Medical Image Processing Task

Omar Ali Amin Elsherif

Section 4

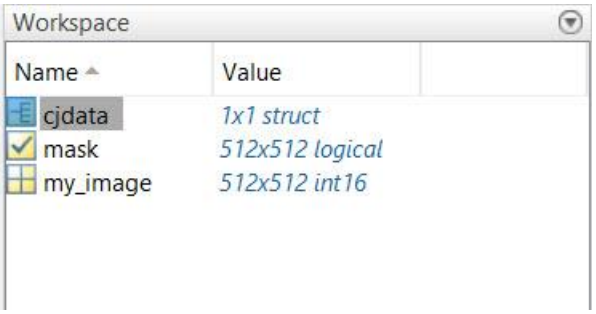
Image 103

ID : 1119103

Supervised by: Dr. Wafaa Shalash

1- Load the image from the file (‘103.m’) as a struct then read the image and mask inside and store them in variables

```
load('103.mat');
my_image = cjdata.image;
%border=cjdata.tumorBorder;
mask = cjdata.tumorMask;
```

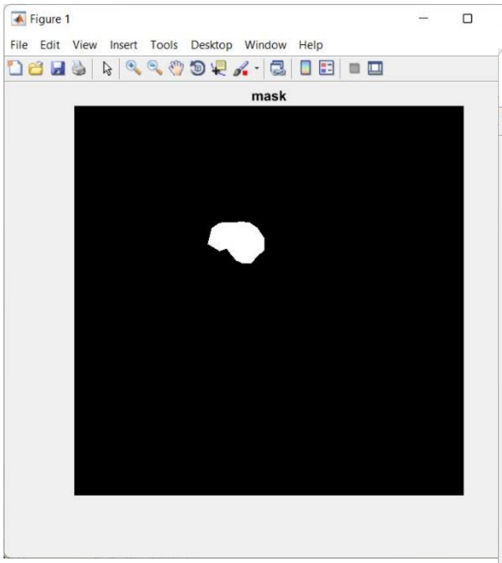


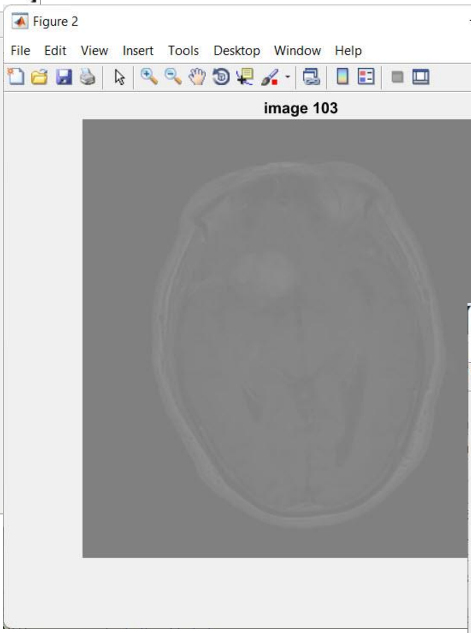
Workspace

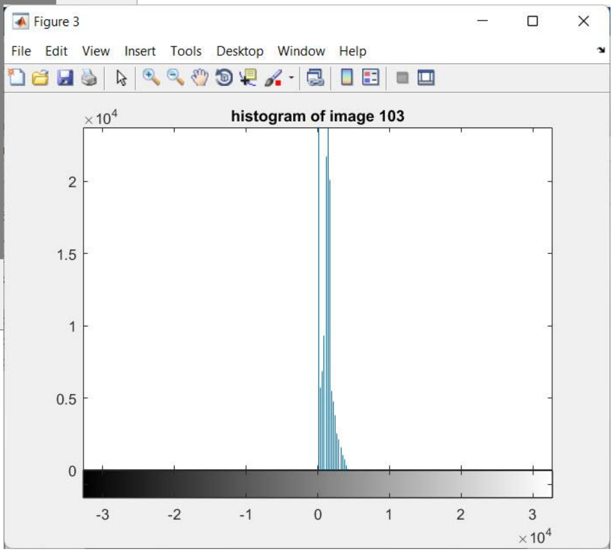
Name	Value
cjdata	1x1 struct
mask	512x512 logical
my_image	512x512 int16

2- Show image, mask and image histogram

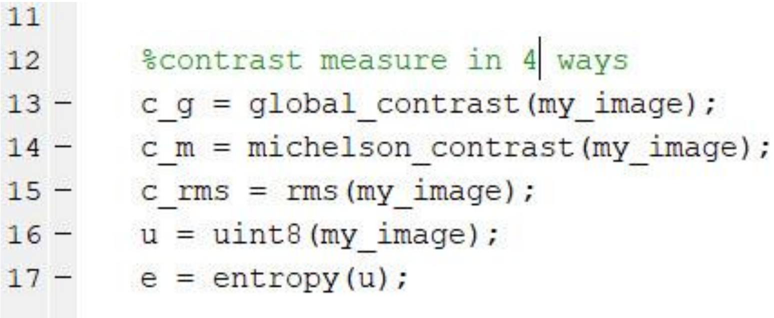
```
figure,imshow(mask);title('mask');
figure,imshow(my_image);title('image 103');
figure,imhist(my_image);title('histogram of image 103');
```



