Image Processing | Lab3

Aim : Implement following Image Enhancement Techniques using Octave.
1. Contrast Stretching
2. Intensity Level Slicing
3. Bit Plane Slicing

- Generalized Equation for contrast stretching:
  s = smin + [(smax-smin)/(rmax-rmin)]*(r-rmin);

- To perform bit plane slicing we use bitget() function:
  c = bitget(A,n)
  It returns the status of bit(s) n of the unsigned integers in A.

Assignment :

1. Do contrast stretching For the Image given in Figure 3.10 of the Textbook. Obtain Contrast stretched Image from Low contrast Image.

Code:

```
#Do contrast stretching For the Image given in Figure 3.10 of the Textbook.
#Obtain Contrast stretched Image from Low contrast Image as given in Figure.

r = imread("lab3images/4.tif");
subplot(1,2,1);
imshow(r);
title("Low contrast Image");
rmin = min(min(r));
rmax = max(max(r));
smax = 255;
smin = 0;
slope = (smax-smin)/(rmax-rmin);
s = smin + slope*(r-rmin);
subplot(1,2,2);
imshow(s);
title("Contrast stretched Image");
```

Output:

2. Take any family photo of yours – convert it into grayscale- reduce it's contrast by using the function that was defined during the lab session. Enhance the contrast of that image using piecewise linear operation for contrast stretching.

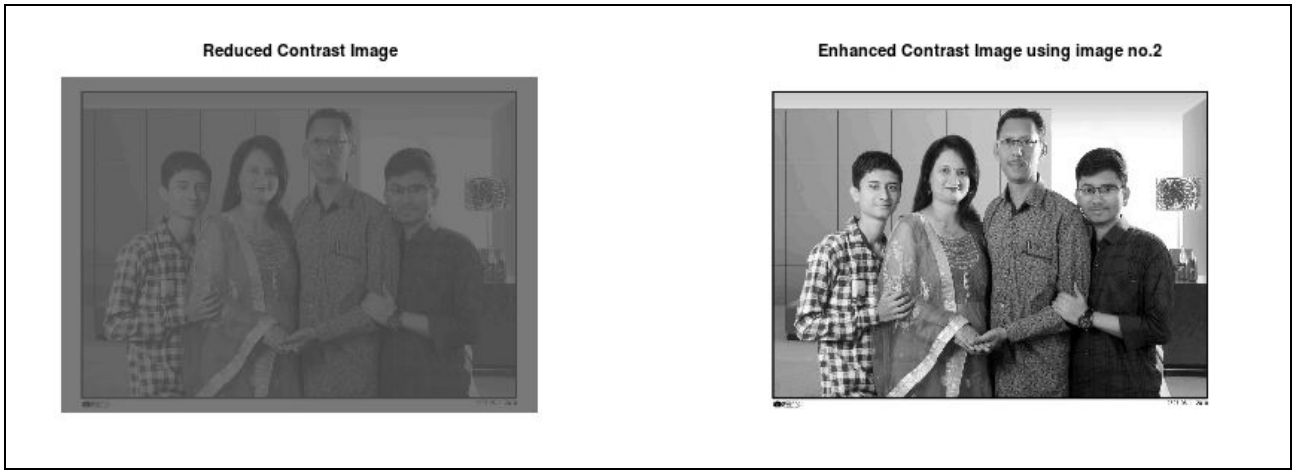
Code:

```
#Take any family photo of yours - convert it into grayscale- reduce it's
#contrast by using the function that was defined during lab session. Enhance
#the contrast of that image using piecewise linear operation for contrast
#stretching.

%task1
r = imread('family.jpg');
r_gray = rgb2gray(r);
subplot(1,3,1);
imshow(r_gray);
title('Original Image');
rmin = min(min(r_gray));
rmax = max(max(r_gray));
smax = 120;
smin = 70;
slope = double(smax-smin)/double(rmax-rmin);
slope
s = smin + slope*(r_gray-rmin);
subplot(1,3,2);
imshow(s);
title("Reduced Contrast Image");
```

```
23  %task2
24  rmin = min(min(s));
25  rmax = max(max(s));
26  smax = 255;
27  smin = 0;
28  slope = double(smax-smin)/double(rmax-rmin);
29  slope
30  s = smin + slope*(s-rmin);
31  subplot(1,3,3);
32  imshow(s);
33  title("Enhanced Contrast Image using image no.2");
```

Output:

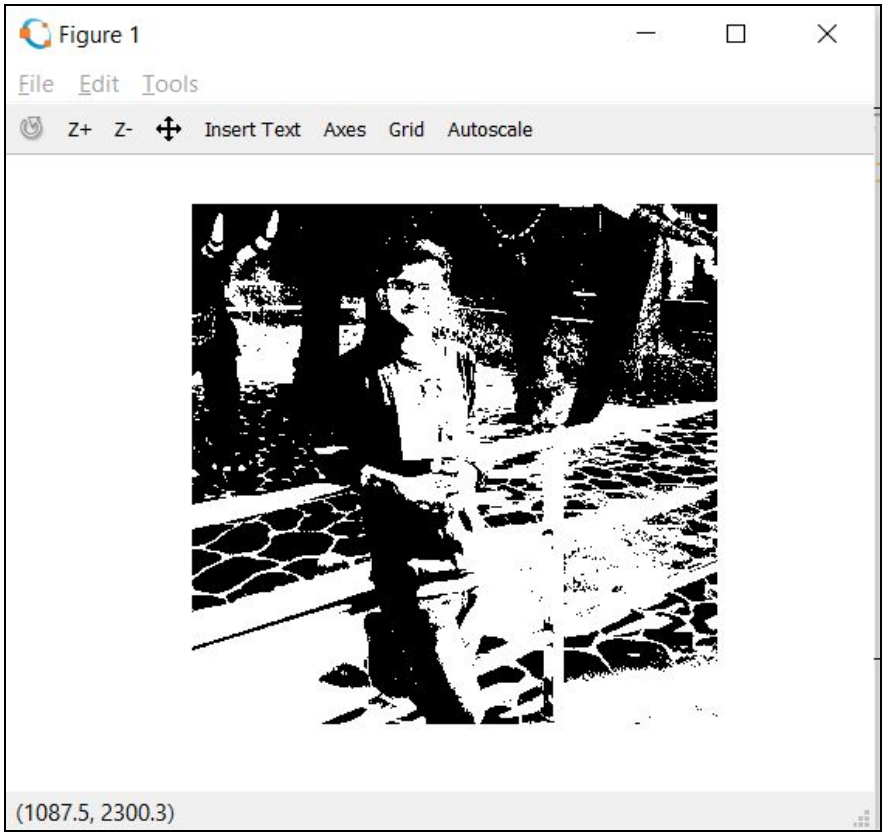


3. Apply thresholding to any of your grayscale photo.

Code:

```
#Apply thresholding to any of your gray scale photo.
r = imread('my_gray_scale.jpg');
s = NaN(size(r));
row = size(r,1);
col = size(r,2);
for i=1:row,
  for j=1:col,
    if (r(i,j)>150)
      s(i,j) = 1;
    else
      s(i,j) = 0;
    endif
  end
end
imshow(s);
```

Output:



4. Take your photo and separate out its bit plains. Reconstruct the given image using higher order 2 bit planes. Reconstruct the given image using higher order 4 bit planes. Experiment with the bit planes and derive your conclusions.

Code:

```
#Take your photo and separate out its bit plains.
#Reconstruct the given image using higher order 2 bit planes.
#Reconstruct the given image using higher order 4 bit planes.
#Experiment with the bit planes and derive your conclusions.

r = imread('my_image.jpg');
r_gray = rgb2gray(r);
subplot(3,3,1);
imshow(r_gray);
title("original Image");
for i=1:8,
  subplot(3,3,i+1);
  s = bitget(r_gray,i);
  imshow(s);
  title(i);
endfor

figure;

h2bit = bitget(r_gray,8).*2^(7) + bitget(r_gray,7).*2^(6);
subplot(1,2,1);
h2bit = uint8(h2bit);
imshow(h2bit);
title("Image using higher order 2 bit planes.");

h4bit = bitget(r_gray,8).*2^(7)+bitget(r_gray,7).*2^(6)+bitget(r_gray,6).*2^(5)+
        bitget(r_gray,5).*2^(4);
subplot(1,2,2);
h4bit = uint8(h4bit);
imshow(h4bit);
title("Image using higher order 4 bit planes.");
```

Output:

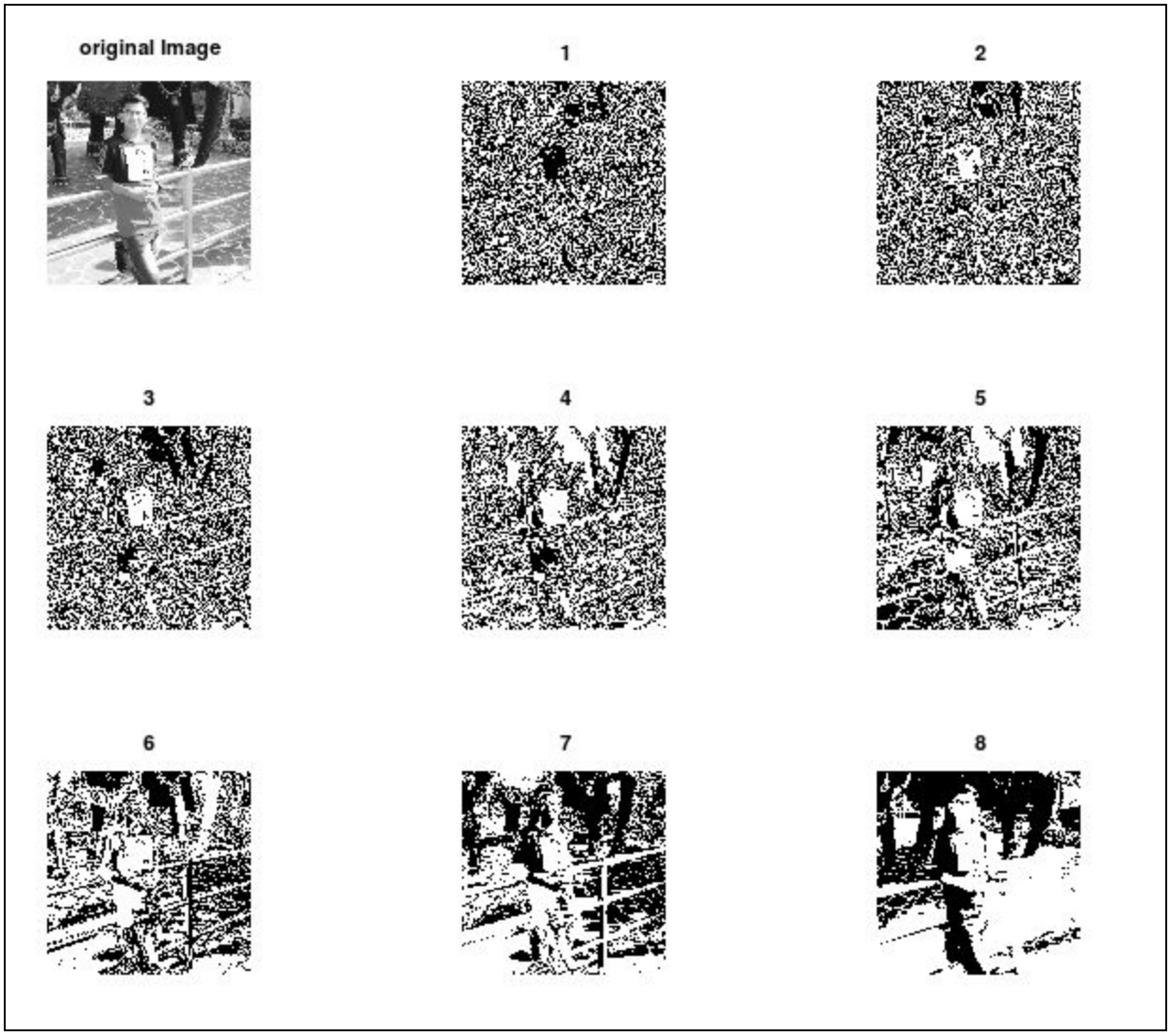


Image using higher order 2 bit planes.

Image using higher order 4 bit planes.

5. Perform intensity slicing to separate out red green balloons from the image 1(check: lab images) given.

Code:

```
#Perform intensity slicing to separate out red green baloons form
#the image 1(check: lab3images) given.
im = imread('lab3images/image1.jpg');
imshow(im);
figure;
slice_image = NaN(size(im));
row = size(im,1);
col = size(im,2);
for i=1:row,
  for j=1:col,
    if (im(i,j,1)<=255 && im(i,j,1)>=180 && im(i,j,2)>=0 && im(i,j,2)<=100 && im(i,j,3)>=0 && im(i,j,3)<=50)
      slice_image(i,j,:) = im(i,j,:);
    elseif (im(i,j,2)>=180 && im(i,j,2)<=255 && im(i,j,1)<=160 && im(i,j,1)>=100 && im(i,j,3)>=0 && im(i,j,3)<=70)
      slice_image(i,j,:) = im(i,j,:);
    else
      slice_image(i,j,1) = 0;
      slice_image(i,j,2) = 0;
      slice_image(i,j,3) = 0;
    endif
  endfor
endfor
slice_image =uint8(slice_image);
imshow(slice_image);
```

Output: