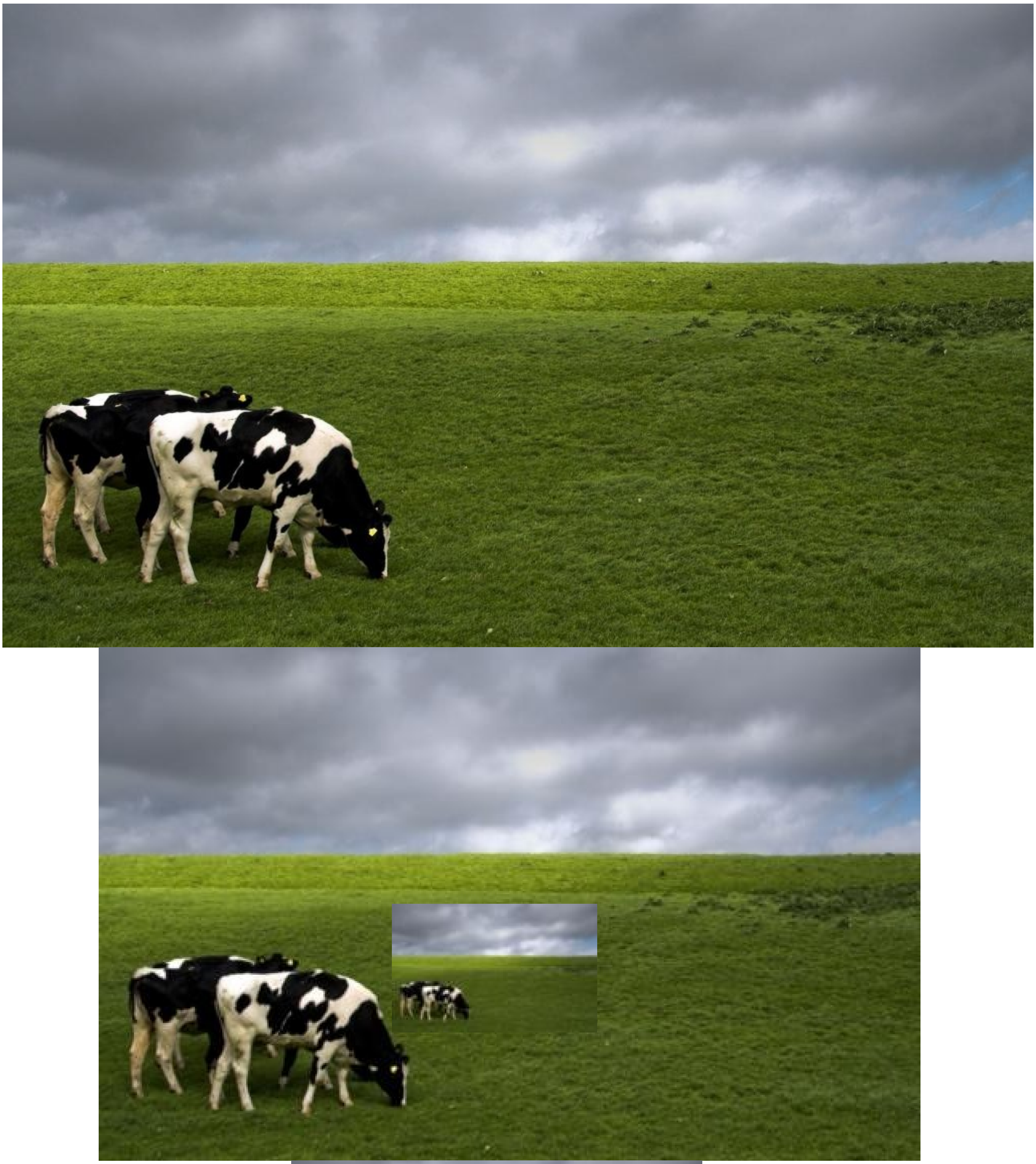
