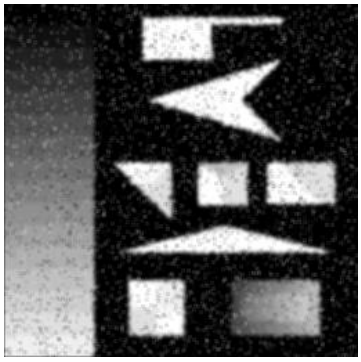
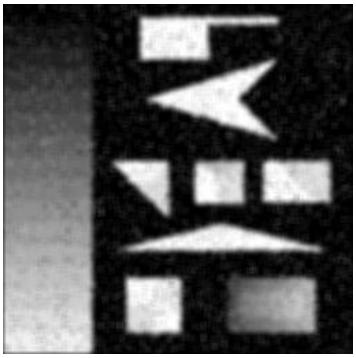
Question 1: Box Filter

Image 1



Kernel = 3



Kernel = 5

Image 2

Kernel = 3

Kernel = 5

This filter does not remove the noise. As the kernel size gets bigger then the images blur even more.

Question 2: Median Filter

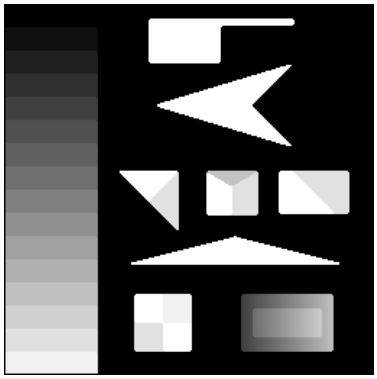
Image 1

Kernel = 3

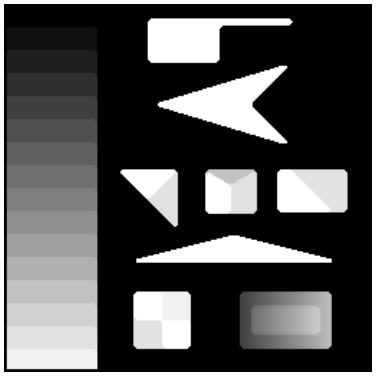
Kernel = 5

Kernel = 7

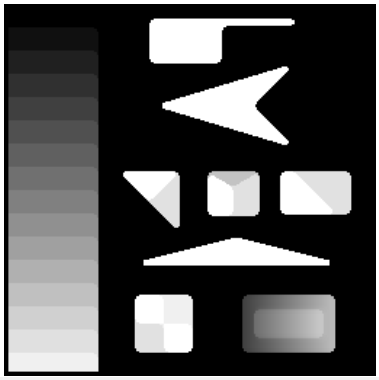
Image 2



Kernel = 3



Kernel = 5



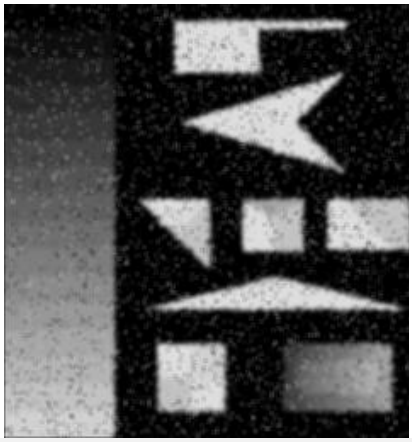
Kernel = 7

This filter removes the noise and at increasing kernel size smooths the edges. The corners become less sharp and more rounded. The rounded edges are more noticeable in the clearer image.

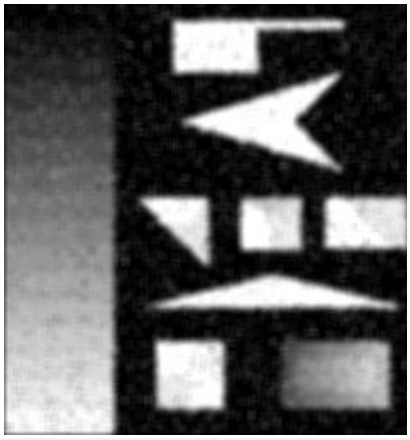
The median filter appears to be most effective at reducing noise.

