Instituto de Computação

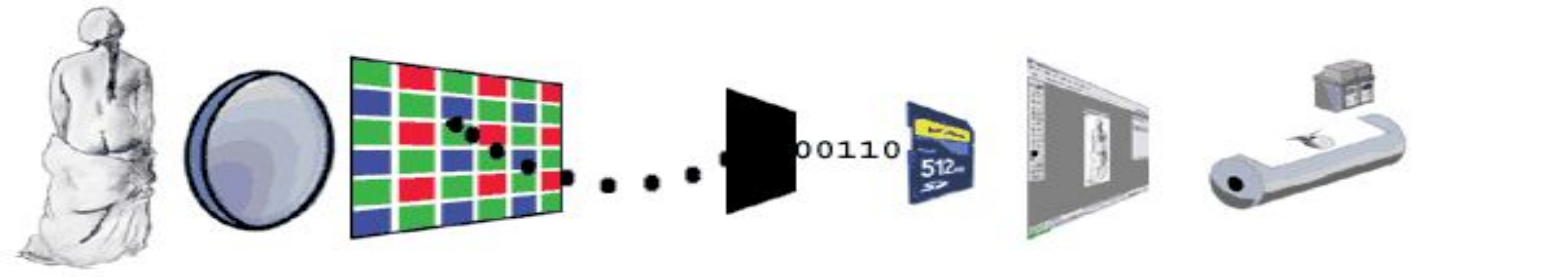
JAI 6 - Deep Learning Teoria e Prática

Cristina Nader Vasconcelos
Universidade Federal Fluminense

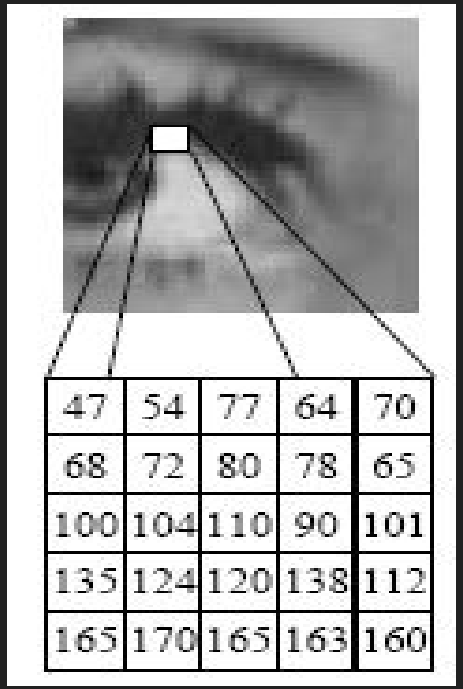
CNNs

Câmeras estão por toda parte!