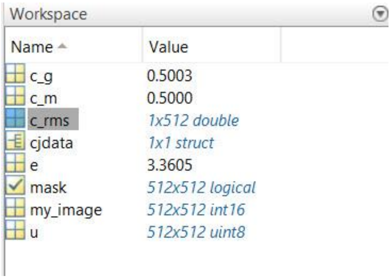




3- Measure contrast of the image with 4 ways

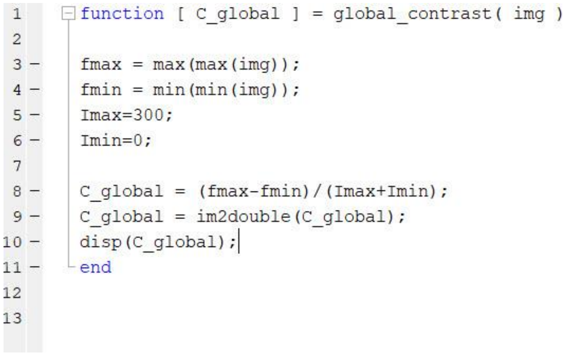


```
%contrast measure in 4 ways
c_g = global_contrast(my_image);
c_m = michelson_contrast(my_image);
c_rms = rms(my_image);
u = uint8(my_image);
e = entropy(u);
```

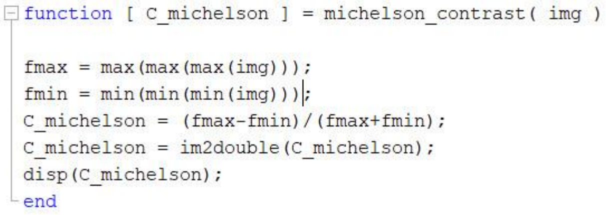


Workspace

Name	Value
c_g	0.5003
c_m	0.5000
c_rms	1x512 double
cjdata	1x1 struct
e	3.3605
mask	512x512 logical
my_image	512x512 int16
u	512x512 uint8

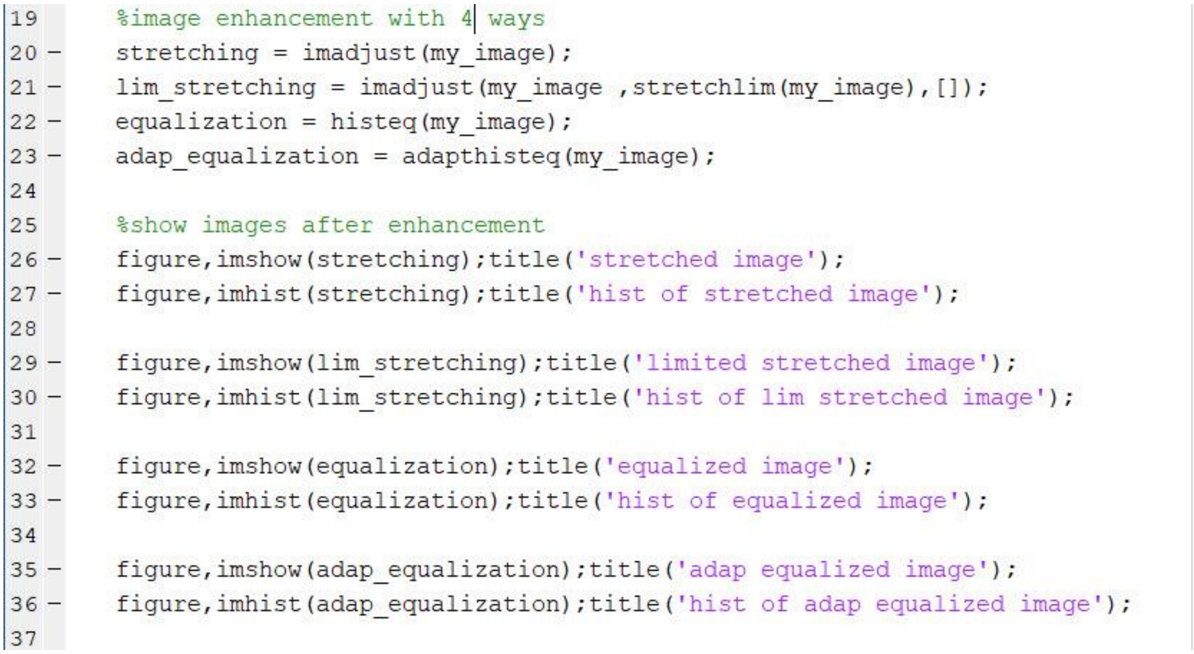


```
function [ C_global ] = global_contrast( img )

fmax = max(max(img));
fmin = min(min(img));
Imax=300;
Imin=0;

C_global = (fmax-fmin)/(Imax+Imin);
C_global = im2double(C_global);
disp(C_global);
end
```

