```
clc
clear all
close all
Irgb = imread('peppers.png');
Igray=rgb2gray(Irgb);
imshow(Igray);
I = double(Igray);

classType = class(Igray);
scalingFactor = double(intmax(classType));
I = I/scalingFactor;
```

Obtain Image Gradient

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

Define Fuzzy Inference System (FIS) for Edge Detection

```
edgeFIS = newfis('edgeDetection');
```

Specify the image gradients, `Ix` and `Iy`, as the inputs of `edgeFIS`.

```
edgeFIS = addvar(edgeFIS,'input','Ix',[-1 1]);
edgeFIS = addvar(edgeFIS,'input','Iy',[-1 1]);
```

Specify a zero-mean Gaussian membership function for each input.

```
sx = 0.1;
sy = 0.1;
edgeFIS = addmf(edgeFIS,'input',1,'zero','gaussmf',[sx 0]);
edgeFIS = addmf(edgeFIS,'input',2,'zero','gaussmf',[sy 0]);
```

Specify the intensity of the edge-detected image as an output of `edgeFIS`.

```
edgeFIS = addvar(edgeFIS,'output','Iout',[0 1]);

wa = 0.1;
wb = 1;
wc = 1;
ba = 0;
bb = 0;
bc = 0.7;
edgeFIS = addmf(edgeFIS,'output',1,'white','trimf',[wa wb wc]);
edgeFIS = addmf(edgeFIS,'output',1,'black','trimf',[ba bb bc]);
```

```
figure
subplot(2,2,1)
plotmf(edgeFIS,'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')
```

Add rules to make a pixel white if it belongs to a uniform region. Otherwise, make the pixel black.

```
r1 = 'If Ix is zero and Iy is zero then Iout is white';
r2 = 'If Ix is not zero or Iy is not zero then Iout is black';
r = char(r1,r2);
edgeFIS = parsrule(edgeFIS,r);
showrule(edgeFIS)
```

Evaluate the output of the edge detector for each row of pixels in `I` using corresponding rows of `Ix` and `Iy` as inputs.

```
Ieval = zeros(size(I));
for ii = 1:size(I,1)
    Ieval(ii,:) = evalfis([(Ix(ii,:));(Iy(ii,:));]',edgeFIS);
end

figure
image(I,'CDataMapping','scaled')
colormap('gray')
title('Original Grayscale Image')

figure
image(Ieval,'CDataMapping','scaled')
colormap('gray')
title('Edge Detection Using Fuzzy Logic')
```

