Assignment-2

Generation of Different Color Shades using RGB Channels

Digital Image Processing Lab

1 Objective

The objective of this experiment is to:

- Generate different shades using RGB channels
- Understand how color intensities affect image formation
- Visualize grayscale and color combinations

2 Methodology

The program creates images of different color intensities using:

$$I(x, y) = value$$

Different RGB combinations are created by assigning intensity values to specific channels.

- x = 0 : Grayscale (All channels equal)
- x = 1 : Red channel
- x = 2 : Green channel
- x = 3 : Blue channel
- x = 4 : Cyan (Green + Blue)
- x = 5 : Magenta (Red + Blue)
- x = 6 : Yellow (Red + Green)

Intensity values used:

$$\{0, 50, 127, 200, 255\}$$

3 Output Result

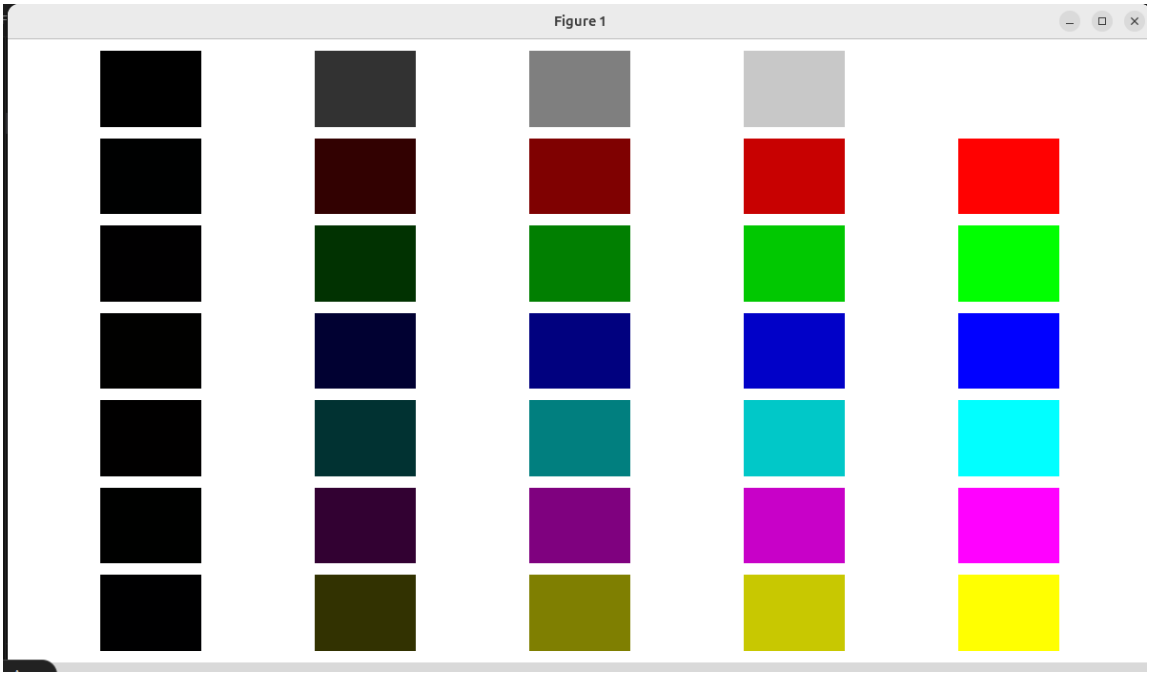

Figure 1: Generated Shades for Different RGB Channel Combinations

(You can save the plot as `shade_output.png` for report inclusion.)

4 Observation

- Increasing intensity increases brightness.
- Single channel activation produces primary colors.
- Combination of two channels produces secondary colors.
- Equal values in all channels produce grayscale.

5 Complete Python Code

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

def shade(height, width, value, x):
    if x == 0:
        img = np.ones((height, width, 3), dtype=np.uint8) * value
        return img

    elif x == 4:
```

```
        img = np.ones((height, width, 3), dtype=np.uint8)
        img[:, :, 1] = value
        img[:, :, 2] = value
        return img

    elif x == 5:
        img = np.ones((height, width, 3), dtype=np.uint8)
        img[:, :, 0] = value
        img[:, :, 2] = value
        return img

    elif x == 6:
        img = np.ones((height, width, 3), dtype=np.uint8)
        img[:, :, 0] = value
        img[:, :, 1] = value
        return img

    img = np.ones((height, width, 3), dtype=np.uint8)
    img[:, :, x-1] = value
    return img

img = cv2.imread("img1.jpg")
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
height, width, channels = img_rgb.shape

values = [0, 50, 127, 200, 255]

pos = 1
plt.figure(figsize=(12,8))

for i in range(7):
    for j in range(5):
        plt.subplot(7,5,pos)
        plt.imshow(shade(height, width, values[j], i))
        plt.axis('off')
        pos += 1

plt.tight_layout()
plt.savefig("shade_output.png")
plt.show()
```