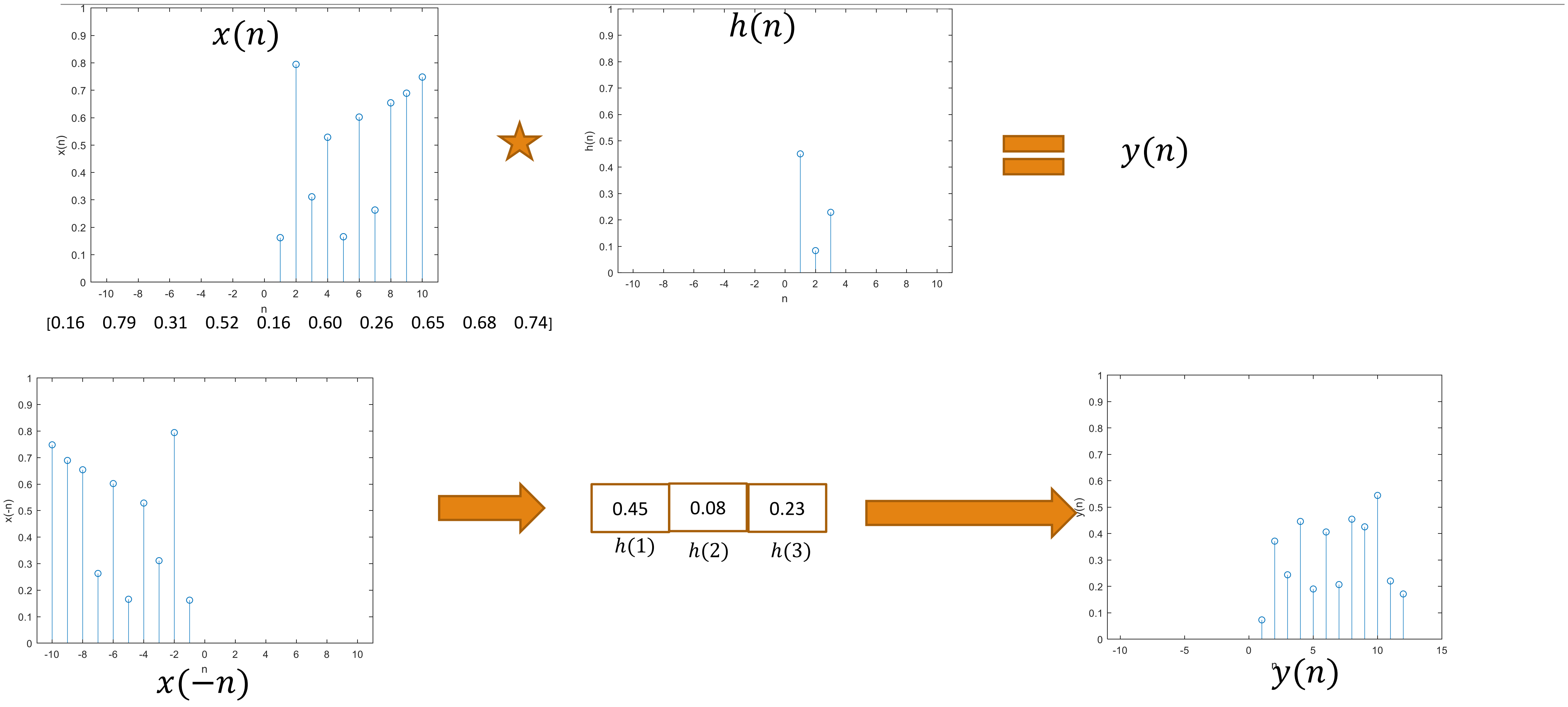
Review and examples

CAP 5415, FALL 2019

Filtering and Convolution



Convolution arises from the process by which a signal propagates through a series of weights (a pipe)

- When the pipe is completely filled, the first sample of the signal is aligned with the last weight
- This is equivalent to a *flipping* the indexed order of the signal samples (i.e. the signal is flipped w.r.t. to the weights).

