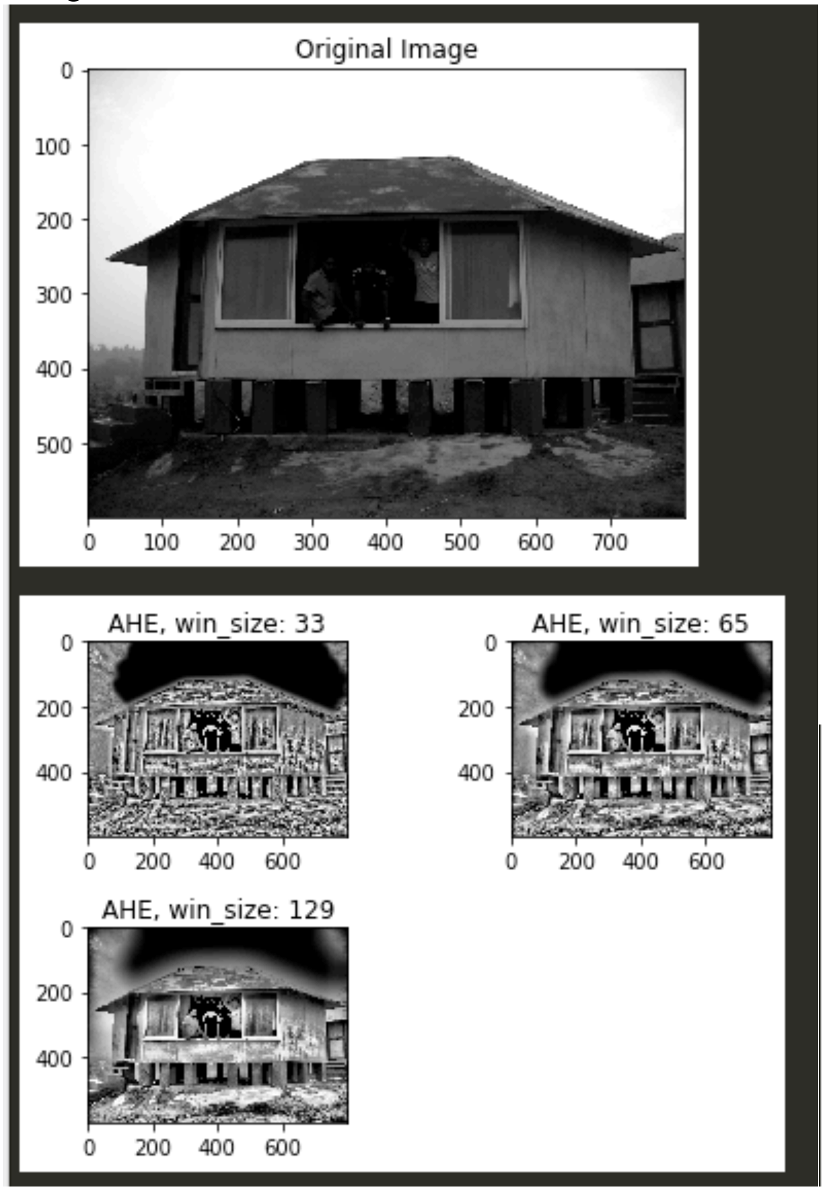
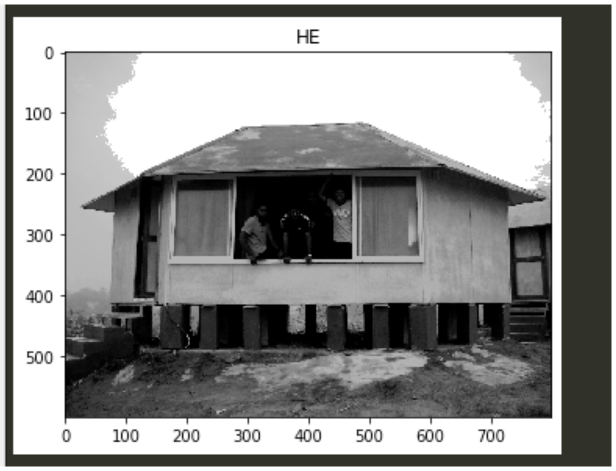
William Argus
A12802324
ECE 253, Fall 2020, Homework 2

Note: All Python and Matlab code is contained in this report at the end, organized by problem

Problem 1

//image





//image

How does the original image qualitatively compare to the images after AHE and HE respectively?

