Image Processing Homework 2
Daymon Wu

Problem 1:
What effect would setting to zero the lower-order bit planes have on the histogram of an image in general? Why? What would be the effect on the histogram if we set to zero the higher-order bit planes instead? Why? Write a Matlab program to verify your argument with LENNA.JPG. Does the outcome of your experiment support your argument? You may use any built-in functions for this experiment.

Setting the lower order bit planes to zero would make the histogram of the image more coarse while reducing the image's and detail. This is because since lower order bits do not play a role in affecting the contrast or brightness it will not make the image any worse because this is in black and white
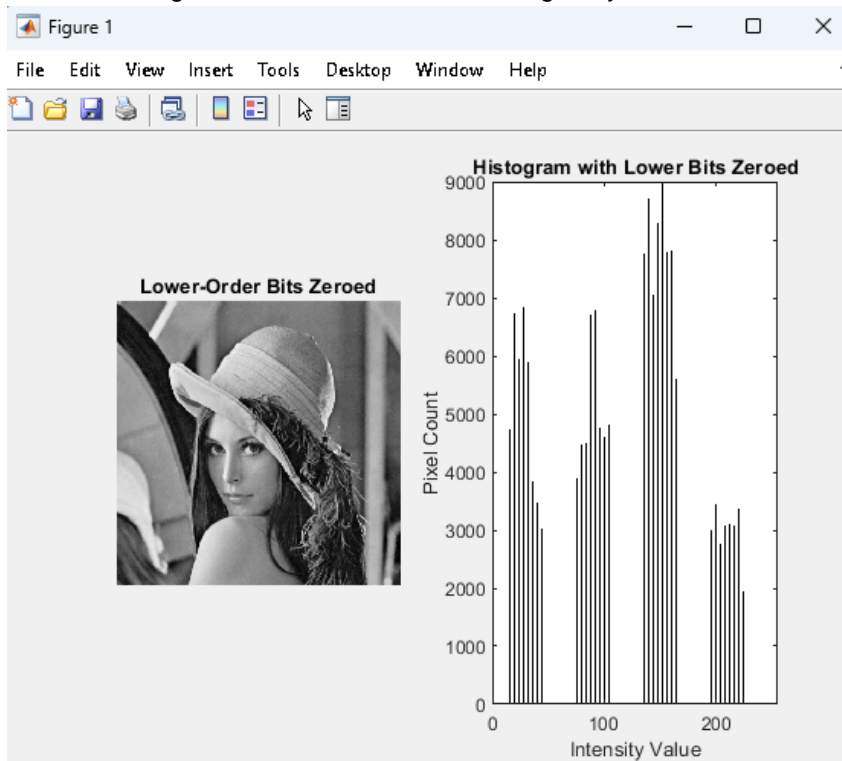

```matlab
imagePath = 'C:\Users\MGSummer1\Desktop\MATLAB\HW2\LENNA.jpg';
I = imread(imagePath);
if size(I, 3) == 3
    I = rgb2gray(I);
end
I_low = I;
for k = [1, 2]
    I_low = bitand(I_low, bitcmp(2^(k-1), 'uint8'));
end
subplot(1,2,1);
imshow(I_low);
title('Lower-Order Bits Zeroed');
counts = hist(double(I_low(:)), 0:255);
subplot(1,2,2);
bar(0:255, counts, 'k');
```

```matlab
xlim([0 255]);
title('Histogram with Lower Bits Zeroed');
xlabel('Intensity Value');
ylabel('Pixel Count');
```

Setting the higher order bit planes to zero might have an intense effect on the histogram and image. When those bits are zero the values of the pixel are reduced which makes the image more darker. We might lose most of the image this way. This is because these bits I think play a big role in contrast to the image.
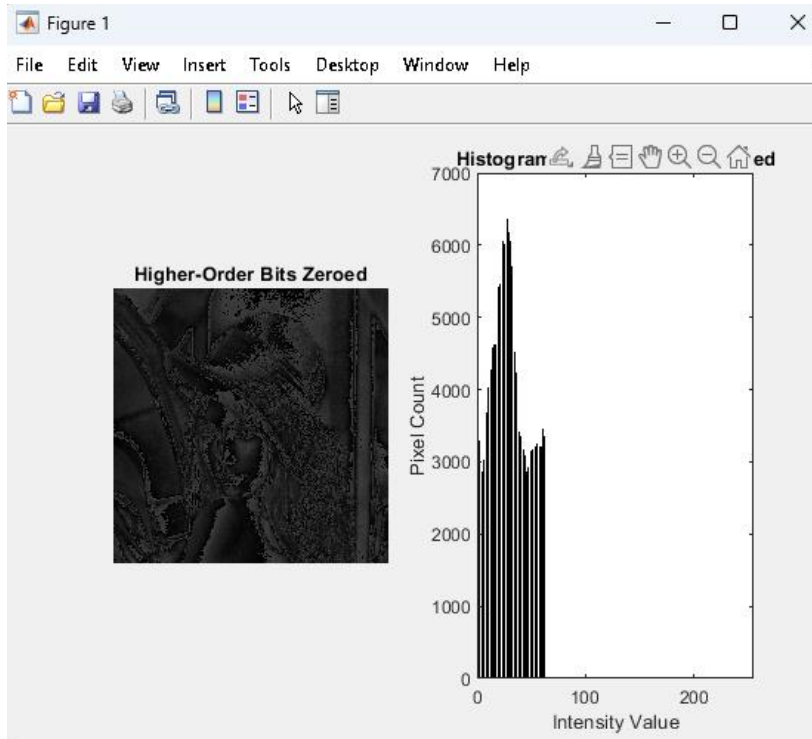

```matlab
imagePath = 'C:\Users\MGSummer1\Desktop\MATLAB\HW2\LENNA.jpg';
I = imread(imagePath);
if size(I, 3) == 3
    I = rgb2gray(I);
end
I_high = I;
for k = [7, 8]
    I_high = bitand(I_high, bitcmp(2^(k-1), 'uint8'));
end
subplot(1,2,1);
imshow(I_high);
title('Higher-Order Bits Zeroed');
counts = hist(double(I_high(:)), 0:255);
subplot(1,2,2);
bar(0:255, counts, 'k');
xlim([0 255]);
title('Histogram with Higher Bits Zeroed');
xlabel('Intensity Value');
```

```matlab
ylabel('Pixel Count');
```
File to run all functions:
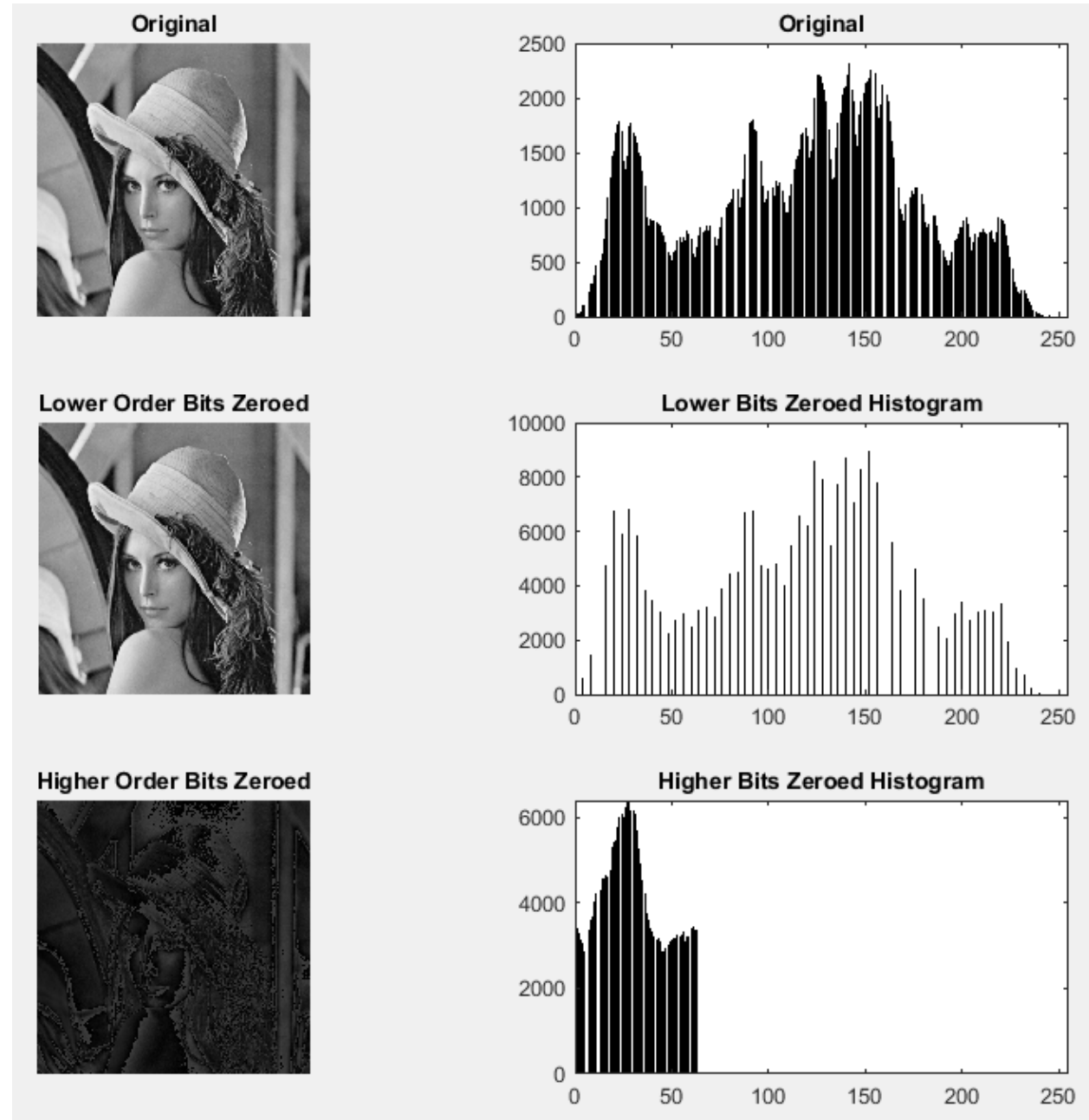

```matlab
imagePath = 'C:\Users\MGSummer1\Desktop\MATLAB\HW2\LENNA.jpg';
I = imread(imagePath);
if size(I, 3) == 3
    I = rgb2gray(I);
end
I_low = I;
for k = [1, 2]
    I_low = bitand(I_low, bitcmp(2^(k-1), 'uint8'));
end
```

```matlab
I_high = I;
for k = [7, 8]
    I_high = bitand(I_high, bitcmp(2^(k-1), 'uint8'));
end
figure;
subplot(3,2,1);
imshow(I);
title('Original');
subplot(3,2,2);
counts_original = hist(double(I(:)), 0:255);
bar(0:255, counts_original, 'k');
xlim([0 255]);
title('Original');
subplot(3,2,3);
imshow(I_low);
title('Lower Order Bits Zeroed');
subplot(3,2,4);
counts_low = hist(double(I_low(:)), 0:255);
bar(0:255, counts_low, 'k');
xlim([0 255]);
title('Lower Bits Zeroed Histogram');
subplot(3,2,5);
imshow(I_high);
title('Higher Order Bits Zeroed');
subplot(3,2,6);
counts_high = hist(double(I_high(:)), 0:255);
bar(0:255, counts_high, 'k');
xlim([0 255]);
title('Higher Bits Zeroed Histogram');
```

Outcome:
Yes, the experiment from my code does support my hypothesis.

Problem 2:
Implement your own Matlab function which performs RGB histogram matching on color images. Your function takes two color images as the input, the first one being the image to be remapped and the second one being the target image. In your function, you should not make function calls to predefined histogram related functions. The output of the function is the RGB remapped image. Test your function with Atlas-Mercury.png and techno-trousers.png. To verify the correctness of the output, compare your result with the output of imhistmatch(.) in a "scientific" way. Include the comparison in the report.

```matlab
function rgbHistogram()
    sourceImagePath = 'C:\Users\MGSummer1\Desktop\MATLAB\HW2\Atlas-Mercury.png';
    targetImagePath = 'C:\Users\MGSummer1\Desktop\MATLAB\HW2\techno-trousers.png';
    sourceImage = imread(sourceImagePath);
    targetImage = imread(targetImagePath);
    sourceImage = imresize(sourceImage, [size(targetImage, 1), size(targetImage, 2)]);
    matchedImage = customHistMatch(sourceImage, targetImage);
    figure;
    subplot(1,3,1), imshow(sourceImage), title('Source Image');
    subplot(1,3,2), imshow(targetImage), title('Target Image');
    subplot(1,3,3), imshow(matchedImage), title('Matched Image');
    mseValue = calculateMSE(matchedImage, targetImage);
    disp(['MSE Value: ', num2str(mseValue)]);
end
function matchedImage = customHistMatch(sourceImage, targetImage)
    matchedImage = zeros(size(sourceImage), class(sourceImage));
    for channel = 1:3
        [sourceCDF, map] = histMatchMapping(sourceImage(:,:,channel),
targetImage(:,:,channel));
        matchedImage(:,:,channel) = map(double(sourceImage(:,:,channel))+1);
    end
end
function [sourceCDF, map] = histMatchMapping(sourceChannel, targetChannel)
    sourceHist = imhist(sourceChannel);
    targetHist = imhist(targetChannel);
    sourceCDF = cumsum(sourceHist) / numel(sourceChannel);
    targetCDF = cumsum(targetHist) / numel(targetChannel);
    map = zeros(256,1,'uint8');
    for idx = 1:256
        [~, closest] = min(abs(sourceCDF(idx) - targetCDF));
        map(idx) = closest - 1;
    end
end
function hist = imhist(channel)
    hist = zeros(256,1);
    for pixelValue = 0:255
        hist(pixelValue+1) = sum(channel(:) == pixelValue);
    end
end
function mseValue = calculateMSE(image1, image2)
```

```matlab
    if any(size(image1) ~= size(image2))
        error('Images must be of the same size.');
    end

    image1 = double(image1);
    image2 = double(image2);
    diffSquared = (image1 - image2).^2;
    mseValue = sum(diffSquared(:)) / numel(image1);
end
```
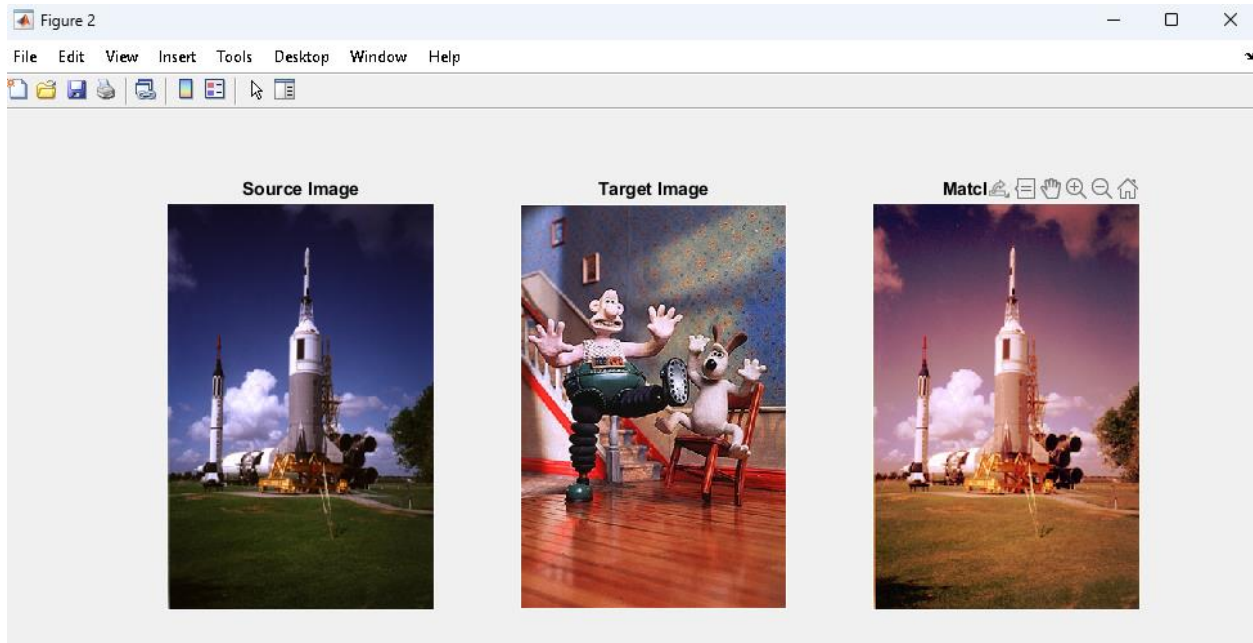
Comparison:

This is a comparison of the source image to the matched image after the source image was altered by the colors in the target image.

Problem 3:
In a given application an averaging mask is applied to input images to reduce noise, and then a Laplacian mask is applied to enhance small details. Would the result be the same if the order of these operations were reversed? Why? Write a Matlab program to verify your argument with LENNA.JPG. How does the outcome of your experiment support your argument? You may use any built-in functions for this experiment.

Hypothesis:
I believe that the order in which the masks are applied matters because it is like baking a cake. If you smoothen out a cake then put it in the oven it will be a smooth cake. But if you don't smoothen the cake before you bake it the cake will have more texture/lines.

Averaging the mask then applying the Laplacian mask:
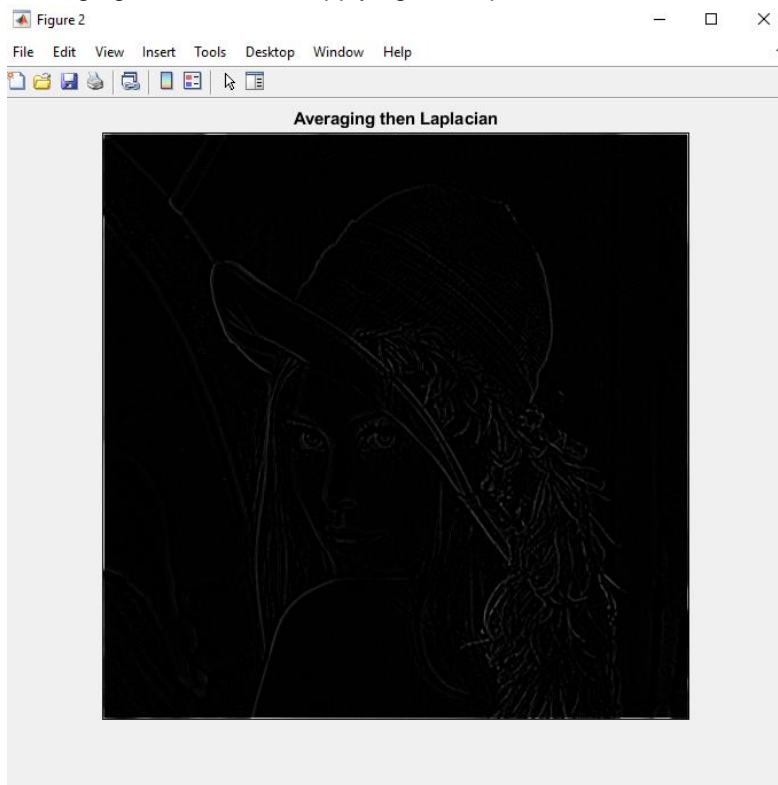
Averaging the mask to reduce the noise can smoothen the image will blur the image by reducing some qualities of the image. By adding the Laplacian filter second, it enhanced the edges by getting the regions that still remain from the blurring. By doing the averaging then the laplacian mask we have an image that seems to have a consistent line strength throughout the image.

Applying the Lapacian mask then averaging the mask:
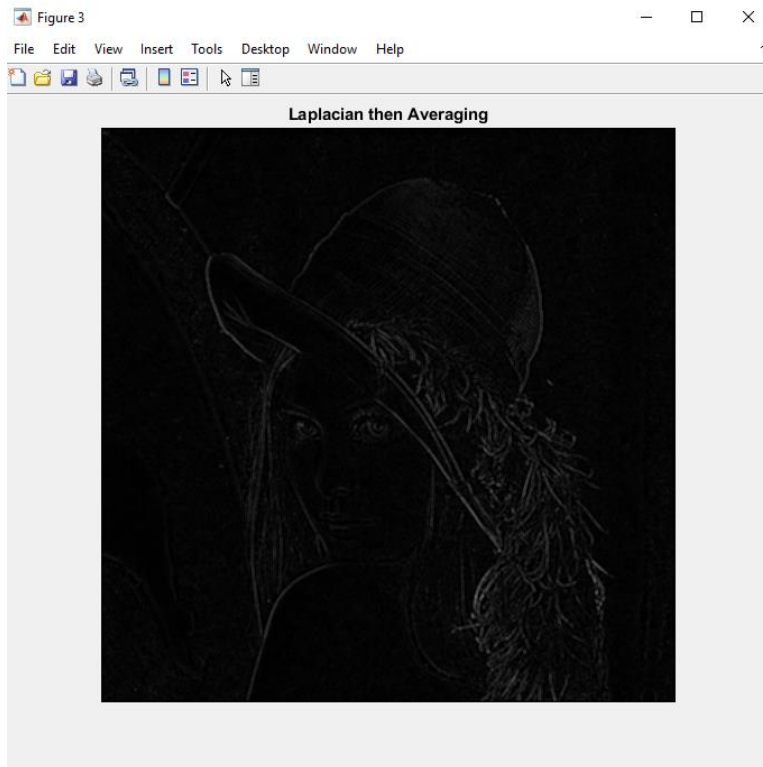
When we apply the Lapacian mask first the edges and the details become more noticeable. Then after applying the averaging filter everything smoothes out which might make the image look more dull or less sharp. But the lines are less consistent making the image pop out more or more realistic because lines are more dynamic.

```
img = imread('LENNA.JPG');
if size(img, 3) == 3
    img = rgb2gray(img);
end
avgFilter = (1/9) * ones(3, 3);
lapFilter = [0 -1 0; -1 4 -1; 0 -1 0];
smoothed = applyFilterManual(img, avgFilter);
avgLap = applyFilterManual(smoothed, lapFilter);
enhanced = applyFilterManual(img, lapFilter);
lapAvg = applyFilterManual(enhanced, avgFilter);
figure, imshow(img), title('Original Image');
figure, imshow(avgLap), title('Averaging then Laplacian');
figure, imshow(lapAvg), title('Laplacian then Averaging');
function output = applyFilterManual(img, filter)
    [rows, cols] = size(img);
    [fRows, fCols] = size(filter);
```

```matlab
    output = zeros(rows, cols, 'like', img);
    padRow = floor(fRows / 2);
    padCol = floor(fCols / 2);

    for i = (1+padRow):(rows-padRow)
        for j = (1+padCol):(cols-padCol)
            segment = double(img((i-padRow):(i+padRow), (j-padCol):(j+padCol)));
            filteredValue = sum(sum(segment .* filter));
            output(i, j) = filteredValue;
        end
    end

    if isinteger(img)
        output = uint8(output);
    end
end
```

Resources:

http://www.ece.northwestern.edu/local-apps/matlabhelp/techdoc/ref/size.html

https://www.educba.com/size-function-in-matlab/

https://www.youtube.com/watch?v=7L32uA3N5Z8

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ece-research.unm.edu/hayat/ece533/Sol_Assignment04_08.pdf

https://mathworks.com/matlabcentral/answers/77323-gray-image-to-8-bit-planes-using-bit-plane-slicing

https://www.researchgate.net/post/What-effect-would-have-to-set-to-zero-the-lower-order-bit-planes-on-the-histogram-of-an-image-in-general

https://www.youtube.com/watch?v=rnWDeG0RFDY

https://www.youtube.com/watch?v=yoouCCFjCE8

https://www.mathworks.com/help/matlab/ref/rgb2gray.html

https://www.mathworks.com/matlabcentral/answers/53657-how-to-implement-a-function-to-apply-the-3x3-average-filter-to-a-gray-scale-image

https://www.youtube.com/watch?v=rxjgLakRqcw

https://www.youtube.com/watch?v=dvqldER3vBA

https://www.geeksforgeeks.org/laplacian-filter-using-matlab/

https://www.mathworks.com/help/images/ref/locallapfilt.html

https://www.geeksforgeeks.org/laplacian-of-gaussian-filter-in-matlab/

https://www.geeksforgeeks.org/laplacian-filter-using-matlab/

https://www.youtube.com/watch?v=4Ho9zgRSHPY

https://www.mathworks.com/help/signal/ug/filter-implementation-and-analysis.html

https://www.mathworks.com/help/matlab/ref/filter.html

https://www.mathworks.com/matlabcentral/answers/1759825-how-to-display-a-result

https://www.mathworks.com/help/matlab/matlab_external/displaying-output-in-matlab-command-window.html

https://www.mathworks.com/help/matlab/ref/disp.html

https://www.mathworks.com/matlabcentral/answers/1703405-function-definitions-in-a-script-must-appear-at-the-end-of-the-file-move-all-statements-after-the

https://www.mathworks.com/help/signal/ug/the-filter-function.html

https://www.mathworks.com/help/matlab/ref/floor.html
https://www.mathworks.com/matlabcentral/answers/599011-floor-function-with-a-space-giving-an-array-output
https://www.mathworks.com/matlabcentral/answers/29837-processing-files-using-a-for-loop
https://www.youtube.com/watch?v=MbbWlSmUNgk
https://www.youtube.com/watch?v=rEKtr8DYHWk
https://www.mathworks.com/help/images/ref/im2uint8.html
https://www.mathworks.com/matlabcentral/answers/595522-how-to-convert-double-to-uint8
https://www.mathworks.com/matlabcentral/answers/868103-my-data-type-changes-to-uint8-after-if-condition-i-want-to-keep-it-int16
https://www.mathworks.com/help/images/ref/imhistmatch.html
https://www.youtube.com/watch?v=vAmbSc7rJKQ
https://www.youtube.com/watch?v=gUx7_mmxsG4
https://www.mathworks.com/videos/getting-started-with-image-processing-1617599799993.html
https://www.mathworks.com/videos/converting-double-precision-design-to-embedded-efficient-fixed-point-design-1504212832864.html
https://www.mathworks.com/matlabcentral/answers/237282-reason-for-converting-image-into-double
https://www.mathworks.com/matlabcentral/answers/258217-how-can-i-found-the-distance-ssd-sum-of-squared-differences-between-two-images
https://stackoverflow.com/questions/29027878/sum-of-squared-differences
https://www.mathworks.com/help/images/ref/immse.html
https://www.mathworks.com/help/deeplearning/ref/dlarray.mse.html
https://www.mathworks.com/matlabcentral/answers/690680-why-am-i-unable-to-connect-to-my-speedgoat-target-computer-after-upgrading-to-r2020b-or-later
https://www.mathworks.com/videos/plot-geographic-data-on-a-map-in-matlab-1545831202291.html
https://www.youtube.com/watch?v=3St0byWYcEI
https://www.youtube.com/watch?v=fD4AEPAa6X4
https://www.mathworks.com/help/images/ref/imhist.html
https://www.mathworks.com/help/images/create-image-histogram.html
https://mathworks.com/help/matlab/ref/subplot.html
https://www.mathworks.com/matlabcentral/answers/50003-what-is-subplot-and-how-to-use-it
https://www.mathworks.com/help/matlab/ref/bar.html
https://www.mathworks.com/help/matlab/ref/xlim.html
http://www.ece.northwestern.edu/local-apps/matlabhelp/techdoc/ref/xlim.html
https://www.mathworks.com/matlabcentral/answers/525395-scale-plot-of-datetime-x-axis
https://www.mathworks.com/matlabcentral/answers/824110-what-does-xlim-do-in-my-code-and-why-are-its-brackets-empty
https://www.youtube.com/watch?v=bBCZQESz78o
https://www.youtube.com/watch?v=pppguq-qQpc
https://www.mathworks.com/matlabcentral/answers/91036-how-do-i-split-a-color-image-into-its-3-rgb-channels
https://www.mathworks.com/help/images/display-separated-color-channels-of-rgb-image.html
https://www.mathworks.com/matlabcentral/answers/111104-how-to-separate-an-image-to-rgb
https://www.mathworks.com/matlabcentral/answers/797742-how-to-separate-the-rgb-channels-from-an-image-and-get-the-average-of-each-rgb-channel-separetly