

## Motivação CNNs

Alto custo computacional

Vulnerabilidade a variações na imagem

Dificuldade em capturar relações espaciais

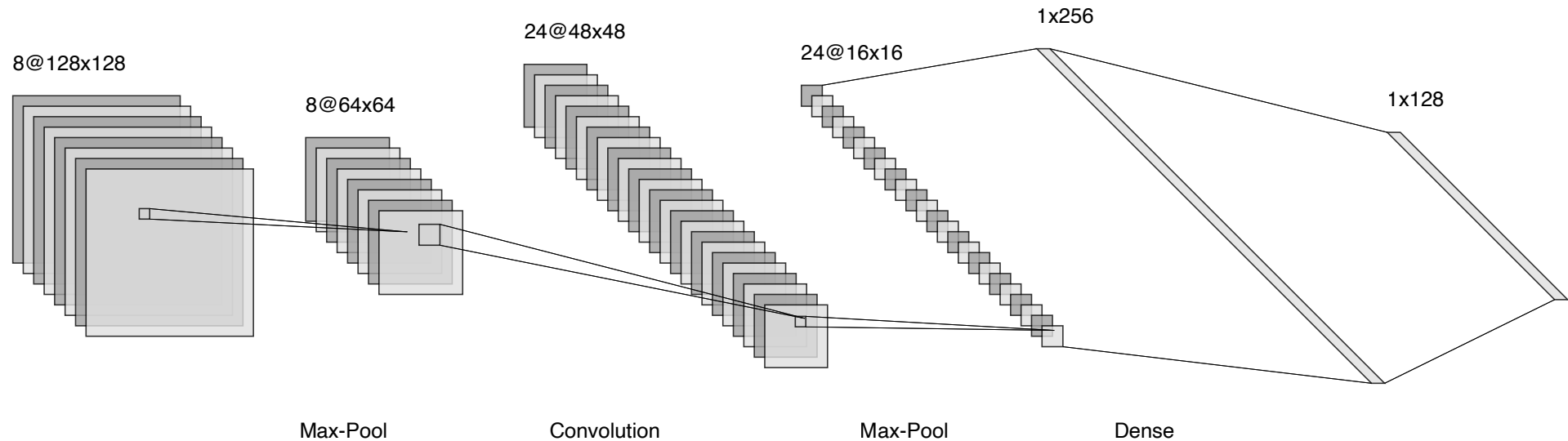
# Propósito

Processar imagens de forma eficiente

Capturar relações espaciais entre os pixels

Aprender relações hierárquicas de características

## Componentes de uma CNN Típica



Camada Convolutacional

Camada Densa

Camada de Pooling

Camada de Ativação

Aplicação: Visão Computacional

# Camada Convolutucional

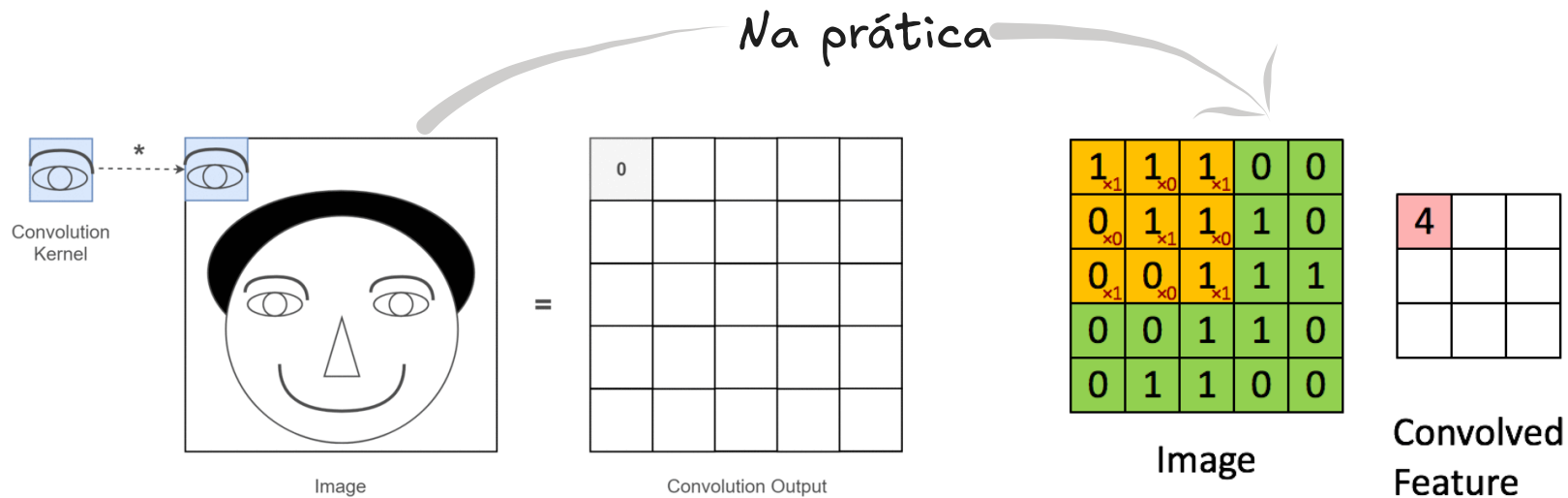
Convolução: aplicar filtros sobre a imagens de entrada

Cada filtro é uma pequena matriz de pesos

Saída do processo: mapa de características

desliza pela imagem

multiplica seus valores pelos dos pixels

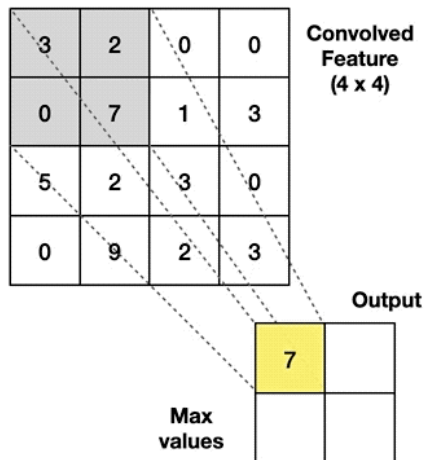


# Camada de Pooling

## Max Pooling

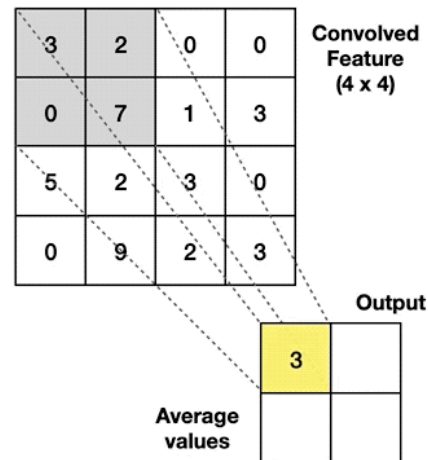
Take the **highest** value from the area covered by the kernel

Example: Kernel of size 2 x 2; stride=(2,2)



## Average Pooling

Calculate the **average** value from the area covered by the kernel



Reduz a dimensionalidade dos mapas de características gerados pelas camadas convolucionais.

Diminui o custo computacional e torna a rede mais robusta a pequenas variações na imagem

Os métodos de pooling mais comuns são:

## Camada de Ativação

Funções mais comuns:

- ReLU
- sigmoid
- tanh

Exemplo

**Input**

-249	-91	-37
250	-134	101
27	61	-153

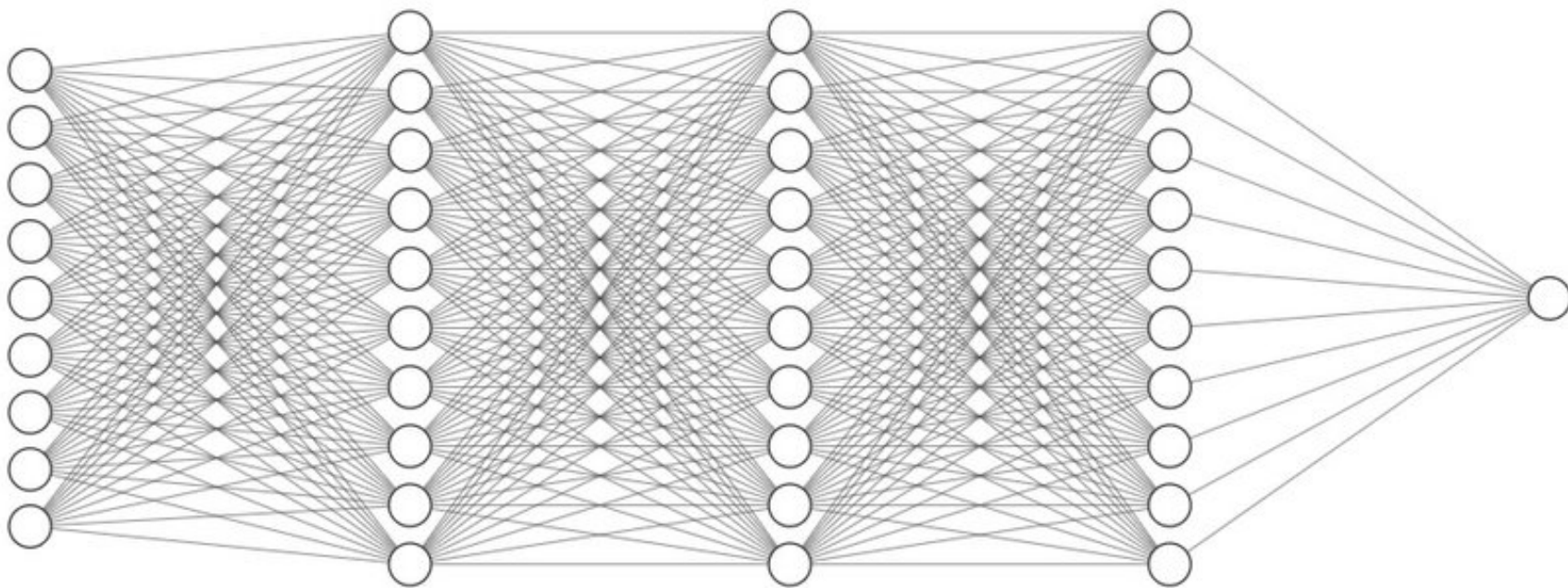
**ReLU**

0	0	0
250	0	101
27	61	0



## Camada Totalmente Conectada

Dense Layer





# Visão Computacional

- Classificação de imagens: determinar a categoria de um objeto presente na imagem (ex.: gato, cachorro, carro).
- Reconhecimento de objetos: identificar e localizar múltiplos objetos em uma cena.
- Segmentação de imagens: dividir a imagem em regiões com significado semântico (ex.: identificar o contorno de um objeto).