



**Universidade Federal do Piauí**  
Laboratório de Inteligência Artificial - LINA

# Introdução à Deep Learning

**Bruno Vicente Alves de Lima**

# Deep Learning



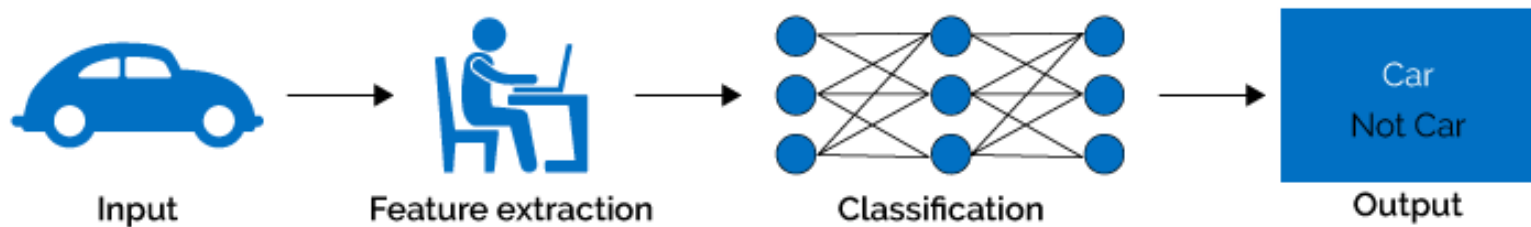
# Deep Learning

Deep Learning resolve problemas introduzindo representações a partir de outras representações mais simples." (Goodfellow et al.,2016);

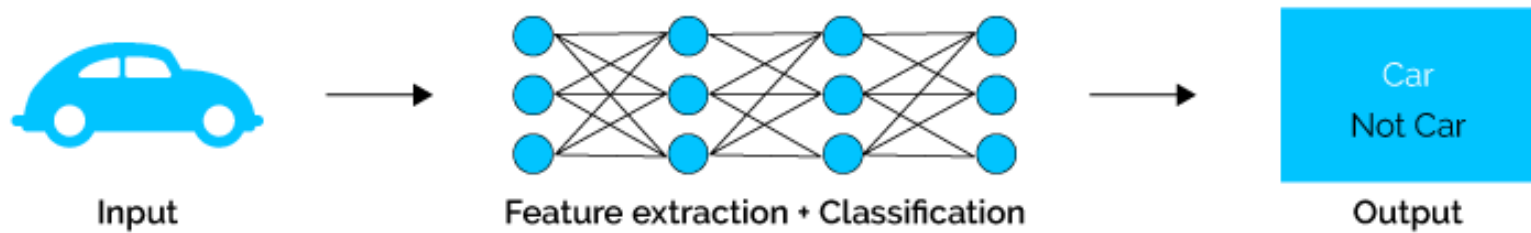
Possuem capacidade de extrair características mais relevantes no problema, reduzindo a necessidade de interferência humana.