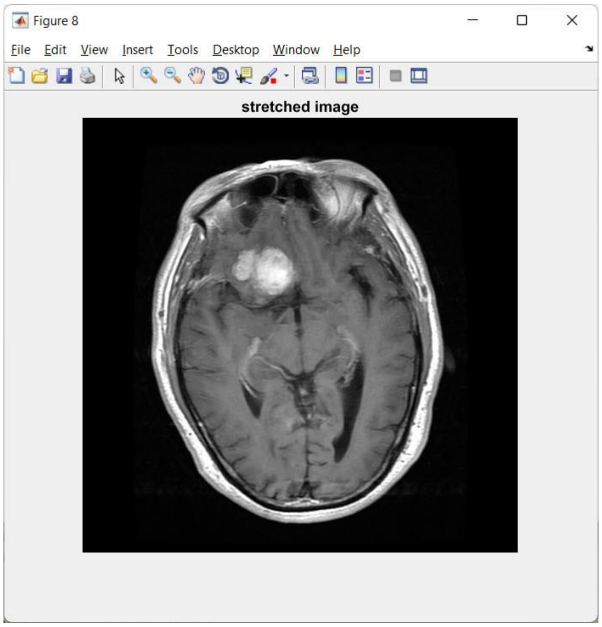


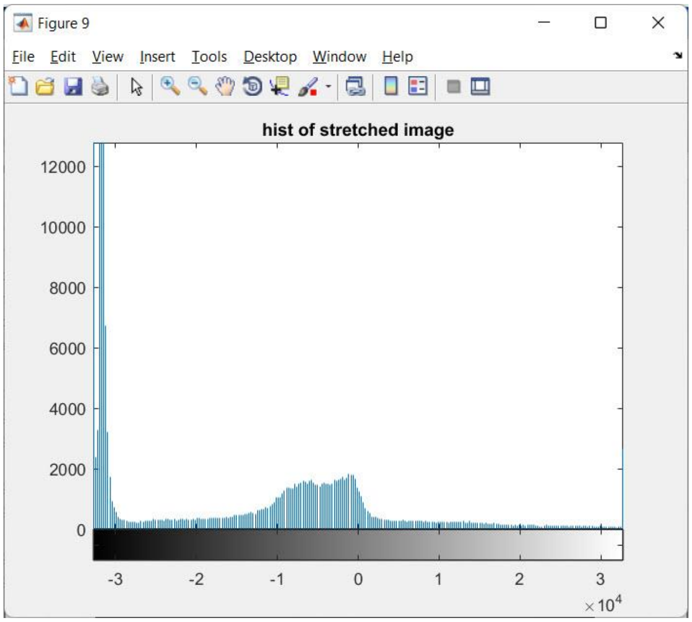
```
function [ C_michelson ] = michelson_contrast( img )

fmax = max(max(max(img)));
fmin = min(min(min(img)));
C_michelson = (fmax-fmin)/(fmax+fmin);
C_michelson = im2double(C_michelson);
disp(C_michelson);
end
```

4- Image enhancement code then show the results and their corresponding histograms



```
%image enhancement with 4 ways
stretching = imadjust(my_image);
lim_stretching = imadjust(my_image ,stretchlim(my_image),[]);
equalization = histeq(my_image);
adap_equalization = adapthisteq(my_image);

%show images after enhancement
figure,imshow(stretching);title('stretched image');
figure,imhist(stretching);title('hist of stretched image');

figure,imshow(lim_stretching);title('limited stretched image');
figure,imhist(lim_stretching);title('hist of lim stretched image');

figure,imshow(equalization);title('equalized image');
figure,imhist(equalization);title('hist of equalized image');

figure,imshow(adap_equalization);title('adap equalized image');
figure,imhist(adap_equalization);title('hist of adap equalized image');

```





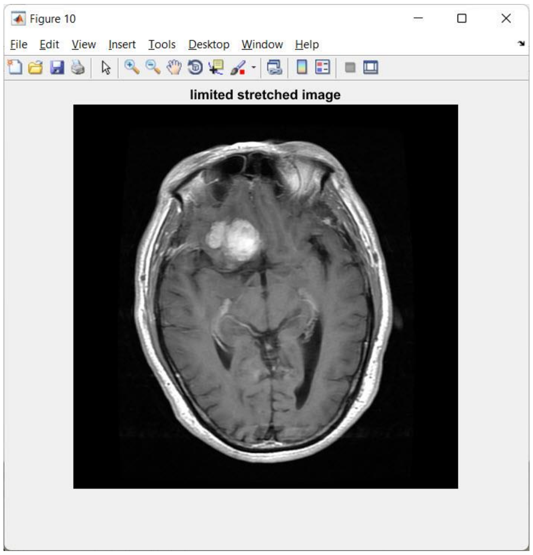


Figure 10
File Edit View Insert Tools Desktop Window Help
limited stretched image

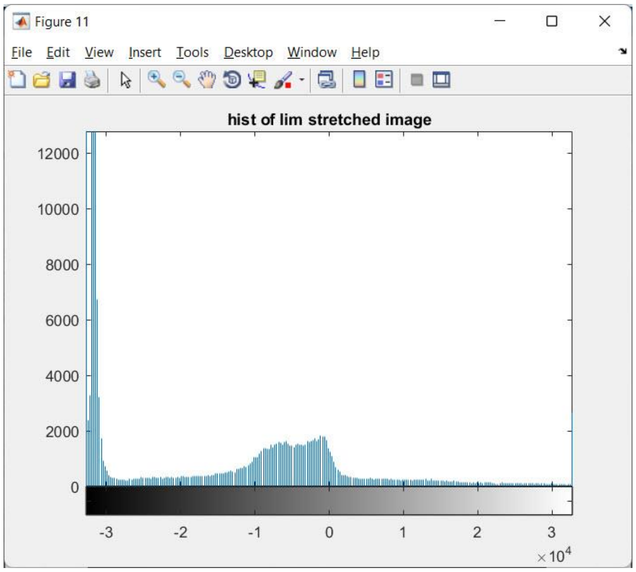


Figure 11
File Edit View Insert Tools Desktop Window Help
hist of lim stretched image
12000
10000
8000
6000
4000
2000
0
-3
-2
-1
0
1
2
3
$\times 10^4$

