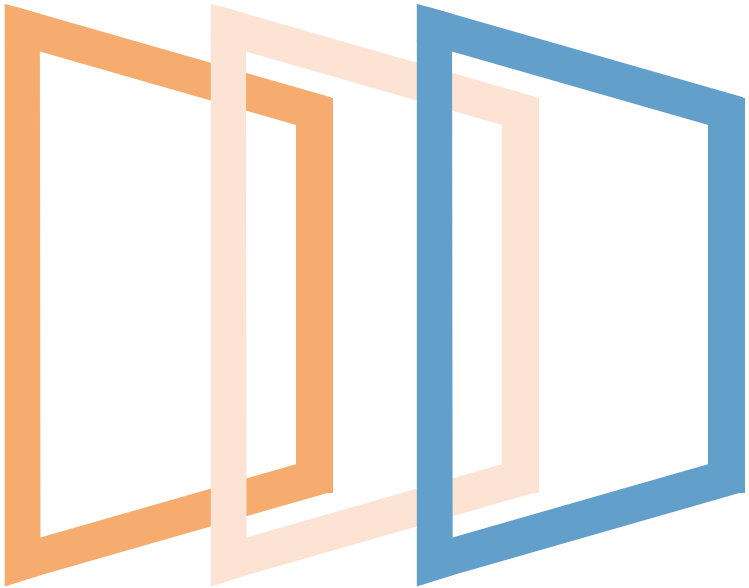
Redes Neurais Intermediário

Diego Alexandre

Práticas Tecnológicas, 11.11.2024



minsait

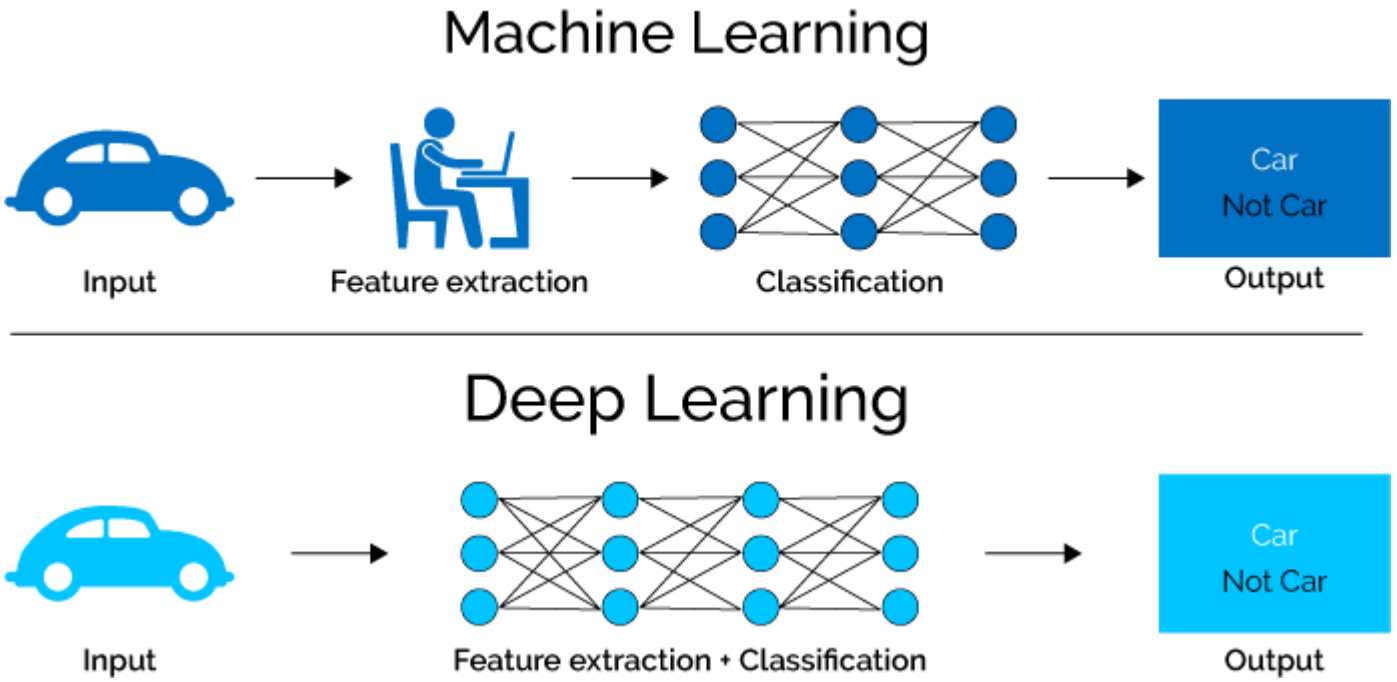
An Indra company

Índice

Machine Learning x Deep Learning

01

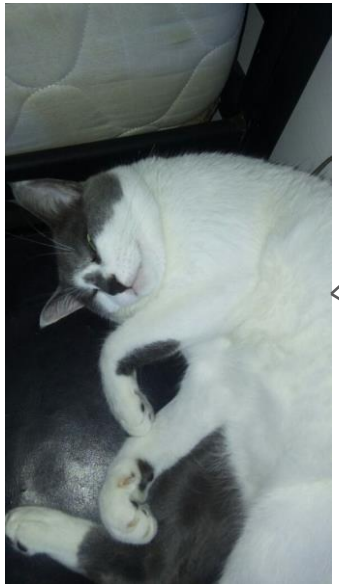
Machine Learning x Deep Learning



Visão Computacional

02

Visão



- Gato
- Branco e cinza
- Dormindo

```
>>> a = cv2.imread('img.jpg
>>> a
array([[[ 87,  93,  82],
        [ 88,  94,  83],
        [ 89,  95,  84],
        ...,
        [ 13,  10,   5],
        [ 13,  10,   5],
        [ 13,  10,   5]],

       [[ 98, 104,  93],
        [101, 107,  96],
        [105, 111, 100],
        ...,
        [ 13,  10,   5],
        [ 13,  10,   5],
        [ 13,  10,   5]],

       [[113, 119, 108],
        [117, 123, 112],
        [122, 128, 117],
        ...,
```

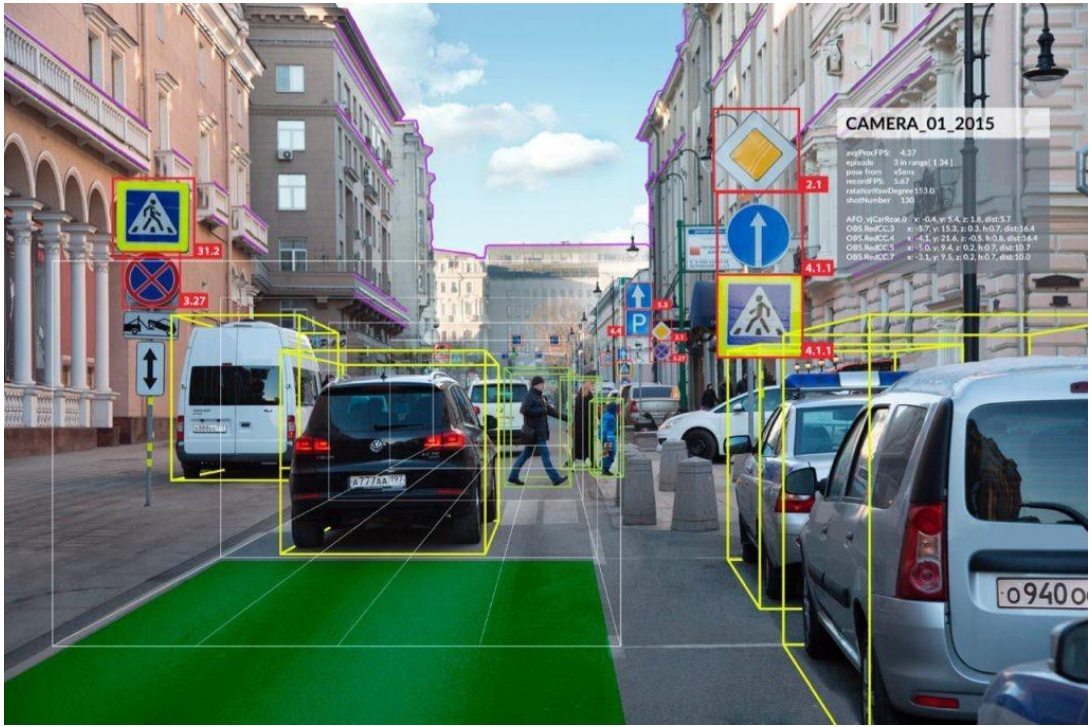
Visão Computacional

- Visão computacional é o campo da ciência que estuda como os computadores podem enxergar e entender o conteúdo de imagens e vídeos

- Isto inclui adquirir, processar e analisar imagens ou vídeos

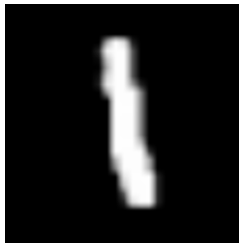
- Problemas ligados à visão computacional muitas vezes são resolvidos facilmente por pessoas, mas não pelas máquinas

Visão Computacional



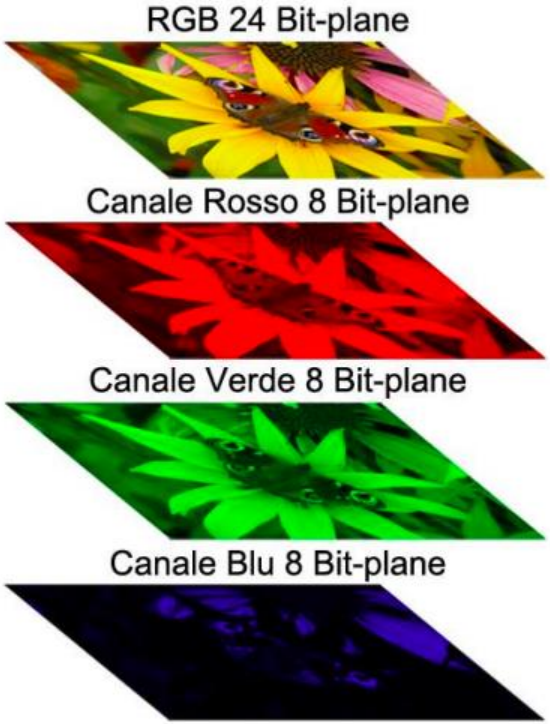
Visão Computacional

Uma imagem é representada como uma matriz



0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0,6	0,8	0	0	0	0	0	0
0	0	0	0	0	0	0,7	1	0	0	0	0	0	0
0	0	0	0	0	0	0,7	1	0	0	0	0	0	0
0	0	0	0	0	0	0,5	1	0,4	0	0	0	0	0
0	0	0	0	0	0	0	1	0,4	0	0	0	0	0
0	0	0	0	0	0	0	1	0,4	0	0	0	0	0
0	0	0	0	0	0	0	1	0,7	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0,9	1	0,1	0	0	0	0
0	0	0	0	0	0	0	0,3	1	0,1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Imagens rgb



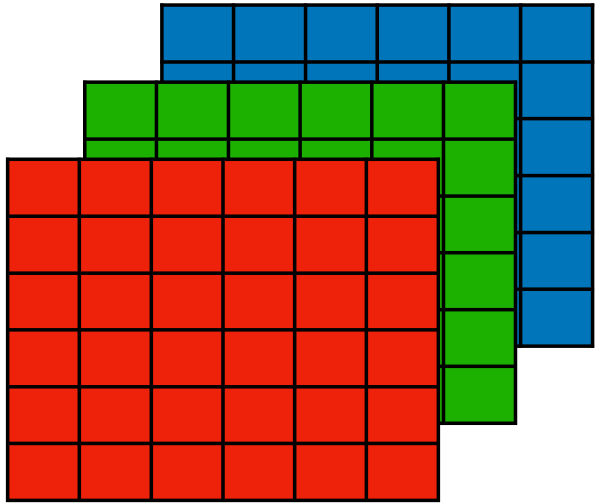


Imagem A x L x 3
Imagens A x L x C
C: no. de canais, ou profundidade

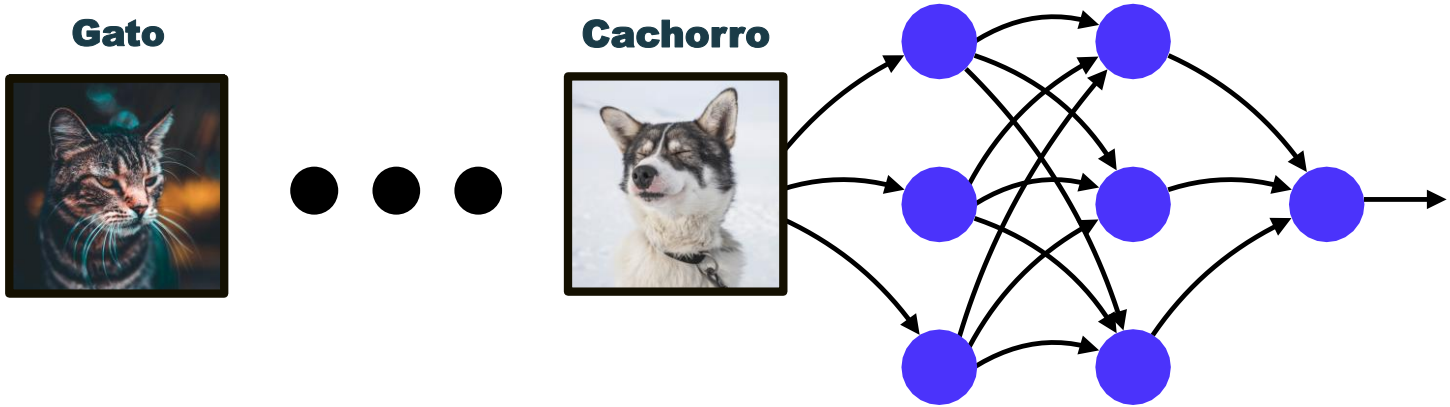
Classificação de Imagens

03

Classificação de imagens

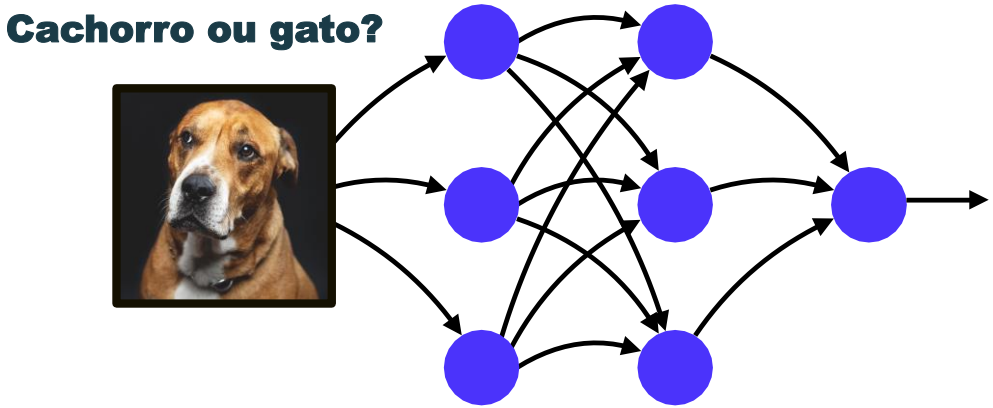
Classificação de imagens

Treinamos uma rede neural com várias imagens de cachorros e gatos



Classificação de imagens

Para que possamos classificar novas entradas

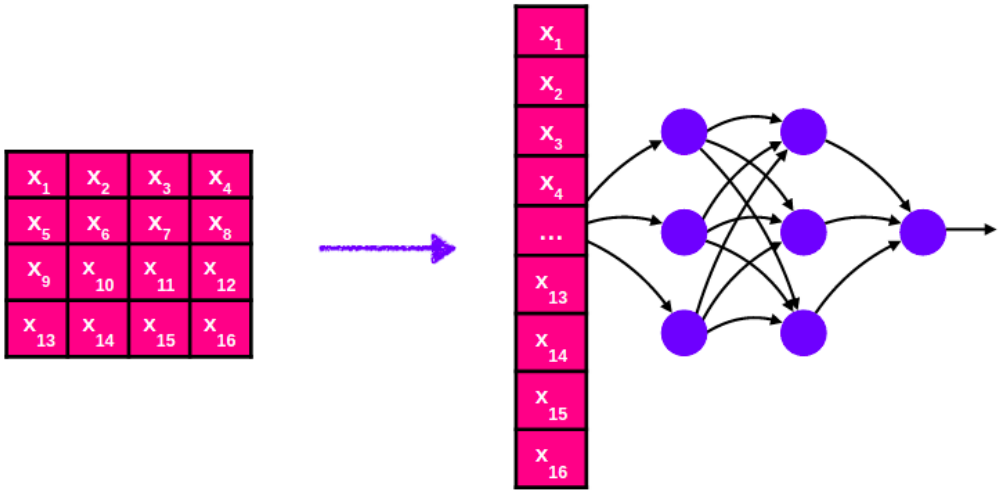


Classificação de imagens

- Como passar uma imagem para uma rede neural?
- A rede recebe entradas numéricas, ela não recebe uma matriz

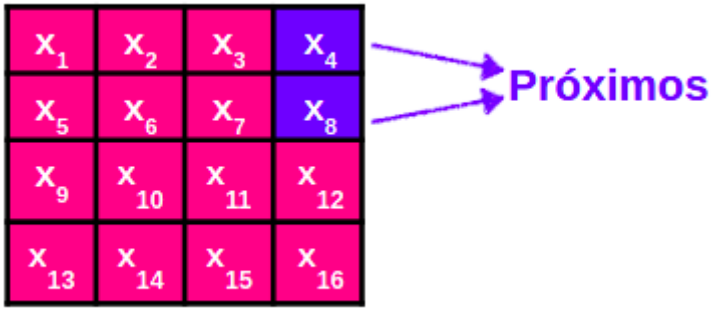
Classificação de imagens

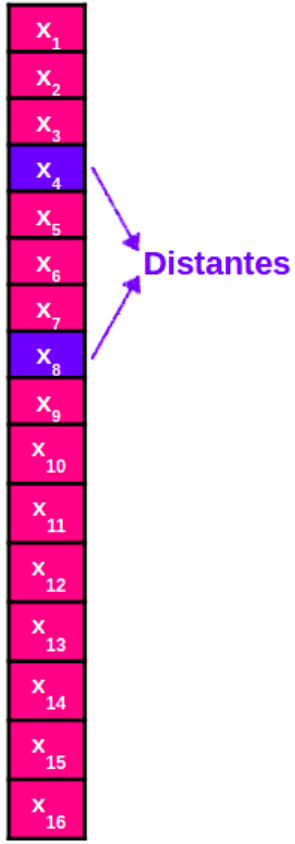
Podemos passar cada valor da matriz como uma entrada



Classificação de imagens

Uma limitação dessa abordagem é que perdemos a noção espacial da imagem



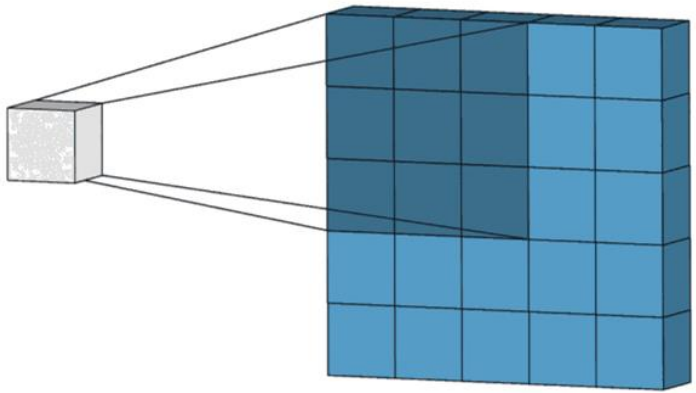


Classificação de imagens

- Outra limitação é que o número de entradas e pesos será muito grande
- Uma imagem colorida 1000x1000 tem 3 milhões de valores
- (A x L x 3)
- Se a primeira camada intermediária possuir 10 neurônios, teremos 30 milhões de pesos só na primeira camada!

