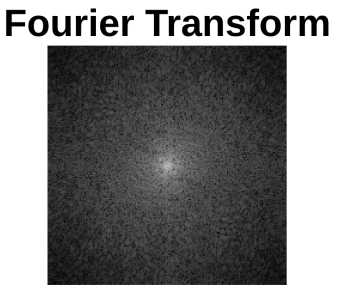
```
%Homework 2 - Q4
im = imread('tint1.jpg');
im_gray = rgb2gray(im);
im_gray = im2double(im_gray);

% Fourier Transform on the image
F = fft2(im_gray);
F_shifted = fftshift(F);
```

```
% Display the magnitude of the Fourier Transform
figure;
subplot(2, 3, 1);
imshow(log(1 + abs(F_shifted)), []);  % Display magnitude of the Fourier
Transform
title('Fourier Transform');
```




```
% Create Butterworth and Gaussian filters
[rows, cols] = size(im_gray);
[X, Y] = meshgrid(1:cols, 1:rows);
centerX = ceil(cols / 2);
centerY = ceil(rows / 2);
```

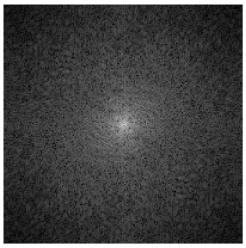
```
D = sqrt((X - centerX).^2 + (Y - centerY).^2);  % Distance from the center
```

```
% Butterworth Filter (Low-pass)
D0 = 50;  % Cutoff frequency
n = 2;    % Filter order
butterworth_filter = 1 ./ (1 + (D ./ D0).^(2 * n));

% Butterworth Filter in the frequency domain
F_butterworth = F_shifted .* butterworth_filter;
```

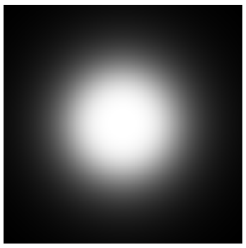
```
% Inverse Fourier Transform to get the filtered image
butterworth_filtered_image = real(ifft2(ifftshift(F_butterworth)));
```

```
% Display Butterworth filter
subplot(2, 3, 2);
imshow(butterworth_filter, []);
title('Butterworth');
```

Fourier Transform

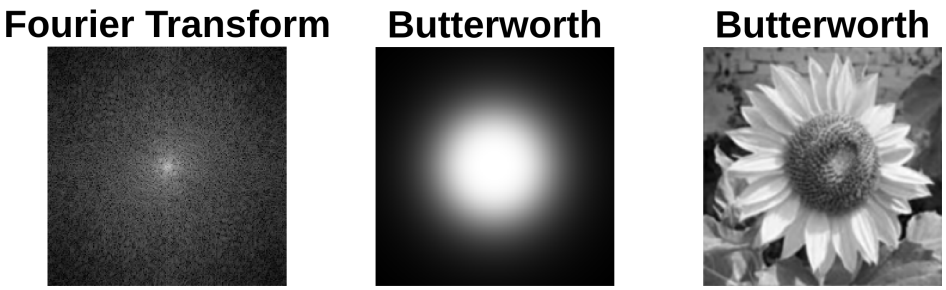


Butterworth



```
subplot(2, 3, 3);
```

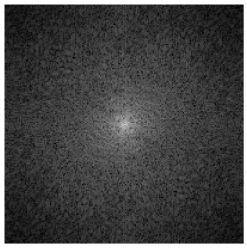
```
imshow(butterworth_filtered_image, []);
title('Butterworth');
```



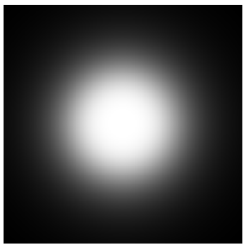

```
% Gaussian Filter (Low-pass)
D0 = 50;  % Cutoff frequency
gaussian_filter = exp(-(D.^2) / (2 * (D0^2)));

% Apply Gaussian Filter in the frequency domain
F_gaussian = F_shifted .* gaussian_filter;

% Inverse Fourier Transform to get the filtered image
gaussian_filtered_image = real(ifft2(ifftshift(F_gaussian)));
```

```
% Display Gaussian filter
subplot(2, 3, 4);
imshow(gaussian_filter, []);
title('Gaussian Filter');

subplot(2, 3, 5);
imshow(gaussian_filtered_image, []);
```

Fourier Transform



Butterworth



Butterworth

Gaussian Filter