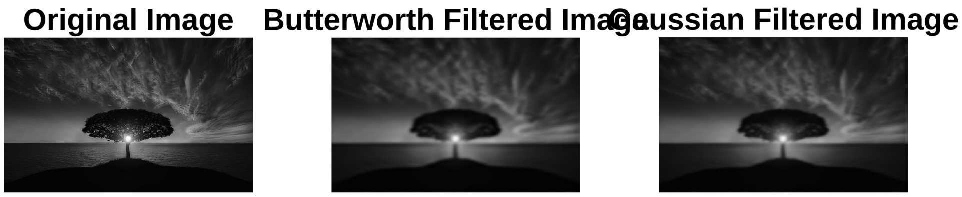
```matlab
% Load the image
img = imread('https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885_640.jpg'); % Replace with your image file
img = rgb2gray(img); % Convert to grayscale if it's a color image
img = im2double(img); % Convert to double precision
% Compute the Fourier Transform
F = fft2(img);
F_shifted = fftshift(F); % Shift zero frequency component to center
% Create a Butterworth filter
[M, N] = size(img);
D0 = 30; % Cutoff frequency
n = 2;   % Order of the filter
[U, V] = meshgrid(-floor(N/2):floor((N-1)/2), -floor(M/2):floor((M-1)/2));
D = sqrt(U.^2 + V.^2); % Distance from the center
H_butterworth = 1 ./ (1 + (D ./ D0).^(2*n));
% Create a Gaussian filter
sigma = 30; % Standard deviation
H_gaussian = exp(-(D.^2) / (2 * sigma^2));
% Apply Butterworth filter
F_butterworth = F_shifted .* H_butterworth;

% Apply Gaussian filter
F_gaussian = F_shifted .* H_gaussian;
% Inverse Fourier Transform
img_butterworth = ifft2(ifftshift(F_butterworth));
img_gaussian = ifft2(ifftshift(F_gaussian));

% Display the results
figure;
subplot(1, 3, 1), imshow(img), title('Original Image');
subplot(1, 3, 2), imshow(abs(img_butterworth)), title('Butterworth Filtered Image');
subplot(1, 3, 3), imshow(abs(img_gaussian)), title('Gaussian Filtered Image');
```



Original Image

Butterworth Filtered Image

Gaussian Filtered Image