EE 440 Autumn 2018

Homework 3 Report

Professor: Ming-Ting Sun

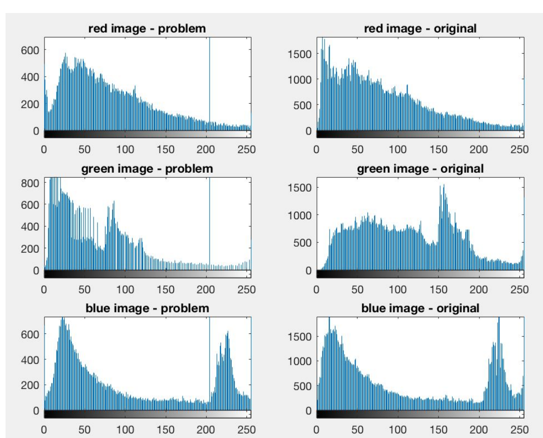
October 16th, 2018

He Feng 1427841

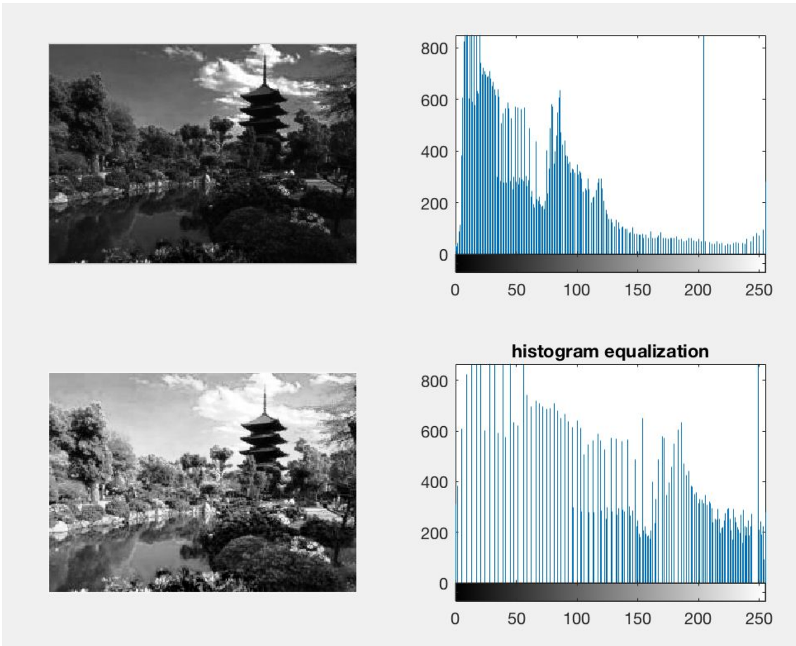
Problem 1

In this problem, I am going to figure out the problem of a picture which does not look right based on another more natural picture on the right of the problem image.

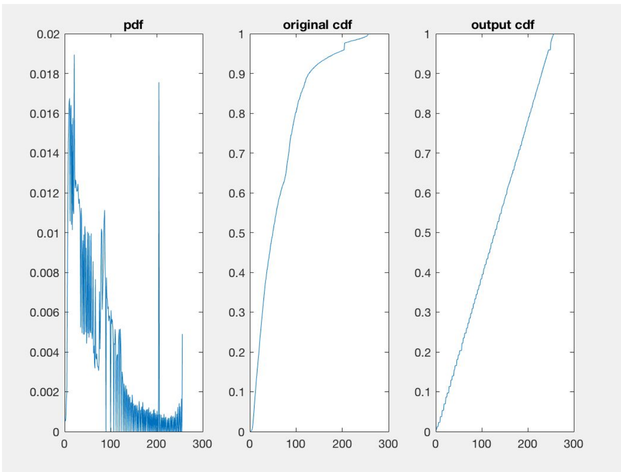
Initially I used imread function to load two images. Then I subtracted the RGB images of the natural picture and problem picture. By printing out the RGB images of two pictures in one figure, I found out the red images and blue images look very similar between each other for two pictures, but their green images are different. To prove this hypothesis, I used the function imhist to print out the histogram of each RGB images. The result is the same, which is the histogram of two pictures are different. Hence there are some problem happened in the green image, and I need to fixed the problem in green image such that I can get the ideal picture.



