Machine Learning for Speech and Computer Vision

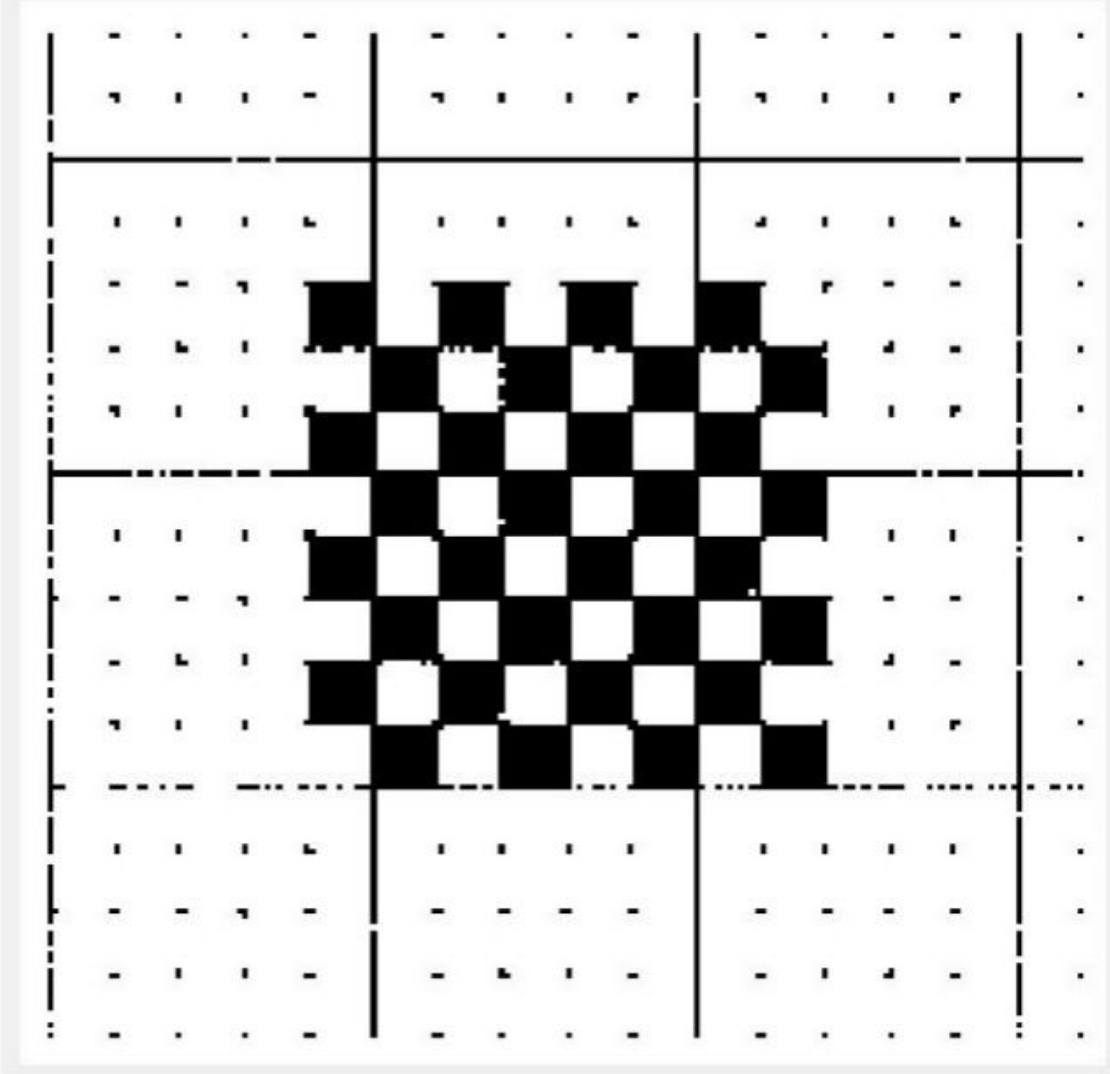
Assignment-1

Name:D.NaveenNischal
Rollno:18BEC011

1)Matlab code:

```
  Clear all;
 Close all;
  Clc
img = imread("Checkerbox1.jpg");
img = rgb2gray(img);
 imshow(img);
img1 = imbinarize(img, 0.25 * graythresh(img));
imshow(img1);
```

Output:



2)Matlab Code:

