Cynthia Li

10/20/2021

EE 440 HW3 Report

1. Since the original image is a gray scale image, the three channels have the same values, or in other words same intensity, we only need to read one channel to produce the negative. By checking the shape of the image, we know the number of pixels in each row and column. Then we can then loop through these pixels and calculate their corresponding intensity for the negative using the formula (2^n – I)-1, in which n = 8 (gray scale intensity value is from 0 to 256) and I is the corresponding value of lena. With the new intensity stored in sequence in lena_neg, we can show the new image labeled as “negative.” The results are show below in Fig. 1. And the code for this part is shown in APPENDIX. I.

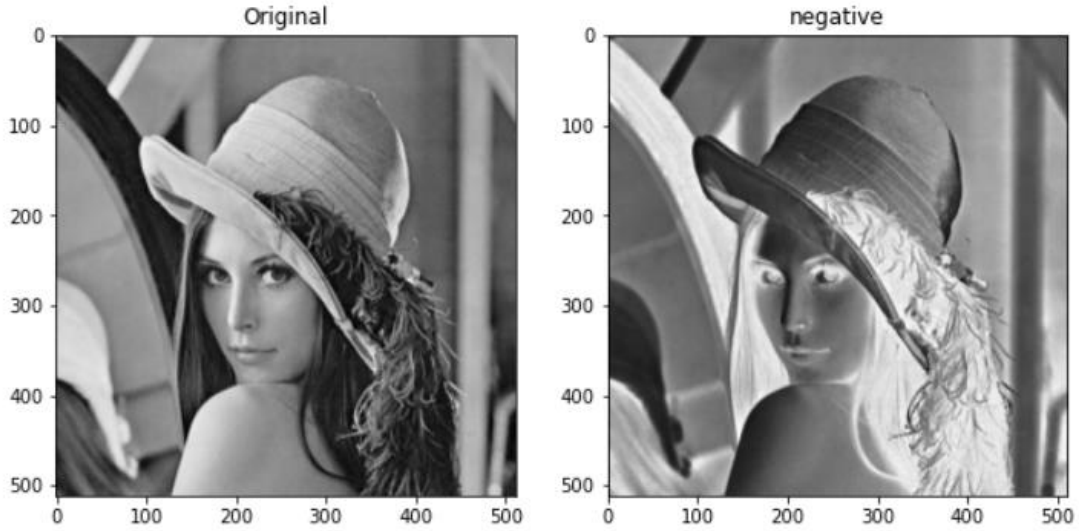

Fig. 1. Original and negative image of 3_1.bmp

2. We first read in the image. Then in i., we can extract red, green and blue’s value from each channel, and display them separately on the plot. The results are shown below in Fig. 2. In ii., we can first convert the original image to hsv, then extract each channel’s value for hue, saturation and value, and lastly display each channels’ value separately on the plot. The results are shown below in Fig. 3.

