Convolution

Multi-channels input and multi-filters layer

Dr. Thanh-Sach LE
LTSACH@hcmut.edu.vn

GVLab:
Graphics and Vision Laboratory

Faculty of Computer Science and Engineering,
HCMUT

Contents

- ❖Multi-channels input
- ❖Multi-filters layer

Convolution

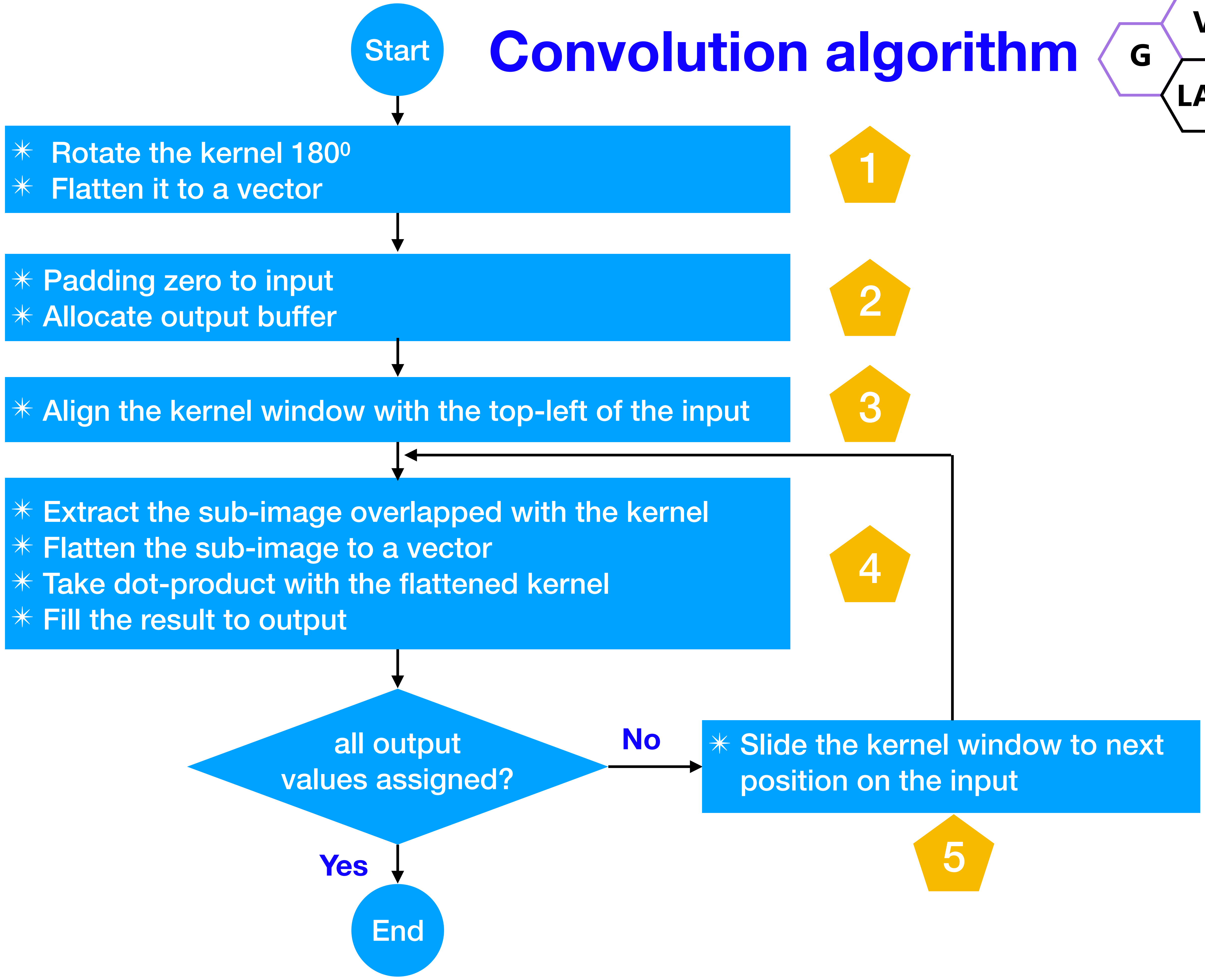
Multi-channels input

Dr. Thanh-Sach LE
LTSACH@hcmut.edu.vn

GVLab:
Graphics and Vision Laboratory

Faculty of Computer Science and Engineering,
HCMUT

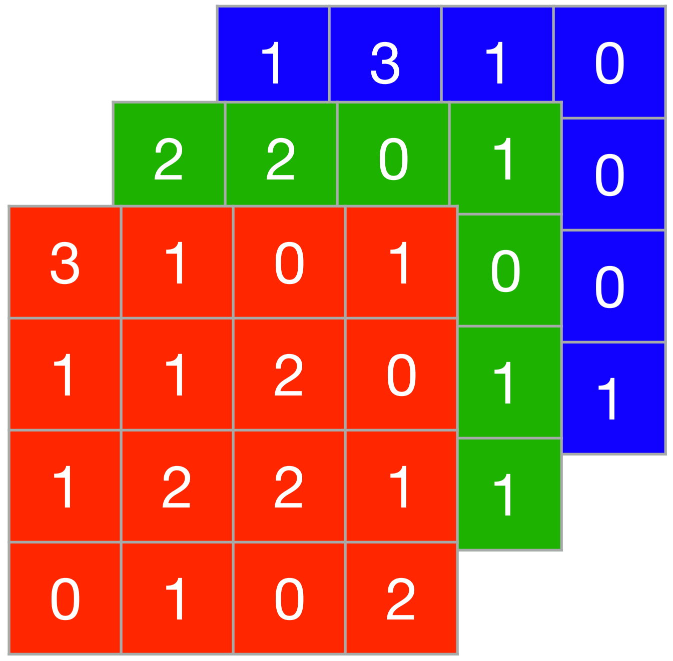
Convolution algorithm


Start
1
✳ Rotate the kernel 180^0
✳ Flatten it to a vector
2
✳ Padding zero to input
✳ Allocate output buffer
3
✳ Align the kernel window with the top-left of the input
4
✳ Extract the sub-image overlapped with the kernel
✳ Flatten the sub-image to a vector
✳ Take dot-product with the flattened kernel
✳ Fill the result to output
all output values assigned?
No
✳ Slide the kernel window to next position on the input
5
Yes
End

5

Multi-channels input

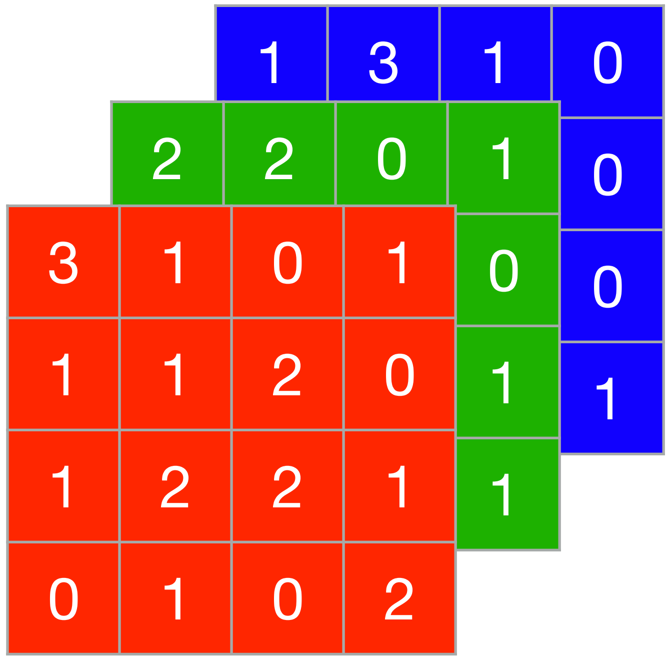
Input image or feature map: multiple channels



Input: 3 channels

Multi-channels input

Input image or feature map: multiple channels



Input: 3 channels

3	1	0	1
1	1	2	0
1	2	2	1
0	1	0	2

Channel 1 (RED)

2	2	0	1
1	1	1	0
1	0	0	1
1	1	0	1

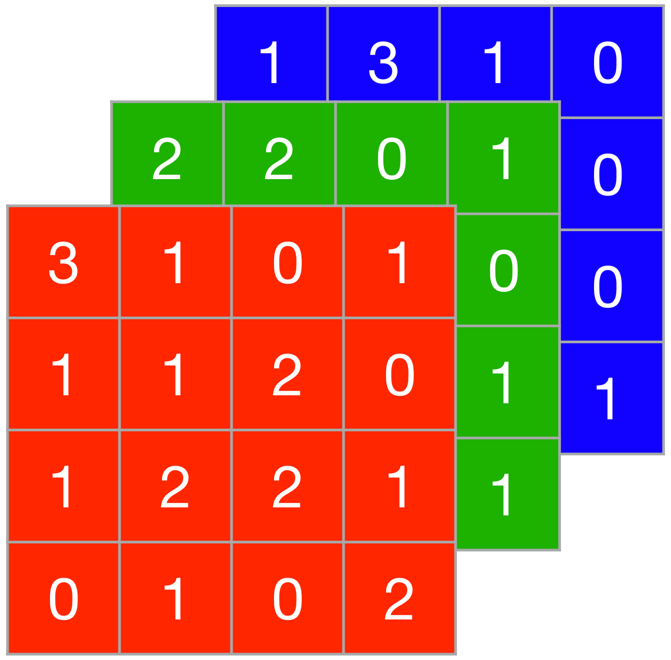
Channel 2 (GREEN)

1	3	1	0
1	1	2	0
2	1	0	0
2	0	0	1

Channel 3 (BLUE)

Multi-channels input

Input image or feature map: multiple channels



Input: 3 channels

3	1	0	1
1	1	2	0
1	2	2	1
0	1	0	2

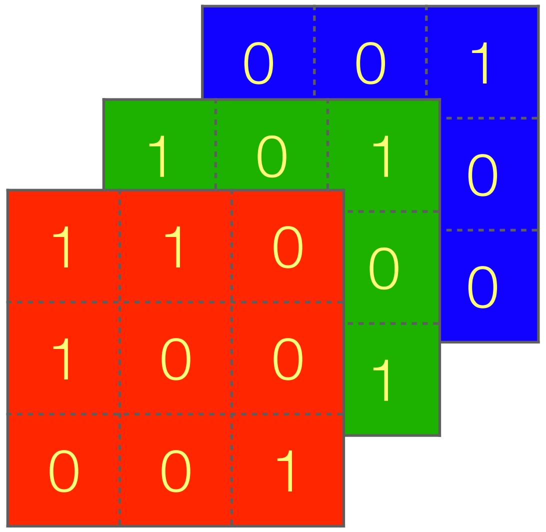
Channel 1 (RED)

2	2	0	1
1	1	1	0
1	0	0	1
1	1	0	1

Channel 2 (GREEN)

1	3	1	0
1	1	2	0
2	1	0	0
2	0	0	1

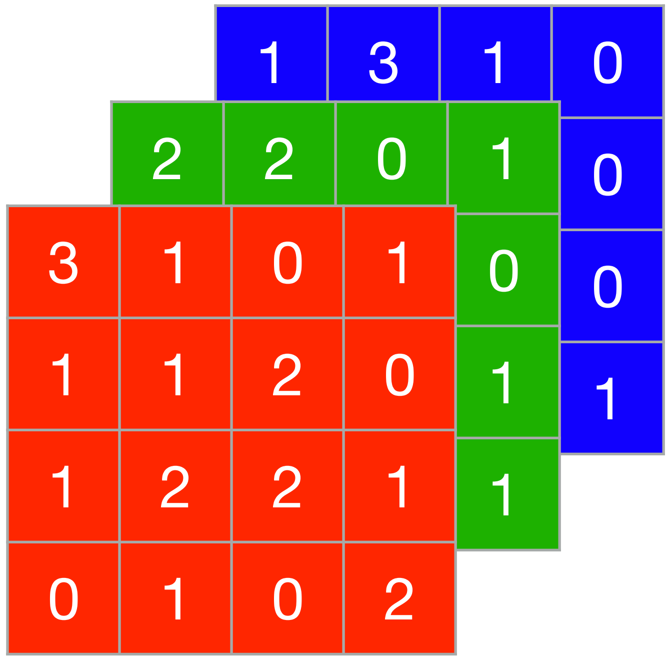
Channel 3 (BLUE)



Kernel: 3 channels

Multi-channels input

Input image or feature map: multiple channels



3	1	0	1
1	1	2	0
1	2	2	1
0	1	0	2

Channel 1 (RED)

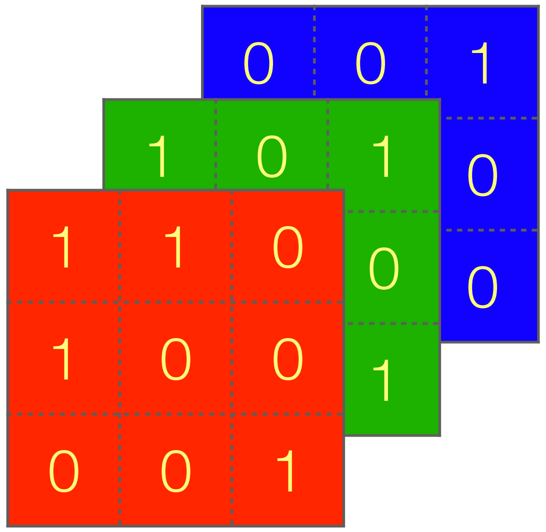
2	2	0	1
1	1	1	0
1	0	0	1
1	1	0	1

Channel 2 (GREEN)

1	3	1	0
1	1	2	0
2	1	0	0
2	0	0	1

Channel 3 (BLUE)

Input: 3 channels



1	1	0
1	0	0
0	0	1

Channel 1 (RED)

1	0	1
1	0	0
0	0	1

Channel 2 (GREEN)

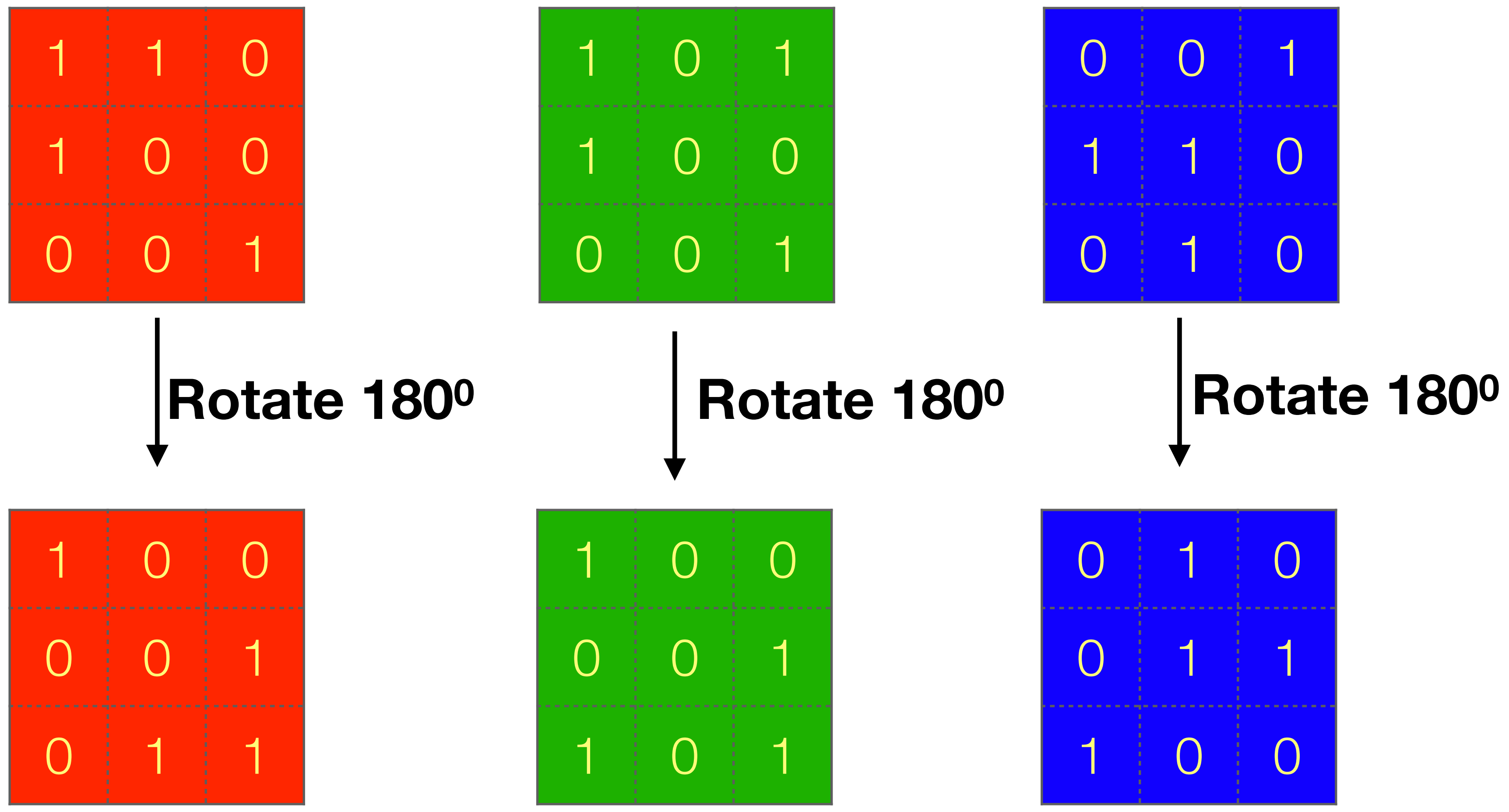
0	0	1
1	1	0
0	1	0

Channel 3 (BLUE)

Kernel: 3 channels

Multi-channels input

1 Rotate kernel 180^0



$$\begin{bmatrix}1&1&0\\1&0&0\\0&0&1\end{bmatrix} \xrightarrow{\text{Rotate }180^0} \begin{bmatrix}1&0&0\\0&0&1\\0&1&1\end{bmatrix}$$

$$\begin{bmatrix}1&0&1\\1&0&0\\0&0&1\end{bmatrix} \xrightarrow{\text{Rotate }180^0} \begin{bmatrix}1&0&0\\0&0&1\\1&0&1\end{bmatrix}$$

$$\begin{bmatrix}0&0&1\\1&1&0\\0&1&0\end{bmatrix} \xrightarrow{\text{Rotate }180^0} \begin{bmatrix}0&1&0\\0&1&1\\1&0&0\end{bmatrix}$$

Multi-channels input

1 Flatten the rotated kernel to a vector

$$\begin{bmatrix}1&1&0\\1&0&0\\0&0&1\end{bmatrix}\xrightarrow{\text{Rotate }180^0}\begin{bmatrix}1&0&0\\0&0&1\\0&1&1\end{bmatrix}$$

$$\begin{bmatrix}1&0&1\\1&0&0\\0&0&1\end{bmatrix}\xrightarrow{\text{Rotate }180^0}\begin{bmatrix}1&0&0\\0&0&1\\1&0&1\end{bmatrix}$$

$$\begin{bmatrix}0&0&1\\1&1&0\\0&1&0\end{bmatrix}\xrightarrow{\text{Rotate }180^0}\begin{bmatrix}0&1&0\\0&1&1\\1&0&0\end{bmatrix}$$

After flattening:

1	0	0	0	0	1	0	1	1	1	0	0	0	0	1	1	0	1	0	1	0	0	1	1	1	0	0

Multi-channels input

2 Padding the input

3	1	0	1
1	1	2	0
1	2	2	1
0	1	0	2

Channel 1 (RED)

2	2	0	1
1	1	1	0
1	0	0	1
1	1	0	1

Channel 2 (GREEN)

1	3	1	0
1	1	2	0
2	1	0	0
2	0	0	1

Channel 3 (BLUE)

(No padding)

Multi-channels input

2 Allocate the output buffer

3	1	0	1
1	1	2	0
1	2	2	1
0	1	0	2

Channel 1 (RED)

2	2	0	1
1	1	1	0
1	0	0	1
1	1	0	1

Channel 2 (GREEN)

1	3	1	0
1	1	2	0
2	1	0	0
2	0	0	1

Channel 3 (BLUE)

input: $i_1 = i_2 = 4$

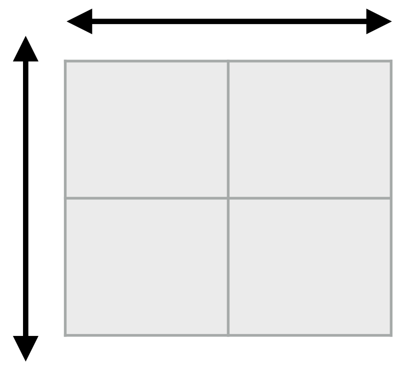
kernel: $k_1 = k_2 = 3$

padding: $p_1 = p_2 = 0$

strides: $s_1 = s_2 = 1$

$i_1 - k_1 + 1 = 2$

$i_2 - k_2 + 1 = 2$



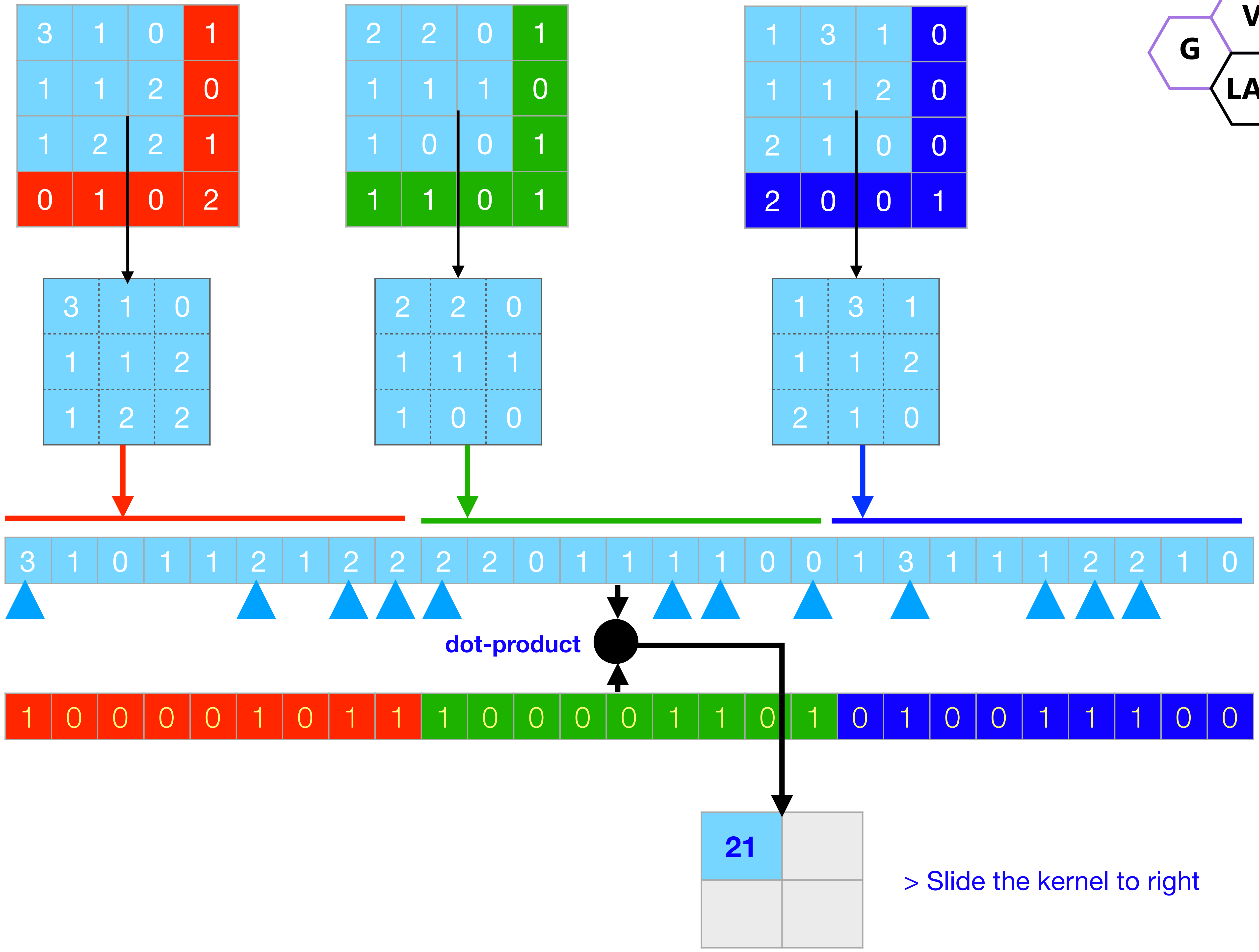
Output

3	1	0	1
1	1	2	0
1	2	2	1
0	1	0	2

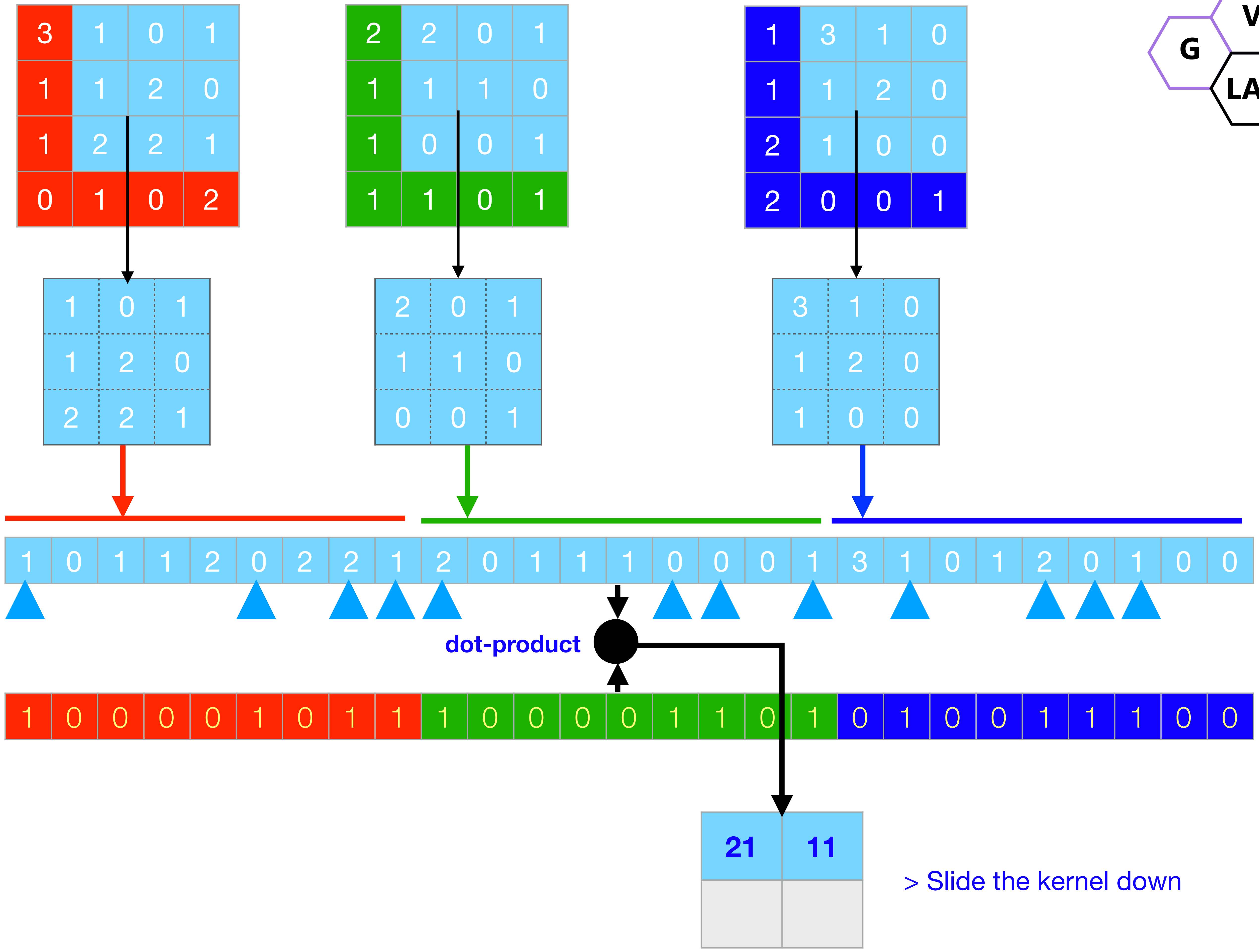
2	2	0	1
1	1	1	0
1	0	0	1
1	1	0	1

1	3	1	0
1	1	2	0
2	1	0	0
2	0	0	1

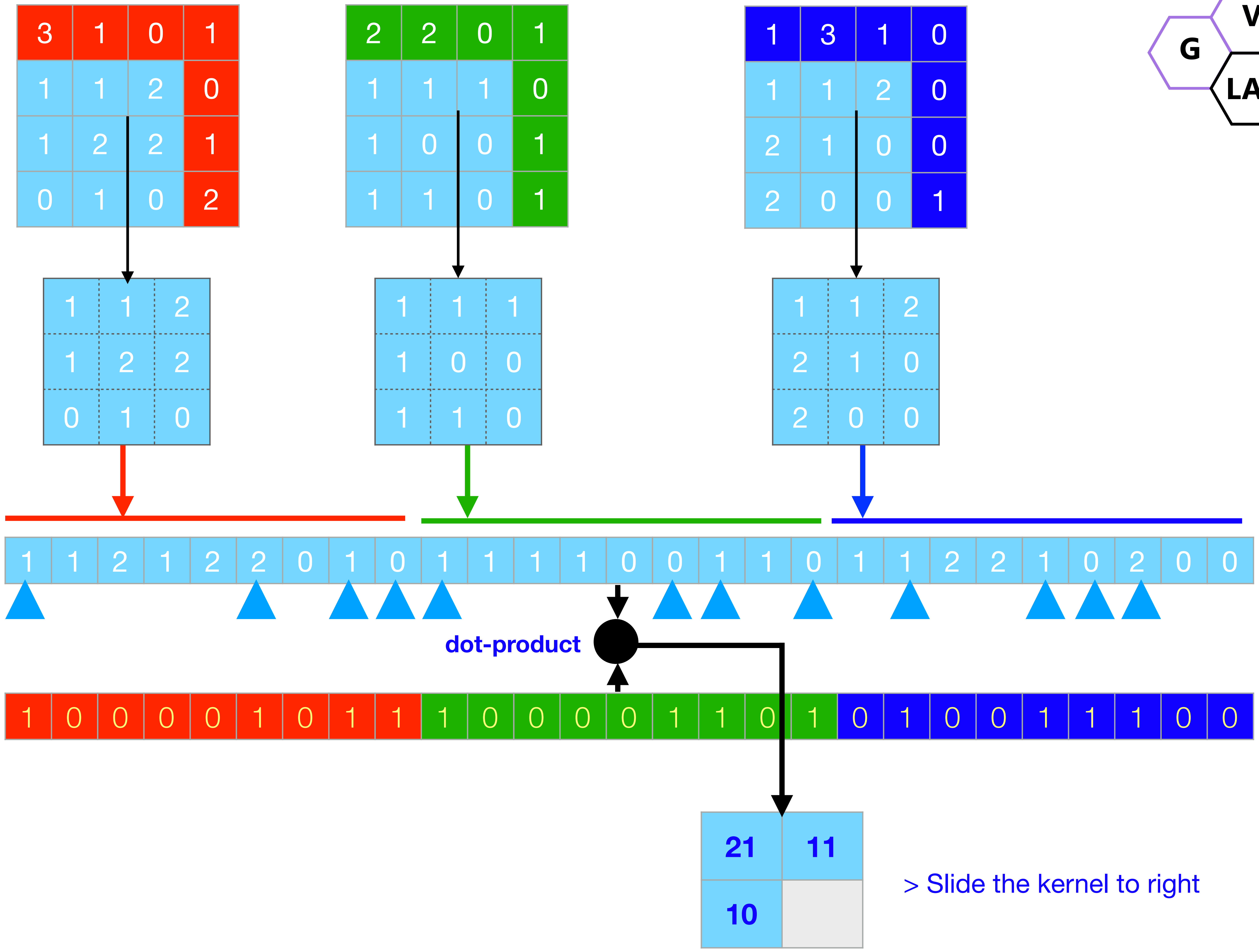
3 Starting the cross-correlation process



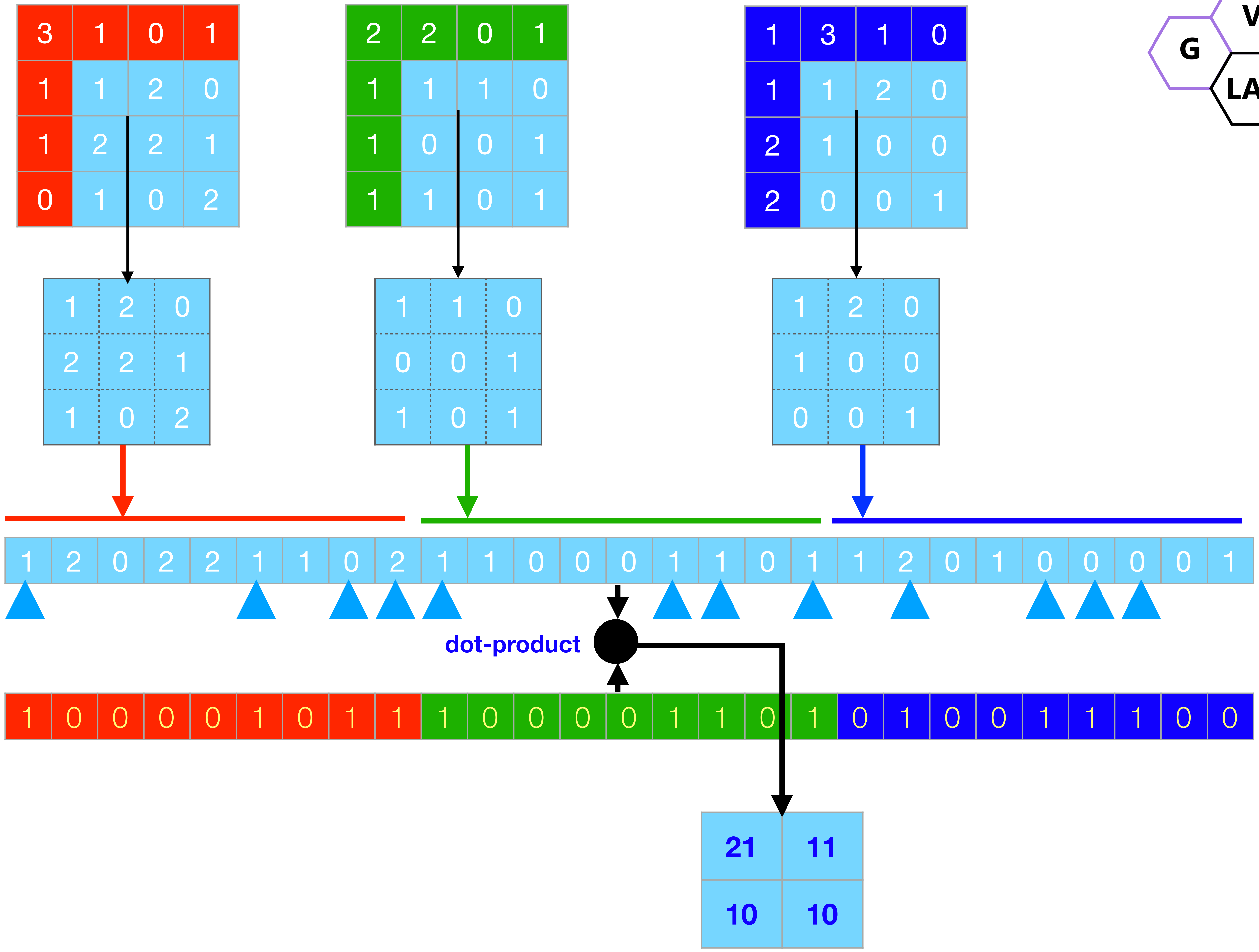
> Slide the kernel to right



> Slide the kernel down

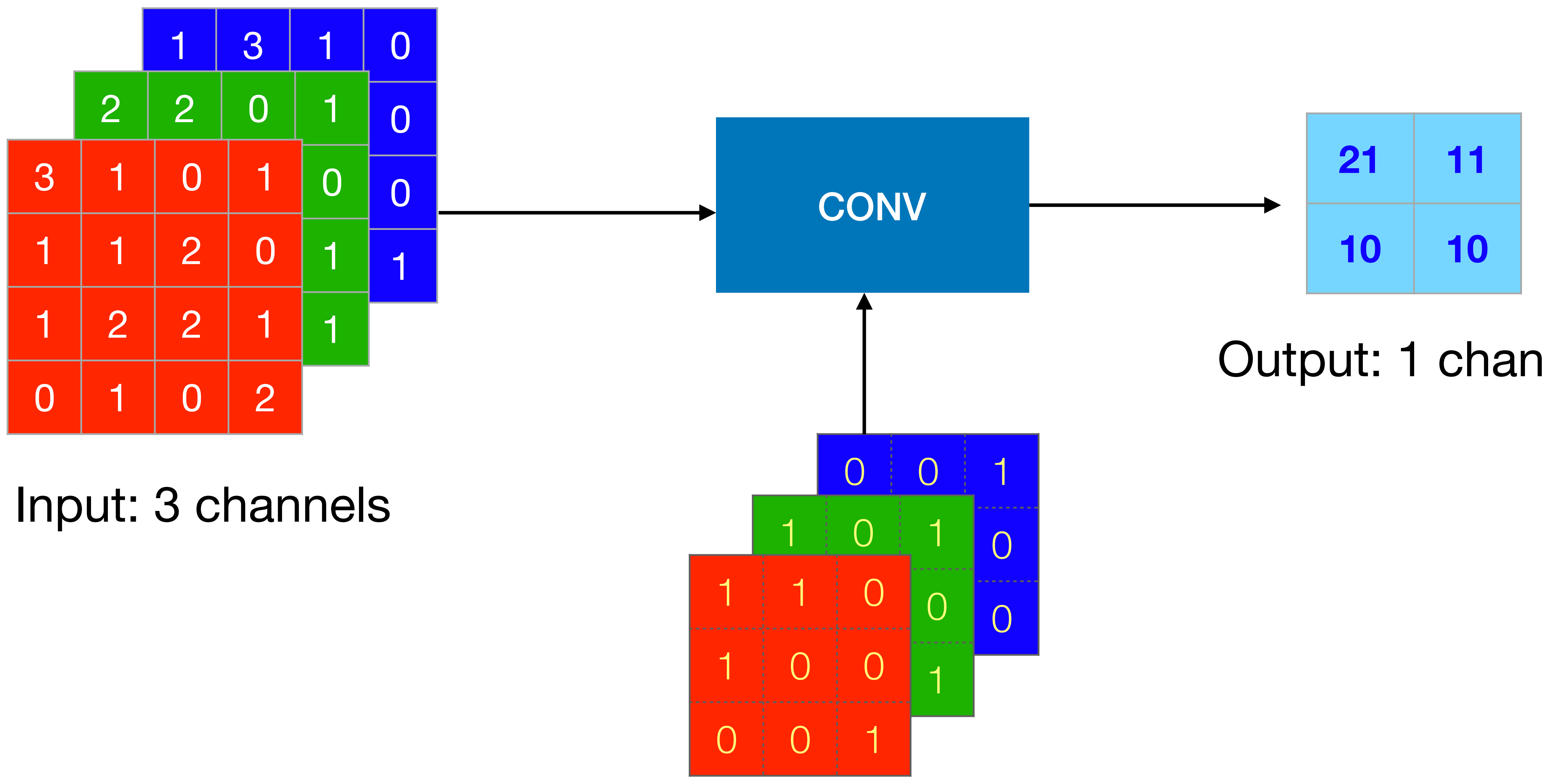


> Slide the kernel to right


dot-product
21
11
10
10

Multi-channels input

Final result



Input: 3 channels

Output: 1 channel

Kernel: 3 channels

Convolution

Multi-filters layer

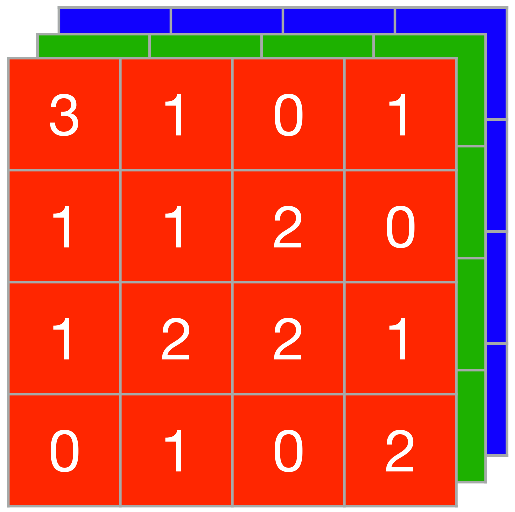
Dr. Thanh-Sach LE
LTSACH@hcmut.edu.vn

GVLab:
Graphics and Vision Laboratory

Faculty of Computer Science and Engineering,
HCMUT

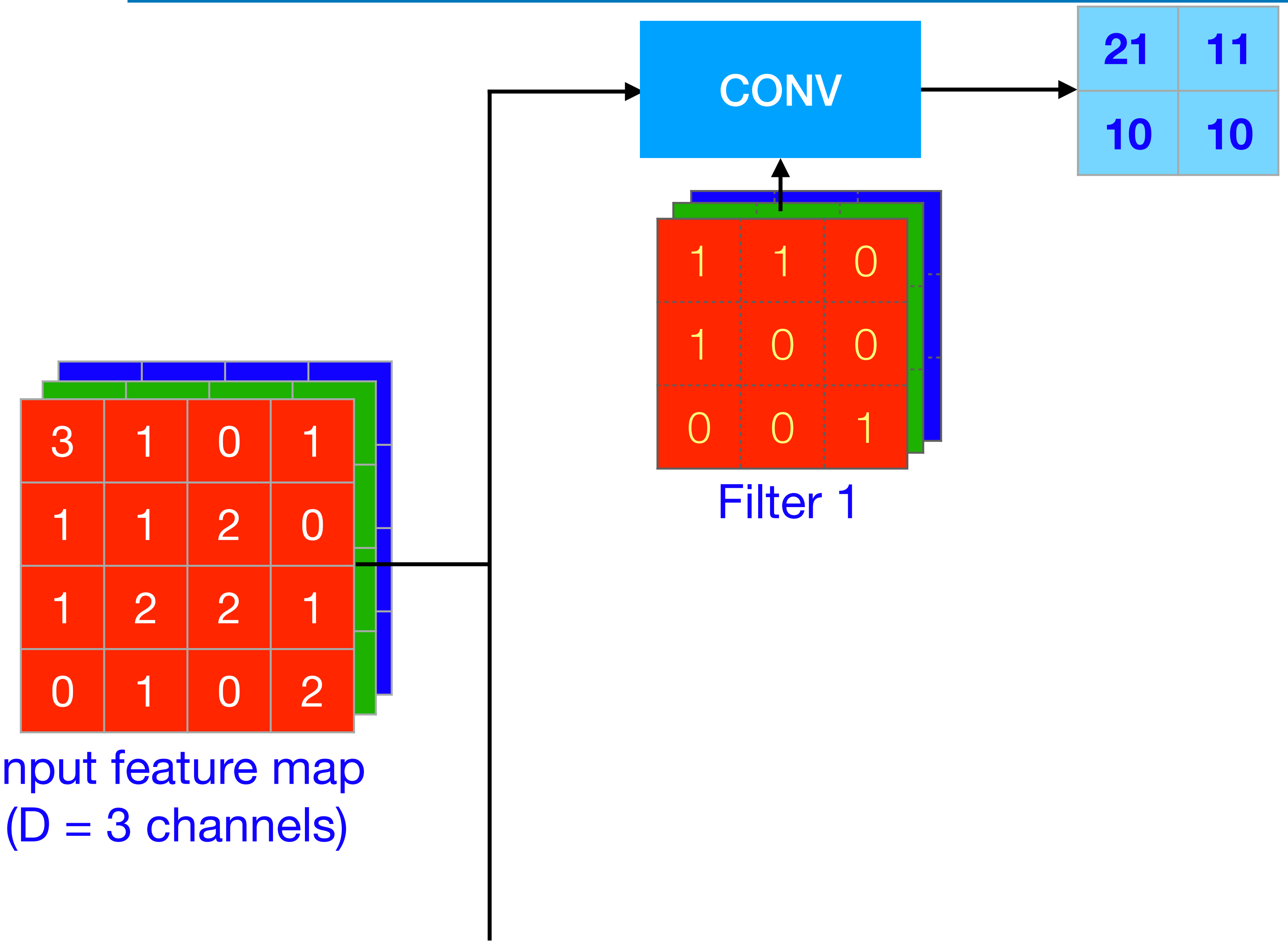
Multi-filters layer

- Input: a feature map of D channels
- Convolution layer in Deep learning frameworks:
  - Consists of multiple filters:
    - Each the filters' kernel has D channels (#channels of the input)
    - All the filters' kernel are the same size, e.g., 3x3
- How the convolution layer computed?

