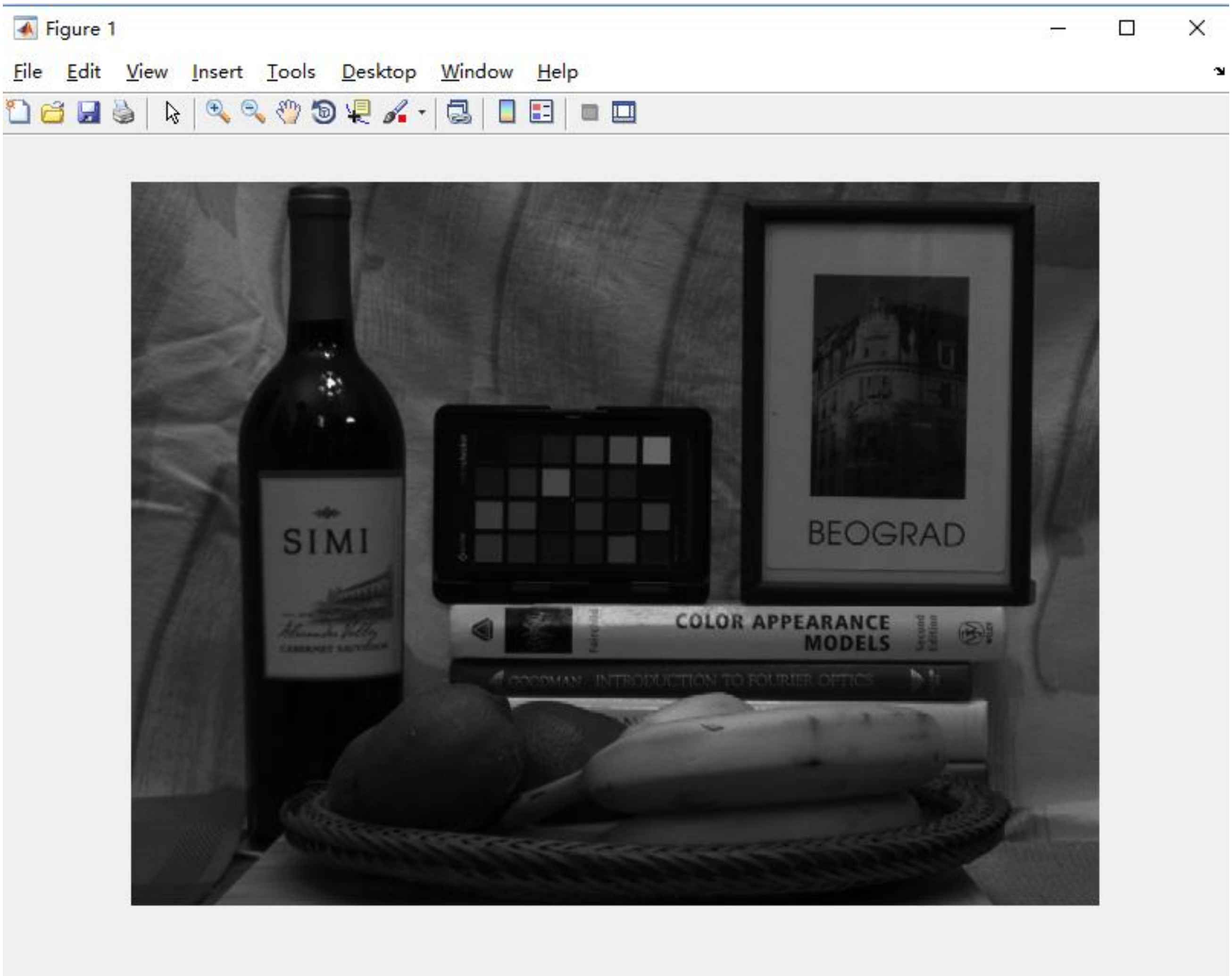
ECE415 -- Homework 2

Name: Wang Yunjuan; UIN:661180568

1) Display the image

Load the grayscale image Image.bmp into MATLAB and display the image. The picture is shown as follows.

```
im=imread('Image.bmp');
imshow(im);
```



