Documented code

Q1a

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

% Image with uniform noise
mean = 40
sd = 20

I_n = imnoise(I_o,'speckle',sd^2)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
r0 = input('Enter the value of r_0 :')
for i = 1:R
    for j = 1:C
        if sqrt(i^2+j^2)<=r0
            H(i,j) =1;
        else
            H(i,j) =0;
        end
    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of LPF H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('low pass filtered image','fontsize',14)
```

Q1b

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

%% Image with gaussian noise
mean = 40
sd = 20
I_n = imnoise(I_o,'gaussian',40,sd^2)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
r0 = input('Enter the value of r_0 :')
for i = 1:R
    for j = 1:C
        if sqrt(i^2+j^2)<=r0
            H(i,j) =1;
        else
            H(i,j) =0;
        end
    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of LPF H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('low pass filtered image','fontsize',14)
```

Q1c

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

%% Images with salt and pper noise
I_n = imnoise(I_o,'salt & pepper',0.1)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
r0 = input('Enter the value of r_0 :')
for i = 1:R
    for j = 1:C
        if sqrt(i^2+j^2)<=r0
            H(i,j) =1;
        else
            H(i,j) =0;
        end
    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of LPF H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('low pass filtered image','fontsize',14)
```

Q2a

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

% Image with uniform noise
mean = 40
sd = 20

I_n = imnoise(I_o,'speckle',sd^2)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
r0 = input('Enter the value of r_0 :')

% Order of butter worth filter
n = 2;
for i = 1:R
    for j = 1:C
         H(i,j) = 1/(1+((sqrt(i^2+j^2))/(r0))^(2*n));

    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of Butterworth filter H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('Buttherworth filtered image','fontsize',14)
```

Q2b

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

%% Image with gaussian noise
mean = 40
sd = 20
I_n = imnoise(I_o,'gaussian',40,sd^2)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
r0 = input('Enter the value of r_0 :')

% Order of butter worth filter
n = 2;
for i = 1:R
    for j = 1:C
         H(i,j) = 1/(1+((sqrt(i^2+j^2))/(r0))^(2*n));

    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of Butterworth filter H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('Buttherworth filtered image','fontsize',14)
```

Q2c

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

%% Images with salt and pper noise
I_n = imnoise(I_o,'salt & pepper',0.1)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
r0 = input('Enter the value of r_0 :')

% Order of butter worth filter
n = 2;
for i = 1:R
    for j = 1:C
         H(i,j) = 1/(1+((sqrt(i^2+j^2))/(r0))^(2*n));

    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of Butterworth filter H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('Buttherworth filtered image','fontsize',14)
```

Q3a

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

% Image with uniform noise
mean = 40
sd = 20

I_n = imnoise(I_o,'speckle',sd^2)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
sigma = input('Enter the value of sigma :')

% Order of butter worth filter
n = 2;
for i = 1:R
    for j = 1:C
         H(i,j) = exp(-sqrt(i^2+j^2)/(2*sigma^2));

    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of Gaussian filter H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('Buttherworth filtered image','fontsize',14)
```

Q3b

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

%% Image with gaussian noise
mean = 40
sd = 20
I_n = imnoise(I_o,'gaussian',40,sd^2)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
sigma = input('Enter the value of sigma :')

% Order of butter worth filter
n = 2;
for i = 1:R
    for j = 1:C
         H(i,j) = exp(-sqrt(i^2+j^2)/(2*sigma^2));

    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of Gaussian filter H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('Buttherworth filtered image','fontsize',14)
```

Q3c

```
clear all
close all
clc

I_o = rgb2gray(imread('cm.jpeg'))
figure
imshow(I_o);colormap gray
title('Original image of Cameraman','fontsize',14)

% Images with salt and pper noise
I_n = imnoise(I_o,'salt & pepper',0.1)

figure
imshow(I_n);colormap gray
title('Noisy image of Cameraman','fontsize',14)

% Find the fft of image and scale it
A = fft2(double(I_n));
A_s=fftshift(A);

% the size of A in R row an C columns
[R C]=size(A); % image size

% Design the filter with the given specifications
sigma = input('Enter the value of sigma :')

% Order of butter worth filter
n = 2;
for i = 1:R
    for j = 1:C
         H(i,j) = exp(-sqrt(i^2+j^2)/(2*sigma^2));

    end
end

figure
imshow(H,[0 1])
title('Magnitude Spectra of Gaussian filter H','fontsize',14)

J=A_s.*H;
J1=ifftshift(J);
B1=ifft2(J1);

figure
imshow(abs(B1),[12 290]), colormap gray
title('Buttherworth filtered image','fontsize',14)
```

Introduction

Noise is always presents in digital images during image acquisition, coding, transmission, and processing steps. It is very difficult to remove noise from the digital images without the prior knowledge of filtering techniques. In this article, a brief overview of various noise filtering techniques. These filters can be selected by analysis of the noise behaviour. In this way, a complete and quantitative analysis of noise and their best suited filters will be presented over here.

