INF721

2023/2

UFV

Aprendizado em Redes Neurais Profundas

A13: Redes Neurais Convolucionais

Logística

Avisos

- Projeto P3: Regularização e Otimização será publicado até sexta-feira!

Última aula

- MLP em Pytorch

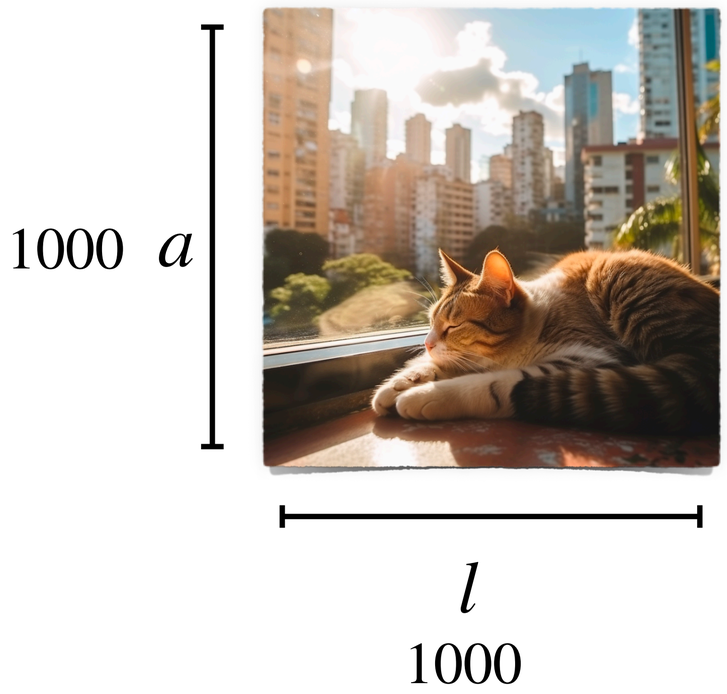
Plano de Aula

- Explosão de Parâmetros
- Filtros
- Convoluções
  - Preenchimento (Padding)
  - Convoluções Passadas (Strided Convolutions)
- Convoluções em Volumes
- Camadas de Padding
- Redes Neurais Convolucionais (CNN)

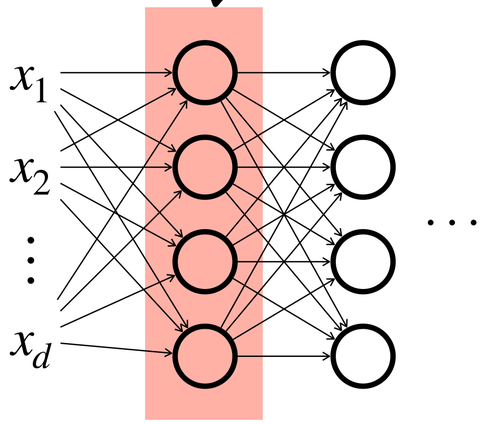
Explosão de parâmetros

Para processar imagens com MLPs, temos que transformá-las em vetores de características.

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



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



3 bilhões de parâmetros! 😱

Problemas:

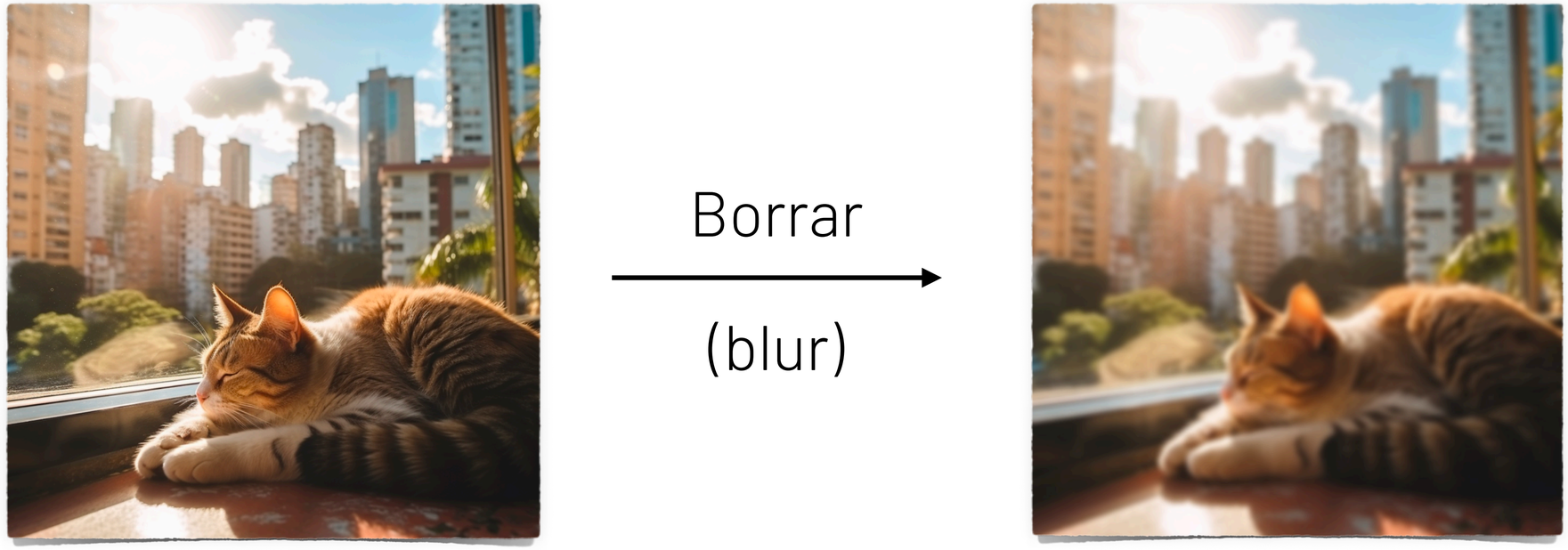
- Voume de dados para evitar sobreajuste
- Recursos computacionais (memória e processamento)

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

UFV

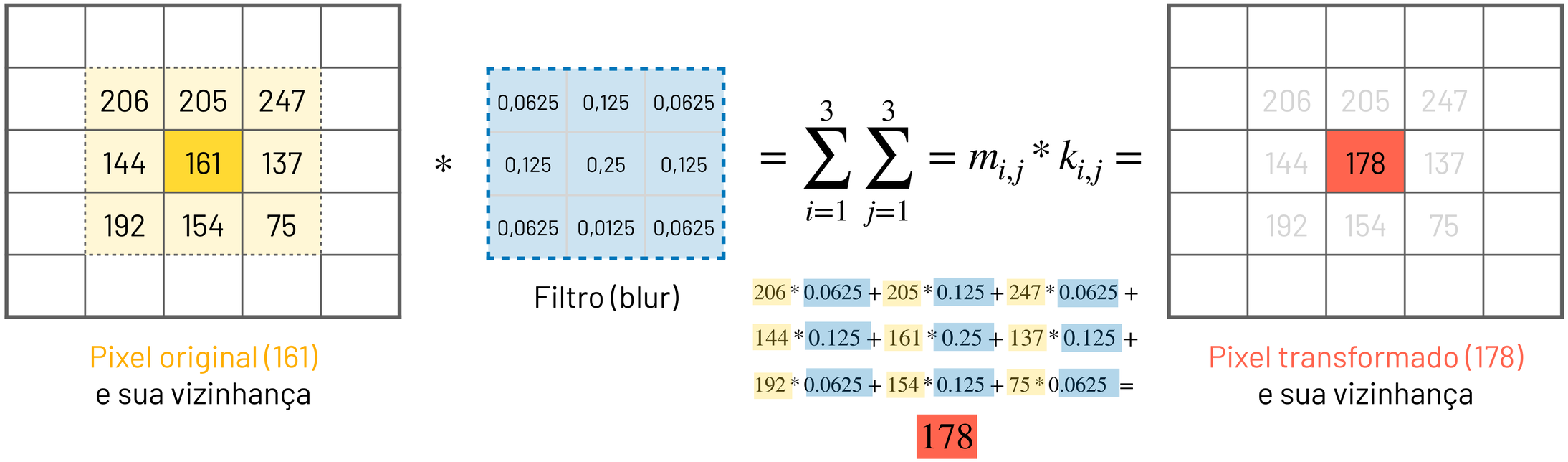
Convoluções

Em processamento de imagens e visão computacional, convoluções são operações para aplicar **filtros** (i.e., transformações) em imagens.



Filtros

Um **filtro** (ou kernel) é uma pequena matriz (geralmente 3x3) de pesos que transforma um **pixel** com a soma ponderada dos pixels de sua vizinhança.



Pixel original (161)
e sua vizinhança

Pixel transformado (178)
e sua vizinhança

Convoluções

Uma **convolução** é uma operação entre uma imagem e um filtro que consiste em aplicar um filtro para cada pixel de uma imagem.

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

Imagem Original
(5×5)

$*$

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

Filtro (blur)

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

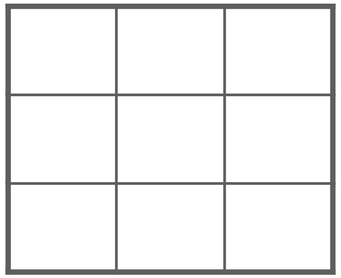
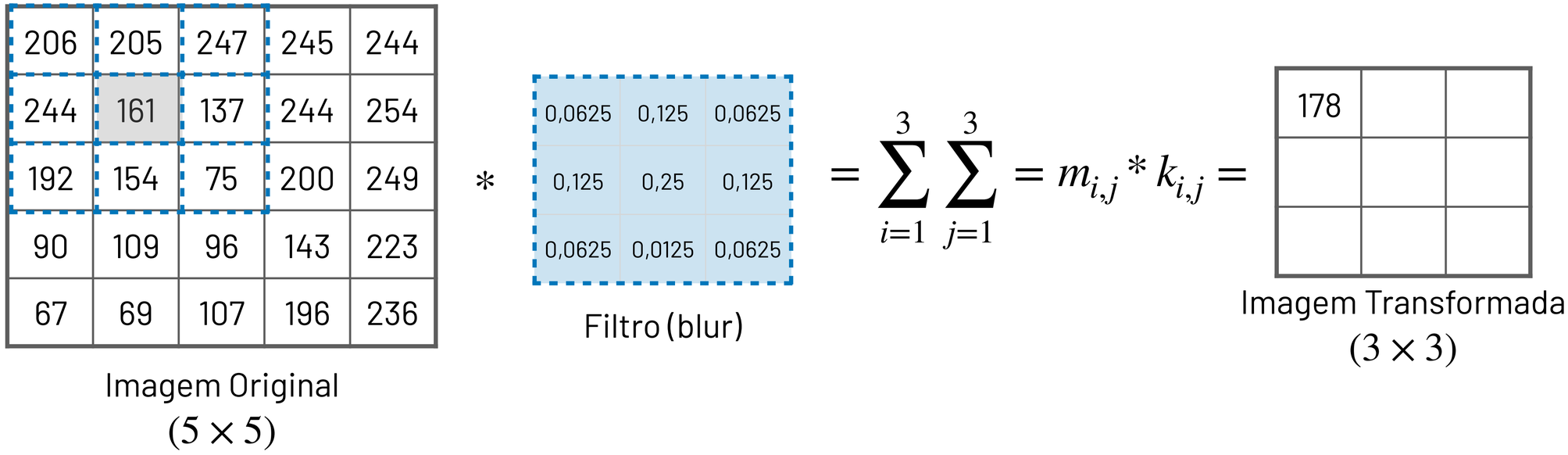


Imagem Transformada
(3×3)

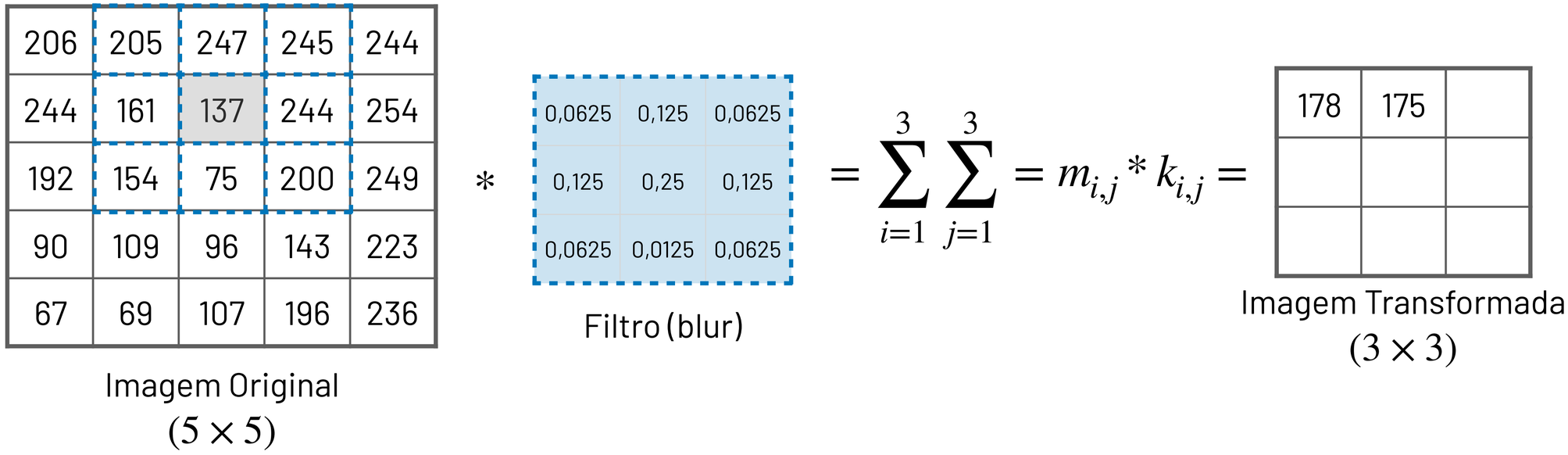
Convoluções

Uma **convolução** é uma operação entre uma imagem e um filtro que consiste em aplicar um filtro para cada pixel de uma imagem.



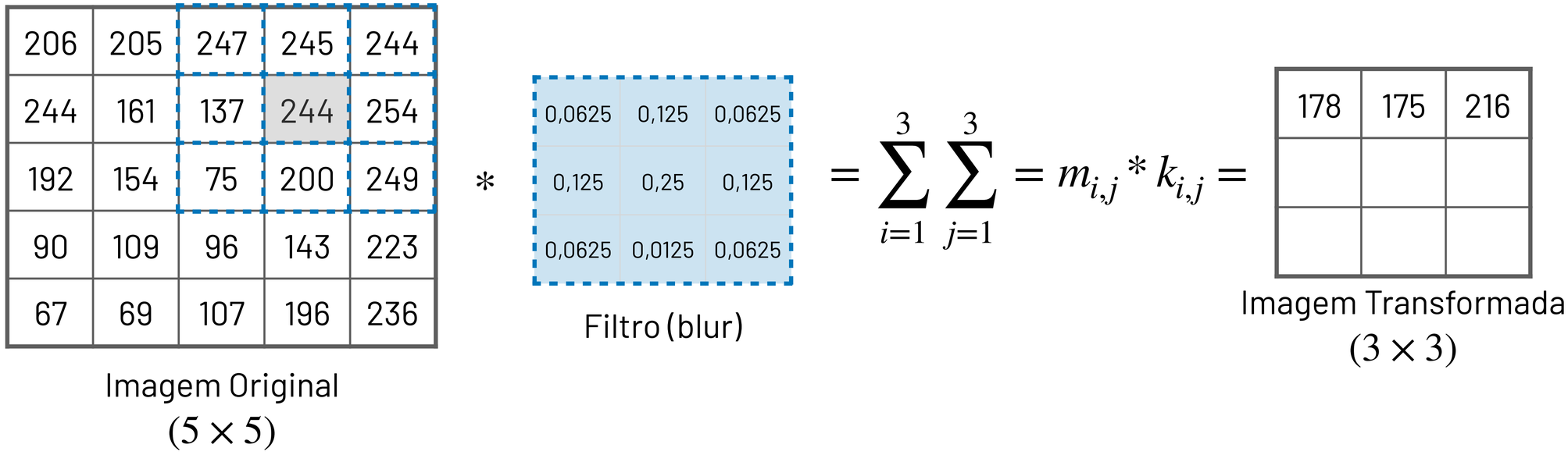
Convoluções

Uma **convolução** é uma operação entre uma imagem e um filtro que consiste em aplicar um filtro para cada pixel de uma imagem.



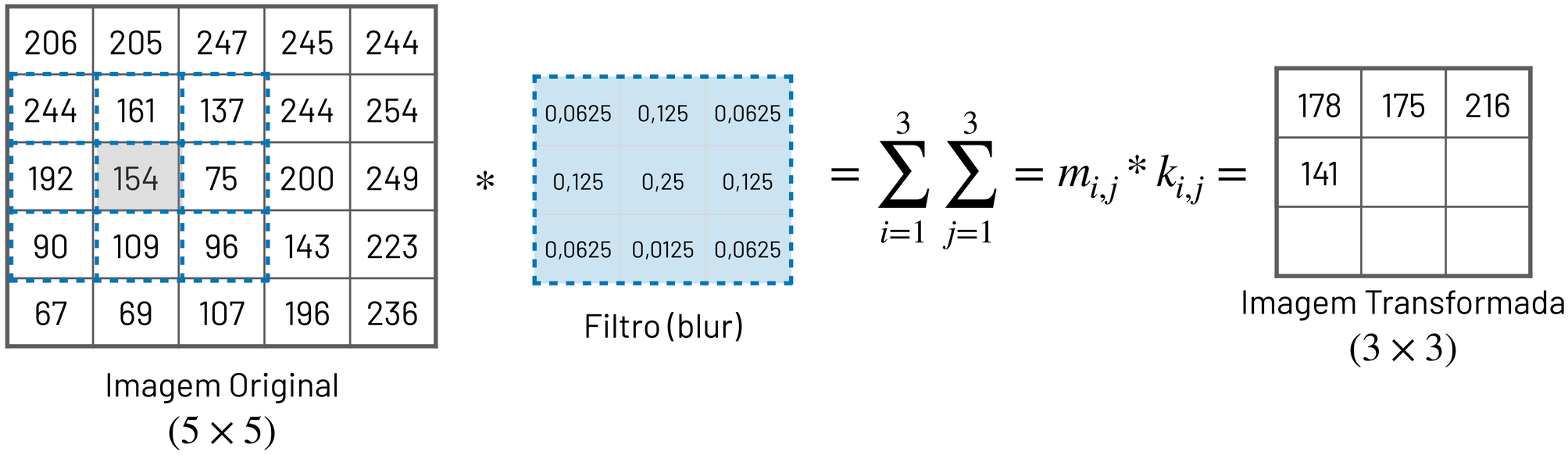
Convoluções

Uma **convolução** é uma operação entre uma imagem e um filtro que consiste em aplicar um filtro para cada pixel de uma imagem.



Convoluções

Uma **convolução** é uma operação entre uma imagem e um filtro que consiste em aplicar um filtro para cada pixel de uma imagem.



Convoluções

Uma **convolução** é uma operação entre uma imagem e um filtro que consiste em aplicar um filtro para cada pixel de uma imagem.

