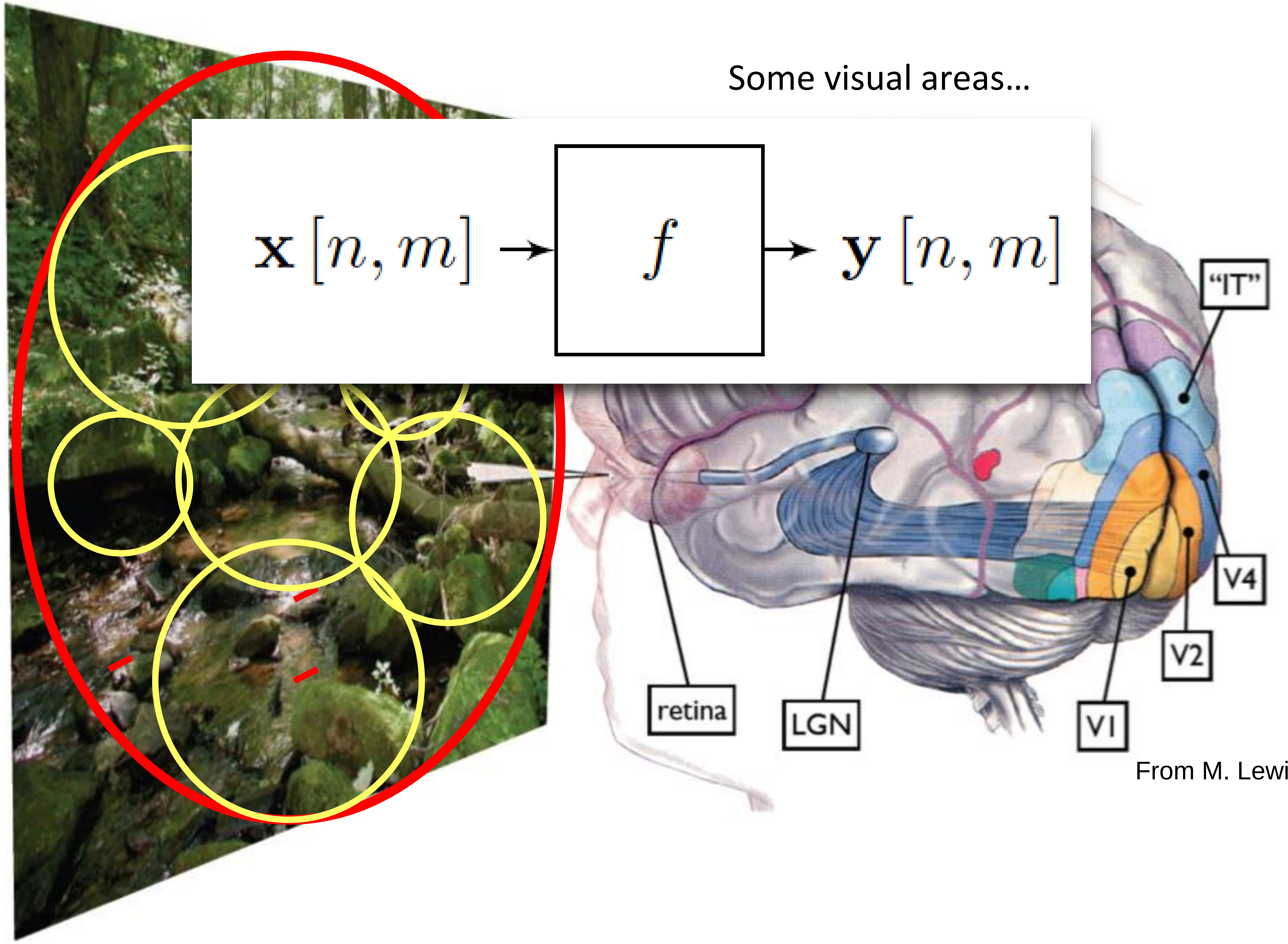
Linear Systems

Alexei Efros, CS280, Spring 2024

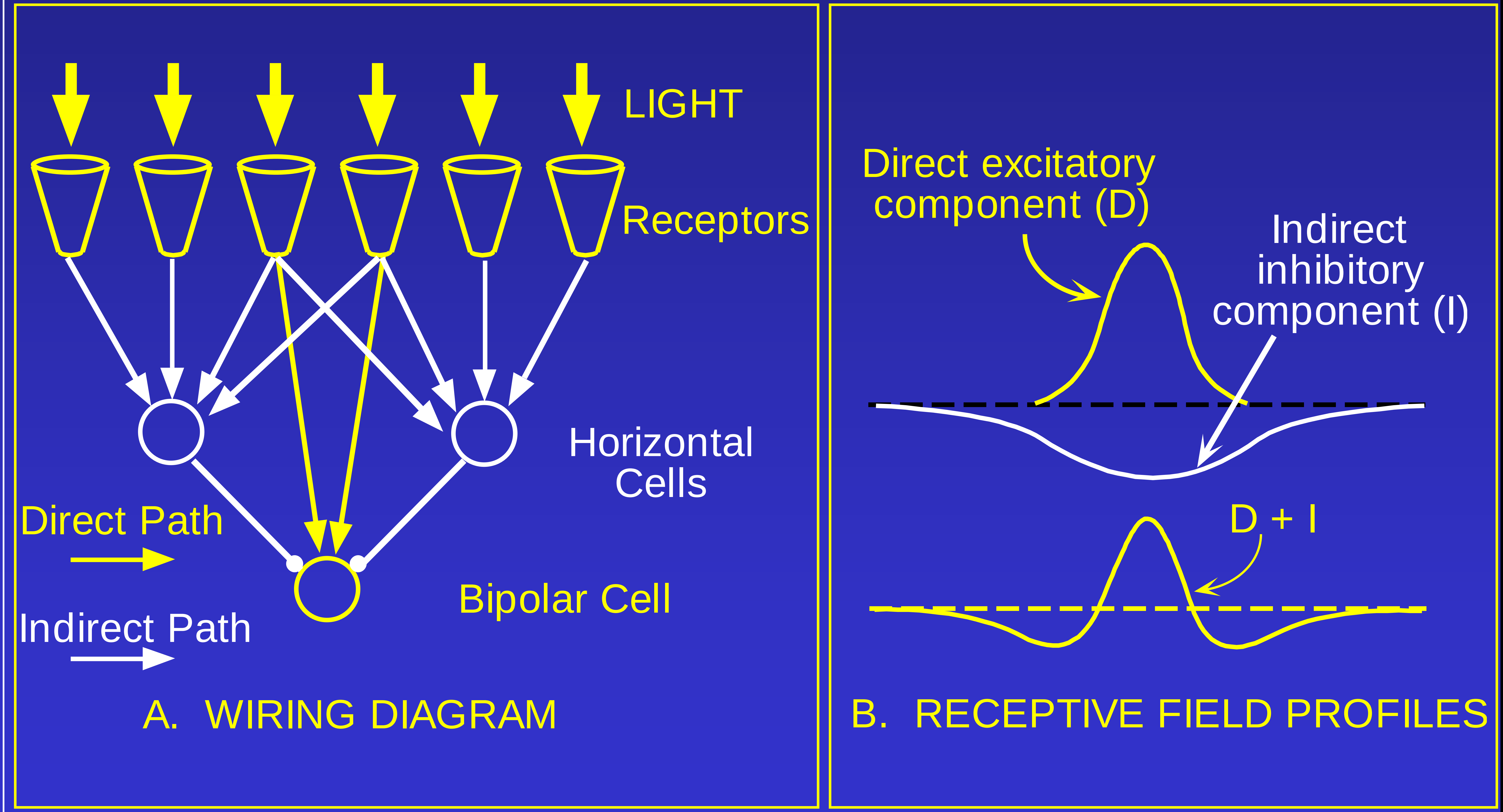
Some visual areas...



From M. Lewicky

Retinal Receptive Fields

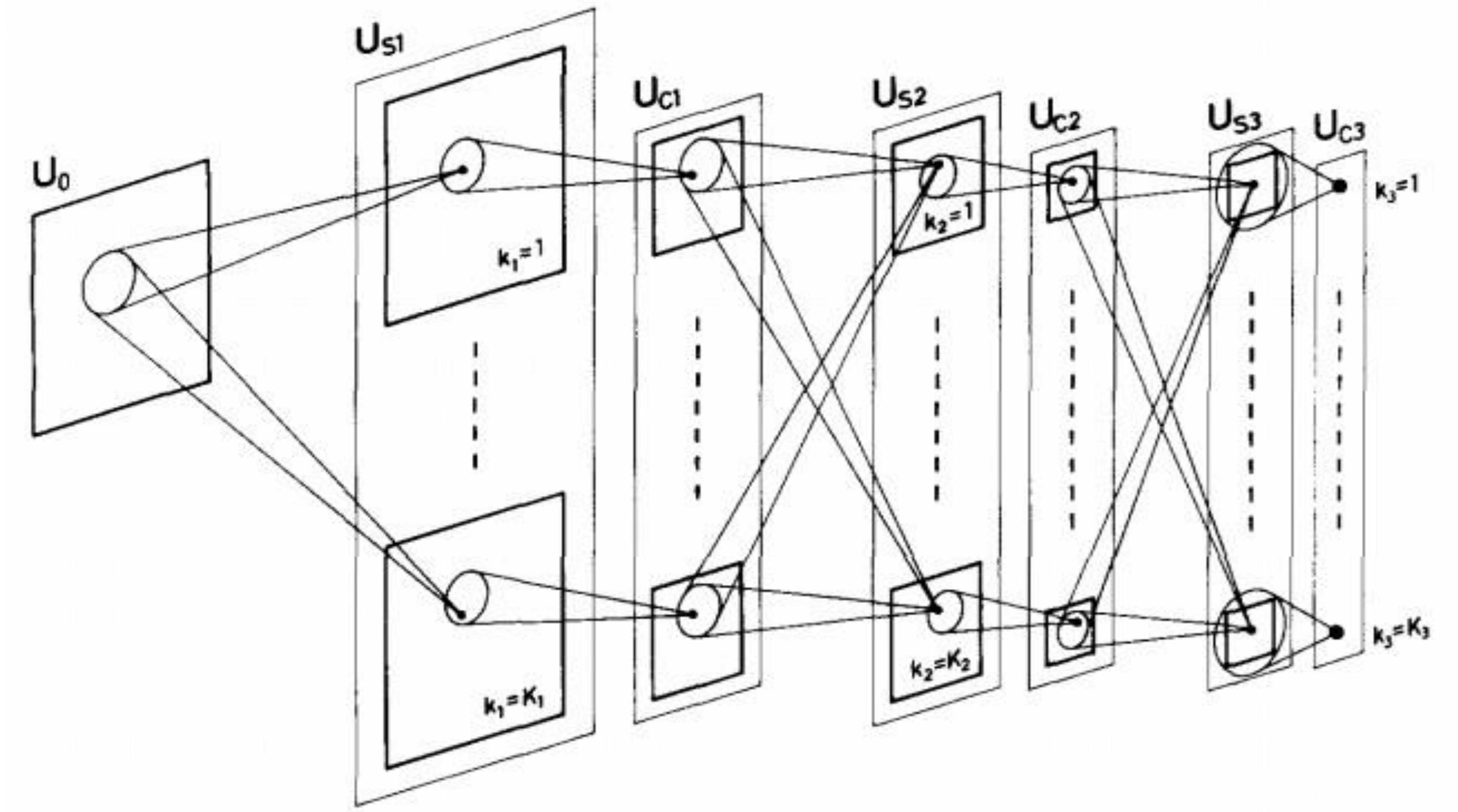
Receptive field structure in bipolar cells

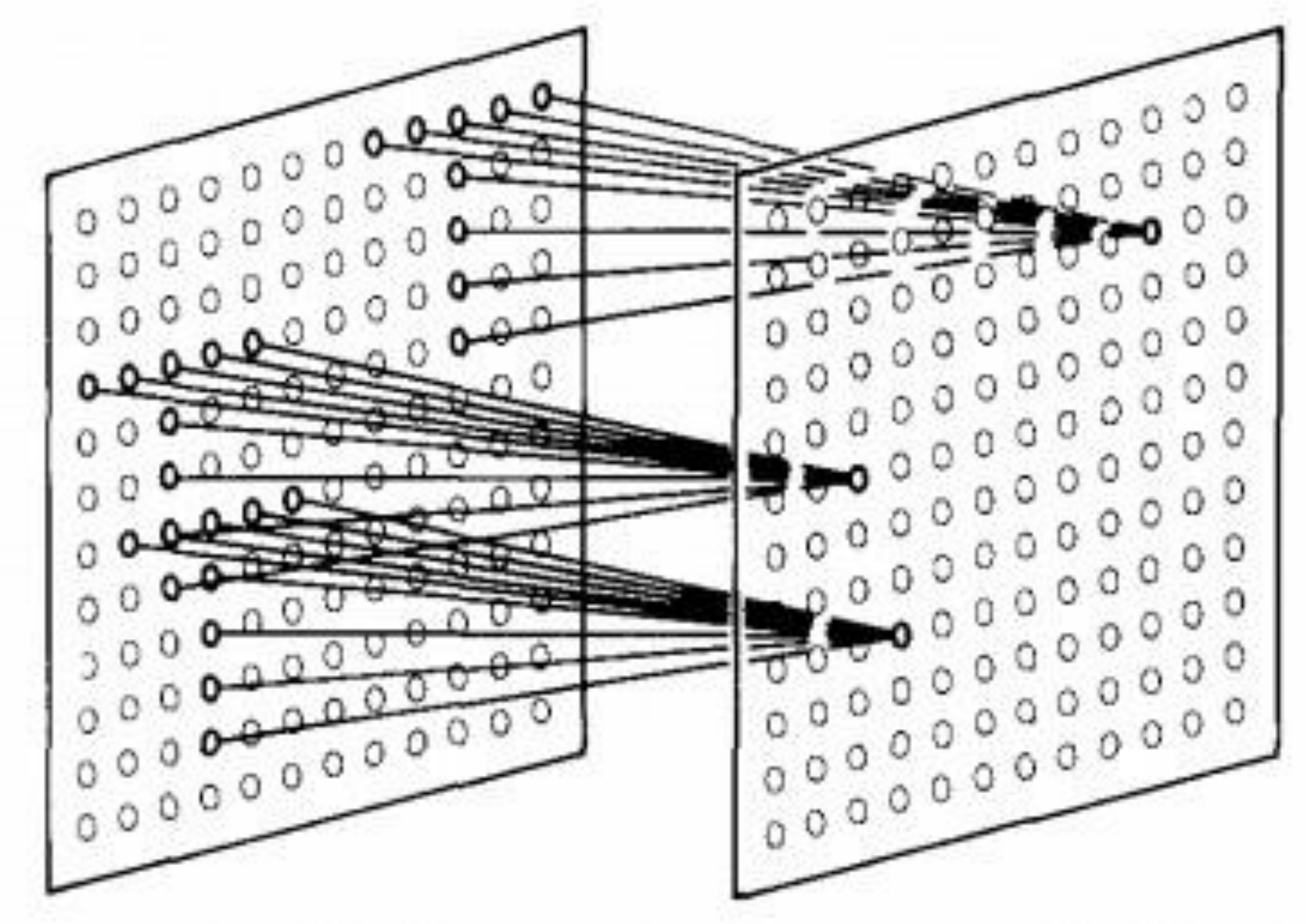


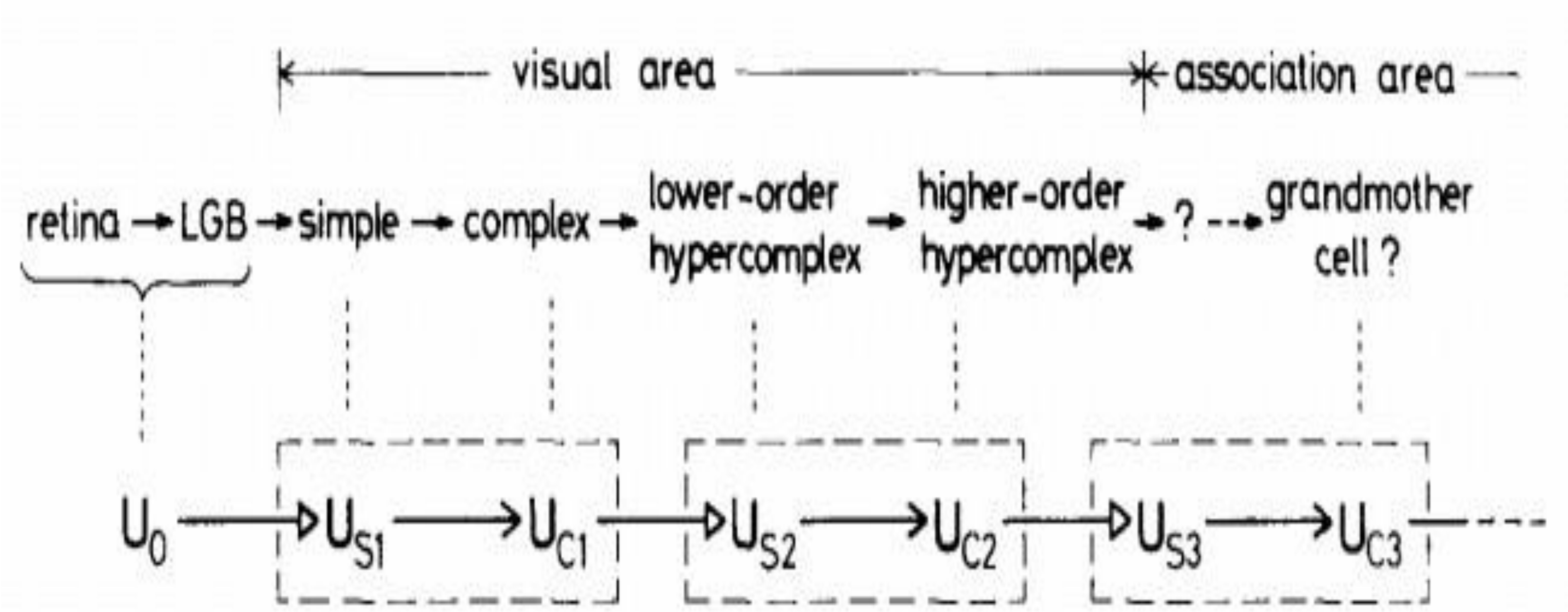
A bit of history:

Neurocognitron
[Fukushima 1980]

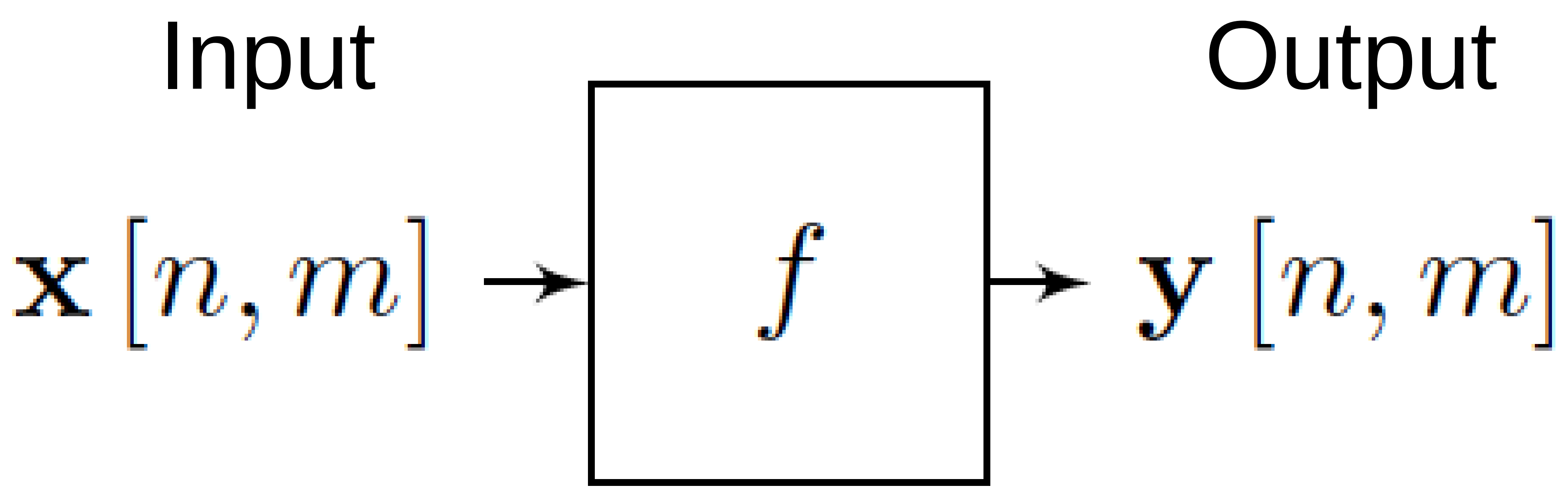
“sandwich” architecture (SCSCSC…)
simple cells: modifiable parameters
complex cells: perform pooling







Linear Systems



One important class of systems is the set of linear systems.

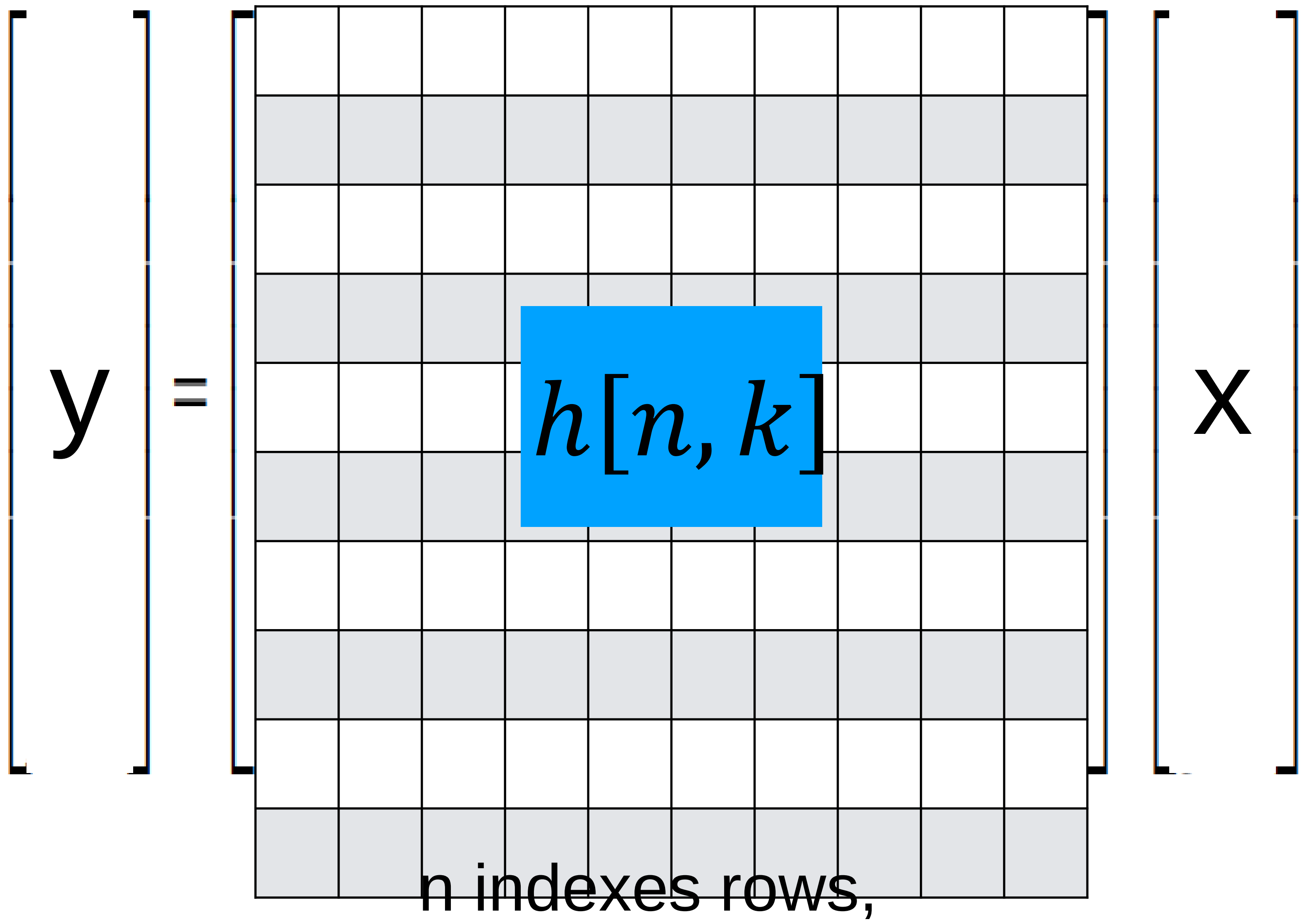
A function f is linear if it satisfies:

$$f(\alpha \mathbf{x}) = \alpha f(\mathbf{x})$$

$$f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$$

Linear system: $\mathbf{y} = f(\mathbf{x})$

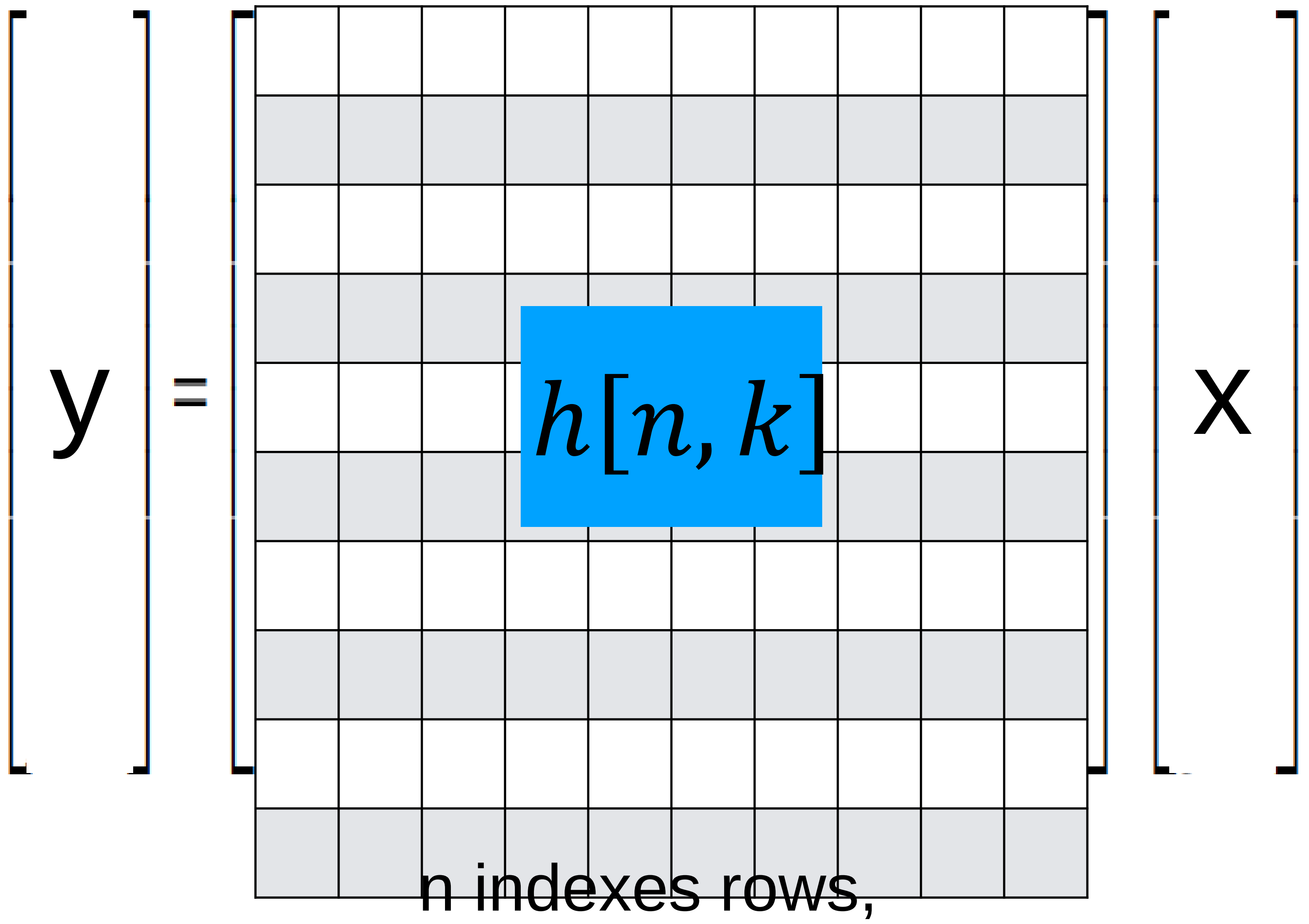
A linear function f can be written as a matrix multiplication:



n indexes rows,
k indexes columns

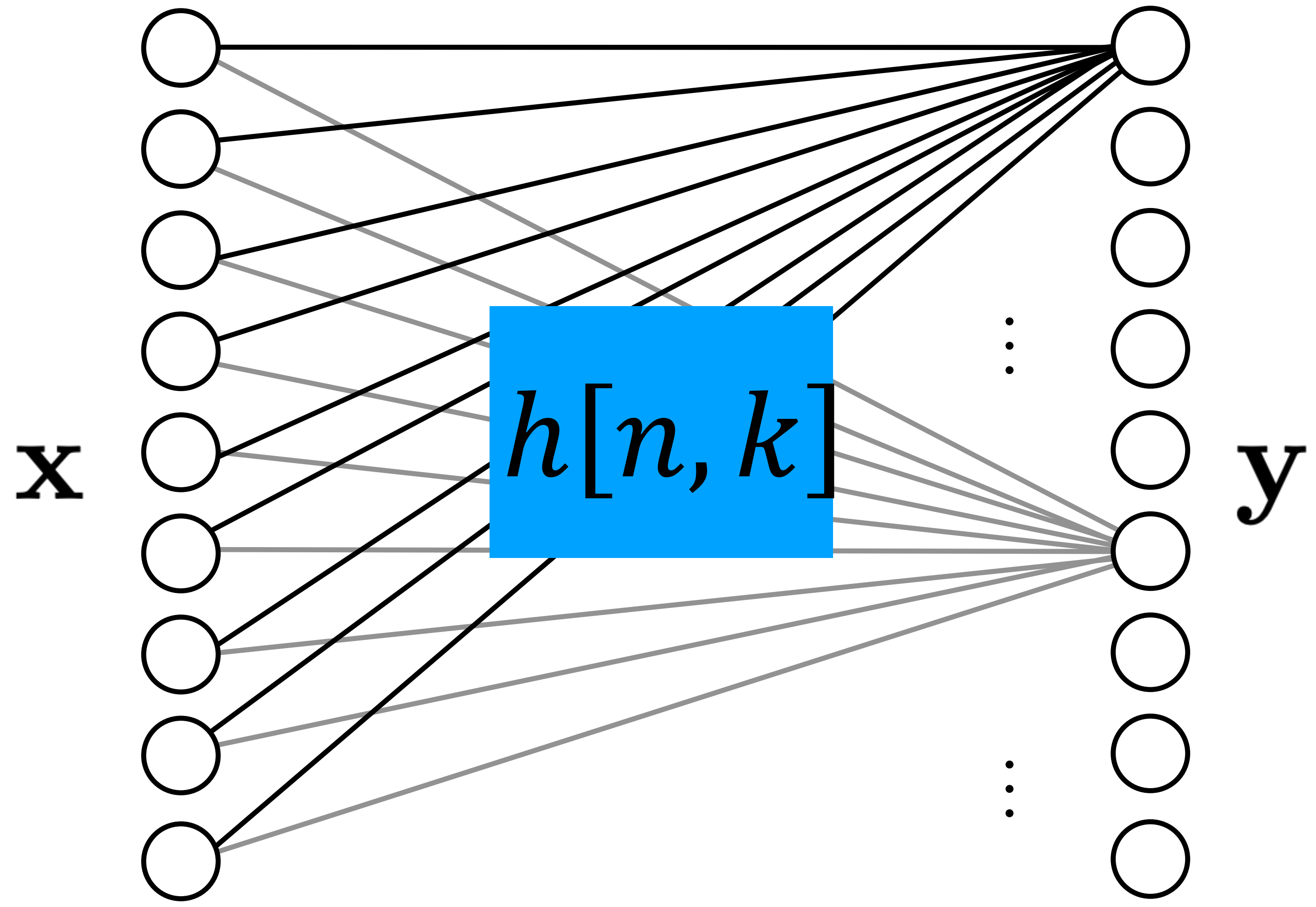
Linear system: $\mathbf{y} = f(\mathbf{x})$