# Deep Learning

## Machine Learning

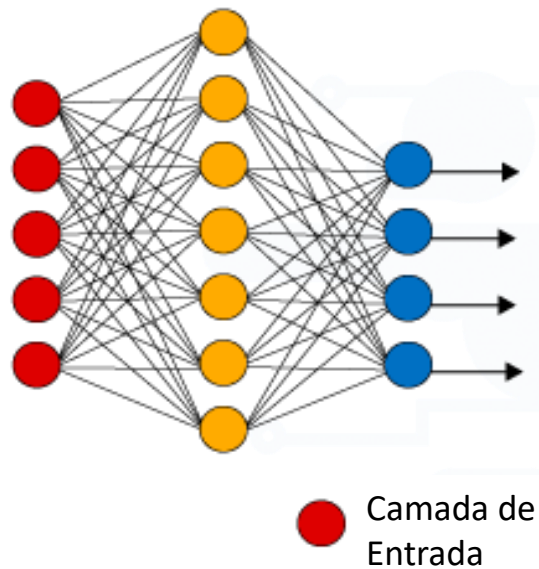


## Deep Learning

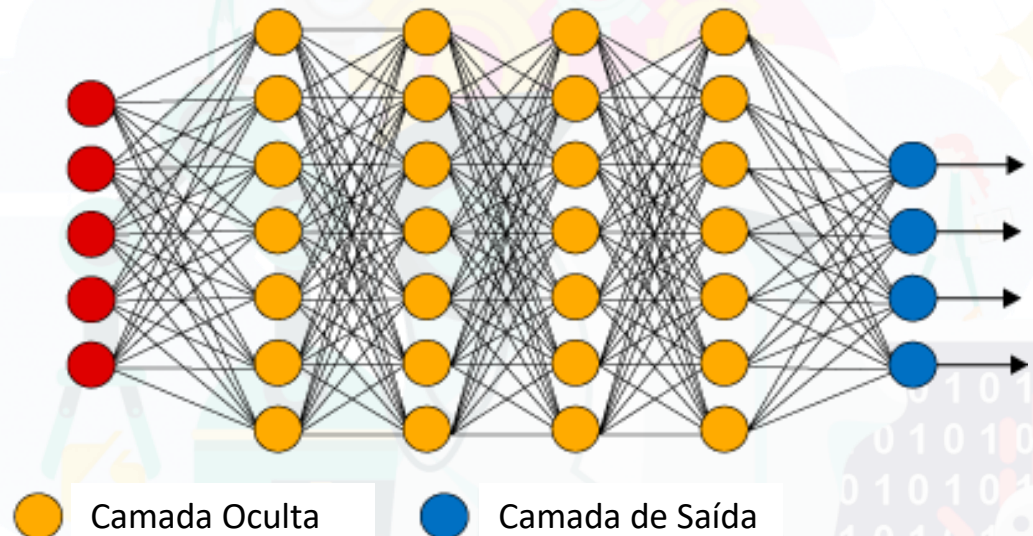


# Deep Learning

Rede Neural Rasa

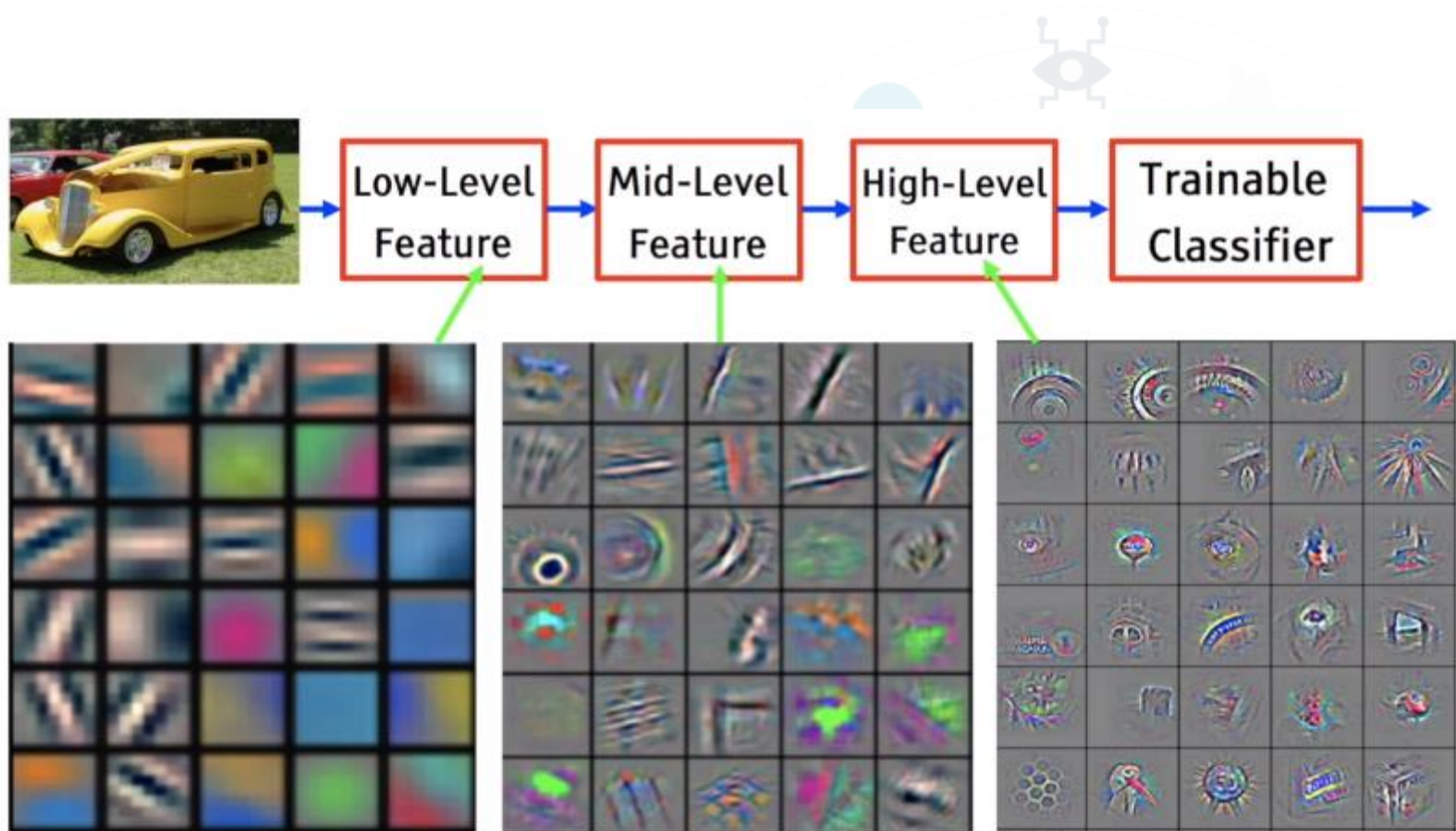


Rede Neural Profunda





# Deep Learning



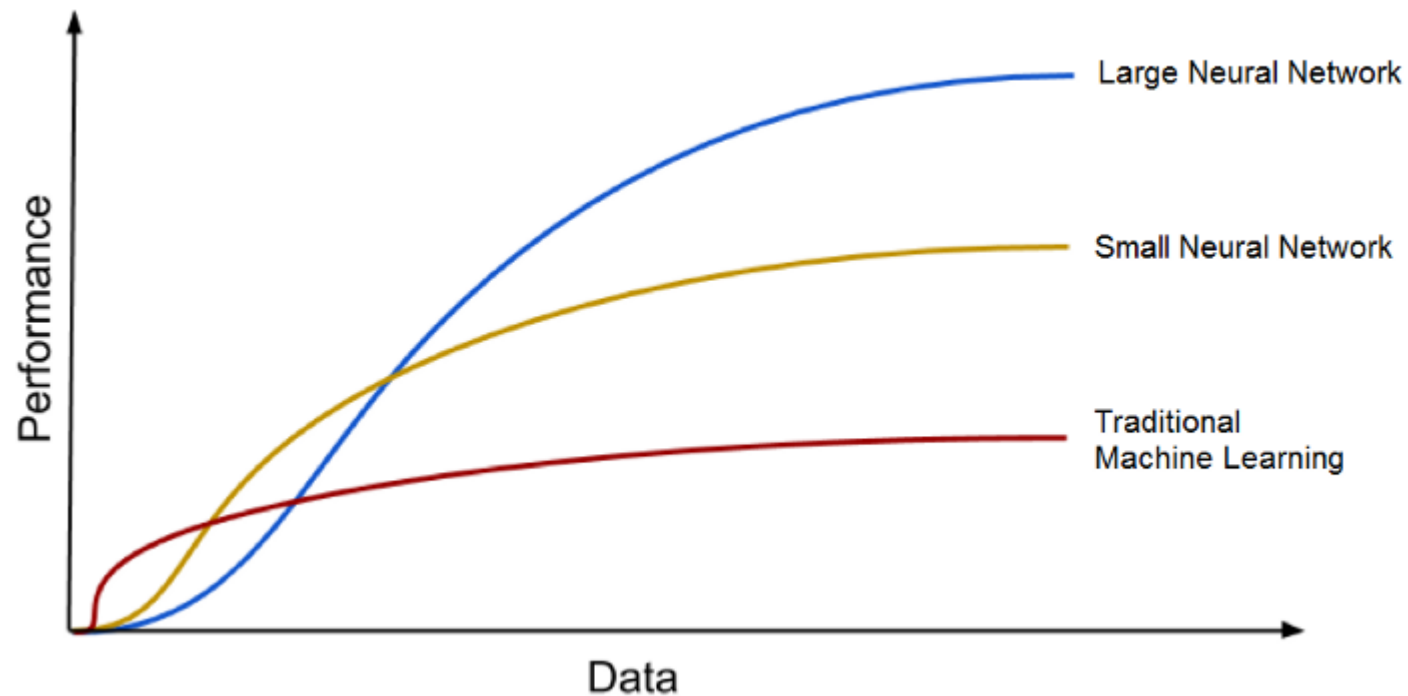
# Deep Learning

## 🌐 Conceitos

- Grande quantidade de dados;
  - Big Data
- Problema de Dimensionalidade;
- Função de Ativação;
  - Utilização da Função Relu
  - Softmax na saída
- Regularização;
  - **Dropout**
  - L1
  - L2

# Deep Learning

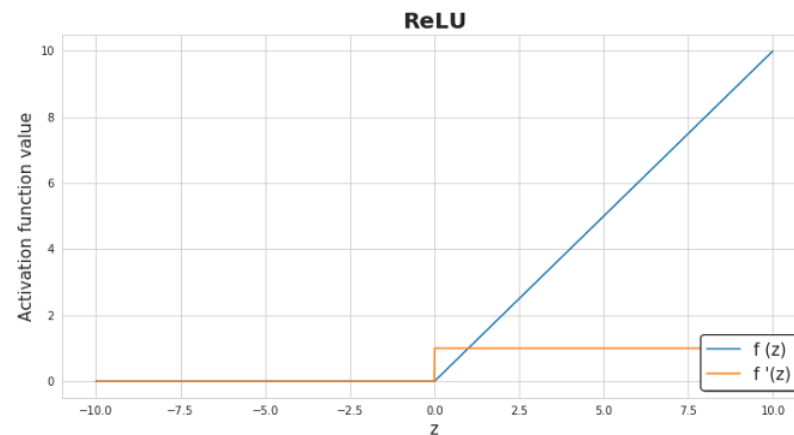
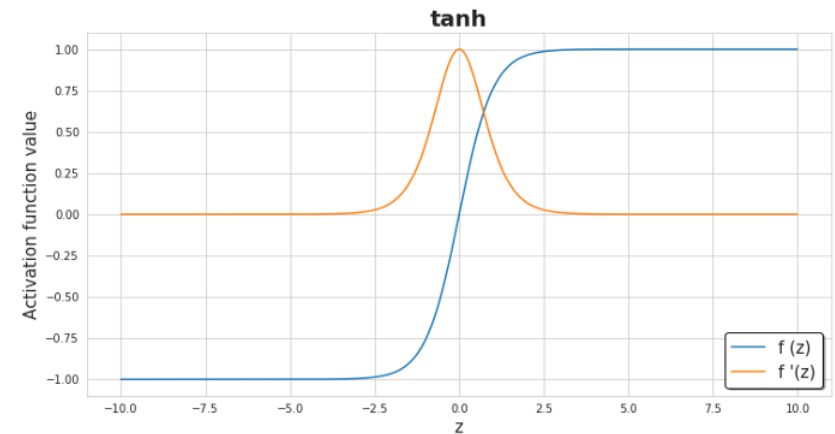
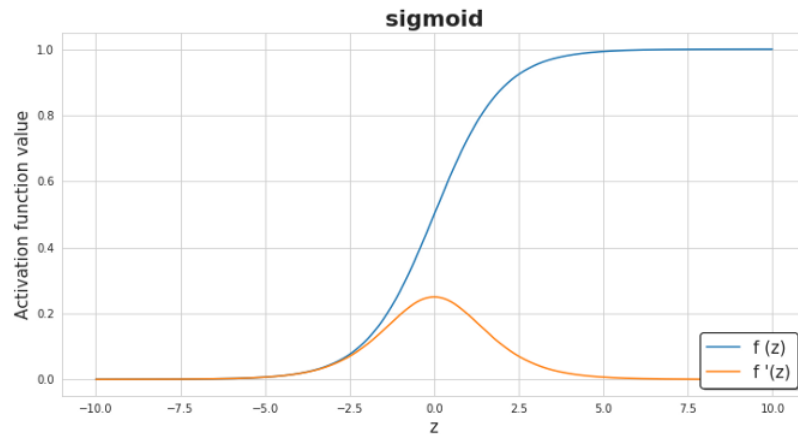
## ● Influência da Quantidade de dados