Filtering image data is a standard process used in almost every image processing system. Filters are used for this purpose. They remove noise from images by preserving the details of the same. The choice of filter depends on the filter behaviour and type of data.

We all know that, noise is abrupt change in pixel values in an image. So when it comes to filtering of images, the first intuition that comes is to replace the value of each pixel with average of pixel around it. This process smooths the image. For this we consider two assumptions.

Assumption:

1. The true value of pixels are similar to true value of pixels nearby
2. The noise is added to each pixel independently.

Though there are many types of filters, we will consider 4 filters which are mostly used in image processing.

1. Gaussian Filter:

In **image processing**, a **Gaussian** blur (also known as **Gaussian** smoothing) is the result of blurring an **image** by a **Gaussian** function (named after mathematician and scientist Carl Friedrich **Gauss**). It is a widely used effect in graphics software, typically to reduce **image** noise and reduce detail.

2. Mean Filter:

Mean filter is a simple sliding window that replace the center value with the average of all pixel values in the window. The window or kernel is usually a square but it can be of any shape.

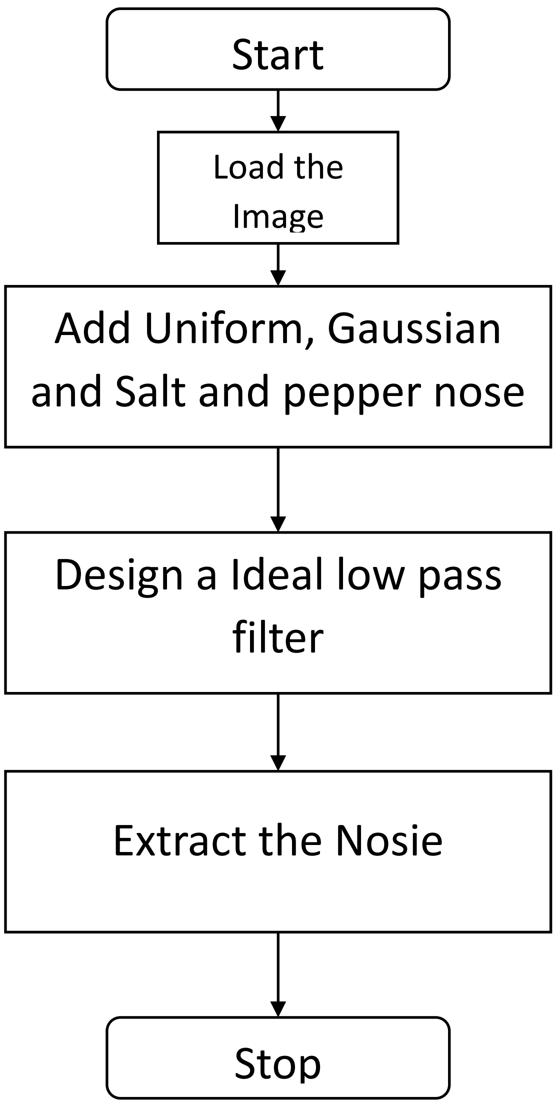
3. Median Filter:

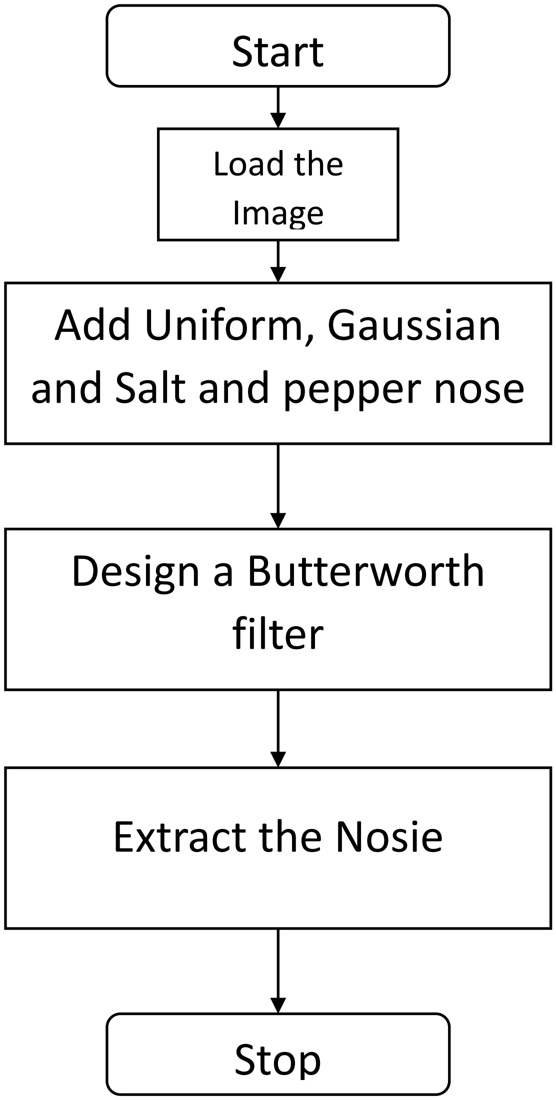
Mean filter is a simple sliding window that replace the center value with the Median of all pixel values in the window. The window or kernel is usually a square but it can be of any shape.

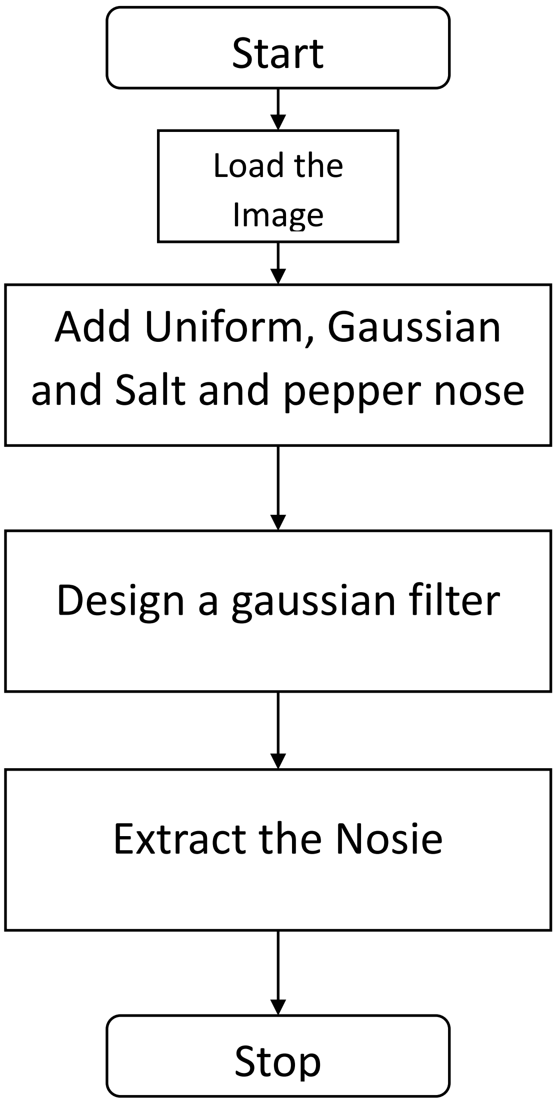
4. Bilateral Filter

Bilateral filter uses Gaussian Filter but it has one more multiplicative component which is a function of pixel intensity difference. It ensures that only pixel intensity similar to that of the central pixel is included in computing the blurred intensity value. This filter preserves edges.

Flow charts for each set of codes

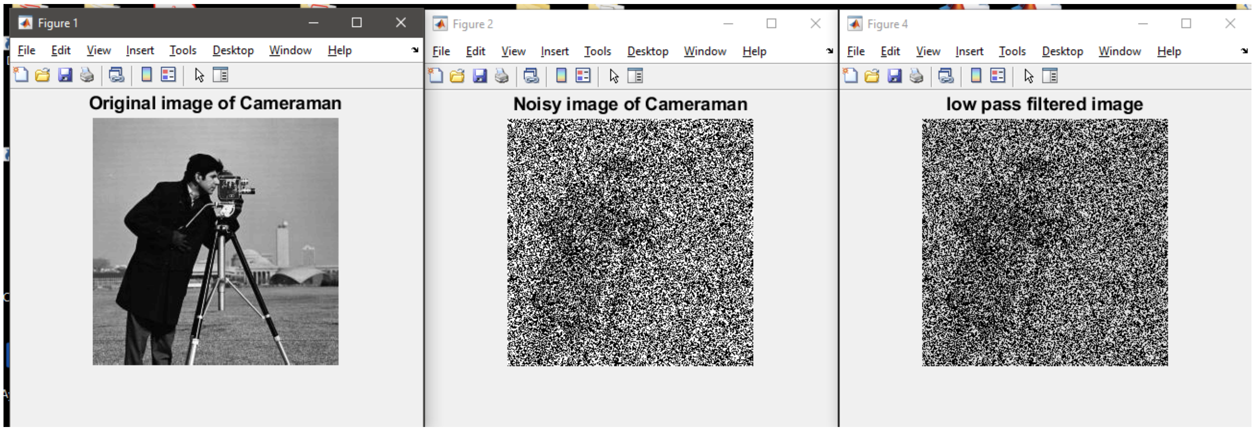



Start
Load the Image
Add Uniform, Gaussian and Salt and pepper nose
Design a Butterworth filter
Extract the Nosie
Stop

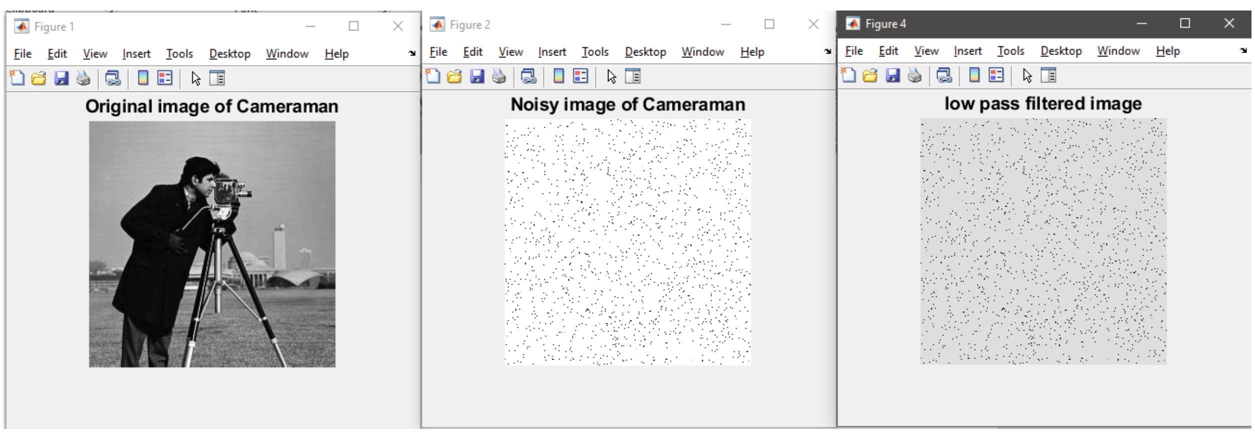

Start
Load the Image
Add Uniform, Gaussian and Salt and pepper nose
Design a gaussian filter
Extract the Nosie
Stop

All generate graphs

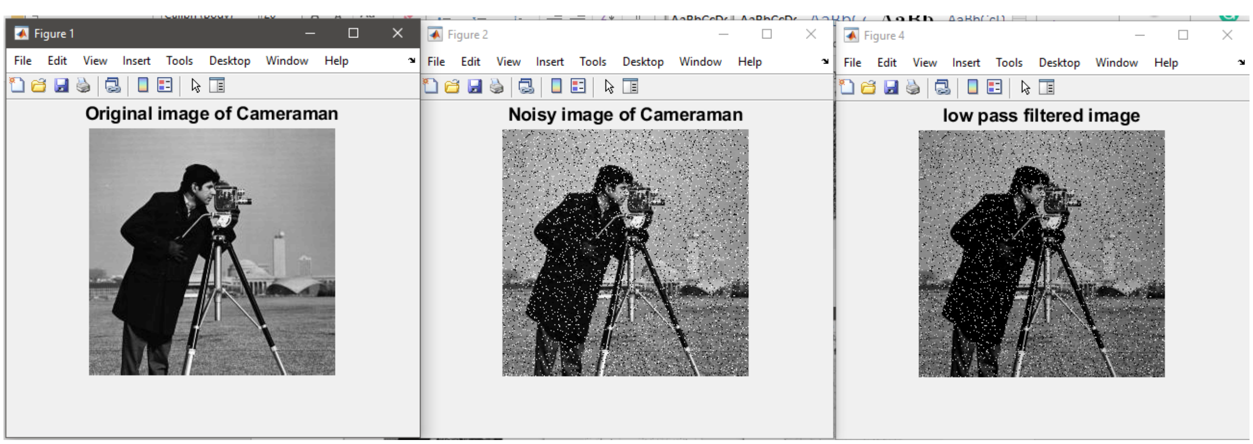
Q1a



Q1b



Q1c



Q2a

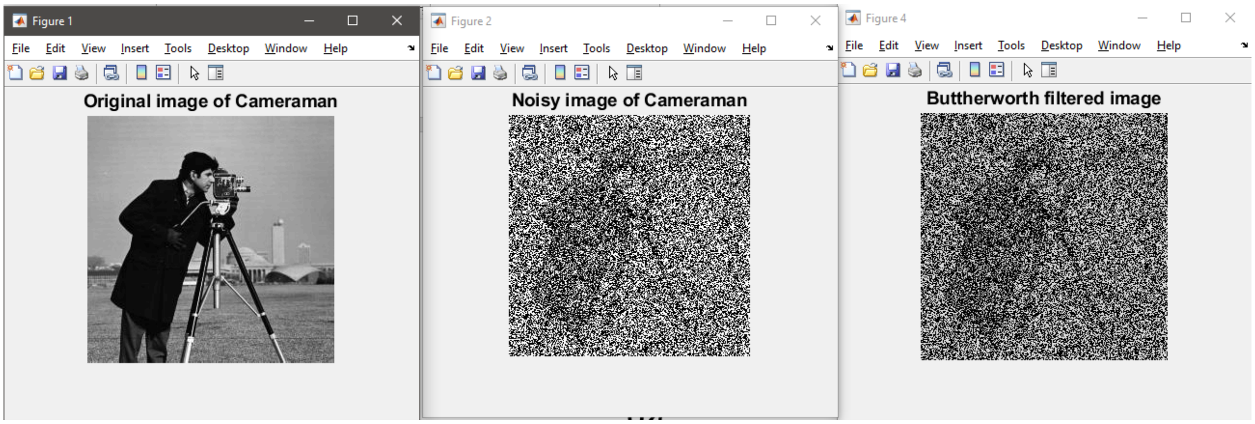


Figure 1
Original image of Cameraman
Figure 2
Noisy image of Cameraman
Figure 4
Buttherworth filtered image

Q2b

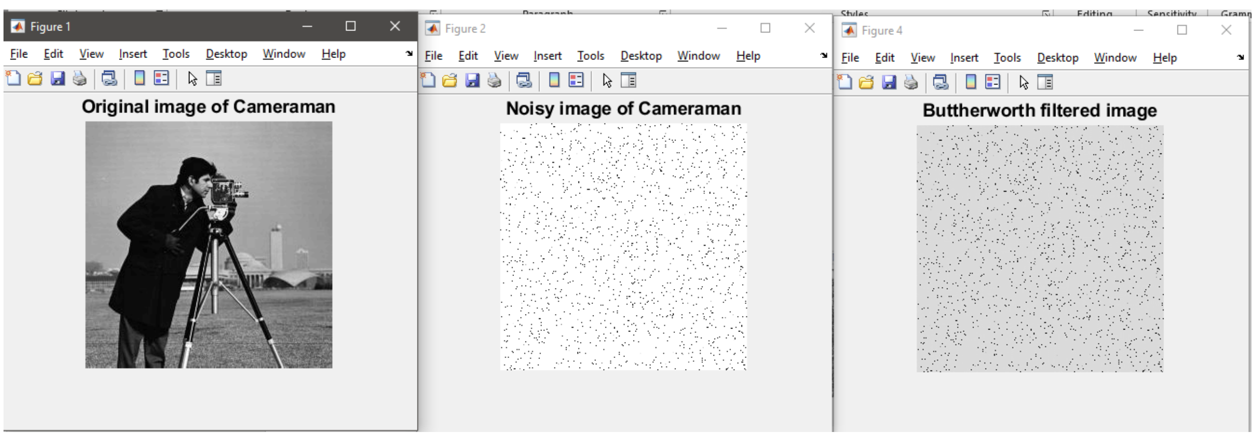


Figure 1
Original image of Cameraman
Figure 2
Noisy image of Cameraman
Figure 4
Buttherworth filtered image

Q2c

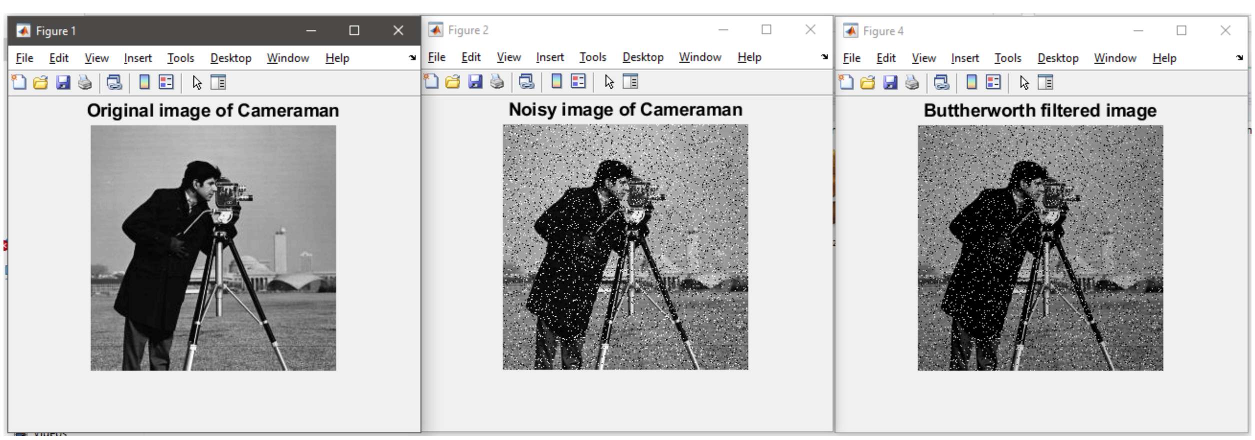


Figure 1
Original image of Cameraman
Figure 2
Noisy image of Cameraman
Figure 4
Buttherworth filtered image

Q3a

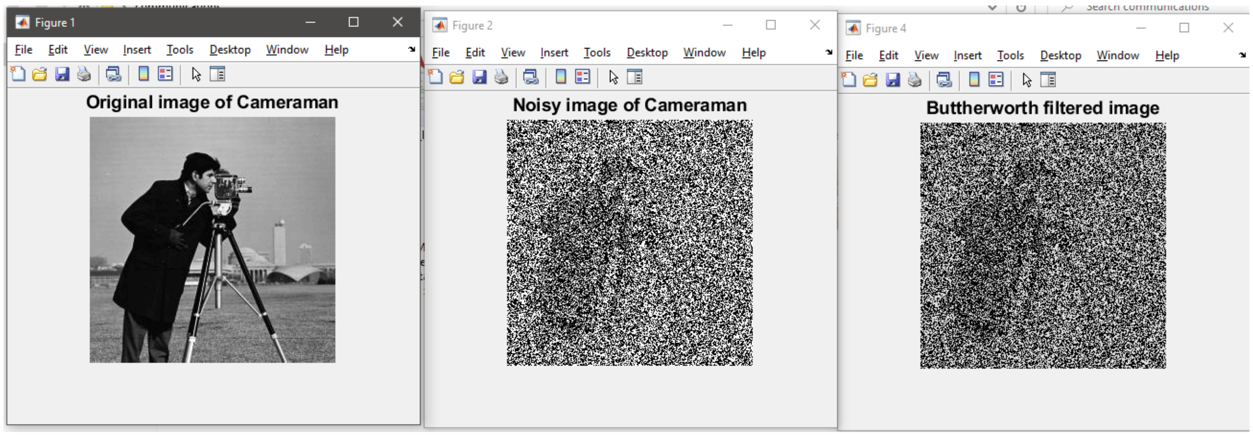


Figure 1
File Edit View Insert Tools Desktop Window Help
Original image of Cameraman
Figure 2
Noisy image of Cameraman
Figure 4
Buttherworth filtered image