A linear function f can be written as a matrix multiplication:



n indexes rows,
k indexes columns

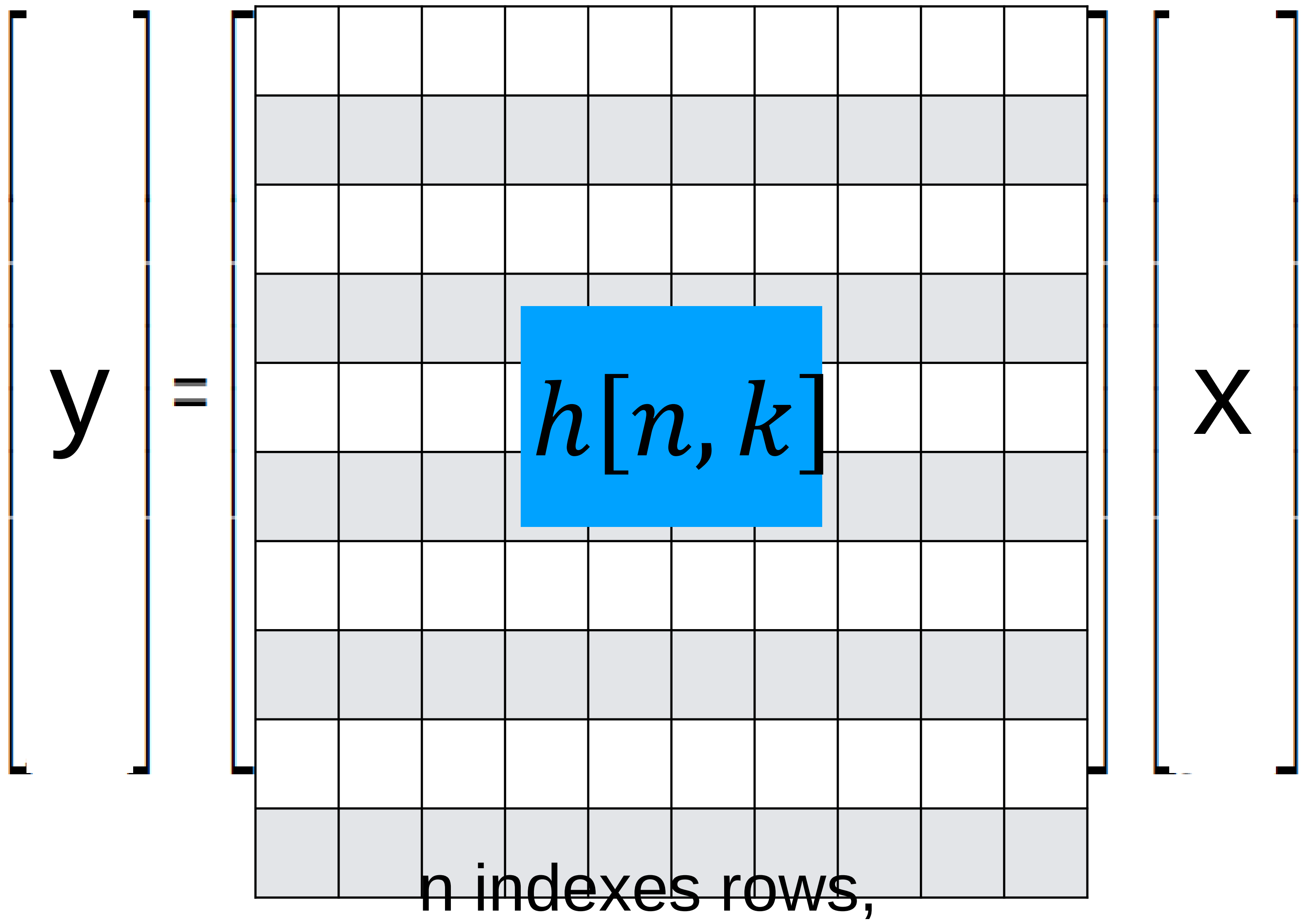
It can also be represented as a fully connected linear neural network



$h[n, k]$ Is the strength of the connection between x[k] and y[n]

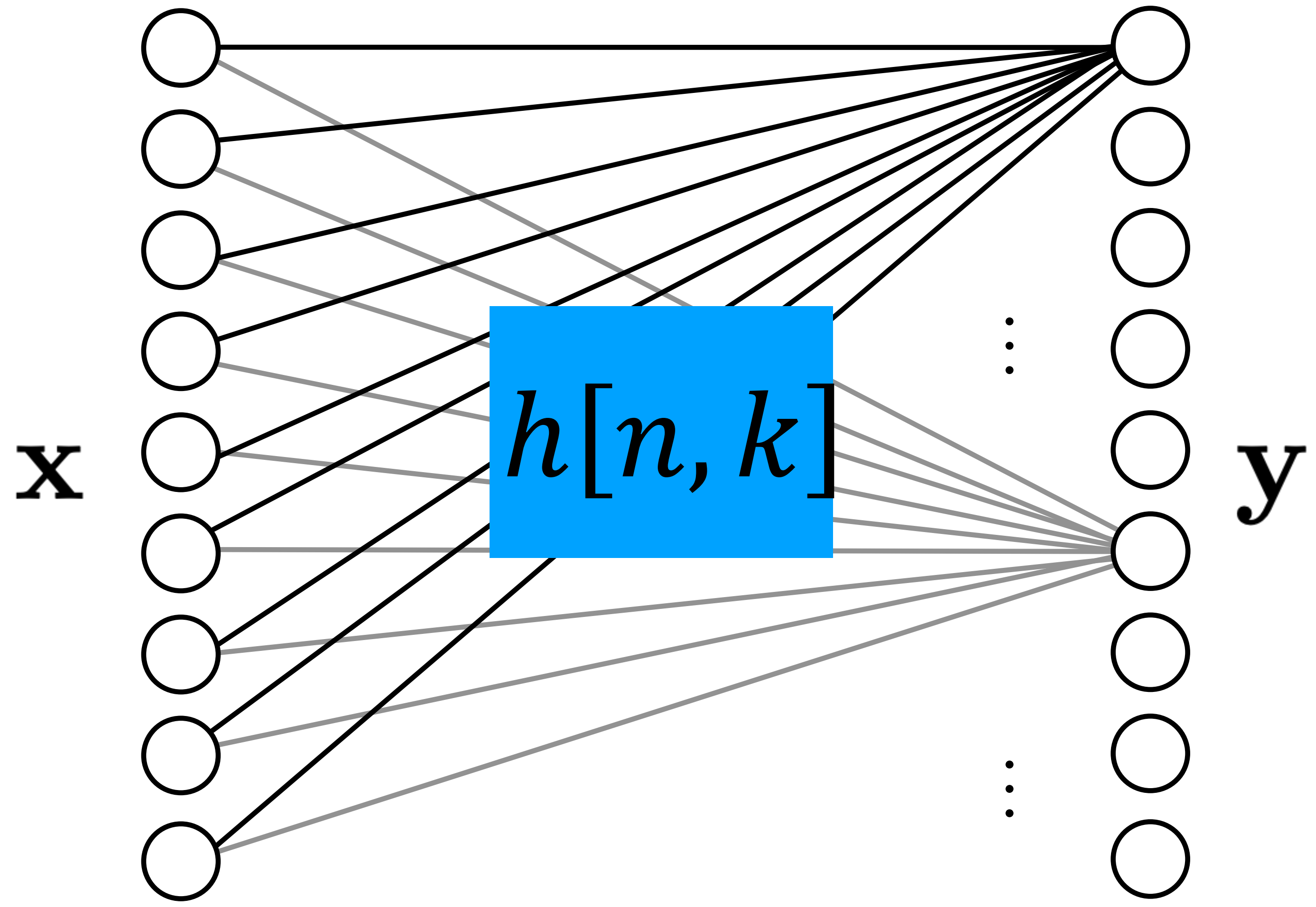
Linear system: $\mathbf{y} = f(\mathbf{x})$

A linear function f can be written as a matrix multiplication:



n indexes rows,
k indexes columns

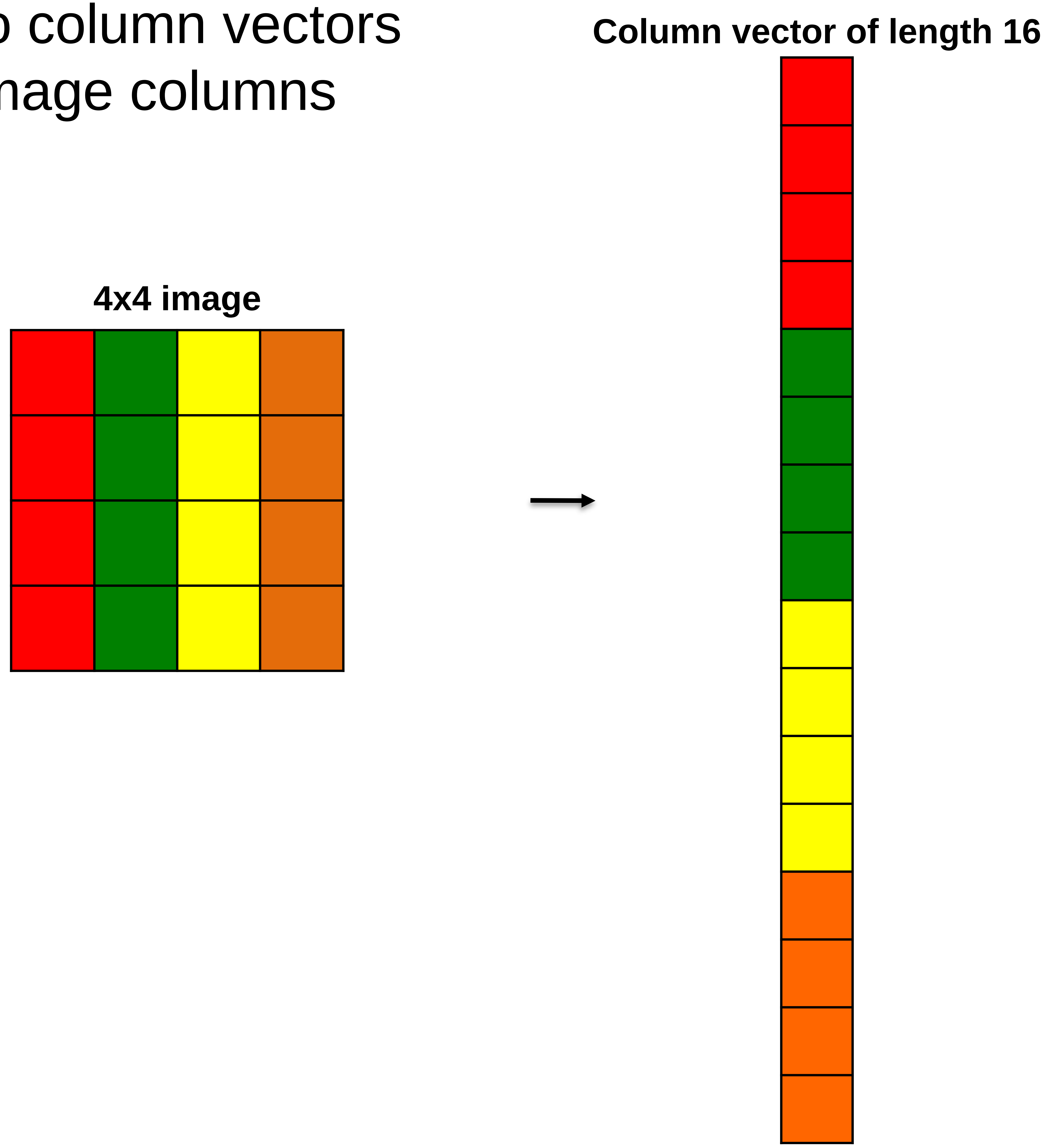
It can also be represented as a fully connected linear neural network

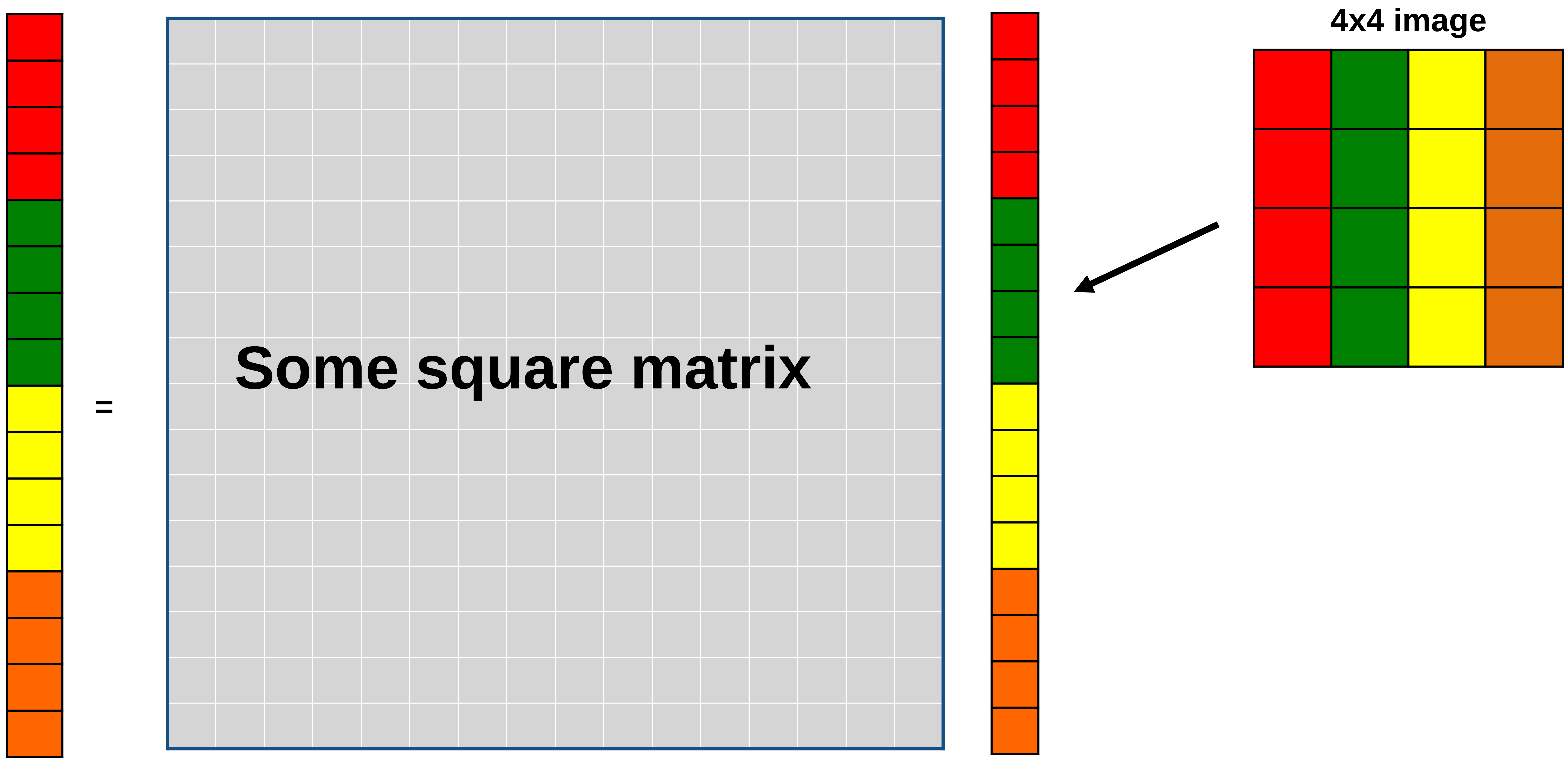


$$y[n] = \sum_{k=0}^{N-1} h[n,k]x[k]$$

$h[n,k]$ Is the strength of the connection between x[k] and y[n]

Images are turned into column vectors by concatenating all image columns



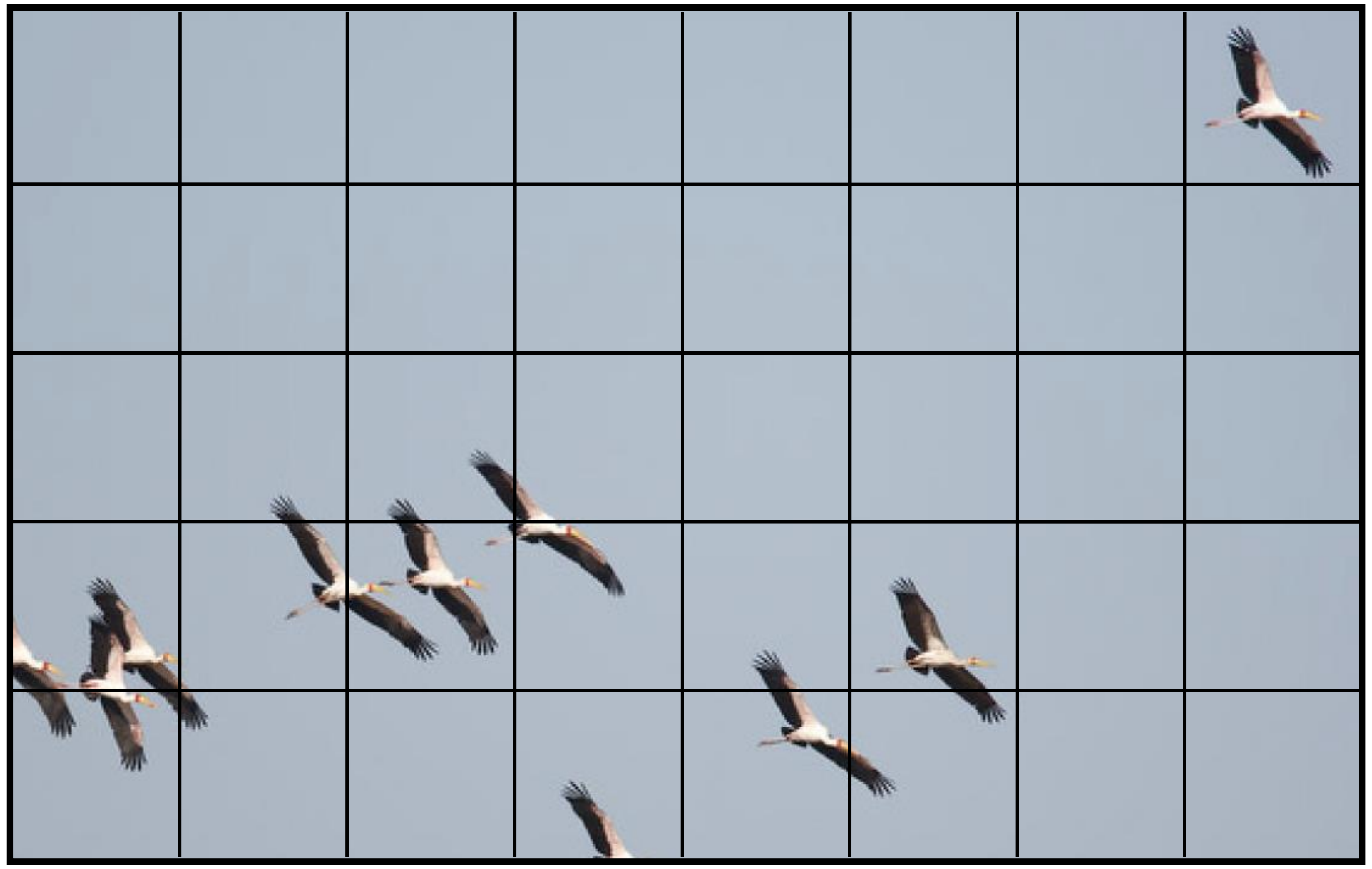


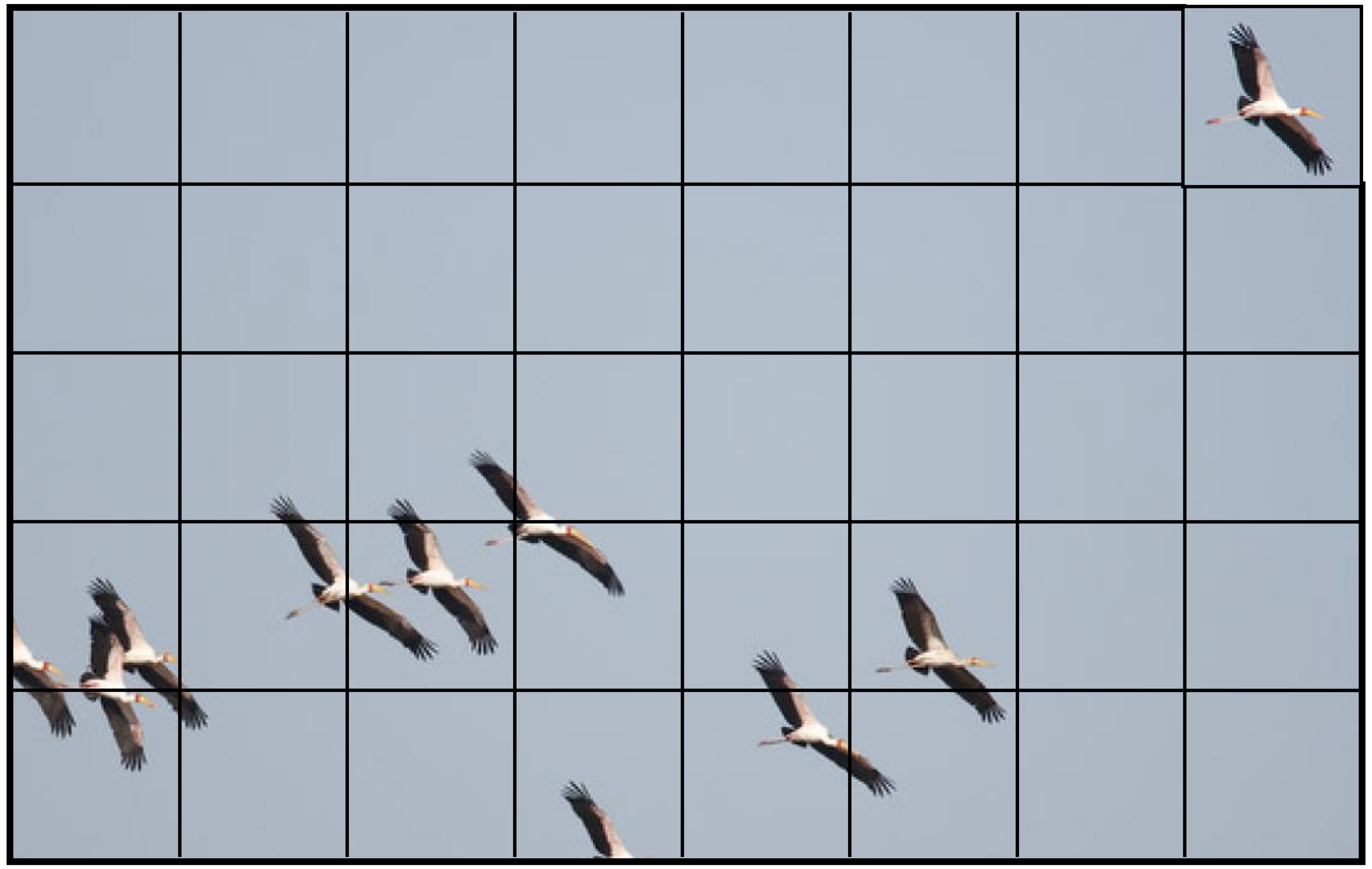
=
Some square matrix
4x4 image

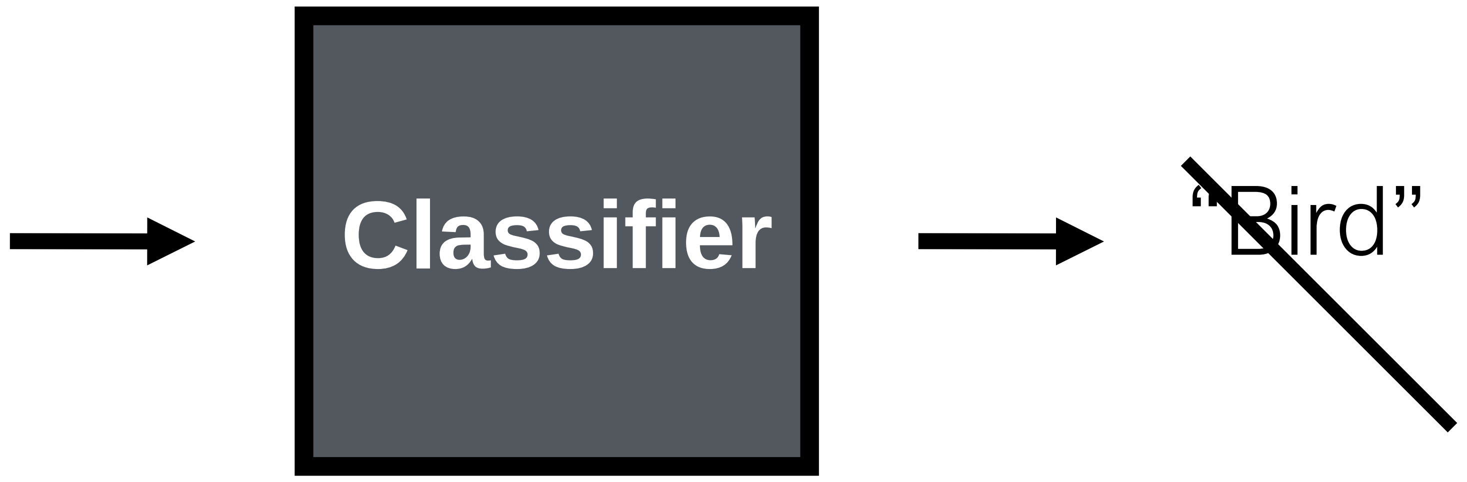
Quiz: what operation is linear?

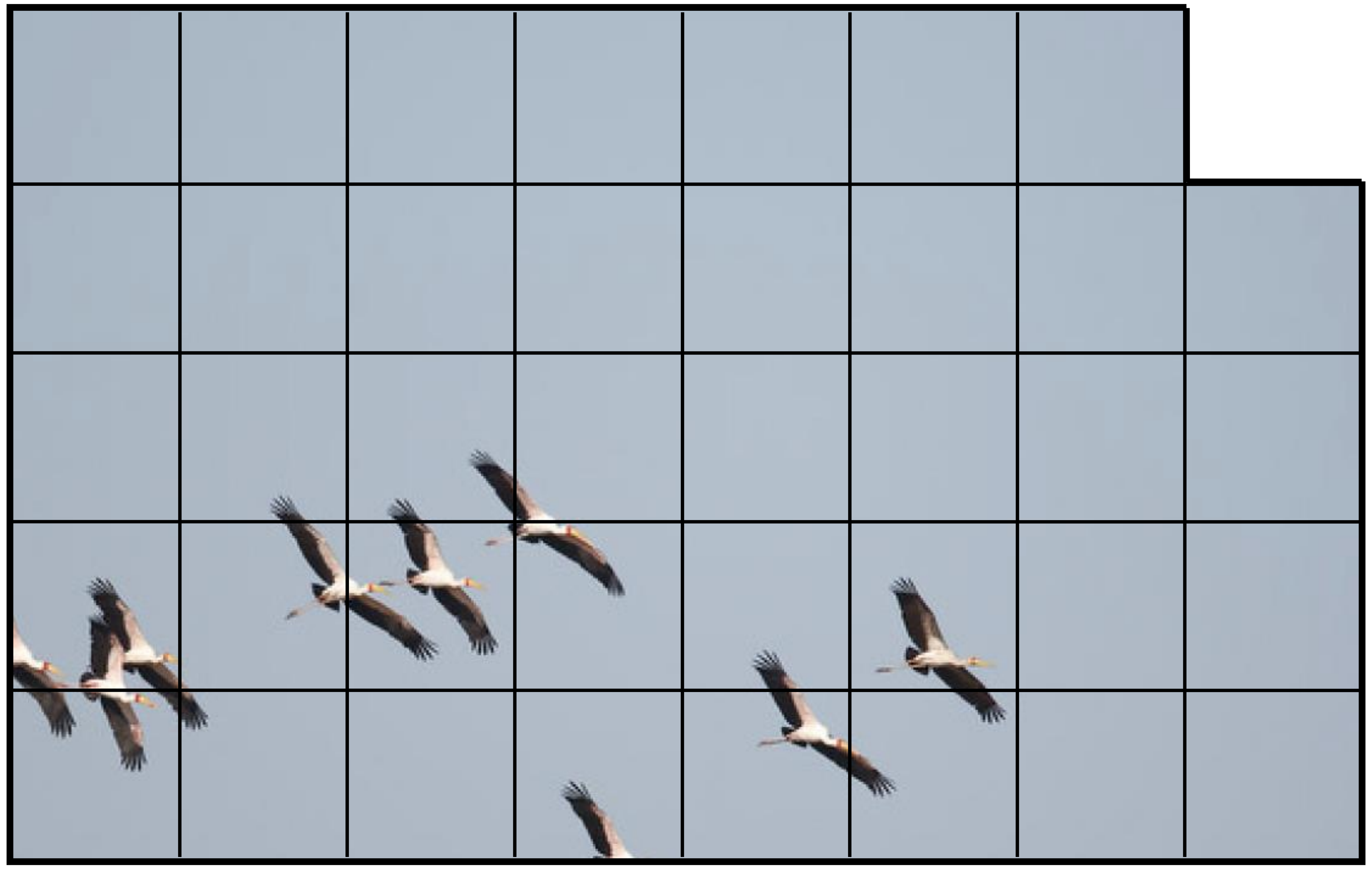
Quiz: what operation is linear?

