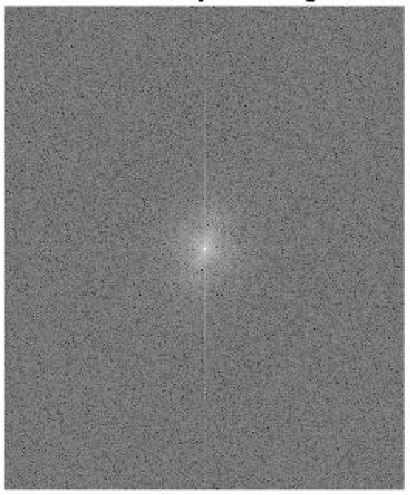
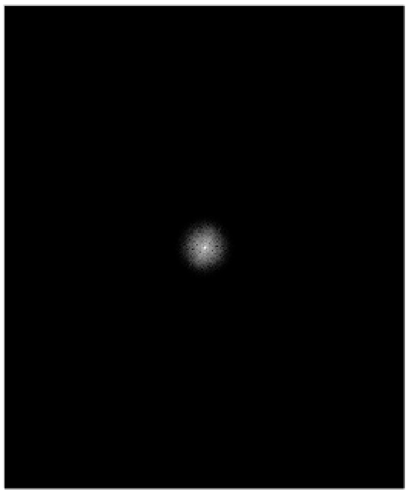
```
image = imread('Digital Image Processing/Assignment 2/charlie.png');
fft_image = fftshift(fft2(image));
imshow(mat2gray(log(1+abs(fft_image))));
title('FFT of Input Image')
```

FFT of Input Image



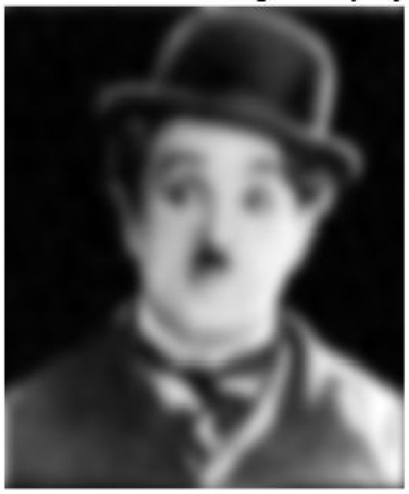
```
%Low Pass Gaussian Filtering

%Creating the Gaussian Filter
[M, N] = size(fft_image);  % Get the size of the FFT image
G_filter = fspecial('gaussian', [M N], 10);  % Create a Gaussian filter with
the same size

%Applying fft of gaussian filter
filtered_fft_image = fft_image.*G_filter;

%for distplaying the fft of filtered image
imshow(mat2gray(log(1+abs(filtered_fft_image))));
title('FFT of Low Pass Gaussian Filtered Image');
```

FFT of Low Pass Gaussian Filtered Image



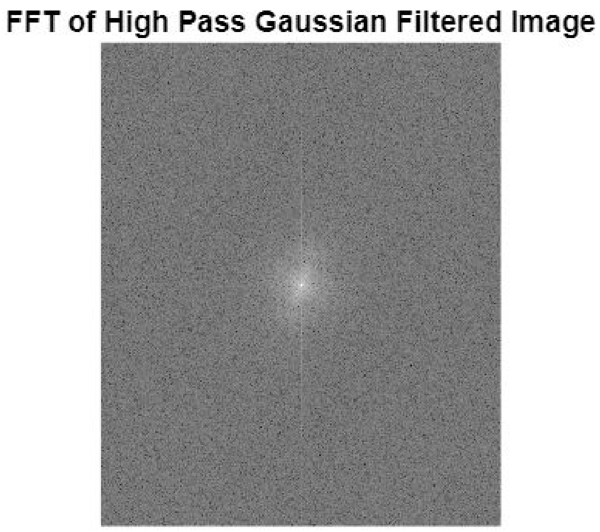
```
%for displaying the filtered image
inverse_filtered_fft_image = ifft2(ifftshift(filtered_fft_image));  % Use
ifftshift before ifft2

%displaying the real part
imshow(mat2gray(real(inverse_filtered_fft_image)));
title('IFFT of Low Pass Gaussian Filtered Image/ Displaying the Filtered
Image')
```

Low Pass Gaussian Filtered Image/ Displaying the Filter

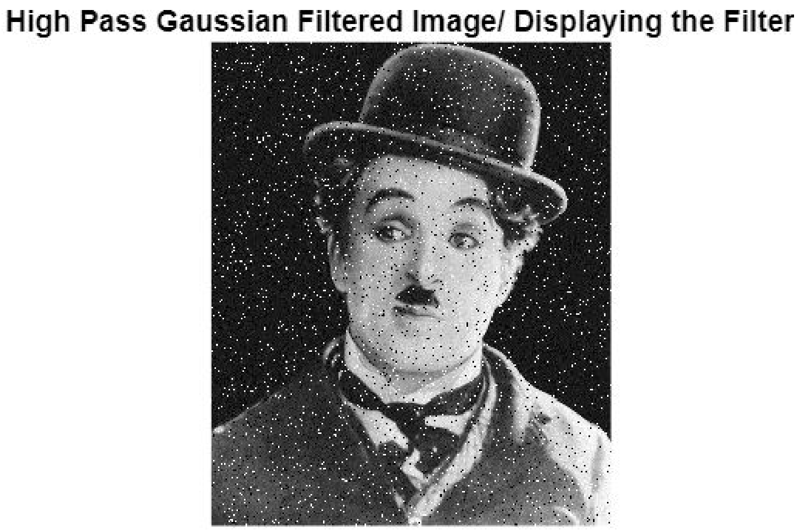


```
%High Pass Gaussian Filter

H_G_filter = 1- G_filter;

%Applying fft of gaussian filter
```

```
H_G_filtered_fft_image = fft_image.*H_G_filter;

%for distplaying the fft of filtered image
imshow(mat2gray(log(1+abs(H_G_filtered_fft_image))));
title('FFT of High Pass Gaussian Filtered Image');
```

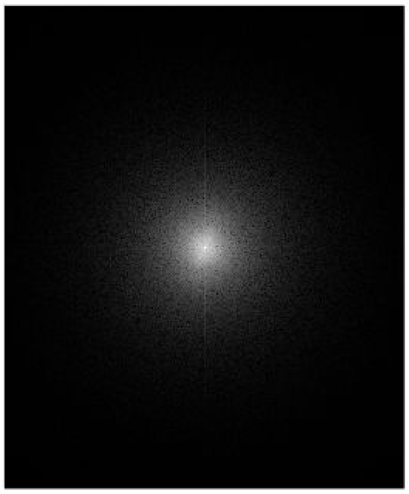


```


%for displaying the filtered image
H_G_inverse_filtered_fft_image = ifft2(ifftshift(H_G_filtered_fft_image));
% Use ifftshift before ifft2

%displaying the real part
imshow(mat2gray(real(H_G_inverse_filtered_fft_image)));
title('IFFT of High Pass Gaussian Filtered Image/ Displaying the Filtered
Image')
```



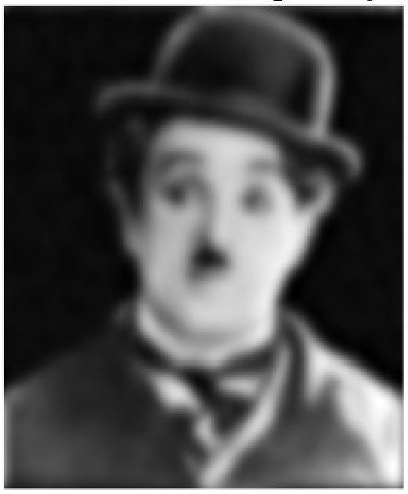
```
%Butterworth Low Pass Filtering

%Creating a Butterworth Filter with D=200 (D is cut off frequency) and n=2
%(n is order of filter

% Create a meshgrid for the filter that matches the size of fft_image
[x, y] = meshgrid(-N/2:(N/2-1), -M/2:(M/2-1));
B_filter = 1 ./ (1 + ((x.^2 + y.^2) / 200).^2);

%Applying B_filter
B_filtered_fft_image = fft_image.*B_filter;

%for distplaying the fft of filtered image
imshow(mat2gray(log(1+abs(B_filtered_fft_image))));
title('FFT of Low Pass Butterworth Filtered Image');
```

FFT of Low Pass Butterworth Filtered Image



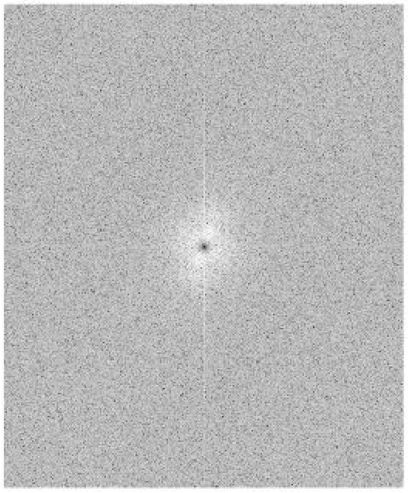
```

%for displaying the filtered image
inverse_B_filtered_fft_image = ifft2(ifftshift(B_filtered_fft_image));  %
Use ifftshift before ifft2

%displaying the real part
imshow(mat2gray(real(inverse_B_filtered_fft_image)));
title('IFFT of Low Pass Butterworth Filtered Image/ Displaying the Filtered
Image')
```

.ow Pass Butterworth Filtered Image/ Displaying the Filte



```
%High Pass Butterworth Filtering

H_B_filter = 1-B_filter;

%Applying H_B_filter
H_B_filtered_fft_image = fft_image.*H_B_filter;

%for distplaying the fft of filtered image
imshow(mat2gray(log(1+abs(H_B_filtered_fft_image))));
title('FFT of High Pass Butterworth Filtered Image');
```

FFT of High Pass Butterworth Filtered Image



```
%for displaying the filtered image
inverse_H_B_filtered_fft_image = ifft2(ifftshift(H_B_filtered_fft_image));
% Use ifftshift before ifft2
```

```
%displaying the real part
imshow(mat2gray(real(inverse_H_B_filtered_fft_image)));
title('IFFT of High Pass Butterworth Filtered Image/ Displaying the Filtered
Image')
```