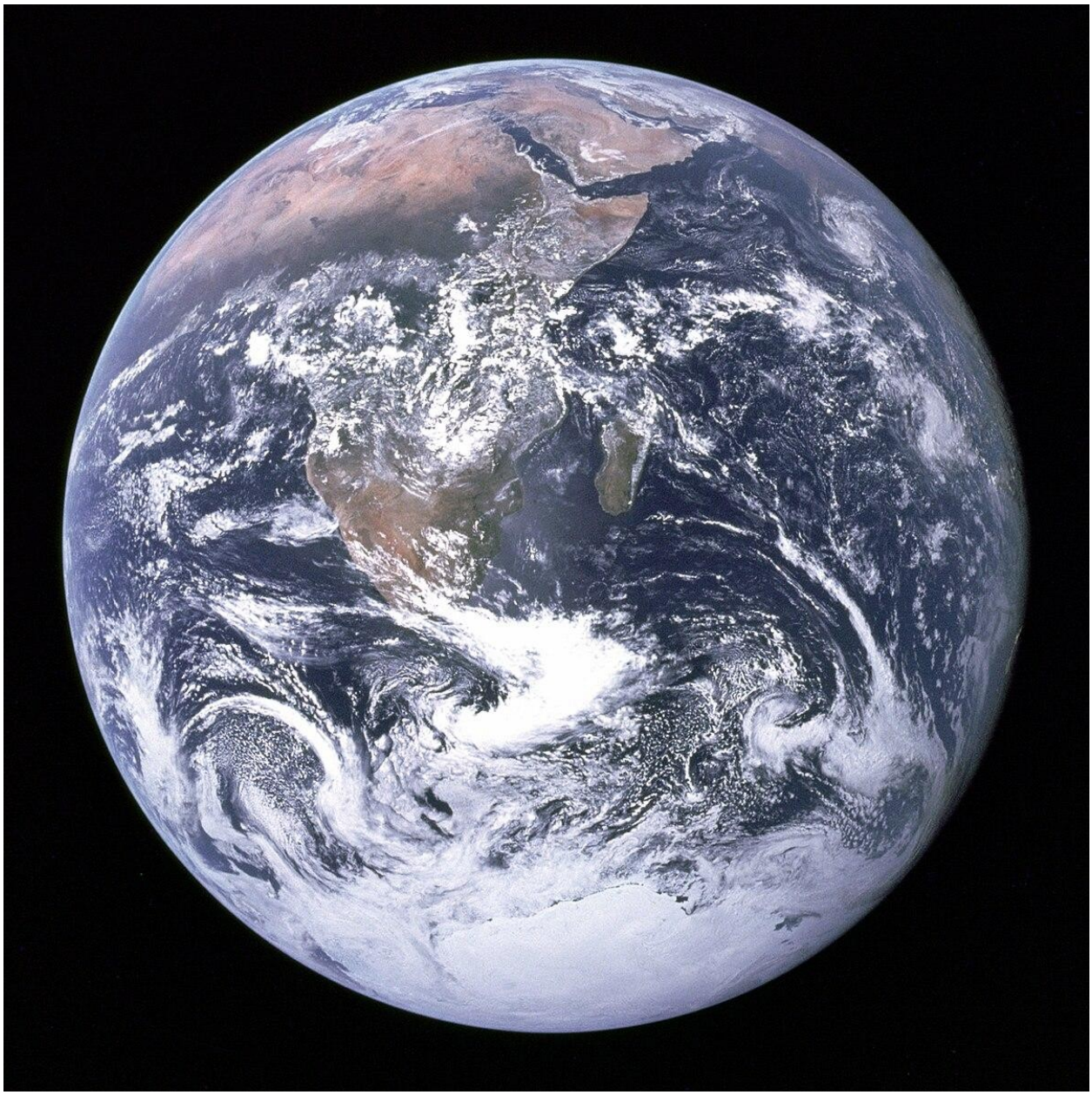
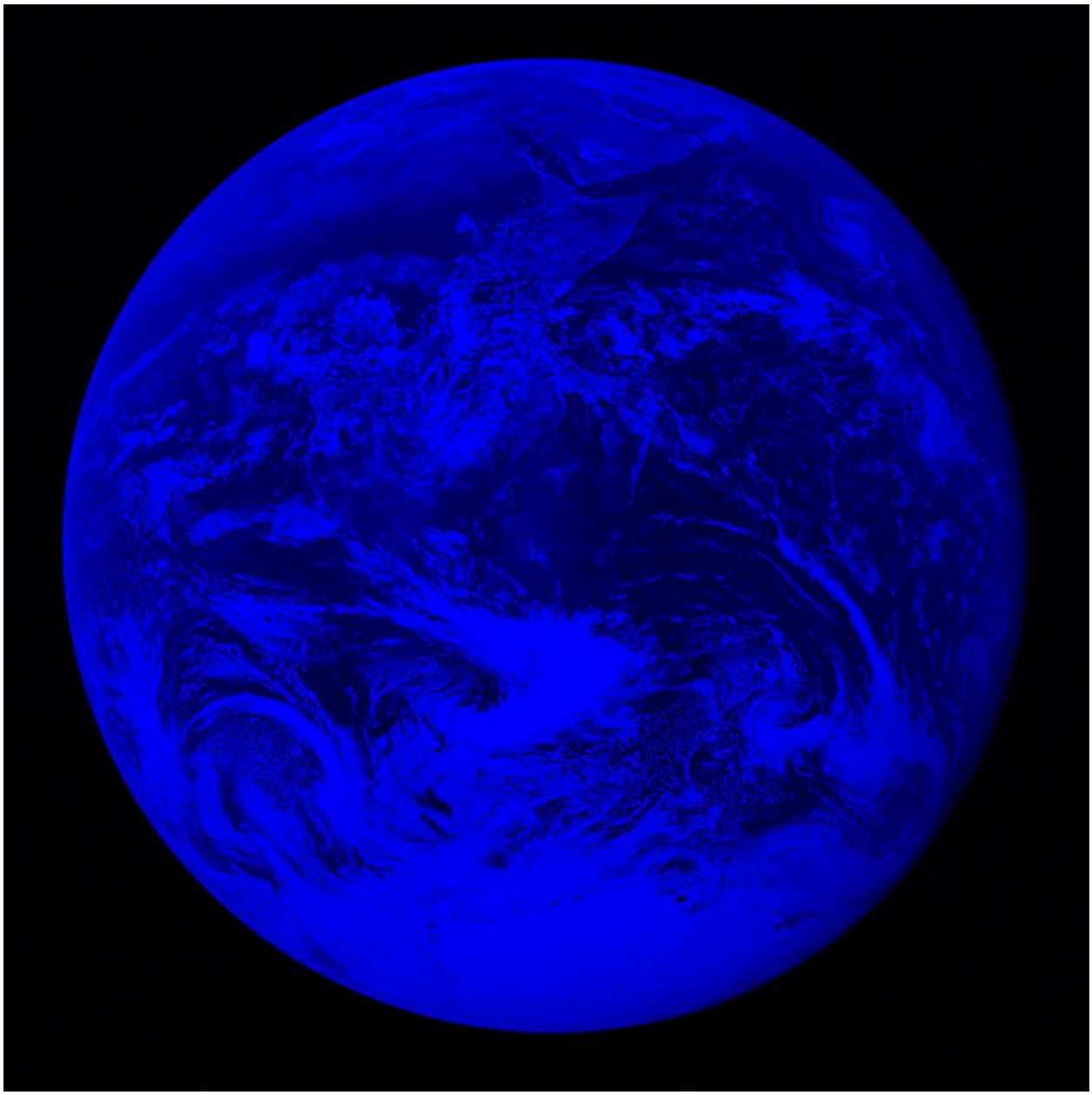
```python
# Import all necessary libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.feature import local_binary_pattern
from google.colab.patches import cv2_imshow
import os

# --- 1. Download and Load a NEW Image from the Internet ---
image_url =
"https://upload.wikimedia.org/wikipedia/commons/thumb/9/97/The_Earth_s
een_from_Apollo_17.jpg/1200px-The_Earth_seen_from_Apollo_17.jpg"
image_filename = "animal_image.jpg"

# Use wget to download the image from the URL and save it with a
specific filename
# The '-O' flag specifies the output filename.
!wget -q -O {image_filename} {image_url}

# Check if the image was downloaded and then load it
if os.path.exists(image_filename):
    image_bgr = cv2.imread(image_filename)
    print("Original Image:")
    cv2_imshow(image_bgr)
    print("\n" + "="*40 + "\n")

    # --- 2. Color Channel Isolation ---
    b, g, r = cv2.split(image_bgr)
    zeros = np.zeros_like(b)

    blue_channel_image = cv2.merge([b, zeros, zeros])
    green_channel_image = cv2.merge([zeros, g, zeros])
    red_channel_image = cv2.merge([zeros, zeros, r])

    print("Blue Channel Isolated:")
    cv2_imshow(blue_channel_image)
    print("\nGreen Channel Isolated:")
    cv2_imshow(green_channel_image)
    print("\nRed Channel Isolated:")
    cv2_imshow(red_channel_image)
    print("\n" + "="*40 + "\n")

    # --- 3. Color Histogram Graph ---
    print("Combined Color Histogram Graph:")
    plt.figure(figsize=(10, 6))
    plt.title("Color Channel Histograms")
    plt.xlabel("Pixel Intensity (0-255)")
    plt.ylabel("Number of Pixels")

    hist_blue = cv2.calcHist([b], [0], None, [256], [0, 256])
```

```
    plt.plot(hist_blue, color='blue', label='Blue Channel')

    hist_green = cv2.calcHist([g], [0], None, [256], [0, 256])
    plt.plot(hist_green, color='green', label='Green Channel')

    hist_red = cv2.calcHist([r], [0], None, [256], [0, 256])
    plt.plot(hist_red, color='red', label='Red Channel')

    plt.legend()
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.xlim([0, 256])
    plt.show()
    print("\n" + "="*40 + "\n")

    # --- 4. LBP Texture Histogram ---
    print("Texture (LBP) Histogram Graph:")
    gray_image = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
    n_points = 8
    radius = 1
    lbp = local_binary_pattern(gray_image, n_points, radius,
method='uniform')
    (hist, _) = np.histogram(lbp.ravel(),
                             bins=np.arange(0, n_points + 3),
                             range=(0, n_points + 2))
    hist = hist.astype("float")
    hist /= (hist.sum() + 1e-6)

    plt.figure(figsize=(10, 6))
    plt.title("Texture (LBP) Histogram")
    plt.xlabel("LBP Code")
    plt.ylabel("Normalized Frequency")
    plt.bar(np.arange(0, n_points + 2), hist, color='gray')
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.xlim([0, n_points + 2])
    plt.show()

else:
    print(f"Error: Could not download the image from the URL.")
```

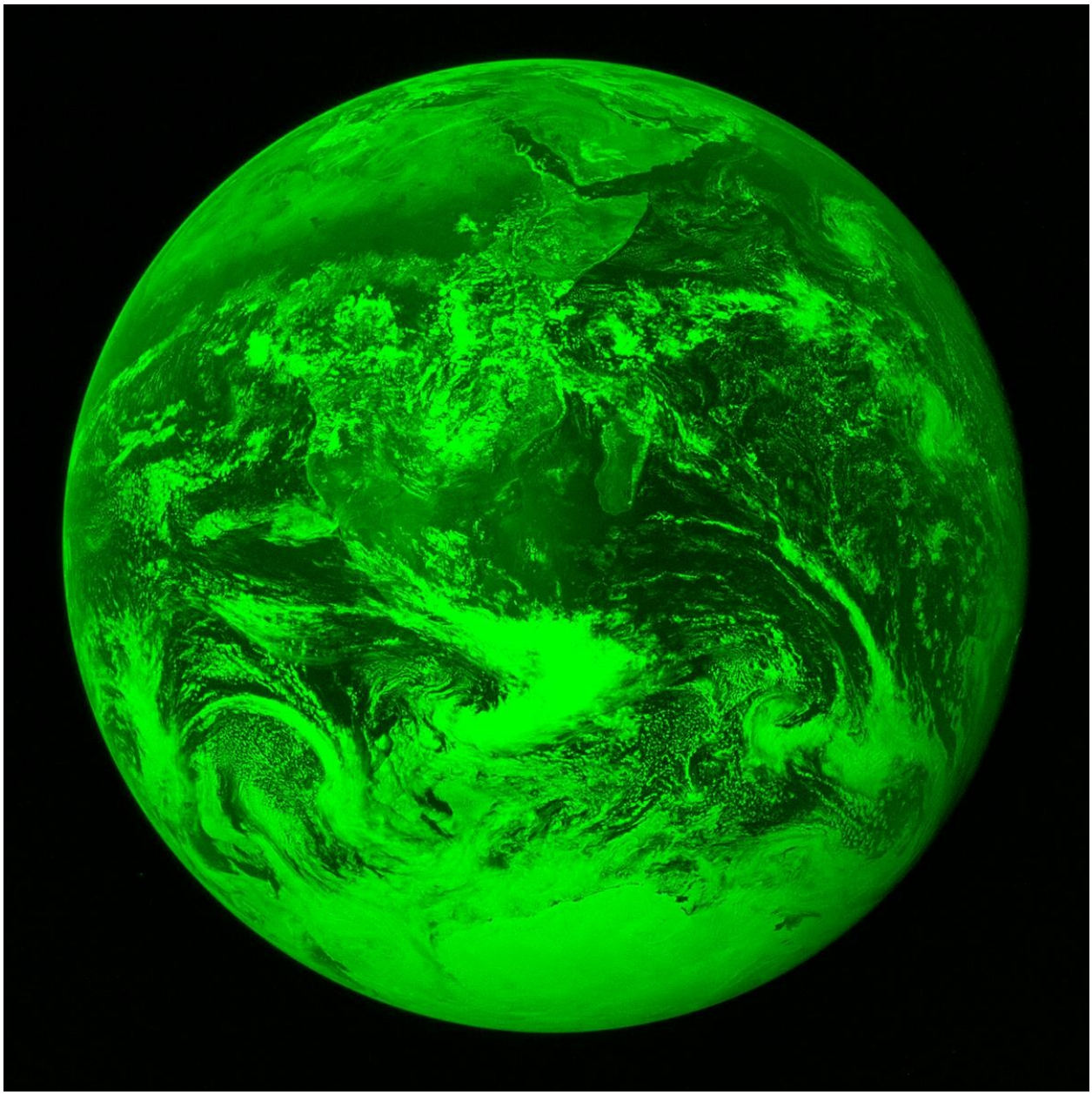
```
Original Image:
```



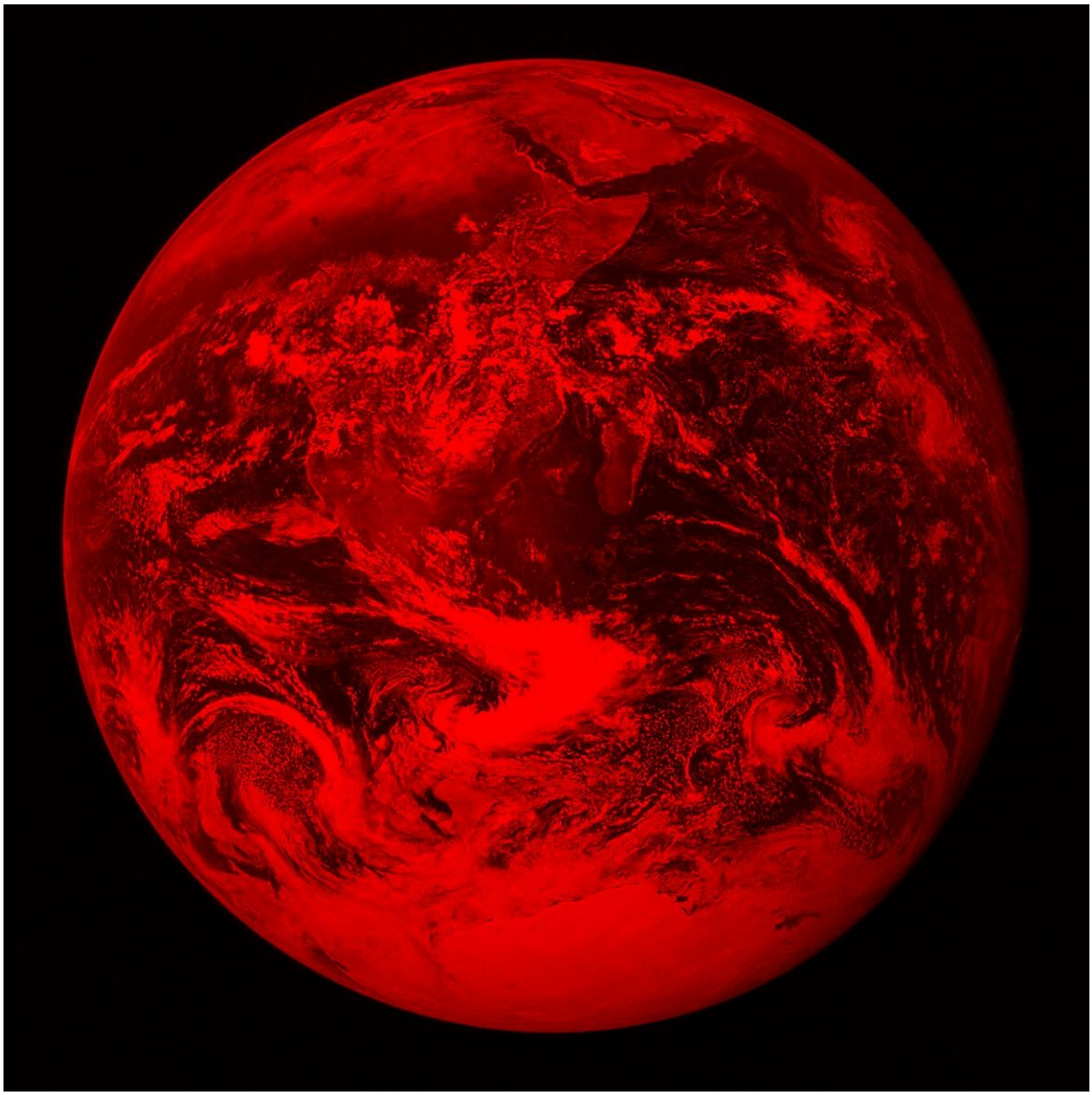
```
========================================

Blue Channel Isolated:
```



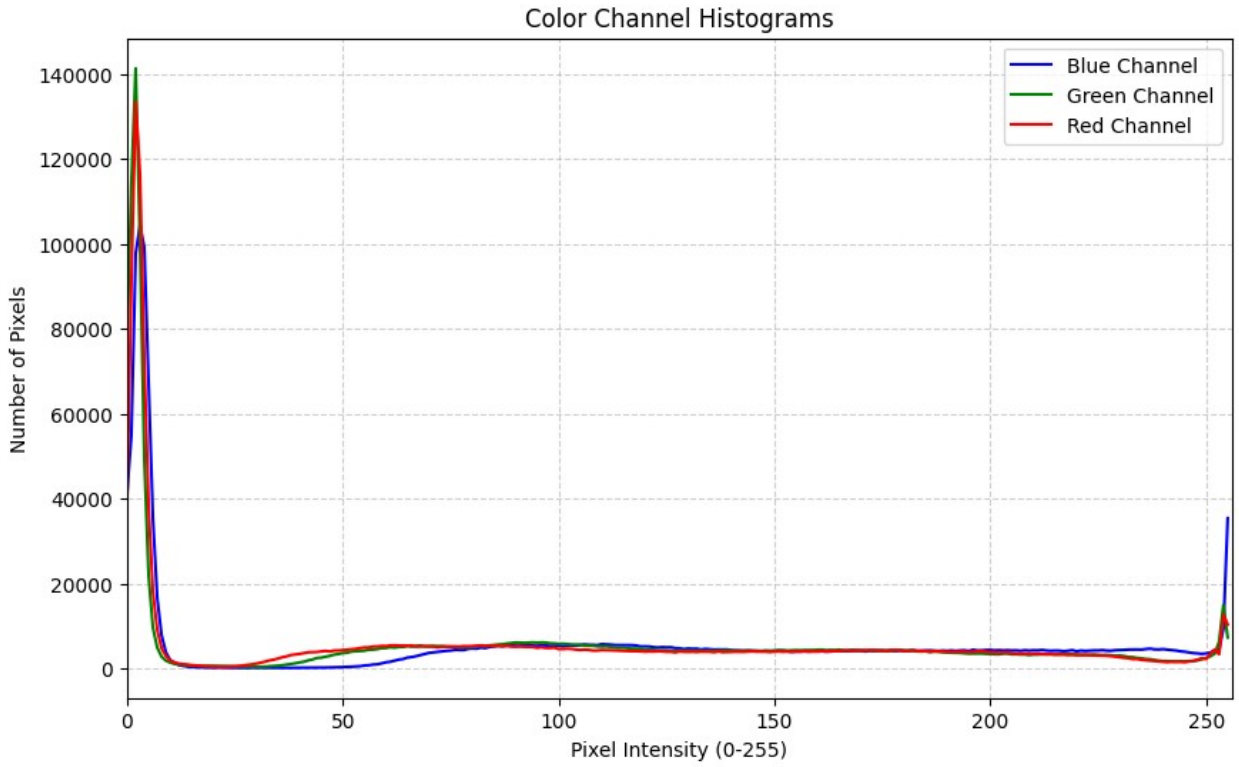
```
Green Channel Isolated:
```



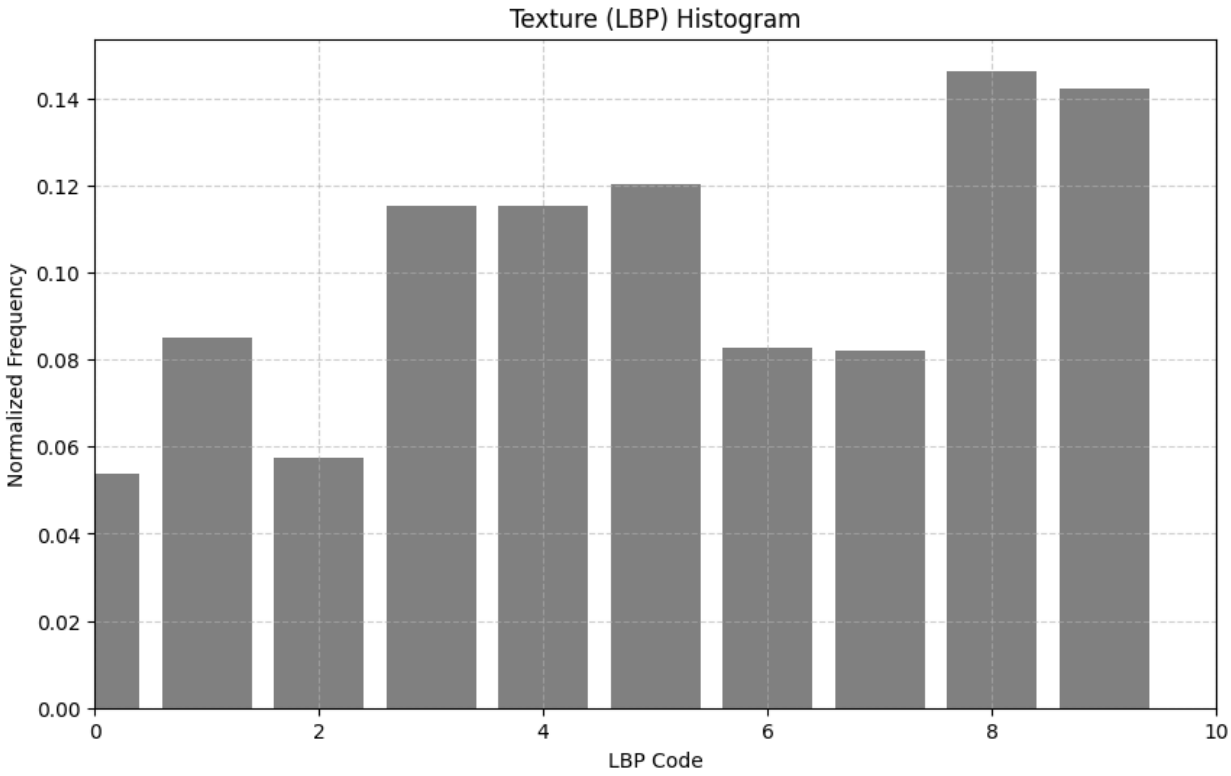
```
Red Channel Isolated:
```



```
========================================

Combined Color Histogram Graph:
```




```
==========================================

Texture (LBP) Histogram Graph:
```


Texture (LBP) Histogram
Normalized Frequency
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0
2
4
6
8
10
LBP Code