# Deep Learning

## 🎓 Morte do Gradiente (Vanishing Gradiente)

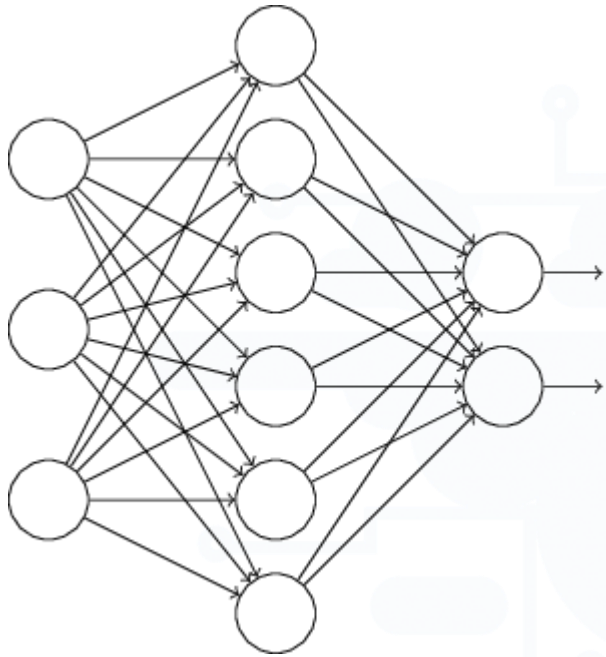


# Regularização Dropout

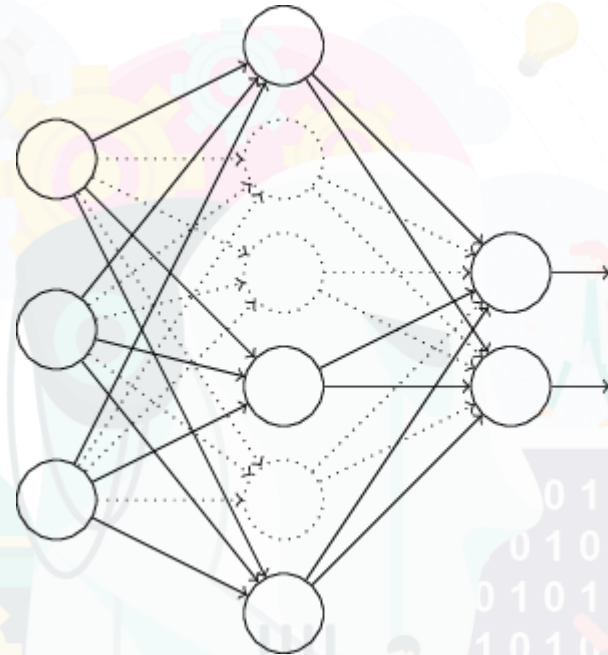
- Elimina aleatoriamente (e temporariamente) alguns dos neurônios ocultos na rede, deixando os neurônios de entrada e saída intocados;
- Reduz o overfitting causado pela profundidade da rede;
- Proposto em:

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). **Improving neural networks by preventing co-adaptation of feature detectors**. arXiv preprint arXiv:1207.0580.

# Regularização Dropout



Sem Dropout



Com Dropout

# Extração de Características “Tradicional”



Vermelho da Camisa – X1

Azul do Calção – X2

Azul do Sapado – X3

Base de Dados de Treinamento

X1	X2	X3	X4	X5	X6	Y
8,97	3,45	2,35	0,00	0,00	0,00	Bart
6,75	0,94	0,52	0,00	0,00	0,00	Bart
9,69	4,10	1,56	0,00	0,00	0,00	Bart
0,00	0,00	0,00	4,68	0,66	0,01	Home
0,00	0,00	0,00	0,12	2,50	0,03	Home
0,00	0,00	0,00	5,80	0,50	1,28	Home

