**Digital Image Processing Sessional 4**

**150222**

**EXERCISE 1**

This problem was about processing Satellite image. The satellite image of a lake, ‘lake.jpg’ was downloaded for this purpose.

1. This part of the problem told to measure the surface area of the lake in pixels. The source image ‘lake.jpg’ was taken using imread(). Then the image was converted from RGB to LAB. After that, only the blue plane of the converted image was taken to extract the blue plane. The blue plane image was then converted into BW image by applying im2bw(). The resultant image was then inverted/complemented and bwareafilt() was applied by using the condition to get the biggest object. Finally, regionprops() was used to get the total number of pixels.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

It is seen that, converting the image into LAB and taking the blue plane, the image has become a gray image where blue colored area is darker and the left area is lighter. Then converting the image makes the dark area black and the rest white. After that, after inverting the image, there were supposed to be some discrete small area which were filtered out and the one biggest object was shown.

Now using regionprops(), the number of white pixels is found which could also be done by using sum(image(:)). The total number of white pixels is **26418**.

1. For this part, the height and width of the image was given as 6.901Km and 5.258 Km respectively. The area and the perimeter were to be calculated. For this reason, the height and width of the image were calculated by using size(). Now by calculating, the area was found in square Km. For the perimeter it was necessary to found the perimeter of the image which was done by using edge() and it could also be done using bwperim(). The number of pixels could be counted from here but as there were holes so imfll() was used to remove those small areas. Then using sum(image(:)) the total number of white pixels was counted and the perimeter was calculated.

|  |  |
| --- | --- |
|  |  |

For calculating area, per pixel area was calculated and then it was multiplied by the pixels number. The area found was **7.4482** square kilometers.

For perimeter, the per pixel value could be calculated from any axis X or Y.

By considering X axis the perimeter is **21.6199** kilometers.

By considering Y axis the perimeter is **23.5558** kilometers.

**CODE :**

I = imread('Lake.jpg');

imshow(I),title('Original Lake image');

C = makecform('srgb2lab');

lab = applycform(I,C);

figure,imshow(lab(:,:,3)),title('image in b plane');

Ib = im2bw(lab(:,:,3));

figure,imshow(Ib),title('thresholded image');

In = ~Ib;

bw = bwareafilt(In,1);

figure,imshow(bw),title('mask image of the lake');

numpixels1 = regionprops(bw,'Area');

xkm = 6.901;

ykm = 5.258;

[x y] = size(bw);

area = ((xkm\*ykm)/(x\*y))\*numpixels1.Area;

Ic = edge(bw,'canny');

figure,imshow(Ic),title('showing perimeter of the lake');

If = imfill(bwareafilt(Ic,1),'holes');

figure,imshow(If),title('actual perimeter');

numpix = sum(If(:));

perimeterx = (xkm/x)\*numpix;

perimetery = (ykm/y)\*numpix;

**EXERCISE 2**

This problem was about detecting edges of objects in an image. For this purpose, ‘bricks.jpg’ was provided.

1. This part told to detect all the edges from the image. After reading the image using imread(), the image was converted to gray image using rgb2gray(). The edge() and ‘Canny’ method was used to mark the edges.

|  |  |
| --- | --- |
|  |  |

By experimentation it was found that, Canny was better for finding edges of sharp objects where nothing else give as much close edge detection and for threshold value of **0.06** the result obtained was relatively better. But it was really hard to get only the edges of objects as shadows of those objects were also shown with edges. Detailing were not properly found in the edges.

1. For this part of the problem, firstly, the edges of blue objects and secondly the edges of the red objects were to found and finally all the edges of both blue and red objects were to found. For doing this the image was converted to LAB format. Then Blue and Red plane were extracted differently. Then for each panel after applying graythresh(), im2bw was used and finally after filtering with bwareafilt() to get the certain amount of objects edge() was used. Finally the resultant images were merged with imadd() to show final image.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Not much changes were seen by changing threshold so threshold was not used while finding out edges. Different color edges can form a perfect shape if they are merged together.

**CODE :**

I = imread('bricks.jpg');

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

I2 = edge(rgb2gray(I),'canny');

figure,imshow(I2),title('All edges of the image');

C = makecform('srgb2lab');

lab = applycform(I,C);

b = lab(:,:,3);

a = lab(:,:,2);

level=graythresh(b);

BW1=edge(bwareafilt(im2bw(b,level),6),'Canny');

figure,imshow(BW1),title('Edges of Blue Blocks');

level=graythresh(a);

BW2=edge(bwareafilt(im2bw(a,level),6),'Canny');

figure,imshow(BW2),title('Edges of Red Blocks');

BW=imadd(BW1,BW2);

figure,imshow(BW),title('Blue+Red edges');

**EXERCISE 3**

This problem was to detect same kind of objects and count their numbers. For this problem ‘coins.png’ and ‘GLUCOSE.tif’ were provided and it was told to count the number of coins and glucose atom from the images. The same process was applied for the two images.

Firstly, the images was read with imread(). Then the images were converted using im2bw(). After that imfill() was used to fill the holes in the BW image. Then finally the bwlabel() was used to get the count. The result of the count was displayed with fprintf().

|  |  |
| --- | --- |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |

AS ‘coins.png’ had darker background then the coins so applying im2bw had turned the coins white and the background black and the opposite case has happened for the ‘glucose.tif’. In coins after conversion there were holes so the holes were filled but for glucose atoms no holes were found so nothing was there to fill. If the holes were not filled then the correct number is not found. But as bwlabel is able to detect white objects so glucose was inverted/complemented. And finally as the sample output had the coins and glucose atoms colored so, those were assigned color by using label2rgb.

And the output for the count is

|  |
| --- |
|  |

**CODE :**

I = imread('coins.png');

imshow(I),title('coins.png');

Ibin = im2bw(I);

figure,imshow(Ibin),title('binary');

Ifill = imfill(Ibin,'holes');

figure,imshow(Ifill),title('hole filled');

[L,N] = bwlabel(Ifill);

figure,imshow(label2rgb(L)),title('colored');

fprintf('total number of coins in coins.png image is %i.\n',N);

I2 = imread('GLUCOSE.tif');

figure,imshow(I2),title('GLICOSE.tif');

Ibin2 = im2bw(I2);

figure,imshow(Ibin2),title('Binary');

figure,imshow(~Ibin2),title('Inverted');

[L2,N2] = bwlabel(~Ibin2);

figure,imshow(label2rgb(L2)),title('colored');

fprintf('total number of atoms in the image of the glucose molecule is %i.\n',N2);