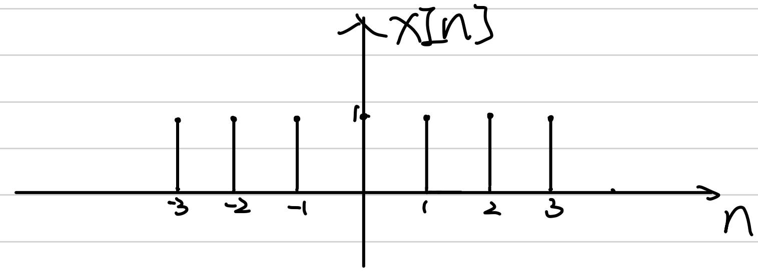
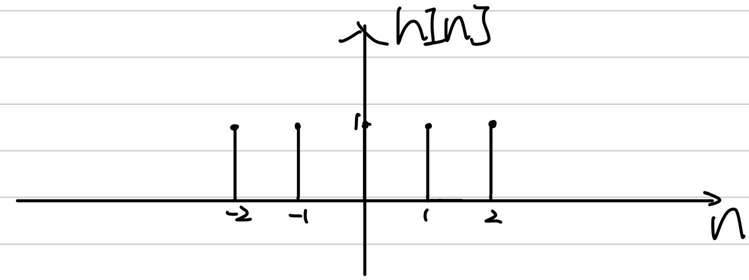
[1.a] (5 marks) Calculate and plot the convolution of $x[n]$ and $h[n]$ specified below:

$$x[n] = \begin{cases} 1 & -3 \le n \le 3 \\ 0 & \text{otherwise} \end{cases} \qquad h[n] = \begin{cases} 1 & -2 \le n \le 2 \\ 0 & \text{otherwise} \end{cases}$$





$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$

$n_{start} = -3+(-2) = -5$

$n_{end} = 3+2 = 5$

when $n=-5$ $\quad y[-5] = \sum_{k=-\infty}^{\infty} x[k]h[-5-k]$

$= x[-3]h[-5-(-3)] = 1$

when $n=-4$ $\quad y[-4] = \sum_{k=-\infty}^{\infty} x[k]h[-4-k]$

$= x[-3]h[-4-(-3)] + x[-2]h[-4-(-2)]$

$= 2$

when $n=-3$ $\quad y[-3] = \sum_{k=-\infty}^{\infty} x[k]h[-3-k]$

$= x[-3]h[-3-(-3)] + x[-2]h[-3-(-2)] + x[-1]h[-3-(-1)]$

$= 3$

when $n=-2$ $\quad y[-2] = \sum_{-\infty}^{\infty} x[k]h[-2-k]$

$= x[-3]h[-2-(-3)] + x[-2]h[-2-(-2)] + x[-1]h[-2-(-1)] + x[0]h[-2-0]$

$= 4$

when $n=-1$ $\quad y[-1] = \sum_{k=-\infty}^{\infty} x[k]h[-1-k]$

$= x[-3]h[-1-(-3)] + x[-2]h[-1-(-2)] + x[-1]h[-1-(-1)] + x[0]h[-1-0] + x[1]h[-1-1]$