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

Imagem Original
(5×5)

$*$

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

Filtro (blur)

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

178	175	216
141	133	183
106	117	167

Imagem Transformada
(3×3)

Detecção de borda

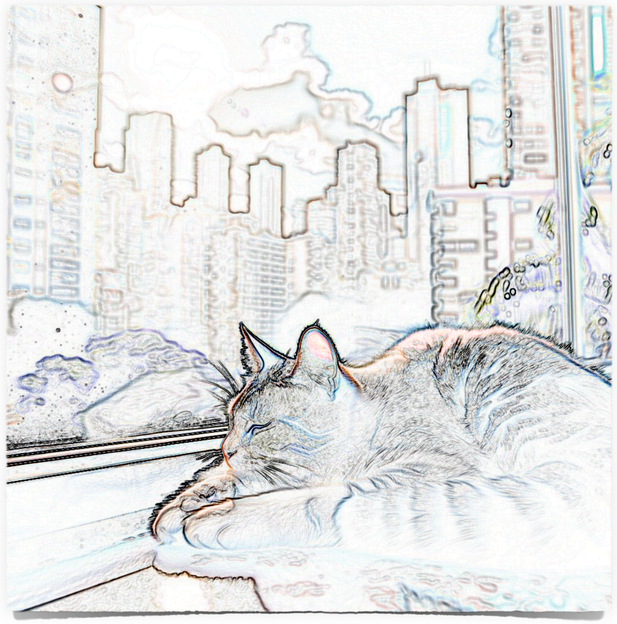
Filtros podem ser utilizados para detecção de borda em imagens, o que é particularmente importante para extração de características.

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

Vertical

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

Horizontal



Desenvolvendo Filtros

Diferentes filtros de detecção de borda foram desenvolvidos cientificamente pela comunidade de processamento de imagens.

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

Aprendendo Filtros

Redes Neurais Convolucionais (CNNs) **aprendem filtros** a partir de imagens e uma função de erro por meio do gradiente descendente.

$*$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Os pesos da uma CNN são organizados em filtros de convolução

Convoluções reduzem o tamanho da imagem

- Aplicações consecutivas de convoluções podem tornar a imagem muito pequena (e.g., 1x1)
- Pixels dos cantos são menos compartilhados que pixels do meio

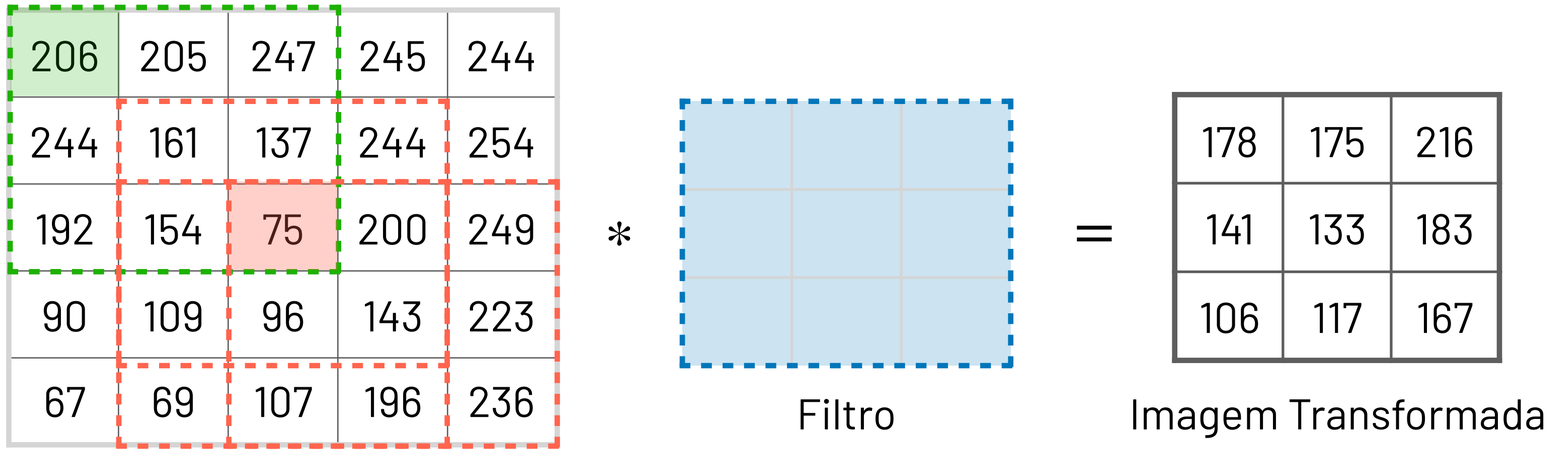


Imagem Original
(5×5)

Filtro
(3×3)

Imagem Transformada
(3×3)

Regra Geral

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

Preenchimento (Padding)

Adicionar uma borda com p pixels na imagem original.

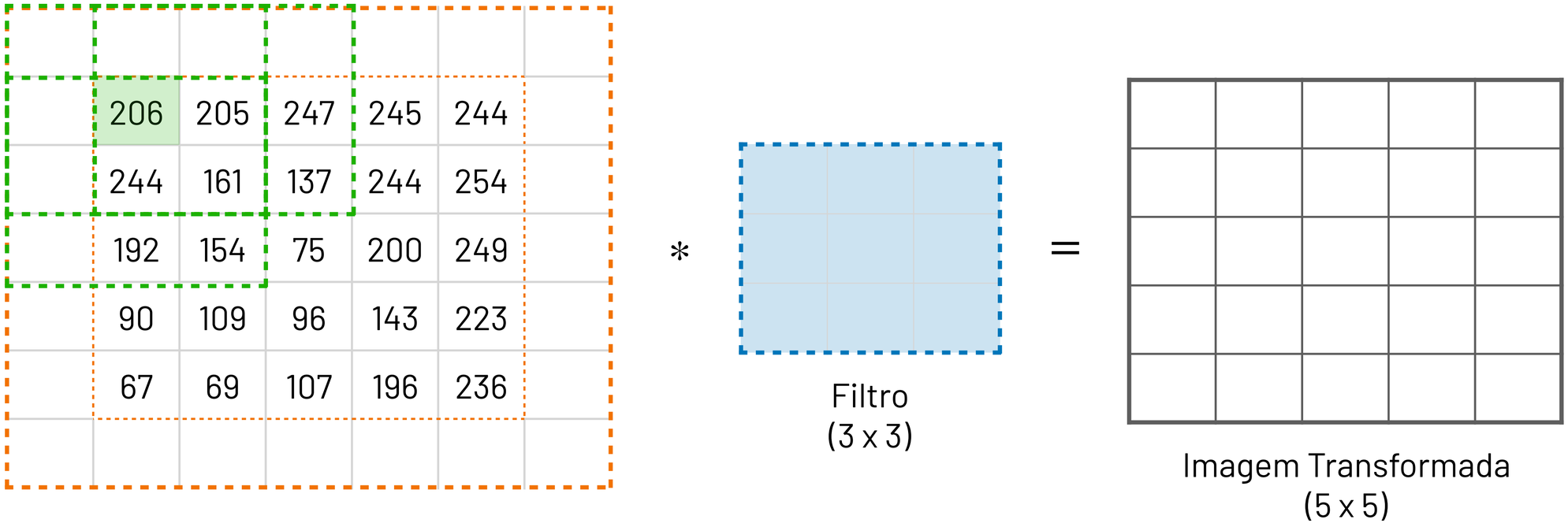


Imagem Original
(5 x 5)

Filtro
(3 x 3)

Imagem Transformada
(5 x 5)

Regra Geral

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

Preenchimento (Padding)

Para encontrar o valor de p que mantém o tamanho de uma imagem $n \times n$ após a convolução com um filtro de tamanho f (ímpar), basta resolver a seguinte equação:

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

Convoluções Passadas (*Strided Convolutions*)

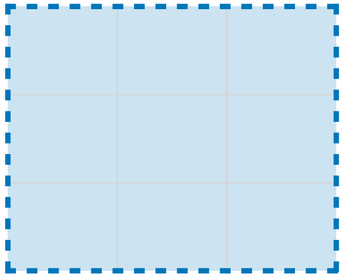
Convoluções podem ser executadas com passos *(strides)* maiores do que 1.

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

Imagem Original
(5×5)

$*$



Filtro
(3×3)

$=$

Regra Geral

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

Convoluções Passadas (*Strided Convolutions*)

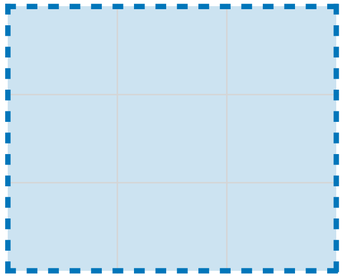
Convoluções podem ser executadas com passos *(strides)* maiores do que 1.

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

Imagem Original
(5×5)

$*$



Filtro
(3×3)

$=$

Regra Geral

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

Convoluções Passadas (*Strided Convolutions*)

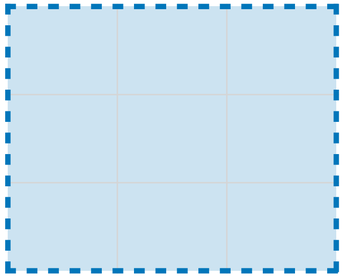
Convoluções podem ser executadas com passos *(strides)* maiores do que 1.

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

Imagem Original
(5×5)

$*$



Filtro
(3×3)

$=$

Regra Geral

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

Convoluções Passadas (*Strided Convolutions*)

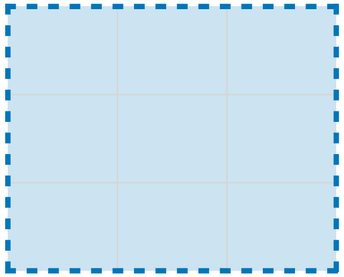
Convoluções podem ser executadas com passos *(strides)* maiores do que 1.

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

Imagem Original
(5×5)

$*$



Filtro
(3×3)

$=$

Regra Geral

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

Convoluções Passadas (*Strided Convolutions*)

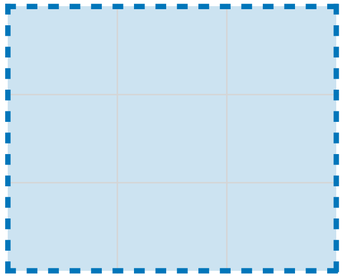
Convoluções podem ser executadas com passos *(strides)* maiores do que 1.

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

Imagem Original
(5×5)

$*$



Filtro
(3×3)

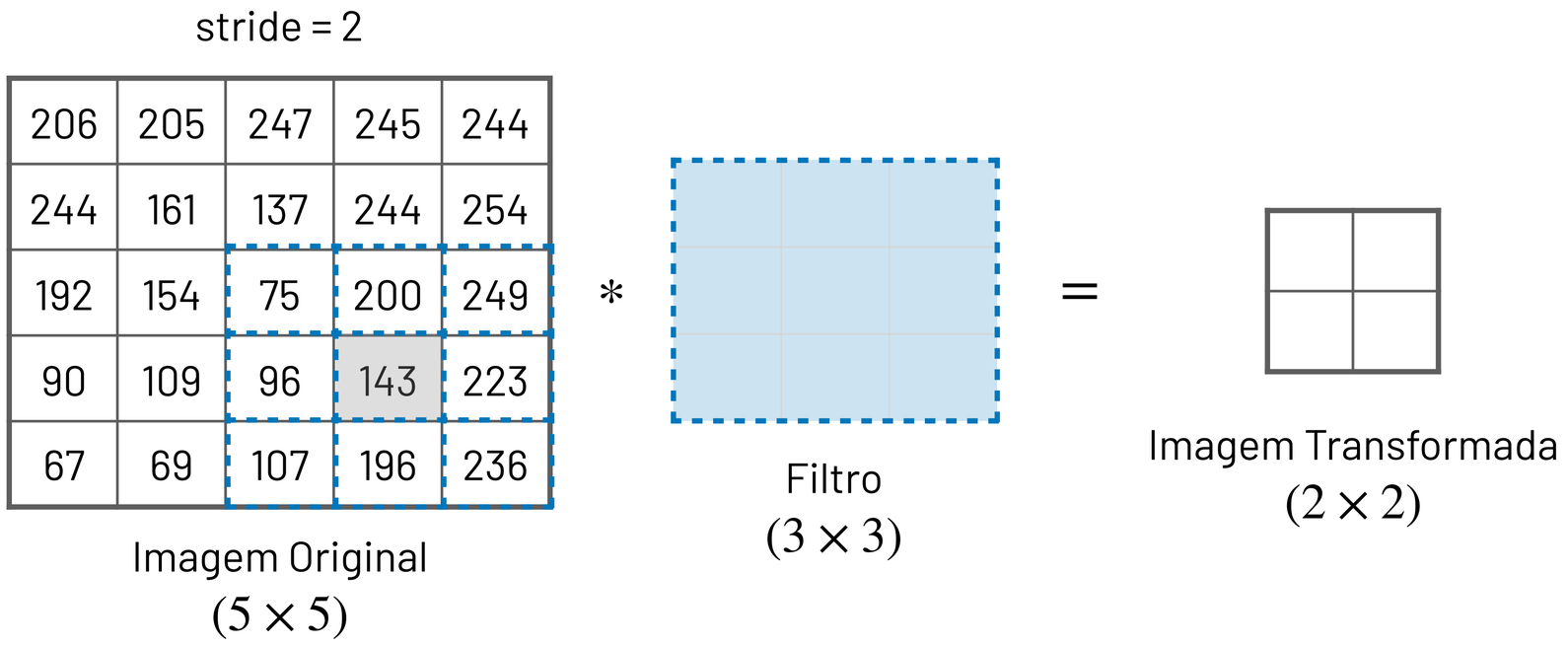
$=$

Regra Geral

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

Convoluções Passadas (*Strided Convolutions*)

Convoluções podem ser executadas com passos *(strides)* maiores do que 1.

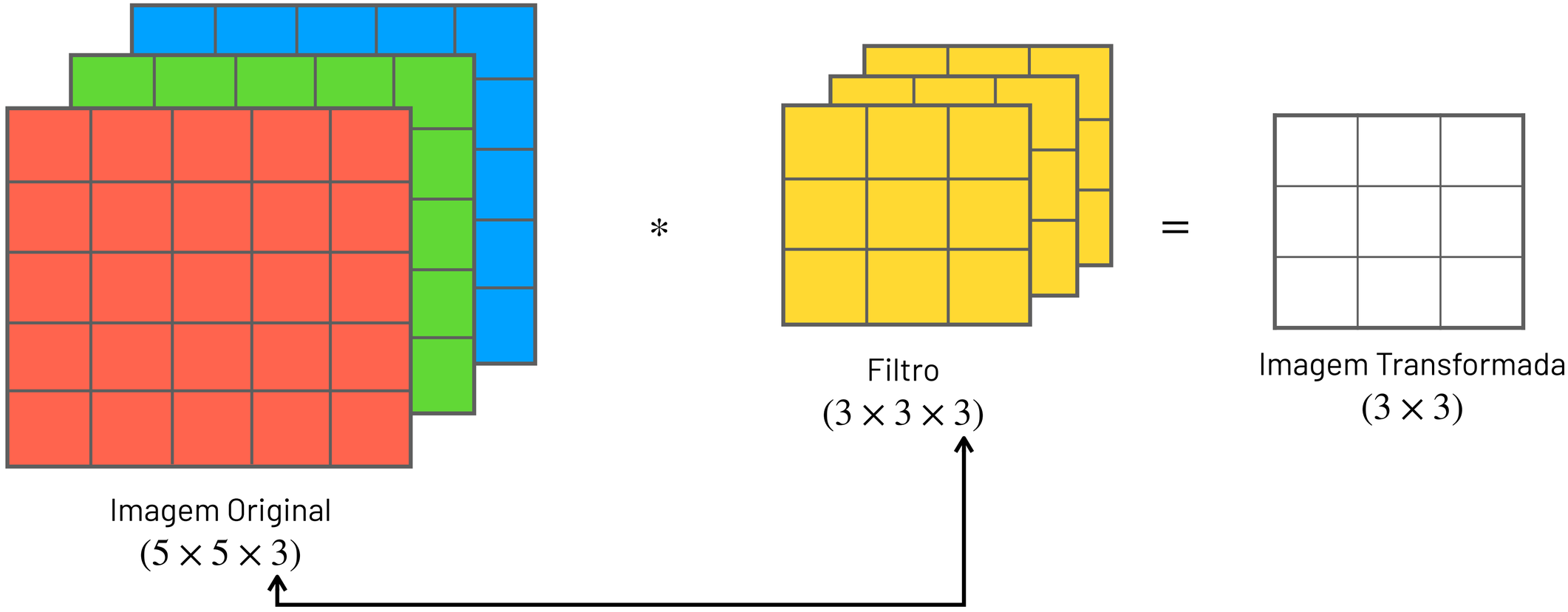


Regra Geral

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

Convoluções em Volumes

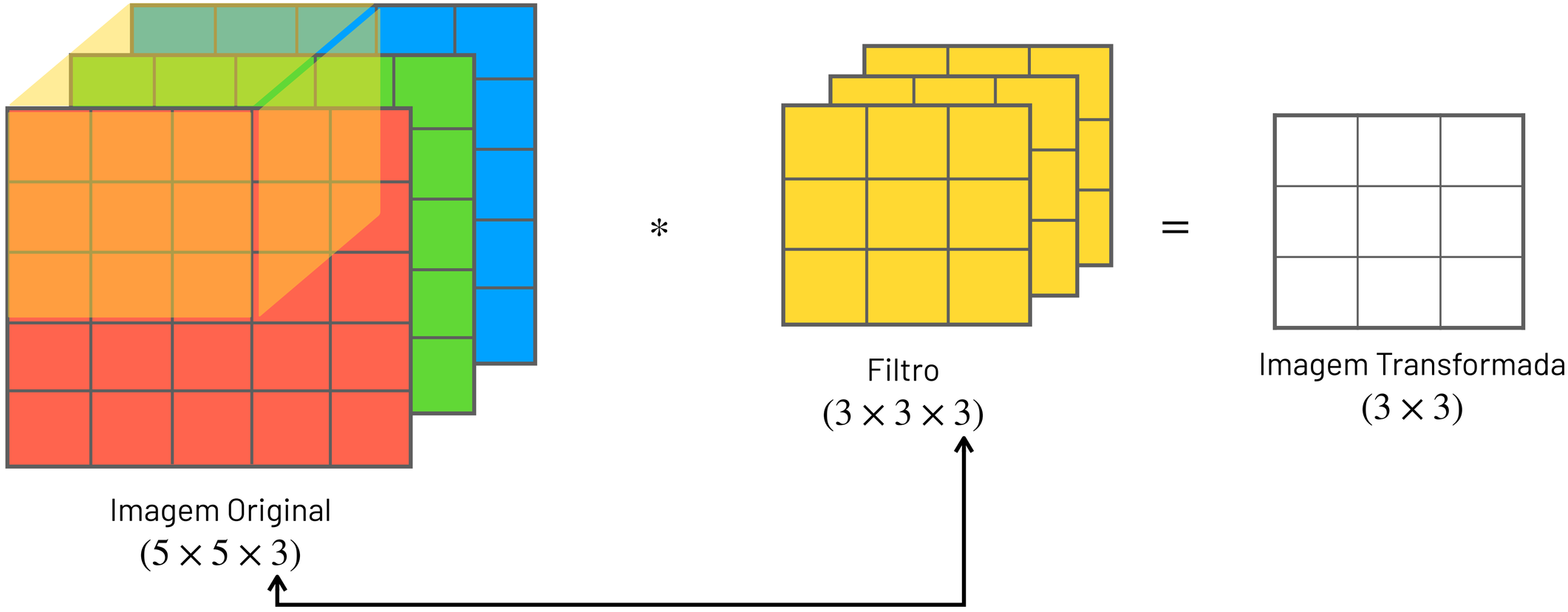
Convoluções em imagens coloridas (R,G,B) necessitam filtros com 3 canais



O número de canais deve ser o mesmo na imagem e no filtro!

