**Problem No:** 02

**Problem Name**: Color conversion of images and histogram generation.

**Description:** Image color conversion is the process of changing an image from one color space to another to make it easier to analyze or process.It helps in simplifying image data and extracting meaningful information. Common conversions include RGB to Gray scale, RGB to Binary, and RGB to HSV. Gray scale images represent brightness only, while binary images show objects in black and white. Color conversion is an essential step in image preprocessing for computer vision and pattern recognition tasks.

**RGB to Gray scale Conversion**: An RGB image contains three color channels **Red (R)**, **Green (G)**, and **Blue(B)**.To convert it into gray scale, the color information is removed, and brightness is calculated using a weighted sum of the channels. The resulting image contains only intensity values (0–255), making processing faster and simpler.

**RGB to Binary Conversion:** A binary image contains only two pixel values: 0 for black and 1 for white. It is created by applying a threshold to a grayscale image, where pixel values above the threshold become white and those below become black. This conversion simplifies the image by highlighting key objects or regions. Binary images are widely used in object detection, segmentation, and pattern recognition.

Conversion of rgb folder to grayscale and binary image folder:

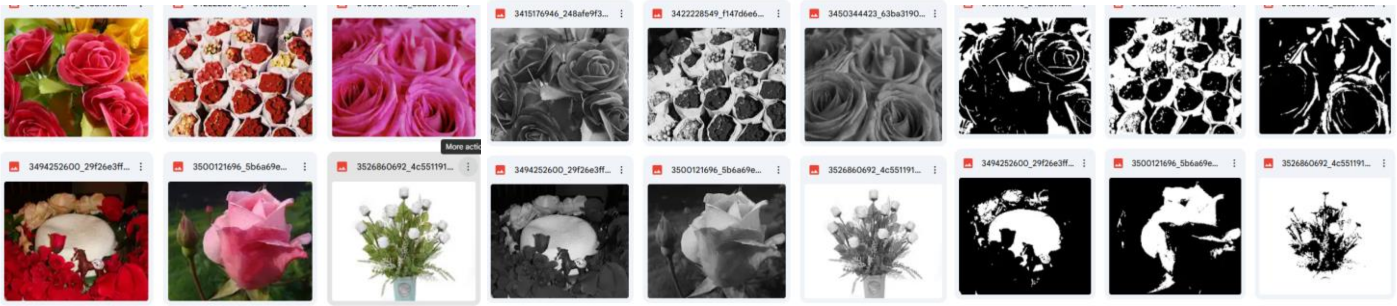


Figure:02: Folder of rgb images, grayscale images and binary images.