Question 3: Gaussian Filter

Image 1



Kernel = 3



Kernel = 5

Kernel = 11

Image 2

Kernel = 3

Kernel = 5

Kernel = 11

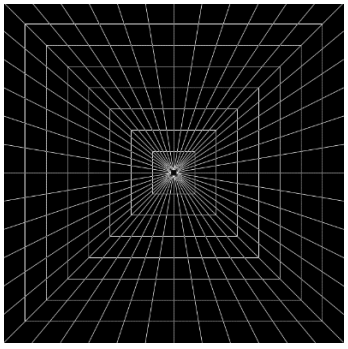
This filter also does not remove the noise and with a higher kernel size if blurs the image even more. Also, at higher kernels the image seems to reduce the noise such as at k=11, the noise is really hard to notice.

If you want to blur the image, then the Gaussian filter will be better as it won't round the edges and a higher pixel intensity in the neighborhood won't affect the blur.

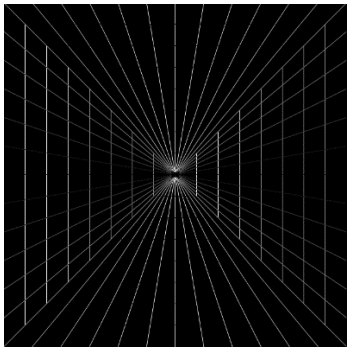
If you want to reduce noise, then the median filter will be useful as it does just that although it does round the edges at higher kernel sizes.