Convoluções em Volumes

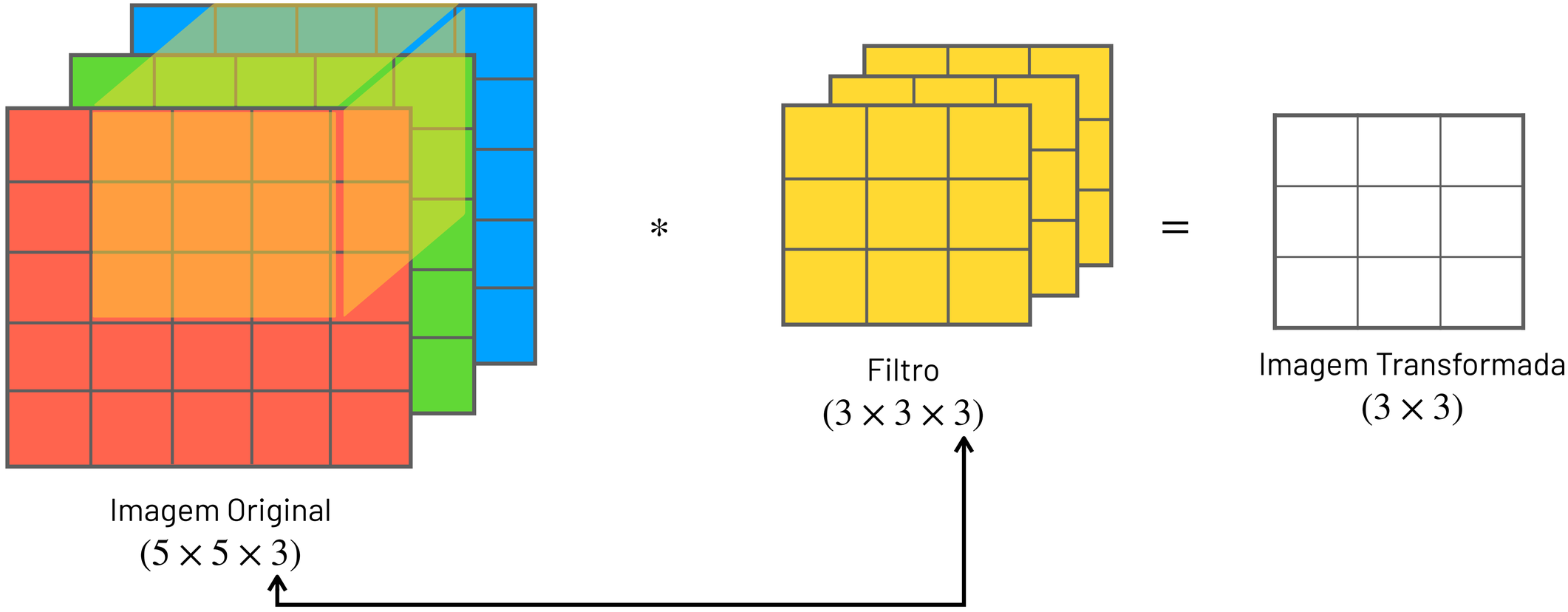
Convoluções em imagens coloridas (R,G,B) necessitam filtros com 3 canais



O número de canais deve ser o mesmo na imagem e no filtro!

Convoluções em Volumes

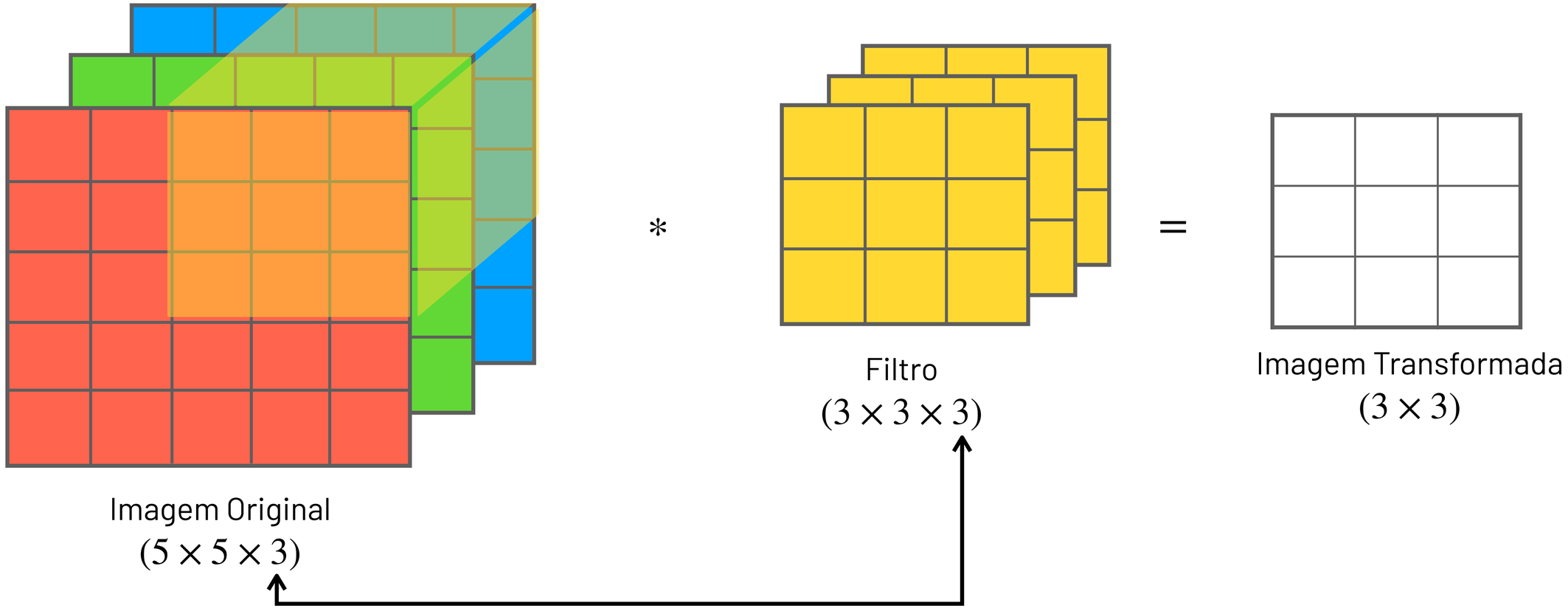
Convoluções em imagens coloridas (R,G,B) necessitam filtros com 3 canais



O número de canais deve ser o mesmo na imagem e no filtro!

Convoluções em Volumes

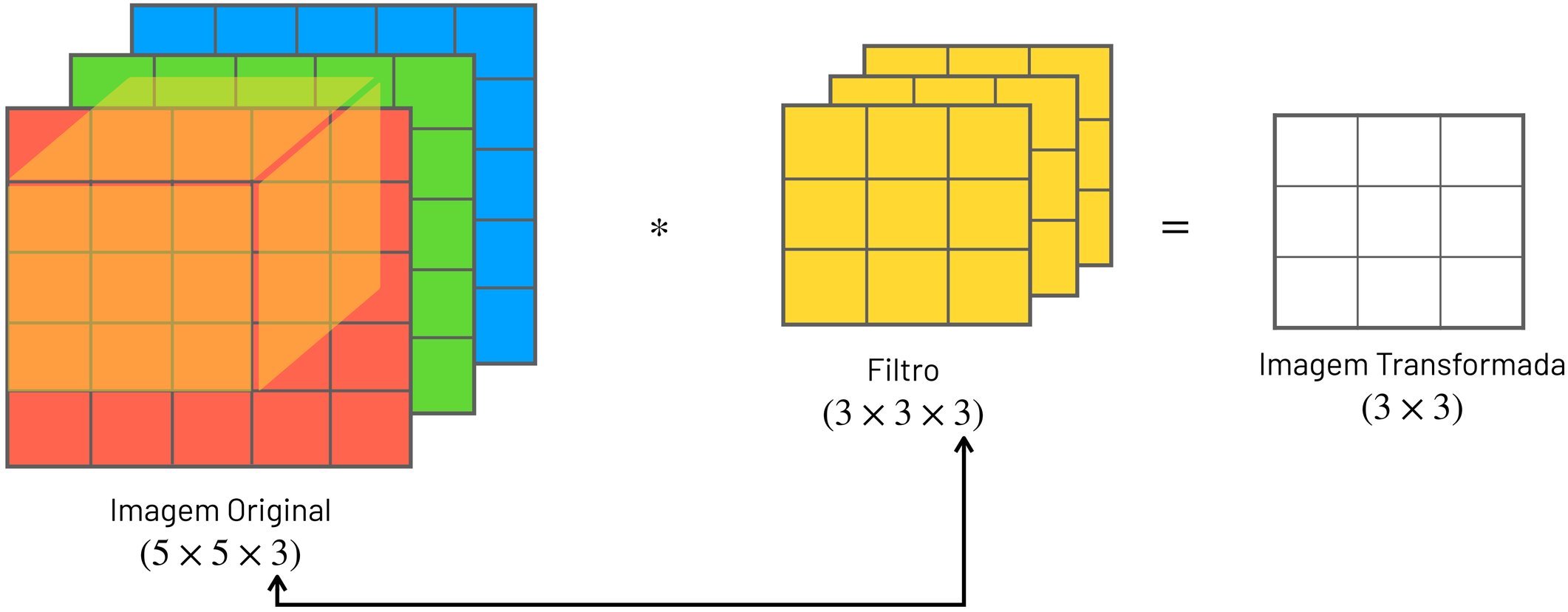
Convoluções em imagens coloridas (R,G,B) necessitam filtros com 3 canais



O número de canais deve ser o mesmo na imagem e no filtro!

Convoluções em Volumes

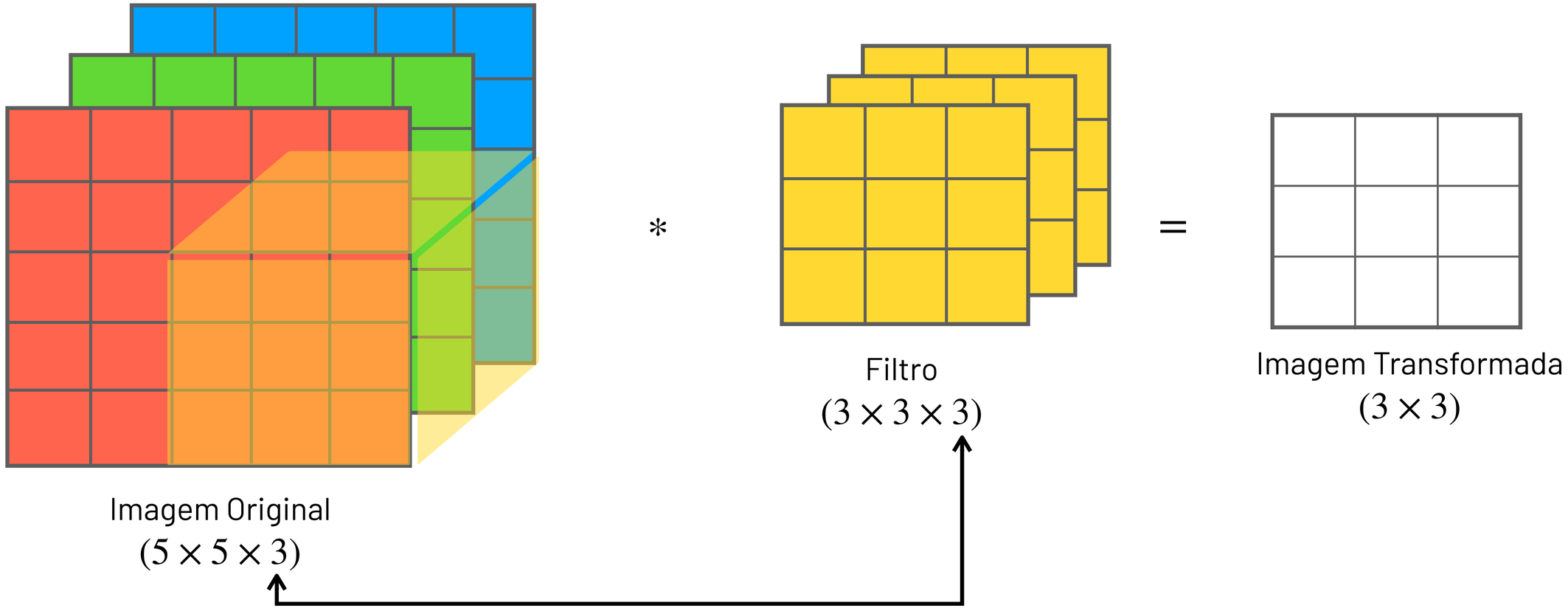
Convoluções em imagens coloridas (R,G,B) necessitam filtros com 3 canais



O número de canais deve ser o mesmo na imagem e no filtro!

Convoluções em Volumes

Convoluções em imagens coloridas (R,G,B) necessitam filtros com 3 canais



O número de canais deve ser o mesmo na imagem e no filtro!

Múltiplos Filtros

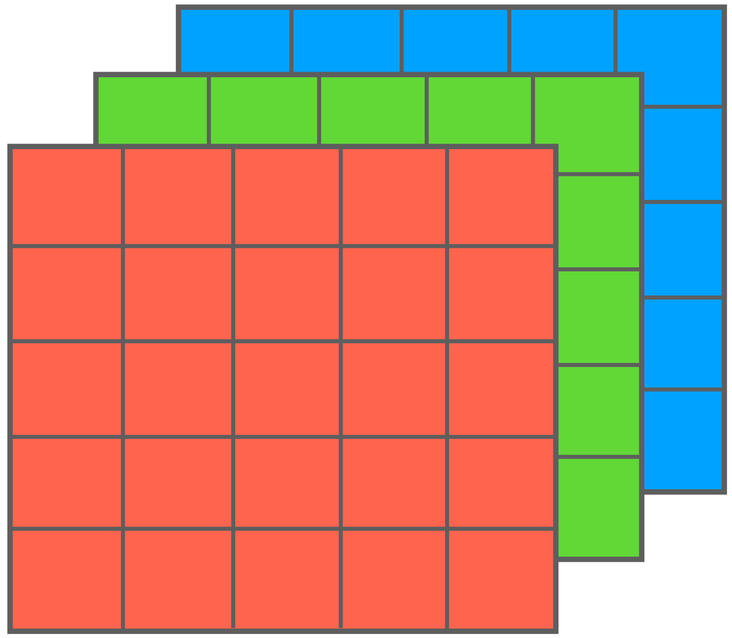
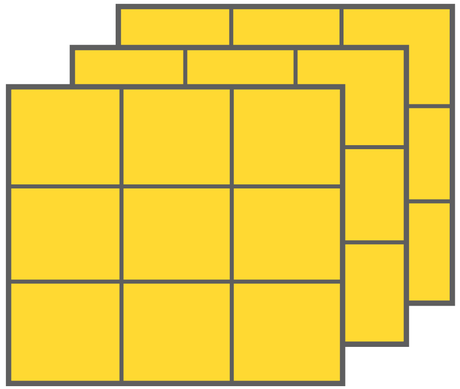


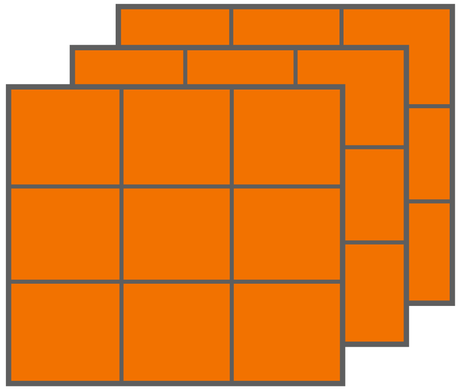
Imagem Original
$(5 \times 5 \times 3)$

$*$



Filtro 1
$(3 \times 3 \times 3)$

$*$



Filtro 2
$(3 \times 3 \times 3)$

$=$

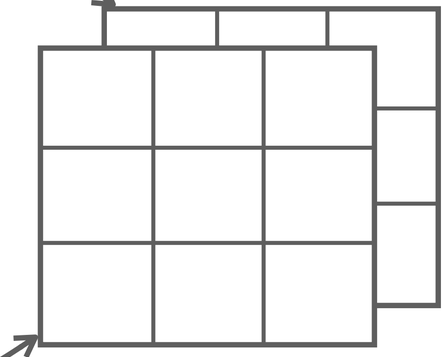
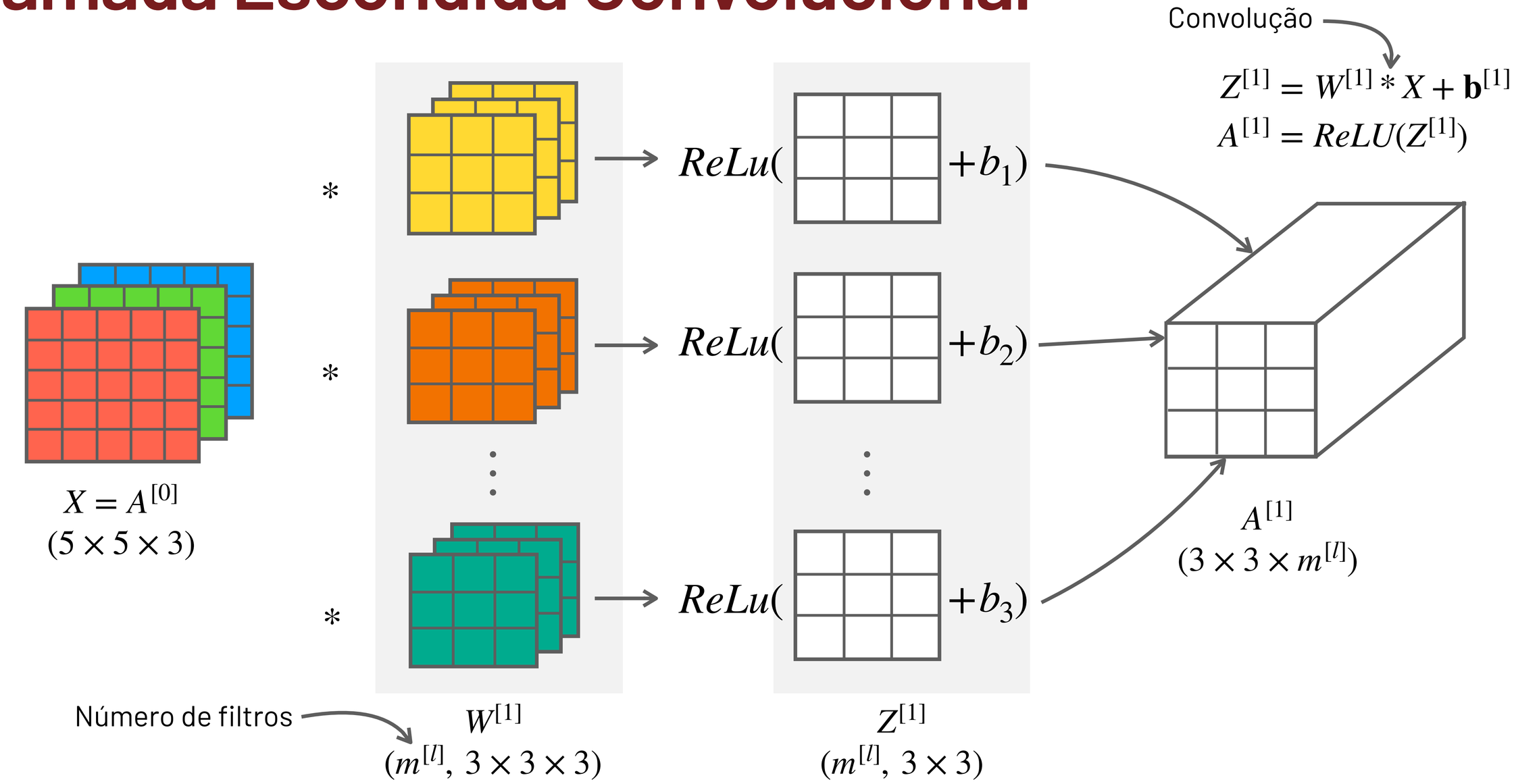
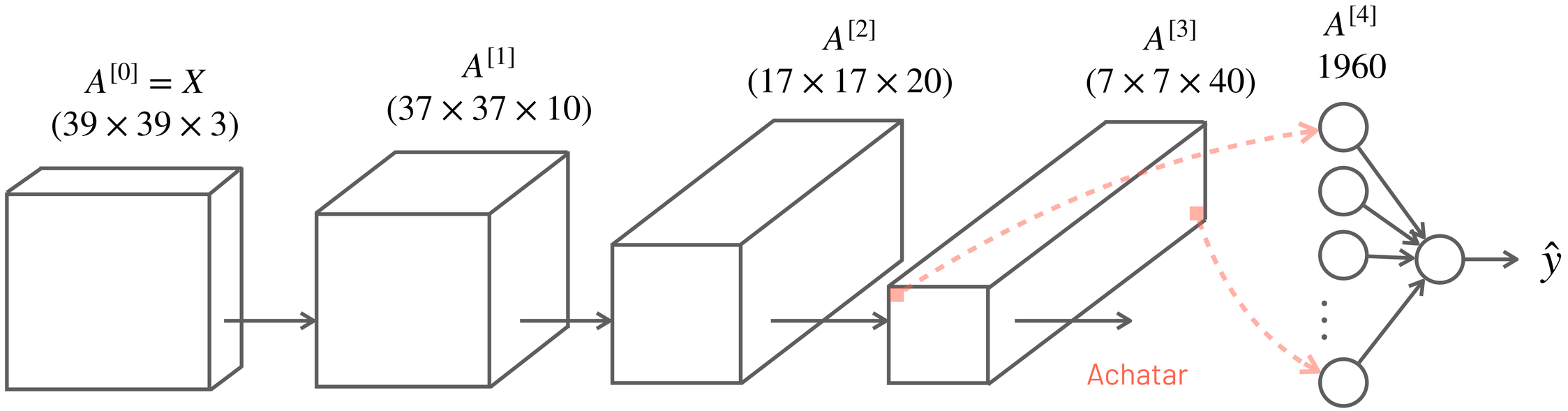


Imagem Transformada
$(3 \times 3 \times 2)$

Camada Escondida Convolucional



Rede Neural Convolucional (CNN) de Classificação



$n^{[0]} = 39$

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

$f^{[2]} = 5$
$s^{[2]} = 2$
$p^{[2]} = 0$
$m^{[2]} = 20$

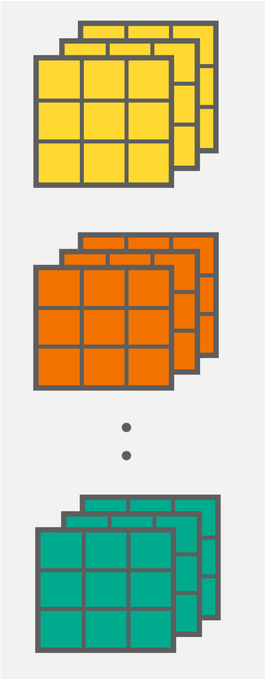
$f^{[3]} = 5$
$s^{[3]} = 2$
$p^{[3]} = 0$
$m^{[3]} = 40$

Notação:

- $f^{[l]}$ tamanho dos filtros da camada l
- $s^{[l]}$ tamanho do stride da camada l
- $p^{[l]}$ tamanho do padding da camada l
- $m^{[l]}$ número de filtros na camada l

Exercício

Quantos parâmetros uma camada com 10 filtros (3x3x3) tem?



$W^{[1]}$

$(10,\ 3 \times 3 \times 3)$

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

Camadas de Pooling

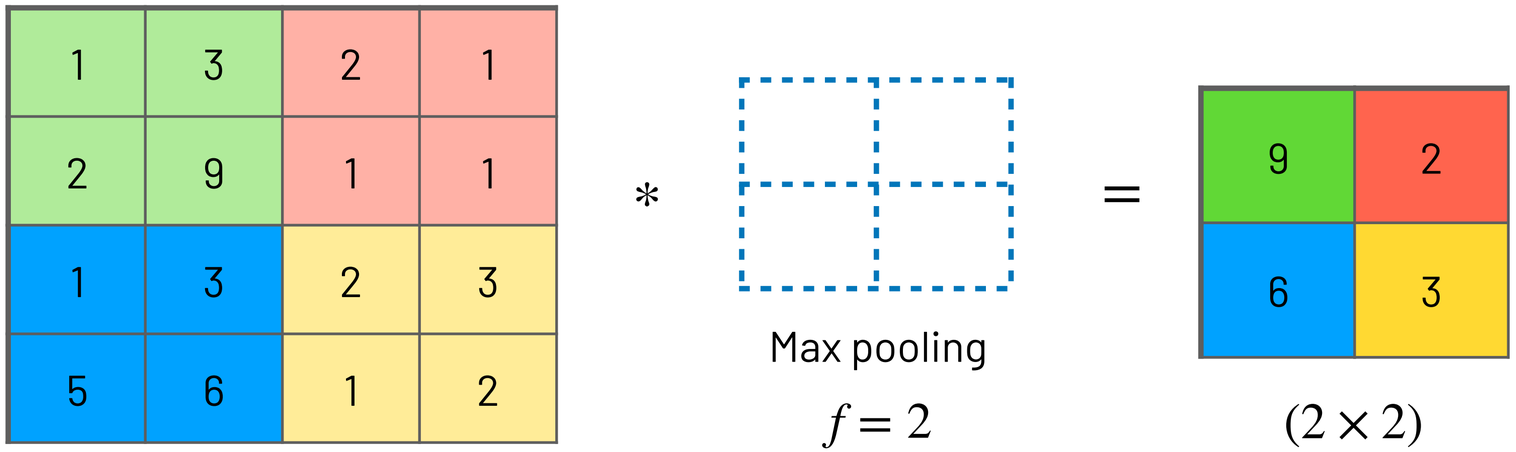
Além das camadas convolucionais, CNNs tipicamente também utilizam **camadas de pooling** para extrair características de imagens:

- Max Pooling
- Average Pooling

Essas camadas realizam computações fixas e por isso não possuem pesos para aprender!

Max pooling

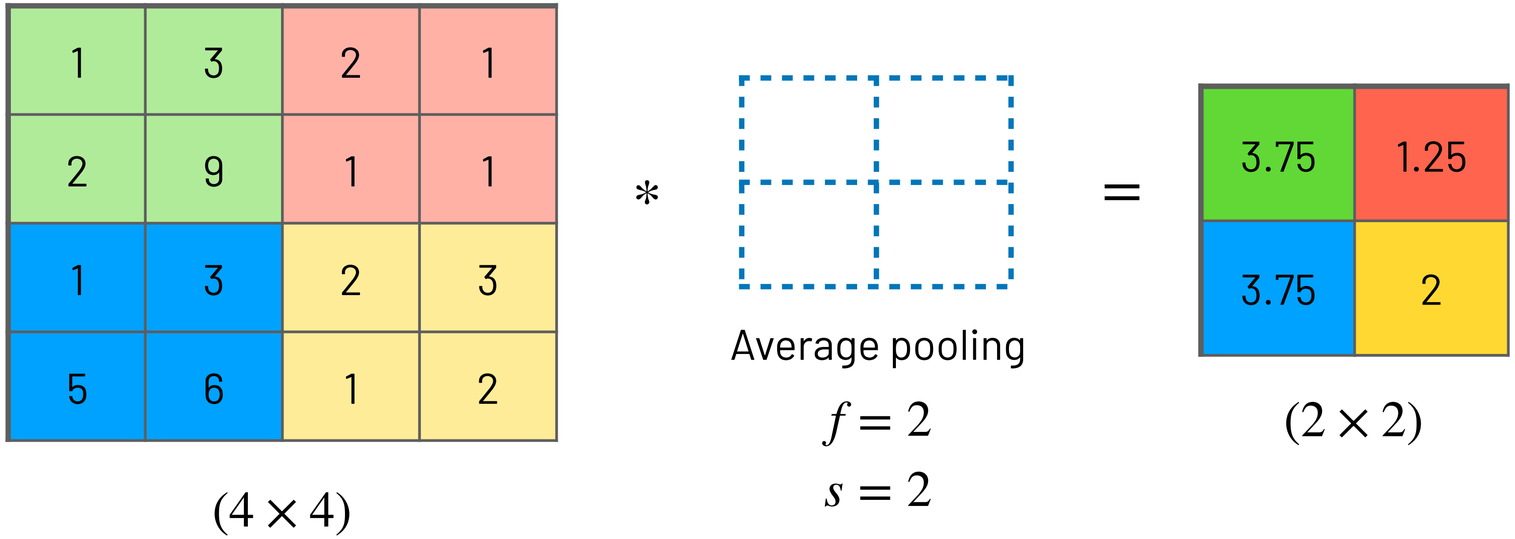
Filtro para extrair o elemento máximo da vizinhança.



Sem pesos para aprender!

Average pooling

Filtro para extrair a média da vizinhança.



(4×4)

Average pooling

$f = 2$

$s = 2$

(2×2)

Sem pesos para aprender!

Exercício

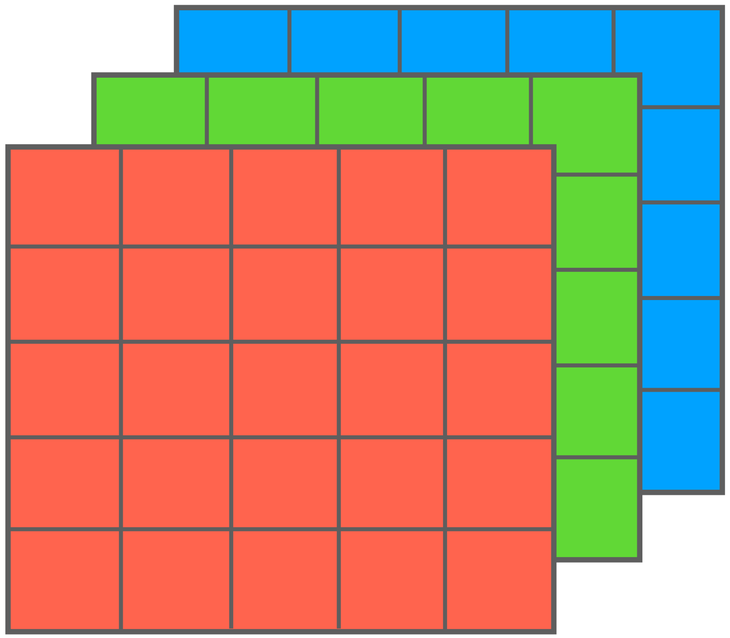
Calcule a matriz resultante da aplicação do filtro max pooling com $f = 3$ e $s = 1$

1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9

(5×5)

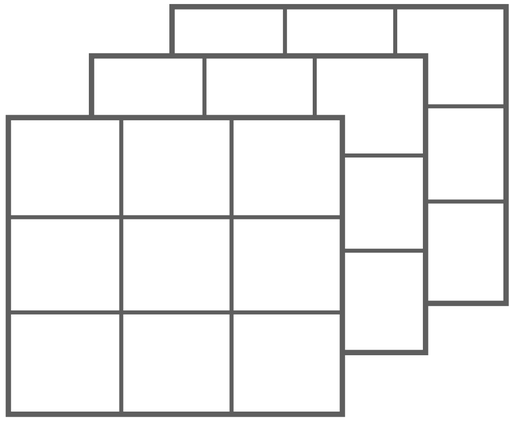
Camadas de Pooling em Volumes

Os filtros de pooling são aplicados de forma independente para cada canal.



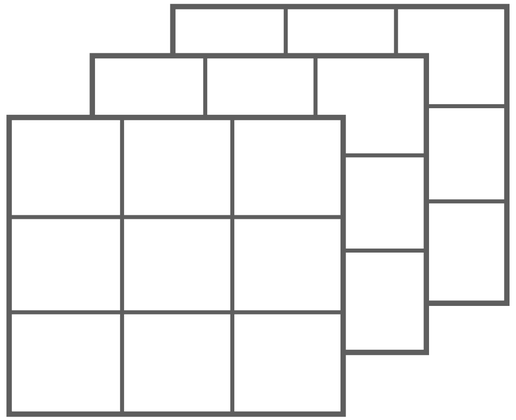
$(5 \times 5 \times 3)$

$*$



$(3 \times 3 \times 3)$

$=$

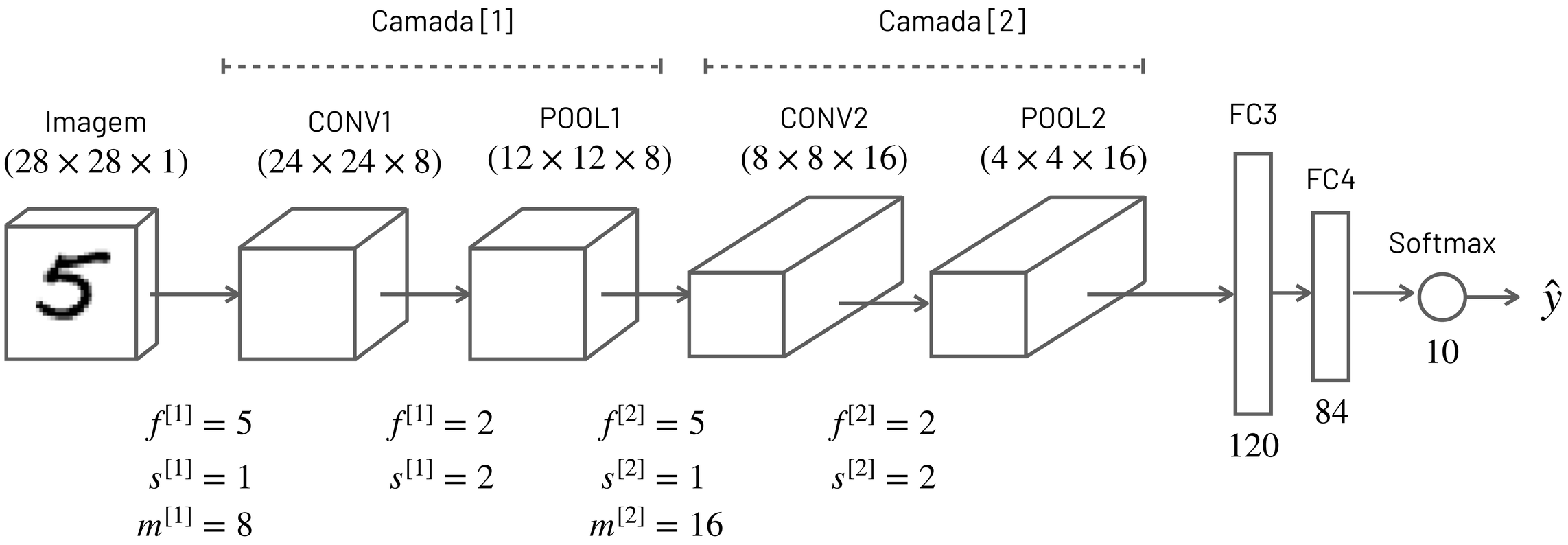


$(3 \times 3 \times 3)$

Pooling em volumes não altera o número de canais!

CNN com Camadas de Pooling

LeNet-5

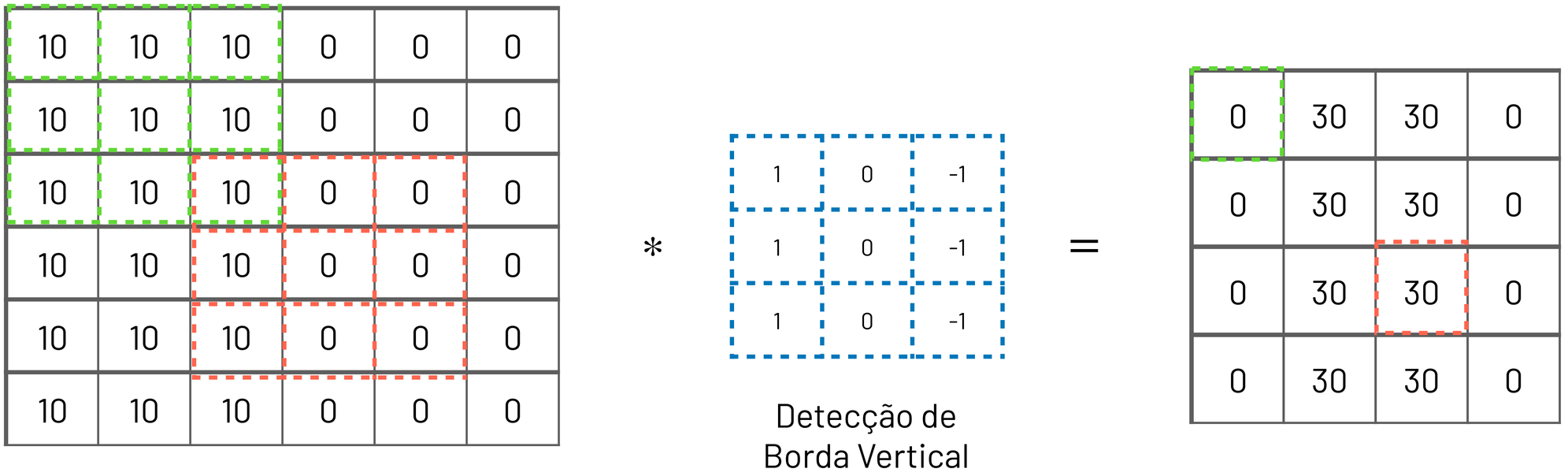


CNN com Camadas de Pooling

	Dimensões da Ativação	Tamanho da Ativação	Número de Parâmetros
Entrada	(28, 28, 1)	784	0
CONV1 (f=5, s=1)	(24, 24, 8)	4608	208
POOL1	(12, 12, 8)	1152	0
CONV2 (f=5, s=1)	(8, 8, 16)	1024	3208
POOL2	(4, 4, 16)	256	0
FC3	(120, 1)	120	30840
FC4	(84, 1)	84	10164
Softmax	(10,1)	10	850

Porque Convoluções?

Redução do número de parâmetros



Detecção de
Borda Vertical

Compartilhamento de parâmetros
Um filtro (e.g., detector de bordas) que funciona bem em uma parte da imagem, provavelmente funciona bem em outra parte da imagem.

Conexões Esparsas
Uma saída depende de apenas um número pequeno de entradas.

Próxima aula

A12: Estudo de Casos de CNNs

Resnet, Inception Network, MobileNet e Efficient Net.