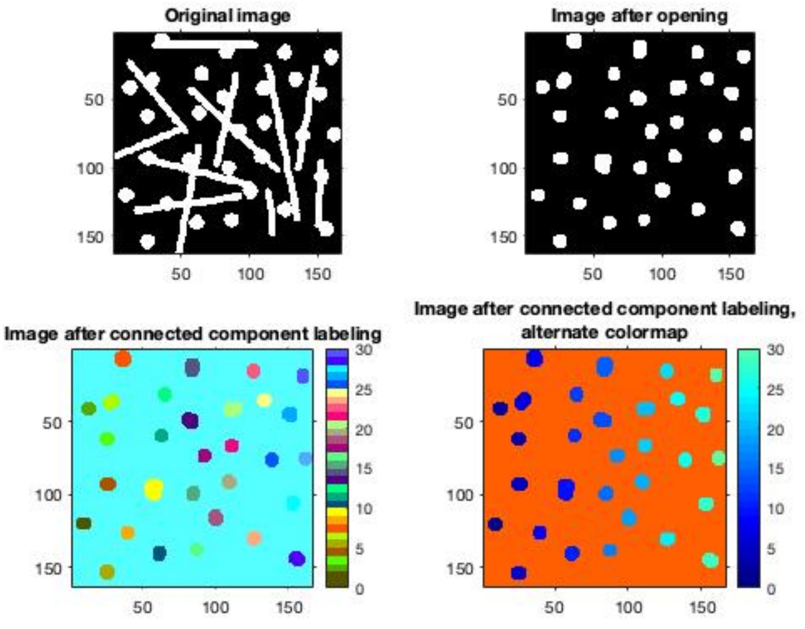
In terms of a human viewing the images they look worse, since for AHE the image is equalized in the given window size, meaning the local contrast all over the image is increased and noise is overamplified, which doesn't look natural; the larger the window size, however, the less the local contrast is increased and the more natural it looks. AHE, in particular the smaller windows, does make it features stand out more due to its high local contrast. After HE, the image in general looks more natural, however the lightest part of the sky (most of the sky) is the lightest part of the image so the equalization turns it white, again making the picture look unnatural.

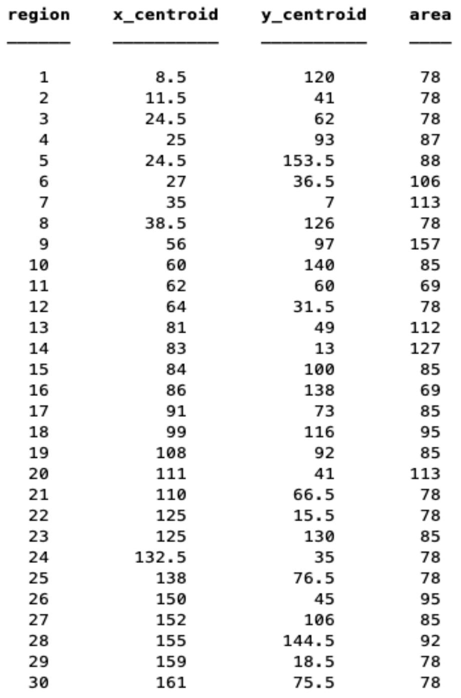
Which strategy (AHE or HE) works best for beach.png and why? Is this true for any image in general?

AHE works best for beach since it brings outs the features better (particularly those in the dark window), and keeps the sky from turning completely white due to the its localized equalization. In general this is true for most images, as AHE makes local features easier to see than HE, and doesn't suffer the issue with the sky turning totally white, seen in HE. It should be noted that AHE can overamplify noise, however this student assumes that histogram equalization is being used to better see features, which AHE is better than HE at, hence the statement that AHE is the better choice.

