EXP NO: 14
DATE:

FUZZY LOGIC - IMAGE PROCESSING

AIM:-

The fuzzy logic approach for image processing allows you to use membership function to define the degree to which a pixel belongs to an edge or uniform.

Import RGB Image and convert to Grayscale:-

Import the Image
Irgb = imread('peppers.png'
Irgb is a 384 X 512 X 3 uint 8 array. The Three channels of IRGB represent red, green, Blue.
convert IRGB to grayscale so that you can work with a 2-D array instead of 3D array
Igray = rgb2gray(Irgb);
figure
image(Igray, 'CDataMapping', 'scaled'),
colormap('gray')
title('Input Image in Grayscale')

Convert Image to Double precision Data:-

The fuzzy inference systems supports only single precision and double precision.
Therefore convert Igray to double array using the im2double function
I = im2double(Igray)
obtain Image gradient

```
Gx = [-1 1];
Gy = Gx';
Ix = conv2(I, Gx, 'same');
Iy = conv2(I, Gy, 'same');
```

Plot the image gradients

figure
image (Ix, 'CDataMapping', 'scaled')
colormap ('gray')
title ('Ix')

figure
image (Iy, 'CDataMapping', 'scaled')
colormap ('gray')
title ('Iy')

Define Fuzzy Inference system (FIS) for edge detection
create a fuzzy inference system for edge detection.
edgeFIS = mamfis ('Name', 'edgeDetection');
edge FIS = addInput (edge FIS, [-1 1], 'Name', 'Ix');
edge FIS = add Input (edge FIS, [-1 1] 'Name', 'Iy');

specify zero - mean Gaussian membership function for each input, If the gradient value for a pixel 0, then it belongs to zero with 1

sx = 0.1;
sy = 0.1;

sx and sy specify the standard deviation for the zero membership function for Ix and Iy inputs.
specify the intensity of edge-detected image as an output of edge FIS
edge FIS = add output (edge FIS, [0 1], 'Name', 'Iout');

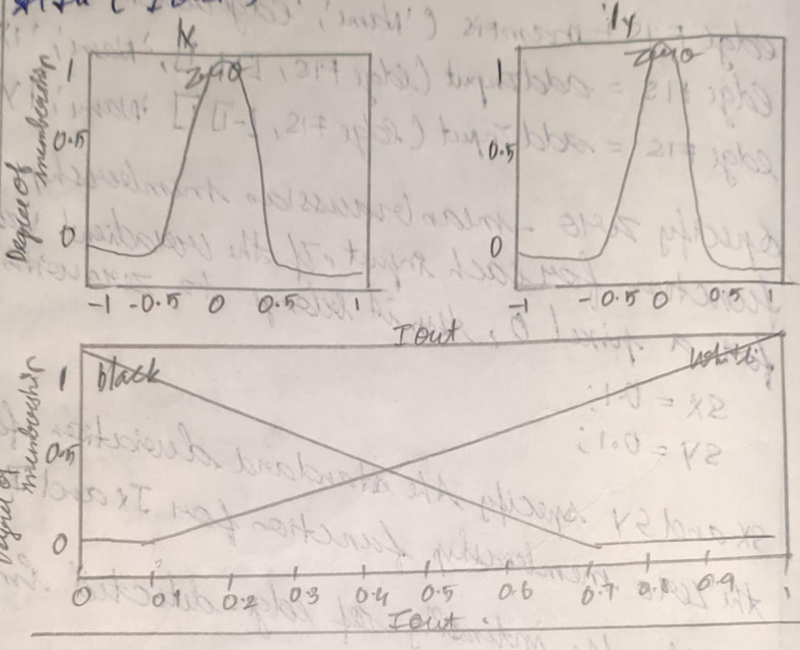
Wa = 0.1;
Wb = 1;
Wc = 1;
ba = 0;
bb = 0;
bc = 0.7;

edge FIS = add MF (edge FIS, 'Iout', 'trimf', [Wa Wb Wc], 'Name', 'white');

edge FIS = addMF (edge FIS, 'Iout', 'trimf', [ba bb bc], 'Name', 'black').

Plot the membership of the inputs and outputs of FIS.

```
figure
subplot (2, 2, 1)
plotmf (edge FIS, 'input', 1)
title ('Ix')
subplot (2,2,2)
plotmf (edgeFIS, 'input', 2)
title ('Iy')
subplot (2,2,[3 4])
plotmf(edgeFIS, 'output', 1)
title ('Iout')
```



Specify FIS rules.

```
r1 = "If Ix is zero and Iy is zero then Iout is white";
r2 = "If Ix is not zero and Iy is not zero then Iout is black";
edge FIS = add rule (edge FIS , [r1 r2]);
edge FIS. rules ans = 1x2 fis rule ans = 1x2
```

1x2 fisrule array with properties:-

Description:-

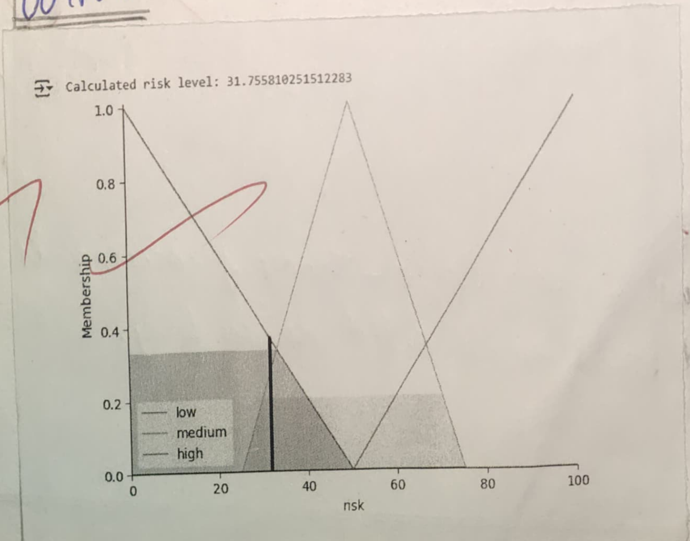
Antecedent
consequent
weight
connection
Details:
Description

1 "Ix == zero && Iy== zero => Iout = white(1)"
2 "Ix ~= zero || Iy ~= zero => Iout = black(1)"

Evaluate FIS
Evaluate the output of the edge detector for each rows in pixeles in I.
Ix and Iy as inputs.
Ieval = zeros(size(I));
for ii = 1: size(I,1)
Ieval(ii,:) = evalfis(edgeFIS,[Ix(ii,:)'; (Iy(ii,:))']);
end

Plot results
Plot the original grayscale image
figure
image(I,'CDataMapping','scaled')
colormap('gray')
title('Original Grayscale Image').

OUTPUT:-



Uploaded Image :-

Gray scale Processing :-

RESULT:

Thus fuzzy logic - Image processing is executed and implemented successfully.