INF721

2024/2

UFV

Deep Learning

L10: Convolutional Neural Networks

Logistics

Announcements

- FP1: Project Proposal deadline has been extended to Oct. 18
- Please fill out the evaluation form:
  https://forms.gle/2g3fXBymVtvh2ij3A

Last Lecture

- Mini-batch Gradient Descent
- Gradient Descent with Momentum
- RMSProp
- Adam

Lecture Outline

- Parameter explosion
- Filters (kernels)
- Convolutions
  - Padding
  - Strided Convolutions
- Convolutions Over Volumes
- Padding Layers
- Convolutional Neural Networks

Parameter Explosion

To process images with MLPs, we have to transform them into feature vectors:

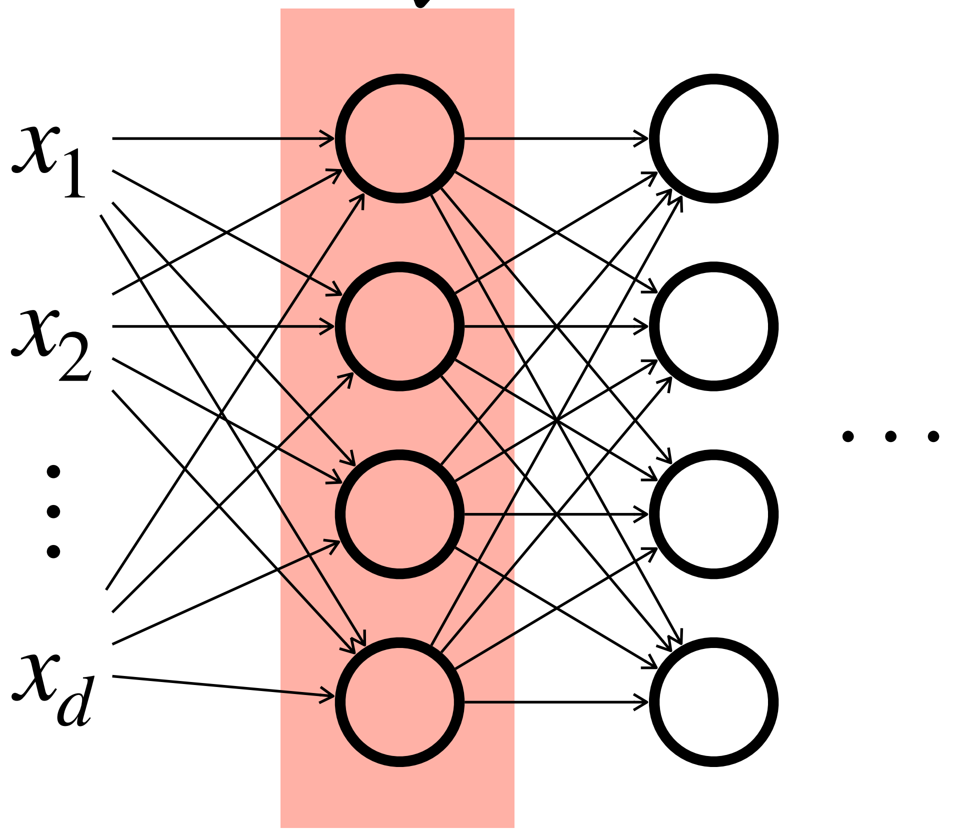
1000 h

w
1000

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

$n^{[1]} = 1000$

$W^{[1]} = (n^{[1]}, n^{[0]}) = (1000,\ 3M)$



$d = h \times w \times 3$

$d = 1000 \times 1000 \times 3 = 3M$

3 billion parameters! 😱

Problems:

- Computational resources (Memory and processing)
- We need lots of data to avoid overfit

UFV

Convolutions

We can use convolutions to process large images with a constant number of parameters.

Convolutions are operations to apply filters (i.e, transformations) to images:

Blur

(Filter)

UFV

Filters

A **filter** (or kernel) is a small matrix (typically 3x3) of weights used to transform a **pixel** by the weighted sum of its neighbours.

	206	205	247	
	144	161	137	
	192	154	75	

Original pixel (161)
and its neighbours

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

$206 * 0.0625 + 205 * 0.125 + 247 * 0.0625 +$

$144 * 0.125 + 161 * 0.25 + 137 * 0.125 +$

$192 * 0.0625 + 154 * 0.125 + 75 * 0.0625 =$

178

	206	205	247	
	144	178	137	
	192	154	75	

Transformed pixel (178)
and its neighbours

Convolutions

In image processing and computer vision, a **convolution** consists of applying a filter to each pixel of an image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

Transformed Image
(3×3)

Convolutions

In image processing and computer vision, a **convolution** consists of applying a filter to each pixel of an image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image

(5×5)

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

178		

Transformed Image

(3×3)

Convolutions

In image processing and computer vision, a **convolution** consists of applying a filter to each pixel of an image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

178	175	

Transformed Image
(3×3)

UFV

Convolutions

In image processing and computer vision, a **convolution** consists of applying a filter to each pixel of an image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

178	175	216

Transformed Image
(3×3)

Convolutions

In image processing and computer vision, a **convolution** consists of applying a filter to each pixel of an image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

178	175	216
141		

Transformed Image
(3×3)

Convolutions

In image processing and computer vision, a **convolution** consists of applying a filter to each pixel of an image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image (5×5)

$*$

0,0625	0,125	0,0625
0,125	0,25	0,125
0,0625	0,0125	0,0625

Filter (blur)

$$= \sum_{i=1}^{3} \sum_{j=1}^{3} = m_{i,j} * k_{i,j} =$$

178	175	216
141	133	183
106	117	167

Transformed Image (3×3)

Edge Detection

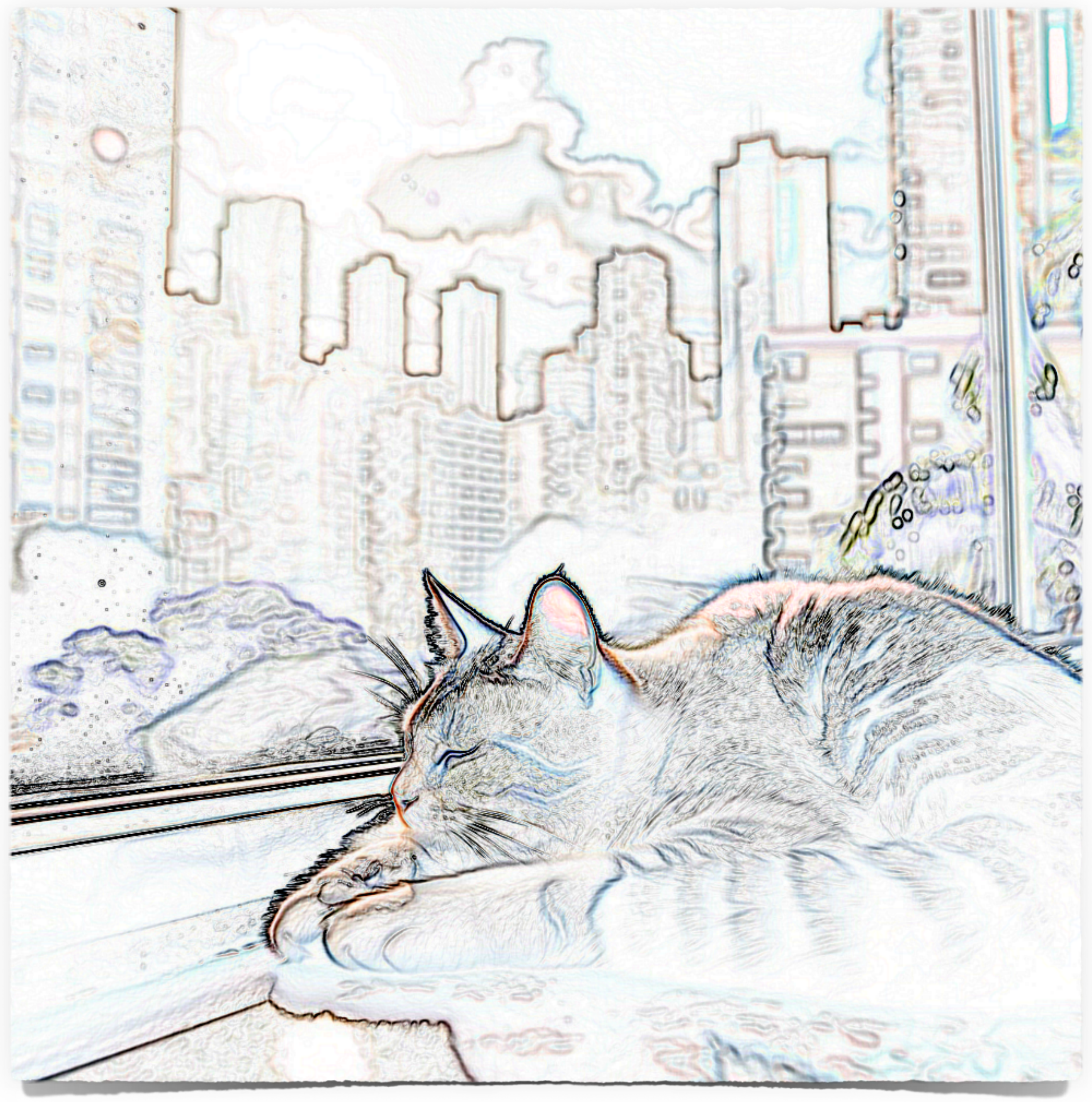
Convolutions can be used to detect edges in images, which is particularly important for feature extraction.

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

Vertical

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

Horizontal



UFV

Manually Designing Filters

Diferent filters for border detection have been developed scientifically by the reseach community in image processing.

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

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

Sobel

3	0	-3
10	0	-10
3	0	-3

Scharr

UFV

https://setosa.io/ev/image-kernels/

Learning Filters

Convolutional Neural Networks (CNNs) **learn filters** from images with a loss function and gradient descent.

*

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

The weights of a CNN are organized in convoluion filters

Convolutions reduce the size of an image

- Consecutive convolutions can make the image very small (e.g., 1x1)
- Corner pixels are less shared among convolution steps than the pixels in the middle

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

Filter
(3×3)

$=$

178	175	216
141	133	183
106	117	167

Transformed Image
(3×3)

General rule

$$(n \times n) * (f \times f) = (n - f + 1 \times n - f + 1)$$

UFV

Padding

Padding consists of adding a border with p pixels to the original image:

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5 x 5)

$*$

Filter
(3 x 3)

$=$

Transformed Image
(5 x 5)

General Rule with Padding

$$(n \times n) * (f \times f) = (n + 2p - f + 1 \times n + 2p - f + 1)$$

Padding

To find the value of p that keeps the size of an $n \times n$ image after a convolution with a filter of size f (odd), one can solve the following equation:

$$
\begin{aligned}
n + 2p - f + 1 &= n \\
2p - f + 1 &= 0 \\
2p &= f - 1 \\
p &= \frac{f-1}{2}
\end{aligned}
$$

Strided Convolutions

Strided convolutions slide the filter more than one step at a time.

stride = 2

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

Filter
(3×3)

$=$

UFV

Strided Convolutions

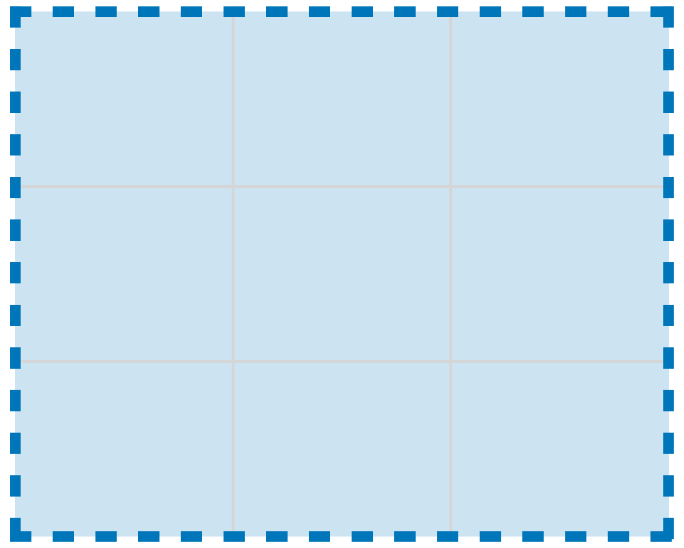
Strided convolutions slide the filter more than one step at a time.

stride = 2

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$



Filter
(3×3)

$=$

Strided Convolutions

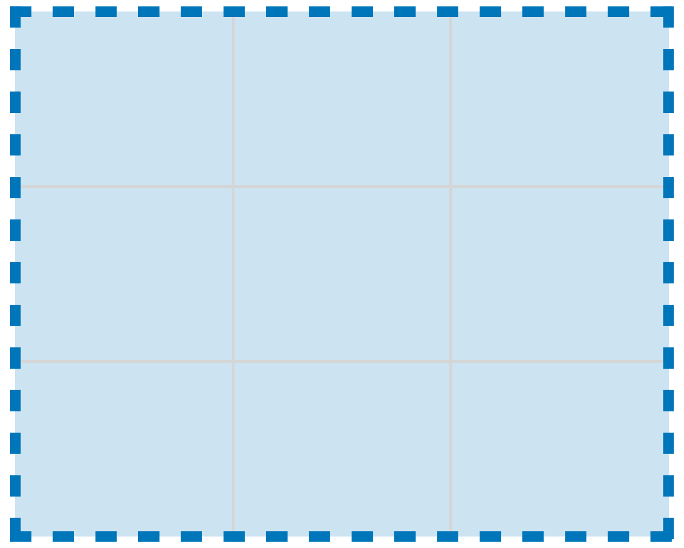
Strided convolutions slide the filter more than one step at a time.

stride = 2

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$



Filter
(3×3)

$=$

UFV

Strided Convolutions

Strided convolutions slide the filter more than one step at a time.

stride = 2

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image

(5×5)

$*$

Filter

(3×3)

$=$

UFV

Strided Convolutions

Strided convolutions slide the filter more than one step at a time.

stride = 2

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$

Filter
(3×3)

$=$

Strided Convolutions

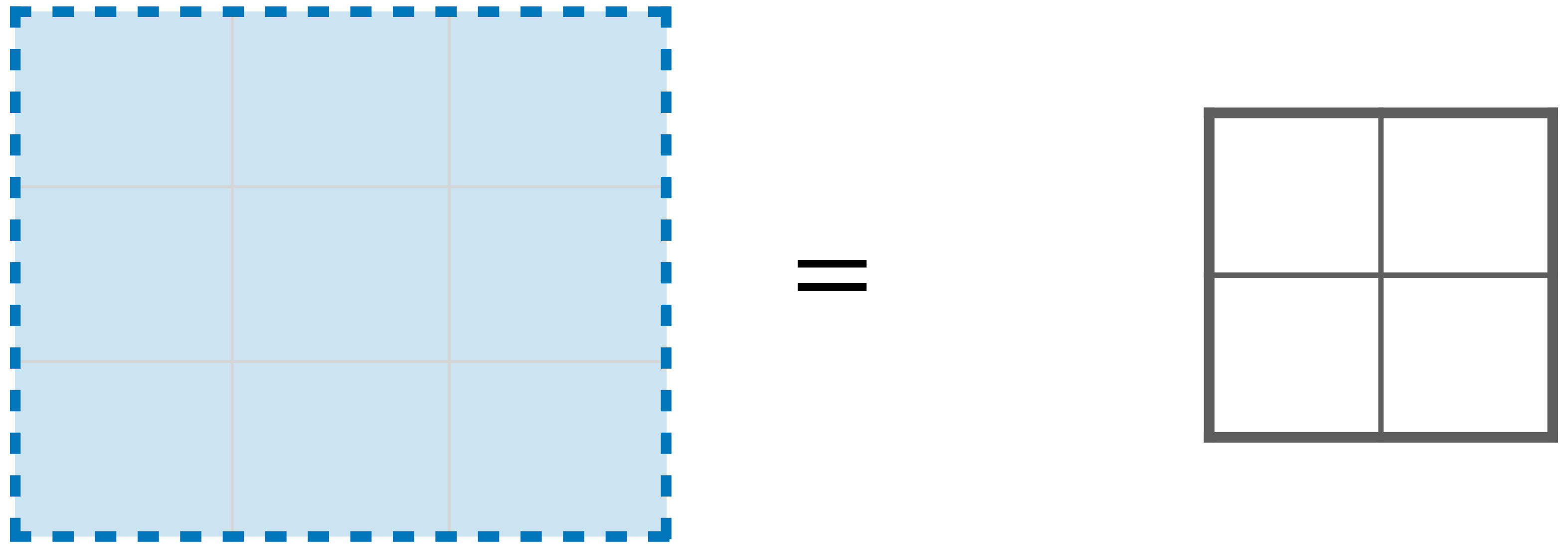
Strided convolutions slide the filter more than one step at a time.

stride = 2 → The stride size is the number of steps used to slide the filter during the convolution.

206	205	247	245	244
244	161	137	244	254
192	154	75	200	249
90	109	96	143	223
67	69	107	196	236

Original Image
(5×5)

$*$



Filter
(3×3)

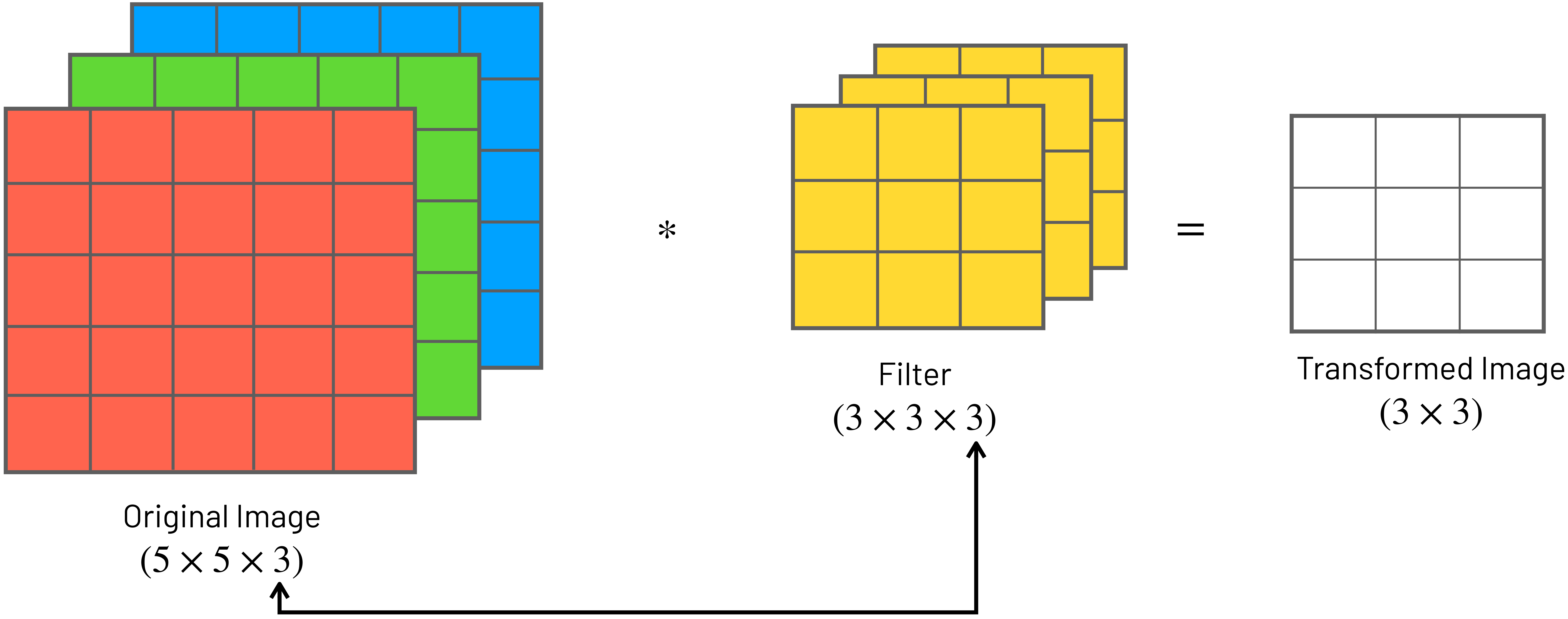
Transformed Image
(2×2)

General Rule with Padding and Stride

$$(n \times n) * (f \times f) = \left(\frac{n+2p-f}{s} + 1 \times \frac{n+2p-f}{s} + 1\right)$$

Convolutions Over Volumes

Convolutions in colored images (R,G,B) need filters with 3 channels:

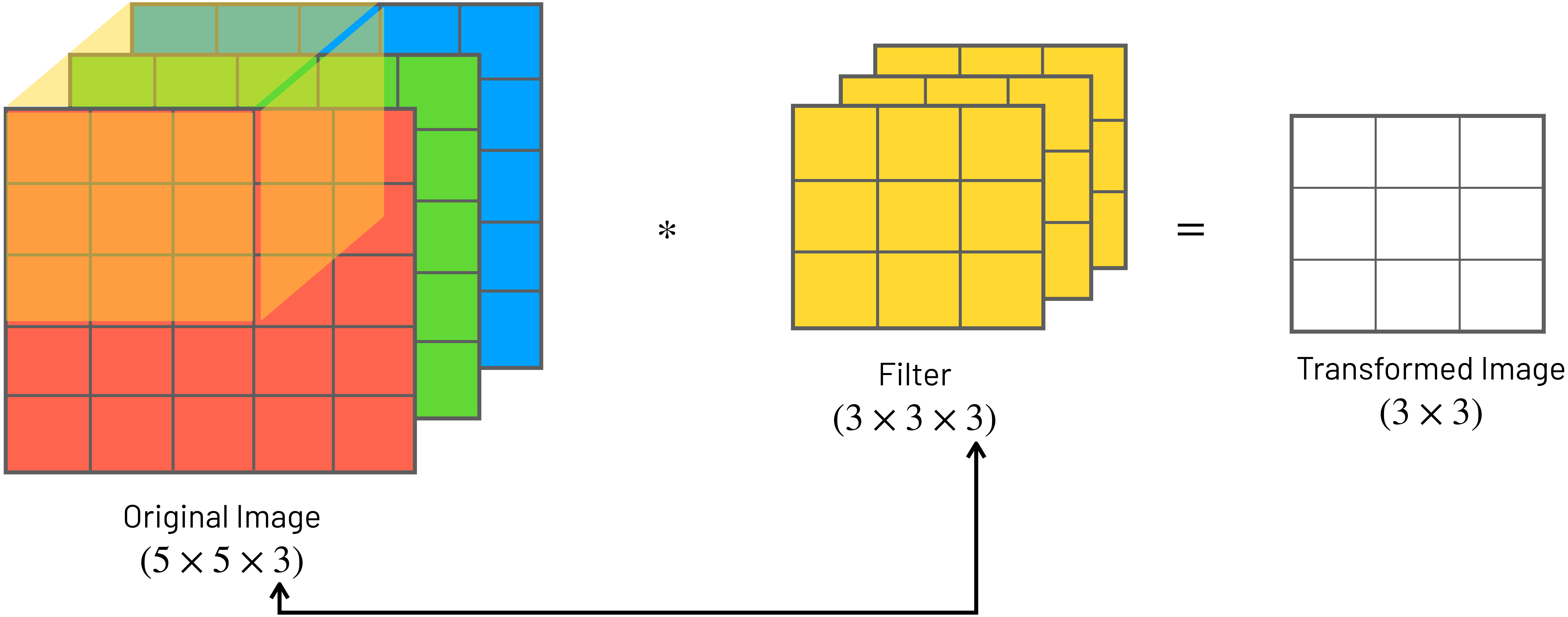


The number of channels in the filter must be the same as in the image!

UFV

Convolutions Over Volumes

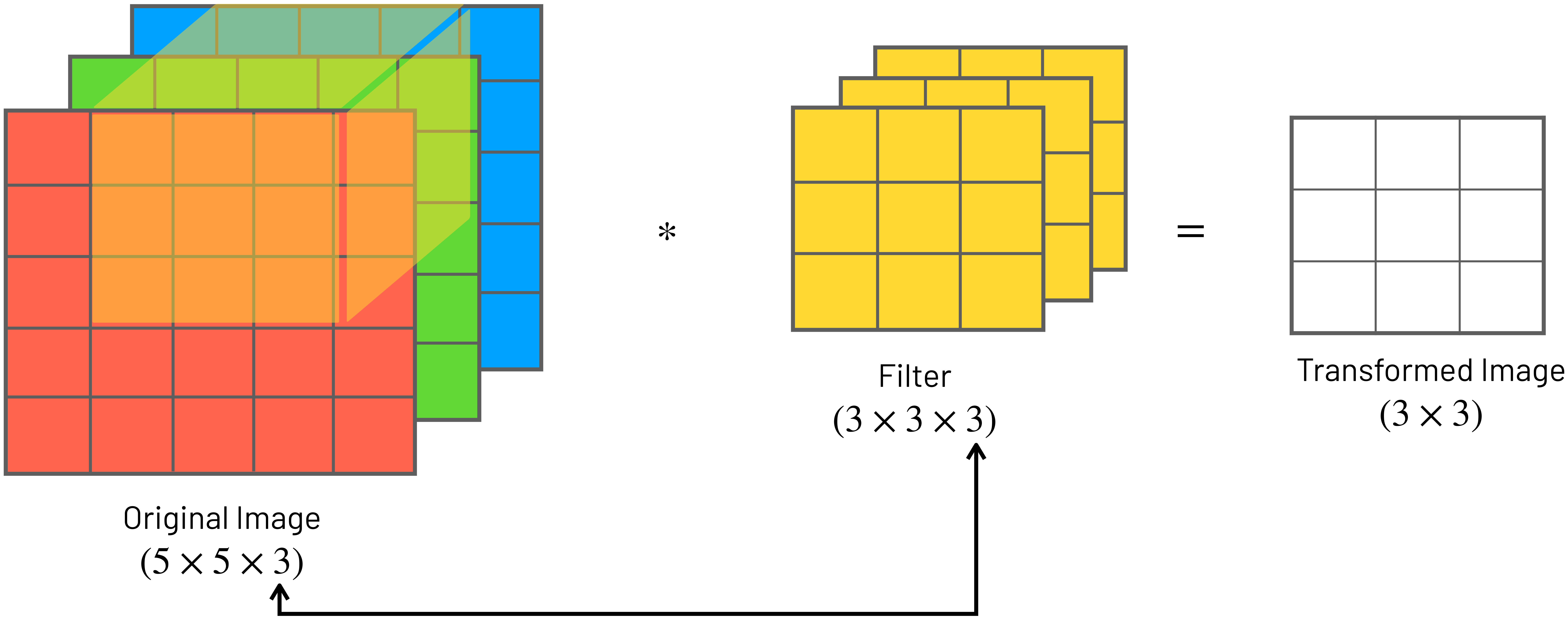
Convolutions in colored images (R,G,B) need filters with 3 channels:



The number of channels in the filter must be the same as in the image!

Convolutions Over Volumes

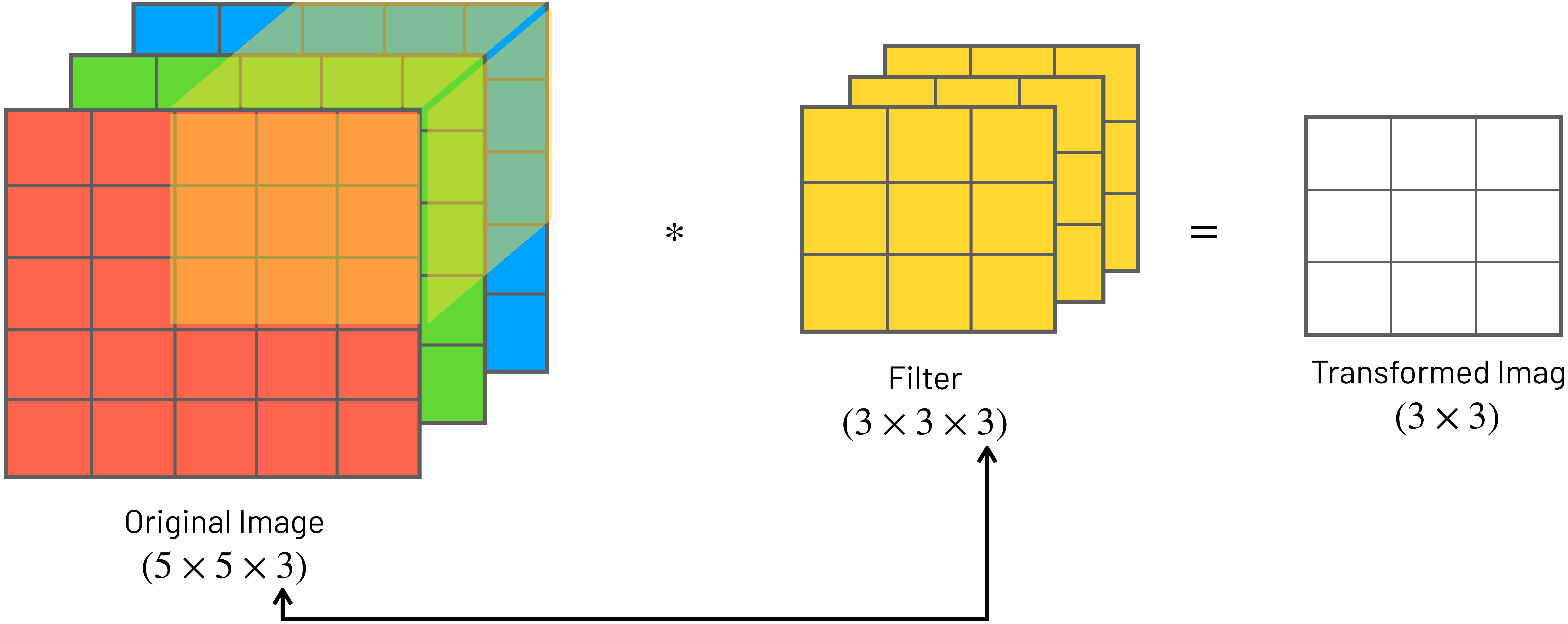
Convolutions in colored images (R,G,B) need filters with 3 channels:



The number of channels in the filter must be the same as in the image!

Convolutions Over Volumes

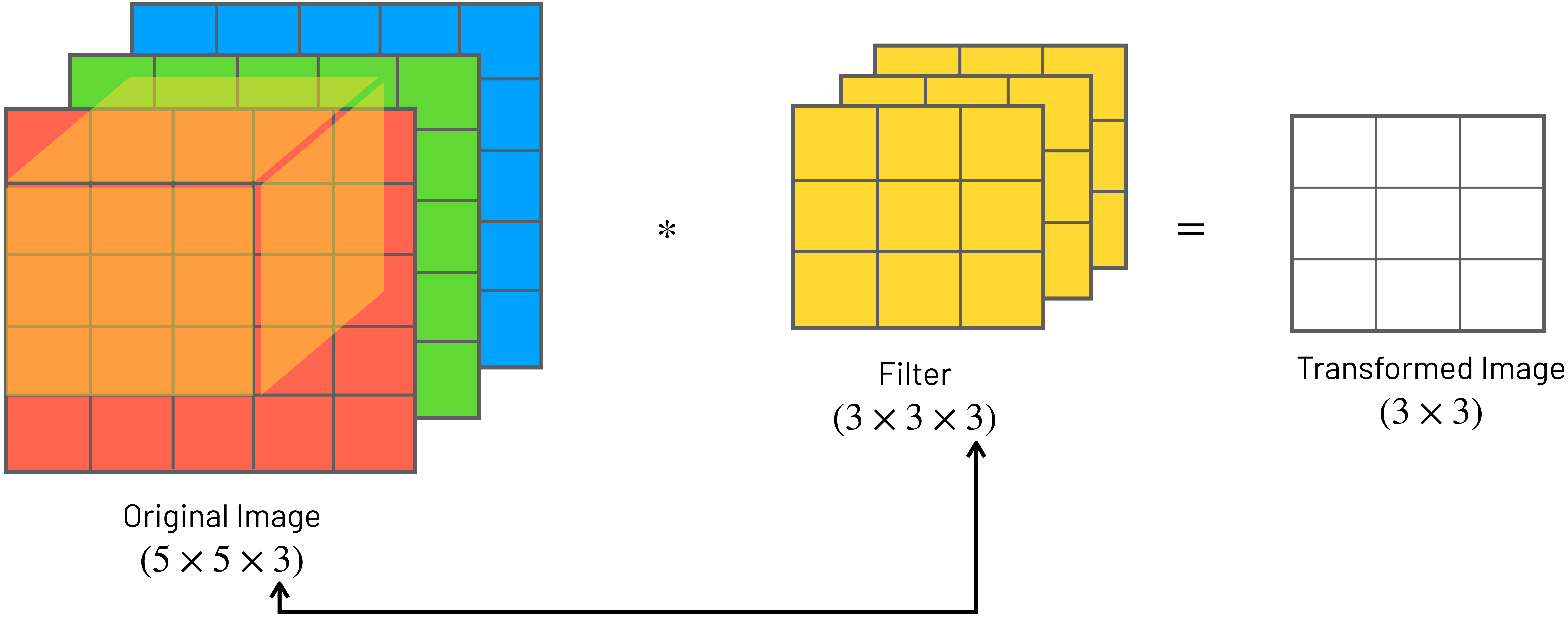
Convolutions in colored images (R,G,B) need filters with 3 channels:



The number of channels in the filter must be the same as in the image!

Convolutions Over Volumes

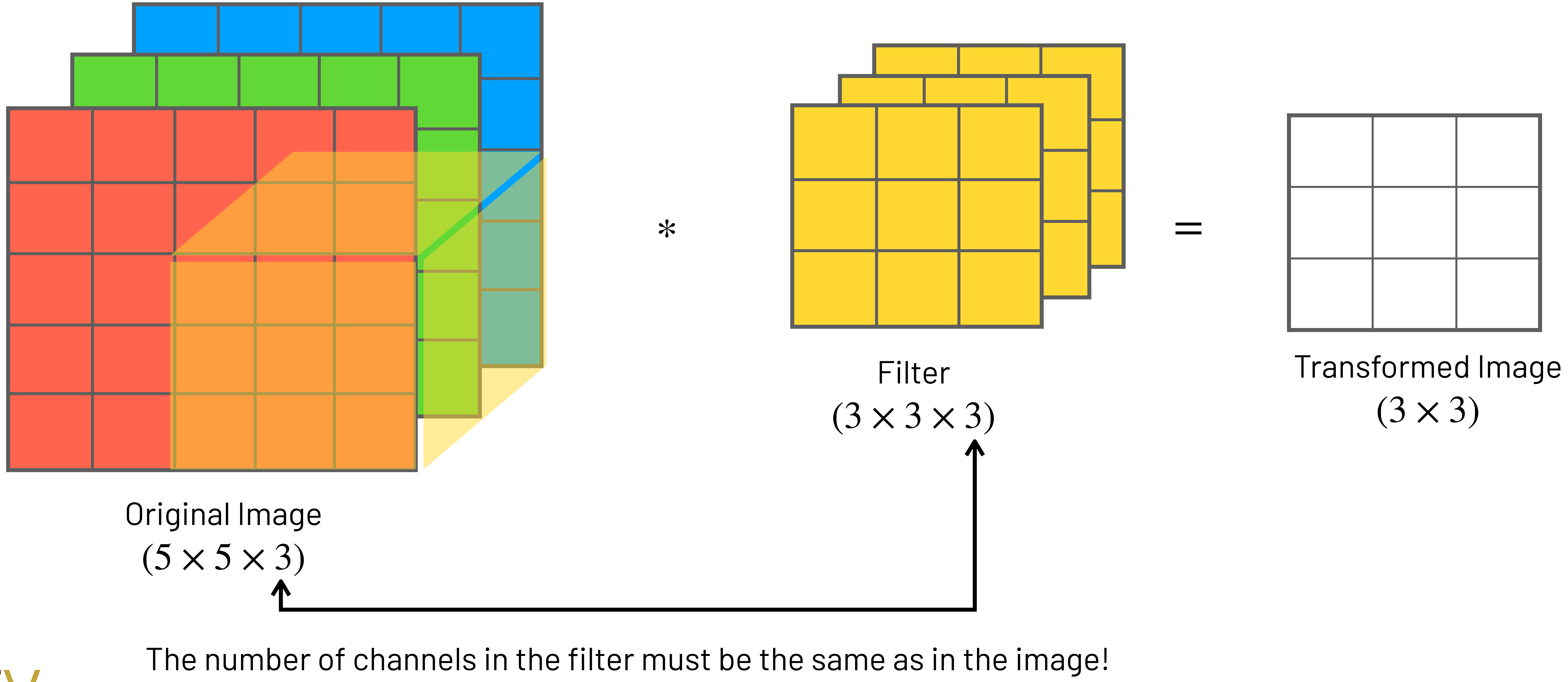
Convolutions in colored images (R,G,B) need filters with 3 channels:



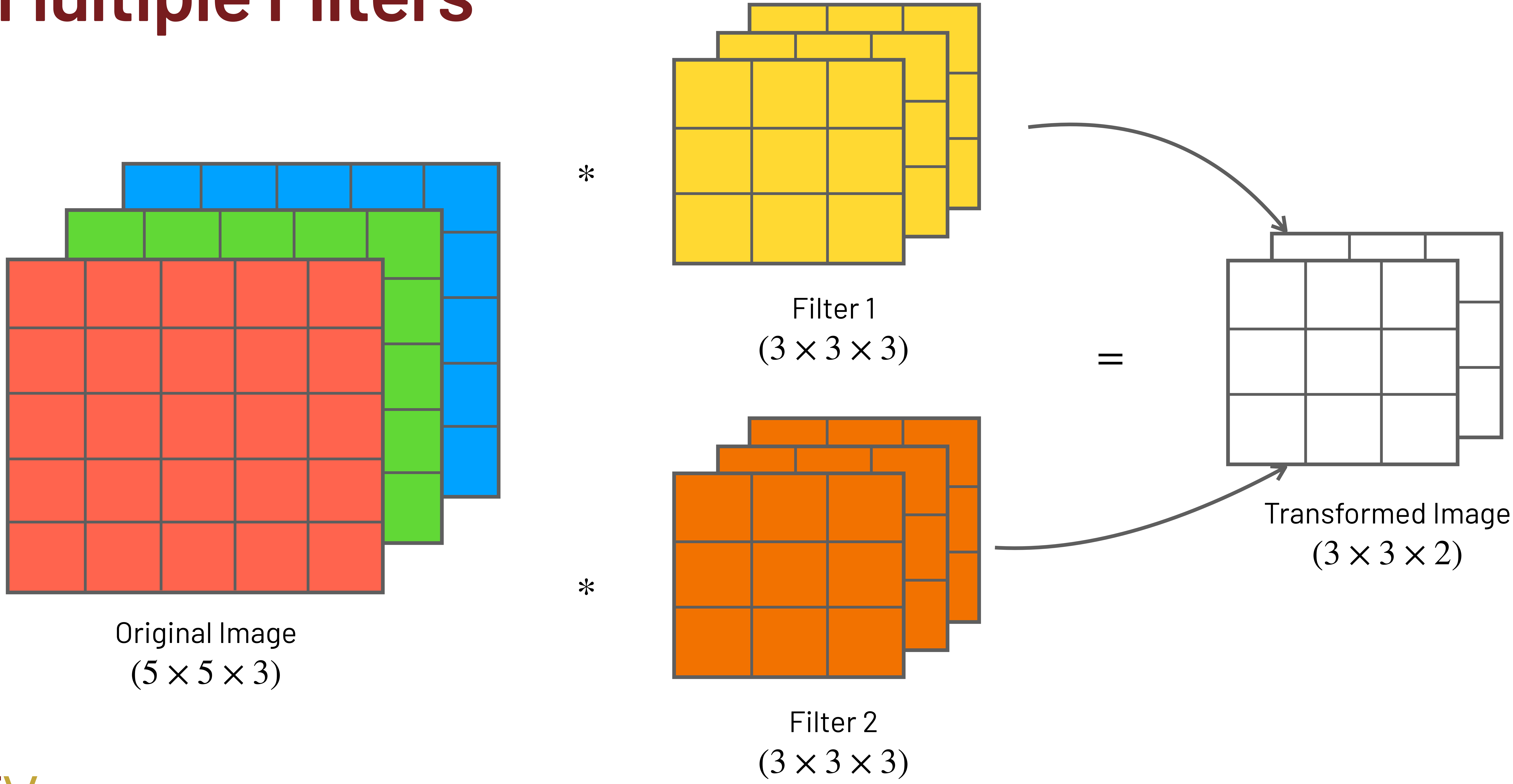
The number of channels in the filter must be the same as in the image!

Convolutions Over Volumes

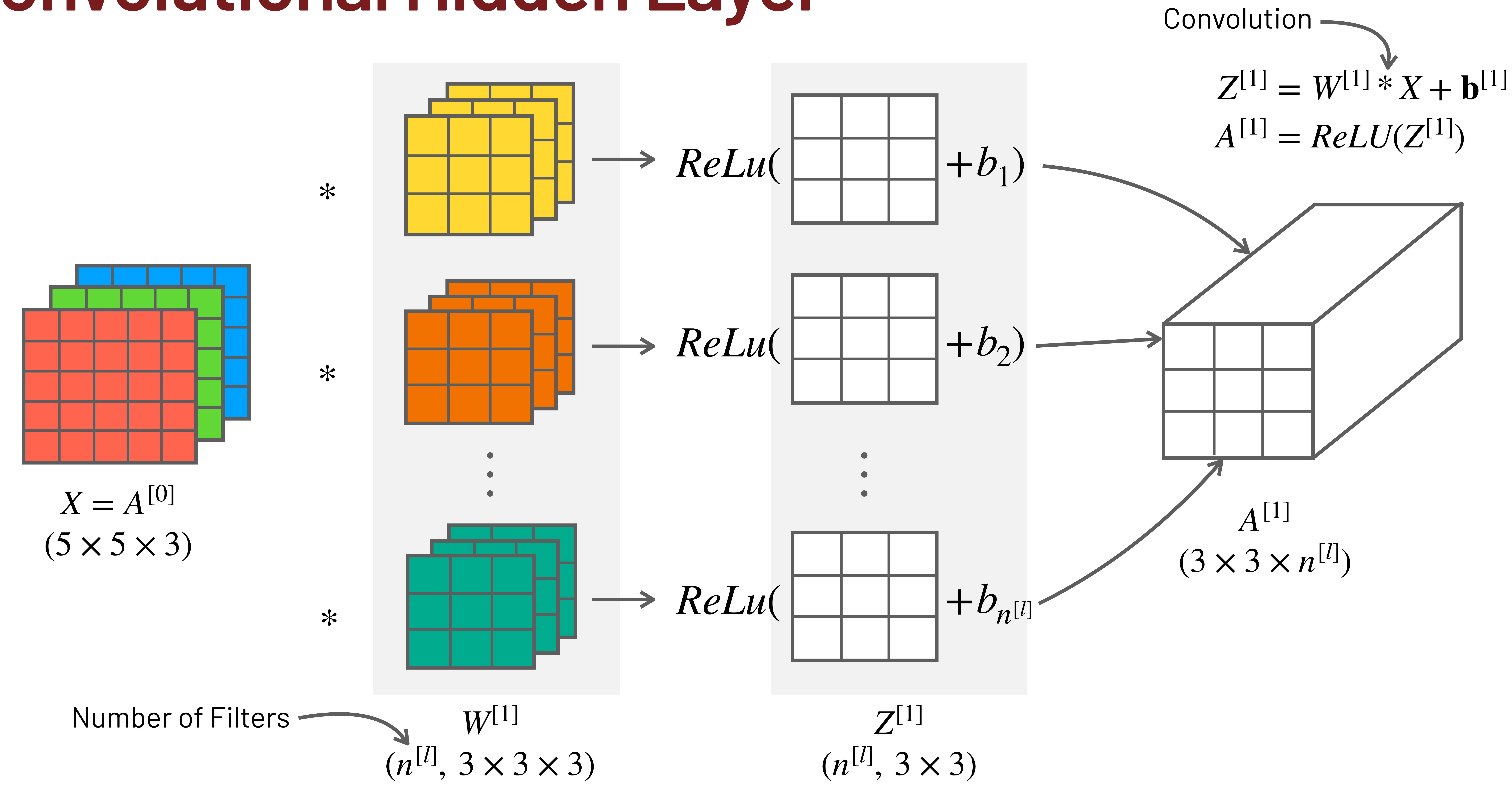
Convolutions in colored images (R,G,B) need filters with 3 channels:



Multiple Filters

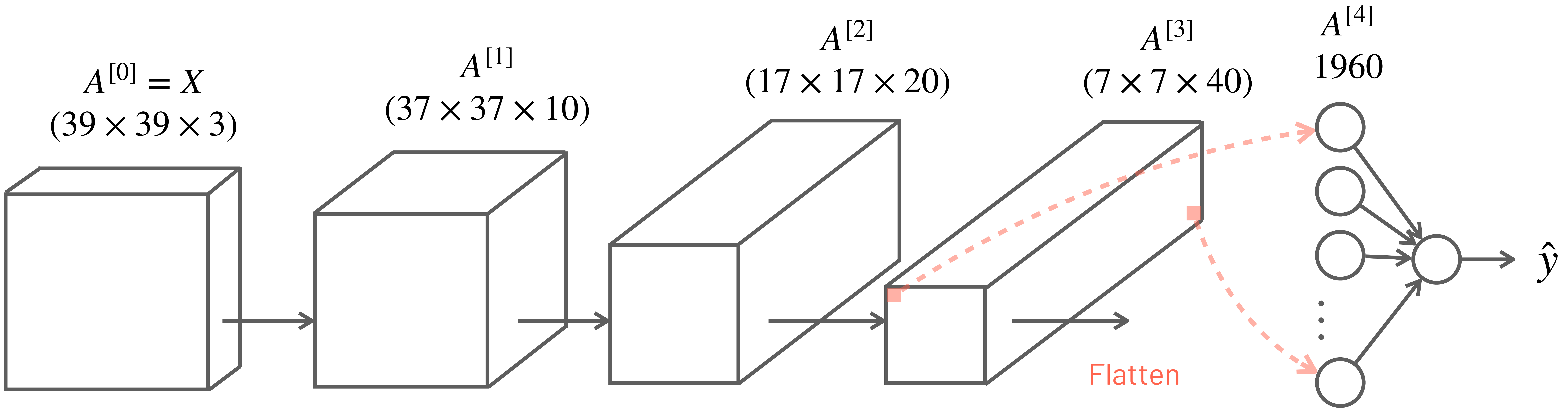


Convolutional Hidden Layer



UFV

CNN for Image Classification



$n^{[0]} = 39$

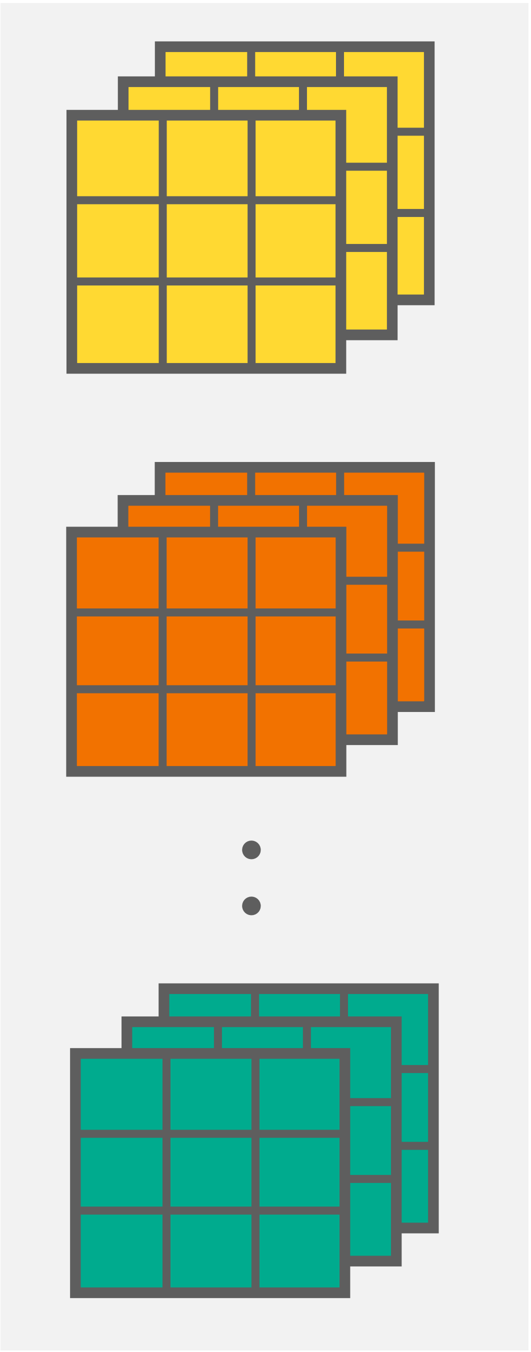
$f^{[1]} = 3$	$f^{[2]} = 5$	$f^{[3]} = 5$
$s^{[1]} = 1$	$s^{[2]} = 2$	$s^{[3]} = 2$
$p^{[1]} = 0$	$p^{[2]} = 0$	$p^{[3]} = 0$
$n^{[1]} = 10$	$n^{[2]} = 20$	$n^{[3]} = 40$

Notation:

- $f^{[l]}$ size of filters in layer l
- $s^{[l]}$ stride size in layer l
- $p^{[l]}$ padding size in layer l
- $n^{[l]}$ number of filters in layer l

Exercise

How many parameters does a layer with 10 filters (3x3x3) have?



$W^{[1]}$
$(10,\ 3 \times 3 \times 3)$

$$
\begin{aligned}
3 \times 3 \times 3 &= 27 \\
&\quad +1 \\
&= 28 \\
&\quad \times 10 \\
&= \underline{280} \text{ Parameters}
\end{aligned}
$$

UFV

Next Lecture

L11: CNN Case Studies

LeNet-5, AlexNet, VGG, ResNet, Inception