Classificação de imagens - Visão Computacional

- ➔ Uma operação linear para reconhecer padrões
- ➔ Operação do filtro/kernel sobre a imagem/sinal
- ➔ Disposição espacial dos dados se transforma em informação



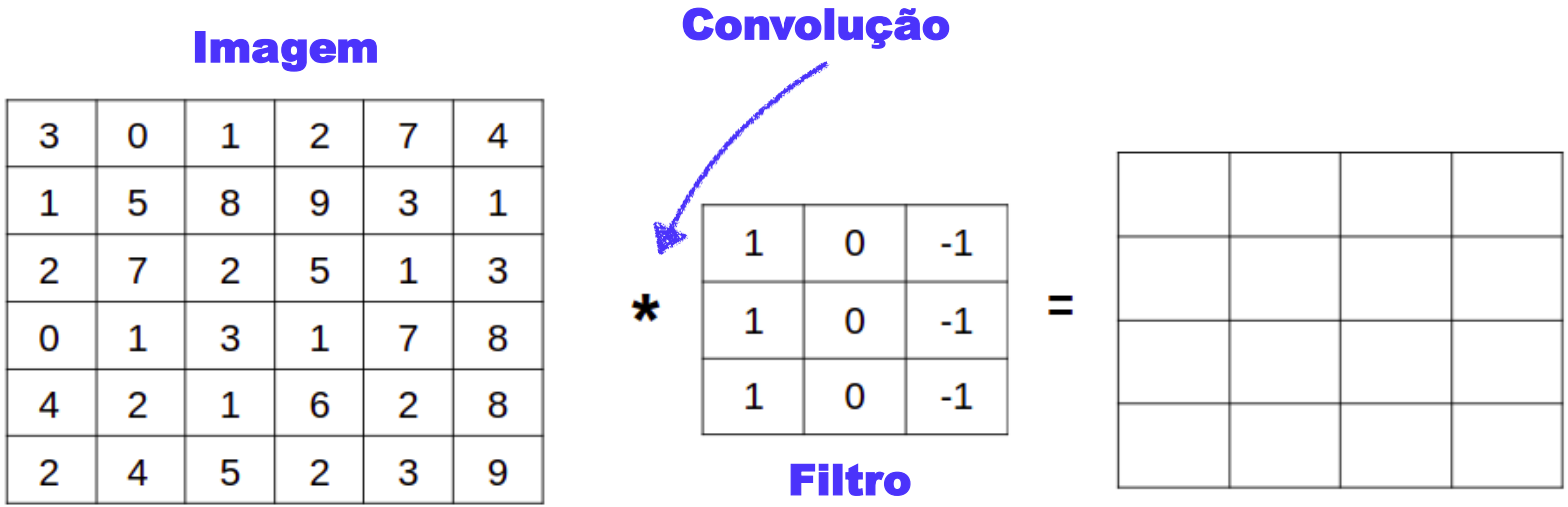
Classificação de imagens

Podemos processar previamente as imagens e depois passar para rede neural:

- ➔ Detectar arestas
- ➔ Detectar cantos
- ➔ Detectar regiões

Detecção de arestas

Detecção de arestas



Operação com filtros

O mesmo filtro é “aplicado” sobre o campo receptivo (CR) que desliza sobre a imagem

A cada iteração, são somados os produtos dos pares (sum(CR x filtro)) gerando um novo valor para a matriz resultante

Detecção de arestas

Campo receptivo

Imagem

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

Convolução

*

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

Filtro

=

Detecção de arestas - Convolução 2D

3x1

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1 + 2x1

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1 + 2x1 + 0x0

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1 + 2x1 + 0x0 + 5x0

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1 + 2x1 + 0x0 + 5x0 + 7x0

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

$$3x1 + 1x1 + 2x1 + 0x0 + 5x0 + 7x0 + 1x-1$$

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1 + 2x1 + 0x0 + 5x0 + 7x0 + 1x-1 + 8x-1

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

3x1 + 1x1 + 2x1 + 0x0 + 5x0 + 7x0 + 1x-1 + 8x-1 + 2x-1

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

Detecção de arestas - Convolução 2D

$$3x1 + 1x1 + 2x1 + 0x0 + 5x0 + 7x0 + 1x-1 + 8x-1 + 2x-1 = -5$$

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

-5			

Detecção de arestas - Convolução 2D

0x1 + 5x1 + 7x1 + 1x0 + 8x0 + 2x0 + 2x-1 + 9x-1 + 5x-1 = -4

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

-5	-4		

Detecção de arestas - Convolução 2D

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

Detecção de arestas - Convolução 2D

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

Detecção de arestas - Convolução 2D

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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

=

Detecção de arestas - Convolução 2D

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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

=

Detecção de arestas - Convolução 2D

10x1 + 10x1 + 10x1 + 10x0 + 10x0 + 10x0 + 10x-1 + 10x-1 + 10x-1 = 0

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

Detecção de arestas - Convolução 2D

10x1 + 10x1 + 10x1 + 10x0 + 10x0 + 10x0 + 10x0 + 10x0 + 10x0 = 30

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

Detecção de arestas - Convolução 2D

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

Detecção de arestas

Filtros diferentes detectam características diferentes:

Arestas verticais

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

Arestas horizontais

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

Redes Neurais Convolucionais

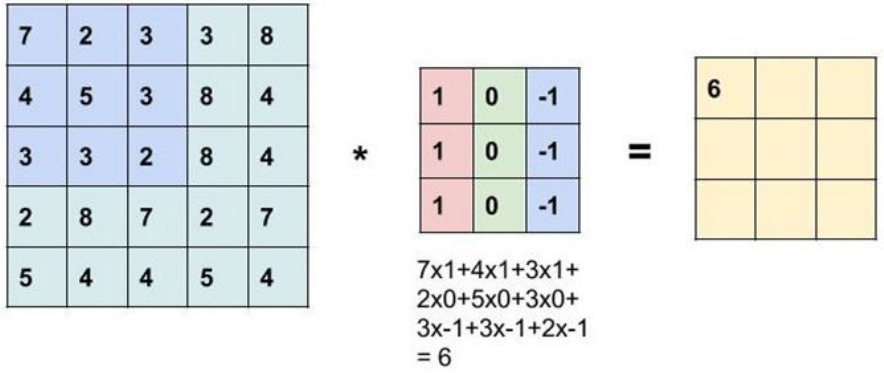
04

Redes neurais convolucionais

$$D_{(0,0)} = (S_{(0,0)}*W_{(0,0)}) + (S_{(1,0)}*W_{(1,0)}) + (S_{(2,0)}*W_{(2,0)}) +$$

$$(S_{(0,1)}*W_{(0,1)}) + (S_{(1,1)}*W_{(1,1)}) + (S_{(2,1)}*W_{(2,1)}) +$$

$$(S_{(0,2)}*W_{(0,2)}) + (S_{(1,2)}*W_{(1,2)}) + (S_{(2,2)}*W_{(2,2)})$$

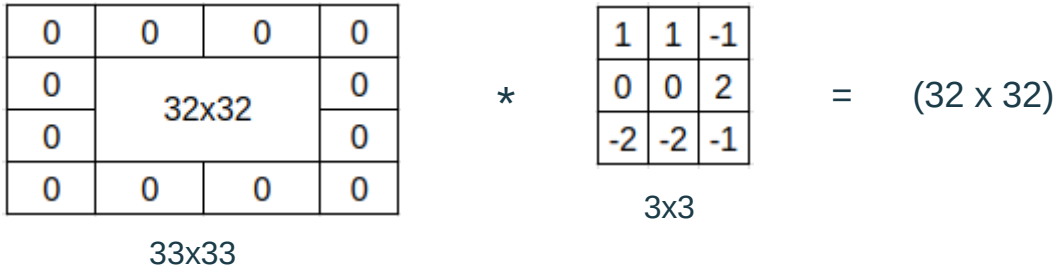


Tamanho Final = (Img - W) + 1 ⇒ (5 - 3) + 1 = 3

Filtro

Padding

Padding = 1; Imagem (32 x 32) -> (33 x 33)

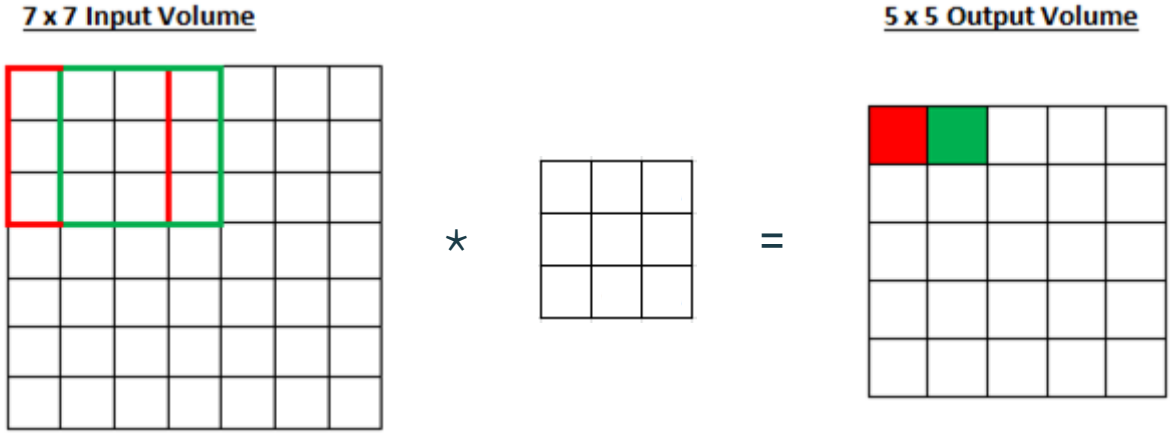


Tamanho Final = (Img - W + 2*P) + 1

Tamanho Final = (32 - 3 + 2*1) + 1 = 32

Stride

Stride = 1 ; Filtro 3x3

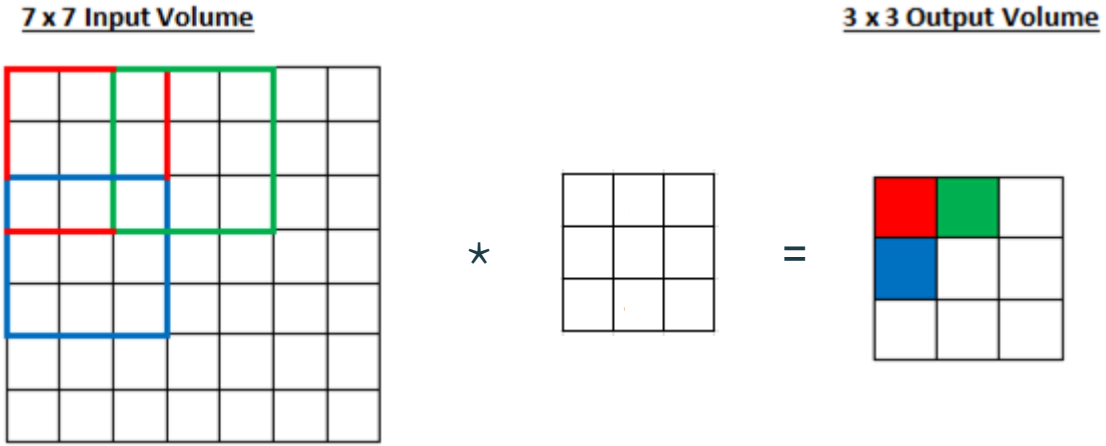


Tamanho Final = [(Img - W + 2*P) / S] + 1

Tamanho Final = [(7 - 3 + 2*0) / 1] + 1 = 5

Stride

Stride = 2; Filtro 3x3



Tamanho Final = [(Img - W + 2*P) / S] + 1

Tamanho Final = [(7 - 3 + 2*0) / 2] + 1 = 3

Batch Normalization

Camada para manter a media próxima a zero e o desvio padrão próximo a 1.

Redes neurais convolucionais

- Podemos tentar criar filtros manualmente escolhendo valores
- Quais filtros criar?
- Como saber se os valores escolhidos são bons? Não sabemos previamente!

Redes neurais convolucionais

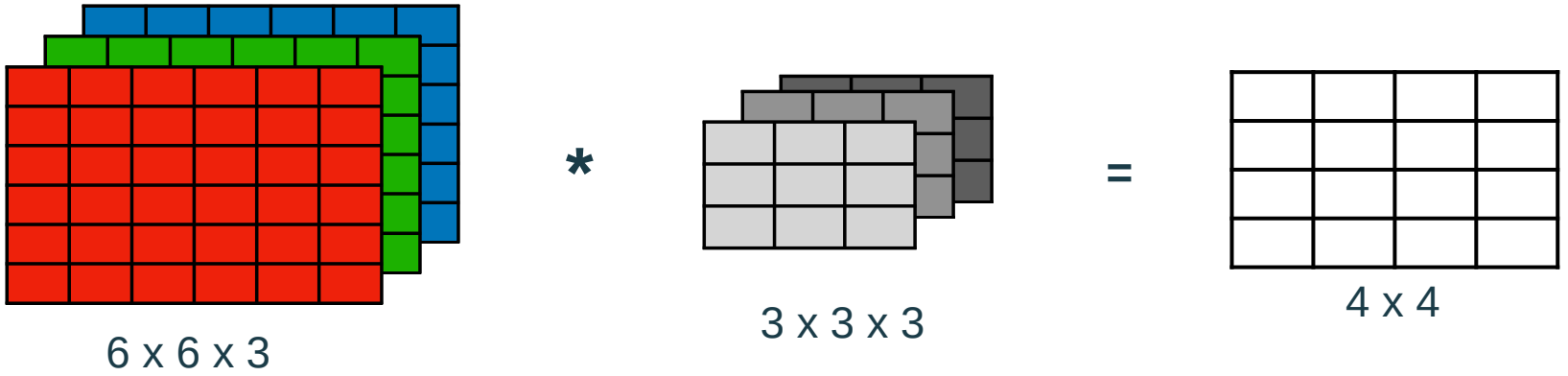
3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

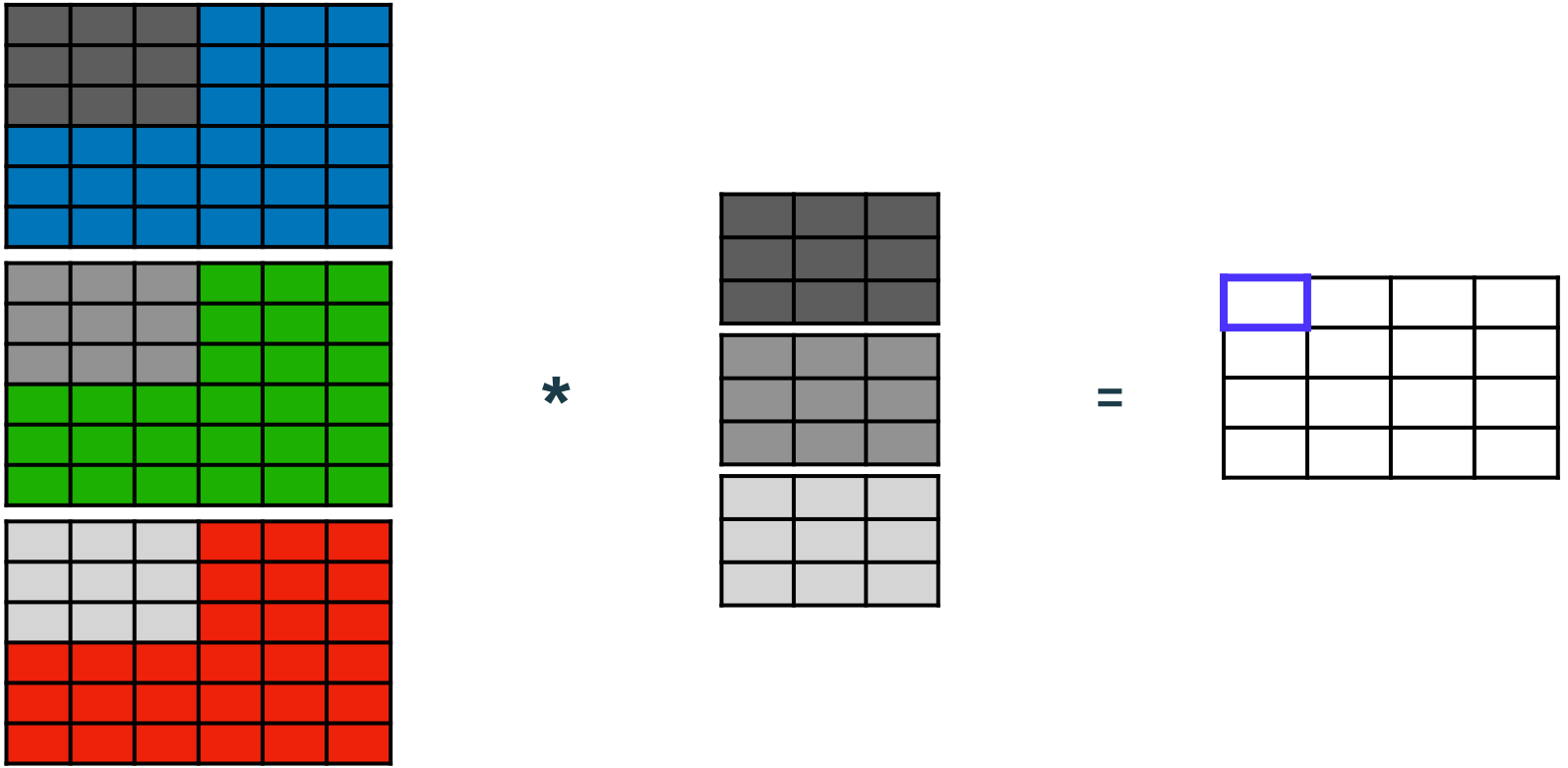
w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Vamos tratar os valores dos filtros como pesos que serão aprendidos pela rede neural através do backpropagation

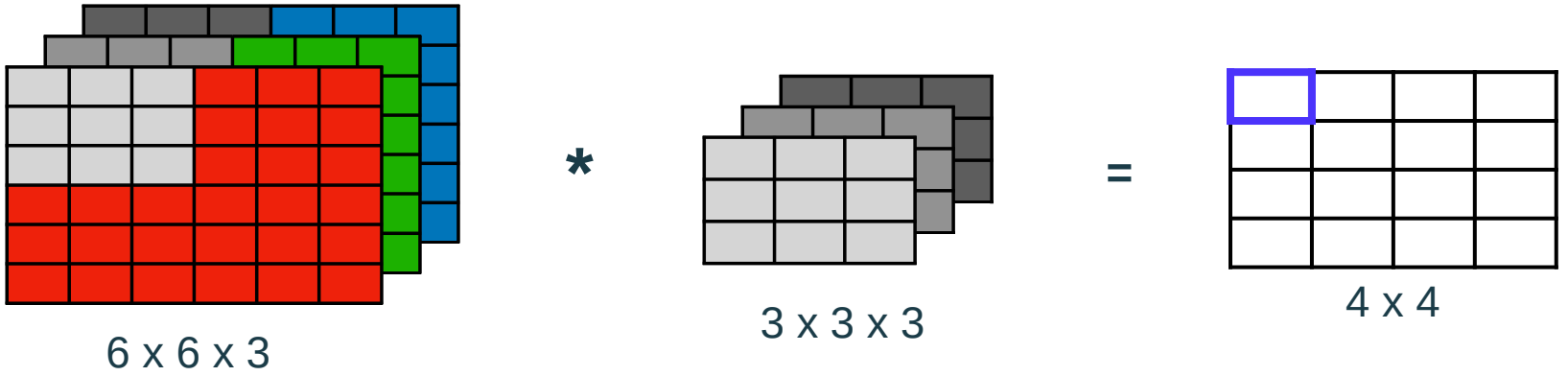
Convolução 3D



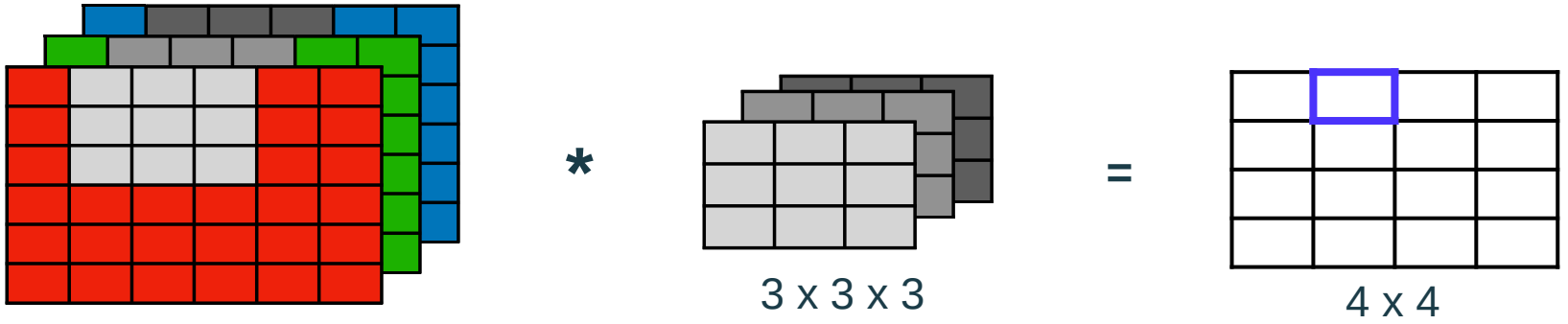
Convolução 3D



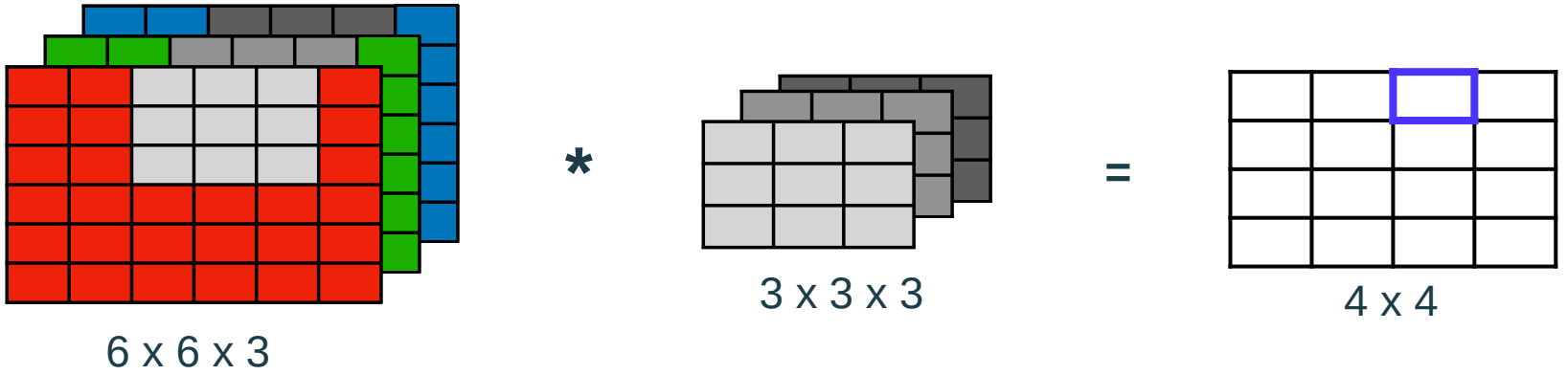
Convolução 3D



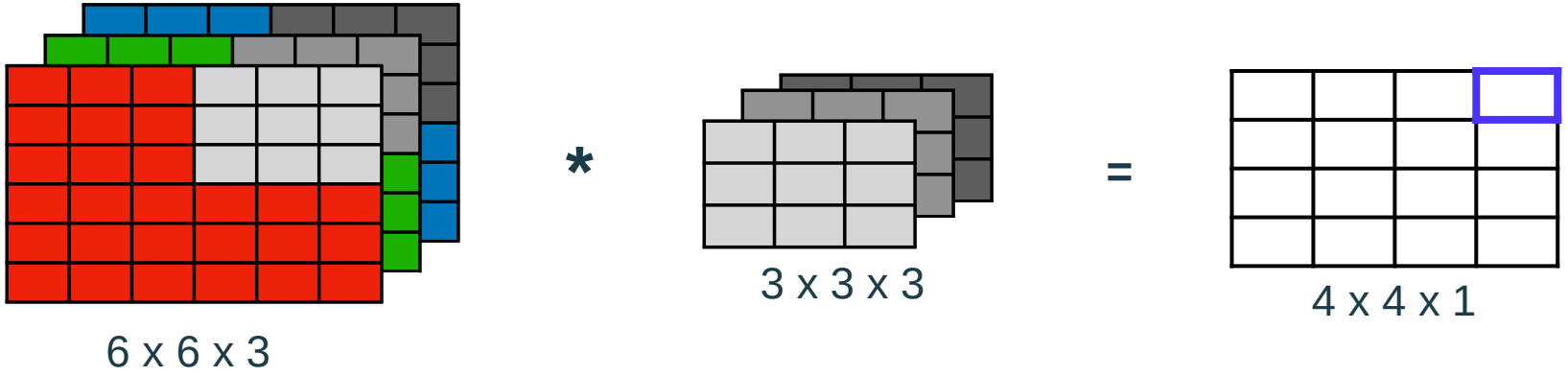
Convolução 3D



Convolução 3D



Convolução 3D



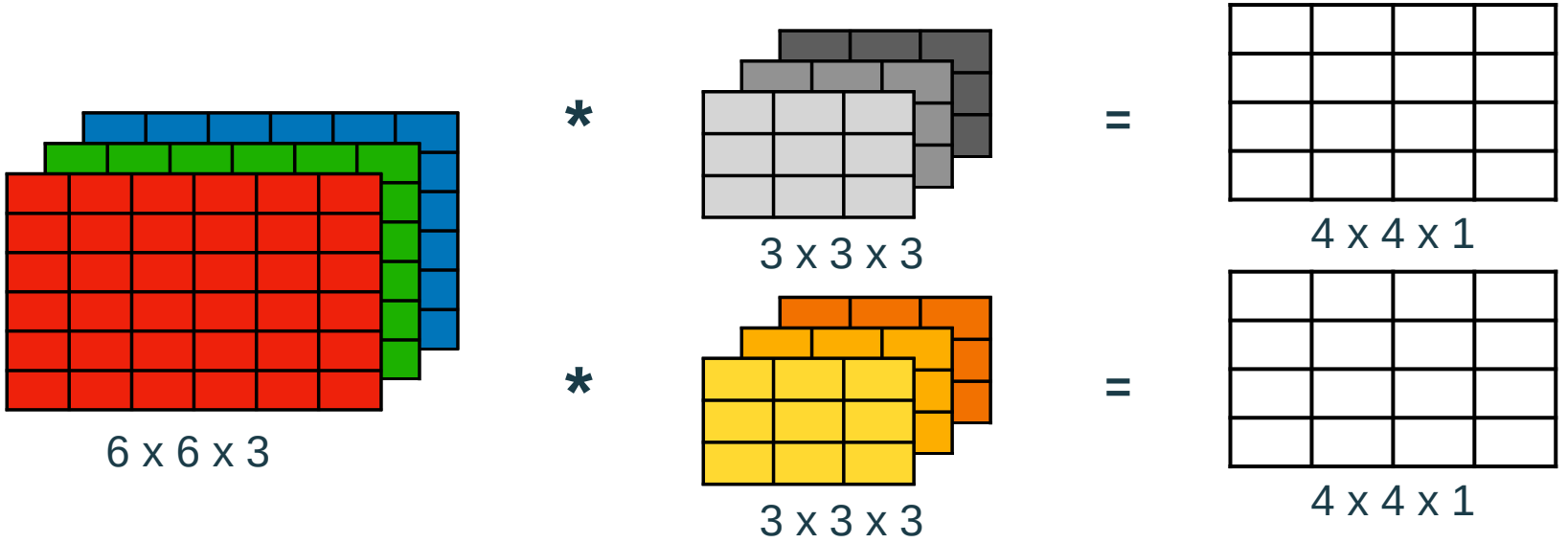
Tamanho Final 3D: $Imagem_{(x,y,c)}$ o N*$Filtros_{(x',y',c)}$ = $Saída_{(x'',y'',N)}$

Importante: x" e y" são calculados como se fosse 2D: (P=0; S=1)

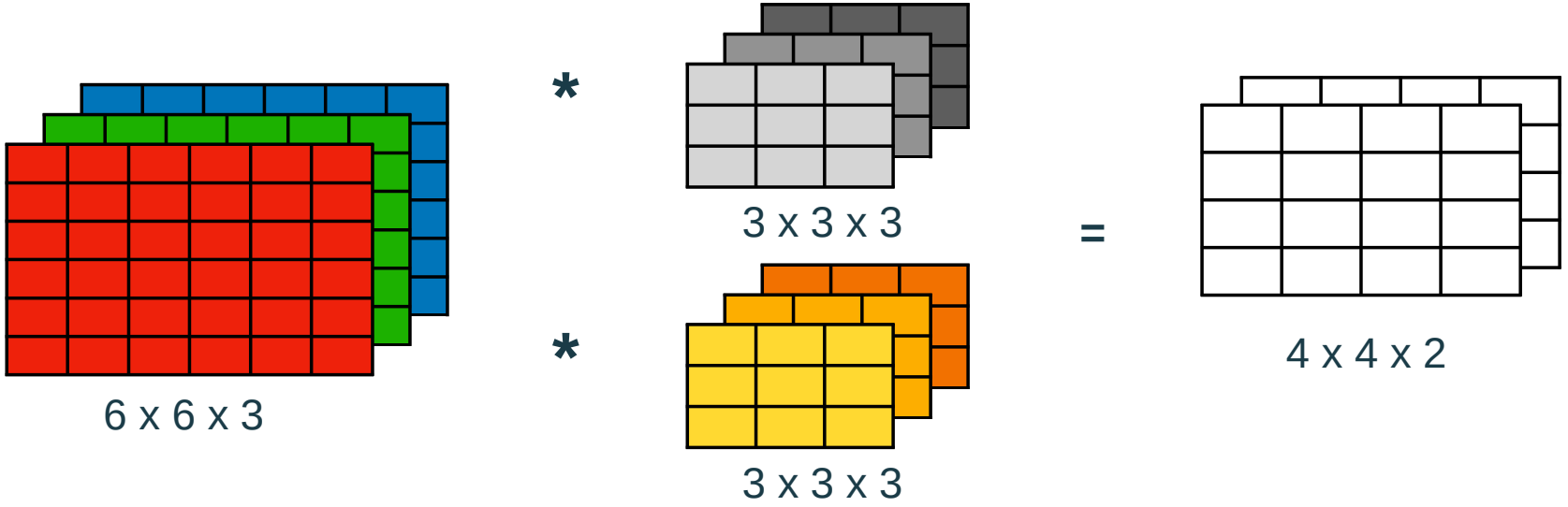
TF = [(Img - W + 2P) / S] + 1 ⟹ [(6 - 3 + 0) / 1] + 1 = 4

Tamanho Final 3D: $Img_{(6, 6, 3)}$ o 1*$W_{(3, 3, 3)}$ = $Saída_{(4, 4, 1)}$

Convolução 3D



Convolução 3D



Tamanho Final 3D: $Imagem_{(x,y,c)}$ o N*$Filtros_{(x',y',c)}$ = $Saída_{(x'',y'',N)}$

x" e y" são definidos pelo mesmo cálculo anterior cujo resultado foi 4

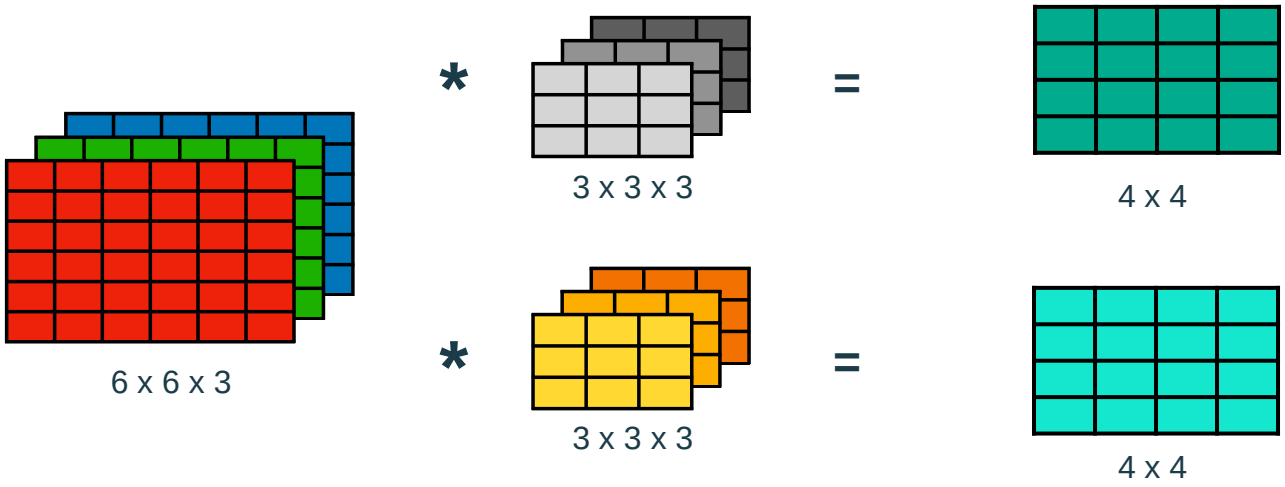
Tamanho Final 3D: $Img_{(6,\ 6,\ 3)}$ o 2*$W_{(3,\ 3,\ 3)}$ = $Saída_{(4,\ 4,\ 2)}$

Redes neurais convolucionais

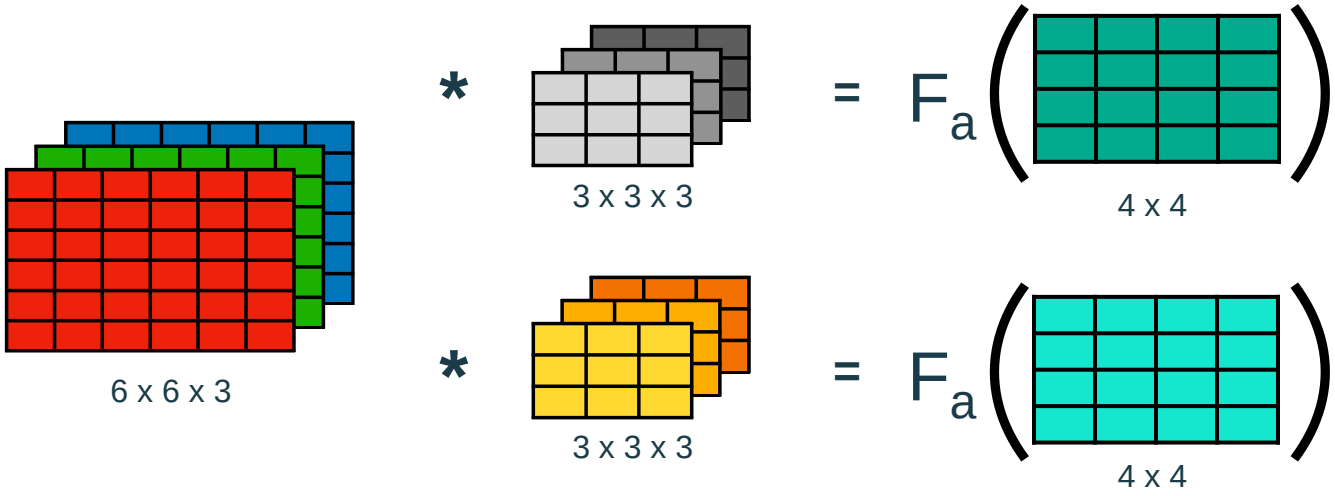
- Temos todas as peças para entender e construir uma rede neural convolucional

- Iniciaremos com uma camada

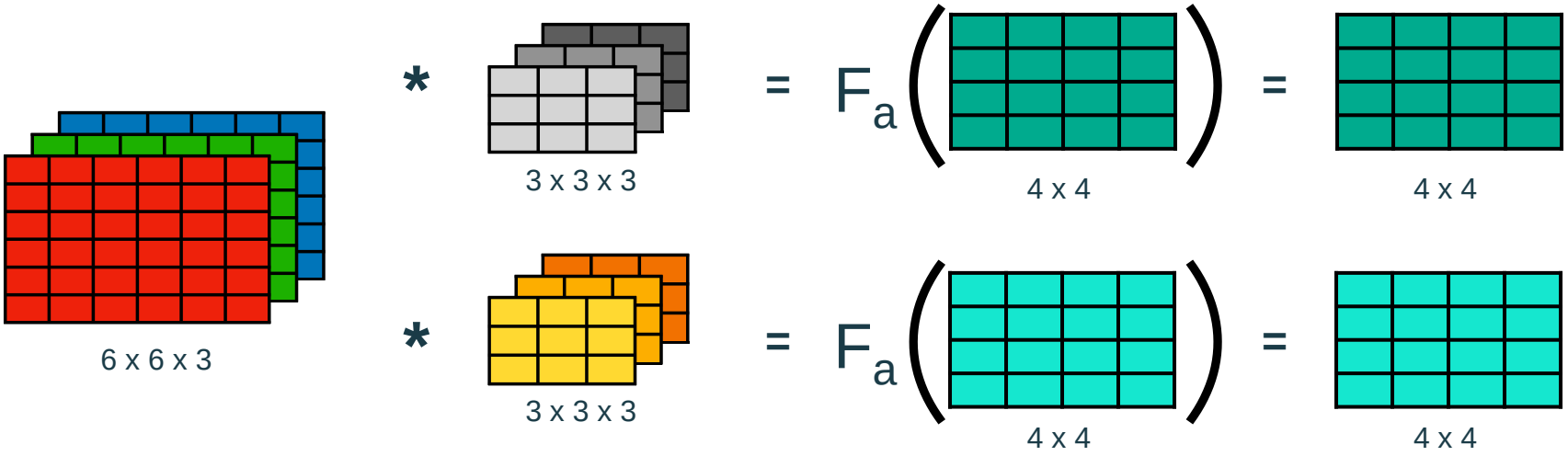
Redes neurais convolucionais



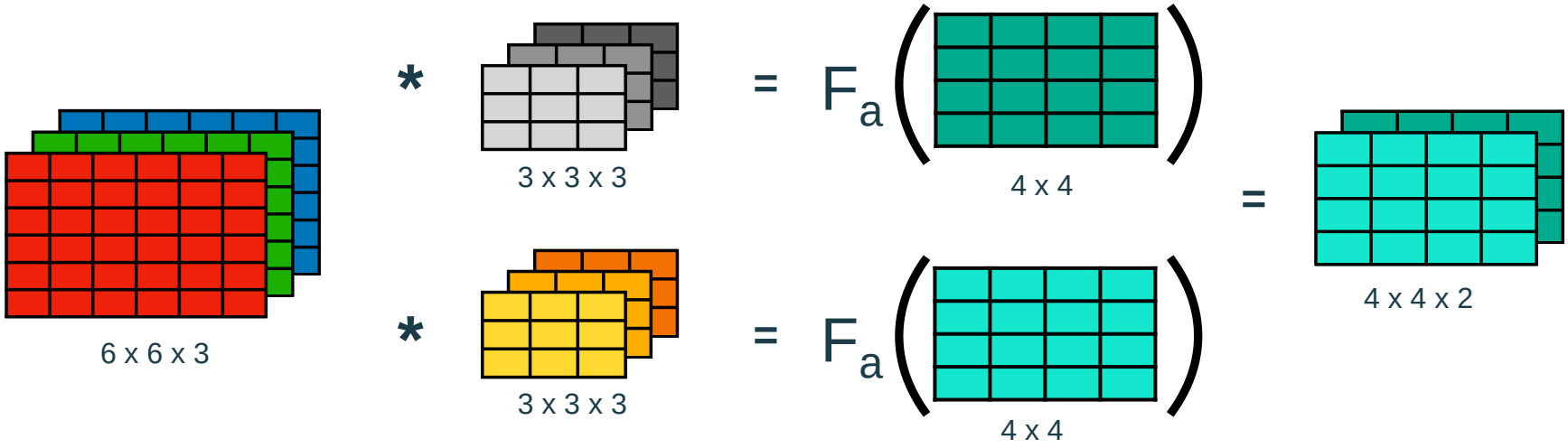
Redes neurais convolucionais



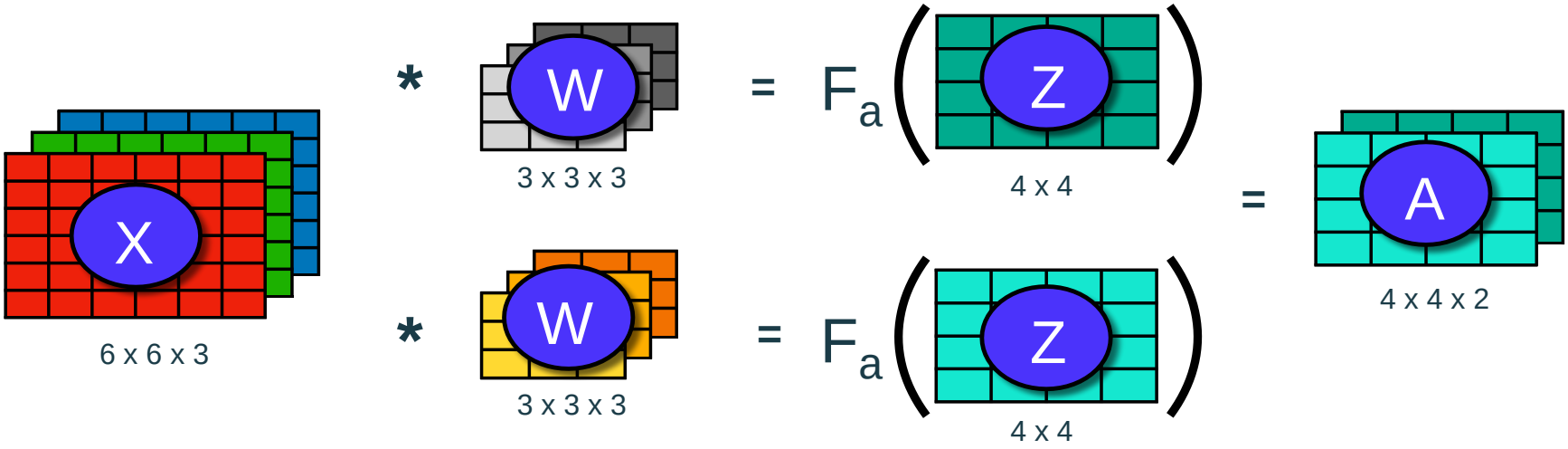
Redes neurais convolucionais



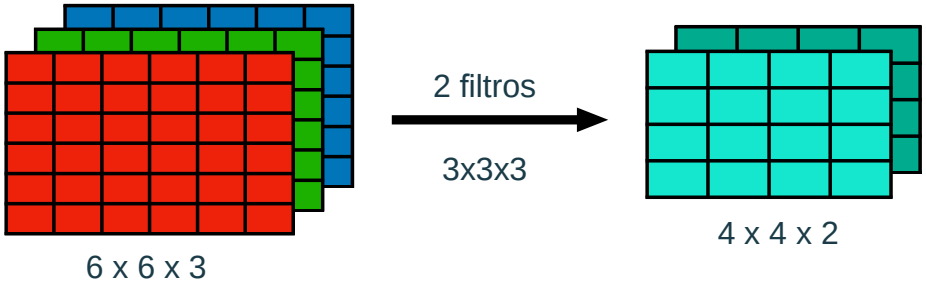
Redes neurais convolucionais



Redes neurais convolucionais

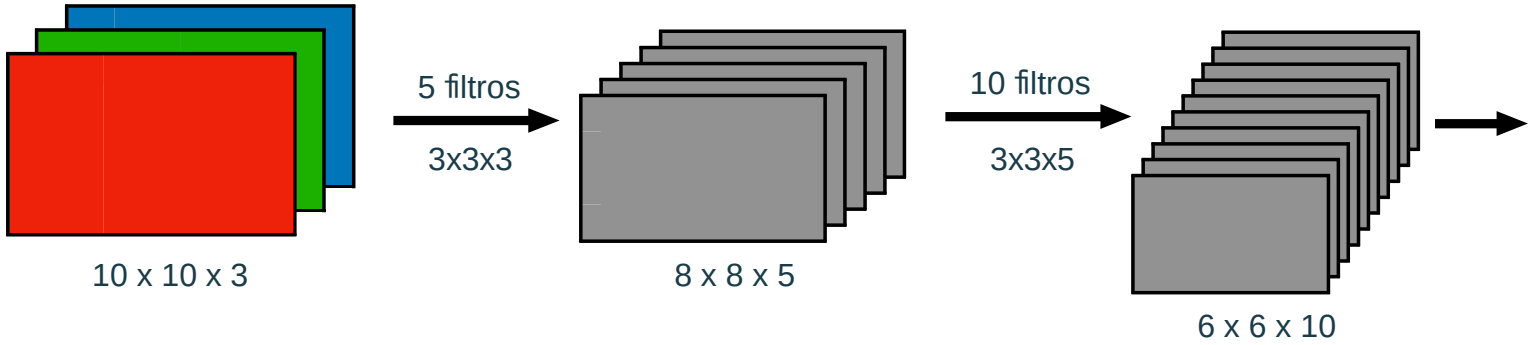


Redes neurais convolucionais



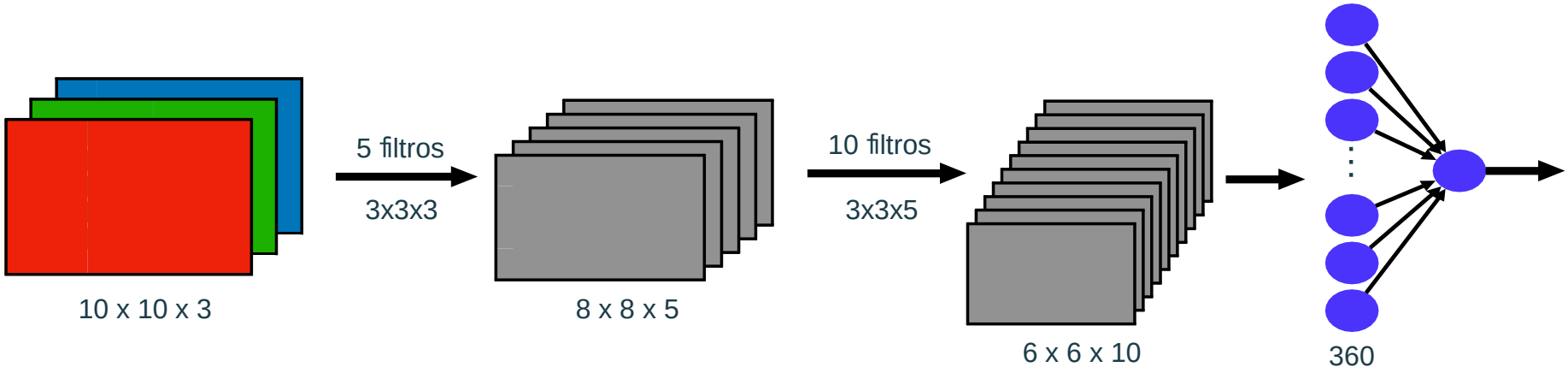
Redes neurais convolucionais

Uma rede convolucional com duas camadas:



Redes neurais convolucionais

No final, usamos uma camada totalmente conectada:



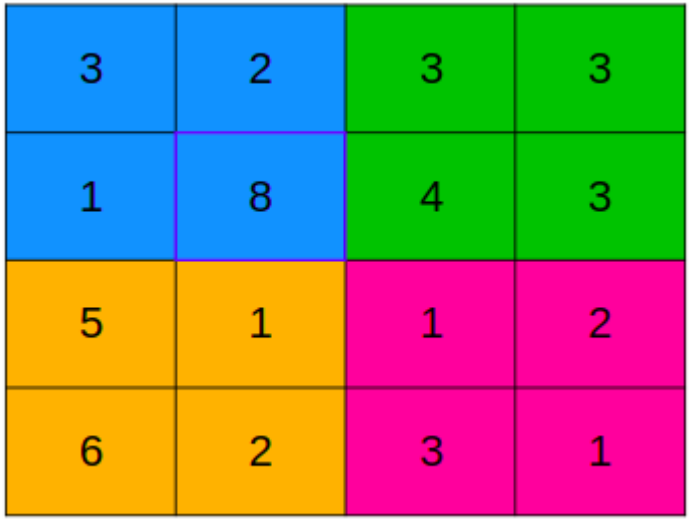
Pooling

- A operação de *pooling* é usada para aumentar a velocidade e diminuir a quantidade de memória usada.

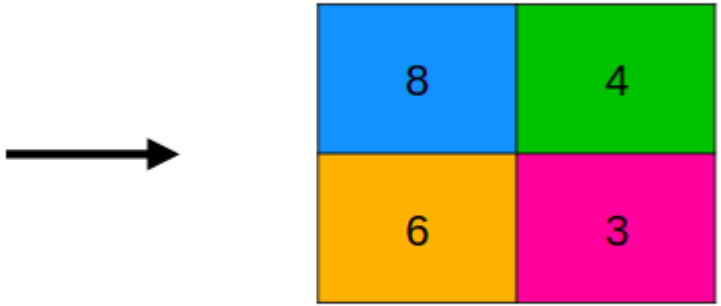
- Ela também realça características encontradas pelos filtros.

Max Pooling

Stride = 2
Filtro 2x2
Tamanho Final = Img / S



4 x 4

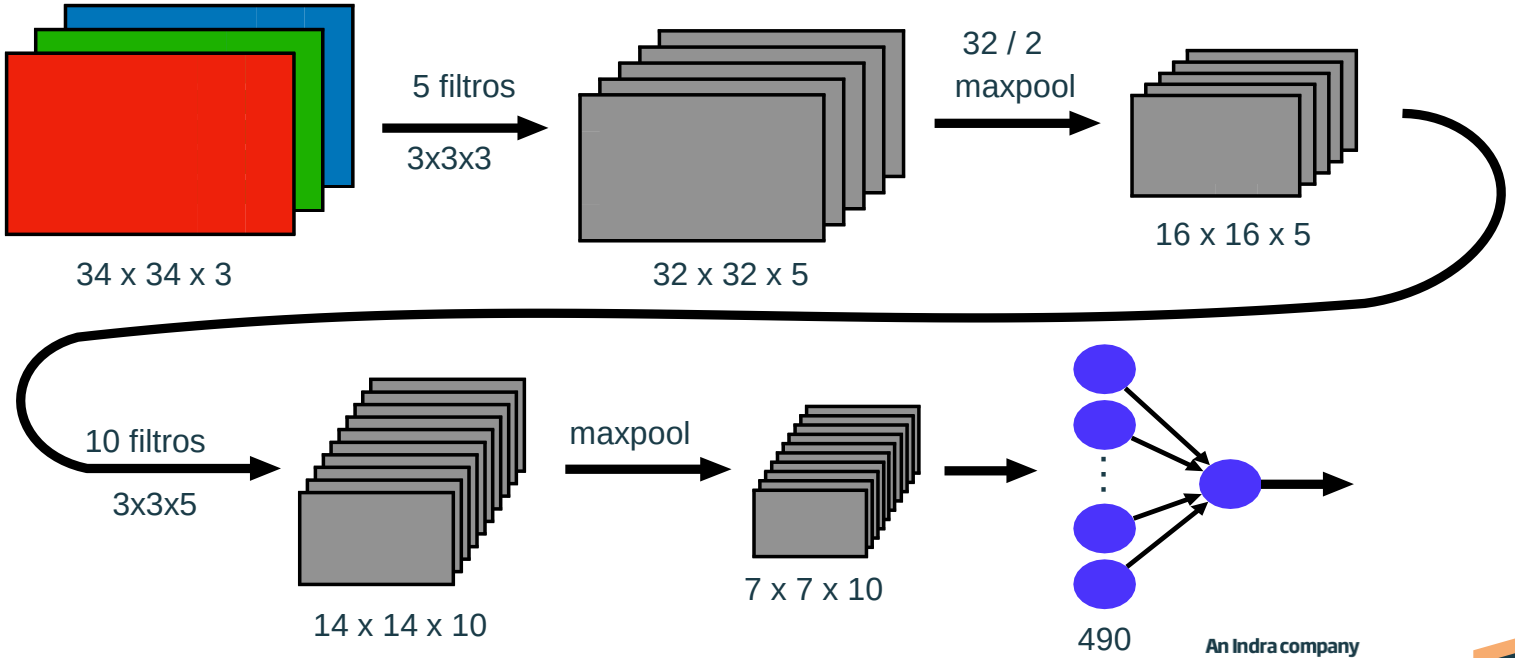


2 x 2

Redes neurais convolucionais

TF 2D: [(34 - 3 + 0) / 1] + 1 = 32

TF 3D: $Img_{(34,34,\ 3)}$ o 5*$W_{(3,\ 3,\ 3)}$ = $Saída_{(32,\ 32,\ 5)}$

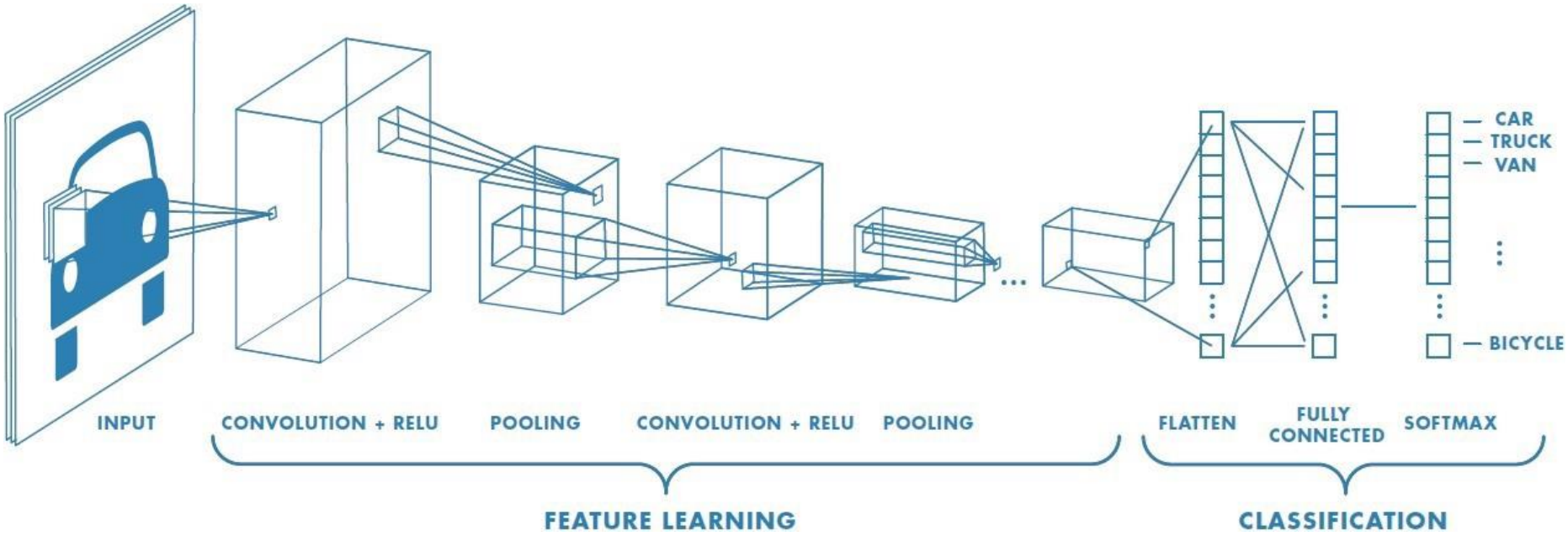


Redes neurais convolucionais

Nas Redes Neurais Convolucionais é muito comum usarmos o padrão:

Convolução ➡ pooling ➡ convolução ➡ pooling ➡ …
➡ convolução ➡ pooling ➡ camada totalmente conectada

Redes neurais convolucionais

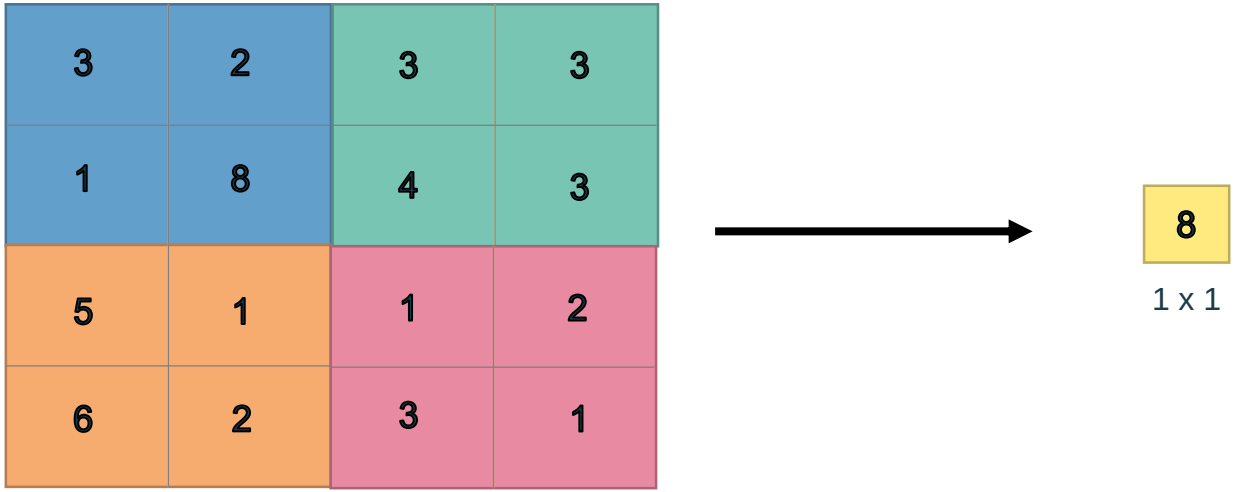


Redes neurais convolucionais

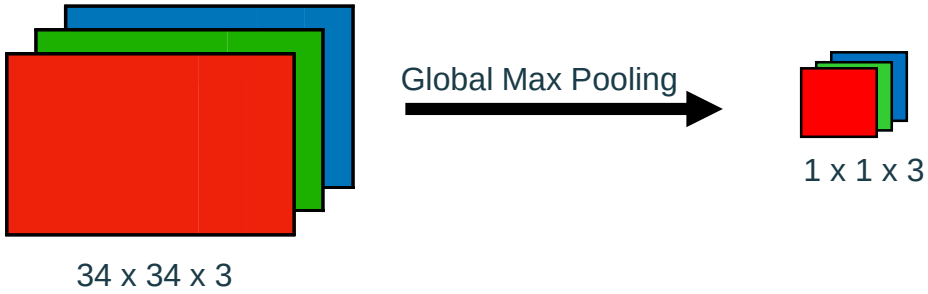
As Redes Neurais Convolucionais, ao longo de suas camadas, vão aprendendo a detectar padrões cada vez mais complexos.

É possível visualizar esses padrões através da criação de imagens artificialmente para maximizar uma das saídas de uma determinada camada.

Global Max Pooling



Global Max Pooling



Conv 1 x 1

Projeção do mapa de atributos

Conv 1 x 1

Redução do mapa de atributos

Conv 1 x 1

Aumento do mapa de atributos

ImageNet

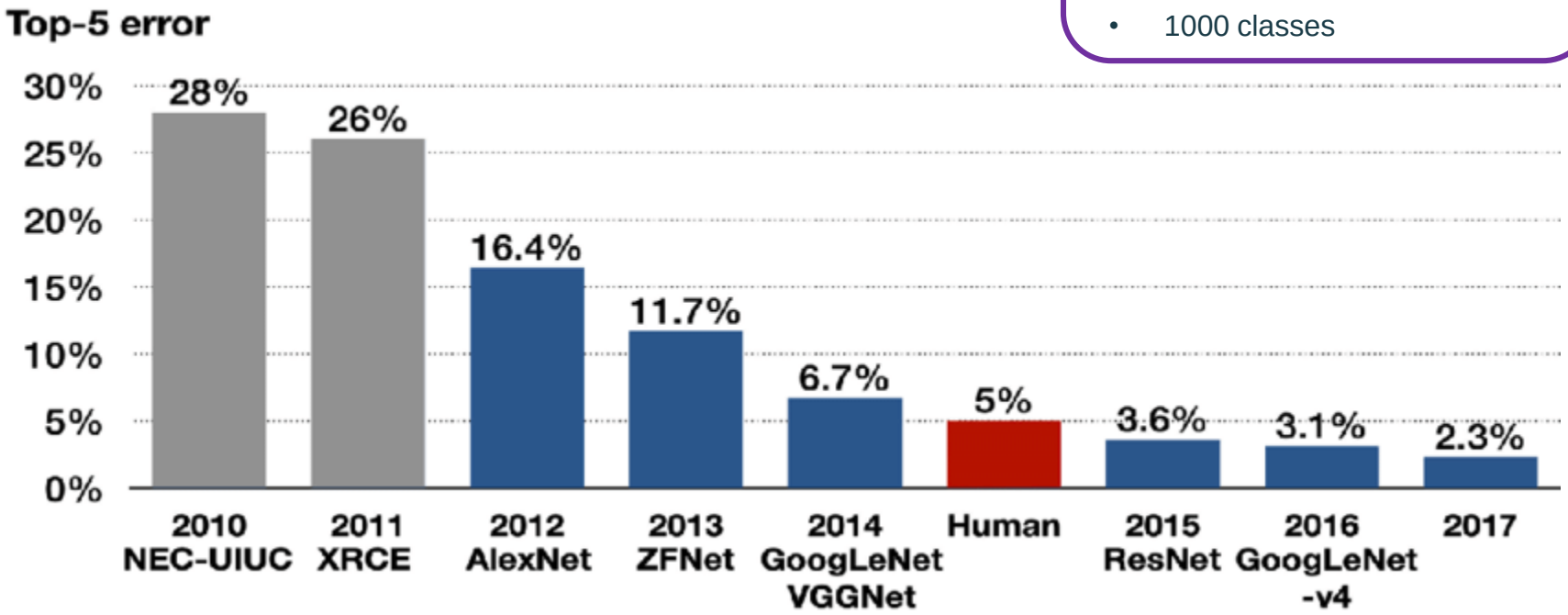
Dataset

- 1 milhão de amostras de treino
- 50 mil validação
- 100 mil teste
- 1000 classes

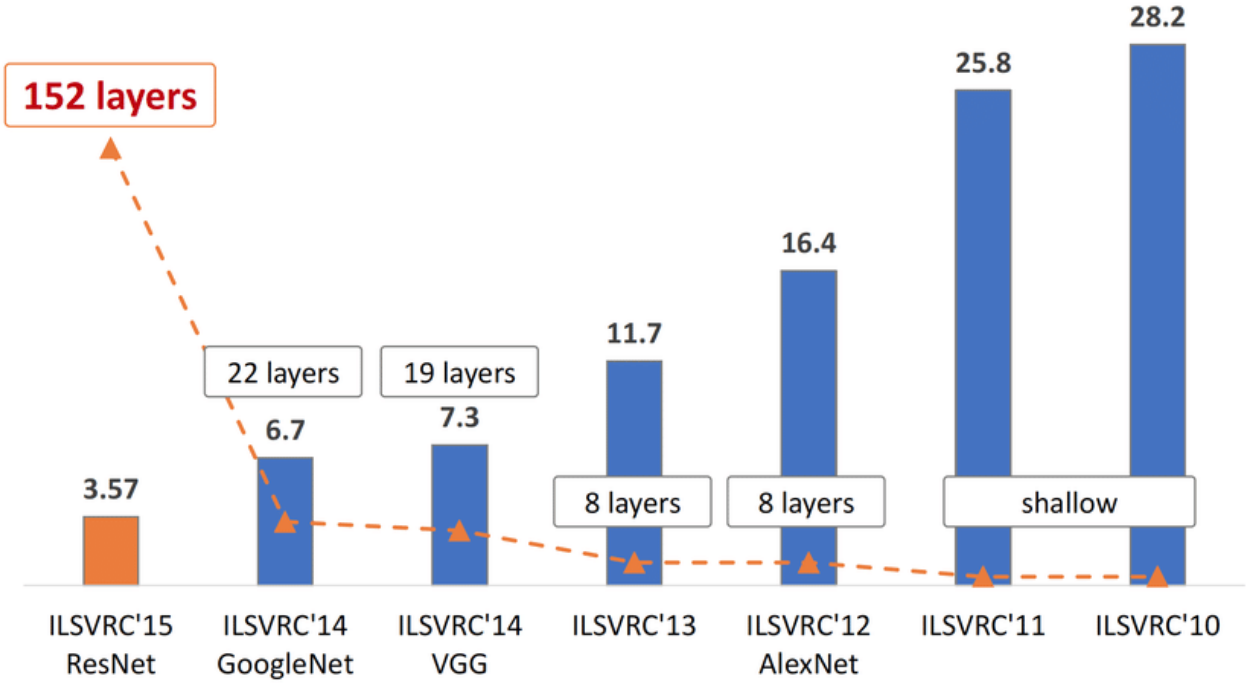
ImageNet

Base de Dados:

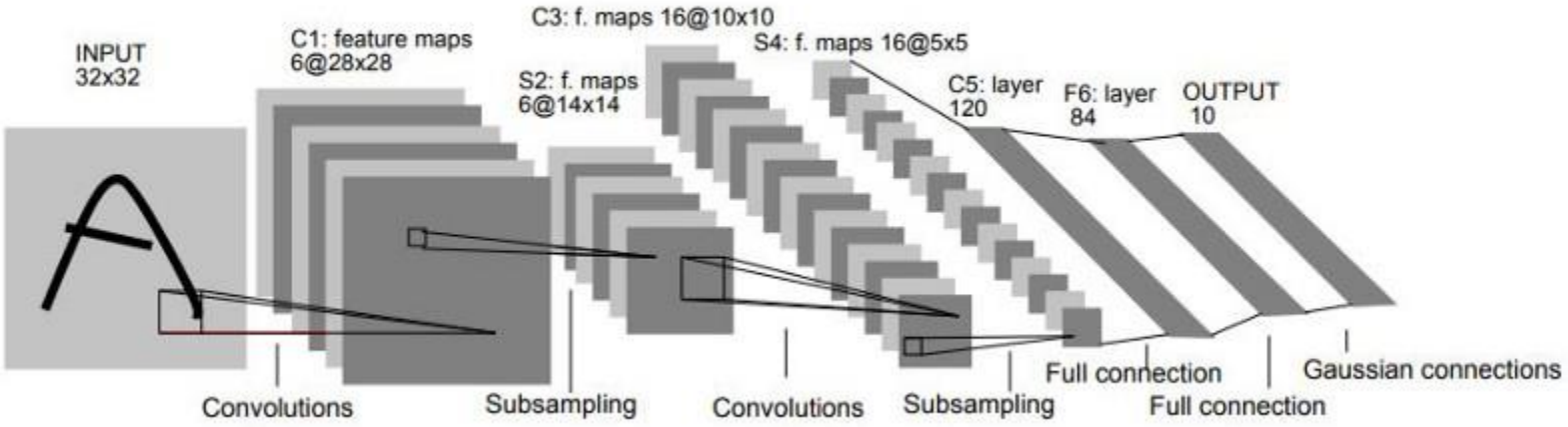
- 1 milhão de amostras de treino
- 50 mil validação
- 100 mil teste
- 1000 classes



ImageNet


152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10

LeNet


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

AlexNet

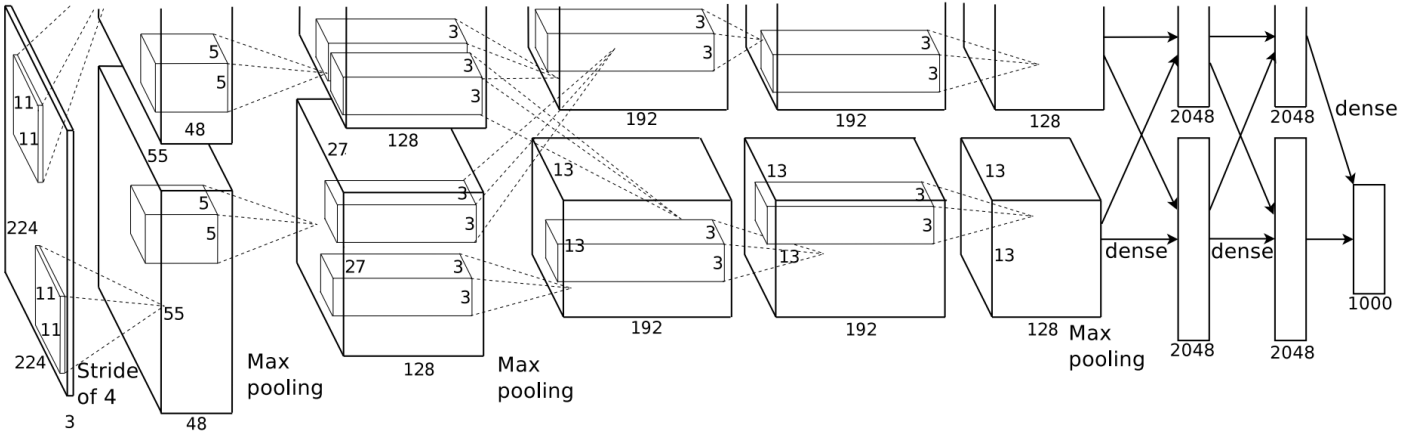
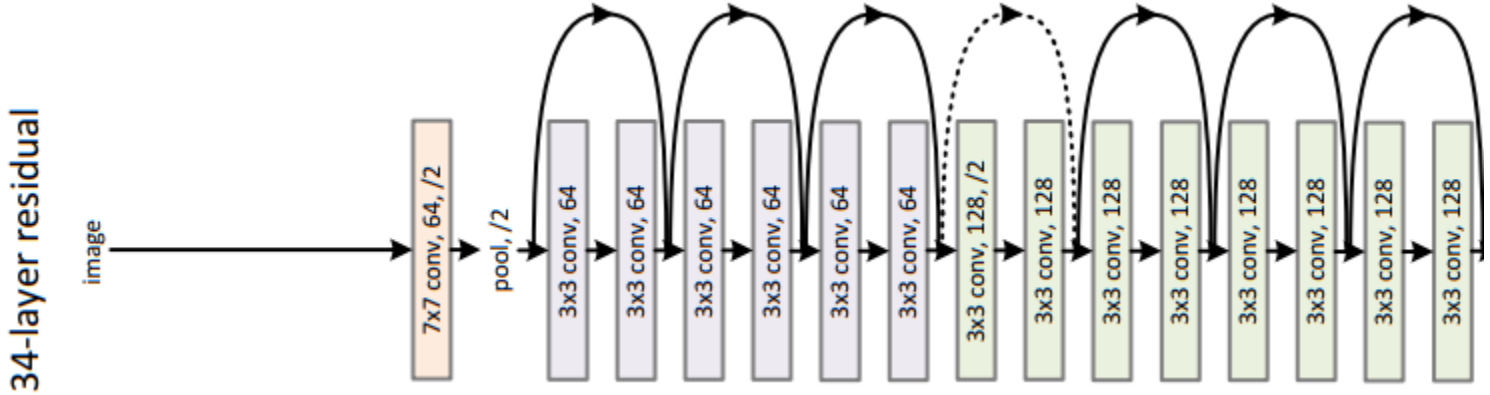
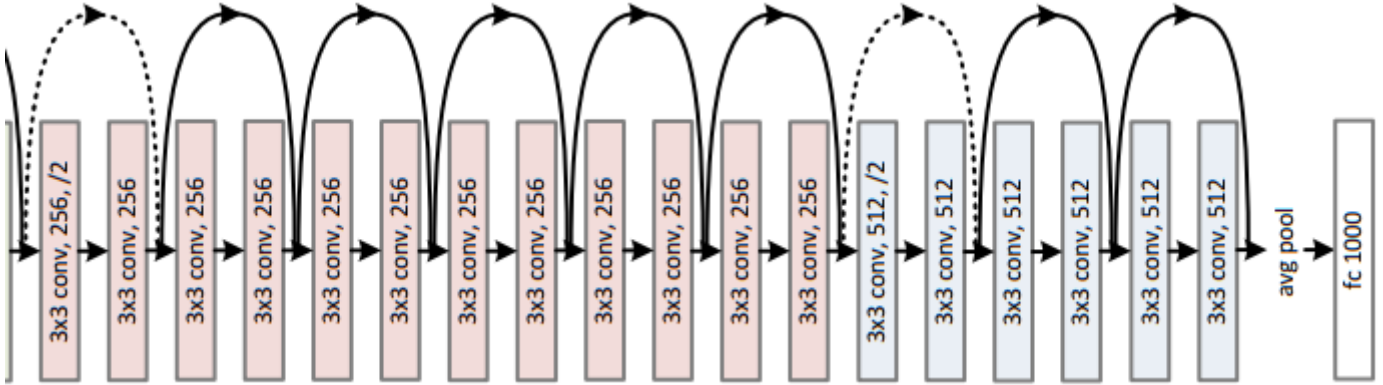


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

ResNet


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128

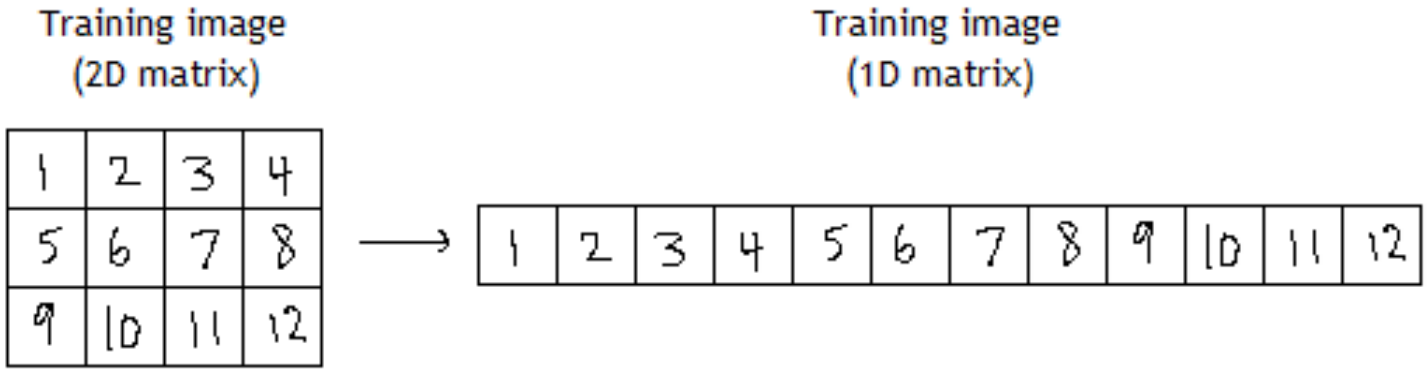
ResNet


3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000

Redes neurais convolucionais em dados categóricos

05

Redes neurais convolucionais 1D



Redes neurais convolucionais 1D

$$4x3 + 1x7 + 3x5 = 34$$

Matriz 1D	4	1	3	8	4	0	3	8	0	7	7	7	1	2
Filtro	3	7	5											
Saída		34												

Redes neurais convolucionais 1D

1x3 + 3x7 + 8x5 = 64

Matriz 1D	4	1	3	8	4	0	3	8	0	7	7	7	1	2
Filtro		3	7	5										
Saída		34	?											

Matriz 1D	4	1	3	8	4	0	3	8	0	7	7	7	1	2
Filtro		3	7	5										
Saída		34	64											

Redes neurais convolucionais 1D

Tamanho Final = [(Matriz - W + 2*P) / S] + 1 (Mesma fórmula da 2D)
Tamanho Final = [(6 - 3 + 2*0) / 1] + 1 = 4

Cálculos:

1º Elemento: 1x2 + 3x0 + 3x1 = 5
2º Elemento: 3x2 + 3x0 + 0x1 = 6
3º Elemento: 3x2 + 0x0 + 1x1 = 7
4º Elemento: 0x2 + 1x0 + 2x1 = 2

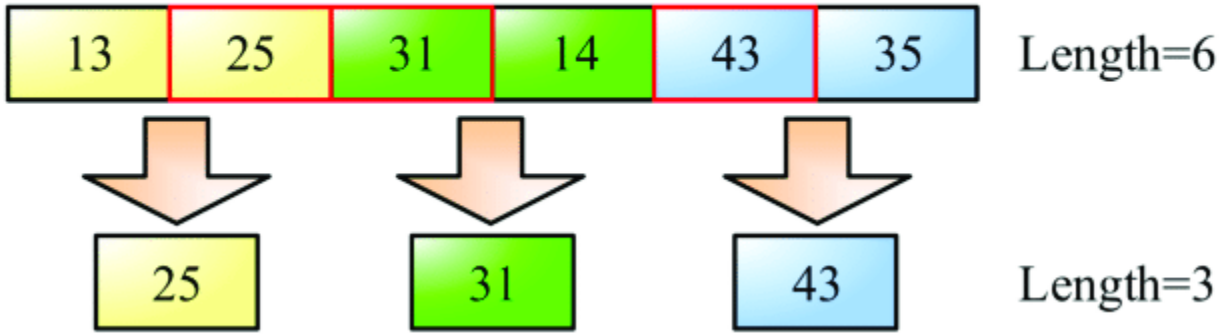
Max Pooling 1D

Stride = 2

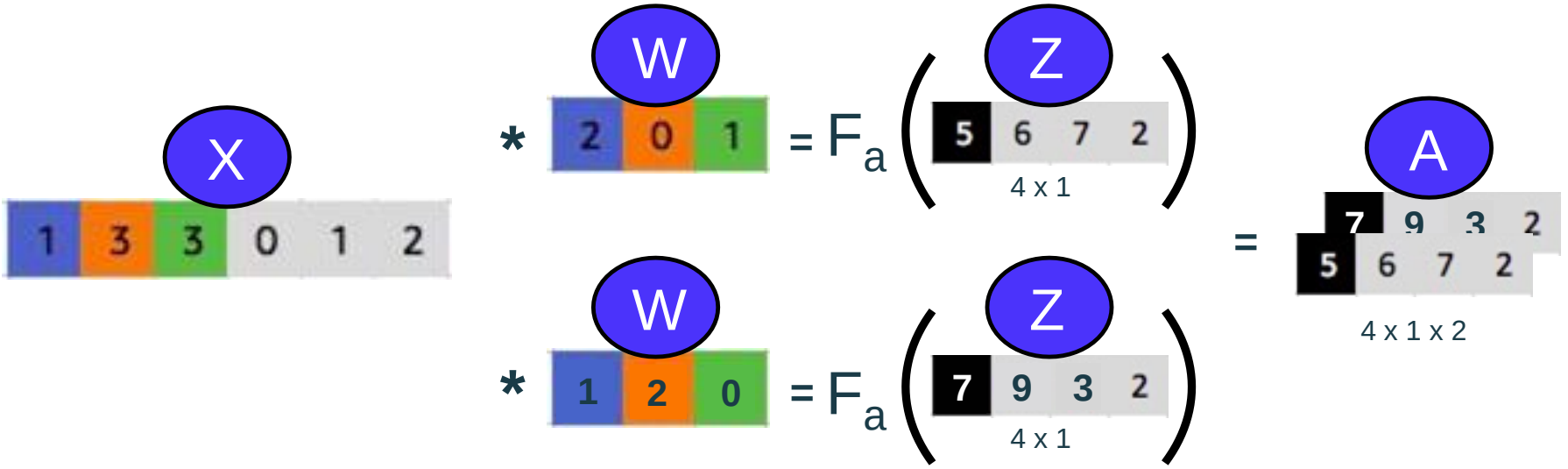
Filtro 2x1

Tamanho Final = Matriz / S

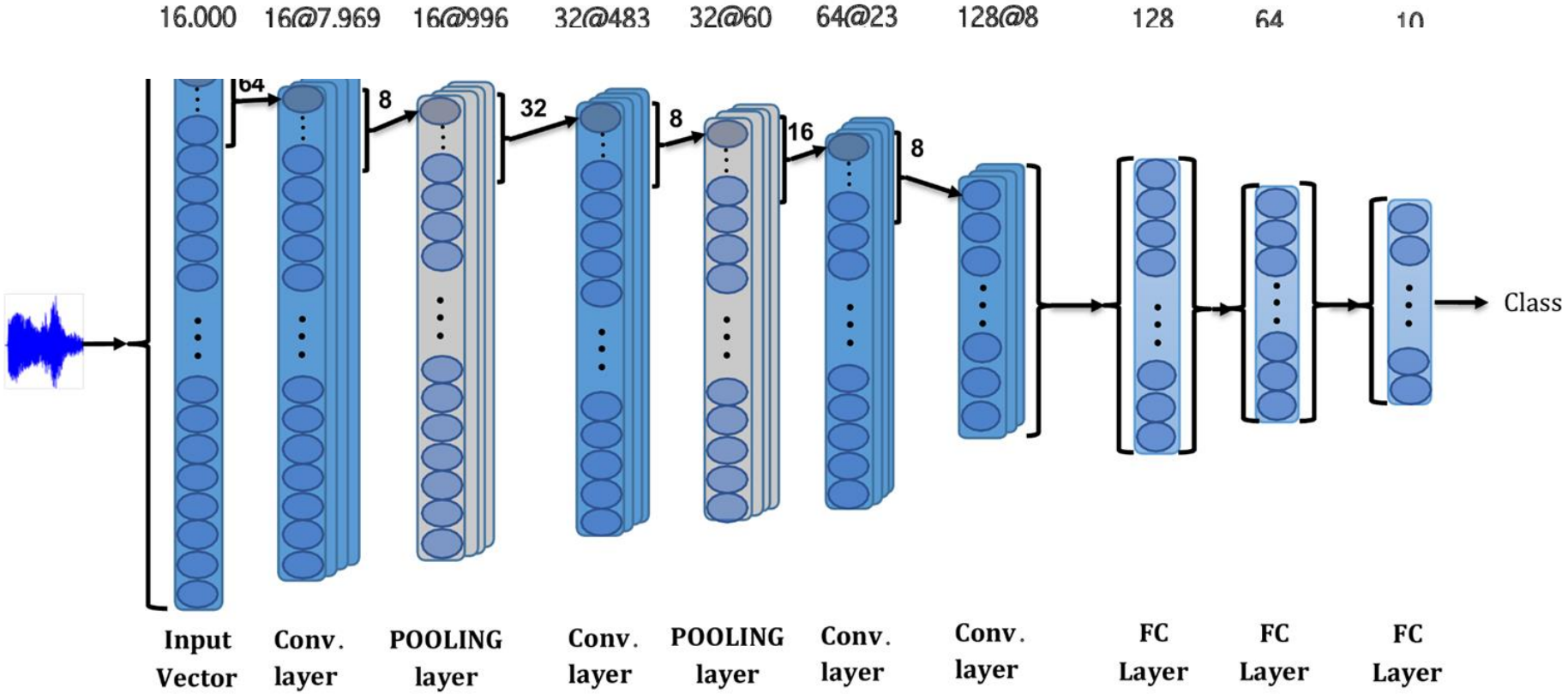
Tamanho Final = 6 / 2 = 3



Redes neurais convolucionais 1D



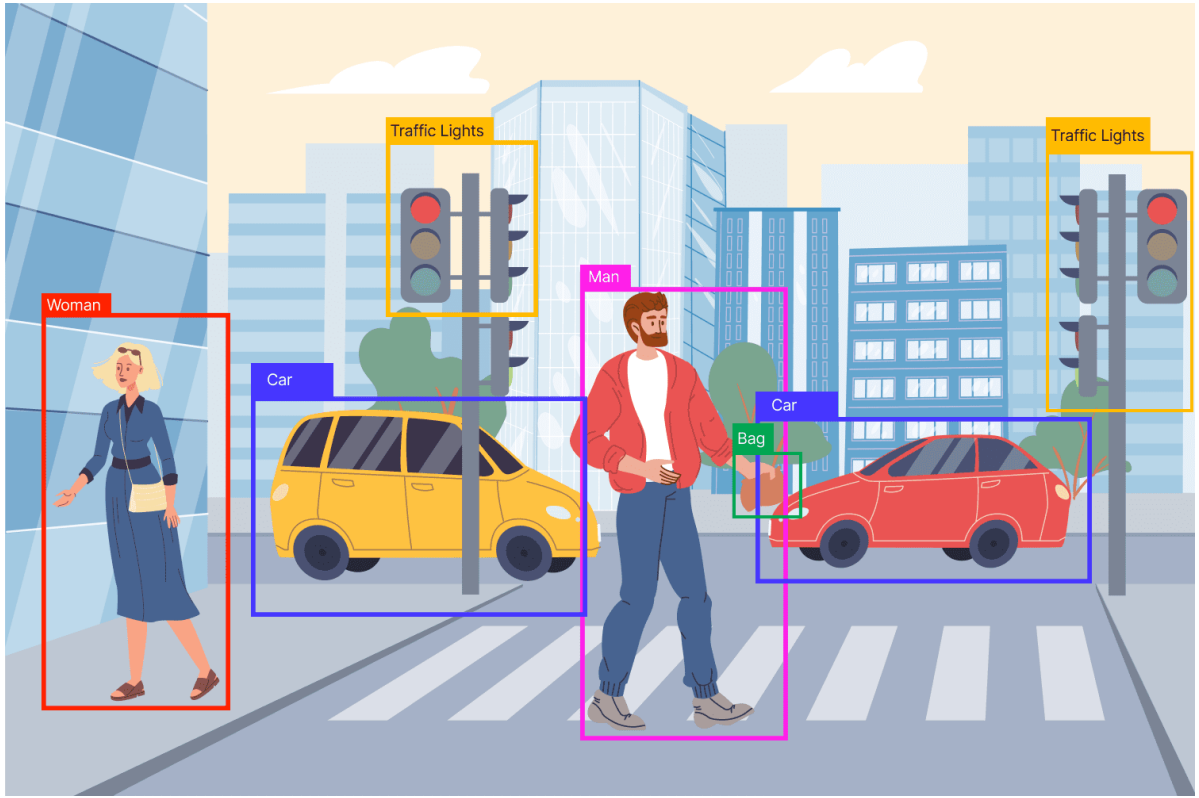
Redes neurais convolucionais 1D



Detecção

06

Detecção



Detecção

- Arquiteturas de detecção iniciam com uma arquitetura de classificação, conhecida a arquitetura escolhida(pode ser qualquer uma das estudadas anteriormente para classificação) é dada o nome de backbone.
- A construção é bem parecida com uma arquitetura de classificação o que muda é a saída.
- Treina-se uma rede para encontrar certos objetos em uma imagem ou video.
- A saída da rede é tanto os bounding boxes quanto a classificação.
- O desempenho da rede é avaliado através de IoU, MaP, Precision e Recall.

IoU

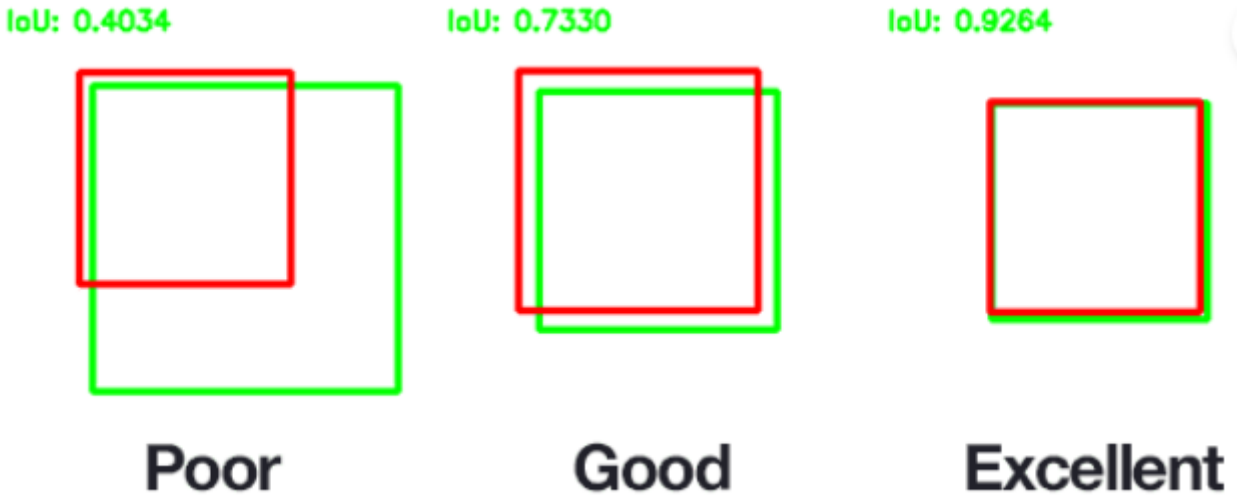
Interseção sobre união ou índice de jaccard.

Calculado usando a bounding box predita e a bound

$$= \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IoU



TP, FP, FN

TP: True Positive(Verdadeiro Positivo) em detecção para ser um verdadeiro positivo:

- A confiança tem que ser maior que o threshold definido.
- O IoU tem que ser maior que o threshold definido.
- A Classe tem que estar correta

FP: False Positive(Falso Positivo) em detecção para ser um falso positivo:

- O IoU tem que ser igual a 0 ou a classe errada tem que ter sido predita errada

FN: False Negative(Falso Negativo) em detecção para ser um falso negativo:

- A Classe tem que estar certa
- O Iou tem que estar abaixo do threshold definido

AP

- Precisão media(Average Precision) para cada classe.
- Calculada usando interpolação de 11 pontos da curva do precision e recall, variando o recall de 0 a 1, com incrementos de 0.1.
- Para cada recall é utilizado o maior valor de precisão que tenha um recall maior ou igual ao recall sendo avaliado.

AP

Considerando essa tabela vamos calcular a interpolação de 11 pontos.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

AP

Para o recall 0 o maior valor de precisão para um recall maior ou igual a 0 é 1, portanto nesse primeiro ponto o valor será 1.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Para o recall 0.1 o maior valor de precisão para um recall maior ou igual a 0.1 é 0.6666, portanto nesse segundo ponto o valor será 0.6666.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Até o momento temos que a AP é

$$1 \div 11(1 + 0.6666)$$

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Para o recall 0.2 o maior valor de precisão para um recall maior ou igual a 0.2 é 0.4285, portanto nesse segundo ponto o valor será 0.4285.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Para o recall 0.3 o maior valor de precisão para um recall maior ou igual a 0.3 é 0.4285, portanto nesse segundo ponto o valor será 0.4285.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Para o recall 0.4 o maior valor de precisão para um recall maior ou igual a 0.4 é 0.4285, portanto nesse segundo ponto o valor será 0.4285.

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Já para os valores a partir de recall a partir de 0.5 não existem valores de precisão portanto a precisão é 0.

Legenda:

- Valores considerados
- Valores escolhido

AP

Assim nossa equação final é:

$1 \div 11(1 + 0.6666 + 3 * 0.4285 + 6 * 0)$

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

Legenda:

- Valores considerados
- Valores escolhido

AP

Cujo resultado é:

0.2685

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

MaP

Média do average precision considerando a average precision de cada classe

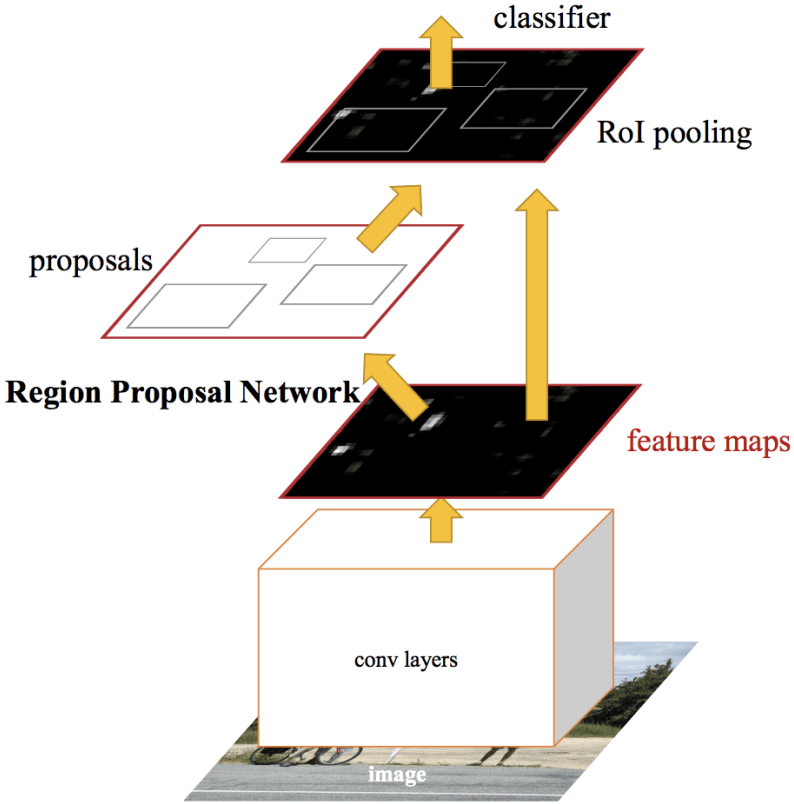
Detecção

Alguns dos datasets usados como benchmark para detecção são o COCO e o Pascal Voc

Faster RCNN

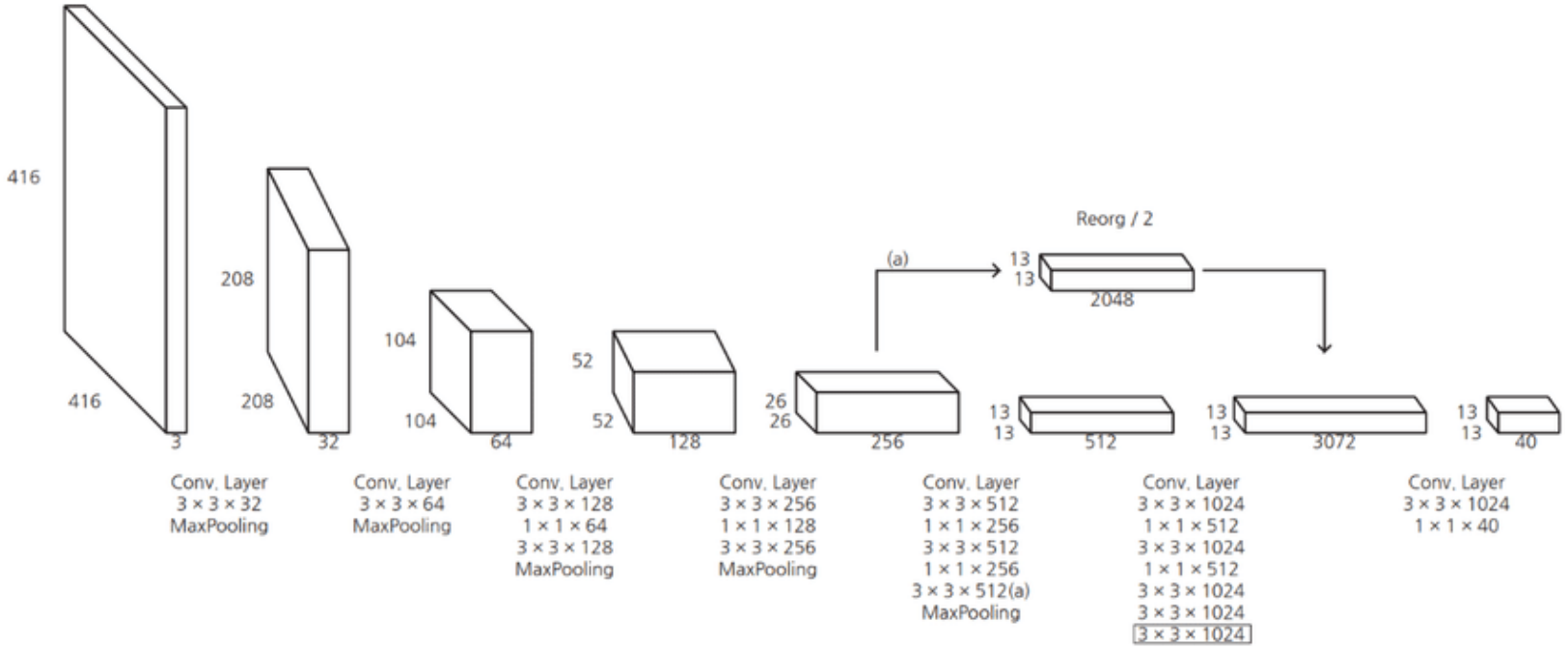
Detecção em dois estágios: primeiro são propostas as bounding boxes, depois são classificadas as imagens.

Usa uma VGG como backbone



YOLO (You Only Look Once)

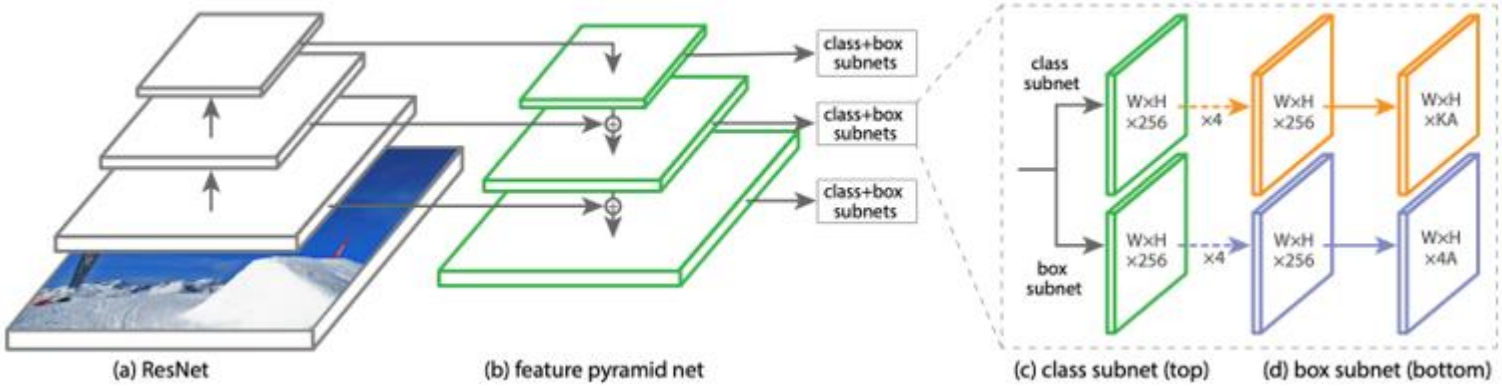
Detecção em um estágio só. Em uma só passada é gerada tanto as coordenadas das bounding boxes como a classe da bounding boxes. Usa uma DarkNet como backbone



RetinaNet

Detecção em um estágio só. Em uma só passada é gerada tanto as coordenadas das bounding boxes como a classe da bounding boxes.

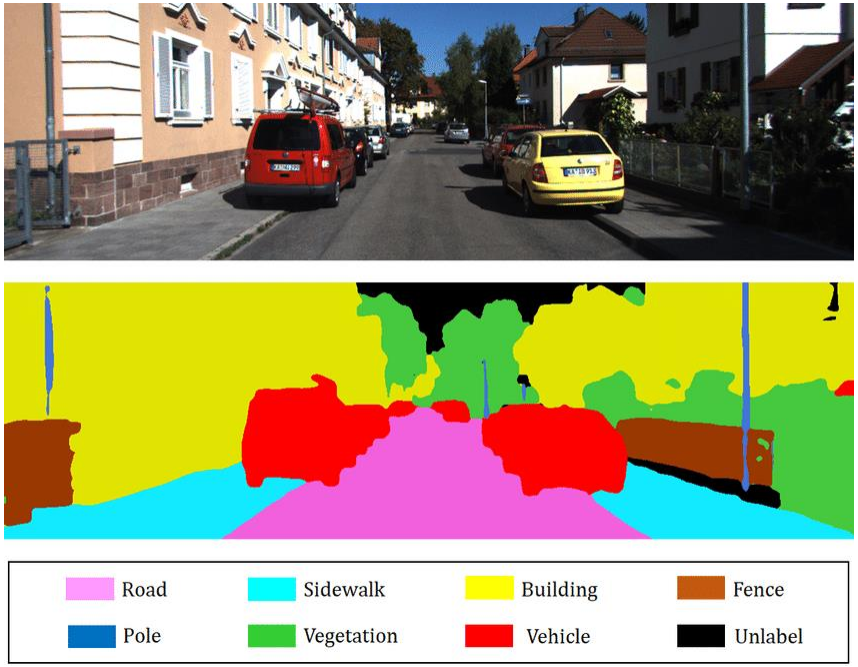
Usa uma ResNet como backbone, usa uma FPN no topo da ResNet e a saída é jogada em duas subredes que calculam paralelamente a posição das bounding boxes e o label.



Segmentação

07

Segmentação

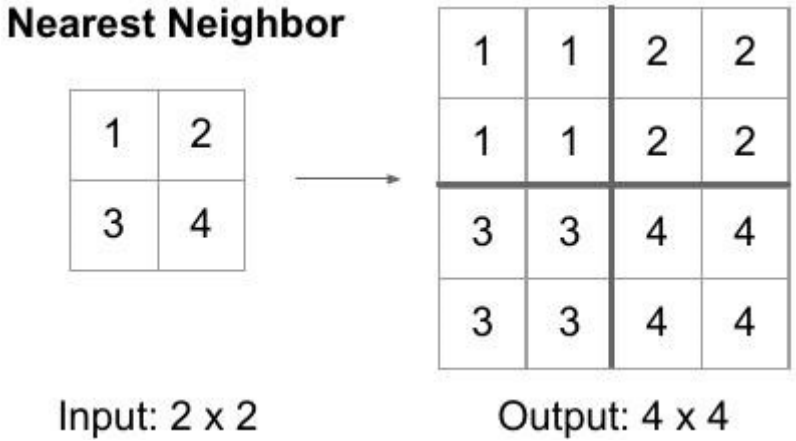


Segmentação

- Consiste em marcar todos os pixels das classes de interesse em uma imagem.
- Assim como arquiteturas de detecção, em segmentação também temos uma arquitetura de classificação(backbone) dentro da arquitetura.
- No entanto, como arquiteturas de segmentação retornam um objeto do mesmo tamanho do input e convoluções(sem padding) e pooling diminuem o tamanho do objeto duas operações novas são introduzidas: upsampling e upconv.

Upsampling

Funciona apenas executando operações no mapa original, podendo ser desde preencher com zeros a bilinear. Abaixo temos um exemplo de upsampling aplicando nearest neighbor que é apenas substituir o valor pelo valor mais próximo.



Upconv

Upconv ou deconvolution ou transpose convolution. Consiste em mover o filtro pelo mapa de saída como se fosse uma convolução e o mapa de entrada dá o peso para o filtro. Além disso soma-se as posições do filtro em que há intersecção.

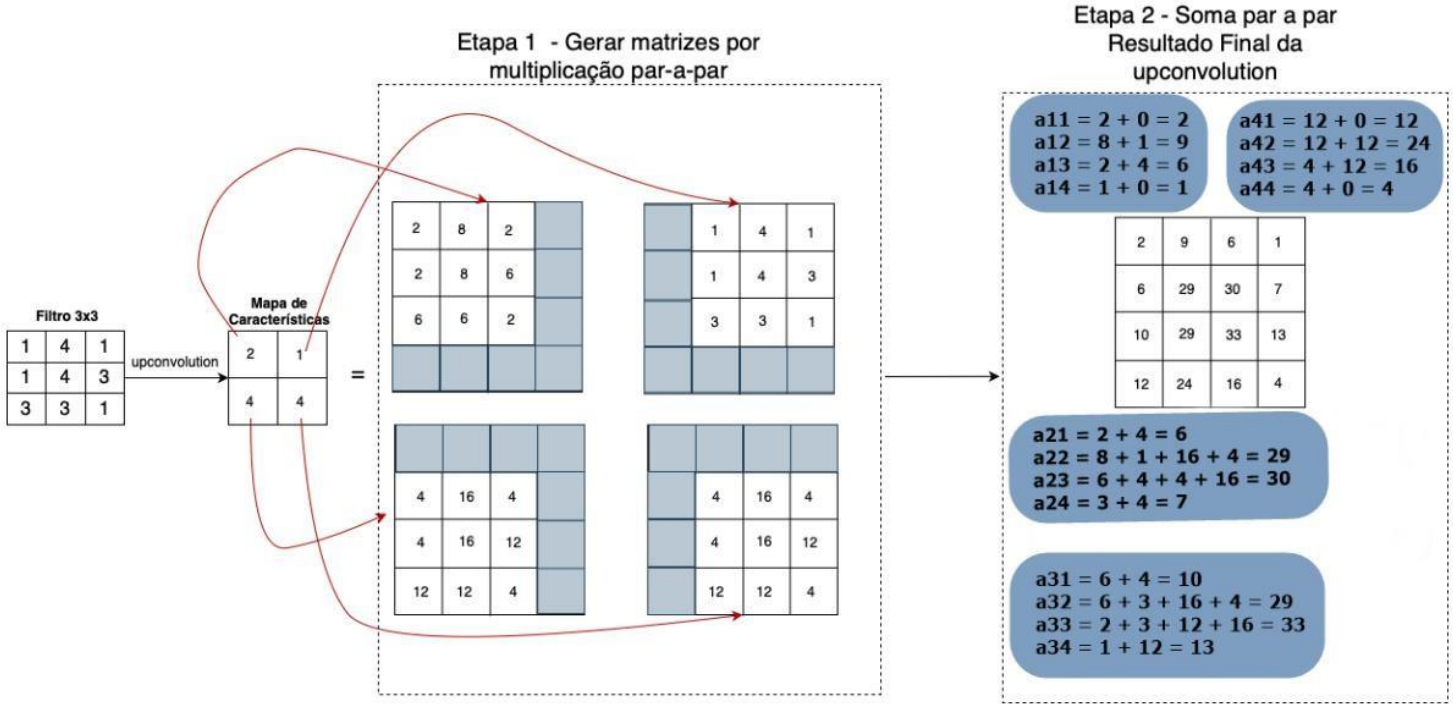
Upconv

Upconv ou deconvolution ou transpose convolution. Consiste em mover o filtro pelo mapa de saída como se fosse uma convolução e o mapa de entrada dá o peso para o filtro. Além disso soma-se as posições do filtro em que há intersecção.

Para ajudar na compreensão do exemplo a seguir, vamos relembrar as posições de uma matriz 4 x 4

a11	**a12**	**a13**	**a14**
a21	a22	a23	a24
a31	a32	a33	a34
a41	a42	a43	a44

Upconv


Etapa 1 - Gerar matrizes por multiplicação par-a-par
Etapa 2 - Soma par a par Resultado Final da upconvolution
Filtro 3x3
1 4 1
1 4 3
3 3 1
upconvolution
Mapa de Características
2 1
4 4
=
2 8 2
2 8 6
6 6 2
1 4 1
1 4 3
3 3 1
4 16 4
4 16 12
12 12 4
4 16 4
4 16 12
12 12 4
a11 = 2 + 0 = 2
a12 = 8 + 1 = 9
a13 = 2 + 4 = 6
a14 = 1 + 0 = 1
a41 = 12 + 0 = 12
a42 = 12 + 12 = 24
a43 = 4 + 12 = 16
a44 = 4 + 0 = 4
2 9 6 1
6 29 30 7
10 29 33 13
12 24 16 4
a21 = 2 + 4 = 6
a22 = 8 + 1 + 16 + 4 = 29
a23 = 6 + 4 + 4 + 16 = 30
a24 = 3 + 4 = 7
a31 = 6 + 4 = 10
a32 = 6 + 3 + 16 + 4 = 29
a33 = 2 + 3 + 12 + 16 = 33
a34 = 1 + 12 = 13

Upconv

Explicando a imagem da página anterior:

- Na primeira posição o filtro é multiplicado por 2. E os valores resultantes são exibidos no quadrado de cima e da esquerda da etapa 1.
- Na segunda posição o filtro é multiplicado por 1. E os valores resultantes são exibidos no quadrado de cima e da direita da etapa 1. Notem que há uma intersecção da segunda posição com a primeira posição do filtro. Essa intersecção ocorre nas colunas 2 e 3 portanto os valores serão somados.
- Na terceira posição o filtro é multiplicado por 4. E os valores resultantes são exibidos no quadrado de baixo de da esquerda da etapa 1. Notem que há intersecção tanto com a posição 1 quanto com a posição 2. Assim os valores da intersecção são somados.
- Na quarta posição o filtro é multiplicado por 4. E os valores resultantes são exibidos no quadrado de baixo de da esquerda da etapa 1. Notem que há intersecção com as posições 1, 2 e 3. Assim os valores da intersecção são somados.

Upconv

Continuando a explicação:

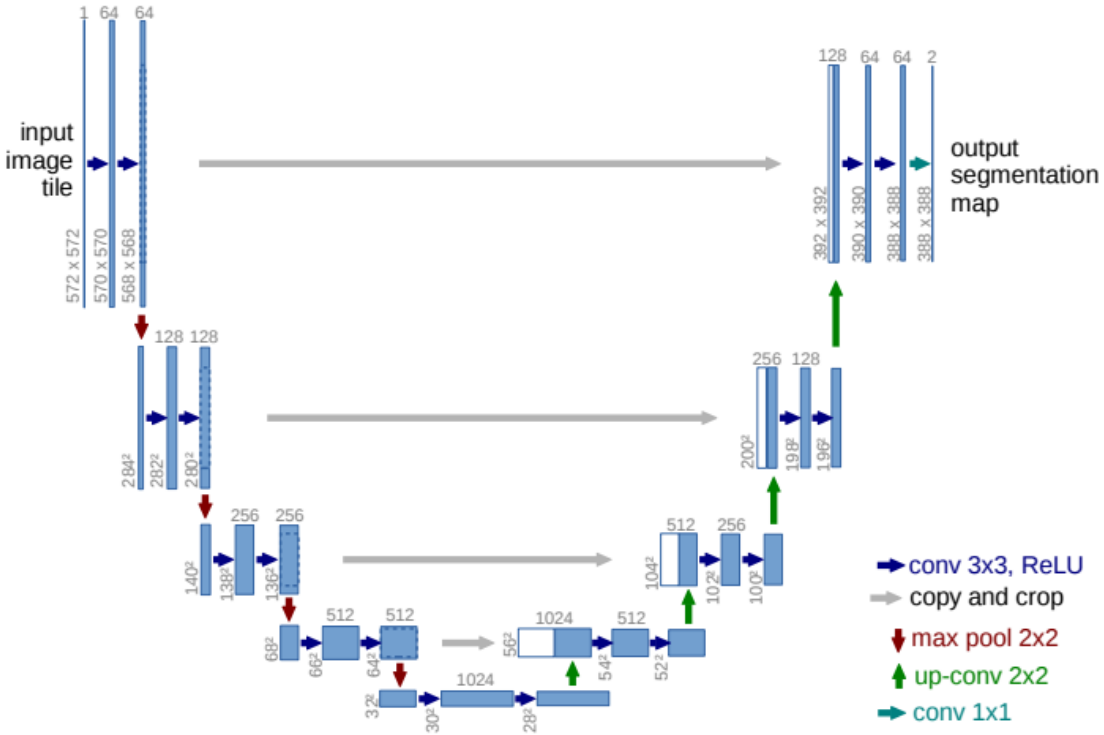
- Na etapa é mostrado o resultado final das operações
- Também é mostrado como foi calculado o valor de cada elemento, isto é, o valor das somas das intersecções para cada elemento.

Métricas e Datasets

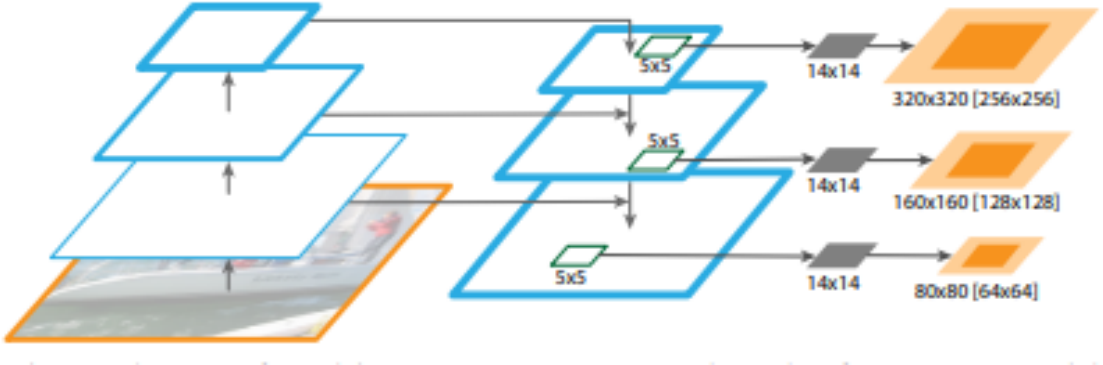
Para segmentação semântica usamos apenas IoU em caso de uma classe e mIoU(media dos IoU) quando temos múltiplas classes.

Alguns datasets usados como benchmark para segmentação são Pascal VOC, CityScapes e Ade20k

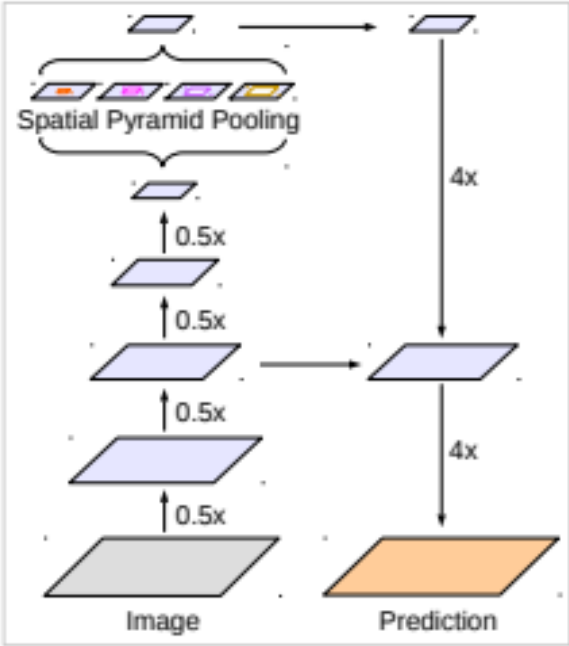
Unet


input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

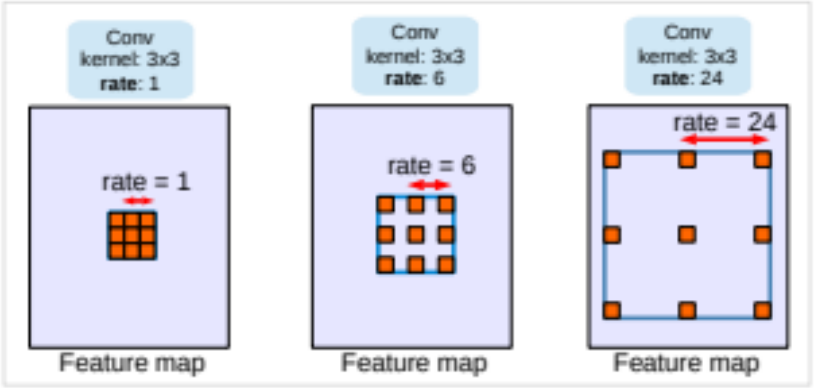
FPN


5x5
5x5
5x5
14x14
14x14
14x14
320x320 [256x256]
160x160 [128x128]
80x80 [64x64]

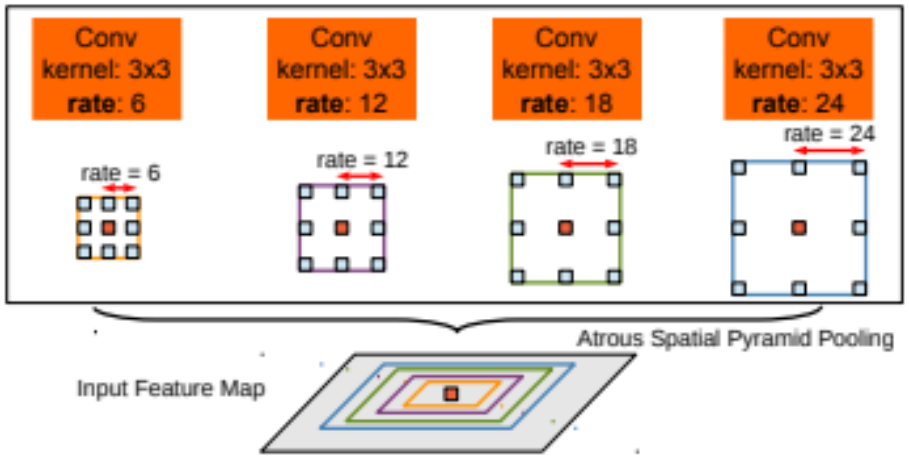
DeepLab


Spatial Pyramid Pooling
4x
0.5x
0.5x
0.5x
4x
0.5x
Image
Prediction

Atrous Convolution



Atrous Spatial Pyramid Pooling(ASPP)



Referências

- Iris Recognition with Off-the-Shelf CNN Features: A Deep Learning Perspective - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-evolution-of-the-winning-entries-on-the-ImageNet-Large-Scale-Visual-Recognition_fig1_321896881 [accessed 23 Oct, 2023]

- Application of Deep Learning in Dentistry and Implantology - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Algorithms-that-won-the-ImageNet-Large-Scale-Visual-Recognition-Challenge-ILSVRC-in_fig2_346091812 [accessed 23 Oct, 2023]

- ImageNet classification with deep convolutional neural networks (acm.org)

- [1512.03385v1] Deep Residual Learning for Image Recognition (arxiv.org)

- LeNet-5 - A Classic CNN Architecture - DataScienceCentral.com
- Exploring Object Detection Applications and Benefits – DeepLobe
- [1708.02002] Focal Loss for Dense Object Detection (arxiv.org)
- Mean Average Precision (mAP) Using the COCO Evaluator – PyImageSearch
- http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf
- https://kharshit.github.io/blog/2019/02/15/autoencoder-downsampling-and-upsampling
- [1505.04597] U-Net: Convolutional Networks for Biomedical Image Segmentation (arxiv.org)

Referências

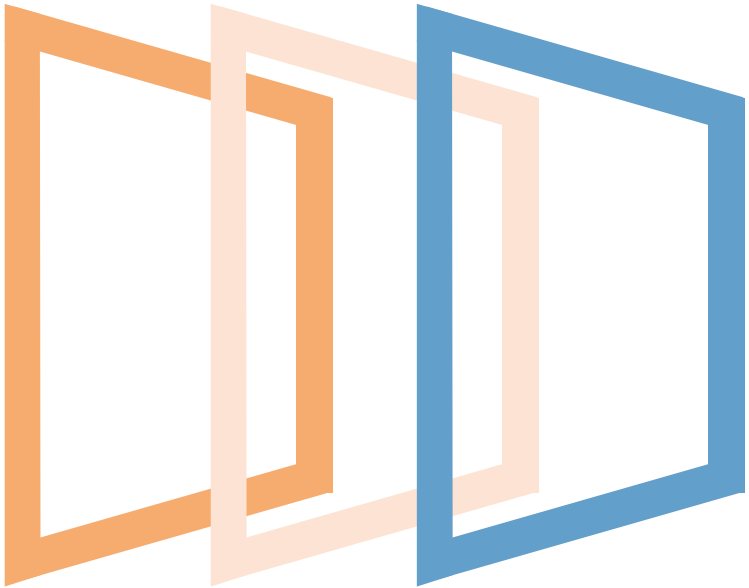
- [1612.03144] Feature Pyramid Networks for Object Detection (arxiv.org)
- [1706.05587v3] Rethinking Atrous Convolution for Semantic Image Segmentation (arxiv.org)
- [1606.00915] DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs (arxiv.org)
- [1802.02611] Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation (arxiv.org)

Redes Neurais Convolucionais

Thaís Ratis

Diego Alexandre

Práticas Tecnológicas, 26.10.2023



minsait

An Indra company