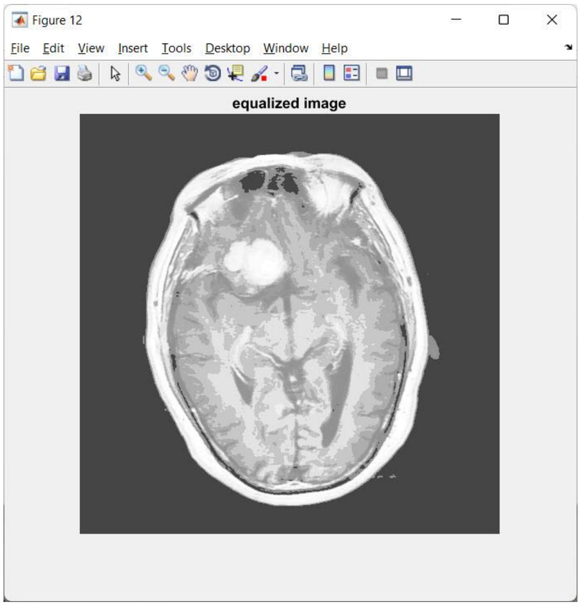


Figure 12
File Edit View Insert Tools Desktop Window Help
equalized image

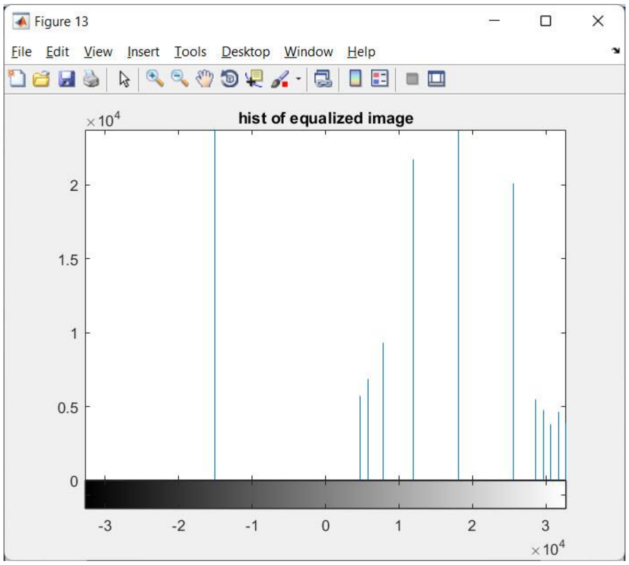


Figure 13
File Edit View Insert Tools Desktop Window Help
hist of equalized image
$\times 10^4$
2
1.5
1
0.5
0
-3
-2
-1
0
1
2
3
$\times 10^4$

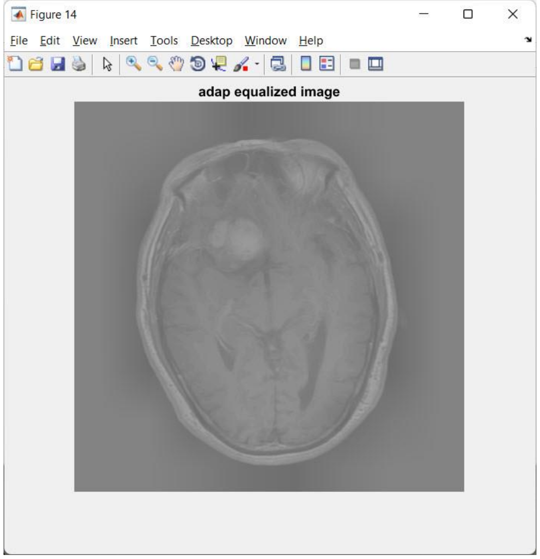


Figure 14
File Edit View Insert Tools Desktop Window Help
adap equalized image

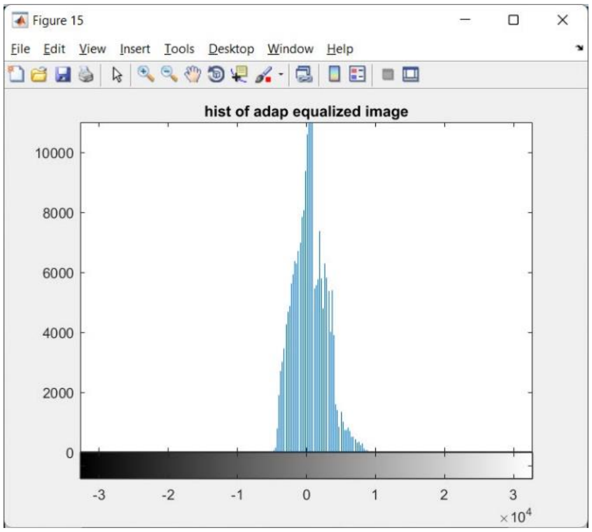
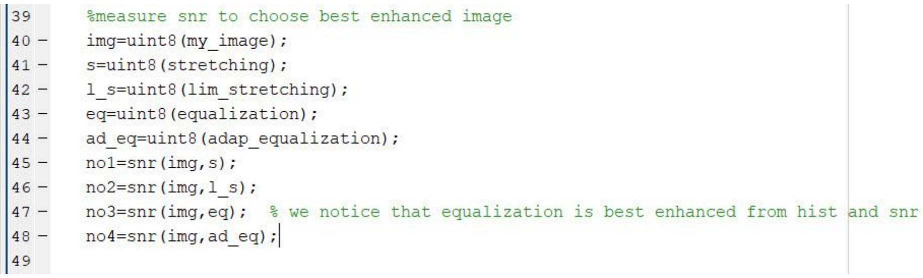
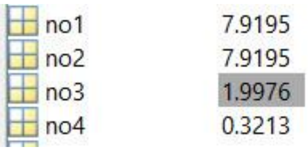


Figure 15
File Edit View Insert Tools Desktop Window Help
hist of adap equalized image
10000
8000
6000
4000
2000
0
-3
-2
-1
0
1
2
3
$\times 10^4$

5- We measure SNR for the enhanced images to determine which is more enhanced.

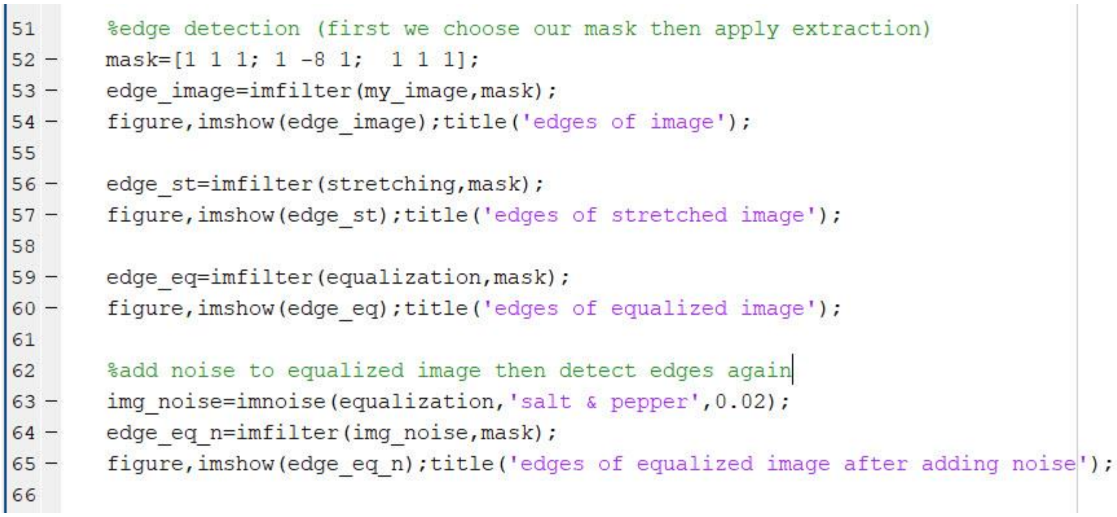


```
%measure snr to choose best enhanced image
img=uint8(my_image);
s=uint8(stretching);
l_s=uint8(lim_stretching);
eq=uint8(equalization);
ad_eq=uint8(adap_equalization);
no1=snr(img,s);
no2=snr(img,l_s);
no3=snr(img,eq);  % we notice that equalization is best enhanced from hist and snr
no4=snr(img,ad_eq);

```



no1	7.9195
no2	7.9195
no3	1.9976
no4	0.3213

6- Edge detection after we choose our mask we apply edge extraction to the enhanced image(equalized) then we add noise to the image and try extracting edges again.



```
%edge detection (first we choose our mask then apply extraction)
mask=[1 1 1; 1 -8 1;  1 1 1];
edge_image=imfilter(my_image,mask);
figure,imshow(edge_image);title('edges of image');

edge_st=imfilter(stretching,mask);
figure,imshow(edge_st);title('edges of stretched image');

edge_eq=imfilter(equalization,mask);
figure,imshow(edge_eq);title('edges of equalized image');

%add noise to equalized image then detect edges again
img_noise=imnoise(equalization,'salt & pepper',0.02);
edge_eq_n=imfilter(img_noise,mask);
figure,imshow(edge_eq_n);title('edges of equalized image after adding noise');

