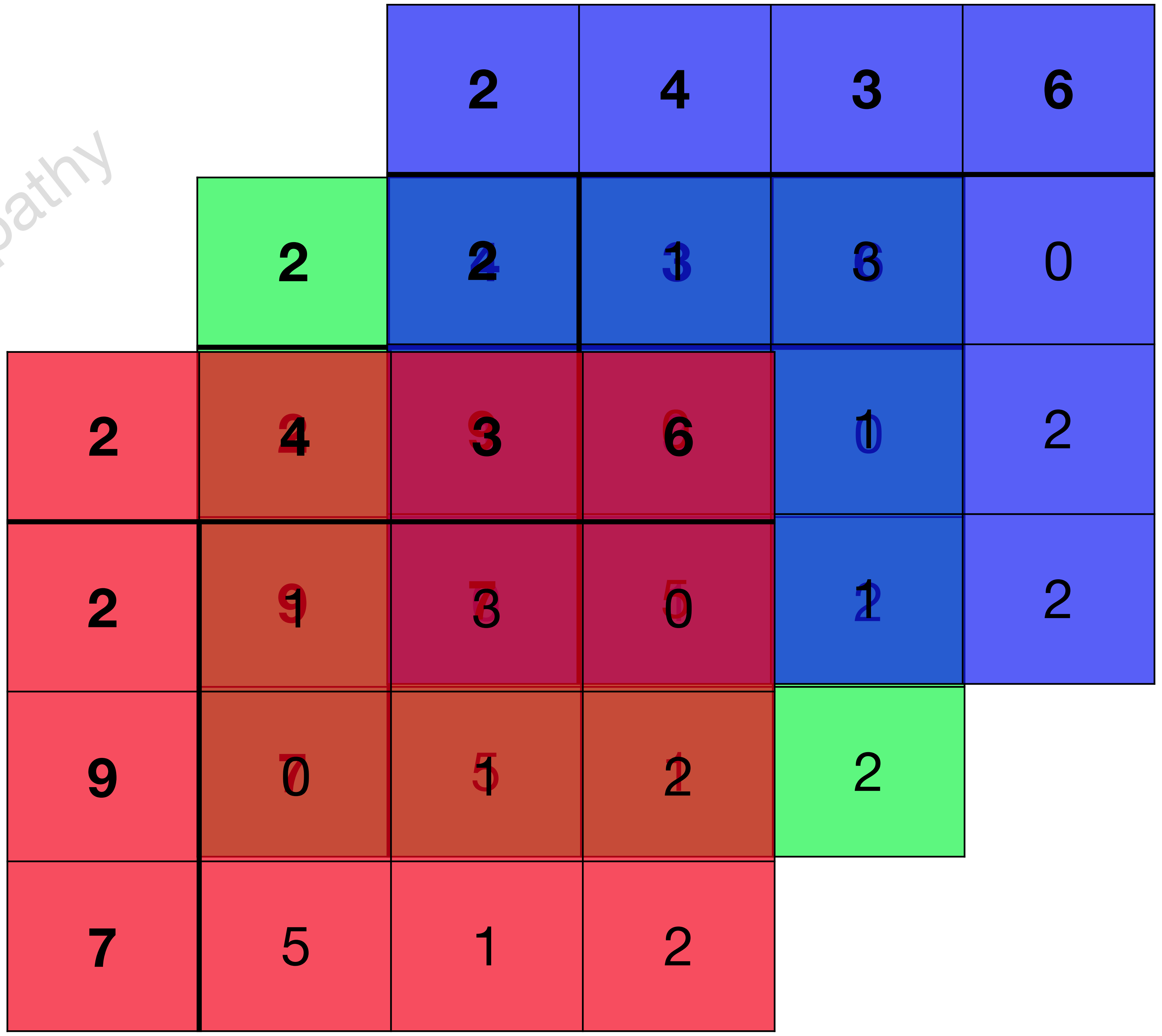
ED5340 - Data Science: Theory and Practise

L28 - Deep Networks

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Three channels - RGB

- RGB (Three channels, one for each colour)



Three channels - RGB

- RGB (Three channels, one for each colour)

