

Colored and Grayscale Image Processing

Student Information

- Name : M. Fazri Nizar
 - NIM : 09021282328053
 - Course : Image Processing L1
-

0. Install All Needed Libraries

```
%pip install matplotlib
%pip install pillow
%pip install requests
%pip install numpy
```

```
Requirement already satisfied: matplotlib in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (3.10.0)
Requirement already satisfied: contourpy>=1.0.1 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from matplotlib)
(1.3.1)
Requirement already satisfied: cyclor>=0.10 in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from matplotlib)
(4.55.8)
Requirement already satisfied: kiwisolver>=1.3.1 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from matplotlib)
(1.4.8)
Requirement already satisfied: numpy>=1.23 in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (from matplotlib) (2.2.2)
Requirement already satisfied: packaging>=20.0 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from matplotlib)
(24.2)
Requirement already satisfied: pillow>=8 in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from matplotlib)
(3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from matplotlib)
(2.9.0.post0)
Requirement already satisfied: six>=1.5 in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (from python-dateutil>=2.7-
>matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pillow in d:\projects\mlprojects\image-
```

```
processing\.conda\lib\site-packages (11.1.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: requests in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in d:\
projects\mlprojects\image-processing\.conda\lib\site-packages (from
requests) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in d:\projects\mlprojects\
image-processing\.conda\lib\site-packages (from requests) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from requests)
(2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in d:\projects\
mlprojects\image-processing\.conda\lib\site-packages (from requests)
(2025.1.31)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: numpy in d:\projects\mlprojects\image-
processing\.conda\lib\site-packages (2.2.2)
Note: you may need to restart the kernel to use updated packages.
```

1. Import All Needed Libraries

```
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
import requests
```

2. Define All Functions

```
# Function to load and display image
def load_and_display_image(image_path, title):
    image = Image.open(image_path)
    plt.imshow(image)
    plt.title(title)
    plt.axis('off')
    plt.show()
    return image

# Function to convert RGB image to Grayscale
def rgb_to_grayscale(image):
    image_array = np.array(image)
    grayscale_array = np.dot(image_array[...,:3], [0.299, 0.587,
0.114])
    grayscale_image =
Image.fromarray(grayscale_array.astype(np.uint8))
```

```

plt.imshow( grayscale_image, cmap='gray')
plt.title('Grayscale Image')
plt.axis('off')
plt.show()
grayscale_image.save("image_gray.jpg")
return grayscale_image

# Function to convert image data to vector
def image_to_vector(image):
    return np.array(image).flatten()

# Function to analyze differences between two vectors
def analyze_difference(colored_vector, grayscale_vector):
    difference_vector = np.abs(colored_vector[:len(grayscale_vector)]
    - grayscale_vector)
    return difference_vector

# Function to plot the difference vector
def plot_difference(difference_vector):
    plt.figure(figsize=(10, 5))
    plt.plot(difference_vector)
    plt.title('Difference Between Colored and Grayscale Image
Vectors')
    plt.xlabel('Pixel Index')
    plt.ylabel('Difference Value')
    plt.ylim(0, 255)
    plt.show()

```

3. Download, Load and Display Colored Images

```

# URL of the image
url =
'https://upload.wikimedia.org/wikipedia/commons/d/d3/Pemandangan_Alam.
jpg'

# Send a GET request to the URL
response = requests.get(url)

image_name = 'image.jpg'

with open(image_name, 'wb') as f:
    f.write(response.content)

print(f"Image downloaded and saved as '{image_name}'")

# Load and display the colored image

```

```
image_path = image_name  
colored_image = load_and_display_image(image_path, 'Colored Image')  
Image downloaded and saved as 'image.jpg'
```