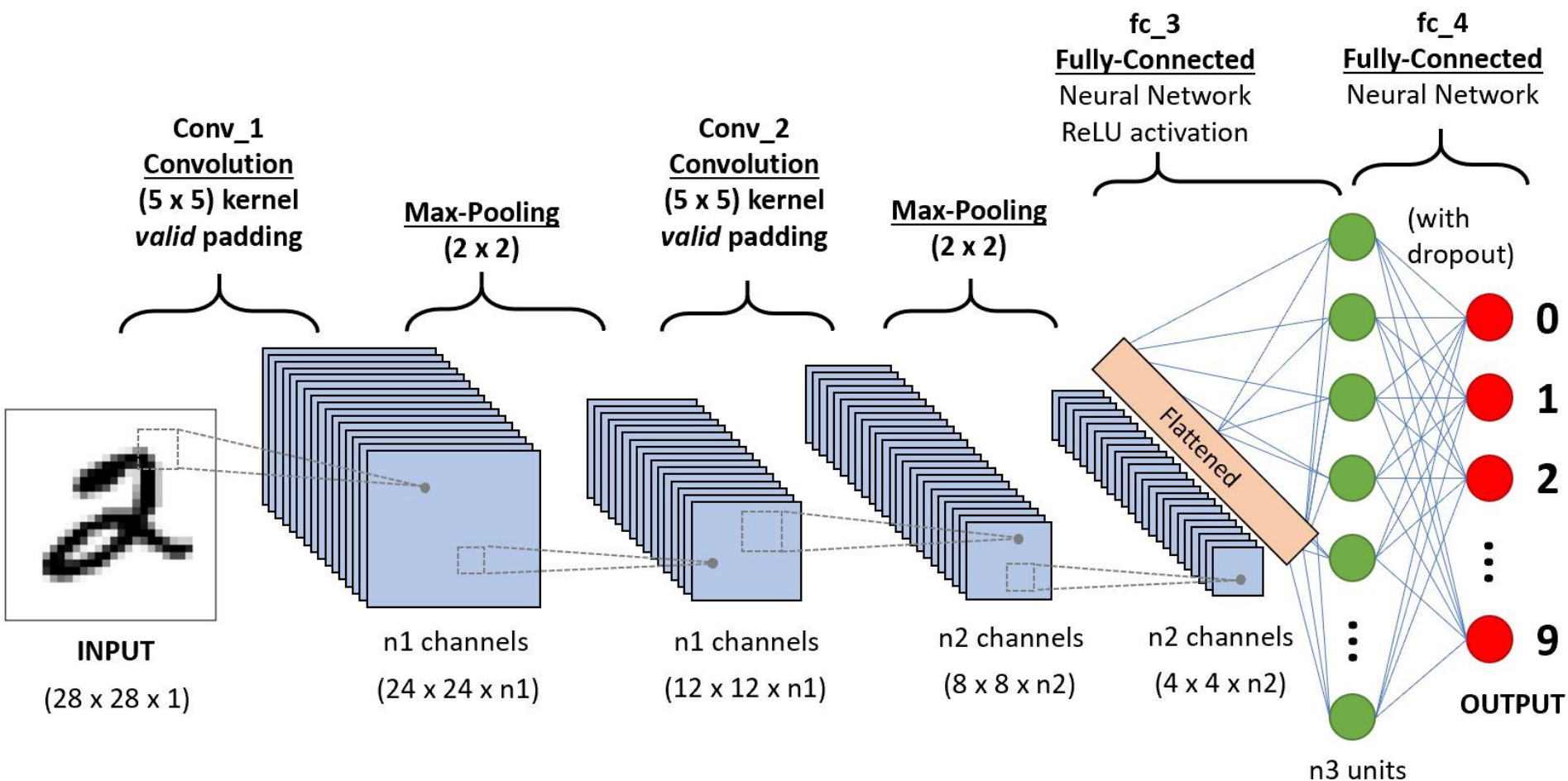


Marrom da boca – X4

Azul da Calça – X5

Cinza do sapado – X6

# Redes Neurais Convolucionais (CNN)





# Redes Neurais Convolucionais (Aplicações)

## 🌐 Descrição de Cenários



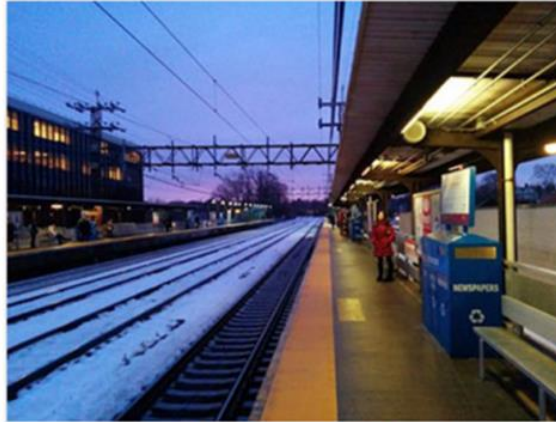
noite

ponte

cidade

ponte suspensa

rio



trem

metrô

ferrovia

via férrea

estação

transporte



competição

trem

tênis

estádio

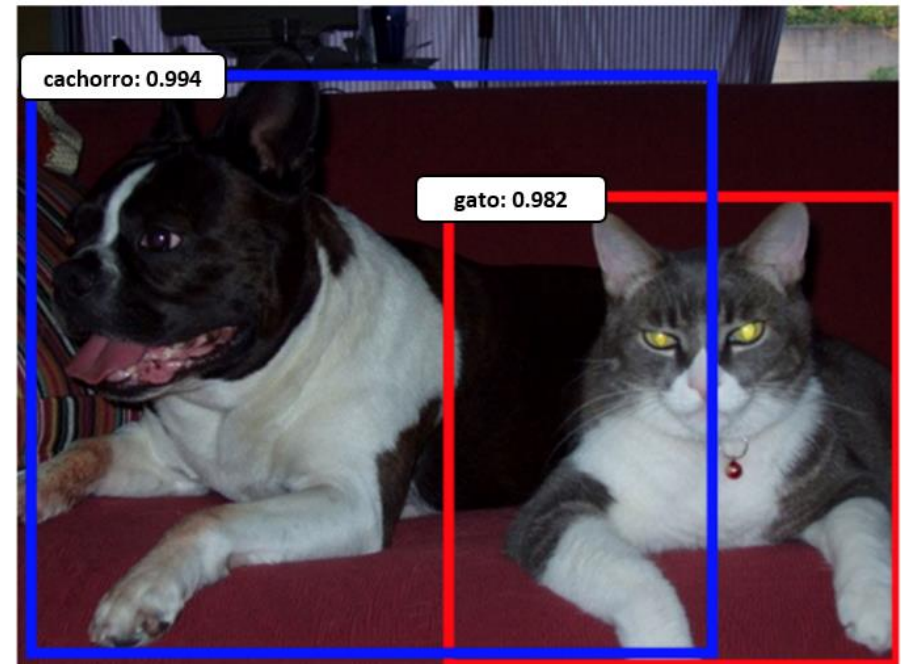
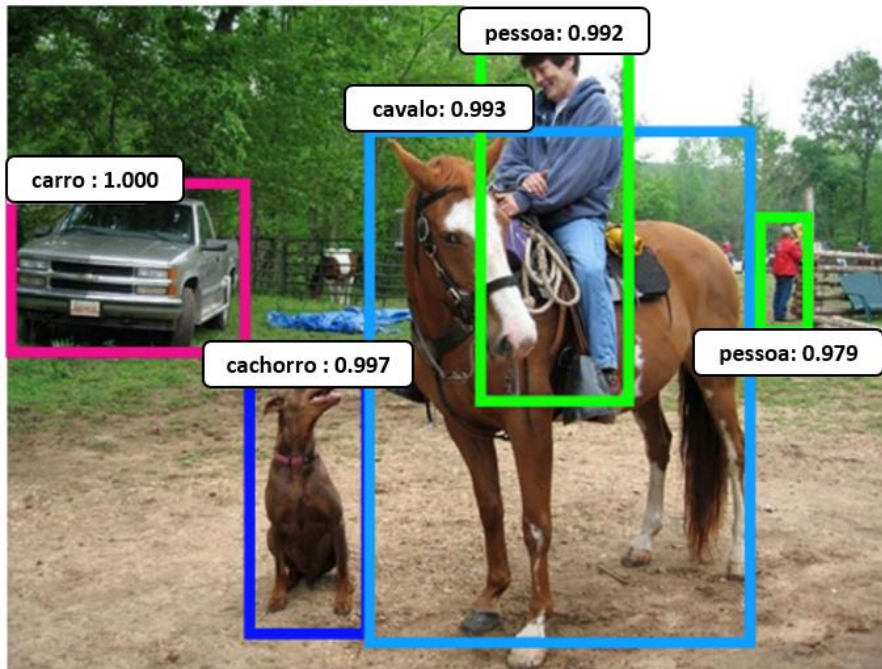
bola

multidão

espectadores

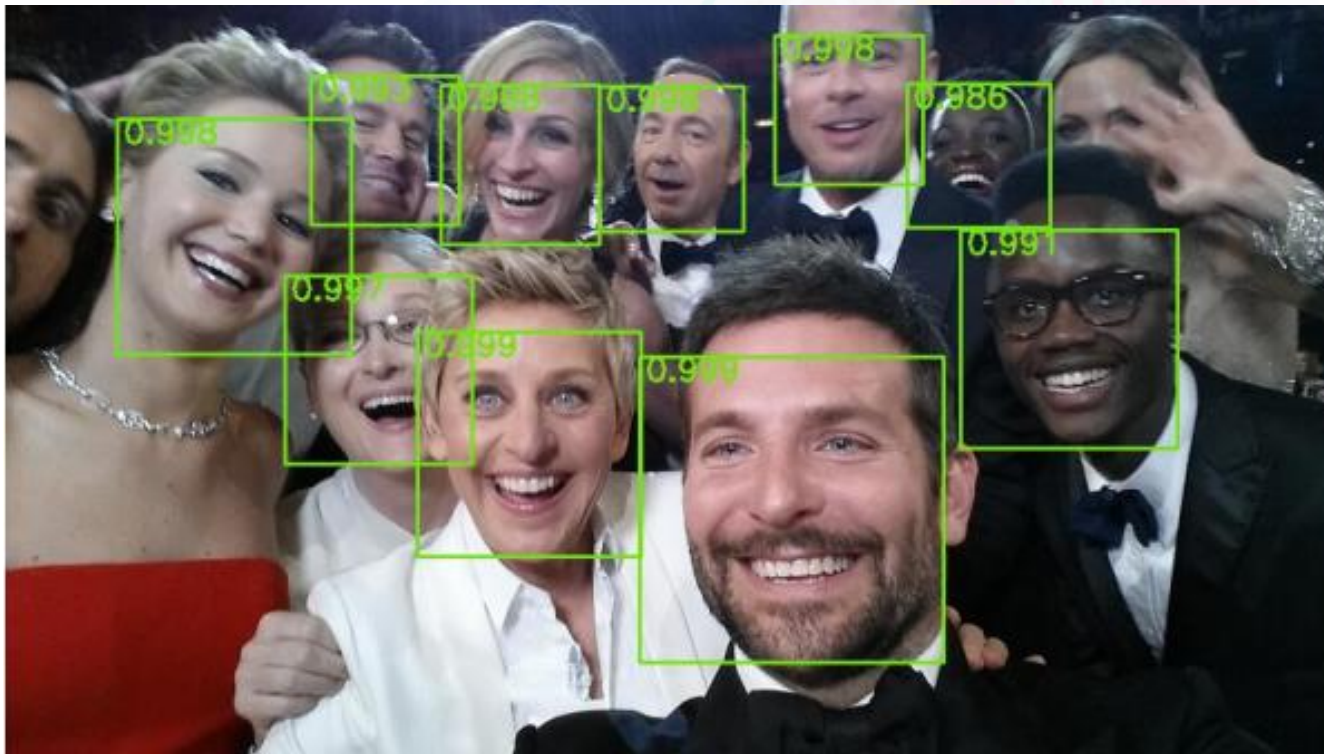
# Redes Neurais Convolucionais (Aplicações)