Problem 2

Part (i):
//image



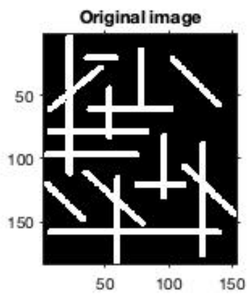


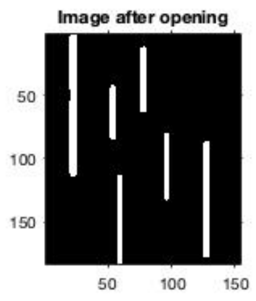
```
region    x_centroid    y_centroid    area
______    __________    __________    ____

   1          8.5           120         78
   2         11.5            41         78
   3         24.5            62         78
   4           25            93         87
   5         24.5         153.5         88
   6           27          36.5        106
   7           35             7        113
   8         38.5           126         78
   9           56            97        157
  10           60           140         85
  11           62            60         69
  12           64          31.5         78
  13           81            49        112
  14           83            13        127
  15           84           100         85
  16           86           138         69
  17           91            73         85
  18           99           116         95
  19          108            92         85
  20          111            41        113
  21          110          66.5         78
  22          125          15.5         78
  23          125           130         85
  24        132.5            35         78
  25          138          76.5         78
  26          150            45         95
  27          152           106         85
  28          155         144.5         92
  29          159          18.5         78
  30          161          75.5         78
```

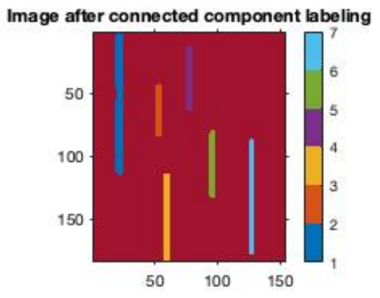
//image
Structuring Element type and size for 'circles_lines': Disk, size: 5

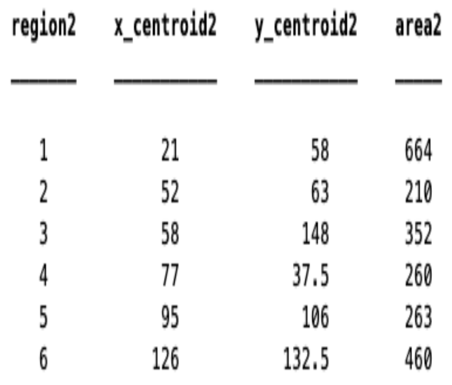
Part (ii):

//image









```
region2    x_centroid2    y_centroid2    area2
_______    ___________    ___________    _____

   1           21              58         664
   2           52              63         210
   3           58             148         352
   4           77            37.5         260
   5           95             106         263
   6          126           132.5         460
```

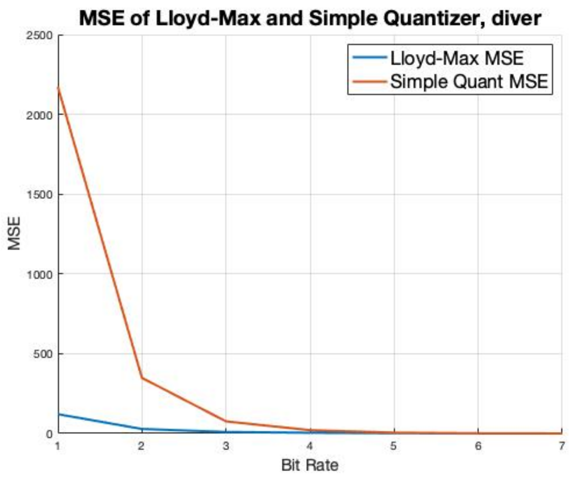
//image

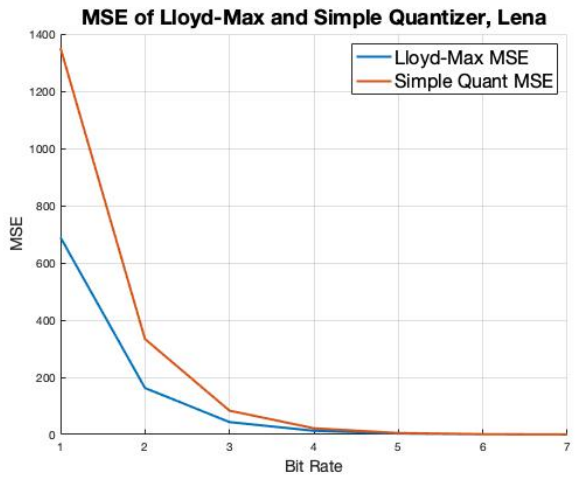
Structuring Element type and size for 'lines': Line, size: 9, degrees of rotation: 90

Problem 3

Parts (i) and (ii):

