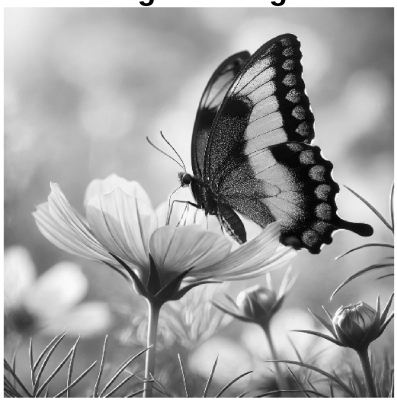
```
% Read the image
I = imread('Images/Butterfly.jpg');

% Convert to grayscale if the image is RGB
if size(I, 3) == 3
    I = rgb2gray(I);
end

% Normalize the image to [0, 1]
I = double(I) / 255;

% Quantize to 32 levels
levels = 32;
I_quantized = round(I * (levels - 1)); % Scale to [0, 31]

% Resize the image (choose a small scale factor to control the detail)
scale_factor = 0.1; % Adjust as necessary
I_resized = imresize(I_quantized, scale_factor, 'nearest');

% Resize back to original size
I_quantized_resized = imresize(I_resized, size(I), 'nearest');

% Convert to uint8
I_quantized_resized = uint8(I_quantized_resized * (255 / (levels - 1))); %
Scale back to [0, 255]

% Display the original and quantized images
figure;
subplot(1, 2, 1), imshow(uint8(I * 255)), title('Original Image');
subplot(1, 2, 2), imshow(I_quantized_resized), title('Quantized
Image (32 Levels)');
```

Original Image



Quantized Image (32 Levels)

Steps:

1. **Load the Image**: The `imread` function is used to load the image into MATLAB.
2. **Convert to Grayscale**: The `rgb2gray` function converts a color image to grayscale. If the image is already in grayscale, this step is skipped.
3. **Downsize the Image**: The image is resized to a much smaller version, reducing its dimensions to 1/32 of the original size using `imresize`. This reduces the color depth by averaging pixel values.
4. **Upscale to Original Size**: The downsized image is then resized back to its original dimensions. This process retains the reduced number of gray levels while smoothing the pixel values.
5. **Display the Results**: The original and quantized images are displayed side by side for comparison.

This method doesn't specifically set the image to exactly 32 levels; instead, it reduces the number of gray levels through resizing. The resulting image will have fewer than 256 gray levels, and typically, the number will approximate around 32 levels, but may vary based on the image content and resizing technique.