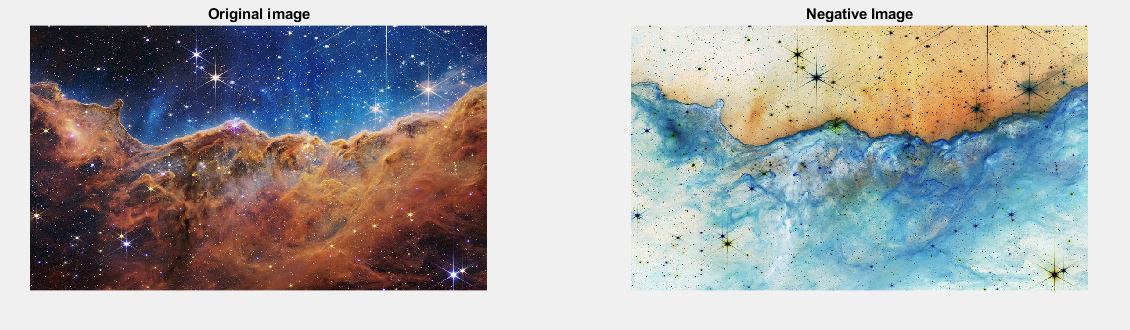
**Source Code:**

|  |
| --- |
| clc;close all;  i1=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  subplot(1,2,1);  imshow(i1);  title('Original image');  neg\_img=255-i1;  subplot(1,2,2);  imshow(neg\_img);  title('Negative Image'); |

**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;  c=input('Input constant c:');  img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST2.jpg');  subplot(1,2,1); imshow(img);  title('Input image');  img2=im2double(img);  log\_trans=c\*log(1+img2);  subplot(1,2,2); imshow(log\_trans);  title('Log transformed Image'); |

**Output:**

Please Input Constant C: 1



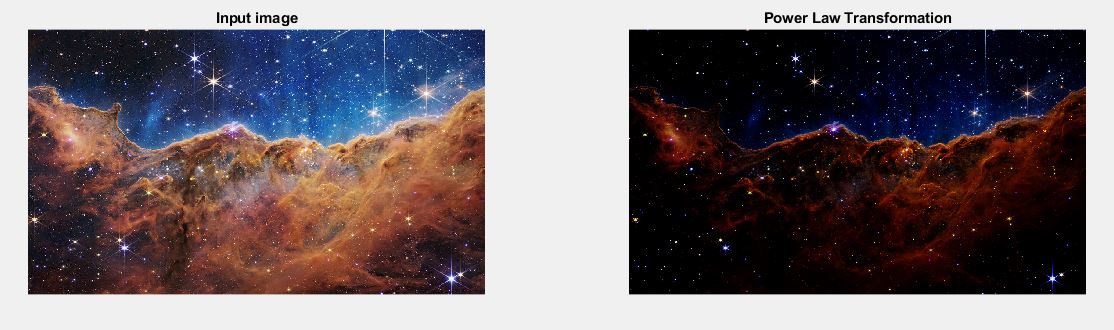
**Source Code:**

|  |
| --- |
| clc;close all;  c=input('Please Input Constant C: ');  gamma=input(' Please Input Constant Gamma: ');  img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  subplot(1,2,1);imshow(img);  title('Input image');  img2=im2double(img);  trans\_img=c\*(img2.^gamma);  subplot(1,2,2);imshow(trans\_img);  title('Power Law Transformation'); |

**Output:**

Please Input Constant C: 1

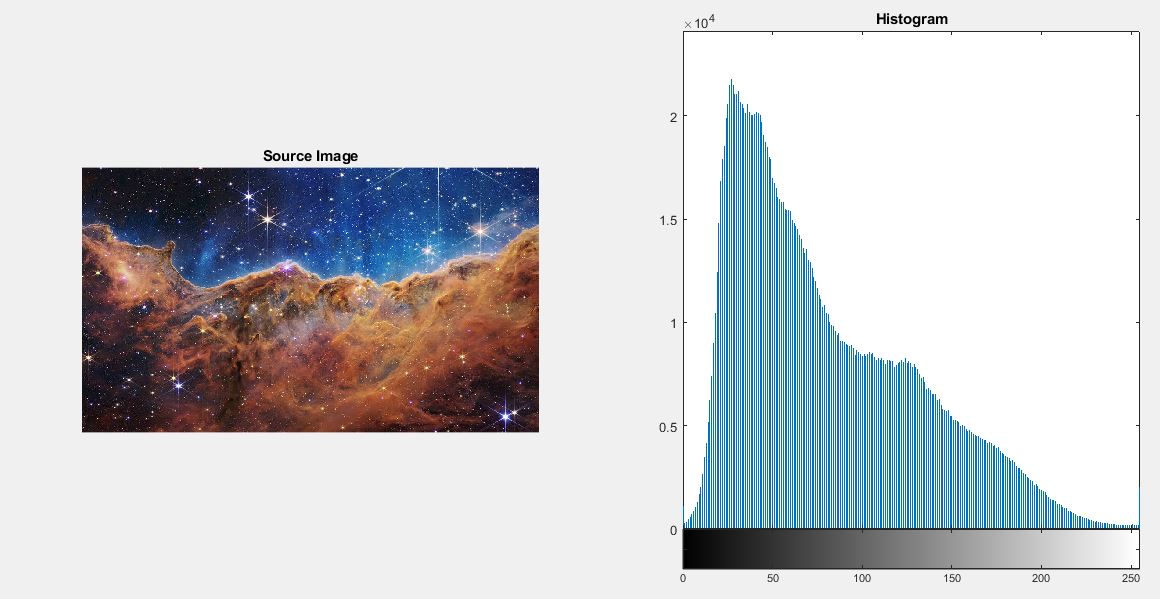
Please Input Constant Gamma: 4



**Source Code:**

|  |
| --- |
| clc;close all;  img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  subplot(1,2,1);  imshow(img);  title('Source Image')  subplot(1,2,2);  imhist(img);  title('Histogram'); |

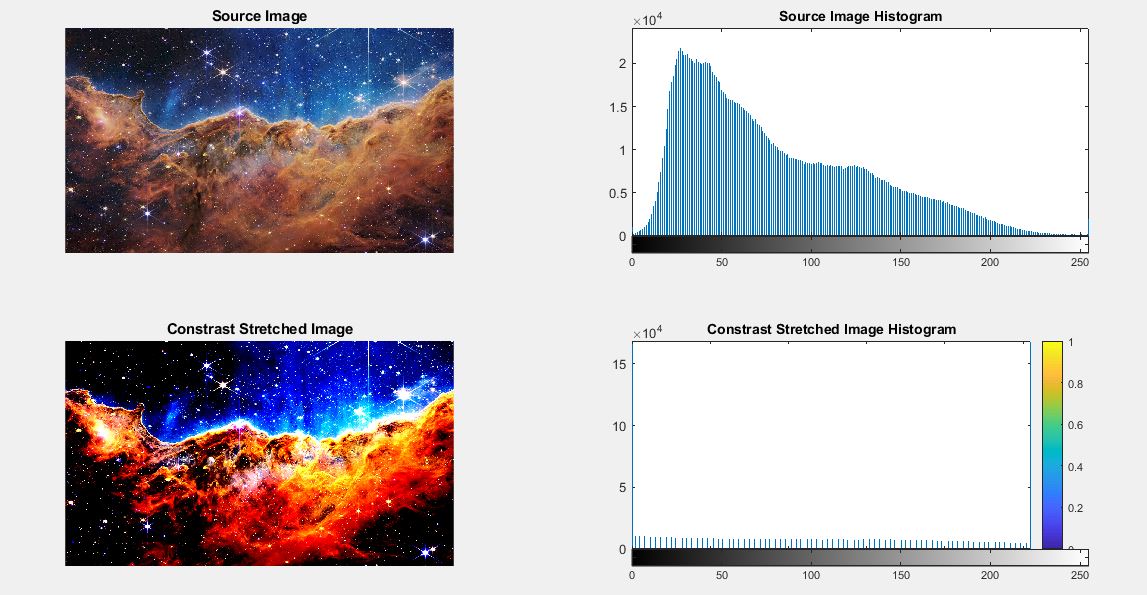
**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;  img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  subplot(2,2,1);imshow(img);  title('Source Image')  subplot(2,2,2);imhist(img);  title('Source Image Histogram')  stretched\_image=imadjust(img,[0.3, 0.6], [0.0, 1.0]);  subplot(2,2,3);imshow(stretched\_image);  title('Constrast Stretched Image');  subplot(2,2,4);imhist(stretched\_image);  title('Constrast Stretched Image Histogram'); |