$= 5$

when $n=0$ $\quad y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$

$= x[-2]h[0-(-2)] + x[-1]h[0-(-1)] + x[0]h[0-0] + x[1]h[0-1] + x[2]h[0-2]$

$= 5$

when $n=1$ $\quad y[1]=\sum_{-\infty}^{+\infty} x[k]h[1-k]$

$=x[-1]h[1-(-1)]+x[0]h[1-0]+x[1]h[1-1]+x[2]h[1-2]+x[3]h[1-3]$

$=5$

when $n=2$ $\quad y[2]=\sum_{-\infty}^{+\infty} x[k]h[2-k]$

$=x[0]h[2-0]+x[1]h[2-1]+x[2]h[2-2]+x[3]h[2-3]$

$=4$

when $n=3$ $\quad y[3]=\sum_{-\infty}^{+\infty} x[k]h[3-k]$

$=x[1]h[3-1]+x[2]h[3-2]+x[3]h[3-3]$

$=3$

when $n=4$ $\quad y[4]=\sum_{k=-\infty}^{+\infty} x[k]h[4-k]$

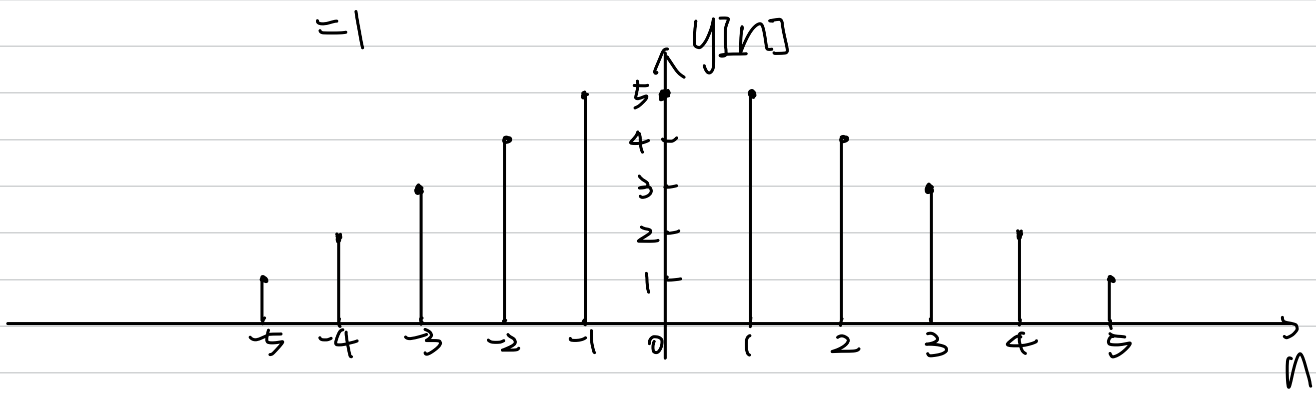
$=x[2]h[4-2]+x[3]h[4-3]$

$=2$

when $n=5$ $\quad y[5]=\sum_{k=-\infty}^{+\infty} x[k]h[5-k]$

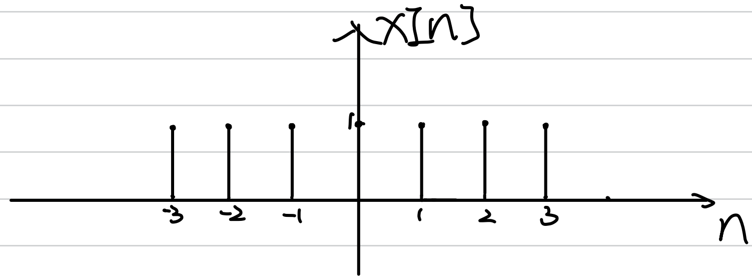
$=x[3]h[4-3]$

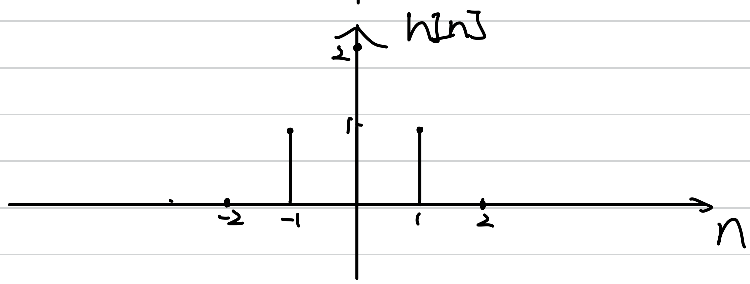
$=1$



[1.b] (5 marks) Calculate and plot the convolution of $x[n]$ and $h[n]$ specified below:

$$x[n] = \begin{cases} 1 & -3 \le n \le 3 \\ 0 & \text{otherwise} \end{cases} \qquad h[n] = \begin{cases} 2 - |n| & -2 \le n \le 2 \\ 0 & \text{otherwise} \end{cases}$$





$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

$$n_{start} = -3 + (-2) = -5$$
$$n_{end} = 3 + 2 = 5$$

when $n = -5$ $\quad y[-5] = \sum_{k=-\infty}^{\infty} x[k]h[-5-k]$

$$= x[-3]h[-5-(-3)] = 0$$

when $n = -4$ $\quad y[-4] = \sum_{k=-\infty}^{\infty} x[k]h[-4-k]$

$$= x[-3]h[-4-(-3)] + x[-2]h[-4-(-2)]$$
$$= 1$$

when $n = -3$ $\quad y[-3] = \sum_{k=-\infty}^{\infty} x[k]h[-3-k]$

$$= x[-3]h[-3-(-3)] + x[-2]h[-3-(-2)] + x[-1]h[-3-(-1)]$$
$$= 2 + 1 + 0 = 3$$

when $n = -2$ $\quad y[-2] = \sum_{k=-\infty}^{\infty} x[k]h[-2-k]$

$$= x[-3]h[-2-(-3)] + x[-2]h[-2-(-2)] + x[-1]h[-2-(-1)] + x[0]h[-2-0]$$
$$= 1 + 2 + 1 + 0 = 4$$

