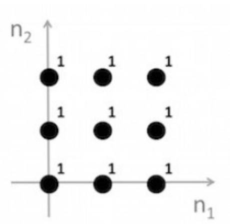
1. After applying spatial filtering to an image, you find that the output image looks more blurry than the original image, i.e., some details like sharp edges are lost. Based on this description, the filter applied is most likely to be which of the following types?

**1 point**

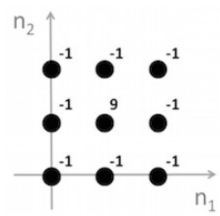
- ( ) A high-pass filter
- (x) A low-pass filter
- ( ) A band-pass filter
- ( ) A band-stop filter

2. Which one of the following impulse responses acts a high-pass filter?

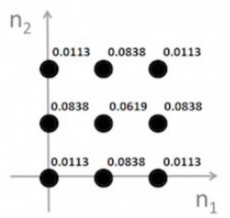
- ( )



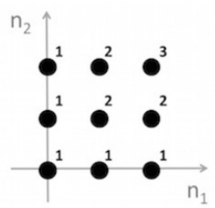

- (x)




- ( )

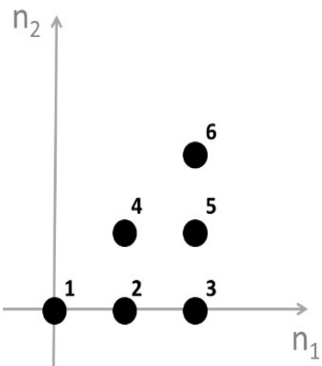



- ( )

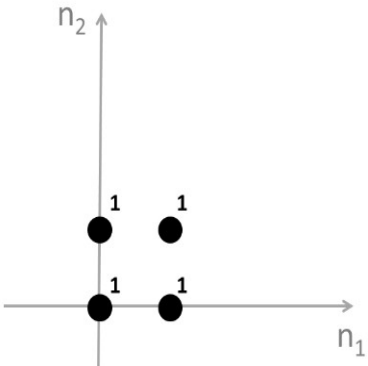



3. What is the linear convolution of $s_1(n_1, n_2)$ and $s_2(n_1, n_2)$?

**1 point**

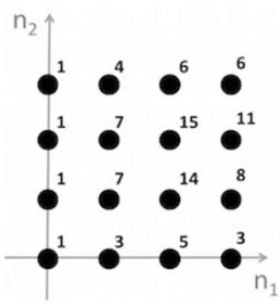



$S_1(n_1,n_2)$



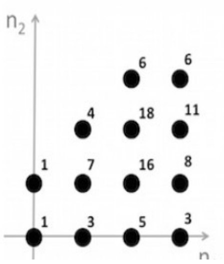

$S_2(n_1,n_2)$

- ( )

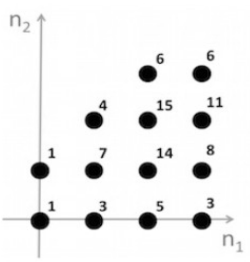



- ( ) None of the above.

- ( )




- (x)



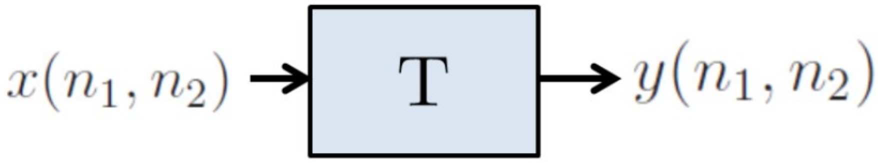
**4.** A linear shift-invariant system is fully characterizable by its impulse response.

1 point

- (•) True
- ( ) False

**5.** Check all the statements that apply to any linear shift-invariant system $T$:

1 point




- [x] If the output to $x(n_1, n_2) = \delta(n_1, n_2)$ is known, it is possible to find the output to any other input.
- [x] The output to $x(n_1, n_2) = e^{j(\omega_1 n_1 + \omega_2 n_2)}$ is always proportional to the input, i.e., $y(n_1, n_2) = C\,x(n_1, n_2)$ where $C$ is a complex constant.
- [ ] It is possible that the zero input (i.e., $x(n_1, n_2) = 0$) results in a non-zero output (i.e., $y(n_1, n_2) \neq 0$).
- [ ] If $y(n_1, n_2) = 0$ then $x(n_1, n_2) = 0$.

