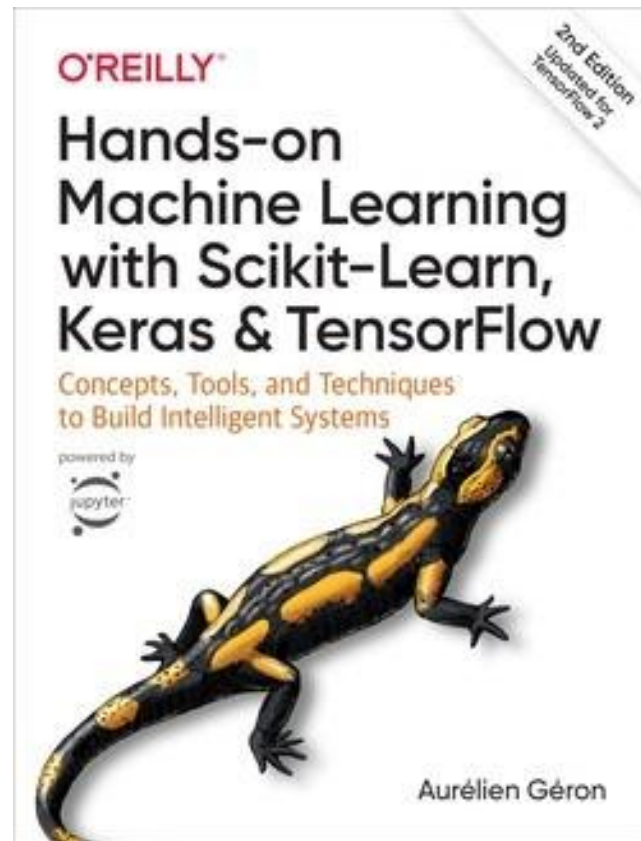
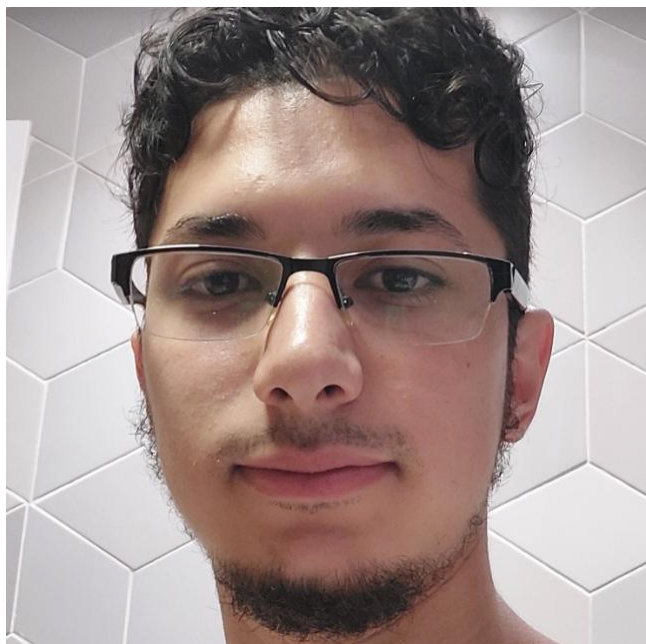



Redes Neuras em Visão Computacional

Apresentações

- Luiz Henrique B. Lago



- Cap 14 - Deep Computer Vision using Convolutional Neural Networks



Visão Computacional

- Analisar, interpretar e extrair informações de imagens;
- Características de animais:
 - Reconhecer faces familiares;
 - Contar objetos;
 - Identificar cenários;

Visão Computacional

- Informações facilmente reconhecidas:
 - São 3 Cachorros;
 - Estão num campo;
 - O do meio carrega algo;
- Como abstrair isso para que uma máquina entenda?