when $n = -1$ $\quad y[-1] = \sum_{k=-\infty}^{\infty} x[k]h[-1-k]$

$$= x[-3]h[-1-(-3)] + x[-2]h[-1-(-2)] + x[-1]h[-1-(-1)] + x[0]h[-1-0] + x[1]h[-1-1]$$
$$= 0 + 1 + 2 + 1 + 0 = 4$$

when $n = 0$ $\quad y[0] = \sum_{k=-\infty}^{\infty} x[k]h[0-k]$

$$= x[-2]h[0-(-2)] + x[-1]h[0-(-1)] + x[0]h[0-0] + x[1]h[0-1] + x[2]h[2-1]$$
$$= 0 + 1 + 2 + 1 + 0 = 4$$

when $n=1$ $\quad y[1]=\sum_{k=-\infty}^{\infty} x[k]h[1-k]$

$=x[-1]h[1-(-1)]+x[0]h[1-0]+x[1]h[1-1]+x[2]h[1-2]+x[3]h[1-3]$

$=0+1+2+1+0=4$

when $n=2$ $\quad y[2]=\sum_{k=-\infty}^{\infty} x[k]h[2-k]$

$=x[0]h[2-0]+x[1]h[2-1]+x[2]h[2-2]+x[3]h[2-3]$

$=0+1+2+1+0=4$

when $n=3$ $\quad y[3]=\sum_{k=-\infty}^{\infty} x[k]h[3-k]$

$=x[1]h[3-1]+x[2]h[3-2]+x[3]h[3-3]$

$=0+1+2=3$

when $n=4$ $\quad y[4]=\sum_{k=-\infty}^{\infty} x[k]h[4-k]$

$=x[2]h[4-2]+x[3]h[4-3]$

$=0+1=1$

when $n=5$ $\quad y[5]=\sum_{k=-\infty}^{\infty} x[k]h[5-k]$

$=x[3]h[5-3]=0$

[**2.a**] (5 marks) Consider a discrete linear time-invariant system T with discrete input signal $x(n)$ and impulse response $h(n)$. Recall that the impulse response of a discrete system is defined as the output of the system when the input is an impulse function $\delta(n)$, i.e. $T[\delta(n)] = h(n)$, where:

$$\delta(n) = \begin{cases} 1, & \text{if } n = 0, \\ 0, & \text{else.} \end{cases}$$

Prove that $T[x(n)] = h(n) * x(n)$, where $*$ denotes convolution operation.

Hint: represent signal $x(n)$ as a function of $\delta(n)$.

$$x(n) = \sum_{k=-\infty}^{\infty} x[k]\delta(n-k)$$

Since T is a LTI system, response to $x(n)$

$$T[x(n)] = T\left[\sum_{k=-\infty}^{\infty} x[k]\delta(n-k)\right]$$

By linearity $T[x(n)] = \sum_{k=-\infty}^{\infty} x[k]\, T[\delta(n-k)]$

For a LTI system, $T[\delta(n-k)] = h(n-k)$

$$T[x(n)] = \sum_{k=-\infty}^{\infty} x[k]h(n-k)$$

$$= h(n) * x(n)$$

[**2.b**] (5 marks) Is Gaussian blurring linear? Is it time-invariant? Make sure to include your justifications.

Gaussian blurring is a filtering operation with each output pixel being a weighted sum of input pixel values where weights are given by Gaussian kernel. It convolves an image with a Gaussian kernel.

Gaussian kernel: $f(x,y) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}$

$$g(x,y) = f(x,y) * I(x,y)$$

$$g(x,y) = \sum_{u=-k}^{k}\sum_{v=-k}^{k} f(u,v)\cdot I(x-u, y-v)$$

where $f(x,y)$ is gaussian kernel
$I(x,y)$ is input image
$g(x,y)$ is blurred output image.

Check if Gaussian kernel satisfies two properties of linearity:

① Additivity: $T[f_1(x,y) + f_2(x,y)] = T[f_1(x,y)] + T[f_2(x,y)]$

② Homogeneity: $T[af_1(x,y)] = aT[f_1(x,y)]$

Check for ①:

$$T[f_1(x,y) + f_2(x,y)] = (f_1(x,y) + f_2(x,y)) * I(x,y)$$
$$= f_1(x,y) * I(x,y) + f_2(x,y) I(x,y)$$
$$= T[f_1(x,y)] + T[f_2(x,y)]$$

b/c convolution is distributive

So, additivity holds.