**6.** The regions of support of two images $x(n_1, n_2)$ and $y(n_1, n_2)$ are given respectively by $\mathcal{S}_x = \{(n_1, n_2) |\, 0 \leq n_1 \leq P_1 - 1,\ 0 \leq n_2 \leq P_2 - 1)\}$ and $\mathcal{S}_y = \{(n_1, n_2) |\, 0 \leq n_1 \leq Q_1 - 1,\ 0 \leq n_2 \leq Q_2 - 1)\}$. Which of the following statements is true regarding the linear convolution of $x(n_1, n_2)$ and $y(n_1, n_2)$, i.e., $z(n_1, n_2) = x(n_1, n_2) \star\star y(n_1, n_2)$.

1 point

- ( ) $z(n_1, n_2)$ is always non-zero over $\mathcal{S}_z = \{(n_1, n_2) |\, 0 \leq n_1 \leq P_1 + Q_1 - 1,\ 0 \leq n_2 \leq P_2 + Q_2 - 1)\}$.
- ( ) $z(n_1, n_2)$ is always non-zero over $\mathcal{S}_z = \{(n_1, n_2) |\, 0 \leq n_1 \leq P_1 + Q_1 - 2,\ 0 \leq n_2 \leq P_2 + Q_2 - 2)\}$.
- ( ) $z(n_1, n_2)$ is always zero outside $\mathcal{S}_z = \{(n_1, n_2) |\, 0 \leq n_1 \leq P_1 + Q_1 - 1,\ 0 \leq n_2 \leq P_2 + Q_2 - 1)\}$.
- (•) $z(n_1, n_2)$ is always zero outside $\mathcal{S}_z = \{(n_1, n_2) |\, 0 \leq n_1 \leq P_1 + Q_1 - 2,\ 0 \leq n_2 \leq P_2 + Q_2 - 2)\}$.

7. In this problem and the next, you will implement spatial-domain low-pass filtering using MATLAB, and evaluate the difference between the filtered image and the original image using two quantitative metrics called Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR). Given two $N_1 \times N_2$ images $x(n_1, n_2)$ and $y(n_1, n_2)$, the MSE is computed as $MSE = \frac{1}{N_1 N_2} \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} [x(n_1, n_2) - y(n_1, n_2)]^2$.

1 point

The PSNR is defined as $PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$, where $MAX_I$ is the maximum possible pixel value of the images. For the 8-bit gray-scale images considered in this problem, $MAX_I = 255$.

Follow the instructions below to finish this problem.

(1) Download the original image from here. The original image is a $256 \times 256$ 8-bit gray-scale image.

(2) Convert the original image from type 'uint8' (8-bit integer) to 'double' (real number).

(3) Create a $3 \times 3$ low-pass filter with all coefficients equal to 1/9, i.e., create a $3 \times 3$ MATLAB array with all elements equal to 1/9.

(4) Low-pass filter the original image (converted to type 'double') with the filter created in step (3). This can be done using the built-in MATLAB function "imfilter". The function "imfilter" takes three arguments and returns one output. The first argument is the original image (converted to type 'double'); the second argument is the low-pass filter created in step (3); and the third argument is a string specifying the boundary filtering option. For this problem, use 'replicate' (including the single quotes) for the third argument. The output of the function "imfilter" is the filtered image.

(5) Compute the PSNR value between the original image (converted to type 'double') and the filtered image by using the formulae given above.

Enter the PSNR value (up to two decimal points).

29.29

8. Repeat steps (3) through (5) in the previous question, this time using a $5 \times 5$ low-pass filter with all coefficients equal to 1/25. Enter the PSNR value (up to two decimal points).

1 point

25.73