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

Convolution on RGB

2D Convolution

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

1	**0**	**-1**
1	0	-1
1	0	-1

1	**0**	**-1**
1	0	-1
1	0	-1

1	0	-1
1	0	-1
1	0	-1

Convolution on RGB

2D Convolution

2 1	4 0	3 -1	6
2 1	1 0	3 -1	0
9 1	0 0	1 -1	2
7	5	1	2

2 1	4 0	3 -1	6
2 1	1 0	3 -1	0
9 1	0 0	1 -1	2
7	5	1	2

2 1	4 0	3 -1	6
2 1	1 0	3 -1	0
9 1	0 0	1 -1	2
7	5	1	2

2*1 + 2*1 + +9 *1
+ 1*0 + 1 *0 +
0*0 + 3*(-1) +
3*(-1) + 1*(-1)
= 6

+

2*1 + 2*1 + +9 *1
+ 1*0 + 1 *0 +
0*0 + 3*(-1) +
3*(-1) + 1*(-1)
= 6

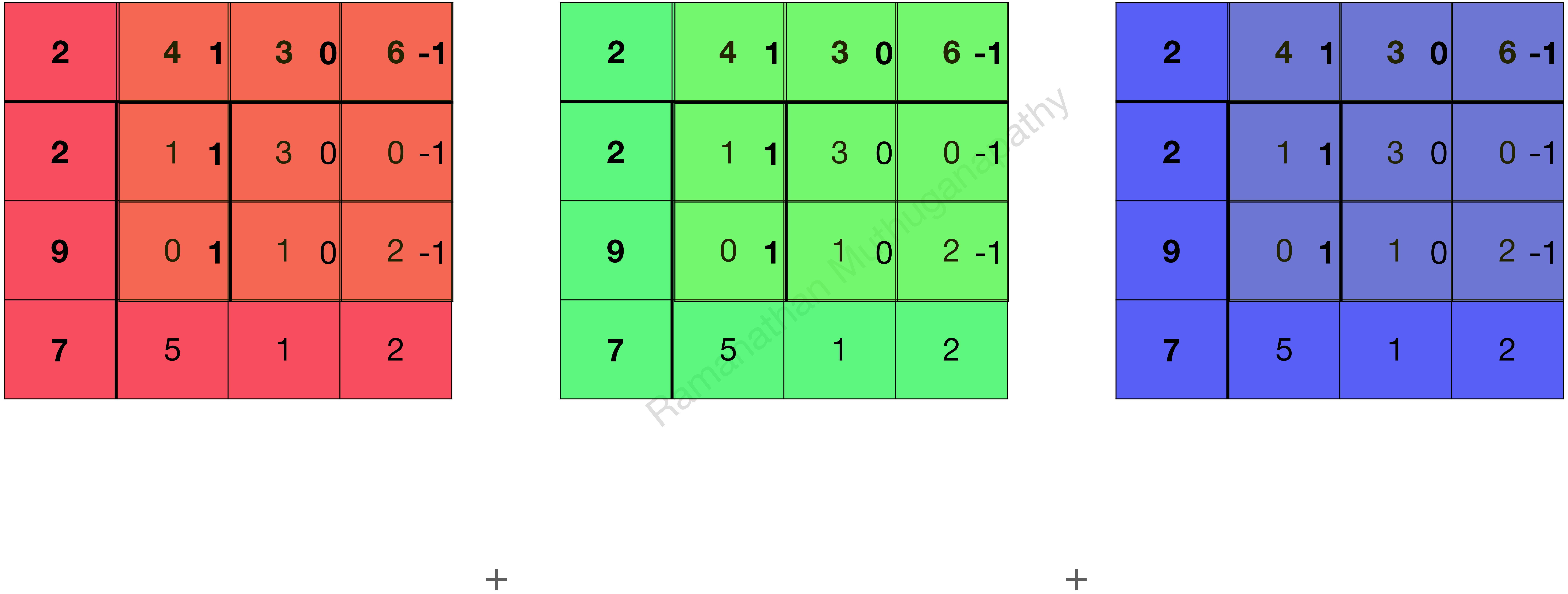
+

2*1 + 2*1 + +9 *1
+ 1*0 + 1 *0 +
0*0 + 3*(-1) +
3*(-1) + 1*(-1)
= 6

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Convolution on RGB

Stride = 1. 2D Convolution



Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Convolution on RGB

Stride = 1. 2D Convolution

2	4	3	6
2 1	1 0	3 -1	0
9 1	0 0	1 -1	2
7 1	5 0	1 -1	2

+

2	4	3	6
2 1	1 0	3 -1	0
9 1	0 0	1 -1	2
7 1	5 0	1 -1	2

+

2	4	3	6
2 1	1 0	3 -1	0
9 1	0 0	1 -1	2
7 1	5 0	1 -1	2

Convolution on RGB

Stride = 1. 2D Convolution

2	4	3	6
2	1 **1**	3 **0**	0 **-1**
9	0 **1**	1 0	2 -1
7	5 **1**	1 0	2 -1

+

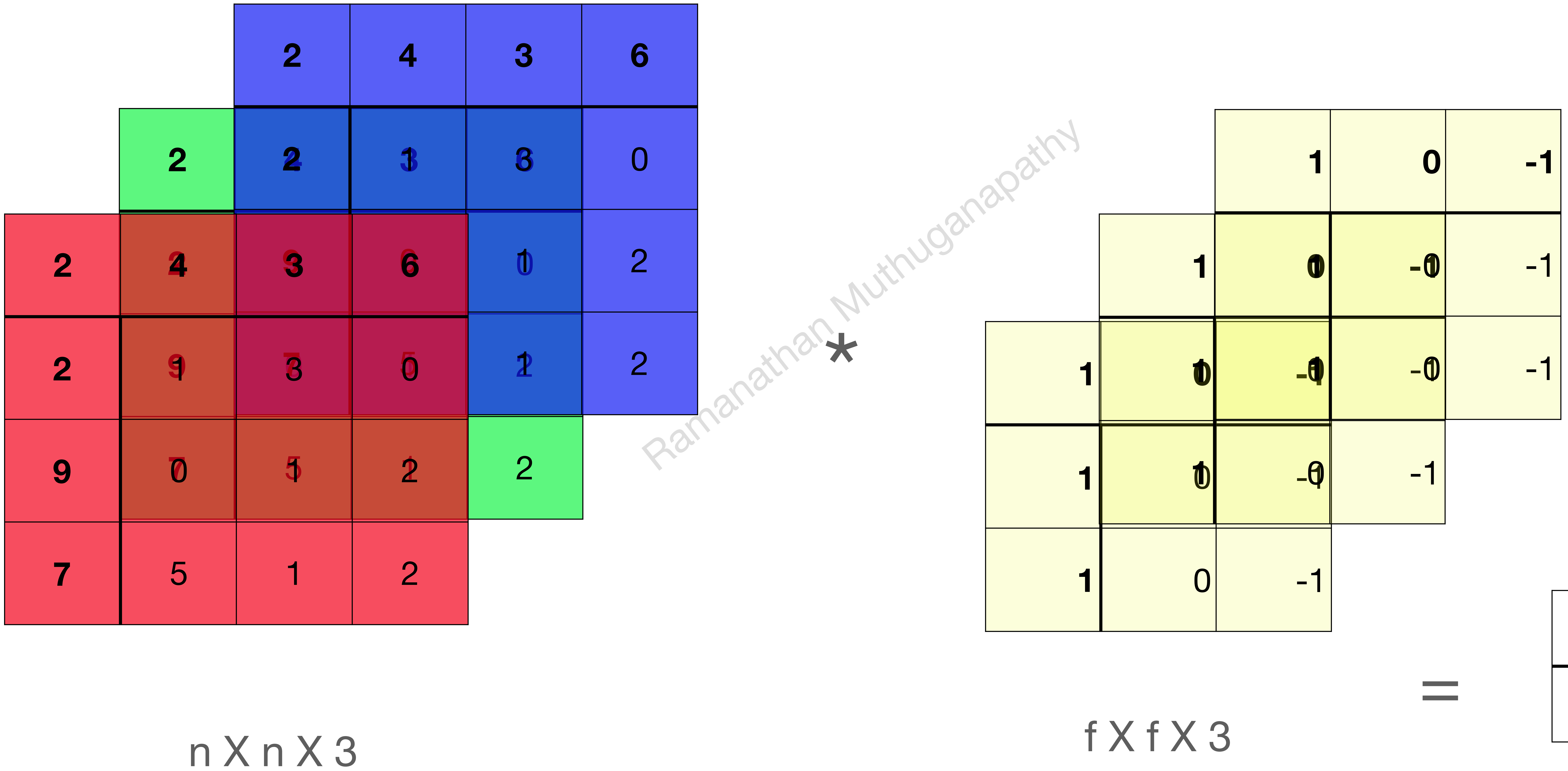
2	4	3	6
2	1 **1**	3 **0**	0 **-1**
9	0 **1**	1 0	2 -1
7	5 **1**	1 0	2 -1

+

2	4	3	6
2	1 **1**	3 **0**	0 **-1**
9	0 **1**	1 0	2 -1
7	5 **1**	1 0	2 -1

Convolution on RGB

2D Convolution (Stride = 1)



18	-9
39	6

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Filter examples

1	**0**	**-1**
1	0	-1
1	0	-1

Vertical Edge

1	**1**	**1**
0	0	0
-1	-1	-1

Horizontal Edge

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

Sobel Edge

w1	**w2**	**w3**
w4	w5	w6
w7	w8	w9

Learning a filter!

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

2D Convolution

- Independent of the number of channels
- Output size is same as in a single channel
- Can be thought of addition of convolution of single channels independently (R, G, B)
- Striding, padding, pooling are done in a similar way

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Take a pic!

- Popular pic used in CV

Take a pic!

- Popular pic used in CV

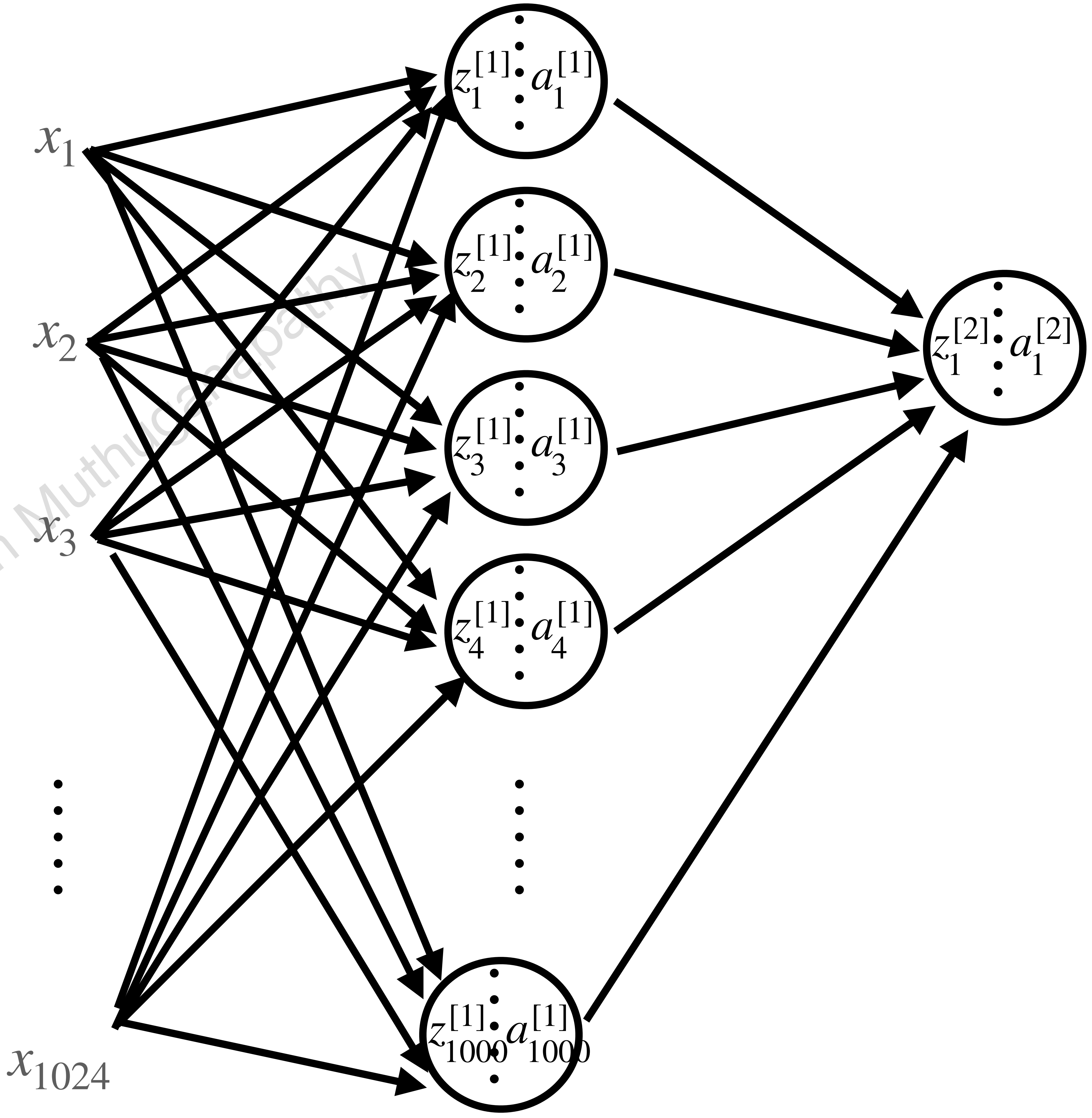
Computing the number of features

- say the pic is of dim 20 X 20
    - 3 Channels
    - nof is 20 X 20 X 3 = 1200
    - 1024 X 1024 X 3 (> 3 million)