Check for ②:

$$T[af(x,y)] = (af(x,y)) * I(x,y)$$
$$= a(f(x,y)) * I(x,y) \quad \text{by scalar multiplication property of convolution.}$$
$$= aG[f(x,y)]$$

So, homogeneity holds.

Thus, Gaussian blurring is linear.

Check if Gaussian blurring is time invariant:

↳ check if shifting the input results in an equivalent shift of the output.

$$T[f(x-a, x-b)] = g(x-a, y-b)$$

Left side : $T[f(x-a, x-b)] = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u-a, v-b) \cdot I(x-u, y-v)$

Right side : $g(x-a, y-b) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u,v) \cdot I(x-a-u, y-b-v)$

$$= \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u,v) \cdot I(x-(a+u), y-(b+v))$$

Let $i = u-a$, $j = v-b$, substitute to left side

$$\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i,j) \cdot I(x-(i+a), y-(j+b))$$

Left side = Right side

Thus, Gaussian blurring is time invariant

[**2.c**] (5 marks) Is time reversal, i.e. $T[x(n)] = x(-n)$, linear? Is it time-invariant? Make sure to include your justifications.

Check if time reversal satisfies the two properties of linearity:

① Additivity : $T[x_1(n)+x_2(n)] = T[x_1(n)] + T[x_2(n)]$

② Homogeneity : $T[ax(n)] = aT[x(n)]$

Check for ①:

Left side : $T[x_1(n)+x_2(n)] = x_1(-n) + x_2(-n)$

Right side : $T[x_1(n)] + T[x_2(n)] = x_1(-n) + x_2(-n)$

Left side = Right side

So, additivity holds.

Check for ②:

Left side : $T[ax(n)] = ax(-n)$

Right side : $aT[x(n)] = ax(-n)$

Left side = Right side

So, homogeneity holds.

Thus, time reversal is linear.

Check if time reversal is time-invariant:

$T[x(n-n_0)] = g(n-n_0)$

Left side: $T[x(n-n_0)] = x(-(n-n_0)) = x(-n+n_0)$

Right side: $g(n-n_0) = x(-n-n_0)$

Left side $\neq$ Right side

Thus, time reversal is not time-invariant.

[Question 3] **Polynomial Multiplication and Convolution (15 marks)**

Vectors can be used to represent polynomials. For example, 3^{rd}-degree polynomial $(a_3x^3 + a_2x^2 + a_1x + a_0)$ can by represented by vector $[a_3, a_2, a_1, a_0]$.
If $\mathbf{u}$ and $\mathbf{v}$ are vectors of polynomial coefficients, prove that convolving them is equivalent to multiplying the two polynomials they each represent.

Hint: You need to assume proper zero-padding to support the full-size convolution.

Two polynomials represented as vectors:

$u = [u_3, u_2, u_1, u_0] \Rightarrow u_3x^3 + u_2x^2 + u_1x + u_0$

$V = [V_3, V_2, V_1, V_0] \Rightarrow V_3x^3 + V_2x^2 + V_1x + V_0$

When we multiply these polynomials, the degree of result will be $3+3=6$.
We need zero padding to handle this in convolution

u-padded $= [u_3, u_2, u_1, u_0, 0, 0, 0]$

V-padded $= [V_3, V_2, V_1, V_0, 0, 0, 0]$

Convolution formula: $y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$

$y[0] = u_3V_3 \rightarrow$ coefficient of x^6

$y[1] = u_3V_2 + u_2V_3 \rightarrow$ coefficient of x^5

$y[2] = u_3V_1 + u_2V_2 + u_1V_3 \rightarrow$ coefficient of x^4

$y[3] = u_3V_0 + u_2V_1 + u_1V_2 + u_0V_3 \rightarrow$ coefficient of x^3

$y[4] = u_2V_0 + u_1V_1 + u_0V_2 \rightarrow$ coefficient of x^2

$y[5] = u_1V_0 + u_0V_1 \rightarrow$ coefficient of x^1

$y[6] = u_0V_0 \rightarrow$ coefficient of x^0

polynomial multiplication $(u_3x^3 + u_2x^2 + u_1x + u_0)$ with $(V_3x^3 + V_2x^2 + V_1x + V_0)$

x^6 term: $u_3V_3x^6$

x^5 term: $(u_3V_2 + u_2V_3)x^5$

x^4 term: $(u_3V_1 + u_2V_2 + u_1V_3)x^4$

x^3 term: $(u_3V_0 + u_2V_1 + u_1V_2 + u_0V_3)x^3$

x^2 term: $(u_2V_0 + u_1V_1 + u_0V_2)x^2$

x^1 term: $(u_1V_0 + u_0V_1)x^1$

x^0 term: $u_0V_0x^0$

The coefficients from polynomial multiplication exactly match the values computed by convolution at each position.
Thus, convolving coefficient vectors is equivalent to multiplying the two polynomials they represent.