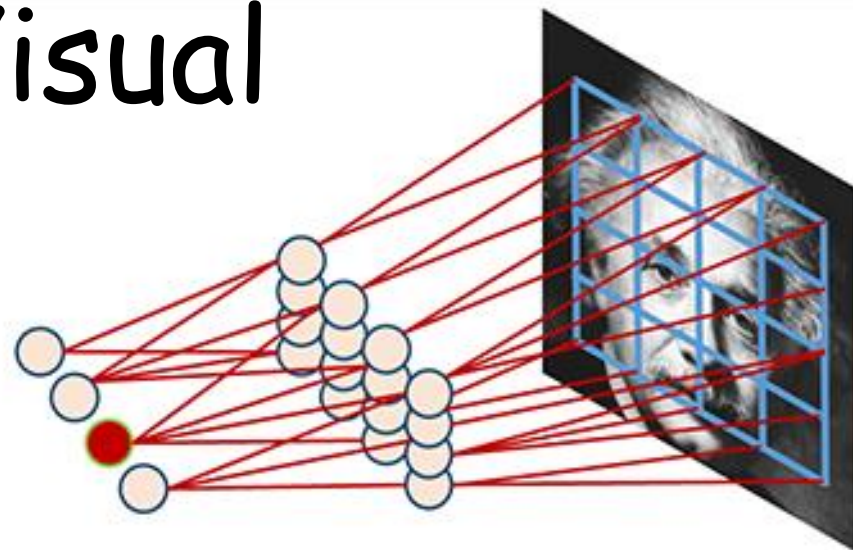




Arquitetura do Córtex Visual

- Como o ser humano entende essas imagens?
- Pesquisa de David H. Hubel e Torsten Wiesel (1958 e 1959);
- Neurônios com campos de percepções pequenos;
- Neurônios especializados:
 - Linhas;
 - Contornos;
 - Formas;
 - Reconhecimento de objeto;
 - Etc..

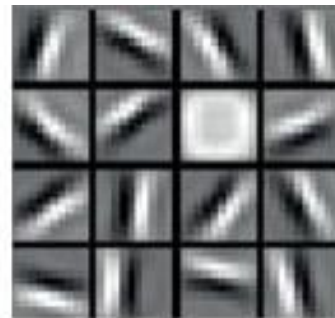
Arquitetura do Córtex Visual



- Neurônios com um campo visual maior:
 - Reagem a padrões mais complexos;
 - Eram baseados na saída dos neurônios de campo visual menor;
- Usavam os padrões simples encontrados para gerar padrões mais complexos;

Redes Convolucionais

- Rede com neurônios:
 - Especializados em reconhecimentos de padrões;
 - Campo de visão pequeno;
- Cada camada da rede se especializa em reconhecimentos mais complexos que a camada anterior;