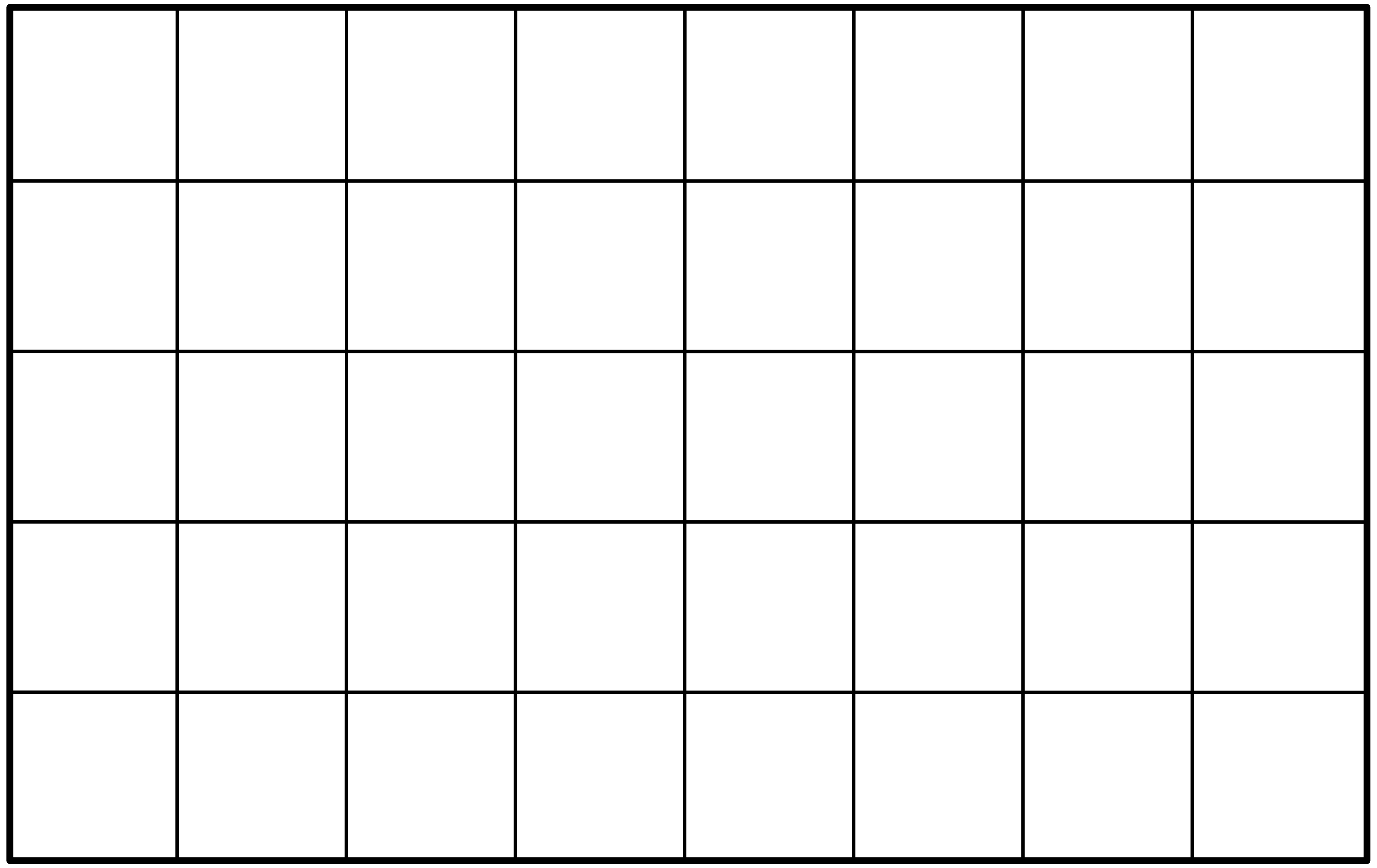
We need translation invariance



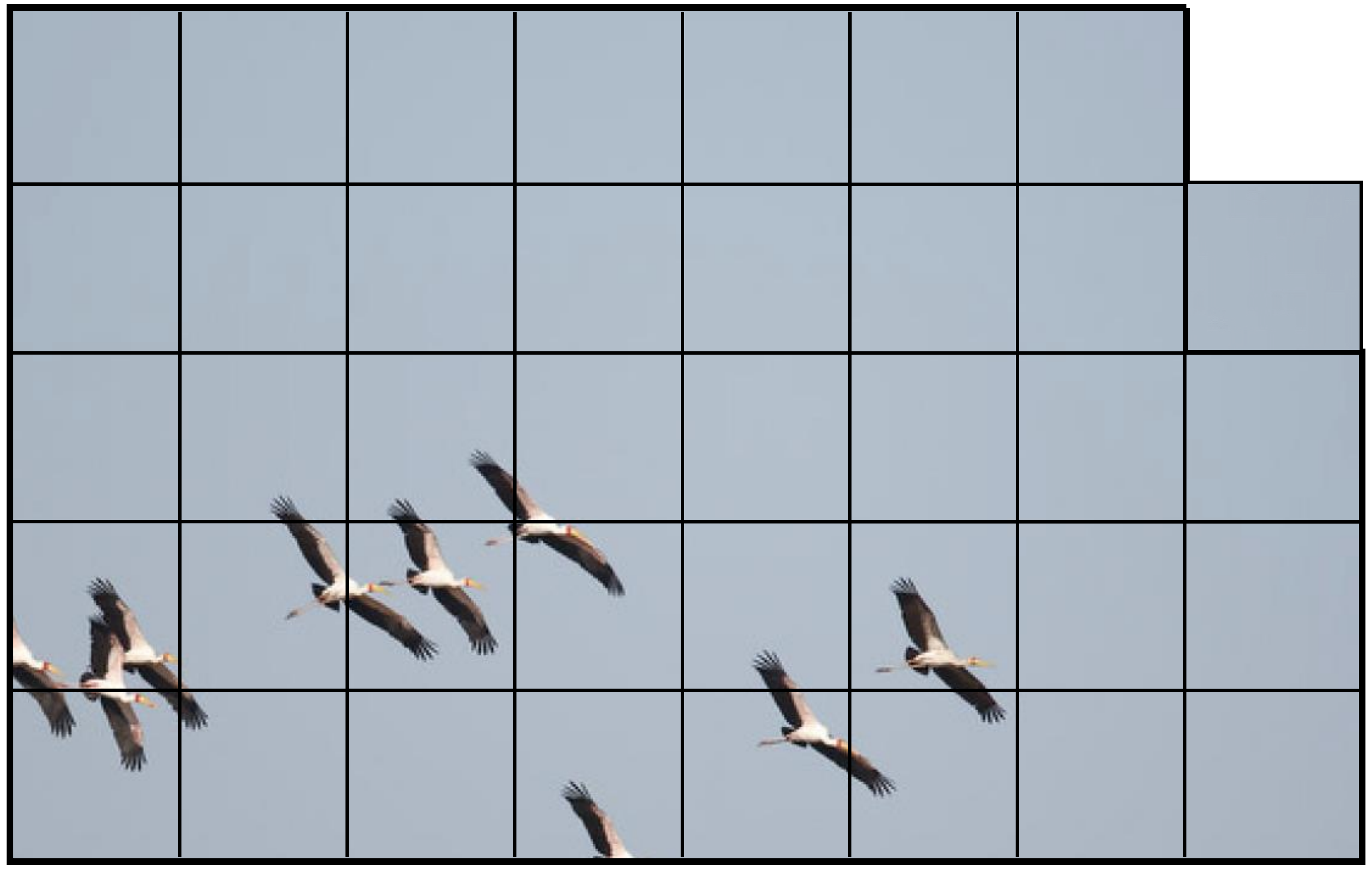


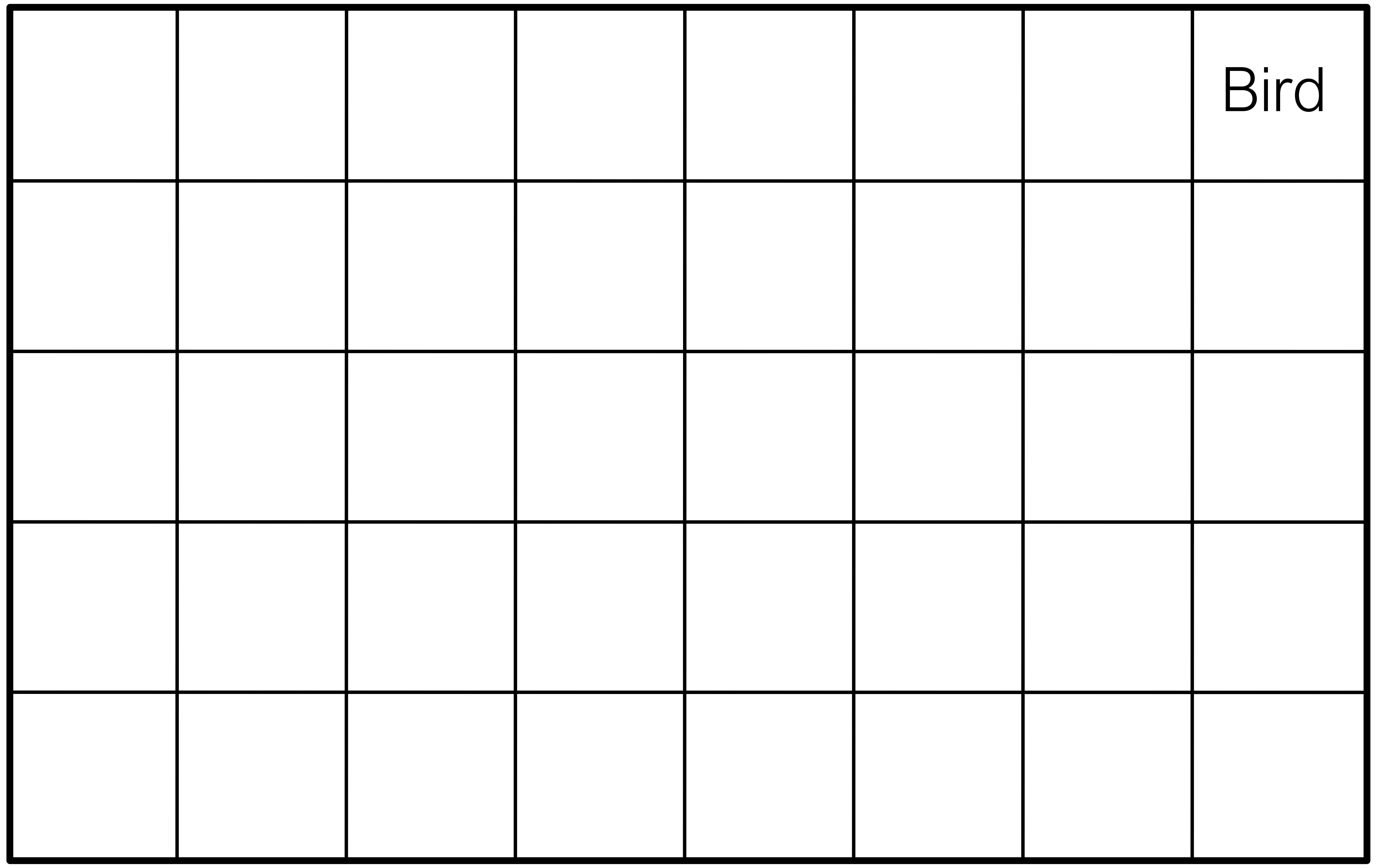

Classifier
~~“Bird”~~

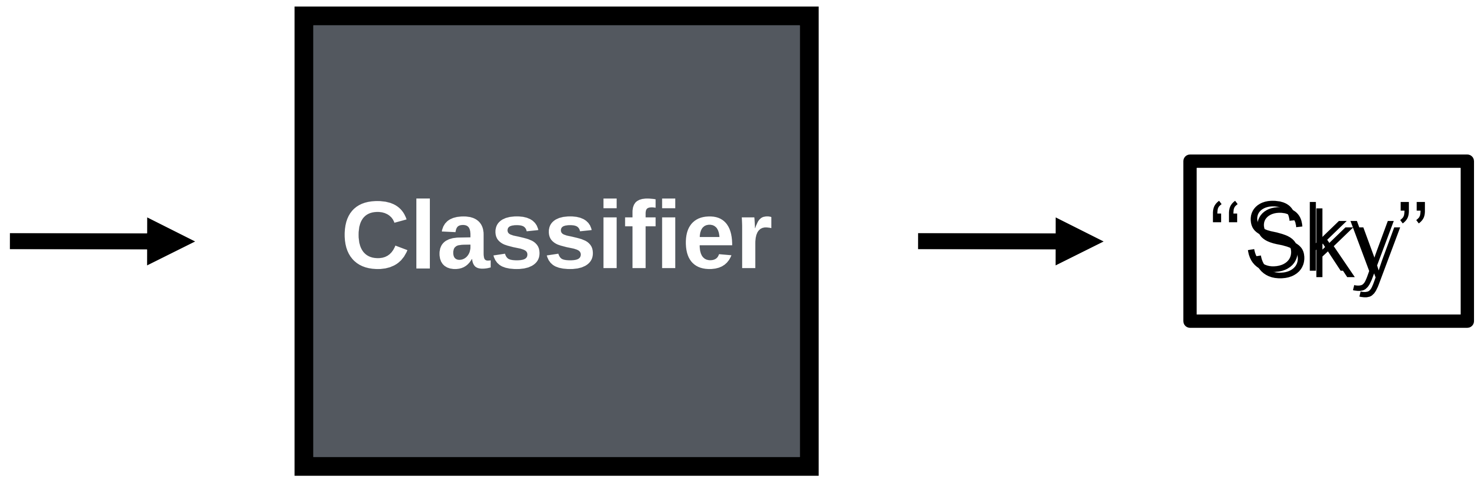


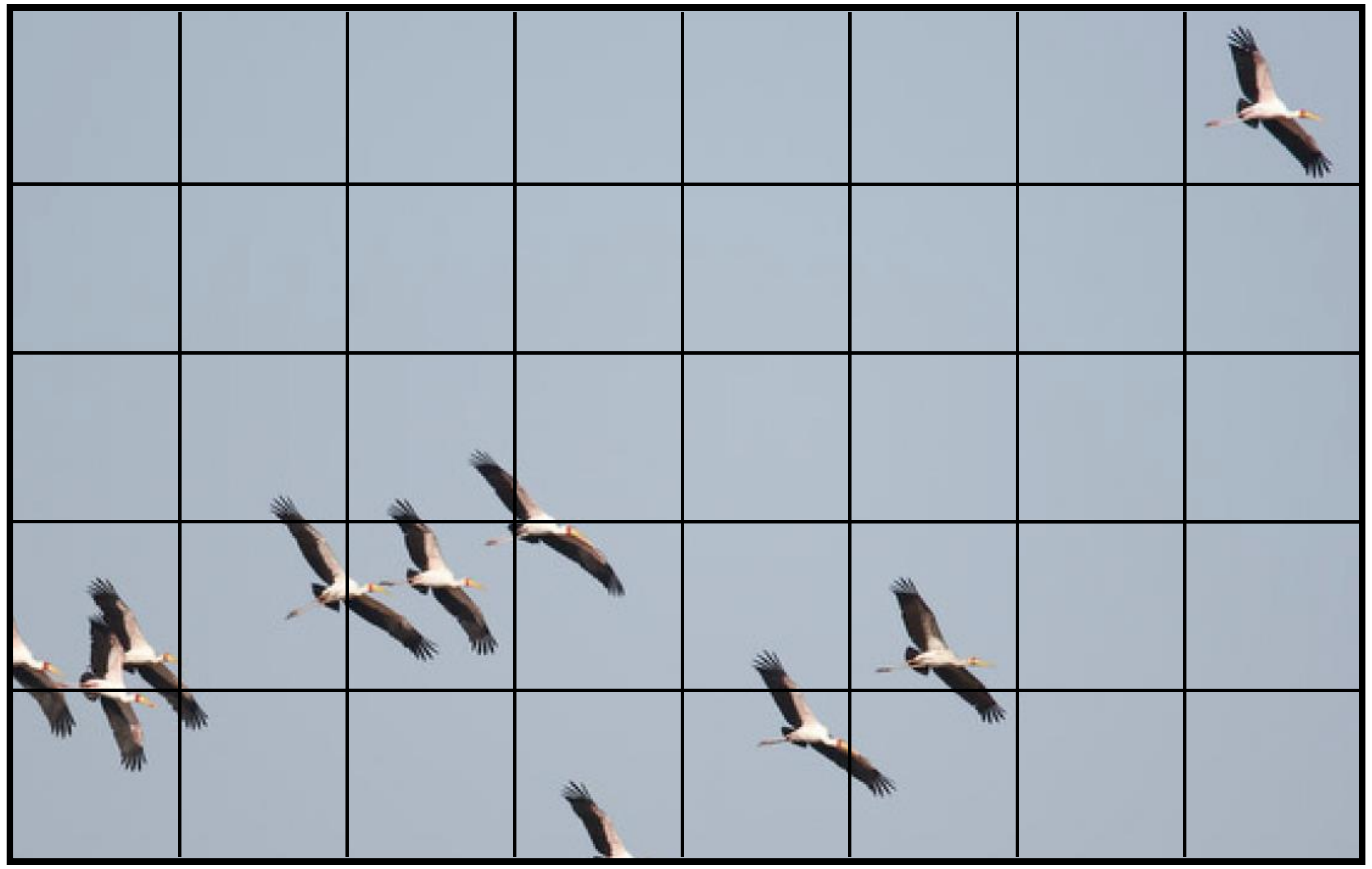



Classifier
“Bird”

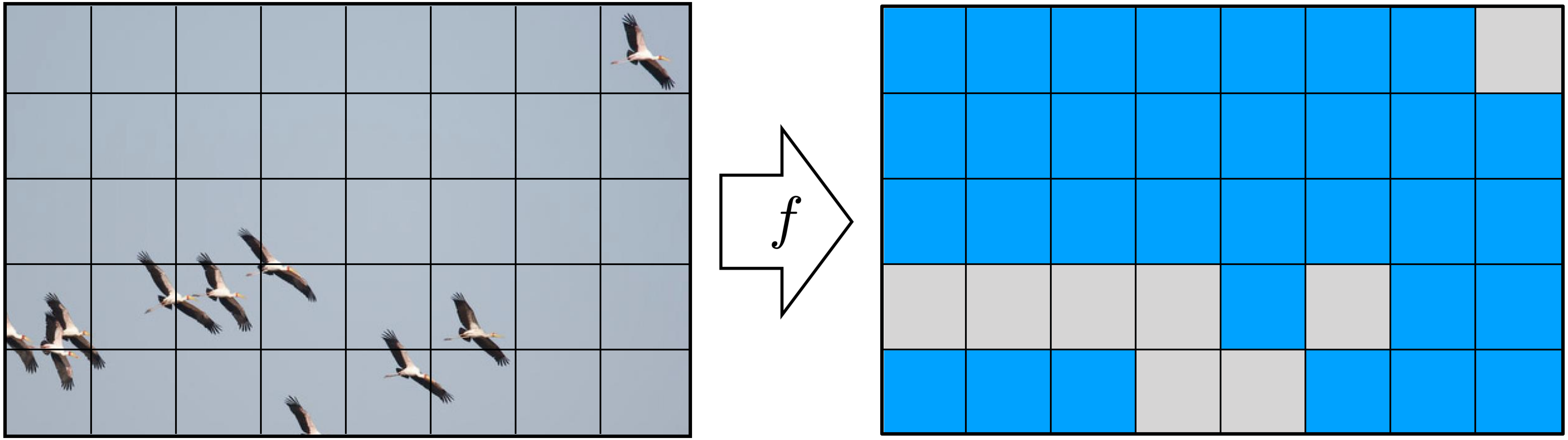



Bird


Classifier
"Sky"

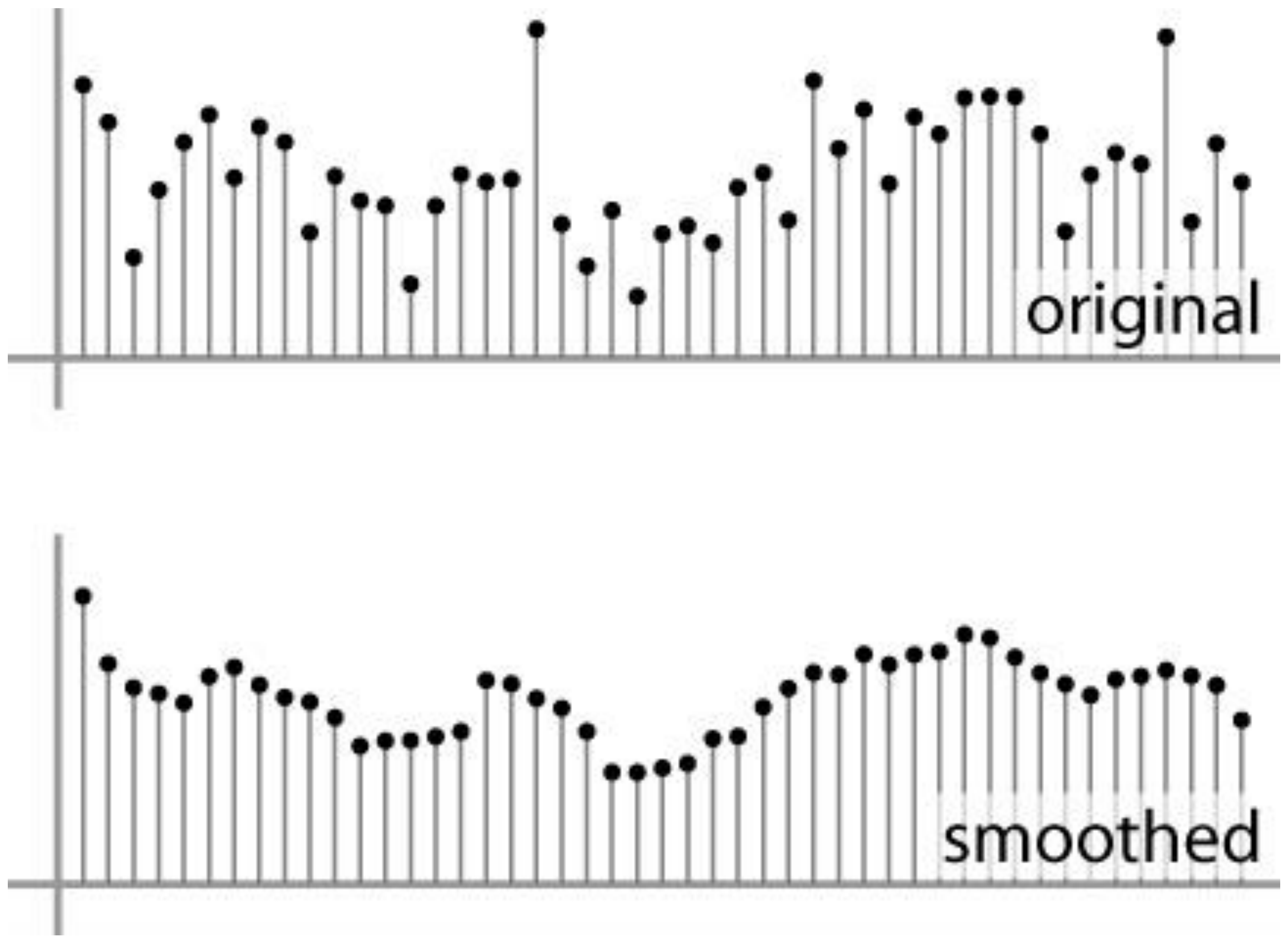


Sky	Sky	Sky	Sky	Sky	Sky	Sky	Bird
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Sky
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Sky
Bird	Bird	Bird	Sky	Bird	Sky	Sky	Sky
Sky	Sky	Sky	Bird	Sky	Sky	Sky	Sky


f

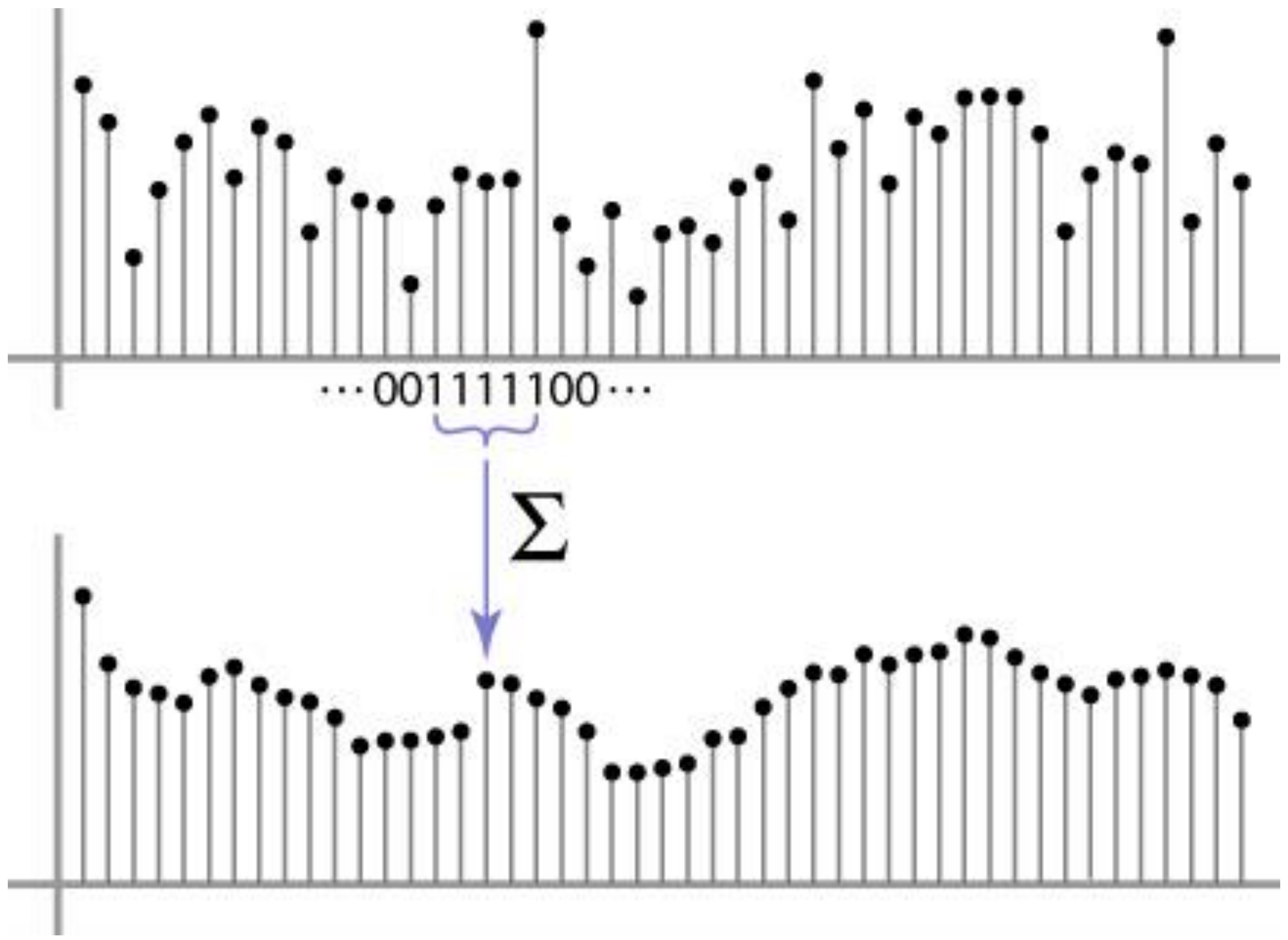
Moving Average

- basic idea: define a new function by averaging over a sliding window
- a simple example to start off: smoothing



Moving Average

- Can add weights to our moving average
- *Weights* [..., 0, 1, 1, 1, 1, 1, 0, ...] / 5



In 2D: box filter

$$h[\cdot\,,\cdot\,]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Slide credit: David Lowe (UBC)

Image filtering

$$h[\cdot\, ,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

$$g[m,n] = \sum_{k,l} h[k,l]\ f[m+k,n+l]$$

Credit: S. Seitz

Image filtering

$$h[\cdot\,,\cdot\,]\;\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$f\,[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$g[.,.]$$

	0	10							

$$g[m,n] = \sum_{k,l} h[k,l]\; f\,[m+k, n+l]$$

Credit: S. Seitz

Image filtering

$h[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20						

Credit: S. Seitz

Image filtering

$h[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f\,[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20	30					

Credit: S. Seitz

Image filtering

$$h[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$g[.,.]$$

	0	10	20	30	30				

Credit: S. Seitz

Image filtering

$$h[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20	30	30				
				?					

Credit: S. Seitz

Image filtering

$$h[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20	30	30				
						?			
				50					

Credit: S. Seitz

Image filtering

$$h[\cdot\,,\cdot\,] \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$g[m,n] = \sum_{k,l} h[k,l]\; f[m+k, n+l]$$

Credit: S. Seitz

Box Filter

What does it do?