[Question 4] Laplacian Operator (20 marks)

The Laplace operator is a second-order differential operator in the "n"-dimensional Euclidean space, defined as the divergence (∇) of the gradient (∇f). Thus if f is a twice-differentiable real-valued function, then the Laplacian of f is defined by:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

where the latter notations derive from formally writing:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right).$$

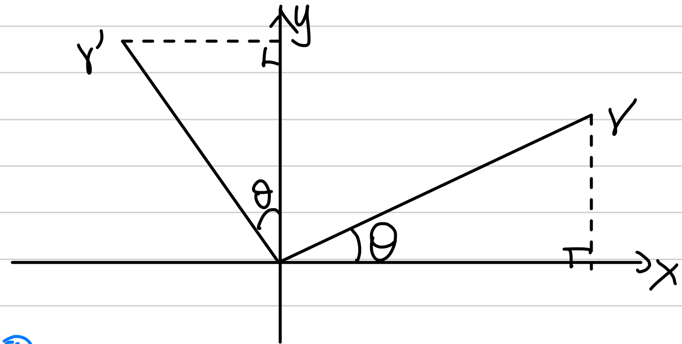
Now, consider a 2D image $I(x, y)$ and its Laplacian, given by $\Delta I = I_{xx} + I_{yy}$. Here the second partial derivatives are taken with respect to the directions of the variables x, y associated with the image grid for convenience. Show that the Laplacian is in fact rotation invariant. In other words, show that $\Delta I = I_{rr} + I_{r'r'}$, where r and r' are any two orthogonal directions.

Hint: Start by using polar coordinates to describe a chosen location (x, y). Then use the chain rule.

Establish the relationship between (x,y) and (r,r')

$x = r\cos(\theta) - r'\sin(\theta)$

$y = r\sin(\theta) + r'\cos(\theta)$



Using chain rule

$$\frac{d}{dr} = \frac{d}{dx}\underbrace{\frac{dx}{dr}}_{\cos\theta} + \frac{d}{dy}\underbrace{\frac{dy}{dr}}_{\sin\theta}$$

$$\frac{d}{dr} = \cos\theta\frac{d}{dx} + \sin\theta\frac{d}{dy} \quad \text{—— ①}$$

$$\frac{d}{dr'} = \frac{d}{dx}\underbrace{\frac{dx}{dr'}}_{-\sin\theta} + \frac{d}{dy}\underbrace{\frac{dy}{dr'}}_{\cos\theta}$$

$$\frac{d}{dr'} = -\sin\theta\frac{d}{dx} + \cos\theta\frac{d}{dy} \quad \text{—— ②}$$

$$I_{rr} = \frac{d}{dr}\left(\frac{d}{dr}I\right) = \frac{d}{dr}\left(\cos\theta\frac{dI}{dx} + \sin\theta\frac{dI}{dy}\right)$$

$$= \frac{d}{dr}\left(\cos\theta\frac{dI}{dx}\right) + \frac{d}{dr}\left(\sin\theta\frac{dI}{dy}\right)$$

plug into equation ①

$$= \cos\theta\frac{d}{dx}\left(\cos\theta\frac{dI}{dx}\right) + \sin\theta\frac{d}{dy}\left(\cos\theta\frac{dI}{dx}\right) + \cos\theta\frac{d}{dx}\left(\sin\theta\frac{dI}{dy}\right) + \sin\theta\frac{d}{dy}\left(\sin\theta\frac{dI}{dy}\right)$$

$$= \cos^2\theta\frac{d}{dx}\frac{d}{dx}I + \sin\theta\cos\theta\frac{d}{dy}\frac{d}{dx}I + \cos\theta\sin\theta\frac{d}{dx}\frac{d}{dy}I + \sin^2\theta\frac{d}{dy}\frac{d}{dy}I$$