```

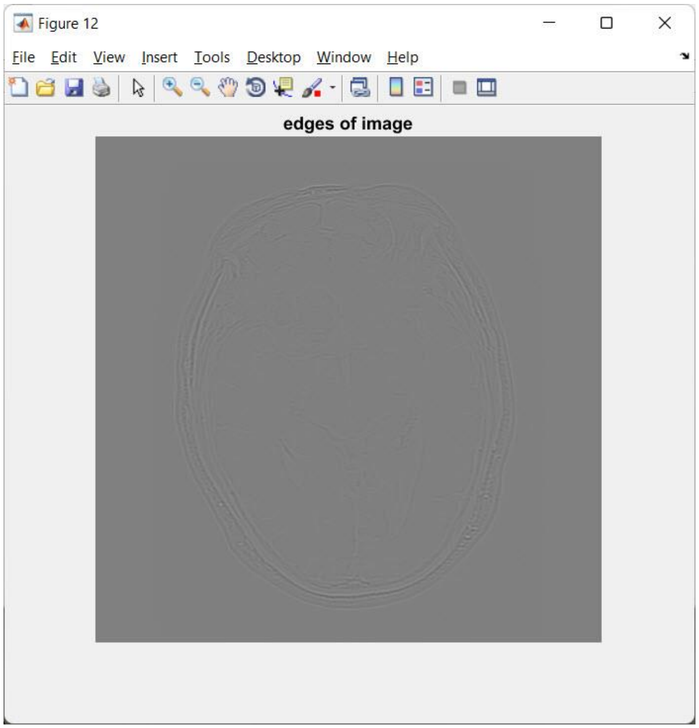


Figure 12
File Edit View Insert Tools Desktop Window Help
edges of image

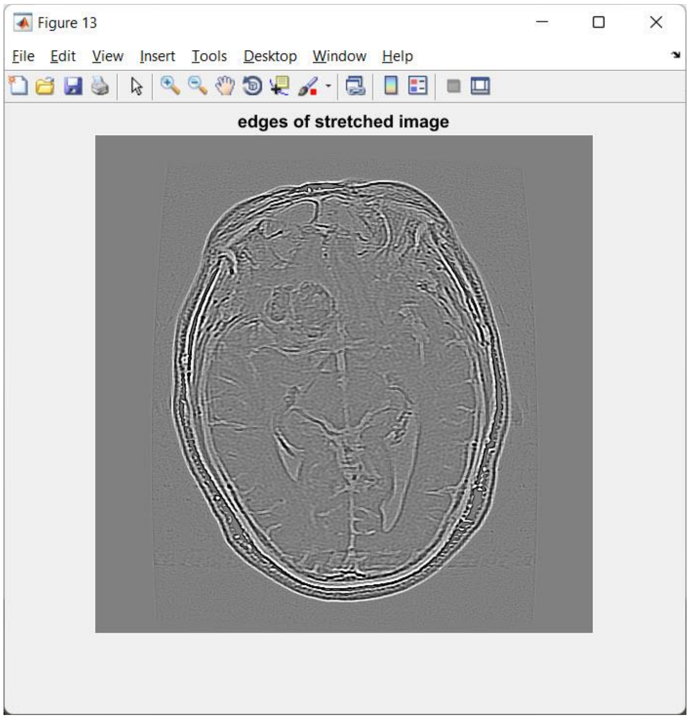


Figure 13
File Edit View Insert Tools Desktop Window Help
edges of stretched image

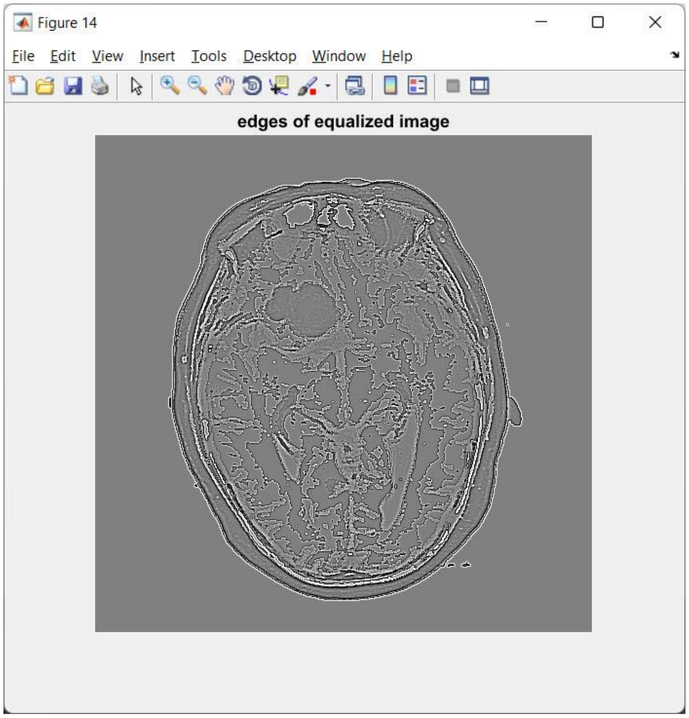


Figure 14
File Edit View Insert Tools Desktop Window Help
edges of equalized image

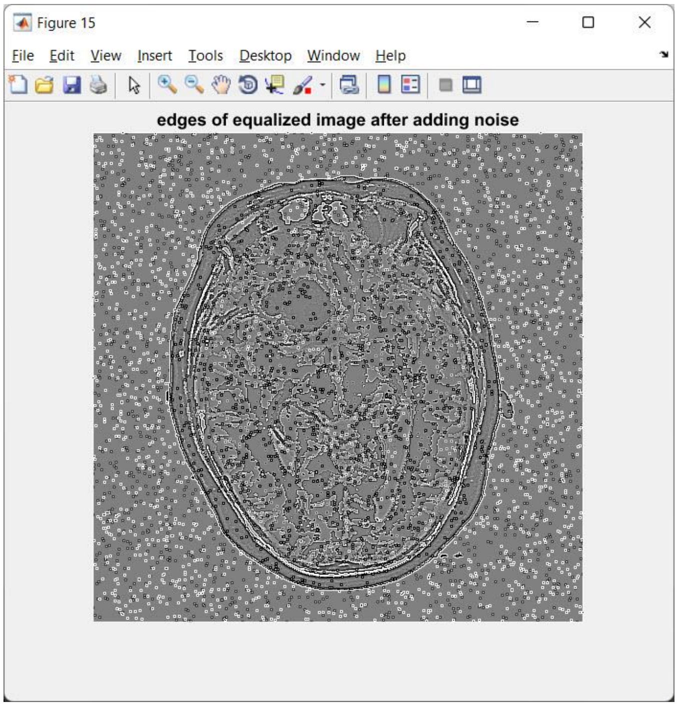
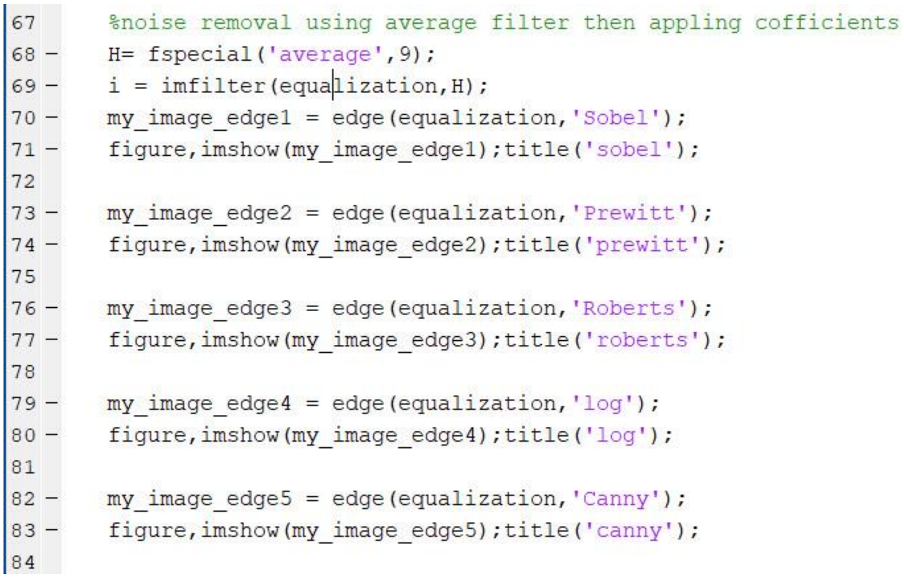
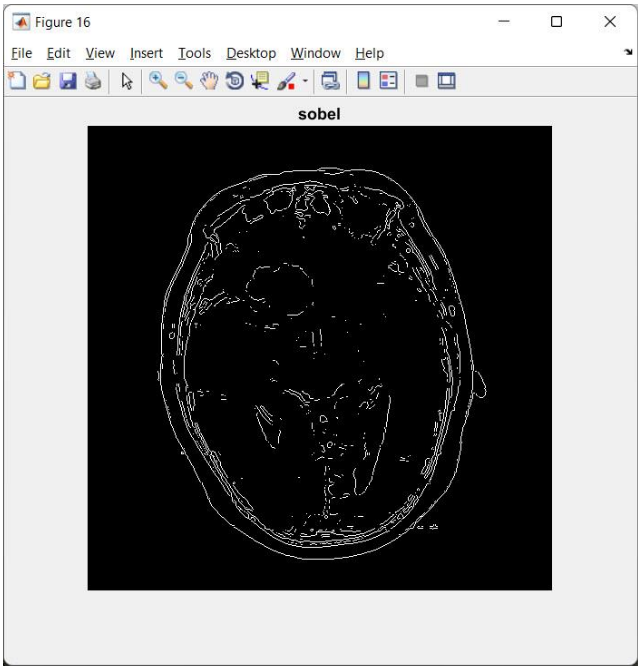