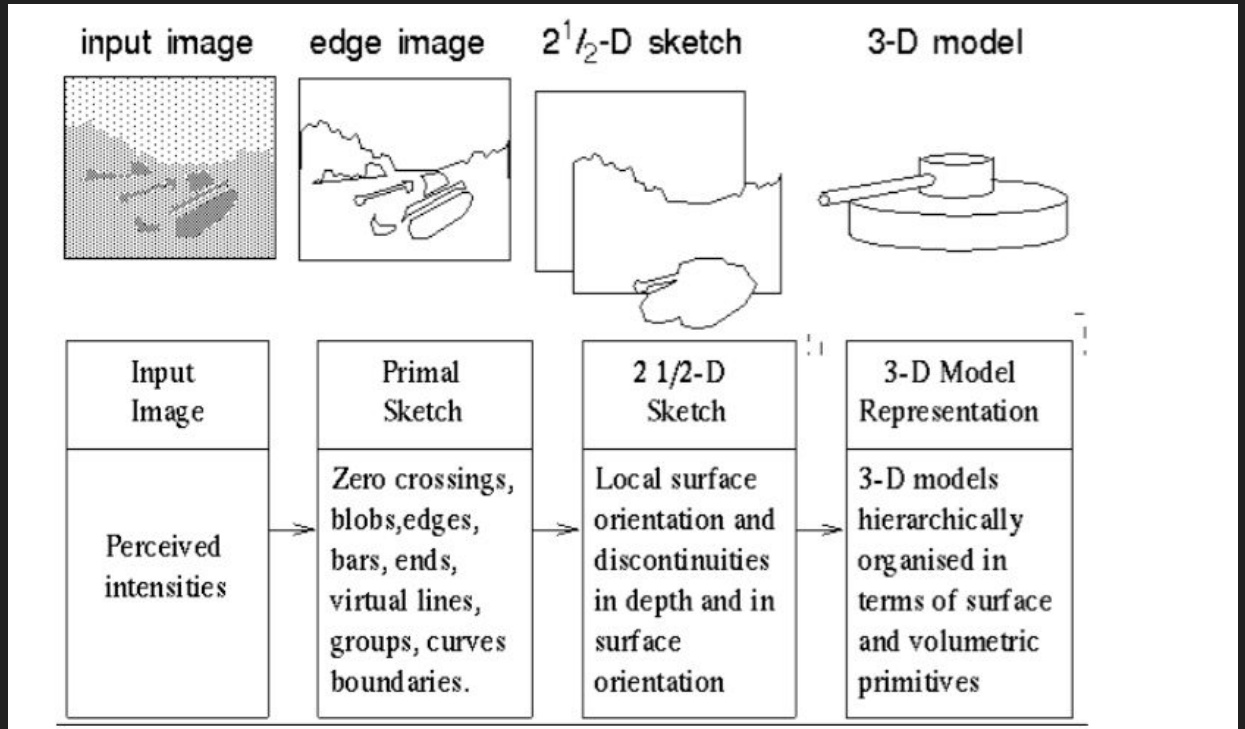
Aquisição de imagens digitais



Representação Digital de Imagens

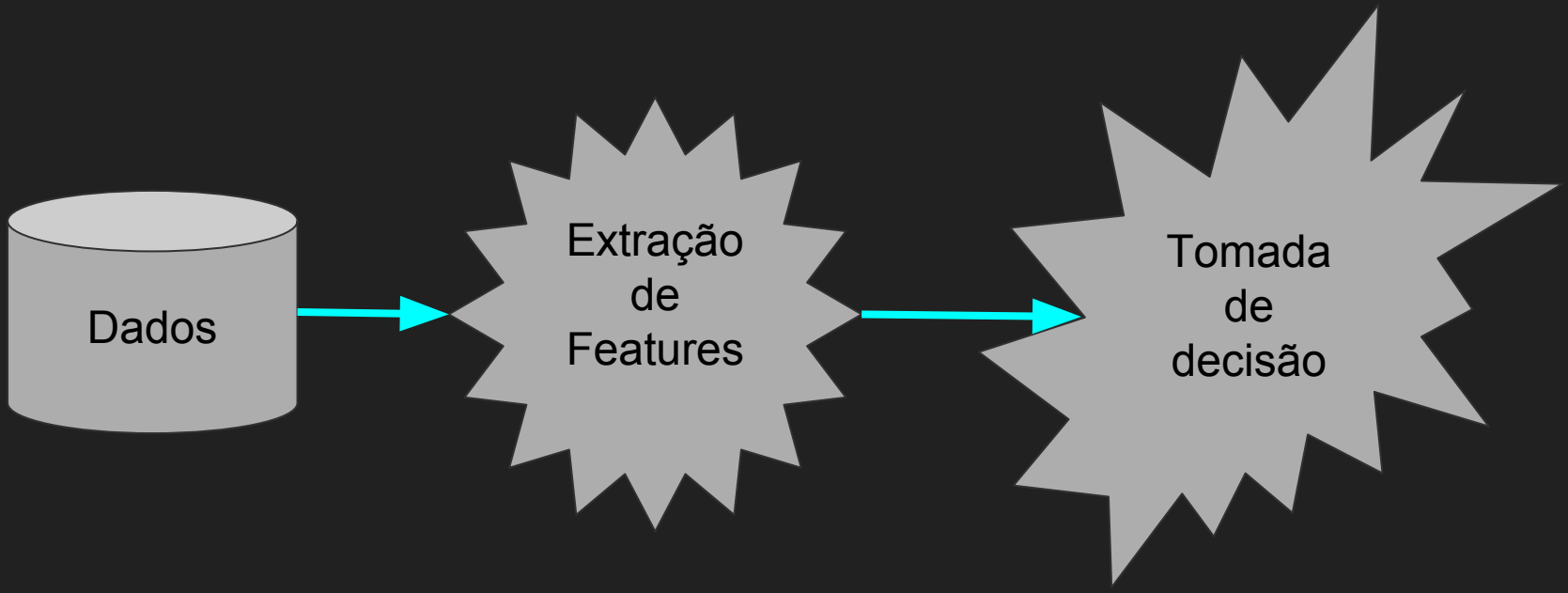


Extração de representação visual

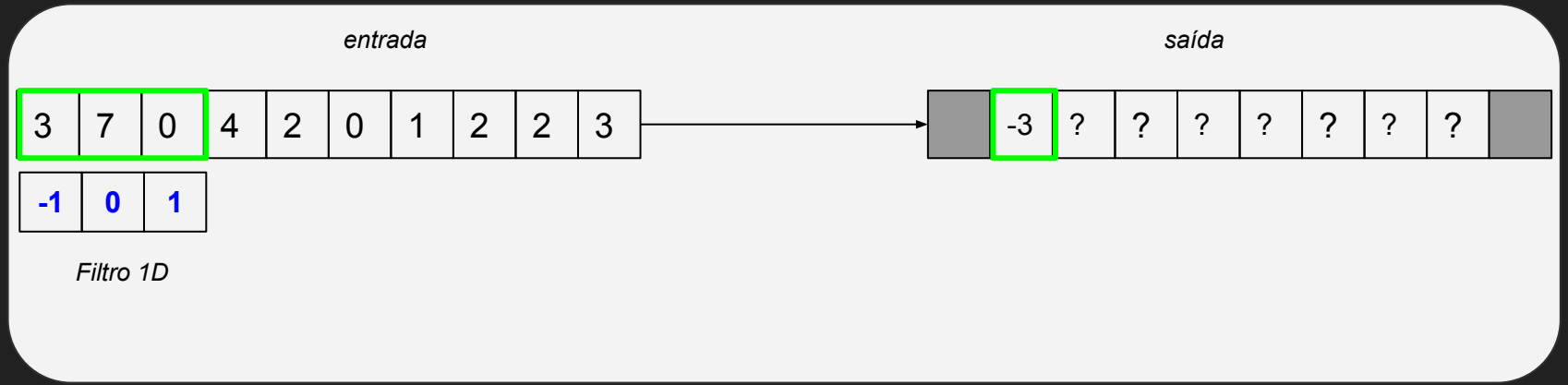


Stages of Visual Representation, David Marr, 1970s

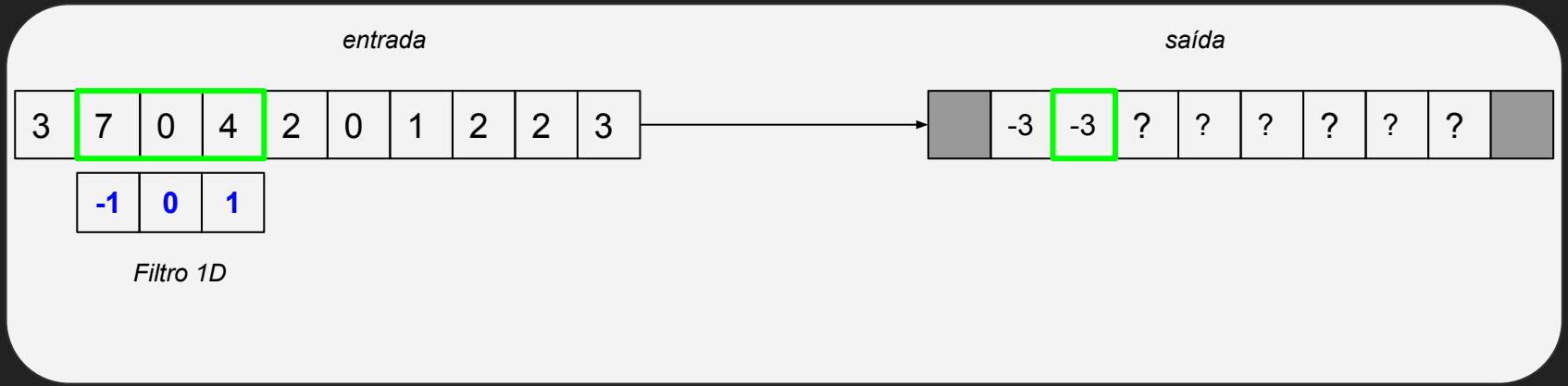
Abordagem clássica



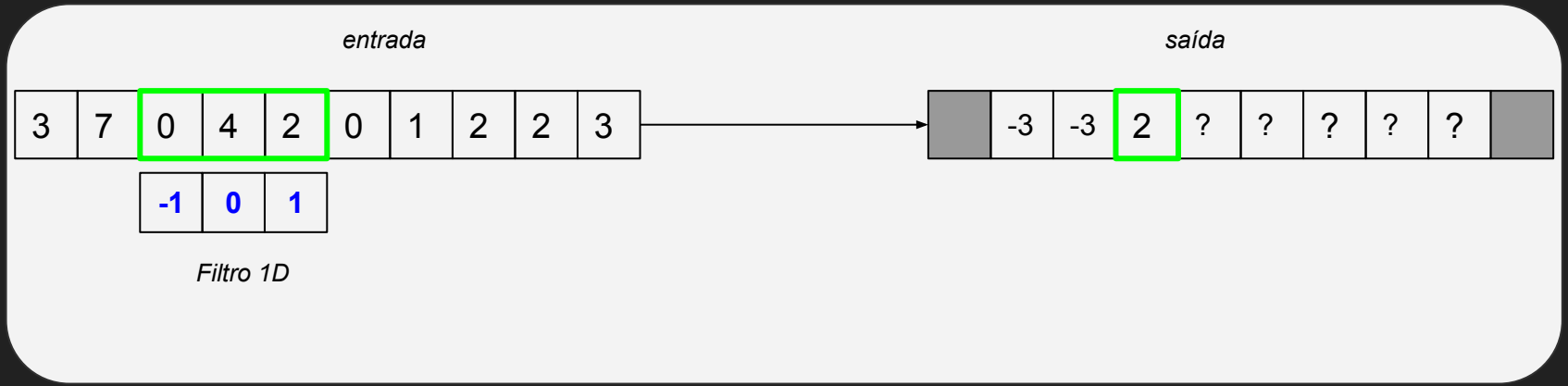
Filtro 1D



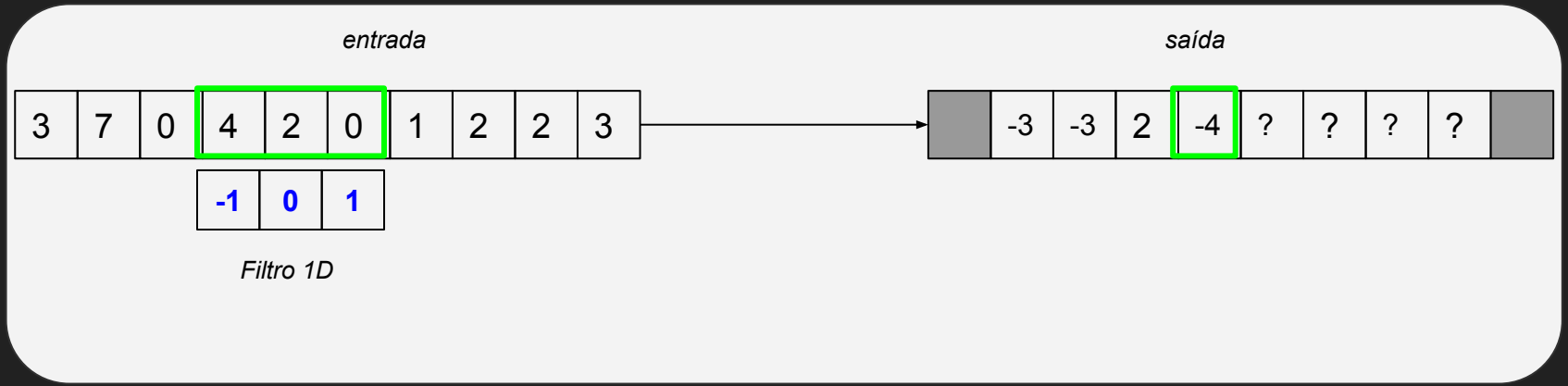
Filtro 1D



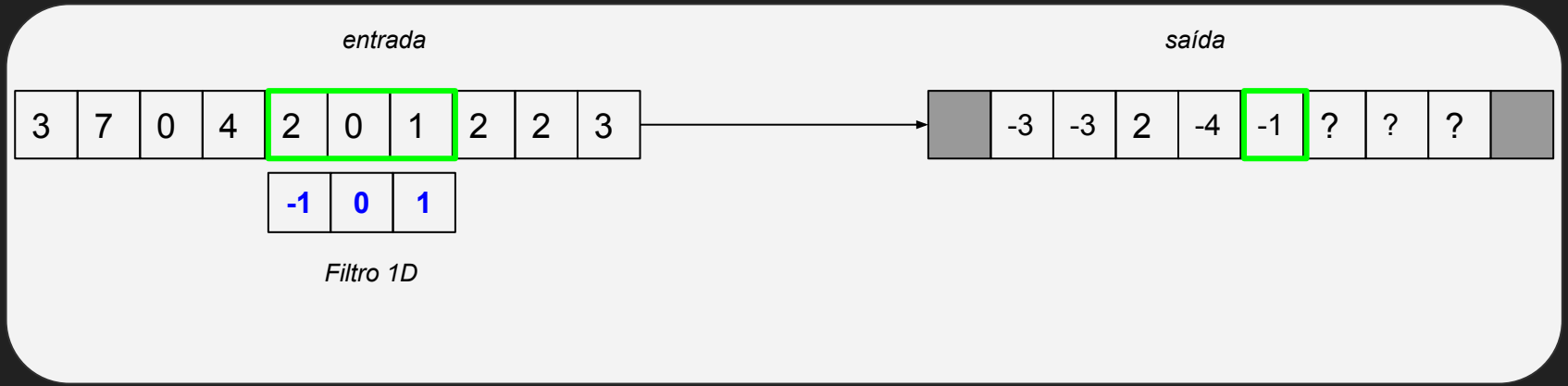
Filtro 1D



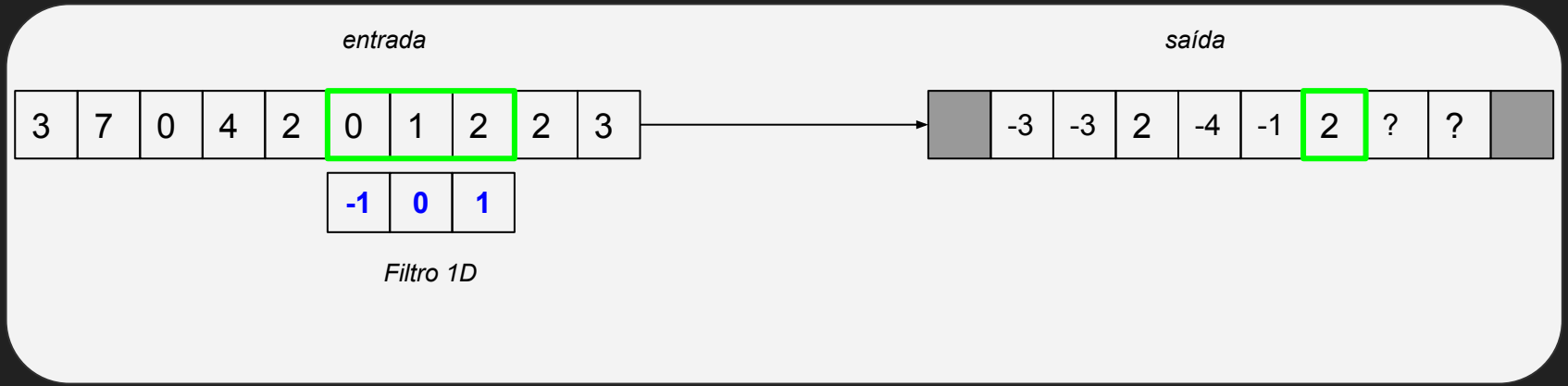
Filtro 1D



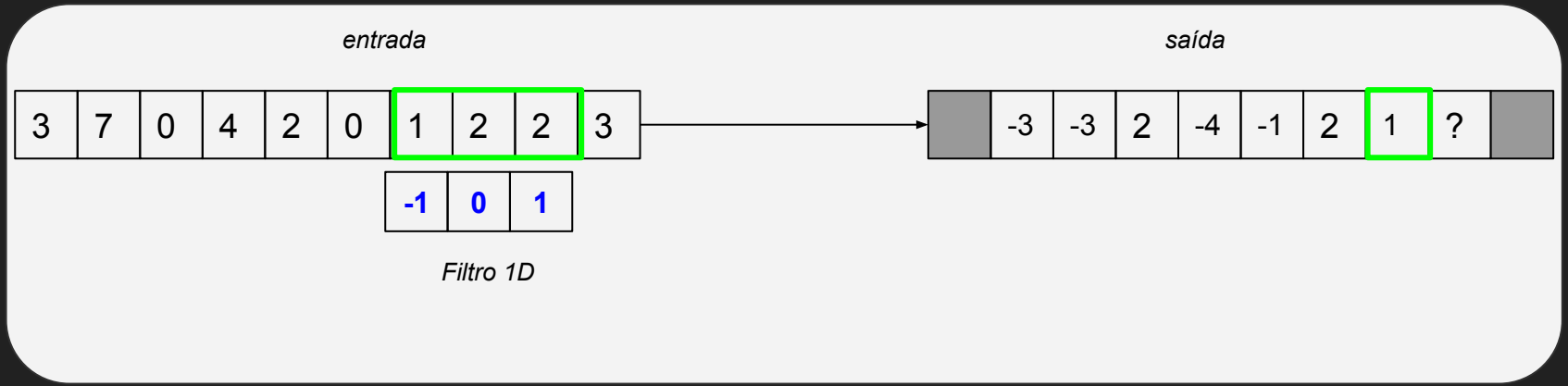
Filtro 1D



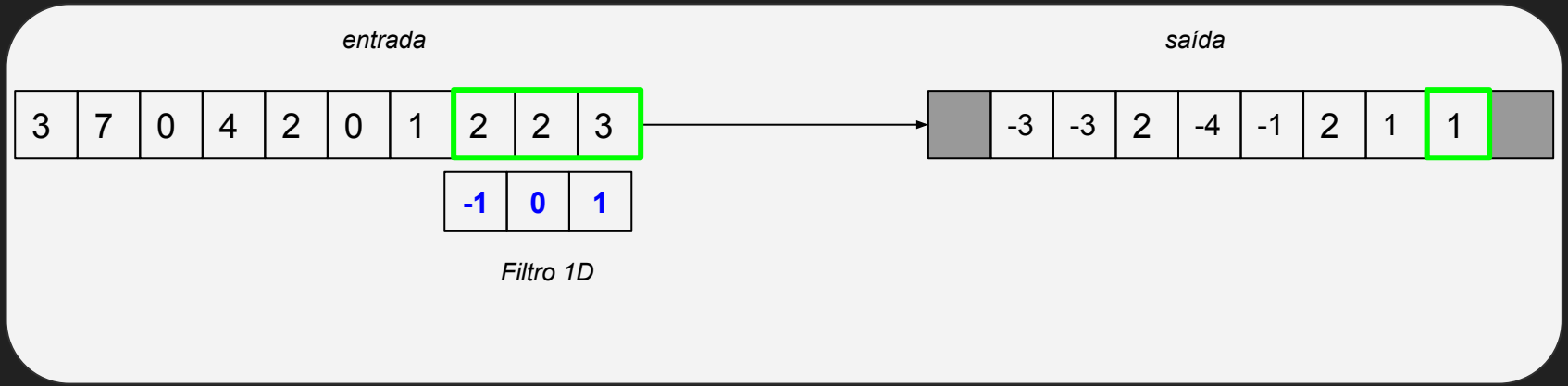
Filtro 1D



Filtro 1D

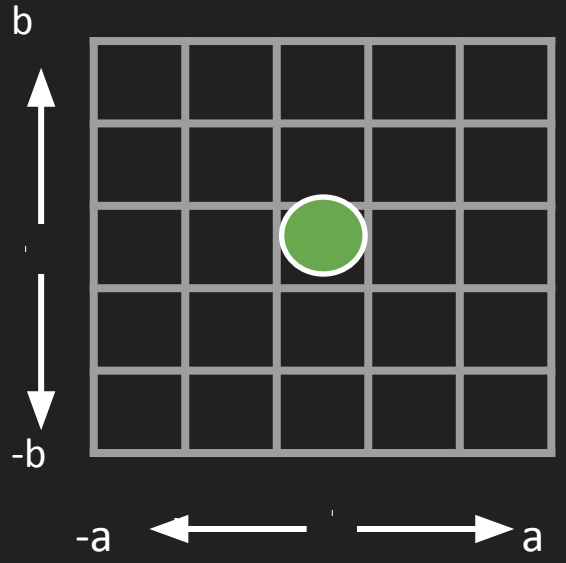


Filtro 1D



Filtragem de imagens

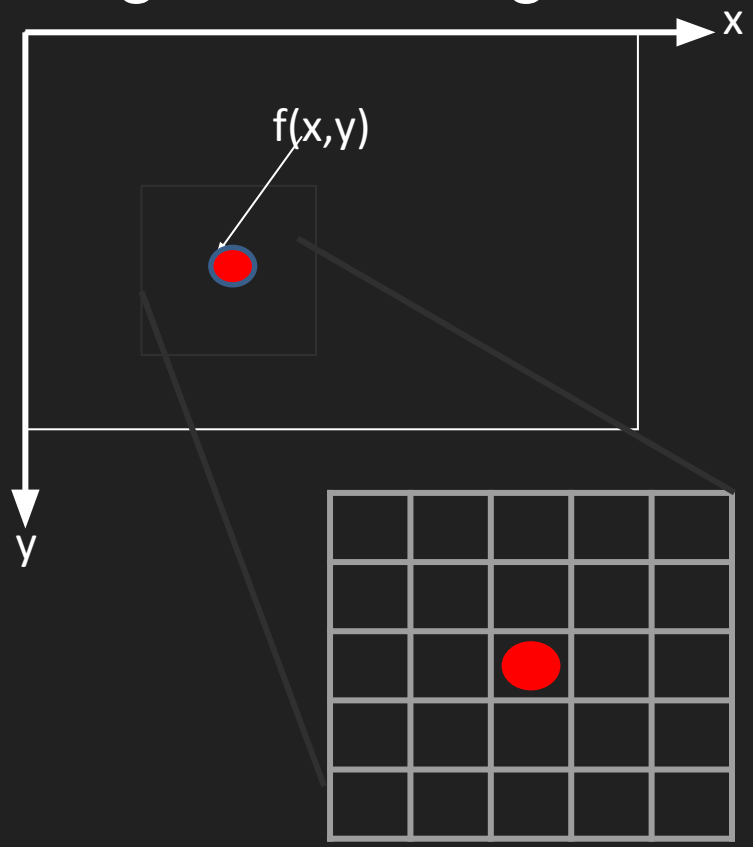
- Matriz de pesos:

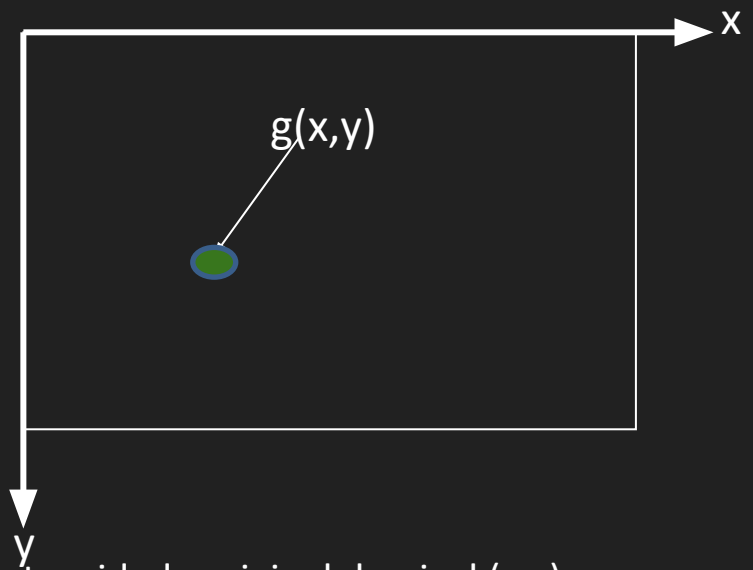


Operação de Filtragem:

$$g(x,y) = \sum_{j=-a}^{a} \sum_{i=-b}^{b} w(i,j) f(x+i, y+j)$$

Filtragem de imagens

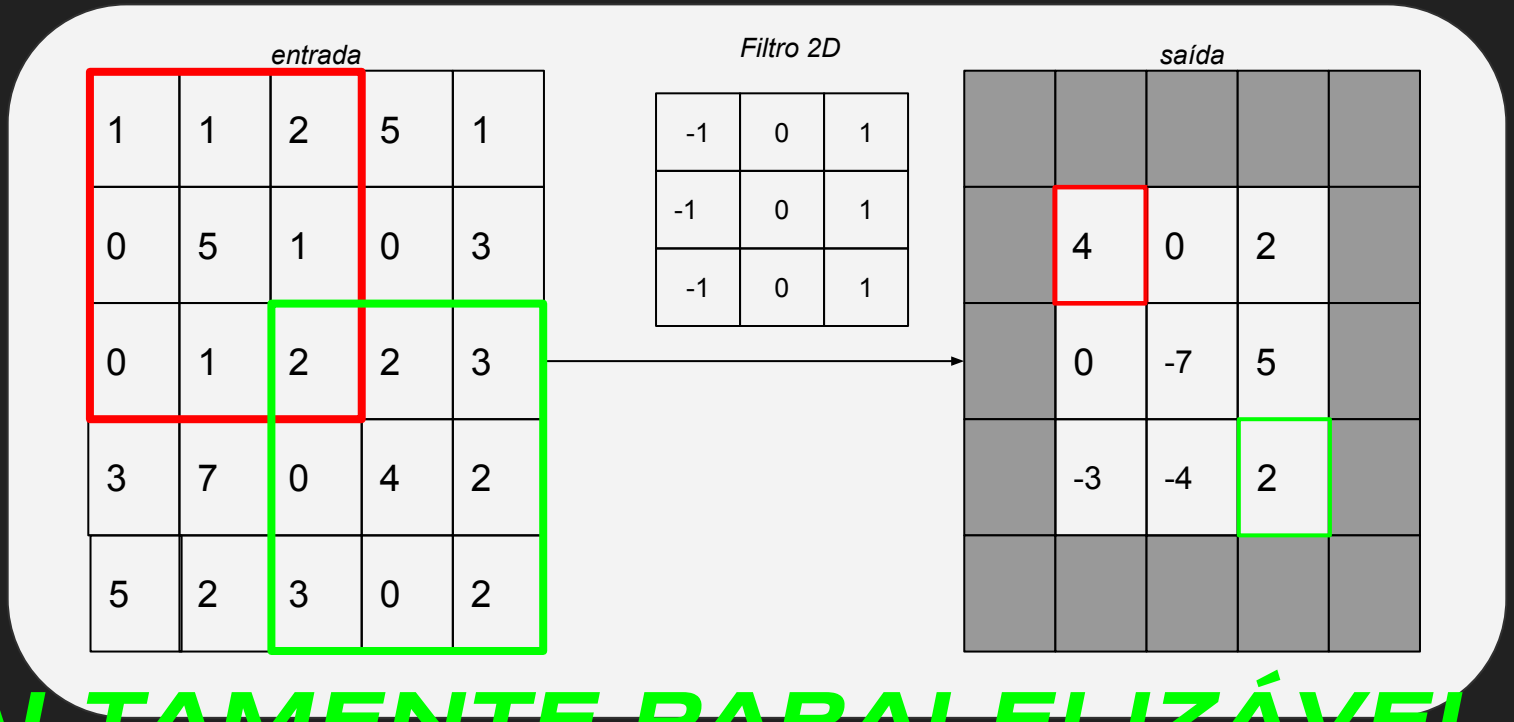




onde:

- f(x,y): intensidade original do pixel (x,y);
- g(x,y): resultado do processamento do pixel (x,y);
- N: vizinhança
- T: operator defino sobre N (x,y), descrito por: $g(x,y) = T_N(f(x,y))$

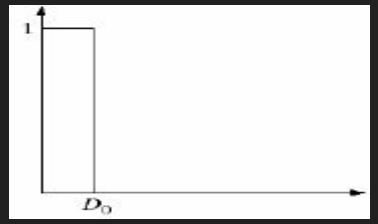
Filtragem de imagens

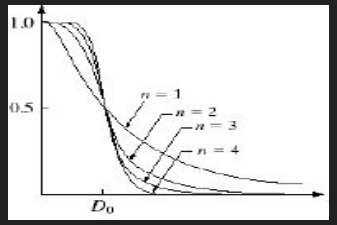


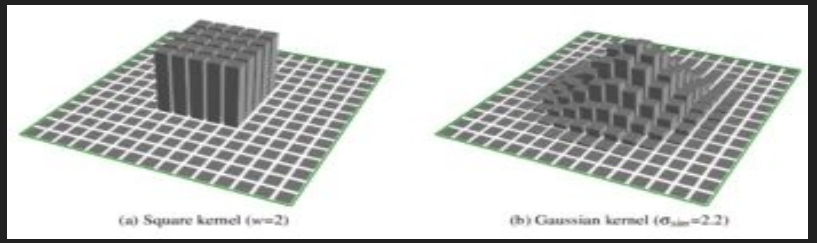
ALTAMENTE PARALELIZÁVEL

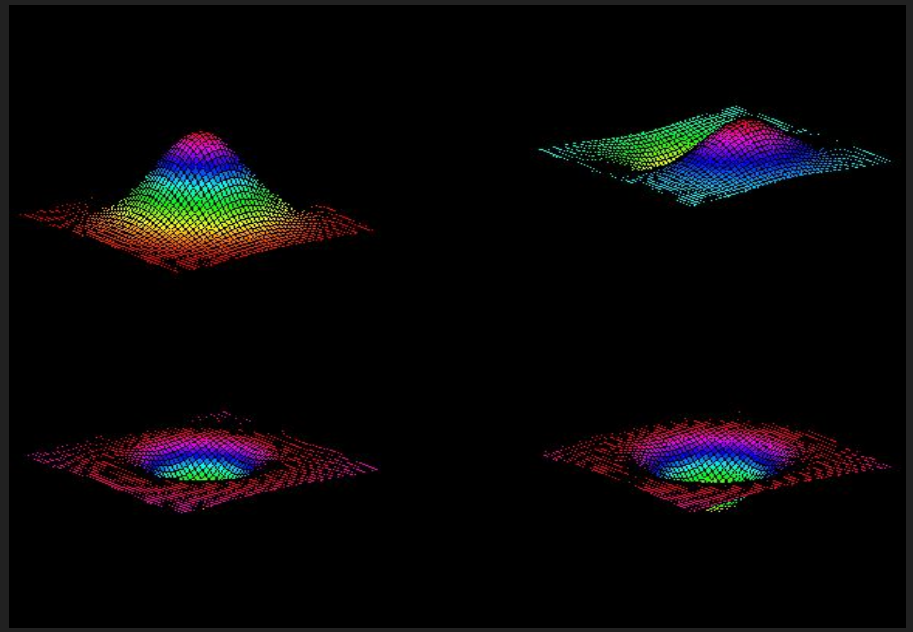
Visualização de Filtros

- Diferentes abordagens:



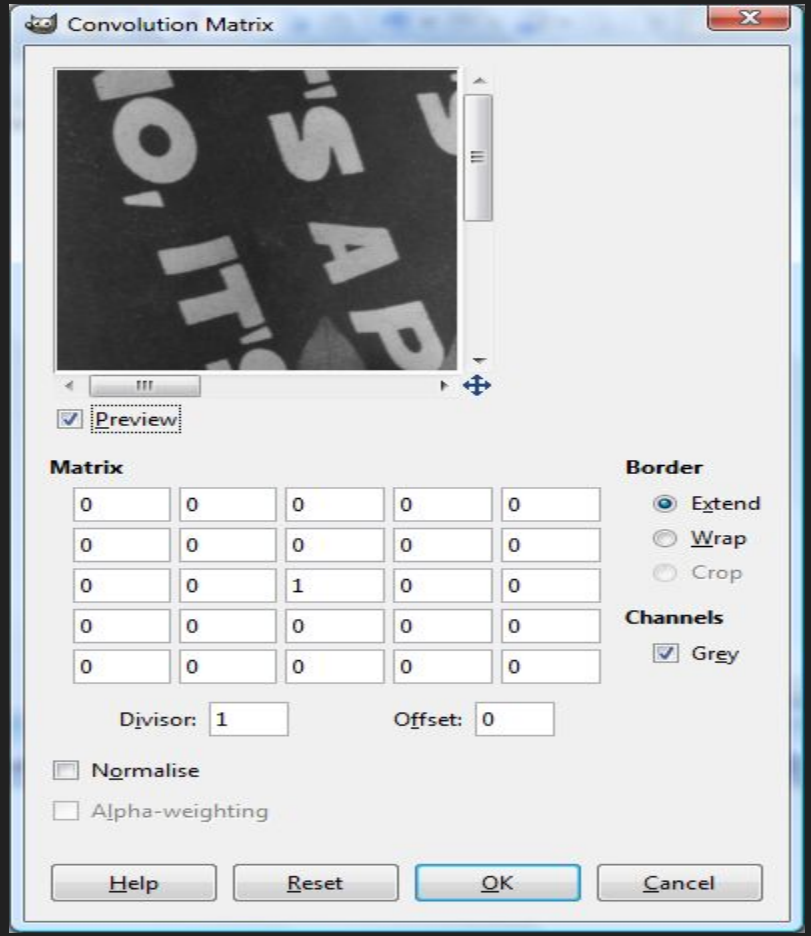






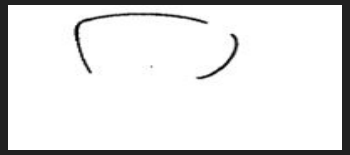
GIMP

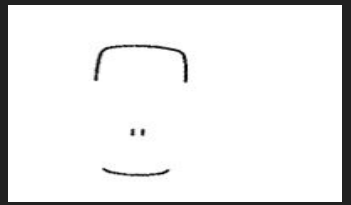
- Menu: Filters
  - Generic
    - Convolution Matrix



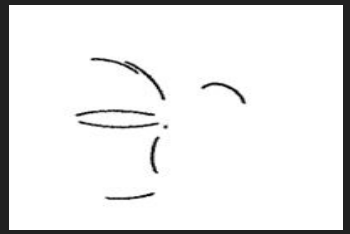
Instituto de Computação

Features Locais: arestas



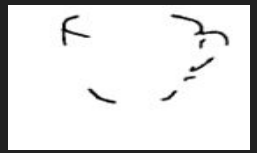


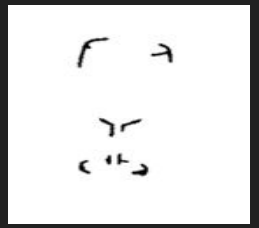
?



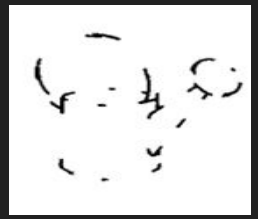
[I. Biederman], “Recognition-by-components: A theory of human image understanding,” *Psychological Review, vol. 2, no. 94, pp. 115–147, 1987.*

Features locais: cantos





?



[I. Biederman], “Recognition-by-components: A theory of human image understanding,” *Psychological Review, vol. 2, no. 94, pp. 115–147, 1987.*

Instituto de Computação

Porque aplicações em sinais naturais/biológicos são problemas difíceis?

Desafios: ponto de vista, deformação, iluminação, ..