$$= \cos^2\theta I_{xx} + \sin^2\theta I_{yy} + 2\sin\theta\cos\theta I_{xy}$$

$$I_{r'r'} = \frac{d}{dr'}\left(\frac{d}{dr'} I\right) = \frac{d}{dr'}\left(-\sin\theta \frac{dI}{dx} + \cos\theta \frac{dI}{dy}\right)$$

$$= \frac{d}{dr'}\left(-\sin\theta \frac{dI}{dx}\right) + \frac{d}{dr'}\left(\cos\theta \frac{dI}{dy}\right)$$

plug into equation ②

$$= -\sin\theta \frac{d}{dx}\left(-\sin\theta \frac{dI}{dx}\right) + \cos\theta \frac{d}{dy}\left(-\sin\theta \frac{dI}{dx}\right) + (-\sin\theta)\frac{d}{dx}\left(\cos\theta \frac{dI}{dy}\right) + \cos\theta \frac{d}{dy}\left(\cos\theta \frac{dI}{dy}\right)$$

$$= \sin^2\theta \frac{d}{dx}\frac{dI}{dx} - \cos\theta\sin\theta \frac{d}{dy}\frac{dI}{dx} - \sin\theta\cos\theta \frac{d}{dx}\frac{dI}{dy} + \cos^2\theta \frac{d}{dy}\frac{dI}{dy}$$

$$= \sin^2\theta I_{xx} - 2\cos\theta\sin\theta I_{xy} + \cos^2\theta I_{yy}$$

$$\Delta I = I_{rr} + I_{r'r'}$$

$$= \cos^2\theta I_{xx} + \sin^2\theta I_{yy} + \cancel{2\sin\theta\cos\theta I_{xy}} + \sin^2\theta I_{xx} \cancel{- 2\cos\theta\sin\theta I_{xy}} + \cos^2\theta I_{yy}$$

$$= (\cos^2\theta + \sin^2\theta) I_{xx} + \underbrace{(\cos^2\theta + \sin^2\theta)}_{=1} I_{yy}$$

$$= I_{xx} + I_{yy}$$

[Question 5] Canny Edge Detector Robustness

Analysis on the performance of the edge detector as a function of noise variance:

1. Original Image and Edges (sigma = 0):

The original edge detection shows clear, continuous contours. Key features are well-defined,such as the person's outline, tripod structure, and background elements.High contrast boundaries are accurately captured.

2. Low Noise (sigma = 10):

Edge detection remains largely robust. Only minor degradation compared to the original. Most significant edges, like the person, tripod, are preserved. Slight increase in small, spurious edge fragments. The overall structure and object boundaries remain clearly visible

3. Medium Noise (sigma = 50):

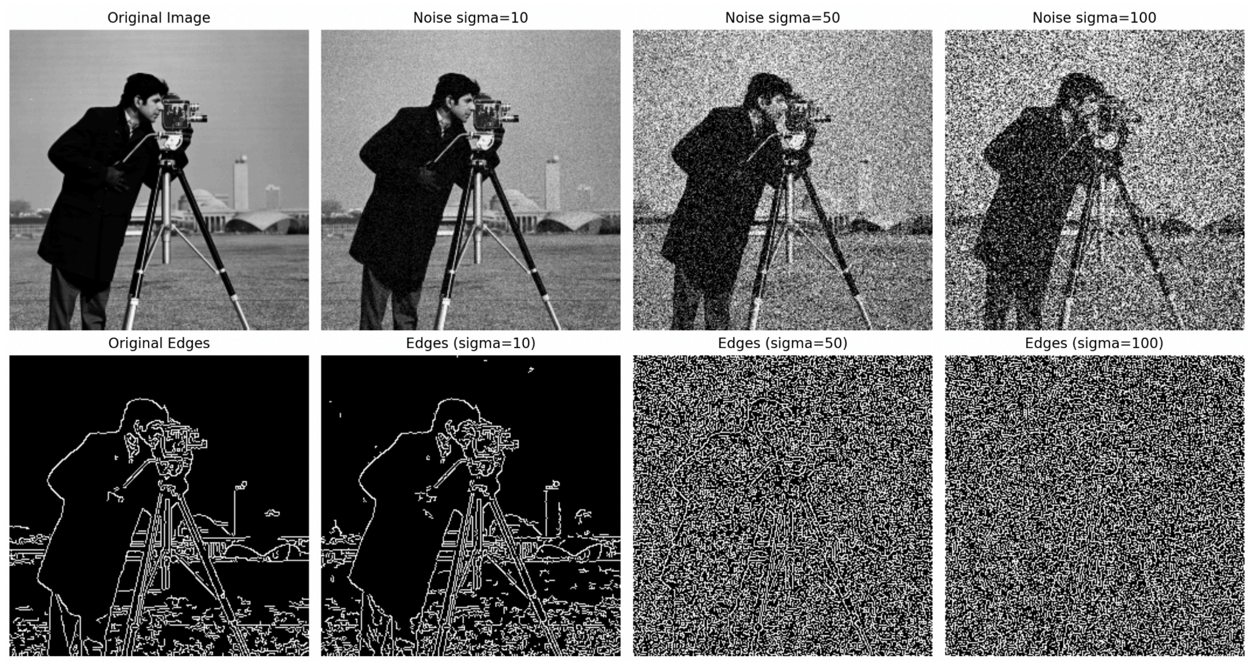
Significant deterioration in edge detection quality. Image is overwhelmed with false edge responses due to noise. Original structure is barely distinguishable. True edges begin to blend with noise-induced edges. The signal-to-noise ratio has dropped dramatically.