//image





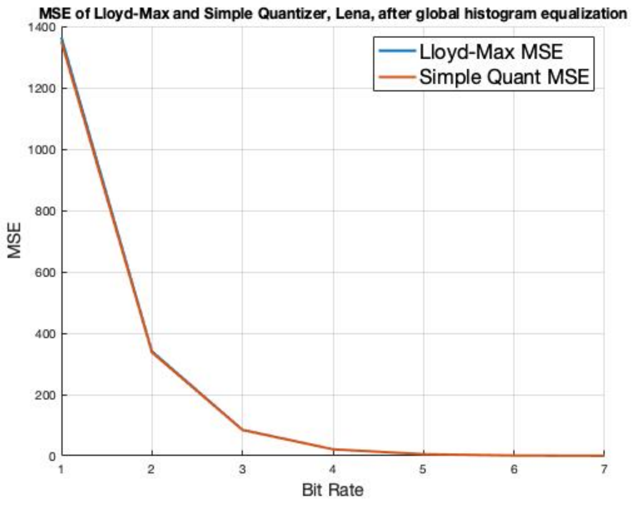
//image

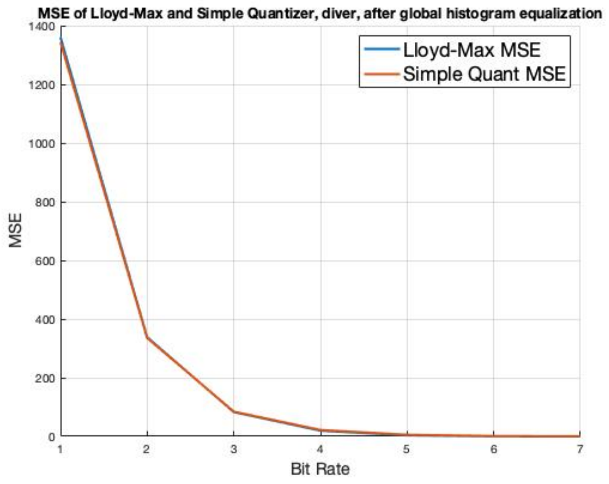
Compare the results for the different quantizers/images and explain them. That is, why does one quantizer outperform the other, and why is the performance gap larger for one image than for the other?

The Lloyd-max quantizer is optimized, as explained the textbook, such that the resulting optimized image has minimum MSE compared to the original image. This involves adjusting the bin thresholds as well as the value assigned to each bin, so it makes sense that the Lloyd-max has smaller MSE than a simple quantizer, especially at lower bit rate. The Lena image had a more equalized histogram, meaning that its pixel values were more evening spread over the range than those of the diver image. As a result, the simple quantizer works better than the diver image as its bins are spread equally over the range. The Lloyd-max doesn't work as well as it did with the diver, since the thresholds have to be more evenly spread to account for the range of pixels, making it closer in performance to the simple quantizer.

Part (iii):

//image





//image

Compare them with the previous set of plots. What has happened to the gap in MSE between the two quantization approaches and why?

The gap in MSE between the two approaches has shrunk a lot, to the point where performance is almost the same. This is because the images underwent histogram equalization, so the advantage of the Lloyd-max is gone as it can no longer optimize its thresholds and bin values – it ends up having thresholds and bin values nearly same as the simple quantizer since that is the optimal placement in an image that is perfectly equalized. (This also ties into what was said above about the performance on the diver image vs the lena image).

Part (iv):
Why is the MSE of the 7-bit Lloyd-Max quantizer zero or near zero for the equalized images? One might have thought that equalization is not to the advantage of the Lloyd-Max quantizer, because equalizing the histogram should be attening the distribution, making it more uniform, which should be to the advantage of the uniform quantizer. Explain this phenomenon.

The MSE is zero or near zero of the Lloyd-Max quantizer at 7-bit because firstly, it started with an 8-bit image so reducing to 7-bit results a smaller loss of information than reducing to, say, 2 bits. Secondly, the point of the Lloyd-Max quantizer is to optimize its parameters, so it isn't really fair to say that the uniform quantizer has the advantage, since if uniform quantization is the best approach, the Llyod-max can simply optimize to be the same as the uniform-quantization. Essentially the performance of the two at 7-bits on a histogram equalized image is very nearly the same, with both having low error due to having 7-bits, meaning the largest possible deviation in pixel value from original to quantized image is less than 0.5.