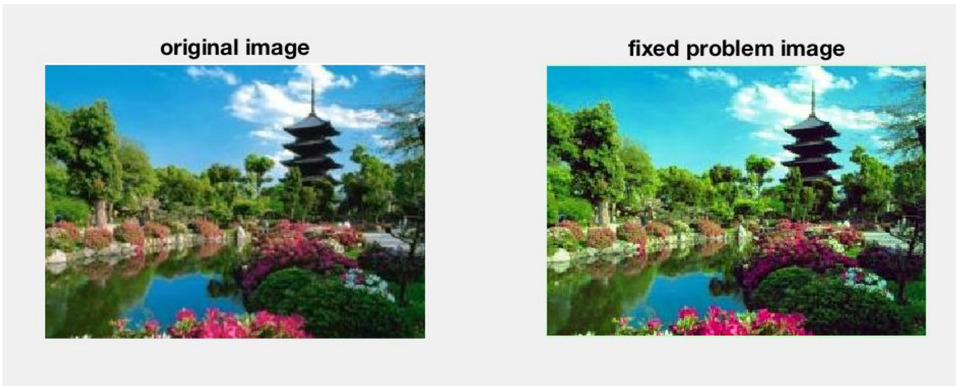
I will use histogram equalization for this situation, which will also be used in the following problem. I used size function and imhist function to get the cdf, and used cumsum function to determine the pdf. Then I used the for loop to check each index point by point. For each bin I found the index vector, and then update the data to the output image by using the information above. After this step I used the same method to determine the outcome picture's cdf, and I print the original and processed image's cdf in one figure. In this case the image is the green image of the problem picture. Finally I print out the image in another figure. Here are the original image with histogram and process image with histogram:



The cdf and pdf graphs are shown below:



At the end of this problem, after I used histogram specification to the green image of the problem picture, I print out the original image and the processed image in one figure such that we can observe the difference between them. By observing the cdf graph, I found the outcome cdf is more flat than the original one. Here is the graphs:



Here are the Matlab code of the first problem and its function code:

problem1.m:

```
clear all;
close all;

% Transfer the screenshot picture into bmp format such that we can use it
% to compare with the given non-perfect image.
pic = imread('3_1.bmp','bmp');
picture = imread('3_11.png');
imwrite(picture,'3_11.bmp','bmp');

% Get eh RGB images of the original image and problem image.
picture_r = picture(:,:,1);
picture_g = picture(:,:,2);
picture_b = picture(:,:,3);

pic_r = pic(:,:,1);
pic_g = pic(:,:,2);
pic_b = pic(:,:,3);

figure(1)
% Red image of the problem picture.
subplot(3,2,1);
imshow(pic_r);
title('red image - problem');
% Red image of the original picture.
subplot(3,2,2);
```

```
imshow(picture_r);
title('red image - original');
% Green image of the problem picture.
subplot(3,2,3);
imshow(pic_g);
title('green image - problem');
% Green image of the original picture.
subplot(3,2,4);
imshow(picture_g);
title('green image - original');
% Blue image of the problem picture.
subplot(3,2,5);
imshow(pic_b);
title('blue image - problem');
% Blue image of the original picture.
subplot(3,2,6);
imshow(picture_b);
title('blue image - original');

% Get the histogram of the RGB images of original image and problem image.
figure(2)
subplot(3,2,1);
imhist(pic_r);
title('red image - problem');
subplot(3,2,2);
```

```
imhist(picture_r);
title('red image - original');
subplot(3,2,3);
imhist(pic_g);
title('green image - problem');
subplot(3,2,4);
imhist(picture_g);
title('green image - original');
subplot(3,2,5);
imhist(pic_b);
title('blue image - problem');
subplot(3,2,6);
imhist(picture_b);
title('blue image - original');

% Histogram equilization of green image by using my own function.
pic(:,:,2) = myhisteq(pic(:,:,2));

figure
subplot(1,2,1);
imshow(picture);
title('original image');
subplot(1,2,2);
imshow(pic);
title('fixed problem image');

