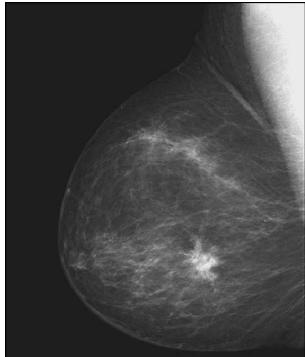


Notes and Outputs - Lab 3

Task 1 - Contrast enhancement

```
imfinfo('assets/breastXray.tif')
f = imread('assets/breastXray.tif');
imshow(f)
```

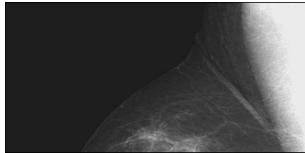
f 571×482 uint8



```
f(3,10) % print the intensity of pixel(3,10)
imshow(f(1:241,:)) % display only top half of the image
```

Output:

```
ans =
28
```



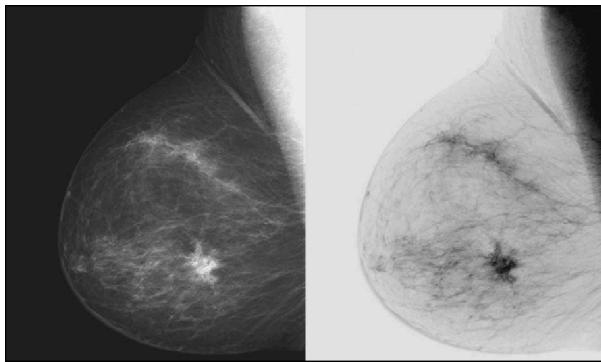
```
[fmin, fmax] = bounds(f(:))
```

Output:

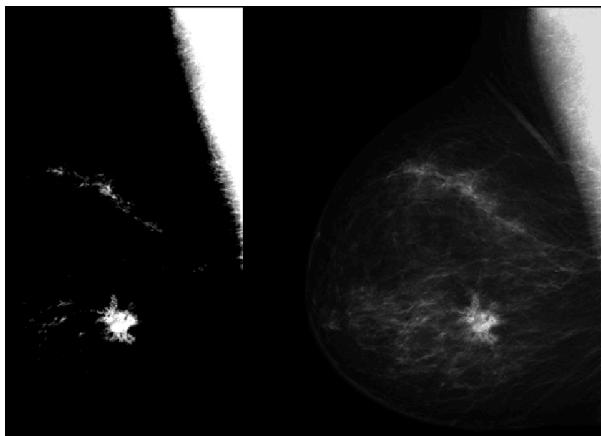
```
fmin =
21
```

```
fmax =
255
```

```
g1 = imadjust(f, [0 1], [1 0])
montage({f, g1})
```



```
g2 = imadjust(f, [0.5 0.75], [0 1]); % gray scale range between 0.5 and 0.75  
mapped to the full range.  
g3 = imadjust(f, [ ], [ ], 2); gamma correct with gamma = 2.0, [ ] is the  
same as [0 1] by default  
montage({g2,g3})
```



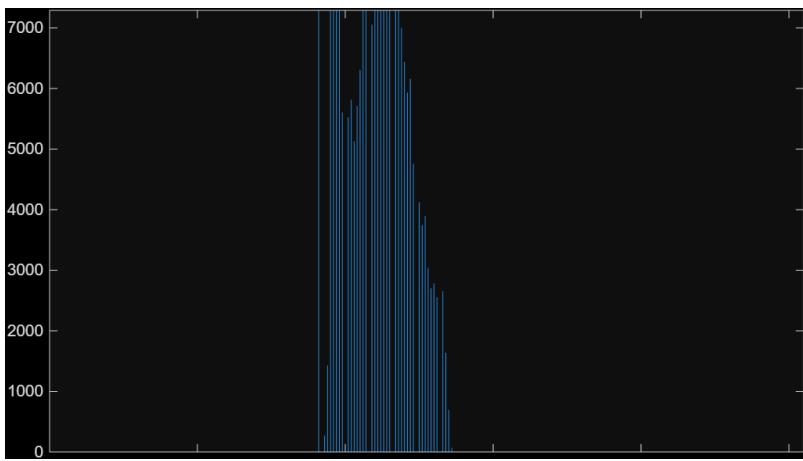
Task 2: Contrast-stretching transformation

```
f = imread('assets/bonescan-front.tif');  
r = double(f); % uint8 to double conversion  
k = mean2(r); % find mean intensity of image  
E = 0.9;  
s = 1 ./ (1.0 + (k ./ (r + eps)) .^ E);  
g = uint8(255*s);  
imshowpair(f, g, "montage")
```



