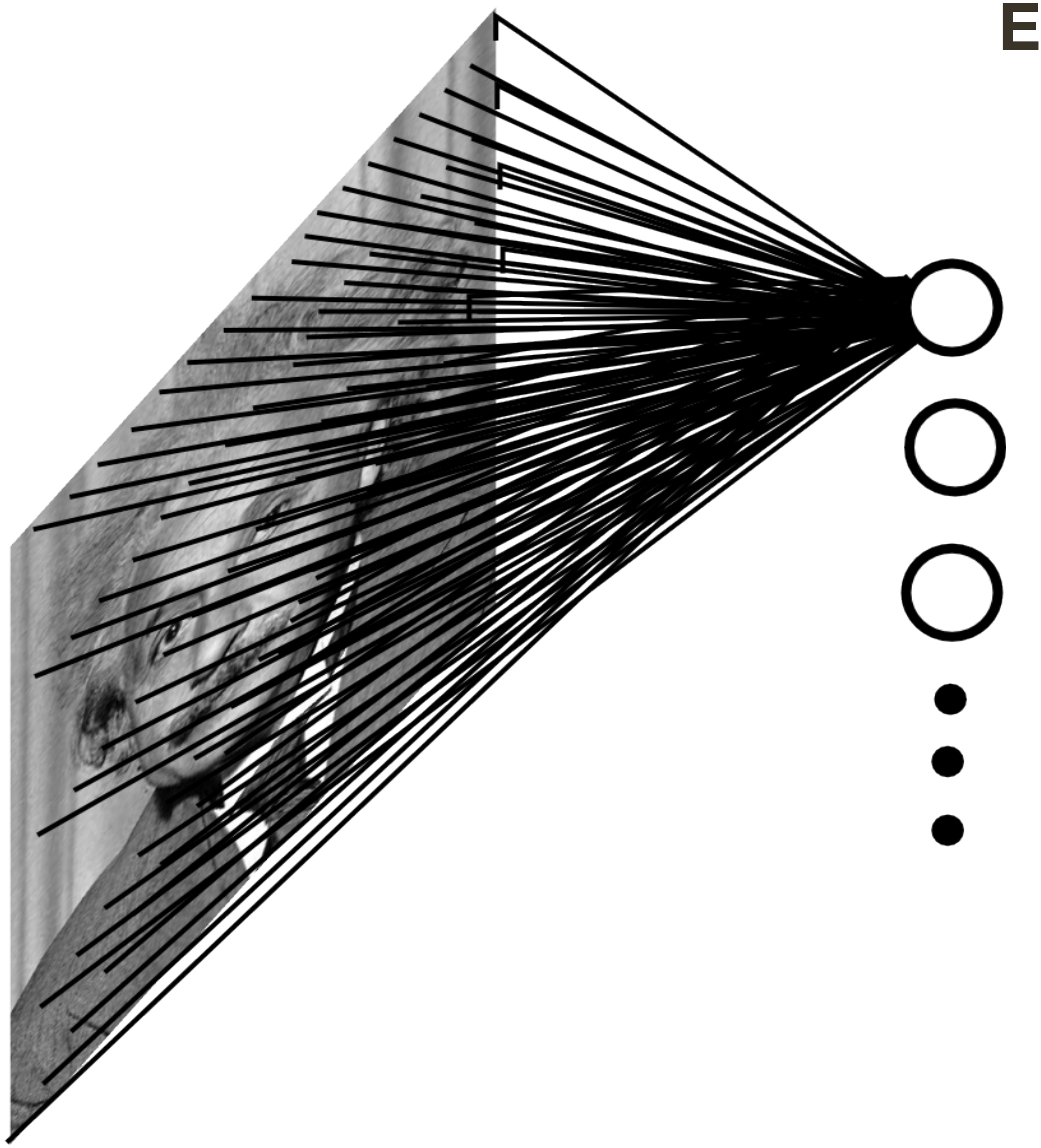
THE UNIVERSITY OF BRITISH COLUMBIA

Topics in AI (CPSC 532S):
Multimodal Learning with Vision, Language and Sound

Lecture 4: Convolutional Neural Networks (Part 1)

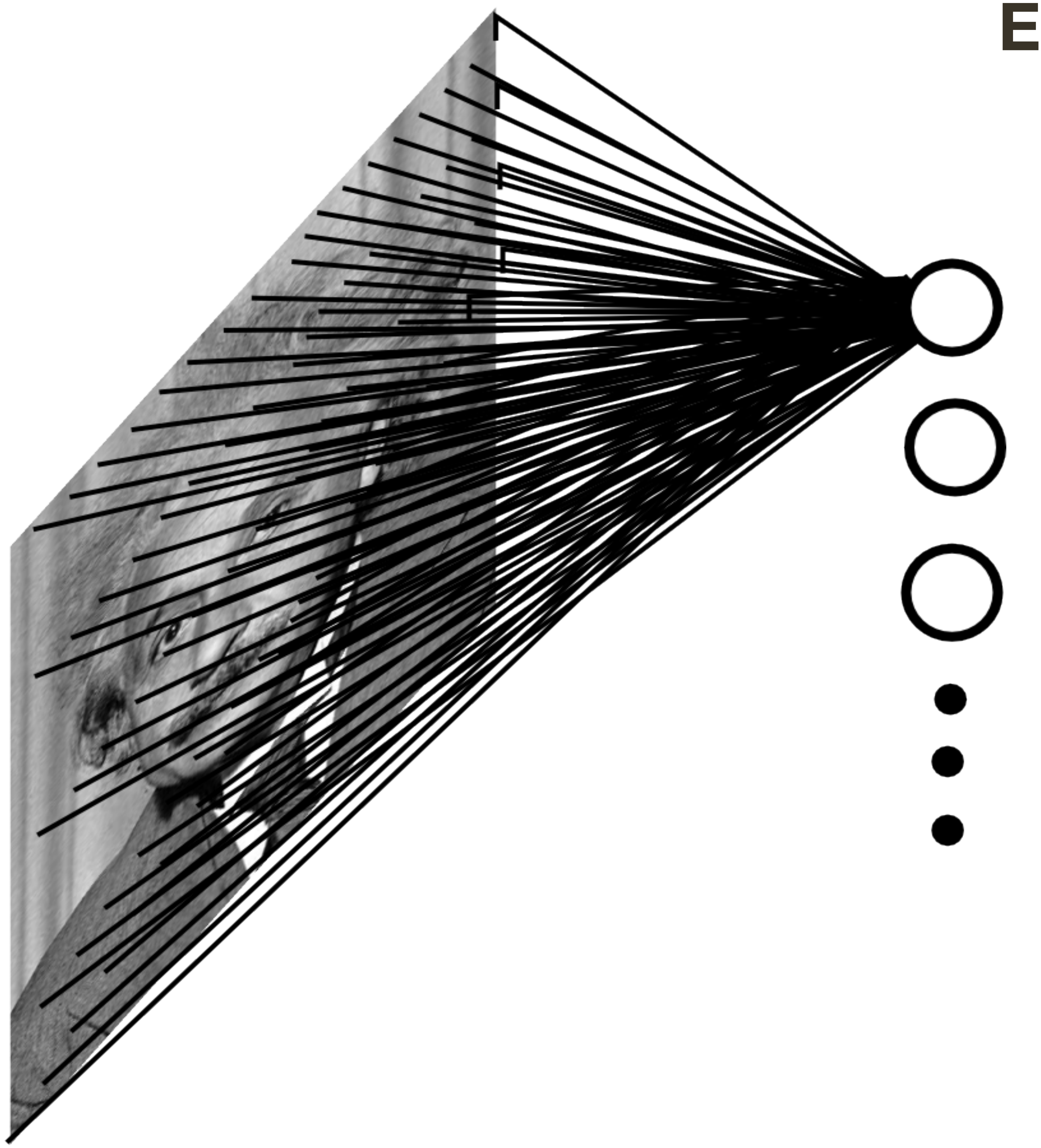
Fully Connected Layer



Example: 200 x 200 image (small)
x 40K hidden units

* slide from Marc'Aurelio Renzato

Fully Connected Layer



Example: 200 x 200 image (small)
x 40K hidden units

= ~ **2 Billion** parameters (for one layer!)

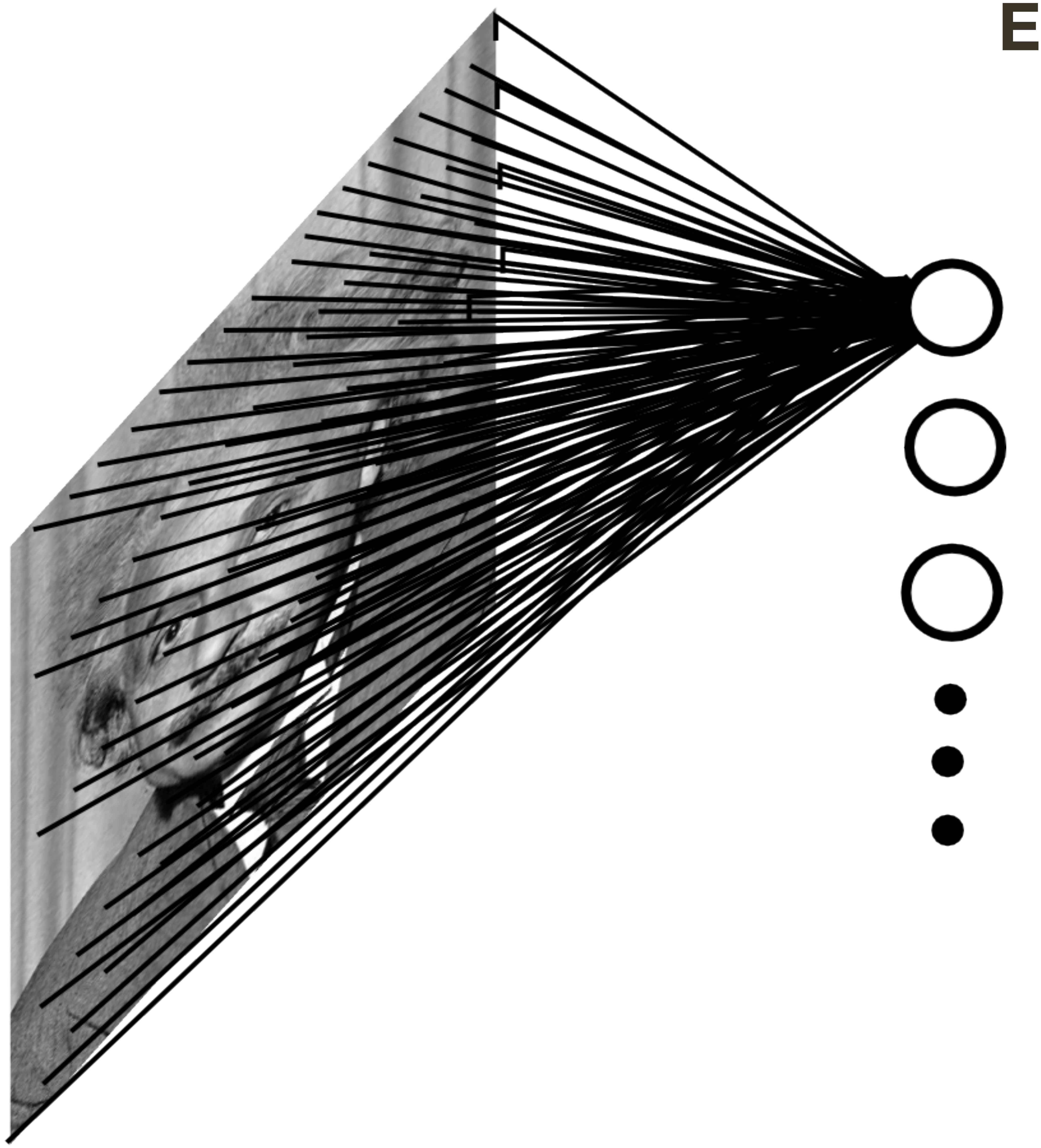
* slide from Marc'Aurelio Renzato

Linear **regression** (review)

	Inputs (features)					Outputs	
	production costs	promotional costs	genre of the movie	box office first week	total book sales	total revenue USA	total revenue international
Training Set 	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_4^{(1)}$	$x_5^{(1)}$	$y_1^{(1)}$	$y_2^{(1)}$
	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$x_4^{(2)}$	$x_5^{(2)}$	$y_1^{(2)}$	$y_2^{(2)}$
	$x_1^{(3)}$	$x_2^{(3)}$	$x_3^{(3)}$	$x_4^{(3)}$	$x_5^{(3)}$	$y_1^{(3)}$	$y_2^{(3)}$
Testing Set 	$x_1^{(4)}$	$x_2^{(4)}$	$x_3^{(4)}$	$x_4^{(4)}$	$x_5^{(4)}$	$\hat{y}_j = \sum_i w_{ji} x_i + b_j$	
	$x_1^{(5)}$	$x_2^{(5)}$	$x_3^{(5)}$	$x_4^{(5)}$	$x_5^{(5)}$		

*slide adopted from V. Ordonex

Fully Connected Layer



Example: 200 x 200 image (small)
x 40K hidden units

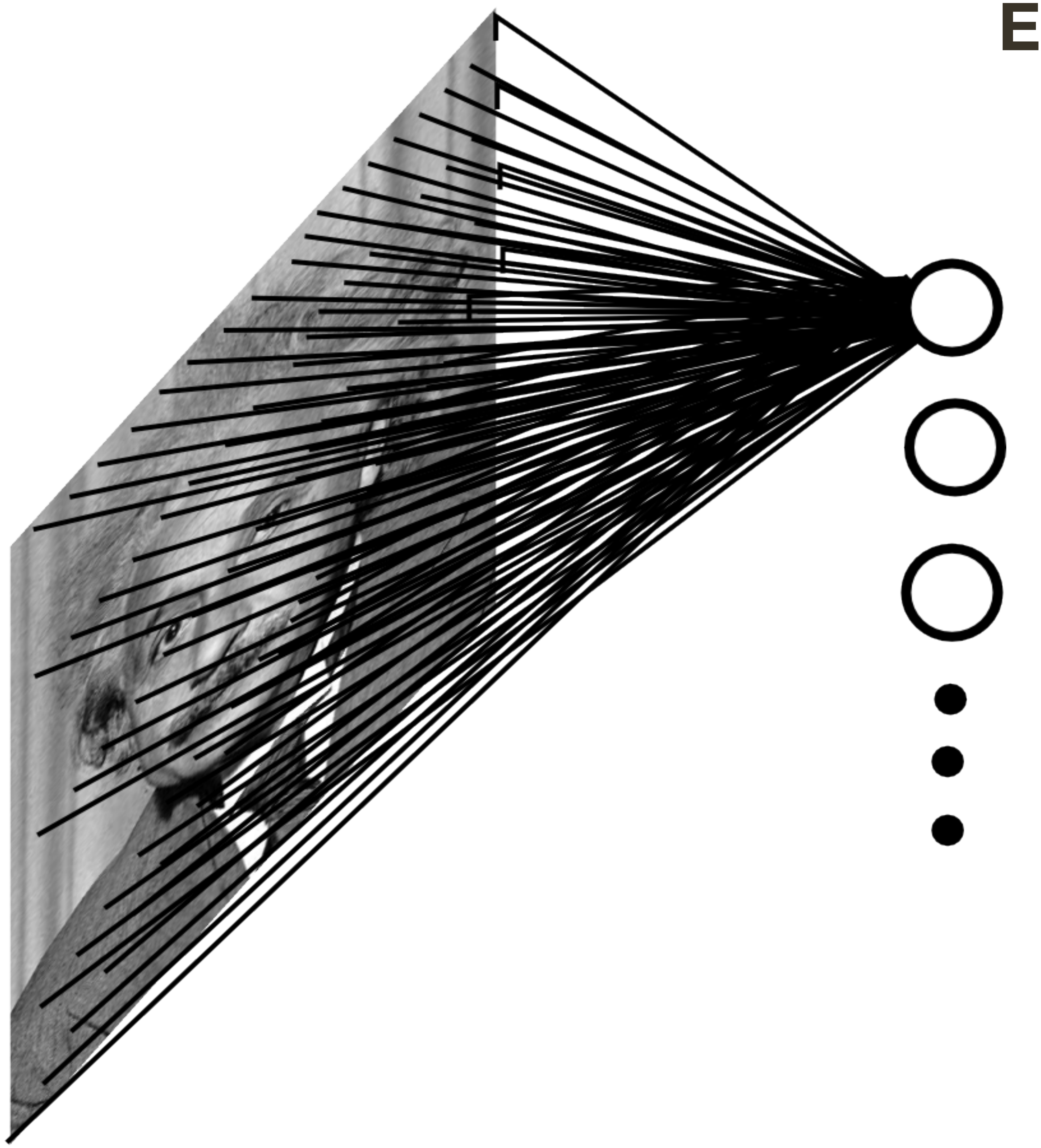
= ~ **2 Billion** parameters (for one layer!)

$$\hat{y}_j = \mathbf{w}_{j,:}^T \mathbf{x} + b_j$$

$$\hat{y}_j = \sum_i w_{ji} x_i + b_j$$

* slide from Marc'Aurelio Renzato

Fully Connected Layer



Example: 200 x 200 image (small)
x 40K hidden units

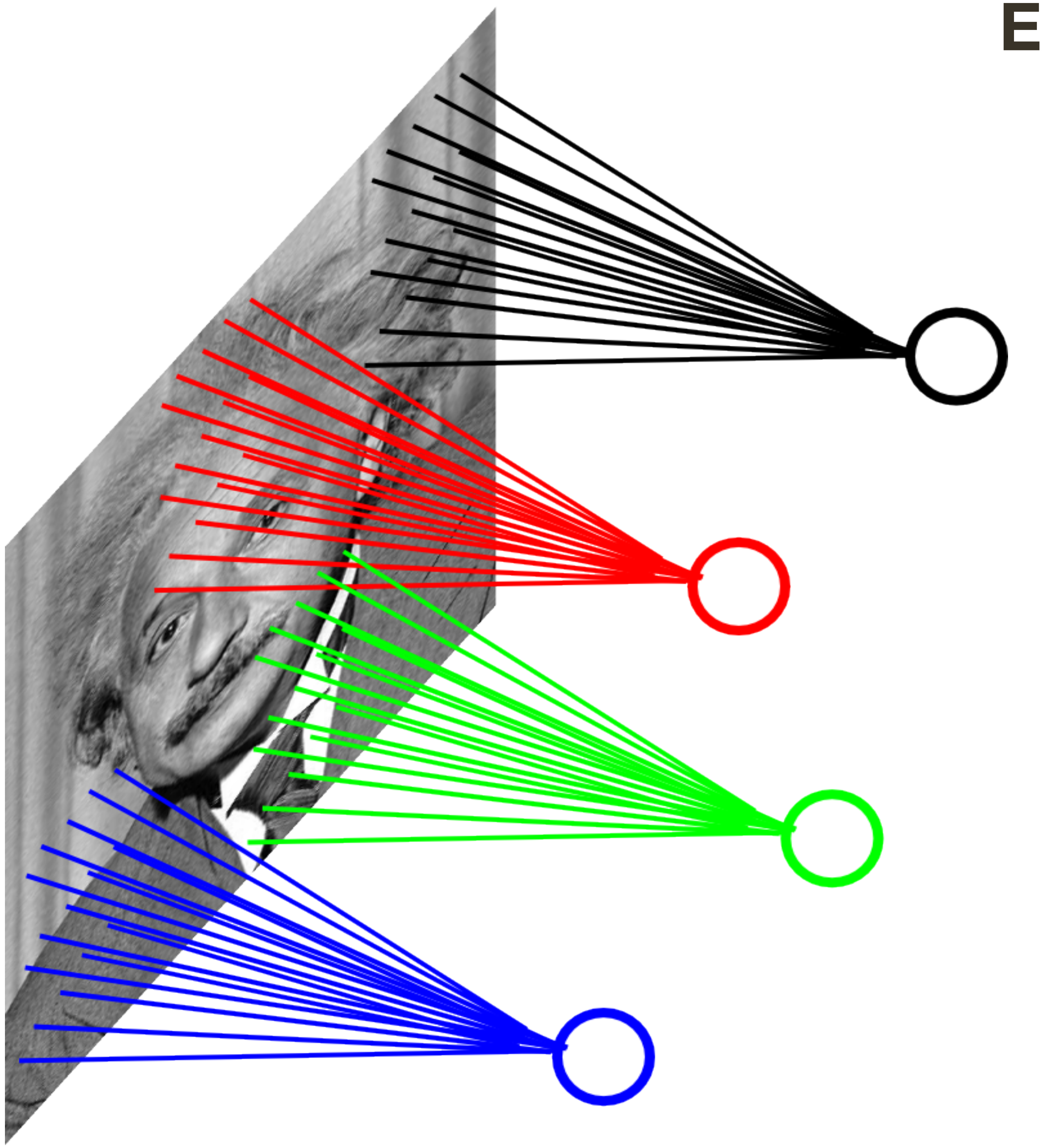
= ~ **2 Billion** parameters (for one layer!)