Figure 15
File Edit View Insert Tools Desktop Window Help
edges of equalized image after adding noise

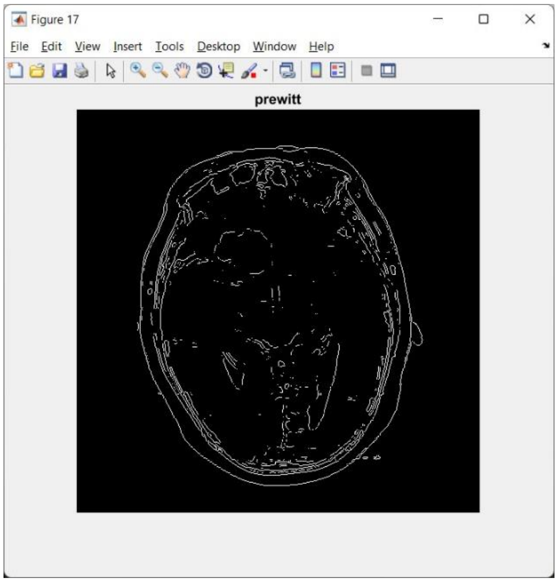
7- Removing noise using average then apply different coefficients to edges



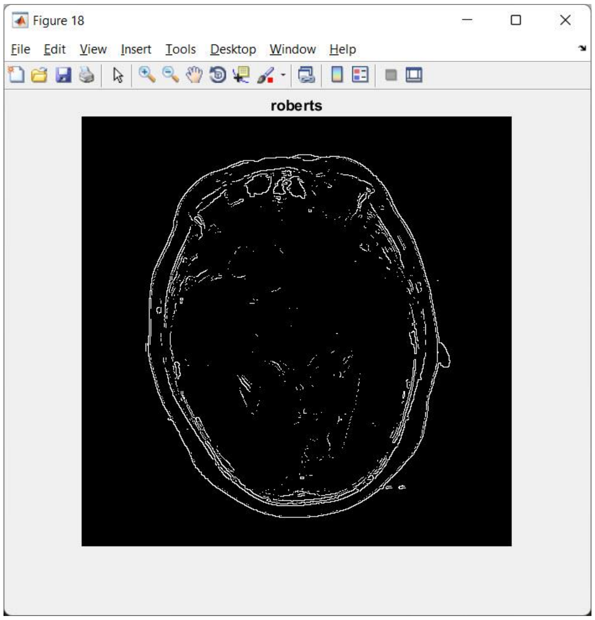
```
%noise removal using average filter then appling cofficients
H= fspecial('average',9);
i = imfilter(equalization,H);
my_image_edge1 = edge(equalization,'Sobel');
figure,imshow(my_image_edge1);title('sobel');

my_image_edge2 = edge(equalization,'Prewitt');
figure,imshow(my_image_edge2);title('prewitt');

my_image_edge3 = edge(equalization,'Roberts');
figure,imshow(my_image_edge3);title('roberts');

my_image_edge4 = edge(equalization,'log');
figure,imshow(my_image_edge4);title('log');

my_image_edge5 = edge(equalization,'Canny');
figure,imshow(my_image_edge5);title('canny');

