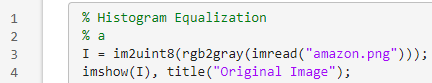
Homework 3 for **Kun**

Introduce to image process

All codes are attached on the last page.

Histogram Equalization

a.



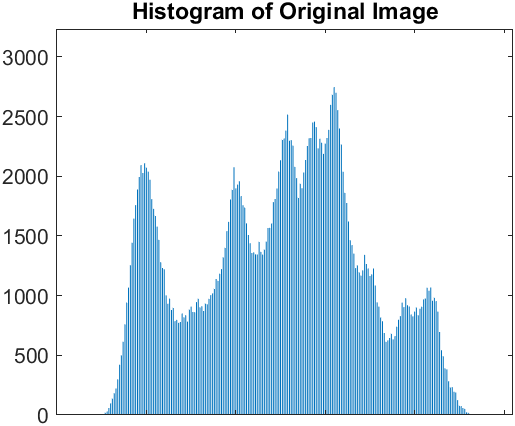
Res:



b.

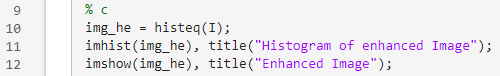


Res:

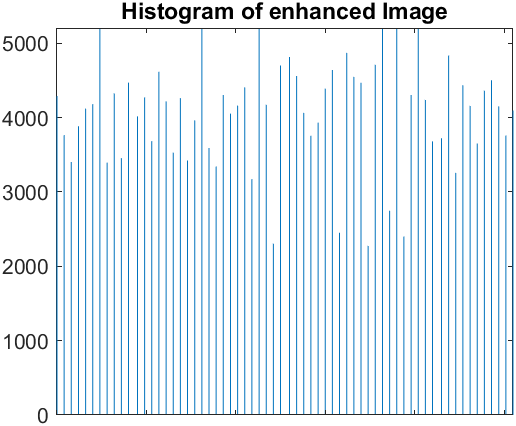




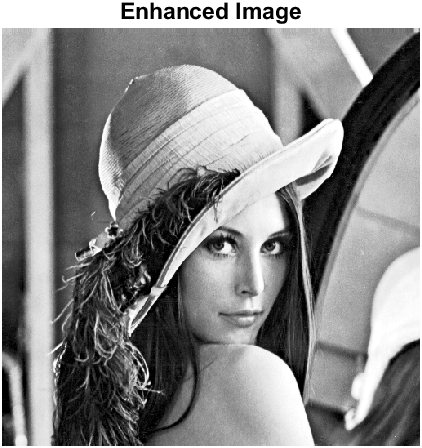
c.



Res:



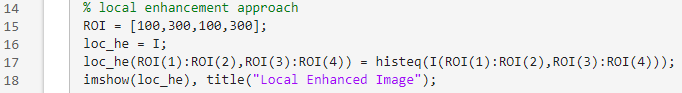




d.

The most significant target(background) has been moved to the medium position.

481 Students.

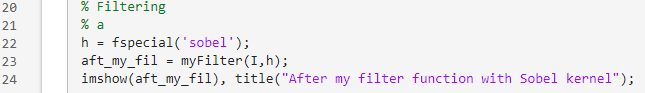


Res:



Filtering

a.



Text

Description automatically generated

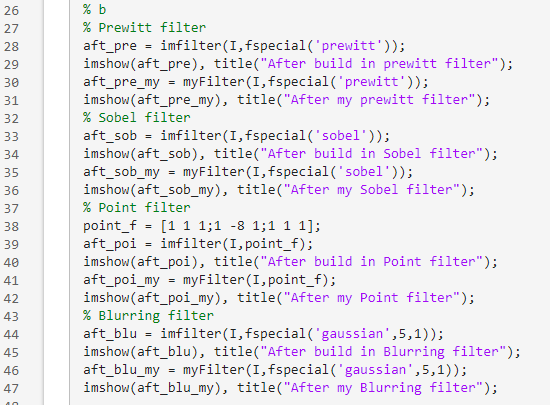
We can add zero padding to original image

Res:

A picture containing text

Description automatically generated

b.



Res:

Text

Description automatically generated

Text

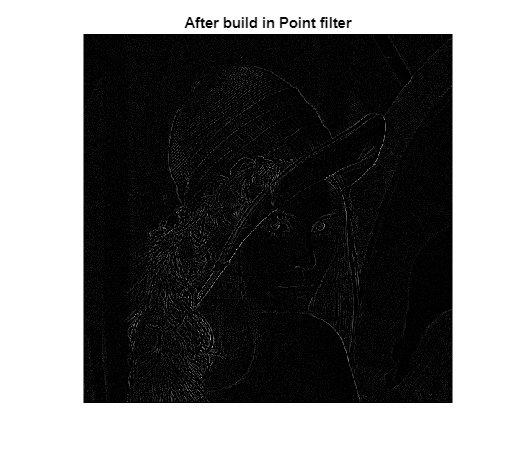
Description automatically generated

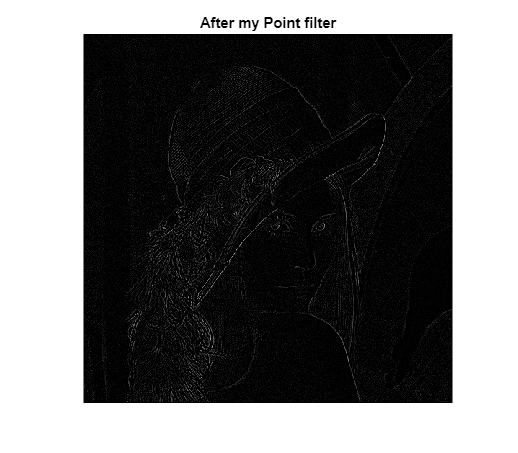
Text

Description automatically generated with low confidence

A picture containing text

Description automatically generated





A person wearing a hat

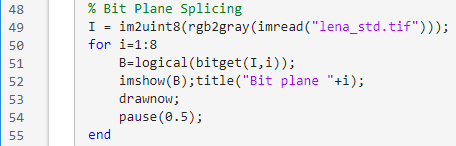
Description automatically generated with medium confidence

A person wearing a hat

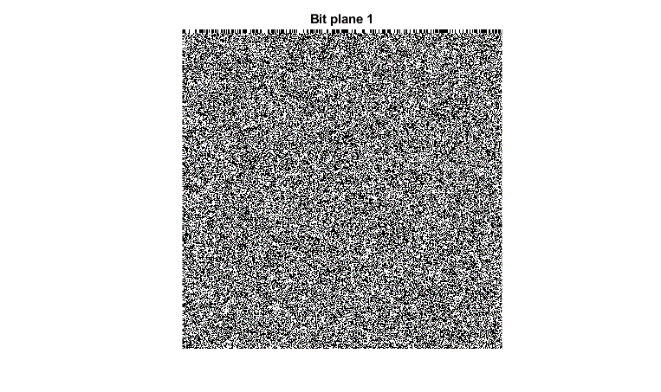
Description automatically generated with medium confidence