Spatial correlations are generally local

Waste of resources + we don’t have enough data to train networks this large

* slide from Marc’Aurelio Renzato

Locally Connected Layer



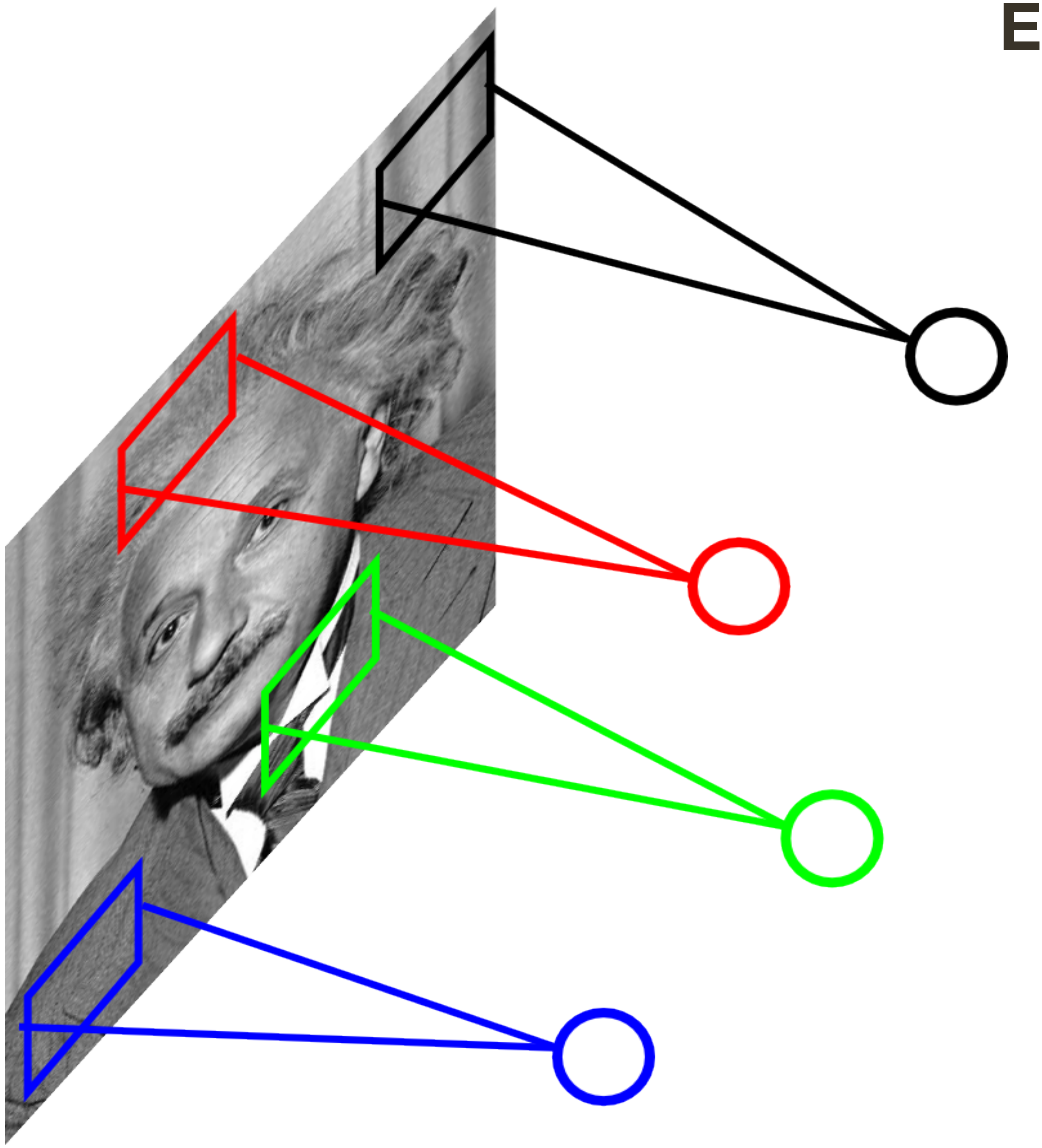
Example: 200 x 200 image (small)

Filter size: 10 x 10

= ~ **4 Million** parameters

* slide from Marc'Aurelio Renzato

Locally Connected Layer



Example: 200 x 200 image (small)

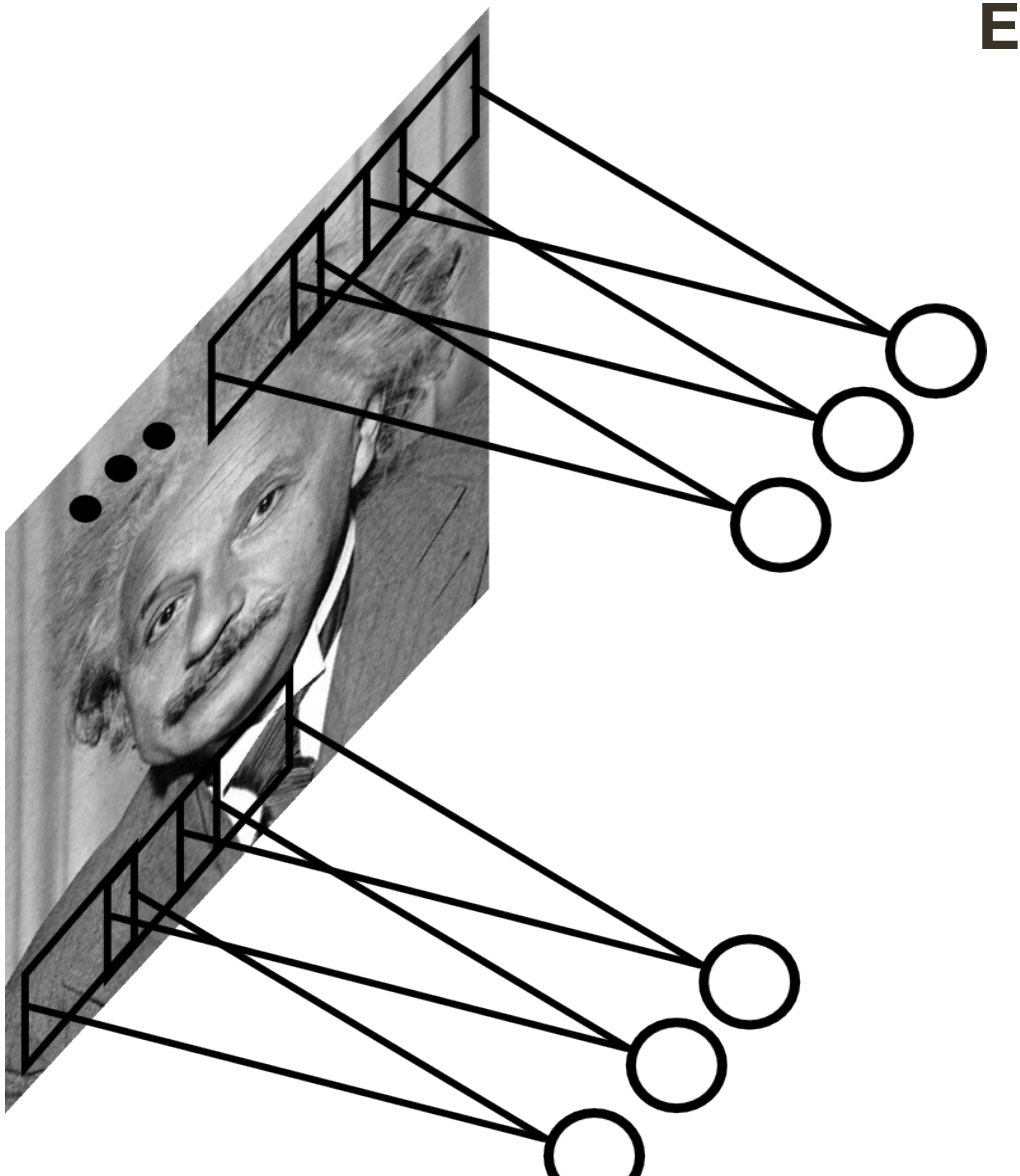
Filter size: 10 x 10

= ~ **4 Million** parameters

Stationarity — statistics is similar at different locations

* slide from Marc'Aurelio Renzato

Convolutional Layer



Example: 200 x 200 image (small)

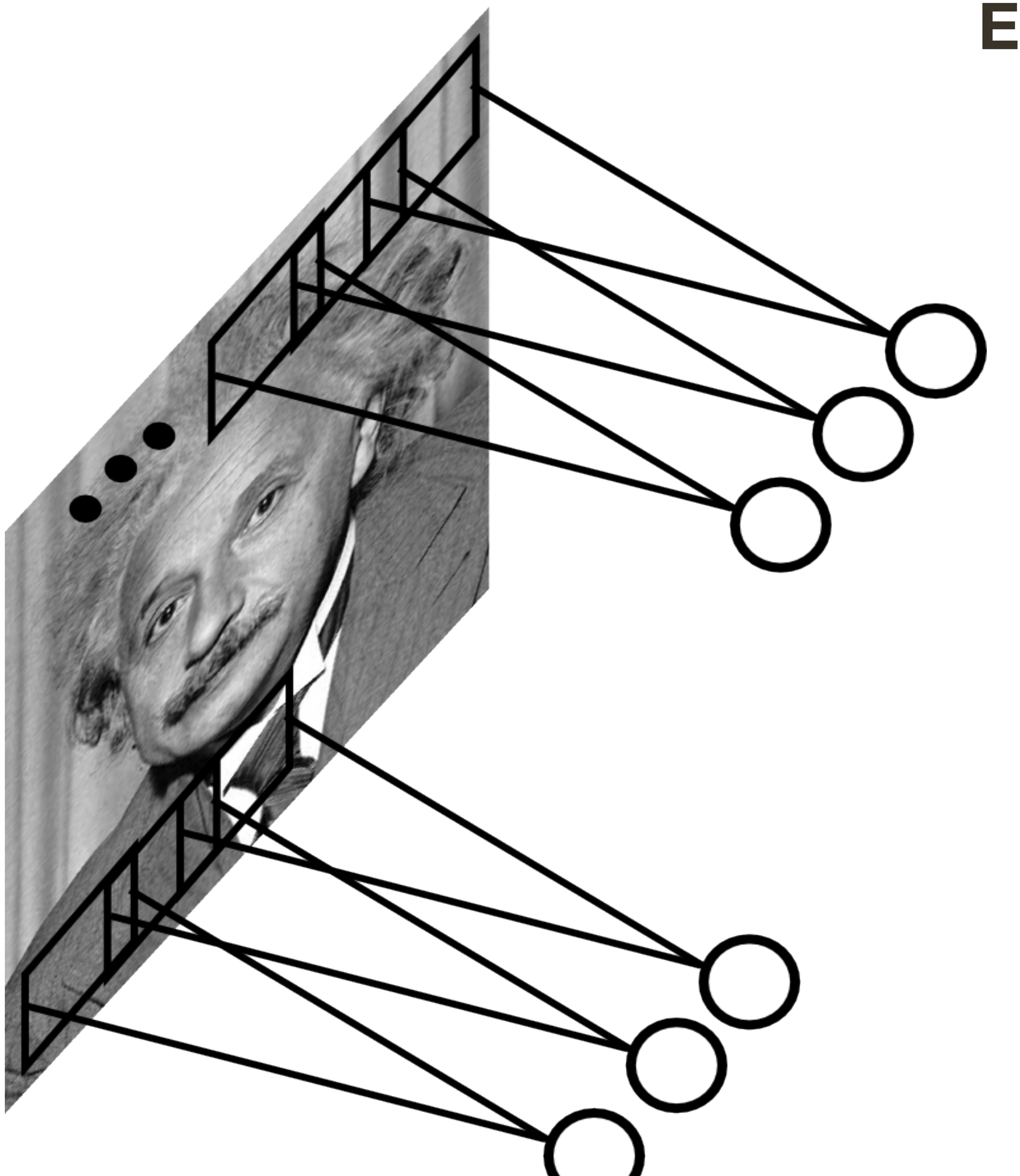
Filter size: 10 x 10

= ~ **4 Million** parameters

Share the same parameters across the locations (assuming input is stationary)

* slide adopted from Marc'Aurelio Renzato

Convolutional Layer



Example: 200 x 200 image (small)

Filter size: 10 x 10

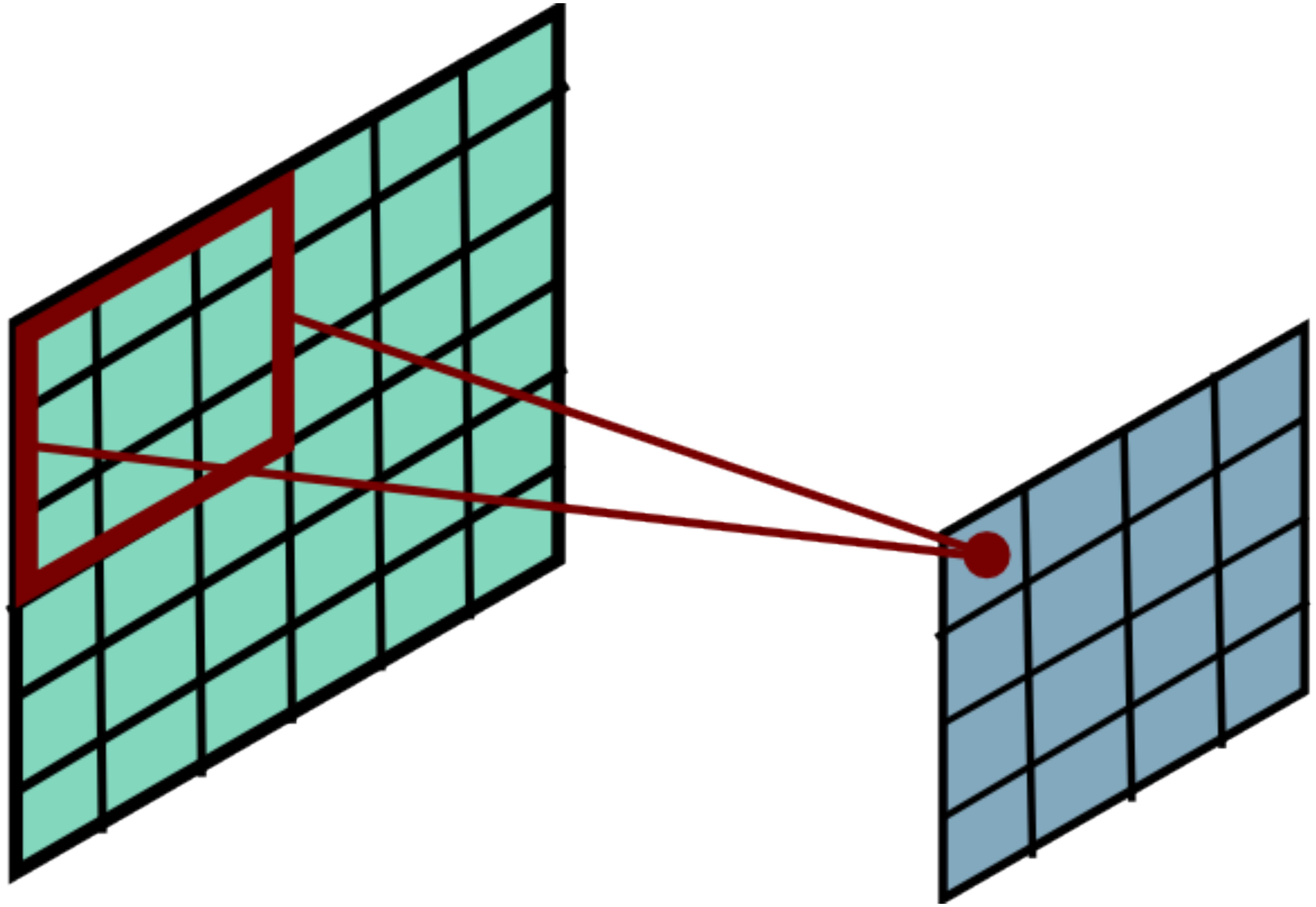
= ~ ~~**4 Million**~~ parameters

= 100+1 parameters

Share the same parameters across the locations (assuming input is stationary)

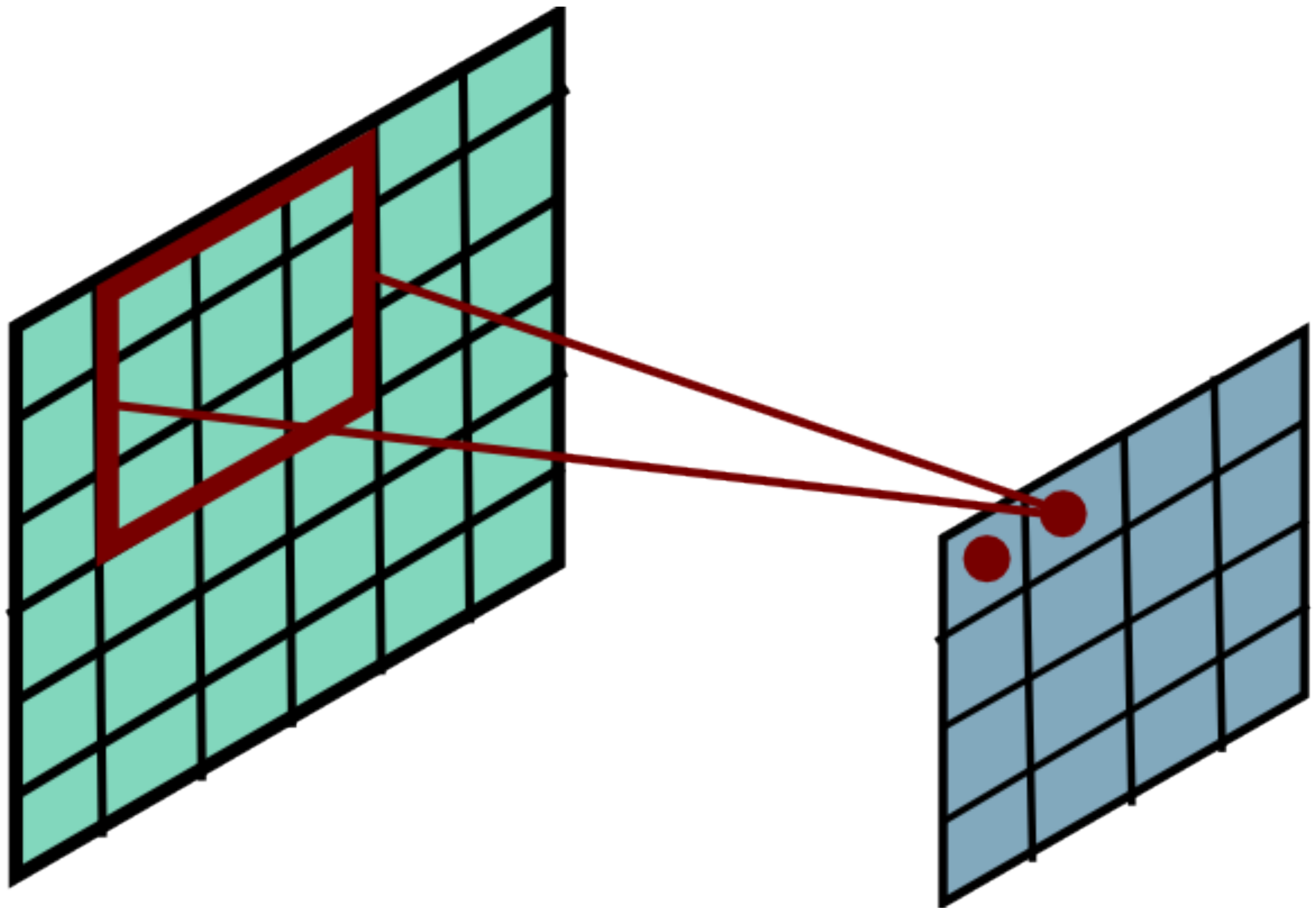
* slide adopted from Marc’Aurelio Renzato