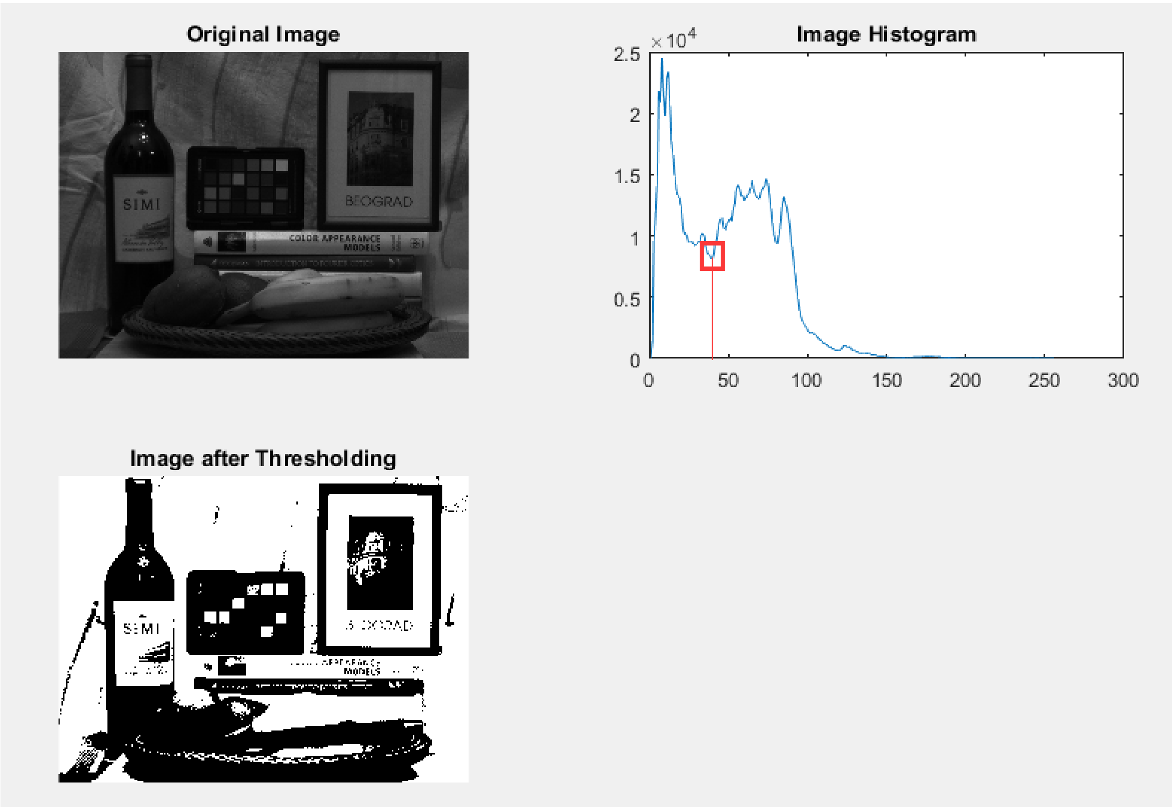
Before solving the following four questions, I first write a function *histogram* to calculate the histogram for the intensity image, which can implement the built-in function of *imhist*. The MATLAB code is as follows. I will use this function in the following questions.

```
function y=histogram(I)
[row col]=size(I);
```

```
h=zeros(1,256);
for i=1:row
    for j=1:col
        h(I(i,j)+1)=h(I(i,j)+1)+1;
    end
end
y=h;
end
```

2) Thresholding

First, we use function *histogram* I have write above to display the histogram of picture. We can easily find a valley in this curve. We roughly evaluate the pixel value of this valley as 40 and choose 40 as our threshold. The result is a binary image. We display the original image and the image after thresholding together to make a comparation.



