Name: Laya, Jhance Bredh C. Section: N32

**Exercises**

**A.**

1. Open the greyscale image cameraman.tif and view it. What data type is it?

uint8

2. Enter the following commands:

[em,map]=imread(’emu.tif’);

e=ind2gray(em,map);

These will produce a greyscale image of type double. View this image.



3. Enter the command

e2=im2uint8(e);

and view the output. What does the function im2uint8 do? What affect does it have on

the function converts the datatype of the image to uint8

(a) the appearance of the image?

It is still the same because it converts the image’s datatype to the datatype of itself.

(b) the elements of the image matrix?

The elements are still the same.

4. What happens if you apply im2uint8 to the cameraman image?

The im2uint8 doesn’t affect the image because it is converting to its original datatype.

Before im2uint8 After im2uint8



5. Experiment with reducing spatial resolution of the following images:

function bit\_slice\_it(I)

B = zeros(size(I));

ctr = 1;

for i = 1:8

B = bitset(B, 8,bitget(I, ctr));

B = uint8(B);

subplot(3,3,i), imshow(B); title(i);

ctr = ctr + 1;

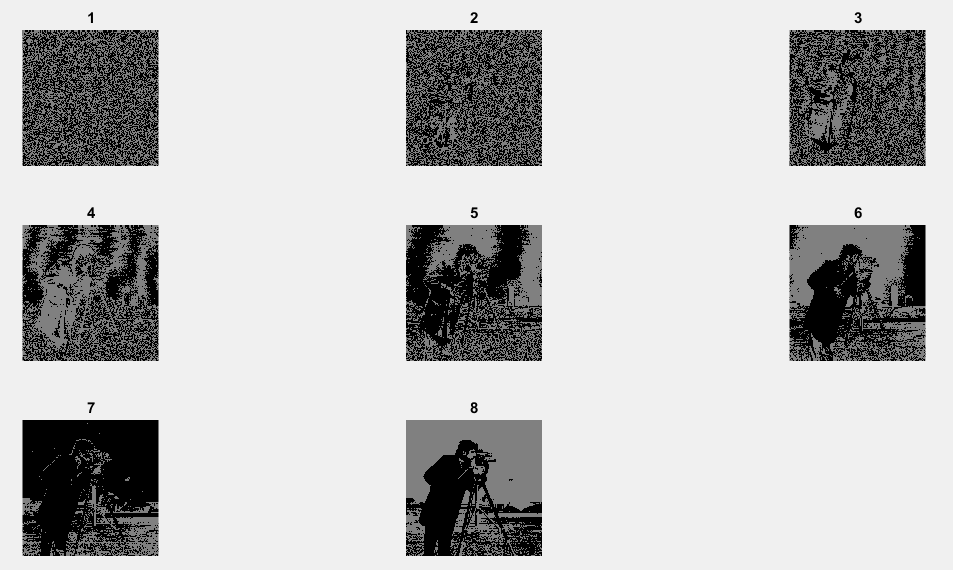
end

end

(a) cameraman.tif

I = imread(image\_here);

bit\_slice\_it(I);