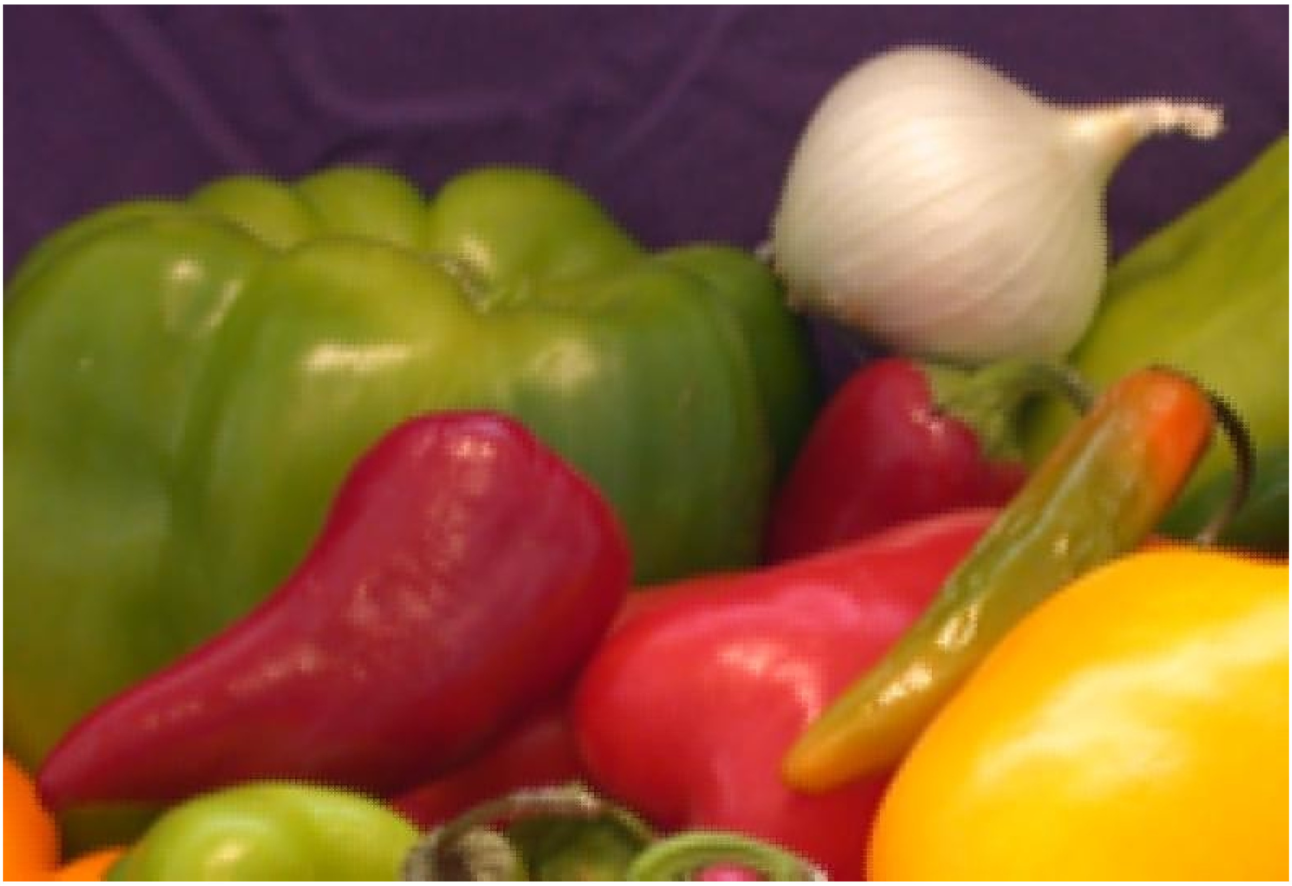
Gaussian Pyramid:(4 levels)