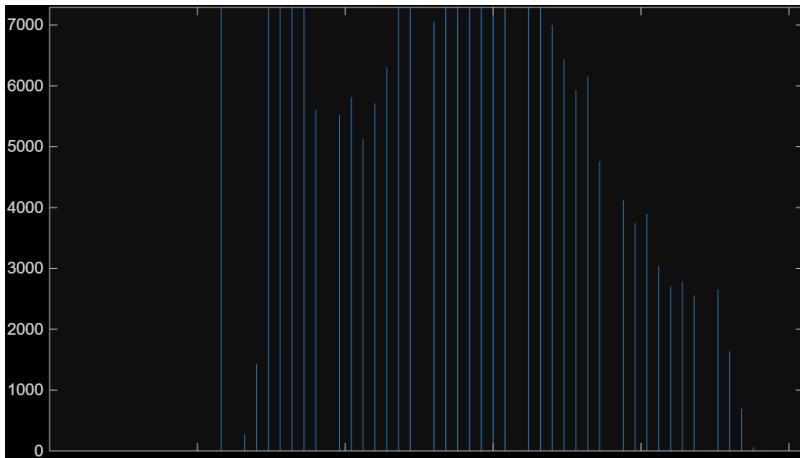
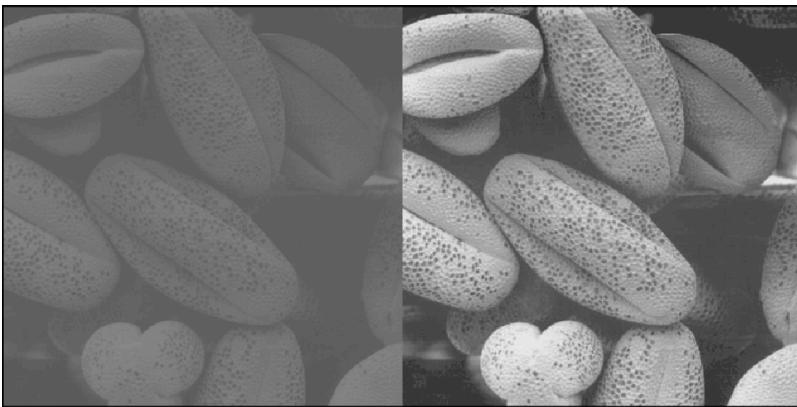
Contrast Enhancement using Histogram

```
f=imread('assets/pollen.tif');  
imhist(f); % calculate and plot the histogram
```



```
g=imadjust(f,[0.3 0.55]); % stretch the intensity between 0.3 and 0.55  
montage({f, g}) % display list of images side-by-side  
  
imhist(g);
```