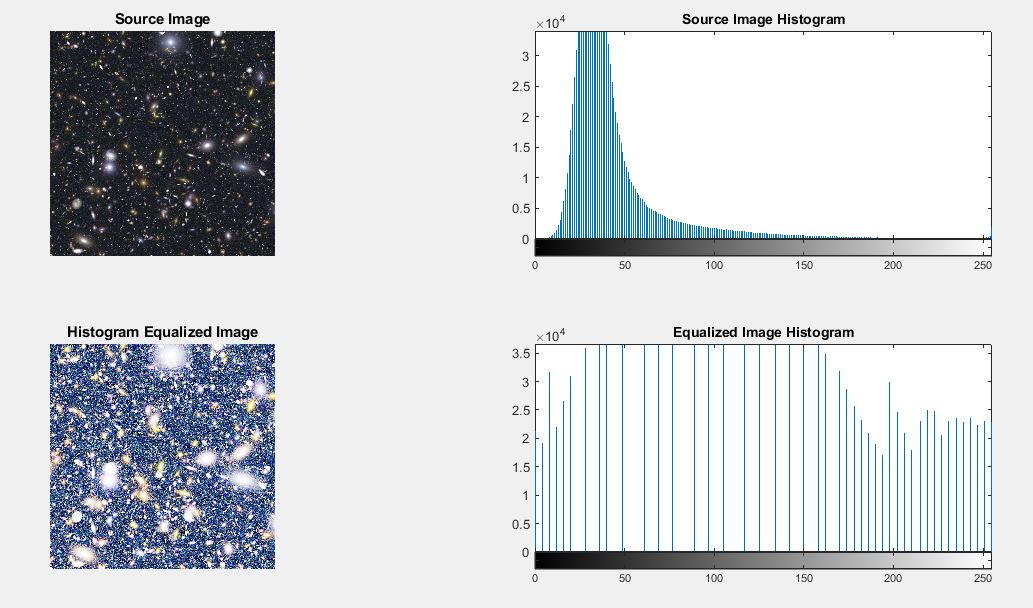
**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;  img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST2.jpg');  subplot(2,2,1);imshow(img);  title('Source Image')  subplot(2,2,2);imhist(img);  title('Source Image Histogram')  hist\_equ\_image =histeq(img);  subplot(2,2,3);imshow(hist\_equ\_image);  title('Histogram Equalized Image');  subplot(2,2,4);imhist(hist\_equ\_image);  title('Equalized Image Histogram'); |

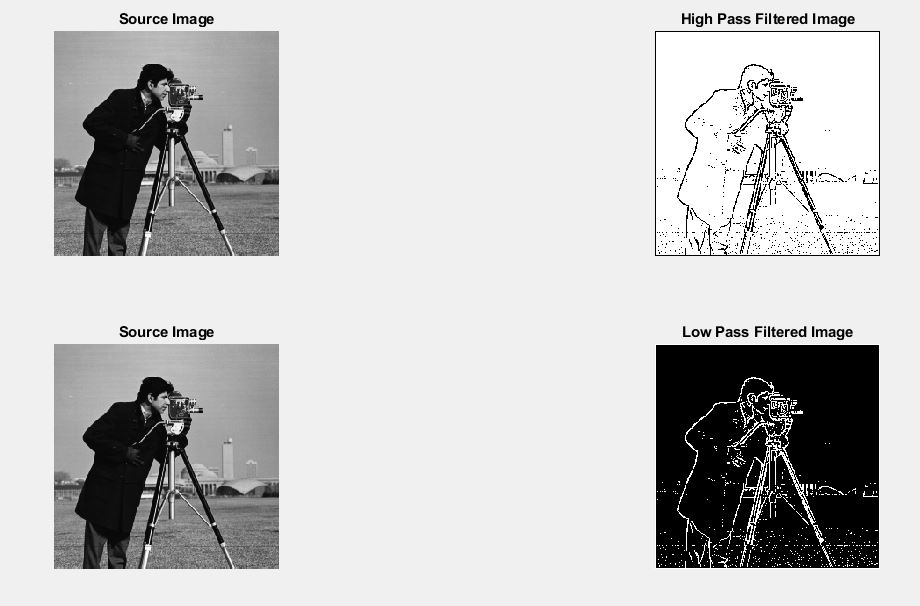
**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;  img=imread('cameraman.tif');  HighKernel = [ -1 -1 -1; -1 10 -1; -1 -1 -1 ];  HPI = conv2(HighKernel, img);  subplot(2,2,1); imshow(img); title('Source Image');  subplot(2,2,2);imshow(HPI);title('High Pass Filtered Image');  LowKernel = [ 1 1 1; 1 -10 1; 1 1 1 ];  LPI = conv2(LowKernel,img);  subplot(2,2,3);imshow(img);  title('Source Image');  subplot(2,2,4);imshow(LPI);  title('Low Pass Filtered Image'); |

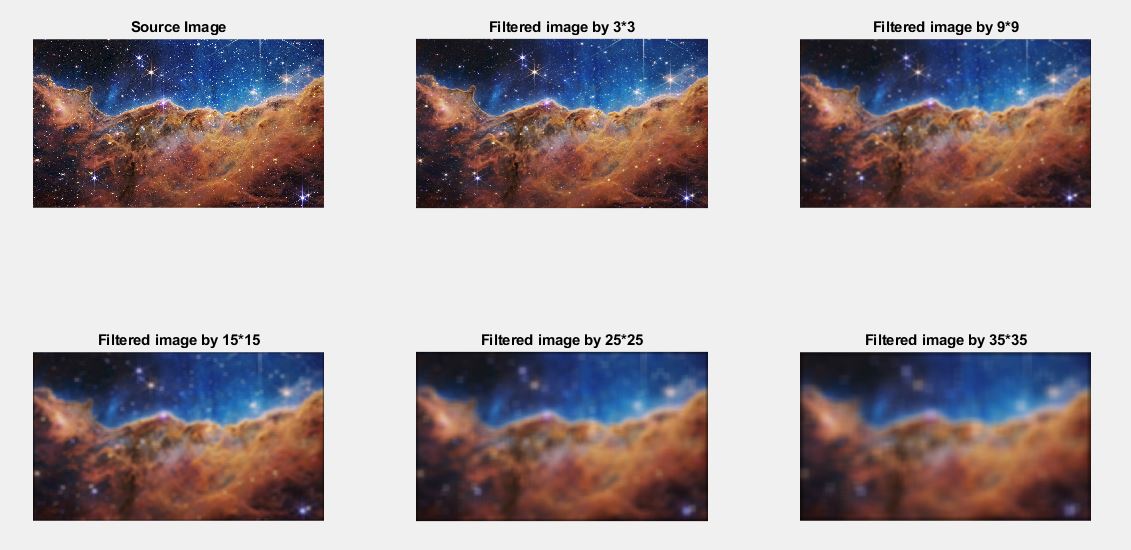
**Output:**



**Source Code:**

|  |
| --- |
| I=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  subplot(231), imshow(I), title('Source Image');  M3=fspecial('average', 3);M9=fspecial('average', 9);  M15=fspecial('average', 15);M25=fspecial('average', 25);  M35=fspecial('average', 35);  J3=imfilter(I, M3);J9=imfilter(I, M9);J15=imfilter(I, M15);  J25=imfilter(I, M25);J35=imfilter(I, M35);  subplot(232), imshow(J3), title('Filtered image by 3\*3');  subplot(233), imshow(J9), title('Filtered image by 9\*9');  subplot(234), imshow(J15), title('Filtered image by 15\*15');  subplot(235), imshow(J25), title('Filtered image by 25\*25');  subplot(236), imshow(J35), title('Filtered image by 35\*35'); |

**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;  img=imread('cameraman.tif');  [r,c]=size(img);img=im2double(img);  subplot(231);imshow(img);title('Source image');  noisy\_img=imnoise(img,'Salt & Pepper');  subplot(232);imshow(noisy\_img);title('Salt & Pepper Noisy Image');  mf\_img=ordfilt2(noisy\_img,5,ones(3,3));  subplot(233);imshow(mf\_img);title('Median Filtered Image');  subplot(234);imshow(img);title('Source Image');  maxf\_img=ordfilt2(noisy\_img,9,ones(3,3));  subplot(235);imshow(maxf\_img);title('Max Filtered Image');  minf\_img=ordfilt2(noisy\_img,1,ones(3,3));  subplot(236);imshow(minf\_img);title('Min Filtered Image'); |