Convolution Example

$x(n) = 1, 3, 2, 5, 8$ $\quad x(-n) = 8, 5, 2, 3, 1$

$h(n) = 1, 0, 2, 2, 1$

$x(n) = 0, 0, 0, 0, 1, 3, 2, 5, 8, 0, 0, 0, 0$ (zero padded)

n -4 -3 -2 -1 0 1 2 3 4 5 6 7 8

$h(n) = 1, 0, 2, 2, 1$

$L = 5$

$$y(k) = \sum_{n=0}^{L-1} x(k-n)h(n)$$

$$\begin{bmatrix} 1\,0\,0\,0\,0 \\ 3\,1\,0\,0\,0 \\ 2\,3\,1\,0\,0 \\ 5\,2\,3\,1\,0 \\ 8\,5\,2\,3\,1 \\ 0\,8\,5\,2\,3 \\ 0\,0\,8\,5\,2 \\ 0\,0\,0\,8\,5 \\ 0\,0\,0\,0\,8 \end{bmatrix}_{\boldsymbol{X}} \begin{bmatrix} 1 \\ 0 \\ 2 \\ 2 \\ 1 \end{bmatrix}_{\boldsymbol{h}} = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 13 \\ 19 \\ 17 \\ 28 \\ 21 \\ 8 \end{bmatrix}_{\boldsymbol{y}}$$

$y(k) = 1, 3, 4, 13, 19, 17, 28, 21, 8$

Convolution can be implemented as a matrix vector multiplication

Two dimensional convolution

$$y(k,l) = \sum_{n}\sum_{m} x(k-m, l-n)h(m,n)$$

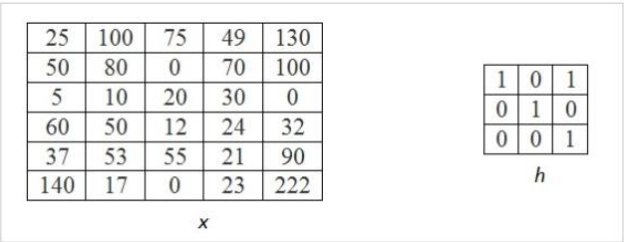


Figure 1: Input matrices, where x represents the original image and h represents the kernel. Image created by Sneha H.L.

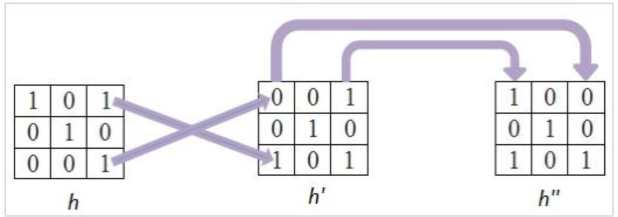


Figure 2: Pictorial representation of matrix inversion. Image created by Sneha H.L.

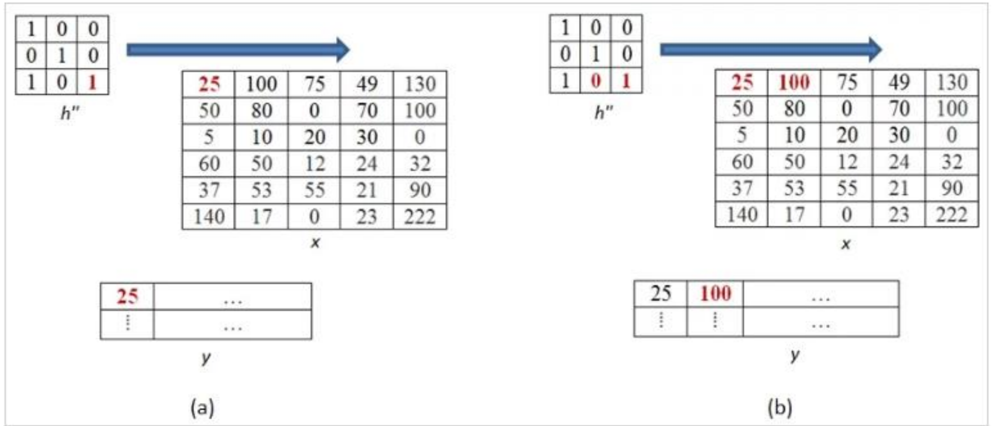


Figure 3a, 3b. Convolution results obtained for the output pixels at location (1,1) and (1,2). Image created by Sneha H.L.

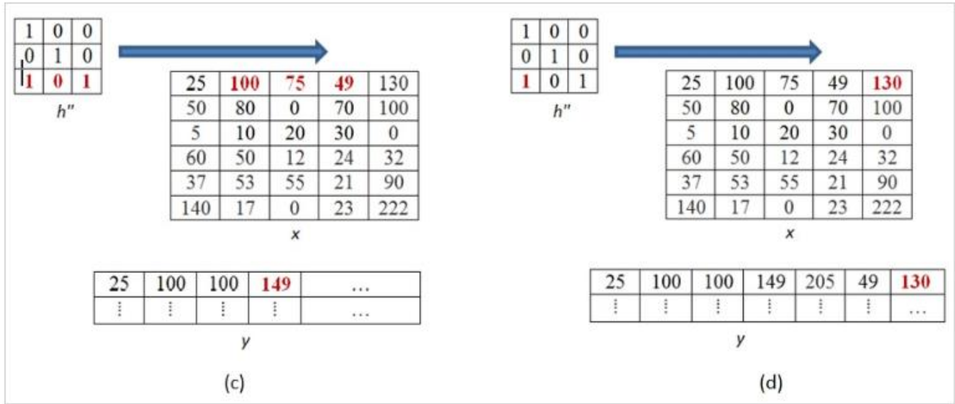


Figure 3c, 3d: Convolution results obtained for the output pixels at location (1,4) and (1,7). Image created by Sneha H.L.

The filter (or kernel) $h(m,n)$ is first flipped in both axes before the "shift-multiply-accumulate" operation.

https://www.allaboutcircuits.com/technical-articles/two-dimensional-convolution-in-image-processing/

Two dimensional convolution

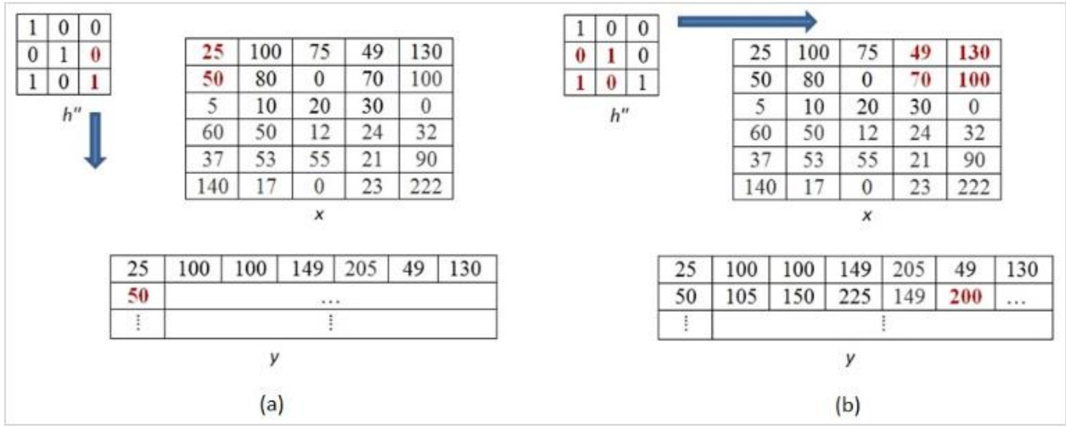


Figure 4a, 4b. *Convolution results obtained for the output pixels at location (2,1) and (2,6). Image created by Sneha H.L.*