**Source Code and Output:**

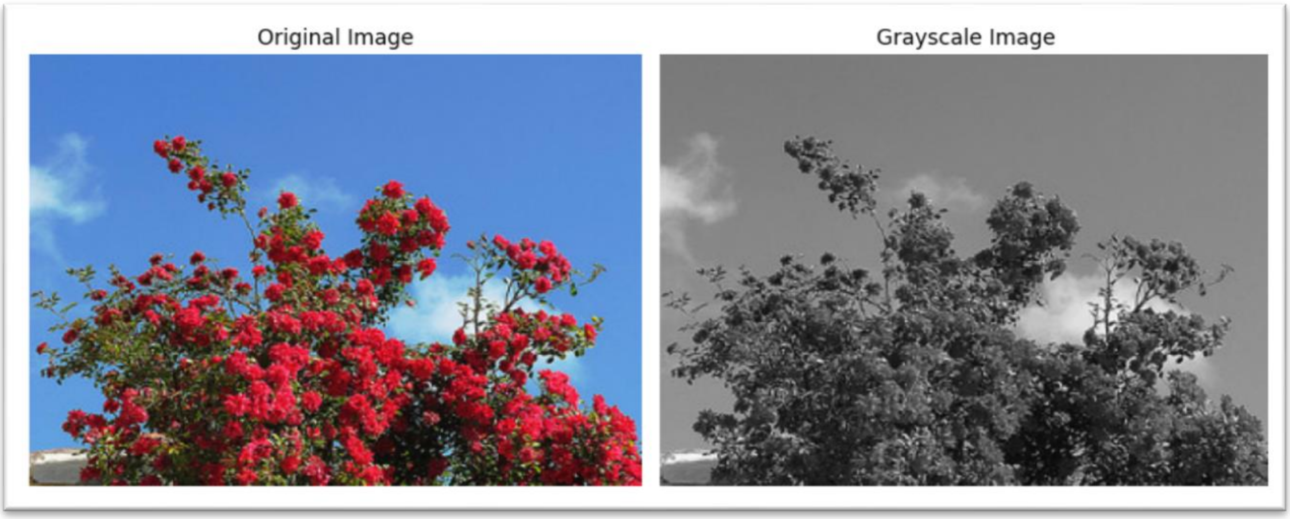
RGB to GrayScale:

```
import os

import cv2

from glob import glob

# Define the destination directory in Google Drive
output_dir = '/content/drive/MyDrive/gray_flower'
# Create the directory if it doesn't exist
os.makedirs(output_dir, exist_ok=True)
# Process each image in the flower_folder
for img_path in flower_folder:
    # Read the image
    img = cv2.imread(img_path)
    # Convert the image to grayscale
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    filename = os.path.basename(img_path)
    output_path = os.path.join(output_dir, filename)
   # Save the grayscale image to Google Drive
    cv2.imwrite(output_path, gray_img)
   print(f"Grayscale images saved to {output_dir}")
```



```
all_pixel_values_gray = []
# Process each image in the gray_folder
for img_path in gray_folder:
    # Read the grayscale image
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    all_pixel_values_gray.extend(img.flatten())
all_pixel_values_gray = np.array(all_pixel_values_gray)
plt.figure(figsize=(10, 6))
plt.hist(all_pixel_values_gray, bins=256, range=[0, 256], color='gray')
plt.title('Histogram of Pixel Values in Grayscale Images (gray_folder)')
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)
plt.show()
```

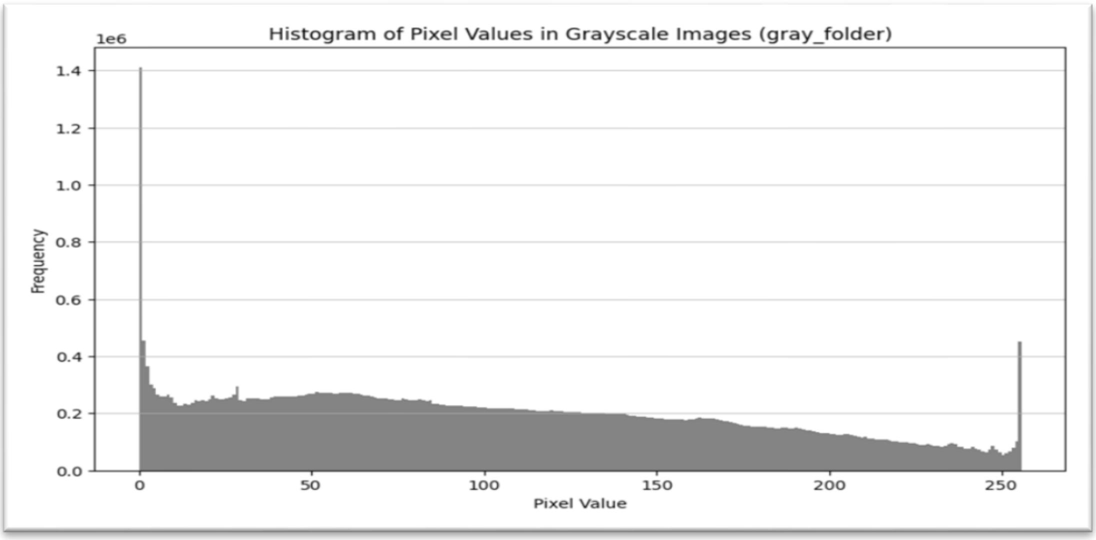


Fig: 04: Histogram of Pixel Values in Grayscale Image.

