Image and Video Processing
Programming Assignment – Report

Week1: Contrast Enhancement using Histogram Equalization

MATLAB code:

```
%main.m
%Main script that imports the image and calls functions to calculate
%histogram and perform histogram equalization

%AUTHOR: SUMANTH SRINIVASAN

a = imread('lena_low_contrast.jpg');
a2 = double(a);
disp('Creating grayscale image...');
ycbcr = rgb_to_ycbcr(a2);
bw = uint8(ycbcr(:,:,1));
imwrite(bw,'lena_grayscale.jpg')

dim = size(a);

disp('Creating negative of the grayscale image')
%To create negative grayscale image
for x = 1:dim(1)
    for y = 1:dim(2)
        neg(x,y) = 255 - a(x,y);
    end
end
imwrite(neg,'lena_negative.jpg')
disp('Computing histogram of the grayscale image')
tic
% computeHist(bw);
tot_time = toc
disp('Done.')

histEq(bw);


%histEq.m
%AUTHOR: SUMANTH SRINIVASAN

function histEq(img)
%Histogram Equalization
%   Equalizes the non-uniform distribution of histogram

dim = size(img);

h = computeHist(img);
h = h/(dim(1)*dim(2));
```

```
% Computing Mapping Fn proportional to Cumulative Dist Fn

map = zeros(256,1);
for k = 1:256
    map(k)=uint8(sum(h(1:k))*255);
end;

figure; subplot(2,1,1), plot(map); title('Mapping function');
subplot(2,1,2), bar(h,'g'); title('Histogram');



% Mapping
tic
% for x = 1:dim(1)
%     for y = 1:dim(2)
%         f = double(img(x,y))+1;
%         histEqImg(x,y) = C(f);
%     end;
% end;

EqImg = map(double(img)+1);
tot_timeMap = toc

figure;
imshow(uint8(EqImg));
imwrite(uint8(EqImg),'lena_Hist_Equalized.jpg')

end



%computeHist.m
%AUTHOR: SUMANTH SRINIVASAN

function [h] = computeHist(img)
%A function to compute the histogram of an imported image
% WORKS ONLY FOR GRAYSCALE OR MONOCHROME IMAGES AT THE MOMENT
histo = zeros(256,1);
 for x = 0:255
    histo(x+1) = sum(sum(img == x));
 end

% figure
% plot(histo);

h = histo;
```

```
end

%rgb_to_ycbcr.m
%AUTHOR: SUMANTH SRINIVASAN


function [img2] = rgb_to_ycbcr(img)
%UNTITLED Converts RGB image to YCbCr format
%   Uses the conversion formula that converts every RGB pixel into YCbCr
%   pixel and returns the final image.


offset = [16;128;128];
coeffMatrix = [0.257 0.504 0.098;-0.148 -0.291 0.439;0.439 -0368 -0.071];
dim = size(img);
img2 = zeros(dim(1),dim(2),dim(3));


for x = 1:dim(1)
    for y = 1:dim(2)
        pix = coeffMatrix *[img(x,y,1);img(x,y,2);img(x,y,3)] + offset;
        img2(x,y,:) = pix(:);
    end
end
imshow(uint8(img2(:,:,1))); title('Grayscale');
end
```

Sample Images, Histogram and Output

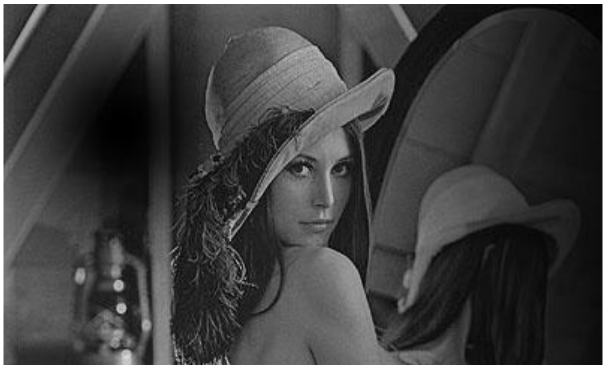

Figure 1 - Lena Grayscale Low Contrast

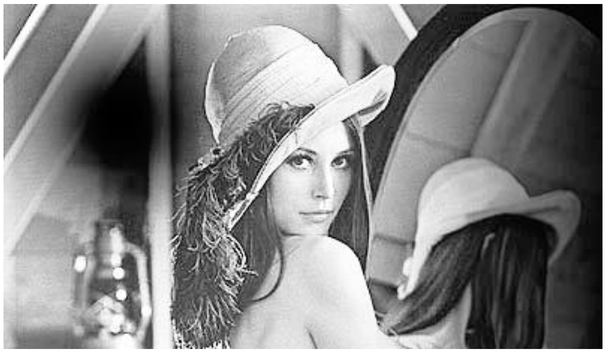

Figure 2 - Lena Grayscale after Histogram Equalization

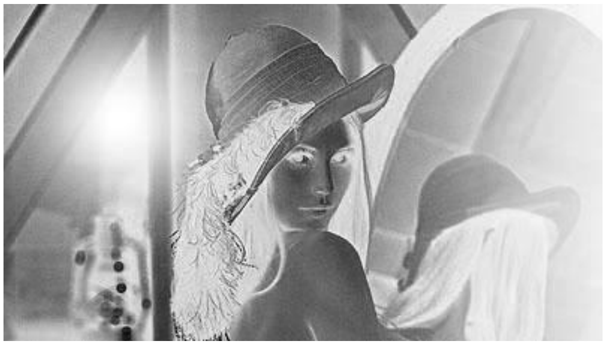

Figure 3 - Lena Grayscale Low Contrast Negative

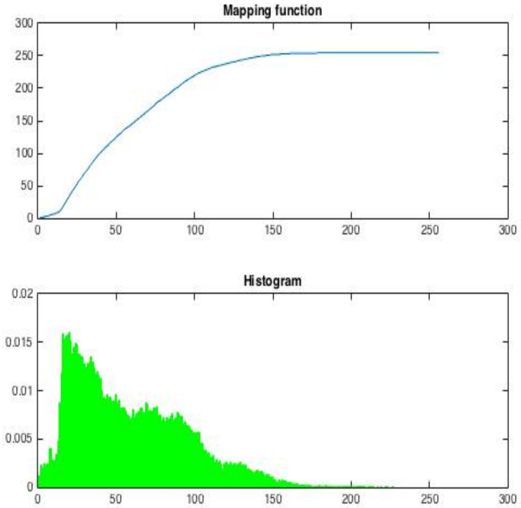

Figure 4 - Low Contrast Histogram and Mapping Function