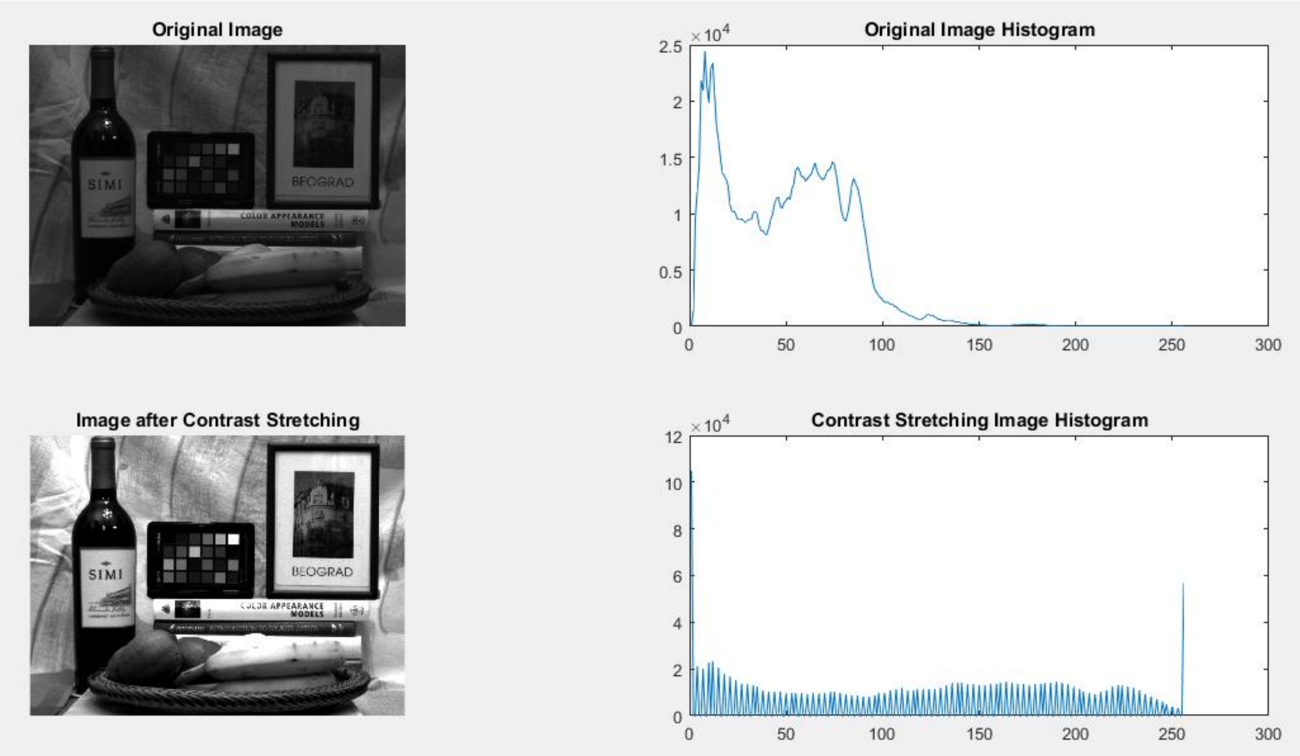
MATLAB code:

```
clear;clc;
im=imread('Image.bmp');
[row col]=size(im);
subplot(2,2,1);
imshow(im);
title('Original Image');
%% Histogram
h=histogram(im);
```

```
subplot(2,2,2);
plot(h);
title('Image Histogram');
%% Thresholding
for i=1:row
    for j=1:col
        if im(i,j)<40
            output(i,j)=0;
        else
            output(i,j)=255;
        end
    end
end
output=uint8(output);
subplot(2,2,3);
imshow(output);
title('Image after Thresholding');
```

3) Contrast Stretching

We calculate the actual range that want to stretch is [7,96], the target range is [0,255]. We display the original image and the image after stretching and both of its histogram together to make a comparation. We can find out that the image after stretching is clearer and brighter. Compared with these two histograms, we can find out that in the contrast stretching image histogram, the pixel value is more balanced.



MATLAB code:

```
clear;clc;
im=imread('Image.bmp');
[row col]=size(im);
imd=double(im);
countleft=0;
countright=0;
h1=histogram(im);
t=0.05*row*col;
for i=1:256
    countleft=countleft+h1(i);
    if countleft>=t
        break
    else
        continue
    end
end

for j=256:-1:1
    countright=countright+h1(j);
    if countright>=t
        break
    else
        continue
    end
end
output=double(zeros(row,col));

for m=1:row
    for n=1:col
        if im(m,n)<=i
            output(m,n)=0;
        else
            if im(m,n)>=j
                output(m,n)=255;
            else
                output(m,n)=(imd(m,n)-i)/(j-i)*255;
            end
        end
    end
end
output=uint8(output);

subplot(2,2,1);
```

```
imshow(im);
title('Original Image');
subplot(2,2,2);
plot(h1);
title('Original Image Histogram');
subplot(2,2,3);
imshow(output);
title('Image after Contrast Stretching');
subplot(2,2,4);
h2=histogram(output);
plot(h2);
title('Contrast Stretching Image Histogram');
```

4) Gamma Correction

I choose gamma equals to 0.5,1.0,2.0 separately and display the final image together as below. It can be seen that the picture is clearer and brighter when gamma is lower than 1, and the picture is darker when gamma is greater than 1.

MATLAB code:

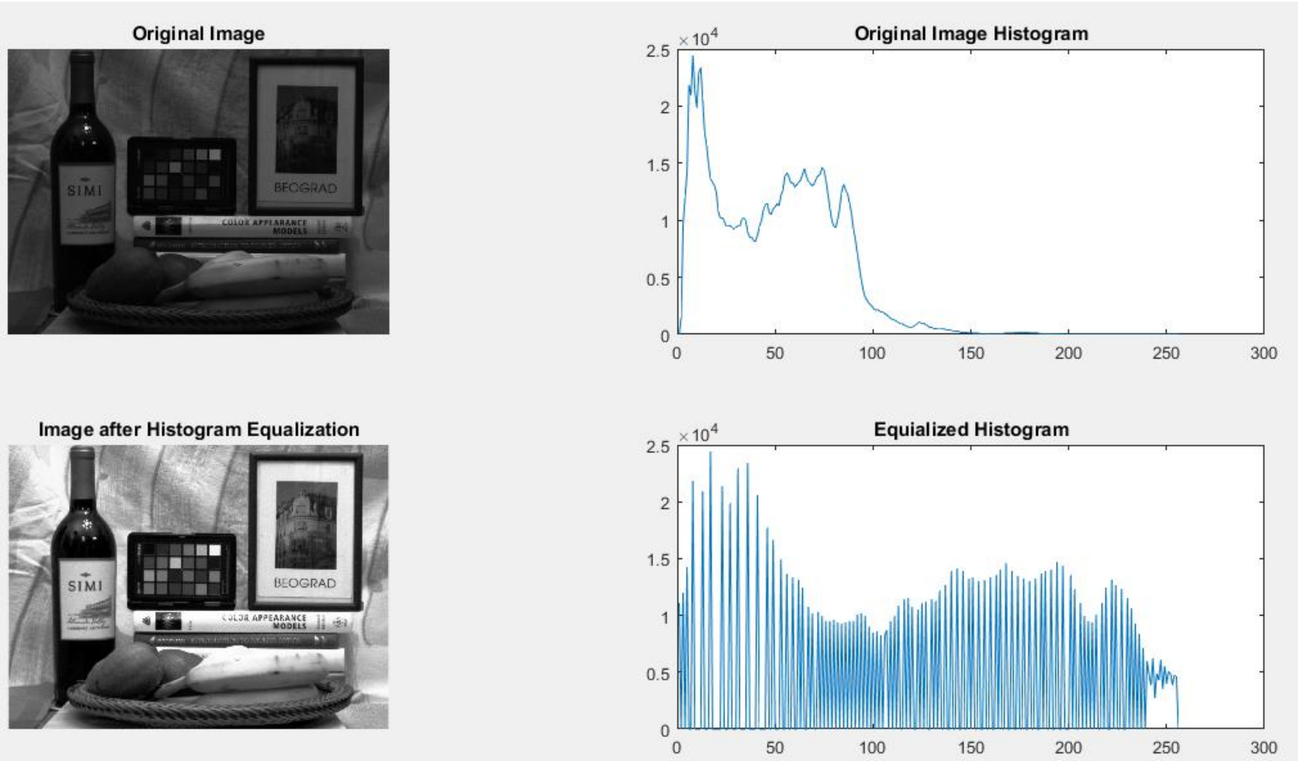
```
clear;clc;
im=imread('Image.bmp');
imd=double(im);
```

```
[row col]=size(imd);
%gamma=0.5
gamma1=0.5;
out1=abs((1*imd).^gamma1);
maxm1=max(out1(:));
minm1=min(out1(:));
for j=1:row
    for k = 1:col
        out1(j,k)=(255*out1(j,k))/(maxm1-minm1);
    end
end
out1=uint8(out1);

%gamma=1.0
gamma2=1.0;
out2=abs((1*imd).^gamma2);
maxm2=max(out2(:));
minm2=min(out2(:));
for j=1:row
    for k = 1:col
        out2(j,k)=(255*out2(j,k))/(maxm2-minm2);
    end
end
out2=uint8(out2);

%gamma=2.0
gamma3=2.0;
out3=abs((1*imd).^gamma3);
maxm3=max(out3(:));
minm3=min(out3(:));
for j=1:row
    for k = 1:col
        out3(j,k)=(255*out3(j,k))/(maxm3-minm3);
    end
end
out3=uint8(out3);

subplot(2,2,1);
imshow(im);
title('Original Image');
subplot(2,2,2);
imshow(out1);
title('Gamma=0.5');
subplot(2,2,3);
```

```
imshow(out2);
title('Gamma=1.0');
subplot(2,2,4);
imshow(out3);
title('Gamma=2.0');
```

5) Histogram equalization

The image after histogram equalization is below. We can see that the final image is clearer and brighter than before. Compared with these two histograms, we can find out that in the equalized histogram, the pixel value is more balanced.



MATLAB code:

```
clear;clc;
im=imread('Image.bmp');
imd=double(im);
[row col]=size(imd);

h1=histogram(im);
p=h1./(row*col);
cdf=zeros(1,256);
cdf(1)=p(1);
for i=2:256
    cdf(i)=cdf(i-1)+p(i);
end
ocdf=zeros(1,256);
```

```
ocdf=floor(255.*cdf);
output=zeros(row,col);
for j=1:255
    output(im==j)=ocdf(j);
end
output=uint8(output);
h2=histogram(output);

subplot(2,2,1);
imshow(im);
title('Original Image');
subplot(2,2,2);
plot(h1);
title('Original Image Histogram');
subplot(2,2,3);
imshow(output);
title('Image after Histogram Equalization');
subplot(2,2,4);
plot(h2);
title('Equilized Histogram');
```