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

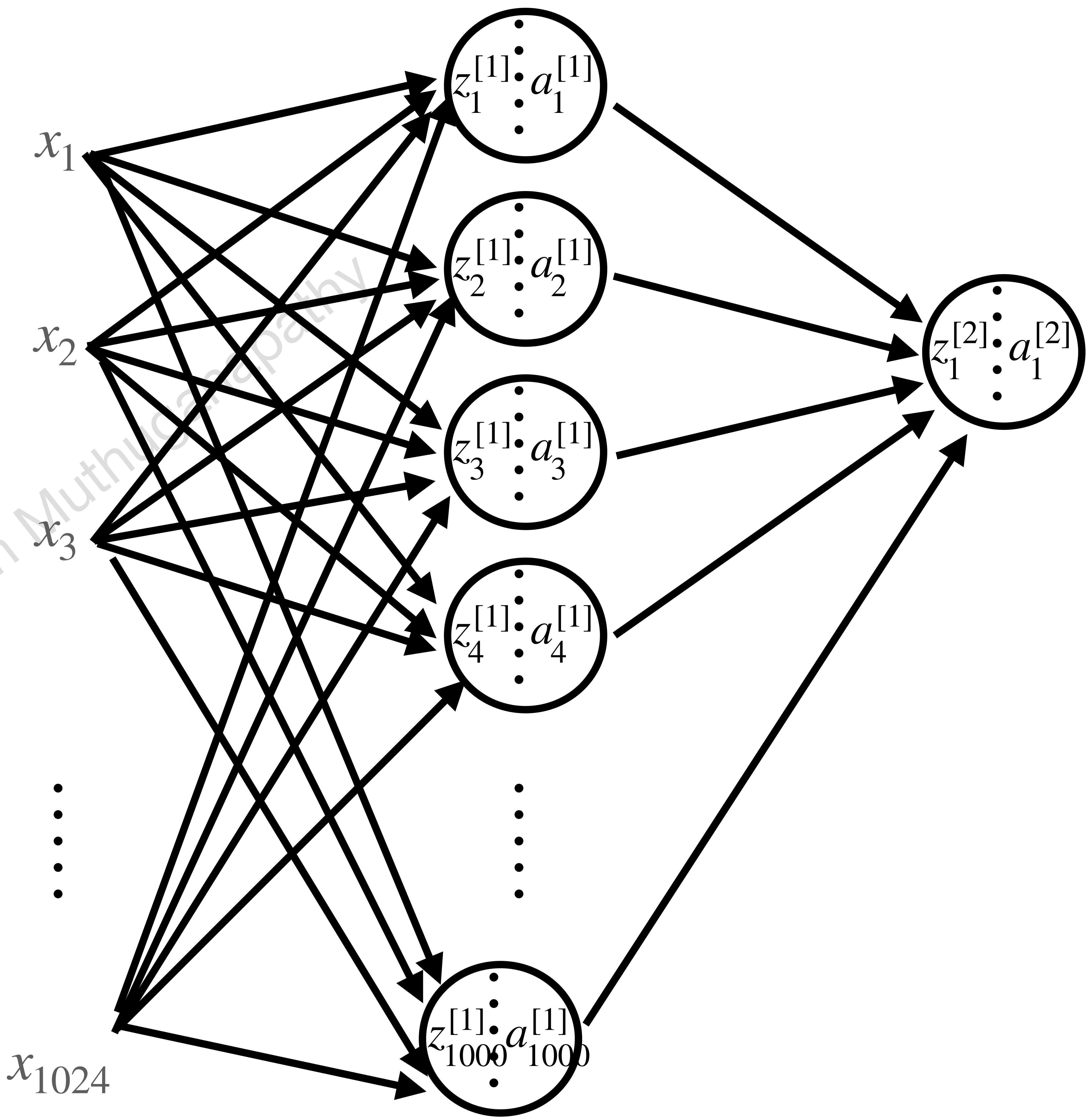
Consider a NN

- 1024 features as input
- 1000 neurons in a layer
- number of weights / parameters?



Consider a NN

- 1024 features as input
- 1000 neurons in a layer
- number of weights / parameters?
  - Parameters will be in billions.
  - Computationally expensive.



Effect of Convolution

On a single channel

2	**4**	**3**	**6**
2	1	3	0
9	0	1	2
7	5	1	2

n X n

*

1	**0**	**-1**
1	0	-1
1	0	-1

f X f

=

6	**-3**
13	2

n-f+1 X n-f+1

1) Original size is not preserved.
2) Boundary pixels are not given importance

Apply Padding

p = 1

0	0	0	0	0	0
0	2	4	3	6	0
0	2	1	3	0	0
0	9	0	1	2	0
0	7	5	1	2	0
0	0	0	0	0	0

n X n, p = 1

*

1	**0**	**-1**
1	0	-1
1	0	-1

f X f

=

-5	**-2**	**-1**	**6**
-5	6	-3	7
-6	18	2	5
-5	14	1	2

n+2p-f+1 X n+2p-f+1

Terminology

- 'Valid' convolution -
  - n-f+1 X n-f+1
  - No padding
- 'Same' convolution
  - Input size is preserved
  - Padding is done
  - n+2p-f+1 X n+2p-f+1

Condition for ‘same’

- $n+2p-f+1 = n$
- $p = (f - 1)/ 2$
- f is usually ‘odd’
- f is 3, $p = 1$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Apply Stride

0 **1**	0 **0**	0 **-1**	0	0	0
0 **1**	2 0	4 -1	3	6	0
0 **1**	2 0	1 -1	3	0	0
0	9	0	1	2	0
0	7	5	1	2	0
0	0	0	0	0	0

-5			

Apply Stride

0	0 **1**	0 **0**	0 **-1**	0	0
0	2 **1**	4 0	3 -1	6	0
0	2 **1**	1 0	3 -1	0	0
0	9	0	1	2	0
0	7	5	1	2	0
0	0	0	0	0	0

-5	**-2**		

Apply Stride

0	0	0	0	0	0
0	2	4	3	6	0
0	2	1	3	0	0
0	9	0	1 **1**	2 **0**	0 **-1**
0	7	5	1 **1**	2 0	0 -1
0	0	0	0 **1**	0 0	0 -1

