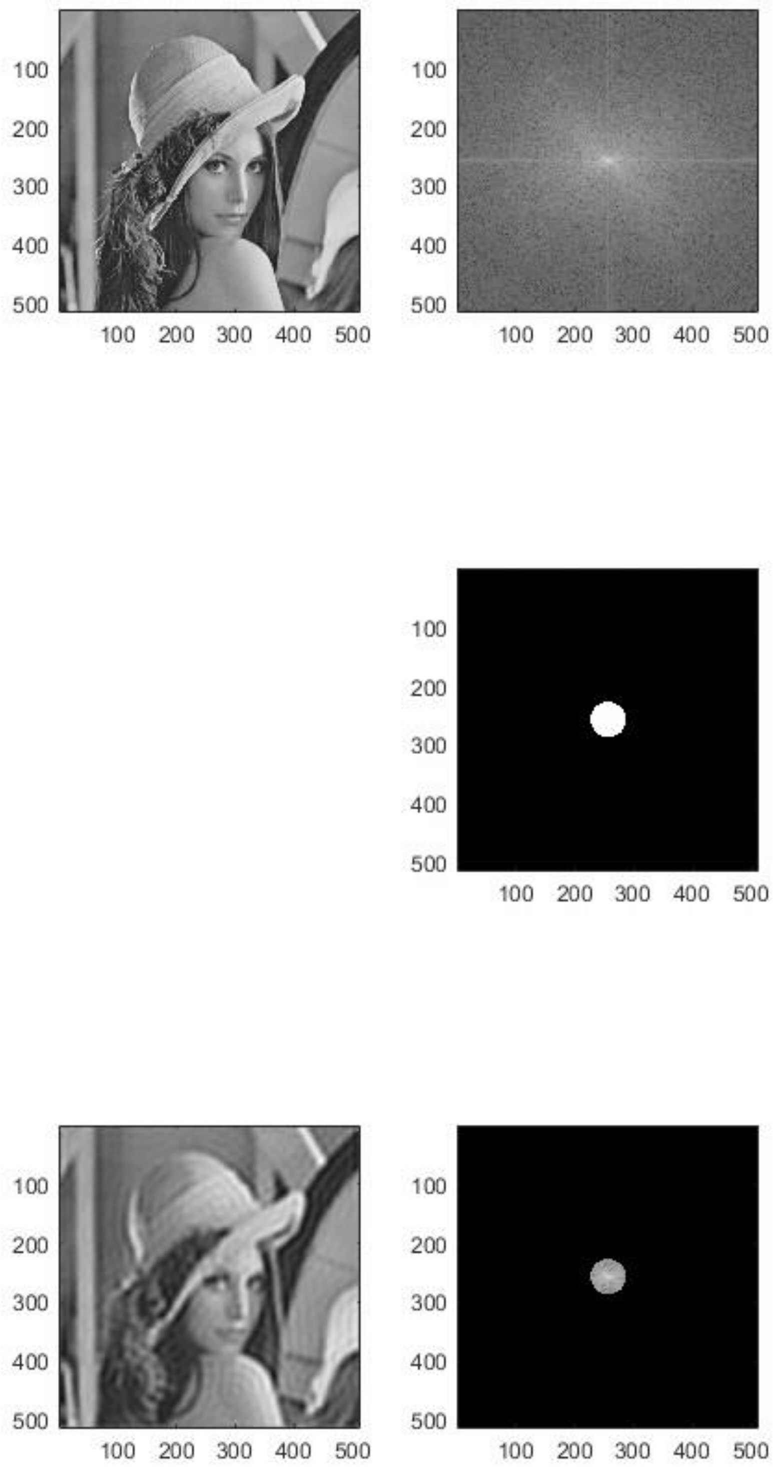
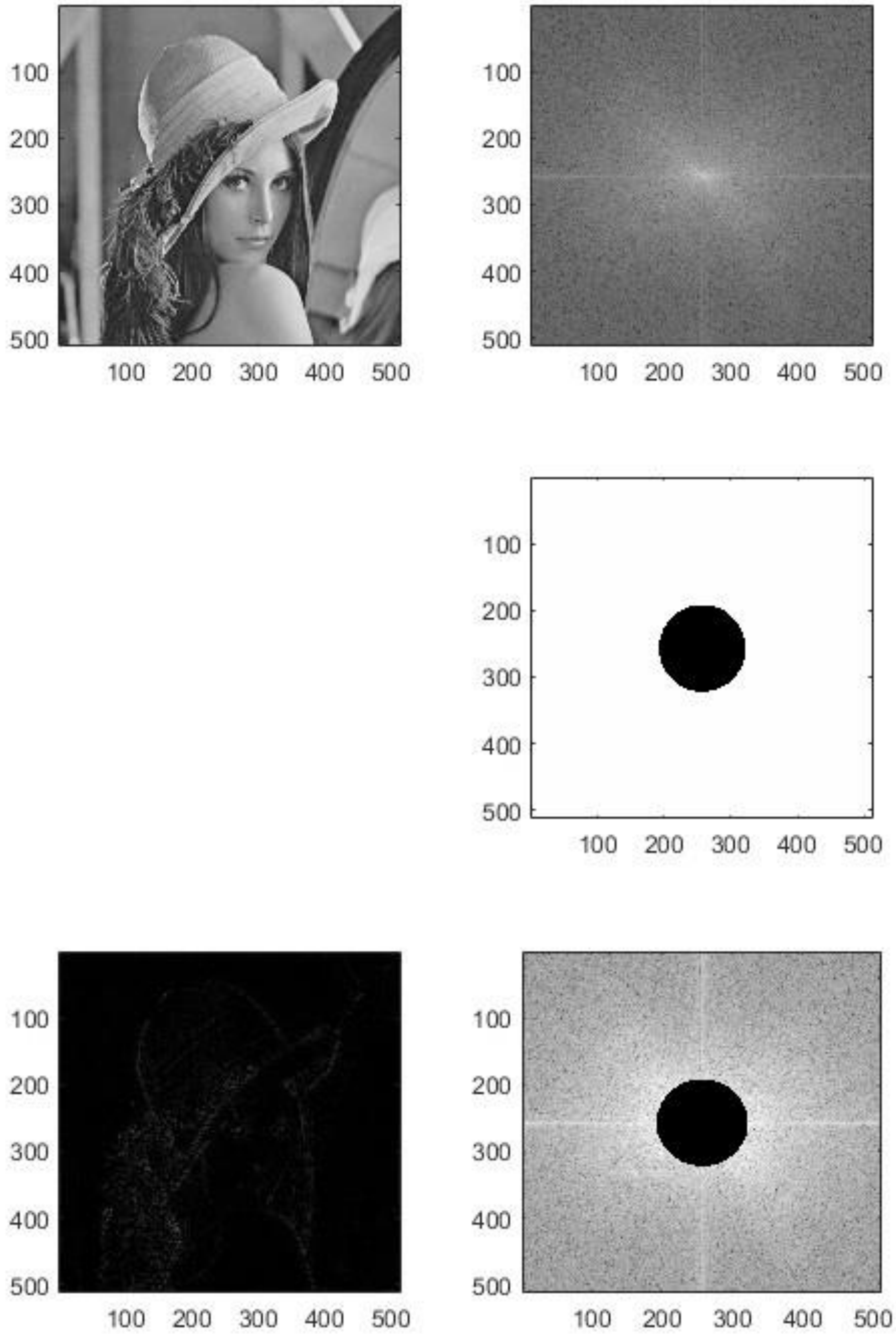
1. Ideal Low-pass filter

```
% change size of z to view the differences of low pass filter
[xLena, map]=imread('lena.bmp'); % xLena = imread('lena.bmp') -> [xLena, map] = imread('lena.bmp')
xLenaf=fftshift(fft2(xLena));
flog = log(1+abs(xLenaf));
fm = max(flog(:));
fig = figure();
fig.Position(3:4) = [1000, 1500];
subplot(3,2,1)
image(xLena);
colormap(map), axis('square')
subplot(3,2,2)
image(flog*255/fm);
colormap(map), axis('square')
[x,y]= meshgrid(-256:255,-256:255);
z = sqrt(x.^2+ y.^2);
% clf= (z<15);
clf = (z < 30); % set z < 30
subplot(3,2,4)
image(255*clf);
colormap(map), axis('square')
subplot(3,2,6)
yLenaf=xLenaf.*clf;
flog = log(1+abs(yLenaf));
fm = max(flog(:));
image(flog*255/fm);
colormap(map), axis('square')
yLena=ifft2(yLenaf);
subplot(3,2,5)
image(abs(yLena));
colormap(map), axis('square')

% If z goes larger, less high-frequency gets filtered out, the resulting
% image become sharper.
% If z goes smaller, more high-frequency get filter out, the resulting
% image become smoother.
```


100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500

2. Ideal High-pass filter

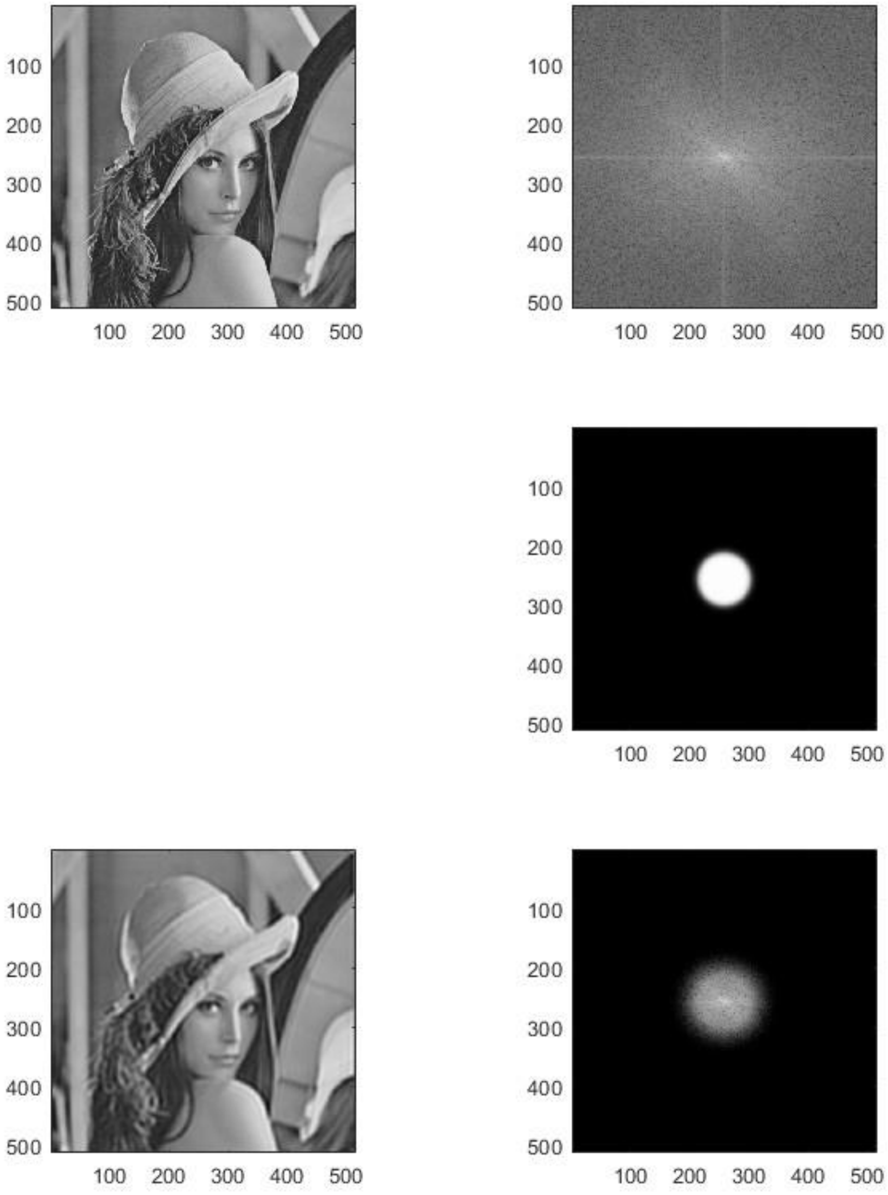
```
% change size of z to view the differences of high pass filter
% ======== new =========
[xLena, map]=imread('lena.bmp');
xLenaf=fftshift(fft2(xLena));
% ======================
fig = figure();
fig.Position(3:4) = [1000, 1500];
subplot(3,2,1)
image(xLena);
colormap(map), axis('square')
subplot(3,2,2)
flog = log(1+abs(xLenaf));
fm = max(flog(:));
image(flog*255/fm);
colormap(map), axis('square')
[x,y]=meshgrid(-256:255,-256:255);
z=sqrt(x.^2+y.^2);
% chf=(z>15);
chf = (z > 65); % set z > 65
subplot(3,2,4)
image(255*chf);
colormap(map), axis('square')
yLenaf=xLenaf.*chf;
flog = log(1+abs(yLenaf));
fm = max(flog(:));
subplot(3,2,6)
image(flog*255/fm);
colormap(map)
yLena=ifft2(yLenaf);
subplot(3,2,5)
image(abs(yLena));
colormap(map), axis('square')

% If z goes larger, more frequency gets filtered out, the resulting image become sharper (more edge detail).
% However, if z goes too large, the high frequency would also be filter out, resulting a dark image.
```


100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500
100
200
300
400
500
100 200 300 400 500

3. Butterworth low-pass filter (blf)

```
% change size of D and n to view the differences of Butterworth low-pass filter
% ======== new =========
[xLena, map]=imread('lena.bmp');
xLenaf=fftshift(fft2(xLena));
% ======================
fig = figure();
fig.Position(3:4) = [1000, 1500];
subplot(3,2,1)
image(xLena);
colormap(map), axis('square')

subplot(3,2,2)
flog = log(1+abs(xLenaf));
fm = max(flog(:));
image(flog*255/fm);
colormap(map), axis('square')

%D=15, n=2
D=2048; % set D = 2048
n=8;    % set n = 8
[x,y]=meshgrid(-256:255,-256:255);
blf=1./(1+((x.^2+y.^2)/D).^n);
subplot(3,2,4)
image(255*blf);
colormap(map), axis('square')

yLenaf=xLenaf.*blf;
flog = log(1+abs(yLenaf));
fm = max(flog(:));
subplot(3,2,6)
image(flog*255/fm);
colormap(map), axis('square')

yLena=ifft2(yLenaf);
subplot(3,2,5)
image(abs(yLena));
colormap(map), axis('square')
```

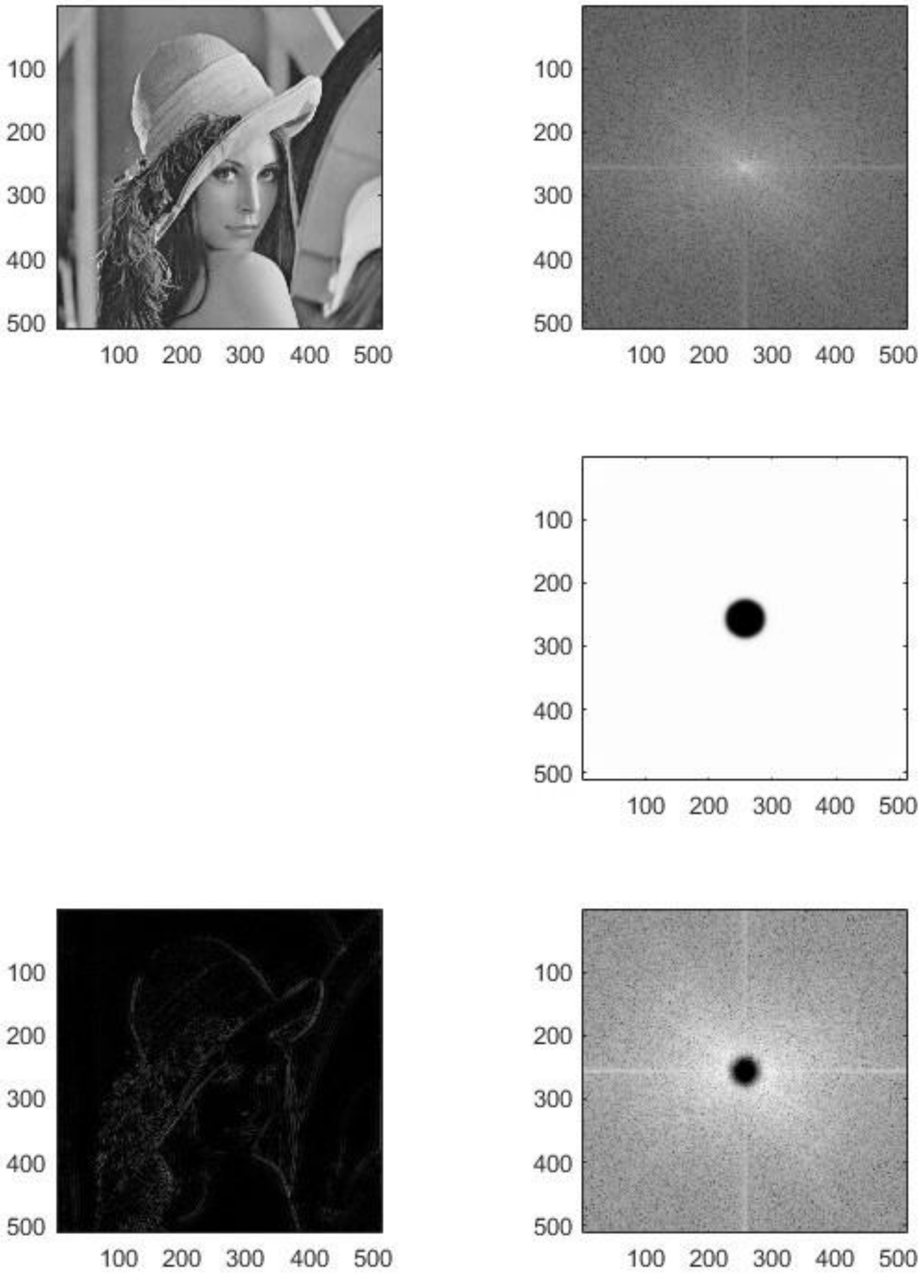
```
% if D goes larger, more frequency would be covered/retained, resulting a clearer(less blur) image.
% if n goes larger, more frequency would be filtered out, resulting a blurer image, and when n reach a certain value, the effect becomes no longer significant.
```



4. Butterworth high-pass filter (bhf)

```
% change size of D and n to view the differences of Butterworth high-pass
filter
% ======== new =========
[xLena, map]=imread('lena.bmp');
xLenaf=fftshift(fft2(xLena));
% ======================
fig = figure();
fig.Position(3:4) = [1000, 1500];
subplot(3,2,1)
image(xLena);
colormap(map), axis('square')

subplot(3,2,2)
flog = log(1+abs(xLenaf));
fm = max(flog(:));
image(flog*255/fm);
colormap(map), axis('square')

%D=15, n=2
D=960;
n=8;
[x,y]=meshgrid(-256:255,-256:255);
blf=1./(1+((x.^2+y.^2)/D).^n);
bhf=1-blf; subplot(3,2,4)
image(255*bhf);
colormap(map), axis('square')

yLenaf=xLenaf.*bhf;
flog = log(1+abs(yLenaf));
fm = max(flog(:));
subplot(3,2,6)
image(flog*255/fm);
colormap(map), axis('square')

yLena=ifft2(yLenaf);
subplot(3,2,5)
image(abs(yLena));
colormap(map), axis('square')
```

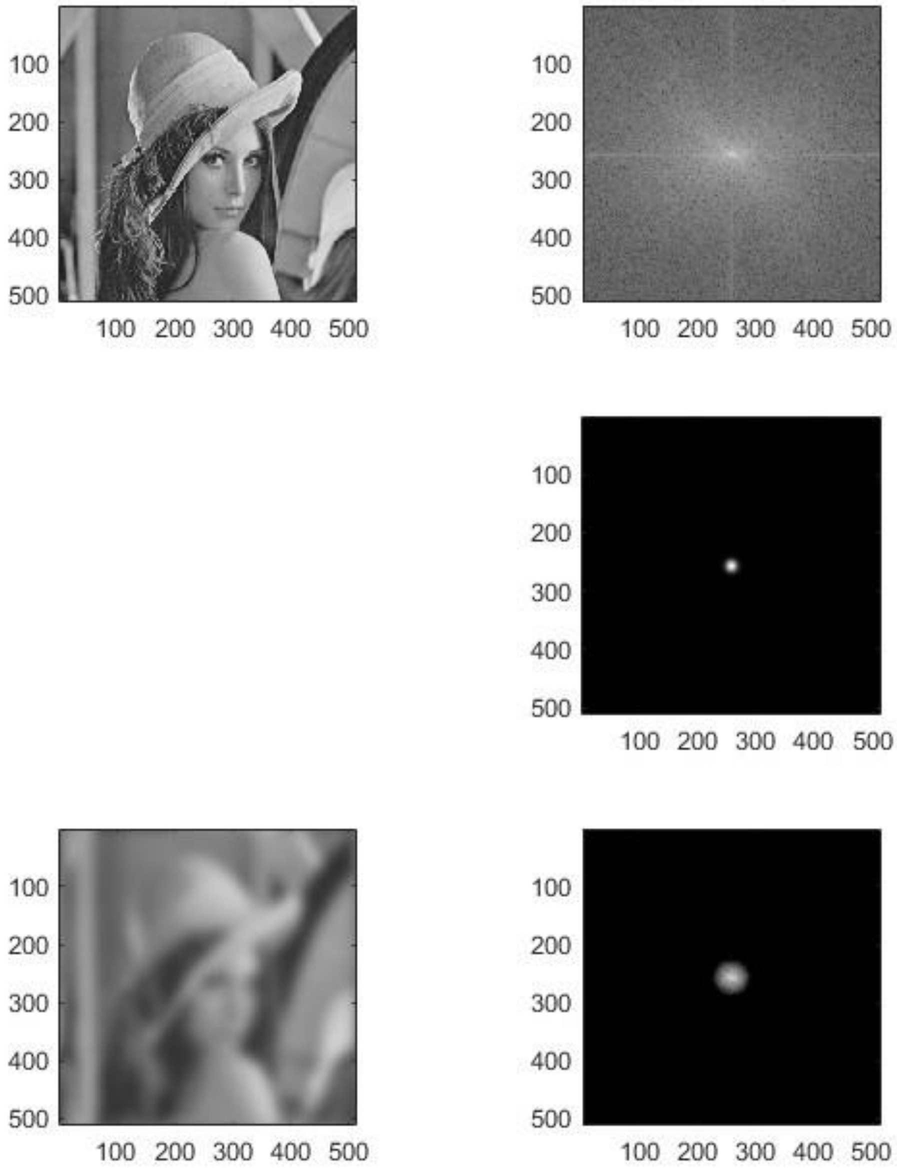
```
% If D goes larger, more low-frequency would be filtered out, resulting a sharper image, vice versa.
% IF N goes larger, more low-frequency would be filtered out, resulting a sharper image, vice versa.
```



5. Gaussian Low-Pass Filter (glf)

```
% change size of std (standard deviation)
% ======== new =========
[xLena, map]=imread('lena.bmp');
xLenaf=fftshift(fft2(xLena));
% ======================
fig = figure();
fig.Position(3:4) = [1000, 1500];
subplot(3,2,1)
image(xLena);
colormap(map), axis('square')

subplot(3,2,2)
flog = log(1+abs(xLenaf));
fm = max(flog(:)); image(flog*255/fm);
colormap(map), axis('square')

gsize=512;
std = 7;
[x,y] = meshgrid(-256:255,-256:255);
arg = -(x.*x + y.*y)/(2*std*std);
glf = exp(arg);
glf(glf<eps*max(glf(:))) = 0;
%sumh = sum(glf(:));
%if sumh ~= 0,
%glf =glf/sumh;
%end;

subplot(3,2,4)
image(255*glf);
colormap(map), axis('square')

yLenaf=xLenaf.*glf;
flog = log(1+abs(yLenaf));
fm = max(flog(:));
subplot(3,2,6)
image(flog*255/fm);
colormap(map), axis("square")
```

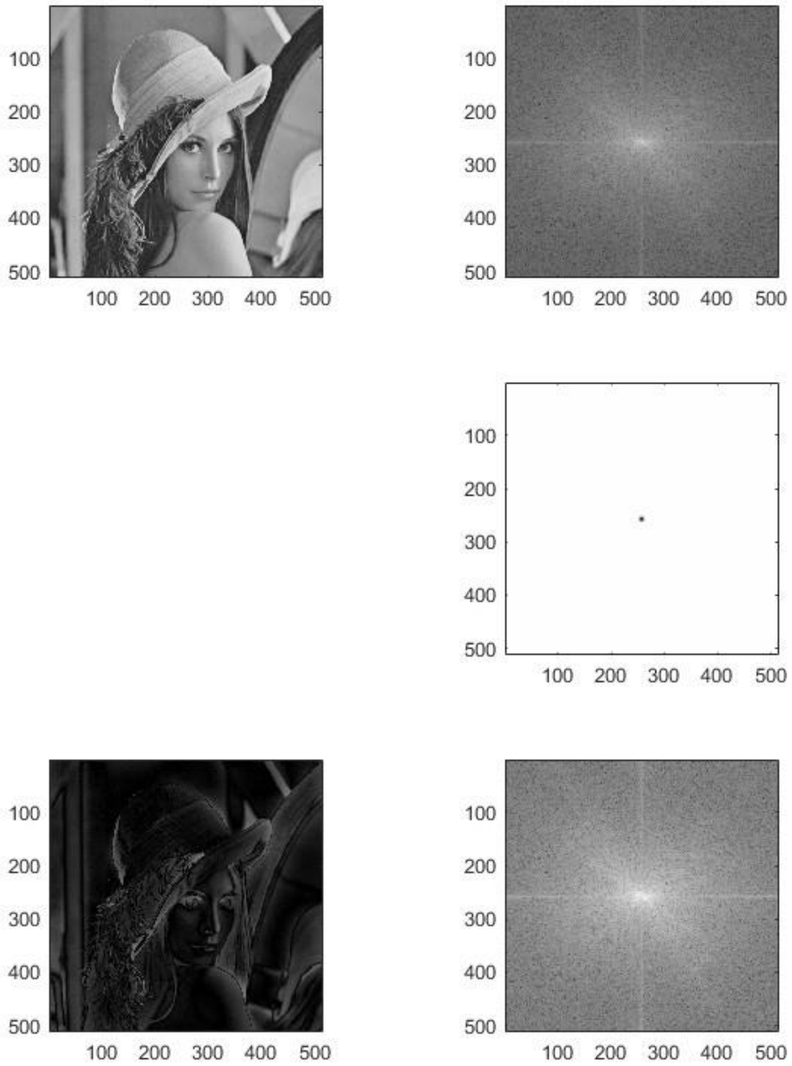
```
yLena=ifft2(yLenaf);
subplot(3,2,5)
image(abs(yLena));
colormap(map),axis('square')

% If std goes smaller, more high-frequency will be flitered out, resulting a smoother image, vice versa.
```



6. Gaussian High-Pass Filter (ghf)

```
% change size of std (standard deviation)
% ======== new =========
[xLena, map]=imread('lena.bmp');
xLenaf=fftshift(fft2(xLena));
% ======================
fig = figure();
fig.Position(3:4) = [1000, 1500];
subplot(3,2,1)
image(xLena);
colormap(map), axis('square')

subplot(3,2,2)
flog = log(1+abs(xLenaf));
fm = max(flog(:)); image(flog*255/fm);
colormap(map), axis('square')

gsize=512;
std = 3;
[x,y] = meshgrid(-256:255,-256:255);
arg = -(x.*x + y.*y)/(2*std*std);
glf = exp(arg);
glf(glf<eps*max(glf(:))) = 0;
sumh = sum(glf(:));
%if sumh ~= 0
%glf =glf/sumh;
%end

% ====== new ======
ghf = 1 - glf;
% ================

subplot(3,2,4)
image(255*ghf); % glf -> ghf
colormap(map), axis('square')

yLenaf=xLenaf.*ghf; % glf -> ghf
flog = log(1+abs(yLenaf));
fm = max(flog(:));
```

```
subplot(3,2,6)
image(flog*255/fm);
colormap(map), axis("square")

yLena=ifft2(yLenaf);
subplot(3,2,5)
image(abs(yLena));
colormap(map),axis('square')

% If std goes larger, more low-frequency will be flitered out, resulting a sharper image, vice versa.
```


100
200
300
400
500
100 200 300 400 500