4. High Noise (sigma = 100):

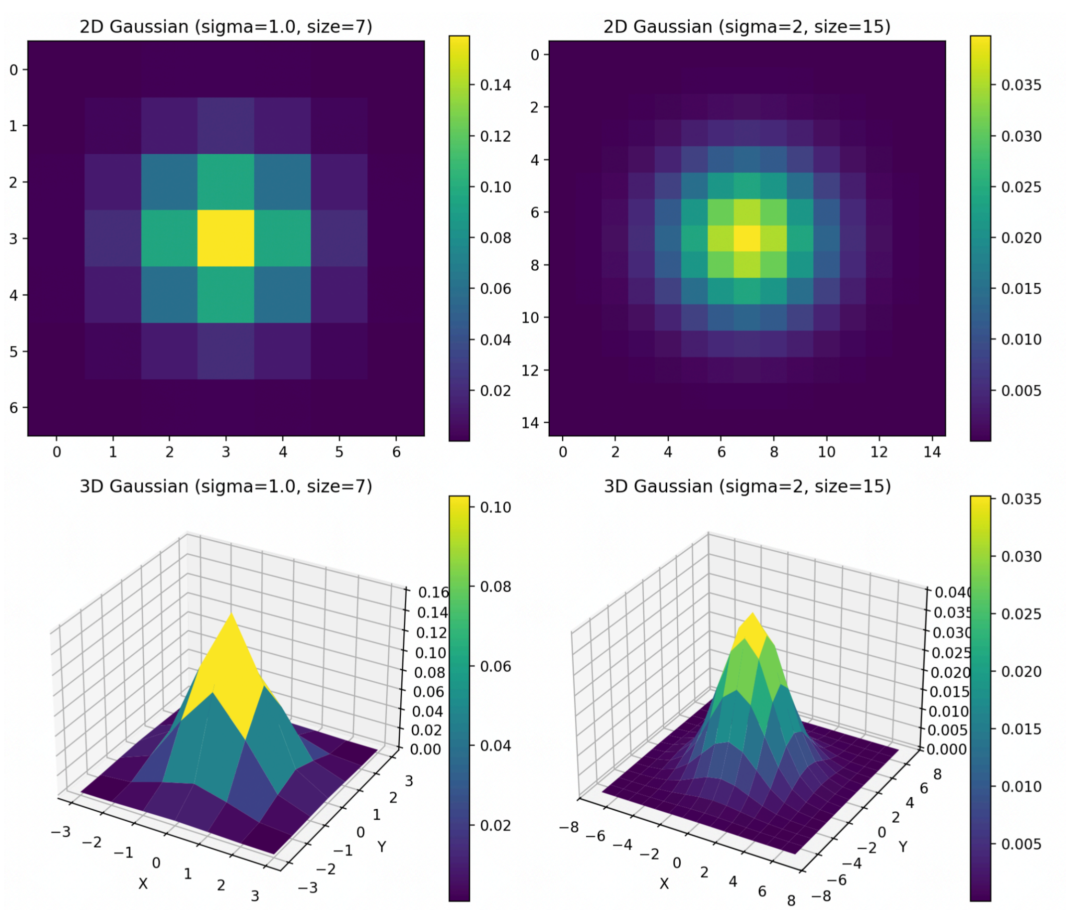
Edge detection has almost completely failed. The output is dominated by noise-induced edges. Original image structure is almost completely lost. Nearly impossible to distinguish true edges from noise.

