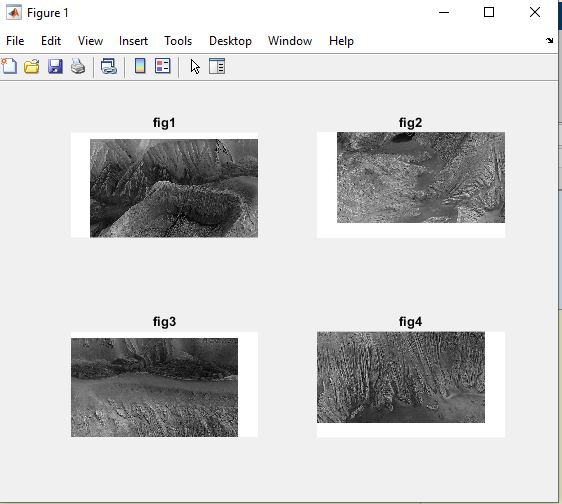
**TASK 2 ON FEATURE CALCULATION**

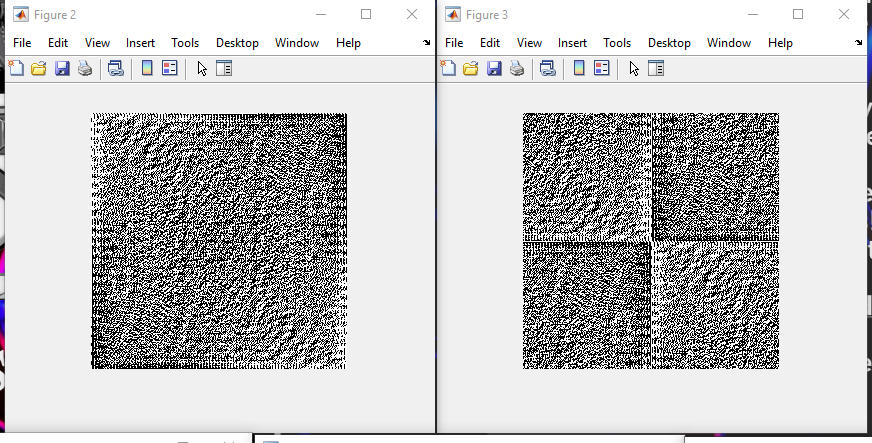
1. **TASK 1**

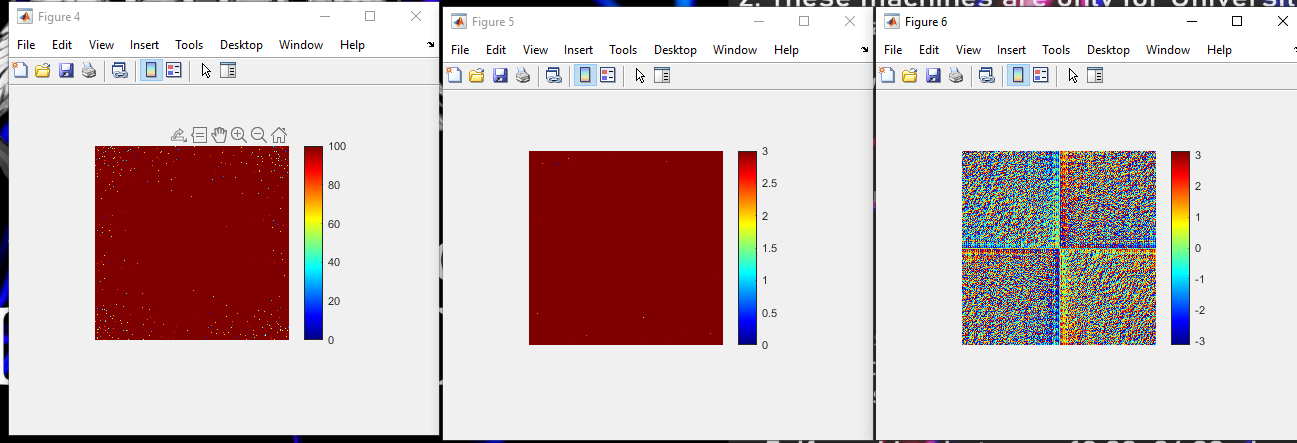


