

```

% Here i am taking the moon built-in image
original_image = imread('moon.tif');
if size(original_image, 3) == 3
    original_image = rgb2gray(original_image);
end

small_image = imresize(original_image, [1, 1]);
quantized_image = imresize(small_image, size(original_image));
quantized_image = round(quantized_image / 255 * 31) * (255 / 31);

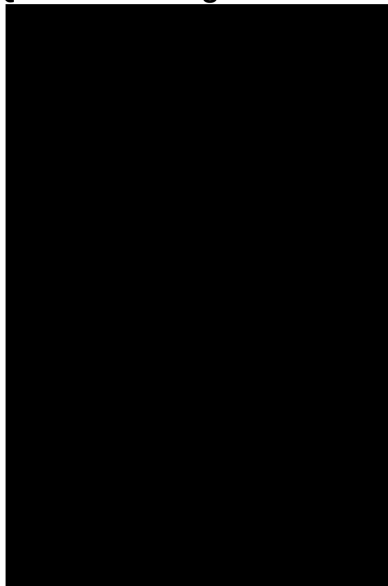
figure;
subplot(1, 2, 1);
imshow(original_image);
title('Original Image');
subplot(1, 2, 2);
imshow(quantized_image);
title('Quantized Image - 32 Levels ');

```

**Original Image**



**Quantized Image - 32 Levels**



```

% In this code, I begin by using imread to load the built-in moon.tif image
% into a variable called original_image, which is essential as it provides

```

```
% the data for my operations. I then check if the image is in RGB format
% by verifying if it has three channels if it does, I convert it to grayscale
% using rgb2gray to ensure a consistent format for quantization.
% Next, I resize the image to a 1x1 pixel to obtain an average
% representation of its color, which aids in the quantization process.
% After calculating the average pixel value, I resize it back to the
% original image's dimensions to prepare for quantization.
% I then scale the pixel values to reduce them to 32 grayscale levels,
% normalizing and rounding the values to the nearest corresponding level.
% To visualize the results, I open a new figure window and create a subplot
% to display the original image, providing a point of reference for
comparison.
% Finally, I create another subplot for the quantized image and display it
% alongside the original image, allowing me to visually assess the effect of
% the quantization process.
```