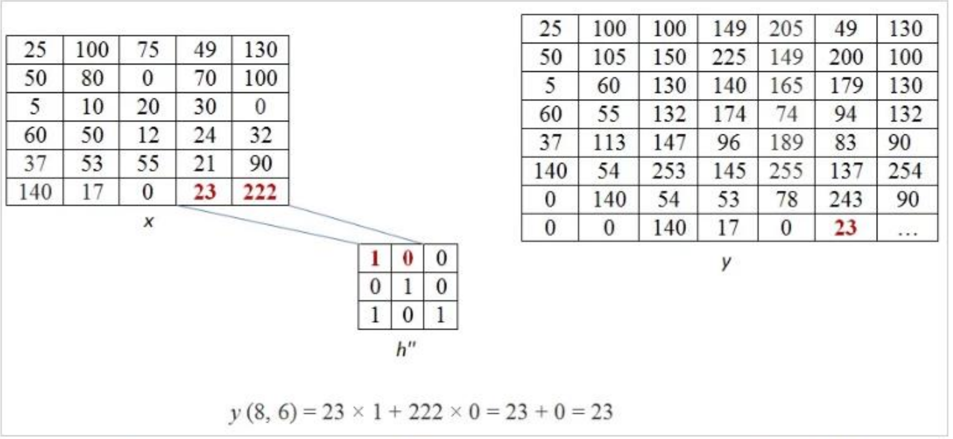


Figure 5c. *Convolution results obtained for the output pixels at (8,6). Image created by Sneha H.L.*

25	100	100	149	205	49	130
50	105	150	225	149	200	100
5	60	130	140	165	179	130
60	55	132	174	74	94	132
37	113	147	96	189	83	90
140	54	253	145	255	137	254
0	140	54	53	78	243	90
0	0	140	17	0	23	255

y

Figure 6. *Our example's resulting output matrix. Image created by Sneha H.L.*

Exercise: Write a code to implement this example

Correlation Example

$$x(n) = 1, 3, 2, 5, 8$$

$$h(n) = 1, 3, 0, 5, 8$$

$$x(n) = 0, 0, 0, 0, 1, 3, 2, 5, 8, 0, 0, 0, 0$$ (zero padded)

n: -4 -3 -2 -1 0 1 2 3 4 5 6 7 8

$$h(n) = 1, 3, 0, 5, 8$$

$L = 5$

$$y(k) = \sum_{n=0}^{L-1} x(k+n)h(n)$$

$$\underbrace{\begin{bmatrix} 8&0&0&0&0 \\ 5&8&0&0&0 \\ 2&5&8&0&0 \\ 3&2&5&8&0 \\ 1&3&2&5&8 \\ 0&1&3&2&5 \\ 0&0&1&3&2 \\ 0&0&0&1&3 \\ 0&0&0&0&1 \end{bmatrix}}_{\boldsymbol{X}} \underbrace{\begin{bmatrix} 1 \\ 3 \\ 0 \\ 5 \\ 8 \end{bmatrix}}_{\boldsymbol{h}} = \underbrace{\begin{bmatrix} 8 \\ 29 \\ 17 \\ 49 \\ 99 \\ 53 \\ 31 \\ 29 \\ 8 \end{bmatrix}}_{\boldsymbol{y}}$$

Largest Value Occurs when There is maximum Match between The signals

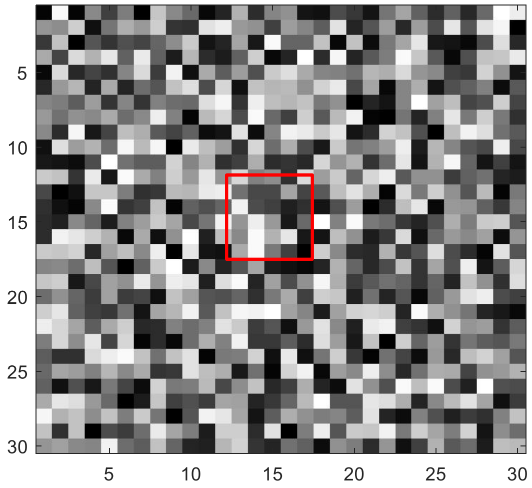
$$y(k) = 8,29,17,49,99,53,31,29,8$$

- Correlation measures how much two signals match
- Same process as convolution, but without reversing the signal

Image Correlations

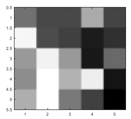
Peak intensity occurs where the two patterns match the best

$x(m,n)$



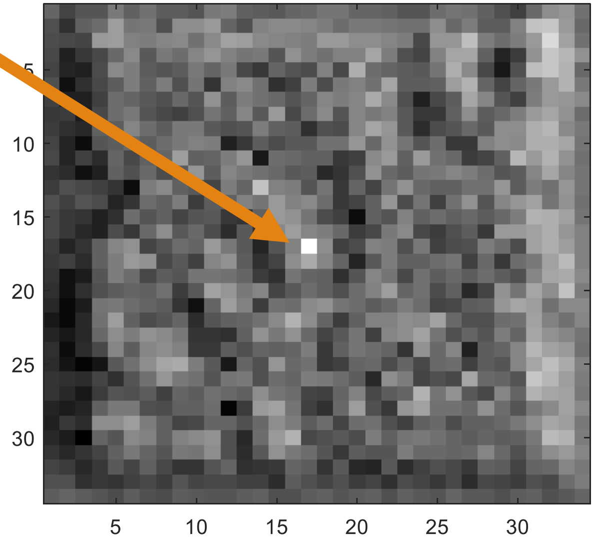
$\otimes$

$h(m,n)$



=

$y(m,n)$

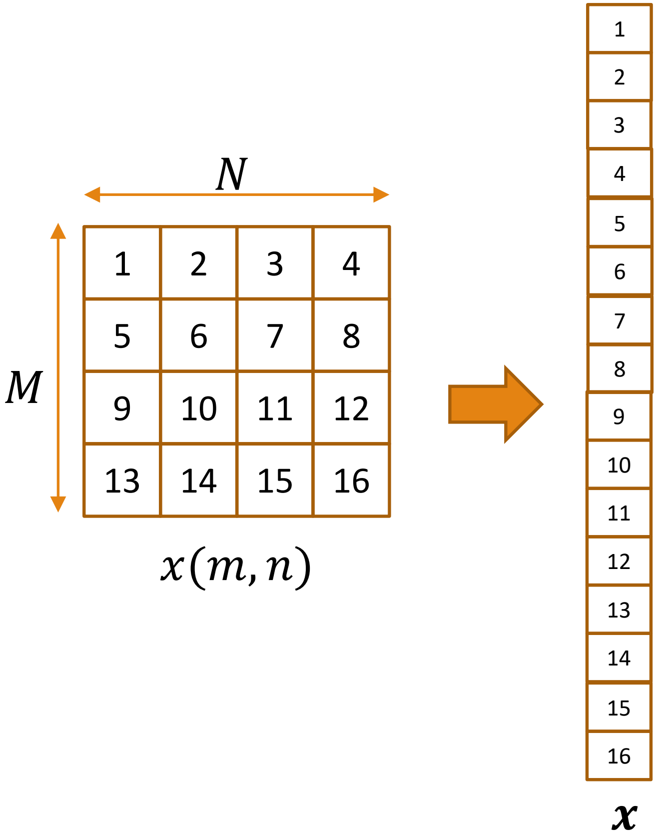


$$y(m,n) = \sum_k \sum_l x(m+k, n+l)h(k,l)$$

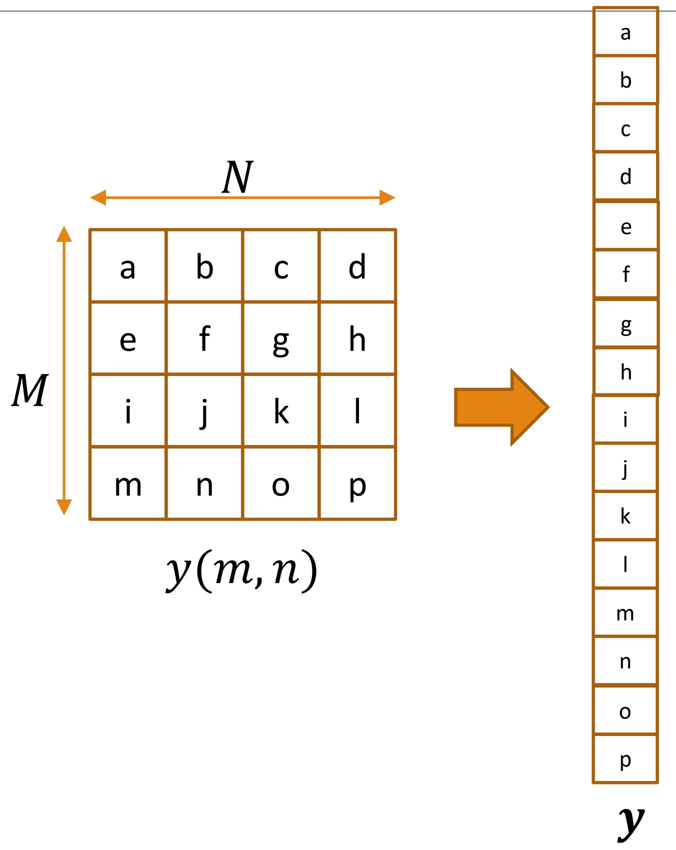
The 2D correlation is similar to 2D convolution except the filter (or kernel) is not flipped.

Maximum value occurs at the location of the “best match”

Vector Representation of Images



$MN \times 1$ vector
of elements x_k



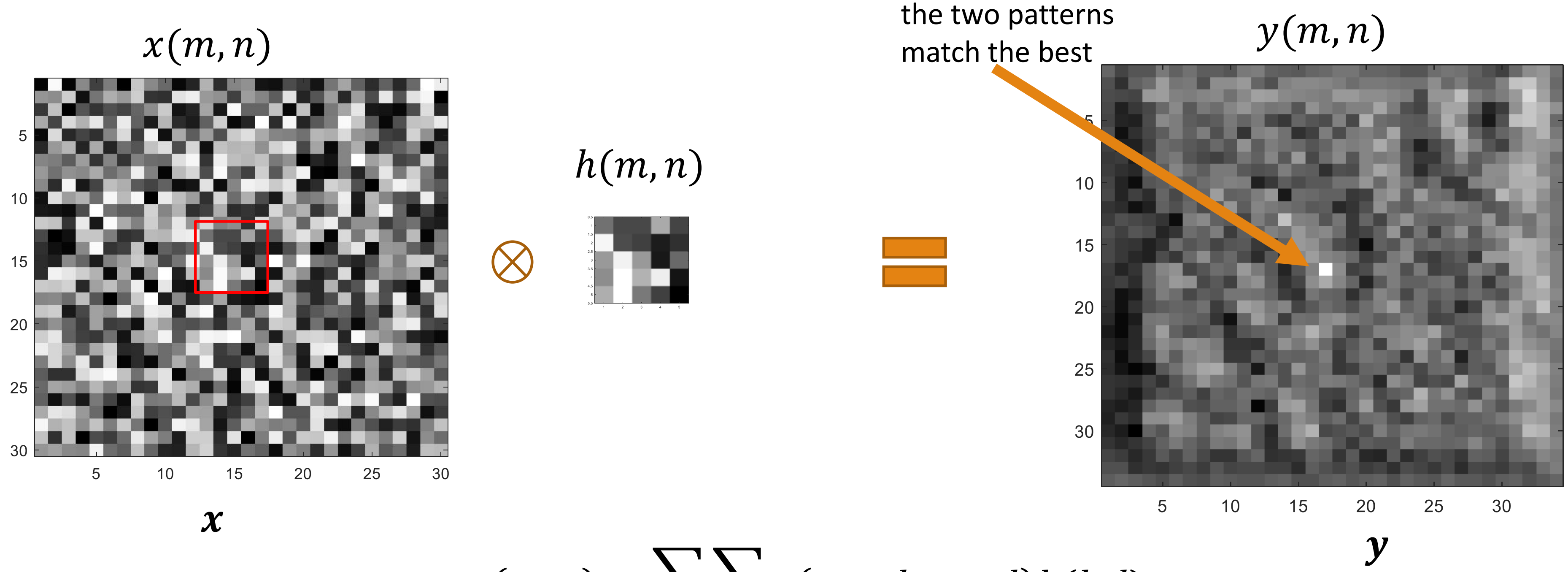
$MN \times 1$ vector
of elements y_k

$$\boldsymbol{x}^T\boldsymbol{y} = \sum_k x_k y_k$$

$$= \sum_m \sum_n x(m,n)y(m,n)$$

- Images can be represented as vectors by lexicographic reordering of the columns
- The sum of the product of the pixels of two images can be represented as an inner product of the two image vectors

Image Correlations

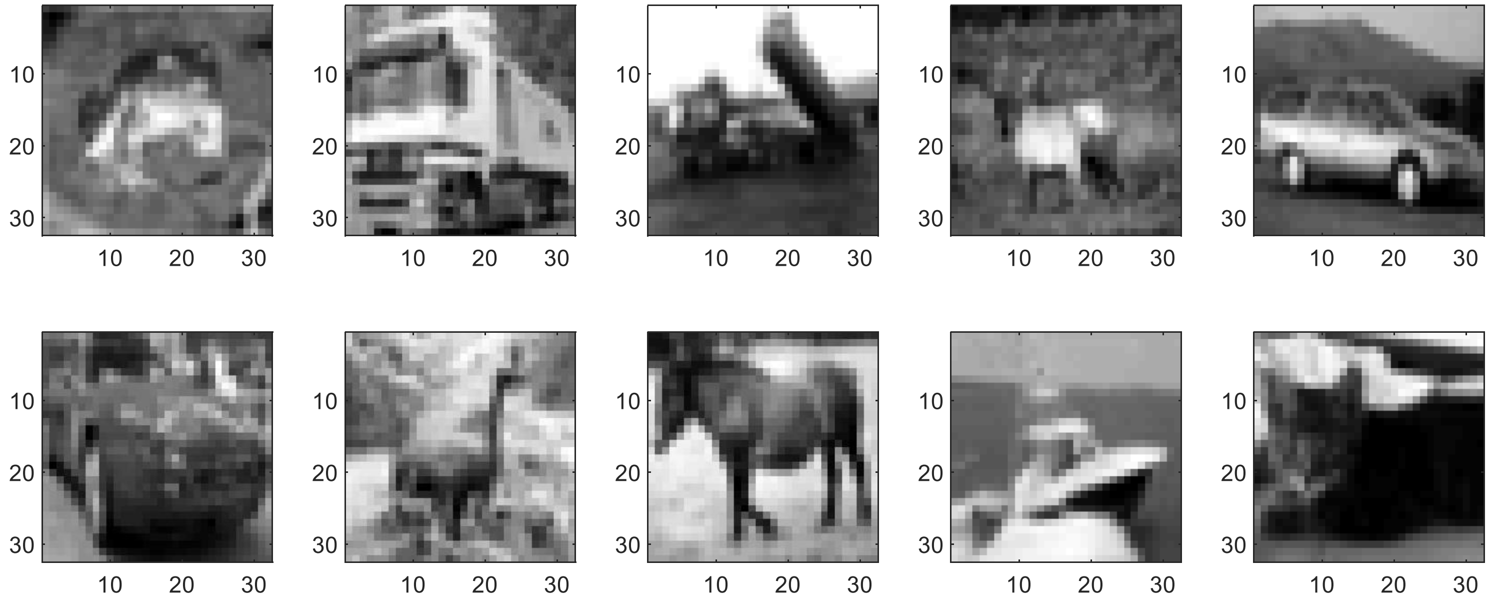


$$y(m,n) = \sum_k \sum_l x(m+k, n+l)h(k,l)$$

$$y(0,0) = \sum_k \sum_l x(k,l)h(k,l) = \boldsymbol{x}^T \boldsymbol{y}$$

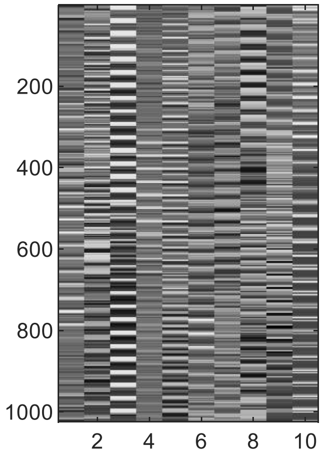
- The value at the center of the cross-correlation (y(0,0)) can be represented as the inner product of the two image vectors

Example of Matrix vector computation using images

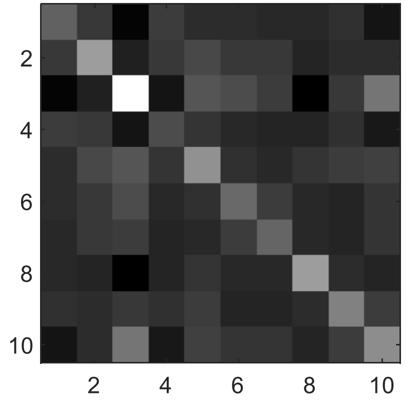


- Find a linear combination of the above images such that the result is identical to the image of the horse (i.e. the 8th image).
  - We know that all the weights should be zero except for the 8th weight which should be 1.0

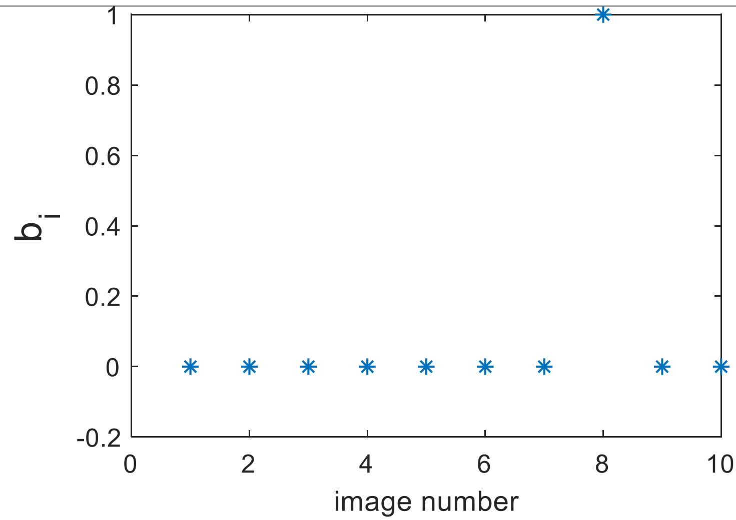
Illustration of MMSE solution using images



$\boldsymbol{A}$



$\boldsymbol{A}^T\boldsymbol{A}$



$\boldsymbol{b}$

- Problem: Find a set of weights b_i such that $\sum_{i=1}^{10} b_i x_i(m,n) = x_8(m,n)$
- Step 1: reshape each image $x_i(m,n)$ in a 1024 x 1 vector $\boldsymbol{x}_i$
- Step 2: Define a matrix $\boldsymbol{A}$ of size 1024 x 10, with each of the image vectors as its columns,
  - i.e. $\boldsymbol{A} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \qquad\qquad \boldsymbol{x}_{10}]$
- Step 3: Compute $\boldsymbol{b} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{x}_8$
  - Note that all elements of $\boldsymbol{b}$ are zero except for b_8