Q3b

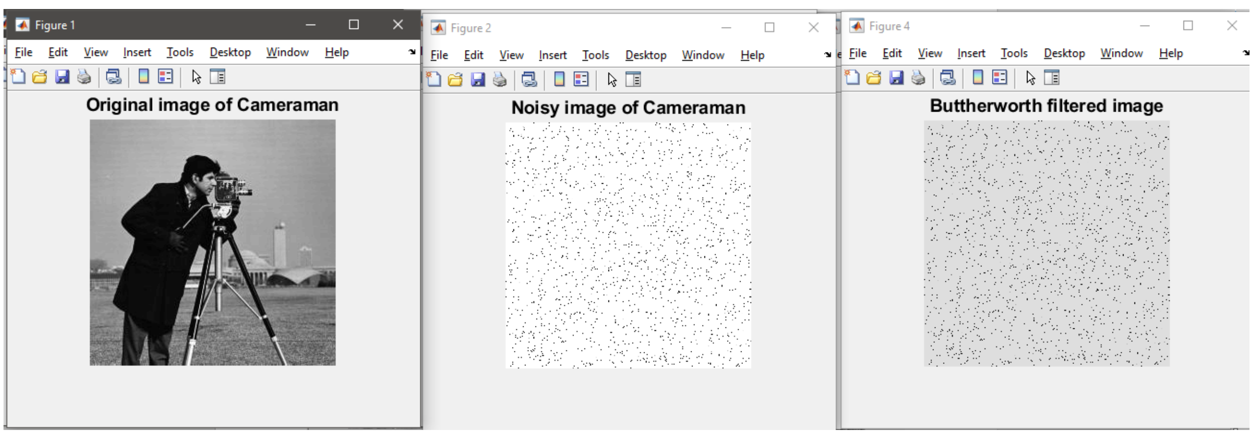


Figure 1
File Edit View Insert Tools Desktop Window Help
Original image of Cameraman
Figure 2
Noisy image of Cameraman
Figure 4
Buttherworth filtered image

Q3c

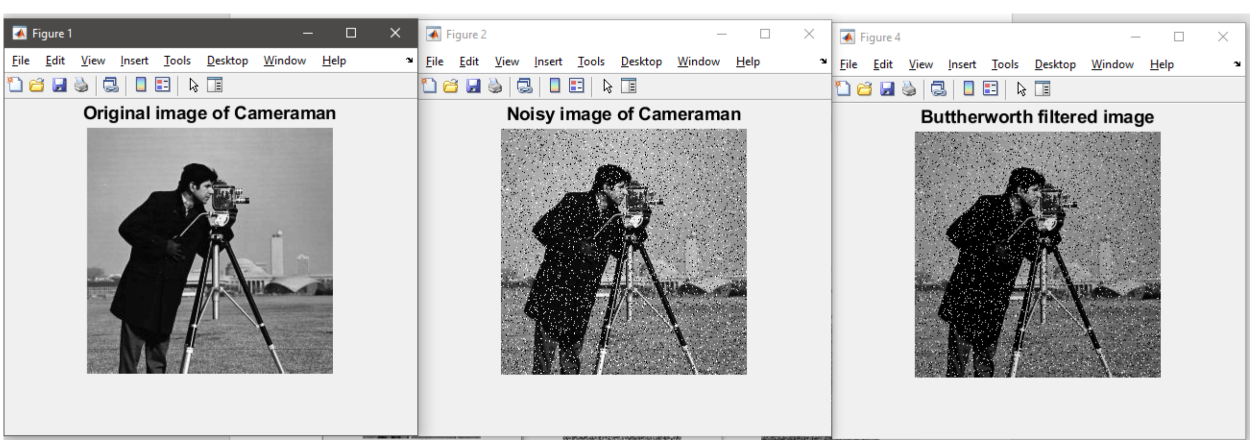


Figure 1
File Edit View Insert Tools Desktop Window Help
Original image of Cameraman
Figure 2
Noisy image of Cameraman
Figure 4
Buttherworth filtered image

Discussion and conclusions

The cameraman image is added with uniform, gaussian and salt and pepper noise. Then the noisy image is filter with the custom filter given in the task. Form the results it can observed the ideal low pass filter and gaussian filter can effectively reduce the impact of noise when compared with Butterworth filter.