```
img = imread('https://img.artpal.com/828282/1-23-6-1-13-33-8m.jpg');

% Convert to grayscale if it's a color image
if size(img, 3) == 3
    img = rgb2gray(img);
end

% Convert to double precision
img = im2double(img);

% Get image dimensions
[M, N] = size(img);

% Compute 2D Fourier Transform
F = fft2(img);
F_shift = fftshift(F);

% Create frequency grid
[u, v] = meshgrid(linspace(-N/2, N/2-1, N), linspace(-M/2, M/2-1, M));

% Distance from origin for each point in the frequency domain
D = sqrt(u.^2 + v.^2);

% Butterworth filter
n = 2; % Order of the Butterworth filter
D0 = 30; % Cutoff frequency
H_butterworth = 1 ./ (1 + (D ./ D0).^(2*n));

% Gaussian filter
sigma = 30;
H_gaussian = exp(-(D.^2) ./ (2 * sigma^2));

% Apply filters
F_butterworth = F_shift .* H_butterworth;
F_gaussian = F_shift .* H_gaussian;

% Inverse Fourier Transform
img_butterworth = real(ifft2(ifftshift(F_butterworth)));
img_gaussian = real(ifft2(ifftshift(F_gaussian)));

% Display results
figure;

subplot(2,3,1);
imshow(img);
title('Original Image');

subplot(2,3,2);
imshow(log(1 + abs(F_shift)), []);
title('Fourier Transform (log scale)');
```

```
subplot(2,3,3);
imshow(H_butterworth);
title('Butterworth Filter');

subplot(2,3,4);
imshow(img_butterworth);
title('Butterworth Filtered Image');

subplot(2,3,5);
imshow(H_gaussian);
title('Gaussian Filter');

subplot(2,3,6);
imshow(img_gaussian);
title('Gaussian Filtered Image');
```