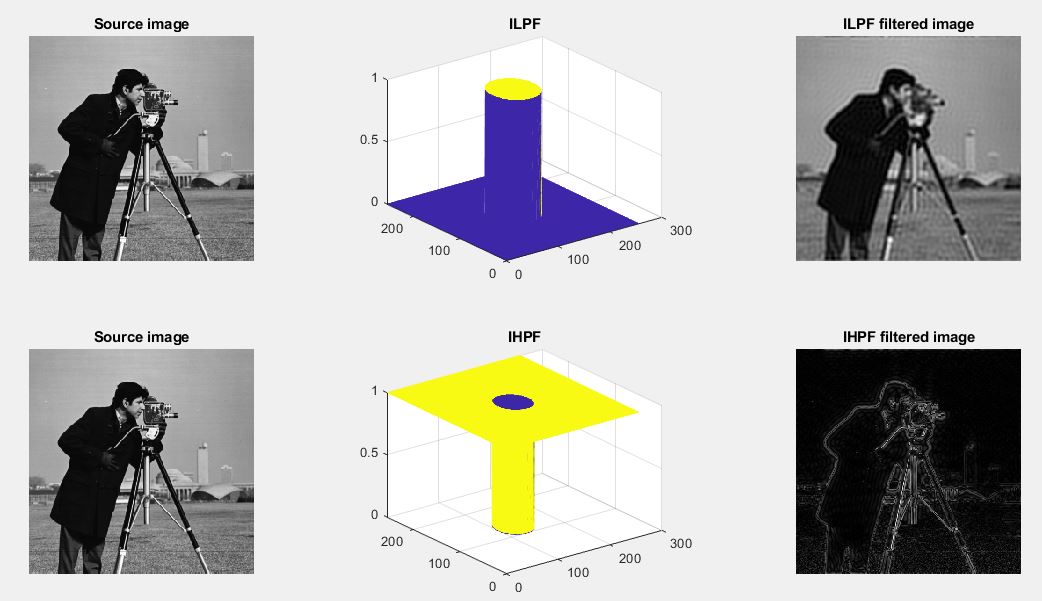
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('cameraman.tif');  [r,c]=size(img);  subplot(231), imshow(img), title('Source image')  IMG=fftshift(fft2(img));  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  D=sqrt(u.^2+v.^2);D0=40;ILPF=(D<=D0);  subplot(232),mesh(double(ILPF)),title('ILPF')  ILPF\_IMG=IMG.\*ILPF;  ilpf\_img=ifft2(ILPF\_IMG);  subplot(233),imshow(mat2gray(abs(ilpf\_img))), title('ILPF filtered image')  subplot(234),imshow(img),title('Source image')  D0=30;IHPF=(D>D0);  subplot(235),mesh(double(IHPF)), title('IHPF')  IHPF\_IMG=IMG.\*IHPF;  ihpf\_img=ifft2(IHPF\_IMG);  subplot(236),imshow(mat2gray(abs(ihpf\_img))), title('IHPF filtered image'); |

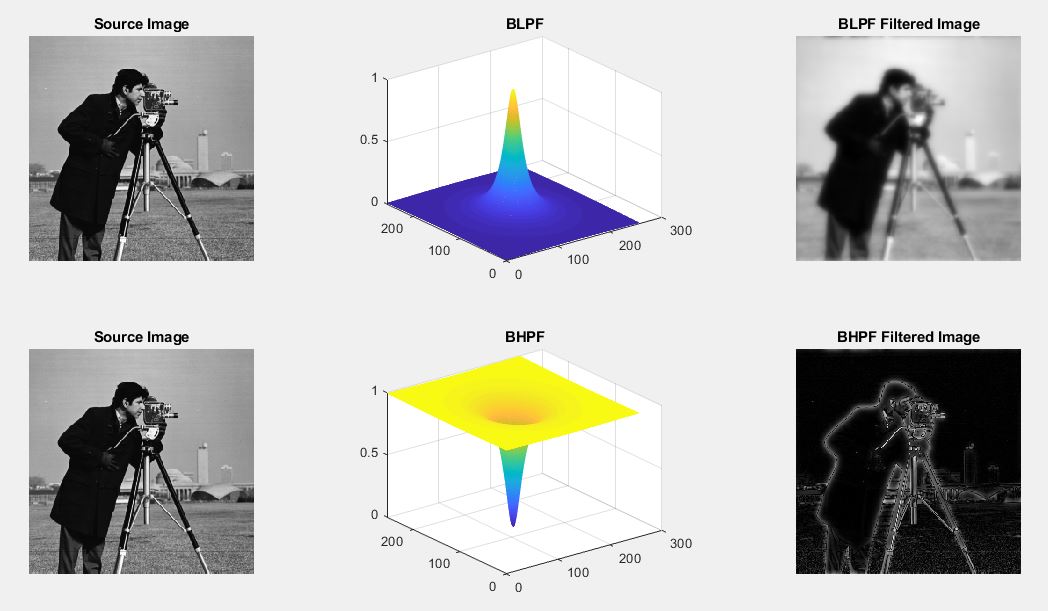
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('cameraman.tif');  [r,c]=size(img);  subplot(231), imshow(img), title('Source Image');  IMG=fftshift(fft2(img));  [u, v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  D=sqrt(u.^2+v.^2);  D0=15;n=1;  BLPF = 1./( 1.+ (D./D0).^(2\*n) );  subplot(232), mesh(BLPF), title('BLPF')  BLPF\_IMG=IMG.\*BLPF;  blpf\_img=ifft2(BLPF\_IMG);  subplot(233), imshow(mat2gray(abs(blpf\_img))), title('BLPF Filtered Image')  subplot(234), imshow(img), title('Source Image')  D0=15;n=1;  BHPF=1./( 1.+ (D0./D).^(2\*n) );  subplot(235), mesh(BHPF), title('BHPF')  BHPF\_IMG=IMG.\*BHPF;  bhpf\_img=ifft2(BHPF\_IMG);  subplot(236), imshow(mat2gray(abs(bhpf\_img))), title('BHPF Filtered Image') |

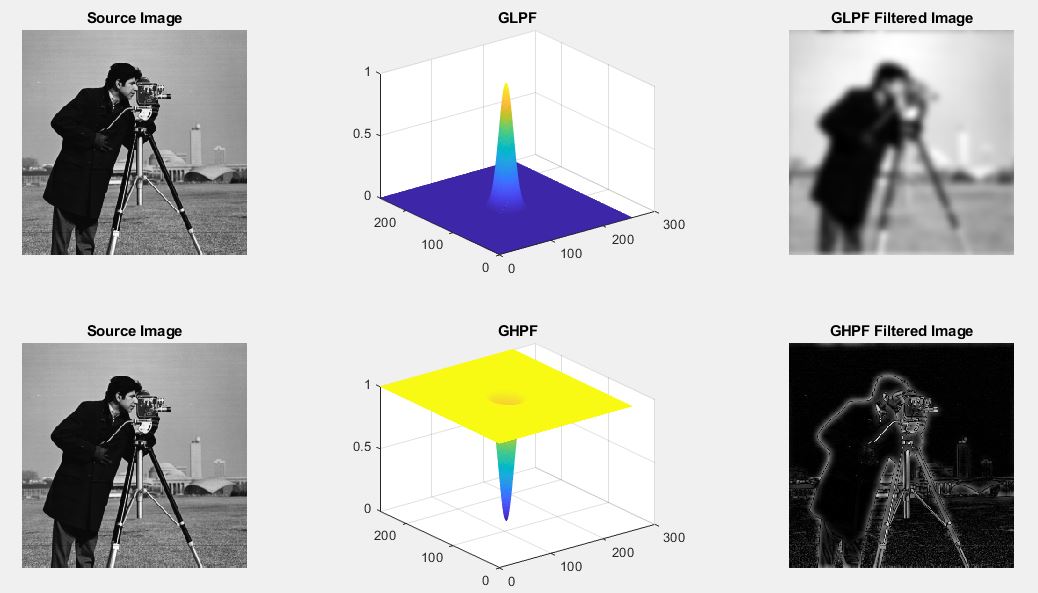
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('cameraman.tif');  [r,c]=size(img);  subplot(231), imshow(img), title('Source Image')  IMG=fftshift(fft2(img));  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  D=sqrt(u.^2+v.^2);D0=10;  GLPF = exp (-(D.^2)./(2\*D0^2)) ;  subplot(232), mesh(GLPF), title('GLPF')  GLPF\_IMG=IMG.\*GLPF;  glpf\_img=ifft2(GLPF\_IMG);  subplot(233), imshow(mat2gray(abs(glpf\_img))), title('GLPF Filtered Image')  subplot(234), imshow(img), title('Source Image')  D0=10;GHPF=1 - exp (-(D.^2)./(2\*D0^2)) ;  subplot(235), mesh(GHPF), title('GHPF')  GHPF\_IMG=IMG.\*GHPF;  ghpf\_img=ifft2(GHPF\_IMG);  subplot(236), imshow(mat2gray(abs(ghpf\_img))), title('GHPF Filtered Image'); |