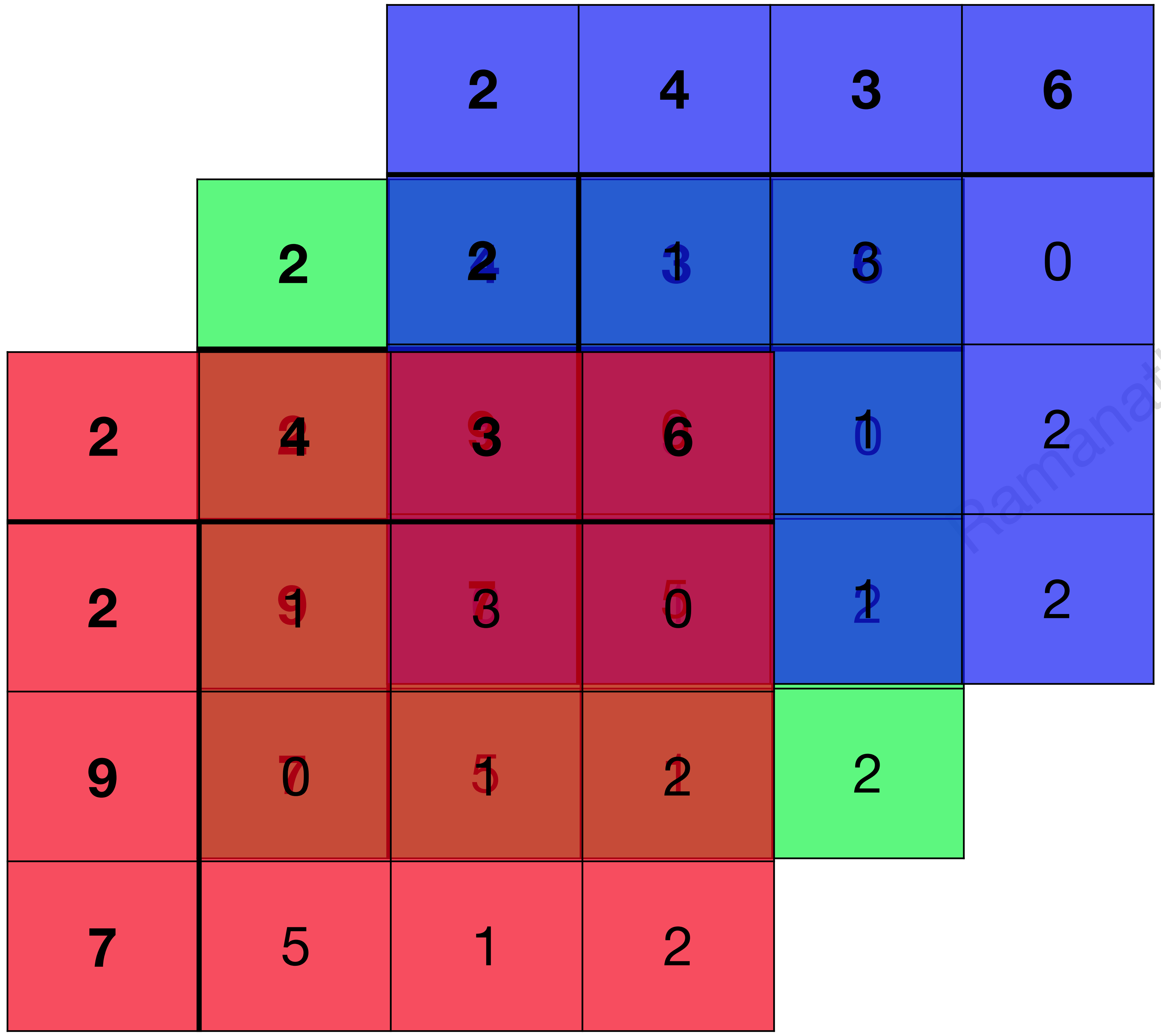
n X n, p = 1, s

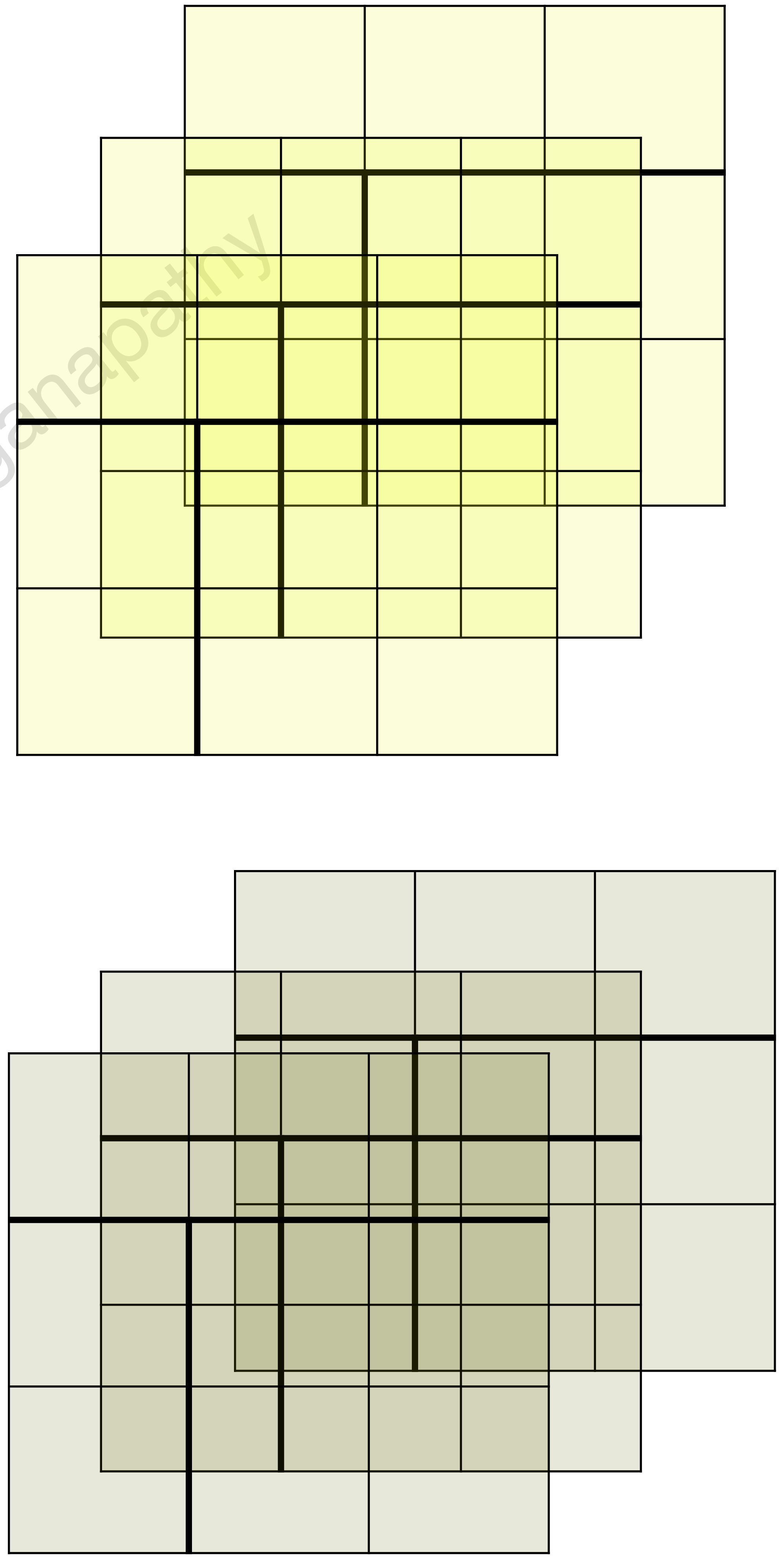
f X f

-5	**-2**	**-1**	**6**
-5	6	-3	7
-6	18	2	5
-5	14	1	2

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

Convolution layer





3X 3 X 3

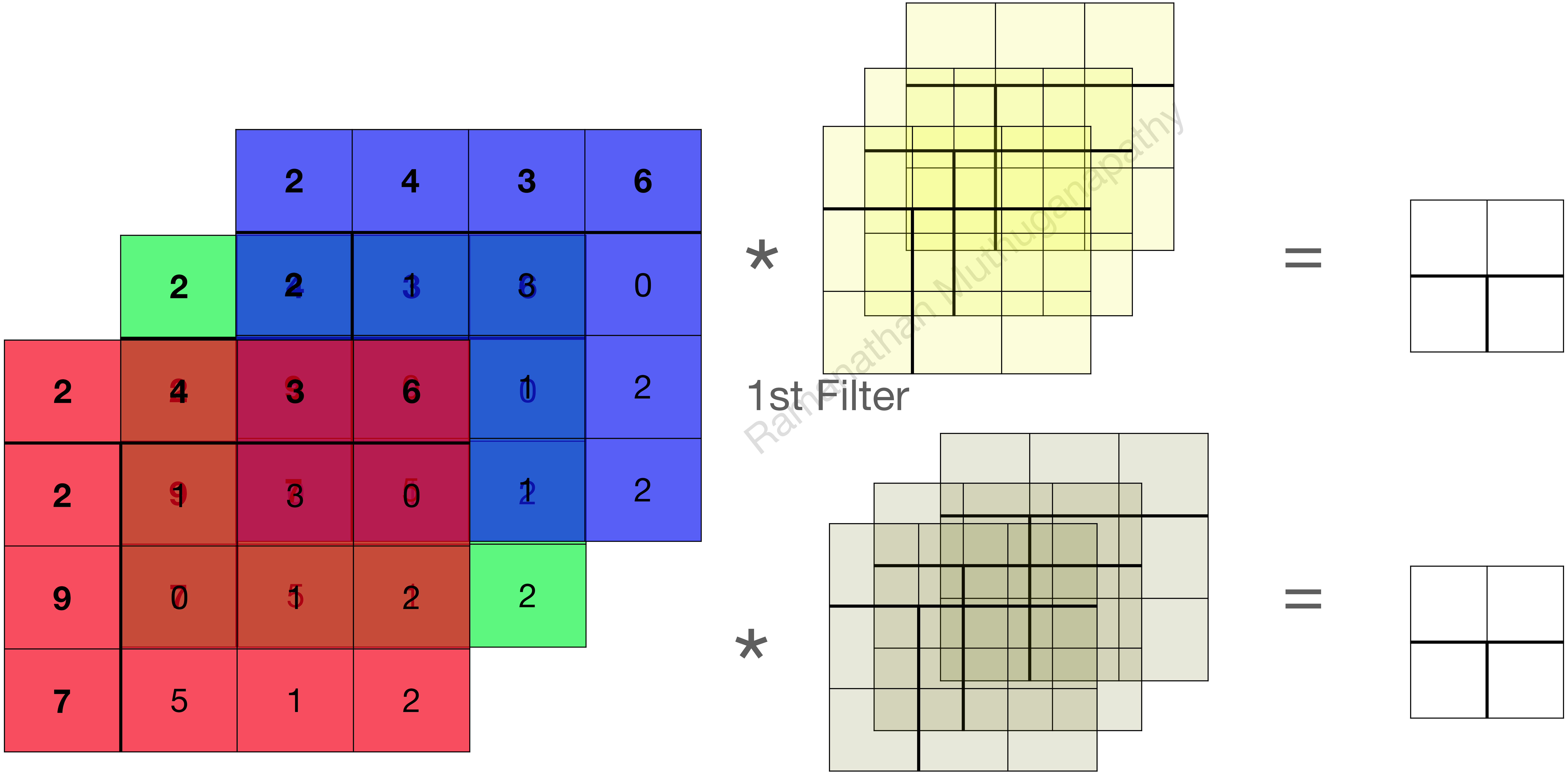
1st Filter

3X 3 X 3

2nd Filter

Convolution layer

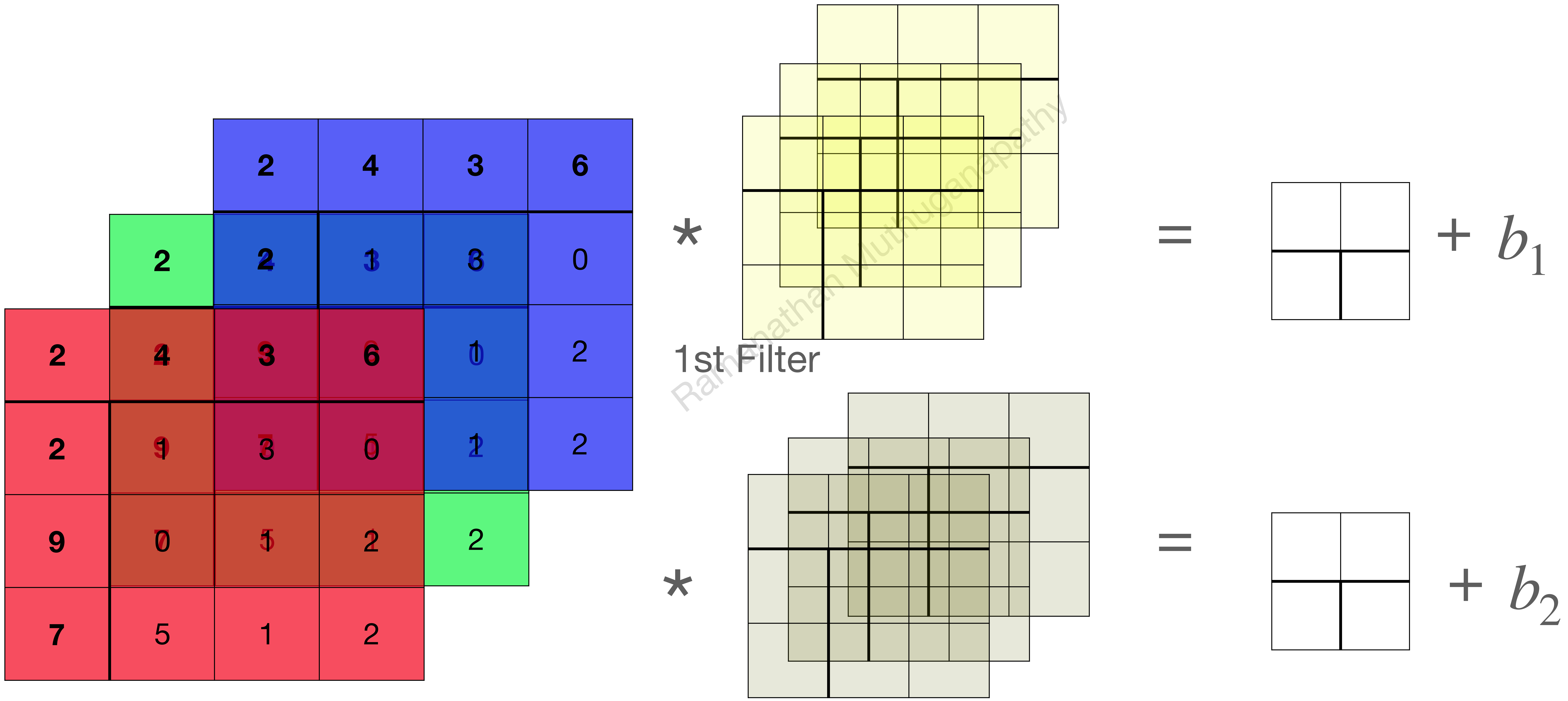
Filter using stride



2nd Filter

Convolution layer

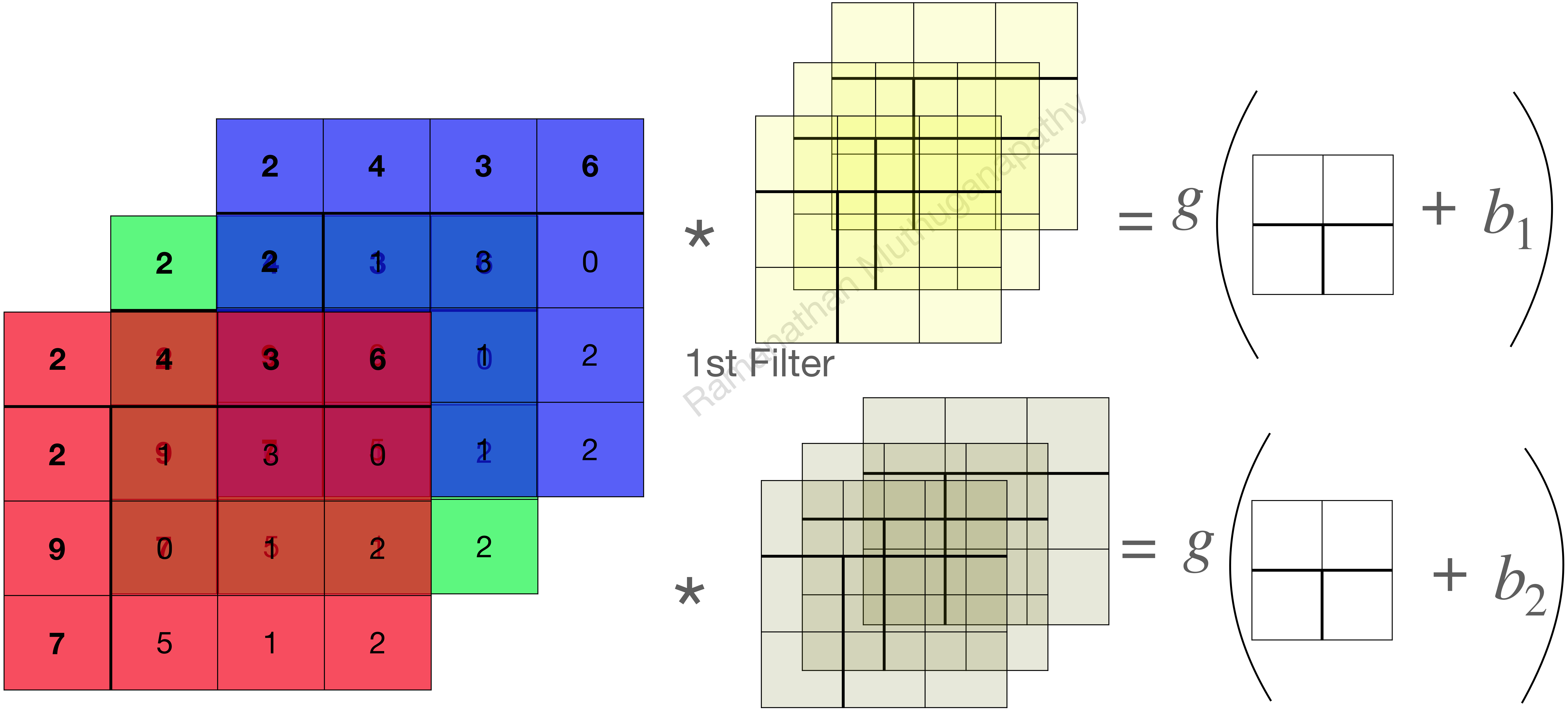
Include bias



Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Convolution layer

Apply activation

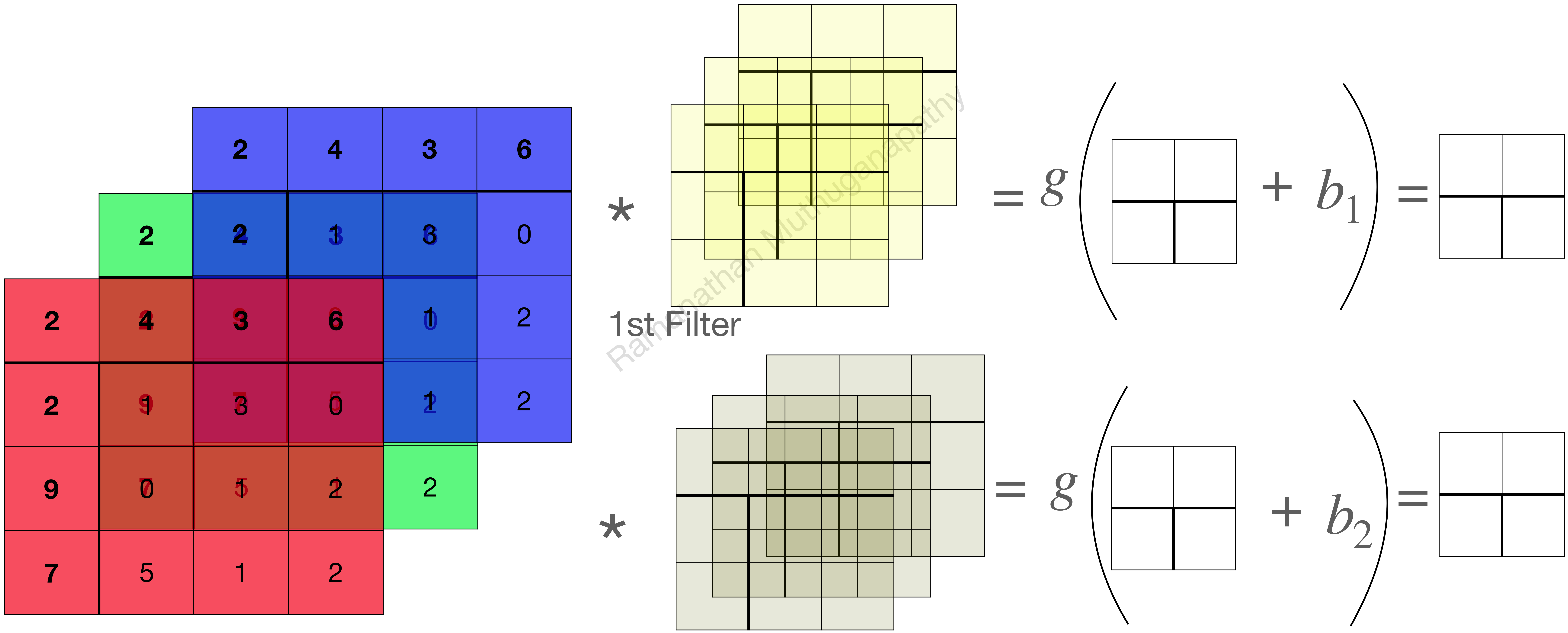


$$* \quad = g\left(\quad + b_1 \right)$$

