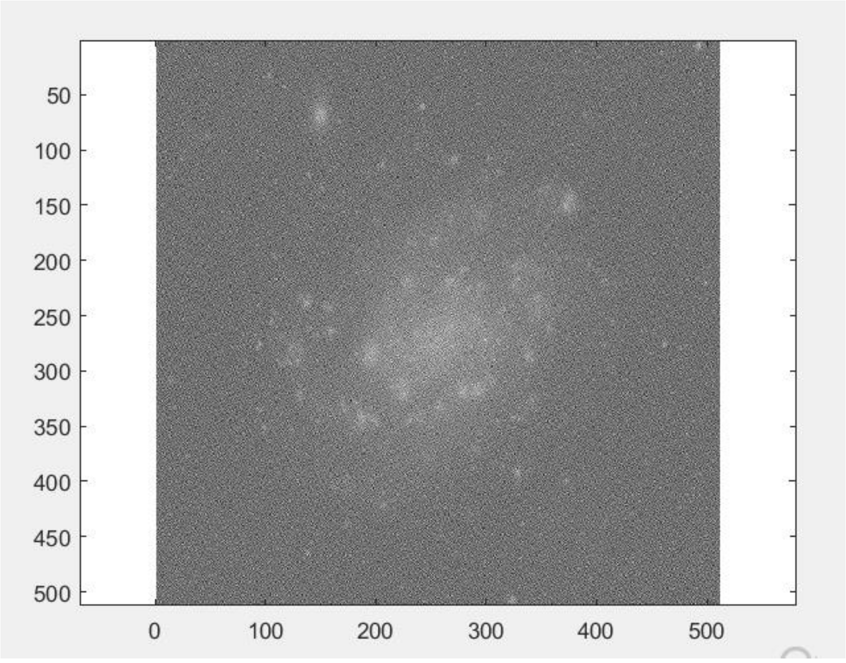
Part 3

Ans.1: For, $\sigma^2 = 0$, using the Wiener filter formulation to estimate the signal,

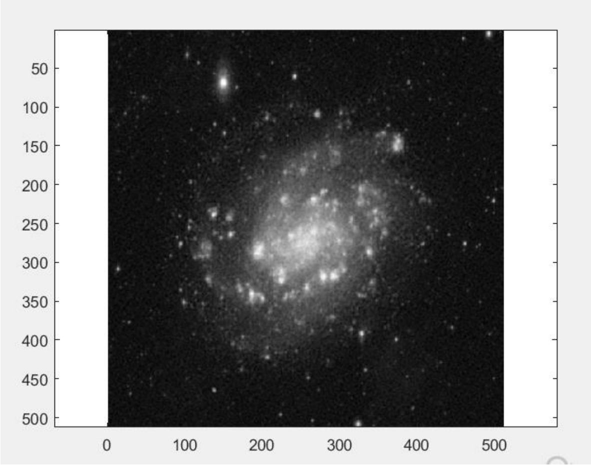


Ans 2: In this the image is not really clear. When $\sigma^2 = 0$ Wiener filter in the frequency domain is inverse of frequency domain impulse response, $H(f,v) = \dfrac{1}{G(f,v)}$. And when frequencies are near zero, then $G(f,v)\big|_{f,v\approx 0} = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} g(n_1,n_2)$. Which makes frequency response of the wiener filter really small. Hence we get a really bad estimate.

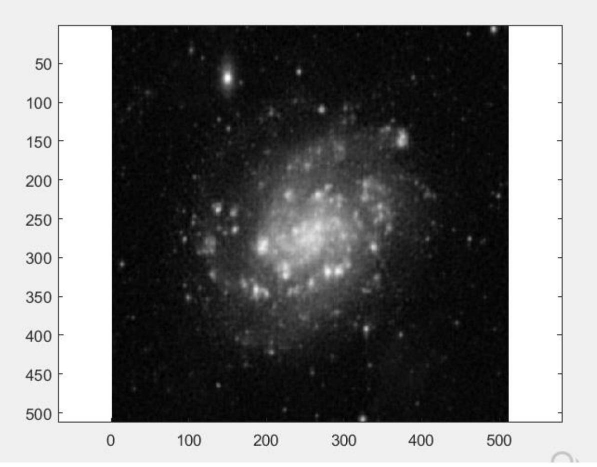
We can not recover frequency components set to zero. As can be seen by above formula.

Ans3:

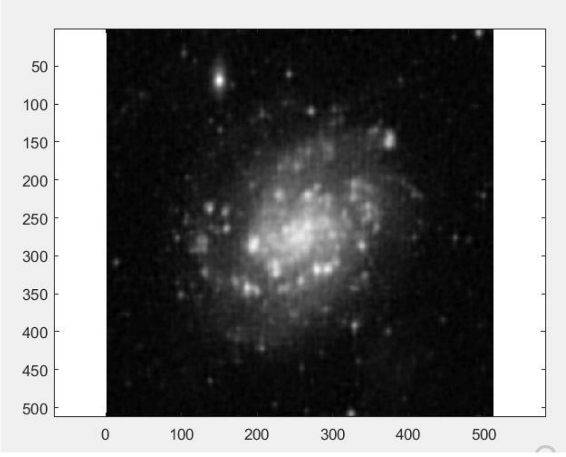
a) $\sigma^2 = 0.00001$



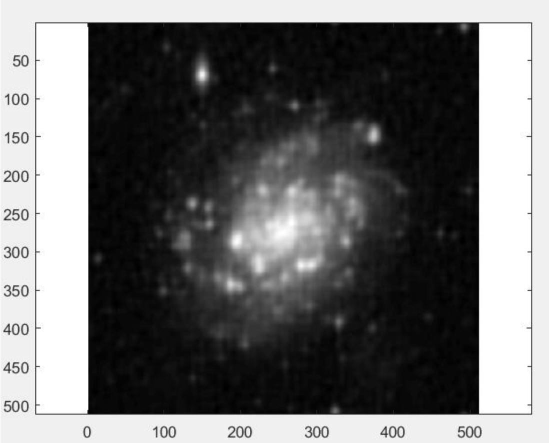
b) $\sigma^2 = 0.0001$



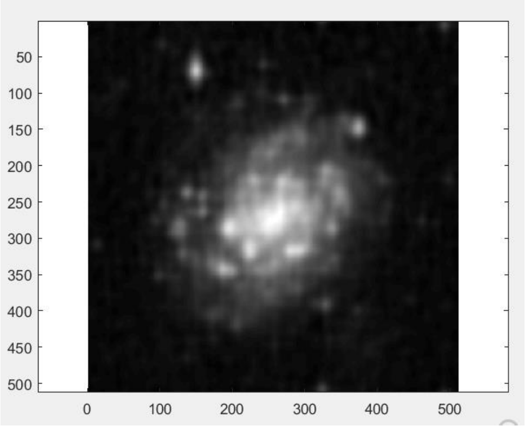
c) $\sigma^2 = 0.001$



d) $\sigma^2 = 0.01$



e) $\sigma^2 = 0.1$



Ans 4: For $\sigma^2 = 0.00001$ the estimate is most clear. As we increase σ^2, noise power spectral density increases, which in turn, reduces the quality of estimate. And frequency response of wiener filter decreases with increasing σ^2.

Part 4:

Used Clean Image

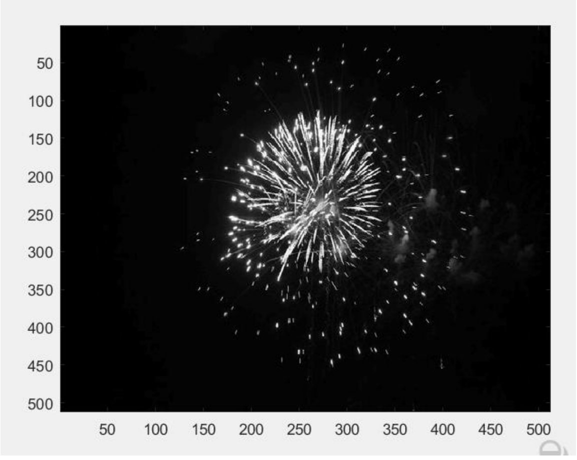
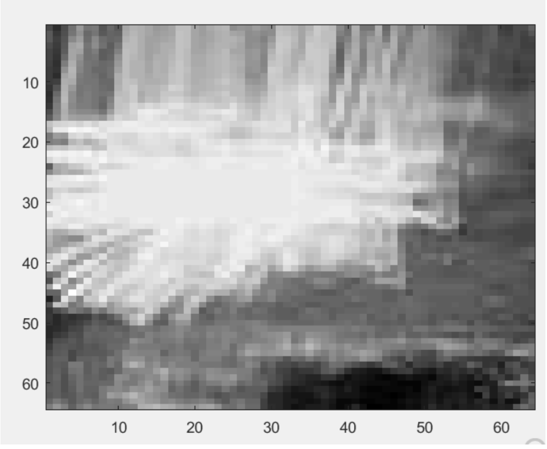
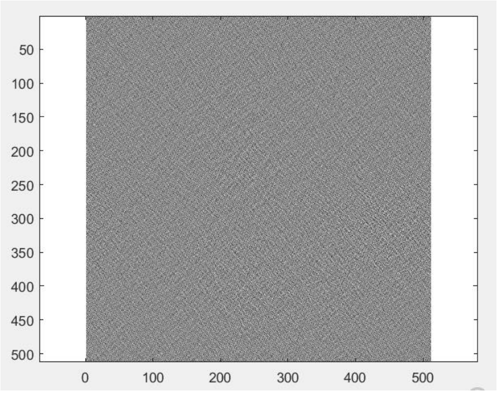


Image used to Estimated_g:



For all σ^2, the estimated image is posted below:



Appendix A: MATLAB Code for Part 3:

```
clear all;
load Hubble.mat;
sigma2=0.1;
zp1=zero_pad(estimated_g,512,512);     % resizing the impulse
response of image
DFTeg=fft2(zp1);          % DFT of zero padded impulse response
N1=512;
N2=512;
DFTcg=fft2(clean_galaxy);                 % DFT of clean galaxy
sDFTcg=(abs(DFTcg)).^2;
Sv1=sDFTcg/(N1*N2);                      % Using periodogram to
approximate the power-spectral density of unknown
```

```
Sw=sigma2;                           % noise power spectral 
density
Fwf=(conj(DFTeg).*Sv1)./(((abs(DFTeg)).^2.*Sv1)+Sw);   % 
Frequency response of wiener filter
DFTbg=fft2(blurred_galaxy);          % FFT of blurred image
Sv2=Fwf.*DFTbg;         % unknown variable frequency response
iSv2=ifft2(Sv2);           % unknown variable in time domain
Rv2=real(iSv2);
imagesc(Rv2); colormap gray; axis equal;
drawnow;
```

Appendix B: Code for Part 4:

```
filename=['s','.jpg'];  % s.jpg is one clean image
filename2=['g','.jpg'];  % estimate
I=imread(filename);
I2=imread(filename2);
X=rgb2gray(I);
Y=rgb2gray(I2);
clean_firework=double(X);
estimate_firework=double(Y);
save firworkFinal.mat clean_firework estimate_firework;
```