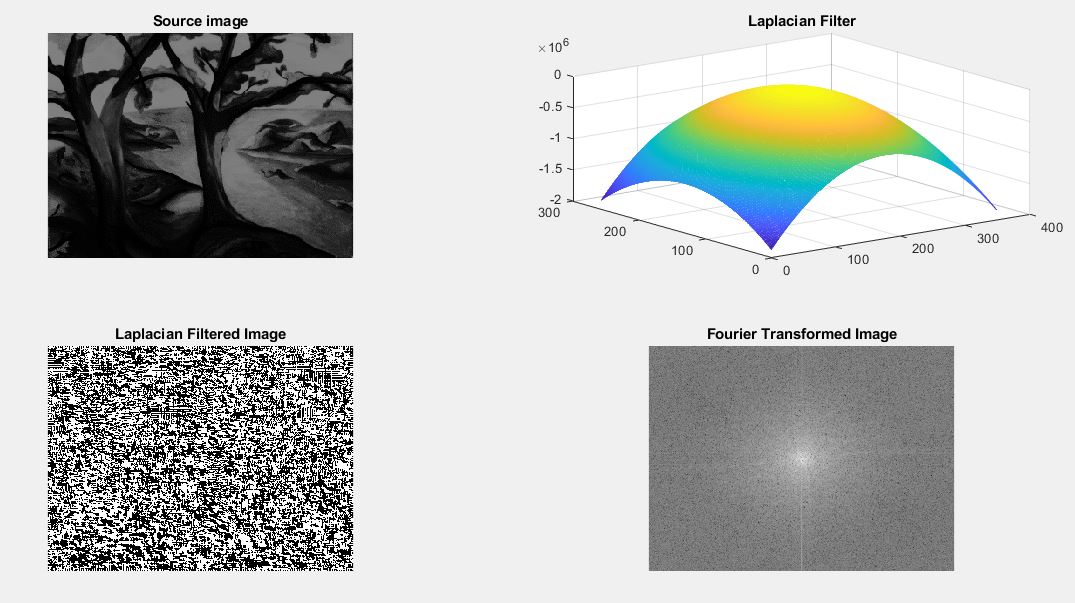
**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;  img=imread('cameraman.tif');  [r,c]=size(img);  subplot(221), imshow(img), title('Source image')  IMG=fftshift(fft2(img));  %Dispaly Fourier Transformed Image  IMG1=log(1+abs(IMG));  m=max(IMG1(:));  subplot(224), imshow(im2uint8(IMG1/m)), title('Fourier Transformed Image');  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  LF = -4\*pi^2\*(u.^2+v.^2) ;  subplot(222), mesh(LF), title('Laplacian Filter');  LF\_IMG=IMG.\*LF;  lf\_img=ifft2(LF\_IMG);  subplot(223),imshow(lf\_img),title('Laplacian Filtered Image'); |

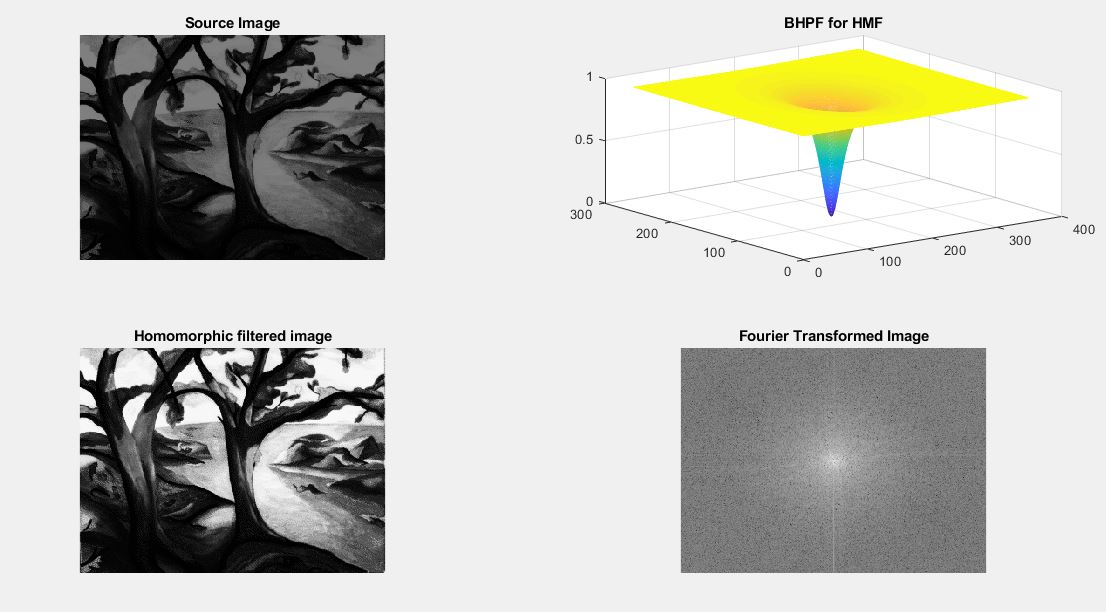
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('trees.tif');  [r,c]=size(img);  subplot(221), imshow(img), title('Source Image')  IMG=fftshift(fft2(img));  IMG1=log(1+abs(IMG));m=max(IMG1(:));  subplot(224), imshow(im2uint8(IMG1/m)), title('Fourier Transformed Image');  img=im2double(img);  IMG=fft2(log(img+0.01));  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  D=sqrt(u.^2+v.^2);  D0=15;n=1;  BHPF = 1./( 1.+ (D0./D).^(2\*n) );  subplot(222), mesh(BHPF), title('BHPF for HMF')  BHPF\_IMG=IMG.\*BHPF;  real\_img=real(ifft2(BHPF\_IMG));exp\_img=exp(real\_img);  subplot(223), imshow(mat2gray(abs(exp\_img))), title('Homomorphic filtered image'); |

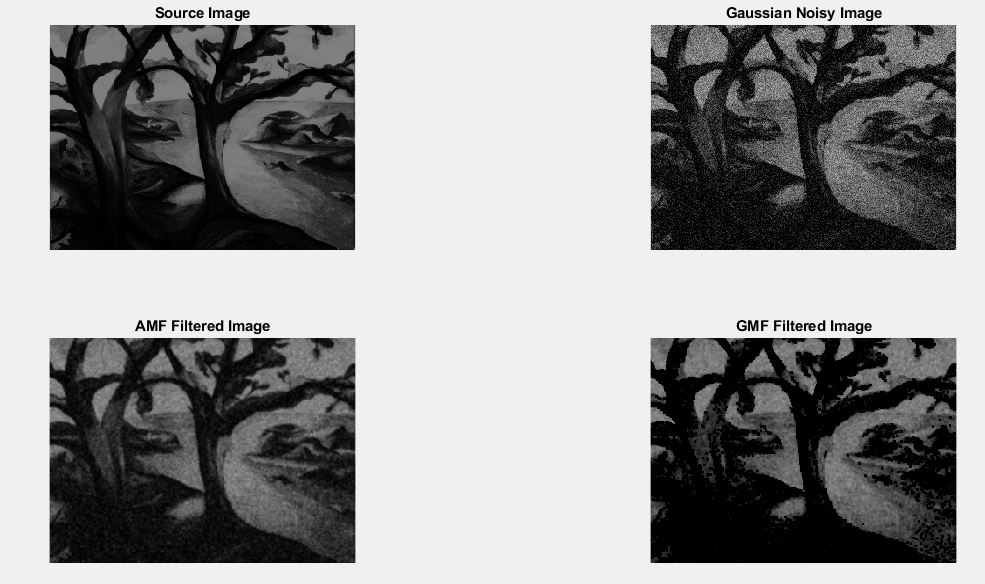
**Output:**



**Source Code:**

|  |
| --- |
| clc;close all;clear all;  img=imread('trees.tif');  [r,c]=size(img);  img=im2double(img);  subplot(221);imshow(img);title('Source Image');  noisy\_img=imnoise(img,'gaussian');  subplot(222);imshow(noisy\_img);title('Gaussian Noisy Image');  for i=1:r-2  for j=1:c-2  window = noisy\_img(i:i+2,j:j+2);  amf\_img(i,j)= mean( window(:) );  gmf\_img(i,j)= geomean( window(:) );  end  end  subplot(223);imshow(amf\_img);title('AMF Filtered Image');  subplot(224);imshow(gmf\_img);title('GMF Filtered Image'); |

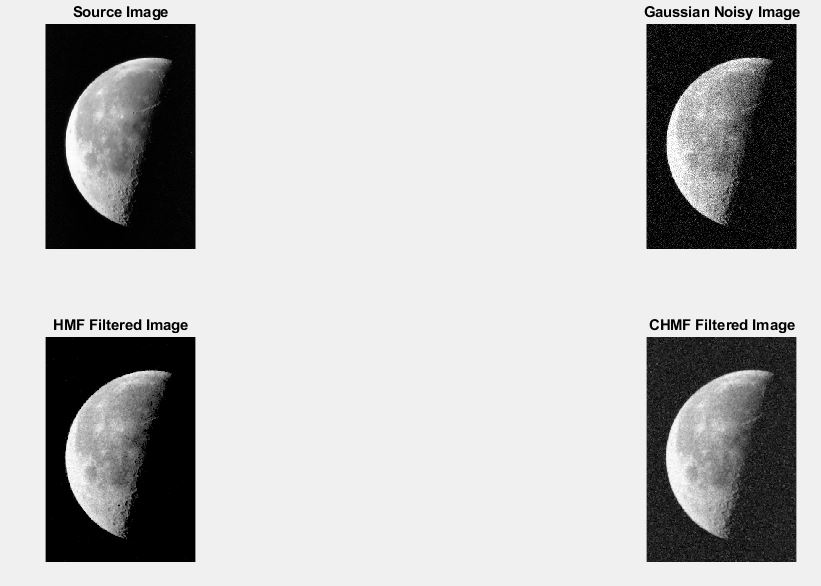
**Output:**



**Source Code:**

|  |
| --- |
| clc;clear all;close all;  img=imread('moon.tif');  [r,c]=size(img);  img=im2double(img);  subplot(221);imshow(img);title('Source Image');  noisy\_img=imnoise(img,'gaussian');  subplot(222);imshow(noisy\_img);title('Gaussian Noisy Image');  Q=1.5;  for i=1:r-2  for j=1:c-2  window = noisy\_img(i:i+2,j:j+2);  hmf\_img(i,j)= harmmean( window(:) );  chmf\_img(i,j)= sum( window(:).^(Q+1) ) ./ sum( window(:).^Q );  end  end  subplot(223);imshow(hmf\_img);title('HMF Filtered Image');  subplot(224);imshow(chmf\_img);title('CHMF Filtered Image'); |

**Output:**



**Source Code:**

|  |
| --- |
| clc;clear all;close all;  img=imread('pout.tif');  [r,c]=size(img);  img=im2double(img);  subplot(221),imshow(img),title('Source Image');  noisy\_img=imnoise(img,'Gaussian');  subplot(222),imshow(noisy\_img),title('Gaussian Noisy Image');  midf\_img= ( ordfilt2(noisy\_img,9,ones(3,3)) + ordfilt2(noisy\_img,1,ones(3,3)) )./2 ;  subplot(223),imshow(midf\_img),title('Midpoint Filtered Image');  d=25;  for i=1:r-2  for j=1:c-2  window = noisy\_img(i:i+2,j:j+2);  atmf\_img(i,j)= trimmean( window(:),d );  end  end  subplot(224),imshow(atmf\_img),title('ATMF Filtered Image'); |

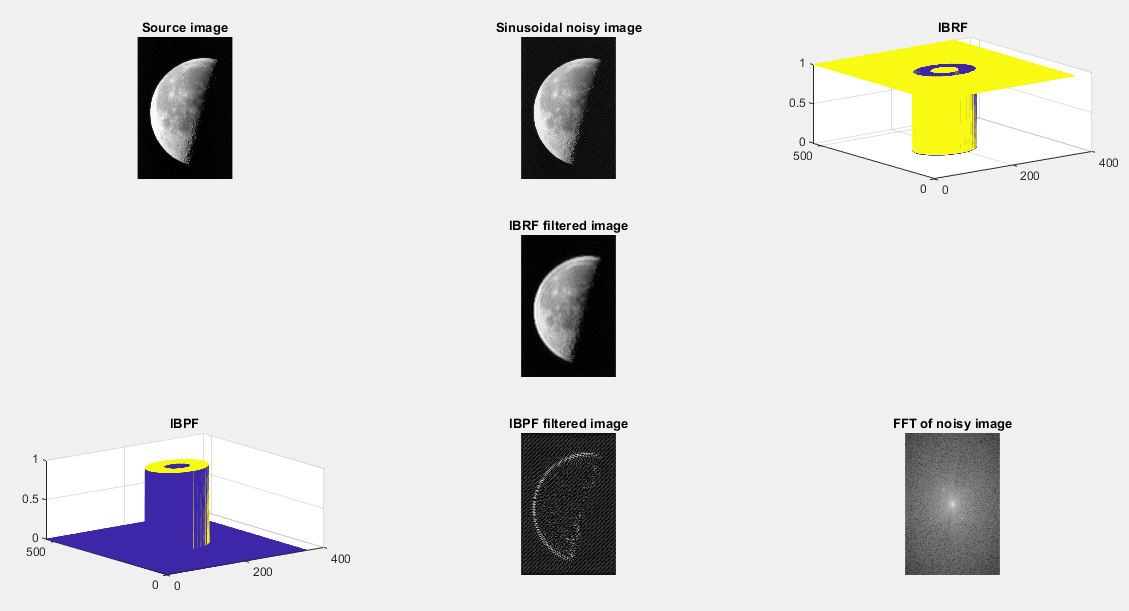
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('moon.tif');  [r,c]=size(img);  subplot(331);imshow(img);title('Source image');  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  sin\_noise= 15\*sin( 2\*pi\*1/10\*u + 2\*pi\*1/10\*v);  noisy\_img=double(img)+sin\_noise;  NOISY\_IMG=fftshift(fft2(noisy\_img));  subplot(332);imshow(noisy\_img,[]);title('Sinusoidal noisy image')  subplot(339);imshow(mat2gray(log(1+abs(NOISY\_IMG))));title('FFT of noisy image');  D=sqrt(u.^2+v.^2);  D0=50;  W=40;  IBRF= ( D<(D0-W/2) | D>(D0+W/2) );  subplot(333);mesh(IBRF);title('IBRF');  IBRF\_IMG=NOISY\_IMG.\*IBRF;  ibrf\_img=ifft2(IBRF\_IMG);  subplot(335);imshow(mat2gray(abs(ibrf\_img)));title('IBRF filtered image');  IBPF= 1 - IBRF ;  subplot(337);mesh(IBPF);title('IBPF')  IBPF\_IMG=NOISY\_IMG.\*IBPF;  ibpf\_img=ifft2(IBPF\_IMG);  subplot(338);imshow(mat2gray(abs(ibpf\_img)));title('IBPF filtered image'); |