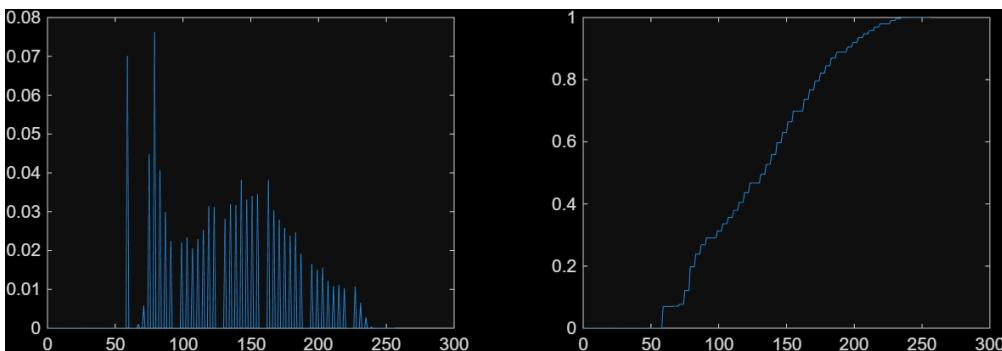
Task 3: Contrast Enhancement using Histogram



```
g_pdf = imhist(g) ./ numel(g); % compute PDF  
g_cdf = cumsum(g_pdf); % compute CDF
```

g_cdf	256x1 double	256x1	double
g_pdf	256x1 double	256x1	double

```
subplot(1,2,1) % plot 1 in a 1x2 subplot  
plot(g_pdf)  
subplot(1,2,2) % plot 2 in a 1x2 subplot  
plot(g_cdf)
```

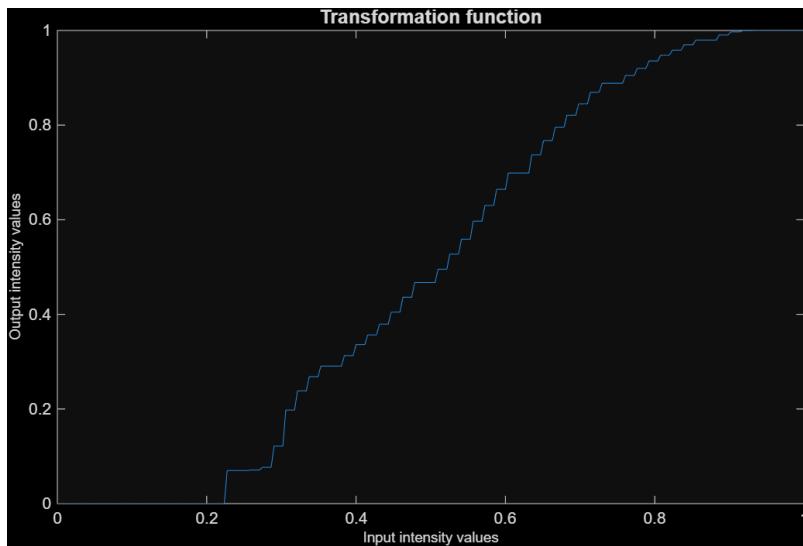


```
x = linspace(0, 1, 256); % x has 256 values equally spaced  
% .... between 0 and 1  
figure  
plot(x, g_cdf)  
axis([0 1 0 1]) % graph x and y range is 0 to 1
```

```

set(gca, 'xtick', 0:0.2:1) % x tick marks are in steps of 0.2
set(gca, 'ytick', 0:0.2:1)
xlabel('Input intensity values', 'fontsize', 9)
ylabel('Output intensity values', 'fontsize', 9)
title('Transformation function', 'fontsize', 12)

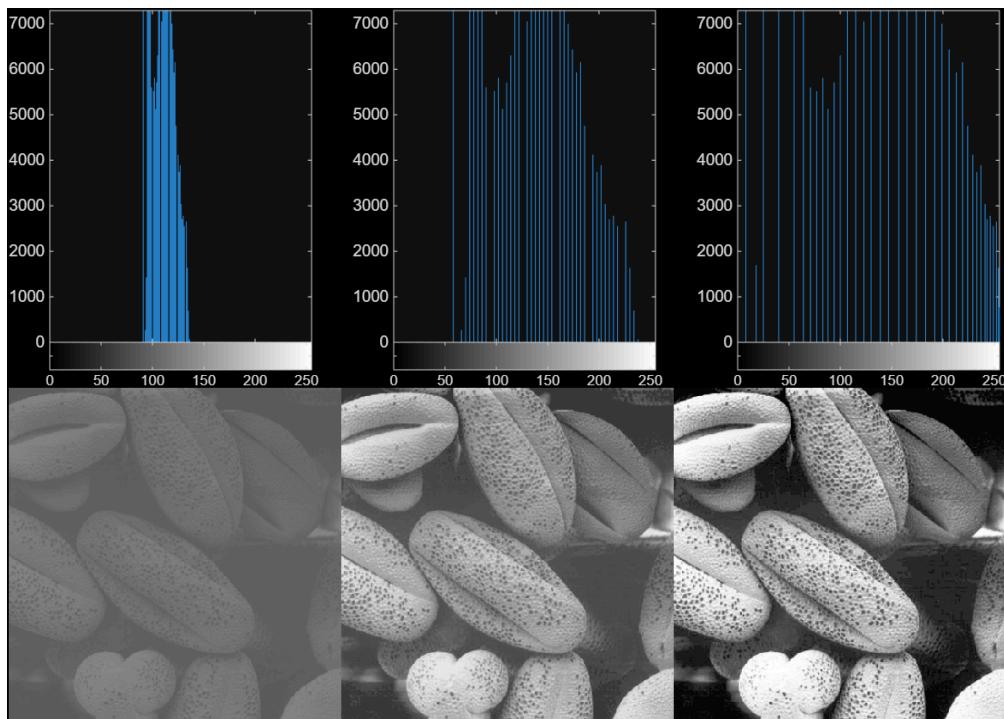
```