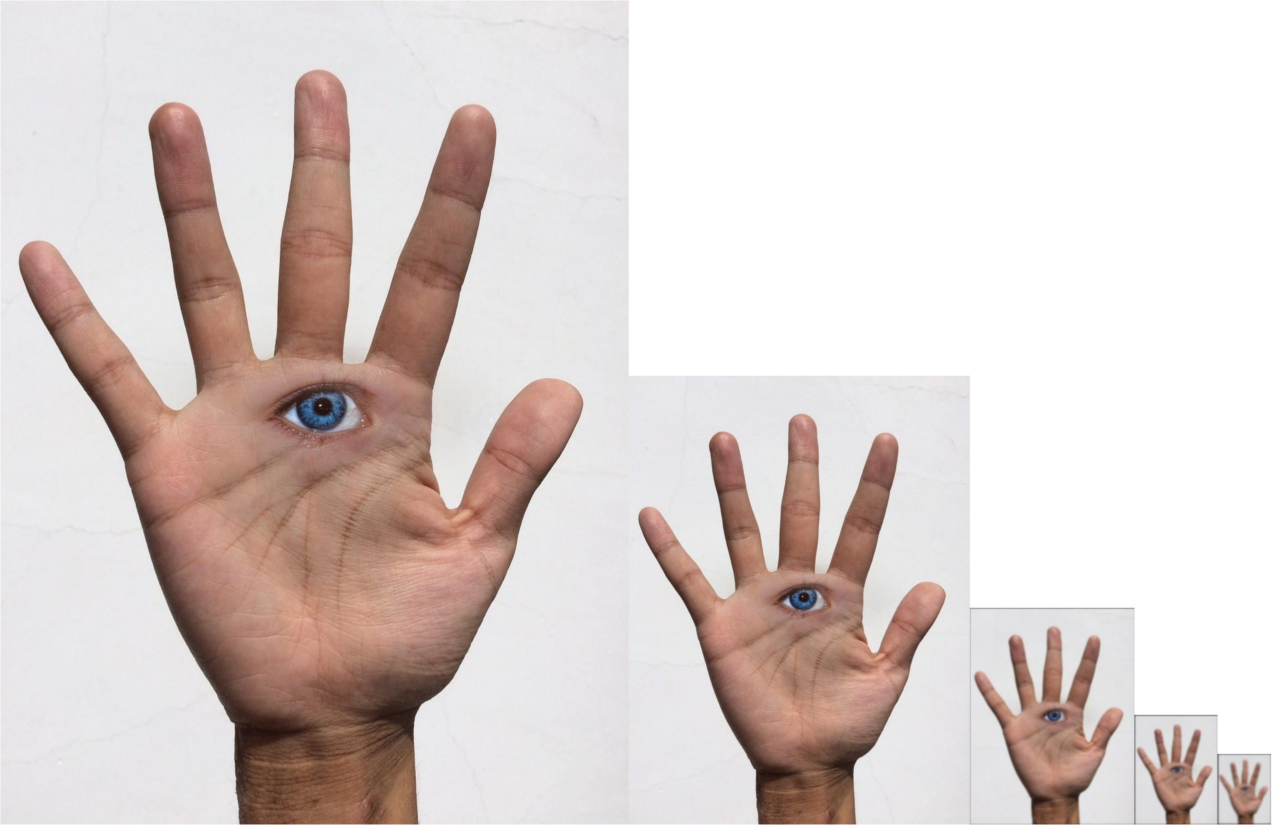


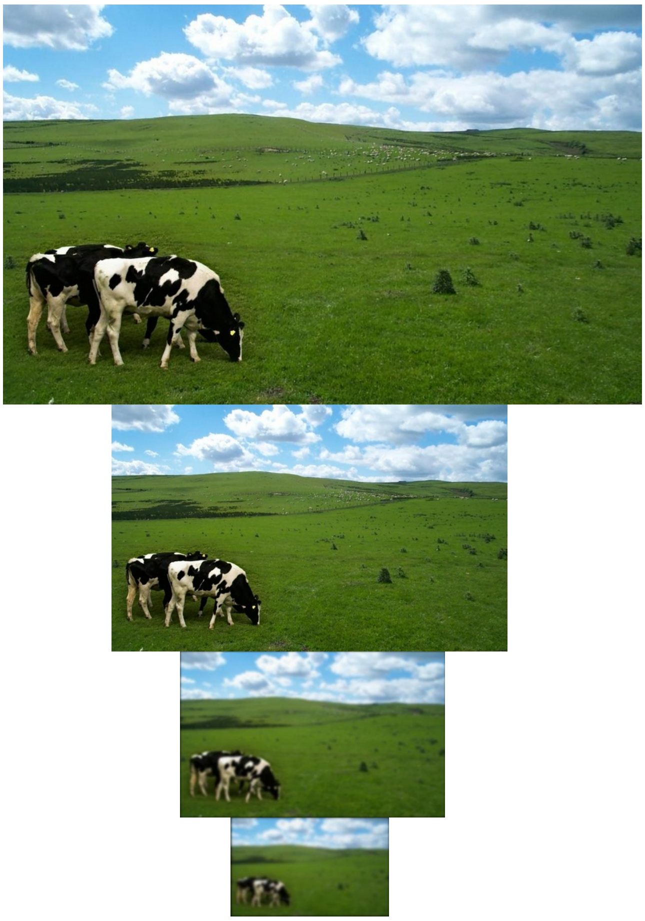
Laplacian Pyramid:(4 levels)