- Replaces each pixel with an average of its neighborhood

- Achieve smoothing effect (remove sharp features)

$h[\cdot\,,\cdot\,]$

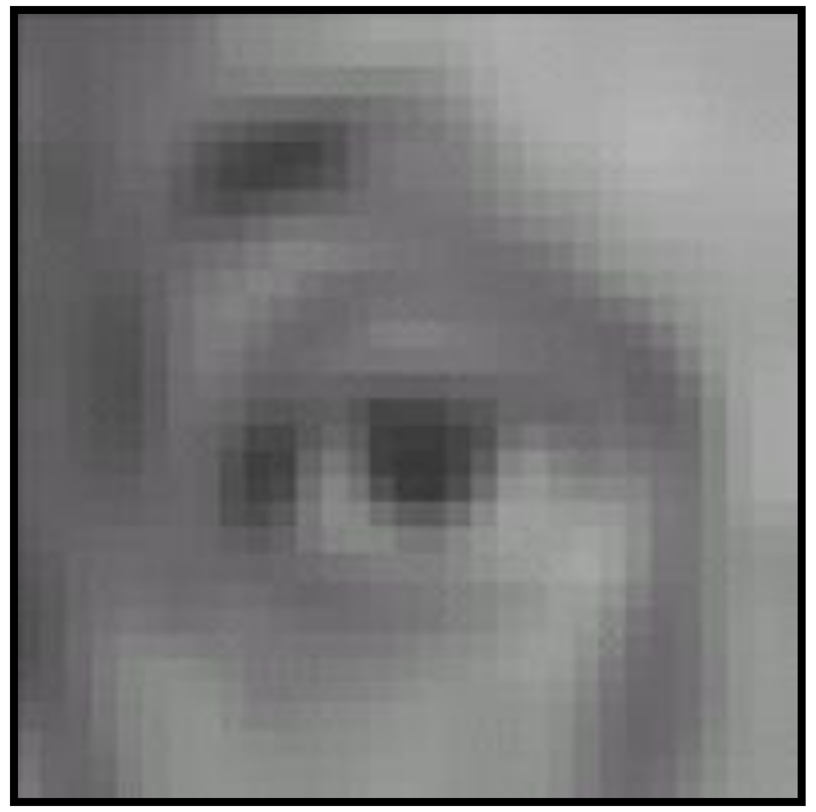
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Slide credit: David Lowe (UBC)

Linear filters: examples

Original

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$



Blur (with a mean filter)

Source: D. Lowe

Cross-correlation

Let F be the image, H be the kernel (of size 2*k*+1 x 2*k*+1), and G be the output image

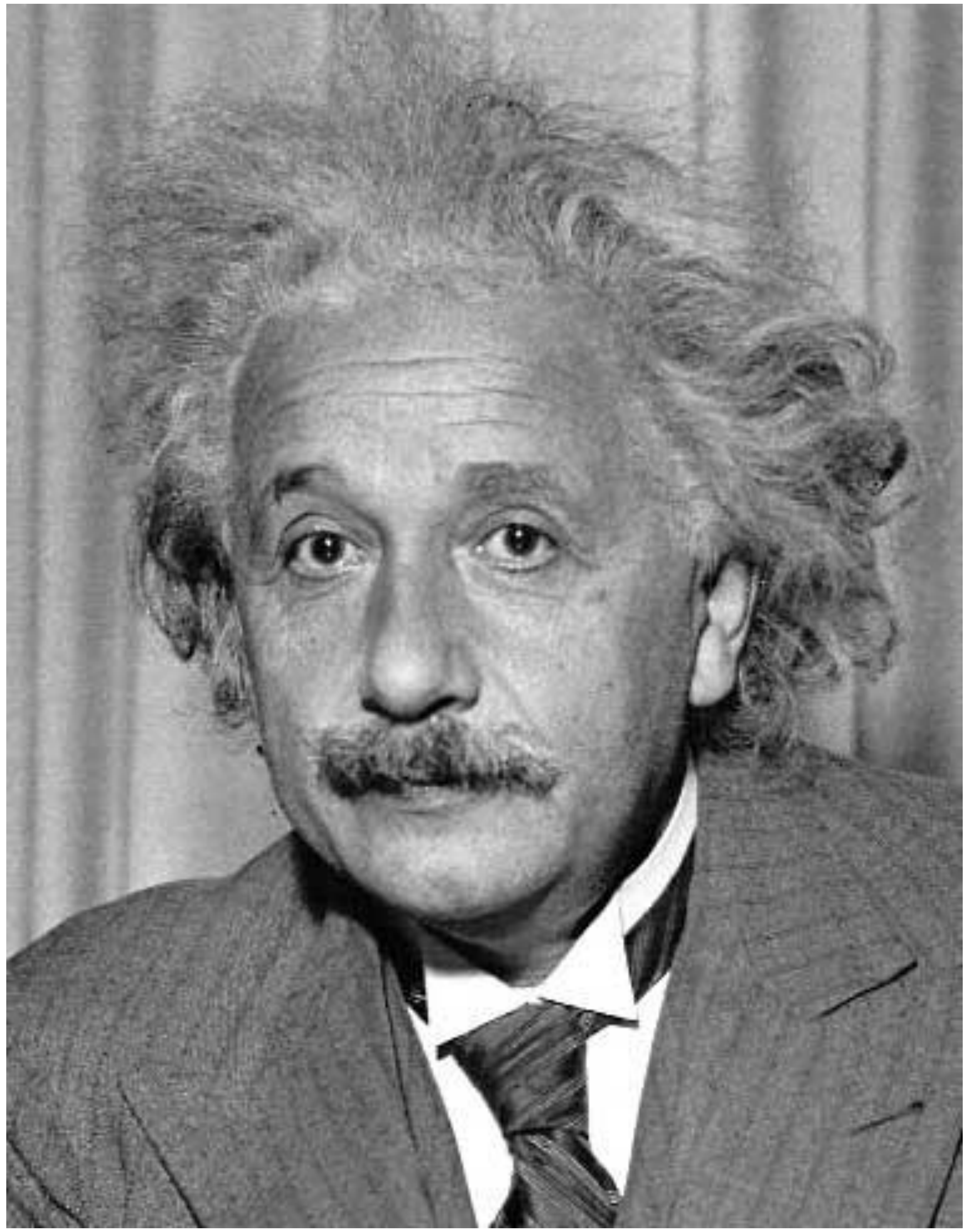
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

This is called a **cross-correlation** operation:

$$G = H \otimes F$$

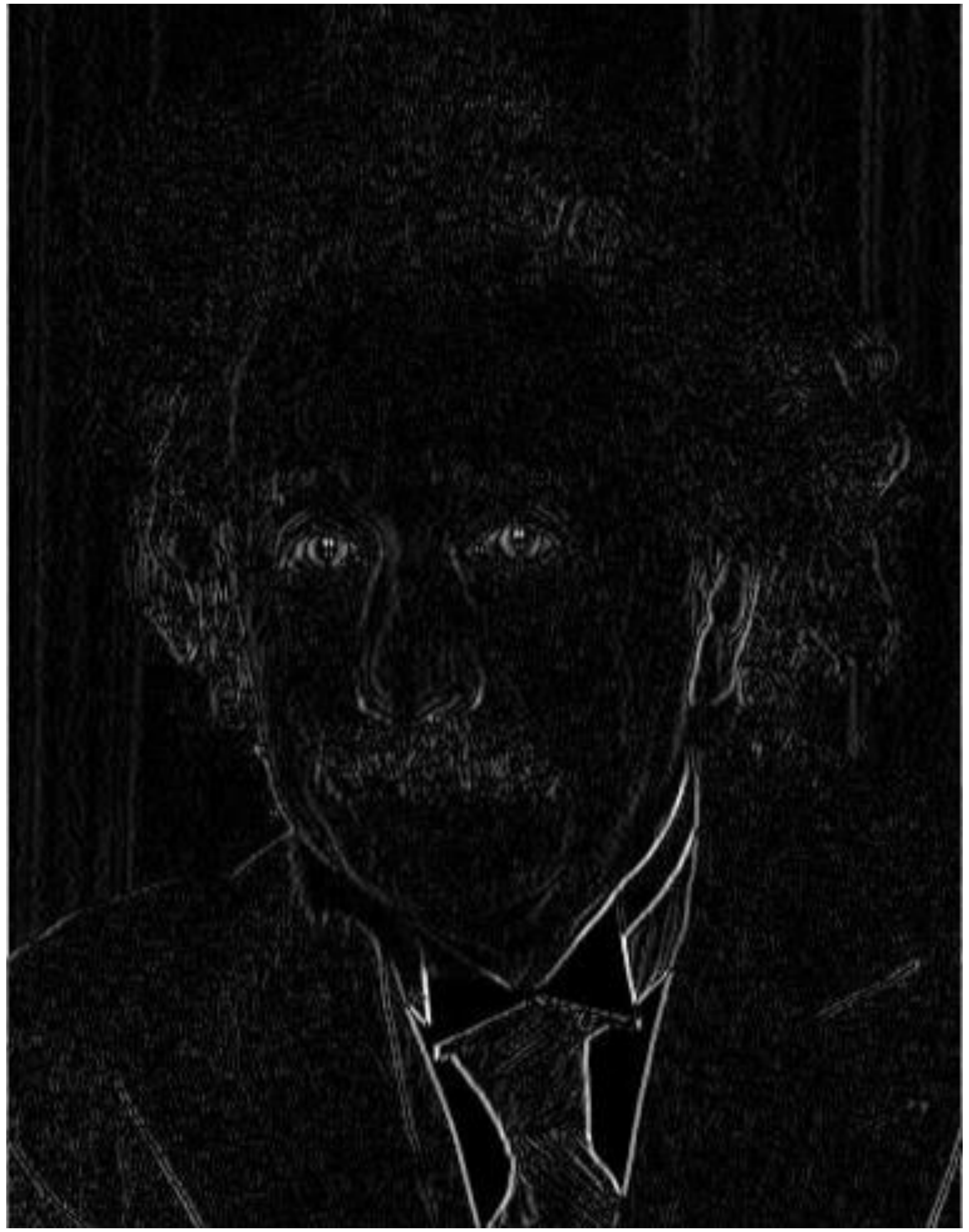
- Can think of as a “dot product” between local neighborhood and kernel for each pixel

Other filters



1	0	-1
2	0	-2
1	0	-1

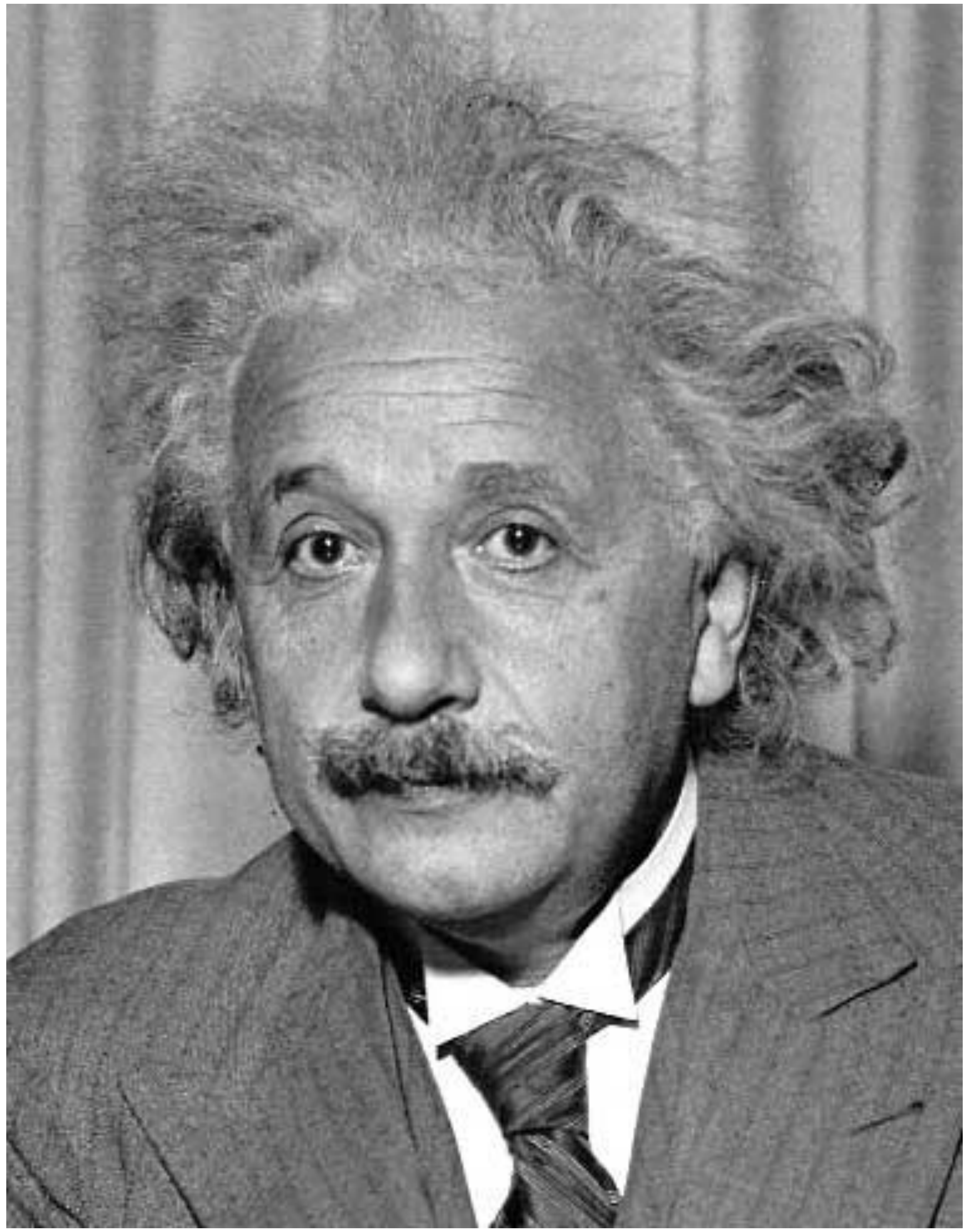
Sobel



Vertical Edge
(absolute value)

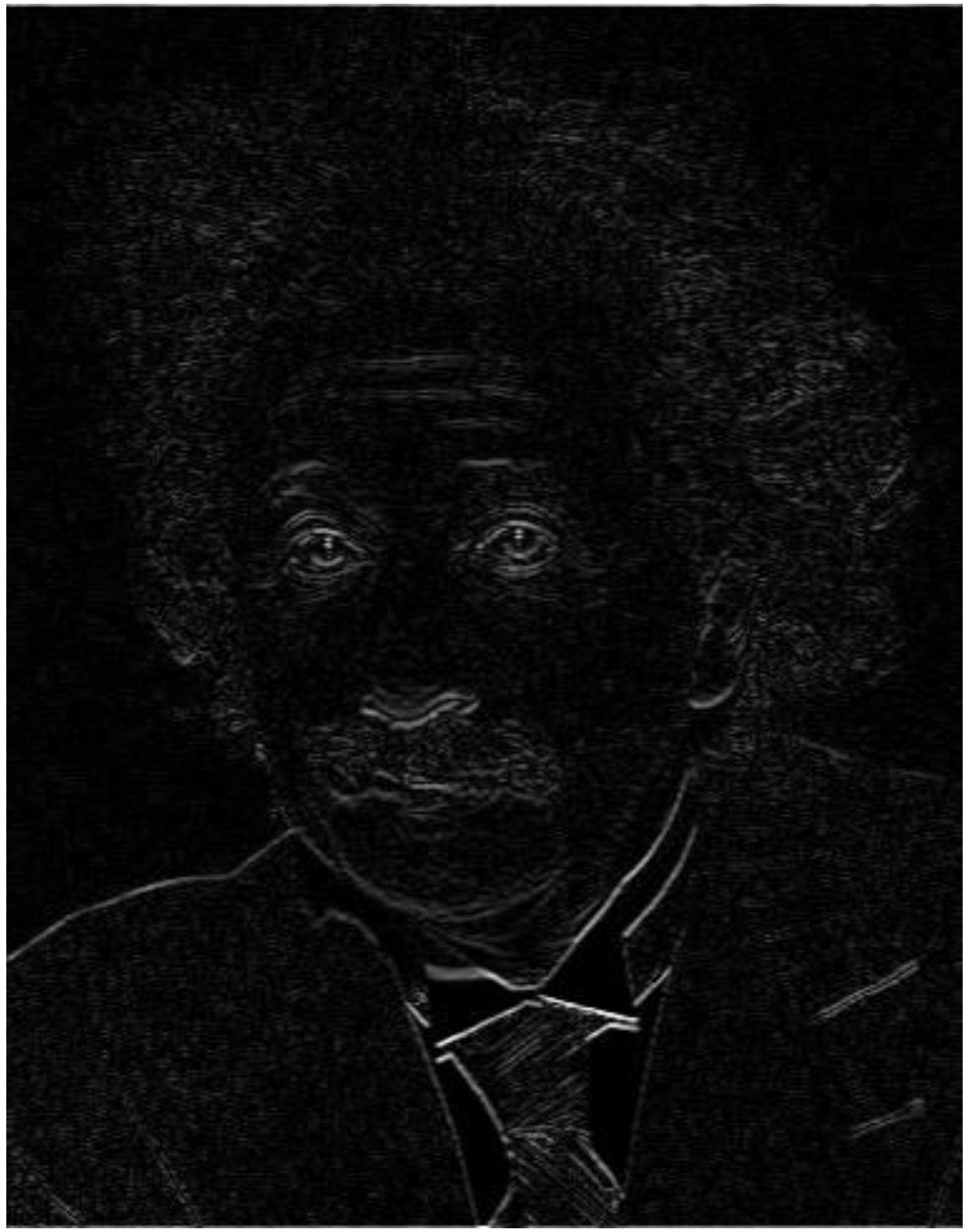
Q?

Other filters



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

Cross-correlation vs. Convolution

cross-correlation:

$$G = H \otimes F$$

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

A **convolution** operation is a cross-correlation where the filter is flipped both horizontally and vertically before being applied to the image:

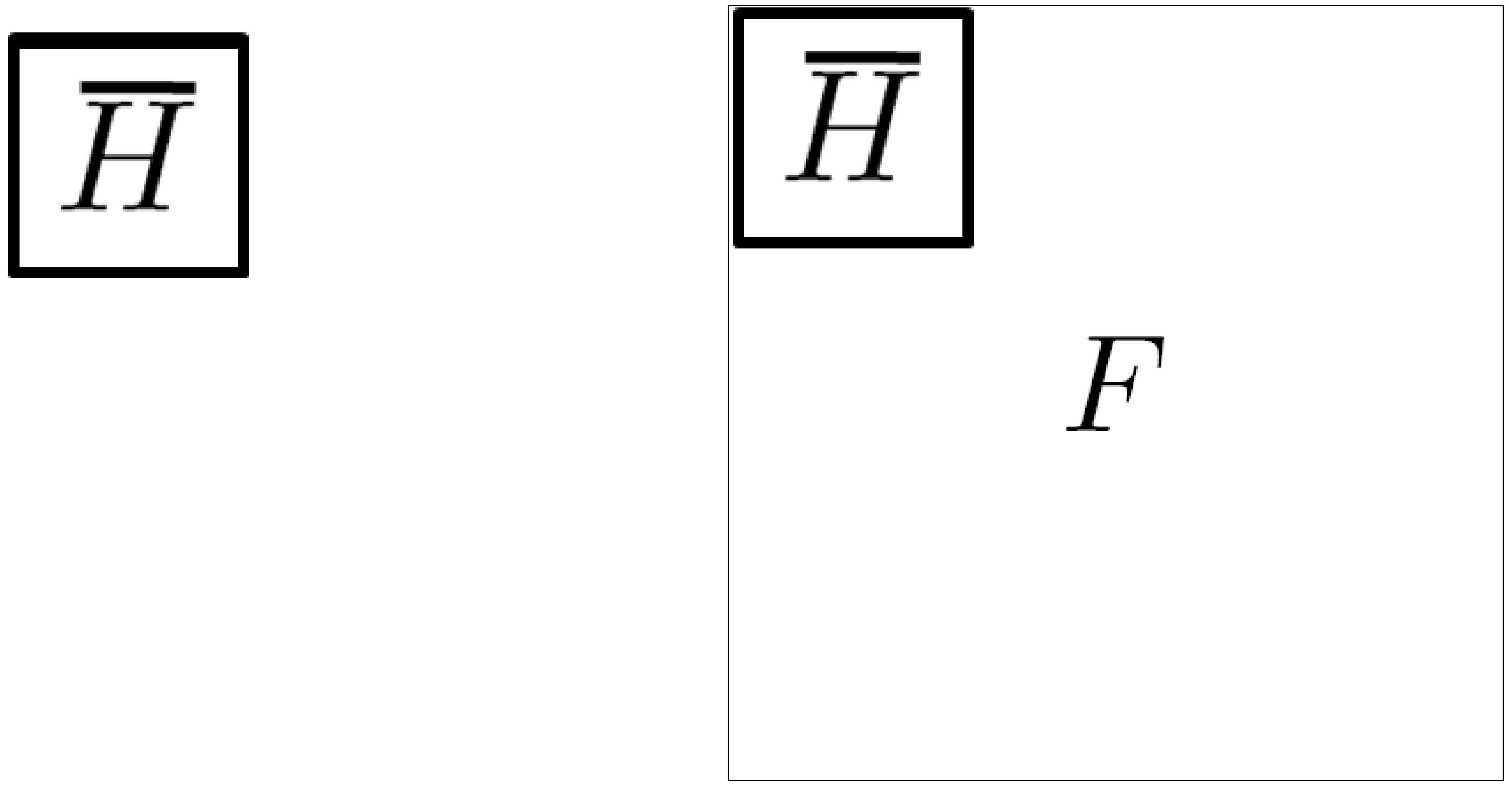
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

It is written:

$$G = H \star F$$

Slide by Steve Seitz

Convolution



Adapted from F. Durand

Convolution is nice!

- Notation: $b = c \star a$
- Convolution is a multiplication-like operation
  - commutative $a \star b = b \star a$
  - associative $a \star (b \star c) = (a \star b) \star c$
  - distributes over addition $a \star (b + c) = a \star b + a \star c$
  - scalars factor out $\alpha a \star b = a \star \alpha b = \alpha(a \star b)$
  - identity: unit impulse *e* = [..., 0, 0, 1, 0, 0, ...]

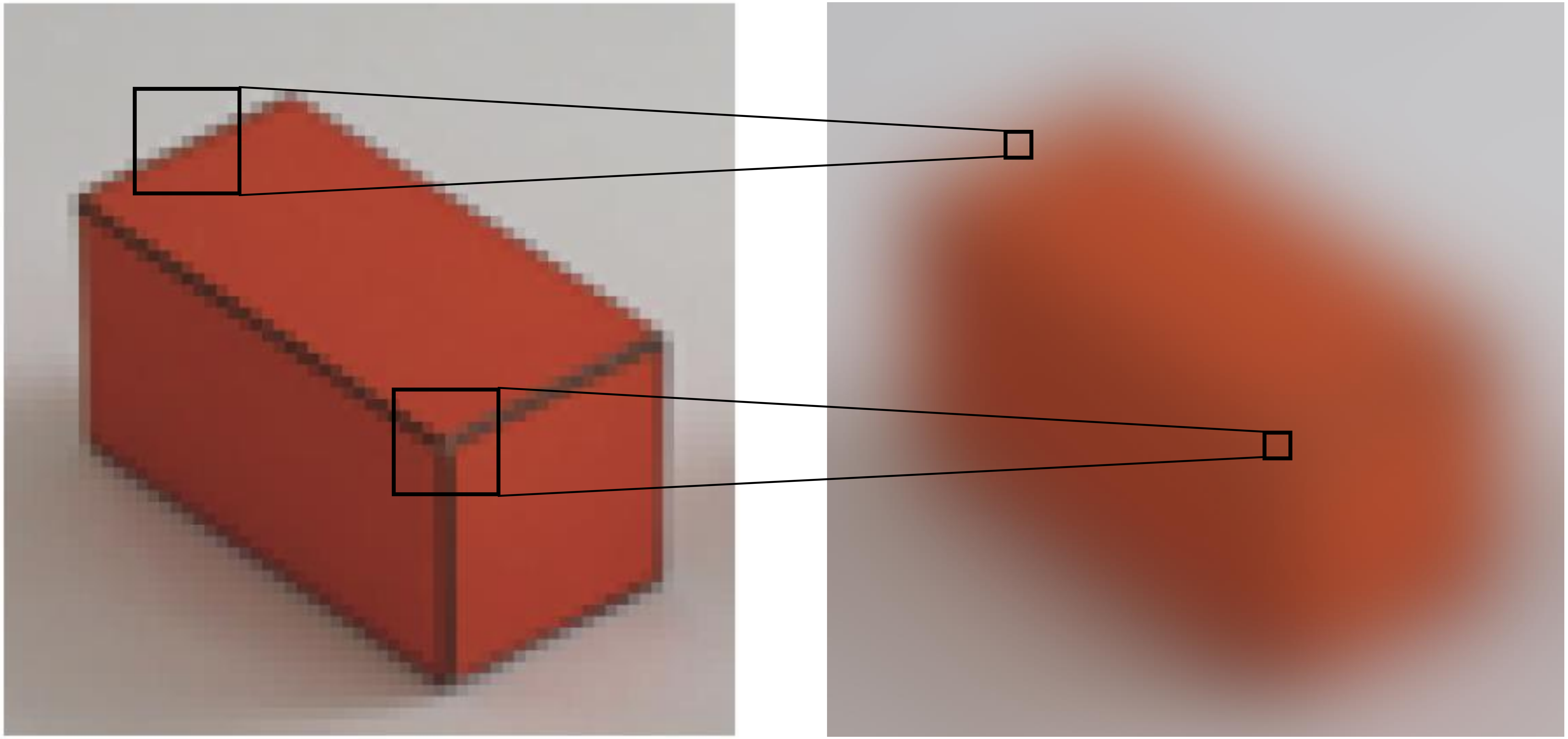
$$a \star e = a$$

- Conceptually no distinction between filter and signal
- Usefulness of associativity
  - often apply several filters one after another: $(((a * b_1) * b_2) * b_3)$
  - this is equivalent to applying one filter: a * $(b_1 * b_2 * b_3)$

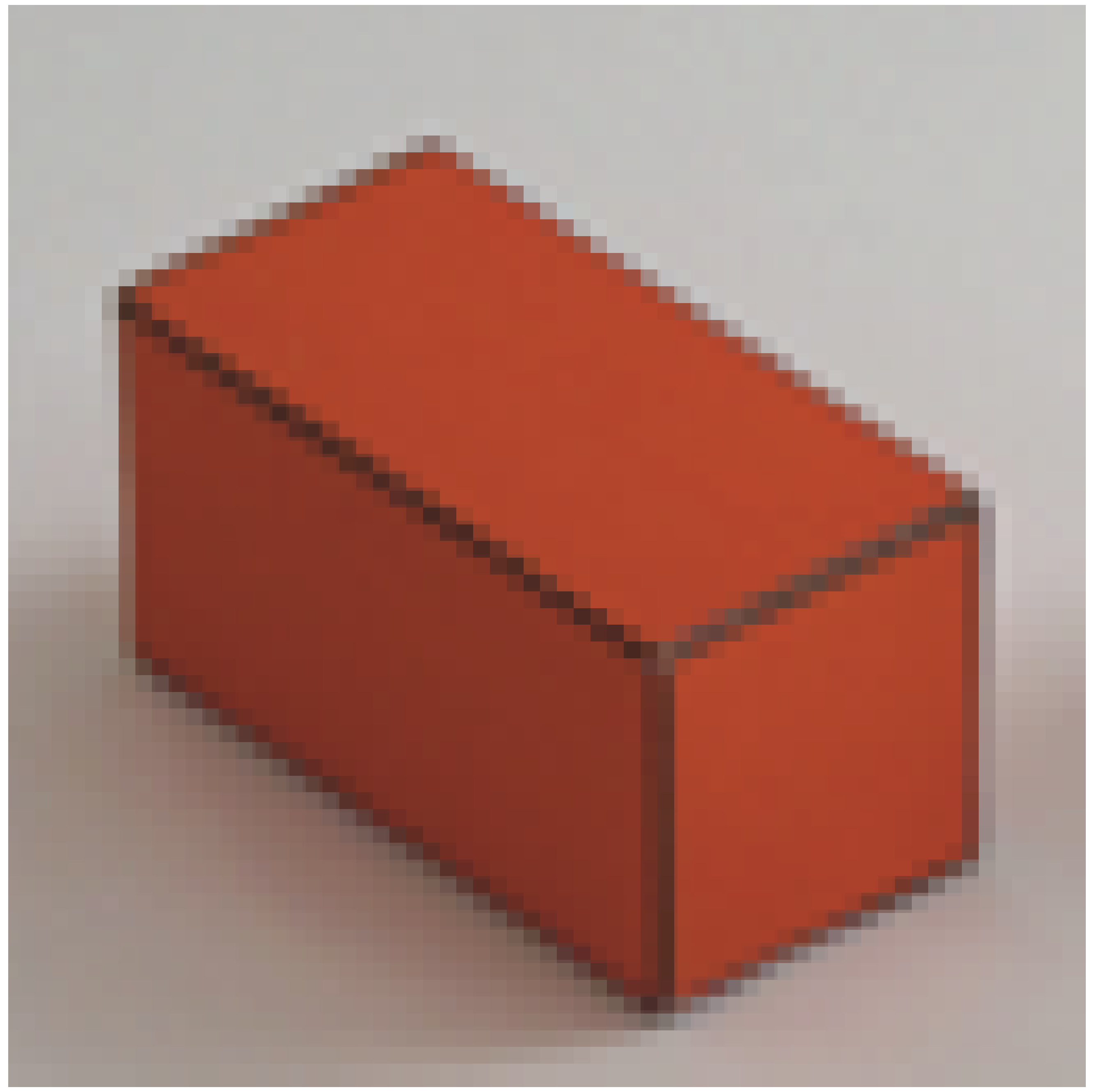
Quiz: what operation is the result of a convolution?

Quiz: what operation is the result of a convolution?

Examples

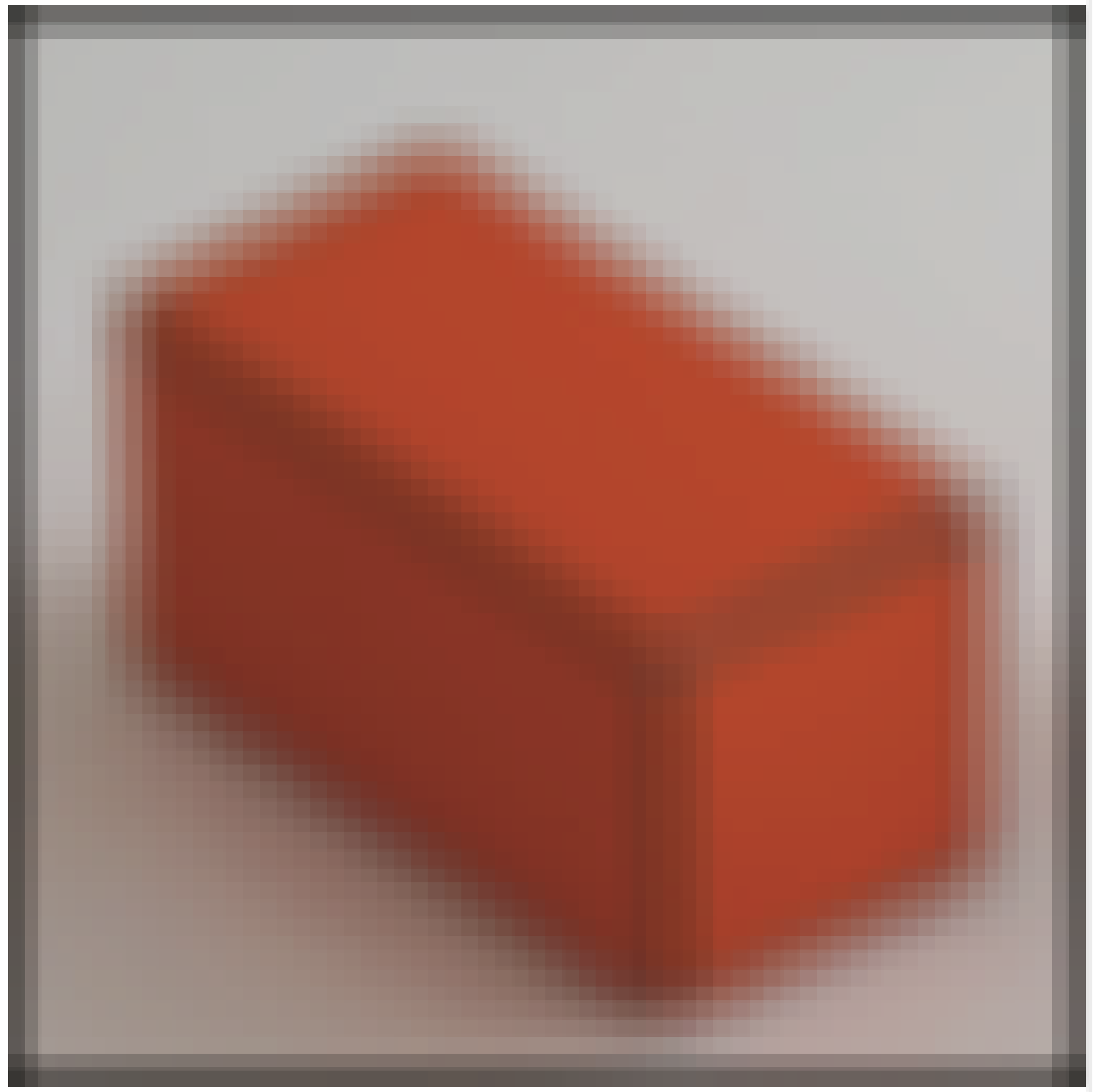


Defocus/blurring



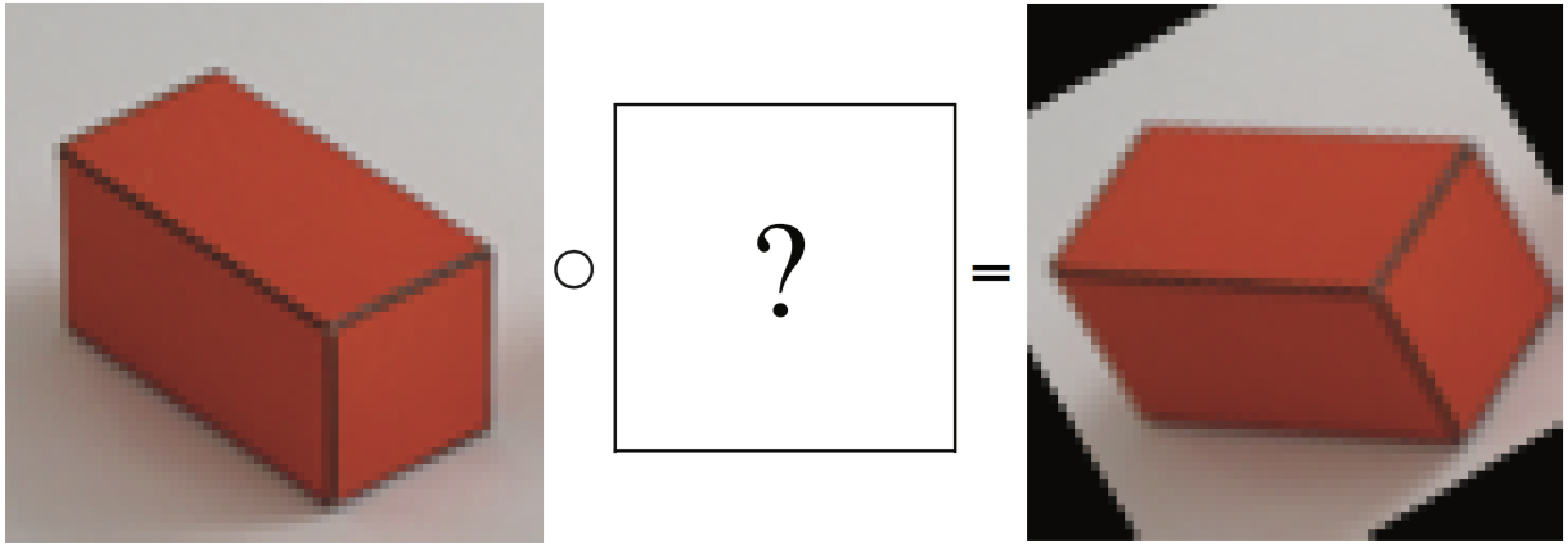
$$\circ \quad \begin{array}{|c|c|c|c|c|}\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline 1&1&1&1&1\\\hline\end{array} \quad =$$