**READ THE IMAGE (ImgPIA.jpg)**



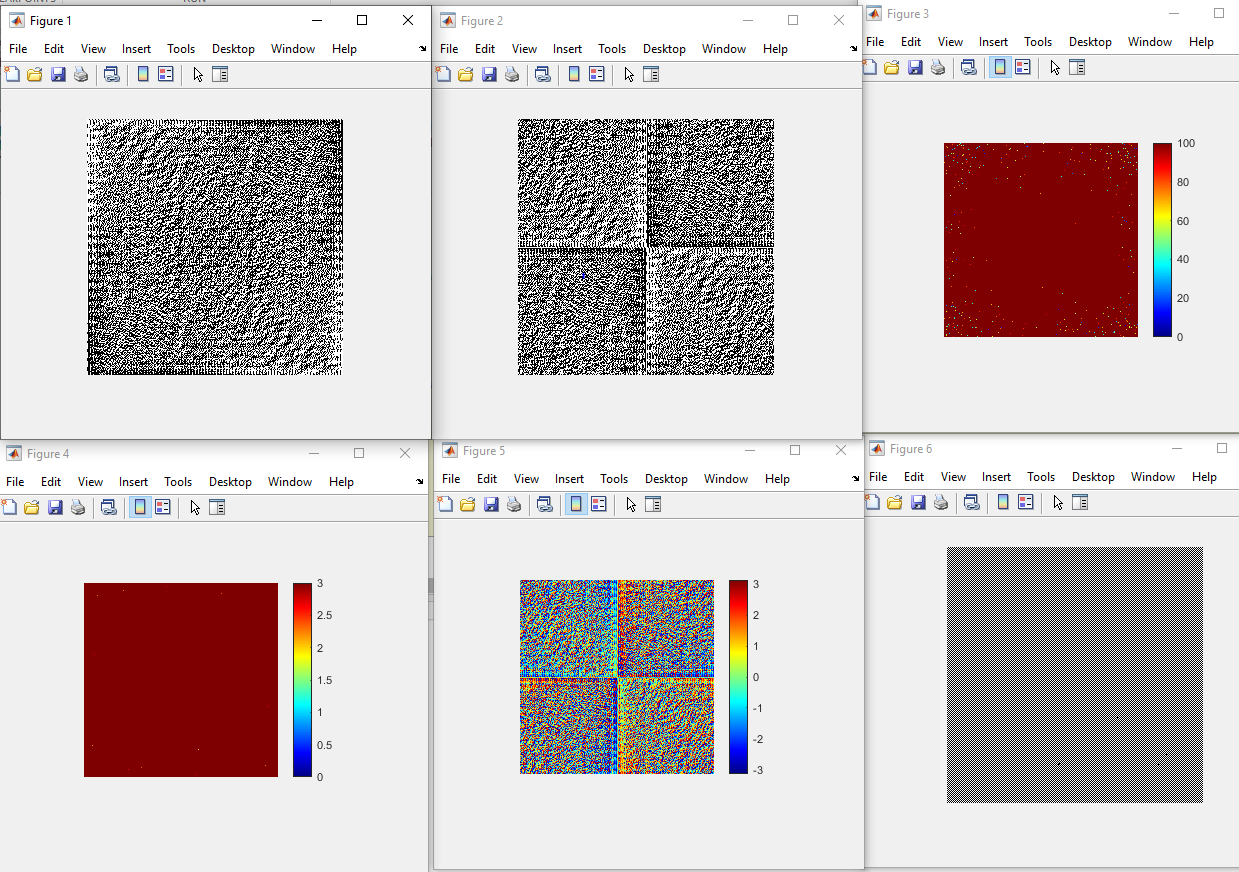
**SEGMENT INTO SECTION**



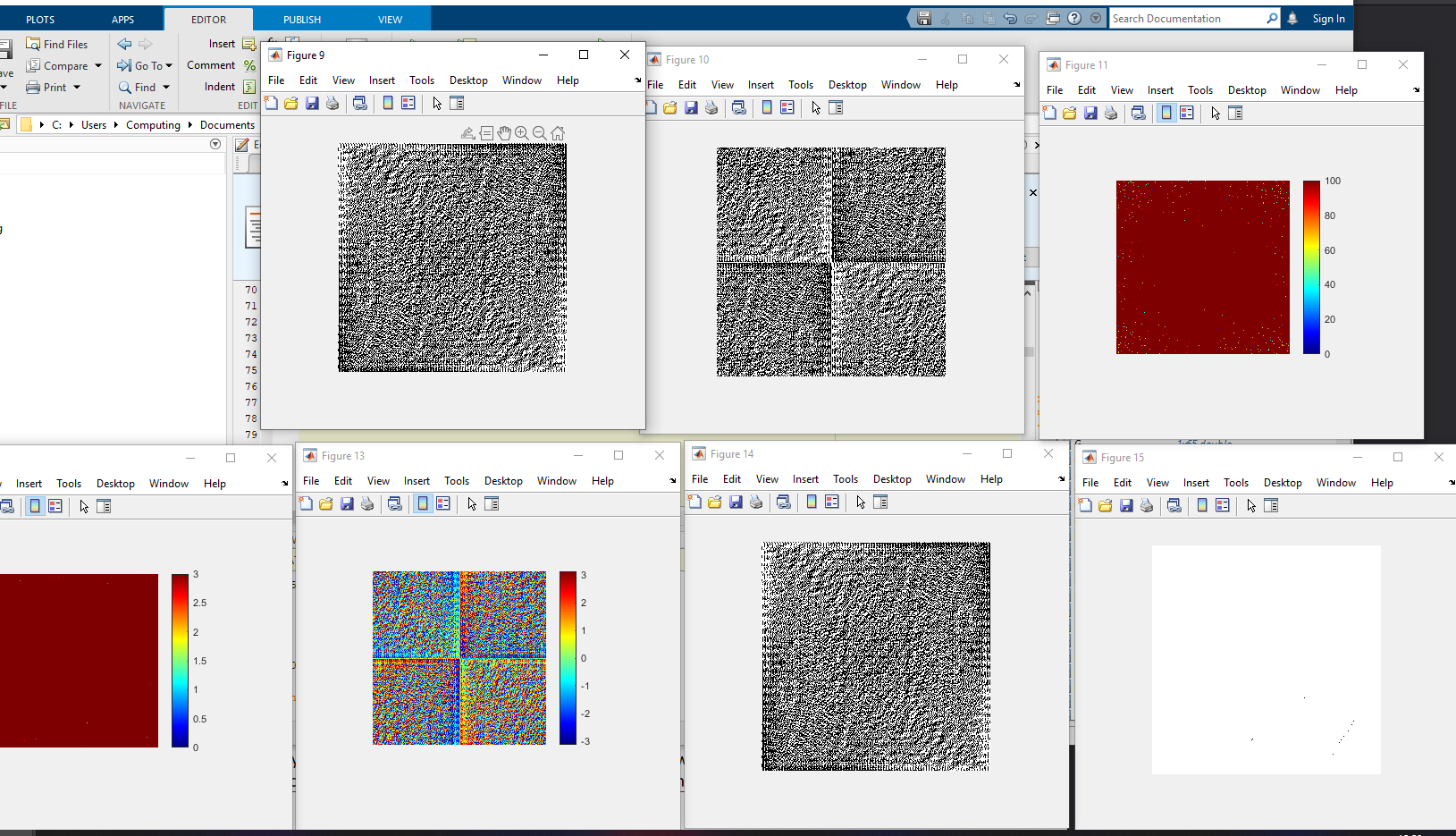


**WHEN FOURIER TRANSFORM WAS FIRST APPLIED** Ans **= 5.158801618675188e+00, Ans = 10136717**

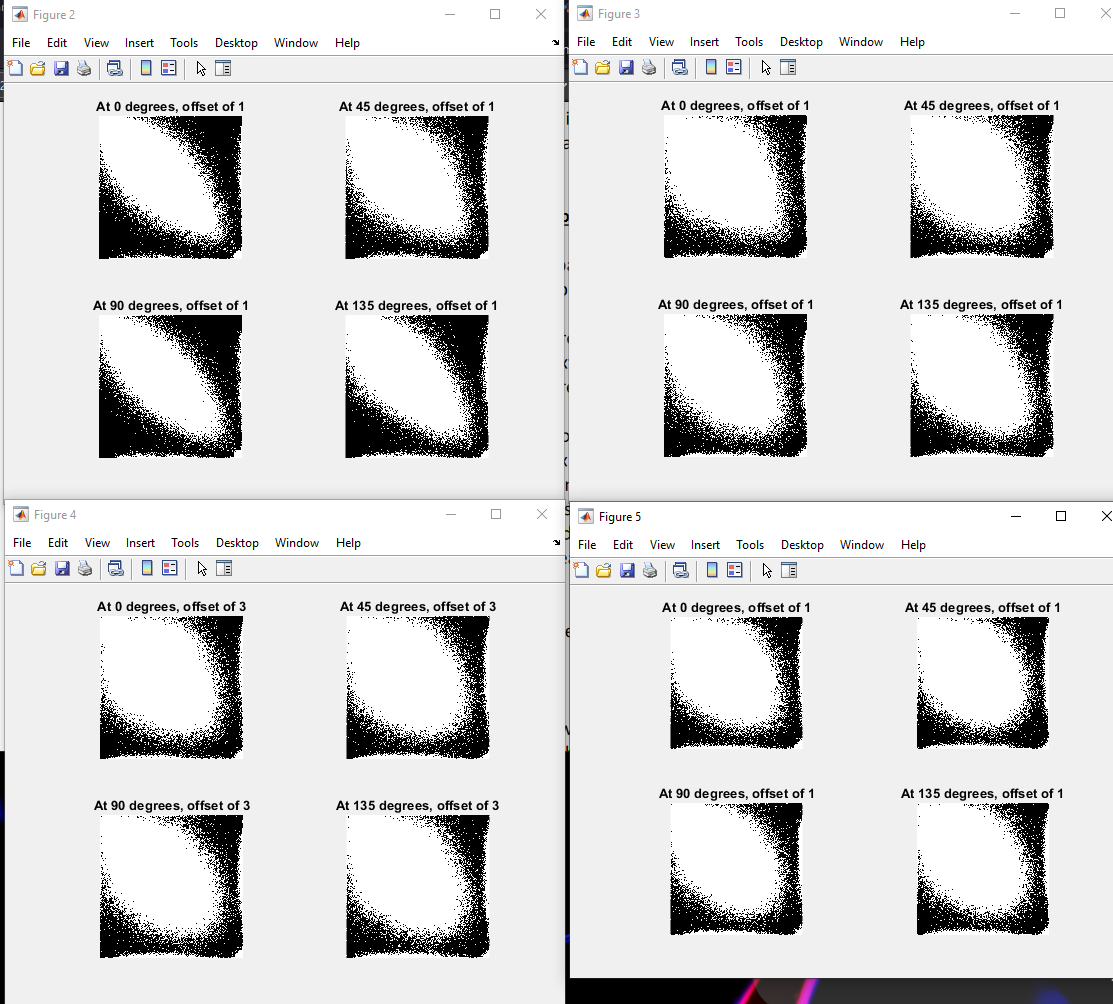
**ans = 5.158801618675188e+00, ans = 10136717**



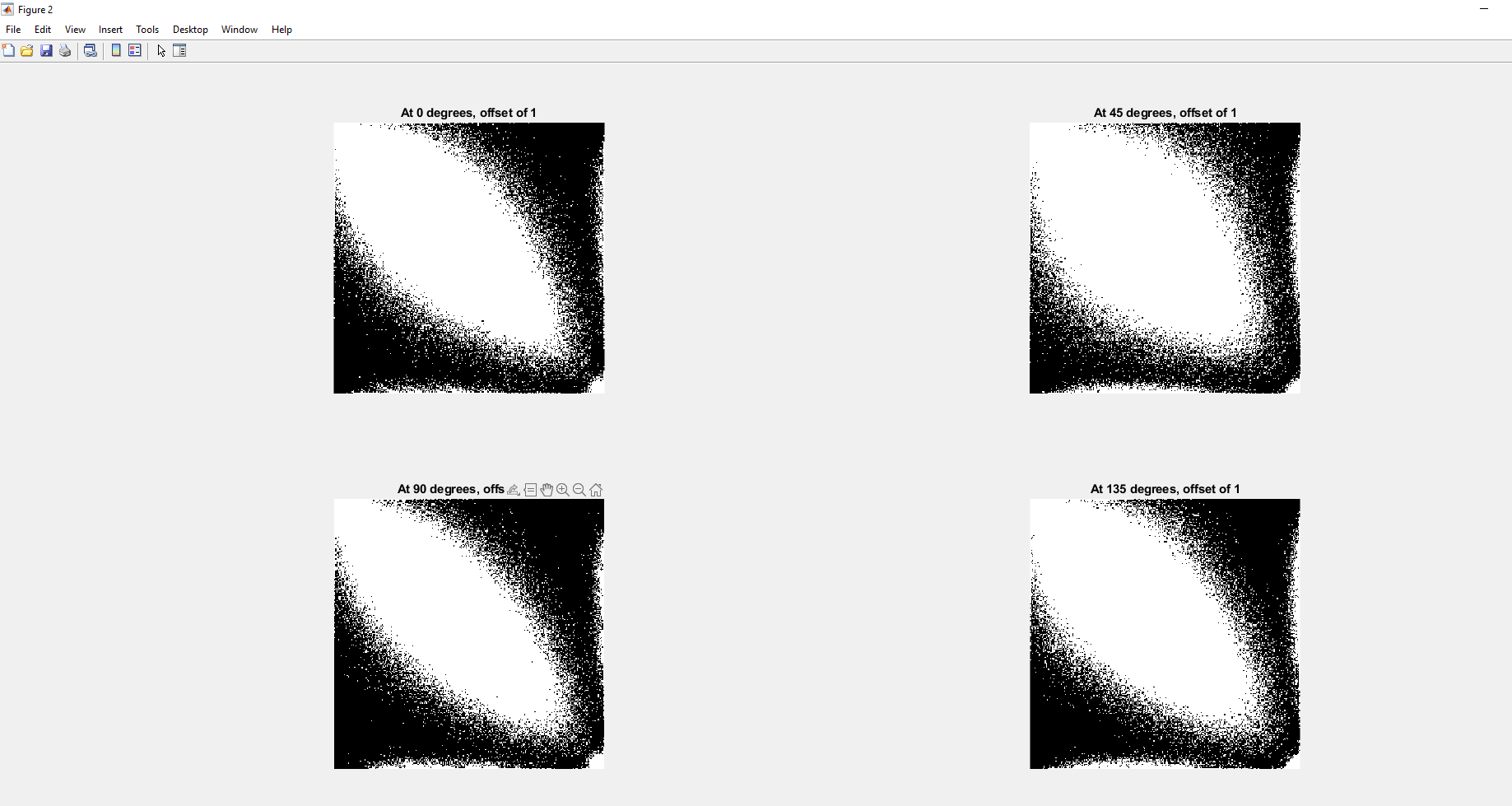
**Ans = 5.158801618675188e+00, Ans =** 10136717



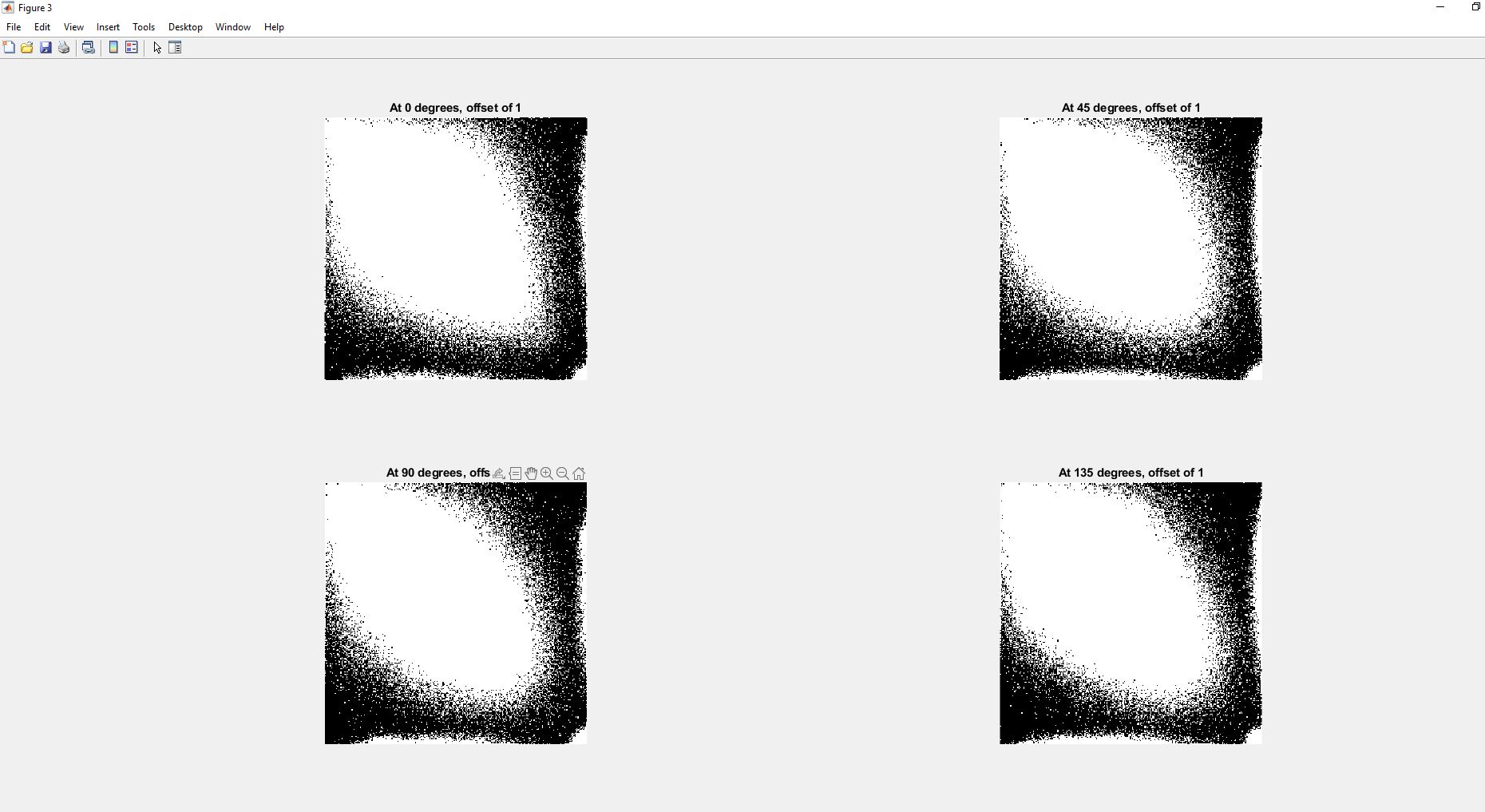
**Ans = 5.158801618675188e+00, Ans = 10136717**



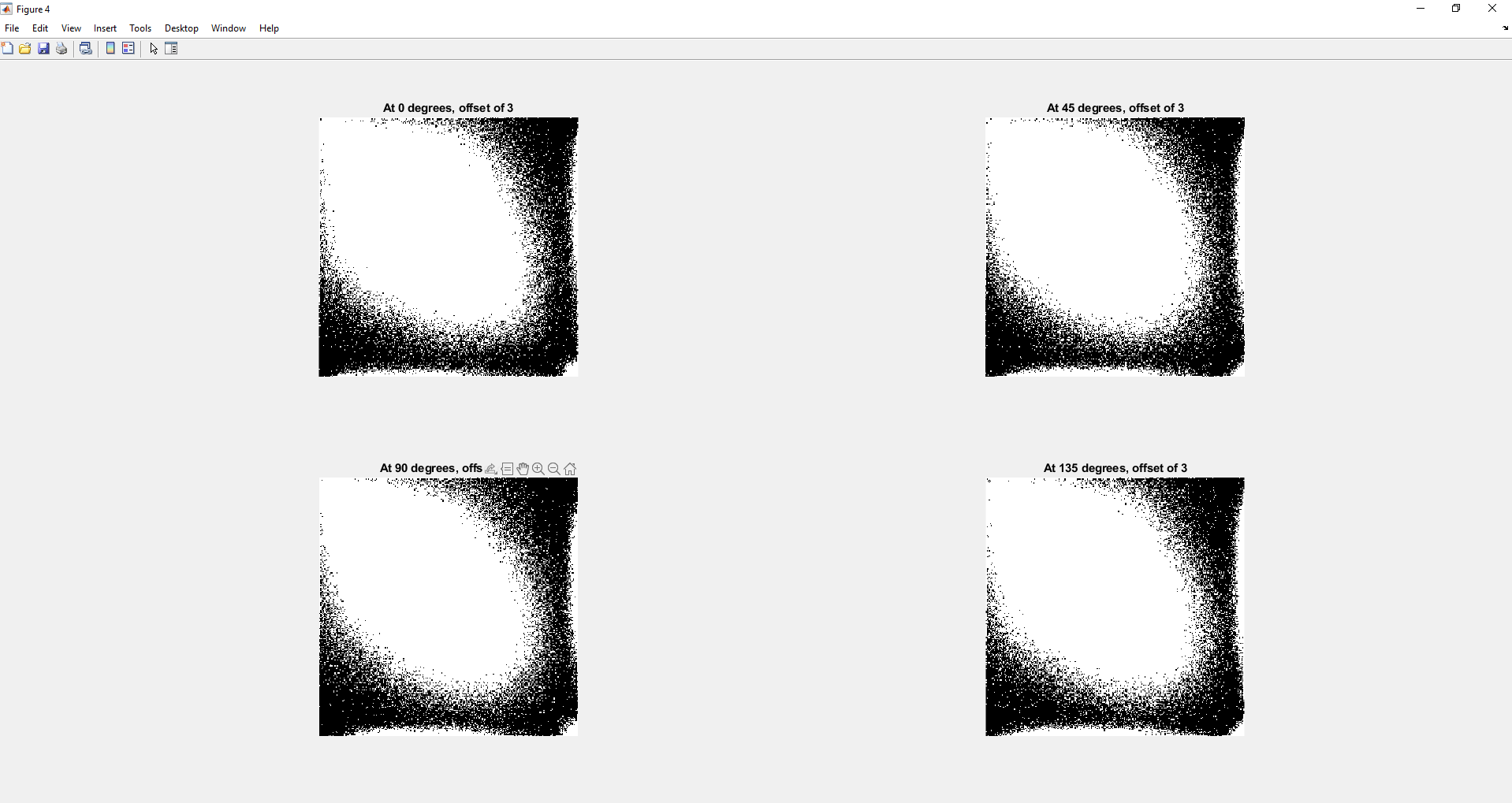
**SELECT THE FEATURES FOR BOTH RADIUS AND DIRECTION AS DESCRIBE IN THE SPECTRAL APPROACH SESSION: FEATURES FOR ALL THE RADIUS AND DIRECTION**



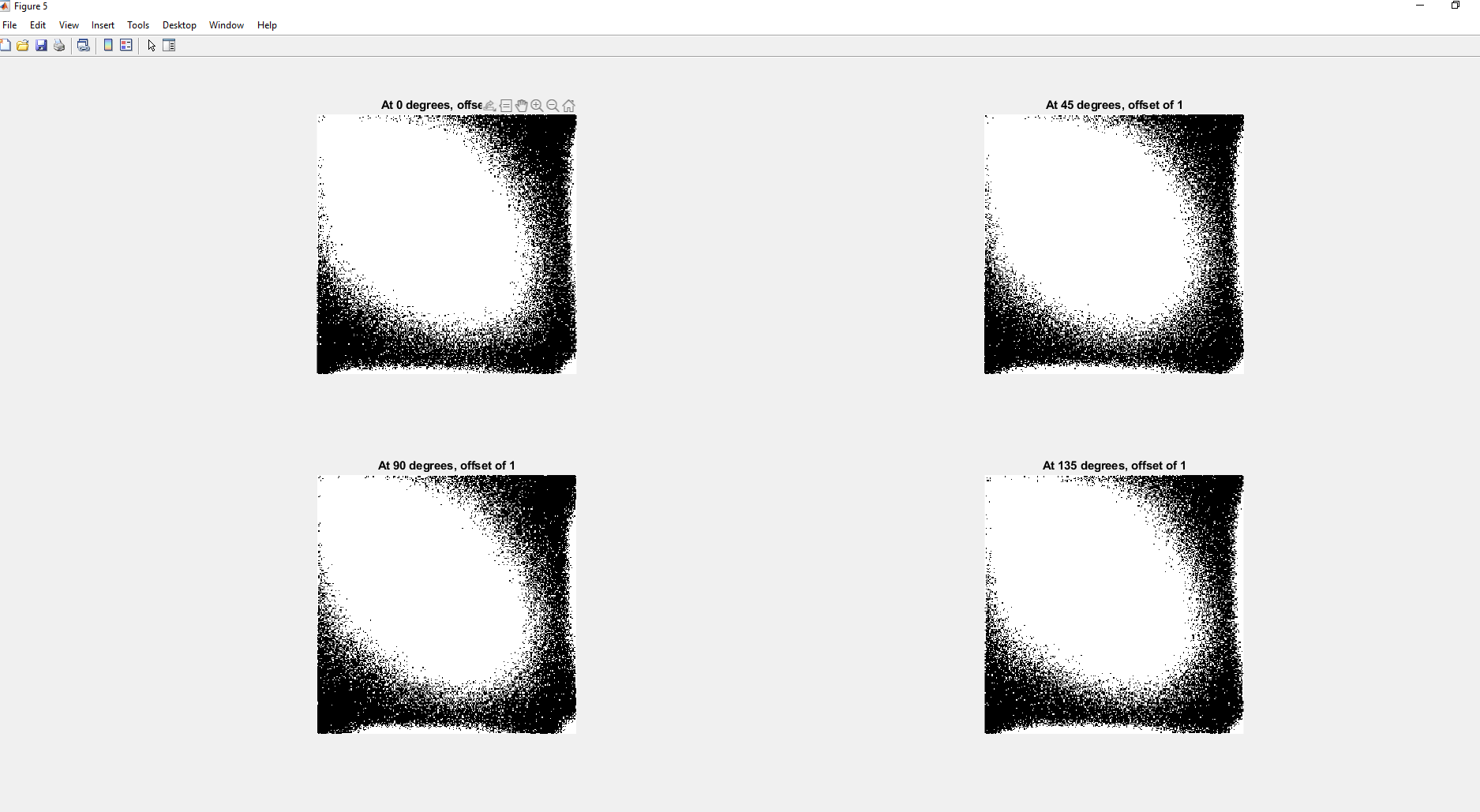
**PARAMETER D AT 1 AT DIFFERENT ANGLE AND OFFSET**



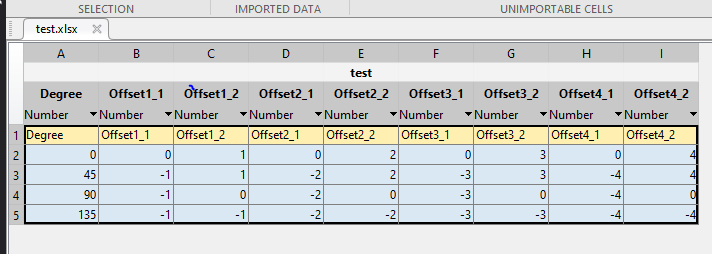
**PARAMETER D AT 2 AT DIFFERENT ANGLE AND OFFSET**



**PARAMETER D AT 3 AT DIFFERENT ANGLE AND OFFSET**



**PARAMETER D AT 4 AT DIFFERENT ANGLE AND OFFSET**

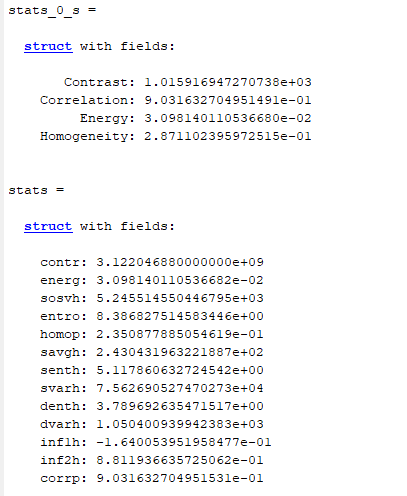


**TABLE FOR FEATURES FOR BOTH THE RADIUS AND DIRECTION**

1. **TASK 2**



**READ THE IMAGE (ImgPIA.jpg)**



**THE CO-OCCURRENCE MATRIX FEATURES**

**THE GRAY LEVEL RUN LENGTH MATRIX FEATURES WITHOUT A BIT QUANTIZATION**

**1. SHORT RUN EMPHASIS (SRE) = 4.5710**

**2. LONG RUN EMPHASIS (LRE) = 14.1076**

**3. GRAY LEVEL NON-UNIFORMITY (GLN) = 122576.2206**

**4. RUN PERCENTAGE (RP) = 57.9390**

**5. RUN LENGTH NON-UNIFORMITY (RLN) = 3640564.9094**

**6. LOW GRAY LEVEL RUN EMPHASIS (LGRE) = 0.0474**

**7. HIGH GRAY LEVEL RUN EMPHASIS (HGRE) = 3142.6580**

**2.4122**

**6.4137**

**8245.4360**

**4.8917**

**156464.0846**

**0.0268**

**211.9722**

**IMAGE QUANTIZATION TO 4 BITS (16 GRAY LEVELS)**

**1. SHORT RUN EMPHASIS (SRE) = 10.0322**

**2. LONG RUN EMPHASIS (LRE) = 16.9052**

**3. GRAY LEVEL NON-UNIFORMITY (GLN) = 13475.2528**

**4. RUN PERCENTAGE (RP) = 20.1838**

**5. RUN LENGTH NON-UNIFORMITY (RLN) = 2906409.4830**

**6. LOW GRAY LEVEL RUN EMPHASIS (LGRE) = 0.0295**

**7. HIGH GRAY LEVEL RUN EMPHASIS (HGRE) = 363.1319**

**2.6319**

**5.7316**

**2498.0348**

**3.1489**

**113921.6059**

**0.0190**

**182.5715**

**IMAGE QUANTIZATION TO 6 BITS (32 GRAY LEVELS)**

**1. SHORT RUN EMPHASIS (SRE) = 4.5978**

**2. LONG RUN EMPHASIS (LRE) = 6.9123**

**3. GRAY LEVEL NON-UNIFORMITY (GLN) = 12008.9827**

**4. RUN PERCENTAGE (RP) = 11.3075**

**5. RUN LENGTH NON-UNIFORMITY (RLN) = 761262.7619**

**6. LOW GRAY LEVEL RUN EMPHASIS (LGRE) = 0.0070**

**7. HIGH GRAY LEVEL RUN EMPHASIS (HGRE) = 1571.5238**

**1.6045**

**2.7103**

**1474.5064**

**2.0905**

**47934.1477**

**0.0057**

**648.3664**

**IMAGE QUANTIZATION TO 8 BITS (64 GRAY LEVELS)**

**1. SHORT RUN EMPHASIS (SRE) = 2.3257**

**2. LONG RUN EMPHASIS (LRE) = 3.4396**

**3. GRAY LEVEL NON-UNIFORMITY (GLN) = 14721.1027**

**4. RUN PERCENTAGE (RP) = 9.0240**

**5. RUN LENGTH NON-UNIFORMITY (RLN) = 308115.9300**

**6. LOW GRAY LEVEL RUN EMPHASIS (LGRE) = 0.0032**

**7. HIGH GRAY LEVEL RUN EMPHASIS (HGRE) = 3080.4826**

**0.9303**

**1.3759**

**588.8442**

**0.9024**

**12324.6372**

**0.0013**

**1232.3645**

%% Task 2: Feature Calculation

%% TASK 1 ON FEATURE CALCULATION

f = imread('ImgPIA.jpg');

figure;

imshow(f);

f = rgb2gray(f);

title('OriginalImage')

ImageFolder ='C:\Users\Computing\Documents\MATLAB';

% % this loop will take 5 pictures and save them in the Matlab folder

%

% file\_name = sprintf('Image%d.png', 0)% name Image with a sequence of number, ex Image1.png , Image2.png....

% fullFileName = fullfile(ImageFolder, file\_name);

% imwrite(f,file\_name,'png') %save the image as a Portable Graphics Format file(png)into the MatLab

% pause(1); % pause for one second

% imshow(f) % display the image for every second

% imgName = [ImageFolder,'\Image\_',num2str(0),'.png'];

%%

F = fft2(f,256,256);

figure;

imshow(F)

%%

I = ifft2(F,256,256);

figure;

imshow(I);

%%

% width = 100; height = 300;

% [rows, columns] = size(f);

% left = columns/2 - width/2;

% top = rows/2 - height/2;

% f1 = imcrop(f, [left, top, width, height]);

% subplot(1,2,1);imshow(f1);

%

% width = 600; height = 300;

% [rows, columns] = size(f);

% left = columns/2 - width/2;

% top = rows/2 - height/2;

% f2 = imcrop(f, [left, top, width, height]);

% subplot(1,2,2);imshow(f2);

% SEGMENT INTO SECTION

I = f;

f1=I(1:size(I,1)/2,1:size(I,2)/2,:);subplot(2,2,1);imshow(f1);

subplot(2,2,1);

title('fig1')

f2=I(size(I,1)/2+1:size(I,1),1:size(I,2)/2,:);subplot(2,2,2);imshow(f2);

subplot(2,2,2);

title('fig2')

f3=I(1:size(I,1)/2,size(I,2)/2+1:size(I,2),:);subplot(2,2,3);imshow(f3);

subplot(2,2,3);

title('fig3')

f4=I(size(I,1)/2+1:size(I,1),size(I,2)/2+1:size(I,2),:);subplot(2,2,4);imshow(f4);

subplot(2,2,4);

title('fig4')

%%

F = fft2(f,256,256);

figure;

imshow(F);

F = fftshift(F); % Center FFT

figure;

imshow(F);

% Measure the minimum and maximum value of the transform amplitude

min(min(abs(F)))

max(max(abs(F)))

figure;

imshow(abs(F),[0 100]); colormap(jet); colorbar

figure;

imshow(log(1+abs(F)),[0,3]); colormap(jet); colorbar

% Look at the phases

figure;

imshow(angle(F),[-pi,pi]); colormap(jet); colorbar

I = ifft2(F,256,256);

figure;

imshow(I);

%%

F = fft2(f,256,256);

figure;

imshow(F);

F = fftshift(F); % Center FFT

figure;

imshow(F);

% Measure the minimum and maximum value of the transform amplitude

min(min(abs(F)))

max(max(abs(F)))

figure;

imshow(abs(F),[0 100]); colormap(jet); colorbar

figure;

imshow(log(1+abs(F)),[0,3]); colormap(jet); colorbar

% Look at the phases

figure;

imshow(angle(F),[-pi,pi]); colormap(jet); colorbar

F = fft2(f,256,256);

figure;

imshow(F);

I = ifft2(F,256,256);

figure;

imshow(I);

%%

% TASK 1: BY SELECTING IT'S FEATURES

figure;

imshow(f);

figure;

offsets = [0 1];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,1);imshow(glcm\_0\_s);

title('At 0 degrees, offset of 1')

offsets = [-1 1];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,2);imshow(glcm\_0\_s);

title('At 45 degrees, offset of 1')

offsets = [-1 0];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,3);imshow(glcm\_0\_s);

title('At 90 degrees, offset of 1')

offsets = [-1 -1];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,4);imshow(glcm\_0\_s);

title('At 135 degrees, offset of 1')

figure;

offsets = [0 2];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,1);imshow(glcm\_0\_s);

title('At 0 degrees, offset of 1')

offsets = [-2 2];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,2);imshow(glcm\_0\_s);

title('At 45 degrees, offset of 1')

offsets = [-2 0];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,3);imshow(glcm\_0\_s);

title('At 90 degrees, offset of 1')

offsets = [-2 -2];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,4);imshow(glcm\_0\_s);

title('At 135 degrees, offset of 1')

figure;

offsets = [0 3];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,1);imshow(glcm\_0\_s);

title('At 0 degrees, offset of 3')

offsets = [-3 3];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,2);imshow(glcm\_0\_s);

title('At 45 degrees, offset of 3')

offsets = [-3 0];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,3);imshow(glcm\_0\_s);

title('At 90 degrees, offset of 3')

offsets = [-3 -3];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,4);imshow(glcm\_0\_s);

title('At 135 degrees, offset of 3')

figure;

offsets = [0 4];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,1);imshow(glcm\_0\_s);

title('At 0 degrees, offset of 1')

offsets = [-4 4];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,2);imshow(glcm\_0\_s);

title('At 45 degrees, offset of 1')

offsets = [-4 0];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,3);imshow(glcm\_0\_s);

title('At 90 degrees, offset of 1')

offsets = [-4 -4];

glcm\_0\_s = graycomatrix(f, 'offset', offsets,'NumLevels', 256, 'Symmetric',true);

subplot(2,2,4);imshow(glcm\_0\_s);

title('At 135 degrees, offset of 1');

% Header = ["Degree";"Offset1";"Offset2";"Offset3";"Offset4"];

Degree = [0;45;90;135];

Offset1 = [0 1; -1 1; -1 0; -1 -1];

Offset2 = [0 2; -2 2; -2 0; -2 -2];

Offset3 = [0 3; -3 3; -3 0; -3 -3];

Offset4 = [0 4; -4 4; -4 0; -4 -4];

T = table(Degree,Offset1,Offset2,Offset3, Offset4)

writetable(T,'test.xlsx','Sheet',1);

%%

stats\_0\_s = graycoprops(glcm\_0\_s,{'contrast','correlation','energy','homogeneity'})

% stats\_0\_ns = graycoprops(glcm\_0\_ns,{'contrast','correlation','energy','homogeneity'})

stats = GLCM\_Features(glcm\_0\_s)

%% GRAY LEVEL RUN LENGTH MATRIX

% THIS PROGRAM SELECT A ROI, QUNATIZE TO LOWER BIT LEVEL AND COMPUTING

% GRAY LEVEL RUN LENGTH MATRIX AND SEVEN TEXTURE PARAMETERS VIZ.,

% 1. SHORT RUN EMPHASIS (SRE)

% 2. LONG RUN EMPHASIS(LRE)

% 3. GRAY LEVEL NON-UNIFORMITY (GLN)

% 4. RUN PERCENTAGE (RP)

% 5. RUN LENGTH NON-UNIFORMITY (RLN)

% 6. LOW GRAY LEVEL RUN EMPHASIS (LGRE)

% 7. HIGH GRAY LEVEL RUN EMPHASIS (HGRE)

%--------------------------------------------------------------------------

im = imread('ImgPIA.jpg');

figure

imshow(im)

im1= f4 %imcrop(im);

im2=im1(1:128,1:128);

im2=double(im2);

[m,n]=size(im2);

% Report is needed

% --------- IMAGE QUANTIZATION TO 4 BITS (16 GRAY LEVELS)------------------

bits = 16; % 4 bits - 16, 6 bits - 32, 8 bits - 64,

Imax=max(max(im2));

Imin=min(min(im2));

newim=im2-Imin;

Nmax=max(max(newim));

Nmin=min(min(newim));

Q=round(Nmax/bits);

[m,n]=size(newim);

Quant=0;

for i=1:m

for j=1:n

I = newim(i,j);

for B = 1:bits

if (I>Quant)&(I<=Quant+Q)

newim(i,j)=B/bits;

Quant=Quant+Q;

end

end

end

end

newmax=max(max(newim));

newim1=newim/newmax;

newim2=round(newim1\*bits)+1;

dir=0;

dist1=1;

if (dir == 1)

newim2=newim2';

end

mx = max(max(newim2));

mn = min(min(newim2));

gl = (mx-mn)+1;

[p,q] = size(newim2);

n=p\*q;

count=1;

c=1;

col=1;

grl(mx,p)=0;

maxcount(p\*q)=0;

mc=0;

%---------------------COMPUTING GRAY LEVEL RUN LENGTH MATRIX---------------

for j=1:p

for k=1:q-dist1

mc=mc+1;

g=newim2(j,k);

f=newim2(j,k+dist1);

if (g==f)&(g~=0)

count=count+1;

c=count;

col=count;

maxcount(mc)=count;

else grl(g,c)=grl(g,c)+1;col=1;

count=1;

c=1;

end

end

grl(f,col)=grl(f,col)+1;

count=1;

c=1;

end

I=(mx:mn);

m=grl(mn:mx,:);

m1=m';

maxrun=max(max(maxcount));

S=0;

G(gl)=0;

R(q)=0;

for u=1:gl

for v=1:q

G(u)=G(u)+m(u,v);

S=S+m(u,v);

end

end

for u1=1:q

for v1=1:gl

R(u1)=R(u1)+m1(u1,v1);

end

end

[dim,dim1]=size(G);

SRE=0; LRE=0; GLN=0; RLN=0; RP=0; LGRE=0; HGRE=0;

for h1= 1:maxrun

SRE=SRE+(R(h1)/(h1\*h1));

LRE=LRE+(R(h1)\*(h1\*h1));

RLN=RLN+(R(h1)\*R(h1));

RP=RP+R(h1);

end

SRE1=SRE/S;

LRE1=LRE/S;

RLN1=RLN/S;

RP1=RP/n;

for h2=1:gl

GLN=(GLN+G(h2)^2);

LGRE=LGRE+(G(h2)/(h2\*h2));

HGRE=HGRE+(h2\*h2)\*G(h2);

end

GLN1=GLN/S;

LGRE1=LGRE/S;

HGRE1=HGRE/S;

clc

% ---------------------------DISPLAY THE PARAMETERS------------------------

disp(sprintf('%6.4f',SRE1))

disp(sprintf('%6.4f',LRE1))

disp(sprintf('%6.4f',GLN1))

disp(sprintf('%6.4f',RP1))

disp(sprintf('%6.4f',RLN1))

disp(sprintf('%6.4f',LGRE1))

disp(sprintf('%6.4f',HGRE1))

**REFERENCE**

Pawel Kleczek (2022). GLCM\_Features(glcm) (https://www.mathworks.com/matlabcentral/fileexchange/56661-glcm\_features-glcm),

%MATLAB Central File Exchange. Retrieved May 6, 2022.

Xunkai Wei (2022). Gray Level Run Length Matrix Toolbox (https://www.mathworks.com/matlabcentral/fileexchange/17482-gray-level-run-length-matrix-toolbox), MATLAB Central File Exchange. Retrieved May 6, 2022.