```
function main()
    % MAIN FUNCTION: reads image, applies histogram equalization,
 displays result

    % Try reading image from file
    try
        img = imread('car1.png');
    catch
        % If image not found, use a sample 8x8 grayscale image
        disp('car1.png not found. Using 8x8 test image.');
        img = uint8(randi([50 150], 8, 8));
    end

    % Apply custom histogram equalization
    output_img = customHistEq_Process(img);

    % Display results
    figure;
    subplot(1,2,1), imshow(img), title('Original Image');
    subplot(1,2,2), imshow(output_img), title('Equalized Image');
end


function output_img = customHistEq_Process(img)
    % Handles both RGB and grayscale images

    if ndims(img) == 3 && size(img,3) == 3
        % Convert RGB to YCbCr and equalize luminance only
        ycbcr = rgb2ycbcr(img);
        ycbcr(:,:,1) = perform_he(ycbcr(:,:,1));
        output_img = ycbcr2rgb(ycbcr);
    else
        % Grayscale image
        output_img = perform_he(img);
    end
end


function equalized = perform_he(I)
    % HISTOGRAM EQUALIZATION FOR GRAYSCALE IMAGE

    I = uint8(I);
    [rows, cols] = size(I);
    num_pixels = rows * cols;

    % Step 1: Compute histogram
    counts = zeros(1,256);
    for i = 0:255
        counts(i+1) = sum(I(:) == i);
    end

    % Step 2: Compute cumulative distribution function (CDF)
```

```
    cdf = cumsum(counts);
    cdf_min = min(cdf(cdf > 0));

    % Step 3: Create intensity mapping
    map = round((cdf - cdf_min) / (num_pixels - cdf_min) * 255);
    map(map < 0) = 0;
    map(map > 255) = 255;

    % Step 4: Apply mapping to image
    equalized = map(double(I) + 1);
    equalized = reshape(equalized, rows, cols);
    equalized = uint8(equalized);
end
```

Published with MATLAB® R2021a