

```

% Create a test image
img_size = 256;
[X, Y] = meshgrid(1:img_size, 1:img_size);
center = img_size/2;
radius = img_size/4;

% Create a circle with some texture
img = double(sqrt((X-center).^2 + (Y-center).^2) <= radius);
texture = sin(X/5) .* cos(Y/5);
img = img + 0.2 * texture + 0.1 * randn(img_size);
img = mat2gray(img); % Normalize to [0, 1]

% Apply Fourier Transform
F = fftshift(fft2(img));
F_mag = log(1 + abs(F));

% Create frequency domain coordinates
[u, v] = meshgrid(linspace(-0.5, 0.5, img_size), linspace(-0.5, 0.5,
img_size));
D = sqrt(u.^2 + v.^2);

% Butterworth lowpass filter
n = 2; % Order of Butterworth filter
D0 = 0.1; % Cutoff frequency
H_butterworth = 1 ./ (1 + (D./D0).^(2*n));
F_butterworth = F .* H_butterworth;
img_butterworth = real(ifft2(ifftshift(F_butterworth)));

% Gaussian lowpass filter
sigma = 0.05;
H_gaussian = exp(-(D.^2) / (2*sigma^2));
F_gaussian = F .* H_gaussian;
img_gaussian = real(ifft2(ifftshift(F_gaussian)));

% Display results
figure;

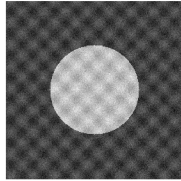
% Original image and its Fourier Transform
subplot(3,2,1), imshow(img), title('Original Image');
subplot(3,2,2), imshow(F_mag, []), title('Fourier Transform Magnitude');

% Butterworth filter
subplot(3,2,3), imshow(H_butterworth, []), title('Butterworth Filter');
subplot(3,2,4), imshow(img_butterworth, []), title('Butterworth Filtered
Image');

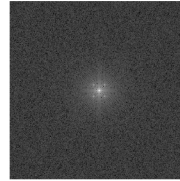
% Gaussian filter
subplot(3,2,5), imshow(H_gaussian, []), title('Gaussian Filter');
subplot(3,2,6), imshow(img_gaussian, []), title('Gaussian Filtered Image');

```

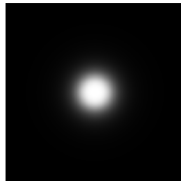
Original Image



Fourier Transform Magnitude



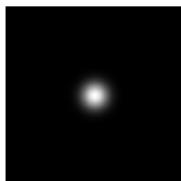
Butterworth Filter



Butterworth Filtered Image



Gaussian Filter



Gaussian Filtered Image



```
% Calculate and display PSNR and SSIM for both filters
psnr_butterworth = psnr(img_butterworth, img);
ssim_butterworth = ssim(img_butterworth, img);
psnr_gaussian = psnr(img_gaussian, img);
ssim_gaussian = ssim(img_gaussian, img);
```

```
fprintf('Butterworth Filter:\n');
```

Butterworth Filter:

```
fprintf(' PSNR: %.2f dB\n', psnr_butterworth);
```

PSNR: 25.60 dB

```
fprintf(' SSIM: %.4f\n', ssim_butterworth);
```

SSIM: 0.4580

```
fprintf('Gaussian Filter:\n');
```

Gaussian Filter:

```
fprintf(' PSNR: %.2f dB\n', psnr_gaussian);
```

PSNR: 24.48 dB

```
fprintf(' SSIM: %.4f\n', ssim_gaussian);
```

SSIM: 0.3912