



Grupo Deep Learning

Redes Neurais Convolucionais

A stylized, low-poly illustration of a study room. In the foreground, there is a dark desk with a laptop and a red office chair. On a shelf in the background, a small panda is sitting next to a book. The room has a warm, orange-toned light.

Beatriz
Daniel
Pedro
Victor



Trajetória até o tema

01



O que é uma CNN?

02



Esmiuçando CNN

03



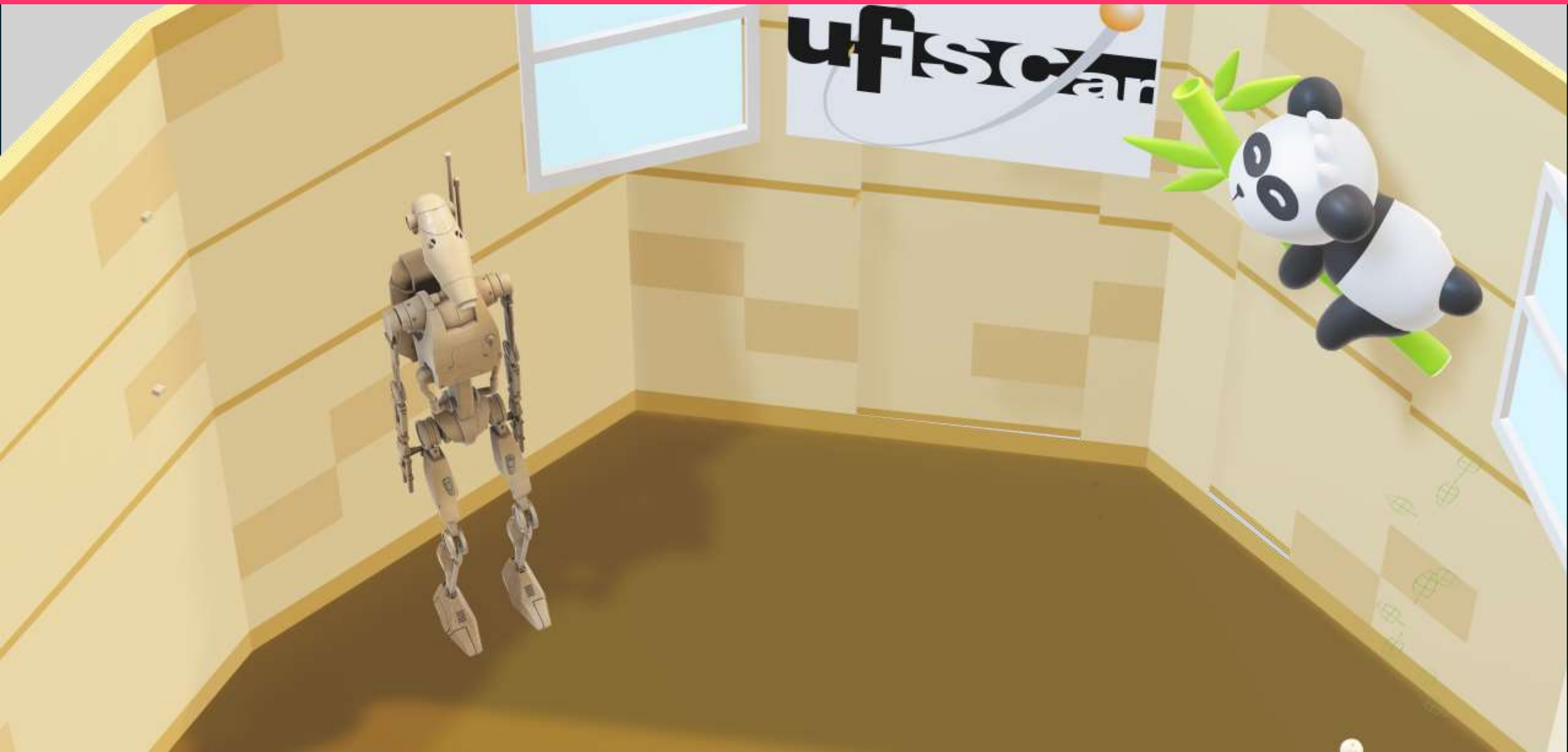
CNN na prática

04

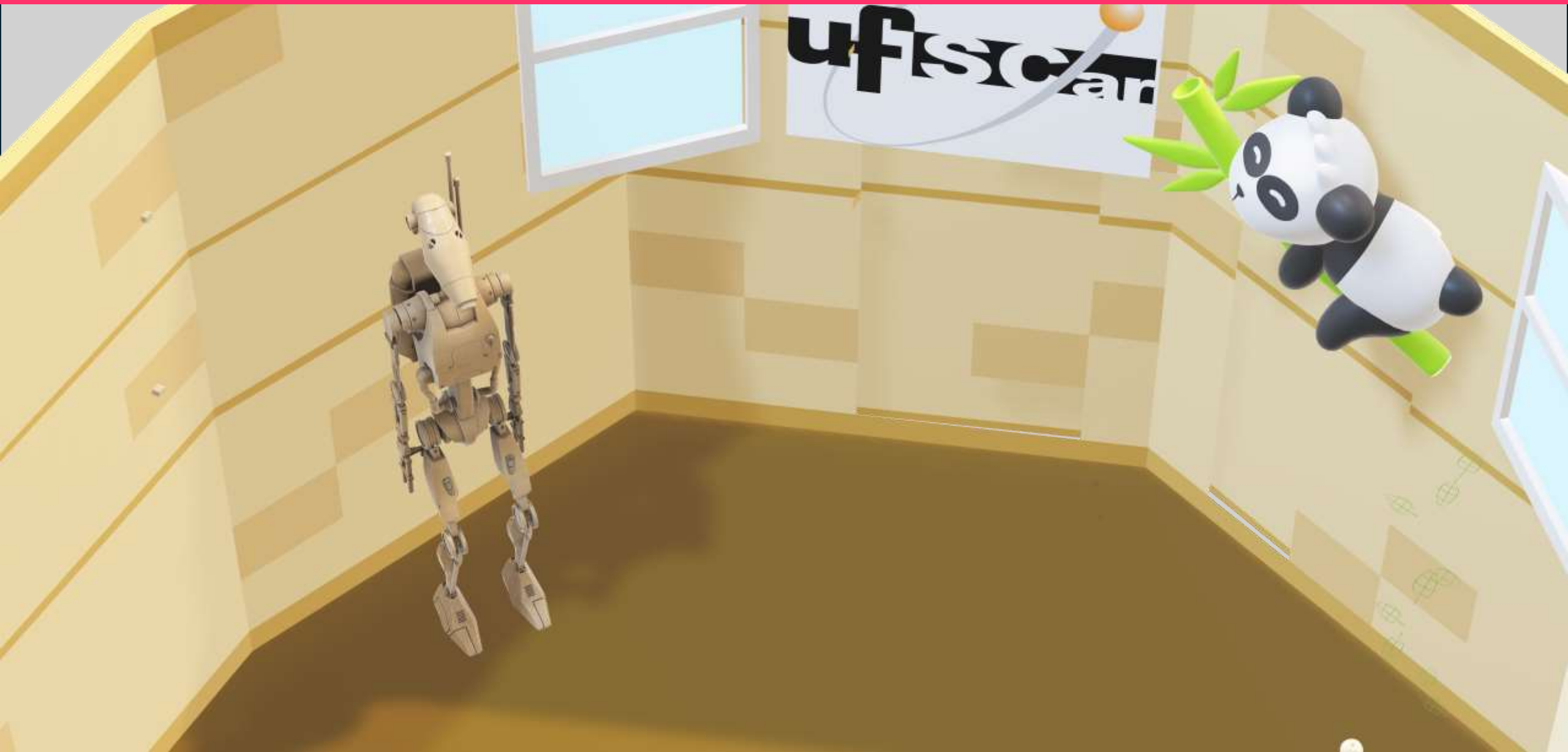
Trajetória até o tema



Trajetória até o tema