Convolutional Layer



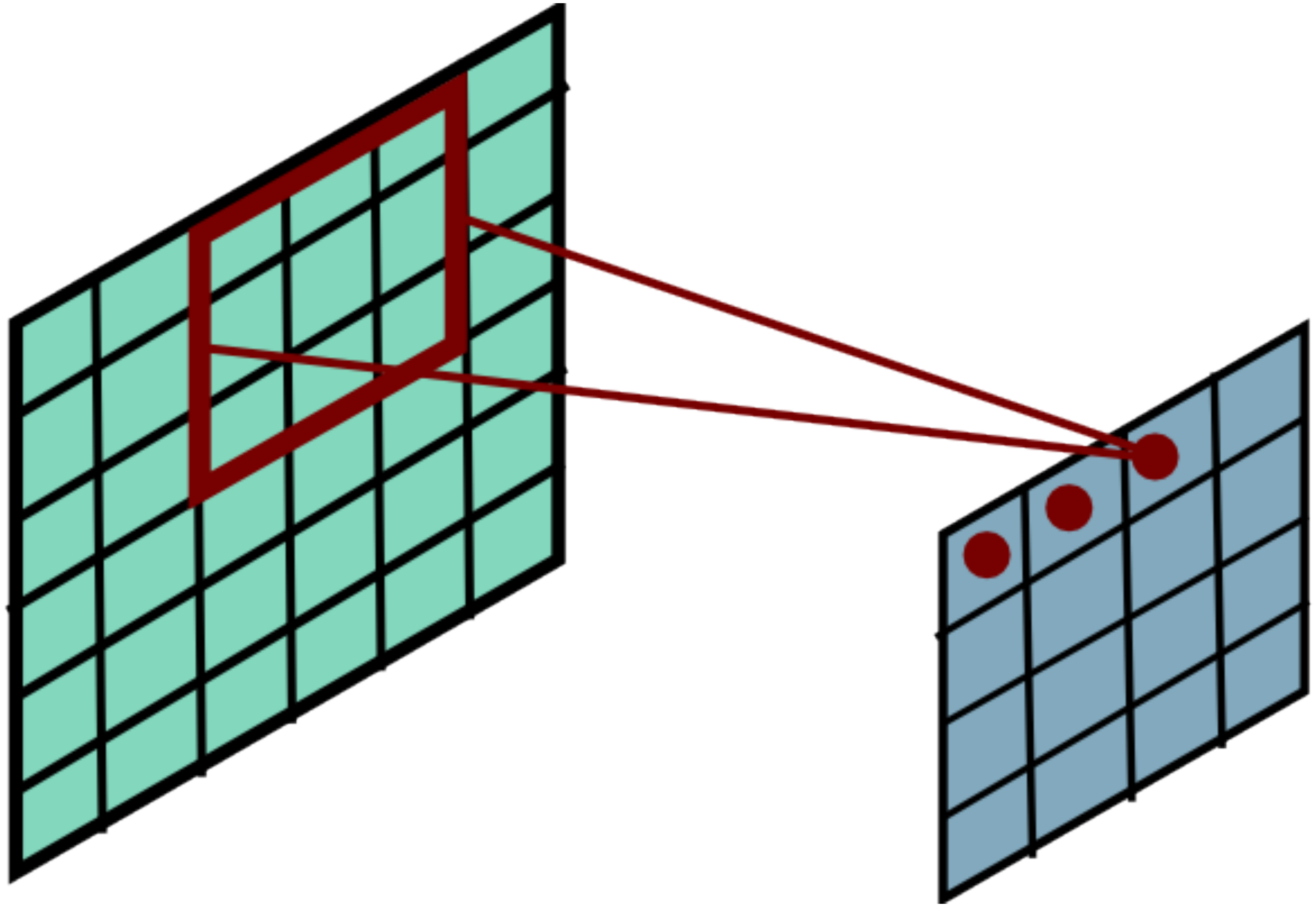
* slide from Marc'Aurelio Renzato

Convolutional Layer



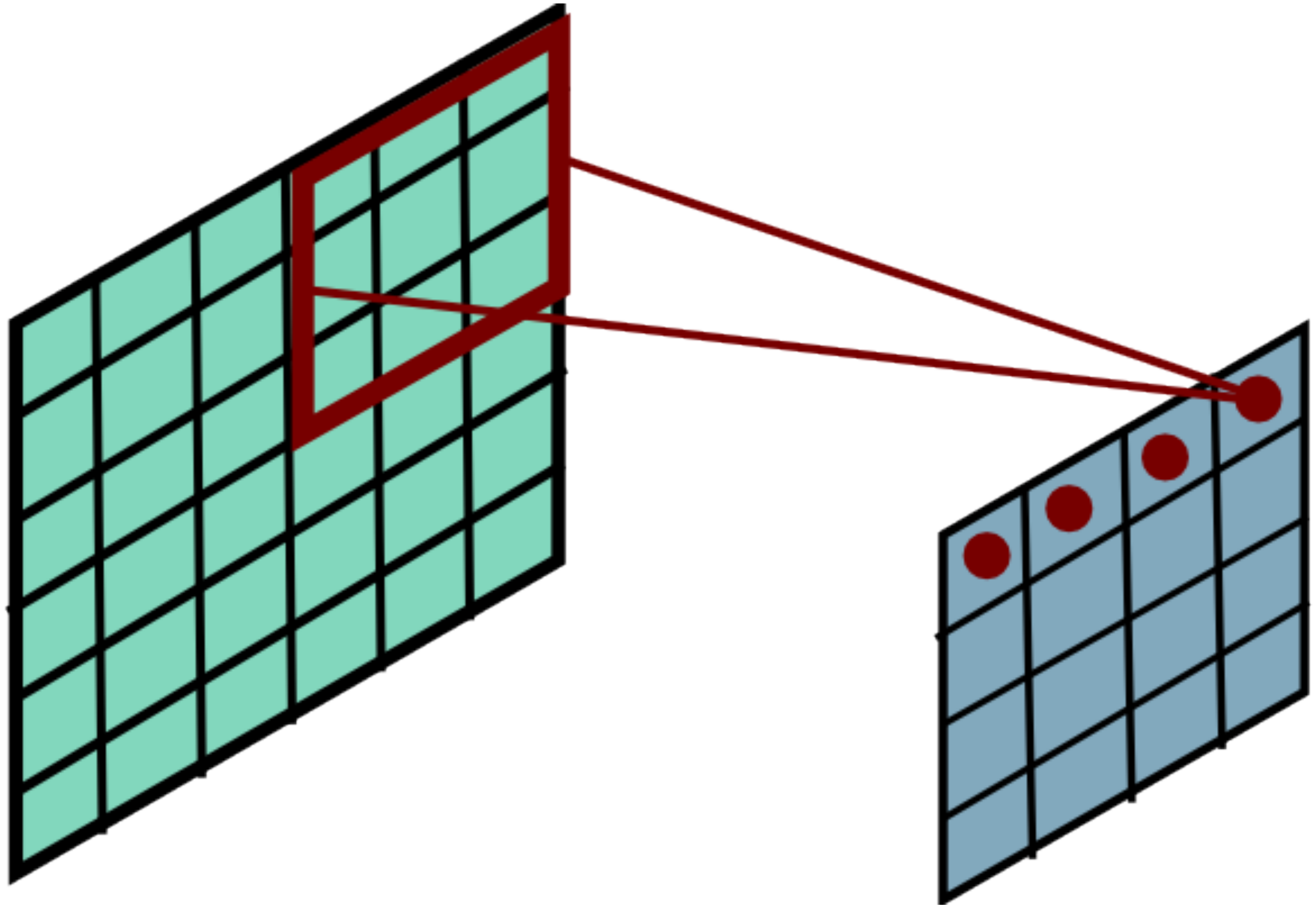
* slide from Marc'Aurelio Renzato

Convolutional Layer



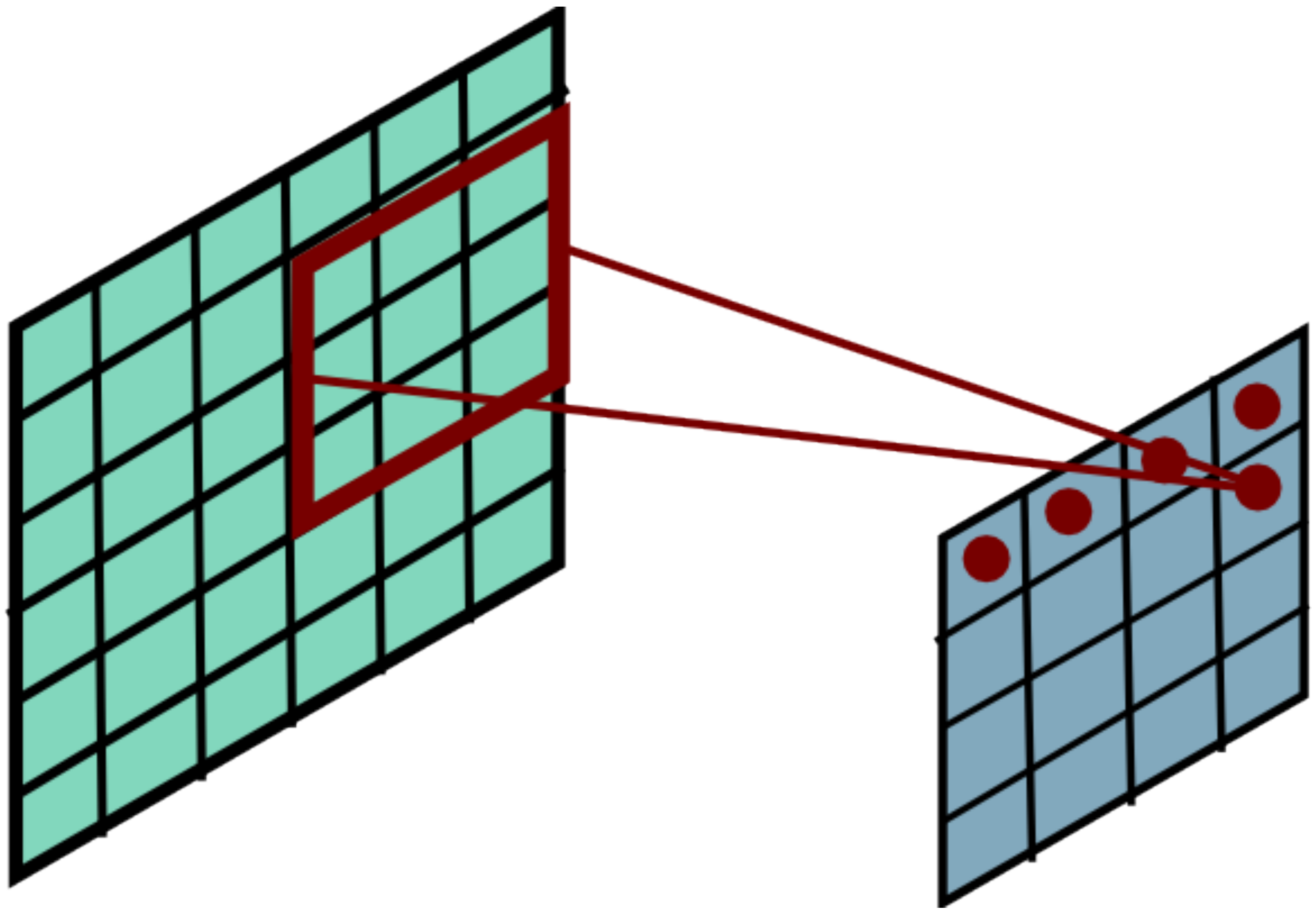
* slide from Marc'Aurelio Renzato

Convolutional Layer



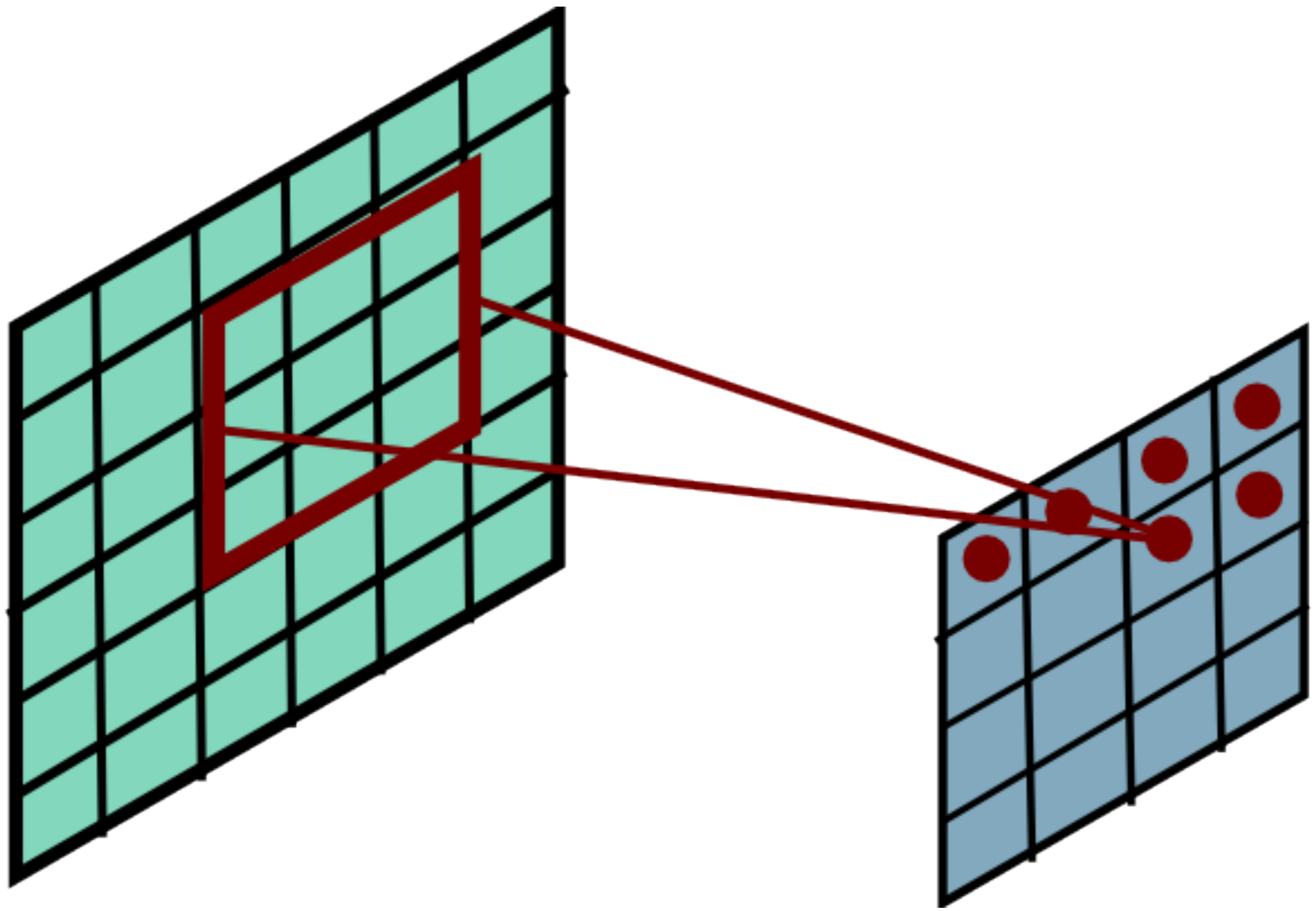
* slide from Marc'Aurelio Renzato

Convolutional Layer



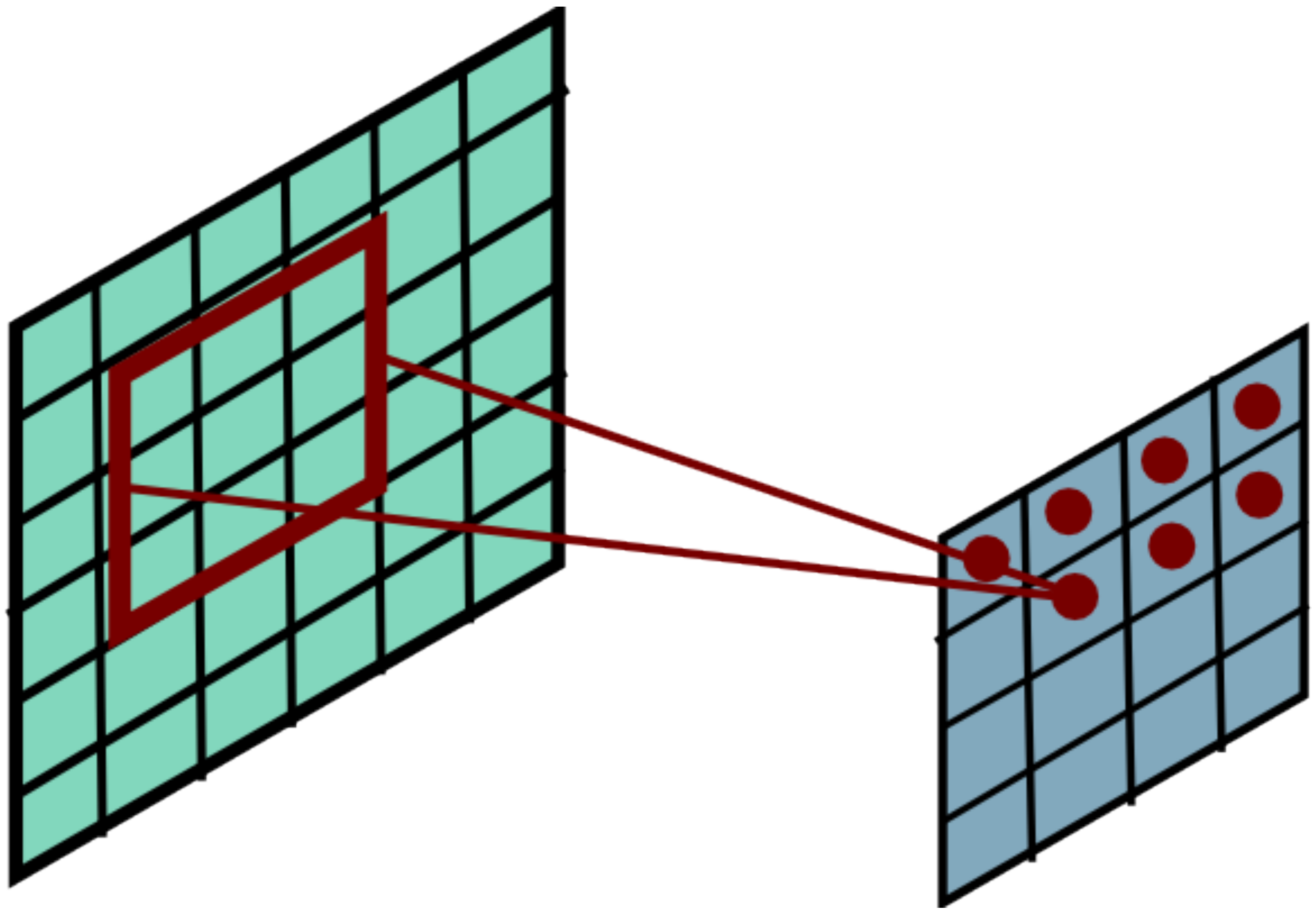
* slide from Marc'Aurelio Renzato

Convolutional Layer



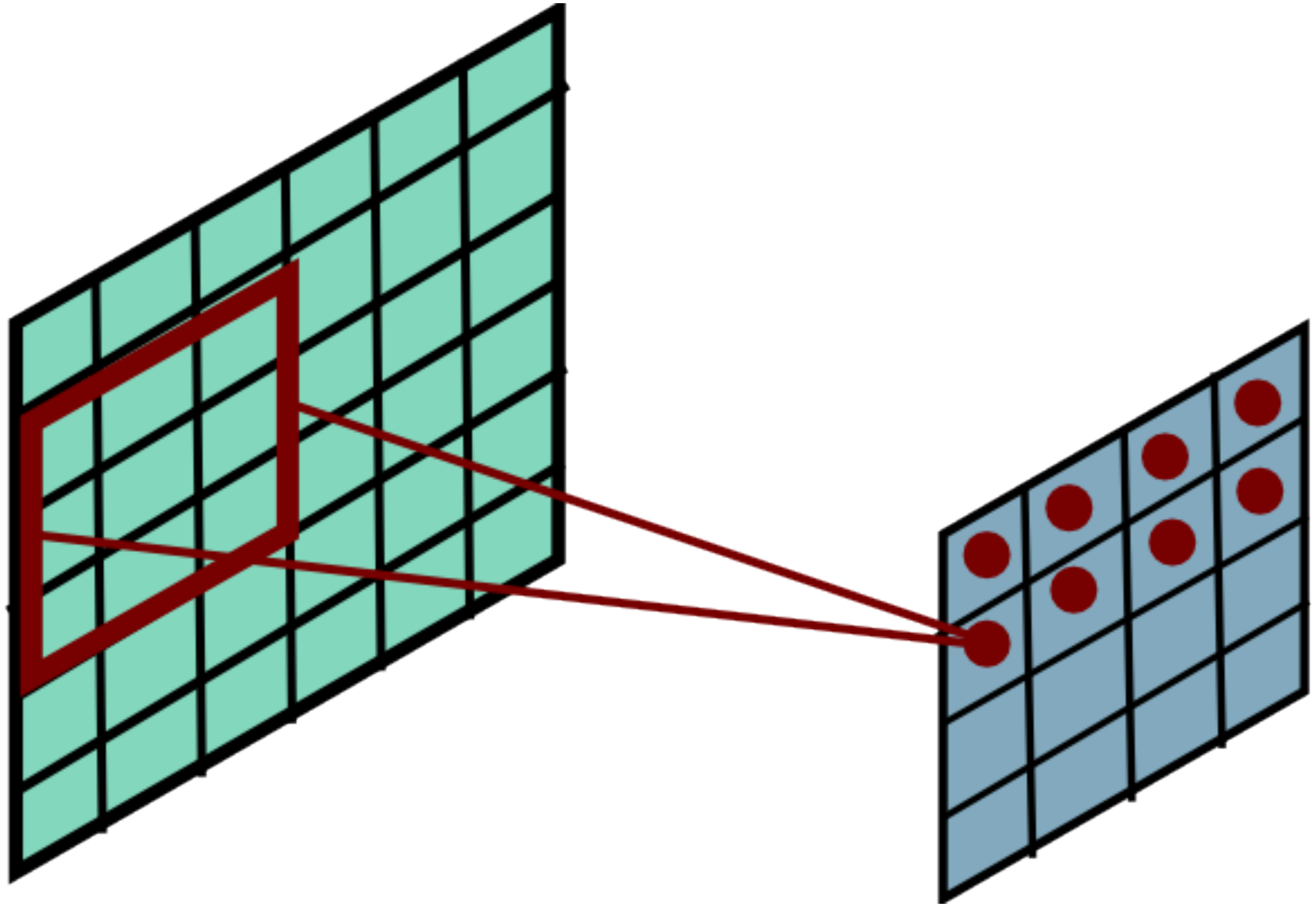
* slide from Marc’Aurelio Renzato

Convolutional Layer



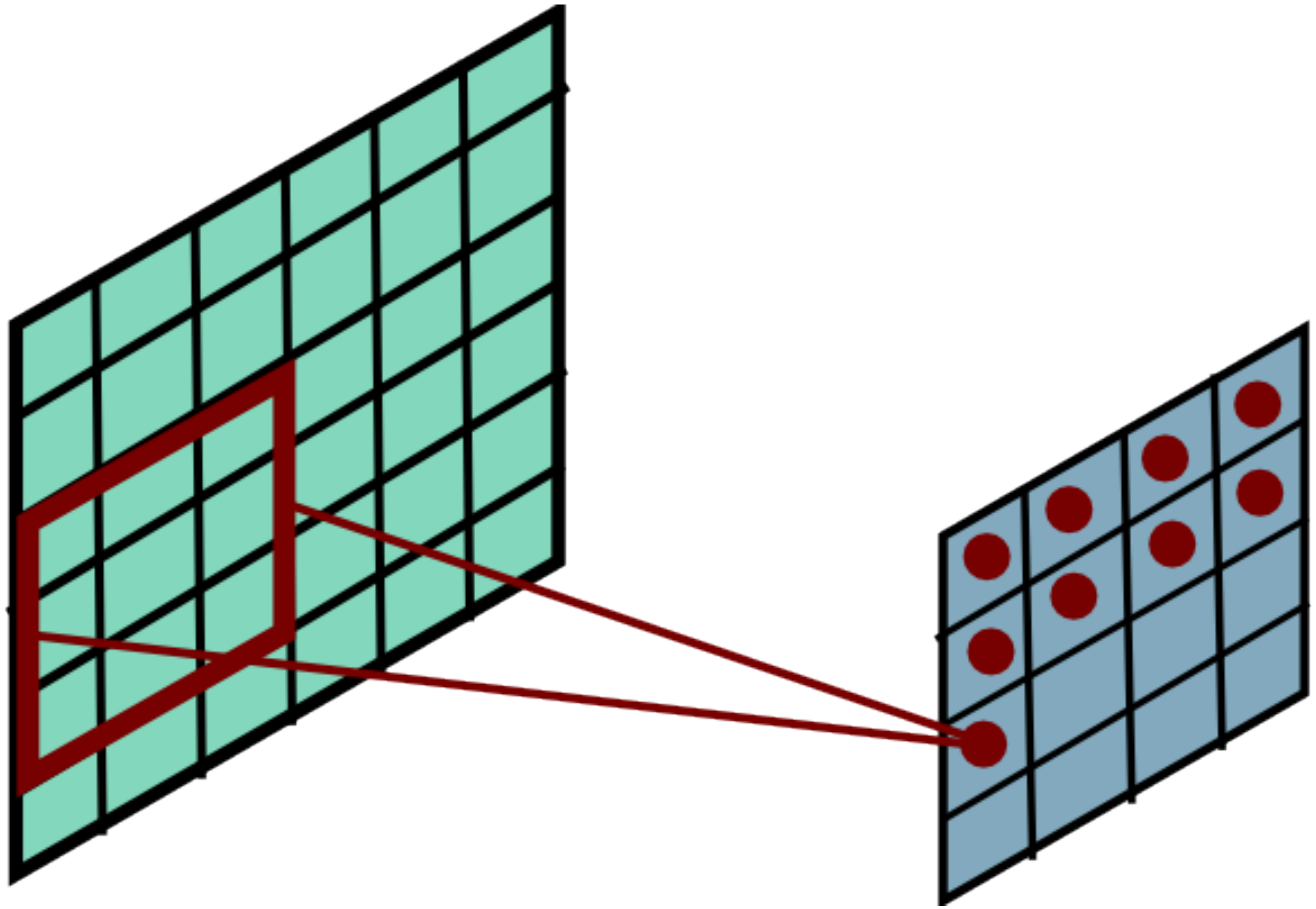