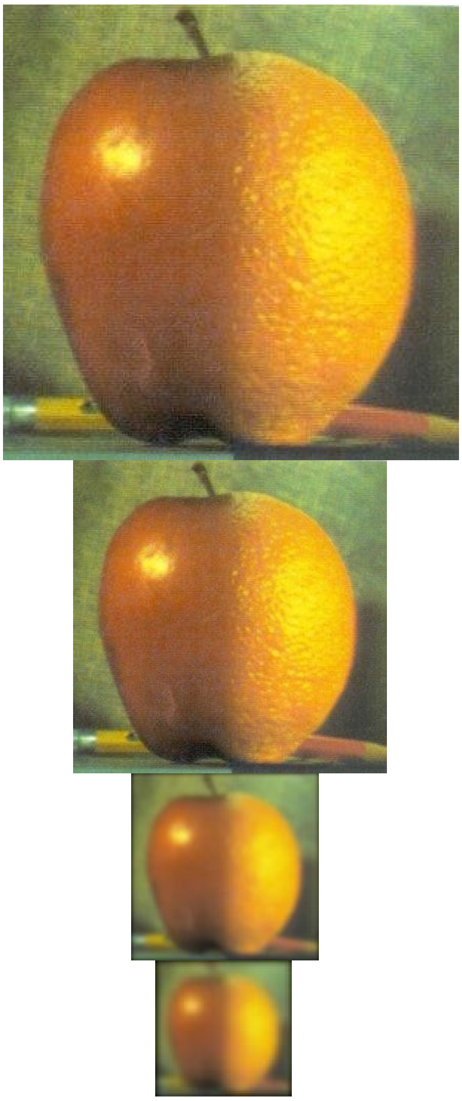
UpSample two times using Nearest Neighbours