```

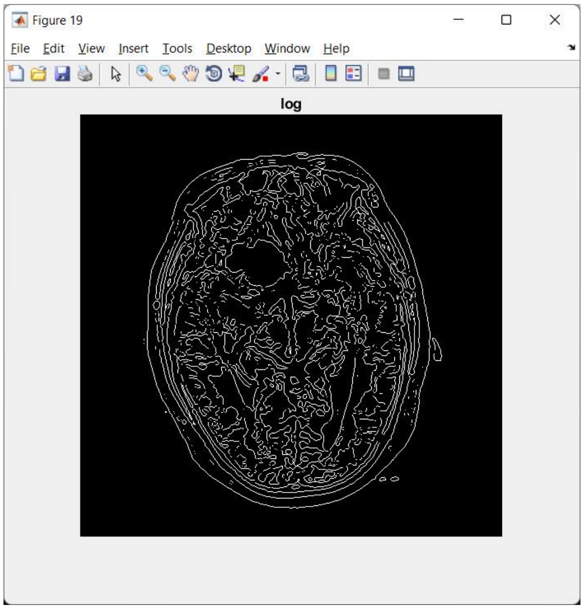


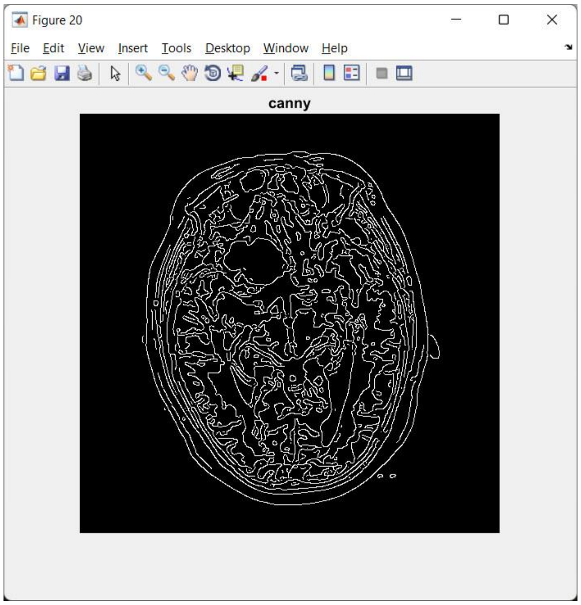
sobel



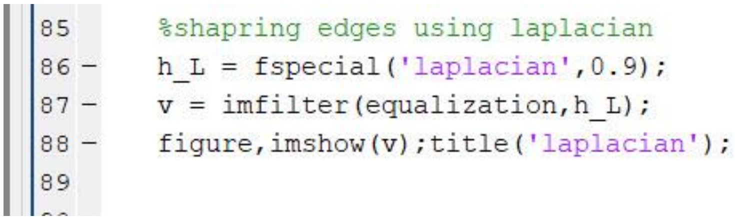
prewitt



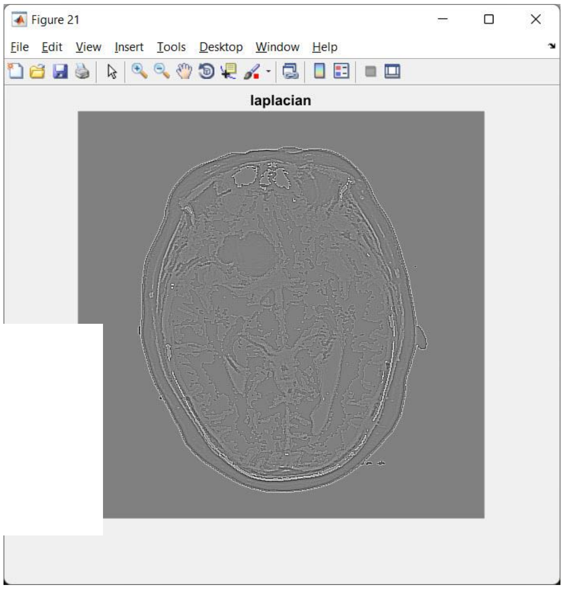




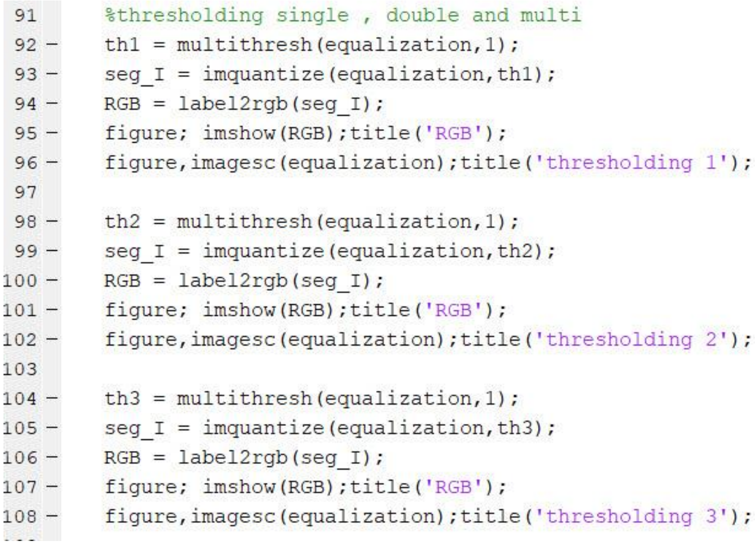
8- Sharping using Laplacian



```
%shapring edges using laplacian
h_L = fspecial('laplacian',0.9);
v = imfilter(equalization,h_L);
figure,imshow(v);title('laplacian');
```



9- Thresholding single , double and multi



```
%thresholding single , double and multi
th1 = multithresh(equalization,1);
seg_I = imquantize(equalization,th1);
RGB = label2rgb(seg_I);
figure; imshow(RGB);title('RGB');
figure,imagesc(equalization);title('thresholding 1');

th2 = multithresh(equalization,1);
seg_I = imquantize(equalization,th2);
RGB = label2rgb(seg_I);
figure; imshow(RGB);title('RGB');
figure,imagesc(equalization);title('thresholding 2');

th3 = multithresh(equalization,1);
seg_I = imquantize(equalization,th3);
RGB = label2rgb(seg_I);
figure; imshow(RGB);title('RGB');
figure,imagesc(equalization);title('thresholding 3');
```