* slide from Marc’Aurelio Renzato

Convolutional Layer



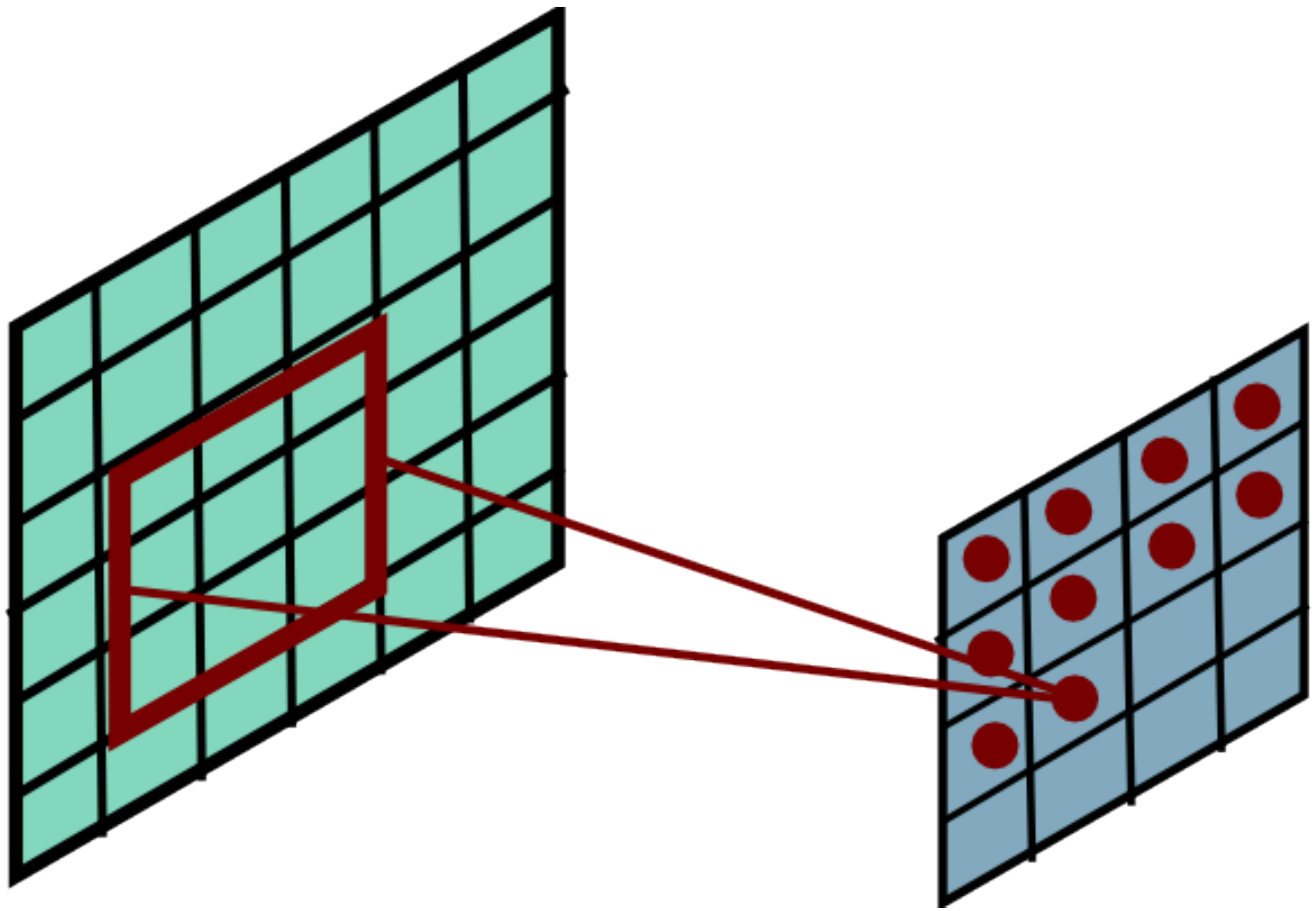
* slide from Marc’Aurelio Renzato

Convolutional Layer



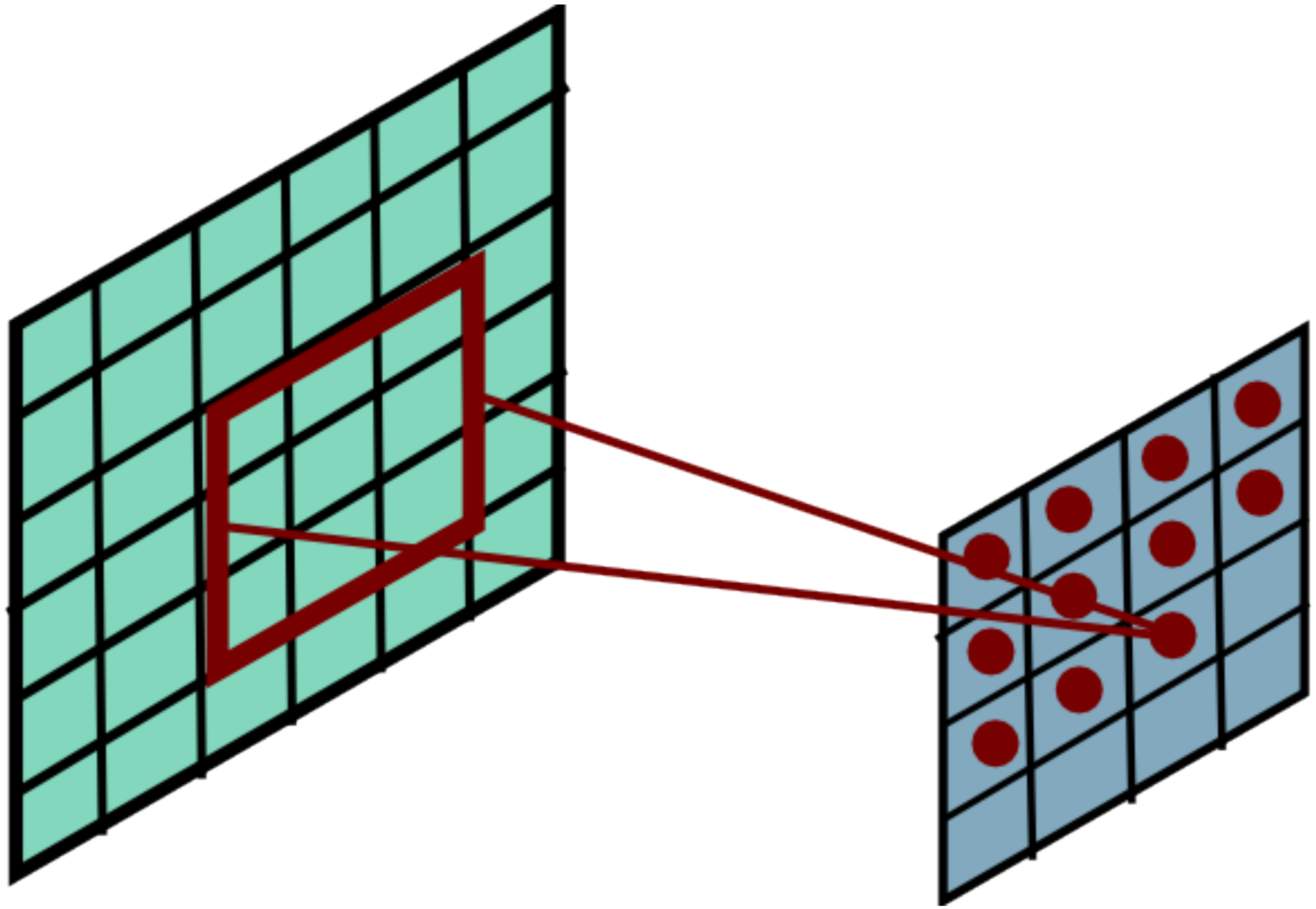
* slide from Marc'Aurelio Renzato

Convolutional Layer



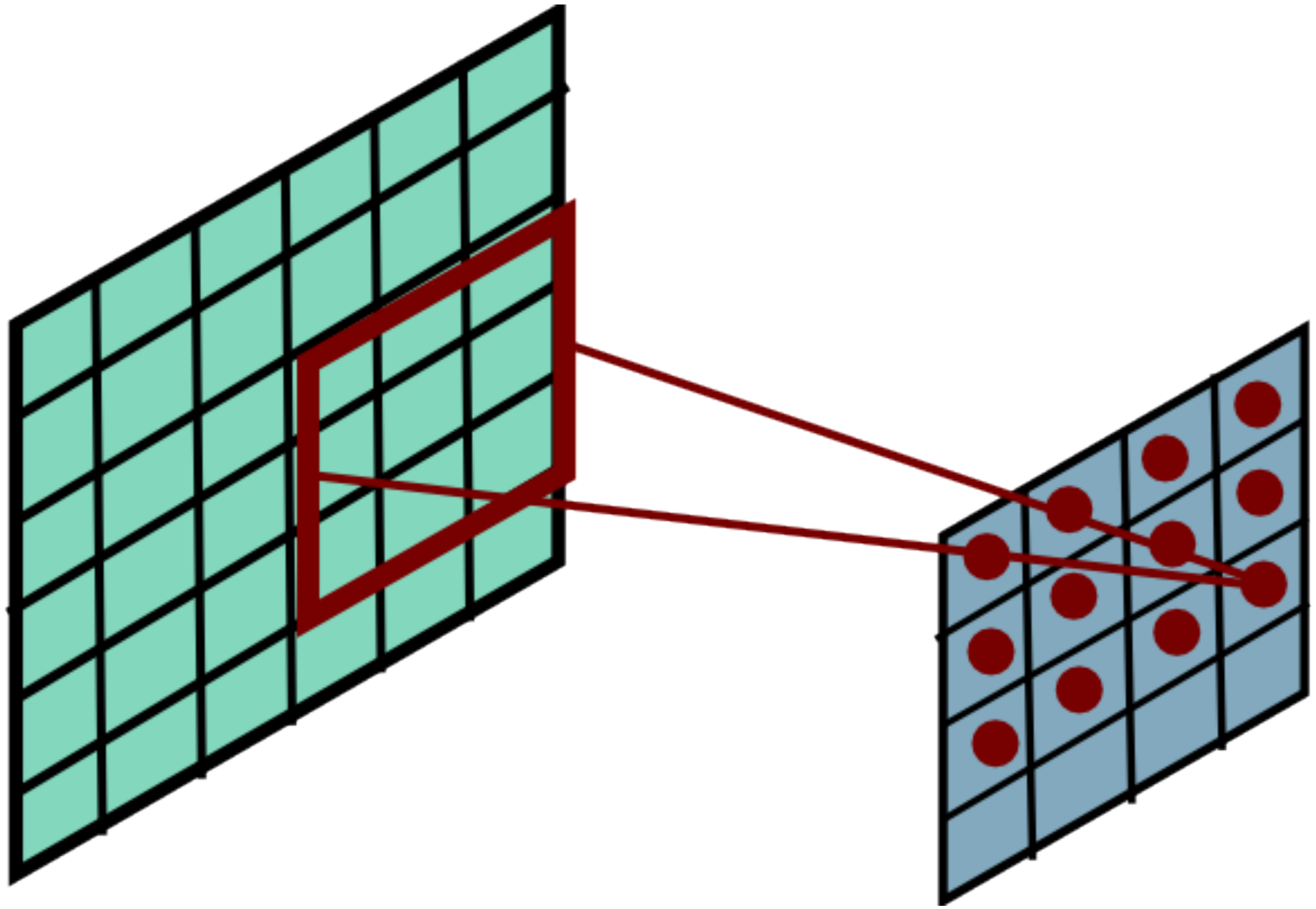
* slide from Marc’Aurelio Renzato

Convolutional Layer



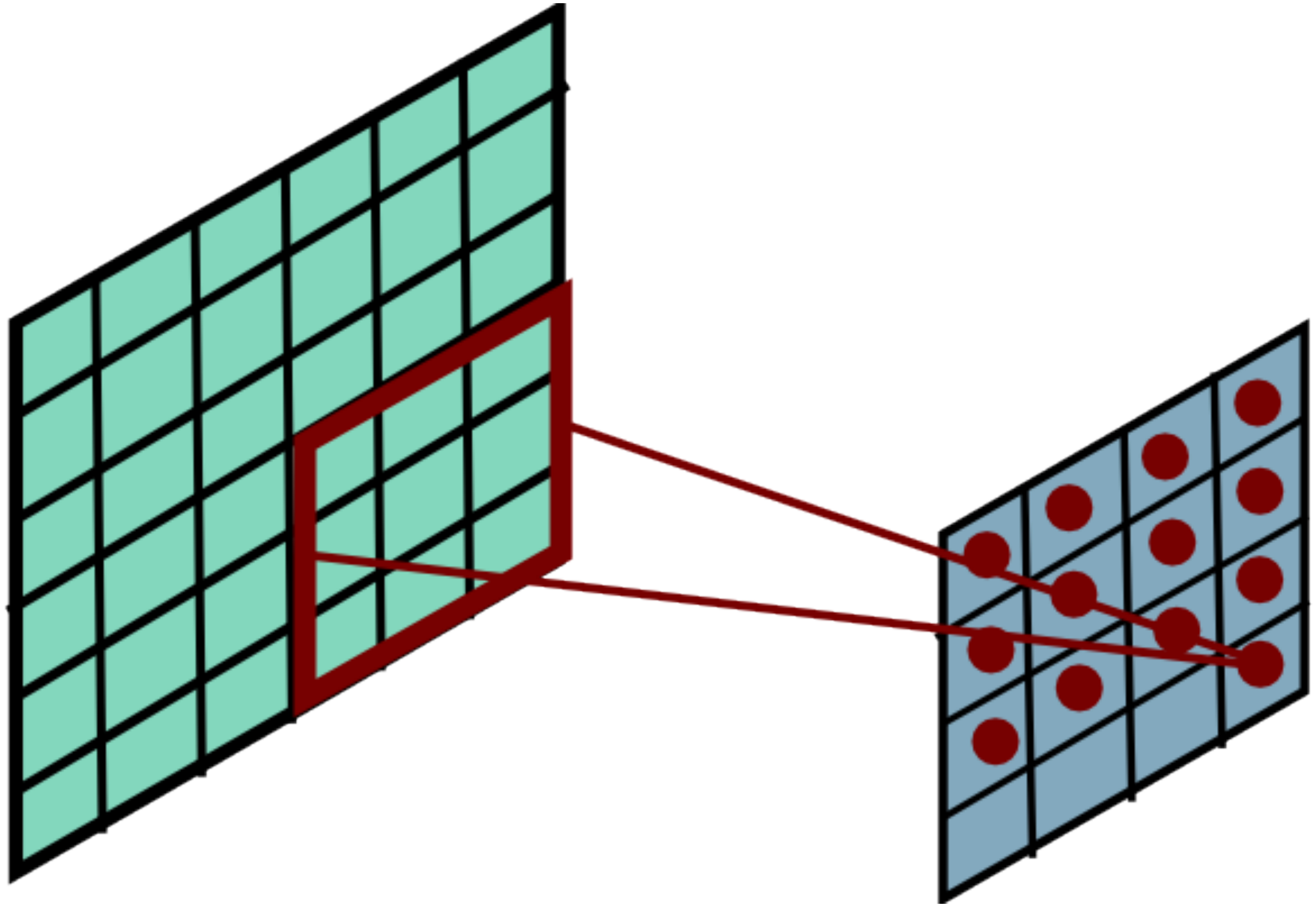
* slide from Marc’Aurelio Renzato

Convolutional Layer



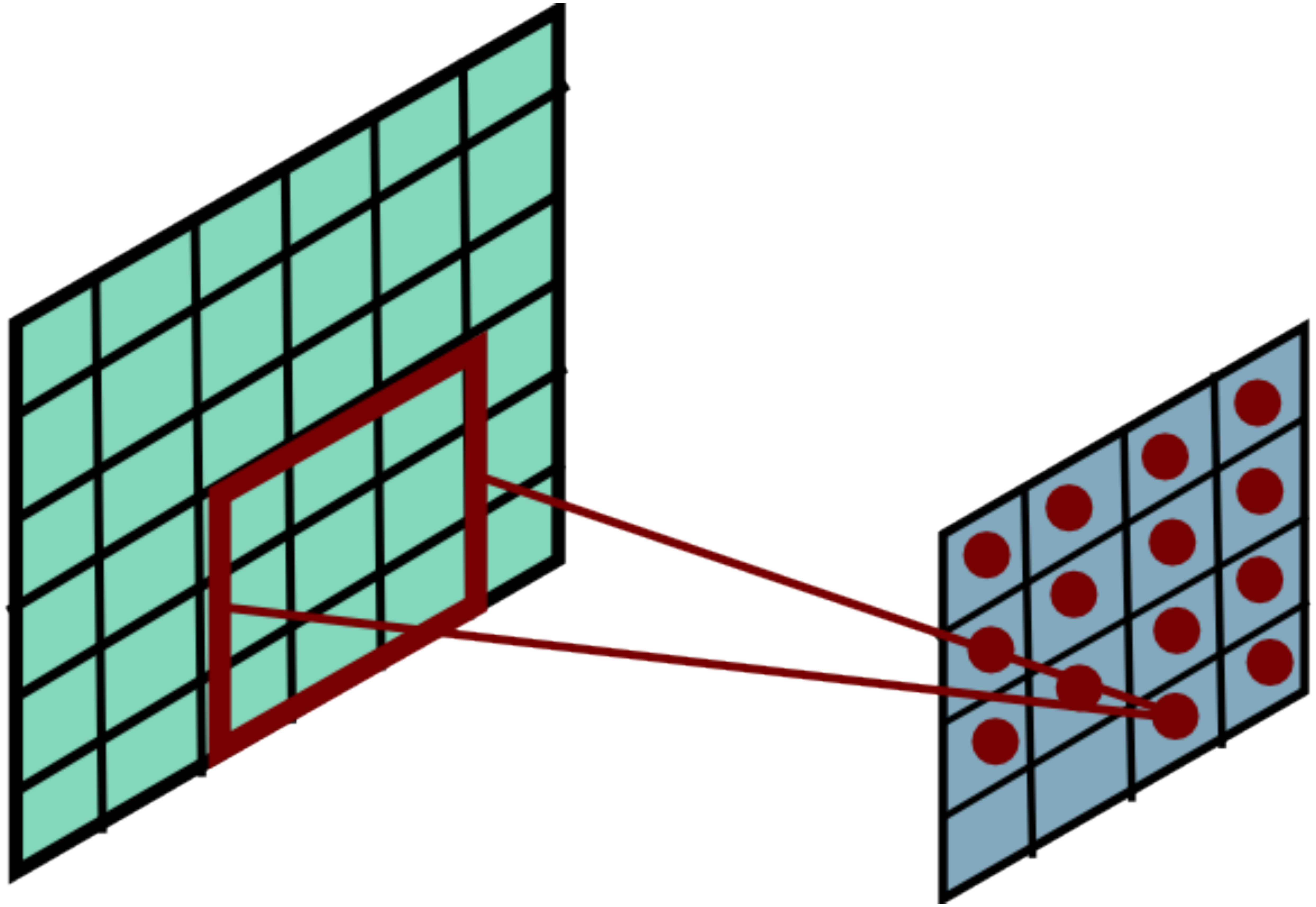
* slide from Marc'Aurelio Renzato

Convolutional Layer



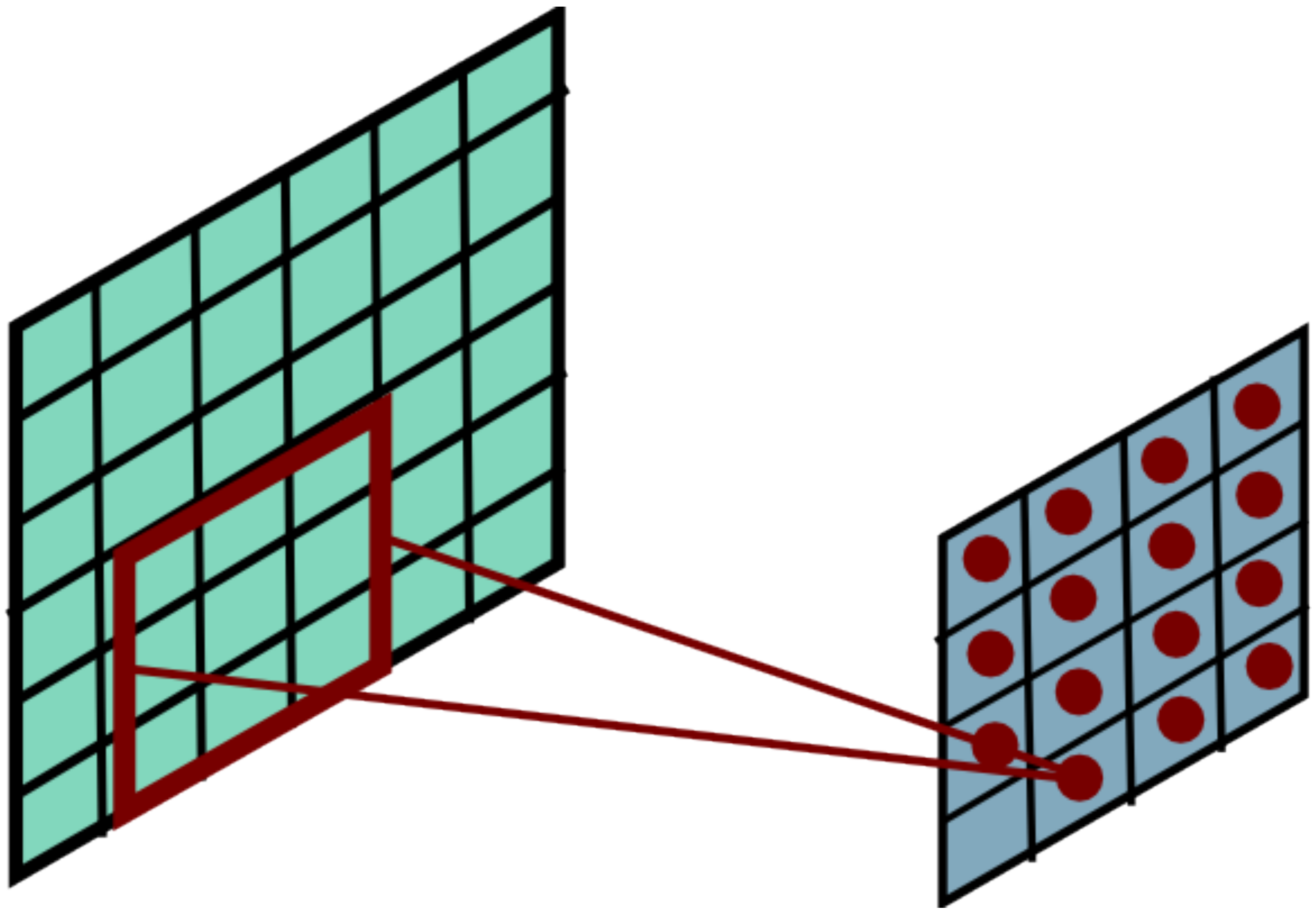
* slide from Marc’Aurelio Renzato