### RGB to Binary Image:

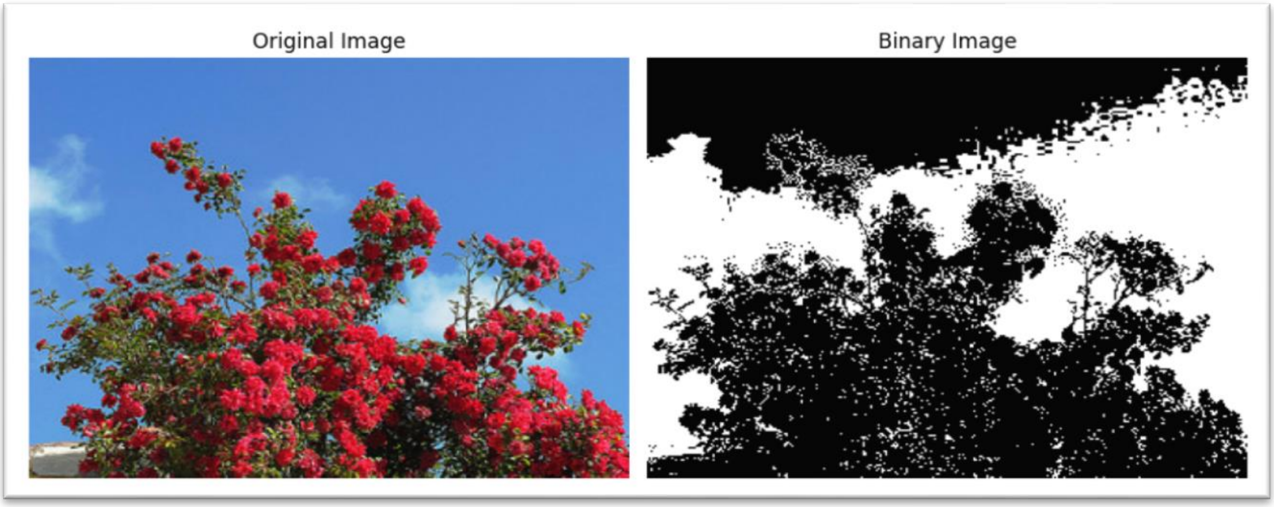
```
import os
import cv2
from glob import glob
# Define the destination directory in Google Drive for binary images
output_dir_binary = '/content/drive/MyDrive/binary_flower'


os.makedirs(output_dir_binary, exist_ok=True)
```

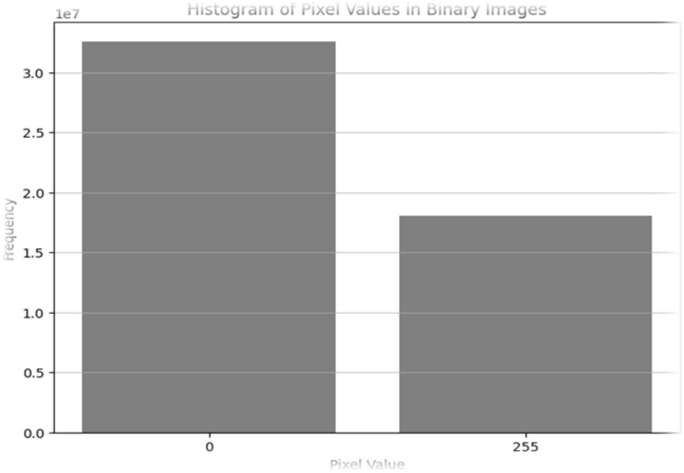
```
# Process each image in the flower_folder
for img_path in flower_folder:
    # Read the image
    img = cv2.imread(img_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    ret, binary_img = cv2.threshold(gray_img, 127, 255, cv2.THRESH_BINARY)

    filename = os.path.basename(img_path)output_path_binary =
os.path.join(output_dir_binary, filename)

cv2.imwrite(output_path_binary, binary_img)print(f"Binary images saved to
{output_dir_binary}")
```



```
import numpy as np
# Initialize a list to store pixel values
all_pixel_values = []
# Process each image in the binary_folder
for img_path in binary_folder:
    # Read the binary image
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
   all_pixel_values.extend(img.flatten())
all_pixel_values = np.array(all_pixel_values)
# Create a histogram of the pixel values
plt.figure(figsize=(8, 6))
plt.hist(all_pixel_values, bins=2, range=[0, 256], color='gray', rwidth=0.8)
plt.title('Histogram of Pixel Values in Binary Images')
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')
plt.xticks([64, 192], ['0', '255']) # Label bins as 0 and 255
plt.grid(axis='y', alpha=0.75)
plt.show()
```


Histogram of Pixel Values in Binary Images
1e7
Frequency
Pixel Value