Trajetória até o tema



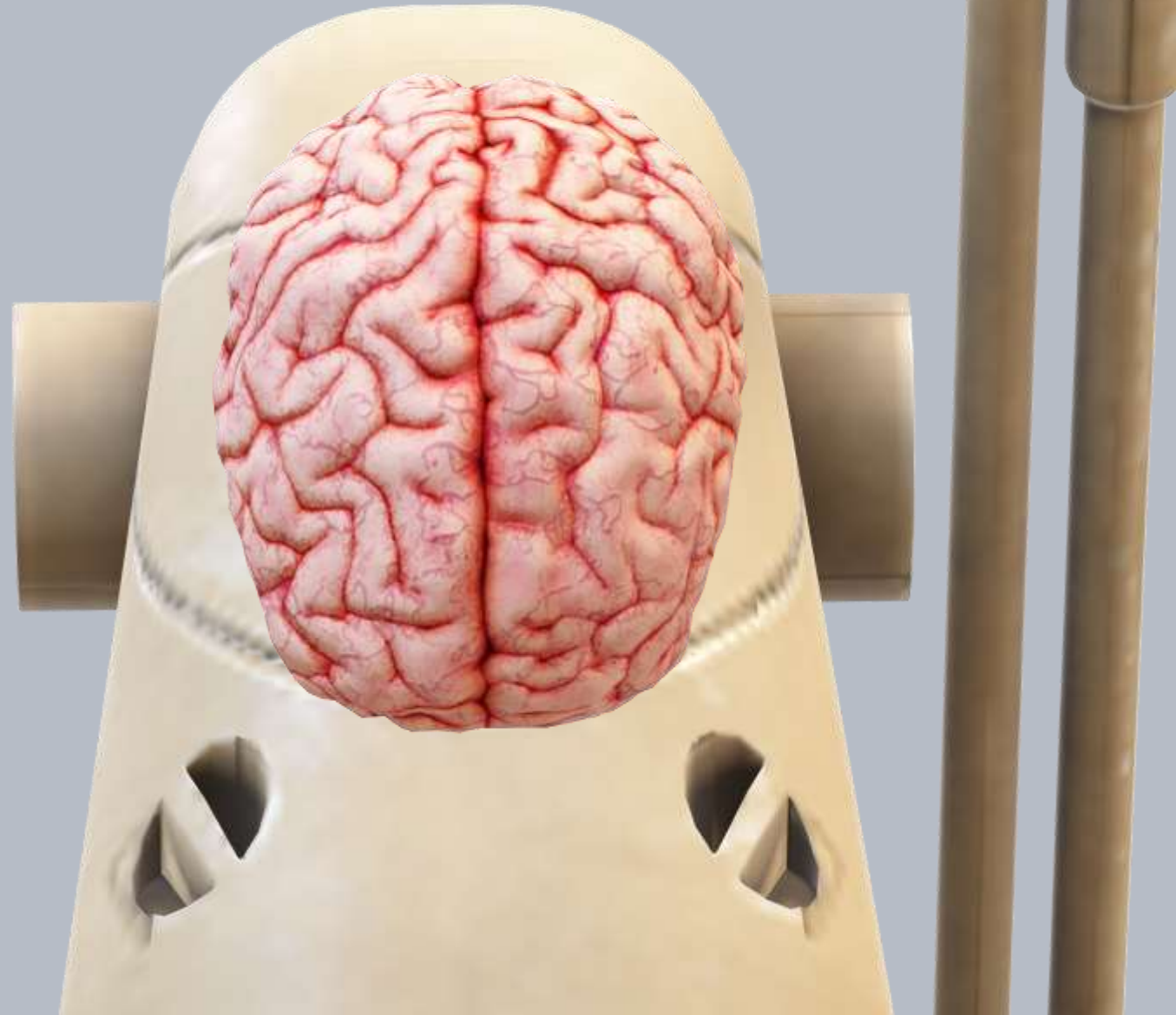
Trajetória até o tema

Deep Learning



Trajetória até o tema

Deep Learning



Trajetória até o tema

Redes Neurais



Trajectoria até o tema

**Deteccção
de Voz**



**Zerar
jogo
sozinho**

**Visão
Computacional**



**Redes
Generativas**

Trajetória até o tema



**Redes
Generativas**

Trajetória até o tema

Redes Generativas



GANs

**Criar samples de gêneros
musicais**

Trajetória até o tema

GANs



Gerador



Discriminador

Trajetória até o tema