Observations:

Without pre-filtering or parameter adjustment, the standard Canny detector fails to be useful beyond moderate noise levels. Additionally, strong edges like the person’s outline persist longer than fine details as noise increases



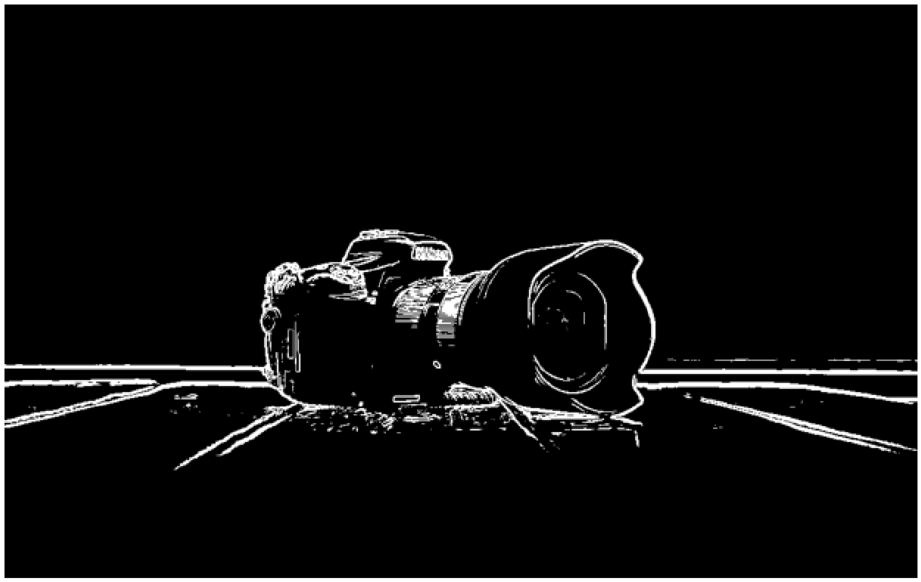
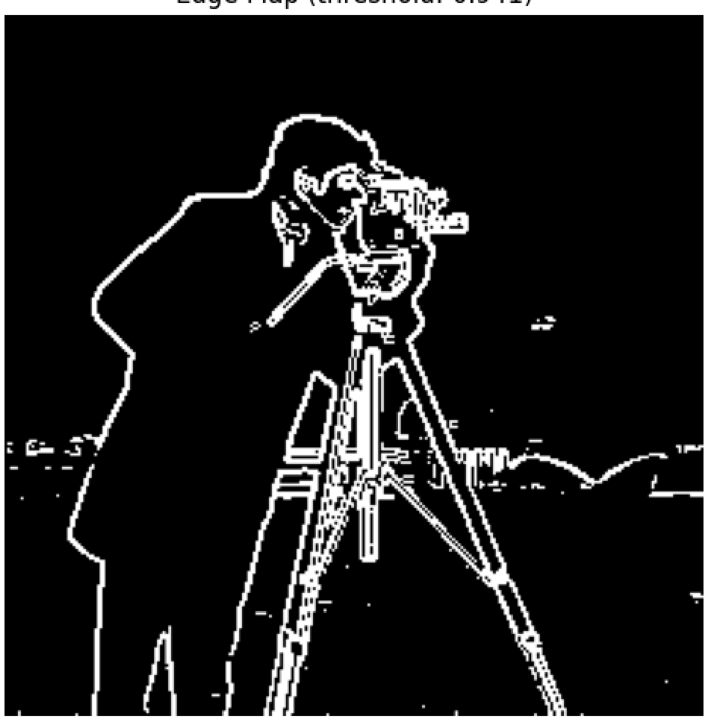
[Question 6] Edge Detection
Step I - Gaussian Blurring:



Step II - Gradient Magnitude:

My Image	
Image Provided	

Step III - Threshold Algorithm:

My Image	Edge Map (threshold: 0.340) 
Image Provided	Edge Map (threshold: 0.941) 

Step IV:

The algorithm demonstrates key strengths. It successfully captures strong edges in both images, particularly evident in the camera's lens shape and the photographer's outline. The adaptive thresholding approach helps separate real edges from background noise. Fine details are also preserved well, such as the camera's control buttons and the tripod's leg segments. However, the algorithm has some limitations. In my image, the reflective surface shows noise in the edge map, indicating that the threshold is too low for handling small intensity changes. The algorithm shows greater sensitivity to horizontal and vertical edges compared to diagonal ones due to the Sobel operator's design, which shows up in the weaker tripod edges.