$$* \quad = g\left(\quad + b_2 \right)$$

Convolution layer

Output



2nd Filter

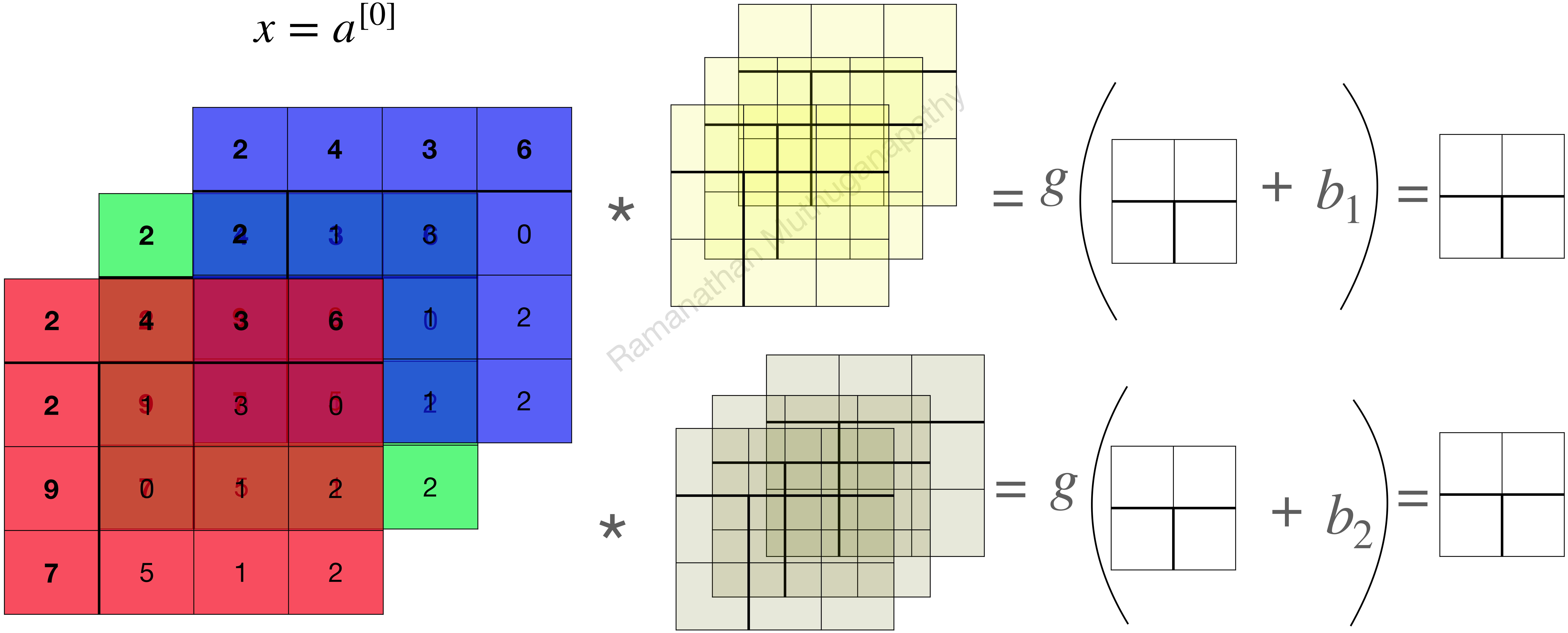
Convolution layer

Output

$$z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \qquad a^{[1]} = g(z^{[1]})$$

$\mathbf{W}^{[1]}$ $\qquad g(z^{[1]})$ $\qquad a^{[1]}$

$x = a^{[0]}$



Output size

n X n X 3

f X f X 3 X nof

n-f+1 X n-f+1 X nof

4 X 4 X 3

3 X 3 X 3 X nof

2 X 2 X nof

Key point in a convolution layer

- Weigths?
    - They are the values of the filters
- For a 3 X 3 X 3 filter, 27 + 1 parameters
- Total par = 28 * nof
- Input size is n X n X 3
- The number of parameters is independent of the input size!

