

```

% Read the image
I = imread('cat.jpeg');
I = rgb2gray(I); % Convert to grayscale if necessary
I = double(I);

% Step 1: Apply Fourier Transform to the image
F = fft2(I); % Compute the 2D Fourier Transform
Fshifted = fftshift(F); % Shift zero-frequency component to the center

% Step 2: Create a grid of frequencies for filtering
[m, n] = size(I);
[X, Y] = meshgrid(1:n, 1:m);
centerX = ceil(n/2); % Center of the frequency domain in X
centerY = ceil(m/2); % Center of the frequency domain in Y
D = sqrt((X - centerX).^2 + (Y - centerY).^2); % Distance matrix for
frequencies

%% Step 3: Apply Butterworth Filter
D0_butterworth = 30; % Cut-off frequency
n_butterworth = 2; % Order of the Butterworth filter
butterworth_filter = 1 ./ (1 + (D ./ D0_butterworth).^(2 * n_butterworth));
% Butterworth low-pass filter

% Apply the Butterworth filter in the frequency domain
F_butterworth = Fshifted .* butterworth_filter;

% Step 4: Apply Gaussian Filter
D0_gaussian = 30; % Cut-off frequency
gaussian_filter = exp(-(D.^2) / (2 * D0_gaussian^2)); % Gaussian low-pass
filter

% Apply the Gaussian filter in the frequency domain
F_gaussian = Fshifted .* gaussian_filter;

%% Step 5: Inverse Fourier Transform to return to spatial domain
% Inverse for Butterworth Filter
F_butterworth_inverse = ifftshift(F_butterworth); % Shift back
I_butterworth = ifft2(F_butterworth_inverse); % Apply inverse FFT
I_butterworth = abs(I_butterworth); % Get the magnitude

% Inverse for Gaussian Filter
F_gaussian_inverse = ifftshift(F_gaussian); % Shift back
I_gaussian = ifft2(F_gaussian_inverse); % Apply inverse FFT
I_gaussian = abs(I_gaussian); % Get the magnitude

%% Step 6: Display the results
figure;

subplot(2, 3, 1);
imshow(I, []);

```

```

title('Original ');

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

subplot(2, 3, 3);
imshow(butterworth_filter, []);
title('Butterworth Filter');

subplot(2, 3, 4);
imshow(I_butterworth, []);
title('Butterworth Filtered');

subplot(2, 3, 5);
imshow(gaussian_filter, []);
title('Gaussian Filter');

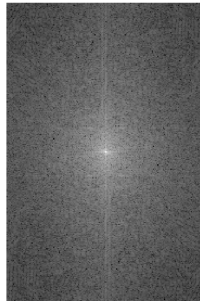
subplot(2, 3, 6);
imshow(I_gaussian, []);
title('Gaussian Filtered');

```

Original



Fourier Transform



Butterworth Filter



Butterworth Filtered



Gaussian Filter



Gaussian Filtered