<u>Code</u>

<u>Problem 1: (Python)</u>
```
##William Argus A12802324
## ECE 253 HW1
'''
Academic Integrity Policy: Integrity of scholarship is essential for an academic commu-
nity. The University expects that both faculty and students will honor this principle and
in so doing protect the validity of University intellectual work. For students, this means
that all academic work will be done by the individual to whom it is assigned, without
unauthorized aid of any kind.
By including this in my report, I agree to abide by the Academic Integrity Policy men-
tioned above.
```

```
'''


import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.axisartist.axislines import Subplot
import math
import cv2

def AHE(im, win_size):
    half = int(win_size/2)
    imPadded = np.pad(im, half, mode = 'symmetric')
    sz = np.shape(im)
    output = np.zeros(np.shape(im))
    for x in range(half, sz[0]+half):
        #print(x)
        for y in range(half, sz[1]+half):
            rank = 0
            for i in range(x-half, x+half):
                for j in range(y-half, y+half):
                    if(imPadded[x,y] > imPadded[i,j]): rank += 1
            output[x-half,y-half] = rank*(255/(win_size**2))
    return output


#set-up call to function
#image beach.png for win size = 33; 65 and 129
win_size = [33, 65, 129]
im = plt.imread('Beach.png')*255


#code to easily downsize without having to change any parameters other than "downsize"
from skimage.transform import rescale
downSize = 1
im = rescale(im, 1/downSize, anti_aliasing=False)
win_size = [33/downSize, 65/downSize, 129/downSize]
win_size = [33, 65, 129]

plt.figure(1)
plt.title('Original Image')
plt.imshow(im, cmap = 'gray')

rows = 2
cols = 2
axes=[]
fig=plt.figure(2)


for a in range(len(win_size)):
    print(a)
    b = AHE(im, win_size[a])
    axes.append( fig.add_subplot(rows, cols, a+1) )
    subplot_title=("AHE, win_size: "+str(win_size[a]))
    axes[-1].set_title(subplot_title)
    plt.imshow(b, cmap = 'gray')
fig.tight_layout()
```

```
plt.show()

axesB=[]
figB=plt.figure(3)
img = cv2.imread('Beach.png',0)
#img = im

b = cv2.equalizeHist(img)
axesB.append( figB.add_subplot(1, 1, 1) )
subplot_title=("HE")
axesB[-1].set_title(subplot_title)
plt.imshow(b, cmap = 'gray')
figB.tight_layout()
plt.show()
```

Problem 2: (Matlab)

```
%%
%William Argus, A12802324

%%
% Part i
image = imread('circles_lines.jpg');
%image = image(:,:,1);
image=im2bw(image,50/255);
figure(1);
subplot(221);
imshow(image)
title('Original image');
axis on;

i=5;
circle = strel('disk',i);
postOpening = imopen(image,circle);

%figure(2);
axes();
subplot(222);
imshow(postOpening);
title('Image after opening');
axis on;

L = bwlabel(postOpening);
temp = max(max(L));

LAlt = L;
LAlt(LAlt == 0) = 31;
%figure(3);
subplot(223);
imshow(LAlt, colorcube);
h = colorbar;
set(h, 'ylim', [0 30])
title('Image after connected component labeling');
axis on;

LAlt = L;
LAlt(LAlt == 0) = 50;
%figure(4);
subplot(224);
imshow(LAlt, jet);
k = colorbar;
set(k, 'ylim', [0 30])
```

```
title({'Image after connected component labeling,', 'alternate colormap'});
axis on;

region = zeros(temp,1);
x_centroid = zeros(temp,1);
y_centroid = zeros(temp,1);
area = zeros(temp,1);
%region, x_centroid, y_centroid, area
sz=size(L);
sz=uint8(sz);
for i = 1:temp
    region(i,1)=i;
    listVals = find(L == i);
    area(i,1) = length(listVals);

    indics = zeros(length(listVals), 2); %x and y indices
    for j = 1:length(listVals)
        x = idivide(listVals(j,1), sz(1));
        y = rem(listVals(j,1), sz(1));
        indics(j,1) = x;
        indics(j,2) = y;
    end
    centroid = median(indics,1);
    x_centroid(i,1) = centroid(1);
    y_centroid(i,1) = centroid(2);
end

tabulate = table(region, x_centroid, y_centroid, area);
tabulate(:,:)

%%
%for i = 5:5
%    circle = strel('disk',i);
%    postOpening = imopen(image,circle);
%    figure(i);
%    imshow(postOpening);
%end

%Part ii
%close all;
%clear all;
image2 = imread('lines.jpg');
image2=im2bw(image2,50/255);
figure(5);
subplot(221);
imshow(image2)
title('Original image');
axis on;

line = strel('line', 9, 90);
postOpening2 = imopen(image2,line);

%figure(6);
axes();
subplot(222);
imshow(postOpening2);
title('Image after opening');
axis on;

L2 = bwlabel(postOpening2);
numLines = max(max(L2));

%figure(7);
```

```
%imshow(L2, lines);
%h = colorbar;
%set(h, 'ylim', [-1 6])
%title('Image after connected component labeling');
%axis on;

L2Alt = L2;
L2Alt(L2Alt == 0) = 7;
%figure(8);
subplot(223);
imshow(L2Alt, lines);
k = colorbar;
set(k, 'ylim', [1 7])
title('Image after connected component labeling');
axis on;


region2 = zeros(numLines,1);
x_centroid2 = zeros(numLines,1);
y_centroid2 = zeros(numLines,1);
area2 = zeros(numLines,1);
%region, x_centroid, y_centroid, area
sz2=size(L2);
sz2=uint8(sz2);
for i = 1:numLines
    region2(i,1)=i;
    listVals2 = find(L2 == i);
    area2(i,1) = length(listVals2);

    indics2 = zeros(length(listVals2), 2); %x and y indices
    for j = 1:length(listVals2)
        x = idivide(listVals2(j,1), sz2(1));
        y = rem(listVals2(j,1), sz2(1));
        indics2(j,1) = x;
        indics2(j,2) = y;
    end
    centroid2 = median(indics2,1);
    x_centroid2(i,1) = centroid2(1);
    y_centroid2(i,1) = centroid2(2);
end

tabulate2 = table(region2, x_centroid2, y_centroid2, area2);
tabulate2(:,:)
```

