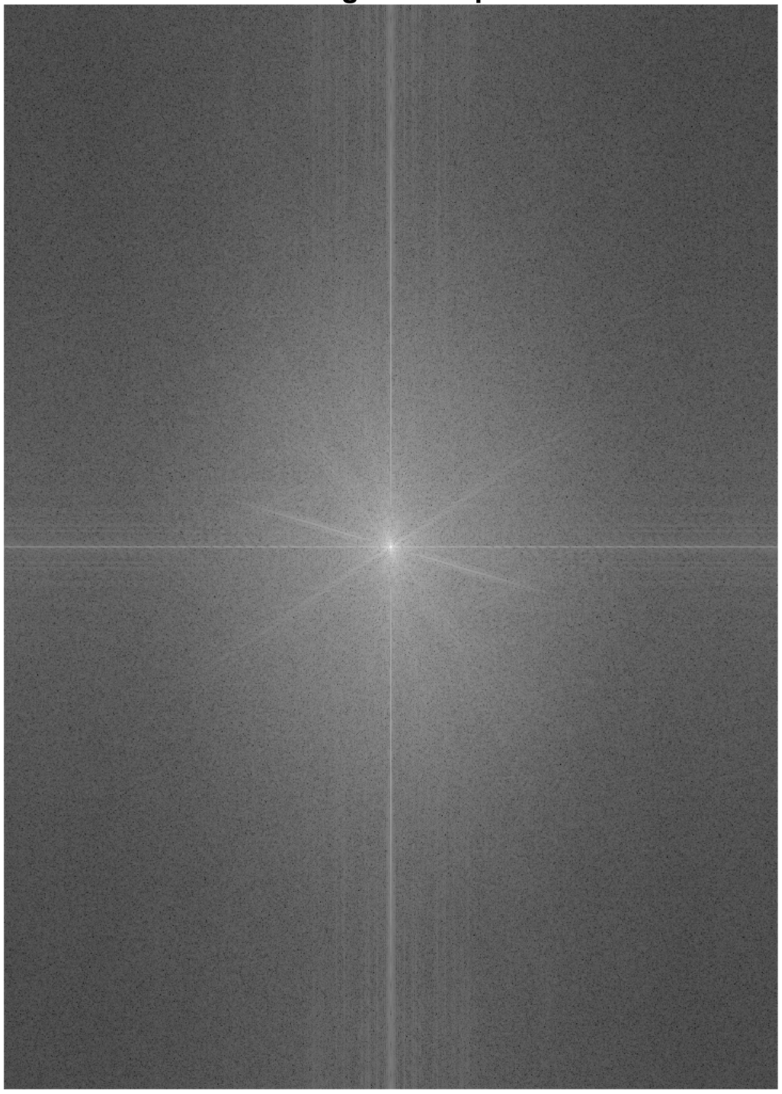
```
% Loading the image from the path and converting it to grayscale
img = imread('C:\Users\USER\Desktop\Assignment Image.JPG');
gray_img = rgb2gray(img);

% Applying Fourier Transform to the grayscale image
fft_img = fftshift(fft2(double(gray_img)));

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

Fourier Magnitude Spectrum



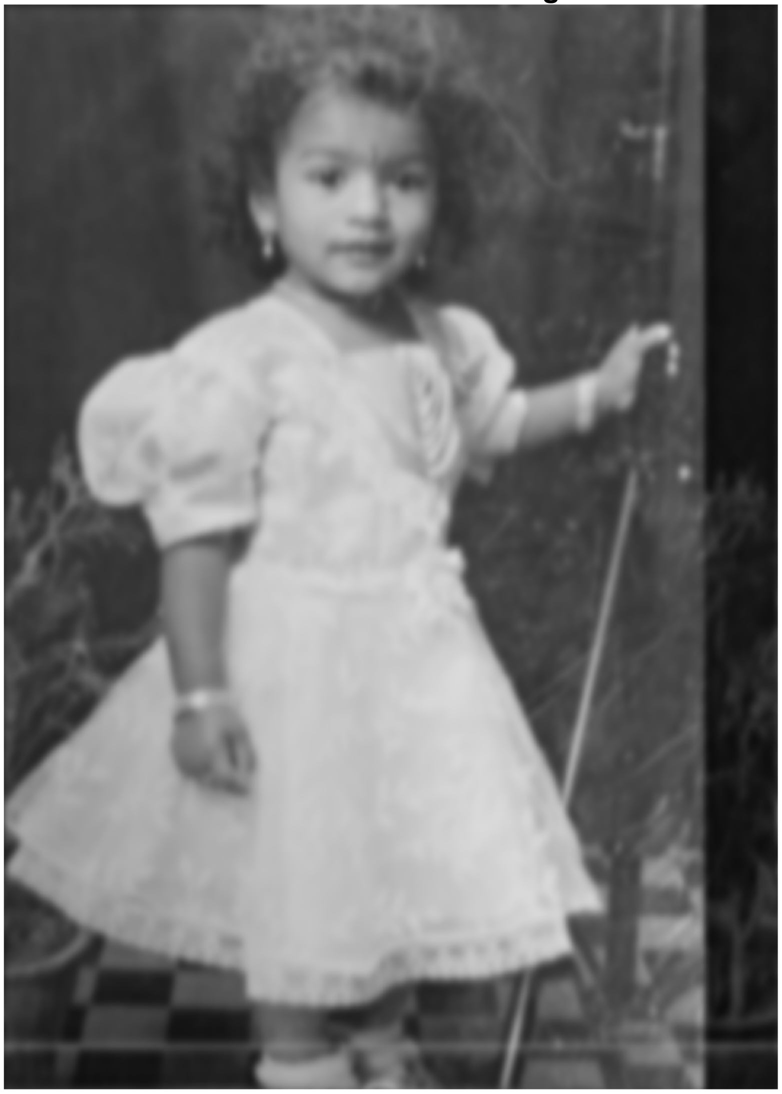
```

% Defining Butterworth filter function
function H = butterworth_lowpass(M, N, D0, order)
    % Creating a grid and finding the center of the image
    [x, y] = meshgrid(1:N, 1:M);
    cx = ceil(N/2); cy = ceil(M/2); % Center coordinates
    D = sqrt((x - cx).^2 + (y - cy).^2); % Calculating distance from center
    % Applying Butterworth low-pass filter formula
    H = 1 ./ (1 + (D ./ D0).^(2 * order));
end

% Defining Gaussian filter function
function H = gaussian_lowpass(M, N, D0)
    % Creating a grid and finding the center of the image
    [x, y] = meshgrid(1:N, 1:M);
    cx = ceil(N/2); cy = ceil(M/2); % Center coordinates
    D = sqrt((x - cx).^2 + (y - cy).^2); % Calculating distance from center
    % Applying Gaussian low-pass filter formula
    H = exp(-D.^2 / (2 * D0^2));
end

% Applying Butterworth filter (using order 2 and cutoff frequency 50)
[M, N] = size(gray_img); % Getting the size of the image
D0 = 50; order = 2;
butter_filter = butterworth_lowpass(M, N, D0, order); % Generating Butterworth
filter
filtered_butter = fft_img .* butter_filter; % Multiplying the filter with the
Fourier Transform
butter_img = ifft2(ifftshift(filtered_butter)); % Inverse Fourier Transform to get
the image back
figure, imshow(uint8(abs(butter_img))), title('Butterworth Filtered Image');
```

Butterworth Filtered Image



```
% Applying Gaussian filter (using cutoff frequency 50)
gaussian_filter = gaussian_lowpass(M, N, D0); % Generating Gaussian filter
filtered_gaussian = fft_img .* gaussian_filter; % Multiplying the filter with the
Fourier Transform
gaussian_img = ifft2(ifftshift(filtered_gaussian)); % Inverse Fourier Transform to
get the image back
figure, imshow(uint8(abs(gaussian_img))), title('Gaussian Filtered Image');
```

Gaussian Filtered Image