LAB - 12

Name	Keval D Gandevia
Roll Number	CE046
ID	19CEUEG017
Subject	Image Processing

Aim: Performing Image Segmentation

Q. 1: Apply laplacian mask [-1 -1 -1;-1 8 -1;-1 -1 -1] to the 'wirebond.tif' image and get the following images. (use conv2 and imfilter to filter the images and see the difference). a. Laplacian of the image. b. Absolute value of the laplacian. c. Positive values of the laplacian.

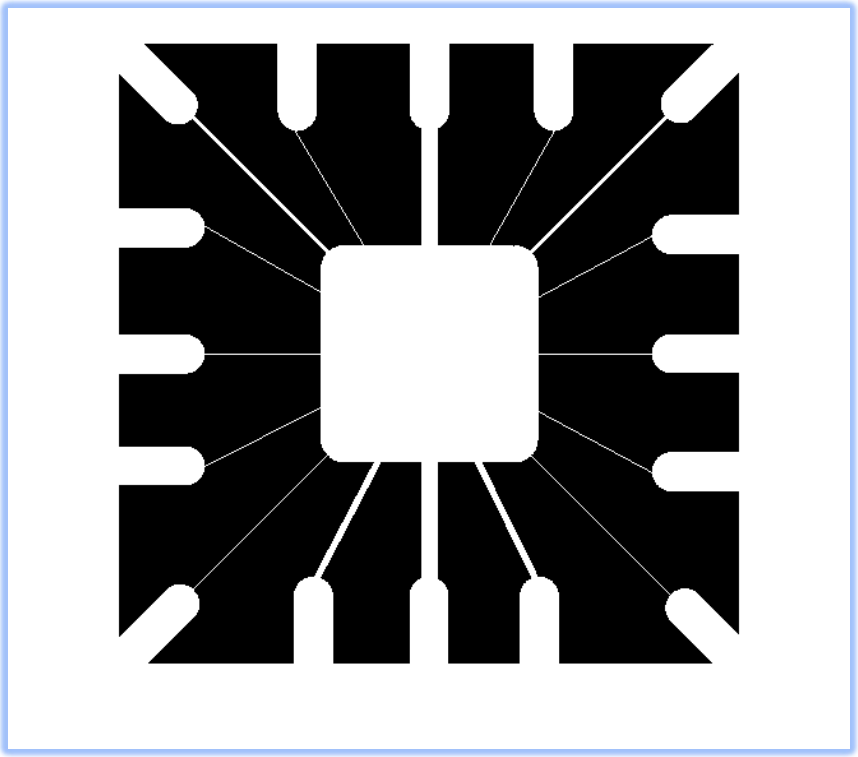
❖ **Code:**

```
img = imread('wirebond.tif');
[m, n] = size(img);
figure;
imshow(img);

mask = [-1 -1 -1;-1 8 -1;-1 -1 -1];

laplacianImg = conv2(img, mask);
% filterredImg = imfilter(img, mask);

figure;
```

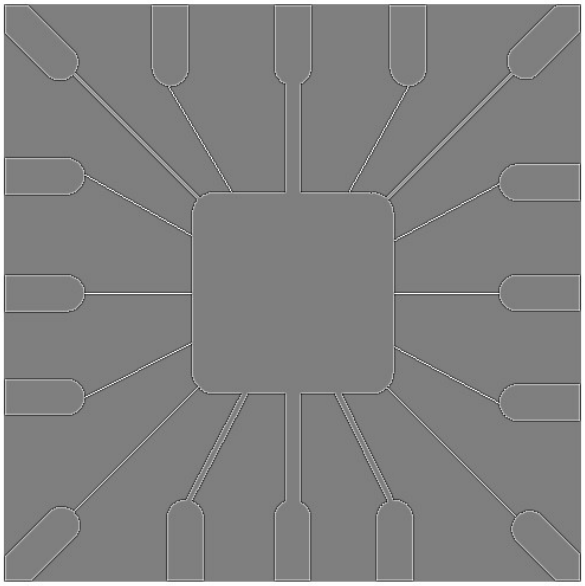
```
imshow(laplacianImg, []);

figure;
imshow(abs(laplacianImg));


for i = 1 : m
   for j = 1 : n
      if(laplacianImg(i, j) < 0)
         new_img(i, j) = 0;
      else
         new_img(i, j) = laplacianImg(i, j);
      end
   end
end

figure;
imshow(new_img);
```

❖ Output:

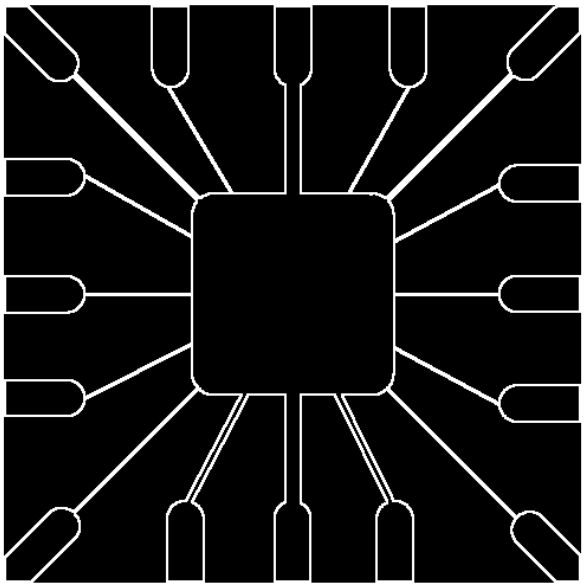
➢ Original Image:



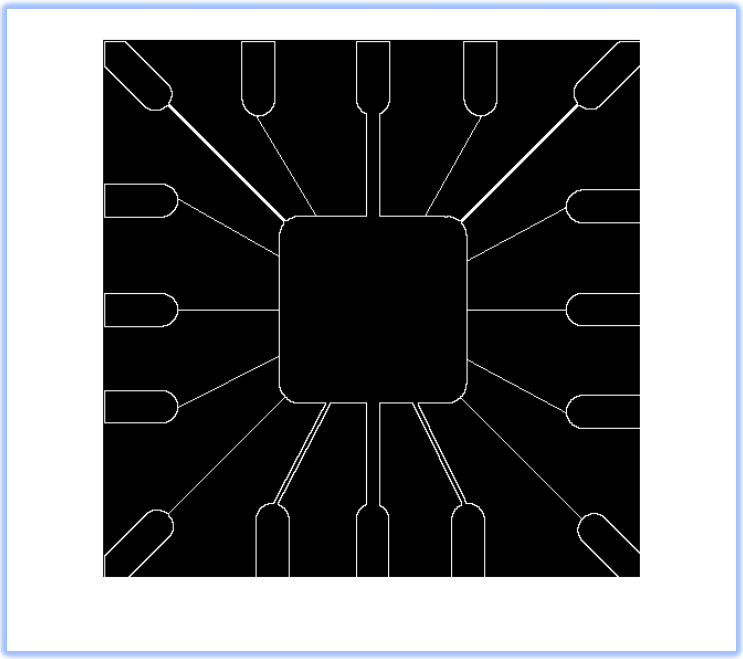
- **Laplacian of the image:**



- **Absolute value of Laplacian:**



➢ **Positive Values of Laplacian:**



Q. 2: Apply gradient filter (Sobel filter) to the image 'building.tif' and output a. Gradient in x direction b. Gradient in y direction c. Absolute Gradient.

❖ **Code:**

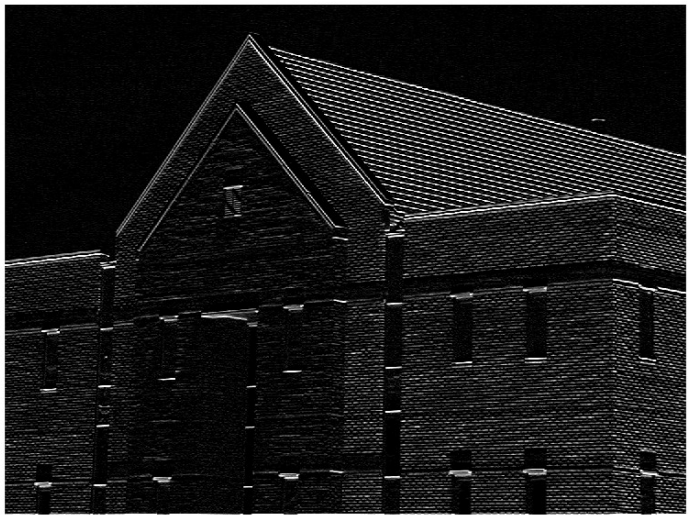
```
img = imread('building.tif');
figure;
imshow(img);

gx = [1 2 1; 0 0 0; -1 -2 -1];
gy = [-1 0 1; -2 0 2; -1 0 1];

xImg = imfilter(img, gx);
yImg = imfilter(img, gy);

magnitude = xImg + yImg;

figure;
imshow(xImg);

figure;
```

```
imshow(yImg);

figure;
imshow(magnitude);
```

❖ Output:

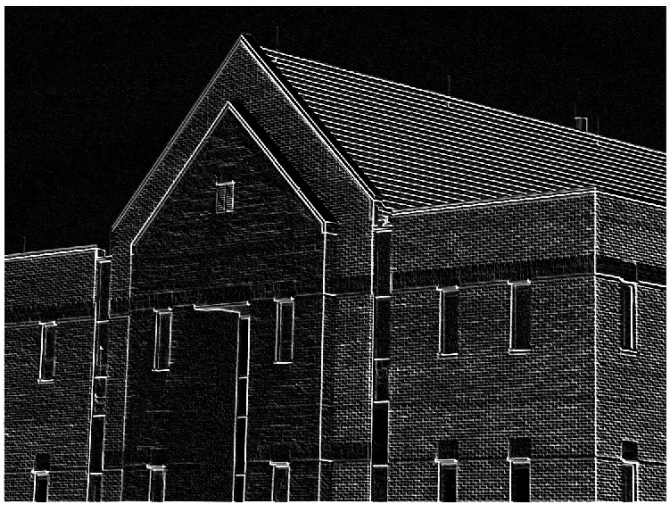
➢ Original Image:

➢ Gradient in x direction:



- **Gradient in y direction:**

➢ **Magnitude of Gradient:**



Q. 3: Implement Otsu's Method to segment 'balloons.jpg' and compare your result with the in-built function graythresh(Image). If both implementations give same result 'ThumbsUp.jpg' should be displayed otherwise 'ThumbsDown.jpg' should be displayed.