$/25$

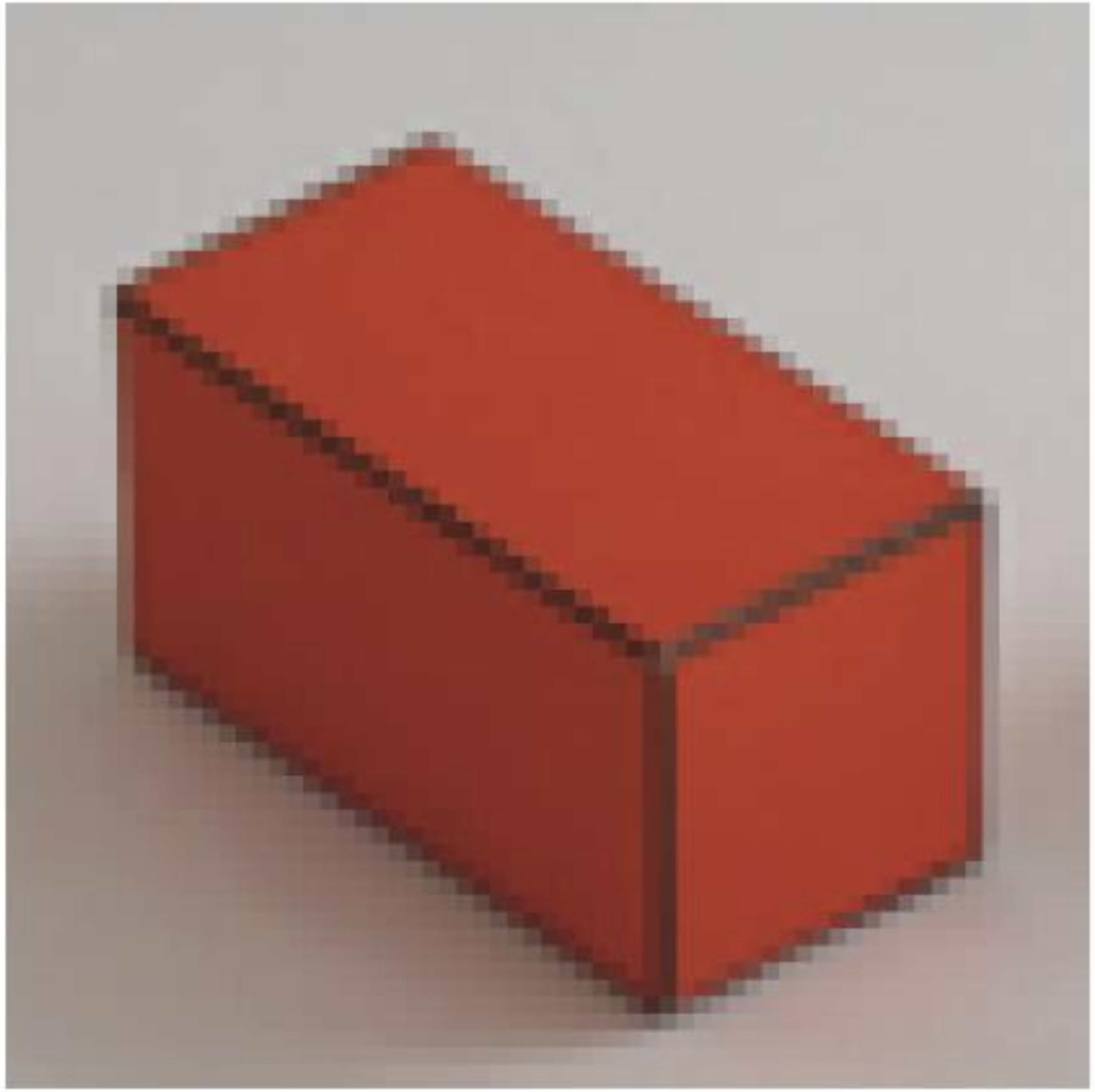


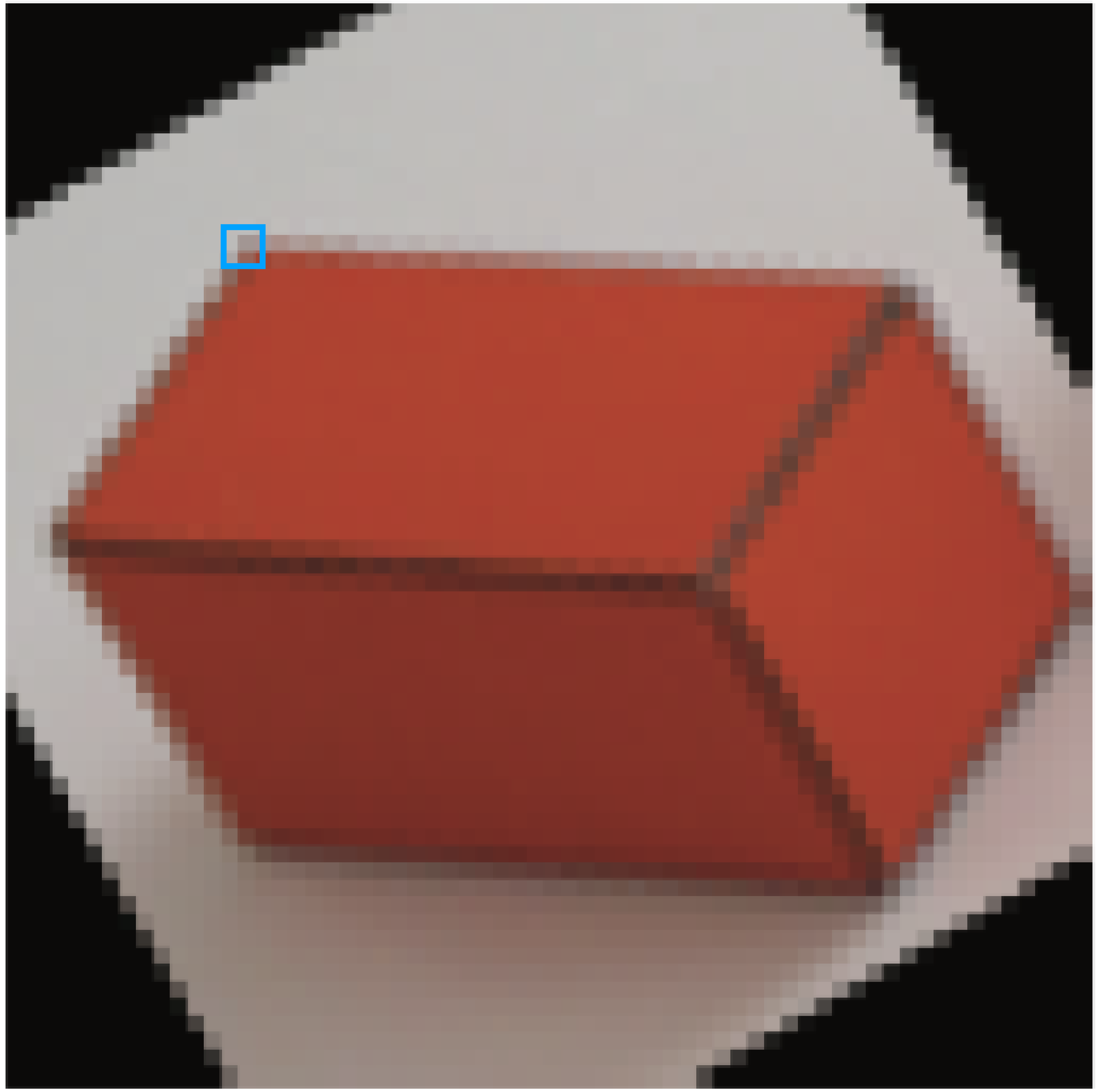
Computes the local average over windows of size 5 x 5 pixels

Examples

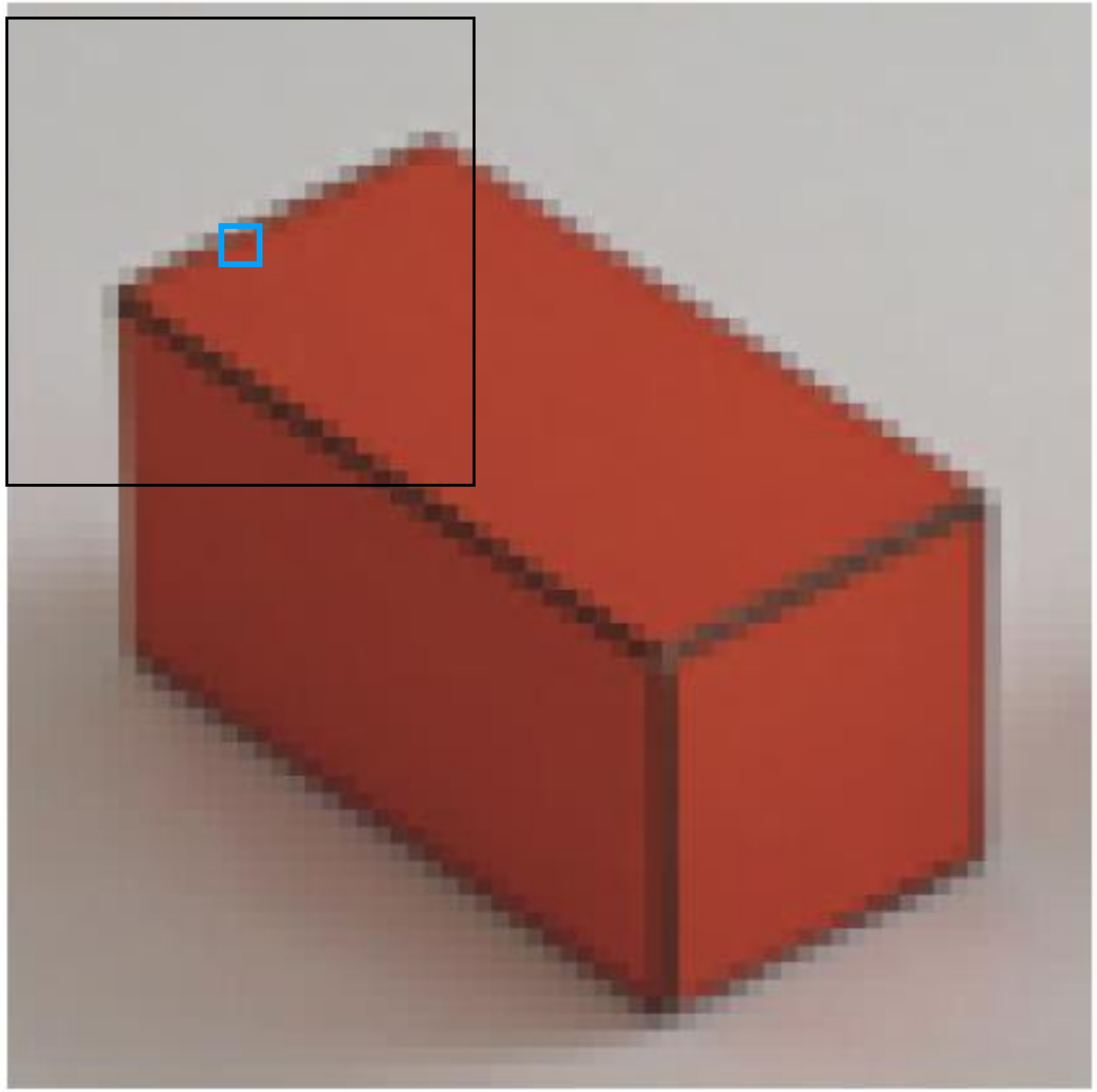


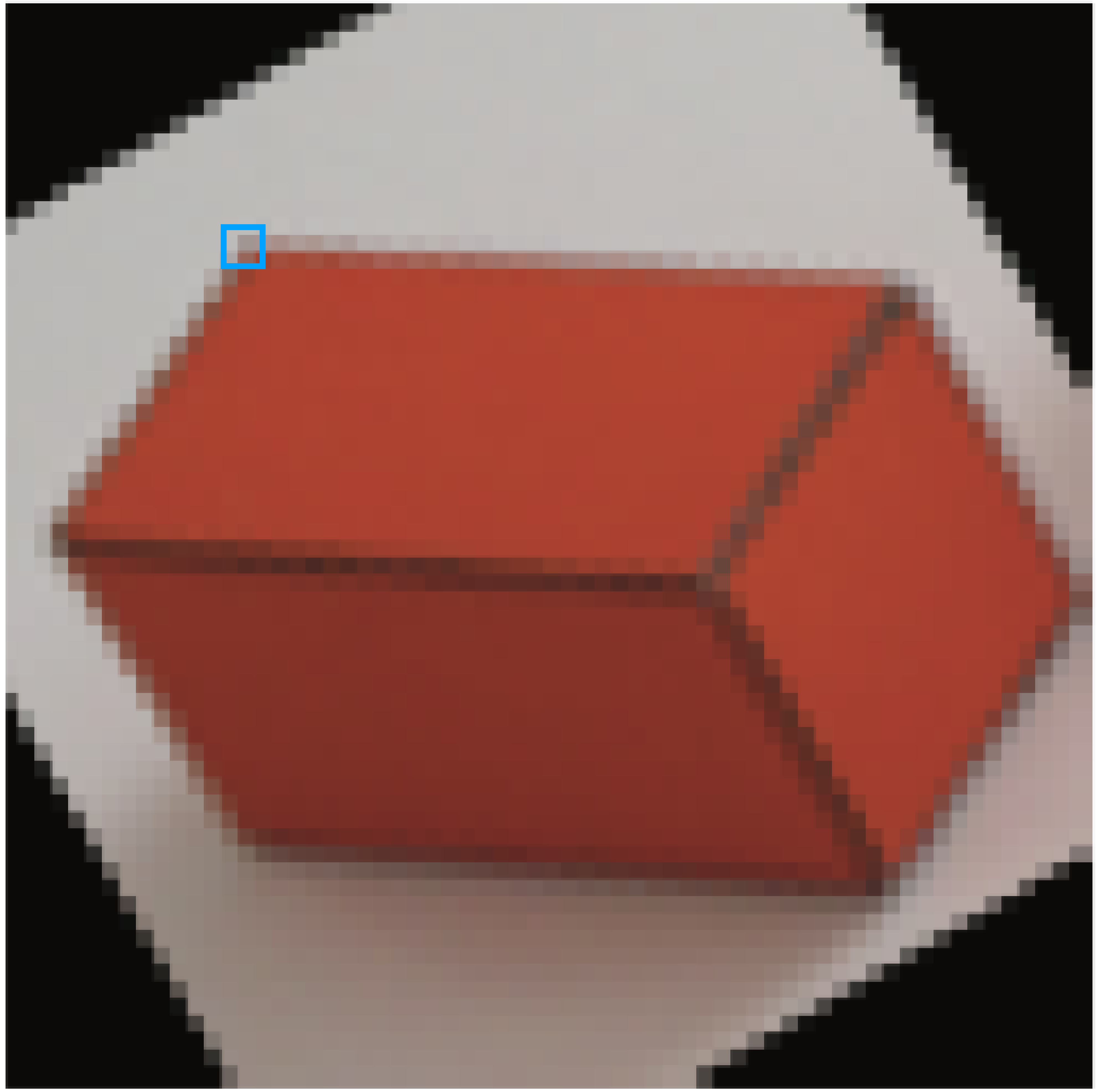
Examples



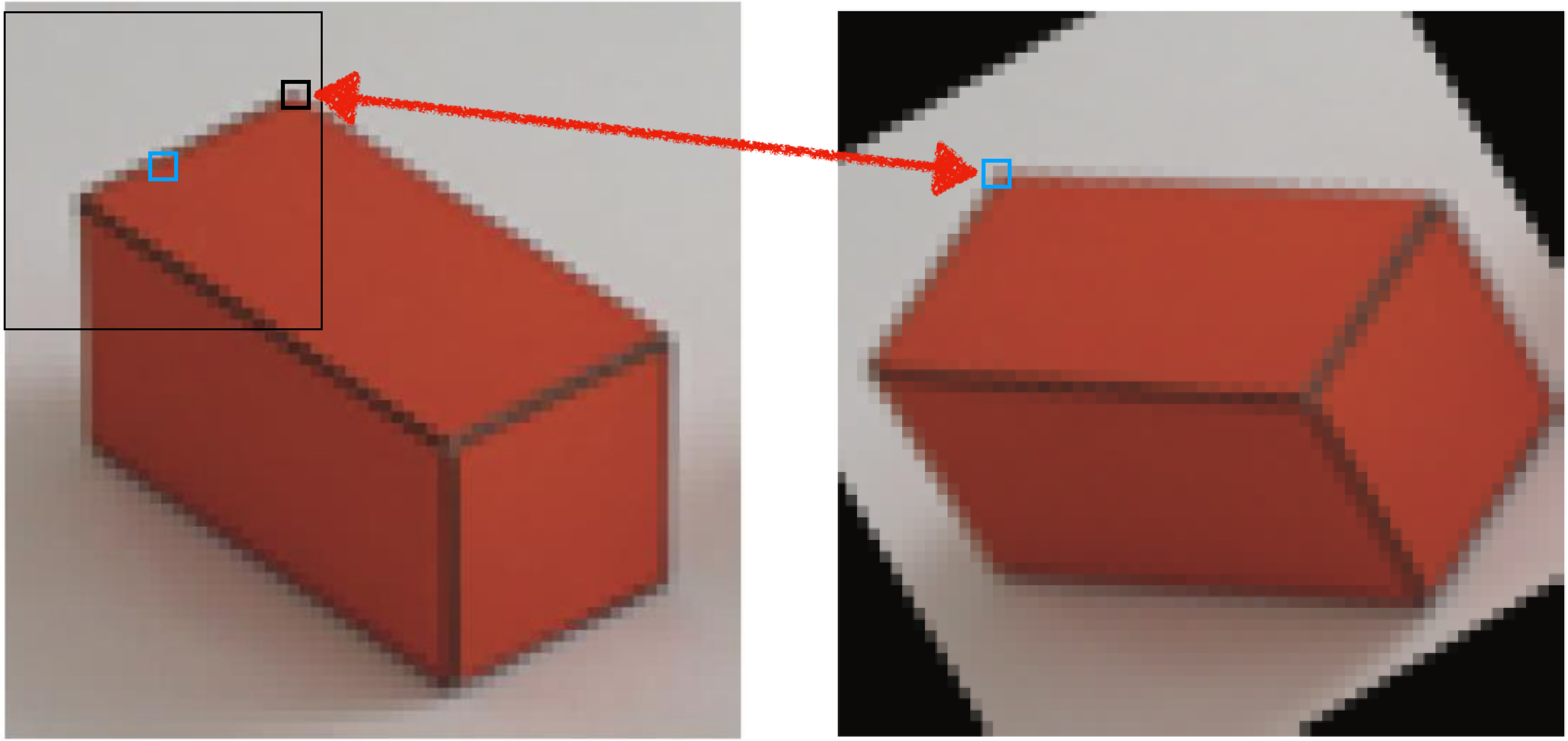


Examples

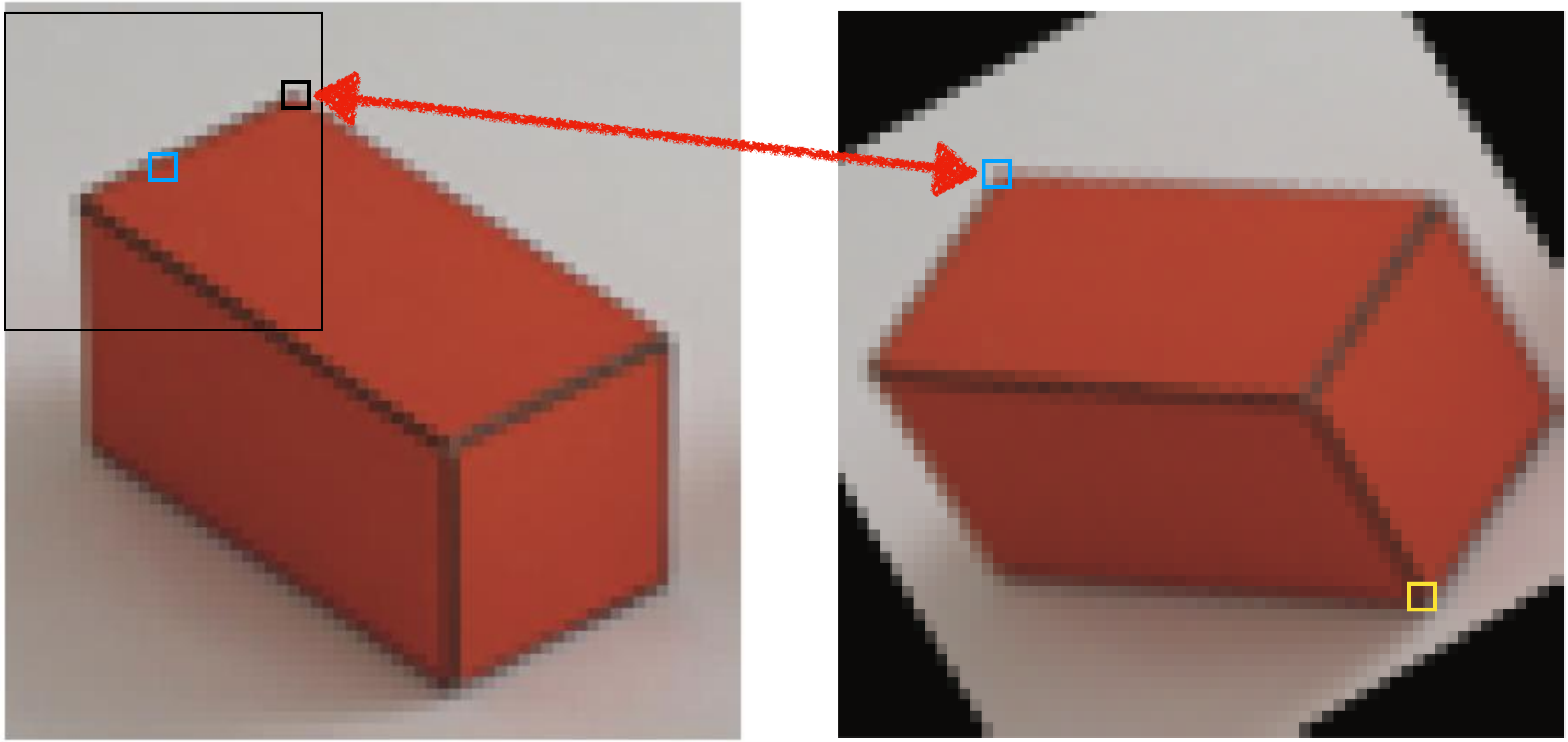




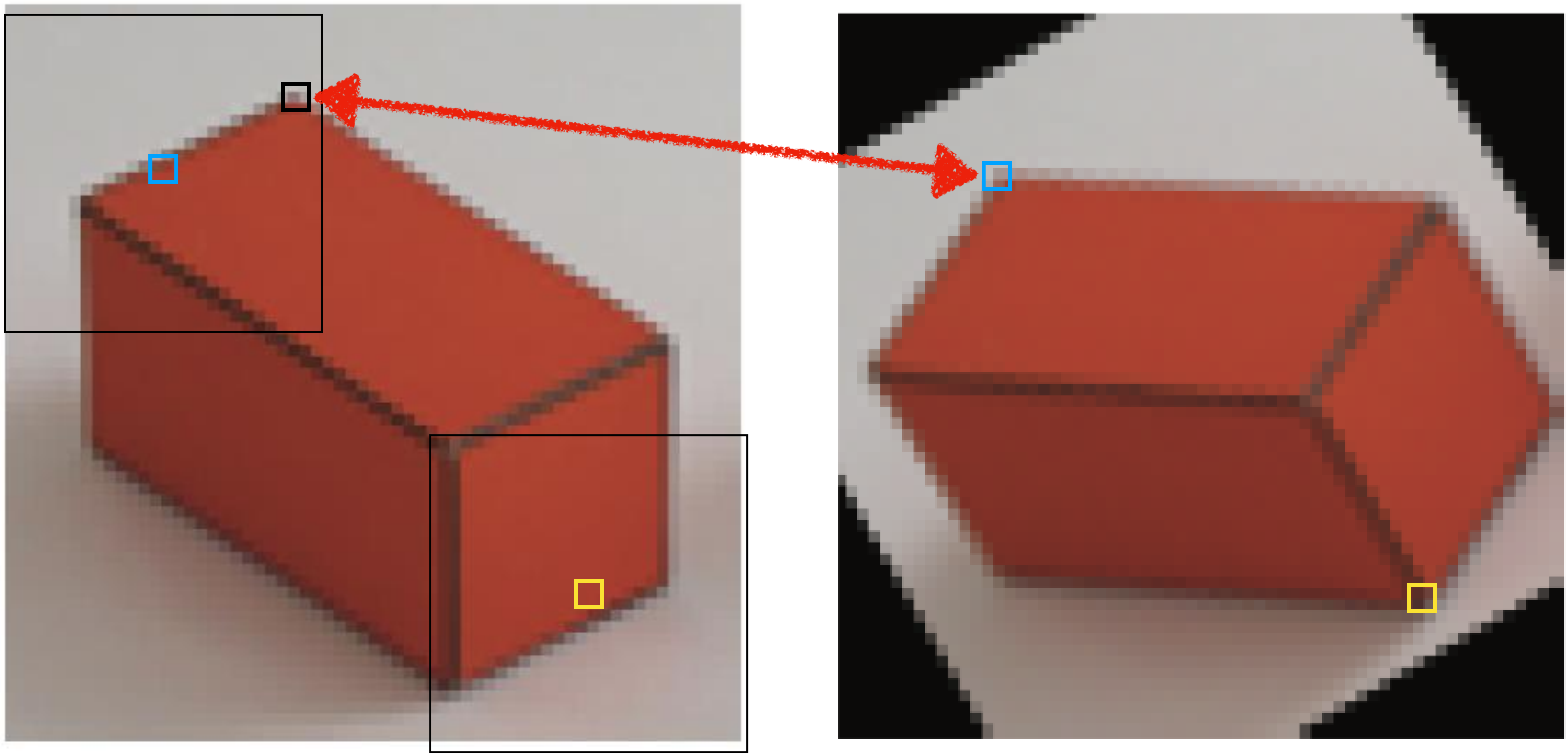
Examples



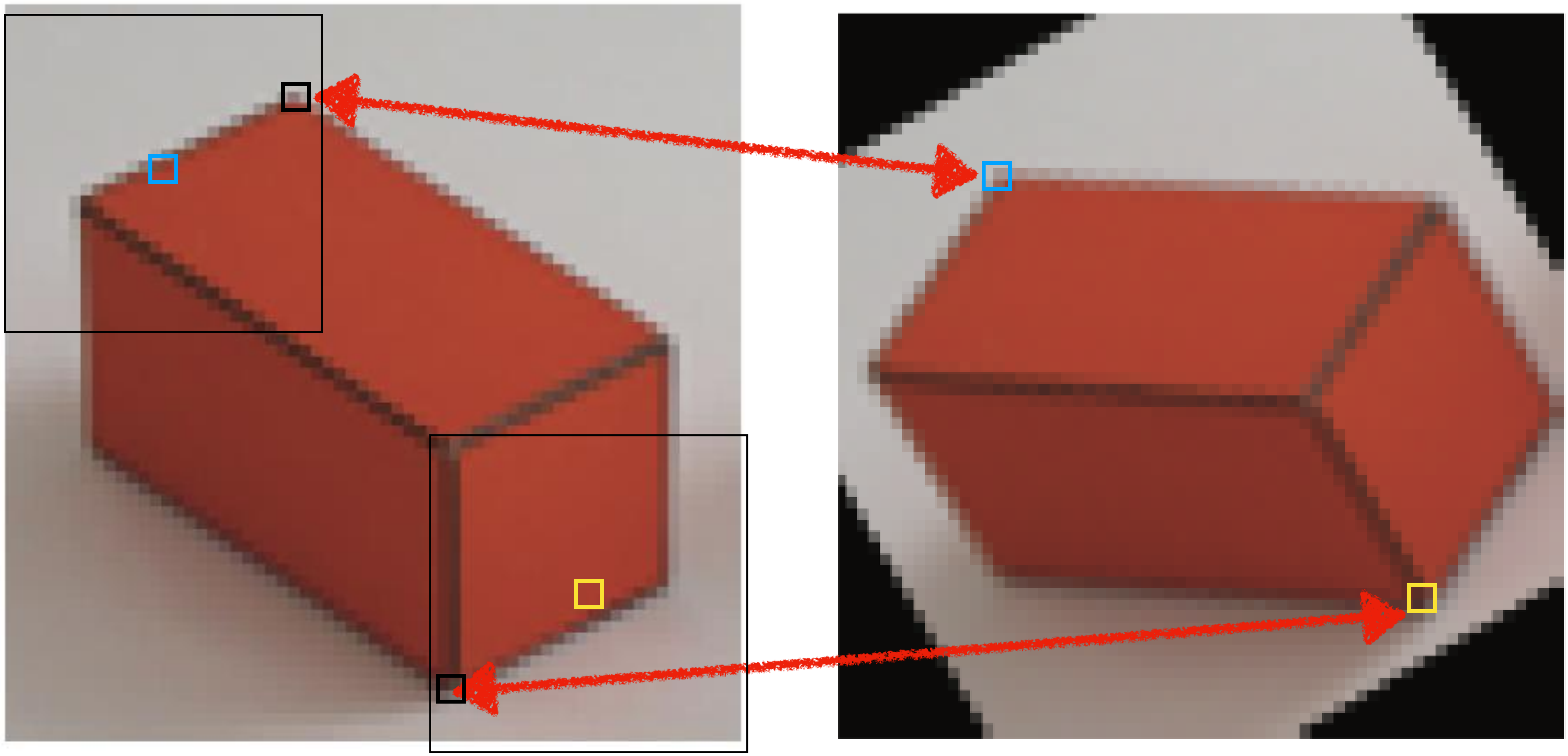
Examples



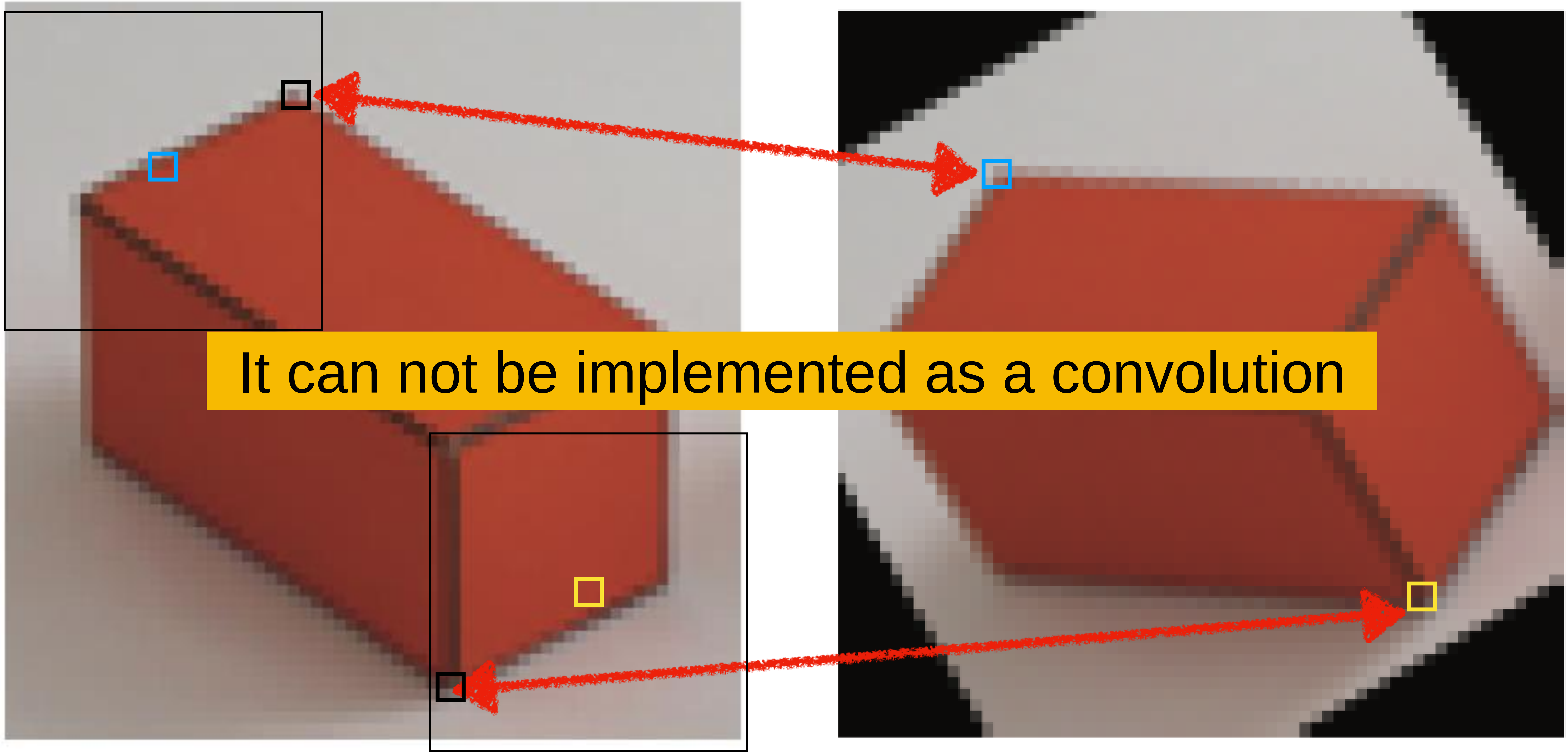
Examples



Examples



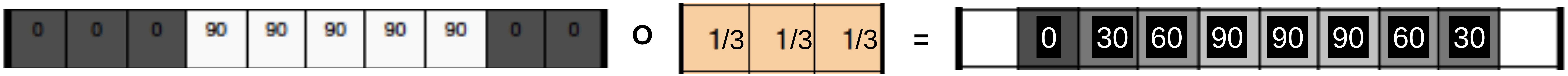
Examples



It can not be implemented as a convolution

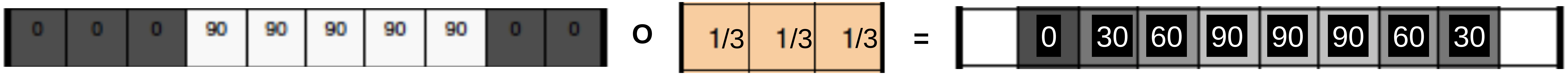
Convolution as matrix multiplication

In the 1D case, it helps to make explicit the structure of the matrix:

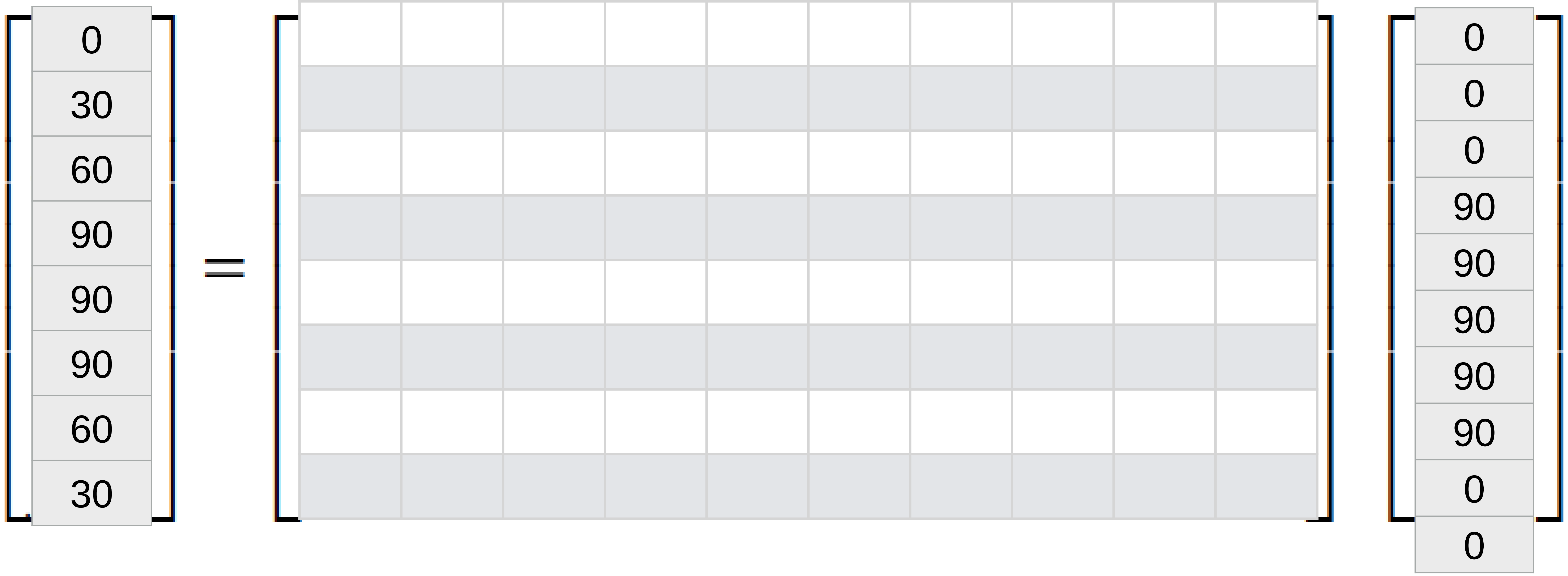


Convolution as matrix multiplication

In the 1D case, it helps to make explicit the structure of the matrix:



In the 1D case, it helps to make explicit the structure of the matrix:



Convolution as matrix multiplication

In the 1D case, it helps to make explicit the structure of the matrix:

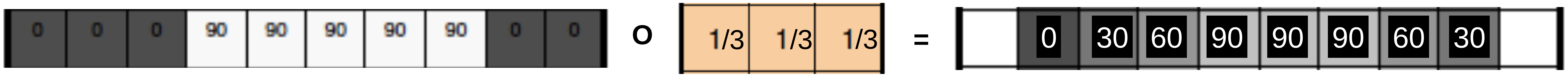
$$\begin{bmatrix} 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix} = \begin{bmatrix} & 0 & 30 & 60 & 90 & 90 & 90 & 60 & 30 & \end{bmatrix}$$

In the 1D case, it helps to make explicit the structure of the matrix:

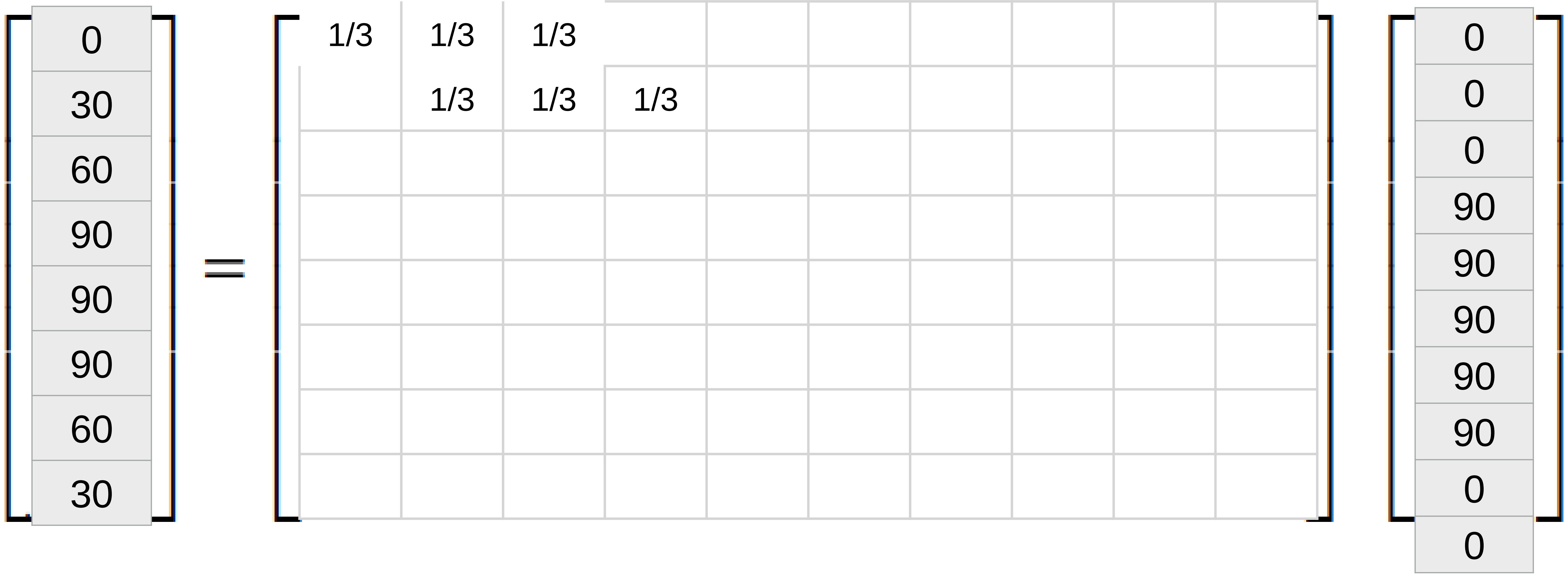
$$\begin{bmatrix} 0 \\ 30 \\ 60 \\ 90 \\ 90 \\ 90 \\ 60 \\ 30 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 90 \\ 90 \\ 90 \\ 90 \\ 90 \\ 0 \\ 0 \end{bmatrix}$$

Convolution as matrix multiplication

In the 1D case, it helps to make explicit the structure of the matrix:

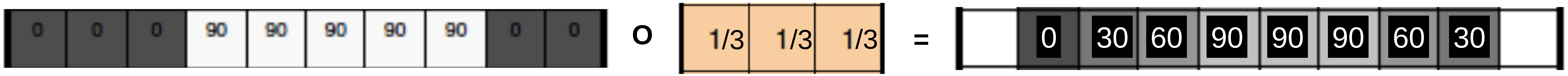


In the 1D case, it helps to make explicit the structure of the matrix:

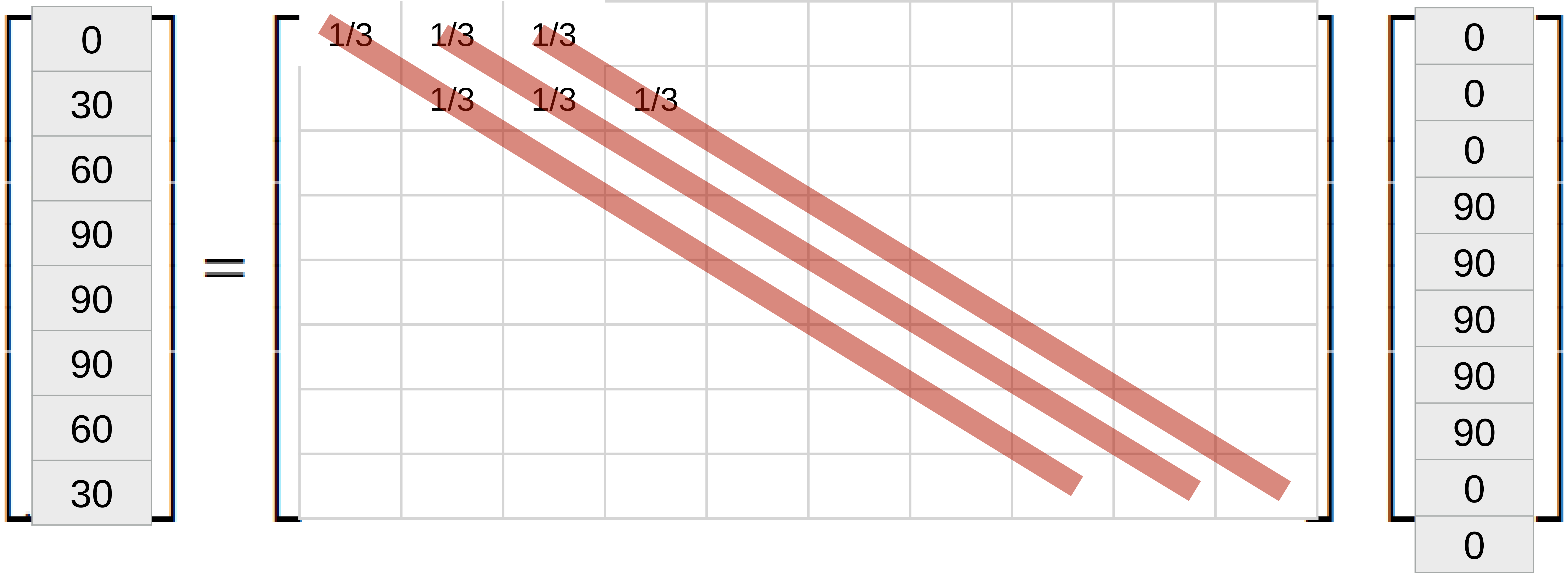


Convolution as matrix multiplication

In the 1D case, it helps to make explicit the structure of the matrix:

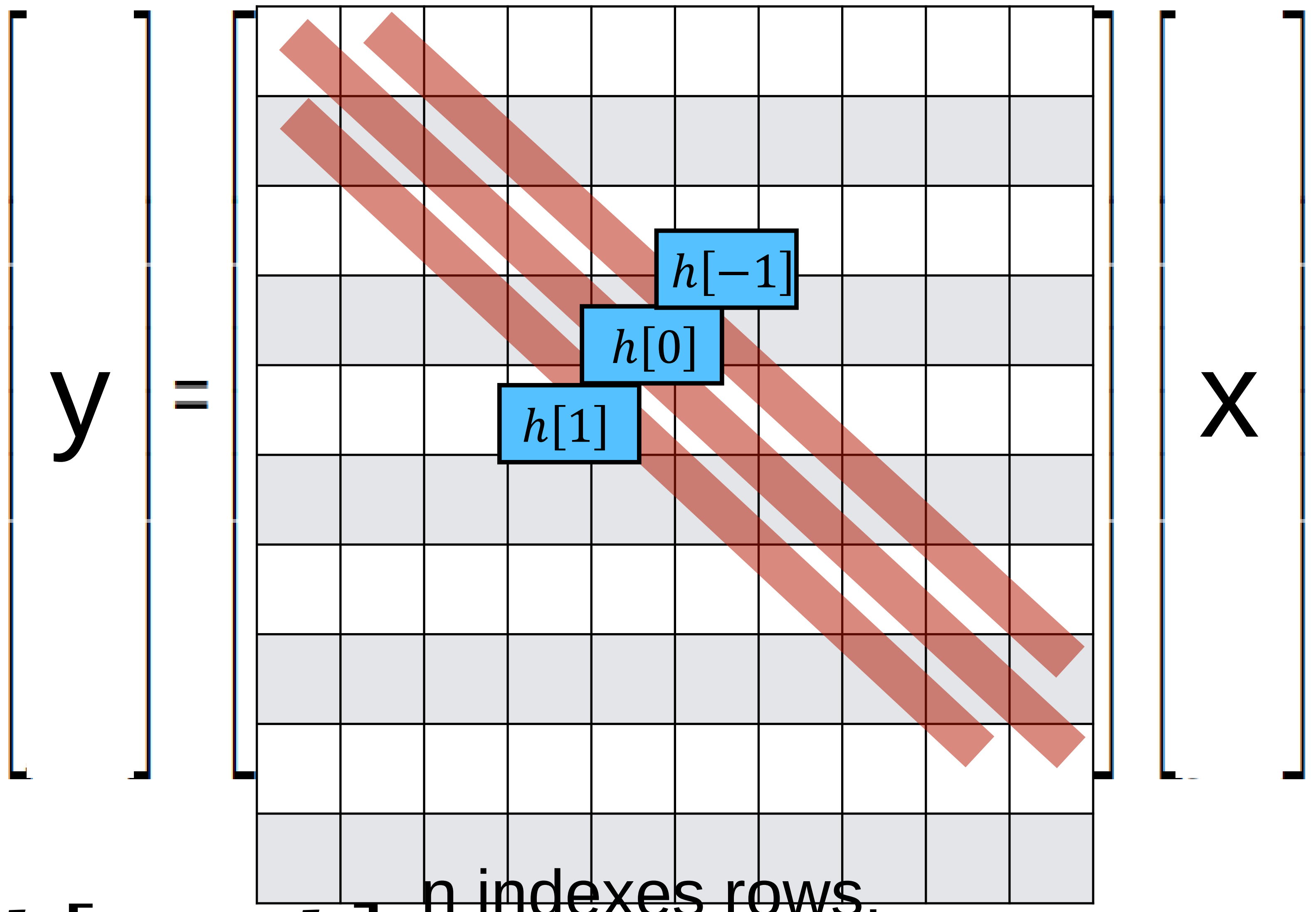


In the 1D case, it helps to make explicit the structure of the matrix:



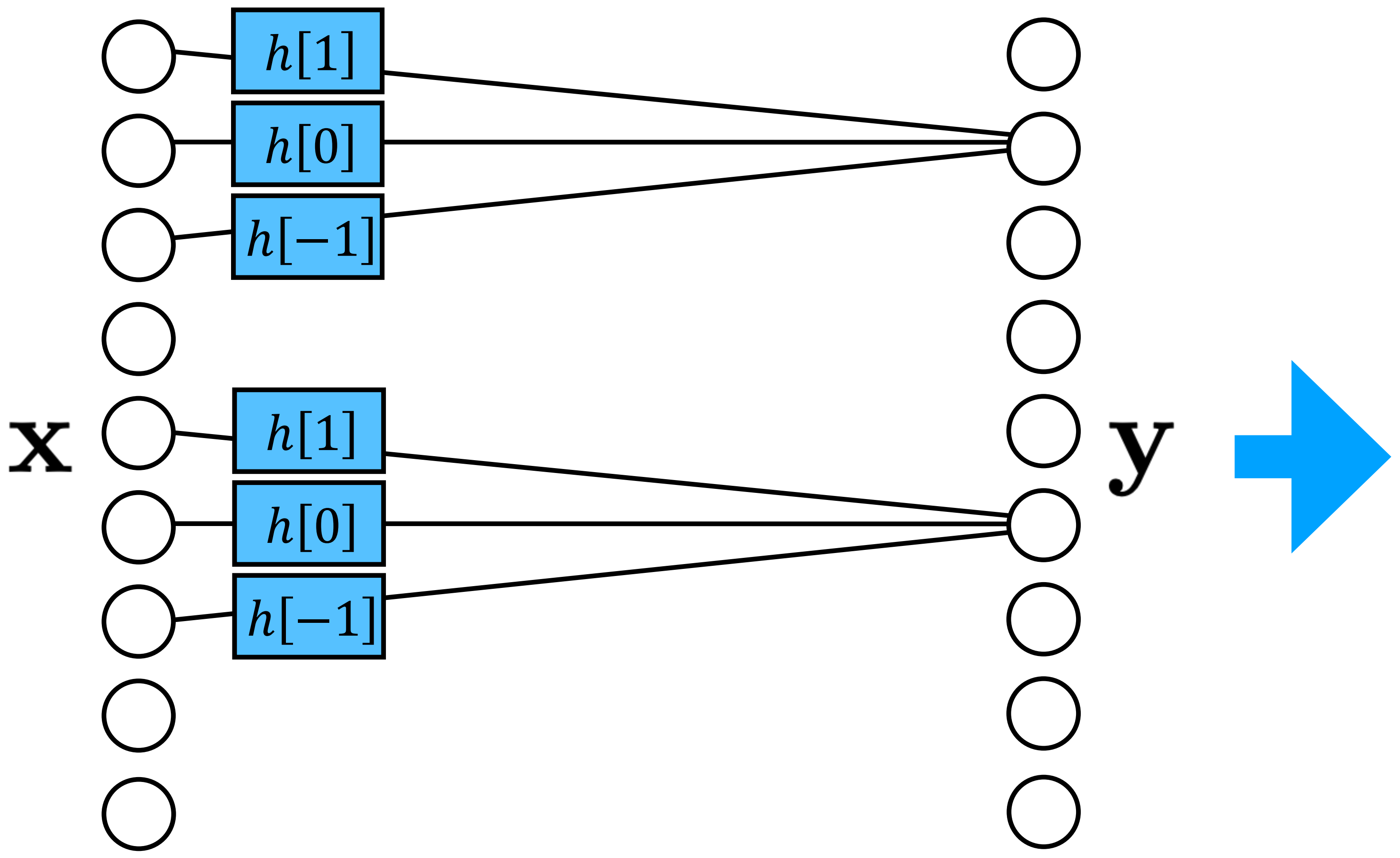
Linear translation invariant system:

A LTI function f can be written as a matrix multiplication:



$h[n-k]$ n indexes rows,
k indexes columns

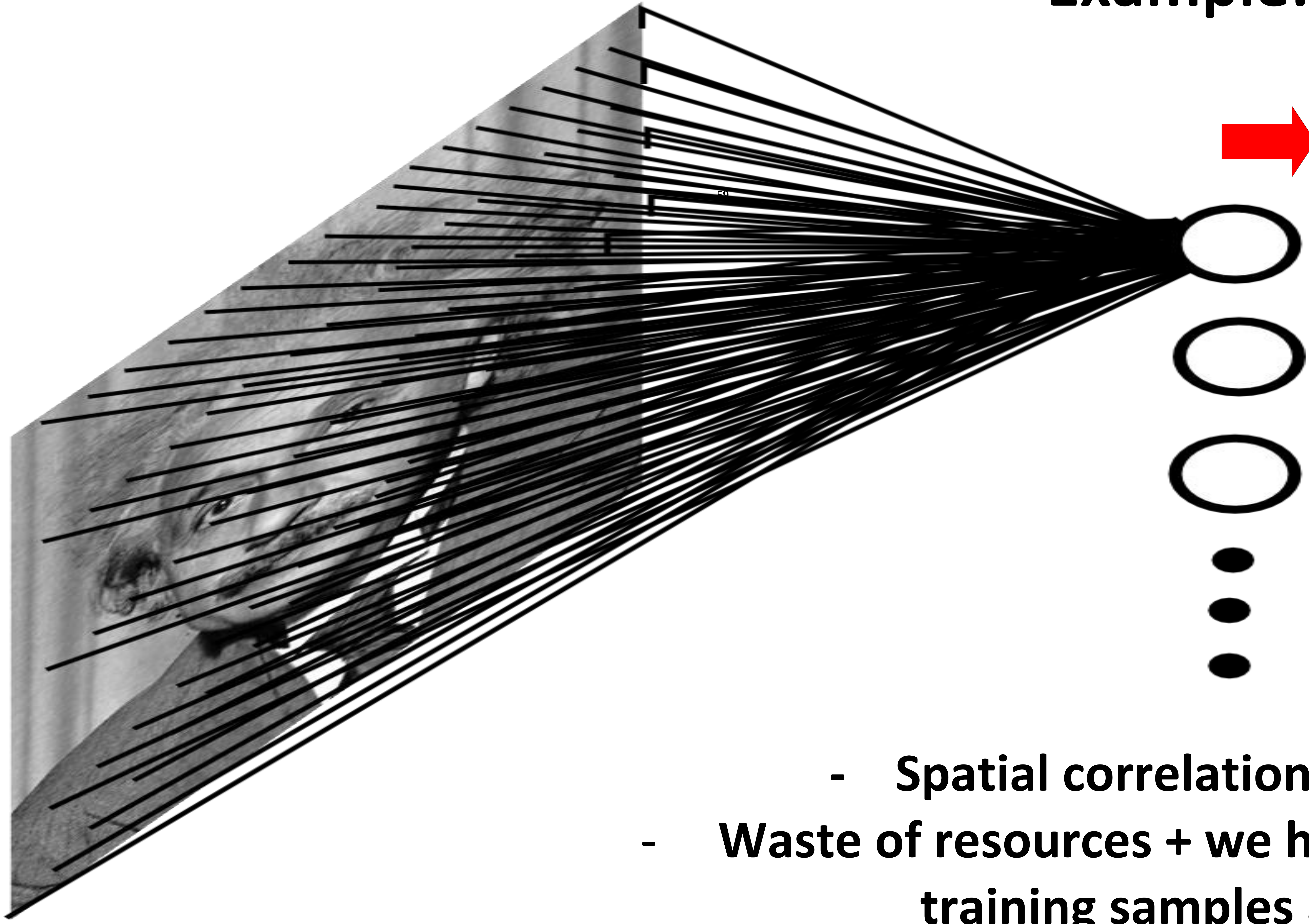
It can also be represented as a convolutional layer of neural net:



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

$h[n-k]$ Is the strength of the connection between x[k] and y[n]

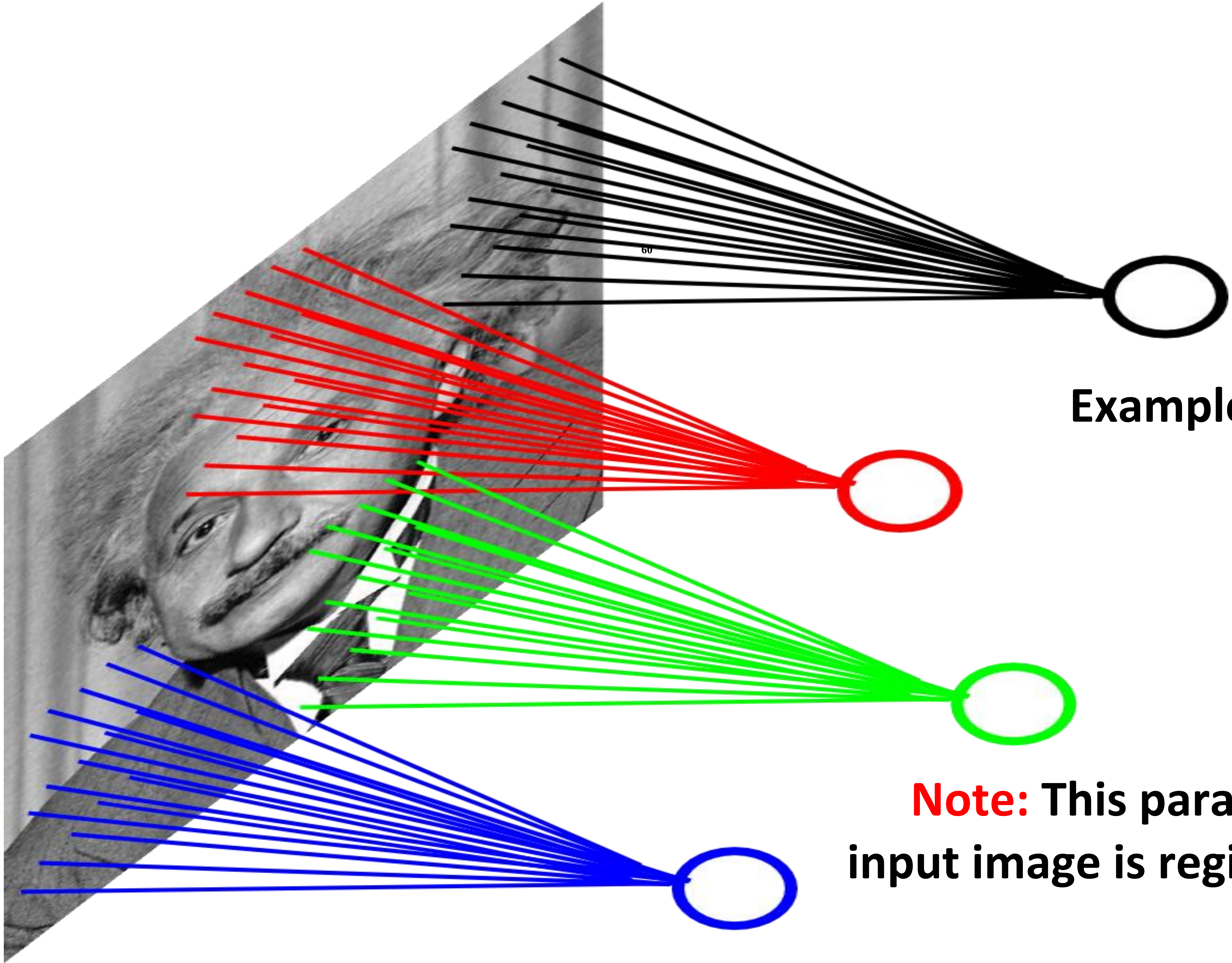
Fully Connected Layer



Example: 200x200 image
40K hidden units
~2B parameters!!!

- Spatial correlation is local
- Waste of resources + we have not enough training samples anyway..

