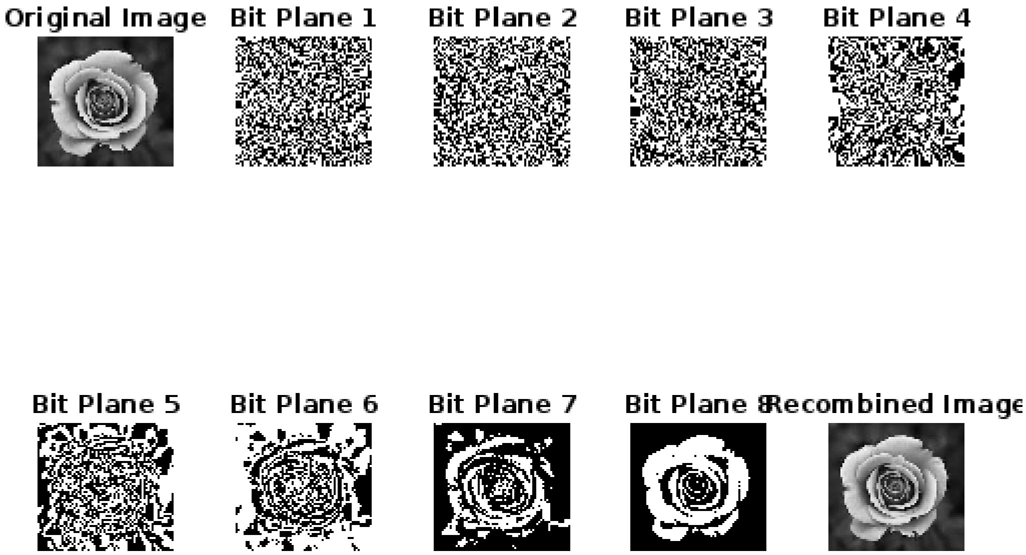
```
% Clearing the output screen
clc; clear;

% Reading image's pixel in c
c = imread('download.jpeg');

% Ensure the image is grayscale
if size(c, 3) == 3
    c = rgb2gray(c);
end

% Convert image to double for bit-plane processing
cd = double(c);

% Extracting all bits one by one from 1st to 8th
c1 = mod(cd, 2);
c2 = mod(floor(cd / 2), 2);
c3 = mod(floor(cd / 4), 2);
c4 = mod(floor(cd / 8), 2);
c5 = mod(floor(cd / 16), 2);
c6 = mod(floor(cd / 32), 2);
c7 = mod(floor(cd / 64), 2);
c8 = mod(floor(cd / 128), 2);

% Combining image again to form equivalent to original grayscale image
cc = c8 * 128 + c7 * 64 + c6 * 32 + c5 * 16 + c4 * 8 + c3 * 4 + c2 * 2 + c1;

% Adjusting figure layout for better spacing
figure;
tiledlayout(2, 5, 'Padding', 'compact', 'TileSpacing', 'compact');

% Plotting original image in first subplot
nexttile;
imshow(c);
title('Original Image');

% Plotting binary image having extracted bit from 1st to 8th
bit_planes = {c1, c2, c3, c4, c5, c6, c7, c8};
for i = 1:8
    nexttile;
    imshow(logical(bit_planes{i})); % Convert to logical for display
    title(['Bit Plane ' num2str(i)]);
end

% Plotting recombined image in the 10th subplot
nexttile;
imshow(uint8(cc));
title('Recombined Image');
```


Original Image
Bit Plane 1
Bit Plane 2
Bit Plane 3
Bit Plane 4
Bit Plane 5
Bit Plane 6
Bit Plane 7
Bit Plane 8
Recombined Image

Published with MATLAB® R2024b