UpSample two times using Bi-Linear Interpolation

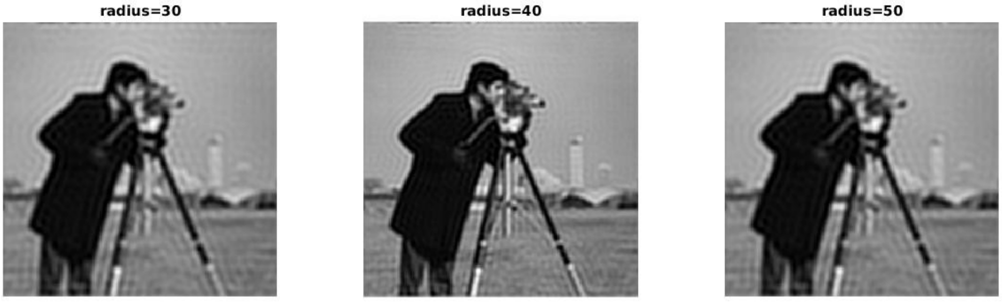




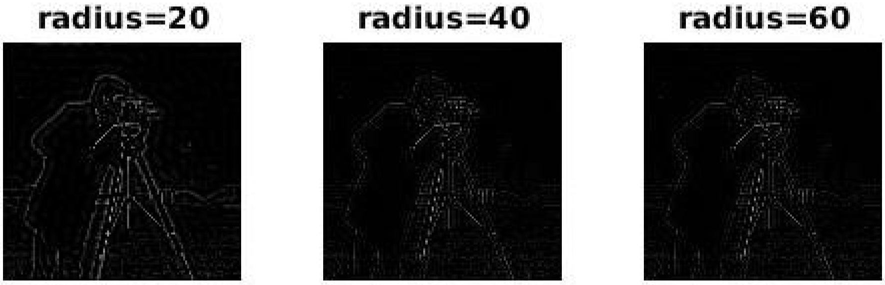




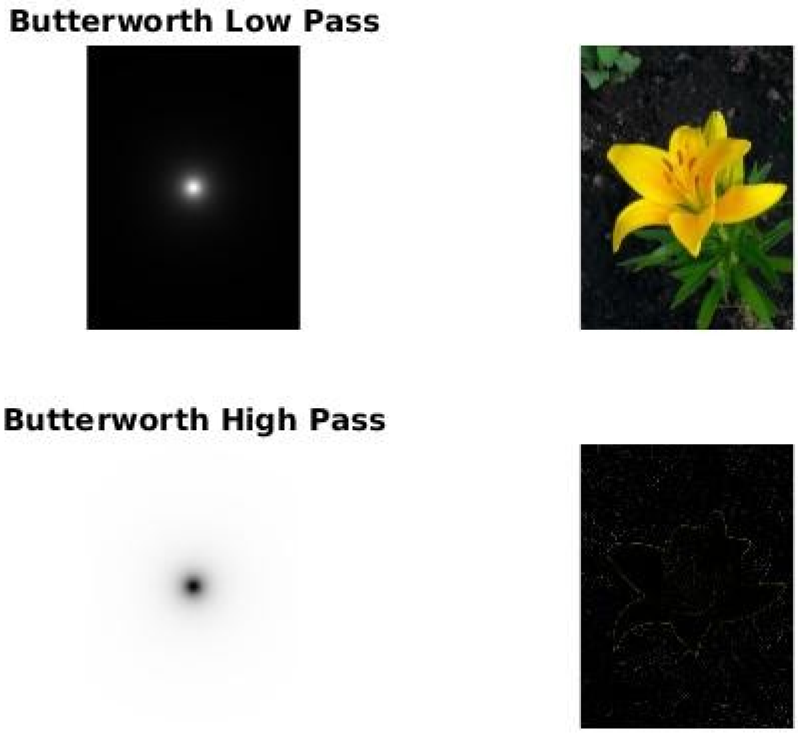
Ideal Low Pass Filter

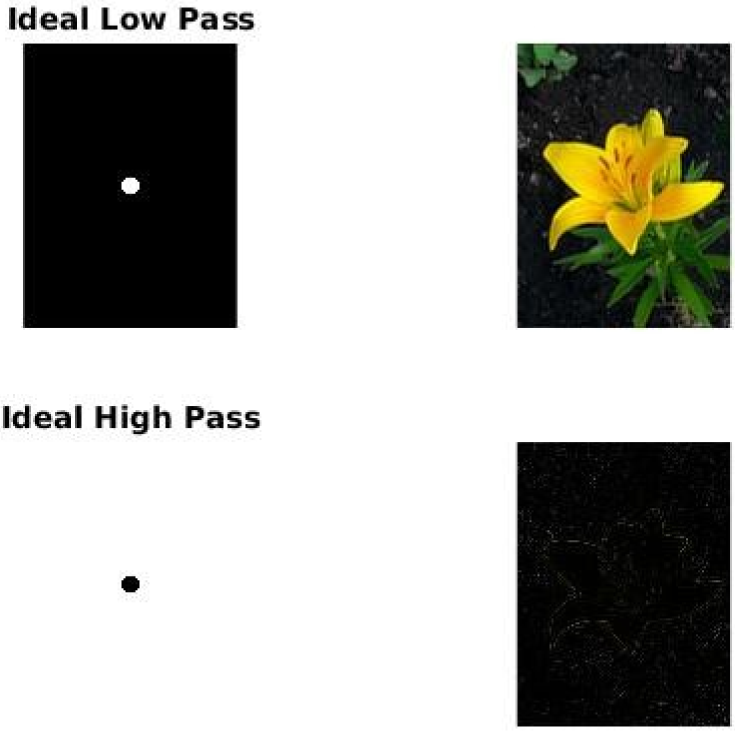


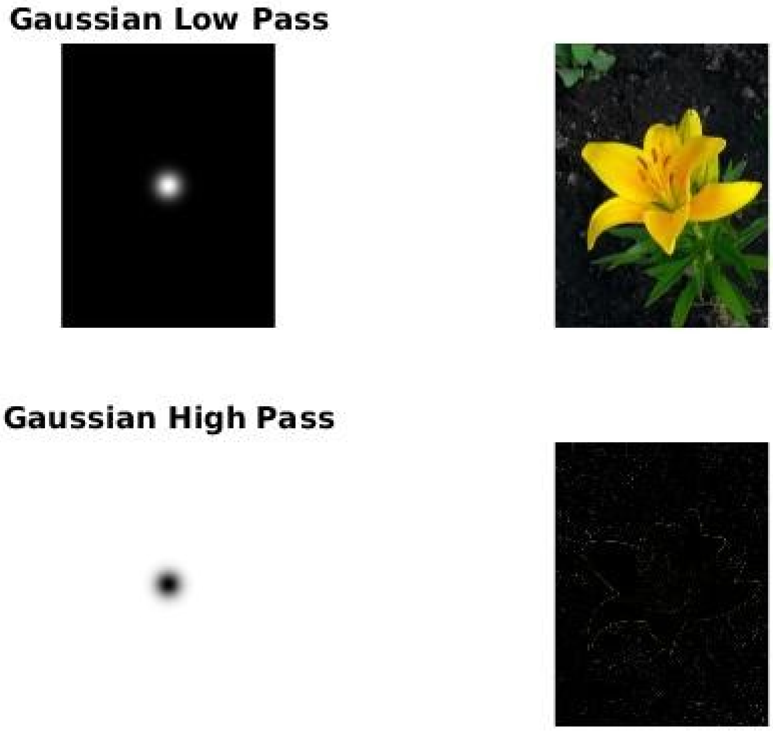
ideal Highpass Fitler



Butterworth Filter




Ideal Low Pass
Ideal High Pass

