**east west university**

**Lab Report - 05**

**Department:** **Computer Science and Engineering**

**Course Title:** Digital Image Processing

**Course Code:** CSE438

**Section No:** 02

**Submitted To**:

Dr. Engr. Ahmed Wasif Reza

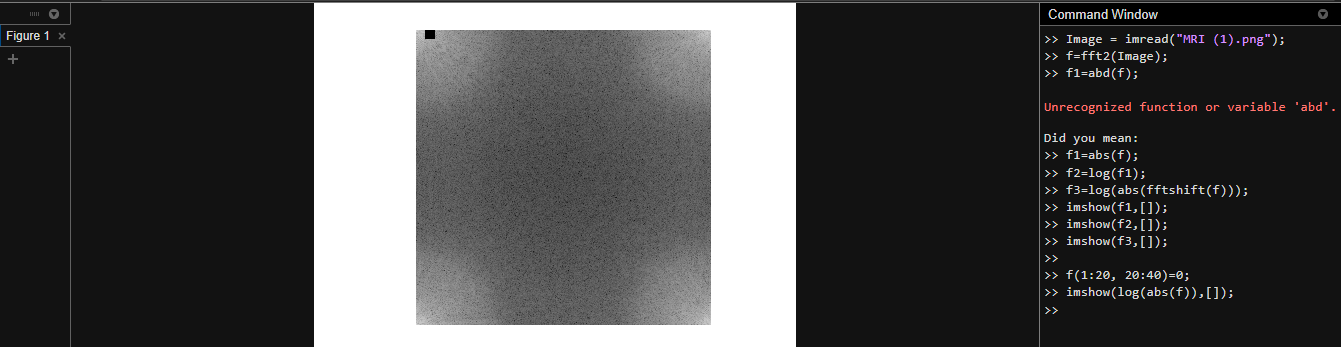
Associate Professor, Department of Computer Science and Engineering

**Submitted By**:

Name: S M Arafat Rahman

ID: **2019-2-60-094**

Apply Fourier transform to transform an image from the spatial domain to the frequency domain.



Code:

Image = imread("MRI (1).png");

f=fft2(Image);

f1=abd(f);

f1=abs(f);

f2=log(f1);

f3=log(abs(fftshift(f)));

imshow(f1,[]);

imshow(f2,[]);

imshow(f3,[]);

f(1:20, 20:40)=0;

imshow(log(abs(f)),[]);

Apply inverse Fourier transform to transform the image from the frequency domain to the spatial domain.



I = imread("MRI (1).png");

i = ifft2(I);

i1 = log(abs(j));

i2 = fftshift(i);

i3 = fftshift(i2);

f(1:20, 20:40) = 0;

i4 = ifft2(i);

imshow(i,[]);

imshow(i1,[]);

imshow(i2,[]);

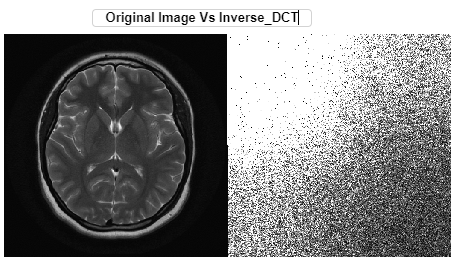
imshow(i3,[]);

imshow(abs(i4),[]);

Apply discrete cosine transform to transform an image from the spatial domain to the frequency domain. Apply inverse discrete cosine transform to transform the image from the frequency domain to the spatial domain.

A picture containing text, old, set

Description automatically generated



Code:

imshow(log(abs(Compress\_Image)),[])

Compress\_Image(abs(Compress\_Image)<5) = 0;

K = idct2(Compress\_Image);

K = rescale(K);

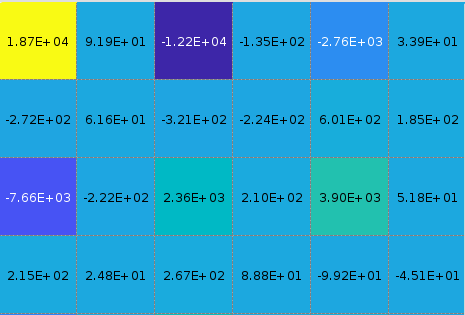
figure,montage({I,K});

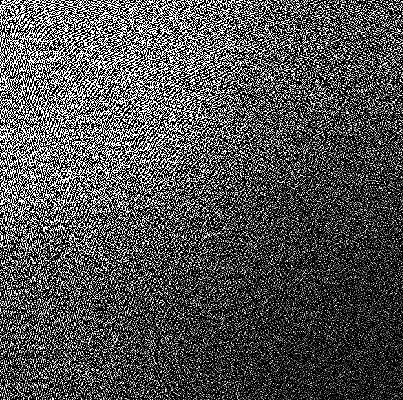
Inverse\_DCT = abs(idct2(I));

figure,imshow(K,Inverse\_DCT);

figure,imshow(Inverse\_DCT);

Compress the image using Discrete Cosine Transform.





Code:

I = imread("MRI (1).png");

ad = dct2(I);

h=imagesc(ad);

impixelregion(h);

imshow(ad);

Compress the image using Haar Transform.

A picture containing shape

Description automatically generated

Code:

I = 'MRI.png';

delta = 0.0001;

disp(delta);

   1.0000e-04

if(delta>1 || delta <0)

error('harr\_wt: Delta value range is between 0 to 1');

end

H1=[0.5 0 0 0 0.5 0 0 0;0.5 0 0 0 -0.5 0 0 0;0 0.5 0 0 0 0.5 0 0 ;0 0.5 0 0 0 -0.5 0 0; 0 0 0.5 0 0 0 0.5 0;0 0 0.5 0 0 0 -0.5 0;0 0 0 0.5 0 0 0 0.5;0 0 0 0.5 0 0 0 -0.5;];

H2=[0.5 0 0.5 0 0 0 0 0;0.5 0 -0.5 0 0 0 0 0;0 0.5 0 0.5 0 0 0 0;0 0.5 0 -0.5 0 0 0 0; 0 0 0 0 1 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];

H3=[0.5 0.5 0 0 0 0 0 0;0.5 -0.5 0 0 0 0 0 0;0 0 1 0 0 0 0 0;0 0 0 1 0 0 0 0;0 0 0 0 1 0 0 0; 0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1;];

H1o = (H1.\*(2^0.5));

H2o = (H2.\*(2^0.5));

H3o = (H3.\*(2^0.5));

Ho=normc(H1o\*H2o\*H3o);

H = H1\*H2\*H3;

x=double(imread(I));

len=length(size(x));

if len~=2

error('harr wt: Input image must be a gray image, use "harr\_wt\_rgb" function to compress RGB image ');

end

yo = zeros(size(x));

y = zeros(size(x));

[r,c]=size(x);

for i=0:8:r-8

for j=0:8:c-8

p=i+1;

q=j+1;

yo(p:p+7,q:q+7)=(Ho')\*x(p:p+7,q:q+7)\*Ho;

y(p:p+7,q:q+7)=(H')\*x(p:p+7,q:q+7)\*H;

end

end

figure;

imshow(x/255);

n1=nnz(y);

zo=yo;

m=max(max(yo));

yo=yo/m;

yo(abs(yo)<delta)=0;

yo=yo\*m;

z=y;

y=y/m;

y(abs(y)<delta)=0;

y=y\*m;

n2=nnz(y);

for i=0:8:r-8

for j=0:8:c-8

p=i+1;

q=j+1;

zo(p:p+7,q:q+7)=Ho\*yo(p:p+7,q:q+7)\*Ho;

z(p:p+7,q:q+7)=inv(H')\*y(p:p+7,q:q+7)\*inv(H);

end

end

figure;

subplot(121);

imshow(x/255);

title("Original Image");

subplot(122)

imshow(z/255);

title("Compressed Image");

imwrite(x/255,'original.tif');

imwrite(z/255,'compossed.tif');

compression\_ratio = n2/n1;