

## **Experiment-2**

### **Implementation of Colour Space Conversions**

**Name : N U Praneeth Reddy**

**Reg.No: 21BAI1500**

**Aim:** To create programs in Python to complete the following image processing tasks.

**Resources Used:** Anaconda Python Environment,  
Google Collab Jupyter Notebook

#### **Theory :**

OpenCV stands as an open-source library designed for computer vision and machine learning applications. Its primary goal is to offer a unified foundation for computer vision projects and to facilitate the integration of machine perception into various commercial products.

On the other hand, NumPy serves as a Python library, enabling support for large, multi-dimensional arrays and matrices, accompanied by an extensive array of high-level mathematical functions for manipulating these arrays.

Additionally, Matplotlib functions as a Python plotting library, directly connected to the numerical mathematics capabilities of NumPy. It delivers an object-oriented API for seamlessly embedding plots within applications.

#### **Tasks :**

1) Import libraries and color image. Perform some common point operations

a) Image Negative

b) Log-Transform

c) Gamma Correction.

Plot the responses and provide your inferences.

2) Perform an operation on the image such that gray values  $<$  threshold is zero and  $>$  threshold is 255 (Image Binarization).

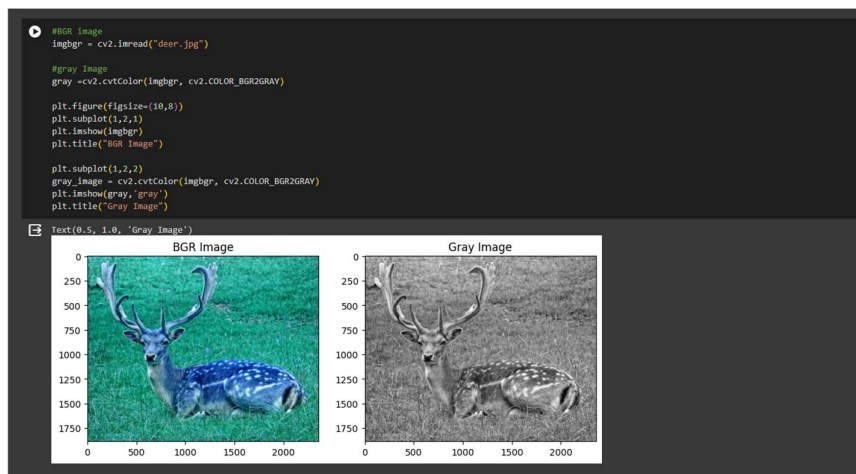
3) Enhance the input gray image using contrast stretching method. Observe the output and also try to plot the histogram of the input and the processed image.

## Procedure :

- Open Google Colab and create a new Jupyter Notebook.
- Import important libraries namely OpenCV, Numpy and Matplotlib

```
[1] #import Libraries
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

- Read the image using imread in the OpenCV library in BGR (Blue-Green-Red) format and convert it to gray scale and print both the images side by side using subplot.



## Task-1:

- a)** Convert the gray image to negative too by subtracting all the values from 255.



**b)** Log Transform for BGR image along with Log Transform for gray image.

1.b)Log transform

```
# BGR log Transform
a1 = imgbgr/255
log_trans_bgr = 1* np.log(1 + a1)

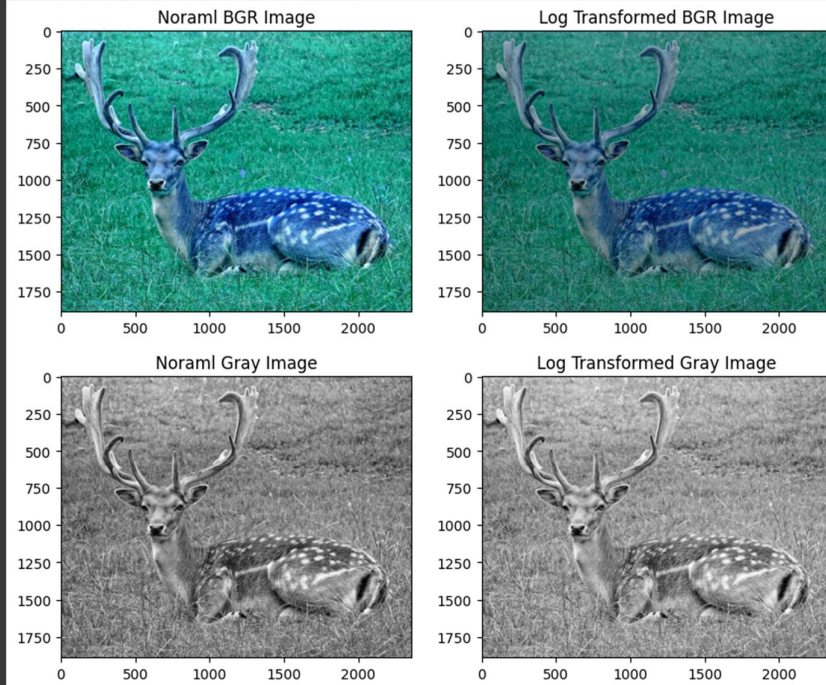
#plotting for BGR Image log transform
plt.figure(figsize=(10,8))
plt.subplot(1, 2, 1,)
plt.title("Noraml BGR Image")
plt.imshow(imgbgr)
plt.subplot(1, 2, 2)
plt.title("Log Transformed BGR Image")
plt.imshow(log_trans_bgr)