Question 4: Gradient Operations

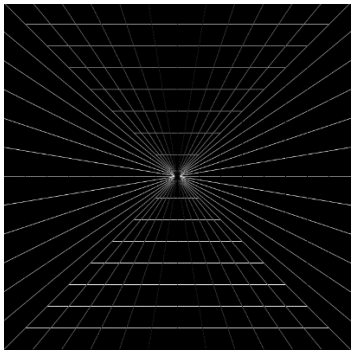
Backward Difference



Magnitude

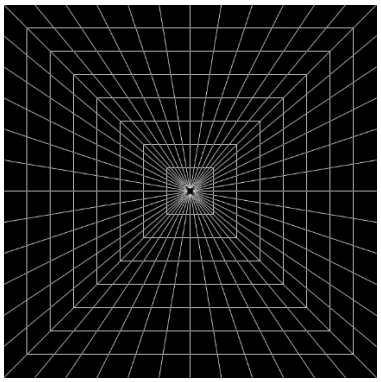


X Direction

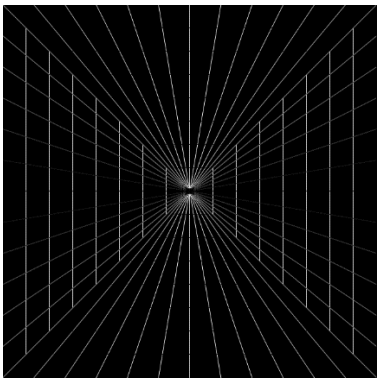


Y Direction

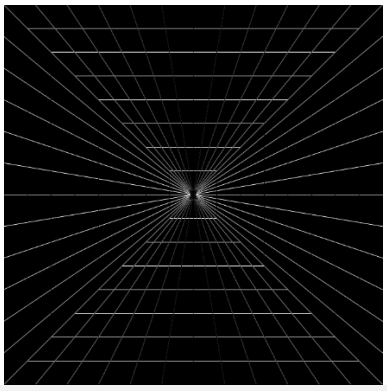
Forward Difference



Magnitude

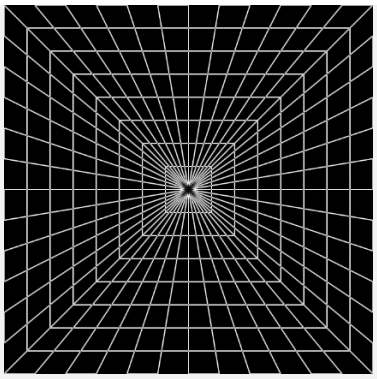


X Direction

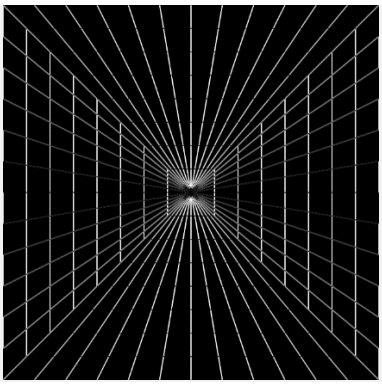


Y Direction

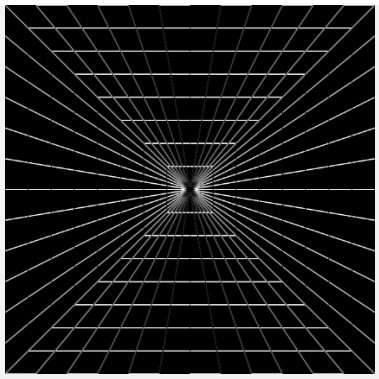
Center Difference



Magnitude



X Direction



Y Direction

Question 5: Sobel Filter

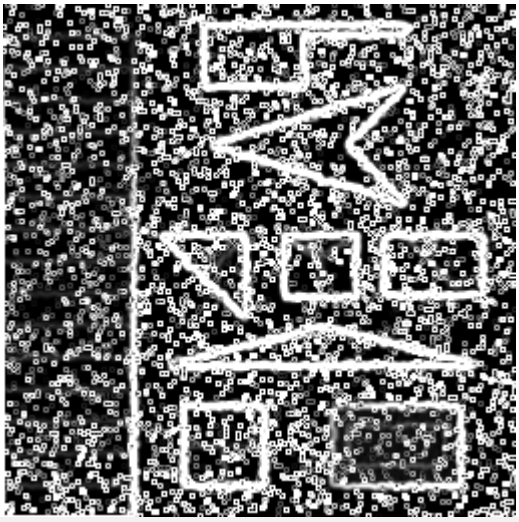


Image 1, Kernel = 3

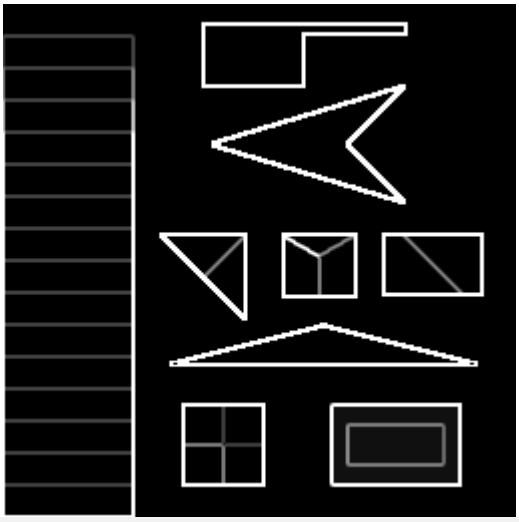


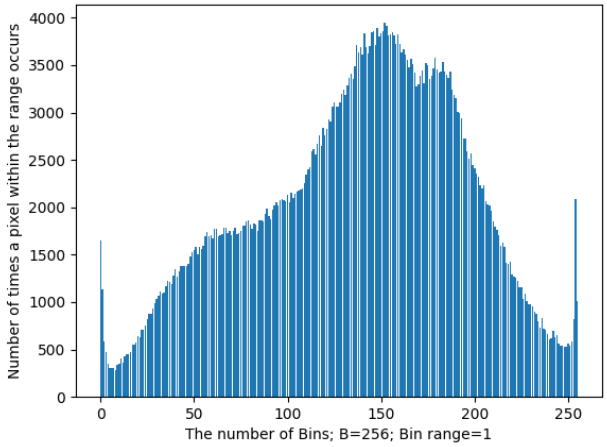
Image 2, Kernel = 3

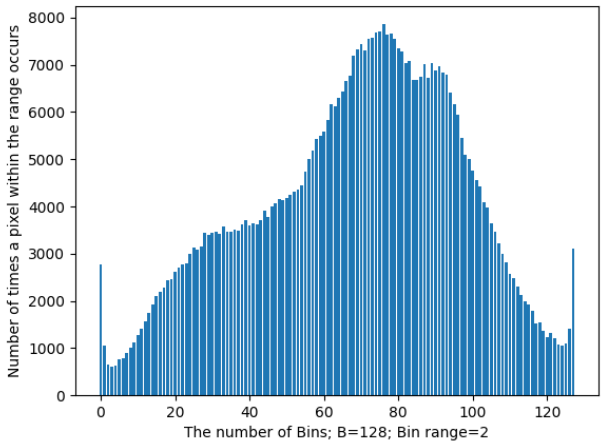
With this filter it seems to work best at finding edges when there is no noise. It seems to also detect the edges of each of the noise dots which may interfere with detecting other edges.

