

```

img = imread('/MATLAB Drive/image1.jpg');
if size(img, 3) == 3
    img = rgb2gray(img);
end

image_quantiz = bitshift(img, -3);
image_quantiz = bitshift(image_quantiz, 3);

figure;
subplot(1, 2, 1), imshow(img), title('Original Grayscale Image');
subplot(1, 2, 2), imshow(image_quantiz), title('Quantized Image (32 Levels)');

```

Original Grayscale Image



Quantized Image (32 Levels)



Steps Followed

Step 1 :Loading the Image

`imread('image1.jpg')` loads the image.

`rgb2gray` is used to convert the image to grayscale, which is required for quantization to grayscale levels.

Step 2 :Normalization

The grayscale image is normalized to a range of $[0, 1]$ using `double(grayImg) / 255`. This is to make it easier to handle the quantization process.

Step 3: Quantization via imresize

`imresize(normalizedImg, [32 32], 'nearest')` shrinks the image to simulate the quantization of intensity levels to 32 distinct levels.

After that, `imresize` is used again to scale the intensity levels back to the original size of the image using 'nearest' to avoid interpolation artifacts.

Step 4: Rounding to Quantize

The quantized values are multiplied by 31 to map the intensity values to 32 distinct levels (from 0 to 31). Then, `round` is applied to make sure that the values are properly quantized.

Step 5: Rescaling to 8-bit range

The image is multiplied by $(255 / 31)$ to rescale the intensity levels to the 8-bit grayscale range of $[0, 255]$.

Step 6: Display

The `subplot` function is used to display the original grayscale image and the quantized image side by side for comparison.

Conclusion:

This approach successfully quantizes the image to 32 distinct grayscale levels using the `imresize` function, simulating quantization by resizing the intensity range and scaling it back to the original image dimensions. The result is a posterized version of the image with only 32 shades of gray.