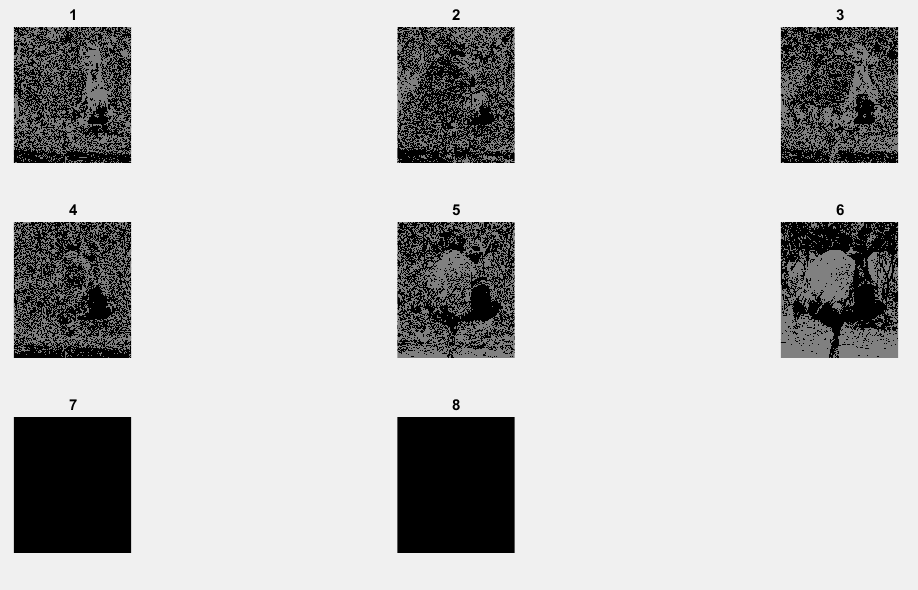


The image cannot be recognize starting at 2.

(b) The greyscale emu image

I = imread(image\_here);

bit\_slice\_it(I);



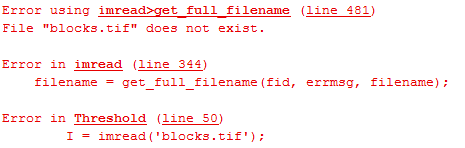
Grayscale image is difficult to recognized when bit-plane slice is applied.

(c) blocks.tif

I = imread(image\_here);

bit\_slice\_it(I);

error 404, image not found;

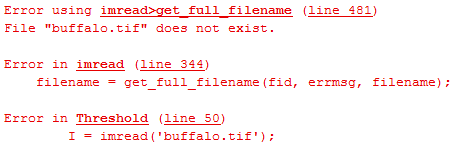


(d) buffalo.tif

I = imread(image\_here);

bit\_slice\_it(I);

error 404, image not found;



In each case note the point at which the image becomes unrecognizable.

**B.**

1. Search lenna.jpg (original)

Convert to the following:

1. [Photographic Negative](http://www.cs.uregina.ca/Links/class-info/425/Lab3/lesson.html#Negative)
2. [Gamma](http://www.cs.uregina.ca/Links/class-info/425/Lab3/lesson.html#Gamma)
3. [Logarithmic](http://www.cs.uregina.ca/Links/class-info/425/Lab3/lesson.html#Logarithm)
4. [Contrast Stretching](http://www.cs.uregina.ca/Links/class-info/425/Lab3/lesson.html#Contrast)
5. fspecial (sobel & prewitt)
6. bit splicing (Gray)

I = imread('lena.jpg');

%1

negative\_I = imcomplement(I);

subplot(4,4,1), imshow(negative\_I); title('Negative');

%2

gamma = 0.8;

gamma\_low = double(I).^gamma;

subplot(4,4,2), imshow(uint8(gamma\_low)); title('Gamma 0.8');

gamma = 1.2;

gamma\_high = double(I).^gamma;

subplot(4,4,3), imshow(uint8(gamma\_high)); title('Gamme 1.2');

%3

i = im2double(I);

J = 1\*log(1+i);

subplot(4,4,4), imshow(J); title('1\*log(1+image)');

J1 = 2\*log(1+i);

subplot(4,4,5), imshow(J1); title('2\*log(1+image)');

J2 = 5\*log(1+i);

subplot(4,4,6), imshow(J2); title('5\*log(1+image)');

%4

prewitt\_method = fspecial('prewitt');

prewitt\_implementation = imfilter(I,prewitt\_method);

subplot(4,4,7), imshow(prewitt\_implementation); title('prewitt');

sobel\_method = fspecial('sobel');

sobel\_implementation = imfilter(I,sobel\_method);

subplot(4,4,8), imshow(sobel\_implementation); title('sobel');

%5

ctr = 1;

Z = zeros(size(I));

for ind = 1:8

Z = bitset(Z, 8,bitget(I, ctr));

Z = uint8(Z);

subplot(4,4,ind+8), imshow(Z); title(ind);

ctr = ctr + 1;

end