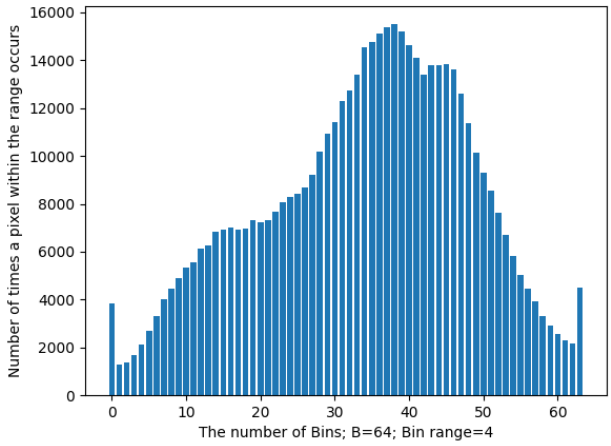
Question 6: Fast Gaussian Filter

This should output the same images as the other Gaussian Filter. Because we're splitting up the filters however it will be faster. In the Q4 Gaussian Filter I did a 2D filter and 2D convolution. This uses nested for loops to find the correlation then convolves the kernel and the image. With the fast gaussian filter because this uses a 1D filter it only needs to loop through the array once per direction and convolve the 1D kernel with the image.

Question 7: Histogram







When we use less bins then we can increase each bin's range. This will group together the pixel intensities that appear which will double the number of times a pixel appears per bin

Question 8: Canny Edge Detection

Gradient Images

Image 1

X Gradient Sigma=1

Y Gradient Sigma =1

Magnitude Sigma = 1

X Gradient Sigma=3

Y Gradient Sigma =3

Magnitude Sigma =3

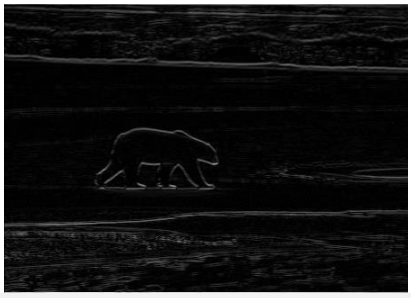
X Gradient Sigma=5

Y Gradient Sigma =5

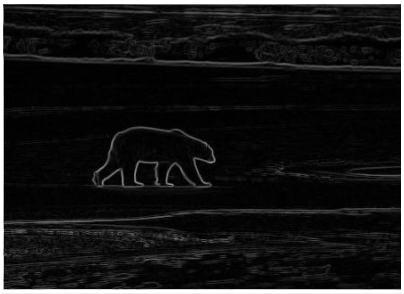
Magnitude Sigma 5

Image 2

X Gradient Sigma=1



Y Gradient Sigma =1



Magnitude Sigma = 1

X Gradient Sigma=3

Y Gradient Sigma =3

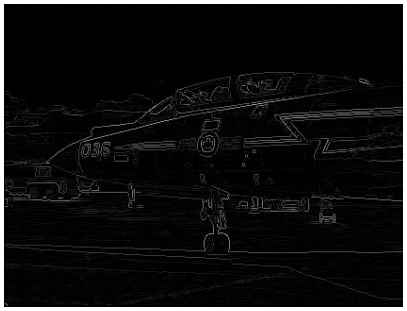
Magnitude Sigma =3

X Gradient Sigma=5

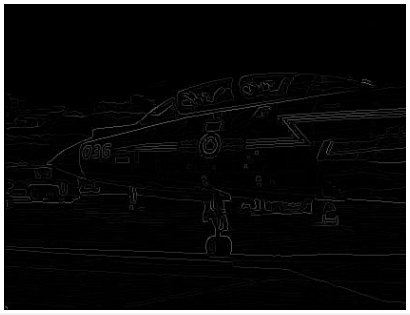
Y Gradient Sigma =5

Magnitude Sigma =5

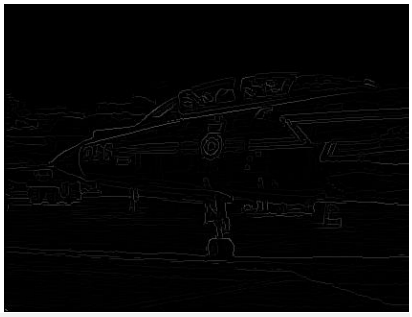
After Non-Max Suppression



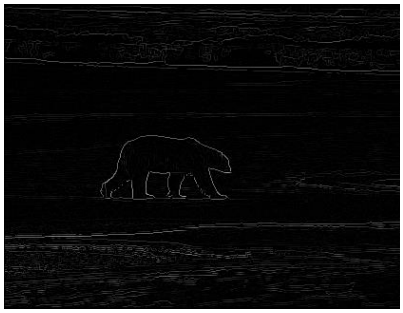
Sigma = 1



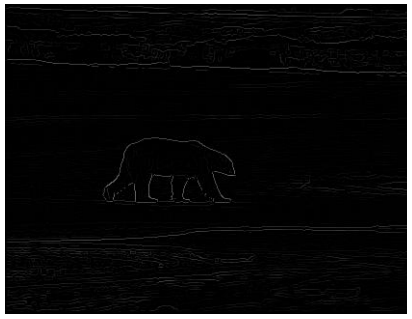
Sigma =3



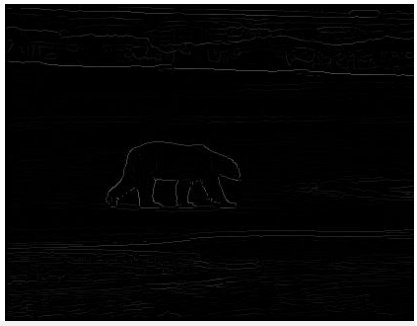
Sigma =5



Sigma = 1

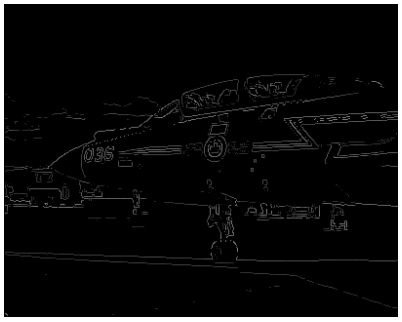


Sigma =3

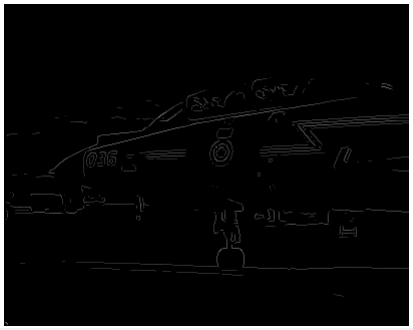


Sigma =5

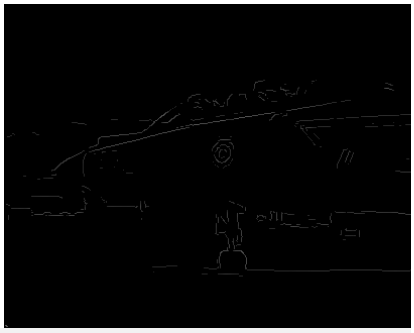
After Hysteresis



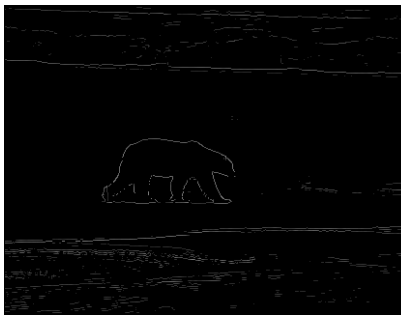
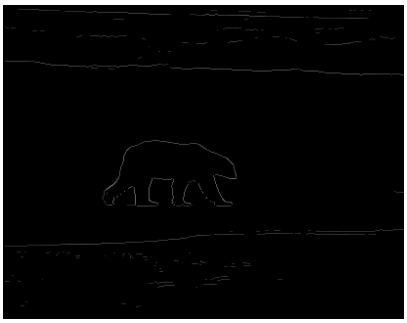
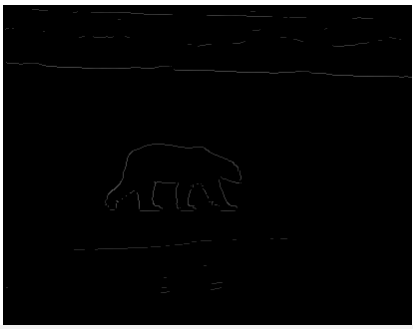
Sigma = 1



Sigma =3



Sigma =5

		
Sigma = 1	Sigma =3	Sigma =5

It seems that at higher sigma values it doesn't detect as many edges as at lower values. If you want the edges that help detail the image, then using a low sigma value will detect these edges. If you want edges that are more "important" then using a higher sigma value will essentially give you a rough outline of the image without the details.

Question 9: Image Segmentation

Binarization with threshold chosen manually

Original | Threshold=50 | Threshold=60 | Threshold=80

Original | Threshold =65 | Threshold =130 | Threshold =170

Original | Threshold =60 | Threshold =100 | Threshold =140

Binarization with Otsu Threshold

Threshold = 33

Threshold=171

Threshold=8

Images from: Egyptian King: https://news.artnet.com/exhibitions/king-tut-london-saatchi-1692016

Cowboy: https://avatars.alphacoders.com/avatars/view/169248

It seems that the Otsu threshold chooses a low threshold value. At lower thresholds the image has more detail than at a higher threshold where it starts showing a basic outline of the image.