Locally Connected Layer

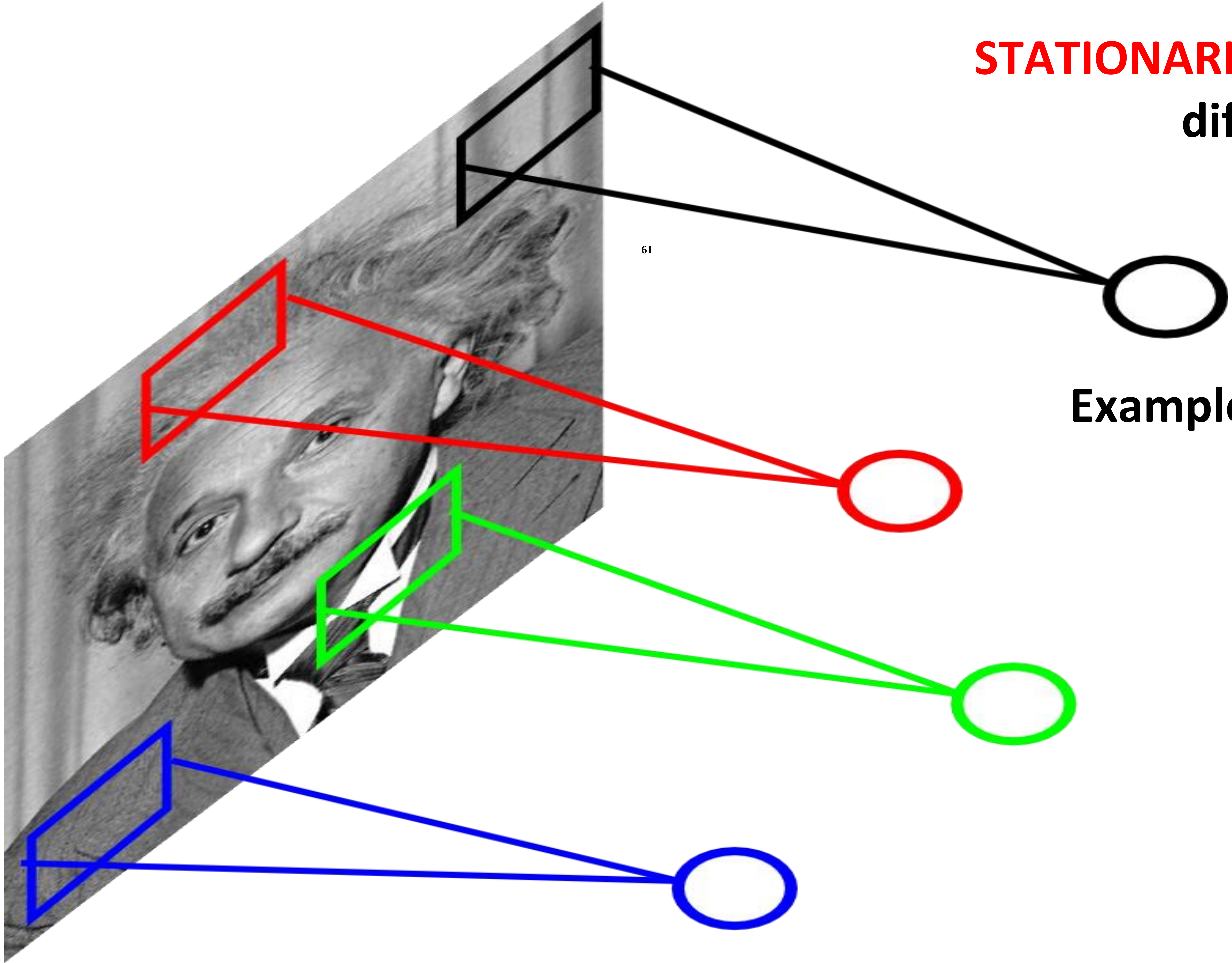


Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

Ranzato

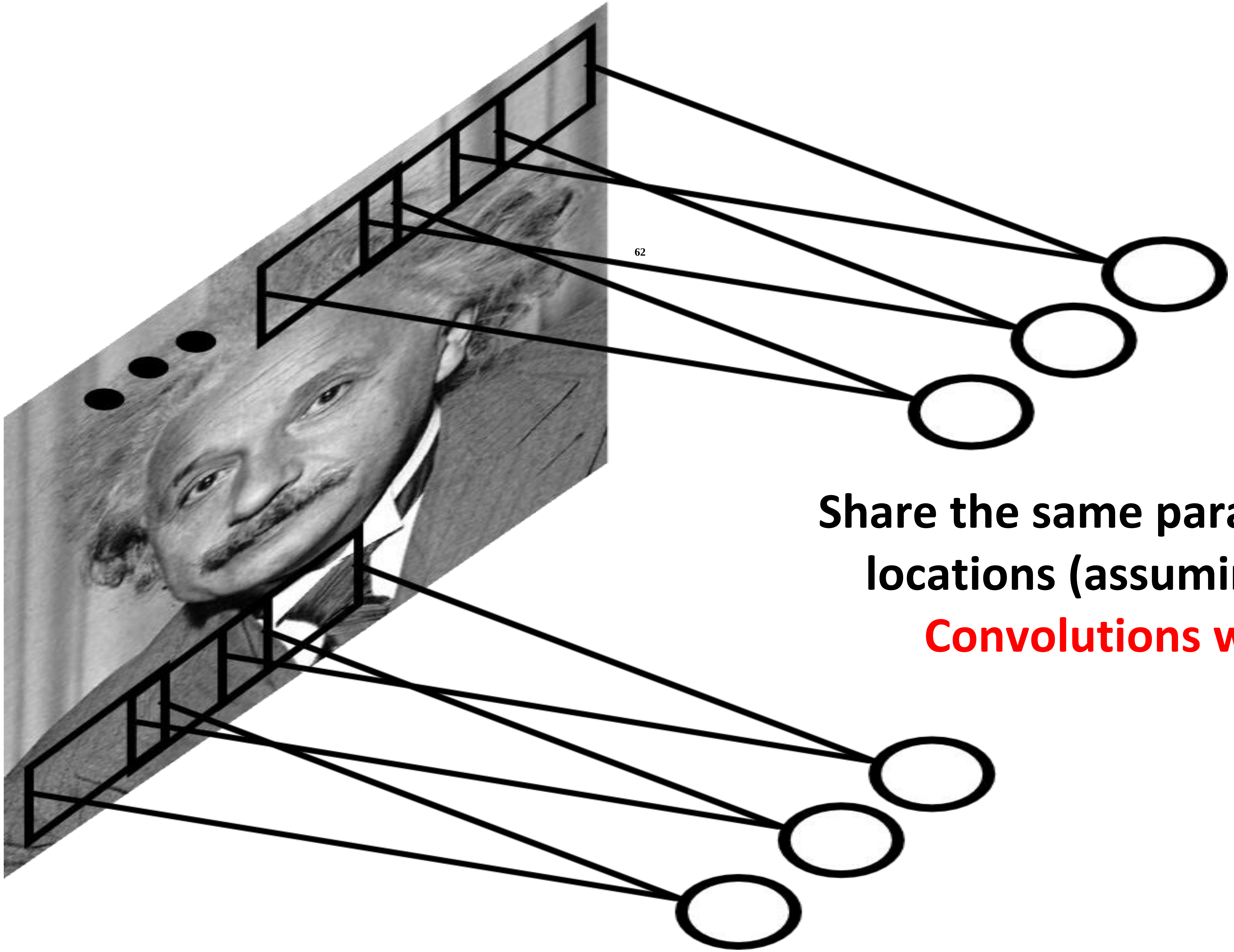
Locally Connected Layer



STATIONARITY? Statistics is similar at different locations

Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Convolutional Layer



Share the same parameters across different locations (assuming input is stationary):
Convolutions with learned kernels

Rectangular filter

⊗

h[m,n]

=

f[m,n]

g[m,n]

Rectangular filter

$\otimes$

=

g[m,n]

h[m,n]

f[m,n]

Rectangular filter

g[m,n]

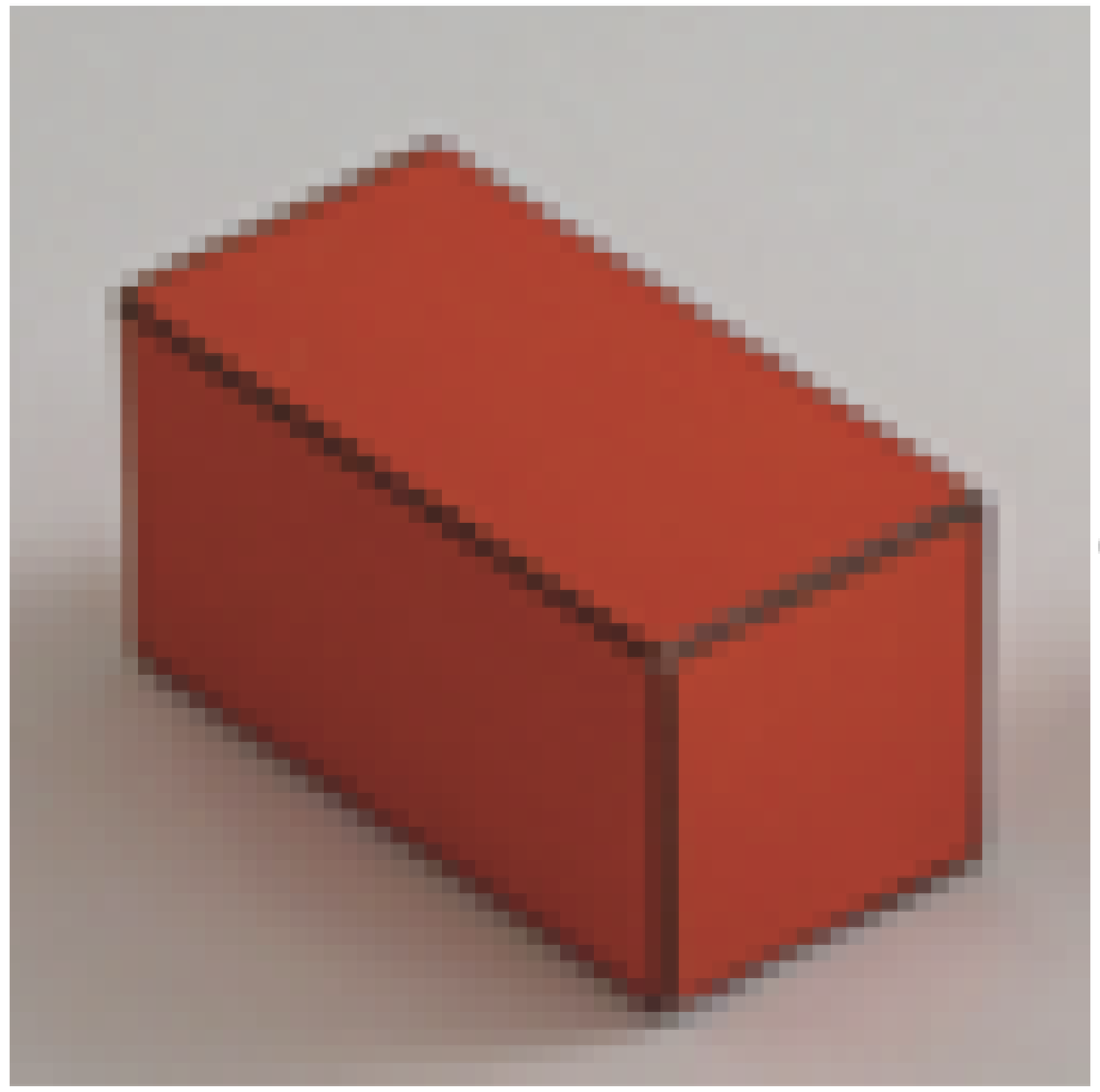
$\otimes$

h[m,n]

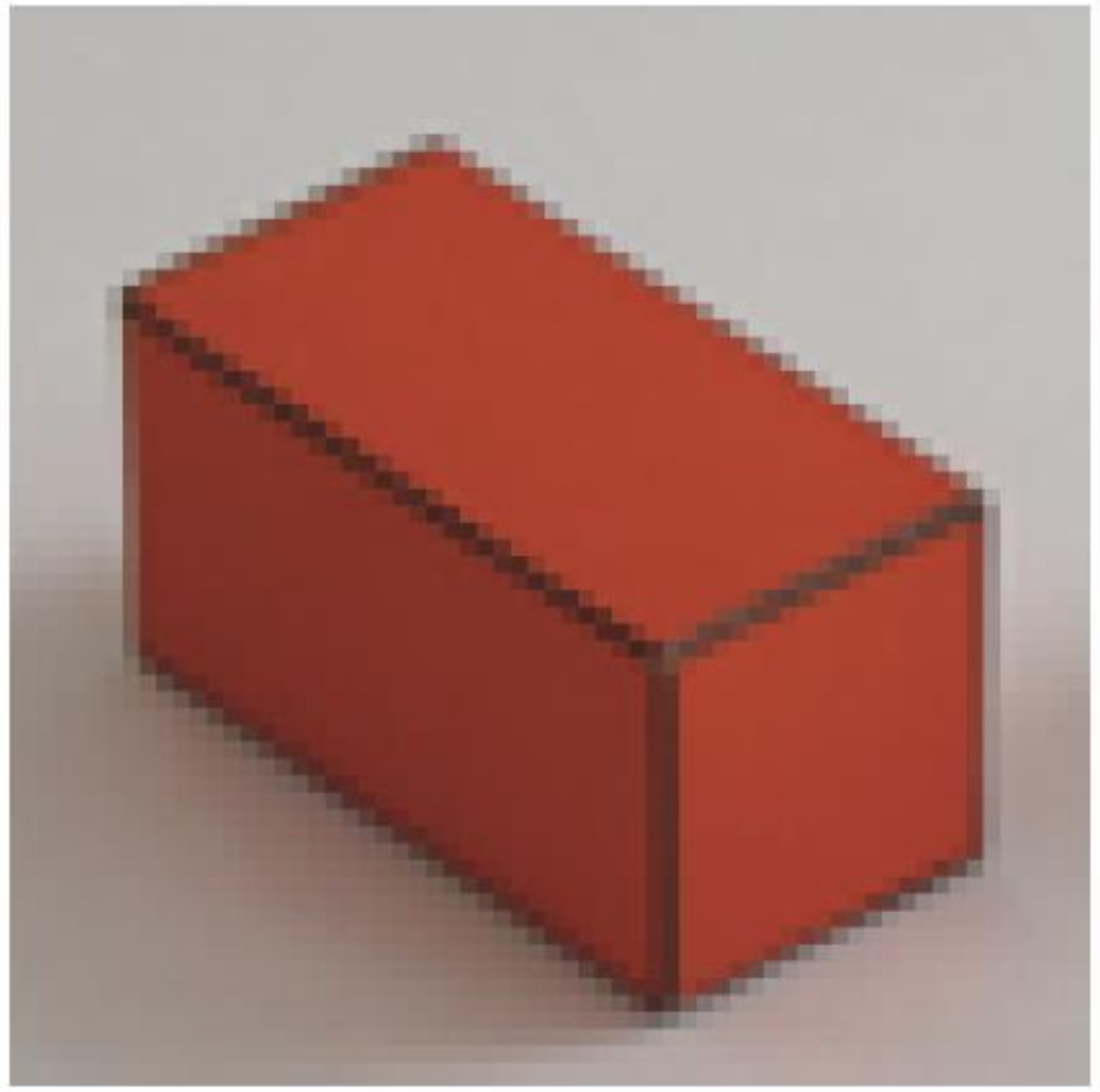
=

f[m,n]

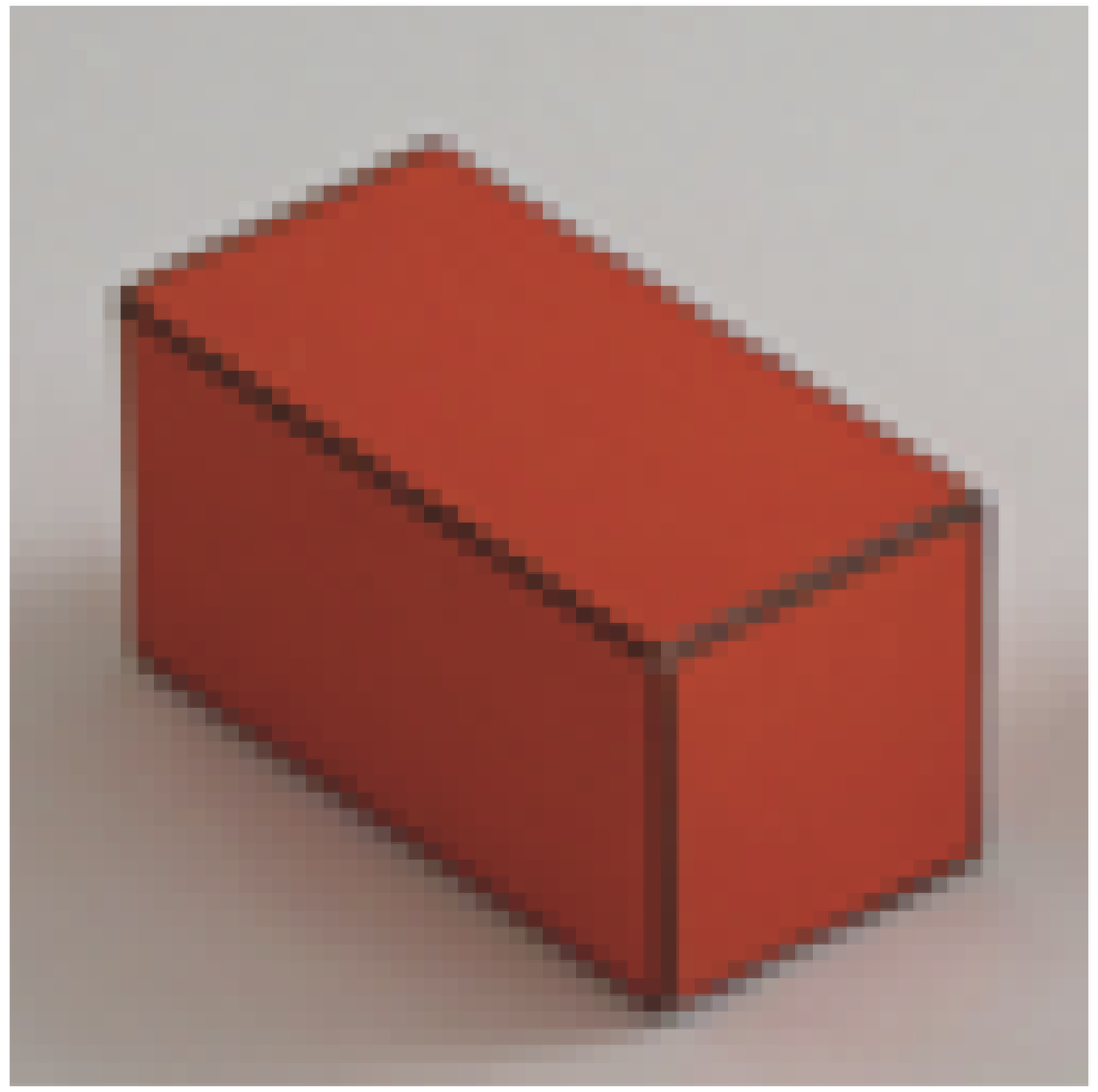
The identity



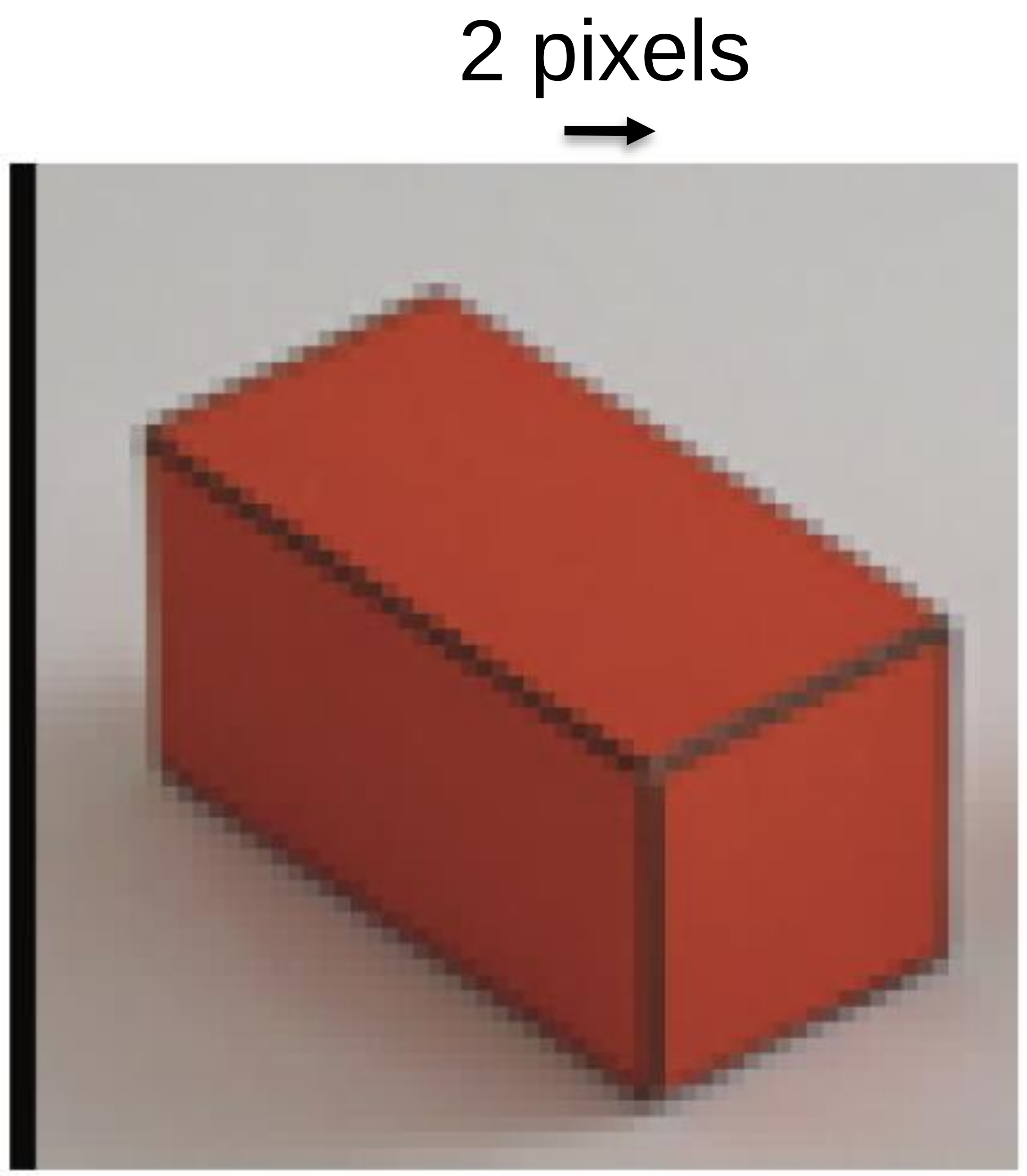
$$\circ \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} =$$



A shift

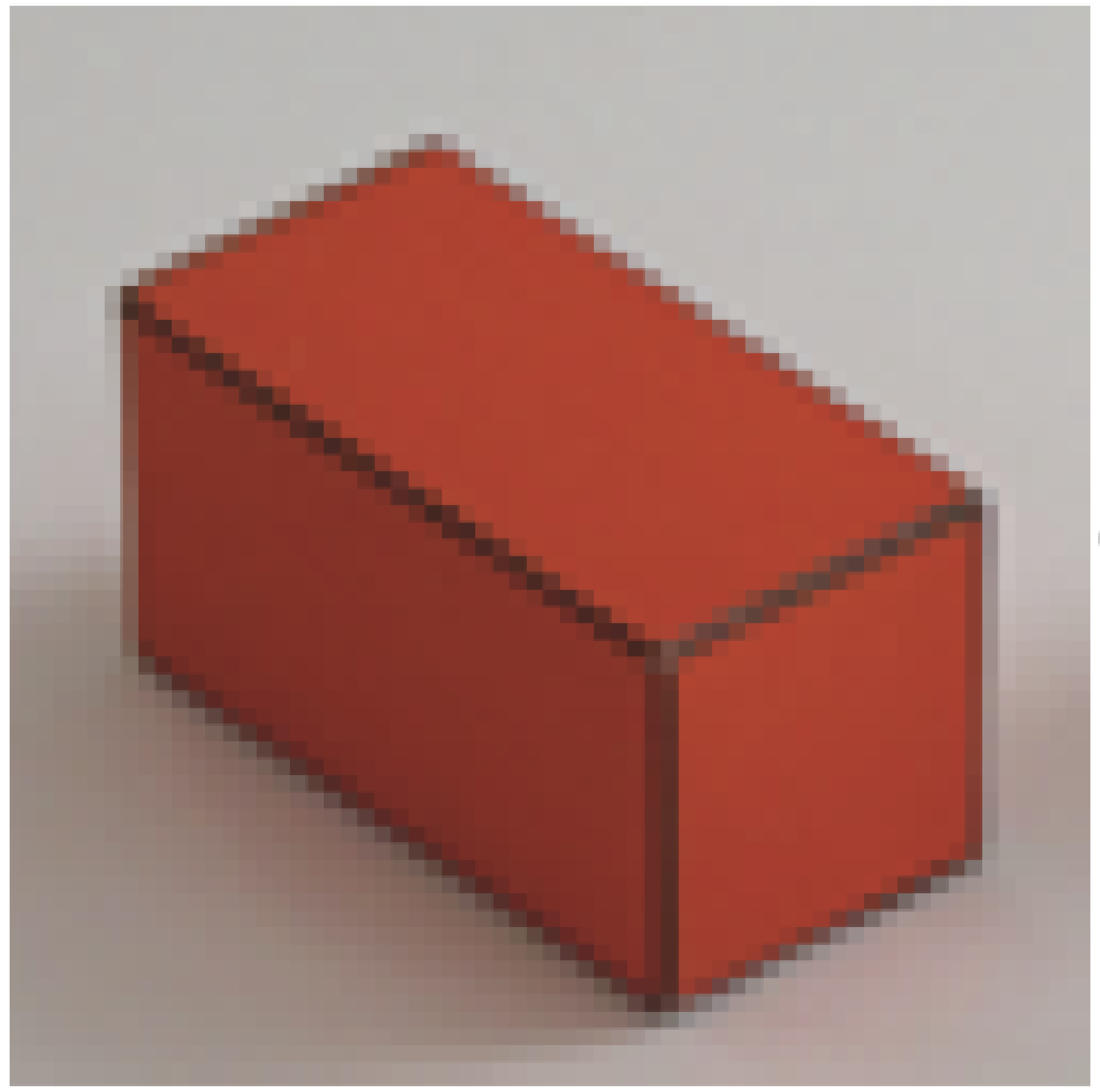


$$\circ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} =$$

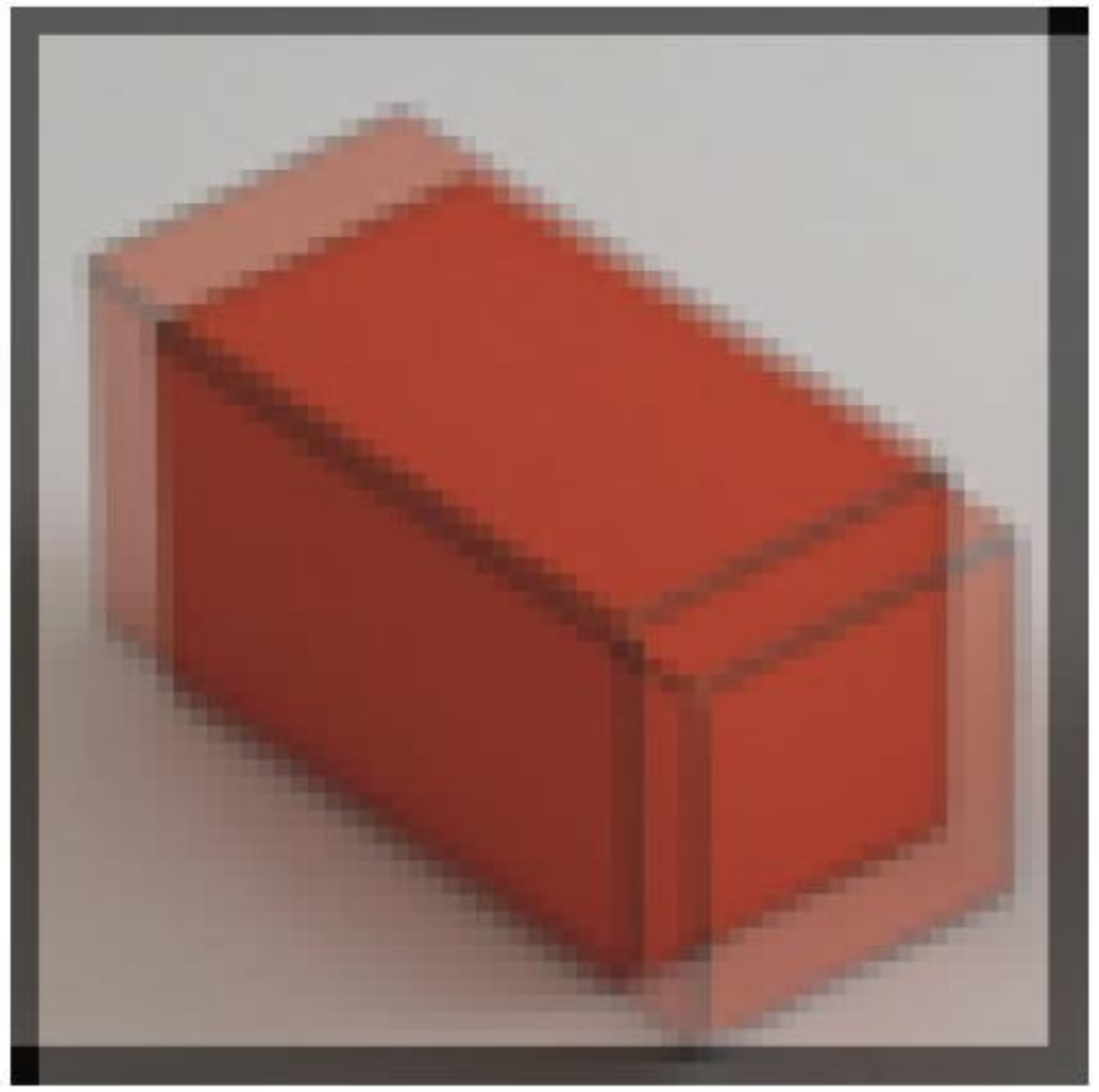


(using zero padding)

Examples



$$\circ \begin{array}{|c|c|c|c|c|}\hline .5 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & .5 \\ \hline \end{array} =$$



“naturally” occurring filters

When we take a picture from a moving car, the resulting picture can be affected by motion blur

a)

Input image

“naturally” occurring filters

When we take a picture from a moving car, the resulting picture can be affected by motion blur

a) 

Input image

c) 

Motion blur

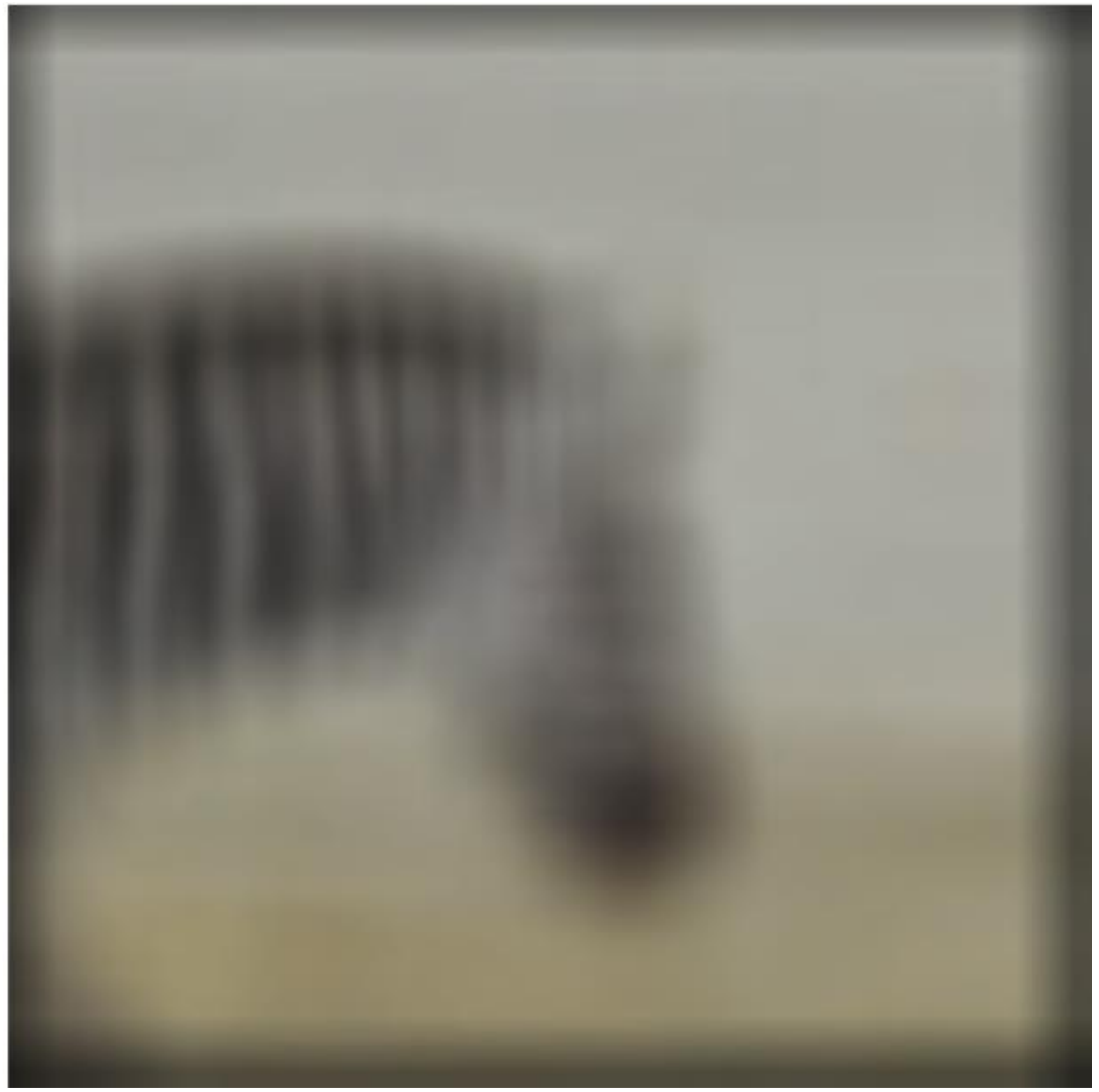
Handling boundaries

Handling boundaries

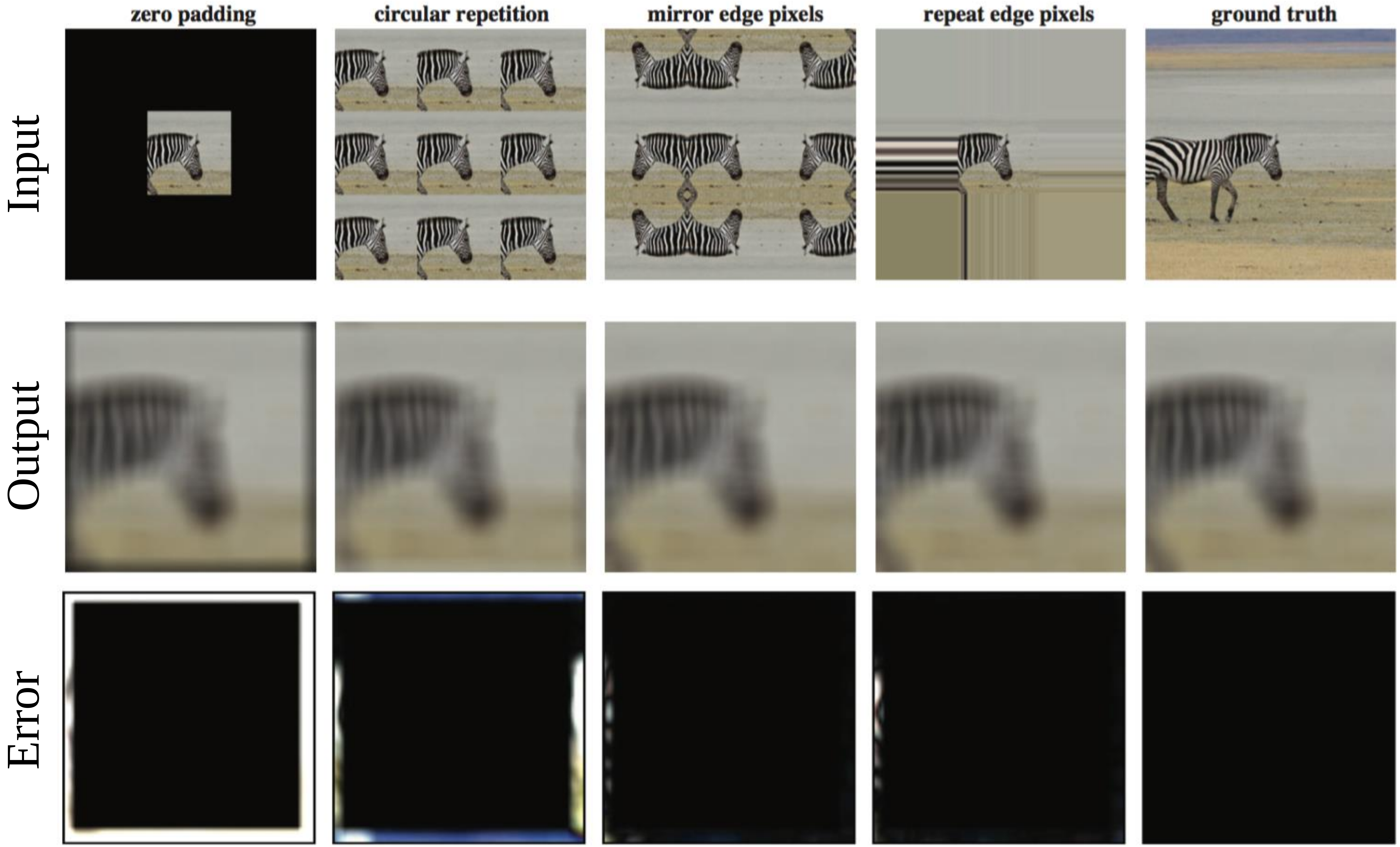
Zero padding

$\circ$ =

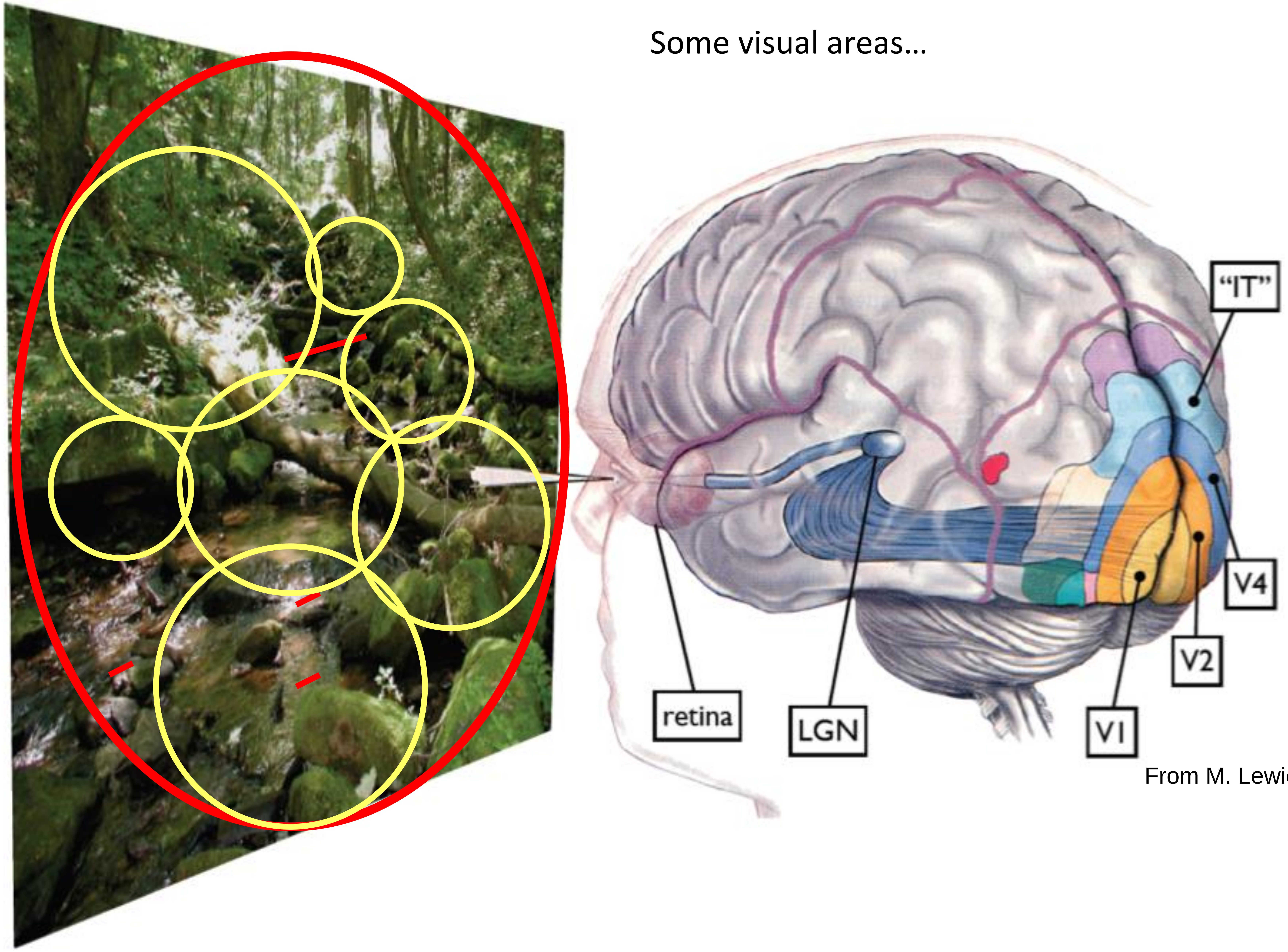
11x11 ones



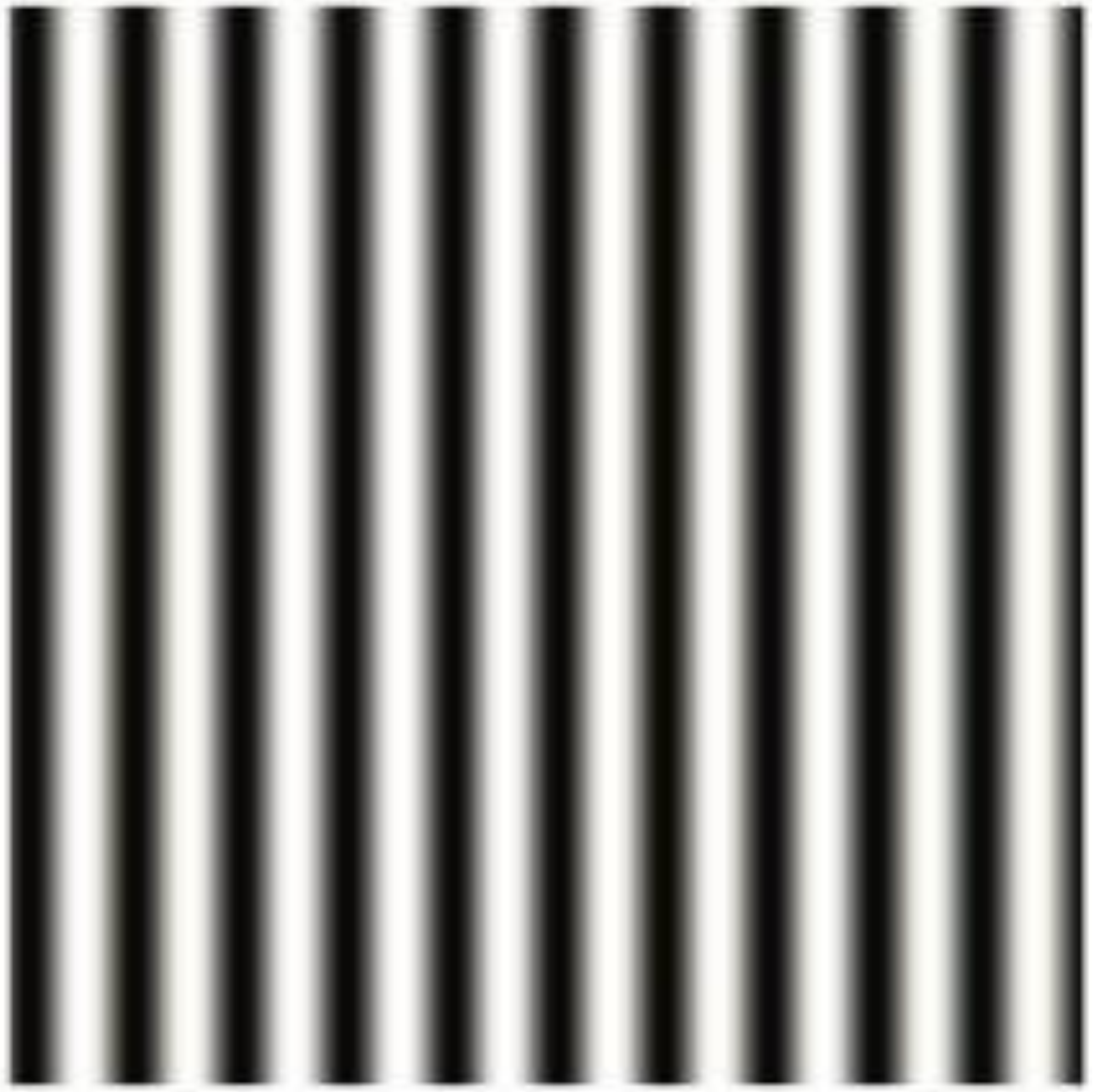
Handling boundaries



Some visual areas...



From M. Lewicky



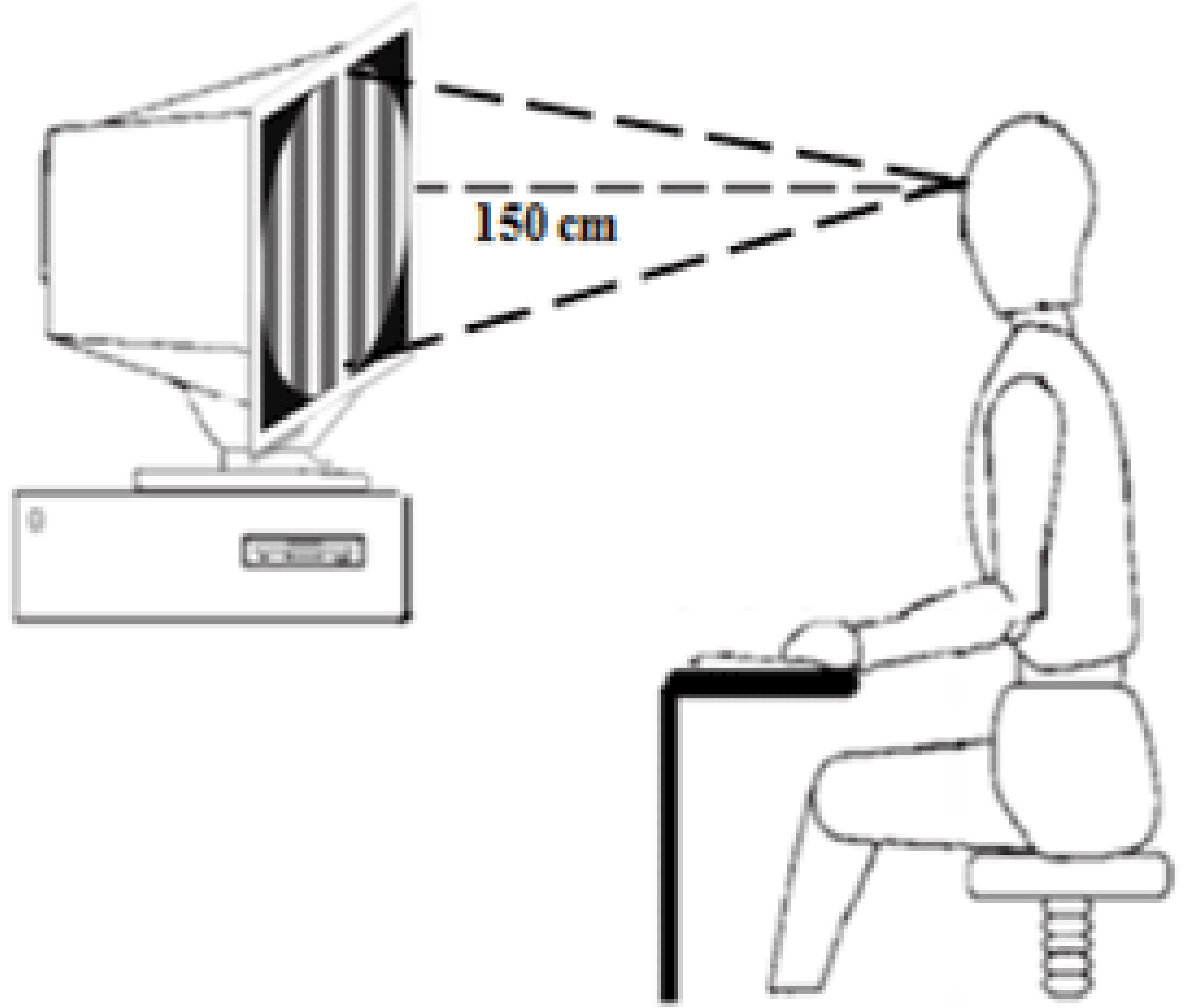


Figure 1. Stimulus presentation scheme. The stimuli were originally calibrated to be seen at a distance of 150 cm in a 19″ display.

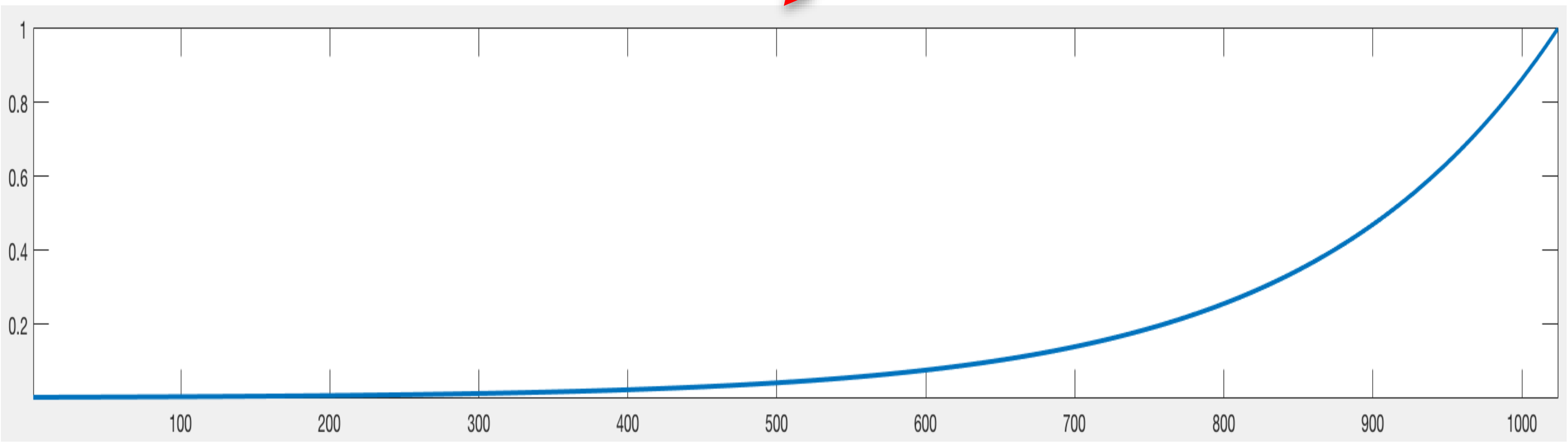
Campbell & Robson chart

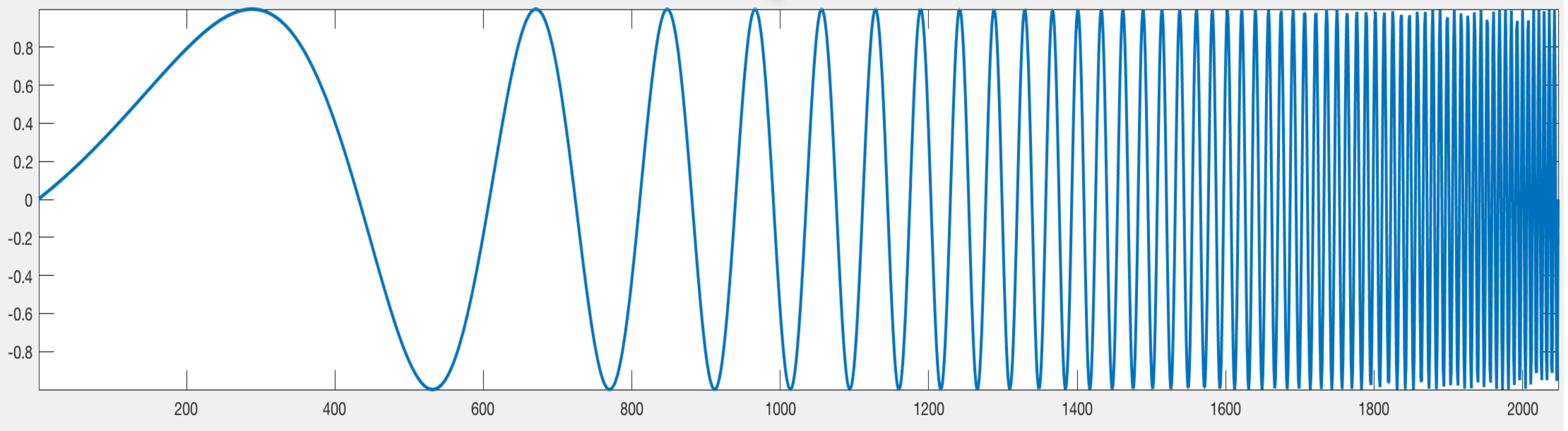
Let's define the following image:

$$\mathbf{I}[n,m] = \boxed{A[n]}\,\boxed{\sin(2\pi f[m]\, m/M)}$$

With:

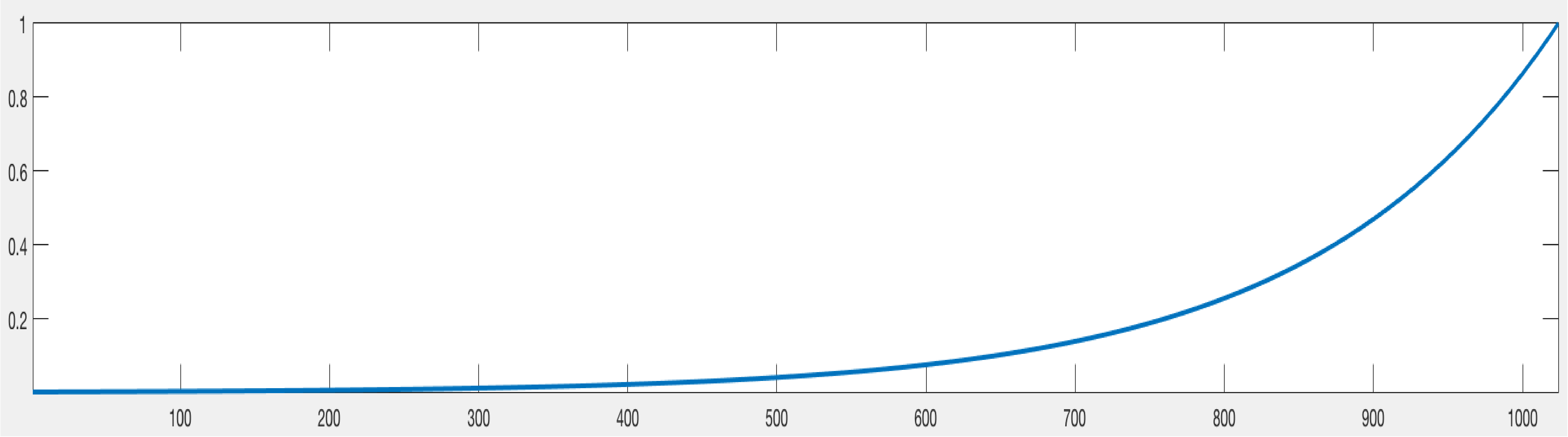
$$A[n] = A_{min}\left(\frac{A_{max}}{A_{min}}\right)^{n/N} \qquad f[m] = f_{min}\left(\frac{f_{max}}{f_{min}}\right)^{m/M}$$

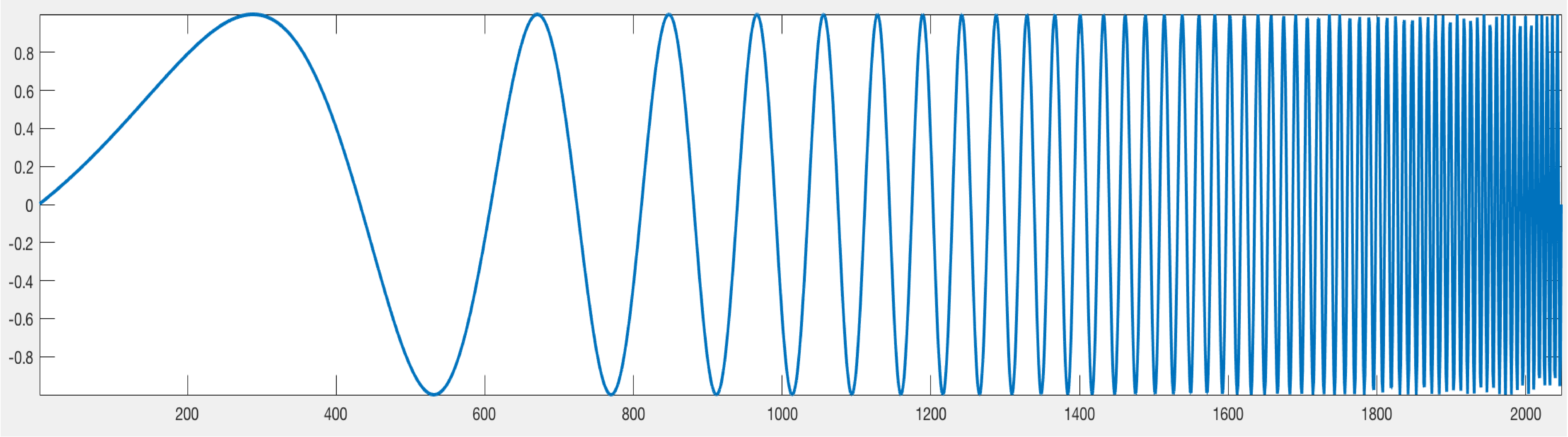




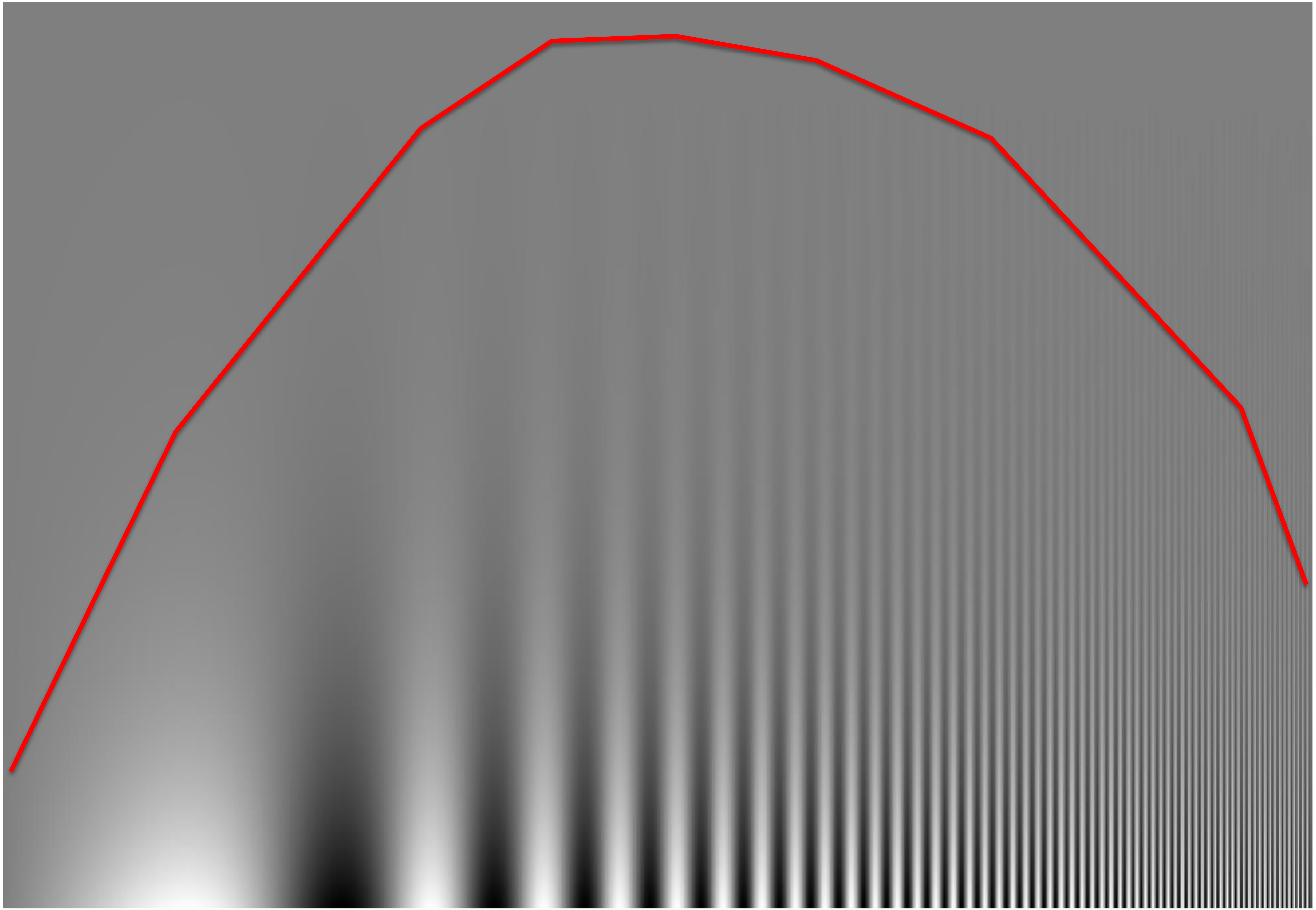
What do you think you should see when looking at this image?

$$\mathbf{I}[n,m] = A[n]\sin(2\pi f[m]\, m/M)$$





$$\mathbf{I}[n,m] = A[n]\sin(2\pi f[m]\, m/M)$$

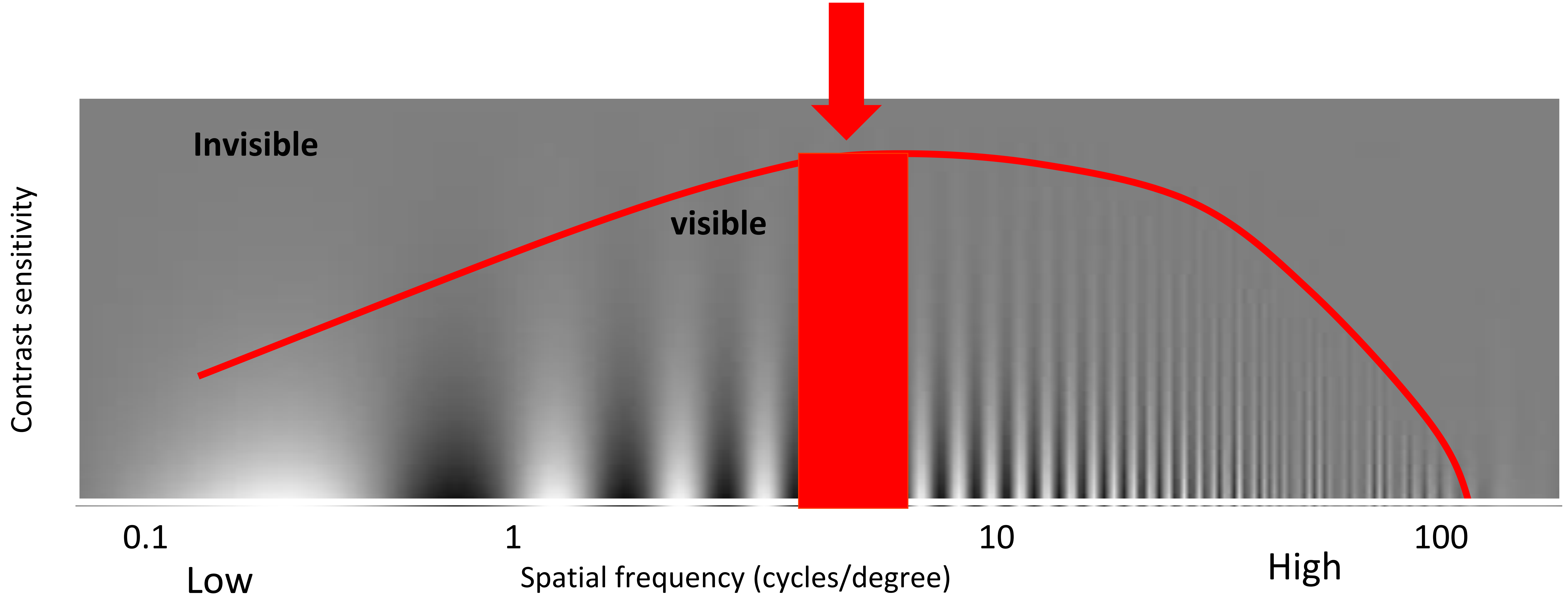


Contrast Sensitivity Function

Blackmore & Campbell (1969)

Maximum sensitivity

~ **6** cycles / degree of visual angle



Things that are very close and/or large are hard to see

Things far away are hard to see