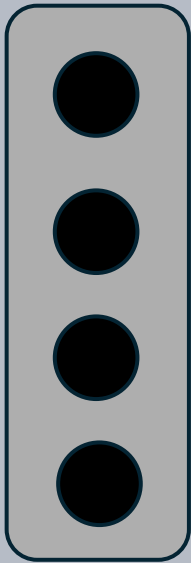
GANs



Gerador

Trajetória até o tema

GANs



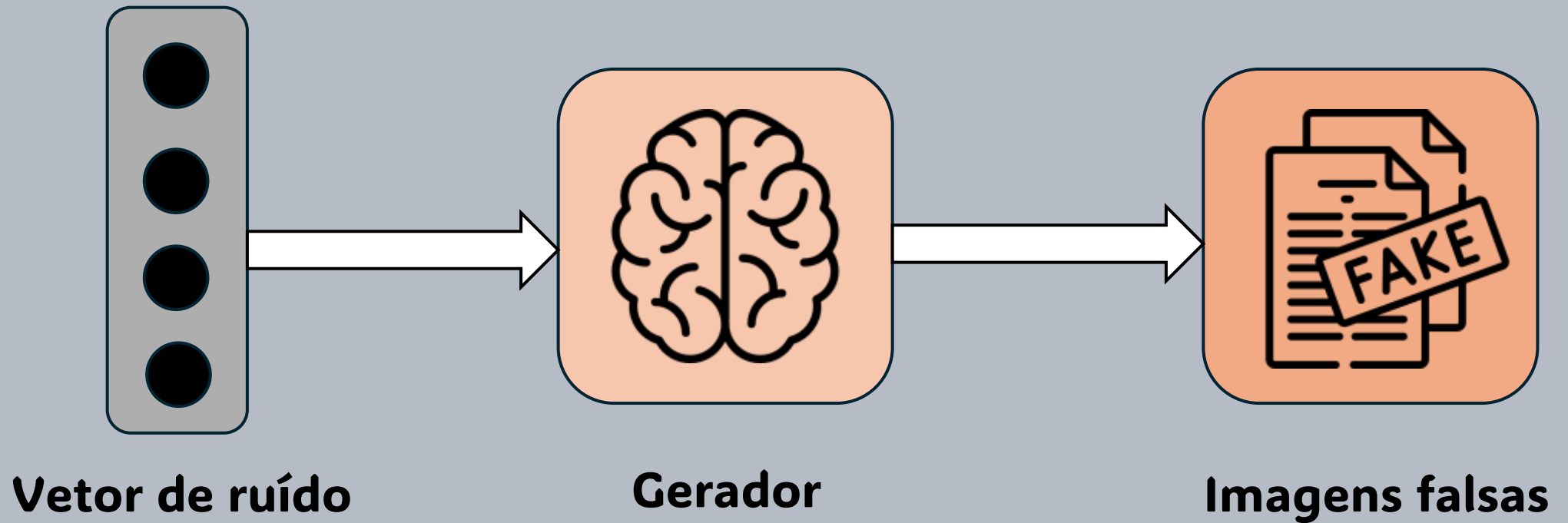
Vetor de ruído



Gerador

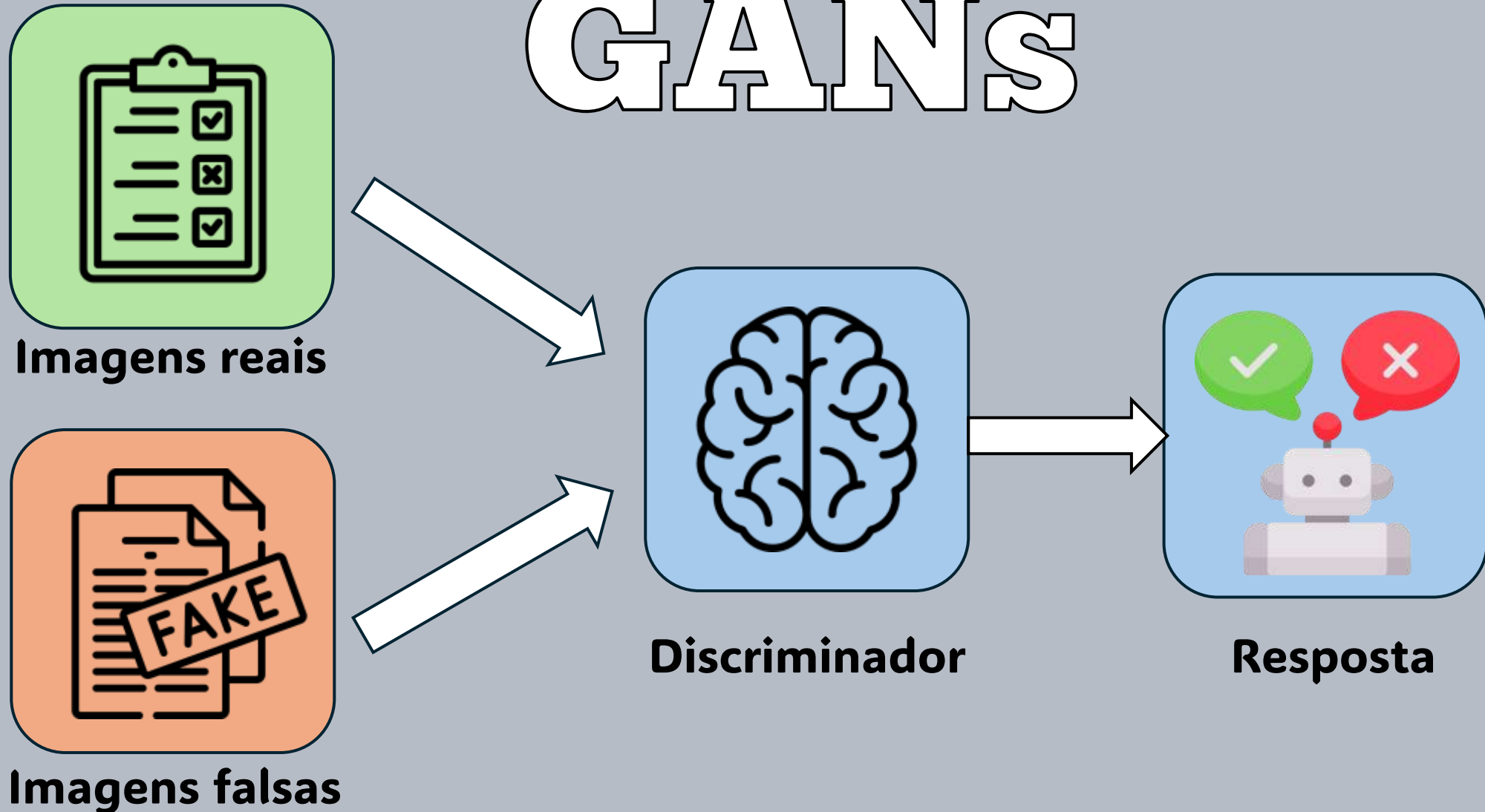
Trajetória até o tema

GANs



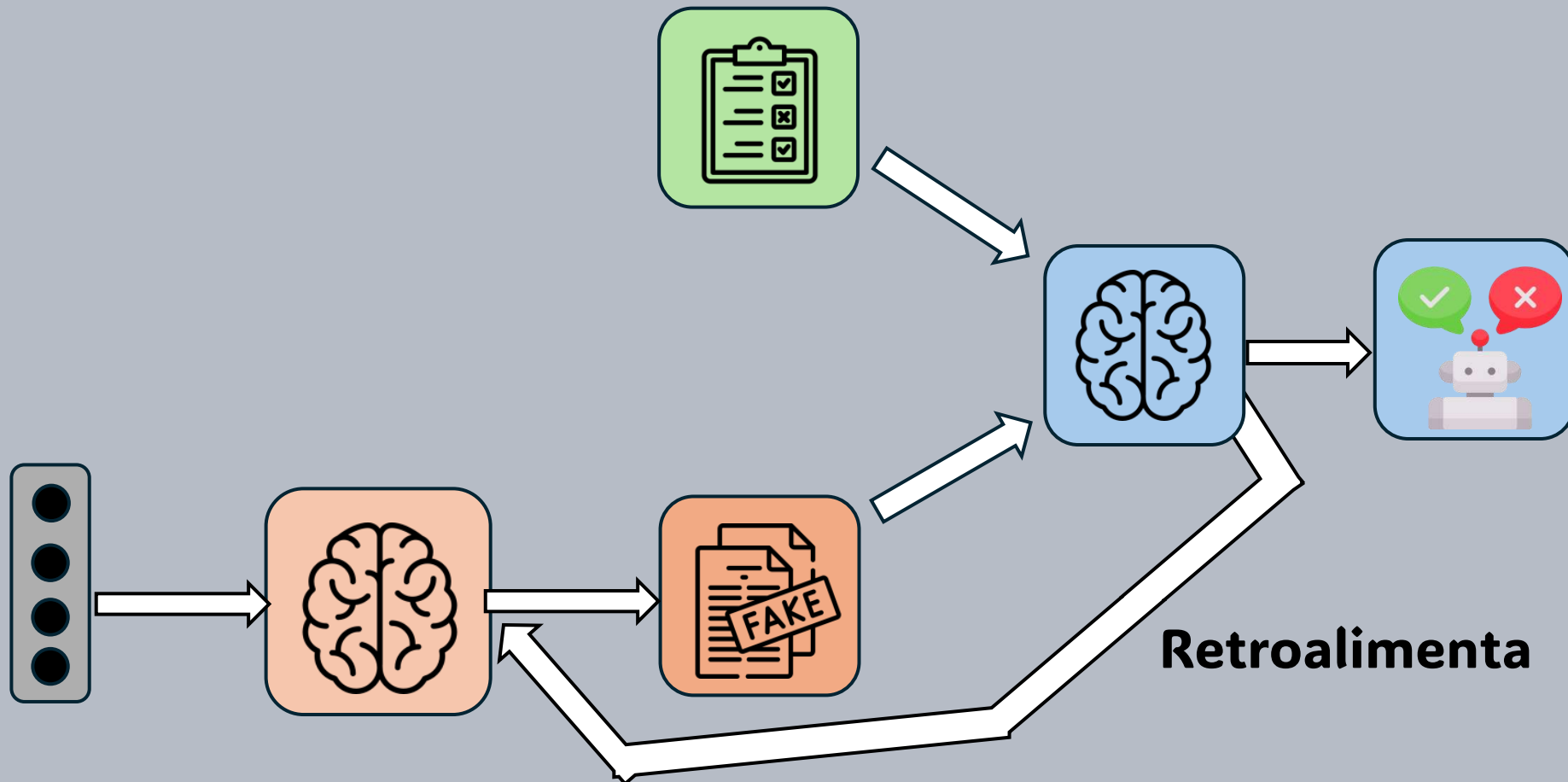
Trajetória até o tema

GANs



Trajetória até o tema

GANs



Trajetória até o tema

GANs

- Para o Gerador: A perda indica quão bem ele conseguiu **enganar o Discriminador**.
- Para o Discriminador: A perda indica quão bem ele conseguiu identificar os dados reais e falsos.