edges



combinations of edges



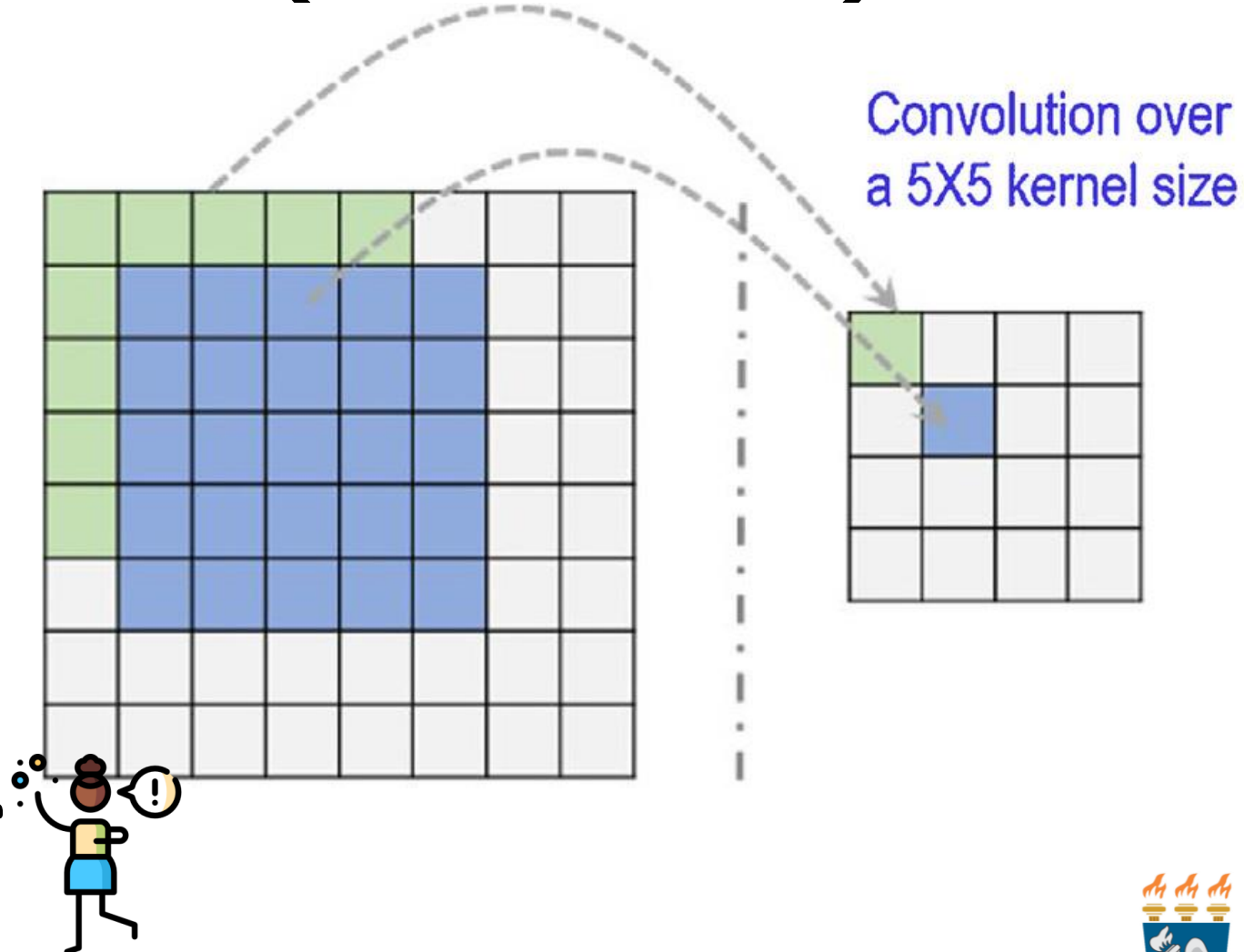
object models

Redes Convolucionais

- Cada camada é conectada a uma fração da camada anterior;
 - Campo de percepção;
- Uma rede pode conter múltiplas camadas;
- A primeira camada é sempre a imagem de entrada;

Redes Convolucionais (Conceitos)

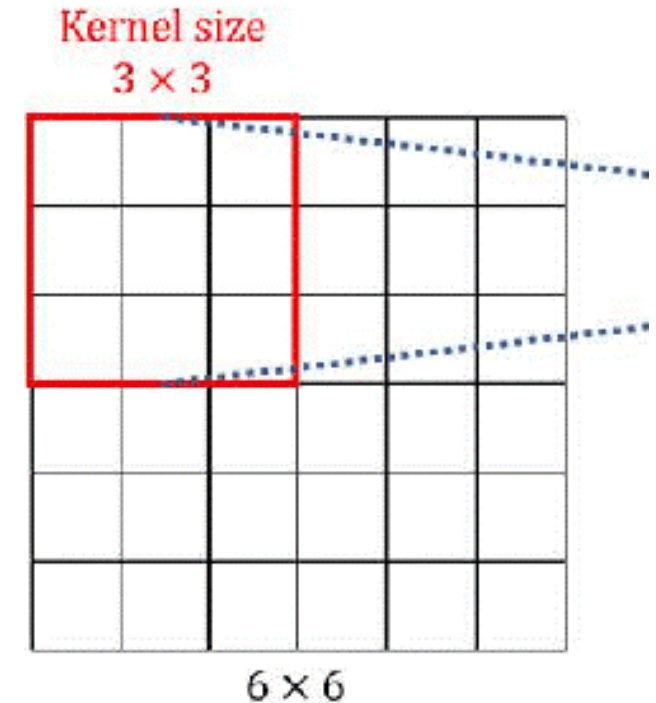
- **Kernel size:**
 - Campo de percepção do neurônio;
- É o tamanho do campo em altura x largura que o vai receber como input da camada anterior;
- Altura e largura podem ter tamanhos diferentes



Redes Convolucionais (Conceitos)

- Tamanho da nova camada atual:
 $novaAltura = (altura - kernelSize) + 1$
 $novaLargura = (largura - kernelSize) + 1$

- Campo de visão do neurônio:
 $f_j = kernelSize[altura]$
 $f_i = kernelSize[largura]$
$$novoN_{ij} = \sum_{x=0}^{f_j} \sum_{y=0}^{f_i} N_{(x+i)(y+j)}$$



Redes Convolucionais (Conceitos)

- **Padding Zero:**
- Algumas vezes queremos que a camada tenha o mesmo tamanho que a anterior;
- Adicionamos um espaçamento de zeros para preencher as casas não existentes;

Redes Convolucionais (Conceitos)

- Mesmo tamanho das camadas;

- Campo de visão do neurônio:

$$f_j = \text{kernelsize}[\text{altura}]$$

$$f_i = \text{kernelsize}[\text{largura}]$$

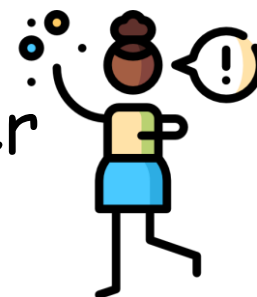
$$\text{nova}N_{ij} = \sum_{x=0}^{f_j} \sum_{y=0}^{f_i} N_{(x+i-\frac{f_i}{2})(y+j-\frac{f_j}{2})}$$