Generalisation on a conv. layer l

Notation (Credit: Andrew Ng) -

$f^{[l]}$ - filter size
$p^{[l]}$ - padding
$s^{[l]}$ - stride

Input - $n_h^{[l-1]} \times n_w^{[l-1]} \times n_c^{[l-1]}$

Output - $n_h^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

$$n_h^{[l]} = \left\lfloor \frac{n_h^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$

$$n_w^{[l]} = \left\lfloor \frac{n_w^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$$

$n_c^{[l]}$ - num of filters
Each filter size is: $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Generalisation on a conv. layer l

Notation (Credit: Andrew Ng)

Activation: $a^{[l]} = n_h^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

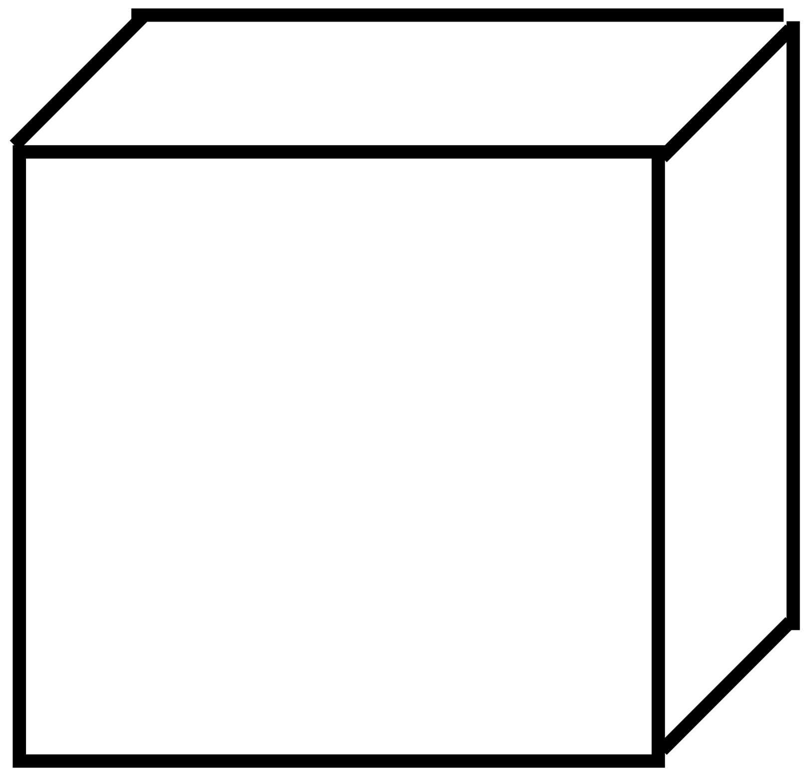
$A^{[l]} = m \times n_h^{[l]} \times n_w^{[l]} \times n_c^{[l]}$

Weights: $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

bias: $n_c^{[l]} - (1,1,1,n_c^{[l]})$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Example