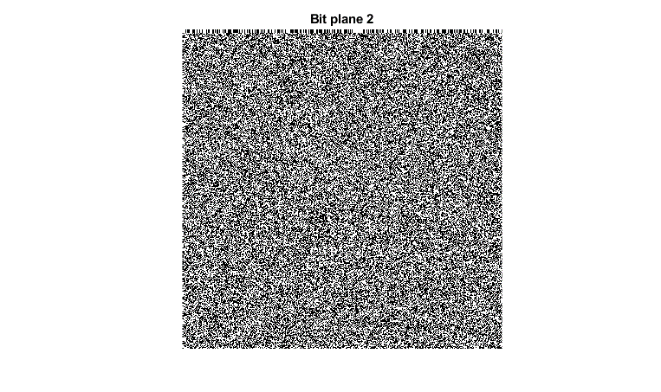
Those are almost same

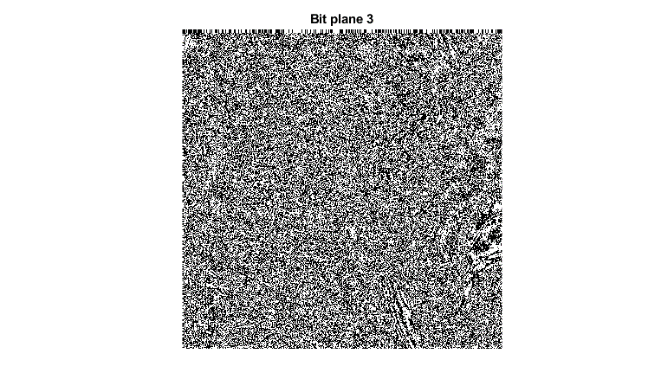
Bit Plane Splicing

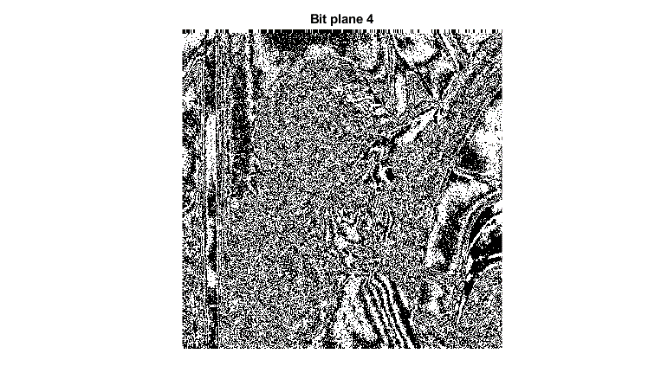


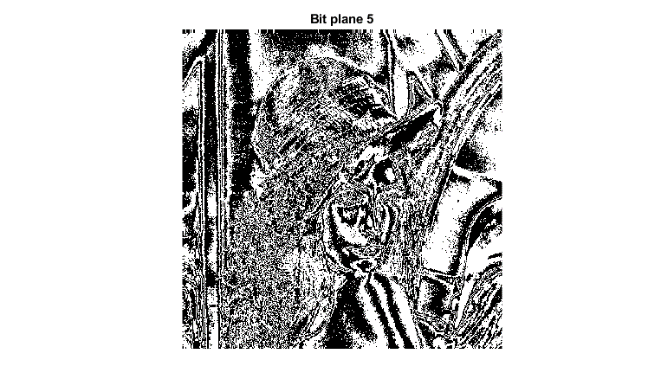
Res:

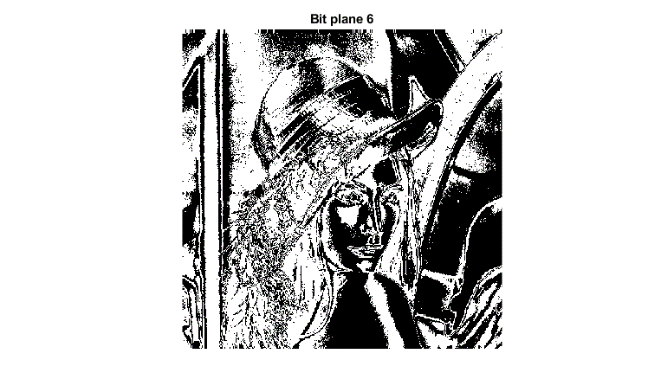
















% Histogram Equalization

% a

I = im2uint8(rgb2gray(imread("lena\_std.tif")));

imshow(I), title("Original Image");

% b

imhist(I), title("Histogram of Original Image");

% c

img\_he = histeq(I);

imhist(img\_he), title("Histogram of enhanced Image");

imshow(img\_he), title("Enhanced Image");

% local enhancement approach

ROI = [100,300,100,300];

loc\_he = I;

loc\_he(ROI(1):ROI(2),ROI(3):ROI(4)) = histeq(I(ROI(1):ROI(2),ROI(3):ROI(4)));

imshow(loc\_he), title("Local Enhanced Image");

% Filtering

% a

h = fspecial('sobel');

aft\_my\_fil = myFilter(I,h);

imshow(aft\_my\_fil), title("After my filter function with Sobel kernel");

% b

% Prewitt filter

aft\_pre = imfilter(I,fspecial('prewitt'));

imshow(aft\_pre), title("After build in prewitt filter");

aft\_pre\_my = myFilter(I,fspecial('prewitt'));

imshow(aft\_pre\_my), title("After my prewitt filter");

% Sobel filter

aft\_sob = imfilter(I,fspecial('sobel'));

imshow(aft\_sob), title("After build in Sobel filter");

aft\_sob\_my = myFilter(I,fspecial('sobel'));

imshow(aft\_sob\_my), title("After my Sobel filter");

% Point filter

point\_f = [1 1 1;1 -8 1;1 1 1];

aft\_poi = imfilter(I,point\_f);

imshow(aft\_poi), title("After build in Point filter");

aft\_poi\_my = myFilter(I,point\_f);

imshow(aft\_poi\_my), title("After my Point filter");

% Blurring filter

aft\_blu = imfilter(I,fspecial('gaussian',5,1));

imshow(aft\_blu), title("After build in Blurring filter");

aft\_blu\_my = myFilter(I,fspecial('gaussian',5,1));

imshow(aft\_blu\_my), title("After my Blurring filter");

% Bit Plane Splicing

I = im2uint8(rgb2gray(imread("lena\_std.tif")));

for i=1:8

B=logical(bitget(I,i));

imshow(B);title("Bit plane "+i);

drawnow;

pause(0.5);

end

function res = myFilter(I,h)

% We can add zero padding to original image

[img\_h,img\_w] = size(I);

hsize = size(h);

padsize = [((hsize(1)-1)/2),((hsize(2)-1)/2)];

padded\_img = zeros(img\_h+padsize(1)\*2,img\_w+padsize(2)\*2);

padded\_img(padsize(1)+1:img\_h+padsize(1), ...

padsize(2)+1:img\_w+padsize(2)) = I;

[padded\_h,padded\_w] = size(padded\_img);

for i=1:img\_h

for j=1:img\_w

copy(i+padsize(1),j+padsize(2)) = sum(sum( ...

padded\_img(i:i+hsize(1)-1,j:j+hsize(2)-1).\*h));

end

end

res = uint8(copy(1+padsize(1):padded\_h-padsize(1), ...

1+padsize(2):padded\_w-padsize(2)));

end