[R. Tao, A. Smeulders, S. Chang] Attributes and Categories for Generic Instance Search from One Example. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp.177 - 186. 2015

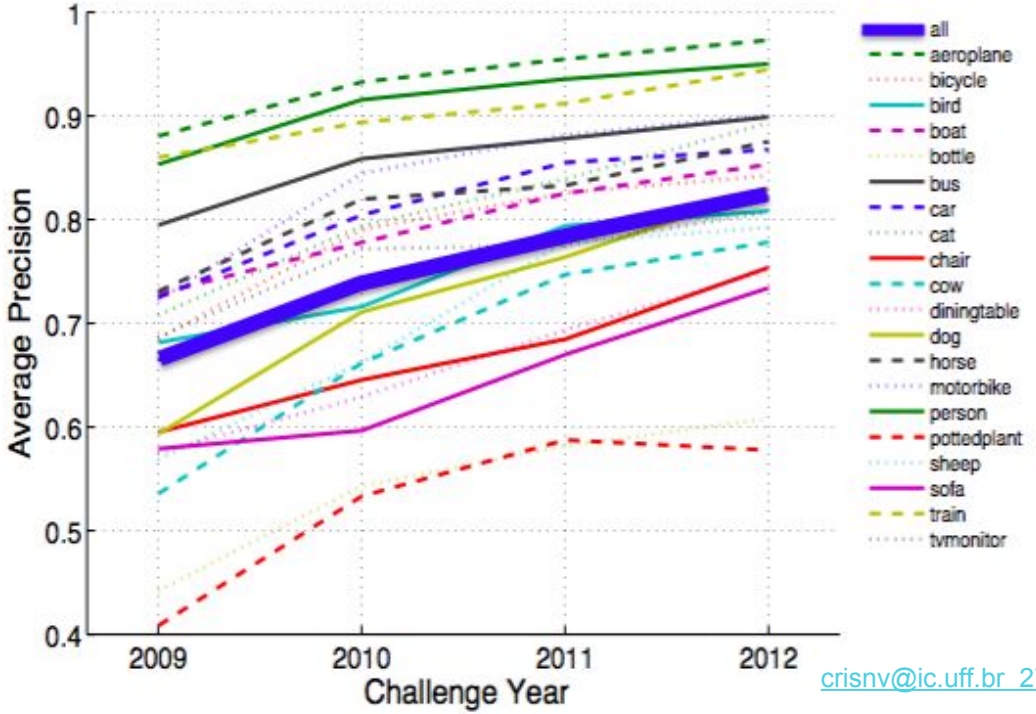
***[K. Patil, S. Bojewar] A Survey on Face Recognition of Identical Twins. International Journal of Scientific & Engineering Research, Vol. 6, Issue 2, Feb.-2015 744.*

Desafios: oclusão, distinção do fundo

Desafios: variação intra-classe

PASCAL Visual Object Challenge (20 object categories)

[Everingham et al. 2006-2012]



"*If we want our machines to think, we need to teach them to see." Fei-Fei Li*

IMAGENET

www.image-net.org

22K categories and **14M** images

- Animals
  - Bird
  - Fish
  - Mammal
  - Invertebrate
- Plants
  - Tree
  - Flower
- Food
- Materials
- Structures
- Artifact
  - Tools
  - Appliances
  - Structures
- Person
- Scenes
  - Indoor
  - Geological Formations
- Sport Activities

Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009

IMAGENET Large Scale Visual Recognition Challenge

Instituto de Computação

The Image Classification Challenge:
1,000 object classes
1,431,167 images

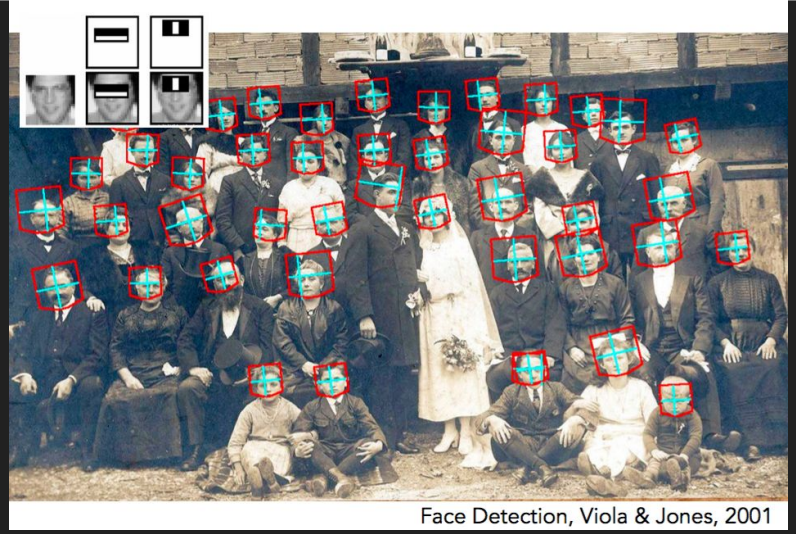
Output:
Scale
T-shirt
Steel drum
Drumstick
Mud turtle

✔

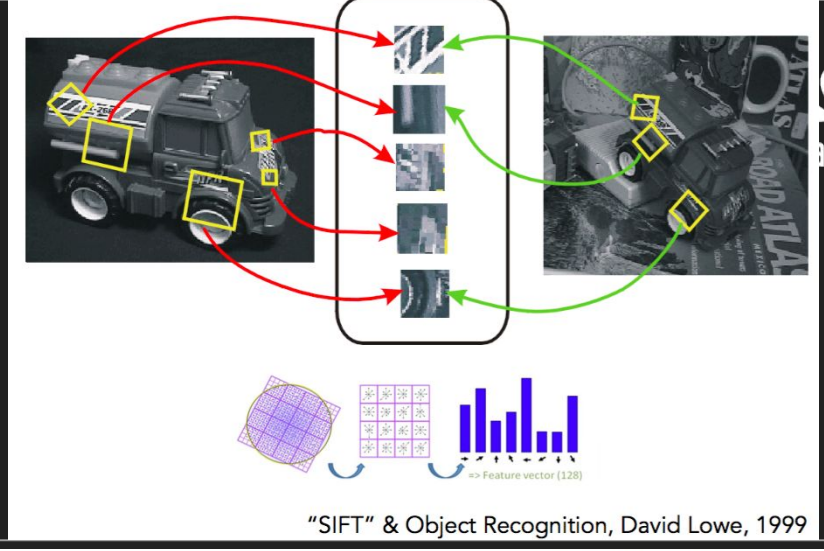
Output:
Scale
T-shirt
Giant panda
Drumstick
Mud turtle

✗

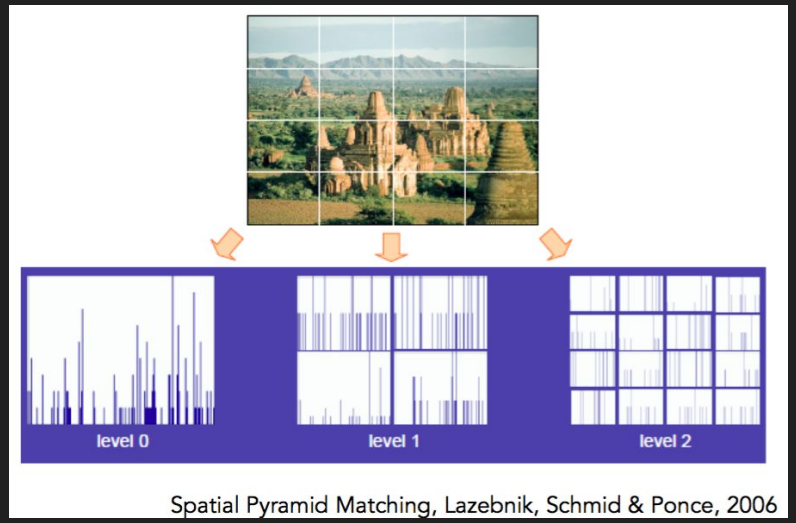
Russakovsky et al. arXiv, 2014



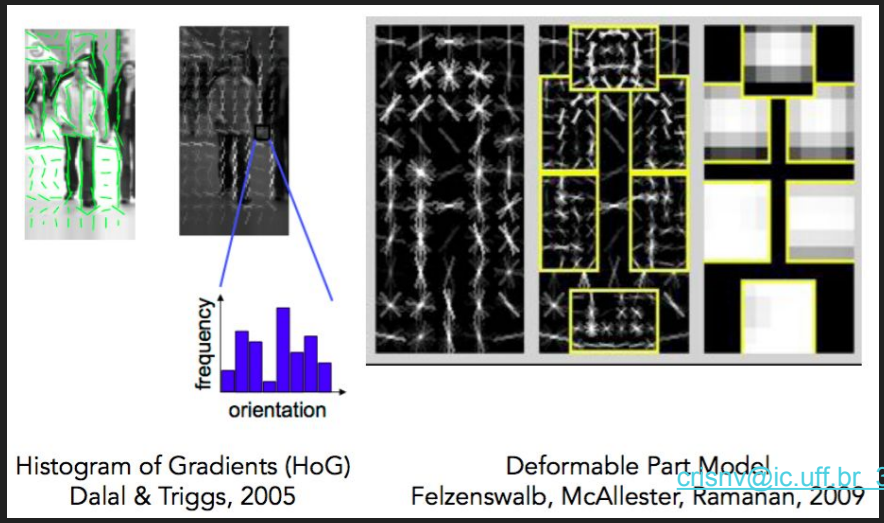
Face Detection, Viola & Jones, 2001



"SIFT" & Object Recognition, David Lowe, 1999



Spatial Pyramid Matching, Lazebnik, Schmid & Ponce, 2006



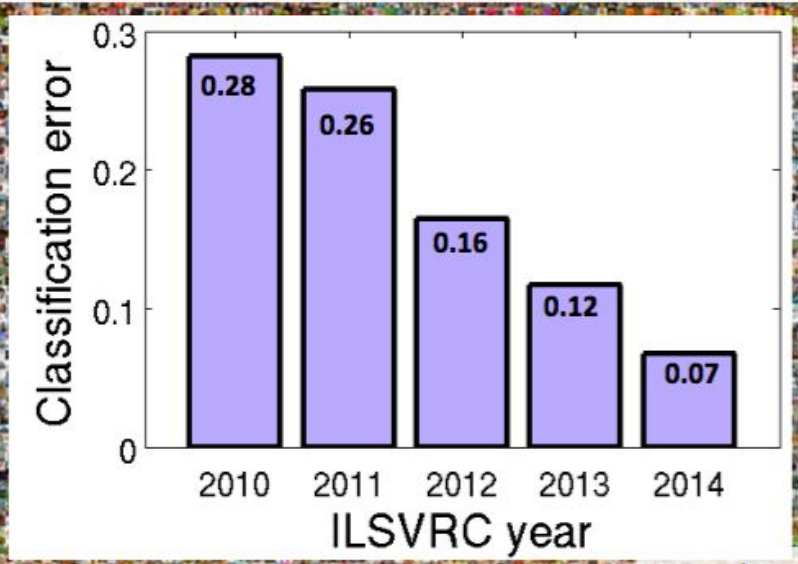
Histogram of Gradients (HoG)
Dalal & Triggs, 2005

Deformable Part Model
Felzenswalb, McAllester, Ramanan, 2009

IMAGENET Large Scale Visual Recognition Challenge

Instituto de Computação

The Image Classification Challenge:
1,000 object classes
1,431,167 images



abaixo de 0.04 2015

Russakovsky et al. arXiv, 2014

IMAGENET Large Scale Visual Recognition Challenge

Year 2010

NEC-UIUC

Dense grid descriptor: HOG, LBP

Coding: local coordinate, super-vector

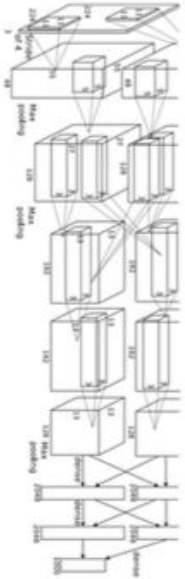
Pooling, SPM

Linear SVM

[Lin CVPR 2011]

Year 2012

SuperVision



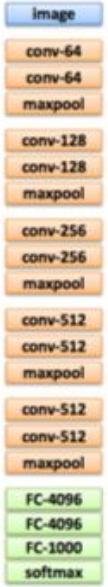
[Krizhevsky NIPS 2012]

Year 2014

GoogLeNet

[Szegedy arxiv 2014]

VGG



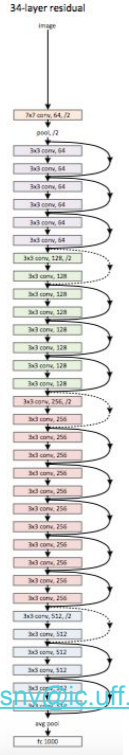
[Simonyan arxiv 2014]

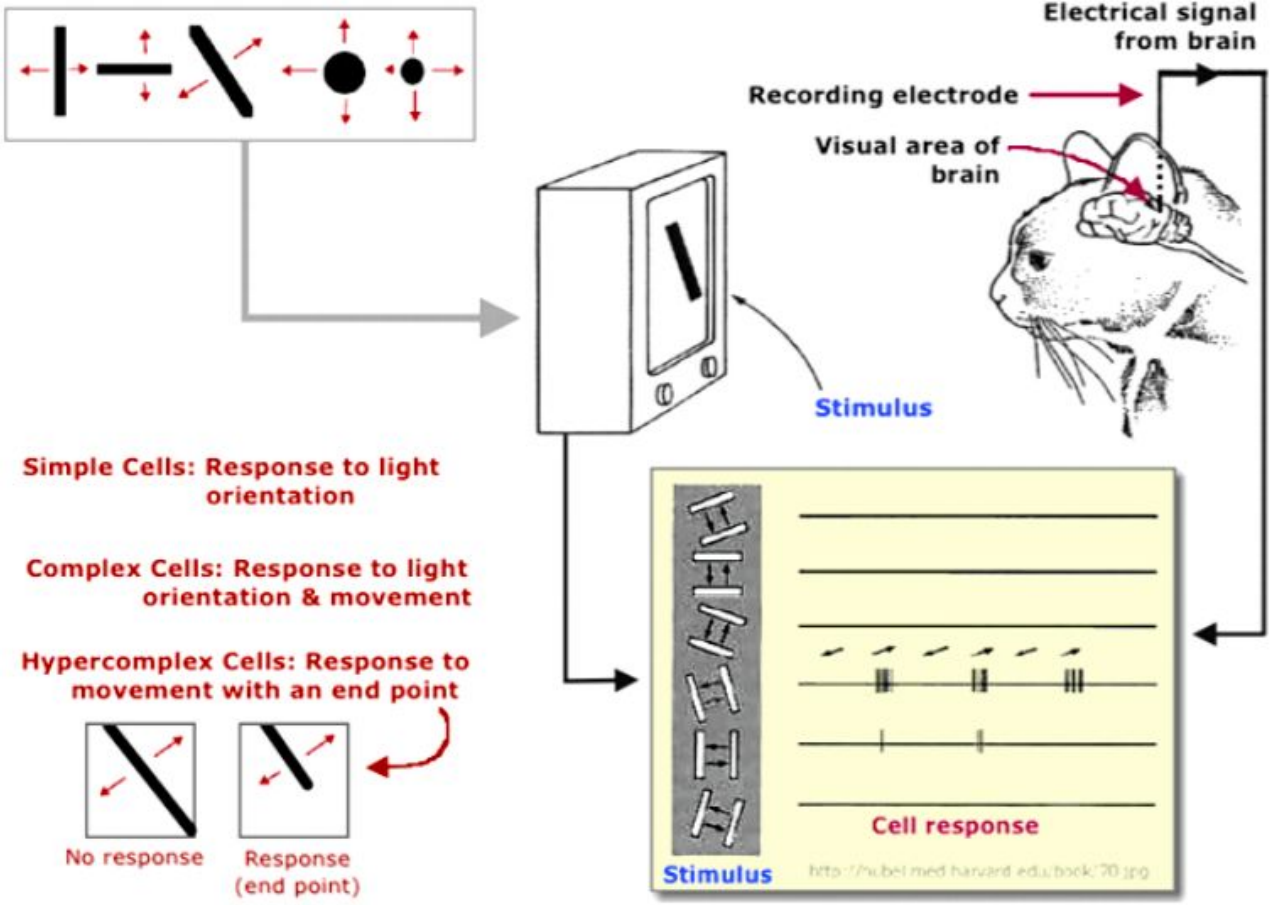
MSRA

[He arxiv 2014]

Year 2015

MSRA

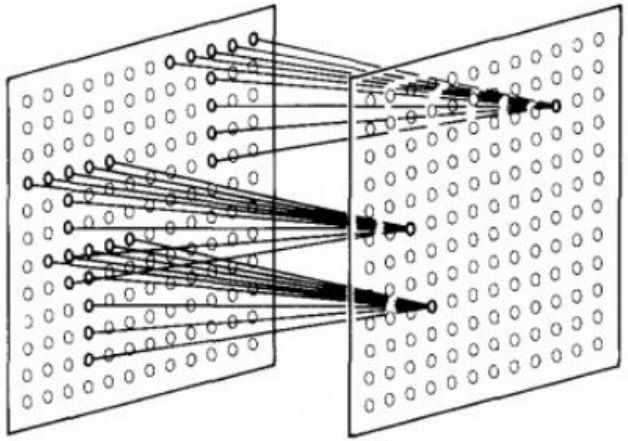




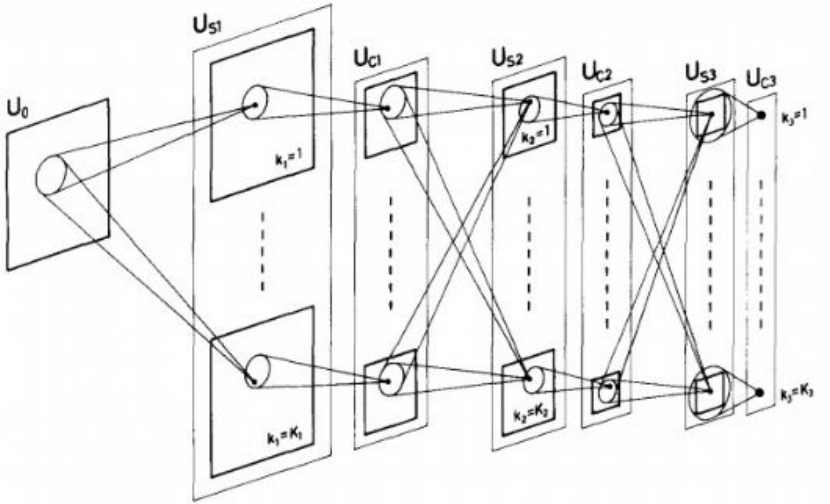
Hubel & Wiesel, 1959

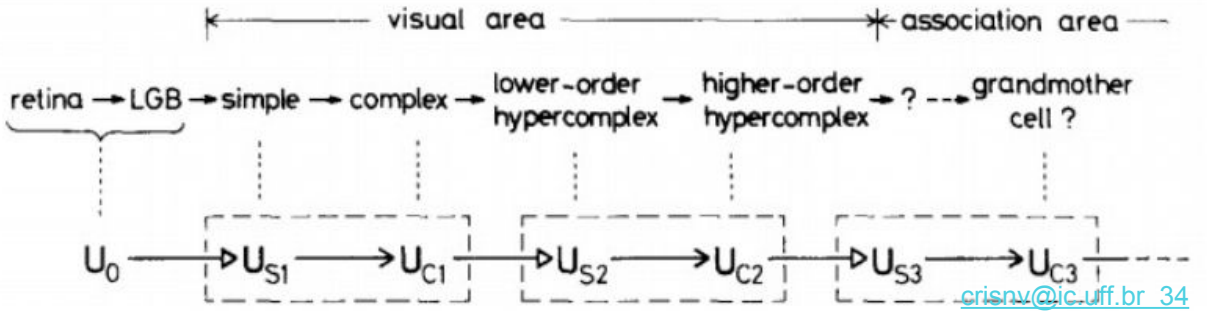
A bit of history:

Neurocognitron
[Fukushima 1980]



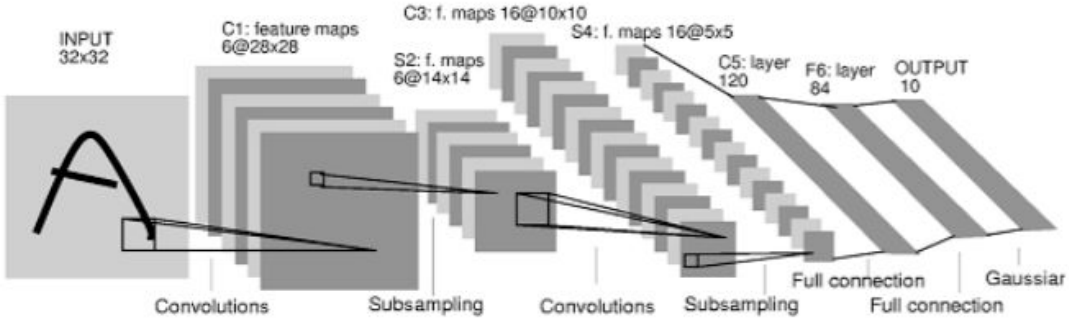
"sandwich" architecture (SCSCSC…)
simple cells: modifiable parameters
complex cells: perform pooling





1998

LeCun et al.



# of transistors

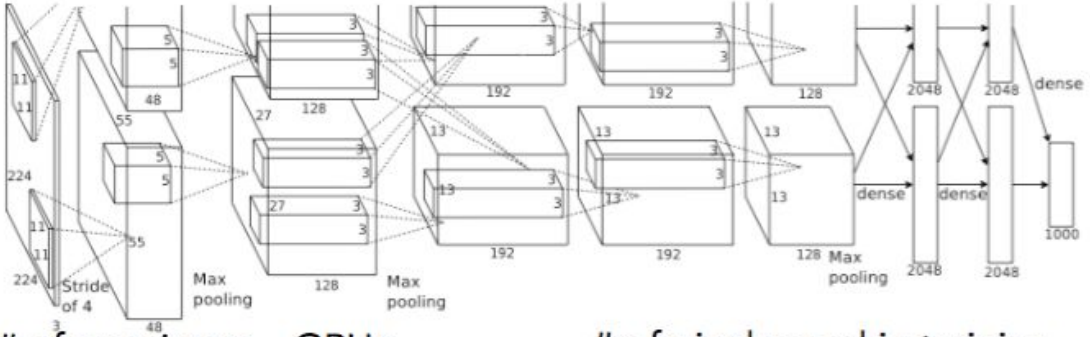
10^6

# of pixels used in training

10^7 NIST

2012

Krizhevsky et al.



# of transistors GPUs

10^9

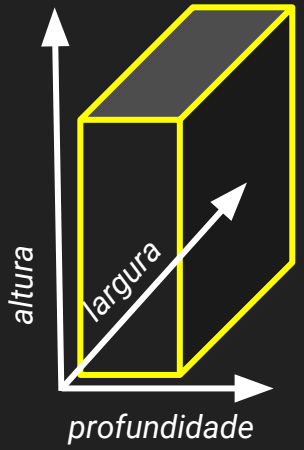
# of pixels used in training

10^{14} IMAGENET

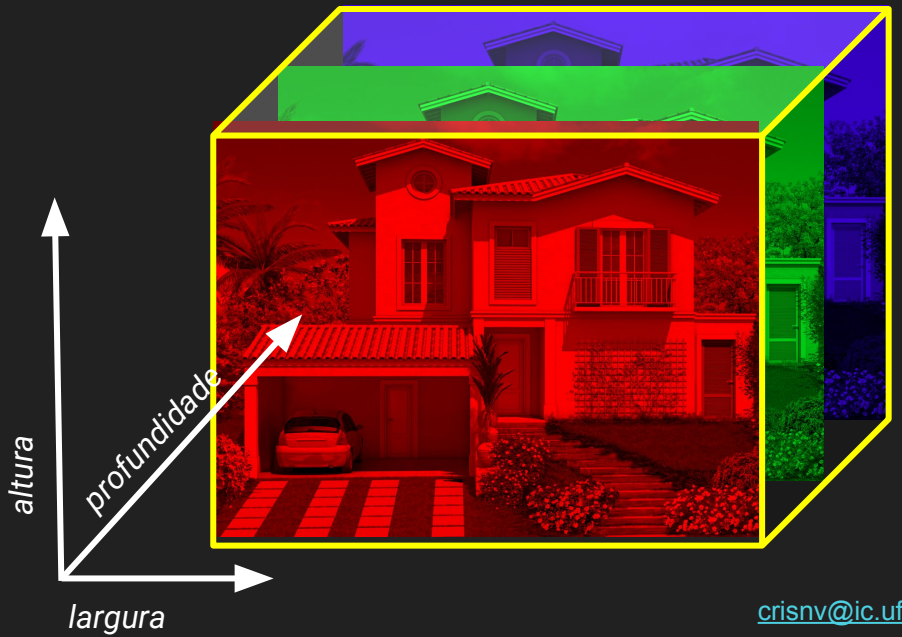
Componentes CNN:

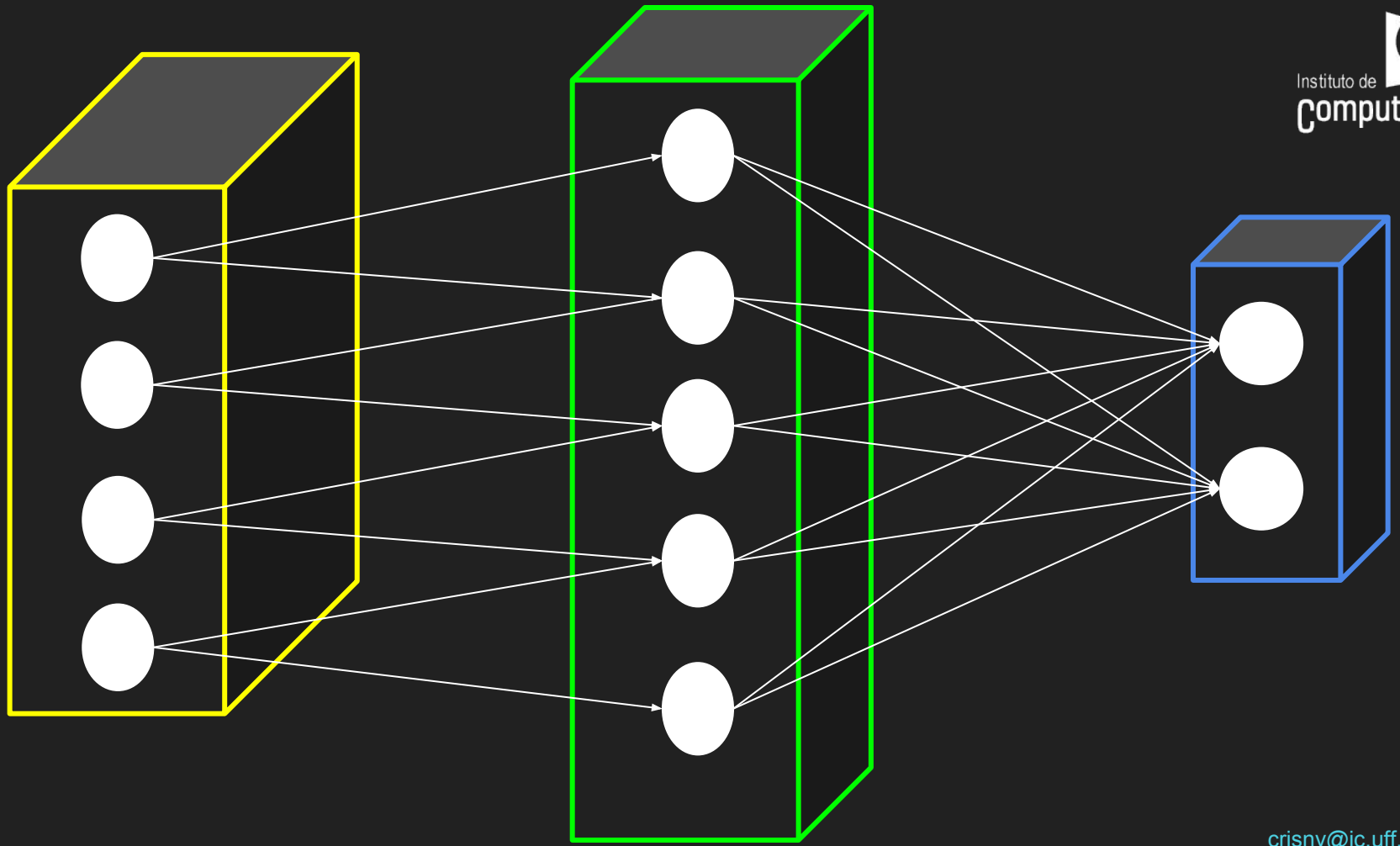
Neurônios e mapas de *features* dispostos em 3 dimensões:

altura x largura X profundidade



exemplo: camada de entrada

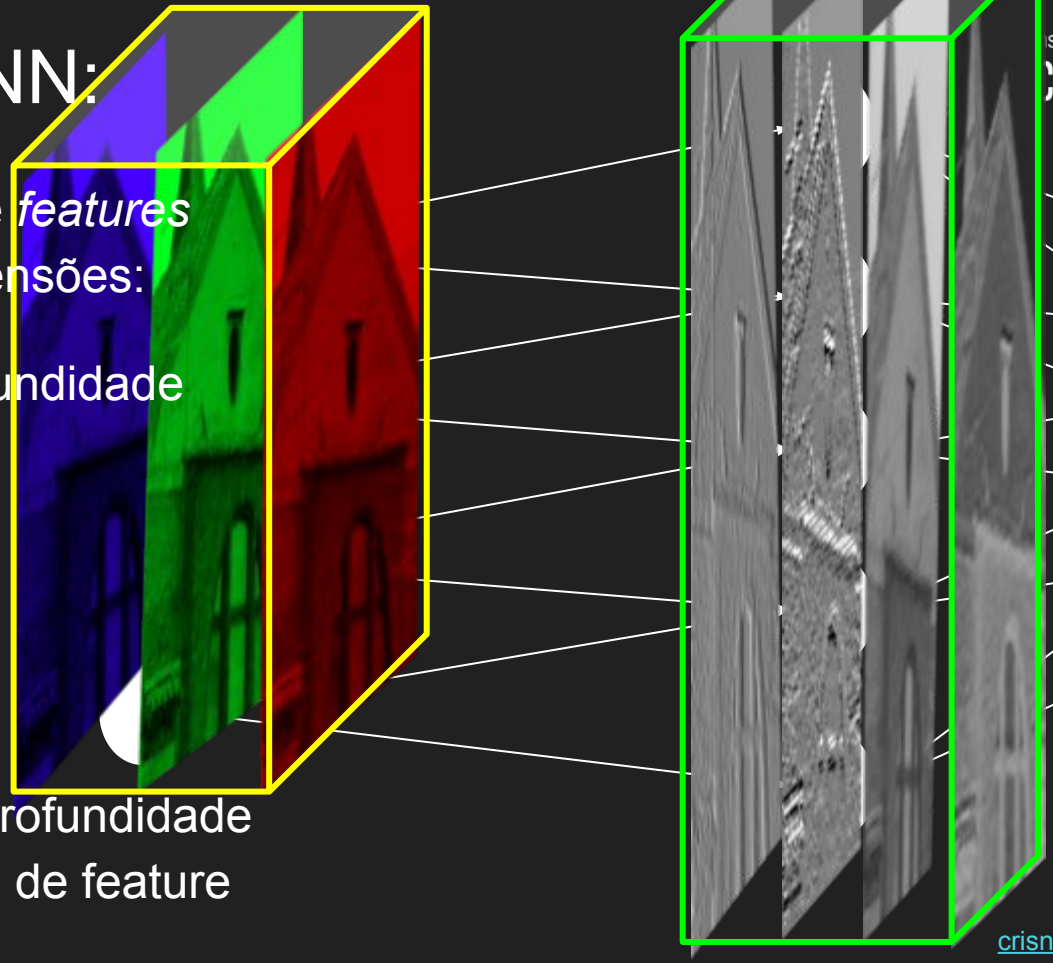




Componentes CNN:

Neurônios e mapas de *features* dispostos em 3 dimensões:

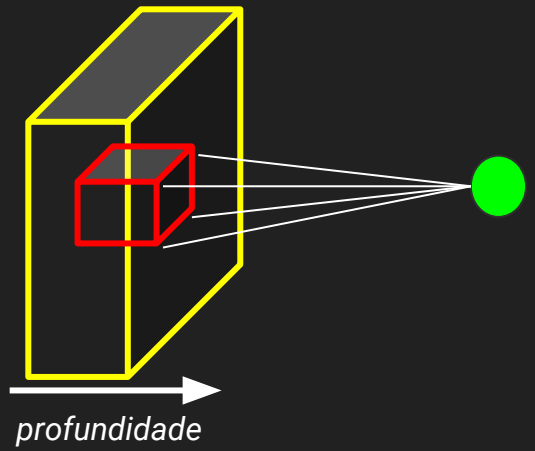
altura x largura X profundidade

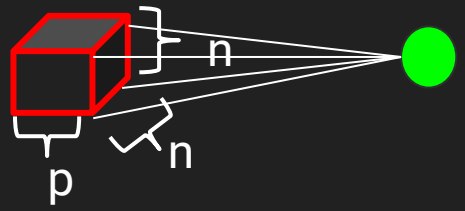


Demais camadas: cada profundidade representando um mapa de feature

Componentes CNN: campos receptivos locais

Conectividade Local: um neurônio em uma camada é ligado apenas a um subconjunto de saídas da camada anterior



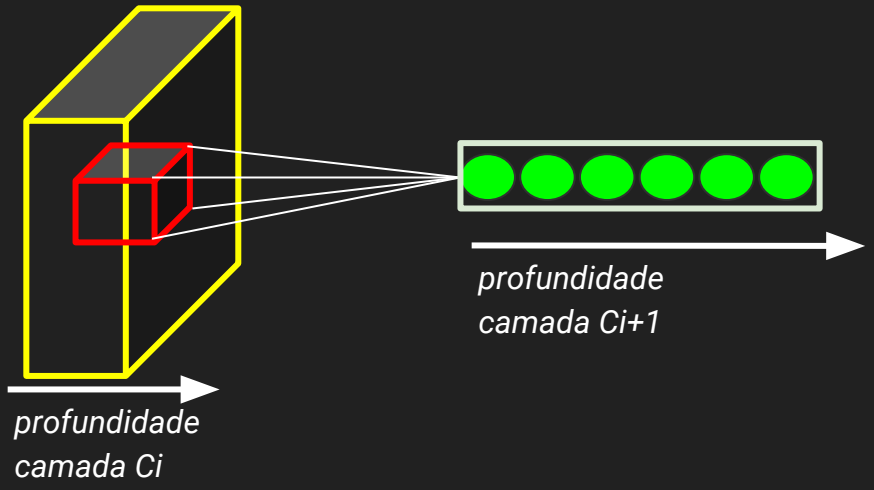


onde: a conectividade é local no plano de features*, mas completa em profundidade*

Logo, um neurônio combina suas estradas usando n*n*p pesos

Componentes CNN:

Conectividade local, múltiplas *features*

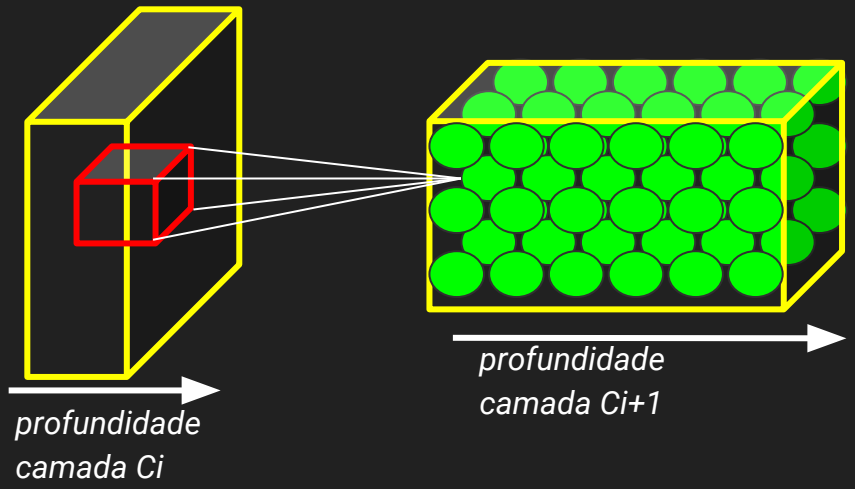


*múltiplos neurônios podem ser **conectados a mesma região** do volume de entrada para sua camada, organizados na dimensão **profundidade da sua grade** responsáveis por diferentes **features***

Exemplo: produz volume com profundidade 6

Componentes CNN:

Conectividade Local: **replicamos** o conjunto de neurônios ao longo das outras duas dimensões do espaço (**altura e largura**) escolhendo um **passo** para essa replicação (stride)



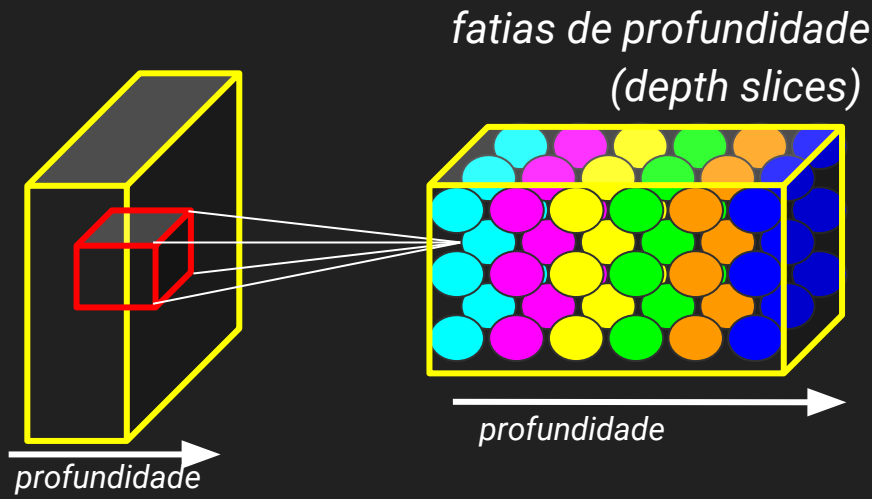
*ex: ao escolher passo 1 criamos um volume com mesma altura e largura da camada anterior**

tratamento de borda: preenchimento (padding**)*

crisnv@ic.uff.br

Componentes CNN:

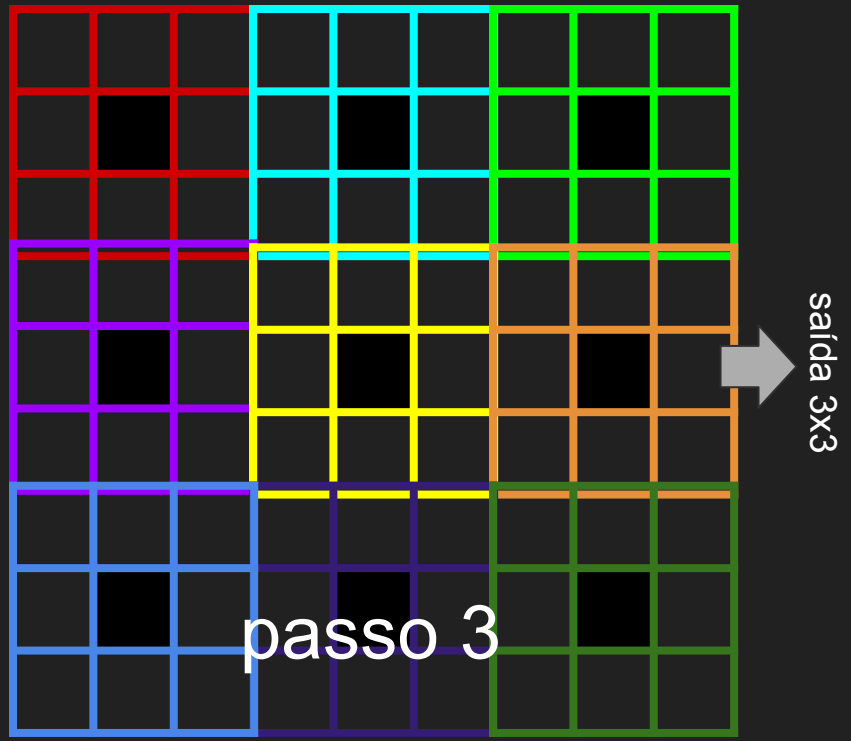
- Compartilhamento de pesos (parâmetros) entre neurônios da mesma camada, na mesma profundidade;
- A saída produzida por uma fatia de profundidade é um mapa de ativação

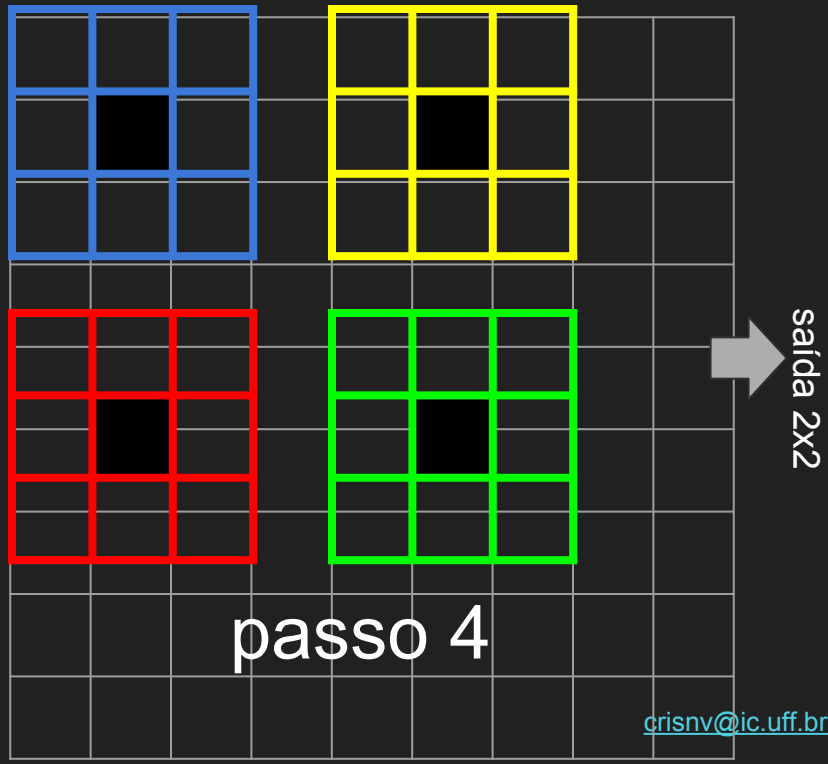


Repartição diminui drasticamente o número de parâmetros por fatia

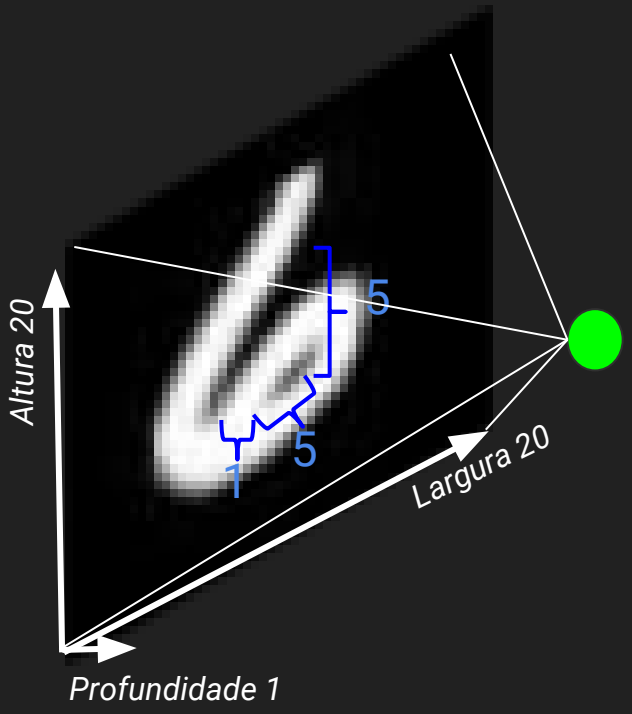
crisnv@ic.uff.br

Exemplos: dado 9x9, filtro 3x3





Cálculo de parâmetros: exemplo



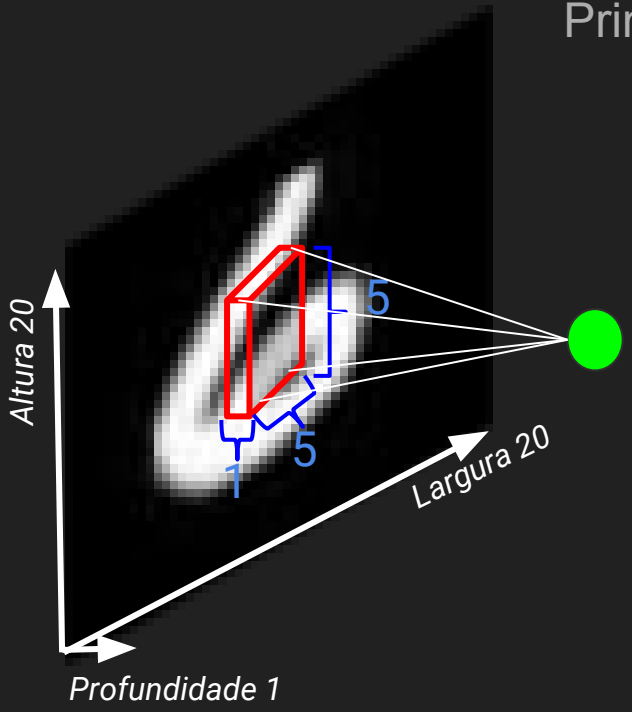
Entrada: 20x20x1

Primeira camada oculta na forma de camada completamente conectada (sem compartilhamento de pesos e sem campo perceptivo):

- cada neurônio possui 20*20 pesos = 400 pesos
- se fizemos um neurônio por pixel para criar uma única camada: 400*400 = 160.000 pesos!!

Cálculo de parâmetros:exemplo

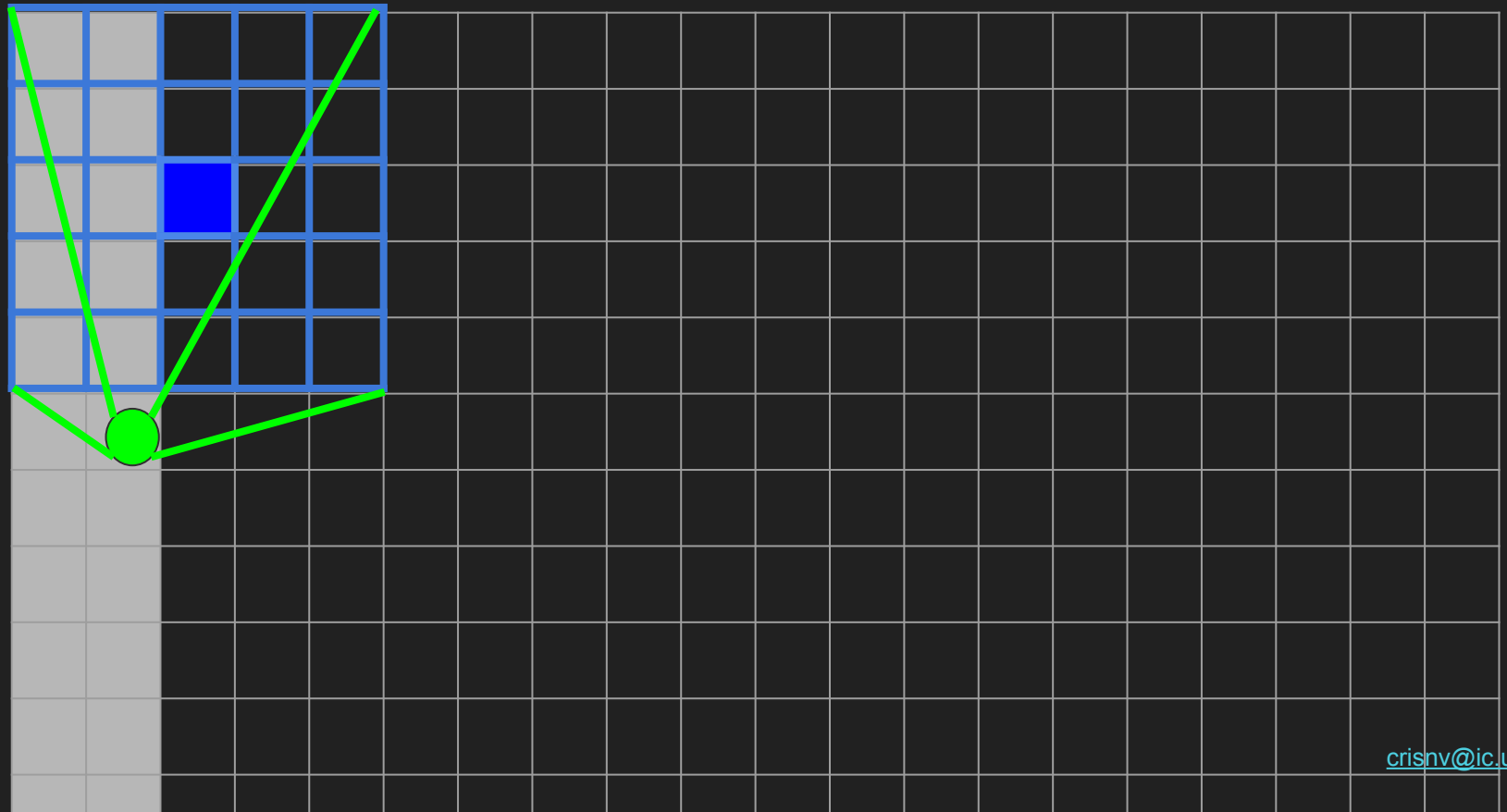
Entrada: 20x20x1



Primeira camada oculta, **sem** compartilhamento de pesos:

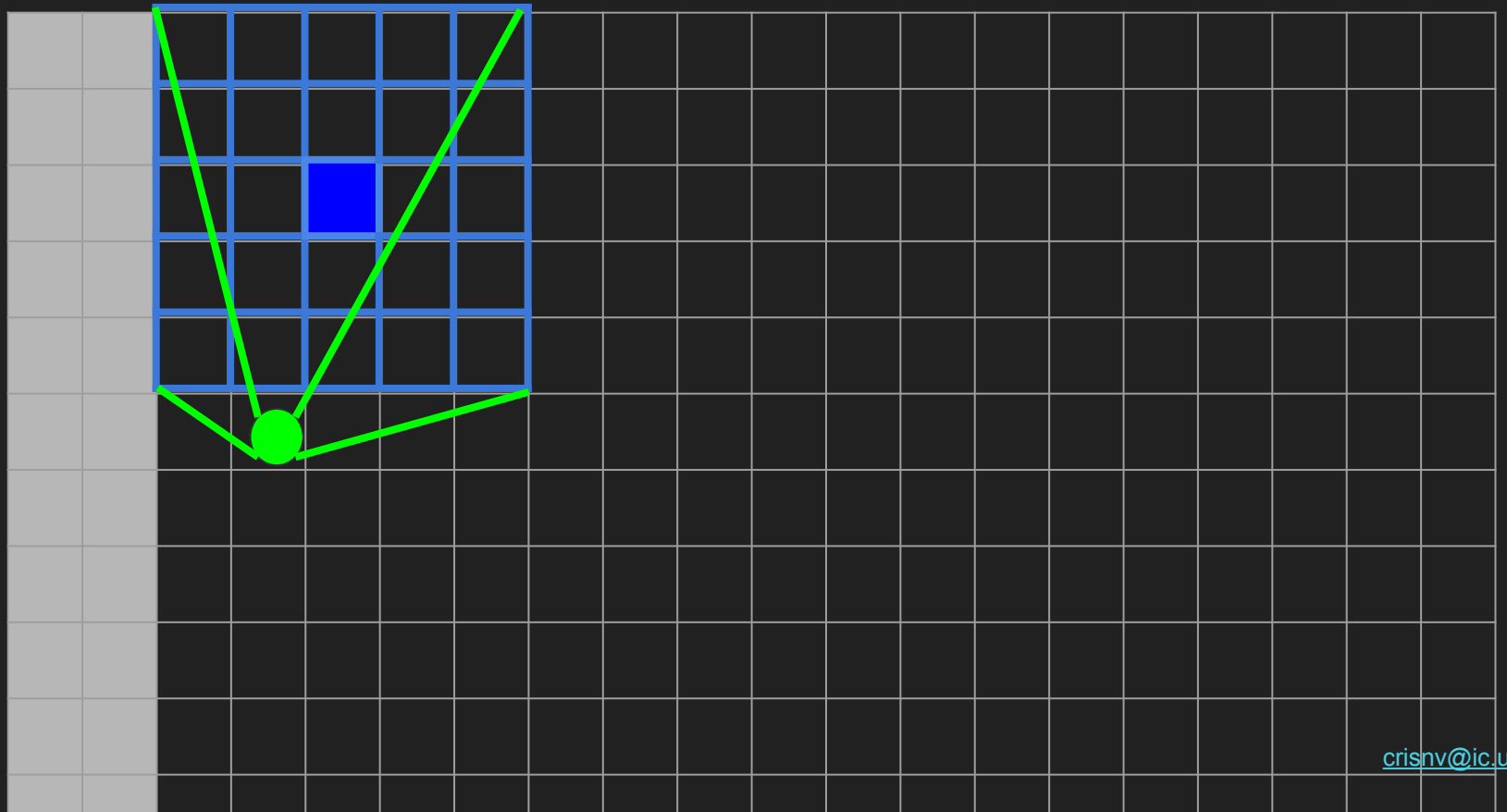
- campo receptivo 5x5:
  cada neurônio: 5x5*1: 25 pesos
- neurônios dispostos com passo 1 e preenchimento 0:
  Total por fatia?
- camada oculta com 10 de profundidade:
  Total na camada?

Exemplo: passo 1 sem preenchimento

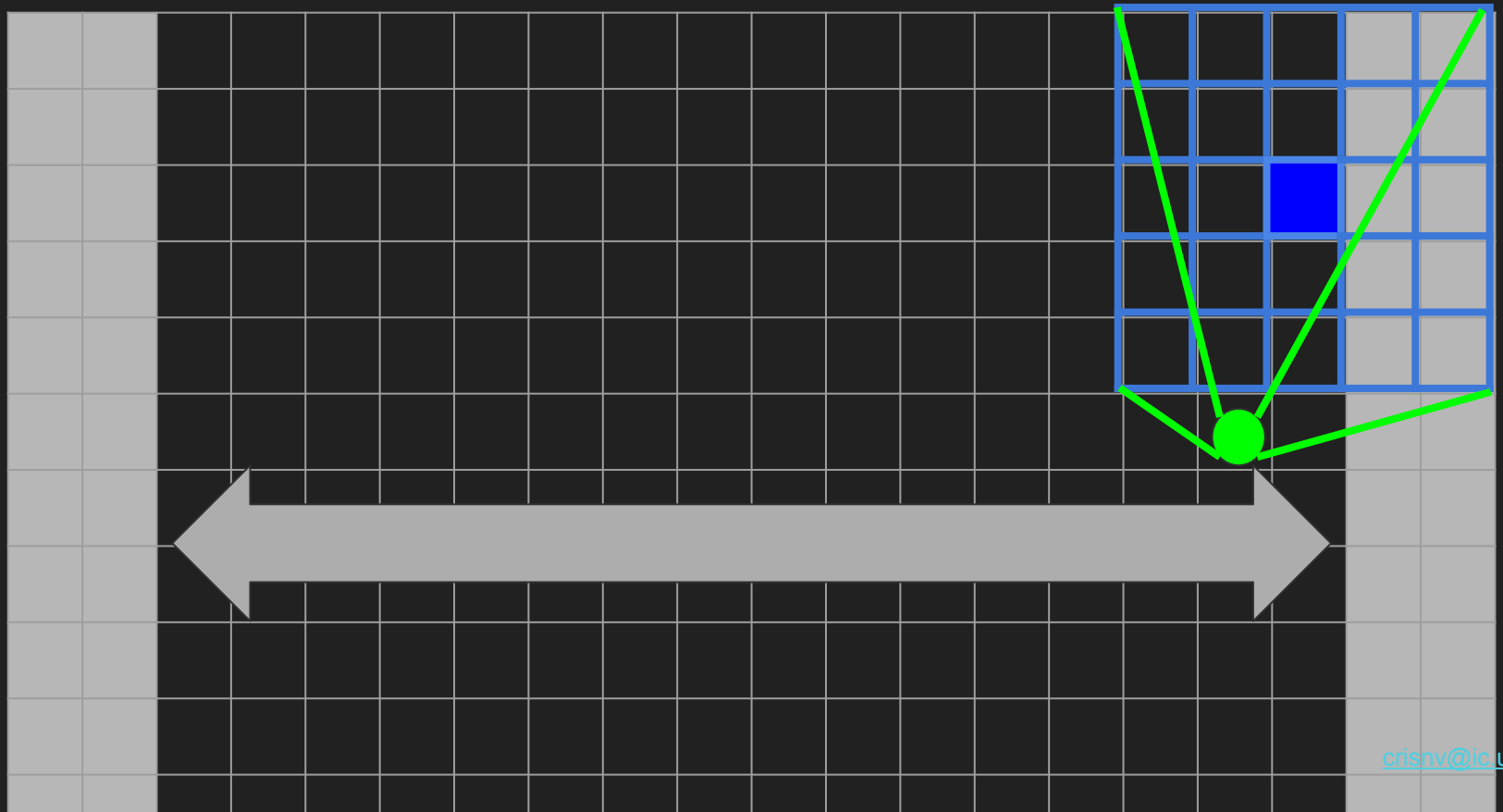


Exemplo: passo 1 sem preenchimento

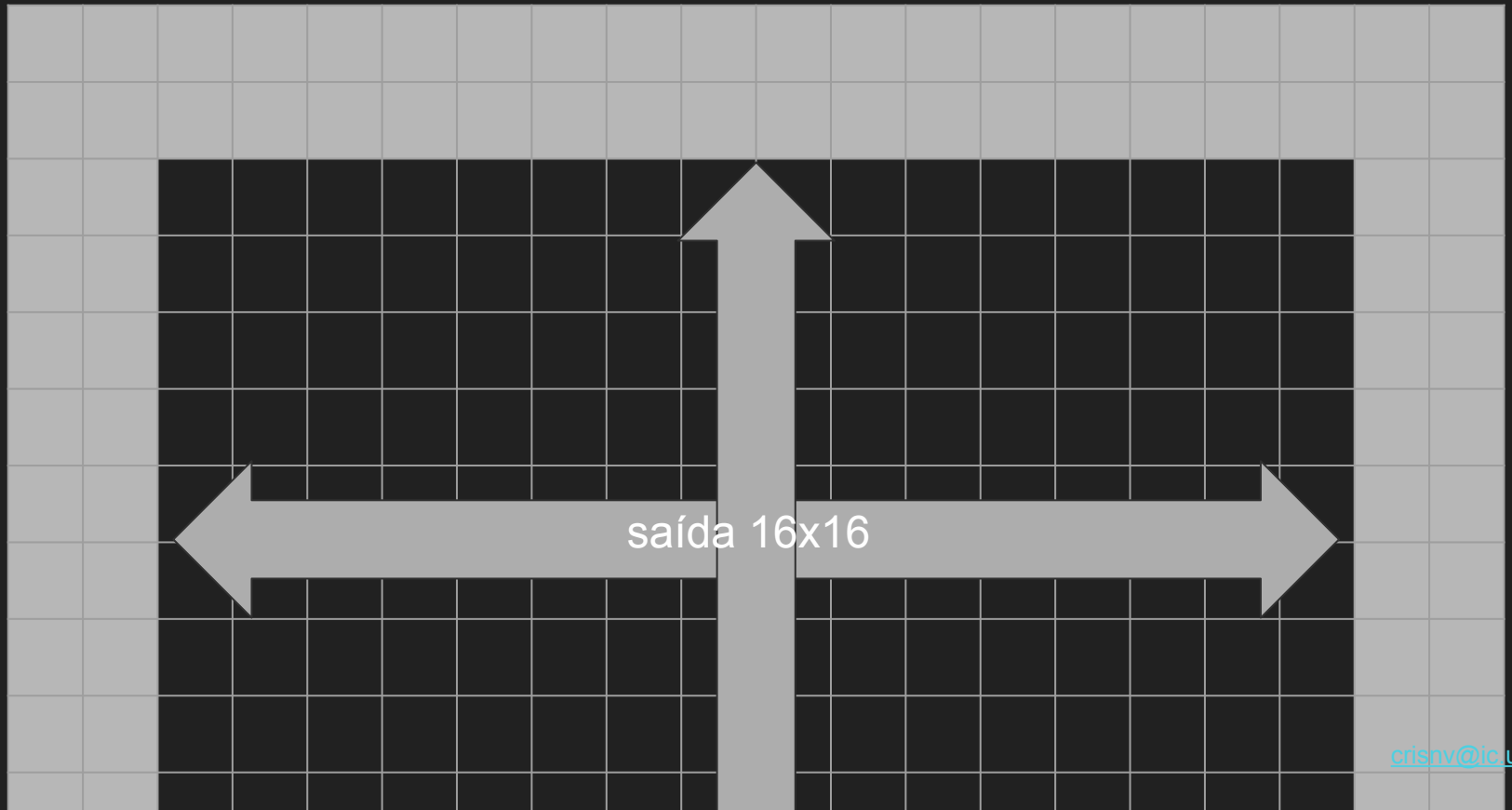
Exemplo: passo 1 sem preenchimento



Exemplo: passo 1 sem preenchimento

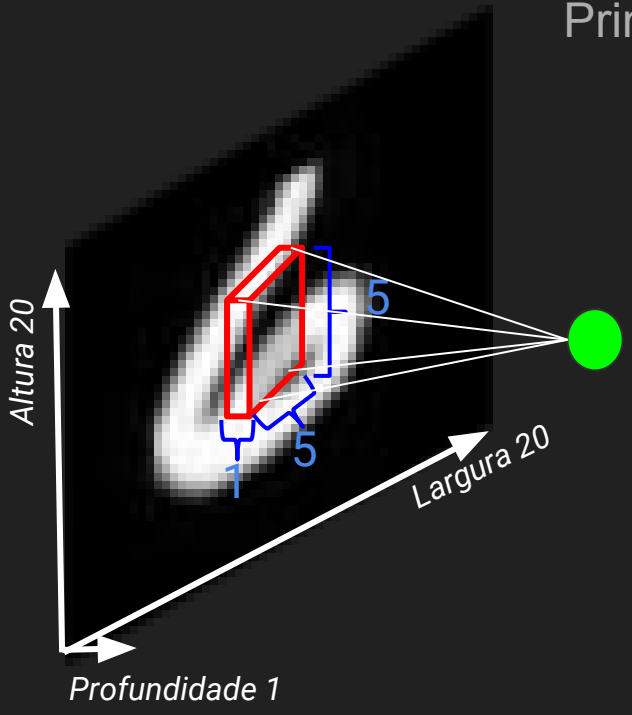


Exemplo: passo 1 sem preenchimento



Cálculo de parâmetros:exemplo

Entrada: 20x20x1

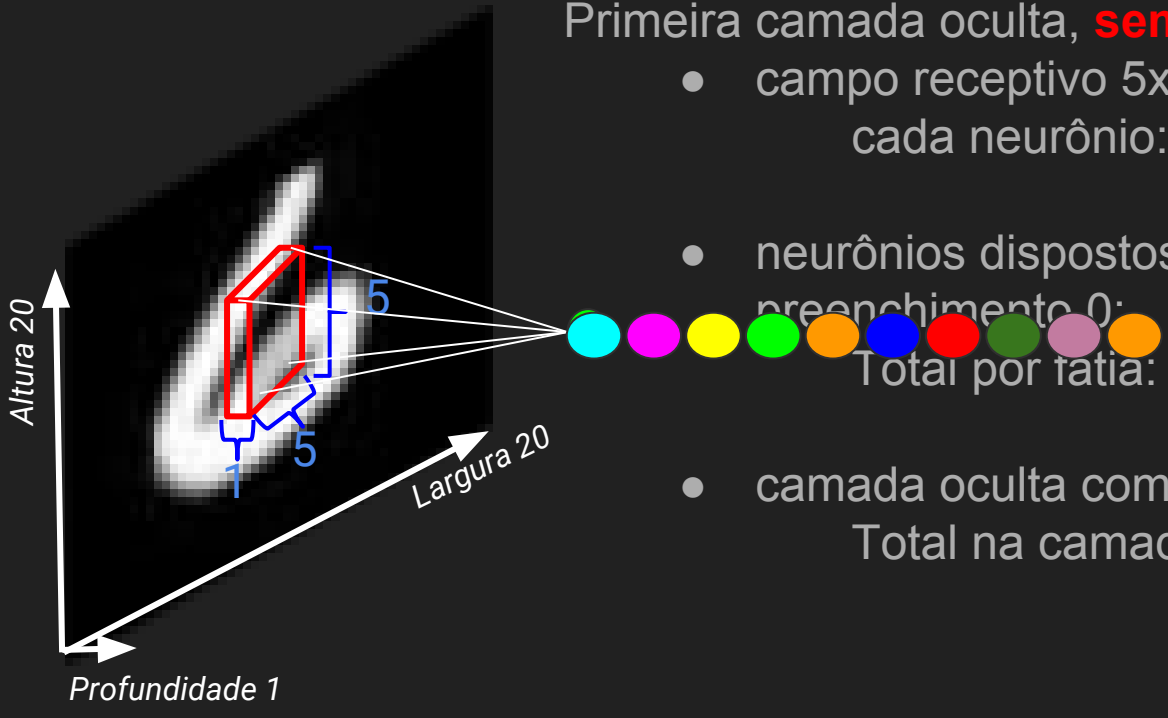


Primeira camada oculta, **sem** compartilhamento de pesos:

- campo receptivo 5x5:
  cada neurônio: 5x5*1: 25 pesos
- neurônios dispostos com passo 1 e preenchimento 0:
  Total por fatia: 16x16
- camada oculta com 10 de profundidade:
  Total na camada: ?

Cálculo de parâmetros:exemplo

Entrada: 20x20x1

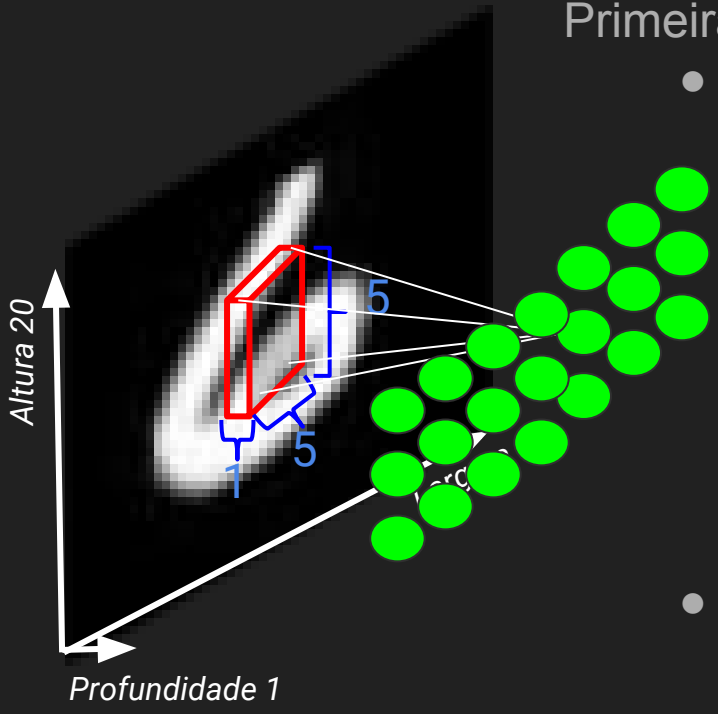


Primeira camada oculta, **sem** compartilhamento de pesos:

- campo receptivo 5x5:
  cada neurônio: 5x5*1: 25 pesos
- neurônios dispostos com passo 1 e preenchimento 0:
  Total por fatia: 16x16
- camada oculta com 10 de profundidade:
  Total na camada: 10*16*16*5*5*1 = **64.000 !!!**

Cálculo de parâmetros:exemplo

Entrada: 20x20x1



Primeira camada oculta, **com** compartilhamento de pesos:

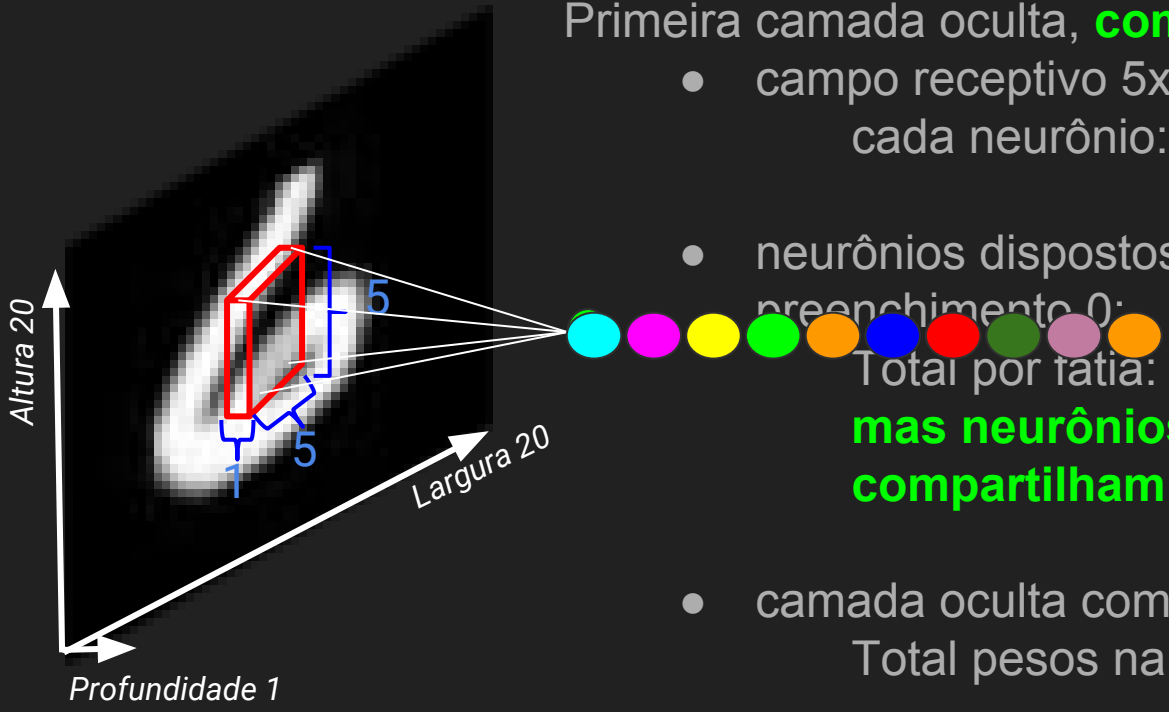
- campo receptivo 5x5:
  cada neurônio: 5x5*1: 25 pesos

  neurônios dispostos com passo 1 e preenchimento 0:
  Total por fatia: 16x16 neurônios,
  mas neurônios na mesma fatia compartilham os 25 pesos entre si

- camada oculta com 10 de profundidade:
  Total na camada: **?**

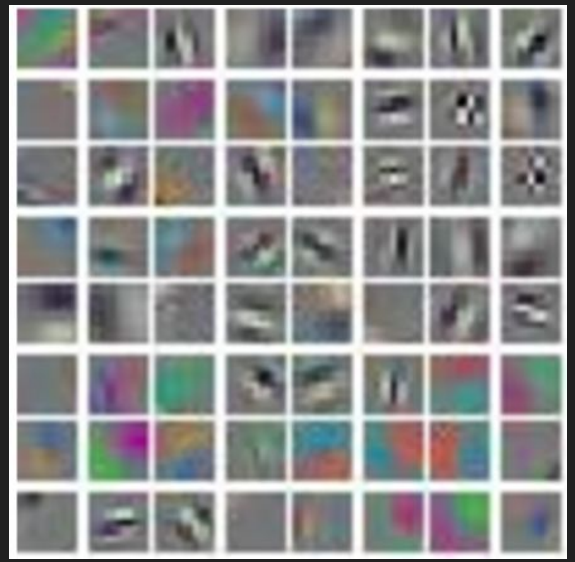
Cálculo de parâmetros:exemplo

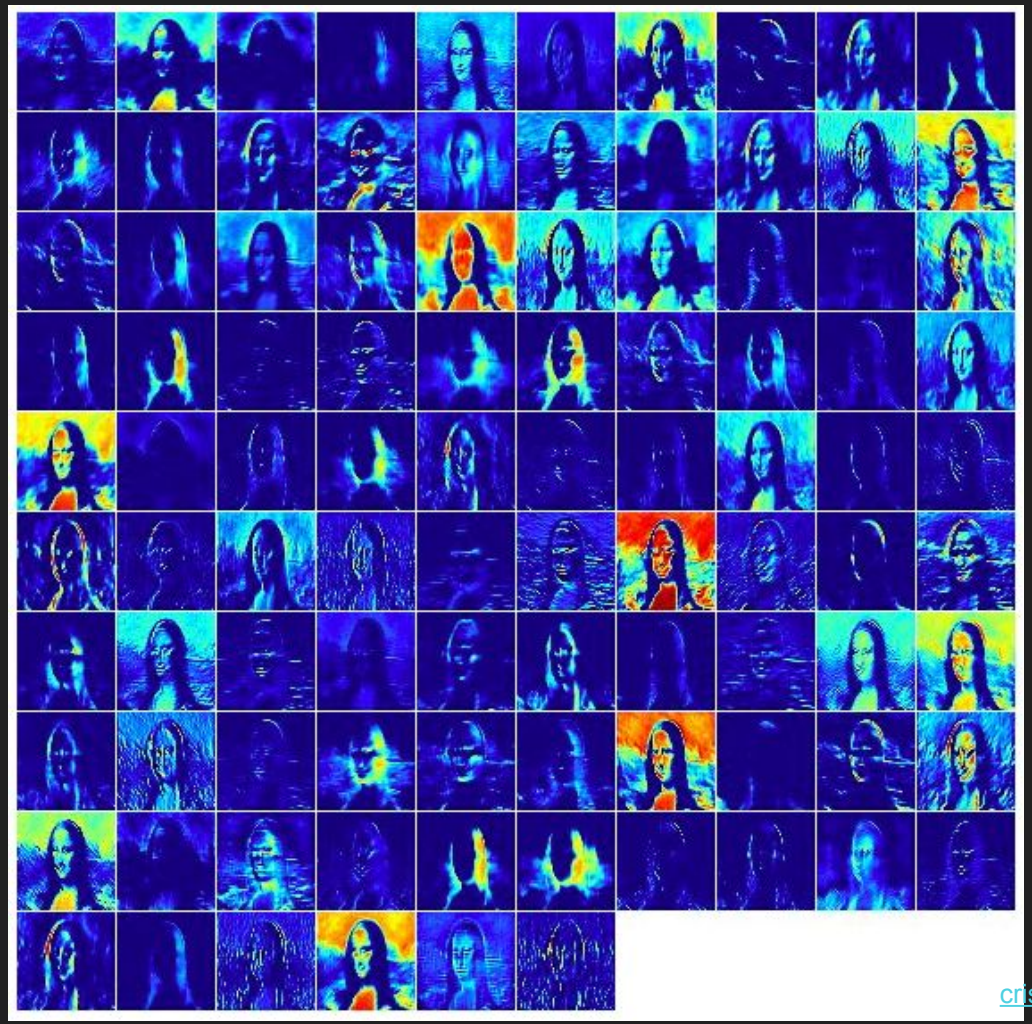
Entrada: 20x20x1



Primeira camada oculta, **com** compartilhamento de pesos:

- campo receptivo 5x5:
  cada neurônio: 5x5*1: 25 pesos
- neurônios dispostos com passo 1 e preenchimento 0:
  Total por fatia: 16x16 neurônios,
  mas neurônios na mesma fatia compartilham os 25 pesos entre si
- camada oculta com 10 de profundidade:
  Total pesos na camada: **250**





Instituto de Computação

Componentes CNN:

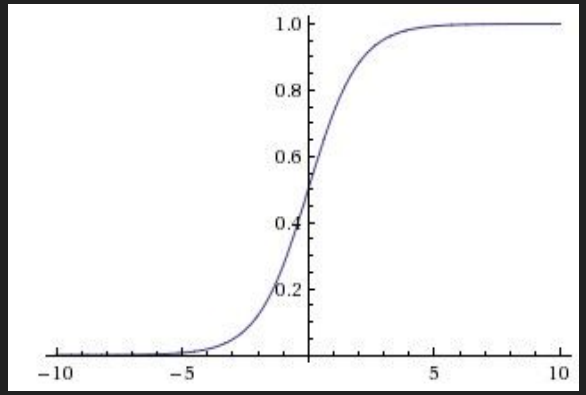
Assim, nas **camadas convolucionais** os neurônios:

1. São conectados apenas aos seus campos receptivos locais (conectividade local);
2. Usam o mesmo conjunto de parâmetros que seus vizinhos de mesma profundidade ou fatia. Tais pesos funcionam como pesos da filtragem sobre o sinal de entrada;
3. Juntas, diversas fatias de uma mesma camada produzem diversos mapas de ativação, respondendo a extração de múltiplas *features;*

Funções de ativação: mais usadas

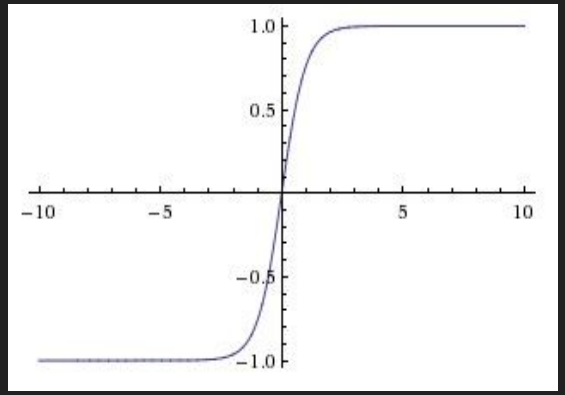
Instituto de Computação

Sigmoid



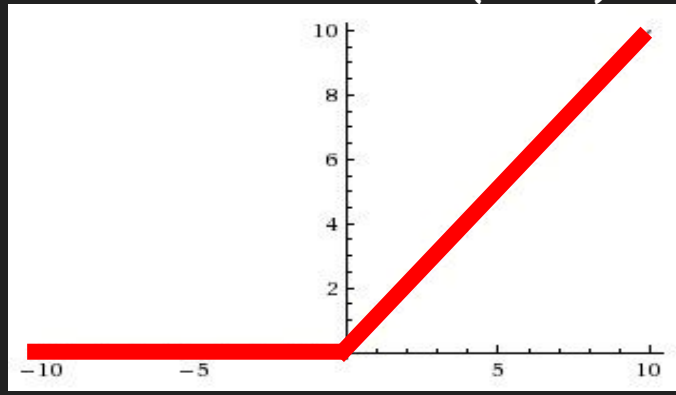
mapeia números reais para o intervalo [0,1]

Tanh



mapeia números reais para o intervalo [-1,1].

Rectified Linear Unit (ReLU)



$f(x) = \max(0, x)$

Camadas de agrupamento *(Pooling layers)*

Camadas de agrupamento *(Pooling layers)*

Camadas de agrupamento *(Pooling layers)*

Camadas de agrupamento *(Pooling layers)*

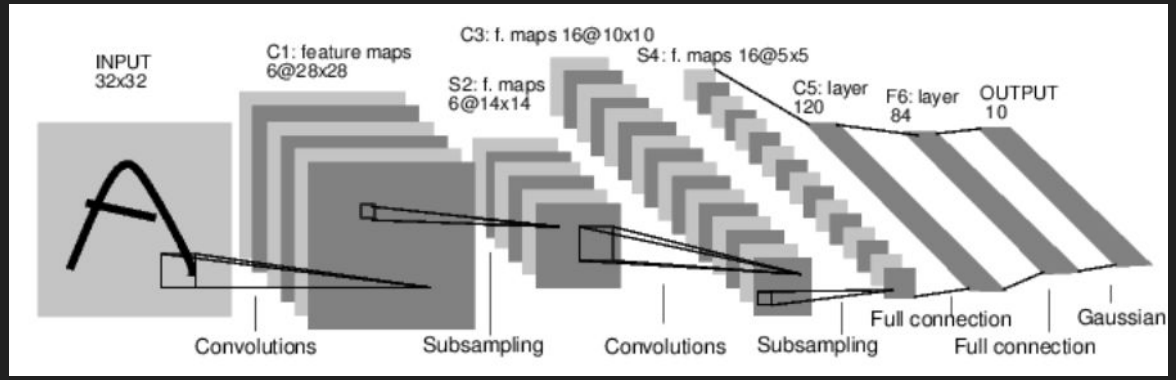
Camada de saída

(# neurônios = # classes)

Saída como vetor de probabilidades -> softmax otimizada com cross-entropy

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

LeNet-5 (1989)



Altura x largura

32x32

6 mapas com filtros de 11x11,passo 4

28x28

Max-pooling 2x2, passo 2

14x14

16 mapas com filtros de 5x5

10x10

Max-pooling 2x2, passo 2

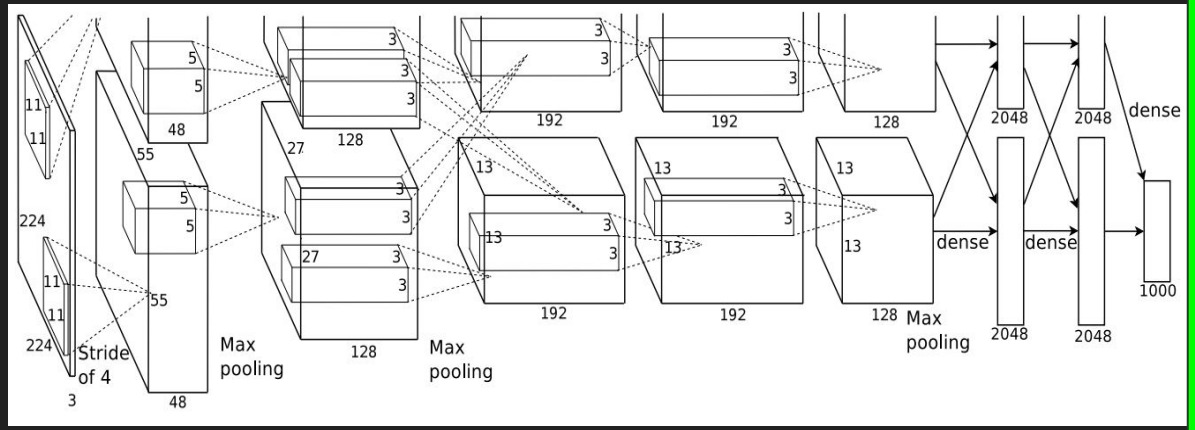
5x5

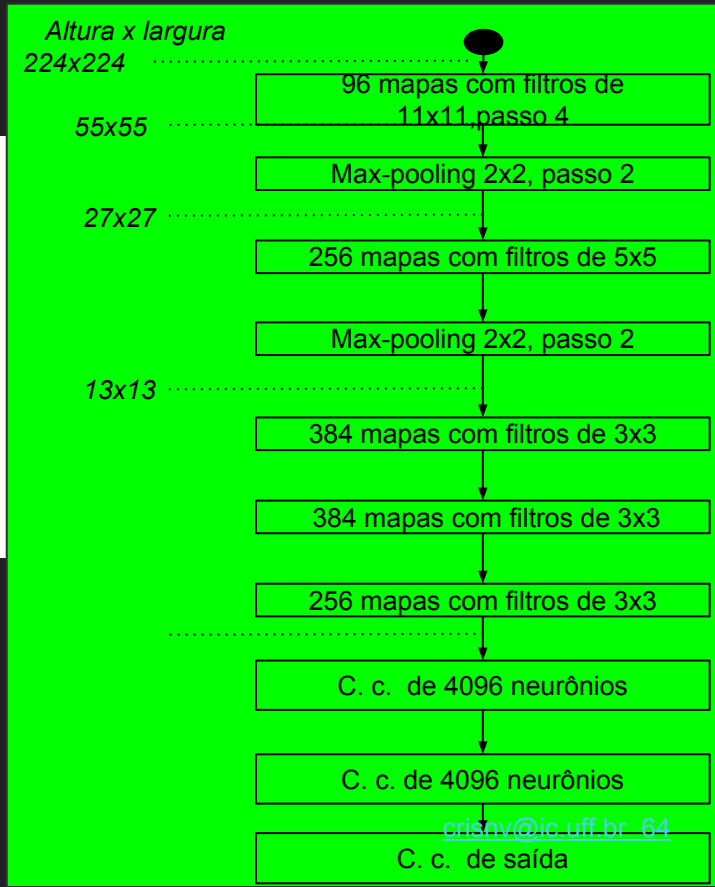
Completamente conectada de 120 neurônios

Completamente conectada de 84 neurônios

completamente conectada de saída (# neurônios = # classes)

AlexNet (2012)





ZFnet (2013)

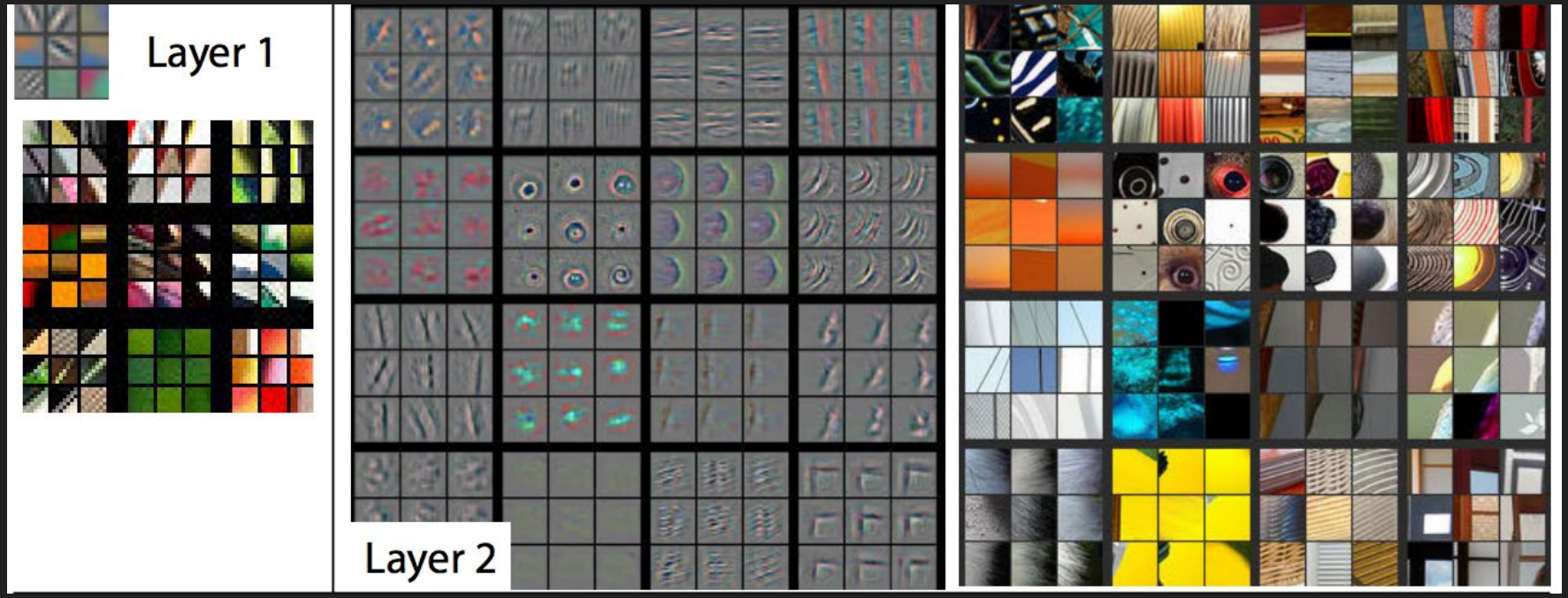
Instituto de Computação

Melhorias sobre os ***hiperparâmetros*** que definem a AlexNet:

- aumentou o número de fatias nas camadas intermediárias expandindo o tamanho das camadas de convolu ̧ca ̃o intermedia ́rias
- reduzindo a passada e o tamanho dos filtros da primeira camada de convolução
- Observa que a retirada das camadas completamente conectadas aumenta pouco o erro

Deconvnet: camadas de reversão da rede que permitem a visualização das ativações produzidas pelos mapas de *features* ao longo da hierarquia de camadas da CNN.

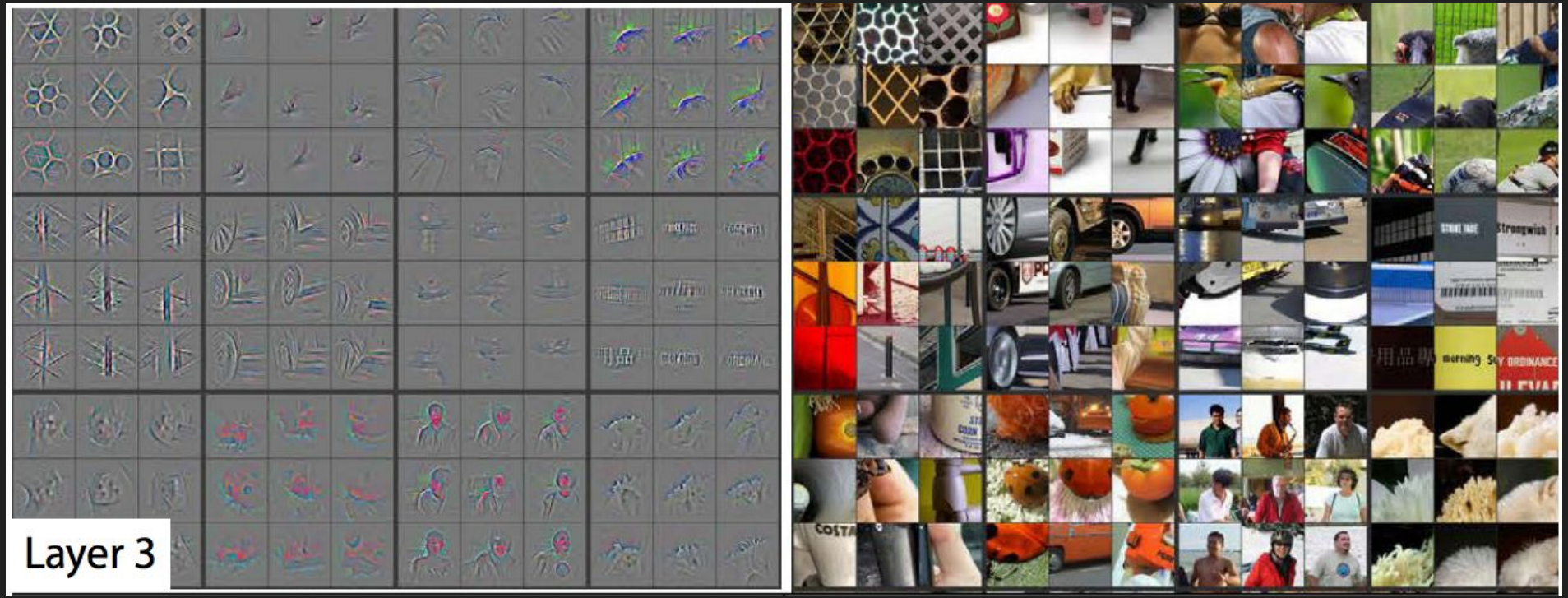
Visualizando as saídas de ativação



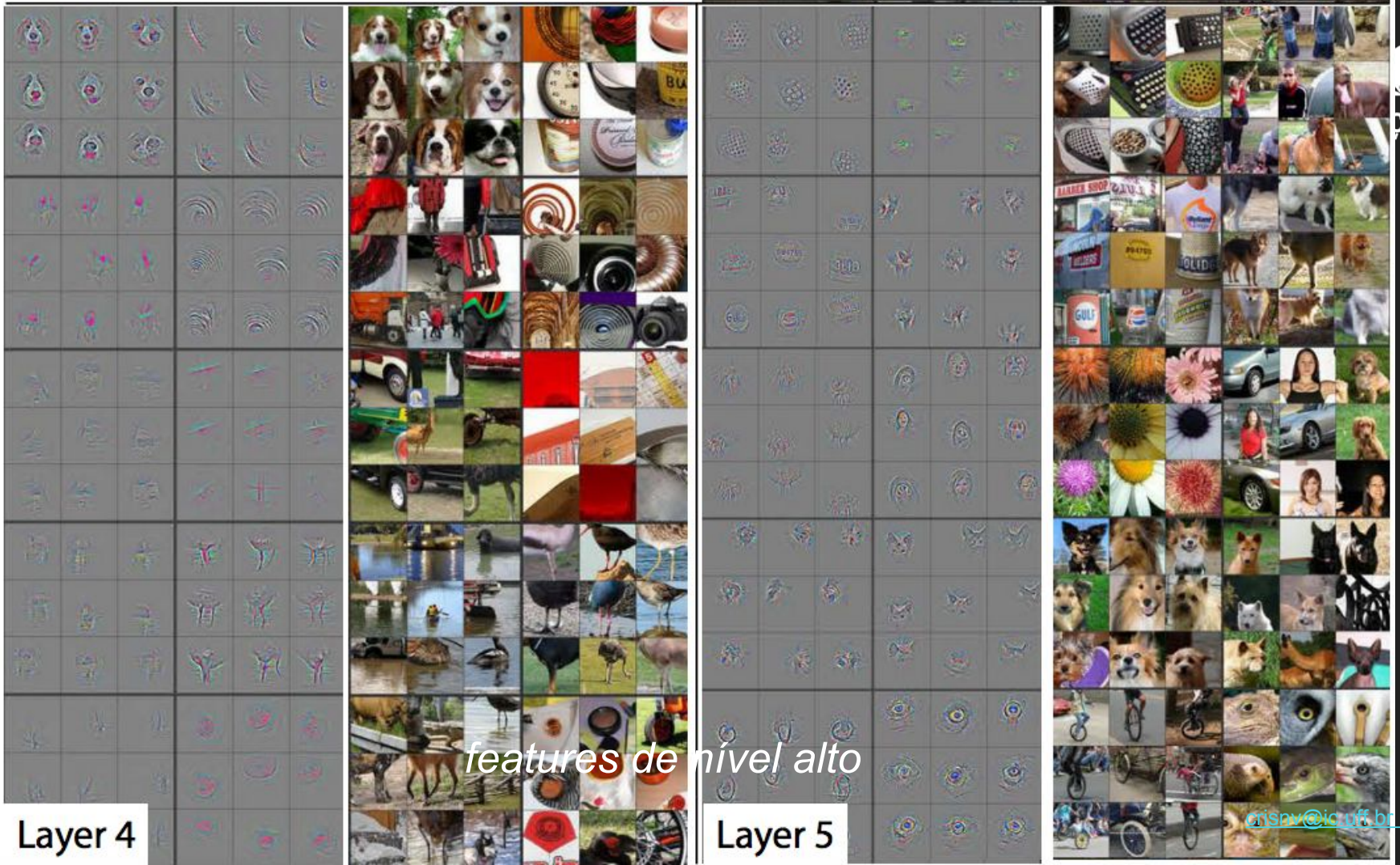
features de nível baixo

D. Zeiler, R. Fergus. Visualizing and Understanding Convolutional Networks. ECCV 2014

Visualizando as saídas de ativação

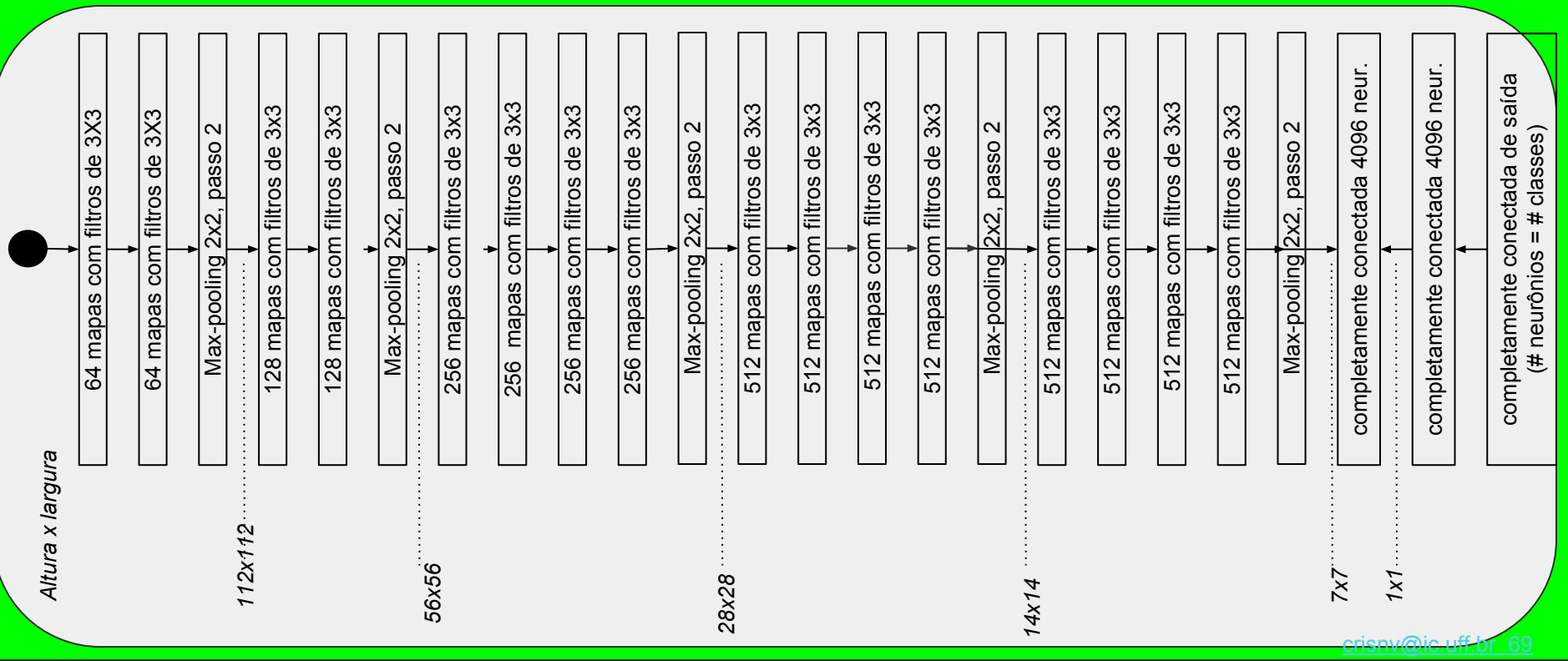


features de nível médio

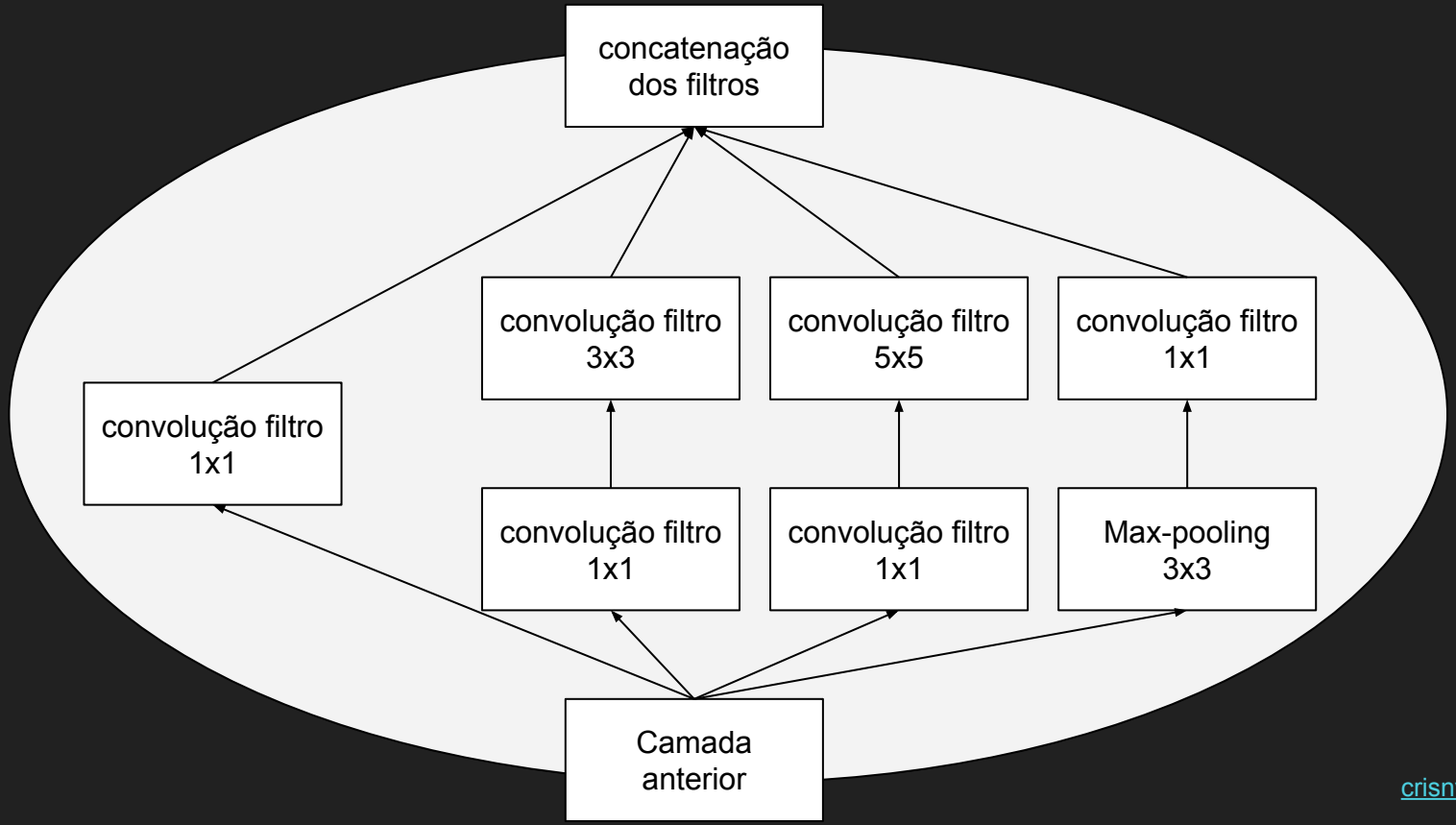


features de nível alto

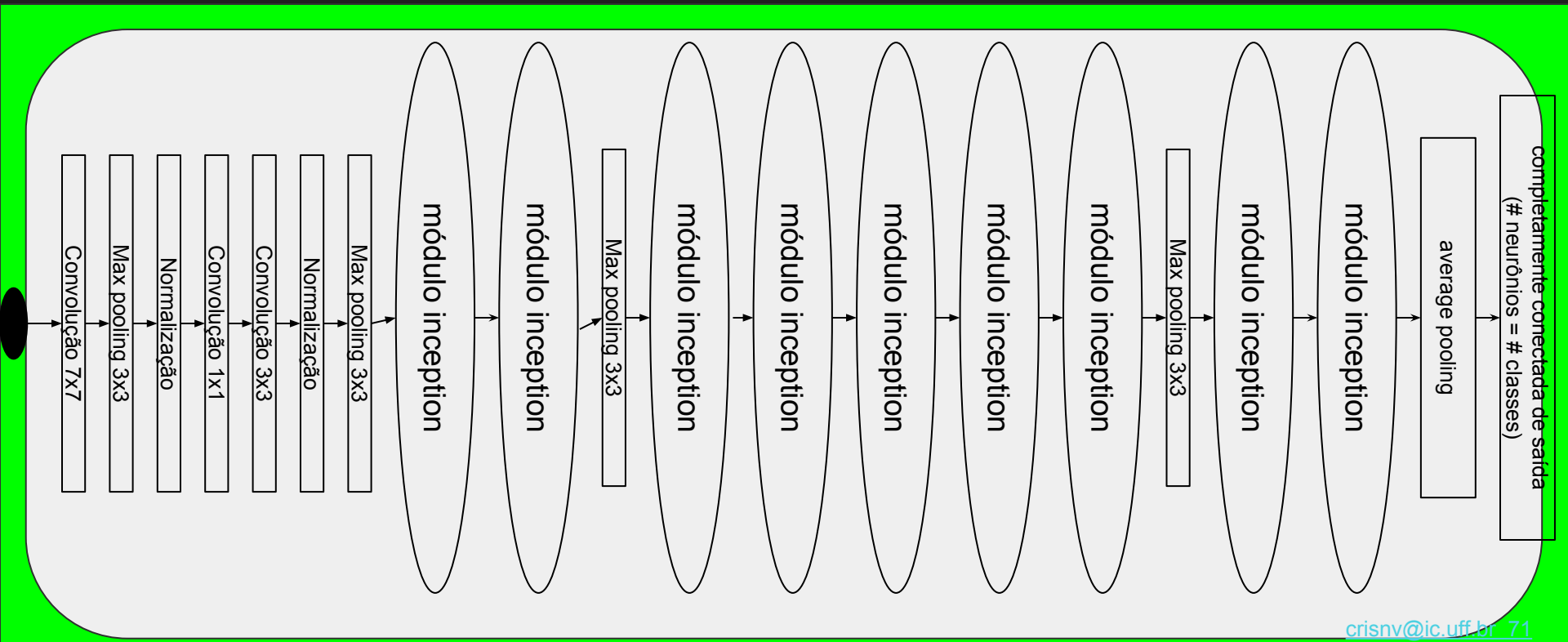
VGG [16~19 camadas] (2014)



Inception (2014)

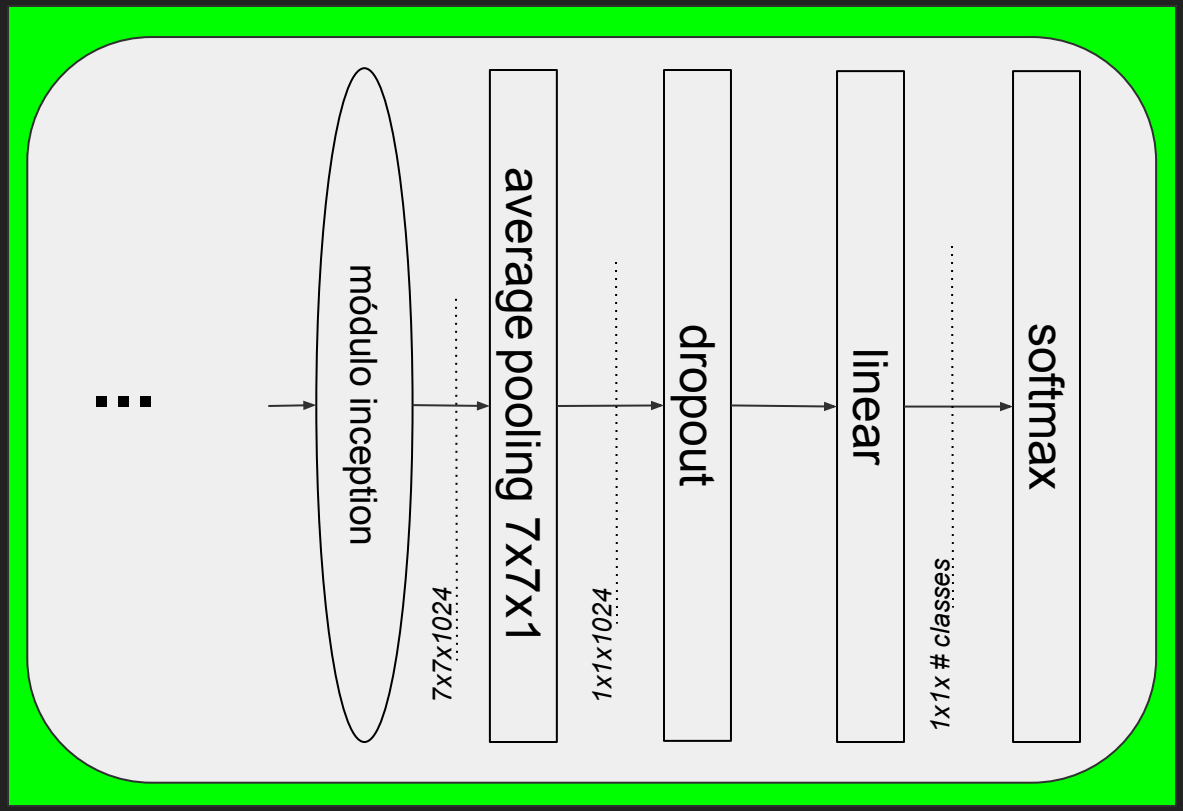


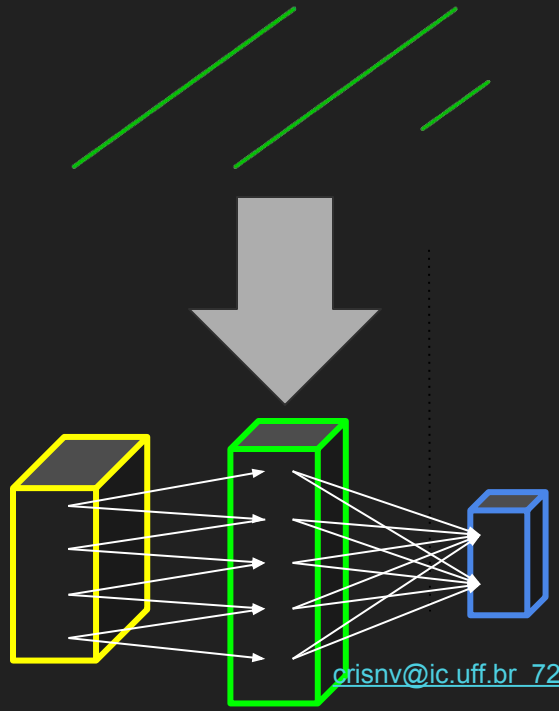
GoogLeNet [22/27 camadas] (2014)



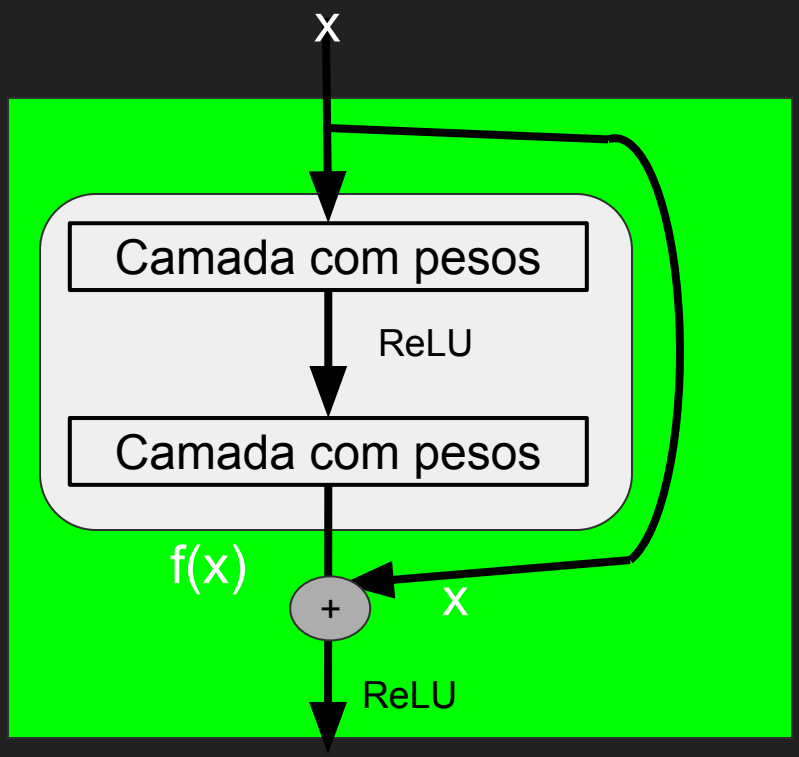
"Eliminação" da camada completamente conectada

Lin et al. Network in Network. 2013





ResNet (2015)



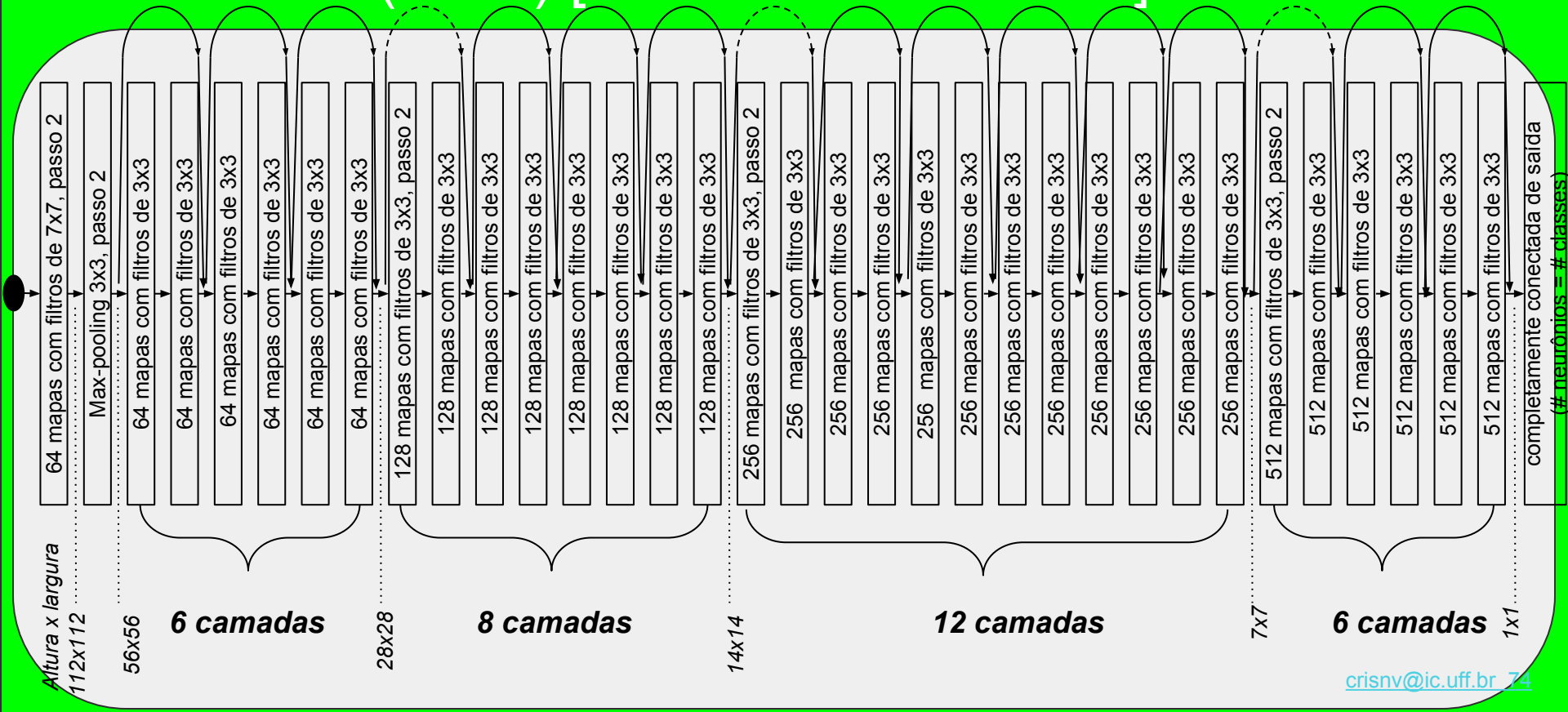
Shortcut connections :

- pulam um ou mais camadas;
- usadas na ResNet para mapear identidade;
- não introduzem novos parâmetros;

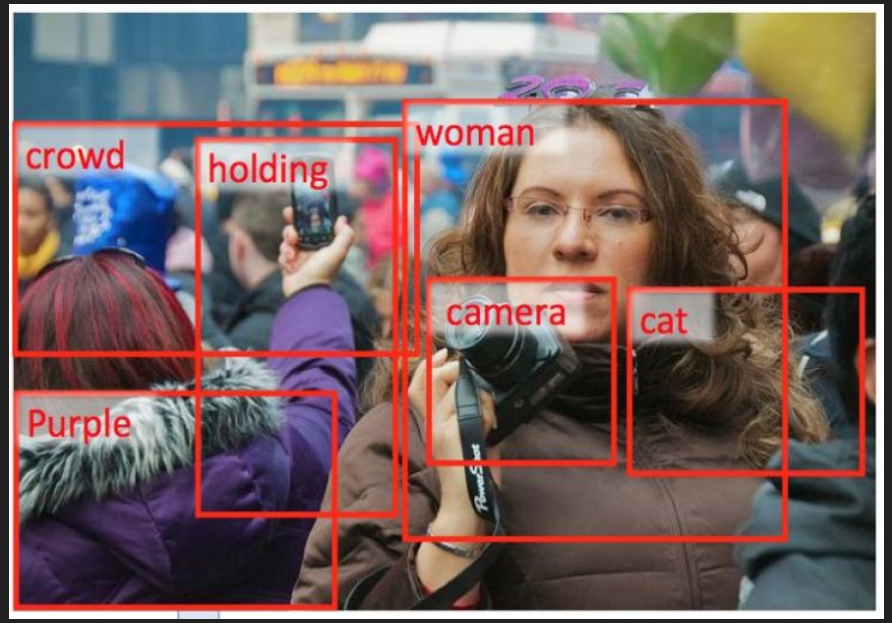
Profundidade da ResNet:

- Shortcut connections;
- Normalização (inicialização e de camadas intermediárias);

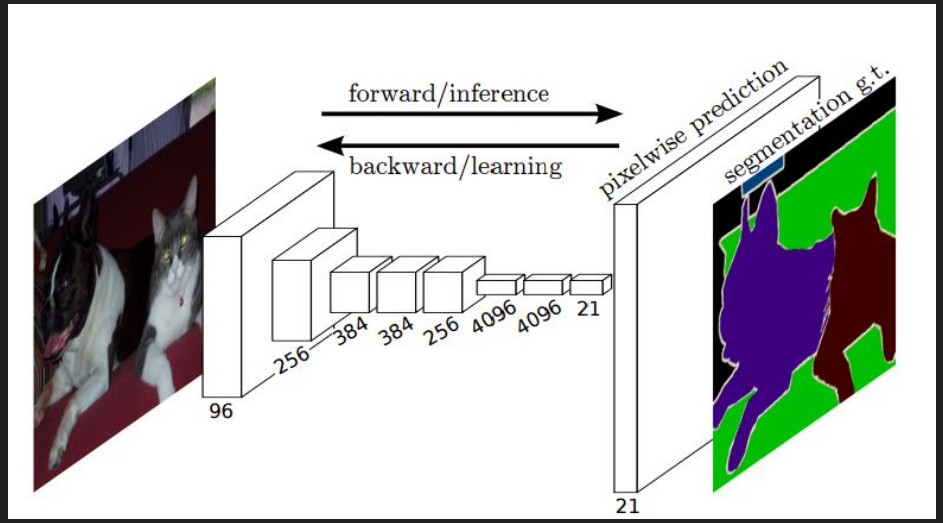
ResNet-34 (2015) [muitas + camadas...]



Outras aplicações de CNN em imagens

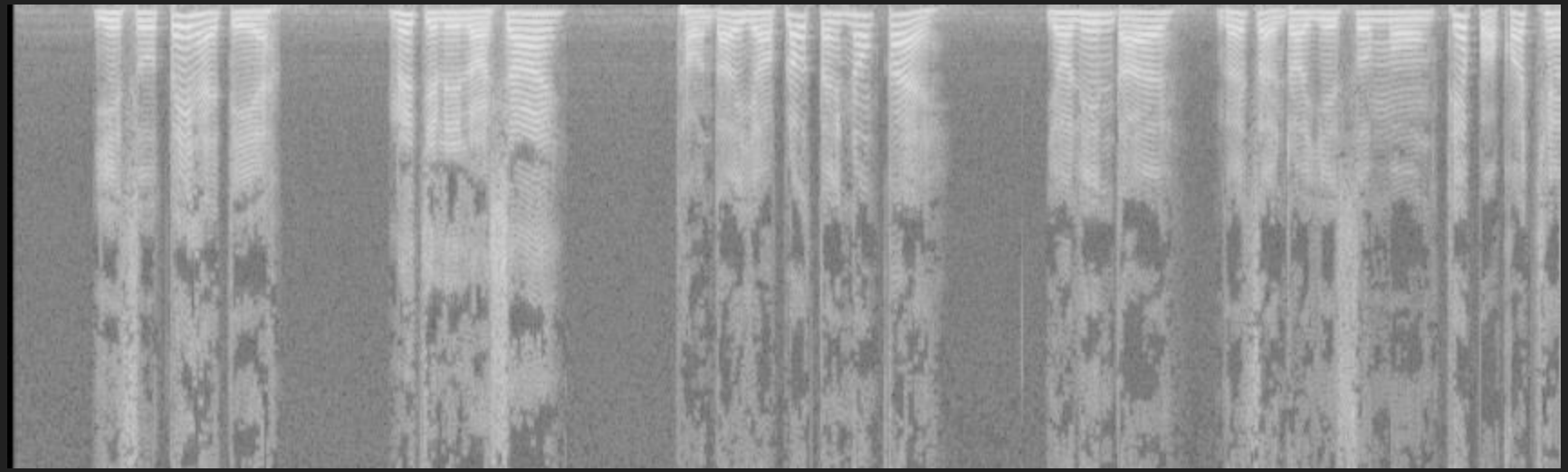


Fang et al. From Captions to Visual Concepts and Back. 2015



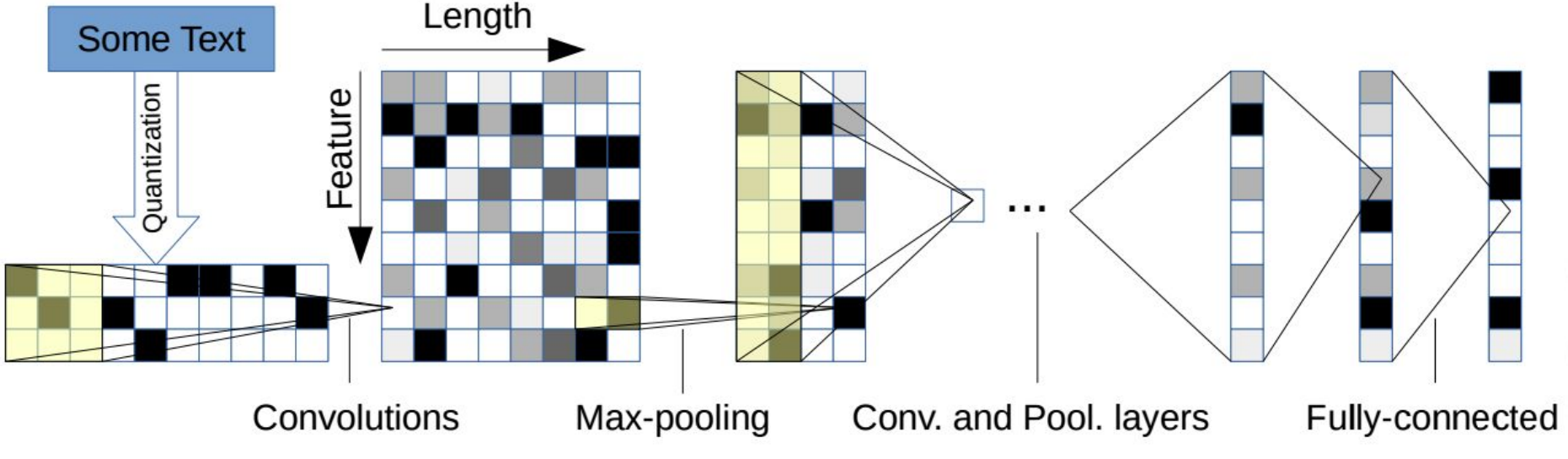
J. Long, E. Shelhamer, T. Darrell. Fully Convolutional Networks for Semantic Segmentation. 2015

Som como imagem



M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani. Exploiting spectro-temporal locality in deep learning based acoustic event detection, 2015

Texto como imagem



Xiang Zhang, Junbo Zhao, Yann LeCun. Character-level Convolutional Networks for Text Classification. 2016

Instituto de Computação

Conclusões

CNNs recentes

- adotam tamanhos de filtros pequenos;
- eliminam/reduzem camadas completamente conectadas;
- adotam tamanhos de janelas de pooling pequenos;
- tamanho de passo 1 (stride)
- muito profundas (MSRA 2015 - experimentos com 150 e 1200 camadas);

Fatores que impulsionaram o DL:

- desenvolvimento de algoritmos de aprendizado eficientes;
- bancos padrões de referência (benchmarks)
- aumento da capacidade de processamento e poder de processamento de fins gerais utilizando hardware gráfico (GPU)

Vantagens de abordagens com aprendizado profundo

Robustez:

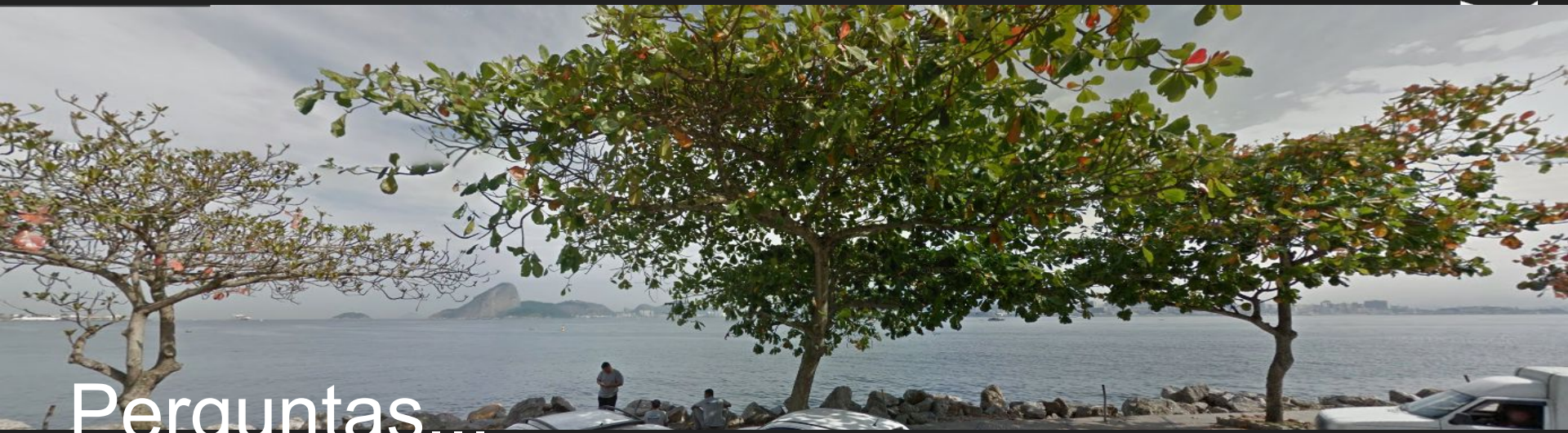
- uma hierarquia de features é aprendida de maneira ótima para a tarefa proposta
- mostra-se capaz de aprender as complexas variações existentes nos dados

Generalizável:

- a mesma arquitetura pode ser usada em diferentes tarefas e tipos de dados

Escalável:

- mais dados levam ao aumento de performance
- método massivamente paralelizável



Perguntas...

crisnv@ic.uff.br
crisnv@ic.uff.br_82