```

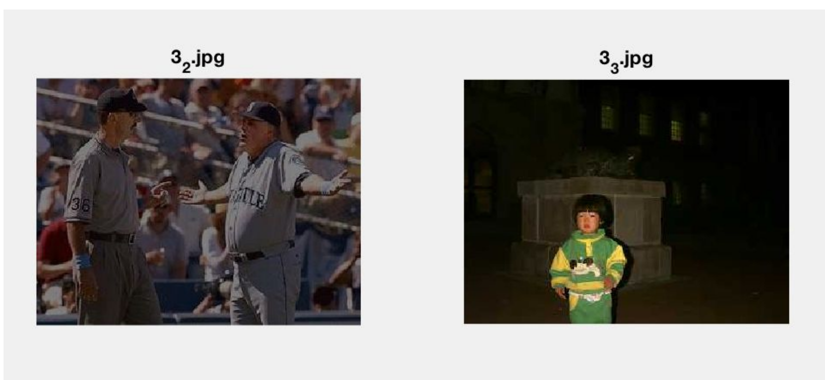
problem2.m:

```
function outIm= HistSpec(im,pdf)

outIm=im;
[m,n]=size(im);
count=imhist(im);
cdf=cumsum(count/(m*n));
outCdf=cumsum(pdf);

figure
subplot(1,2,1);plot(1:256,cdf);
title('original cdf ');
subplot(1,2,2);plot(1:256,outCdf);
title('outcome cdf ');

newBin=zeros(1,256);
% To find the index where the difference between cdf and outCdf is the
% smallest.
for i=1:256
    [minDiff,index]=min(abs(cdf(i)-outCdf));
    newBin(i)=index;
end

```

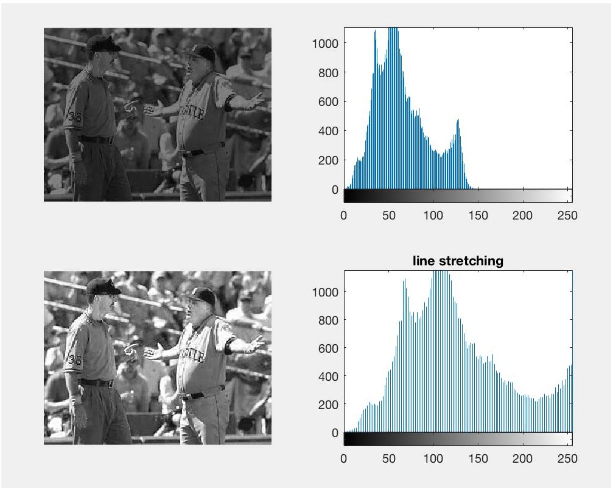
```
figure
% To plot the new mean
plot(newBin);
title('newbin');

for i=1:m
    for j=1:n
        outIm(i,j)=newBin(outIm(i,j)+1);
    end
end

figure
subplot(2,2,1);
imshow(im);
subplot(2,2,2);
imhist(im);
subplot(2,2,3);
imshow(outIm);
subplot(2,2,4);
imhist(outIm);
end
```

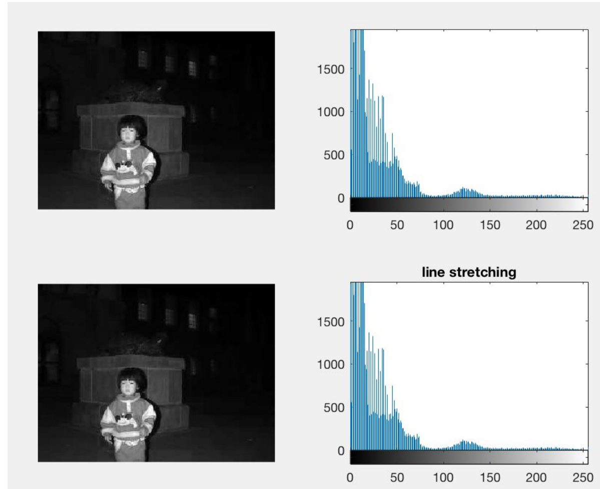
Problem 2

In this problem, I will use three methods to enhance a picture. I initially load the pictures by using imread function. Then I convert the image into RSV type, and then subtract the V image and transfer it into uint8 type. After this step I draw the given two images, and the output are shown below:



The first method I used is called line stretching. I wrote a function to support the main code to get the ideal result. In the function, I used min and max function to get the minimum pixel and maximum pixel of the given image. Then I used “outIm = k*(im-minPix)+b;” to get the outcome image. After this step, I draw the plot and histogram of given image and outcome image, which is shown below:



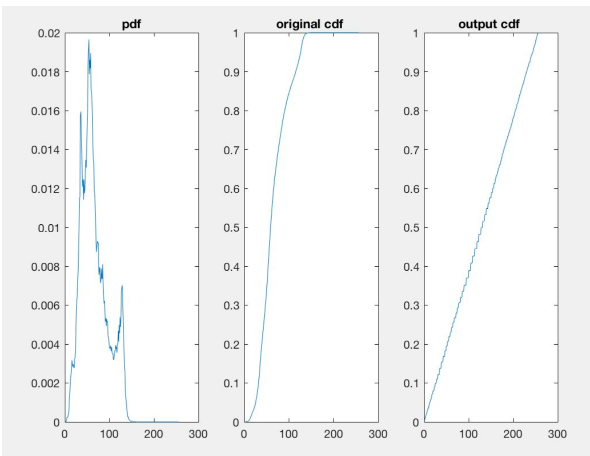


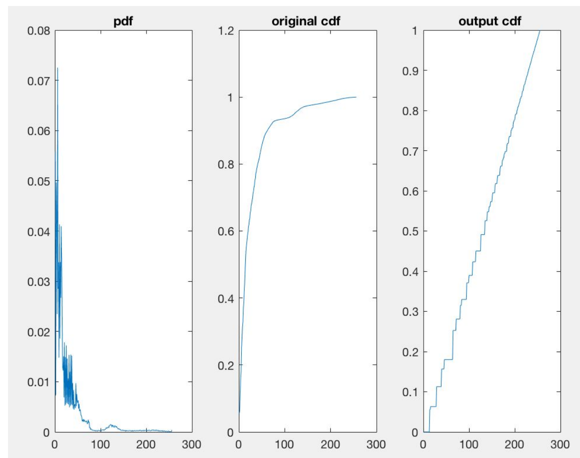
The two given images' line stretching format are given below:

Here is the Matlab code of the function LinearStretching.m:

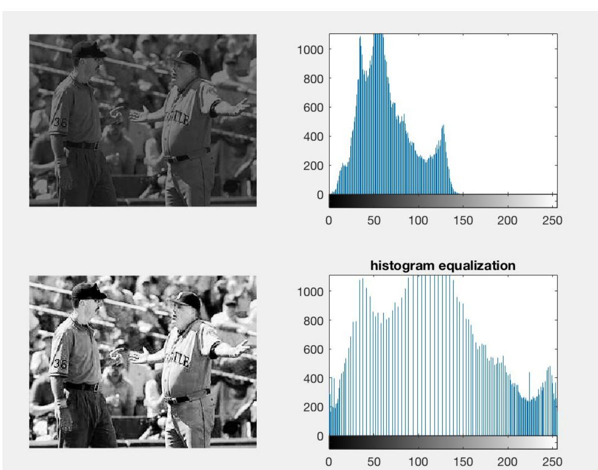
```matlab
function outIm = LineStretching(im)

% The minimum pixel and maximum pixel of the image.
minPix = min(im(:));
maxPix = max(im(:));
k = 255/(maxPix-minPix);
b = 0;
outIm = k*(im-minPix)+b;

figure
subplot(2,2,1);
imshow(im);
subplot(2,2,2);
imhist(im);
subplot(223);
imshow(outIm);
subplot(224);
imhist(outIm);
title('line stretching')
end
```

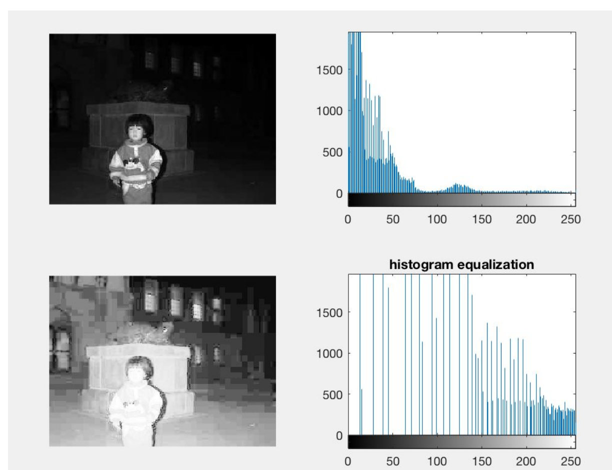
The second method I used histogram equalization, and I also use a function to support the main code. In the function, I used size function and imhist function to determine the pdf, then I used comsum function to get the cdf. Then I used the for loop to discover each index step by step. Initially I found the index vector of each element, then I used an equation to get the new bin. After this step I update the data to get a new image. By using the same method I mentioned before, I got the new cdf such that I can compare them in one figure. The following are the pdf, cdf and outcome cdf of two given images:





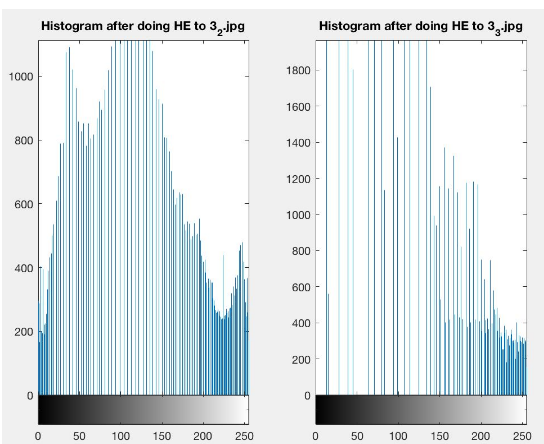
Then I pot the given images and outcome images of the original two pictures:

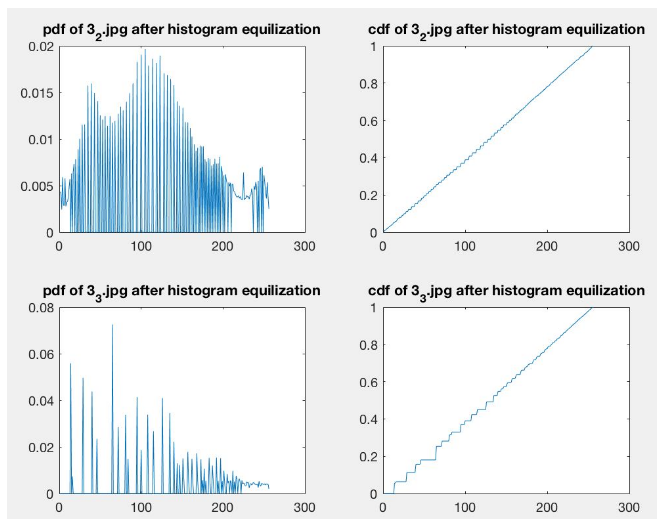




After the step above, I got a updated V image. I combined the updated V image with the original H image and S image, then I convert the HSV image into RGB image. Finally I plot the two images coming through histogram equalization, the output images are given below:

In the function of histogram equalization, I plot the histogram, pdf and cdf of the picture after histogram equalization such that we can get more information to understand this process. Here are the output graphs and images:



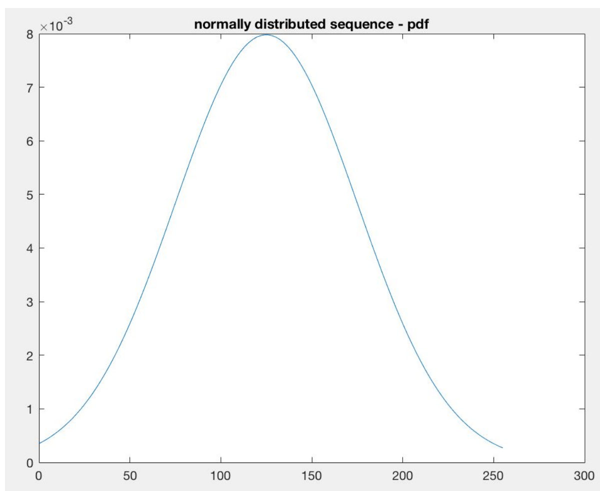


Here are the Matlab code of the function myhisteq.m:

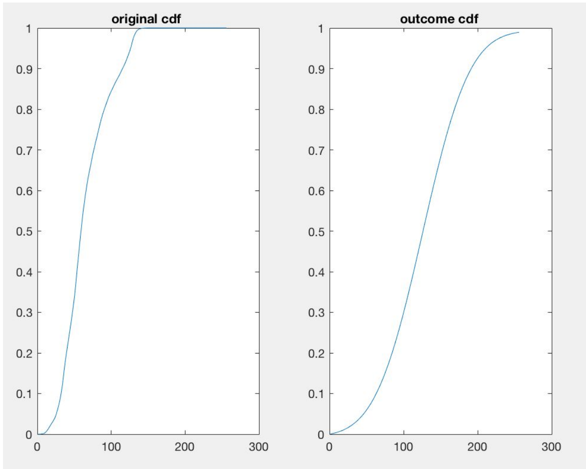
```
function outIm= myhisteq(im)

outIm=im;
[m,n]=size(im);
count=imhist(im);

pdf=count/(m*n);
cdf=cumsum(pdf);

for i=1:256
    % find index vector of bin i
    index=find(im==i-1);
    newBin=round(cdf(i)*256-1);
    for j=1:length(index)
        % Update the data to the output image
        outIm(index(j))=newBin;
    end
end

new_count=imhist(outIm);
outCDF=cumsum(new_count/(m*n));
```

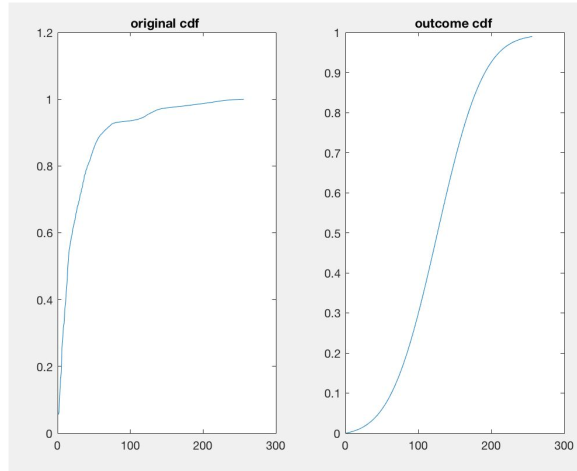
```
figure
subplot(1,3,1);
plot(1:256,pdf);
title('pdf');

subplot(1,3,2);
plot(1:256,cdf);
title('original cdf ')

subplot(1,3,3);
plot(1:256,outCDF);
title('output cdf ');

figure
subplot(2,2,1);
imshow(im);
subplot(2,2,2);
imhist(im);
subplot(2,2,3);
imshow(outIm);
subplot(2,2,4);
imhist(outIm);
title('histogram equalization');
end
```

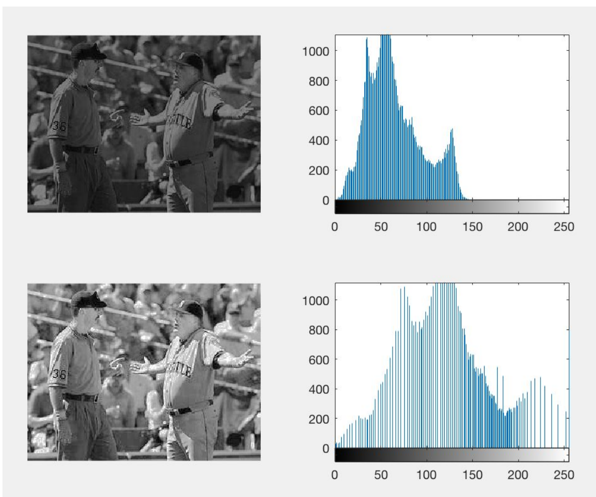
The third method is called histogram specification. I initially used the function normpdf to get a normally distributed sequence. Here is the graph of the normally distributed sequence:

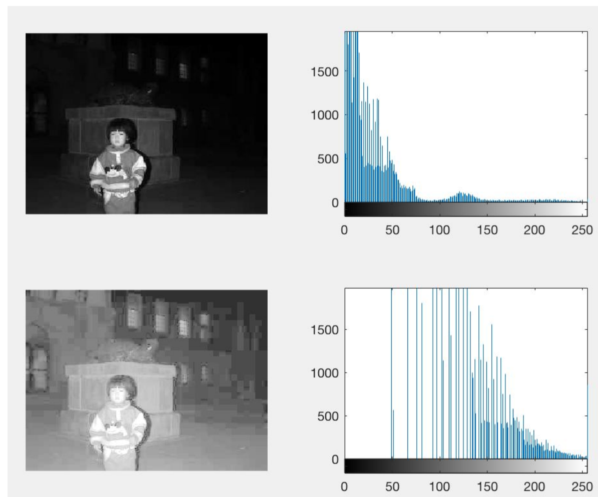


I also used a function to support my code. In the function, I used size function and imhist function to get the cdf similar to the steps in the last method. Then I use cumsum function to determine the output cdf based on the given pdf in the input catalog. Then I built an index vector which contains the index which can cause the smallest difference between cdf and outcome cdf. Then I get the output image based on the previous index vector. At the end of this function, I plot the two given V images and the histogram of them. I also plot the cdf and outcome cdf of these two images. Here are the images:









The images coming through the histogram specification are:

Here are the Matlab code of the function HistSpec.m:

```matlab
function outIm= HistSpec(im,pdf)

outIm=im;
[m,n]=size(im);
count=imhist(im);
cdf=cumsum(count/(m*n));
outCdf=cumsum(pdf);

figure
subplot(1,2,1);plot(1:256,cdf);
title('original cdf ');
subplot(1,2,2);plot(1:256,outCdf);
title('outcome cdf ');

newBin=zeros(1,256);
% To find the index where the difference between cdf and outCdf is the
% smallest.
for i=1:256
    [minDiff,index]=min(abs(cdf(i)-outCdf));
    newBin(i)=index;
end

for i=1:m
    for j=1:n
        outIm(i,j)=newBin(outIm(i,j)+1);
    end
end

figure
subplot(2,2,1);
imshow(im);
subplot(2,2,2);
imhist(im);
subplot(2,2,3);
imshow(outIm);
subplot(2,2,4);
imhist(outIm);
end
```

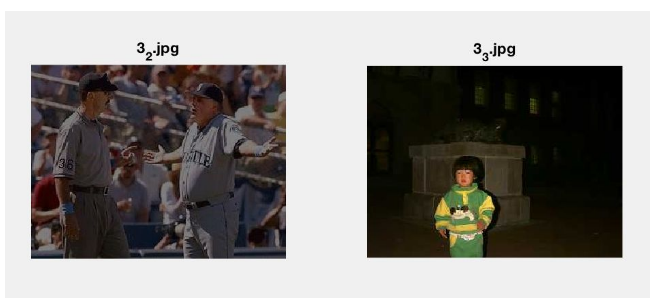
A common point for the three methods is in all of the cases I converted the image into HSV images. Then I do the processes in V images and combine it back to the original H images and S images. At last I convert the new HSV images into RGB images.

Here are the Matlab code of the main code: problem2.m

```matlab
close all;
clear all;

im1 = imread('3_2.jpg');
im2 = imread('3_3.jpg');
hsvIm1 = rgb2hsv(im1);
hsvIm2 = rgb2hsv(im2);
V1 = hsvIm1(:,:,3);
V1 = uint8(V1*255);
V2 = hsvIm2(:,:,3);
V2 = uint8(V2*255);


%% The original images.
figure(1)
subplot(1,2,1);
imshow(im1);
title('3_2.jpg');
subplot(1,2,2);
imshow(im2);
title('3_3.jpg');

%% The images after line stretching.
outV1_1= LineStretching(V1);
outV1_2= LineStretching(V2);
figure
subplot(1,2,1);
hsvIm1(:,:,3)=double(outV1_1)/255;
outIm1=hsv2rgb(hsvIm1);
imshow(outIm1)
title('3_2.jpg after line stretching');
imwrite(outIm1, '3_2_line_stretch.jpg');
subplot(1,2,2);
hsvIm2(:,:,3)=double(outV1_2)/255;
outIm2=hsv2rgb(hsvIm2);
imshow(outIm2)
title('3_3.jpg after line stretching');
imwrite(outIm2, '3_3_line_stretch.jpg');

%% The images after doing histogram equilization.

outV2_1 = myhisteq(V1);
outV2_2 = myhisteq(V2);
figure
hsvIm1(:,:,3)=double(outV2_1)/255;
outIm=hsv2rgb(hsvIm1);
subplot(1,2,1);
imshow(outIm);
title('3_2.jpg after histogram equalization');
imwrite(outIm, '3_2_his_equal.jpg');
hsvIm2(:,:,3)=double(outV2_2)/255;
outIm=hsv2rgb(hsvIm2);
subplot(1,2,2);
imshow(outIm);
title('3_3.jpg after histogram equalization');
imwrite(outIm, '3_3_his_equal.jpg');

% Histogram after two images doing the histogram equalization.
figure
subplot(1,2,1);
imhist(outV2_1);
count1 = imhist(outV2_1);
title('Histogram after doing HE to 3_2.jpg');
subplot(1,2,2);
imhist(outV2_2);
count2 = imhist(outV2_2);
title('Histogram after doing HE to 3_3.jpg');

[m1_1,n1_1] = size(outV2_1);
[m1_2,n1_2] = size(outV2_2);
pdf1_1 = count1/(m1_1*n1_1);
```

```
cdf1_1 = cumsum(pdf1_1);
pdf1_2=count2/(m1_2*n1_2);
cdf1_2=cumsum(pdf1_2);
figure
subplot(2,2,1);
plot(1:256,pdf1_1);
title('pdf of 3_2.jpg after histogram equilization');
subplot(2,2,2);
plot(1:256,cdf1_1);
title('cdf of 3_2.jpg after histogram equilization');
subplot(2,2,3);
plot(1:256,pdf1_2);
title('pdf of 3_3.jpg after histogram equilization');
subplot(2,2,4);
plot(1:256,cdf1_2);
title('cdf of 3_3.jpg after histogram equilization');

%% The images after doing histogram specification
x =0:255;
pdf = normpdf(x,125,50);
figure
plot(x,pdf)
title('normally distributed sequence - pdf');

outV3_1 = HistSpec(V1,pdf);
hsvIm1(:,:,3)=double(outV3_1)/255;
outIm3_1 = hsv2rgb(hsvIm1);
outV3_2 = HistSpec(V2,pdf);
hsvIm2(:,:,3) = double(outV3_2)/255;
outIm3_2 = hsv2rgb(hsvIm2);
figure
subplot(1,2,1);
imshow(outIm3_1);
title('3_2.jpg after histogram specification');
imwrite(outIm3_1, '3_2_his_spec.jpg');
subplot(1,2,2);
imshow(outIm3_2);
title('3_3.jpg after histogram specification');
imwrite(outIm3_2, '3_3_his_spec.jpg');
```

The output images from these three methods are:



By observing four pairs of pictures above. For the image 3_2.jpg, the image after line stretching is brighter than the given image. The image after equalization has an too obvious contrast. The image after histogram specification give me a clear image which details. I will choose to use histogram specification because it has a appropriate contrast and other factors make the picture comfortable to see. For the image 3_3.jpg, the image after line stretching looks clear and more likely to the given image. The image after histogram equalization looks too yellow and too bright. The image after histogram specification is not as bright as the upper image, but it looks too blurry. I will use line stretching because the same image in the other two methods are too blurry, and I can only say the image after line stretching got enhanced because the other two have really low quality.