0 ₂	0 ₀	0 ₁	0	0	0	0
0 ₁	2 ₀	2 ₀	3	3	3	0
0 ₀	0 ₁	1 ₁	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

1	6	5
7	10	9
7	10	8

Redes Convolucionais (Conceitos)

- **Stride:**
- Define o passo entre um campo e percepção e outro;
- Podem ser definidos verticalmente e horizontalmente independentes;
- Sem a opção **padding zero**, parte do input podem ser ignorado;



Redes Convolucionais (Conceitos)

- Campo de visão do neurônio:

$$f_j = \text{kernelsize}[\text{altura}]$$

$$f_i = \text{kernelsize}[\text{largura}]$$

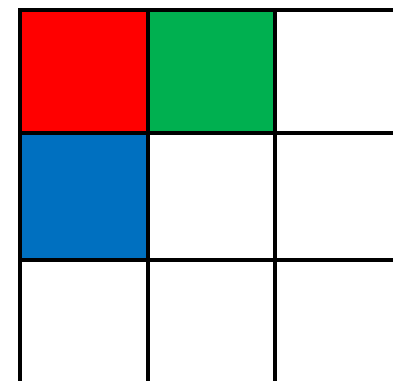
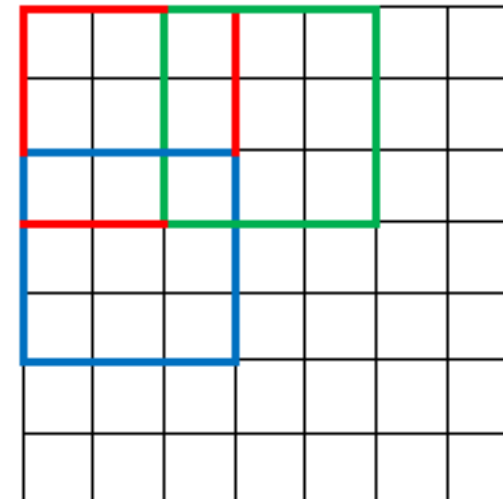
$$s_j = \text{stride}[\text{altura}]$$

$$s_i = \text{stride}[\text{largura}]$$

$$\text{nova}N_{ij} = \sum_{x=s_i}^{f_j} \sum_{y=0}^{f_i} N_{(x+i*s_i)(y+j*s_j)}$$

- Novo tamanho é dividido pelo stride;

7 x 7 Input Volume



Filtros

- É a matriz de pesos aplicada ao campo de percepção do neurônio.
- Os filtros vão ser produzidos a partir do treinamento da rede.

Input (+pad 1) (7x7x1) Filter W_0 (3x3x1) Output (3x3x1)

0	0	0	0	0	0	0
0	0	1	2	0	1	0
0	1	2	2	0	0	0
0	0	1	2	1	0	0
0	0	2	1	1	0	0
0	0	0	1	0	2	0
0	0	0	0	0	0	0