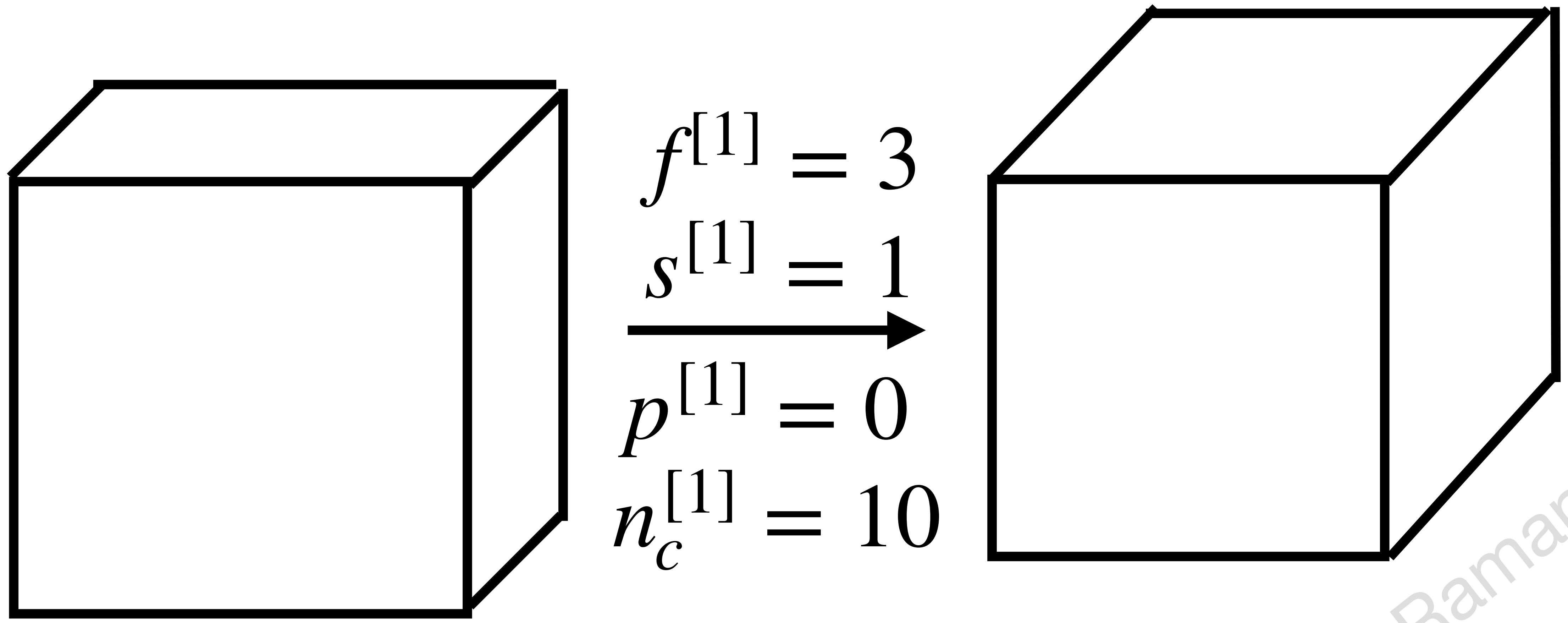


$100 \times 100 \times 3$

$n_h^{[0]} = n_w^{[0]} = 100$

$n_c^{[0]} = 3$

Example



$$f^{[1]} = 3$$
$$s^{[1]} = 1$$
$$p^{[1]} = 0$$
$$n_c^{[1]} = 10$$

$100 \times 100 \times 3$

$n_h^{[0]} = n_w^{[0]} = 100$

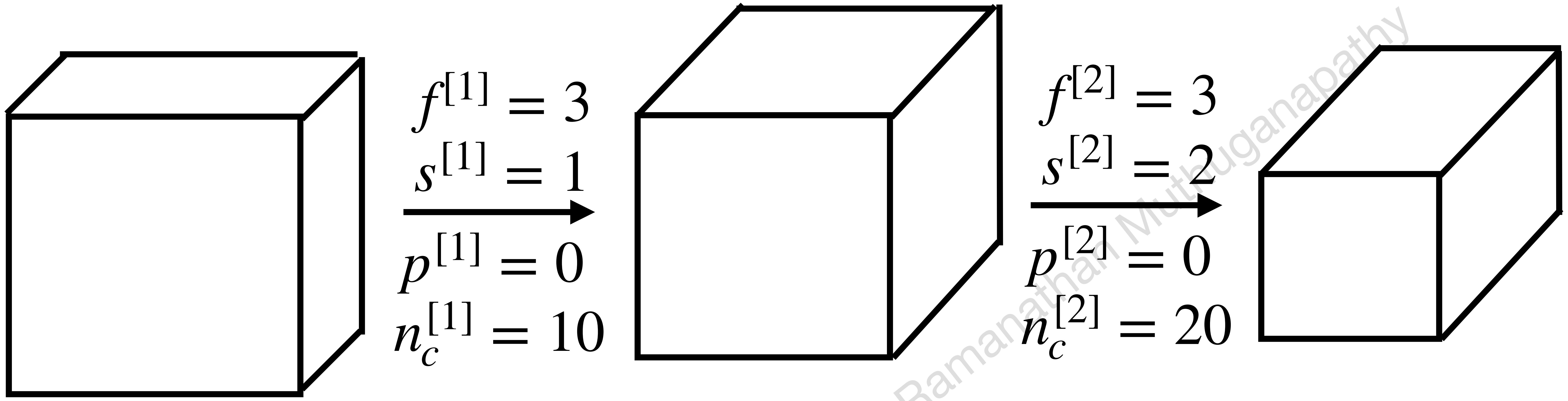
$n_c^{[0]} = 3$

$98 \times 98 \times 10$

$n_h^{[1]} = n_w^{[1]} = 98$

$n_c^{[1]} = 10$

Example



$100 \times 100 \times 3$

$n_h^{[0]} = n_w^{[0]} = 100$

$n_c^{[0]} = 3$

$98 \times 98 \times 10$

$n_h^{[1]} = n_w^{[1]} = 98$

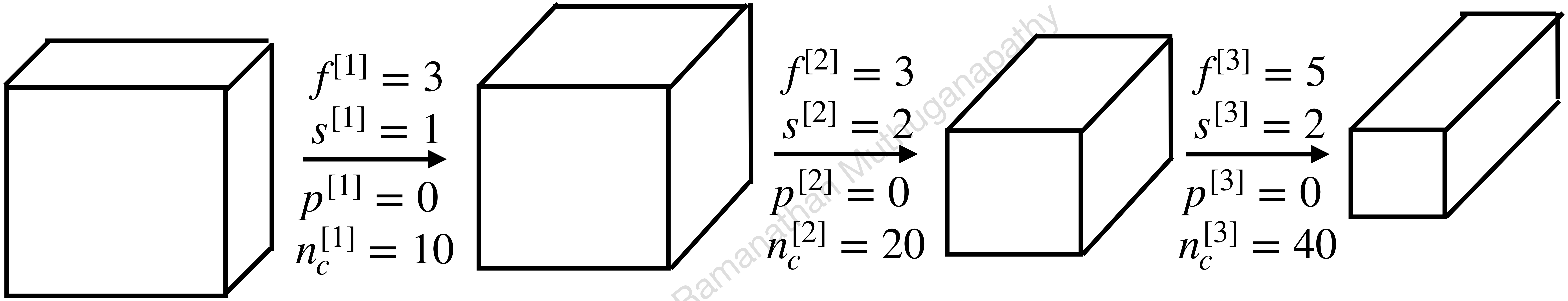
$n_c^{[1]} = 10$

$48 \times 48 \times 20$

$n_h^{[2]} = n_w^{[2]} = 48$

$n_c^{[2]} = 20$

Example



$100 \times 100 \times 3$

$n_h^{[0]} = n_w^{[0]} = 100$

$n_c^{[0]} = 3$

$98 \times 98 \times 10$

$n_h^{[1]} = n_w^{[1]} = 98$

$n_c^{[1]} = 10$

$48 \times 48 \times 20$

$n_h^{[2]} = n_w^{[2]} = 48$

$n_c^{[2]} = 20$

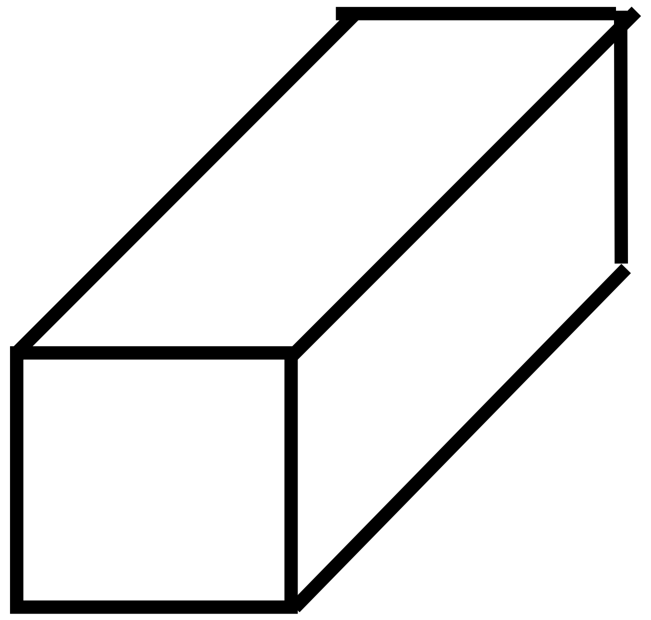
$22 \times 22 \times 40$

$n_h^{[3]} = n_w^{[3]} = 22$

$n_c^{[3]} = 40$

Example

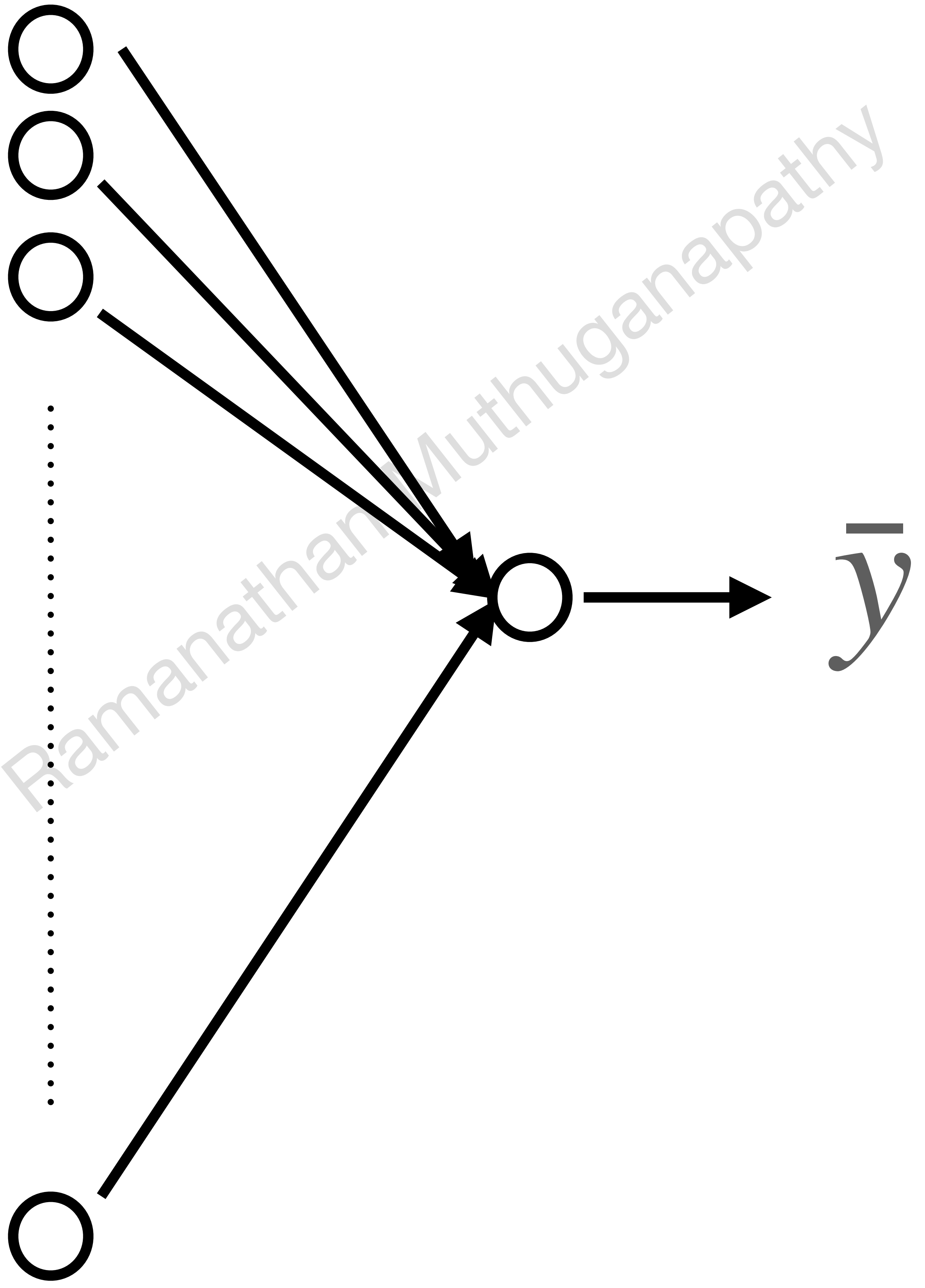
Flattening



$$22 \times 22 \times 40$$

$$n_h^{[3]} = n_w^{[3]} = 22$$

$$n_c^{[3]} = 40$$



$$19360 \times 1$$

Layers in CNN

- Convolution
- Pooling (Padding not used)
- FC (Fully connected)

Filter with size 2 X 2

Max. pooling

-5	**-2**	**-1**	**6**
-5	6	-3	7
-6	18	2	5
-5	14	1	2

Filter with size 2 X 2 and stride 1

Max. pooling

-5	**-2**	**-1**	**6**
-5	6	-3	7
-6	18	2	5
-5	14	1	2

6	**6**	**7**
18	18	7
18	18	5

$$\left\lfloor \frac{n_h - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_w - f}{s} + 1 \right\rfloor$$

