')

data2 = [[0,1,0,1],[1,0,1,0]]

# Binary image: 0=black, 1=white
# 8-bit image: 255=white, 0=white
# So, if only 0 and 1 is there, it is binary image
# but if we write 1 in 'data' it will be considered as black as it is a 8 bit image.
```



```python
data = [[1,0,1,0],[0,1,0,1]]
#plt.imshow(data) # shows color, plt uses 24 bit data
plt.imshow(data,cmap='gray')
```

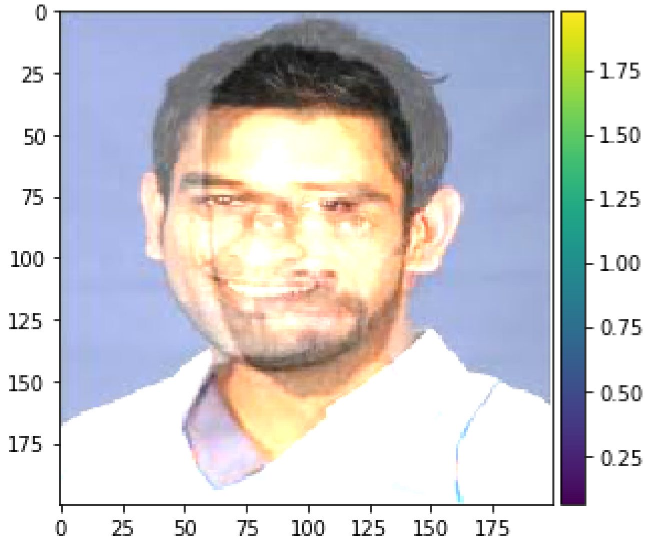
```
<matplotlib.image.AxesImage at 0x7f7cbdaafb90>
```



Activity 2 : Image Blending

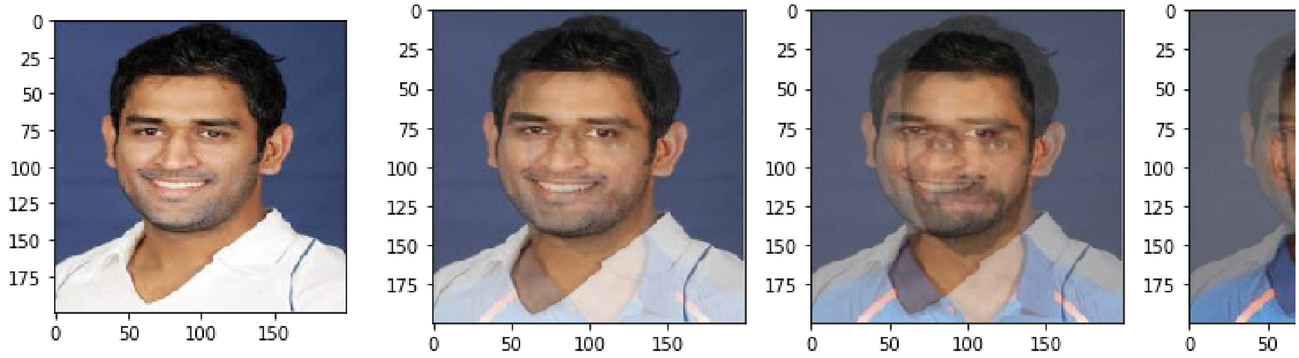
```
from skimage.transform import resize
I1=io.imread("ViratKohli.jpg")/255
I2=io.imread("dhoni.jpg")/255
#IB=I1+I2 #Error because image size is not same
#Checking size and making sizes of both the image same
I2new = resize(I2,(I1.shape[0],I1.shape[1],3))

IB=I1+I2new
io.imshow(IB)
```

```
/usr/local/lib/python3.7/dist-packages/skimage/io/_plugins/matplotlib_plugin.py:150:
  lo, hi, cmap = _get_display_range(image)
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
<matplotlib.image.AxesImage at 0x7f4d39cfc210>
```



```
from skimage.transform import resize
import numpy as np

I1=io.imread("ViratKohli.jpg")/255
I2=io.imread("dhoni.jpg")/255

#Checking size and making sizes of both the image same
I2new = resize(I2,(I1.shape[0],I1.shape[1],3))


i=1
plt.figure(figsize=(15,15))
for alpha in np.linspace(0,1,5):
    IB=(alpha*I1)+((1-alpha)*I2new)
    plt.subplot(1,5,i)
    io.imshow(IB)
    i=i+1

plt.show()
plt.tight_layout()
```

```
/usr/local/lib/python3.7/dist-packages/skimage/io/_plugins/matplotlib_plugin.py:150:
  lo, hi, cmap = _get_display_range(image)
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or
```



```
<Figure size 432x288 with 0 Axes>
```

Conclusion :

1. We implemented code to read and view gray scale and colour image
2. We implemented modification of images such as darkening and image blending
3. We converted gcolor image into gray scale
4. We converted image format
5. We plotted array