Colored Image



4. Convert Colored Image to Grayscale

```
grayscale_image = rgb_to_grayscale(colored_image)
```

Grayscale Image



5. Extract Image Data to Vectors/Arrays

```
colored_image_vector = image_to_vector(colored_image)
grayscale_image_vector = image_to_vector(grayscale_image)

colored_image_vector_length = len(colored_image_vector)
grayscale_image_vector_length = len(grayscale_image_vector)

print(f"Colored Image Vector Length: {colored_image_vector_length}")
print(f"Grayscale Image Vector Length: {grayscale_image_vector_length}")

# Display the first 100 elements of the vectors
print(f"First 100 elements of Colored Image Vector: {colored_image_vector[:100]}")
print(f"First 100 elements of Grayscale Image Vector: {grayscale_image_vector[:100]}")

# Save the full vectors to text files
np.savetxt('colored_image_vector.txt', colored_image_vector, fmt='%d')
np.savetxt('grayscale_image_vector.txt', grayscale_image_vector,
fmt='%d')

print("Full vectors saved to 'colored_image_vector.txt' and 'grayscale_image_vector.txt'")

Colored Image Vector Length: 768000
Grayscale Image Vector Length: 256000
```

```

First 100 elements of Colored Image Vector: [ 38  59 106  38  59 106
38  59 106  38  59 106  38  59 106  38  59 106
 38  59 106  38  59 106  40  61 108  40  61 108  40  61 108  40  61
108
 40  61 108  40  61 108  40  61 108  40  61 106  37  60 104  37  60
102
 37  60 102  37  60 102  37  60 102  37  60 102  37  60 102  37  60
102
 37  60 102  37  60 102  37  60 102  37  60 102  37  60 102  37  60
102
 37  60 102  37  60 102  37  60 102  37]
First 100 elements of Grayscale Image Vector: [58 58 58 58 58 58 58 58
60 60 60 60 60 60 60 59 58 57 57 57 57 57 57 57
57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57 57
57
57 57 57 57 57 57 57 57 56 56 56 56 56 56 56 56 56 56 56 56 56 56
56
55 55 55 55 55 55 55 56 55 55 55 55 55 55 55 55 58 58 58 58 58 58 58
58
55 55 55 55]
Full vectors saved to 'colored_image_vector.txt' and
'grayscale_image_vector.txt'

```

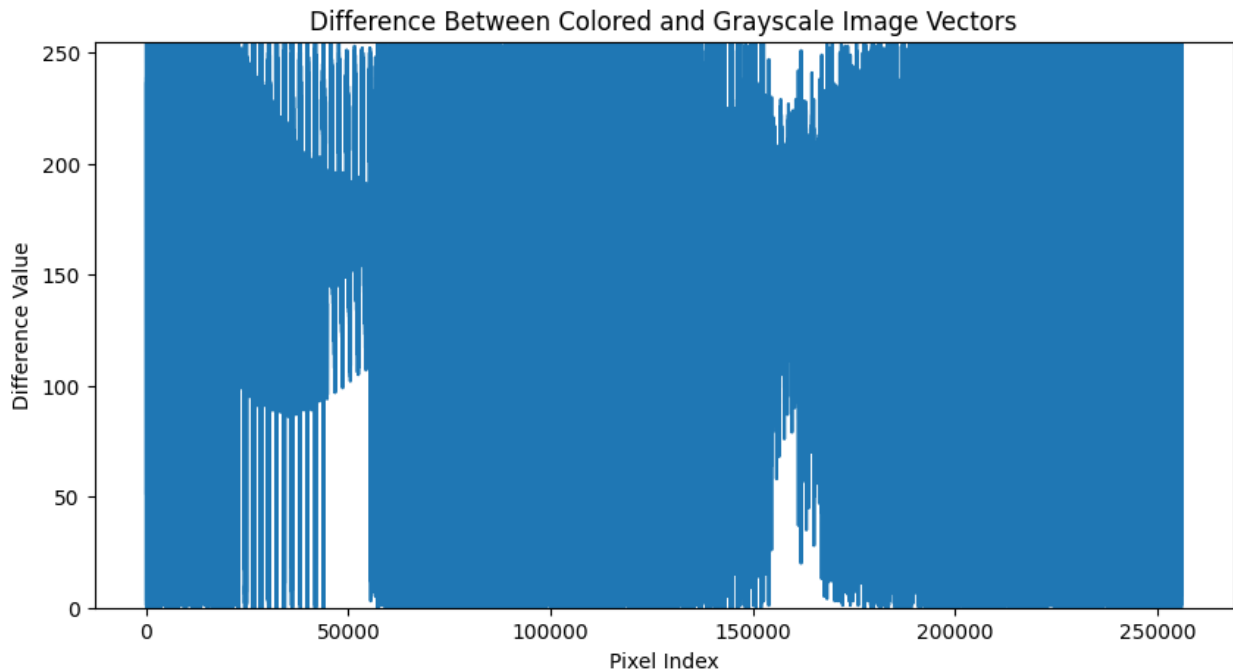
6. Analyze Differences Between Colored and Grayscale Images

```

# Calculate the difference vector
difference_vector = analyze_difference(colored_image_vector,
grayscale_image_vector)

# Plot the difference vector
plot_difference(difference_vector)

```



Conclusion

Steps done:

1. The colored image is downloaded from a URL and saved locally.
2. The colored image is loaded and displayed using a custom function.
3. The colored image is converted to a grayscale image manually using numpy operations.
4. Both the colored and grayscale images are converted to vectors/arrays for analysis.
5. The difference between the colored and grayscale image vectors is calculated and plotted.
6. The plot shows the absolute difference in pixel values between the colored and grayscale images.

The grayscale image is a single-channel image where each pixel represents the intensity of light, whereas the colored image has three channels (Red, Green, Blue) representing color information. The difference vector highlights the changes in pixel values when converting from colored to grayscale.