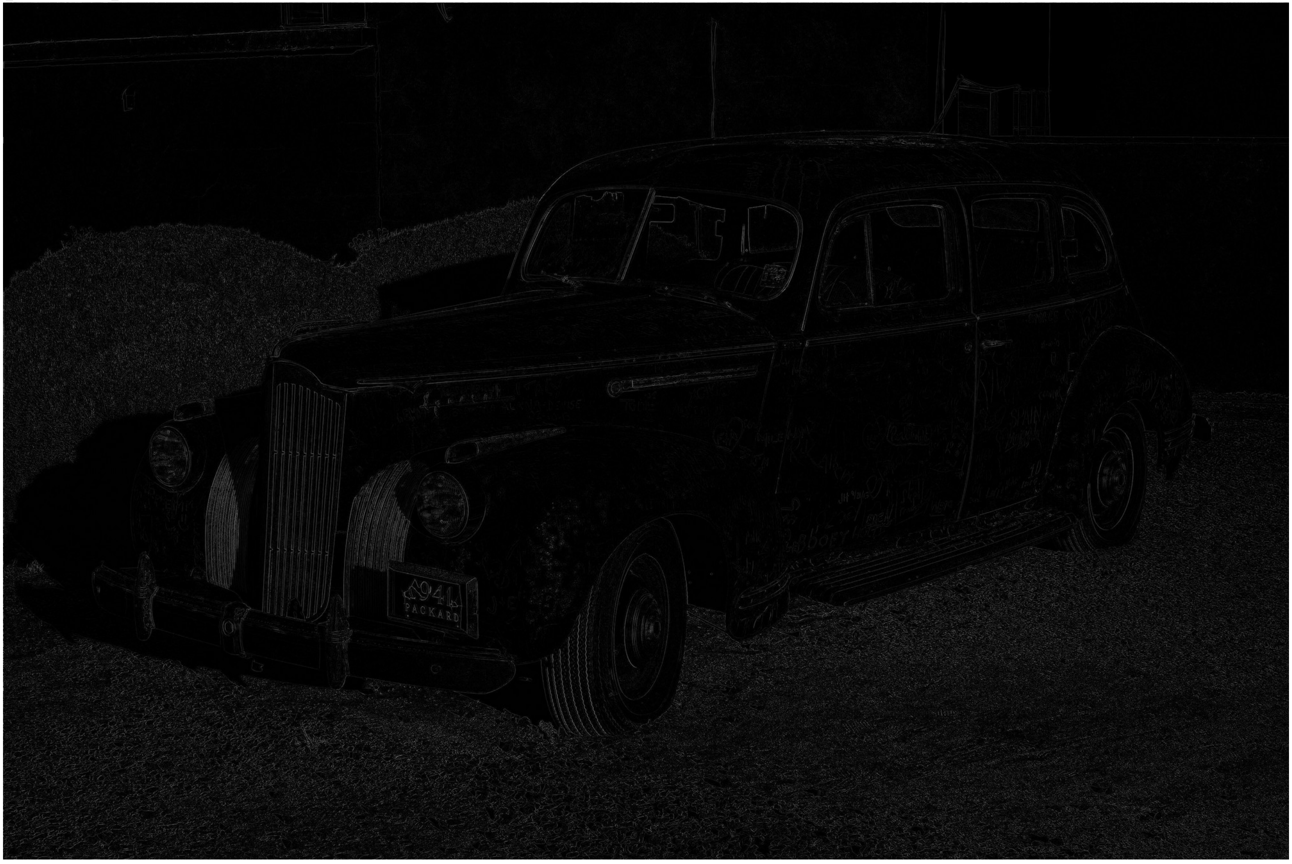

Gaussian Low Pass
Gaussian High Pass

Laplase Filter

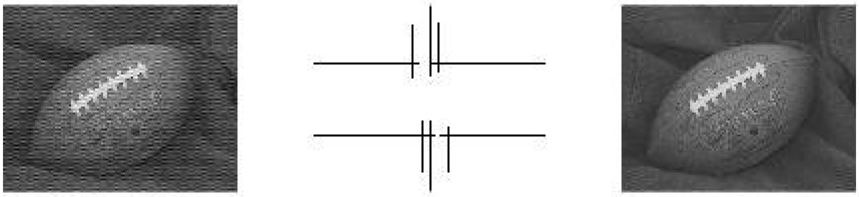
[0 1 0]

[1 -4 1]

[0 1 1]



Notch Pass Filter



```
function [F]=FFT2(a)
    [~,N]=size(a);
    if N==1
        F=a;
    else
        a_even_even=FFT2(a(1:2:end,1:2:end));
        a_even_odd=FFT2(a(1:2:end,2:2:end));
        a_odd_even=FFT2(a(2:2:end,1:2:end));
        a_odd_odd=FFT2(a(2:2:end,2:2:end));

        Q1=exp(-2*pi*1i*(repmat((0:N/2-1)',1,N/2)')./N);
        Q2=exp(-2*pi*1i*(repmat((0:N/2-1),N/2,1)')./N);
        Q3=Q1.*Q2;

        F1=a_even_even;
        F2=a_even_odd.*Q1;
        F3=a_odd_even.*Q2;
        F4=a_odd_odd.*Q3;
        F=[(F1+F2+F3+F4) (F1-F2+F3-F4);(F1+F2-F3-F4) (F1-F2-F3+F4)];
    end
end
```