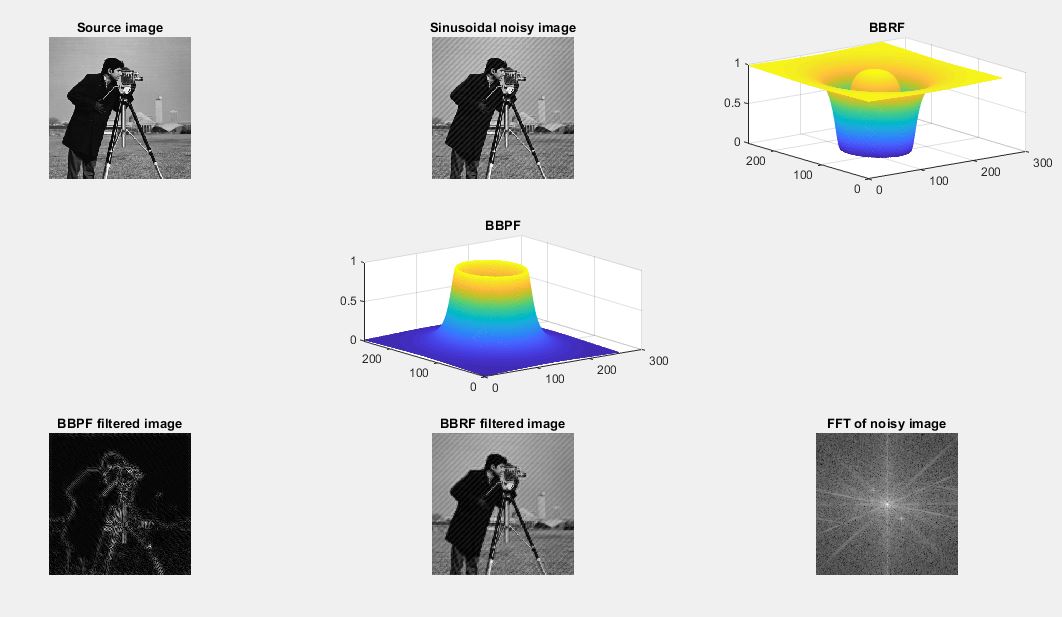
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('cameraman.tif');  [r,c]=size(img);  subplot(331);imshow(img);title('Source image');  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  sin\_noise= 15\*sin( 2\*pi\*1/10\*u + 2\*pi\*1/10\*v);  noisy\_img=double(img)+sin\_noise;  NOISY\_IMG=fftshift(fft2(noisy\_img));  subplot(332);imshow(noisy\_img,[]);title('Sinusoidal noisy image');  subplot(339);imshow(mat2gray(log(1+abs(NOISY\_IMG))));title('FFT of noisy image');  D=sqrt(u.^2+v.^2);  D0=50;n=1;W=20;  BBRF=1./( 1.+ ( (D.\*W) ./ (D.^2-D0.^2) ) .^(2\*n) );  subplot(333);mesh(BBRF);title('BBRF')  BBRF\_IMG=NOISY\_IMG.\*BBRF;  bbrf\_img=ifft2(BBRF\_IMG);  subplot(338);imshow(mat2gray(abs(bbrf\_img)));title('BBRF filtered image')  BBPF= 1 - BBRF;  subplot(335);mesh(BBPF);title('BBPF')  BBPF\_IMG=NOISY\_IMG.\*BBPF;  bbpf\_img=ifft2(BBPF\_IMG);  subplot(337);imshow(mat2gray(abs(bbpf\_img)));title('BBPF filtered image'); |

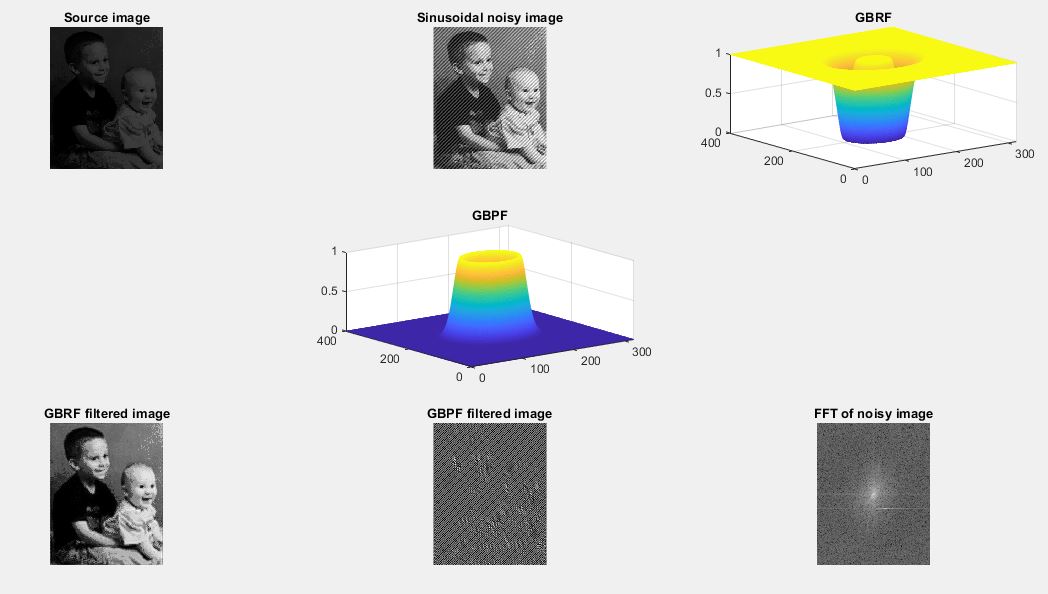
**Output:**



**Source Code:**

|  |
| --- |
| img=imread('kids.tif');  [r,c]=size(img);  subplot(331);imshow(img);title('Source image')  [u,v]=meshgrid(-floor(c/2):floor((c-1)/2),-floor(r/2):floor((r-1)/2));  sin\_noise= 15\*sin( 2\*pi\*1/10\*u + 2\*pi\*1/10\*v);  noisy\_img=double(img)+sin\_noise;  NOISY\_IMG=fftshift(fft2(noisy\_img));  subplot(332);imshow(noisy\_img,[]);title('Sinusoidal noisy image');  subplot(339);imshow(mat2gray(log(1+abs(NOISY\_IMG))));title('FFT of noisy image');  D=sqrt(u.^2+v.^2);  D0=50;W=20;  GBRF= 1 - exp ( -(1/2).\* ( ((D.^2)-(D0.^2)) ./ (D.\*W) ).^2 ) ;  subplot(333);mesh(GBRF);title('GBRF');  GBRF\_IMG=NOISY\_IMG.\*GBRF;  gbrf\_img=ifft2(GBRF\_IMG);  subplot(337);imshow(mat2gray(abs(gbrf\_img)));title('GBRF filtered image');  GBPF=1 - GBRF;  subplot(335);mesh(GBPF);title('GBPF');  GBPF\_IMG=NOISY\_IMG.\*GBPF;  gbpf\_img=ifft2(GBPF\_IMG);  subplot(338);imshow(mat2gray(abs(gbpf\_img)));title('GBPF filtered image'); |

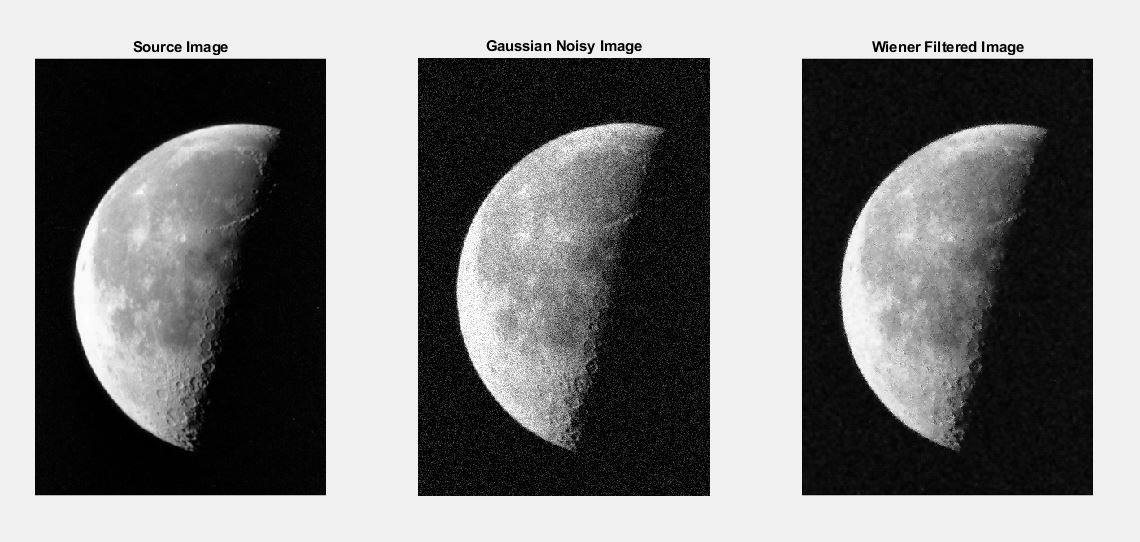
**Output:**

****

**Source Code:**

|  |
| --- |
| clc;clear all;close all  img=imread('moon.tif');  subplot(131)  imshow(img)  title('Source Image')  noisy\_img=imnoise(img,'gaussian');  subplot(132);  imshow(noisy\_img);  title('Gaussian Noisy Image');  wiener\_img=wiener2(noisy\_img,[5 5]);  subplot(133)  imshow(wiener\_img);  title('Wiener Filtered Image'); |

