Shri Ramdeobaba College of Engineering and Management, Nagpur

Department of Electronics Engineering

Digital Image Processing (ENT 355-3)

Name: Prajwal Pandurang Shette

Roll No: B1- 12

Experiment No: 06

Aim: Image enhancement using point processing by following transformation functions:

a) Negative Transform

b) Log Transform

c) Power law Transform

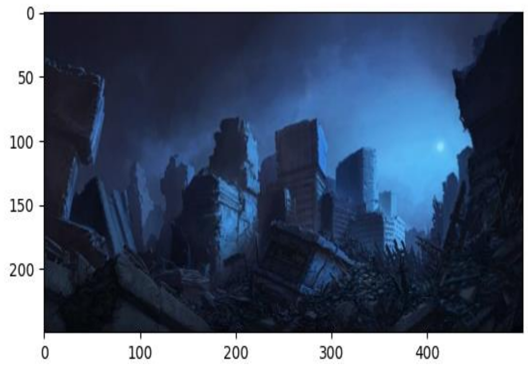
a) Negative Transform

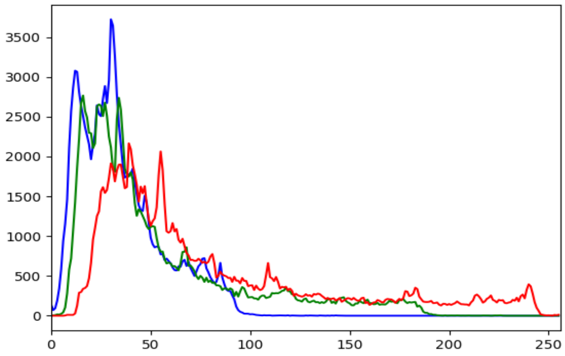
Code:

```
import cv2
import matplotlib.pyplot as plt

# Read an image
img_bgr = cv2.imread('image1.jpg', 1)
plt.imshow(img_bgr)
plt.show()

# Histogram plotting of the image
color = ('b', 'g', 'r')

for i, col in enumerate(color):

    histr = cv2.calcHist([img_bgr],
                         [i], None,
                         [256],
                         [0, 256])

    plt.plot(histr, color = col)

    # Limit X - axis to 256
    plt.xlim([0, 256])

plt.show()

# get height and width of the image
height, width, _ = img_bgr.shape

for i in range(0, height - 1):
    for j in range(0, width - 1):

        # Get the pixel value
        pixel = img_bgr[i, j]

        # Negate each channel by
        # subtracting it from 255

        # 1st index contains red pixel
        pixel[0] = 255 - pixel[0]

        # 2nd index contains green pixel
        pixel[1] = 255 - pixel[1]

        # 3rd index contains blue pixel
        pixel[2] = 255 - pixel[2]

        # Store new values in the pixel
        img_bgr[i, j] = pixel

# Display the negative transformed image
plt.imshow(img_bgr)
plt.show()

# Histogram plotting of the
# negative transformed image
color = ('b', 'g', 'r')

for i, col in enumerate(color):

    histr = cv2.calcHist([img_bgr],
                         [i], None,
                         [256],
                         [0, 256])

    plt.plot(histr, color = col)
    plt.xlim([0, 256])

plt.show()
```

Input:

Output:





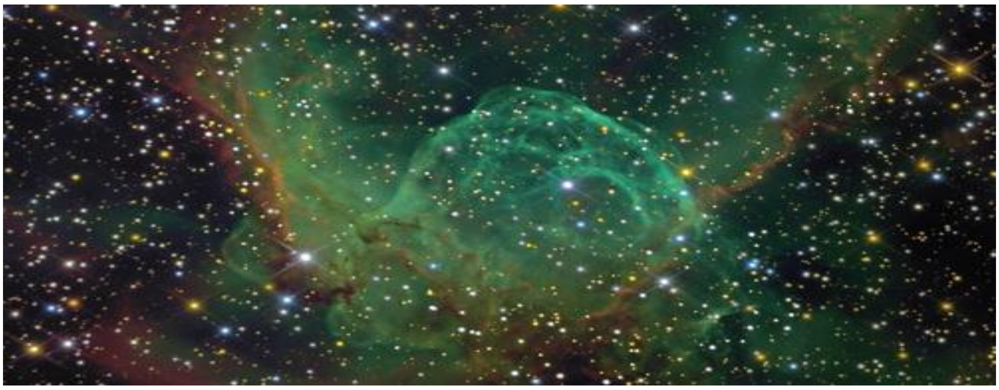
b) Log Transform

Code:

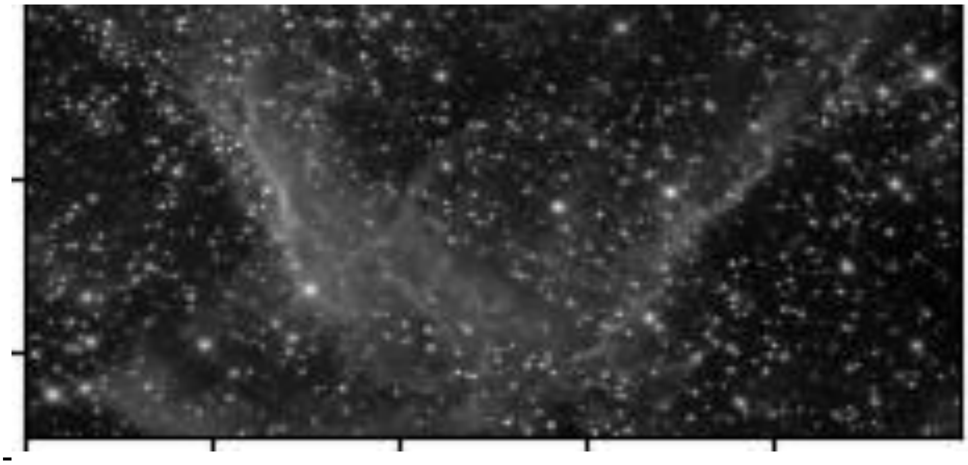
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import math

# Read an image
image = cv2.imread('image1.png')

# Apply log transformation method
c = 255 / np.log(1 + np.max(image))
log_image = c * (np.log(image + 1))

# Specify the data type so that
# float value will be converted to int
log_image = np.array(log_image, dtype = np.uint8)
```

```
# Display both images
plt.imshow(image)
plt.show()
plt.imshow(log_image)
plt.show()
```

Input:



Output:



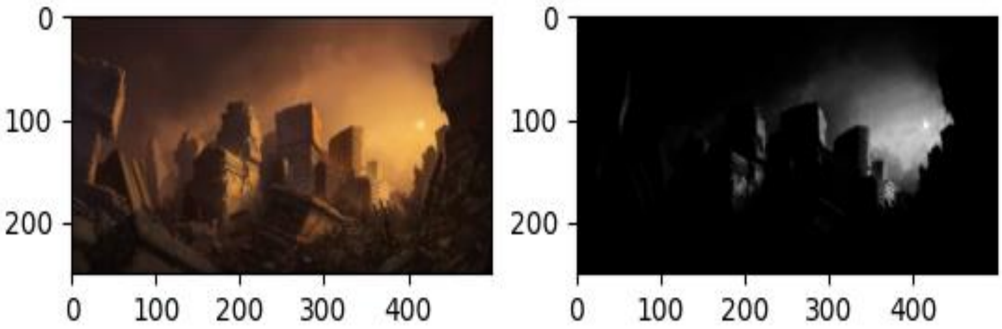
b) **Power Transform**

Code:

```
import numpy
import matplotlib.pyplot as plt
from copy import deepcopy
from PIL import Image
from math import cos, sin


def getGrayColor(rgb):
    return rgb[0]


def setGrayColor(color):
    return [color, color, color]


img = Image.open('image1.jpg')
img = numpy.asarray(img)

c = 1
y = float(input("input y :"))

# copy list not reference
pwl = deepcopy(img)


min = 99999999999999
max = 0

for i in range(len(img)):
    for j in range(len(img[i])):
        s = (c*img[i][j][0])**y
        if(s > max):
            max = s
        if(s < min):
            min = s

for i in range(len(img)):
    for j in range(len(img[i])):
        s = (c*img[i][j][0])**y
        s = (s-min)/(max-min)
        s = s*254
        pwl[i][j] = setGrayColor(s)

print('min:',min,'max:',max)
```

```
plt.subplot(2, 2, 1)
plt.imshow(img)
plt.subplot(2, 2, 2)
plt.imshow(pwl)


plt.show()
```

Input & Output:



input y :5

min: 16807.0 max: 1078203909375.0