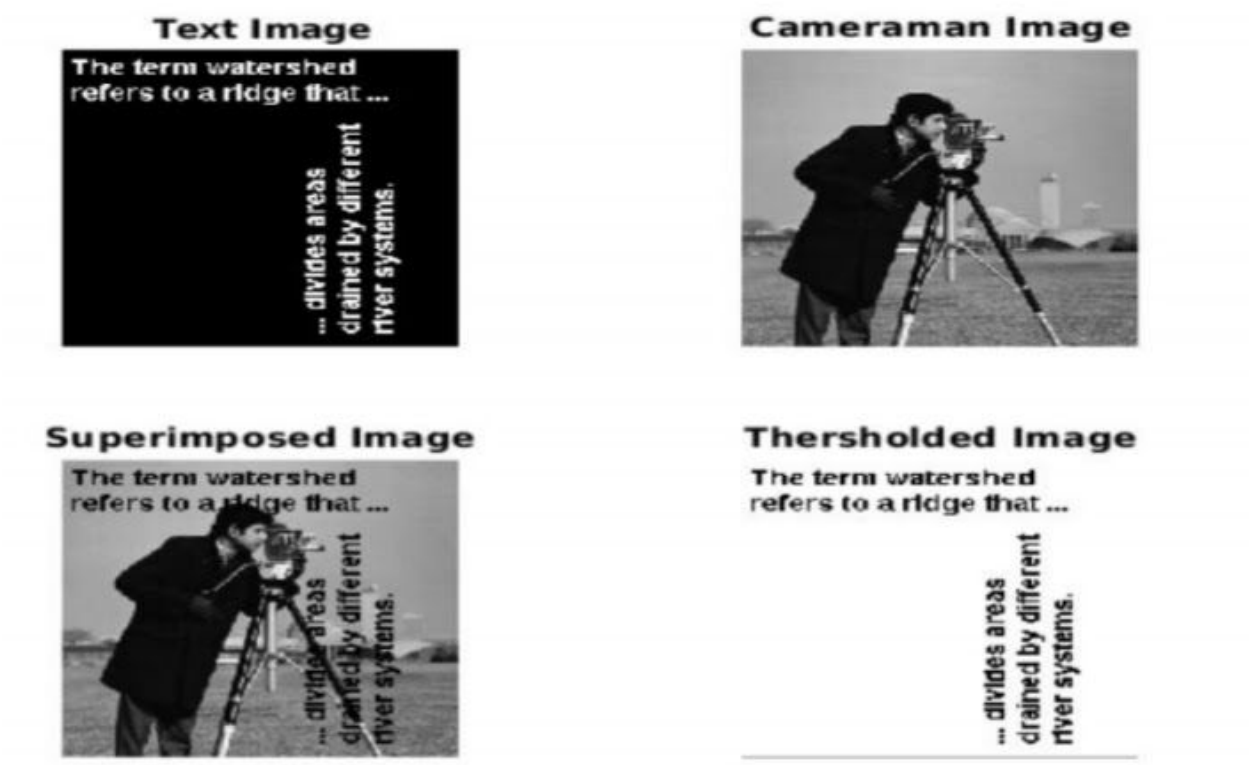
```
t = imread("text.tif");
c = imread("cameraman.tif");
m = uint8(double(c)+255*double(t));
thresholdValue = 250; % varies between [0 255]
output_image = max(m , thresholdValue);
output_image(output_image == round(thresholdValue))=0;
subplot(2,2,1);imshow(t);title("Text Image");
subplot(2,2,2);imshow(c);title("Cameraman Image");
subplot(2,2,3);imshow(m);title("Superimposed Image");
subplot(2,2,4);imshow(output_image);
title("Threshold Image");
```

Output:

3)Matlab Code:

```
t = imread("text.tif");
c = imread("cameraman.tif");
m = uint8(double(c).*double(~t));
thresholdValue = 5; % varies between [0 255]
output_image = min(m , thresholdValue);
output_image(output_image == round(thresholdValue)) = 255;
subplot(2,2,1);imshow(t);title("Text Image");
subplot(2,2,2);imshow(c);title("Cameraman Image");
subplot(2,2,3);imshow(m);title("Superimposed Image");
subplot(2,2,4);imshow(output_image);title("Threshold Image");
```

Output:

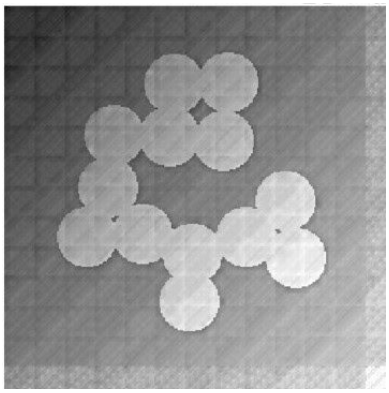


4)Matlab code:

```
clc
clear all
close all
t = imread('circles.tif');
imshow(t)
[x,y] = meshgrid(1:256,1:256);
t2 = double(t).*((x+y)/2+64)+x+y;
t3 = uint8(255*mat2gray(t2));
[m,n] = size(t3)
fun = @(block_struct) adaptthresh(block_struct.data);
T = blockproc(t3,[20 20],fun)
figure,imshow(T)
fun2 = @(block_struct) adapthisteq(block_struct.data);
T2 = blockproc(t3,[20 20],fun2)
figure,imshow(T2)
```

Output:

:



5)Matlab Code:

```
close all;
clear all;
clc;

c = imread("cameraman.tif");
c1 = imnoise(c,'salt & pepper',0.1);
c2 = imnoise(c,'gaussian',0,0.02);

%Sobel Filter
sobel1 = edge(c1,'Sobel');
sobel2 = edge(c2,'Sobel');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(sobel1);title("Sobel filter on c1(Salt and peper noise)");
subplot(2,2,4);imshow(sobel2);title("Sobel filter on c2(Gaussian noise)");

%Roberts
roberts1 = edge(c1,'Roberts');
roberts2 = edge(c2,'Roberts');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(roberts1);title("Roberts filter on c1(Salt and peper noise)");
subplot(2,2,4);imshow(roberts2);title("Roberts filter on c2(Gaussian noise)");

%Prewitt
prewitt1 = edge(c1,'Prewitt');
prewitt2 = edge(c2,'Prewitt');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(prewitt1);title("Prewitt filter on c1(Salt and peper noise)");
subplot(2,2,4);imshow(prewitt2);title("Prewitt filter on c2(Gaussian noise)");
%Laplacian
log1 = edge(c1,'log');
log2 = edge(c2,'log');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(log1);title("Laplacian of Gaussian(LoG) filter on c1");
subplot(2,2,4);imshow(log2);title("Laplacian of Gaussian(LoG) filter on c2");
```

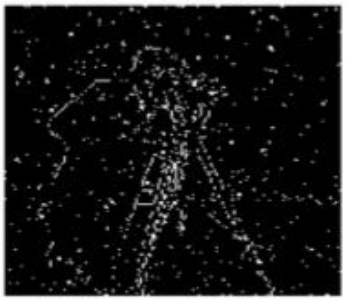
```
 %Canny
canny1 = edge(c1,'Canny');
canny2 = edge(c2,'Canny');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(log1);title("Canny filter on c1");
subplot(2,2,4);imshow(log2);title("Canny filter on c2");
%zerocross
zerocross1 = edge(c1,'zerocross');
zerocross2 = edge(c2,'zerocross');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(zerocross1);title("zerocross filter on c1");
subplot(2,2,4);imshow(zerocross2);title("zerocross filter on c2");
%approxcanny
approxcanny1= edge(c1,'approxcanny');
approxcanny2= edge(c2,'approxcanny');
subplot(2,2,1);imshow(c1);title("Salt & peper noise");
subplot(2,2,2);imshow(c2);title("Gaussian noise");
subplot(2,2,3);imshow(approxcanny1);title("approxcanny filter on c1");
subplot(2,2,4);imshow(approxcanny2);title("approxcanny filter on c2");
```

Outputs:1)Sobel Filter:

Salt & peper noise

Gaussian noise

Sobel filter on c1(Salt and peper noise)



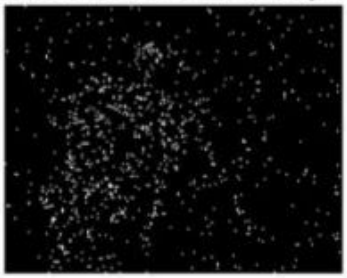
Sobel filter on c2(Gaussian noise)

2)Roberts Filter:

Salt & peper noise

Gaussian noise

Roberts filter on c1(Salt and peper noise)



Roberts filter on c2(Gaussian noise)

3)Prewitt filter

Salt & peper noise

Gaussian noise

Prewitt filter on c1(Salt and peper noise)

Prewitt filter on c2(Gaussian noise)

4)Laplacian Filter:

Salt & peper noise

Gaussian noise

Laplacian of Gaussian(LoG) filter on Salt & peper noise

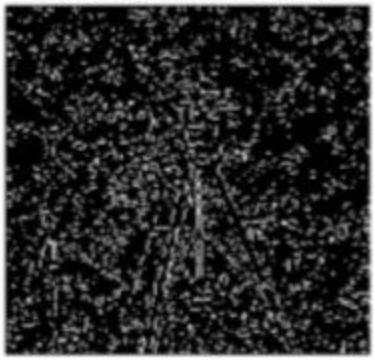
Laplacian of Gaussian(LoG) filter

5)Canny Filter:

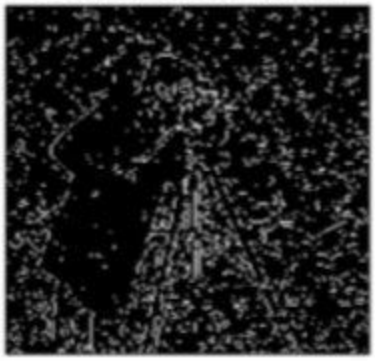
Salt & peper noise

Gaussian noise

Canny filter on c1



Canny filter on c2

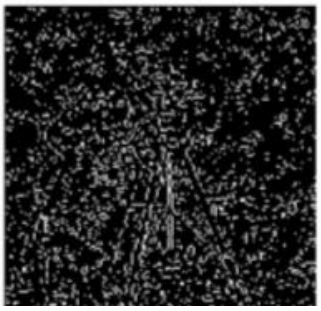


6)Zero cross Filter:

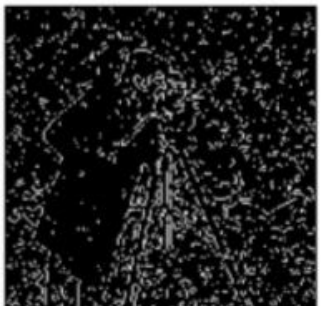
Salt & peper noise

Gaussian noise

zerocross filter on c1



zerocross filter on c2



7)Approx canny Filter:

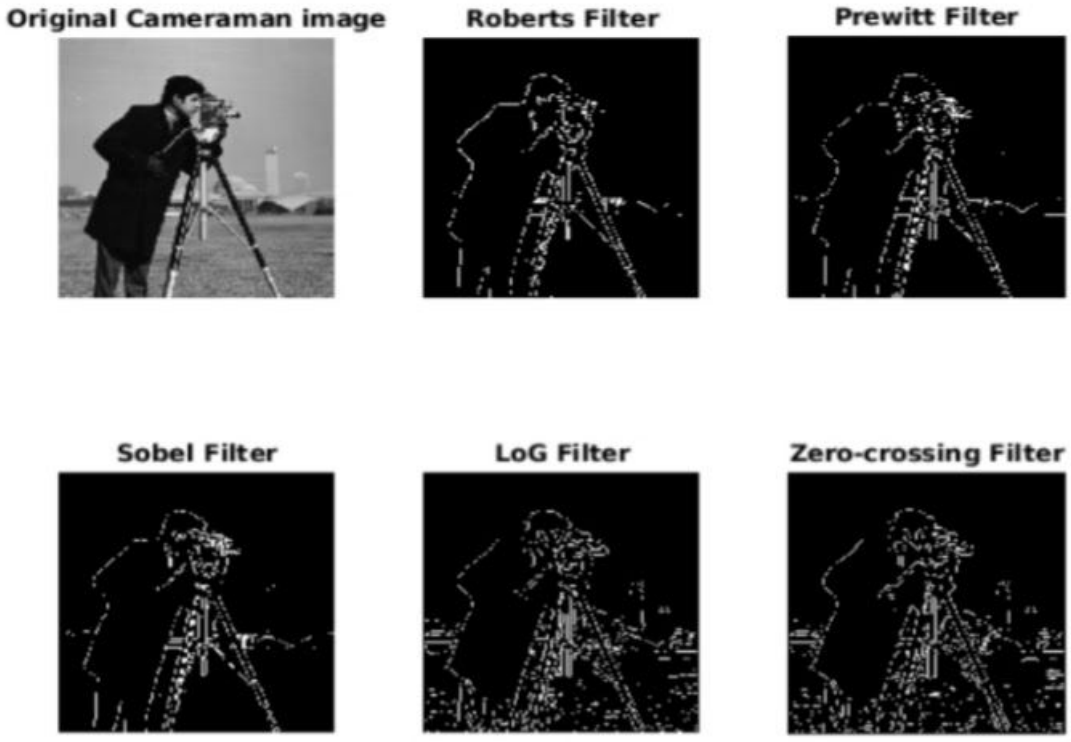
Observations:

As per above outputs Sobel and Prewitt Filter techniques produced the best results in the presence of noise. Both these techniques detected edges more accurately . Roberts ,Laplacian of Gaussian, Canny, Approxcanny and Zerocross Filter techniques produced the worst results in the presence of noise.

6A)Matlab Code:

```
clear all;
close all;
clc;
c=imread("cameraman.tif");
roberts = edge(c,'Roberts');
prewitt = edge(c,'Prewitt');
sobel = edge(c,'Sobel');
log = edge(c,'log');
zerocross=edge(c,'zerocross');
subplot(2,3,1);imshow(c);title("Original Cameraman image");
subplot(2,3,2);imshow(roberts);title("Roberts Filter");
subplot(2,3,3);imshow(prewitt);title("Prewitt Filter");
subplot(2,3,4);imshow(sobel);title("Sobel Filter");
subplot(2,3,5);imshow(log);title("LoG Filter");
subplot(2,3,6);imshow(zerocross);title("Zero-crossing Filter");
```

Output:

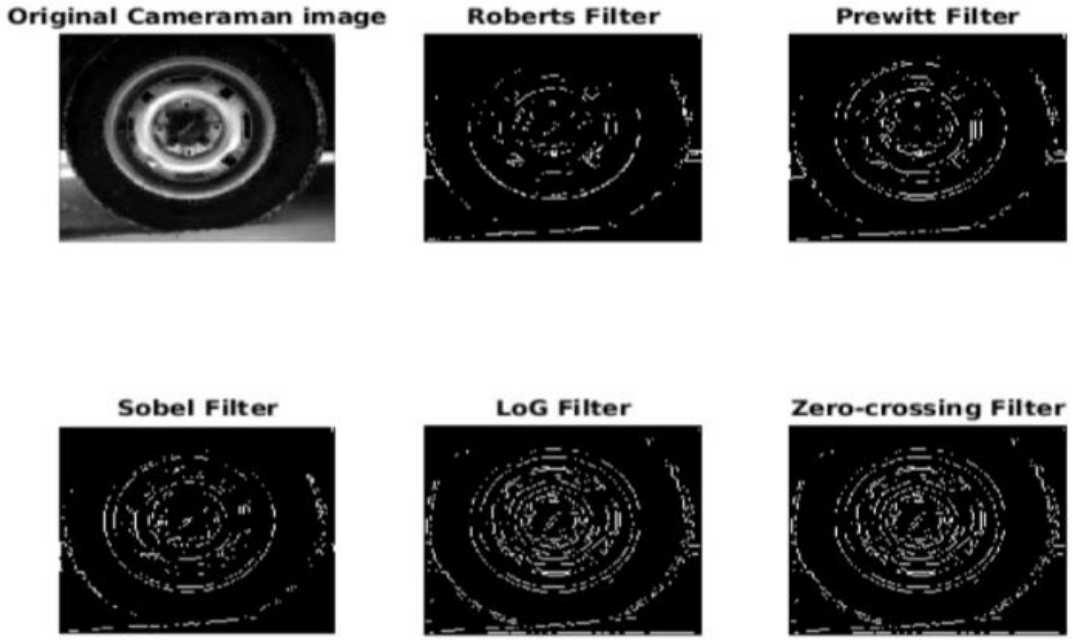


Observation:As per above output Robert, Sobel and Prewitts Filter techniques detected edges more accurately. whereas LoG and Zero-Crossing Filter techinques failed to detect edges accurately.

6B)Matlab Code:

```
clear all;
close all;
clc;
t=imread("tire.tif");
roberts = edge(t,'Roberts');
prewitt = edge(t,'Prewitt');
sobel = edge(t,'Sobel');
log = edge(t,'log');
zerocross=edge(t,'zerocross');
subplot(2,3,1);imshow(t);title("Original Cameraman image");
subplot(2,3,2);imshow(roberts);title("Roberts Filter");
subplot(2,3,3);imshow(prewitt);title("Prewitt Filter");
subplot(2,3,4);imshow(sobel);title("Sobel Filter");
subplot(2,3,5);imshow(log);title("LoG Filter");
subplot(2,3,6);imshow(zerocross);title("Zero-crossing Filter");
```

Output:



Observation:As per above output the Robert, Sobel, Prewitts, LoG and Zero-Crossing all Filter techniques detected the edges perfectly, But in case of LoG and Zero-crossing techniques the detected edges are thick when compared to edges detected by Robert, Sobel and Prewitts techniques.