## 🌐 Detecção de Objetos



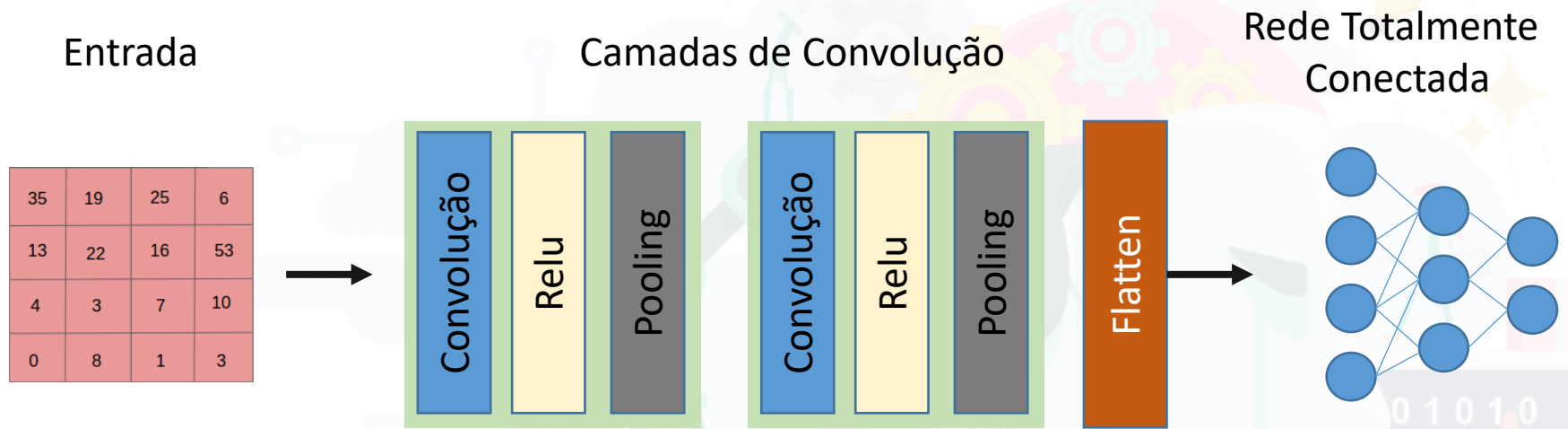
# Redes Neurais Convolucionais (Aplicações)

- Detecção e Reconhecimento Facial





# Camada de Convolução

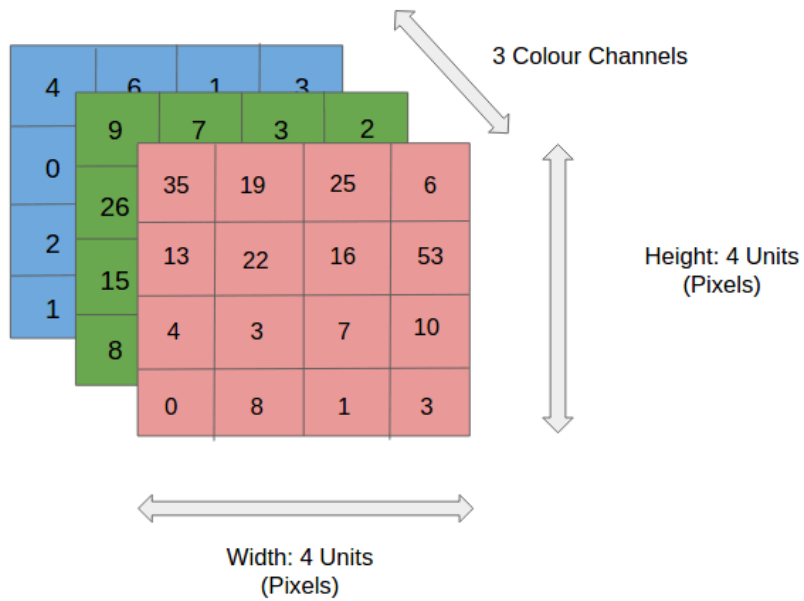


# Entrada

- Para CNN a entrada é uma imagem;
- Uma imagem é uma matriz usualmente tridimensional;
- Pode ser representada no RGB;
- Pode ser preto e branco;



# Entrada



Colorida

Diagram illustrating a 2D input tensor for a grayscale image. The tensor is represented as a 4x4 grid of gray blocks containing numerical values.

35	19	25	6
13	22	16	53
4	3	7	10
0	8	1	3

Preto e Branco

# Etapa de Convolução

- Operação linear, que a partir de duas funções gera uma terceira;
- Utiliza um filtro ou kernel que realiza operação de multiplicação de matrizes em pedaços da imagem;
- Em aplicação do kernel, a região é alterada de acordo com um parâmetro chamado ***stride***;
- O tamanho do kernel é predefinido antes do treino;
- Reduz a dimensionalidade da entrada.

# Etapa de Convolução

🌐 Dado a Imagem de Entrada e o Filtro

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

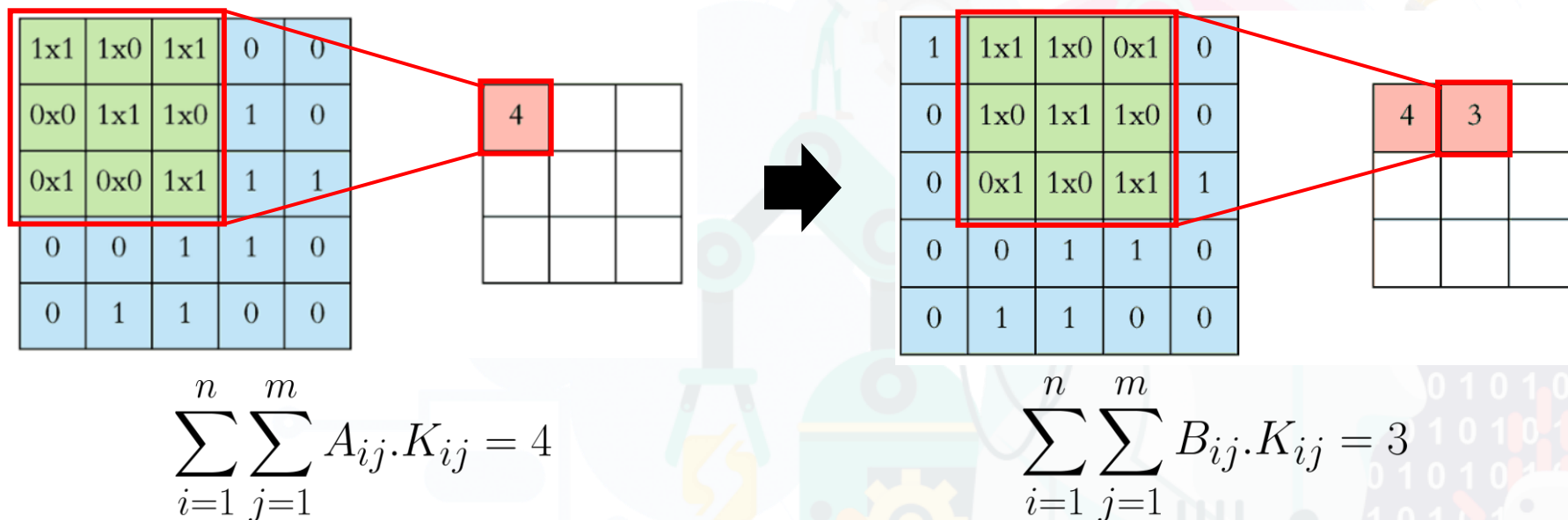
Entrada

1	0	1
0	1	0
1	0	1

Kernel

# Etapa de Convolução

- O Kernel é aplicado à imagem, considerando stride = 1 para esse exemplo;
- Sucessivas multiplicações entre matrizes;



- Sendo, **A** e **B** são as matrizes (pedaços da imagem) e **K** o kernel e **n** e **m** o tamanho do Kernel.

