```matlab
% date:26/1/2026
% created by siddharth khandelwal

clc;
clear;
close all;

% Read a color image
rgb_img = imread('fort_image.jpeg');

figure;
imshow(rgb_img);
title('Original Color Image');

% Convert the color image to grayscale
gray_img = rgb2gray(rgb_img);
[rows, cols] = size(gray_img);

figure;
imshow(gray_img);
title('Grayscale Image');

% Convert grayscale image into a column vector
img_data = gray_img(:);

% Find pixel values and their probabilities
pixel_values = unique(img_data);
pixel_count  = histc(img_data, pixel_values);
pixel_prob   = pixel_count / sum(pixel_count);

% Create Huffman dictionary based on probabilities
huff_dict = huffmandict(pixel_values, pixel_prob);

% Encode the grayscale image using Huffman coding
encoded_stream = huffmanenco(img_data, huff_dict);

% Decode the Huffman encoded data
decoded_data = huffmandeco(encoded_stream, huff_dict);

% Convert decoded data back into image form
decoded_img = uint8(reshape(decoded_data, rows, cols));

figure;
imshow(decoded_img);
title('Reconstructed Image after Huffman Coding');

% Calculate compression ratio
original_bits   = numel(img_data) * 8;
compressed_bits = length(encoded_stream);

compression_ratio = original_bits / compressed_bits;

fprintf('Original size (bits)   : %d\n', original_bits);
fprintf('Compressed size (bits) : %d\n', compressed_bits);
fprintf('Compression Ratio      : %.2f\n', compression_ratio);
```

```
Original size (bits)   : 6750000
Compressed size (bits) : 6412295
Compression Ratio      : 1.05
```

Original Color Image

Grayscale Image

Reconstructed Image after Huffman Coding