ASSIGNMENT-04(A)

6.4 Enlarge the profile 1 4 7 4 3 6 to lengths of (i) 9 (ii) 11 and (iii) 15 pixels, using (a) nearest-neighbor and (b) linear interpolation.

Ans.

6.5 Enlarge the image

8 8 13 9

1 13 1 15

5 4 7 7

5 10 3 7

to sizes (i) 7 x 7 (ii) 8 x 8 (iii) 10 x 10, by hand, using (a) nearest-neighbor and (b) bilinear interpolation

Ans.

(i) a.

	C1.0	C1.5	C2.0	C2.5	C3.0	C3.5	C4.0
R1.0	**8**	**8**	**8**	**13**	**13**	**9**	**9**
R1.5	**1**	**13**	**13**	**1**	**1**	**15**	**15**
R2.0	**1**	**13**	**13**	**1**	**1**	**15**	**15**
R2.5	**5**	**4**	**4**	**7**	**7**	**7**	**7**
R3.0	**5**	**4**	**4**	**7**	**7**	**7**	**7**
R3.5	**5**	**10**	**10**	**3**	**3**	**7**	**7**
R4.0	**5**	**10**	**10**	**3**	**3**	**7**	**7**

b. A 2-stage interpolation

Interpolation along the rows-

	C1.0	C1.5	C2.0	C2.5	C3.0	C3.5	C4.0
R1	**8**	**8**	**8**	**10.5**	**13**	**11**	**9**
R2	**1**	**7**	**13**	**7**	**1**	**8**	**15**
R3	**5**	**4.5**	**4**	**5.5**	**7**	**7**	**7**
R4	**5**	**7.5**	**10**	**6.5**	**3**	**5**	**7**

Interpolation along the columns-

	C1.0	C1.5	C2.0	C2.5	C3.0	C3.5	C4.0
R1.0	**8**	**8**	**8**	**11**	**13**	**11**	**9**
R1.5	**5**	**8**	**11**	**9**	**7**	**10**	**12**
R2.0	**1**	**7**	**13**	**7**	**1**	**8**	**15**
R2.5	**3**	**6**	**9**	**6**	**4**	**8**	**11**
R3.0	**5**	**5**	**4**	**6**	**7**	**7**	**7**
R3.5	**5**	**6**	**7**	**6**	**5**	**6**	**7**
R4.0	**5**	**8**	**10**	**7**	**3**	**5**	**7**

(ii) a.

	C1.00	C1.43	C1.86	C2.29	C2.71	C3.14	C3.57	C4.00
R1.00	**8**	**8**	**8**	**8**	**13**	**13**	**9**	**9**
R1.43	**8**	**8**	**8**	**8**	**13**	**13**	**9**	**9**
R1.86	**1**	**1**	**13**	**13**	**1**	**1**	**15**	**15**
R2.29	**1**	**1**	**13**	**13**	**1**	**1**	**15**	**15**
R2.71	**5**	**5**	**4**	**4**	**7**	**7**	**7**	**7**
R3.14	**5**	**5**	**4**	**4**	**7**	**7**	**7**	**7**
R3.57	**5**	**5**	**10**	**10**	**3**	**3**	**7**	**7**
R4.00	**5**	**5**	**10**	**10**	**3**	**3**	**7**	**7**

b. Along the rows-

	C1.00	C1.43	C1.86	C2.29	C2.71	C3.14	C3.57	C4.00
R1	**8**	**8**	**8**	**9.43**	**11.57**	**12.43**	**10.71**	**9**
R2	**1**	**6.14**	**11.29**	**9.57**	**4.43**	**3**	**9**	**15**
R3	**5**	**4.57**	**4.14**	**4.86**	**6.14**	**7**	**7**	**7**
R4	**5**	**7.14**	**9.29**	**8**	**5**	**3.57**	**5.29**	**7**

Down the columns-

	C1.00	C1.43	C1.86	C2.29	C2.71	C3.14	C3.57	C4.00
R1.00	**8**	**8**	**8**	**9**	**12**	**12**	**11**	**9**
R1.43	**5**	**7**	**9**	**9**	**9**	**8**	**10**	**12**
R1.86	**5**	**6**	**11**	**10**	**5**	**4**	**9**	**14**
R2.29	**2**	**6**	**9**	**8**	**5**	**4**	**8**	**13**
R2.71	**4**	**5**	**6**	**6**	**6**	**6**	**8**	**9**
R3.14	**5**	**5**	**5**	**5**	**6**	**7**	**7**	**7**
R3.57	**5**	**6**	**7**	**7**	**5**	**5**	**6**	**7**
R4.00	**5**	**7**	**9**	**8**	**5**	**4**	**5**	**7**

(iii) a.