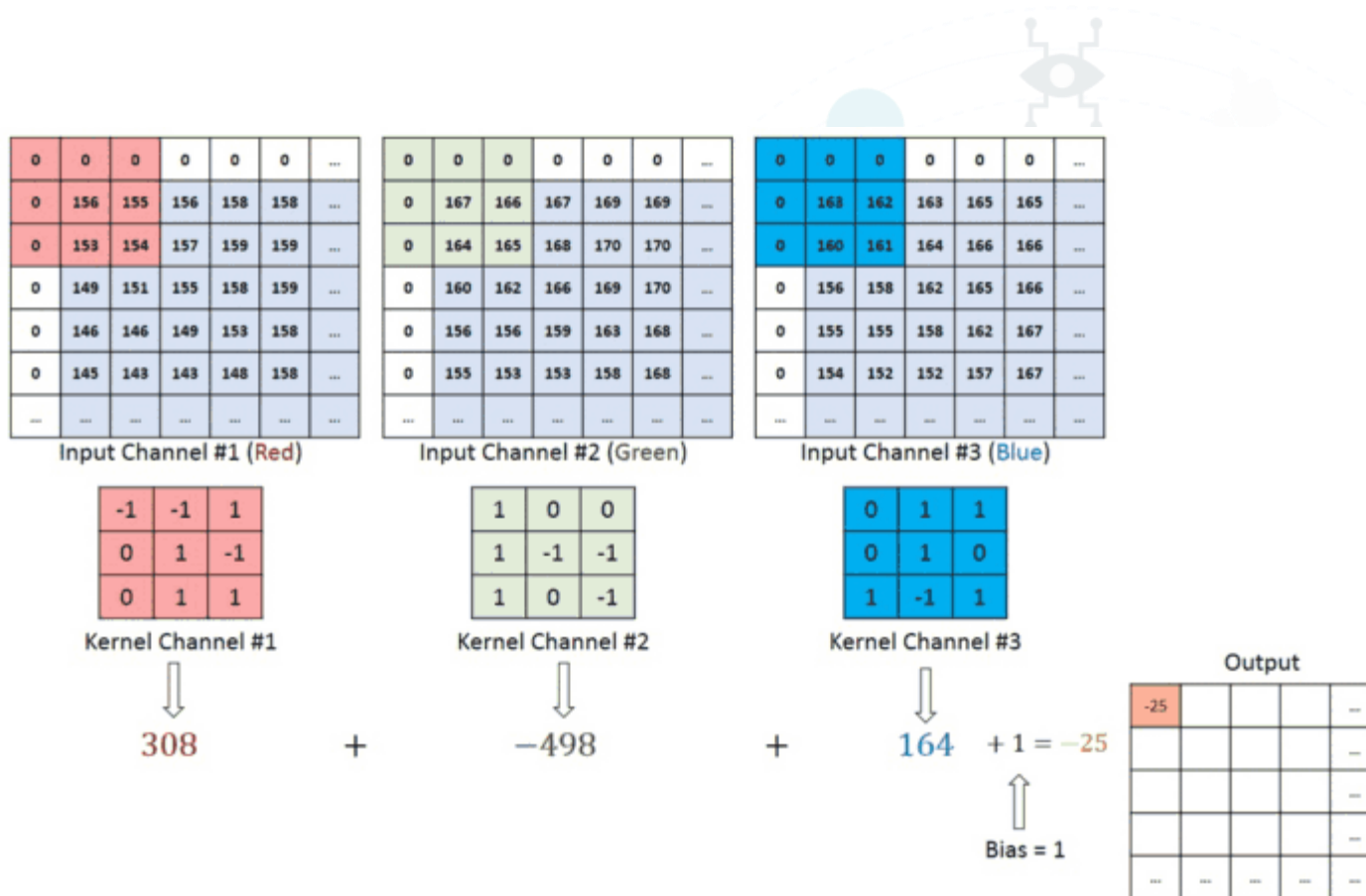
# Etapa de Convolução

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

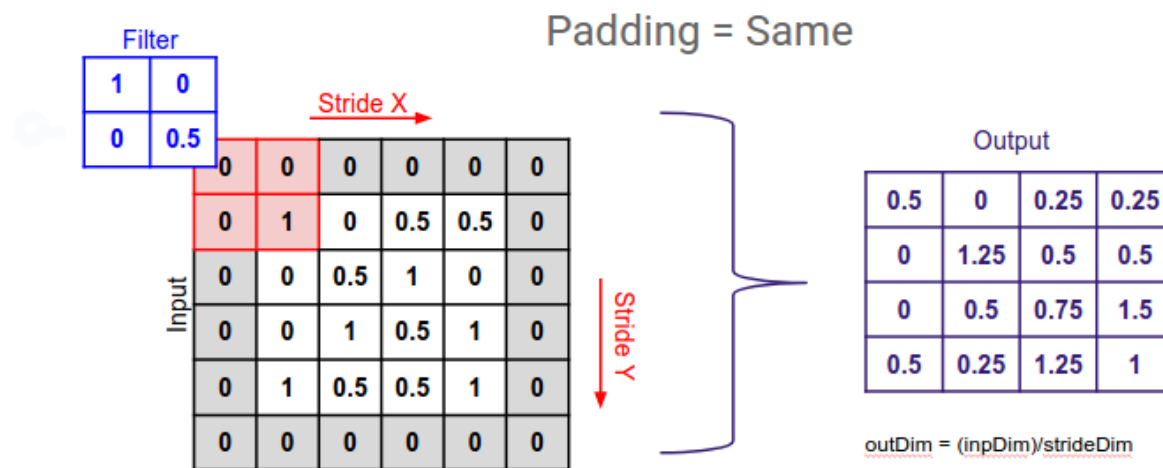


# Etapa de Convolução



# Padding

- Processo de adicionar alguns pixels (com valor Zero) ao redor da imagem antes da convolução;
- Finalidade de manter a dimensionalidade da imagem resultante durante a operação;
- Esse processo é utilizado porque essas imagens resultantes podem conter elementos que facilitam a identificação da classe alvo para a rede.








# Etapa de Convolução

Tamanho da saída:

$$O = \frac{(W - K + 2P)}{S} + 1,$$

$W$ =tamanho (altura) da entrada,  $K$ =tamanho do filtro,  $P$ =padding,  
 $S$  = stride.

# Etapa de Convolução

Operation	Kernel $\omega$	Image result $g(x,y)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

# Etapa de Relu (Ativação)

- Após a convolução, aplica-se a função de ativação Relu na imagem;

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

- Com a convolução pode-se criar uma saída linear, a função de ativação adiciona a não-linearidade;
- Como utiliza-se função Relu, os valores dos pixels ficam todos maiores ou igual a zero;



# Etapa de Pooling

- Serve para simplificar a informação da camada anterior, reduzindo a dimensionalidade;
- Resolve problemas de invariância na imagem;

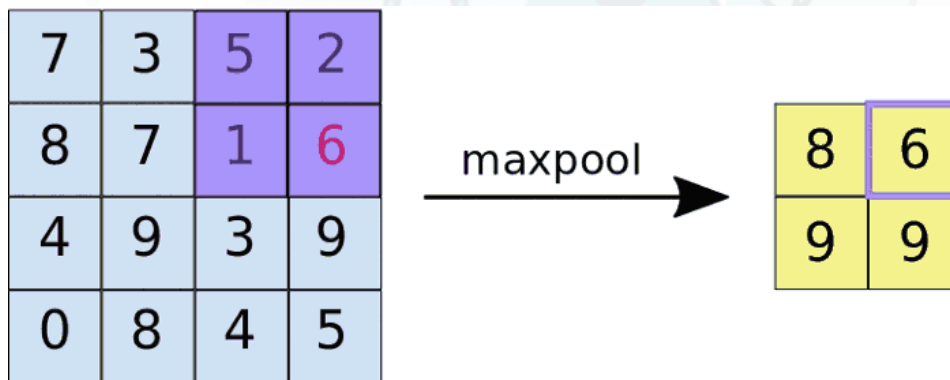
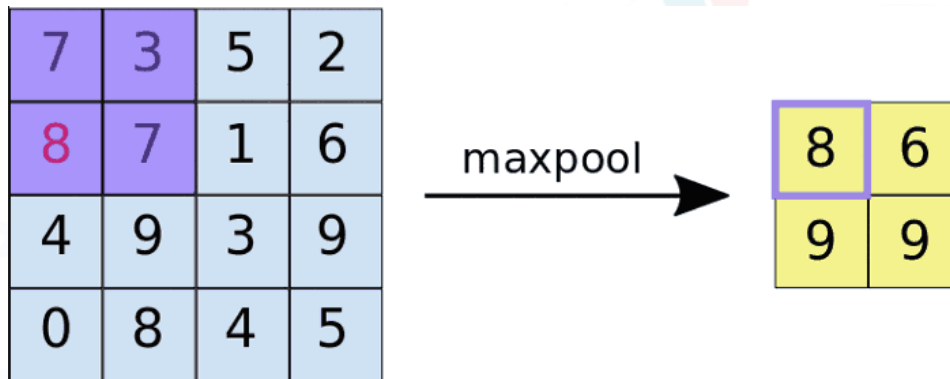


# Etapa Pooling

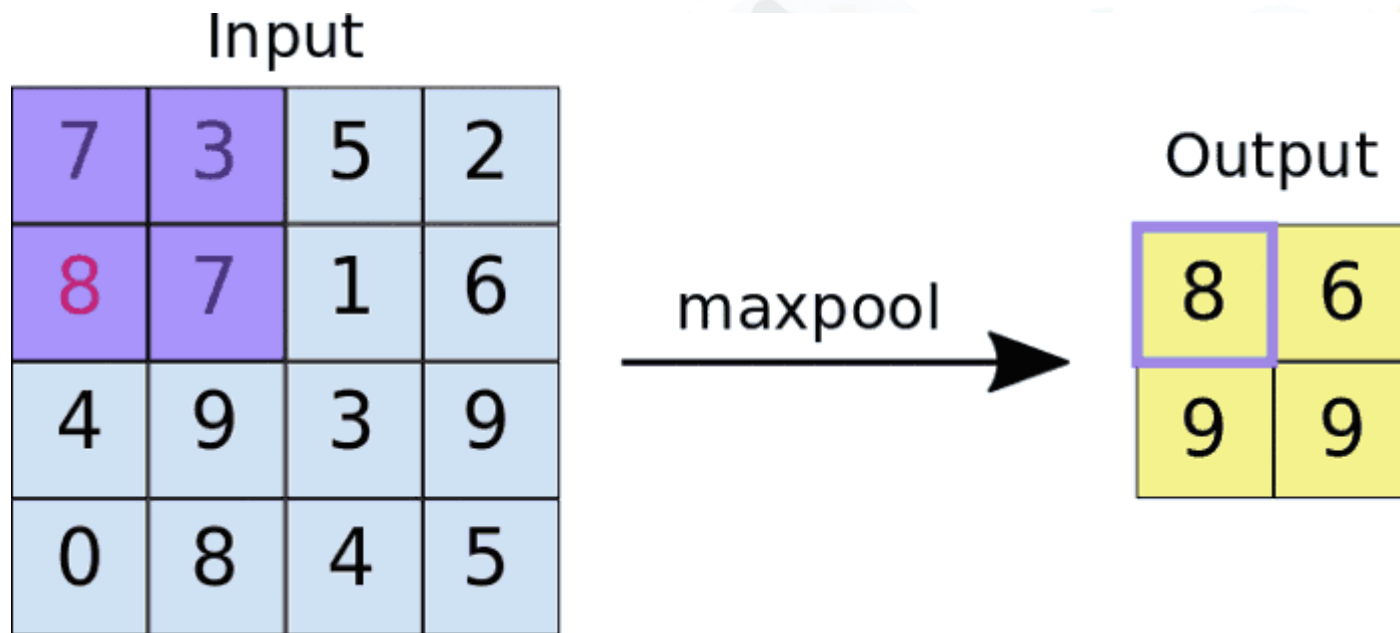
- Uma unidade de área é escolhida para transitar por toda a saída após a ativação;
- Se a entrada foi de  $24 \times 24$  a saída será  $12 \times 12$ , utilizando uma área de  $2 \times 2$ ;
- Utiliza-se um método para realizar a sumarização no pooling, o mais utilizado é MaxPooling;
- Outros Possíveis: Sumpooling, AveragePooling;

# Etapa Pooling

● Utilizando o maxpooling no exemplo:

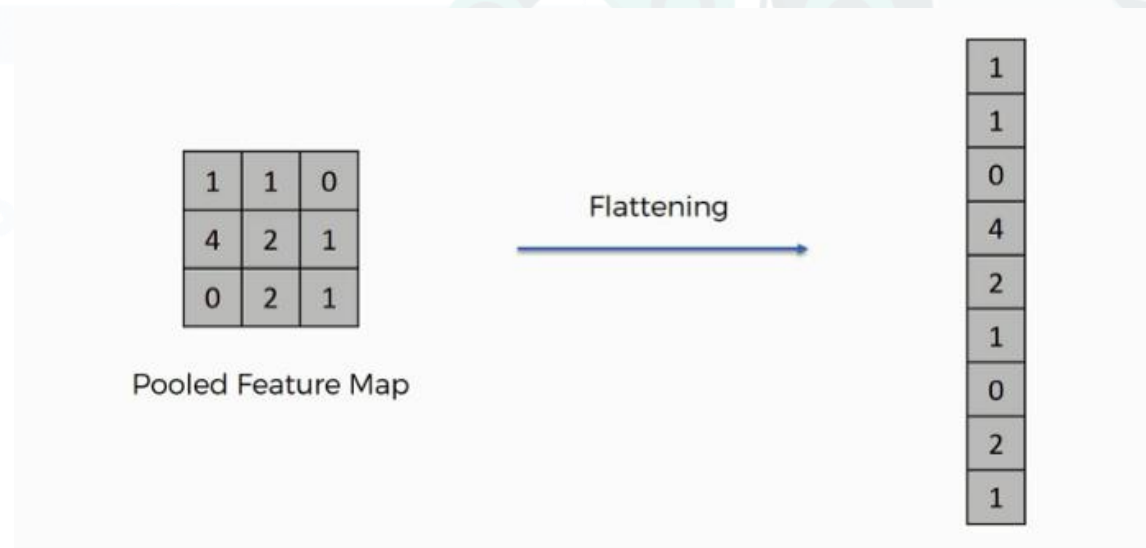


# Etapa Pooling



# Flatten

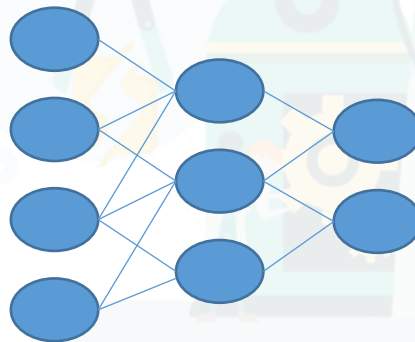
- Utilizada na divisão das duas partes da CNN;
- Transforma a matriz da imagem para um array de 1 dimensão;