❖ **Code:**

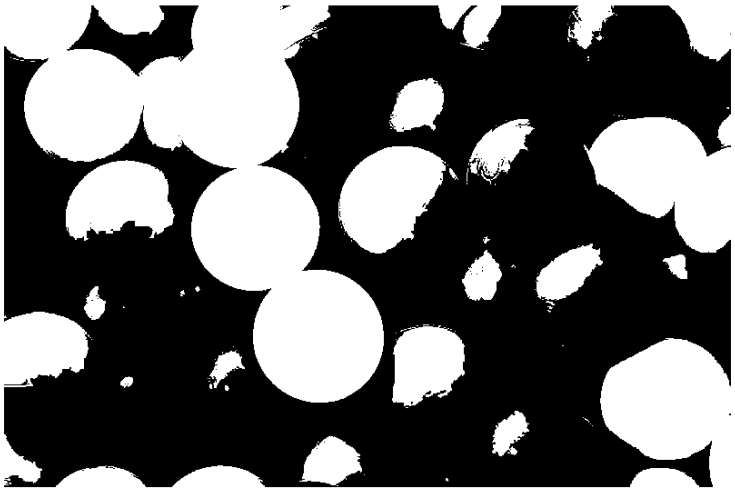
```
clear all;
img = imread('balloons.jpg');
img = rgb2gray(img);
[m, n] = size(img);
values = linspace(0, 255, 256);
hist_array = imhist(img);

figure;
imshow(img);

min = 99999999;
min_index = -1;

for i = 2 : 254
   left = hist_array(1 : i);
```

```
    right = hist_array(i+1 : 255);
    wl = sum(left) / (m * n);
    wr = sum(right) / (m * n);

    ml = sum(dot(left, values(1:i))) / sum(left);
    mr = sum(dot(right, values(i+1:255))) / sum(right);

    vl = sum(dot(power(values(1:i) - ml, 2), left)) / sum(left);
    vr = sum(dot(power(values(i+1:255) - mr, 2), right)) / sum(right);

    within_class_var = wl * vl + wr * vr;

    if within_class_var < min
        min = within_class_var;
        min_index = i;
    end
end

for i = 1 : m
    for j = 1 : n
        if(img(i, j) < min_index)
            output(i, j) = 0;
        else
            output(i, j) = 1;
    end
    end
end

figure;
imshow(output);

L = graythresh(img);
new_img = im2bw(img, L);

figure;
imshow(new_img);

if(output == new_img)
    figure;
    imshow('ThumbsUp.jpg');
else
    figure;
    imshow('ThumbsDown.jpg');
end
```

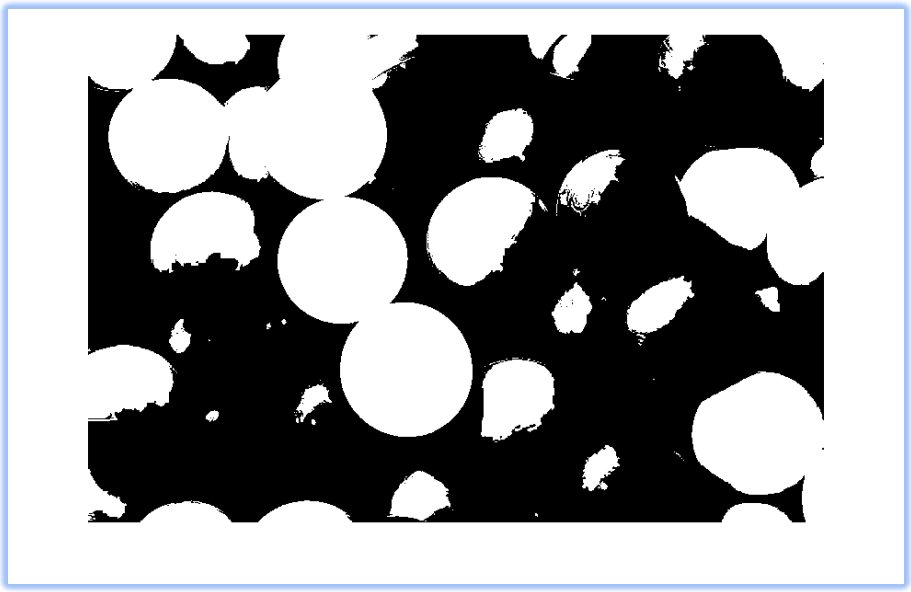
❖ **Output:**

➢ **Original Image:**

➢ **Otsu Implementation:**



- **In-built function:**



- **Comparison:**