Convolutional Layer



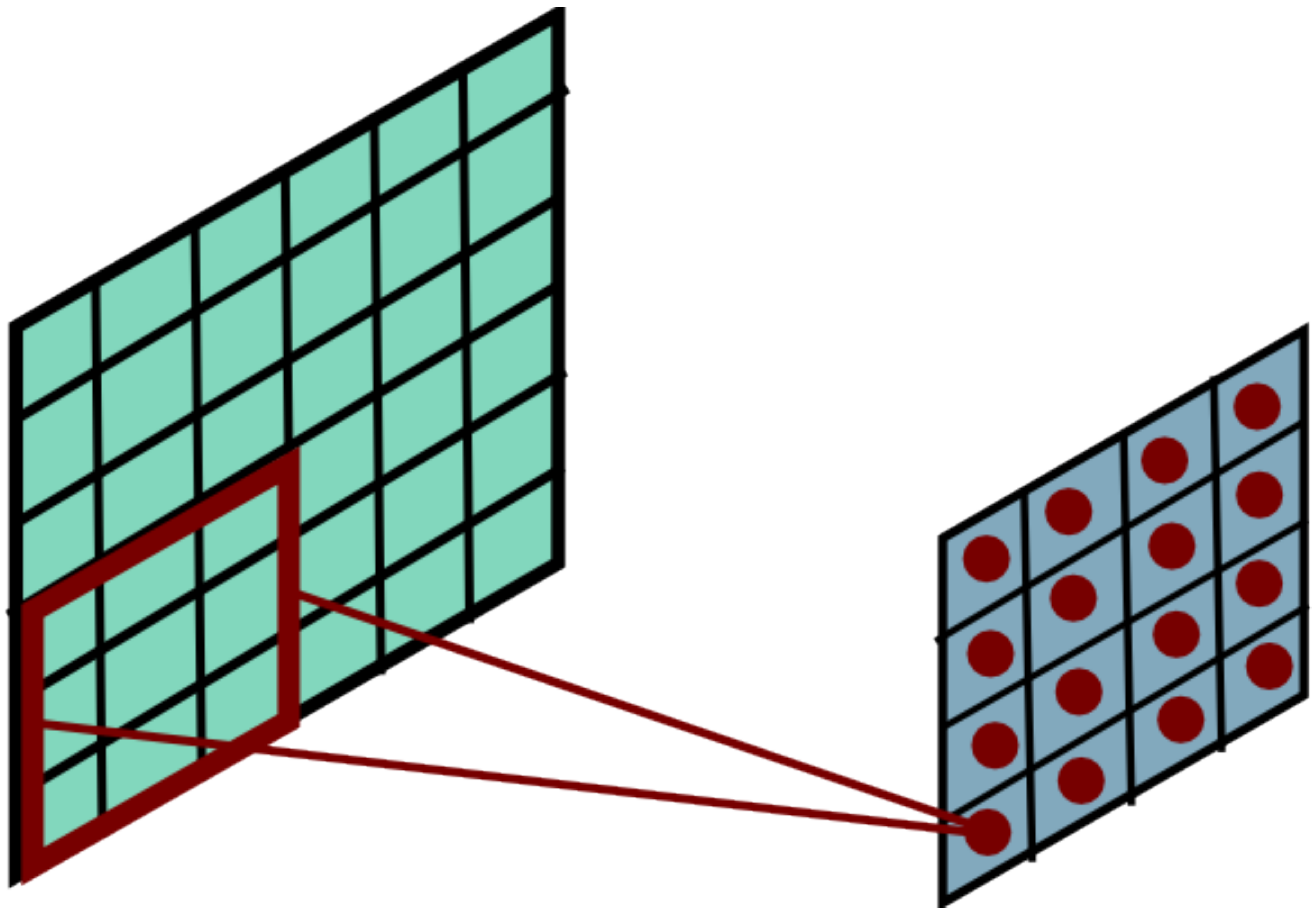
* slide from Marc’Aurelio Renzato

Convolutional Layer



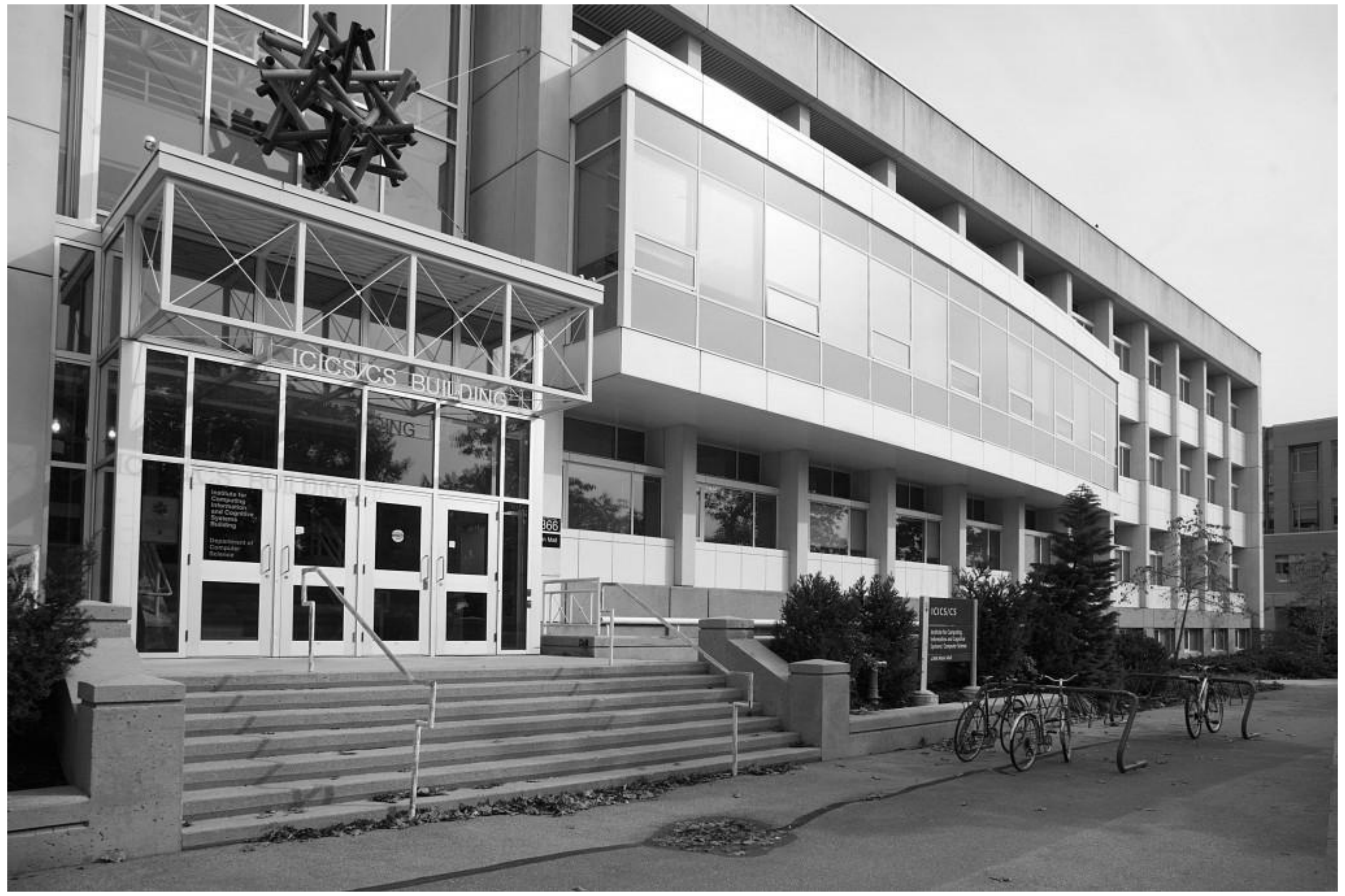
* slide from Marc'Aurelio Renzato

Convolutional Layer

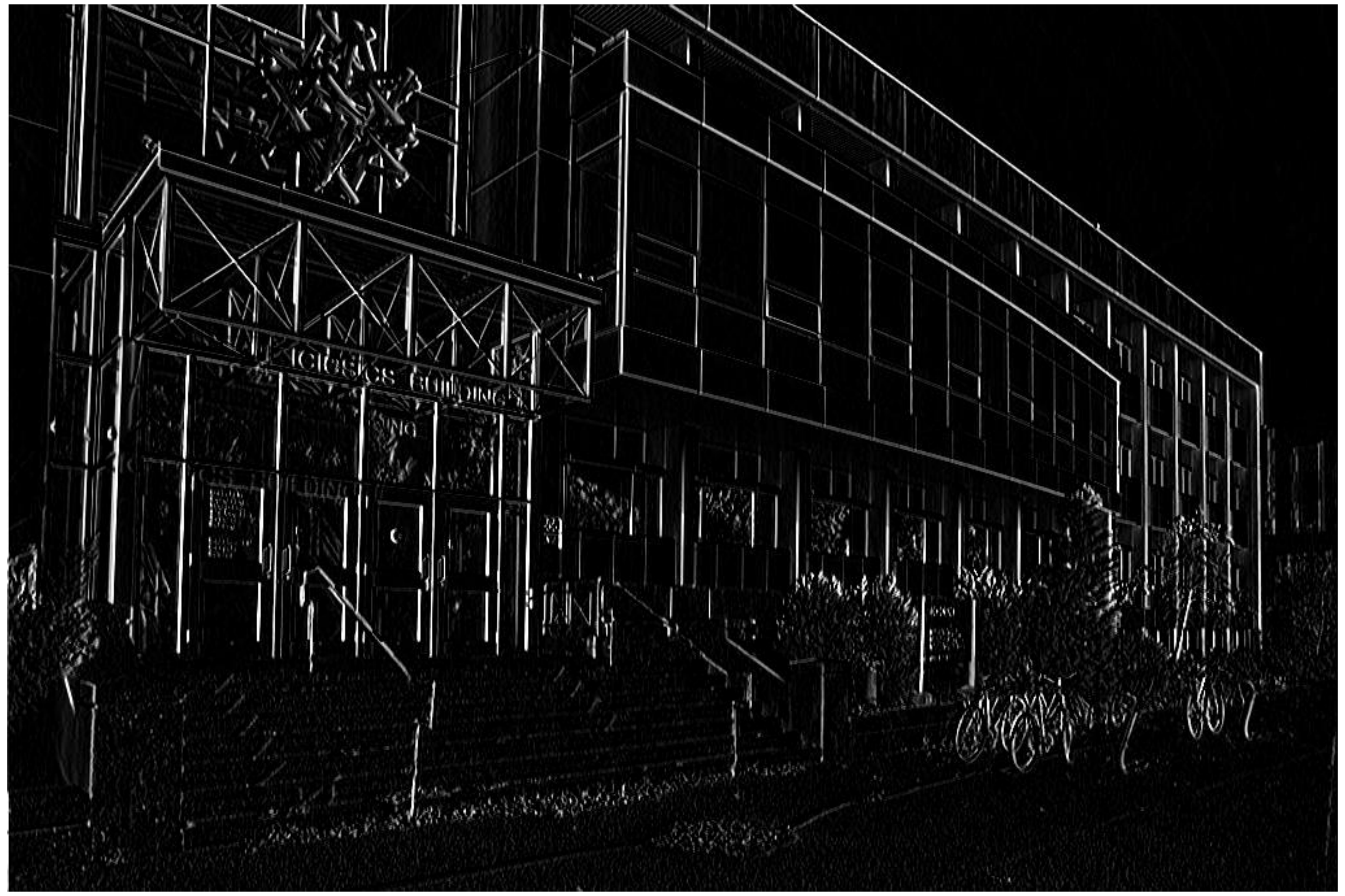


* slide from Marc’Aurelio Renzato

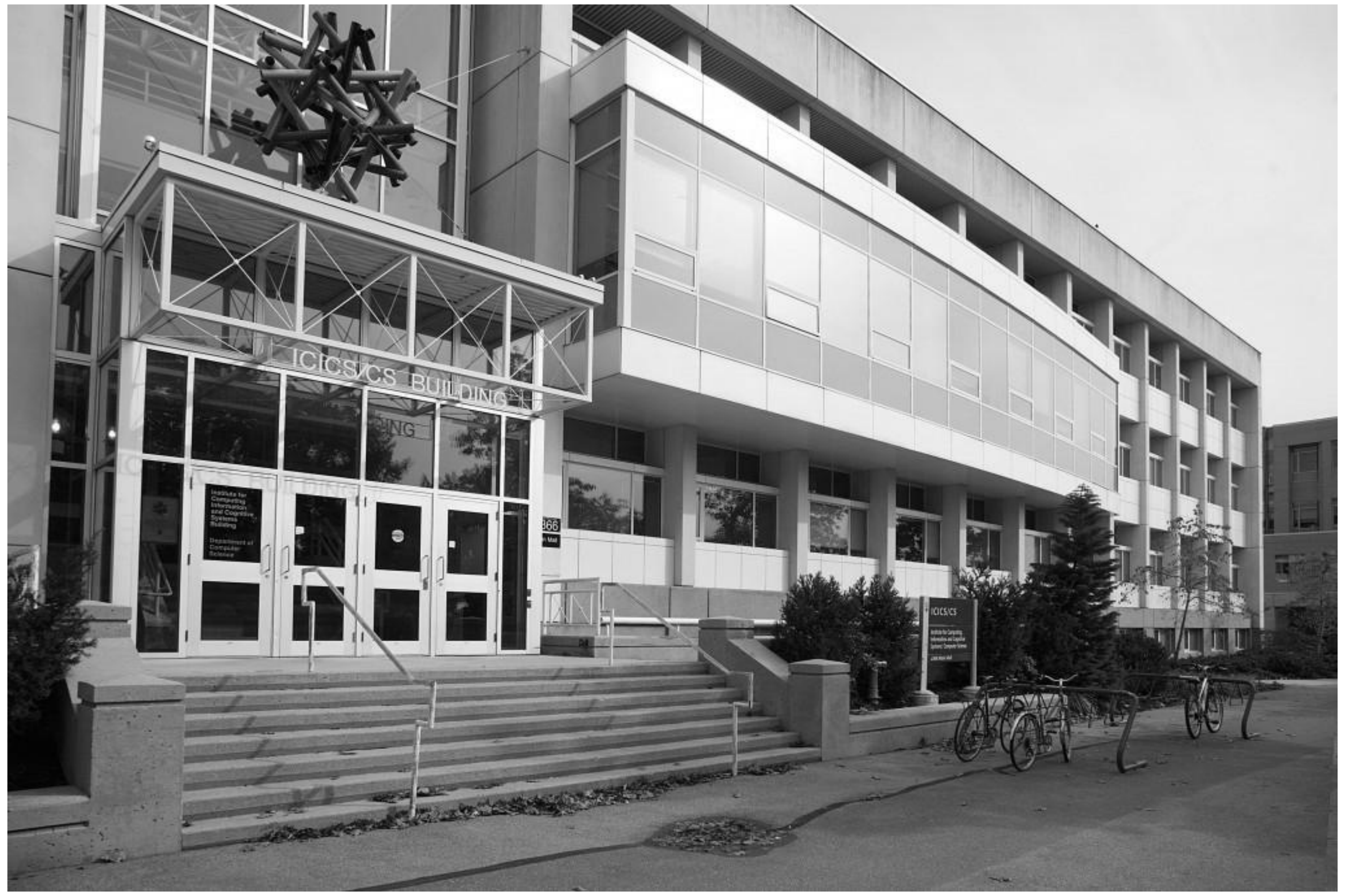
Convolution Layer



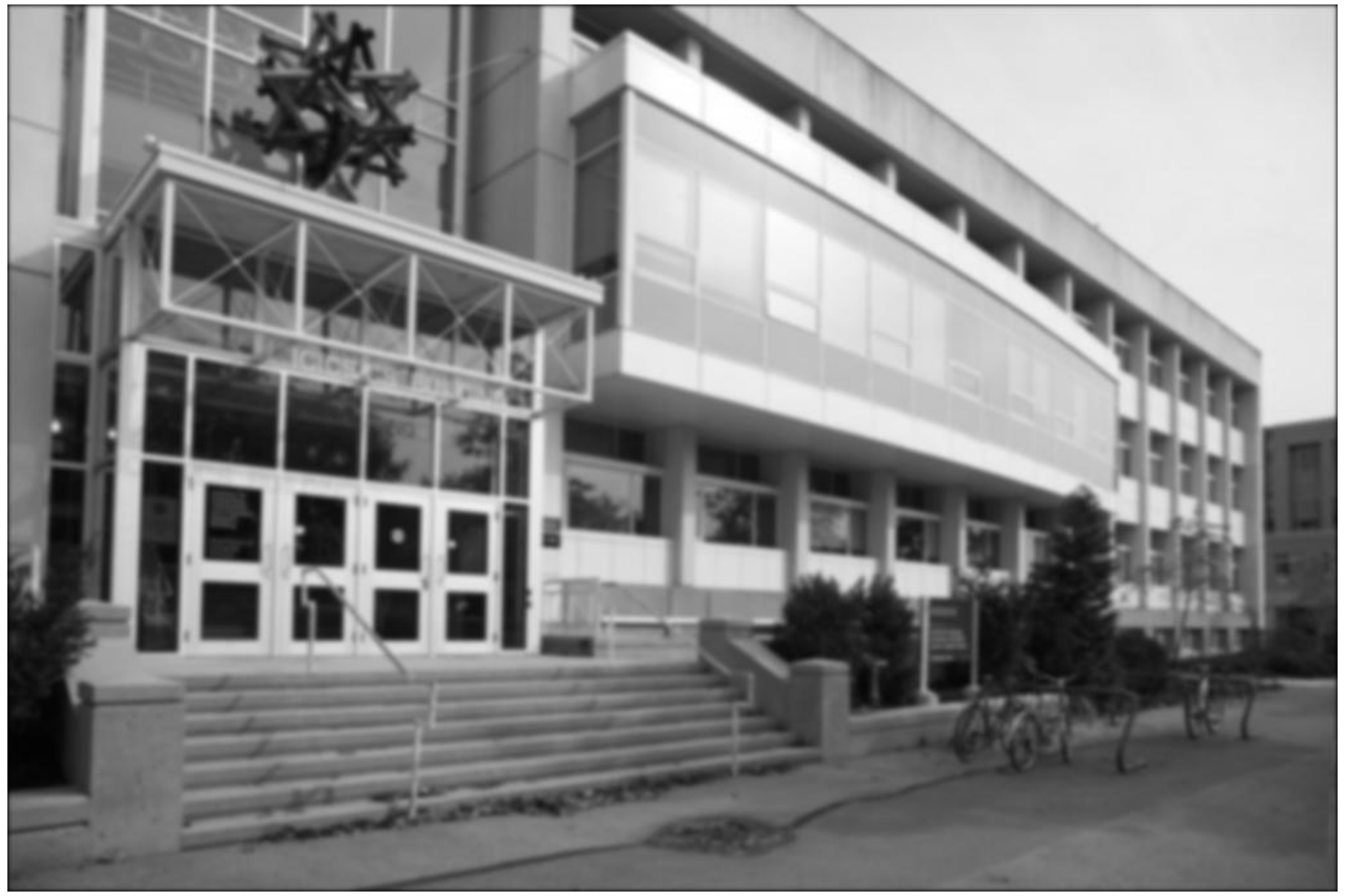
$$\star \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \longrightarrow$$



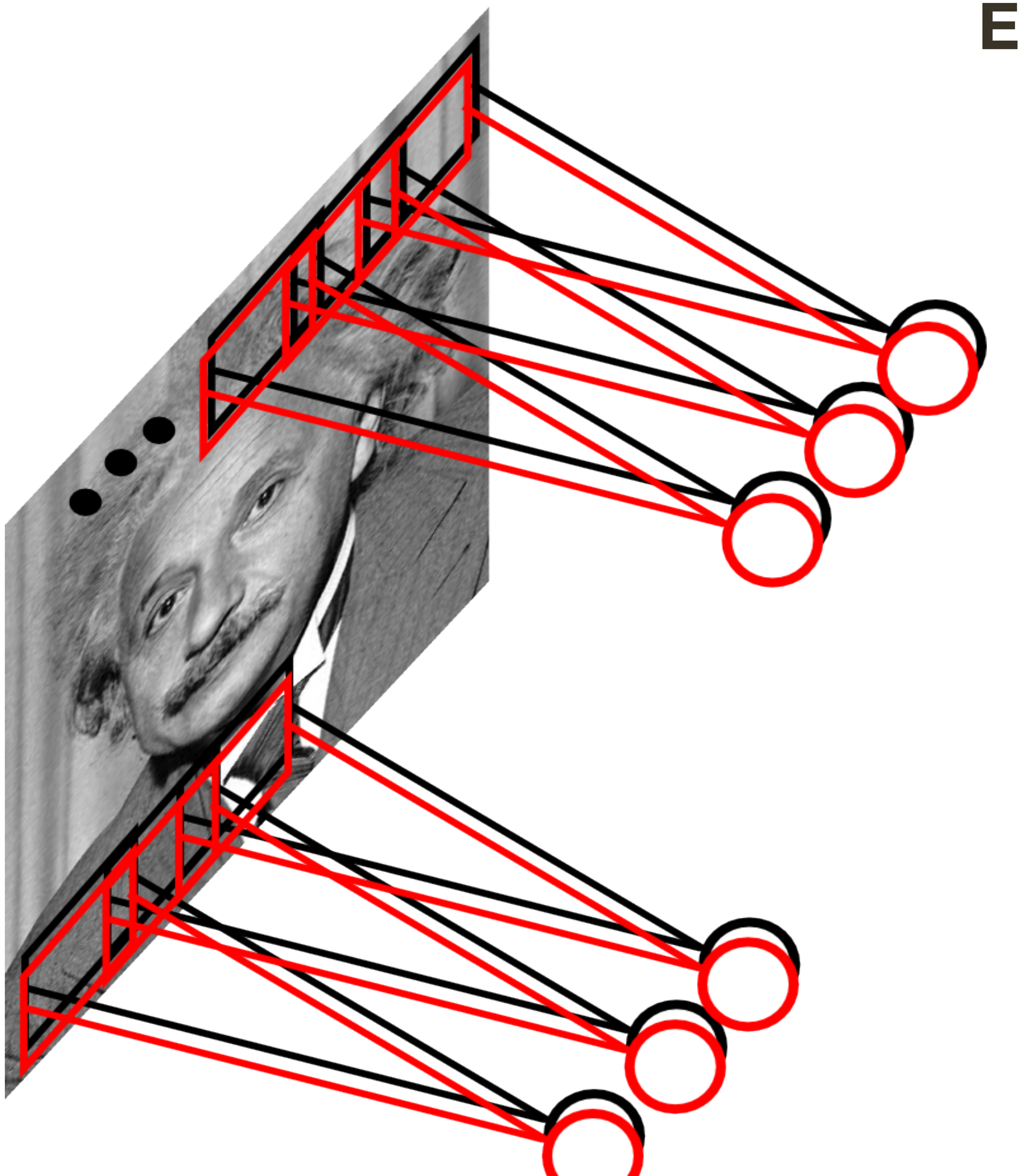
Convolution Layer



$$\star \begin{bmatrix} 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \\ 0.11 & 0.11 & 0.11 \end{bmatrix} \rightarrow$$