# Fully Connected

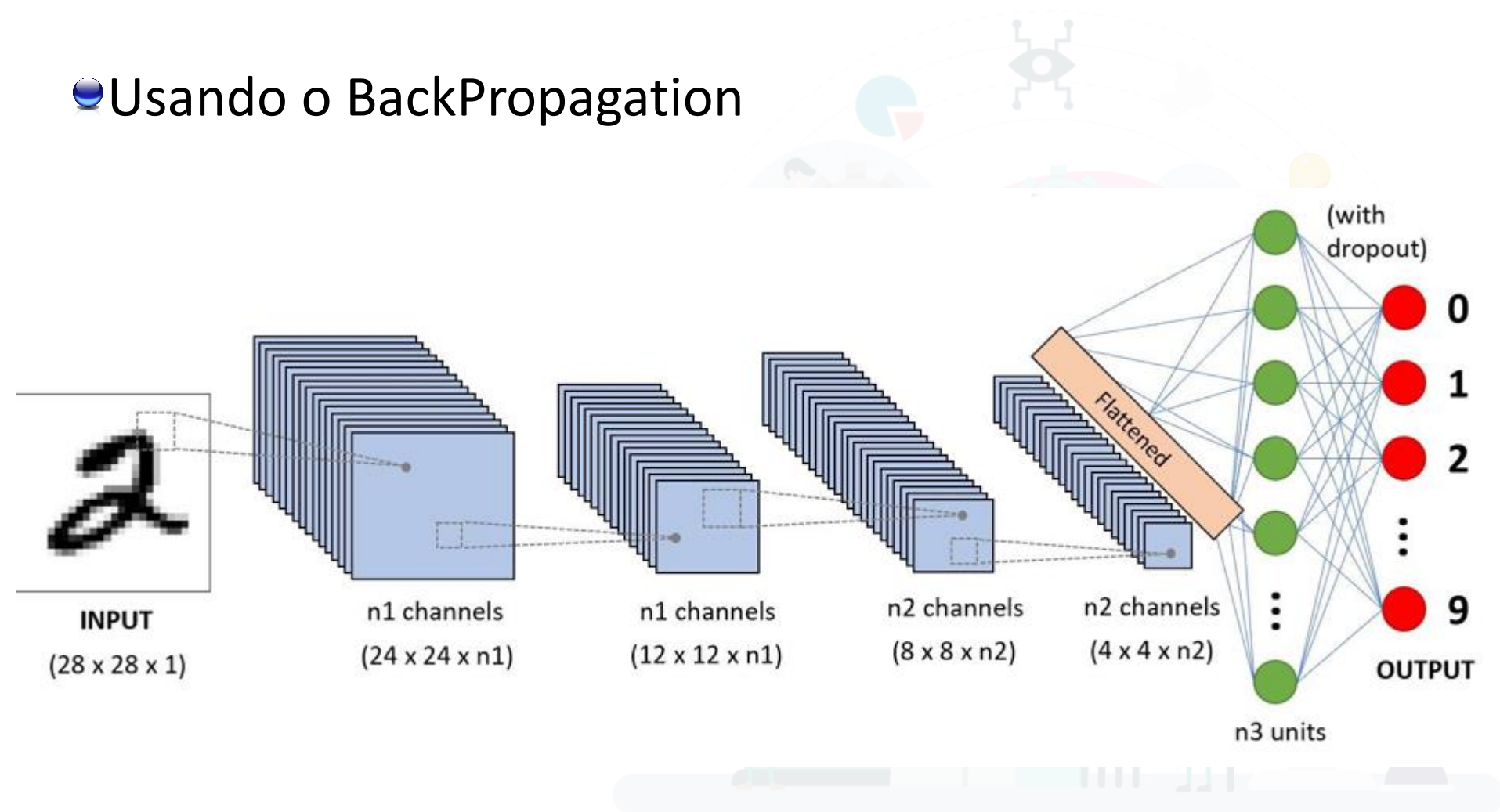
- Ao final é colocada uma camada totalmente conectada;
- Pode ser uma MLP rasa ou profunda;
- Essa rede recebe como entrada o resultado do Flatten;

1
1
0
4
2
1
0
2
1



y

## Usando o BackPropagation





# Outras Arquiteturas

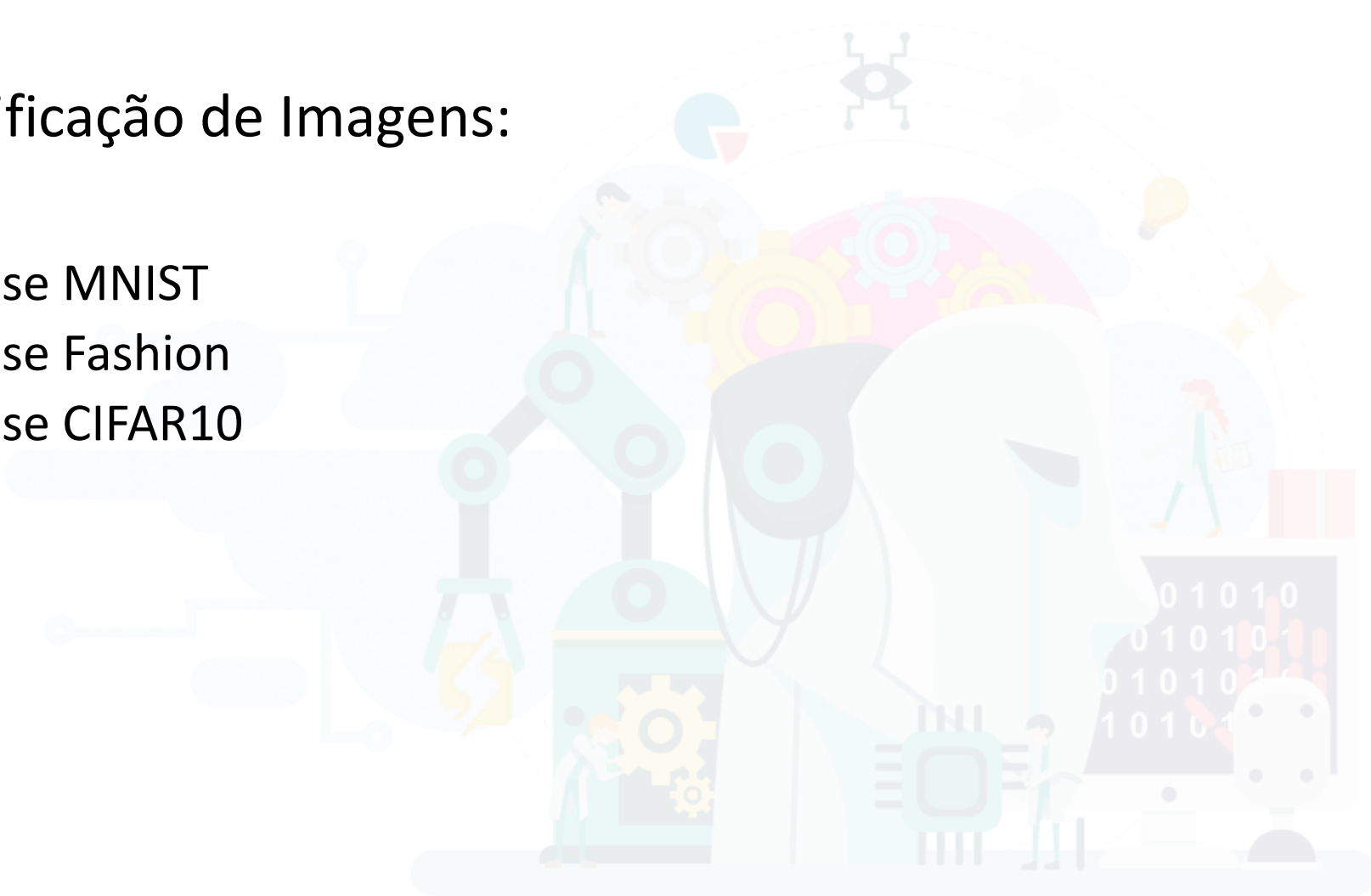
- 🌐 O ILSVRC 2012 ficou marcado como o primeiro ano no qual uma CNN atingiu o primeiro lugar desse desafio.

Ano	Descrição	Erro
2010	SIFT + LBP + Fisher Vector + PCA + SVM	28.2
2011	Otimização do método de 2010	25.8
<b>2012</b>	<b>AlexNet</b>	<b>16.4</b>
2013	Zf Net	11.7
2014	GoogLeNet	6.7
2015	ResNet	3.6
2016	DenseNet	3.0
2017	SENet	2.3

# Exemplo Prático

## 🌐 Classificação de Imagens:

- Base MNIST
- Base Fashion
- Base CIFAR10





**Universidade Federal do Piauí**  
Laboratório de Inteligência Artificial - LINA

# Introdução à Deep Learning

**Bruno Vicente Alves de Lima**