Gerador



Discriminador

Trajetória até o tema

GANs



Discriminador



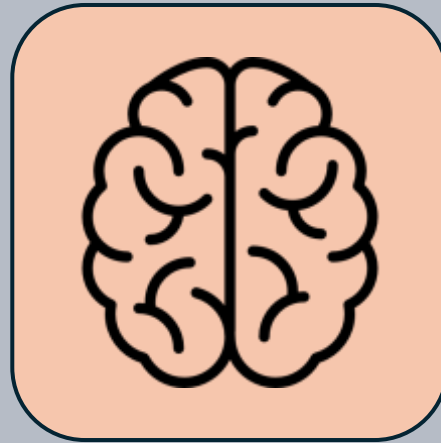
Gerador

Trajetória até o tema

GANs



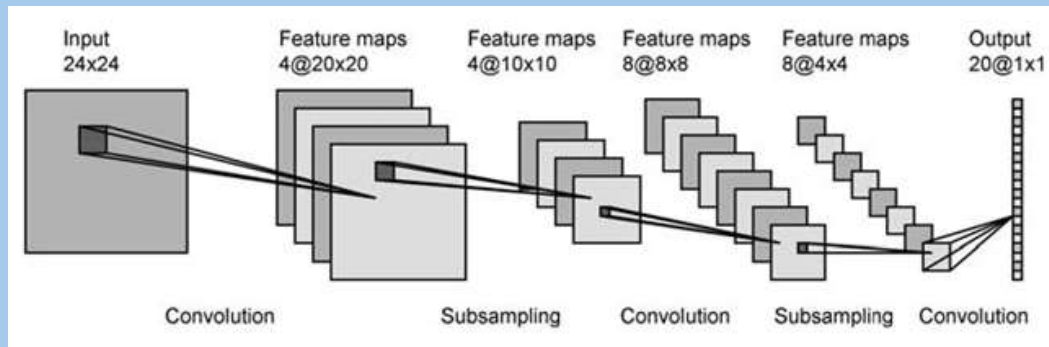
Discriminador



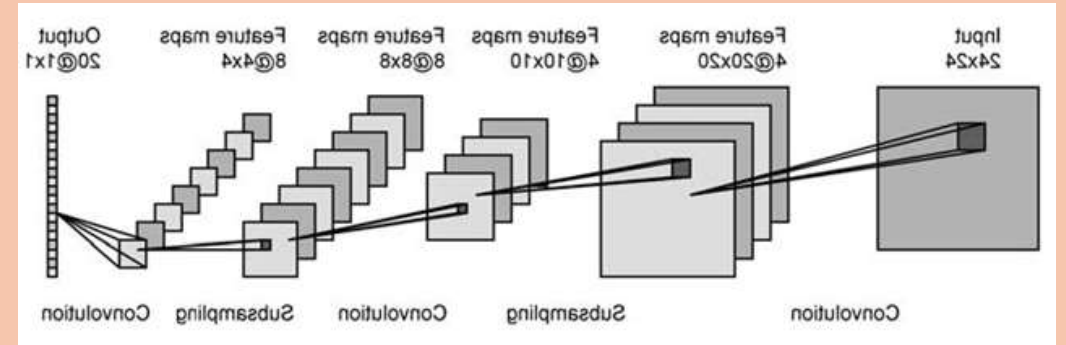
Gerador

Trajetória até o tema

GANs



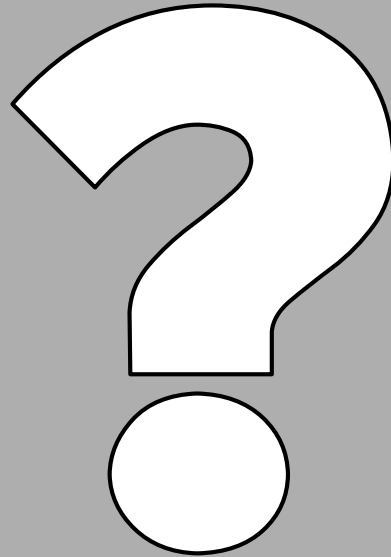
Discriminador



Gerador

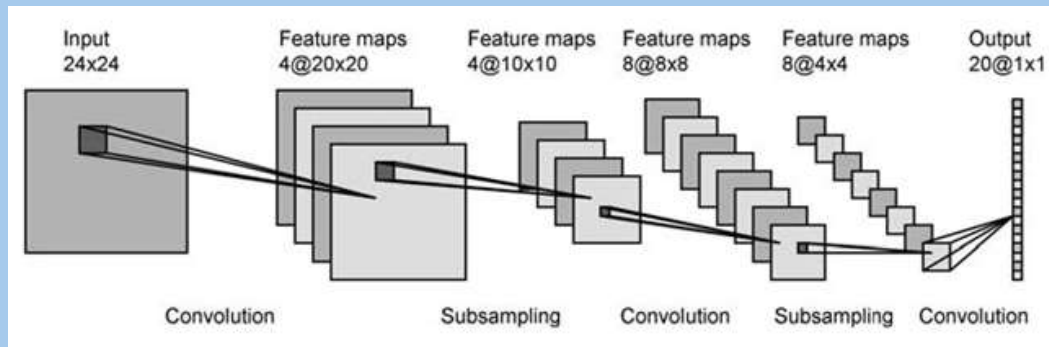
Trajetória até o tema

GANs

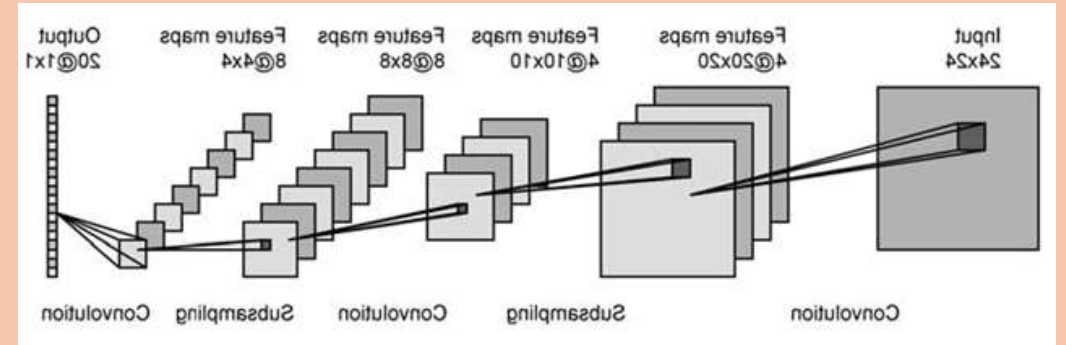


Trajetória até o tema

CNNs

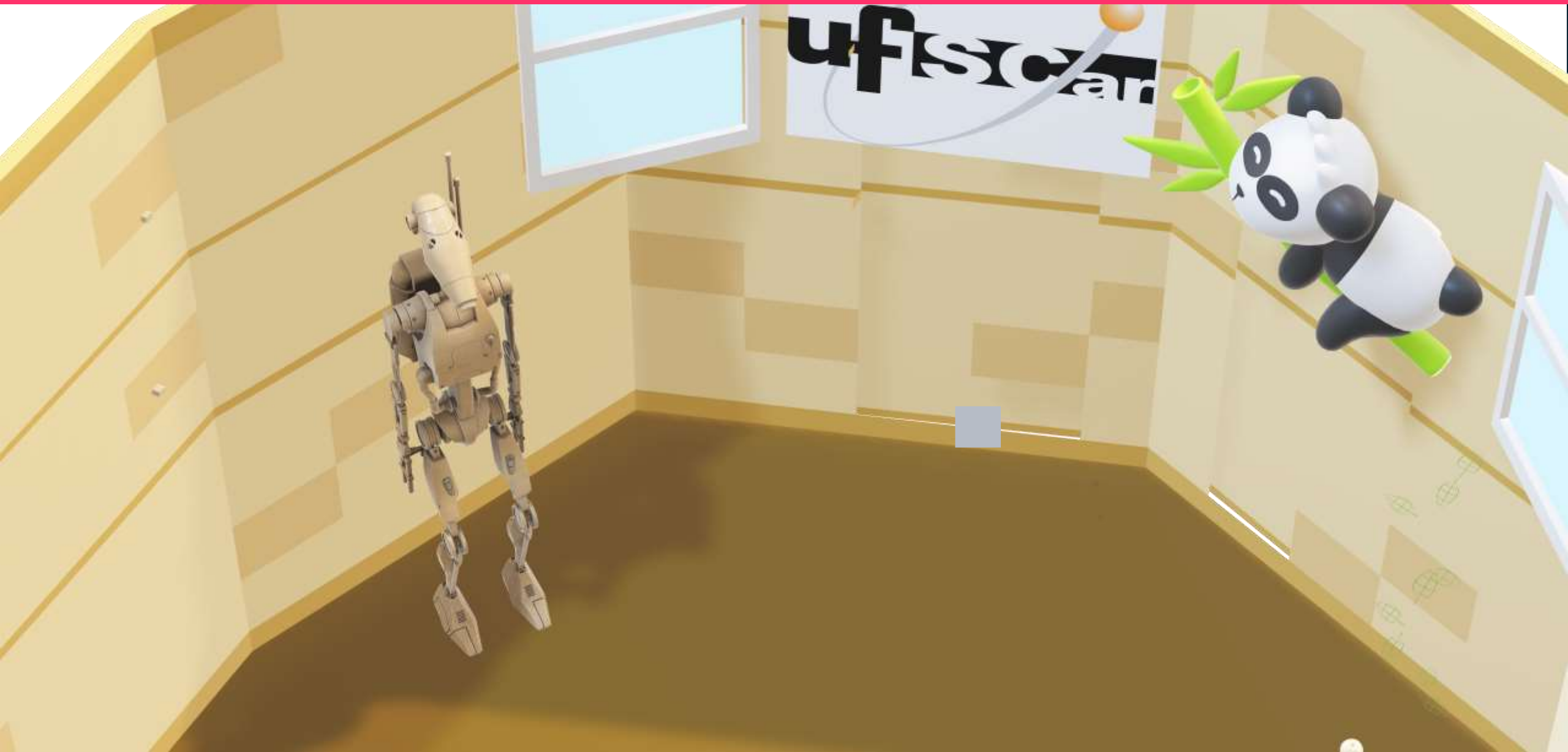


Discriminador



Gerador

Trajetória até o tema





Trajetória até o tema

01



O que é uma CNN?

02