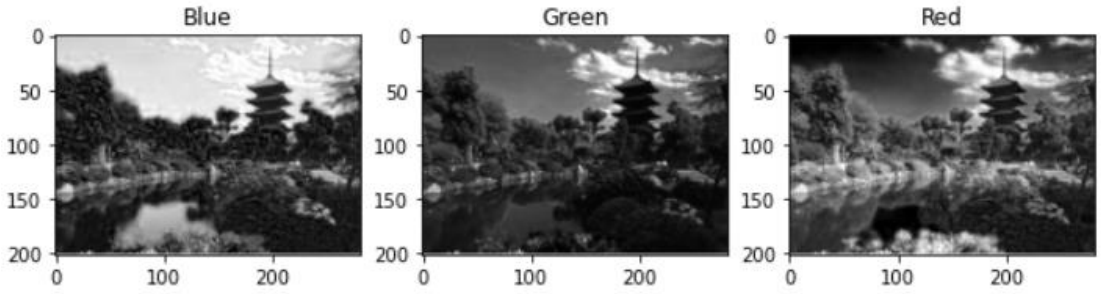

Fig. 2. RGB image of 3_2.bmp

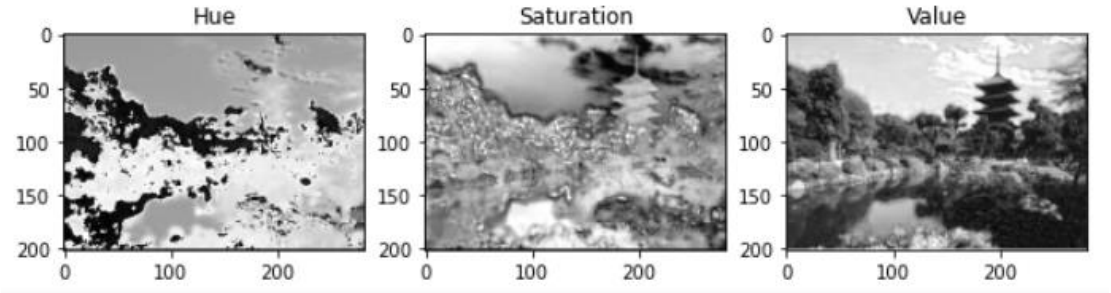

Fig. 3. HSV image of 3_2.bmp

The code for this problem is shown in APPENDIX. II.

APPENDIX

I.

```
# This is the import cell
import numpy as np
import cv2
import matplotlib.pyplot as plt
from skimage import color
```

```
# Problem 1   Negate 3_1.bmp

# to display multiple pictures
fig = plt.figure(figsize=(10, 7))

# read and show the image 3_1.bmp Lena
fig.add_subplot(1, 2, 1)
lena = cv2.imread('3_1.bmp', 0)
plt.imshow(lena)
plt.title("Original")

# check the shape of the image to see the number
# of pixels in each row and column
print(np.shape(lena))

# calculate the negative of lena
# n = 8, I = lena[i, j, k]
neg = np.zeros((512, 512))
lena_neg = neg.astype('uint8')
for i in range (1, 512):
    for j in range (1, 512):
            lena_neg[i, j] = (256 - lena[i, j]) - 1

# show the negative image
fig.add_subplot(1, 2, 2)
plt.imshow(lena_neg, cmap='gray')
plt.title("negative")
```

II.

```
# Problem 3    Color Channels of 3_2.bmp

# i. Display the RGB image of 3_2.bmp
# read the image in BGR order
X = cv2.imread('3_2.bmp')

# extract single channel intensity
# blue channel
B = X[:, :, 0]

# green channel
G = X[:, :, 1]

# red channel
R = X[:, :, 2]

# to display rgb images
fig2 = plt.figure(figsize=(10, 7))
# to display blue channel
fig2.add_subplot(1, 3, 1)
plt.imshow(B)
plt.title("Blue")
# to display green channel
fig2.add_subplot(1, 3, 2)
plt.imshow(G)
plt.title("Green")
# to display red channel
fig2.add_subplot(1, 3, 3)
plt.imshow(R)
plt.title("Red")
```

```
# ii. Display the HSV image of 3_2
# convert the original image to HSV
X_hsv = cv2.cvtColor(X, cv2.COLOR_BGR2HSV)

# extract hsv values from each channel
# hue
H = X_hsv[:, :, 0]
# saturation
S = X_hsv[:, :, 1]
# value
V = X_hsv[:, :, 2]

# to display hsv images
fig3 = plt.figure(figsize=(10, 7))
# to display hue channel
fig3.add_subplot(1, 3, 1)
plt.imshow(H)
plt.title("Hue")
# to display saturation channel
fig3.add_subplot(1, 3, 2)
plt.imshow(S)
plt.title("Saturation")
# to display value channel
fig3.add_subplot(1, 3, 3)
plt.imshow(V)
plt.title("Value")
```