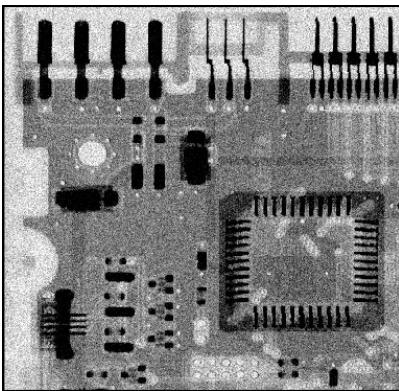
```

h = histeq(g,256); % histogram equalize g
close all
montage({f, g, h}, 'Size',[1 3])
figure;
subplot(1,3,1); imhist(f);
subplot(1,3,2); imhist(g);
subplot(1,3,3); imhist(h);

```



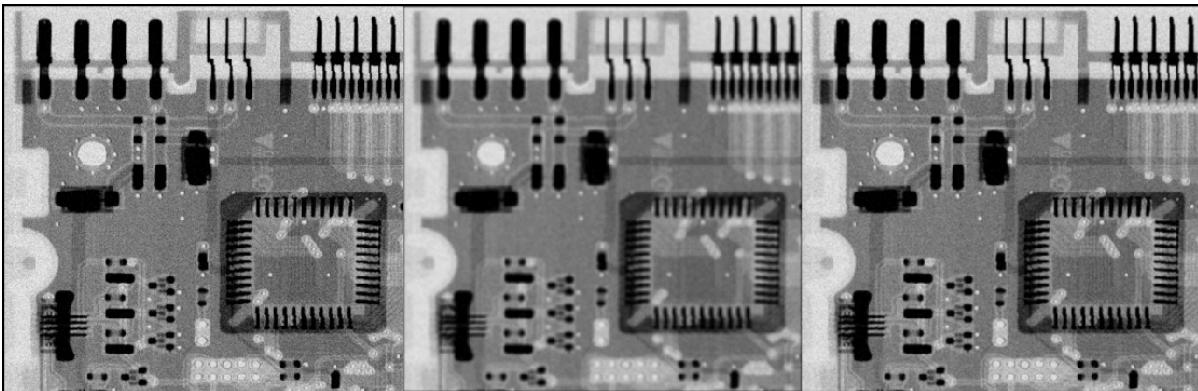
Task 4 - Noise reduction with lowpass filter



```
w_box = fspecial('average', [9 9])
w_gauss = fspecial('Gaussian', [7 7], 1.0)
```

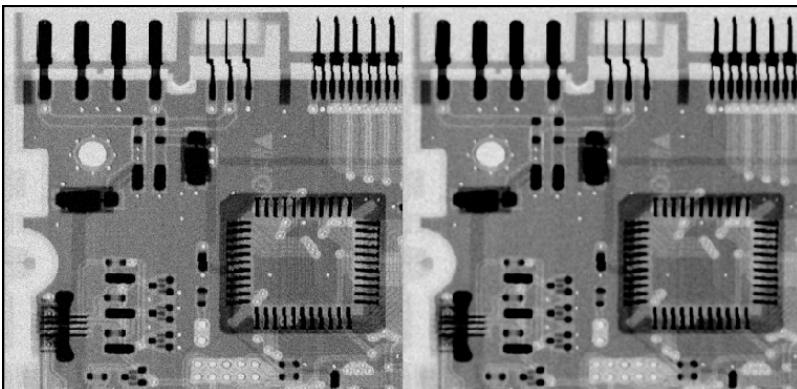
w_box	<i>9×9 double</i>	9×9	double
w_gauss	<i>7×7 double</i>	7×7	double

```
g_box = imfilter(f, w_box, 0);
g_gauss = imfilter(f, w_gauss, 0);
figure
montage({f, g_box, g_gauss})
```



Task 5 - Median Filtering

```
g_median = medfilt2(f, [7 7], 'zero');
figure; montage({f, g_median})
```



Understanding the Additional filters

```
% r and c - row and column are natural numbers
% sig > 0, likely 0.5
% 0 < alpha < 1
% length > 0, natural, default is 9
% theta, from -180 to 180 or 0 to 360
-----
f_average = fspecial('average', [r c])
f_disk = fspecial('disk', r)
f_gauss = fspecial('Gaussian', [r c], sig)
f_laplacian = fspecial('laplacian', alpha)
f_log = fspecial('log', [r c], sig)
f_motion = fspecial('motion', len, theta)
f_prewitt = fspecial('prewitt')
f_sobel = fspecial('sobel')
f_unsharp = fspecial('unsharp', alpha)
```

Task 6 - Sharpening the image with Laplacian, Sobel and Unsharp filters

```
f = imread('moon.tif');
imshow(f)
```



```
f_moon = imread('assets/moon.tif');
w_lap = fspecial('laplacian', 0.2);
edges = imfilter(f_moon, w_lap);
g_sharp = f_moon - edges;
imshowpair(f_moon, g_sharp, 'montage');
```



Task 7a - Contrast Improvement

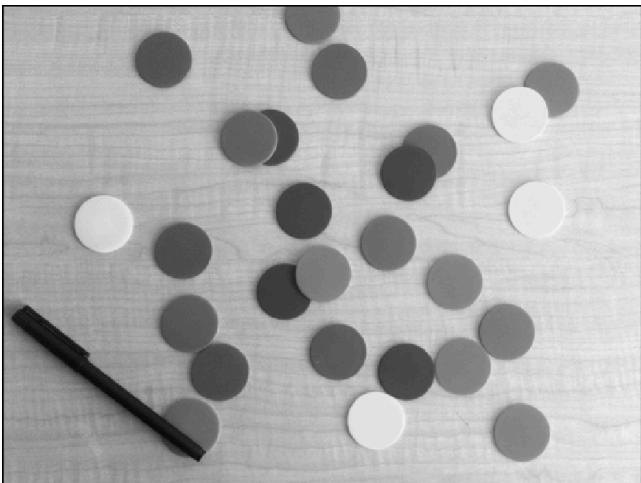
```
f_tree = imread('assets/lake&tree.png');
img = imread('assets/lake&tree.png');
g_enhanced = adapthisteq(img);
imshowpair(f_tree,g_enhanced,'montage');
```



Task 7b - Sobel Filter

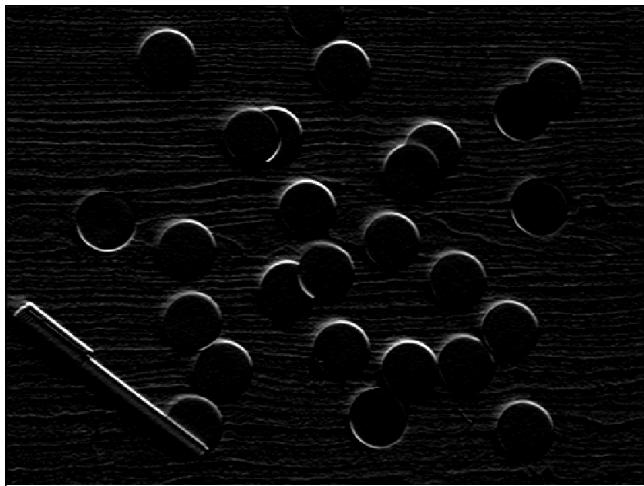
original image:

```
c = imread('assets\circles.tif');
imshow(c)
```



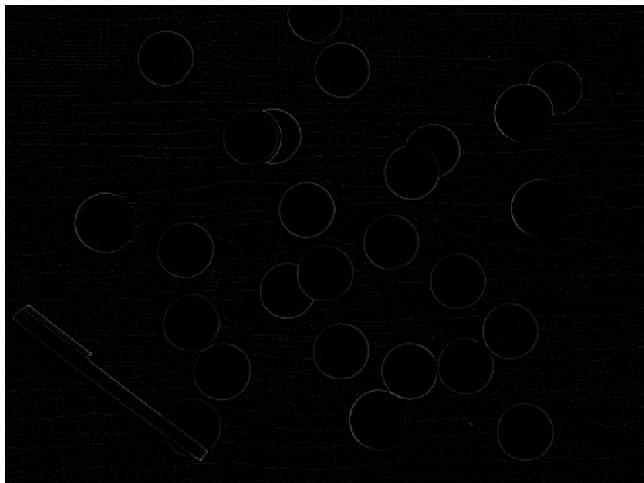
Attempt 1: Using only sobel:

```
w_sobel = fspecial('sobel');
imshow(imfilter(c, w_sobel, 0))
```



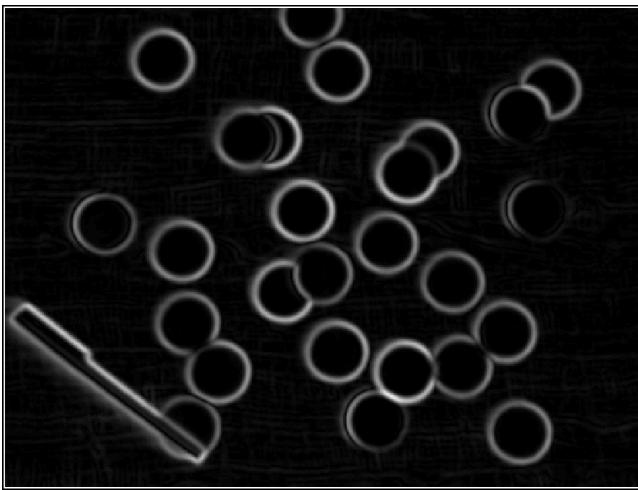
Attempt 2: Using only laplacian

```
w_laplacian = fspecial('laplacian', 0.5);
imshow(imfilter(c, w_laplacian, 0))
```



Attempt 3: Using a combination of sobel and and gaussian smoothing

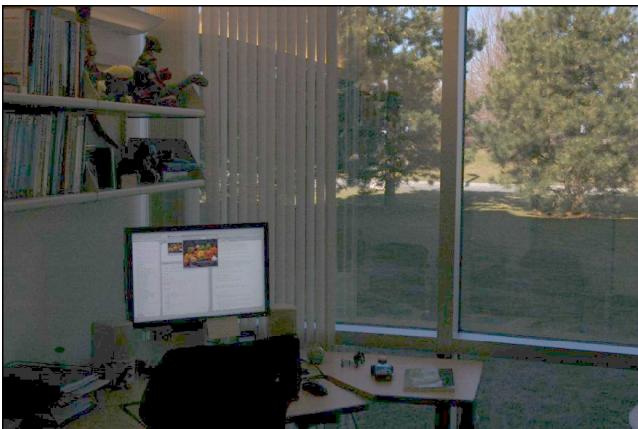
```
w_smooth = imgaussfilt(c, 2);
w_sobel = fspecial('sobel');
edges_v = imfilter(double(w_smooth), w_sobel);
edges_h = imfilter(double(w_smooth), w_sobel');
magnitude = sqrt(edges_v.^2 + edges_h.^2);
imshow(uint8(magnitude));
```



Task 7c - Sobel Filter

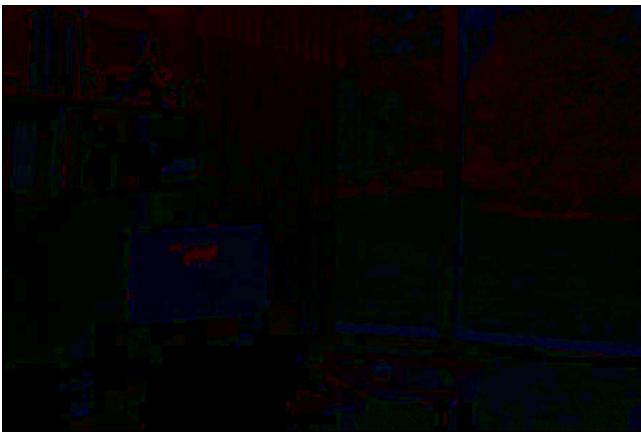
```
rgb = imread('assets/office.jpg');
imshow(rgb)
```

Original Image



Attempt 1:

```
lab = rgb2lab(rgb);
lab(:,:,:,1) = adapthisteq(lab(:,:,:,1), 'ClipLimit', 0.02, 'Distribution',
'rayleigh');
saturation_factor = 1.3;
lab(:,:,:,2) = lab(:,:,:,2) * saturation_factor;
lab(:,:,:,3) = lab(:,:,:,3) * saturation_factor;
enhanced_rgb = lab2rgb(lab);-
figure;
imshow(enhanced_rgb)
```



Attempt 2:

```
% 1. Read and convert to double
f = im2double(imread('office.jpg'));

% 2. Reduce Noise (Median Filtering)
% Effective for "salt and pepper" noise or digital sensor grain.
% We apply it to each channel to remove color spikes.
f_denoised = f;
for i = 1:3
    f_denoised(:,:,i) = medfilt2(f(:,:,i), [3 3]);
end

% 3. Smoothen the Image (Gaussian Filter)
% A light Gaussian blur helps blend textures after denoising.
w_gauss = fspecial('Gaussian', [5 5], 0.5);
f_smooth = imfilter(f_denoised, w_gauss, 'replicate');

% 4. Convert to LAB to isolate Intensity
lab = rgb2lab(f_smooth);
L = lab(:,:,1) / 100; % Normalize L to 0-1

% 5. Increase Exposure and Contrast
% We use Gamma < 1 to lift shadows and stretchlim to widen the range.
% A gamma of 0.4 provides a very bright exposure boost.
L_final = imadjust(L, stretchlim(L), [], 0.4);

% 6. Reassemble and Convert back
lab(:,:,1) = L_final * 100;
g = lab2rgb(lab);

% 7. Final Safety Clamp to prevent glitches
g = max(0, min(1, g));

% Display
figure;
montage({f, g})
title('Original (Left) vs Denoised, Smoothened, and High Exposure (Right)')
```