Original Image

Image Ix

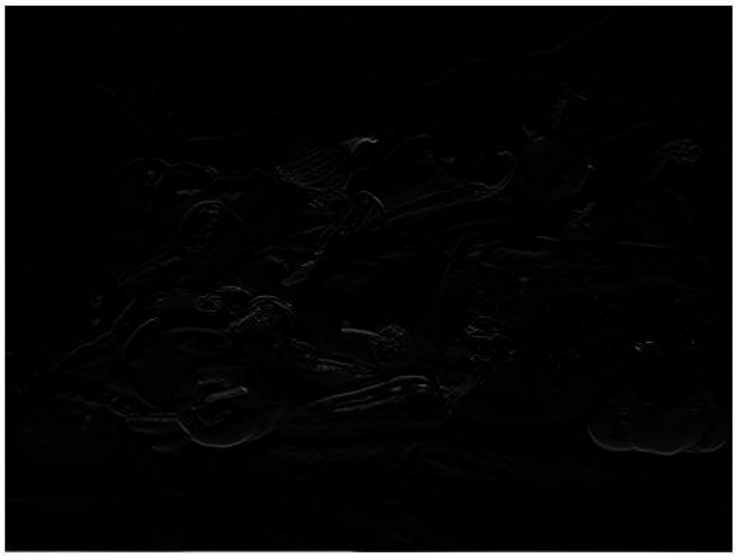
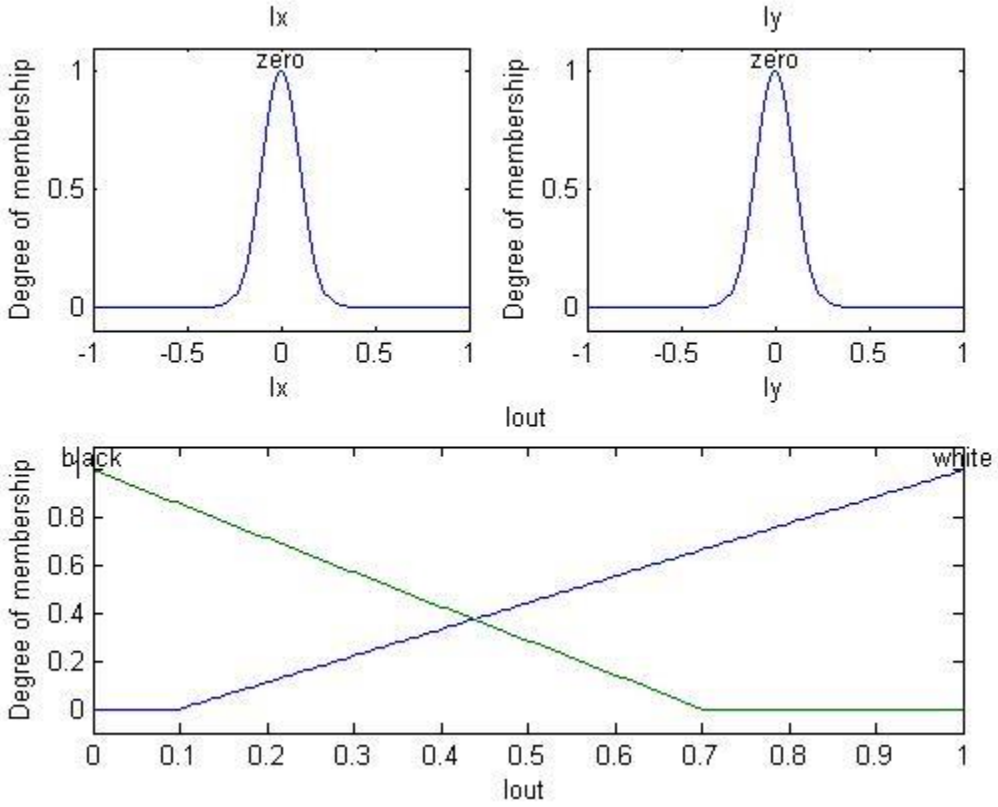
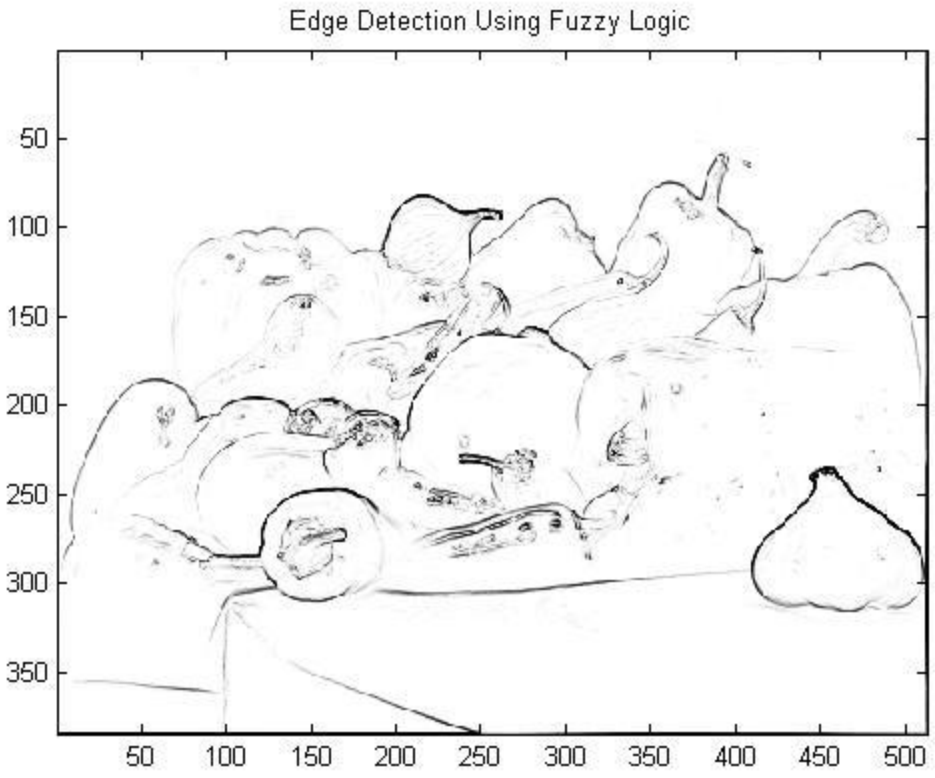


Image Iy



Membership Functions



Output Image