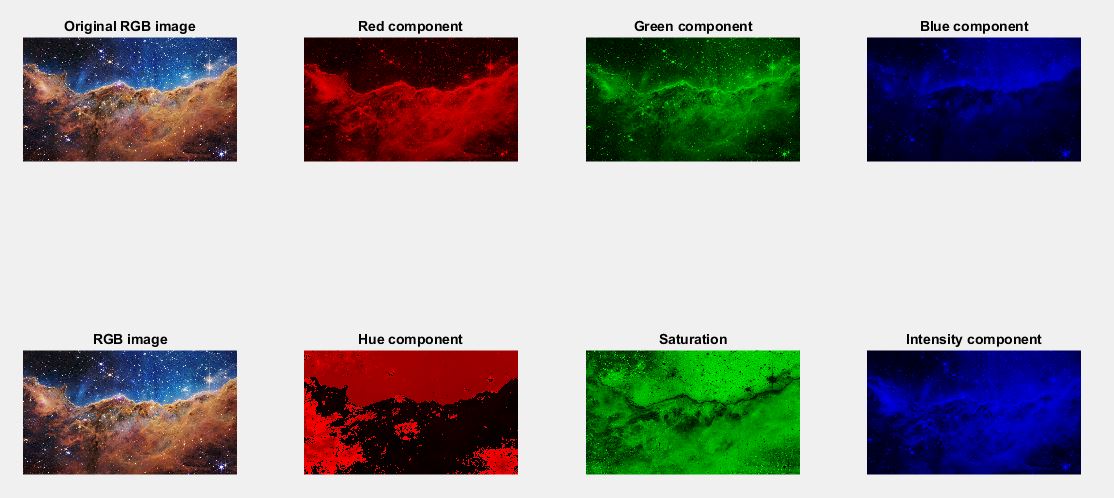
**Output:**



**Source Code:**

|  |
| --- |
| clc;clear all;close all;  rgb\_img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  rgb\_img=im2double(rgb\_img);  subplot(241);imshow(rgb\_img);title('Original RGB image');  R=rgb\_img; R(:,:,2)=0; R(:,:,3)=0;  G=rgb\_img; G(:,:,1)=0; G(:,:,3)=0;  B=rgb\_img; B(:,:,1)=0; B(:,:,2)=0;  subplot(242);imshow(R);title('Red component');  subplot(243);imshow(G);title('Green component');  subplot(244);imshow(B);title('Blue component');  subplot(245);imshow(rgb\_img);title('RGB image');  hsi\_img=rgb2hsv(rgb\_img);  H=hsi\_img; H(:,:,2)=0; H(:,:,3)=0;  S=hsi\_img; S(:,:,1)=0; S(:,:,3)=0;  I=hsi\_img; I(:,:,1)=0; I(:,:,2)=0;  subplot(246);imshow(H);title('Hue component');  subplot(247);imshow(S);title('Saturation');  subplot(248);imshow(I);title('Intensity component'); |

**Output:**

****

**Source Code:**

|  |
| --- |
| rgb\_img=imread('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\\JWST.jpg');  rgb\_img=im2double(rgb\_img);  subplot(241);imshow(rgb\_img);title('Original RGB image');  avg\_filter=fspecial('average',[5 5]);laplacian\_filter=fspecial('laplacian',0.2);  R=rgb\_img(:,:,1); avg\_R=imfilter(R,avg\_filter);lap\_R=imfilter(R,laplacian\_filter);  G=rgb\_img(:,:,2); avg\_G=imfilter(G,avg\_filter);lap\_G=imfilter(G,laplacian\_filter);  B=rgb\_img(:,:,3); avg\_B=imfilter(B,avg\_filter);lap\_B=imfilter(B,laplacian\_filter);  avg\_rgb\_img=cat(3,avg\_R,avg\_G,avg\_B);lap\_rgb\_img=cat(3,lap\_R,lap\_G,lap\_B);  hsi\_img=rgb2hsv(rgb\_img);  H=hsi\_img(:,:,1);S=hsi\_img(:,:,2);I=hsi\_img(:,:,3);  avg\_I=imfilter(I,avg\_filter); lap\_I=imfilter(I,laplacian\_filter);  avg\_hsi\_img=cat(3,H,S,avg\_I);lap\_hsi\_img=cat(3,H,S,lap\_I);  avg\_rgb\_img\_from\_hsi=hsv2rgb(avg\_hsi\_img);lap\_rgb\_img\_from\_hsi=hsv2rgb(lap\_hsi\_img);  diff\_avg\_img=avg\_rgb\_img-avg\_rgb\_img\_from\_hsi;diff\_lap\_img=lap\_rgb\_img-lap\_rgb\_img\_from\_hsi;  subplot(242);imshow(avg\_rgb\_img);title('Smoothing using RGB model');  subplot(243);imshow(avg\_rgb\_img\_from\_hsi);title('Smoothing using HSI model');  subplot(244);imshow(diff\_avg\_img);title('Difference');  subplot(245);imshow(rgb\_img);title('Original RGB image');  subplot(246);imshow(lap\_rgb\_img);title('Sharpening using RGB model');  subplot(247);imshow(lap\_rgb\_img\_from\_hsi);title('Sharpening using HSI model');  subplot(248);imshow(diff\_lap\_img);title('Difference'); |

**Output:**

****

**Source Code:**

|  |
| --- |
| clc; clear all; close all;  I=imread('cameraman.tif');  subplot(231), imshow(I), title('Origianl image');  BW1=edge(I, 'sobel');  BW2=edge(I, 'canny');  subplot(232), imshow(BW1), title('Soble filter');  subplot(233), imshow(BW2), title('Canny filter');  I=imread('circuit.tif');  subplot(234), imshow(I), title('Origianl image');  BW1=edge(I, 'canny');  BW2=edge(I, 'prewitt');  subplot(235), imshow(BW1), title('canny filter');  subplot(236), imshow(BW2), title('prewit filter'); |

**Output:**

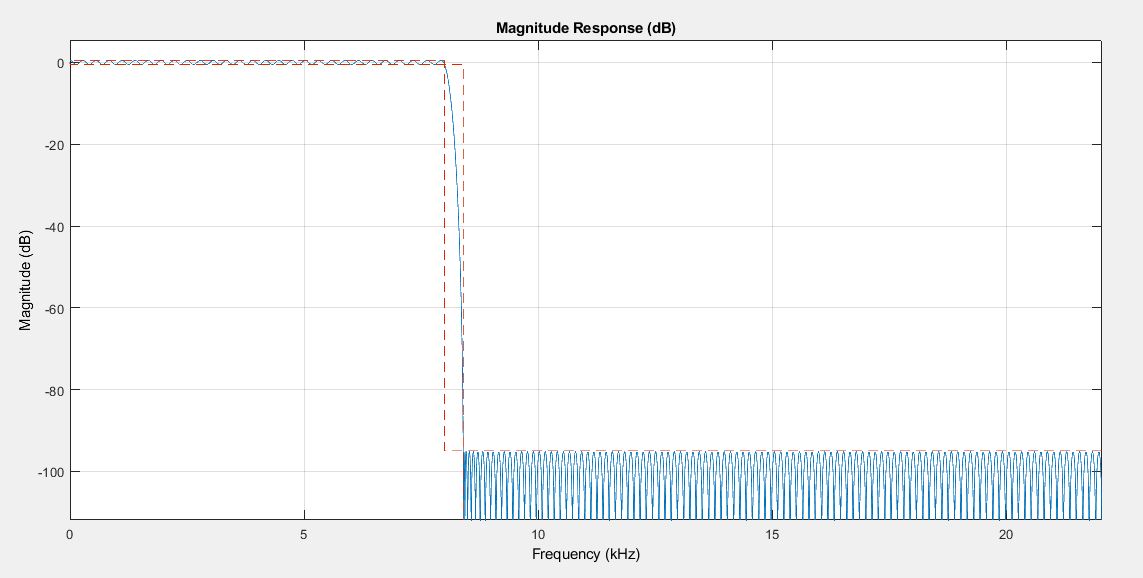
****

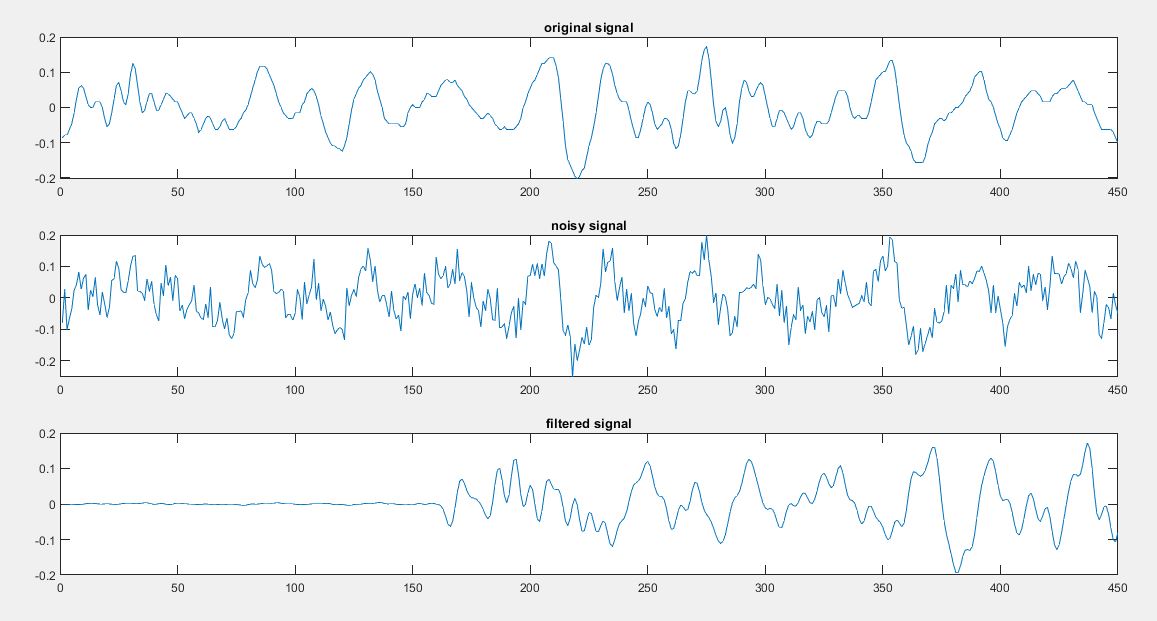
**Source Code:**

|  |
| --- |
| [filename,pathname]=uigetfile('\*.\*', 'select the input audio');  [x, Fs]=audioread(num2str(filename));  Fsf=44100; %sampling frequency  Fp=8e3; %passband frequency in Hz  Fst=8.4e3; %stopband frequency in Hz  Ap=1; %passband ripple in db  Ast=95; %stopband attenuation in db  %design the filter  df=designfilt('lowpassfir', 'PassbandFrequency', Fp, 'StopbandFrequency', Fst, 'PassbandRipple', Ap, 'StopbandAttenuation', Ast, 'SampleRate', Fsf);  fvtool(df) %visualize frequency response fvtool(df);  xn=awgn(x,15, 'measured'); %signal corrupted by white gaussian noise  y=filter(df, xn);  subplot(3,1,1), plot(x(1:450)); title('original signal');  subplot(3,1,2), plot(xn(1:450)); title('noisy signal');  subplot(3,1,3), plot(y(1:450)); title('filtered signal'); |

**Output:**

Fst= 8400  
Type in Command Window(MATLAB)::  
sound(x, Fs)  
We’ll hear a sound without noise.  
sound(xn, Fs)  
This time we will hear a sound with noise.  
sound(y, Fs)  
the sound is filtered, We’ll hear the sound without noise.

****

****

**Source Code:**

|  |
| --- |
| [filename, pathname]=uigetfile('\*.\*', 'select the input audio');  [x, Fs]=audioread(num2str('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\taunt.wav'));  n=length(x); %get the length of the audio file  a=0.8; %attenuation factor  d=2000; %delay  y=zeros(n+d, 1); %initialize the output audio file  xn=padarray(x, [d,0], 0, 'Pre');  for i=(d+1):1:n  y(i-d, 1)=x(i)+a\*xn(i-d);  end |

**Output:**

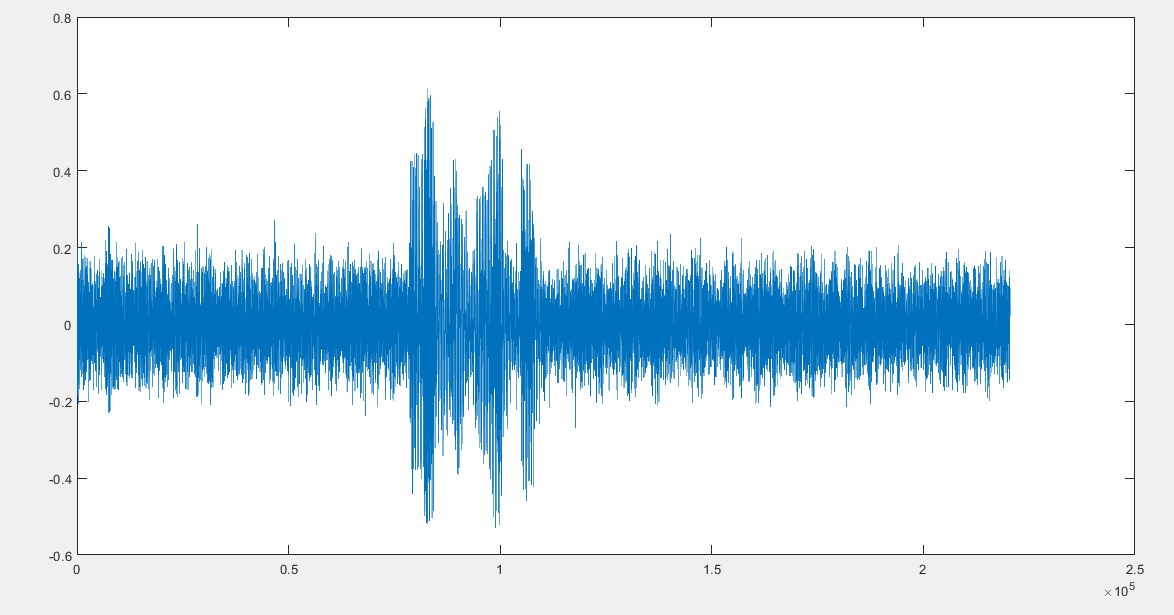
Type in Command Window(MATLAB):  
sound(x, Fs)  
We’ll hear the input audio signal.  
sound(y, Fs)  
this time we will hear the audio signal with echo.

**Source Code:**

|  |
| --- |
| Fs=44100; %sampling frequency, default value is 8192  noc=1; %no of input channel  nob=16; %no of bits per sample, default value is 8bits per sample  recObj=audiorecorder(Fs, nob, noc);  %record(recObj)  %pause(5); % no ofseconds  recordblocking(recObj, 5); %start recording for 5 seconds  play(recObj); %play back the recording  myRecording=getaudiodata(recObj); %getting recorded audio data as an array  plot(myRecording);  sound(myRecording, Fs); %play back the recording  audiowrite('E:\3-2\ICE-3207 DISP\ICE-3208 Sessional\Exm\Lab14\miketest.wav', myRecording, Fs); |

**Output:**

If we run the MATLAB code, the microphone will start recording the audio right away, if we say “Hello!! This is a test”, then the computer will record the audio and save it to the desired location then play it.

****

**Source Code:**

|  |
| --- |
| NET.addAssembly('System.speech');  mySpeaker=System.Speech.Synthesis.SpeechSynthesizer;  mySpeaker.Rate=3;  mySpeaker.Volume=100;  Speak(mySpeaker, 'Hello, This is a Test!'); |

**Output:**

If we run the MATLAB code, we will hear a voice saying

Hello, This is a Test!