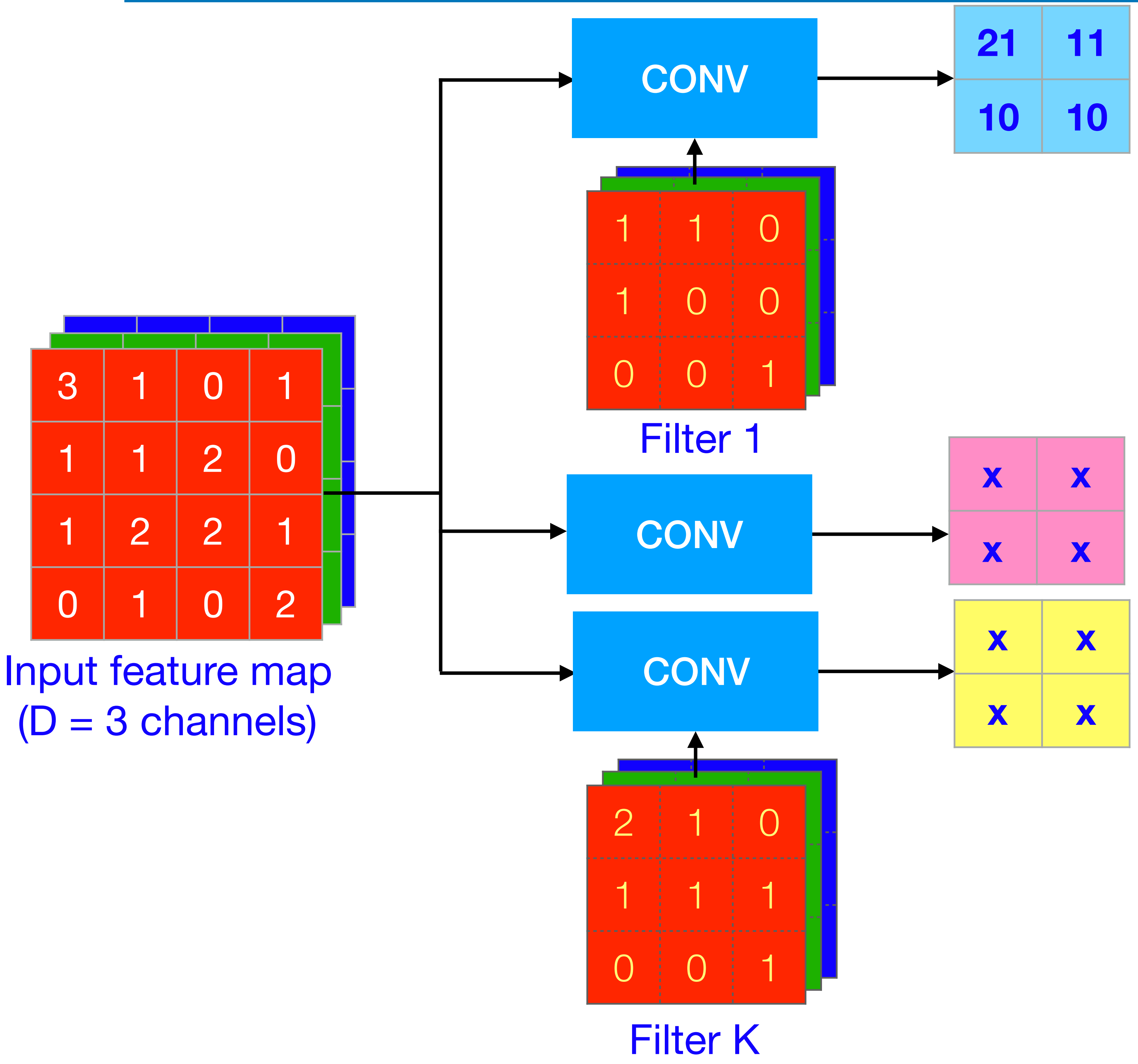


Input feature map (D = 3 channels)

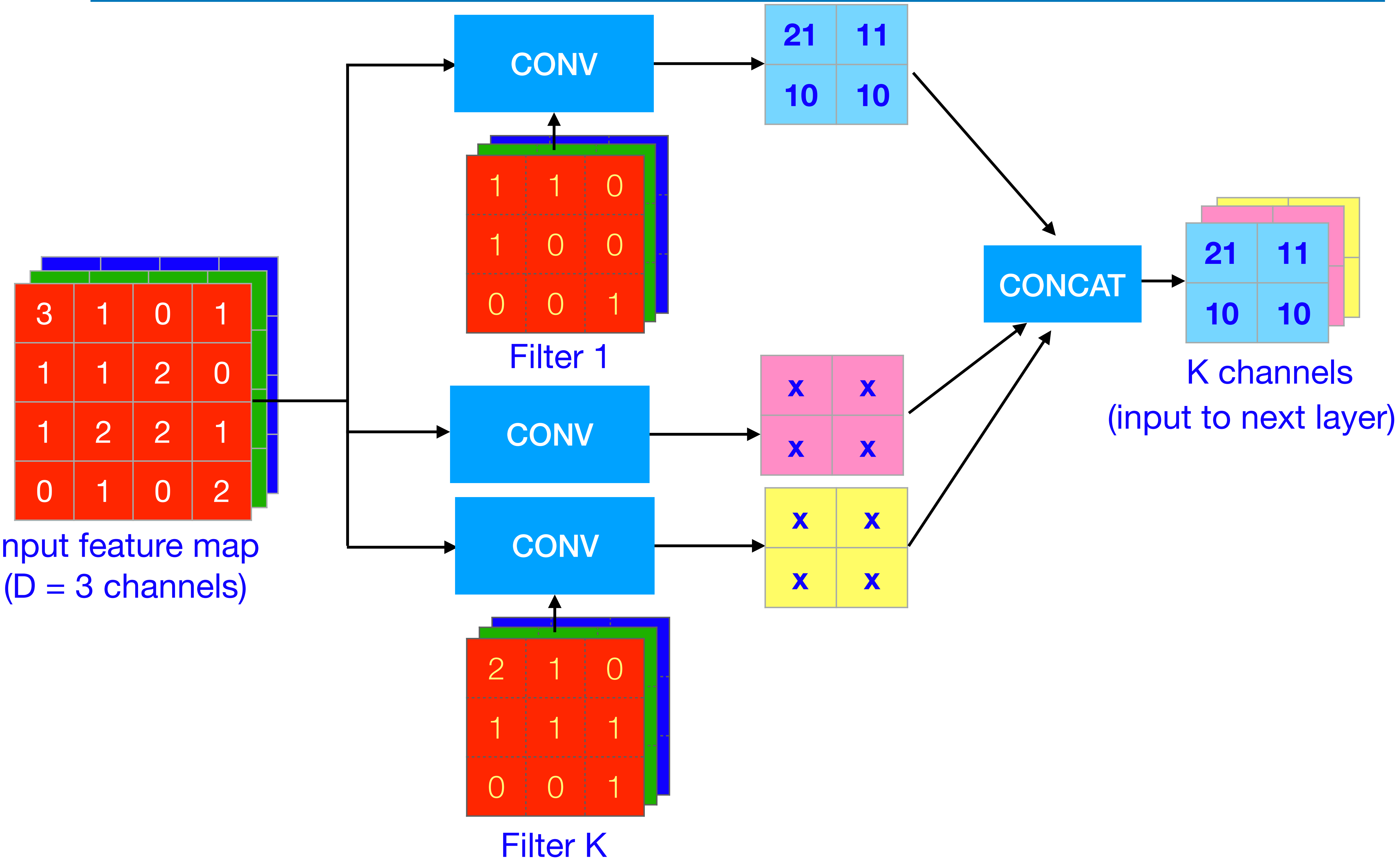
Multi-filters layer



Multi-filters layer



Multi-filters layer


CONV
21 11
10 10
1 1 0
1 0 0
0 0 1
Filter 1
CONV
x x
x x
CONV
x x
x x
2 1 0
1 1 1
0 0 1
Filter K
3 1 0 1
1 1 2 0
1 2 2 1
0 1 0 2
Input feature map
(D = 3 channels)
CONCAT
21 11
10 10
K channels
(input to next layer)