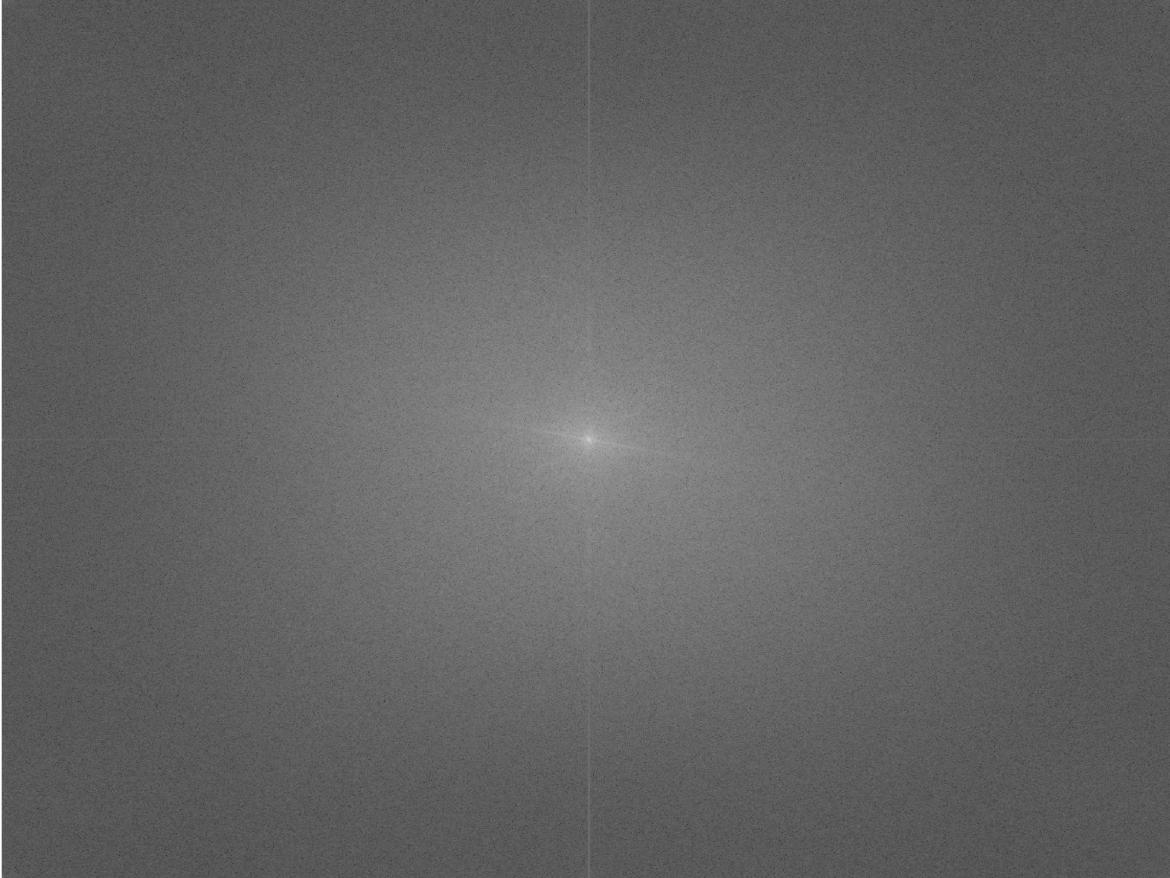


```
% Loading the image from the specified path and converting it to grayscale
image = imread('C:\Users\USER\Pictures\IMG_5745.JPG');
gray_image = rgb2gray(image);

% Performing the Fourier Transform on the grayscale image
fft_image = fftshift(fft2(double(gray_image)));

% Displaying the magnitude spectrum of the Fourier Transform
figure, imshow(log(1 + abs(fft_image)), []), title('Magnitude Spectrum of Fourier
Transform');
```

Magnitude Spectrum of Fourier Transform



```
% Function to create a Butterworth filter
function H = create_butterworth_filter(rows, cols, cutoff, order)
    % Creating meshgrid and calculating the distance from the center
    [X, Y] = meshgrid(1:cols, 1:rows);
    centerX = ceil(cols/2); centerY = ceil(rows/2);
    distance = sqrt((X - centerX).^2 + (Y - centerY).^2);
    % Defining Butterworth low-pass filter
    H = 1 ./ (1 + (distance ./ cutoff).^(2 * order));
end
```

```
% Function to create a Gaussian filter
function H = create_gaussian_filter(rows, cols, cutoff)
    % Creating meshgrid and calculating the distance from the center
    [X, Y] = meshgrid(1:cols, 1:rows);
    centerX = ceil(cols/2); centerY = ceil(rows/2);
    distance = sqrt((X - centerX).^2 + (Y - centerY).^2);
    % Defining Gaussian low-pass filter
    H = exp(-distance.^2 / (2 * cutoff^2));
end

% Applying Butterworth filter (order 2, cutoff frequency 60)
[rows, cols] = size(gray_image);
cutoff_butter = 60; order = 2;
butterworth_filter = create_butterworth_filter(rows, cols, cutoff_butter, order);
filtered_fft_butter = fft_image .* butterworth_filter;
butter_image = ifft2(ifftshift(filtered_fft_butter));
figure, imshow(uint8(abs(butter_image))), title('Image After Butterworth Filter');
```

Image After Butterworth Filter



```
% Applying Gaussian filter (cutoff frequency 60)
cutoff_gaussian = 60;
gaussian_filter = create_gaussian_filter(rows, cols, cutoff_gaussian);
filtered_fft_gaussian = fft_image .* gaussian_filter;
```

```
gaussian_image = ifft2(ifftshift(filtered_fft_gaussian));  
figure, imshow(uint8(abs(gaussian_image))), title('Image After Gaussian Filter');
```

Image After Gaussian Filter