	C1.00	C1.33	C1.66	C2.00	C2.33	C2.66	C3.00	C3.33	C3.66	C4.00
R1.00	**8**	**8**	**8**	**8**	**8**	**13**	**13**	**13**	**9**	**9**
R1.33	**8**	**8**	**8**	**8**	**8**	**13**	**13**	**13**	**9**	**9**
R1.66	**1**	**1**	**13**	**13**	**13**	**1**	**1**	**1**	**15**	**15**
R2.00	**1**	**1**	**13**	**13**	**13**	**1**	**1**	**1**	**15**	**15**
R2.33	**1**	**1**	**13**	**13**	**13**	**1**	**1**	**1**	**15**	**15**
R2.66	**5**	**5**	**4**	**4**	**4**	**7**	**7**	**7**	**7**	**7**
R3.00	**5**	**5**	**4**	**4**	**4**	**7**	**7**	**7**	**7**	**7**
R3.33	**5**	**5**	**4**	**4**	**4**	**7**	**7**	**7**	**7**	**7**
R3.66	**5**	**5**	**10**	**10**	**10**	**3**	**3**	**3**	**7**	**7**
R4.00	**5**	**5**	**10**	**10**	**10**	**3**	**3**	**3**	**7**	**7**

b. Along the rows-

	C1.00	C1.33	C1.66	C2.00	C2.33	C2.66	C3.00	C3.33	C3.66	C4.00
R1	**8**	**8**	**8**	**8**	**9.66**	**11.33**	**13**	**11.66**	**10.33**	**9**
R2	**1**	**5**	**9**	**13**	**9**	**5**	**1**	**5.66**	**10.33**	**15**
R3	**5**	**4.66**	**4.33**	**4**	**5**	**6**	**7**	**7**	**7**	**7**
R4	**5**	**6.66**	**8.33**	**10**	**7.66**	**5.33**	**3**	**4.33**	**5.66**	**7**

Down the columns-

	C1.00	C1.33	C1.66	C2.00	C2.33	C2.66	C3.00	C3.33	C3.66	C4.00
R1.00	**8**	**8**	**8**	**8**	**10**	**11**	**13**	**12**	**10**	**9**
R1.33	**6**	**7**	**8**	**10**	**9**	**9**	**9**	**10**	**10**	**11**
R1.66	**3**	**6**	**9**	**11**	**9**	**7**	**5**	**8**	**10**	**13**
R2.00	**1**	**5**	**9**	**13**	**9**	**5**	**1**	**6**	**10**	**15**
R2.33	**2**	**5**	**7**	**10**	**8**	**5**	**3**	**6**	**9**	**12**
R2.66	**4**	**5**	**6**	**7**	**6**	**6**	**5**	**7**	**8**	**10**
R3.00	**5**	**5**	**4**	**4**	**5**	**6**	**7**	**7**	**7**	**7**
R3.33	**5**	**5**	**6**	**6**	**6**	**6**	**6**	**6**	**7**	**7**
R3.66	**5**	**6.**	**7**	**8**	**7**	**6**	**4**	**5**	**6**	**7**
R4.00	**5**	**7**	**8**	**10**	**8**	**5**	**3**	**4**	**6**	**7**

6.6 Suppose an image is scaled upwards in size by a factor k, and the result is then scaled downwards in size by the same factor. Is the final result exactly the same as the original image? If not, why not? What if the image is reduced in size first, and the result enlarged?

Ans. It's not going to be the same this time. Scaling upwards would fill in the vacant pixels with the values of the nearest pixels neighboring them, using the nearest-neighbor technique. When the image is shrunk, however, the nearest-neighbor algorithm may replace the incorrect pixels, i.e., the pixels that were originally in the image may be replaced by pixels formed during the expansion. We should get the same result if k is an even integer (e.g. extending a 4 x 4 matrix to an 8 x 8, then down again), but not for all other values of k. Different pixel values are generated when bilinear interpolation is utilized in the expansion, and the original values are not regenerated while shrinking. It is even less likely that the outcome will be the same as the original image if the image is reduced first and then expanded. This can only happen if you use nearest-neighbor interpolation on an image where the pixel values repeat over a distance of k pixels. The only caveat is that this, too, is dependent on the algorithm.

6.7 Suppose an image is rotated and then the result rotated back by the same amount (using either (i) nearest neighbor or (ii) bilinear interpolation). Is the resulting image exactly the same as the original? If not, why not?

Ans.

6.8 What is the result of convolving the following (symmetric) one-dimensional masks? (In each case you should slide one (M1) past the other (M2), doing a sum-of-products, replacing the center term (pixel) of M2, and then shifting M1 by a single term (pixel).

Ans.

6.8

(i) $\begin{vmatrix} 1 & 1 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 1 & 1 \end{vmatrix} = [1 \; 2 \; 3 \; 2 \; 1]$

(ii) $\begin{vmatrix} 1 & 1 & 1 \end{vmatrix} * \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix} = [1 \; 1 \; 1; 1 \; 1 \; 1; 1 \; 1 \; 1]$

(iii) $\begin{vmatrix} 1 & 2 & 1 \end{vmatrix} * \begin{vmatrix} 1 \\ 2 \\ 1 \end{vmatrix} = [-1 \; -1 \; -1; 0 \; 0 \; 0; 1 \; 1 \; 1]$

(weighted average)

(iv) $\begin{vmatrix} 1 & 1 & 1 \end{vmatrix} * \begin{vmatrix} -1 \\ 0 \\ 1 \end{vmatrix} = [-1 \; -1 \; -1; 0 \; 0 \; 0; 1 \; 1 \; 1]$

(Prewitt - horizontal edges)

(v) $\begin{vmatrix} -1 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix} = [-1 \; 0 \; 1; -1 \; 0 \; 1; -1 \; 0 \; 1]$

[Prewitt - horizontal edges)

(vi) $\begin{vmatrix} -1 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 \\ 2 \\ 1 \end{vmatrix} = [-1\ 0\ 1;\ -2\ 0\ 2;\ -1\ 0\ 1]$

(Sobel – vertical edges)

(vii) $\begin{vmatrix} 1 & 2 & 1 \end{vmatrix} * \begin{vmatrix} -1 \\ 0 \\ 1 \end{vmatrix} = [-1\ -2\ -1,\ 0\ 0\ 0,\ 1\ 2\ 1]$

(Sobel – horizontal edges)

(Note: the results of (iv) and (v) give the Prewitt operator and the results of (vi) and (vii) give the Sobel operator).

6.9 Applying a 3×3 averaging mask twice does not produce the same result as applying a 5×5 averaging mask once. To what is it equivalent?

Ans. Applying a 3 x 3 averaging mask twice produces the same result as applying a mask once, which is the 3 × 3 mask convolutioned with itself. The following mask is created by combining a 3 x 3 averaging mask with itself:

1 2 3 2 1

2 4 6 4 2

3 6 9 6 3

2 4 6 4 2

3 2 3 2 1

this is not the same as a 5 x 5 (straight) average mask; it is a 5 x 5 weighted average mask.

6.10 Is the 3 x 3 median mask separable (i.e., can it be implemented by a 3 x 1 mask followed by a 1 x 3 mask?). Explain your answer.

Ans. No, because it's non-linear, it's not separable. To get the correct median value with the 3 x 3 median mask, you'll need all 9 values.

6.11 Can unsharp masking be used to reverse the effect of blurring? Choose an image and apply an unsharp mask after a 3 x 3 averaging mask. Describe the result.

Ans. Just a little bit. To restore the original and reverse the blurring, the blurring and sharpening effects would have to be the inverse of each other (as in the Wiener filter).

6.12 An averaging mask is applied to an image to reduce noise, and then a Laplacian mask is applied to the result to enhance small details. Would the result be the same if the order of these operations were reversed? Explain.

Ans. These aren't commutative operations at all. The outcome will be different if their sequence is reversed.