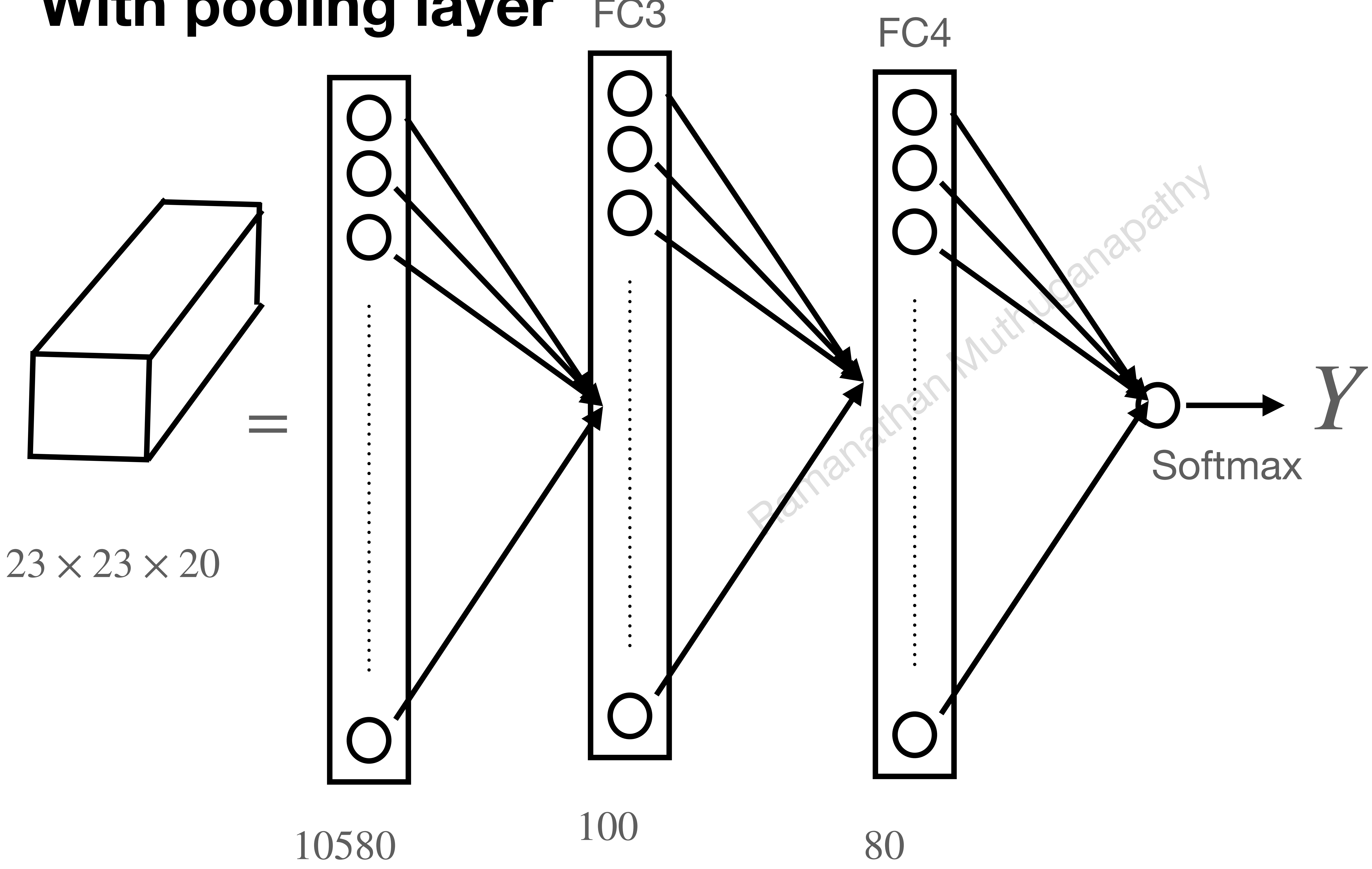
No Parameters to learn

Example

With pooling layer

Example

With pooling layer



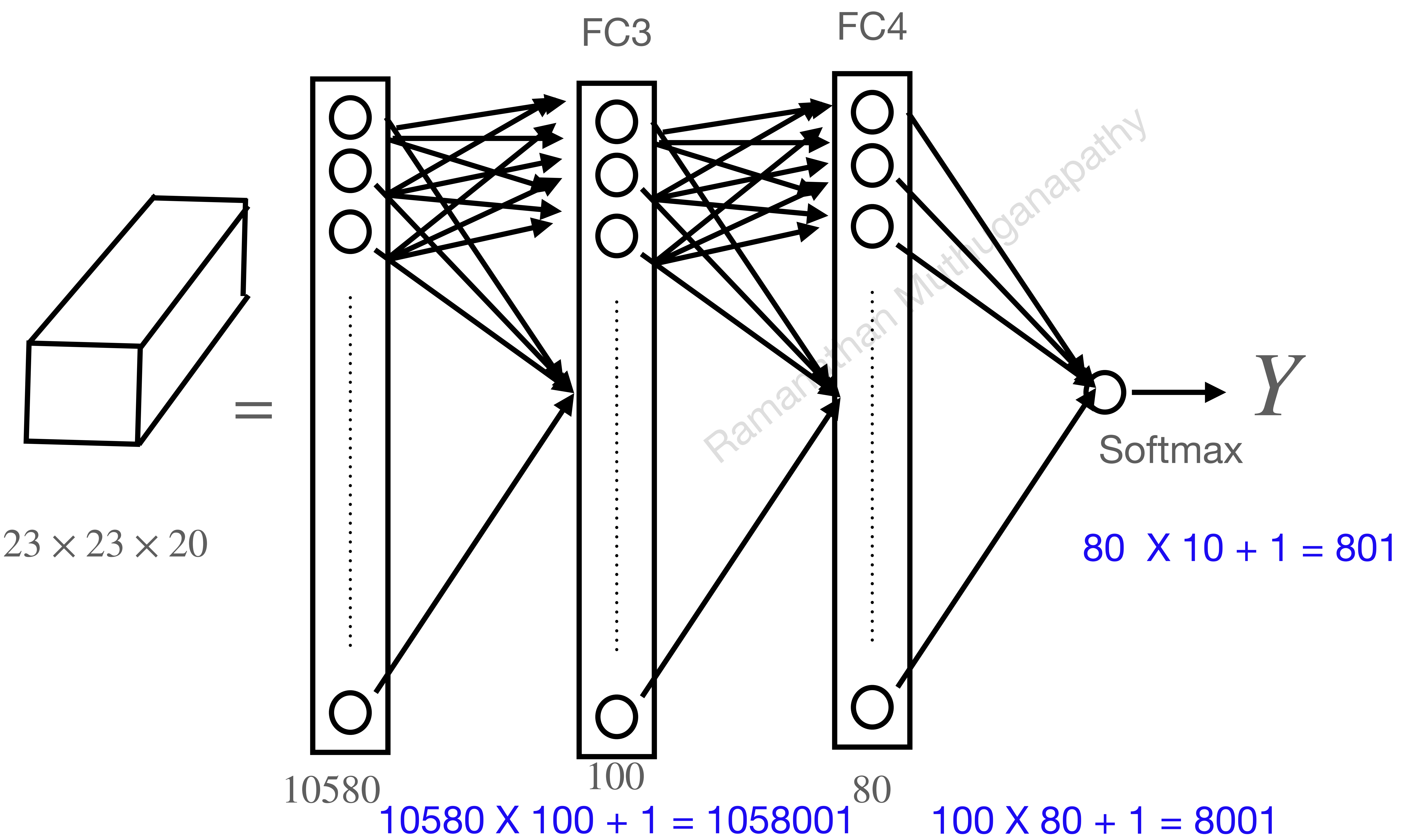
Example

Number of parameters

5 X 5 X 3 X 10 + 10 = 760   0   5 X 5 X 20 + 20 = 520   0

Example

Number of parameters



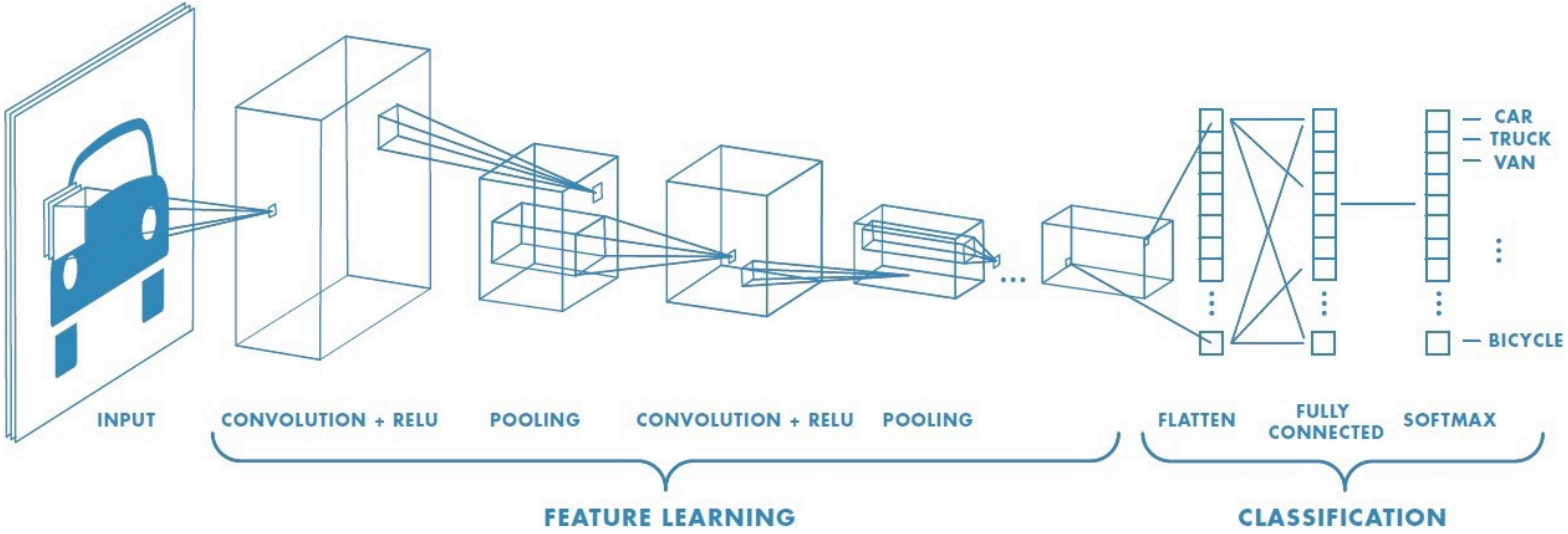
Outcomes of using convolution

- Convolution reduces the number of parameters
- Pooling layers don't have parameters
- Parameters are dominated by Fully-connected layers
- Reduced from billion pars. to million par. !!!

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

CNN

[towardsdatascience.com]()



Convolutions

- 2D convolution (that generates volumes)
- 3D convolution (Voxel-based, equivalent of pixels, but in 3D)
- 1D convolution (time-series data), not same as 1 X 1 convolution

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras