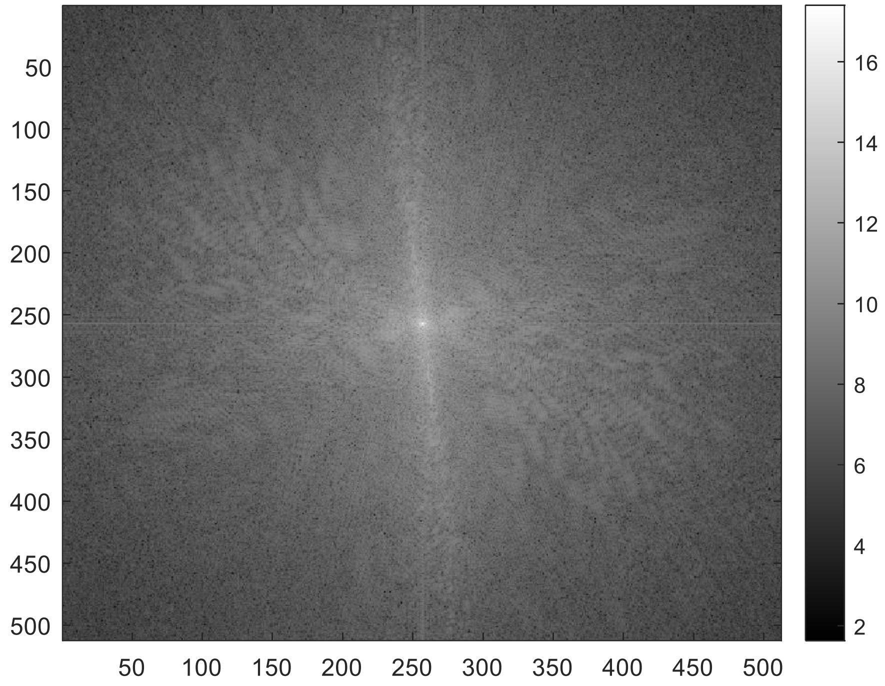
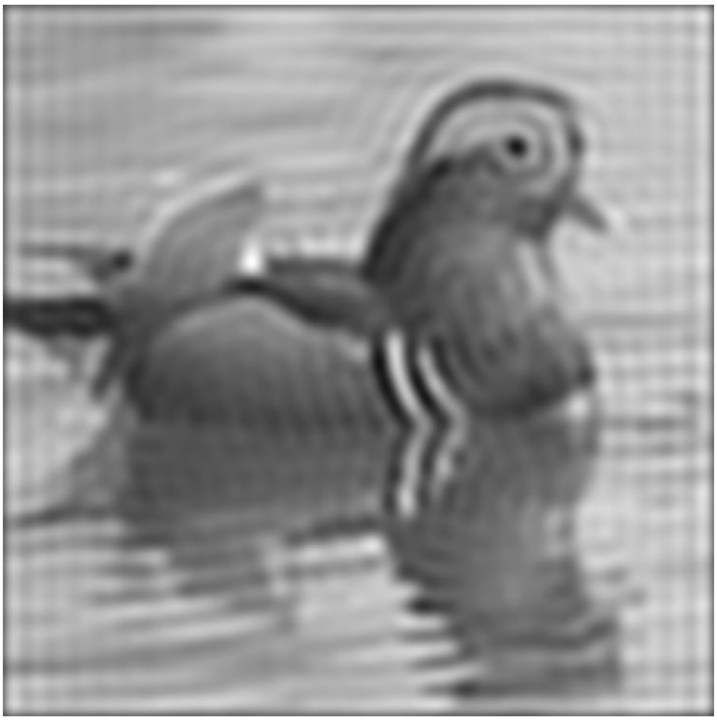
Project 2

1. Figure of the Fourier magnitude spectra using Log scale.

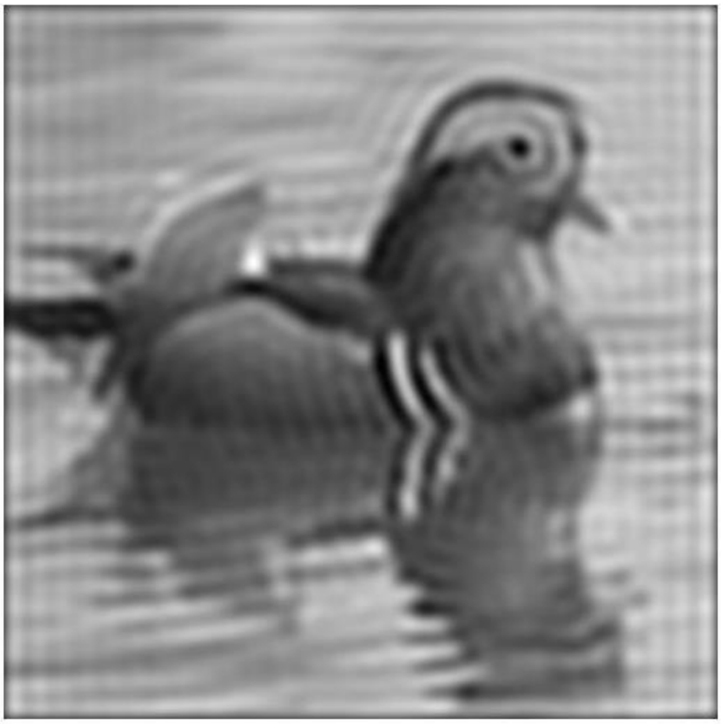


2. Image constructed by DFT coefficients inside the circular region with radius =30 pixels.

Take real part after iDft

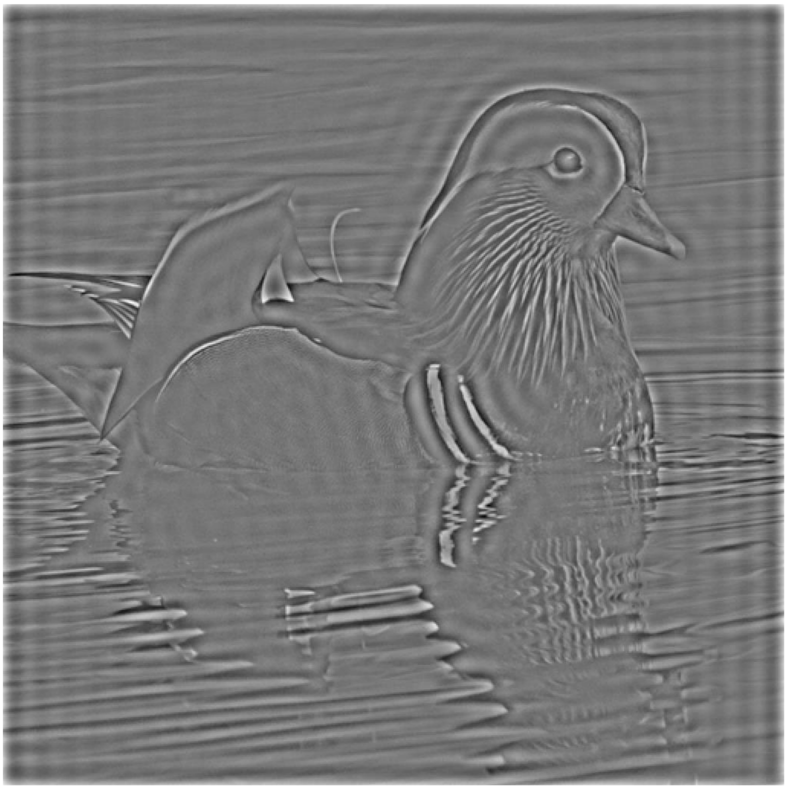


Take absolute value after iDft

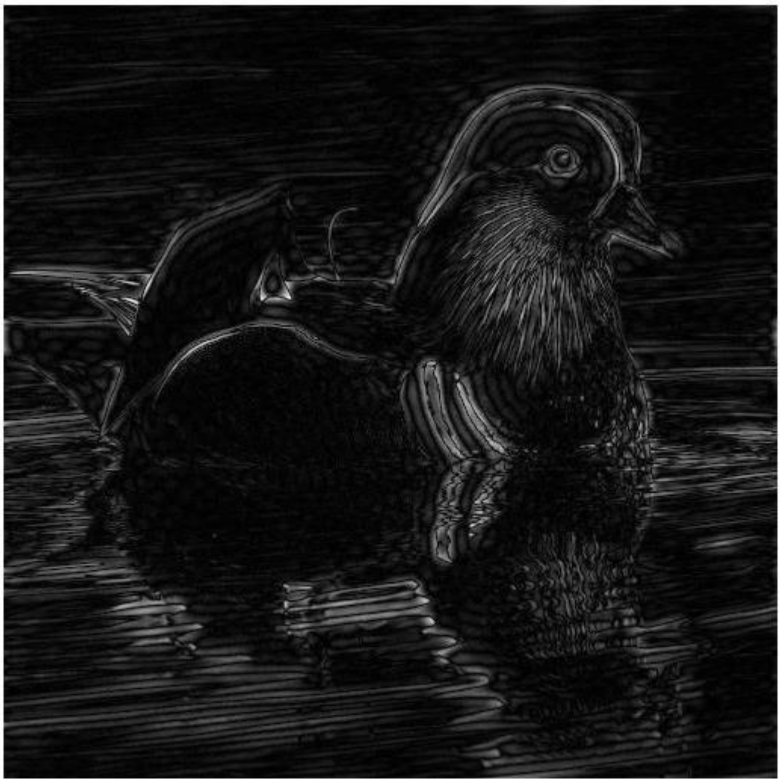


3. Image constructed by DFT coefficients outside the circular region with radius =30 pixels.

Take real part after iDft



Take absolute value after iDft



4. Table of top 25 DFT frequencies (*u*, *v*) in the left half frequency region

Top25	*u*	*v*
1	256	254
2	256	255
3	255	255
4	257	255
5	257	254
6	253	255
7	259	254
8	258	255
9	259	255
10	253	254
11	256	253
12	258	252
13	254	254
14	258	253
15	252	253
16	248	255
17	254	255

18	254	252
19	260	254
20	262	255
21	254	253
22	255	252
23	255	254
24	252	255
25	261	254

5. Source code

```
clc
clear

% Read the image, data type: uint8
I=imread('Bird 2.tif');

% Get Fourier transform of input image and change the data type to double
F=fft2(double(I));

% Shift zero_frequency component to center of spectrum
S_F=fftshift(F);

% The Fourier magnitude spectra using Log scale
F_log=log(1+abs(S_F));

% Show the image's Fourier magnitude in Log scale
figure(1)
imagesc(F_log)
colormap('gray') % Let the image present gray-level
colorbar % show colorbar
%title('Fourier magnitude of the image')

% Images re-synthesize (inside, outside)
M = size(I,1);
N = size(I,2);
```

```matlab
center_v = 2*M/2;  % the centered coordinate of the image (v)
center_u = 2*N/2;  % the centered coordinate of the image (u)

w_inside30 = zeros(2*M,2*N);
w_outside30 = zeros(2*M,2*N);

d=30; % radius=30

% Padded image of size(2M*2N)
I_1024 = zeros(1024,1024);
I_1024(1:M,1:N) = double(I);

% Construct  filters
for i=1:2*M
   for ii=1:2*N
      if sqrt((i-center_v).^2 + (ii-center_u).^2) < 2*d
         w_inside30(i,ii)=1;
         w_outside30(i,ii)=0;
      else
         w_inside30(i,ii)=0;
         w_outside30(i,ii)=1;
      end
   end
end

F_1024=fft2(I_1024);
S_F_1024=fftshift(F_1024);

output=w_inside30.*S_F_1024;
output1=w_outside30.*S_F_1024;


% Shift the zero-frequency component back
output=ifftshift(output);
output1=ifftshift(output1);

% Get the output image using 2-D fast Foirier transform
output=ifft2(output);
```

```
output1=ifft2(output1);

% Adjust the scale range to 0-255
output = real(output);  %Both absolute value and real part is fine
output = output-min(output(:));
output_inside30 = output ./ max(output(:)).*255;

output1 = real(output1); %Both absolute value and real part is fine
output1 = output1-min(output1(:));
output_outside30 = output1 ./ max(output1(:)).*255;

%Crop M*N image
output_inside30=output_inside30(1:M,1:N);
output_outside30=output_outside30(1:M,1:N);

% Show the out put image
figure(2)
imshow(uint8(output_inside30));
%title('Synthesized image inside 30')

figure(3)
imshow(uint8(output_outside30));
%title('Synthesized image outside 30')

% Top 25 freq in the half freq region(0 <= u <= M-1, 0 <= v <= N/2-1)
M = size(I,1);
N = size(I,2);
input_top25=[];
img_log= F_log;
for kk=1:25
    a=max(max(img_log(1:M,1:N/2)));
    for u=1:M
        for v=1:N/2
            if img_log(u,v) == a
                input_top25{kk}=[u-1,v-1]; %start from(0,0)
                img_log(u,v) = 0;
            end
        end
    end
end
```