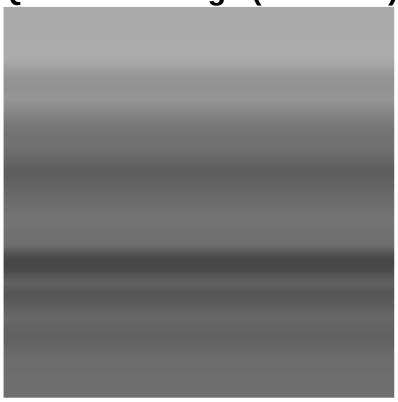
```
% Step 1: Load the image
img = imread('cameraman.tif');  % Using a built-in MATLAB image
if size(img, 3) > 1
    img = rgb2gray(img);  % Convert to grayscale if it's a color image
end

% Step 2: Normalize the image to [0, 1] range
img_normalized = double(img) / 255;

% Step 3: Resize the image to a single column
[height, width] = size(img_normalized);
img_column = imresize(img_normalized, [height * width, 1]);

% Step 4: Resize the column to 32 rows
img_quantized_column = imresize(img_column, [32, 1]);

% Step 5: Resize back to original dimensions
img_quantized = imresize(img_quantized_column, [height, width]);

% Step 6: Scale back to [0, 255] range and convert to uint8
img_quantized = uint8(img_quantized * 255);

% Display results
figure;
subplot(1,2,1), imshow(img), title('Original Image');
subplot(1,2,2), imshow(img_quantized), title('Quantized Image (32 levels)');
```

Original Image

Quantized Image (32 levels)



```
% Calculate and display PSNR and SSIM
psnr_value = psnr(img_quantized, img);
ssim_value = ssim(img_quantized, img);

fprintf('Peak Signal-to-Noise Ratio (PSNR): %.2f dB\n', psnr_value);
```

```
Peak Signal-to-Noise Ratio (PSNR): 13.25 dB
```

```
fprintf('Structural Similarity Index (SSIM): %.4f\n', ssim_value);
```

```
Structural Similarity Index (SSIM): 0.4570
```

```
% Display unique gray levels
unique_levels = unique(img_quantized);
fprintf('Number of unique gray levels: %d\n', length(unique_levels));
```

```
Number of unique gray levels: 73
```

```
fprintf('Unique gray levels: ');
```

```
Unique gray levels:
```

```
fprintf('%d ', unique_levels);
```

```
71 72 73 74 75 77 80 84 87 88 89 90 91 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 11
```

```
fprintf('\n');
```