*

0	0	0
1	-1	0
1	1	-1

=

-1	3	-1
-2	1	2
0	-1	-2

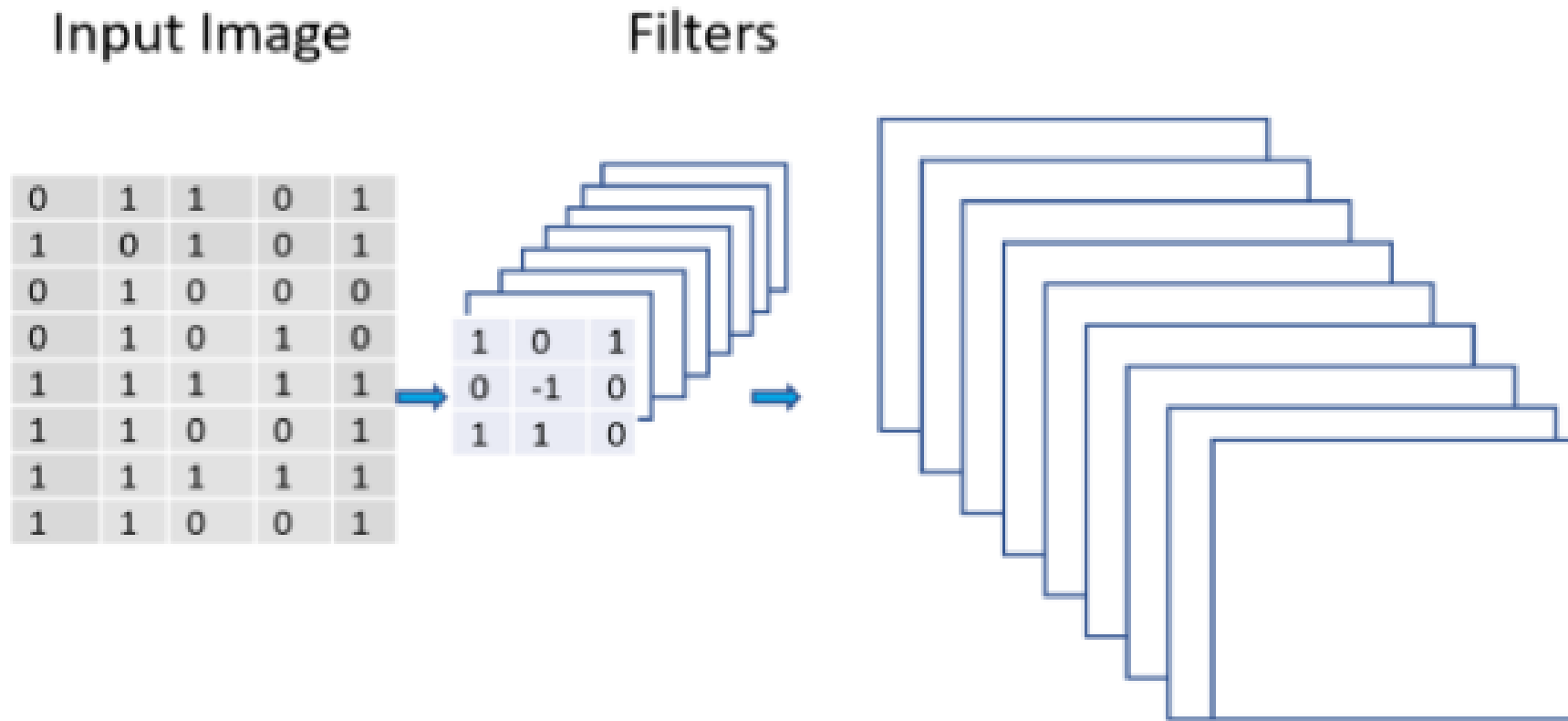
Mapa de Características

- Uma camada de neurônios que aplicam o mesmo filtro criam um **feature map**;
- Tem como objetivo destacar áreas da imagem no qual o filtro foi melhor ativado;
 - Mapa de ativação;
- A quantidade de **feature maps** definem o comprimento da camada;