#gray log transform
a = gray/255
log_trans = 3 * np.log(1 + a)

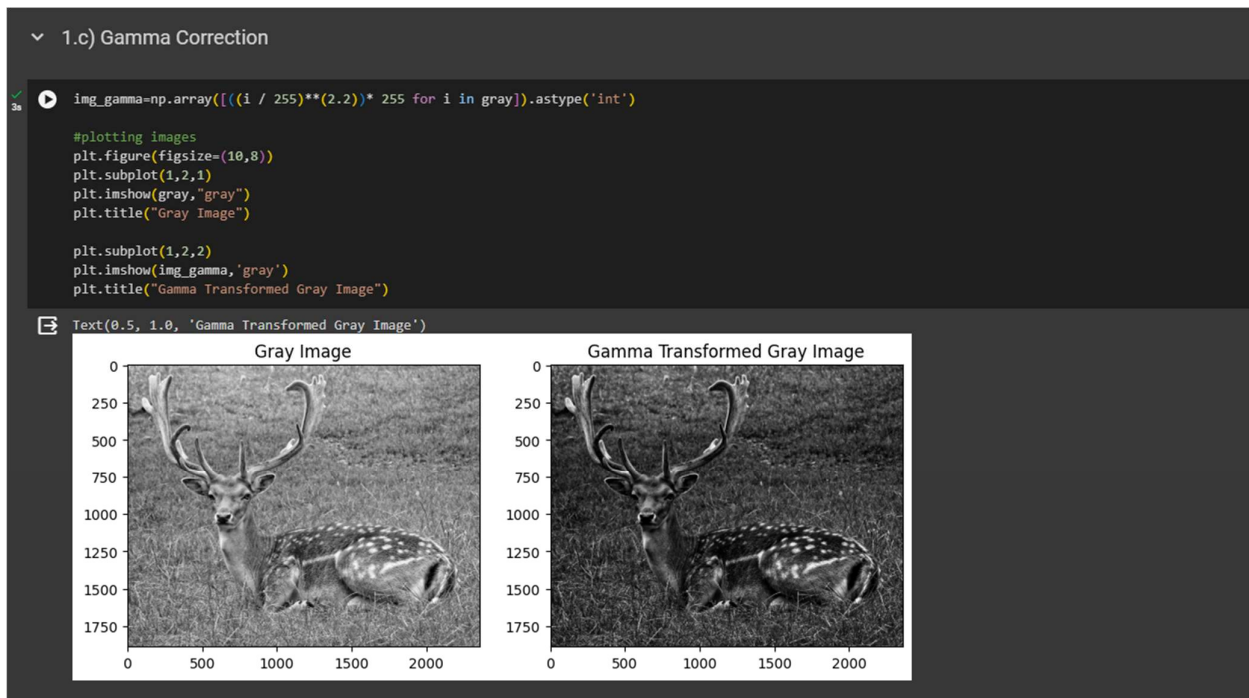
#plotting for gray log transform
plt.figure(figsize=(10,8))
plt.subplot(1, 2, 1,)
plt.title("Noraml Gray Image")
plt.imshow(gray, 'gray')

plt.subplot(1, 2, 2)
plt.title("Log Transformed Gray Image")
plt.imshow(log_trans, 'gray')
```

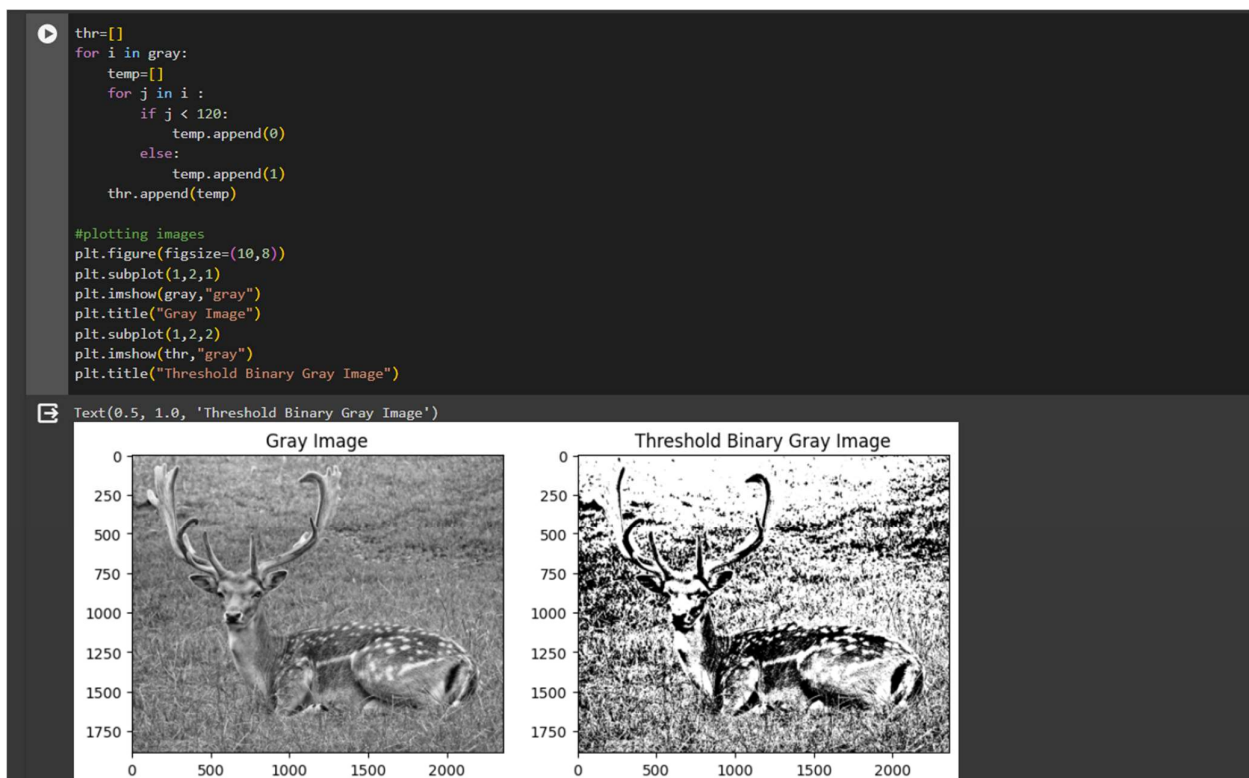
<matplotlib.image.AxesImage at 0x7fed194740d0>



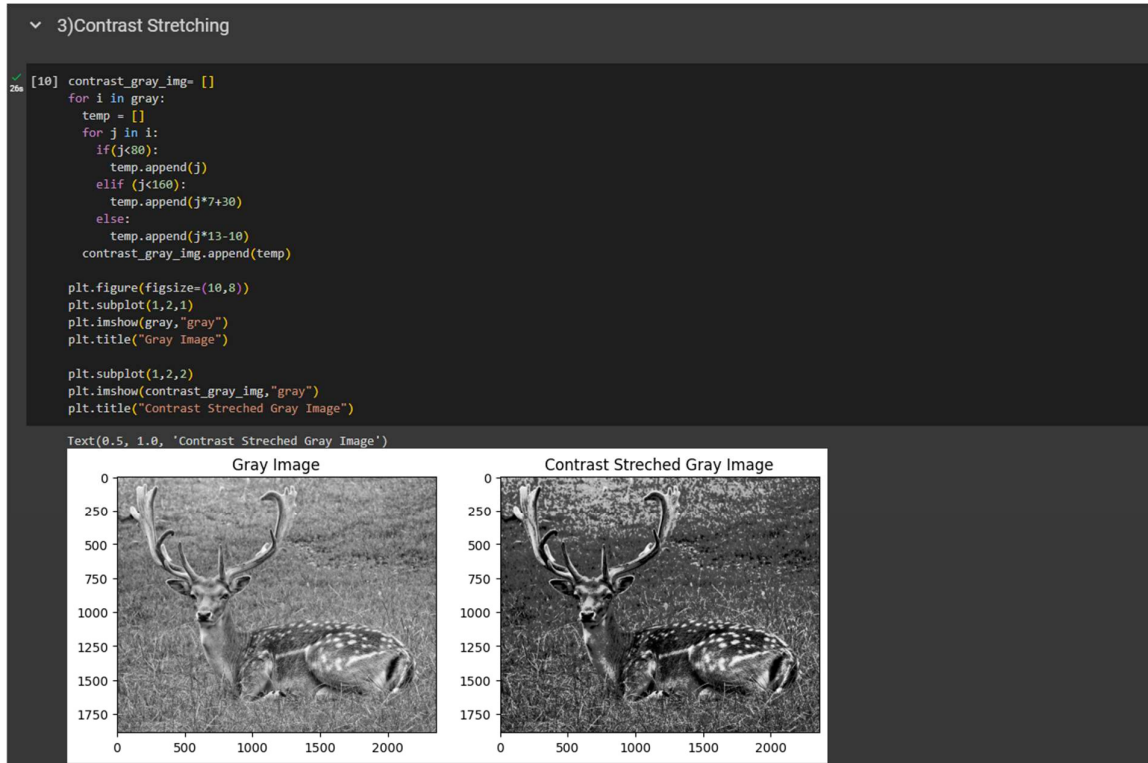
### c) Gamma Transformed Gray image



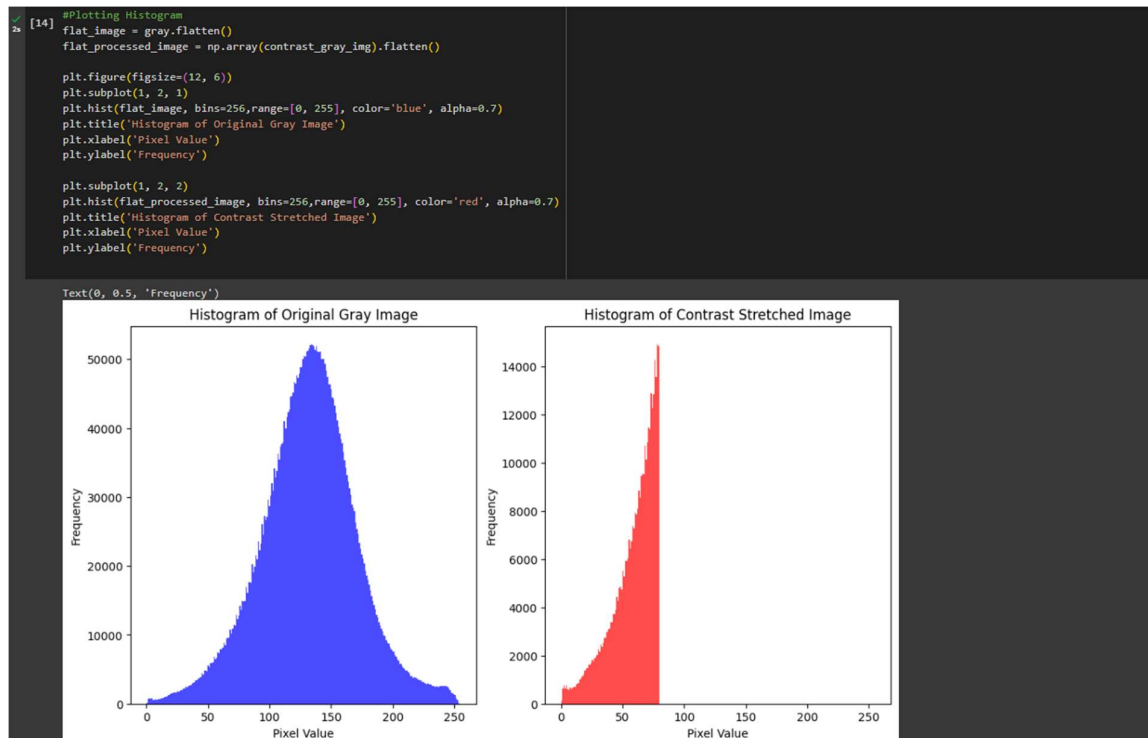
**Task-2 :** Perform an operation on the image such that gray values  $< 150$  is zero and  $> 150$  is 255 (Image Binarization).



**Task-3 :** Enhance the input gray image using contrast stretching method. Observe the output and also try to plot the histogram of the input and the processed image.



- The histogram for the gray image and contrast stretched gray image



**Results:** The given tasks have been done using programs in Python.

**Conclusion:** Python programs has been created to generate negative image, to perform log transform, gamma transform, to perform operation such that the values below the values are converted to zero and above the threshold are converted to one, to enhance the input image using contrast stretching.

Google Collab Link :

[https://colab.research.google.com/drive/1qudQqAuL6\\_7twNwwXewd44aLTvyQhqPq?usp=sharing](https://colab.research.google.com/drive/1qudQqAuL6_7twNwwXewd44aLTvyQhqPq?usp=sharing)