Convolutional Layer



Example: 200 x 200 image (small)
x 40K hidden units

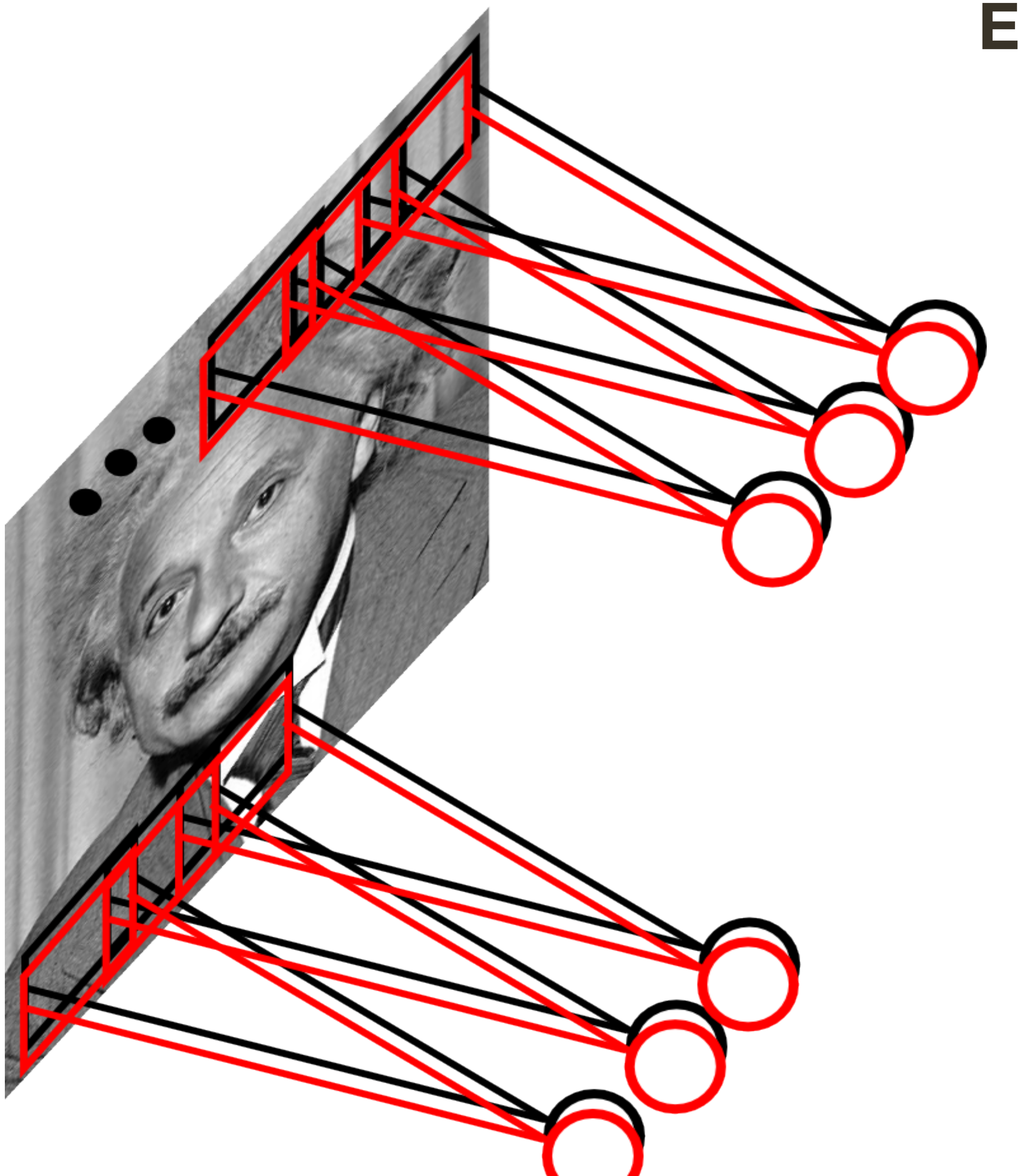
Filter size: 10 x 10

# of filters: 20

Learn **multiple filters**

* slide from Marc'Aurelio Renzato

Convolutional Layer



Example: 200 x 200 image (small)
x 40K hidden units

Filter size: 10 x 10

# of filters: 20

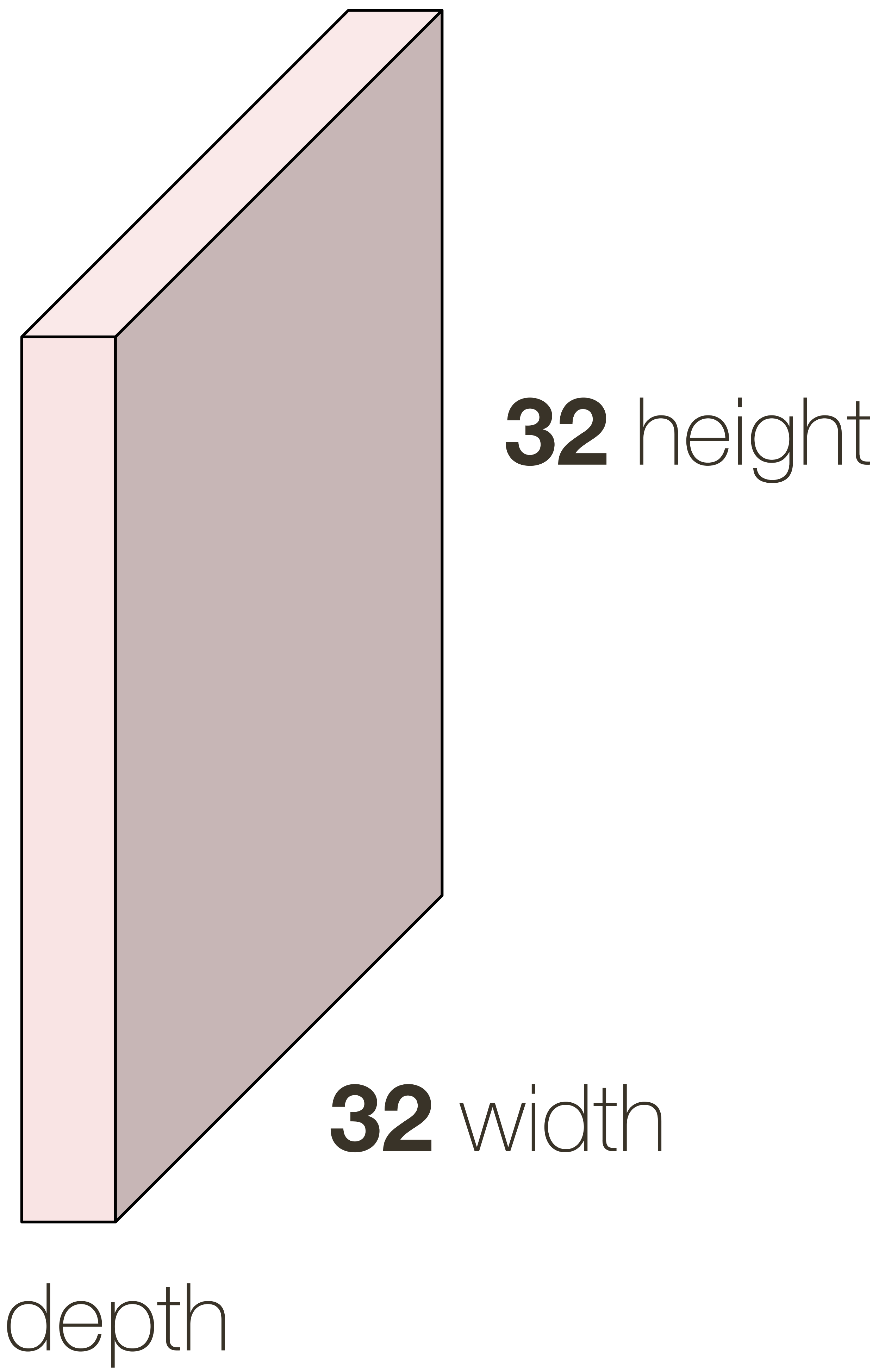
= 2000 parameters

Learn **multiple filters**

* slide from Marc'Aurelio Renzato

Convolutional Layer

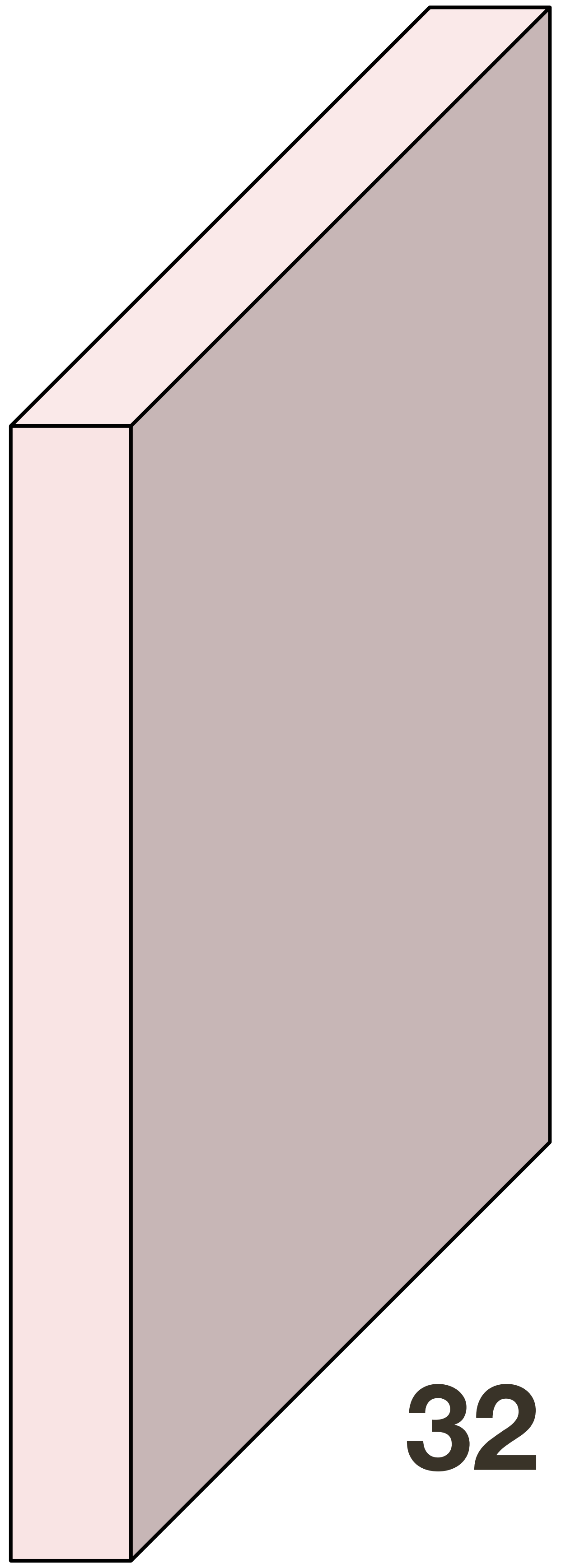
32 x 32 x 3 **image** (note the image preserves spatial structure)



* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Convolutional Layer

32 x 32 x 3 **image**

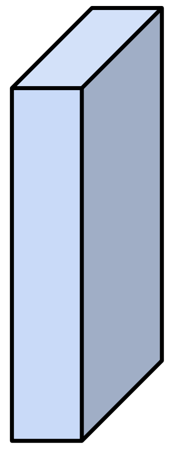


32 height

32 width

3 depth

5 x 5 x 3 **filter**

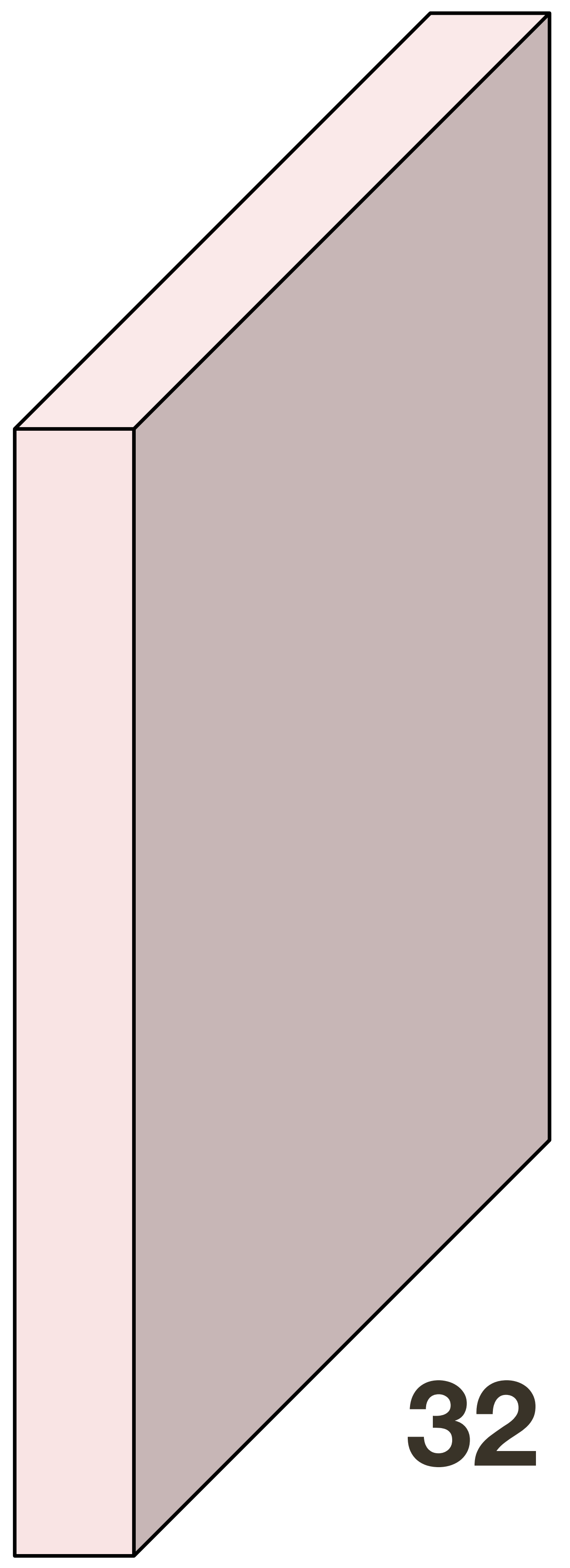


Convolve the filter with the image (i.e., “slide over the image spatially, computing dot products”)

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

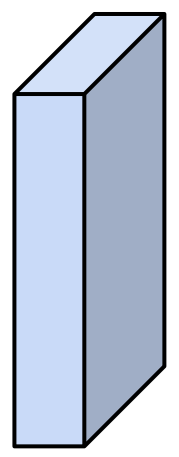
Convolutional Layer

32 x 32 x 3 **image**



Filters always extend the full depth of the input volume

5 x 5 x 3 **filter**

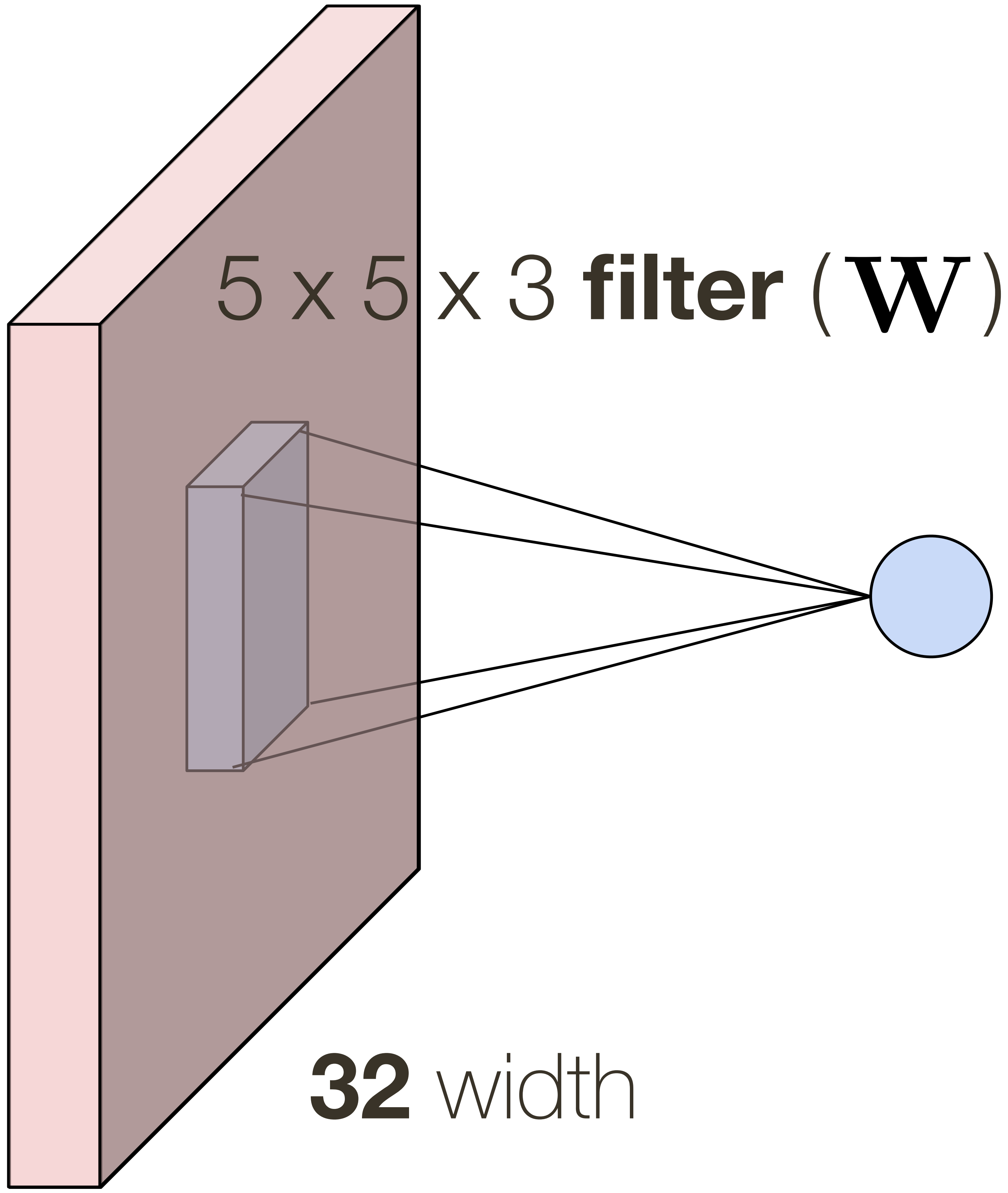


Convolve the filter with the image (i.e., “slide over the image spatially, computing dot products”

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Convolutional Layer

32 x 32 x 3 **image**



5 x 5 x 3 **filter** ($\mathbf{W}$)

32 width

3 depth

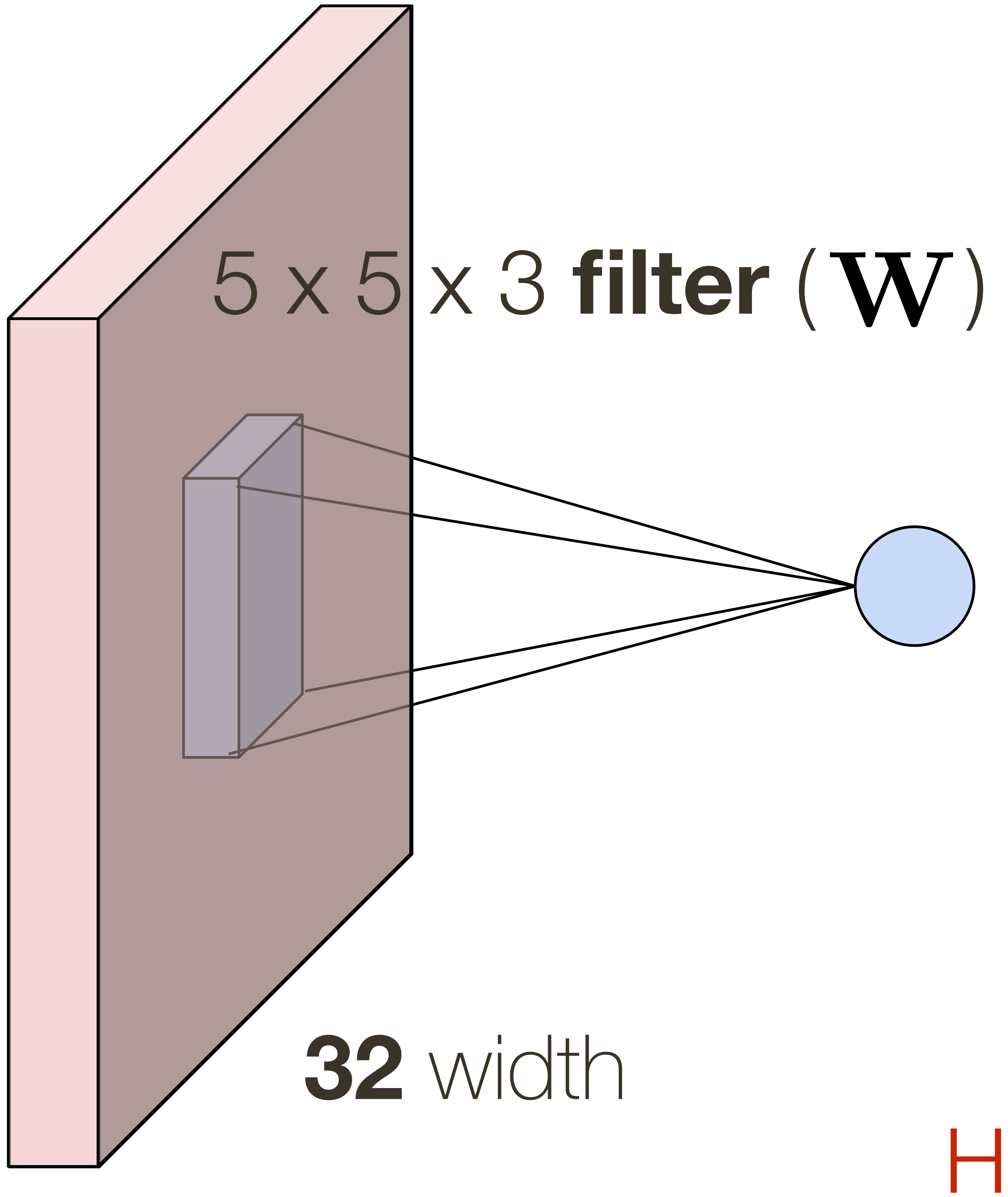
1 number: the result of taking a dot product between the filter and a small 5 x 5 x 3 part of the image

$$\mathbf{W}^T\mathbf{x} + b, \text{where } \mathbf{W}, \mathbf{x} \in \mathbb{R}^{75}$$

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Convolutional Layer

32 x 32 x 3 **image**



5 x 5 x 3 **filter** ($\mathbf{W}$)

32 width

3 depth

1 number: the result of taking a dot product between the filter and a small 5 x 5 x 3 part of the image

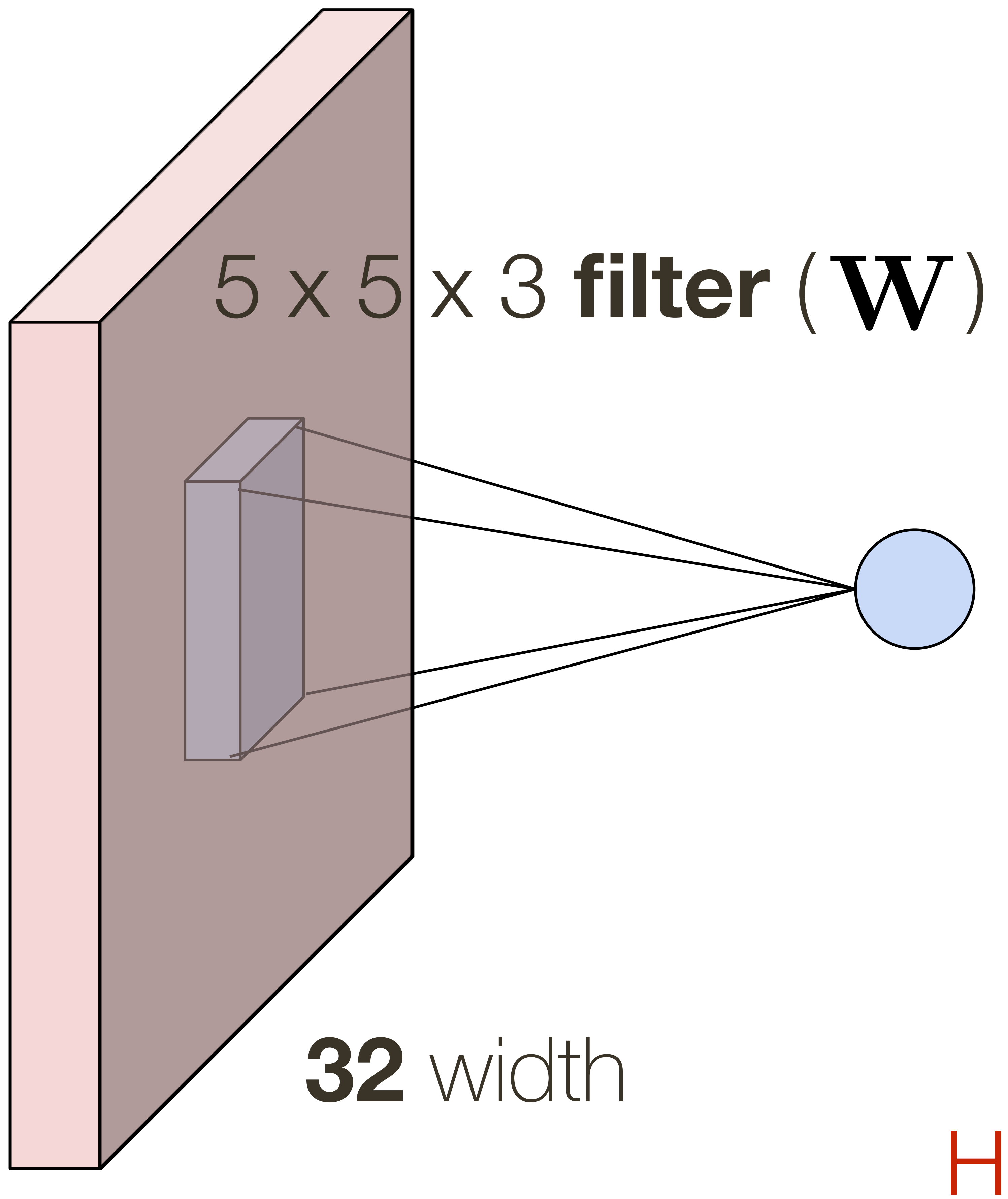
$$\mathbf{W}^T\mathbf{x} + b, \text{where } \mathbf{W}, \mathbf{x} \in \mathbb{R}^{75}$$

How many **parameters** does the layer have?

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Convolutional Layer

32 x 32 x 3 **image**



5 x 5 x 3 **filter** ($\mathbf{W}$)

32 width

3 depth

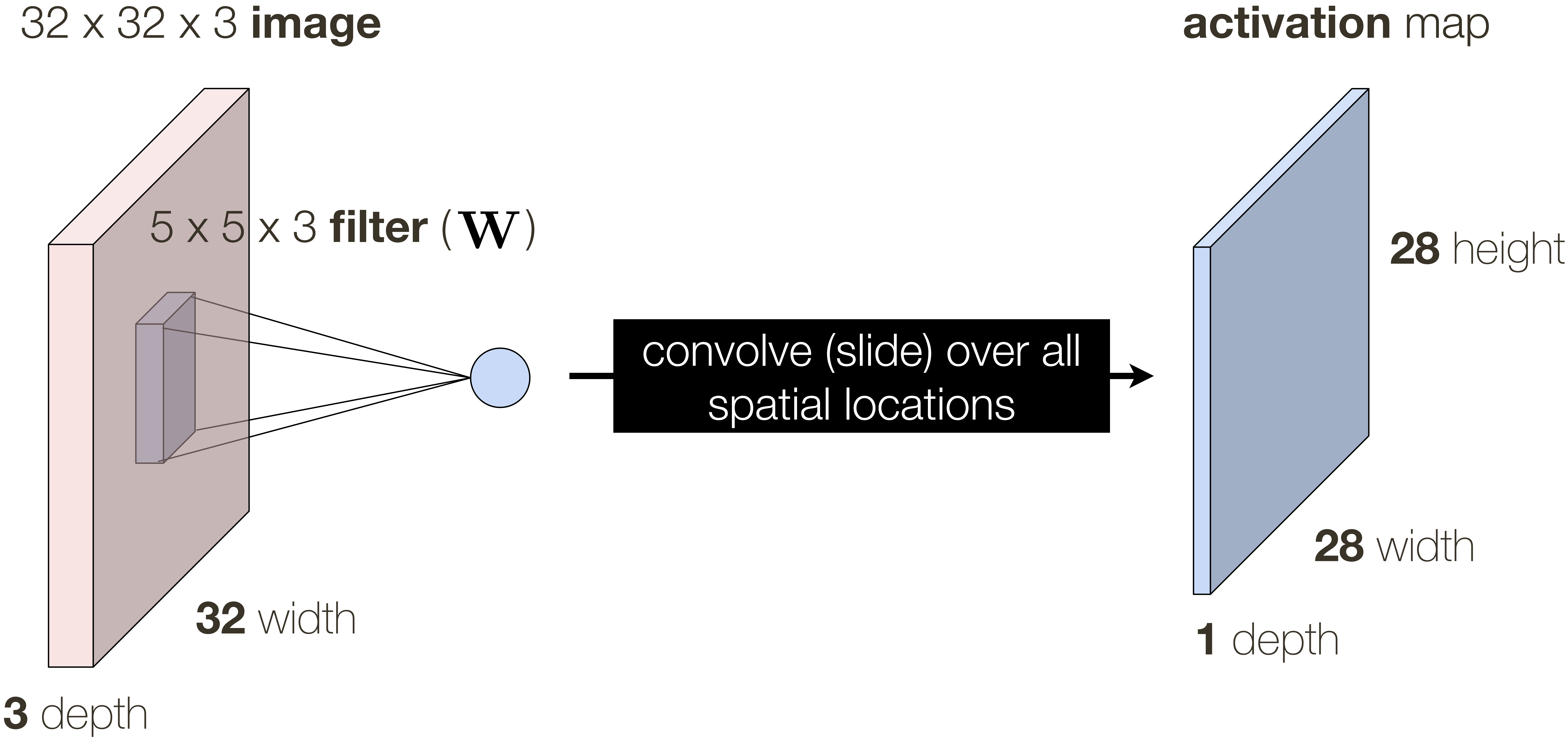
1 number: the result of taking a dot product between the filter and a small 5 x 5 x 3 part of the image

$$\mathbf{W}^T\mathbf{x} + b, \text{where } \mathbf{W}, \mathbf{x} \in \mathbb{R}^{75}$$