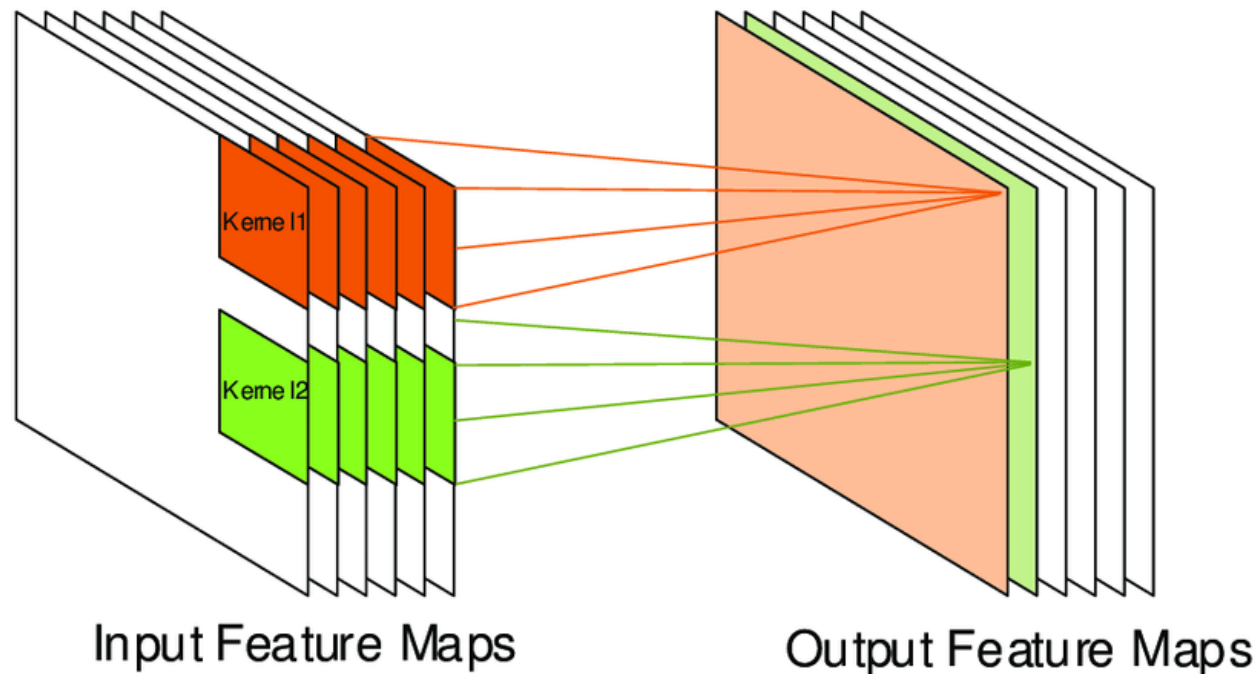
Mapa de Características

- Para cada filtro se obtém uma representação do mapa de características, em que cada pixel é a saída de um neurônio;



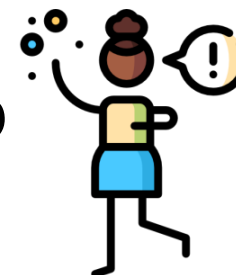
Mapa de Características

- Um neurônio da camada K está conectado aos neurônios da camada $K-1$ no campo de percepção dentro de todos os mapas de características;



Camada Convolutiva

- Uma camada convolutiva é o principal bloco de construção de uma rede convolutiva;
- Ela contém um grupo de filtros e gera um vários mapas de características;
- Nem todas as camadas de uma rede convolutiva são camadas convolutivas;



Canais

- Define o comprimento de cada nível;
- Numa imagem, geralmente os canais são divididos pelos tons de cores (RGB);
- A quantidade de **feature maps** representa também a quantidade de canais;

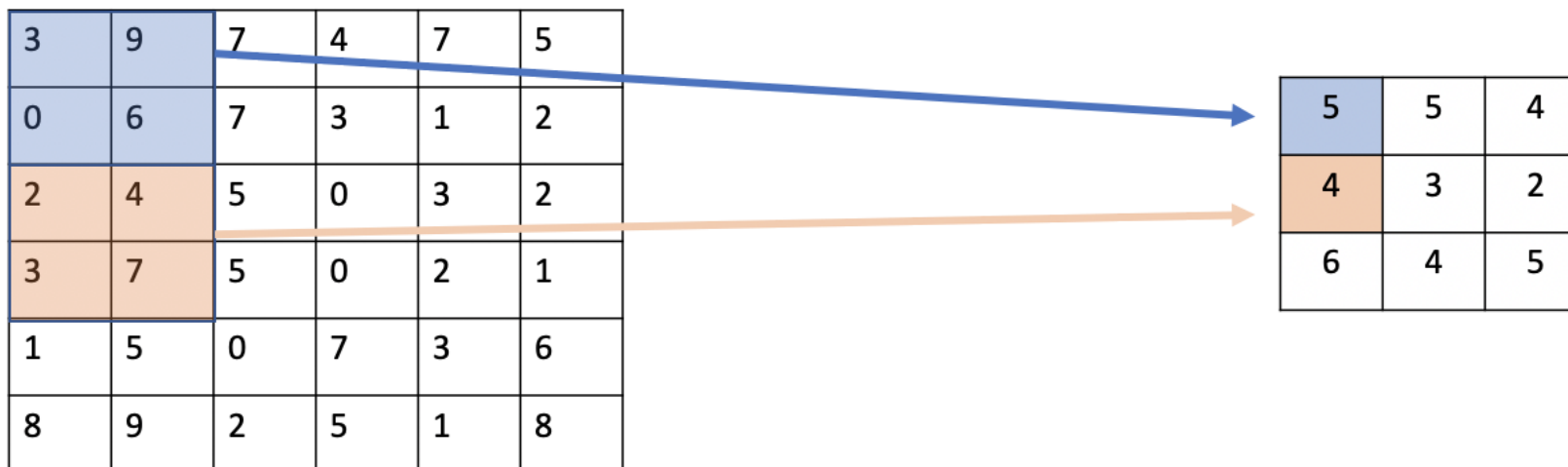
Camadas de Agrupamento

- Tem como objetivo reduzir a matriz de informações de uma camada para outra;
- Cada neurónio também tem seu campo de percepção;
- Não contém pesos, ao invés, utiliza funções de agregação;
 - Eg. Max, Min, Sum
- A saída terá a mesma quantidade de canais que a entrada;



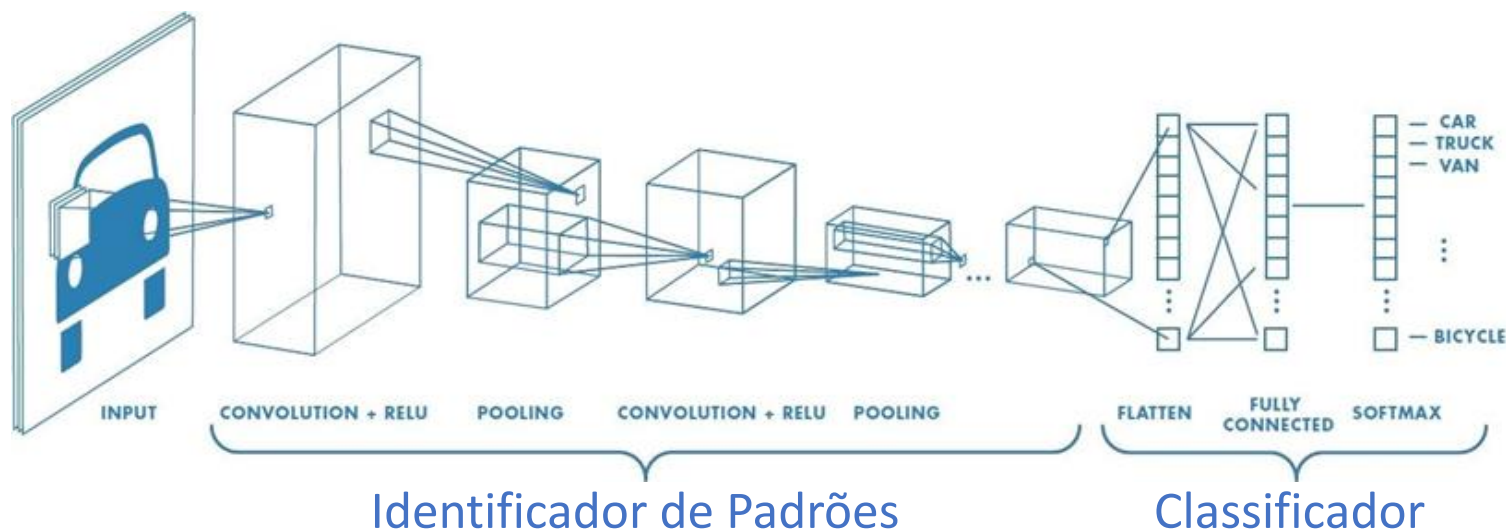
Camada de Agrupamento

- Exemplos concretos:
 - Sem zero padding;
 - Kernel size 2;
 - Stride 2;
- Executa a média arredondada nos dados de entrada;



Construção de uma rede Convolucional

- Geralmente uma rede convolucional é composta por:
 - Grupos de camadas convolucionais e uma camada de pooling;
 - Ao final algumas camadas de uma rede totalmente conectada;
 - A ultima camada resulta na predição de rede;



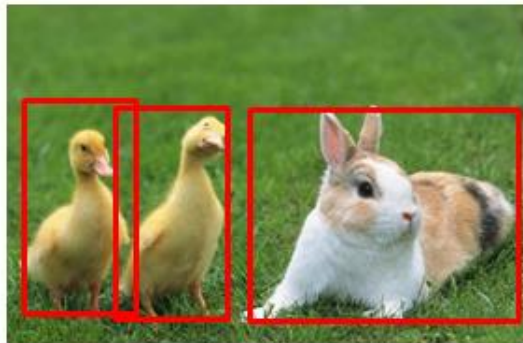
A green square and an orange square are positioned at the top left of the slide.

Construção de uma rede Convolutacional

- (Aqui vai um slide mais voltado para a construção de uma rede dessas. Talvez envolva código?? Quem sabe...)

Construindo a saída

- A saída de uma rede pode variar entre:
 - Classificação;
 - Localização;
 - Detecção de Objetos;
 - Segmentação;

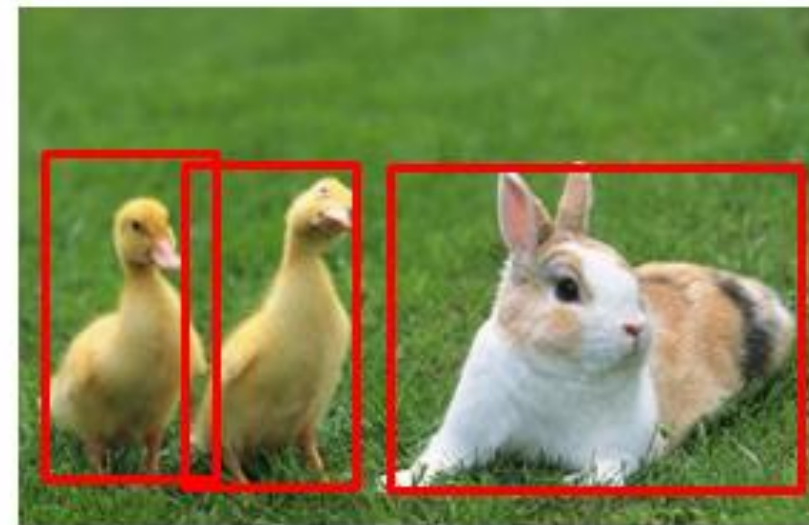


Classificação e Localização

- As ultimas camadas são uma rede neural totalmente conectada;
- Classificação:
 - Função softmax;
 - Cada neurônio de saída representa sim ou não;
- Localização:
 - Valor entre 0 e 1;
 - Cada neurônio de saída representa um eixo na posição do objeto detectado;

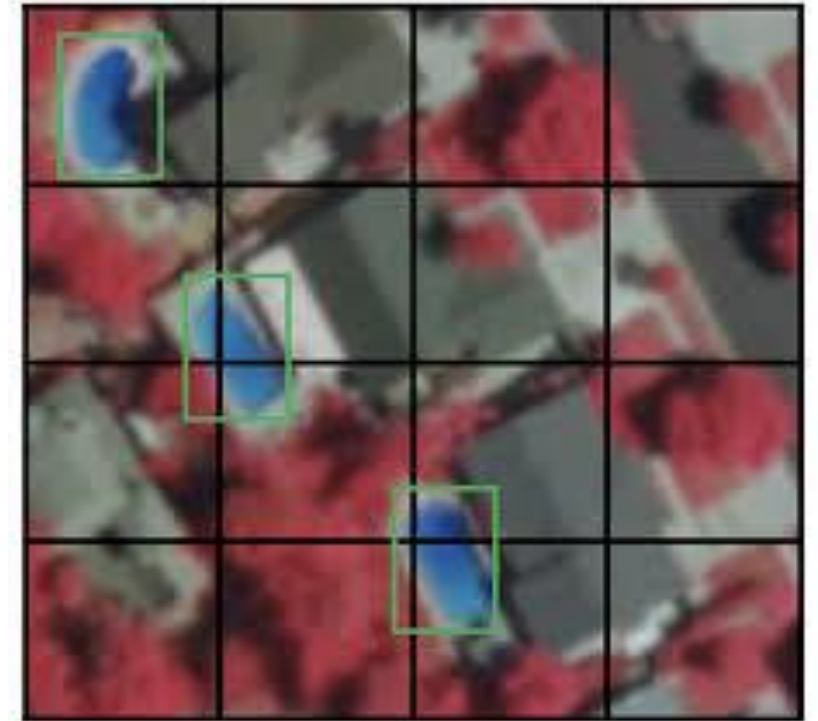
Classificação e Localização

- Número fixo de neurônios de saída;
 - Problemas para detectar mais de um objeto;
- Precisa que os dados de treinamento já estejam classificados ou localizados;



Detecção de Objetos

- Método simples:
 - Dividir a imagem principal;
 - Testar a classificação e localização de cada parte;
- Lista de todas as localizações;
- Desvantagens:
 - Não existe tamanho padrão;
 - Reprocessamento;
 - Dois objetos no mesmo pedaço;
 - Pode detectar o mesmo objeto;





Rede Totalmente Convolutacional

- Substituir a rede totalmente conectada:
 - Imagens de qualquer tamanho podem ser carregadas;
 - Imagens maiores resultam saídas maiores;
 - CNN podem ser convertidas em FCN;



Rede Totalmente Conectada

- Funcionamento:

Aqui vai uma imagem que ainda não terminei de confeccionar

Aqui vai uma explicação bem difícil e complicada

YOLO!!!!

- Arquitetura usando FCN;
- Método rápido e preciso de detecção de objetos;
 - Detecta em tempo real para vídeos;
- Código aberto desenvolvido em C;
 - Diversas variantes;





Implementação?

- Sei não....