Problem 3: (Matlab)

```
%%
% William Argus, A12802324

%%
% Part i

%image = imread('diver.tif');
image = imread('lena512.tif');
%image = histeq(image,256);
figure(10);
imshow(image);

s = [1,2,3,4,5,6,7];

quantMSE = zeros(length(s),1);
```

```
for i = 1:7
    startPoint = 256/(2^s(i));
    numBins = 2^s(i);
    numThresh = numBins-1;
    %create thresholds
    thresh = [0];
    for j = 1:numThresh+1
        thresh = [thresh (startPoint*j -1)];
    end

    %create bin values
    firstBin = thresh(2)/2;
    increment = thresh(2) +1;
    binVals = [firstBin];
    for k = 2:numBins
       binVals = [binVals (binVals(k-1)+increment) ];
    end

    imageQuant = double(image);
    sz=size(imageQuant);
    for x = 1:sz(1)
       for y = 1:sz(2)
           for l = 1:numBins
               %if l == 1
               %    if imageQuant(x,y) <= thresh(l+1)
               %        imageQuant(x,y) = binVals(l);
               %    end
               if imageQuant(x,y) > thresh(l)
                   if imageQuant(x,y) <= thresh(l+1)
                       imageQuant(x,y) = binVals(l);
                   end
               end
           end
       end
    end
    figure(i);
    plotImageQuant = uint8(imageQuant);
    imshow(plotImageQuant);

    imageError = double(image);
    imageError = mean(mean((double(image) - imageQuant).^2));
    quantMSE(i) = imageError;
end

%% lloyd portion

imageDouble = double(image);
[M,N] = size(imageDouble);
training_set = reshape(imageDouble,N*M,1);

lloydMSE = zeros(length(s),1);

for i = 1:7
    len = 2^i;
    [PARTITION, CODEBOOK, DISTORTION] = lloyds(training_set, len);
    lloydMSE(i) = DISTORTION;
end



%% plot portion

figure(11);
```

```
hold on;
a(1) = plot(s,lloydMSE, 'DisplayName','Lloyd-Max MSE','LineWidth',2.0);
a(2) = plot(s,quantMSE, 'DisplayName','Simple Quant MSE','LineWidth',2.0);
%title('MSE of Lloyd-Max and Simple Quantizer, diver','FontSize',12);
title('MSE of Lloyd-Max and Simple Quantizer, Lena','FontSize',12);
%title('MSE of Lloyd-Max and Simple Quantizer, diver, after global histogram
equalization','FontSize',12);
%title('MSE of Lloyd-Max and Simple Quantizer, Lena, after global histogram
equalization','FontSize',12);
xlabel('Bit Rate','FontSize',14)
ylabel('MSE','FontSize',14)
grid on;
legend({}, 'Location','northeast', 'FontSize',16)
hold off;
```