How many **parameters** does the layer have? **76**

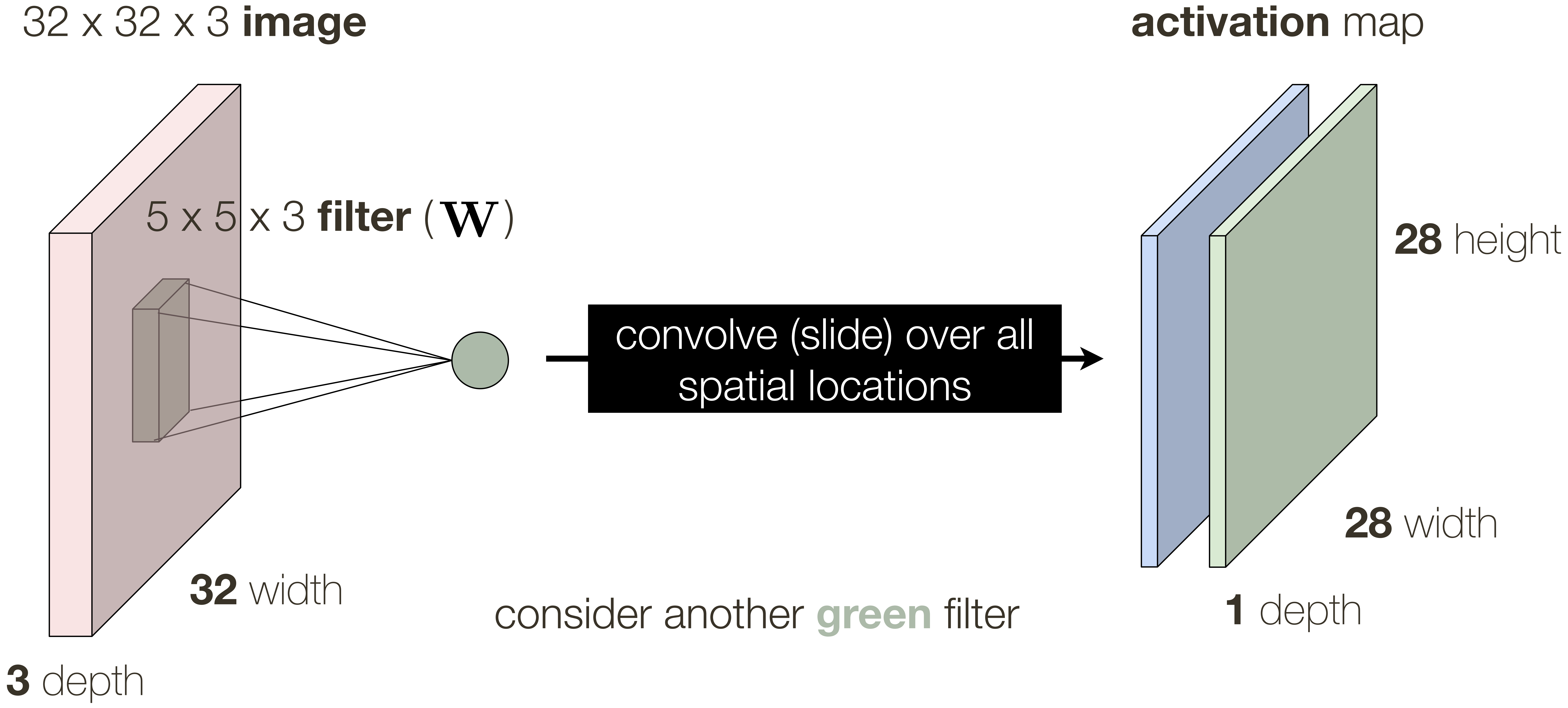
* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Convolutional Layer


32 x 32 x 3 **image**
5 x 5 x 3 **filter** ($\mathbf{W}$)
32 width
3 depth
convolve (slide) over all spatial locations
activation map
28 height
28 width
1 depth

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

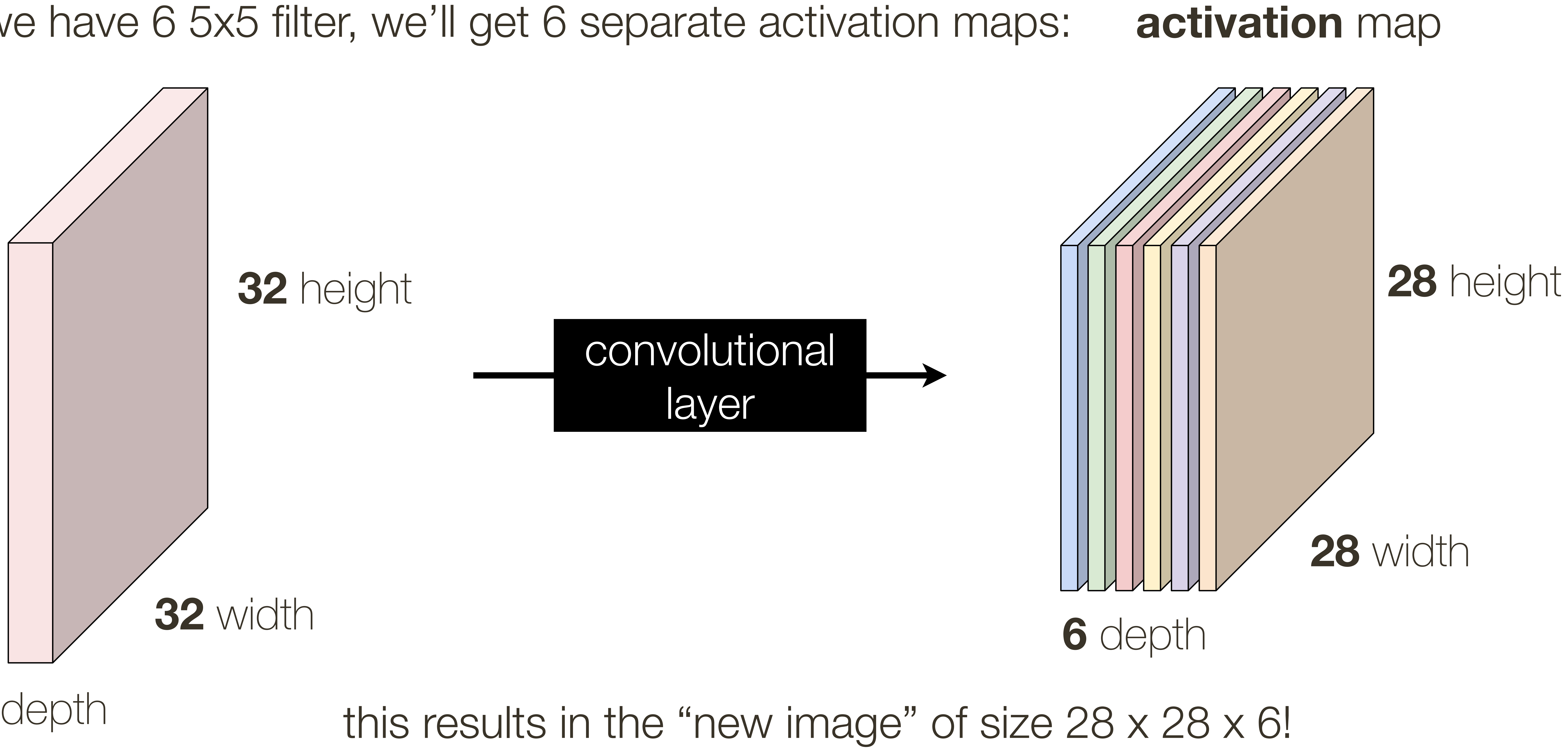
Convolutional Layer


32 x 32 x 3 **image**
5 x 5 x 3 **filter** ($\mathbf{W}$)
32 width
3 depth
convolve (slide) over all spatial locations
consider another green filter
activation map
28 height
28 width
1 depth

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

Convolutional Layer

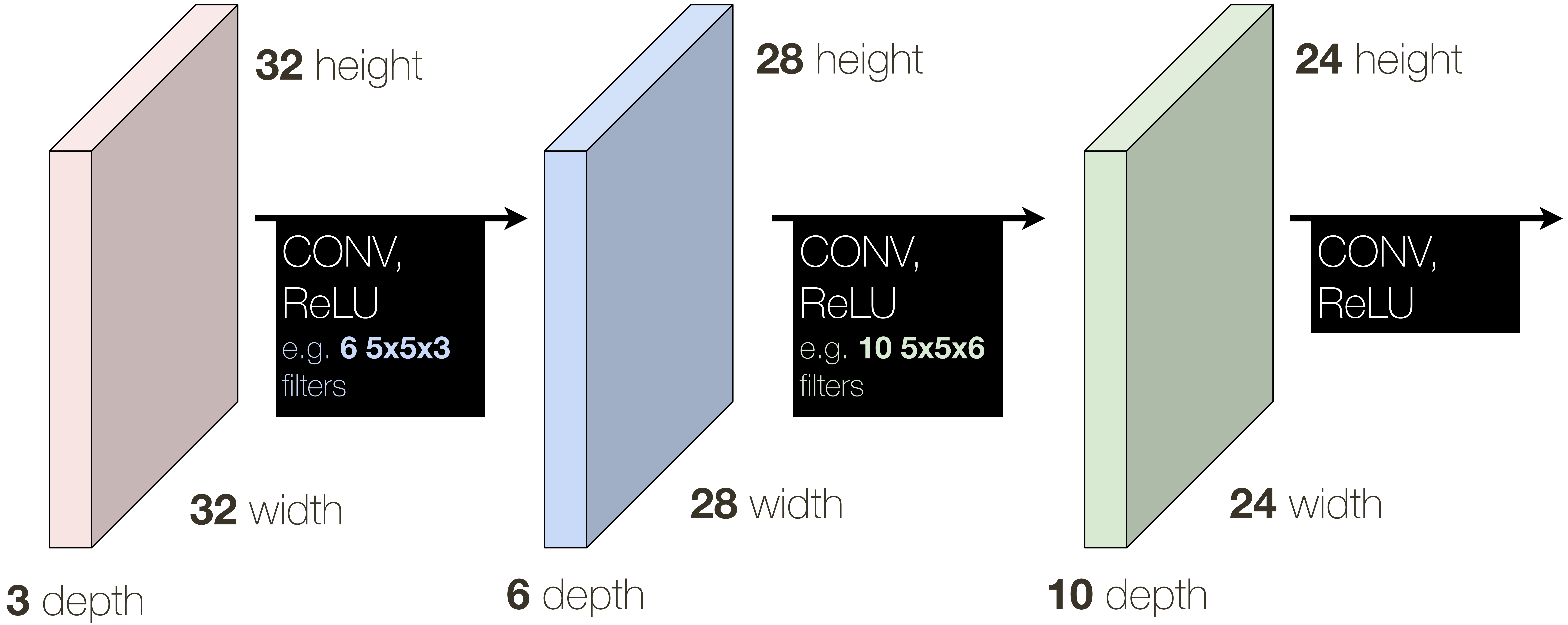
If we have 6 5x5 filter, we'll get 6 separate activation maps:



this results in the "new image" of size 28 x 28 x 6!

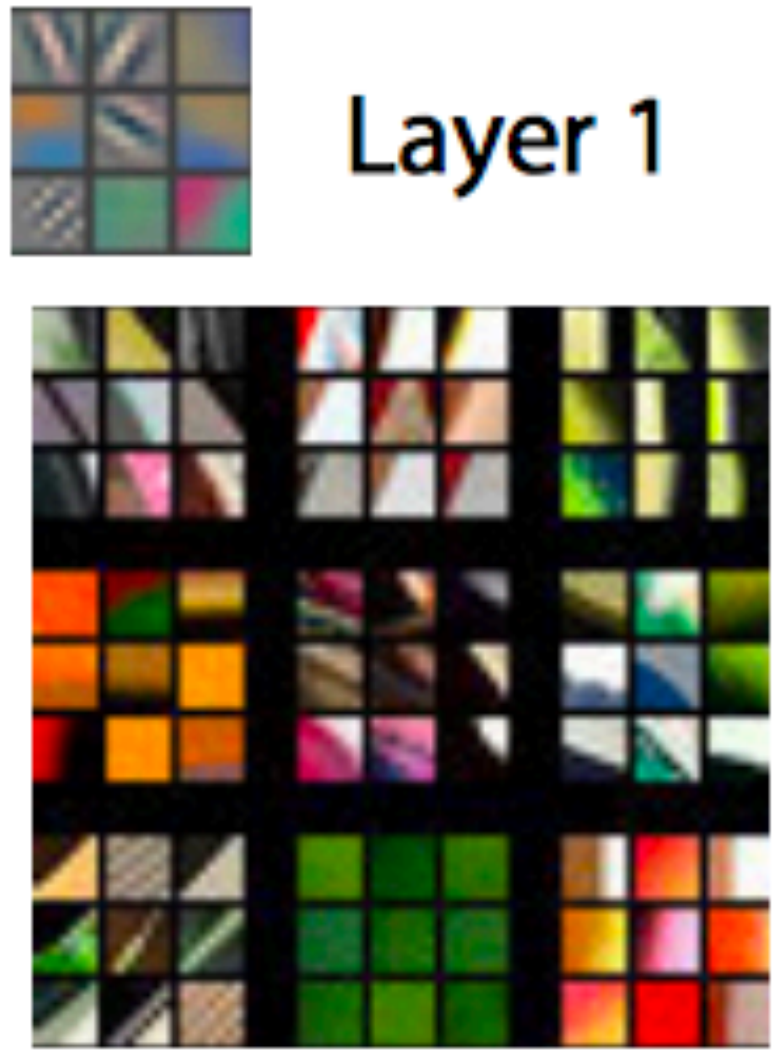
* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

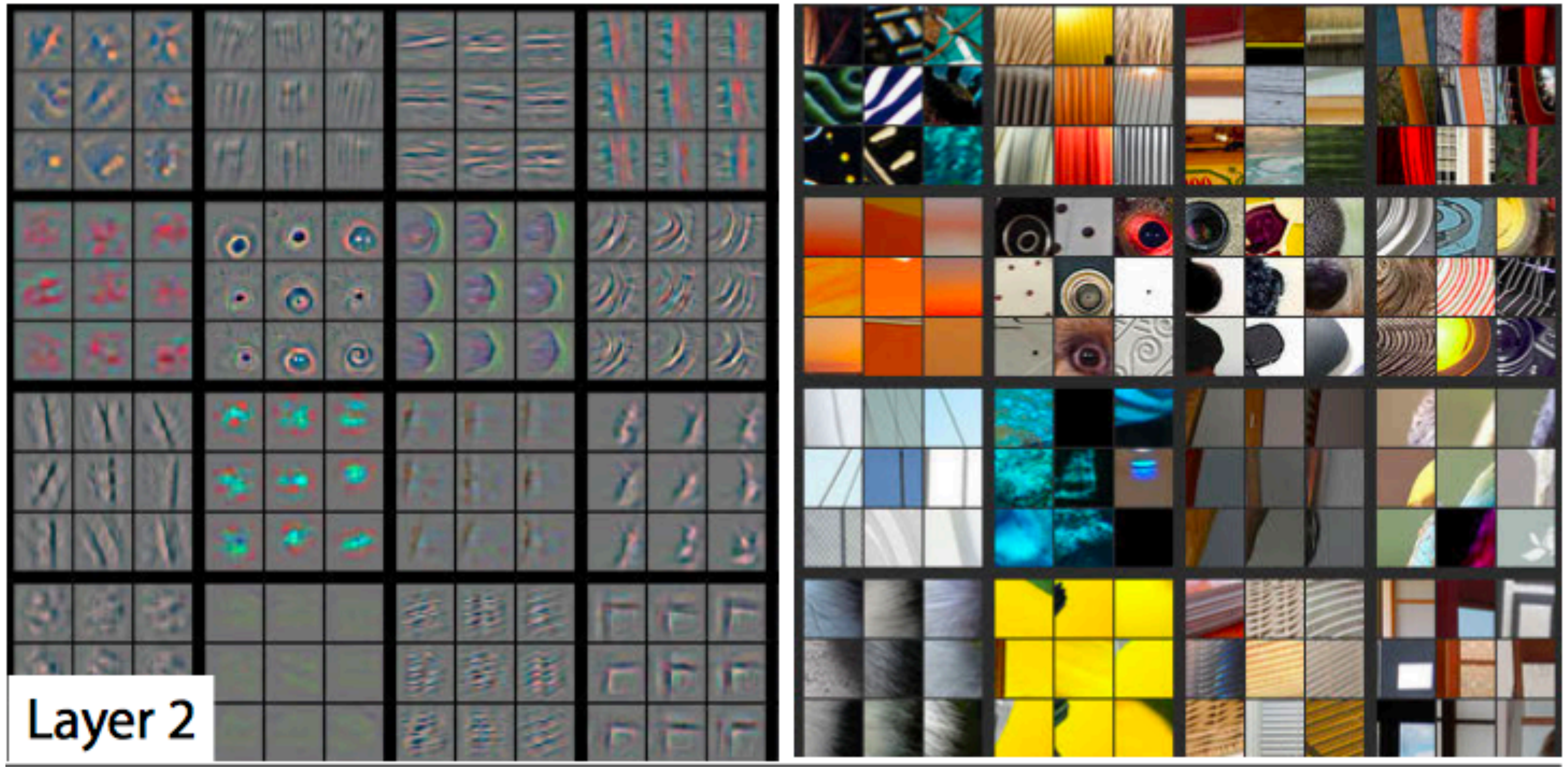
Convolutional Neural Network (ConvNet)



* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

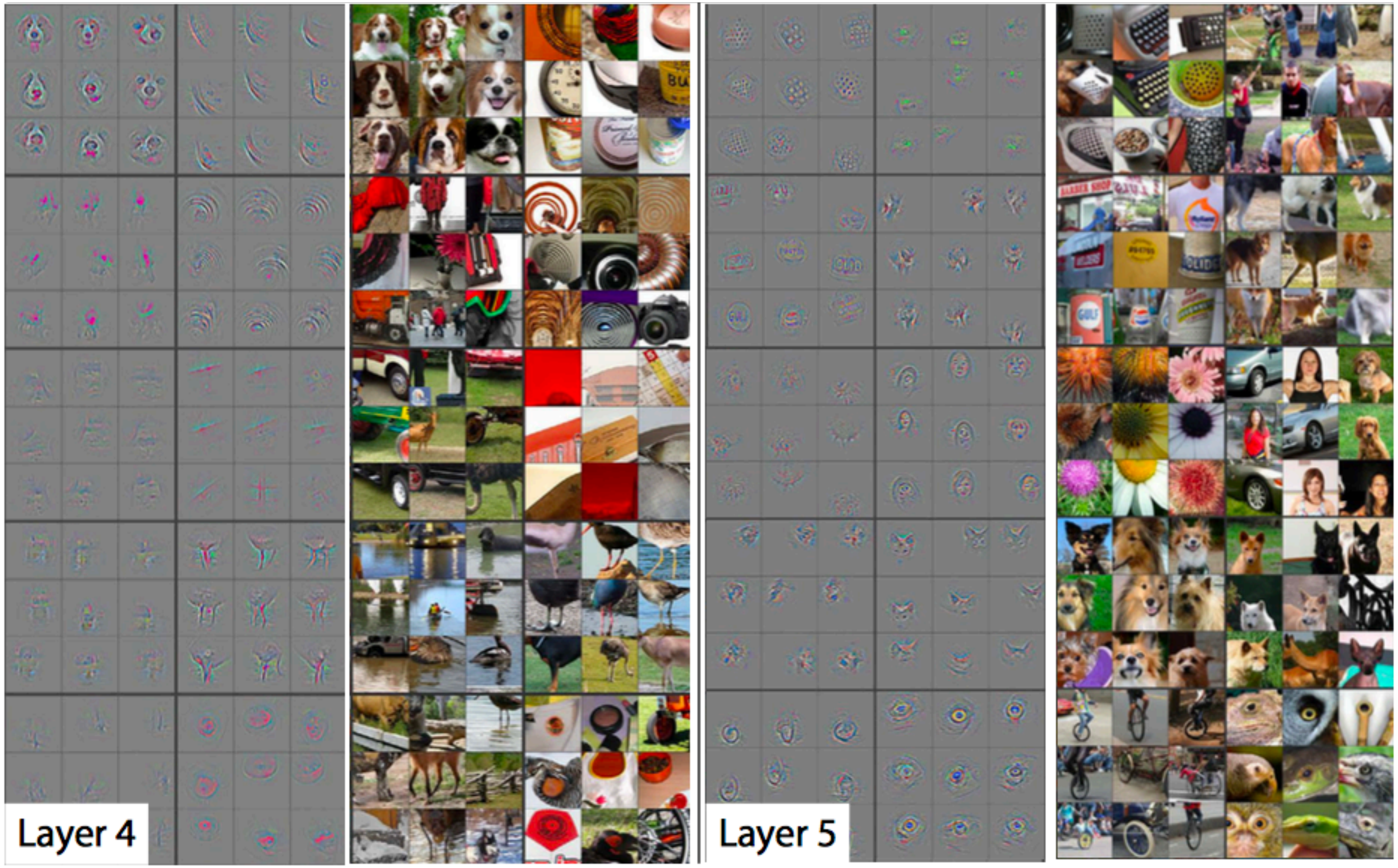
What **filters** do networks learn?





[Zeiler and Fergus, 2013]

What **filters** do networks learn?



[Zeiler and Fergus, 2013]