

407 Comp Lab 6

Dr. Hewayda Lotfy

Q1- Add salt and pepper noise to an image, Apply the 3x3 median filter to remove noise and show the output

```
clear
clc
image = imread('pout.tif');
templImage = imread('pout.tif');
imageWithNoise = imnoise(templImage, 'salt & pepper');
[r,c] = size(imageWithNoise);
for i = 2 : r-1
    for j = 2 : c-1
        imageFilter = imageWithNoise( i-1 : i+1 , j-1 : j+1 ) ;
        resultImg(i, j) = median(median(imageFilter)) ;
    end
end

resultImg=uint8(resultImg);

subplot(2,2,1);imshow(image)
subplot(2,2,2);imshow(imageWithNoise)
subplot(2,2,3);imshow(resultImg)
```

```
clear
clc
image = imread('pout.tif');
imageWithNoise = imnoise(image, 'salt & pepper');
outPutImage = medfilt2(imageWithNoise);

subplot(2,2,1);imshow(image)
subplot(2,2,2);imshow(imageWithNoise)
subplot(2,2,3);imshow(outPutImage)
```

for making filter in colored image

```
i(:, :, 1) = imfilter2(i(:, :, 1))
i(:, :, 2) = imfilter2(i(:, :, 2))
i(:, :, 3) = imfilter2(i(:, :, 3))
```

Q2- Write a code for the derivative filters, Laplacian and sobel and apply to an image using NO matlab function for filtering. Then apply Laplacian for the enhancement of the image

```
clear
clc

image = imread('pout.tif');

[r, c] = size(image);
imageWithPadding = zeros(r+2, c+2);

imageWithPadding ( 2:r+1, 2:c+1 ) = image;
x = imageWithPadding ;
laplacianFilter = imageWithPadding;
for i = 2 : r+1
    for j = 2 : c+1
        laplacianFilter(i, j) = x(i+1, j) + x(i-1, j) + x(i, j+1) + x(i, j-1) + x(i+1, j-1) + x(i-1, j+1) + x(i-1, j-1) + x(i+1, j+1) - 8 * x(i, j);
    end
end

outPutImage = uint8(imageWithPadding-laplacianFilter);

subplot(2,2,1);imshow(image)
subplot(2,2,2);imshow(laplacianFilter)
subplot(2,2,3);imshow(outPutImage)
```

```
clear
clc
image=imread('pout.tif');
[r,c]=size(image);
imageWithPadding=zeros(r+2, c+2);
imageWithPadding(2:r+1,2:c+1)=image;
x=imageWithPadding;
sobelFilter=imageWithPadding;
for i=2:r+1
    for j=2:c+1
        sobelFilter(i,j)=abs((x(i,j-1)+2*x(i+1,j)+x(i+1,j+1))-(x(i-1,j-1)+2*x(i-1,j)+x(i-1,j+1)))+abs((x(i-1,j+1)+2*x(i,j+1)+x(i+1,j+1))-(x(i-1,j-1)+2*x(i,j+1)+x(i+1,j-1))));
    end
end
outPutImage=uint8(imageWithPadding(2:r+1,2:c+1)-sobelFilter(2:r+1,2:c+1));
subplot(2,2,1);imshow(image)
subplot(2,2,2);imshow(sobelFilter)
subplot(2,2,3);imshow(outPutImage)
```

Q3-segmentation

Segment an image using the Automatic basic global threshold method

1- Using the binary image resulted, count number of objects, get the pixel ratio of objects to background, the area and diameter of objects. Use region properties function.

2- Apply edge detection on original image using the MATLAB edge function (open the help documentation to see how to apply it)

```
clear
clc
image=imread('pout.tif');
image=im2bw(image);
[comp,num]=bwlabel(image,4);
```

```
Measurements=regionprops(comp,'all');
for k=1:num
    objArea(k)=Measurements(k).Area;
    objPerimeter(k)=Measurements(k).Perimeter;
end
imshow(comp);
```

```
clear
clc
a=imread('pout.tif');
edg=edge(a);
subplot(1,2,1);imshow(a);
subplot(1,2,2);imshow(edg)
```