Esmiuçando CNN

03



CNN na prática

04

O que é uma CNN?



O que é uma CNN?



O que é uma CNN?

- **Antes das CNNs**, métodos manuais e demorados de extração de características eram usados para identificar objetos em imagens.



!=



O que é uma CNN?

É um algoritmo que:

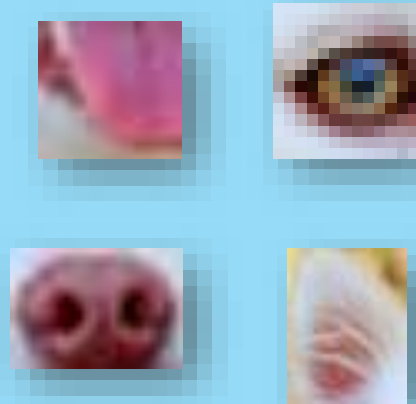
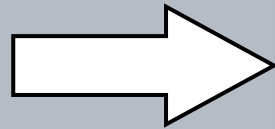
- Capta uma imagem de entrada



O que é uma CNN?

É um algoritmo que:

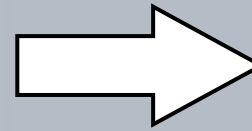
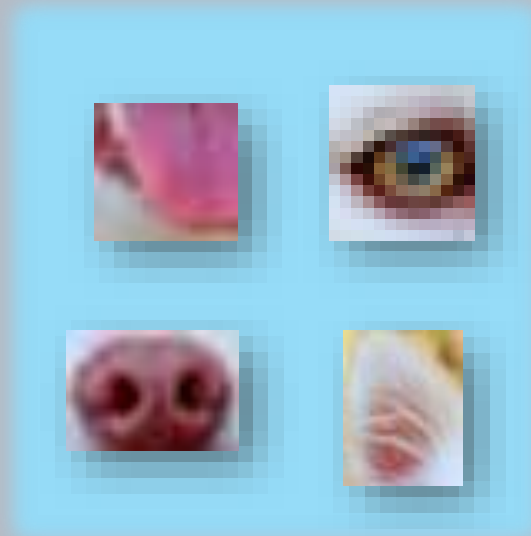
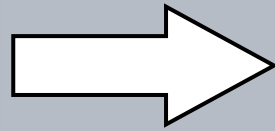
- Capta uma imagem de entrada
- Atribui importância a vários aspectos da imagem



O que é uma CNN?

É um algoritmo que:

- Capta uma imagem de entrada
- Atribui importância a vários aspectos da imagem
- Consegue dizer a % que representa



O que é uma CNN?



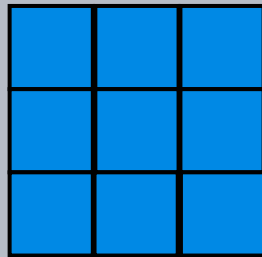
O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?

O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?

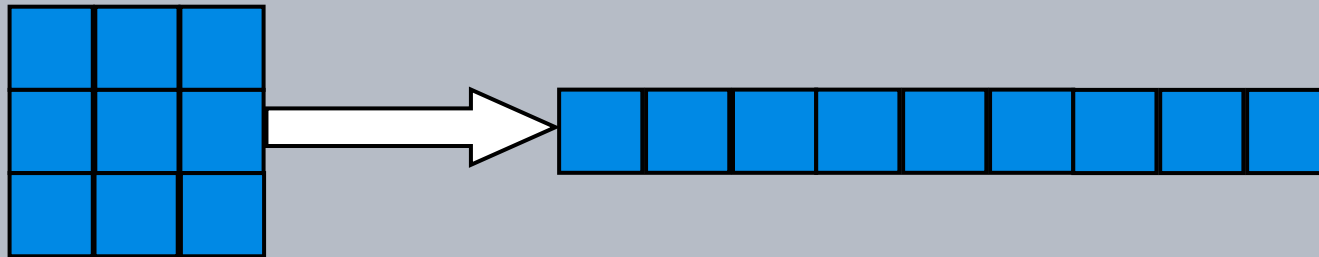
Imagens são matrizes, por que não utilizar?



O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?

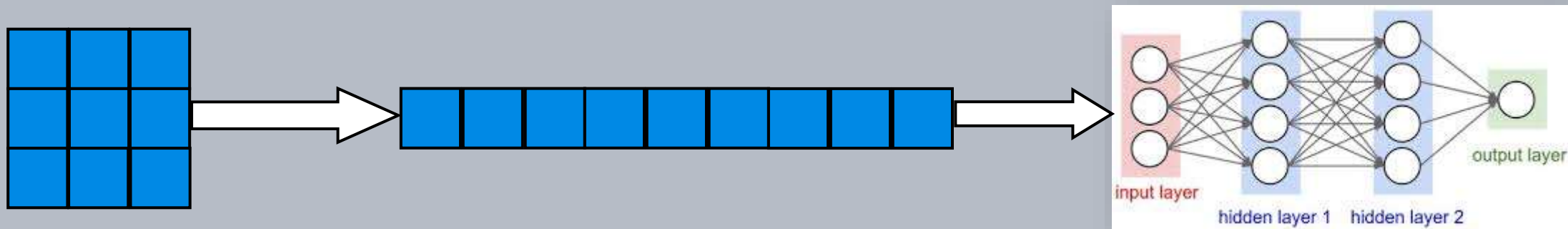
Basta achatar a imagem e alimentá-lo para um Perceptron Multi-Layer para fins de classificação



O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?

Basta achatar a imagem e alimentá-lo para um Perceptron Multi-Layer para fins de classificação



O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?

Em casos de imagens binárias **extremamente** básicas, o método **pode mostrar uma pontuação de precisão média** durante a previsão de classes, **mas teria pouca** ou nenhuma precisão quando se trata de **imagens complexas** com dependências de pixel por toda parte.



O que é uma CNN?

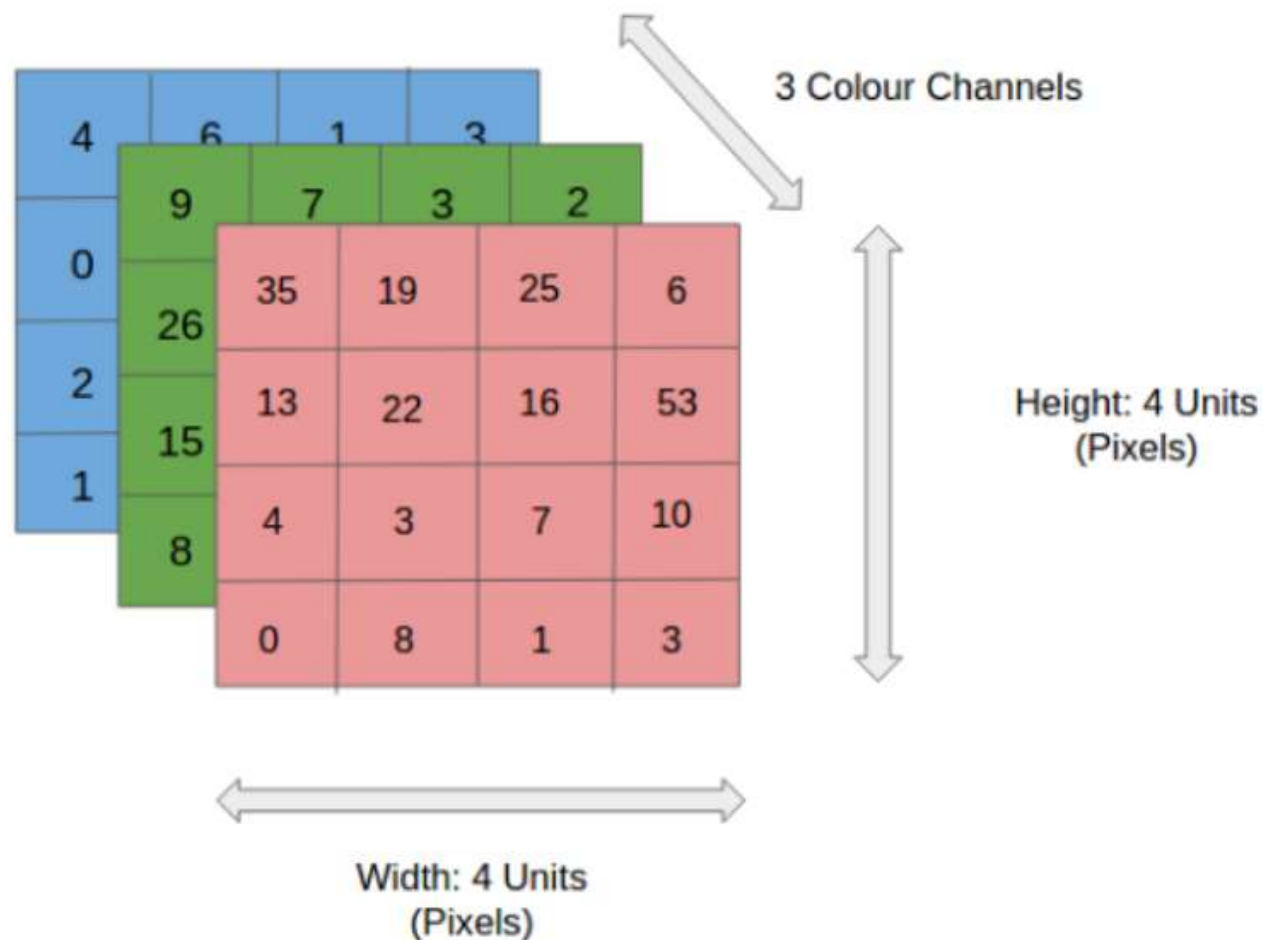
Por que usar ConvNets e não rede feed-forward?

Uma CNN é capaz de capturar com sucesso as dependências espaciais e temporais em uma imagem através da **aplicação de filtros relevantes**.

A arquitetura executa um **melhor ajuste** ao conjunto de dados da imagem devido à redução no número de parâmetros envolvidos e à capacidade de reutilização dos pesos.

O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?



O que é uma CNN?

Por que usar ConvNets e não rede feed-forward?

Na figura mostrada, temos uma imagem RGB (Red – Green – Blue) que foi separada por seus três planos coloridos – Vermelho, Verde e Azul.

Existem vários desses espaços de cores nos quais existem imagens – Escala de cinza, RGB, HSV, CMYK etc.

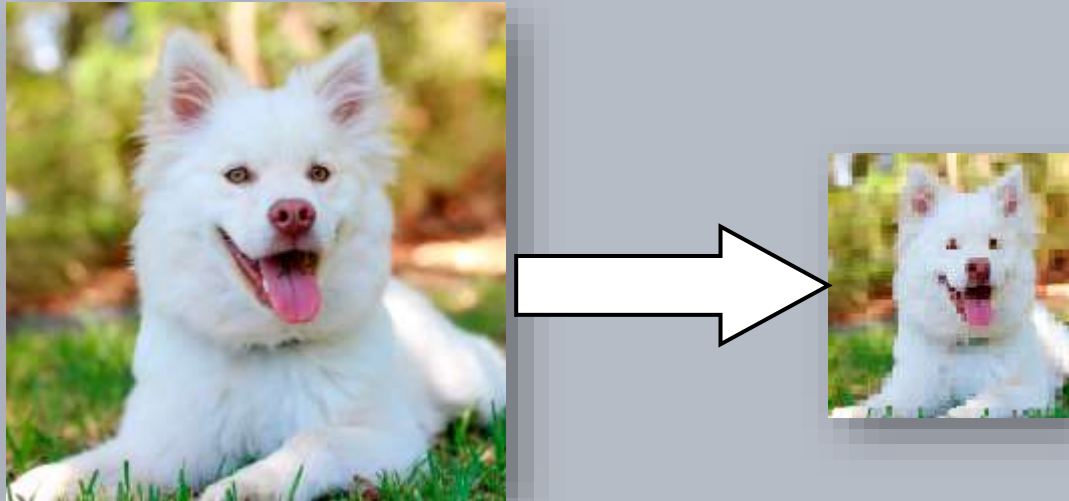
Você pode imaginar como a computação ficaria intensiva assim que as imagens atingissem dimensões, digamos, 8K (7680×4320):

33.177.600 (1 Canal)

O que é uma CNN?

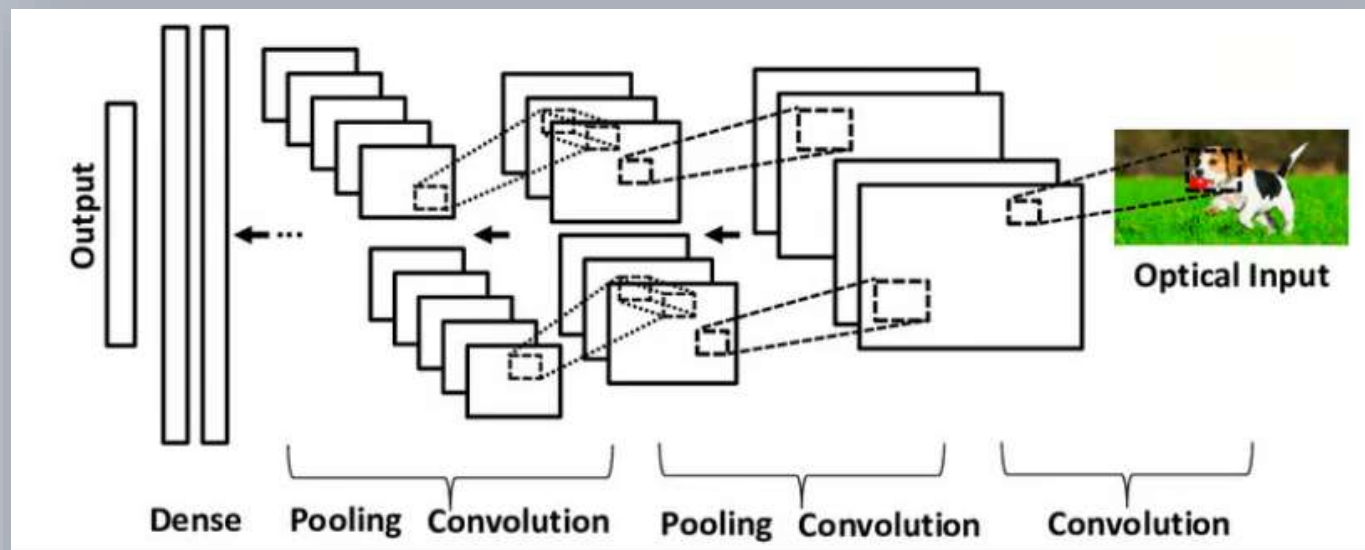
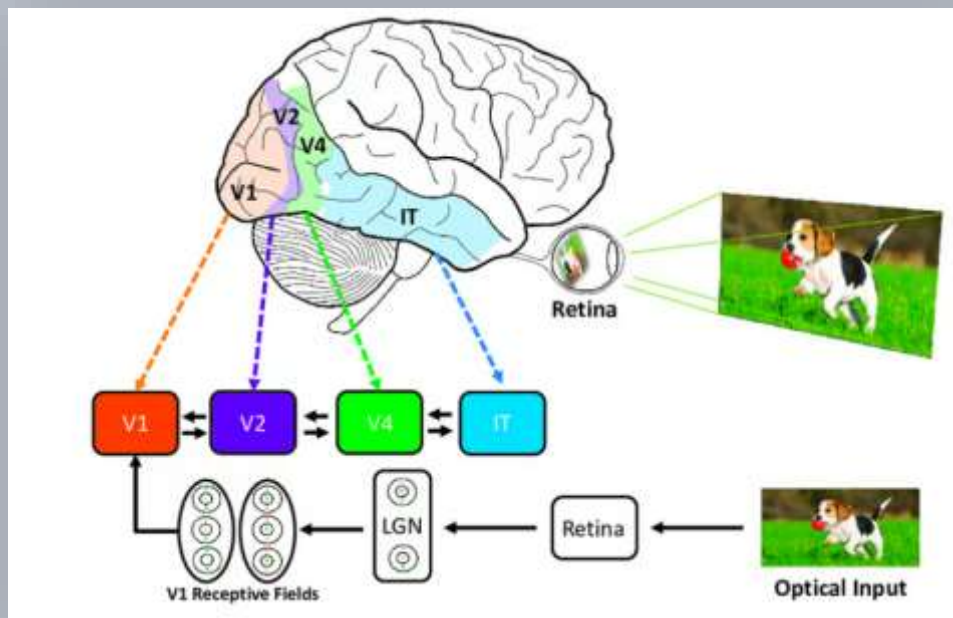
A função da CNN é **reduzir as imagens** para uma forma mais fácil de processar, **sem perder recursos** que são críticos para obter uma boa previsão.

Isso é importante quando queremos projetar uma arquitetura que não seja apenas boa em recursos de aprendizado, **mas que também seja escalável** para conjuntos de dados massivos.



O que é uma CNN?

Paralelismo com o cérebro humano





Trajetória até o tema

01



O que é uma CNN?

02



Esmiuçando CNN

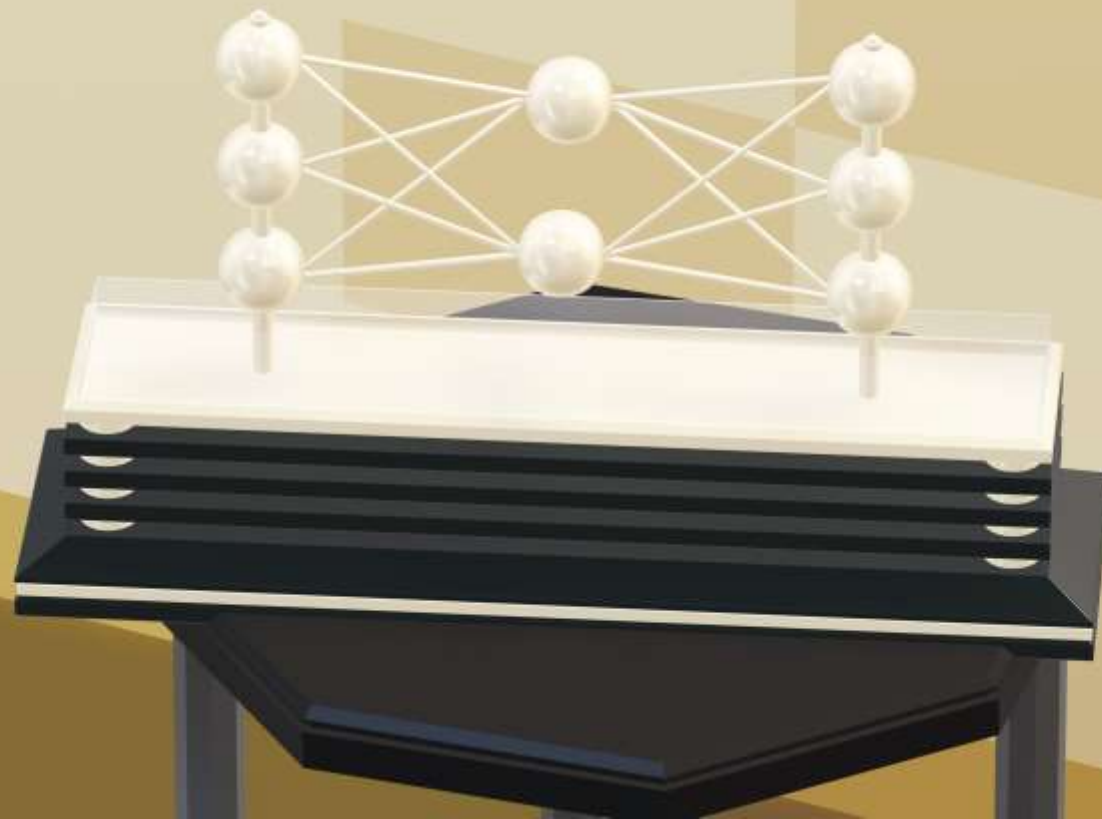
03



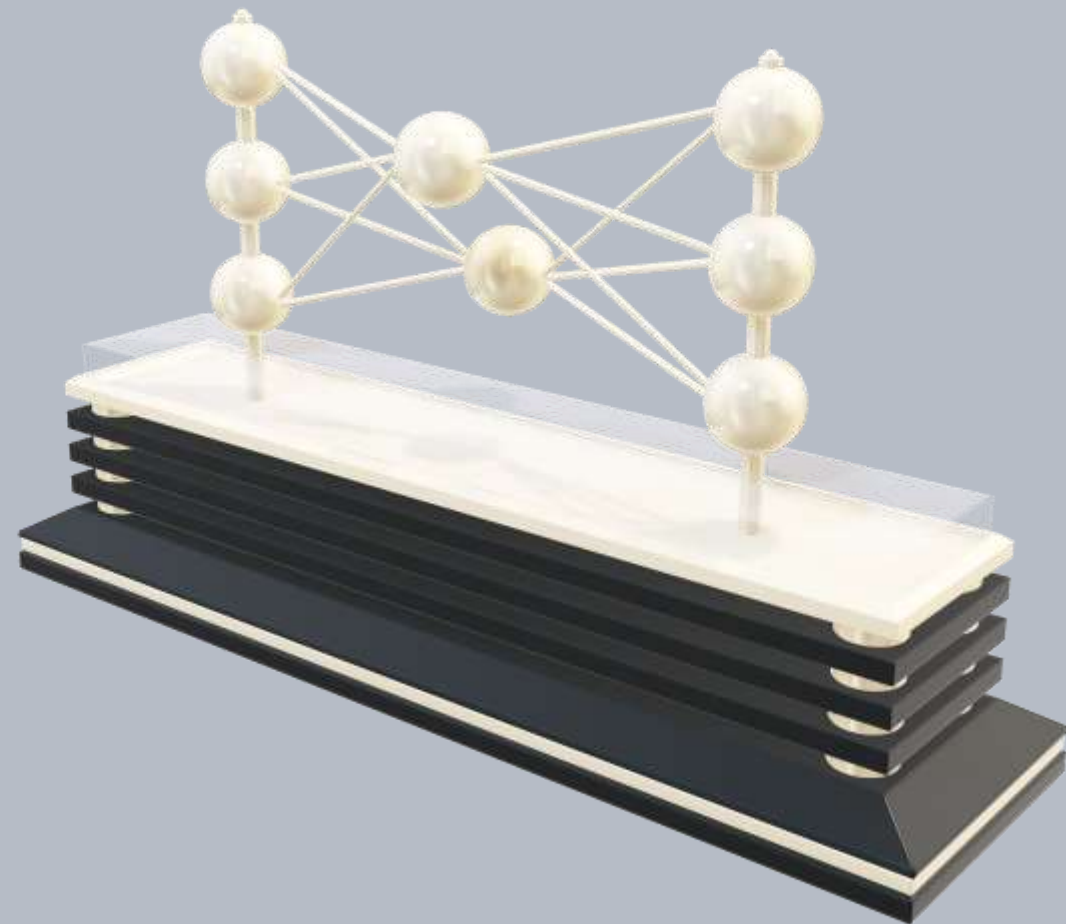
CNN na prática

04

Esmiuçando CNN



Esmiuçando CNN



Esmiuçando CNN

O que é kernel?

No processamento de imagens, um **kernel** (também chamado de **filtro**) é uma pequena matriz usada na **operação de convolução** para modificar ou extrair características de uma imagem, como bordas, texturas e padrões.



Esmiuçando CNN

Exemplos de kernel

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

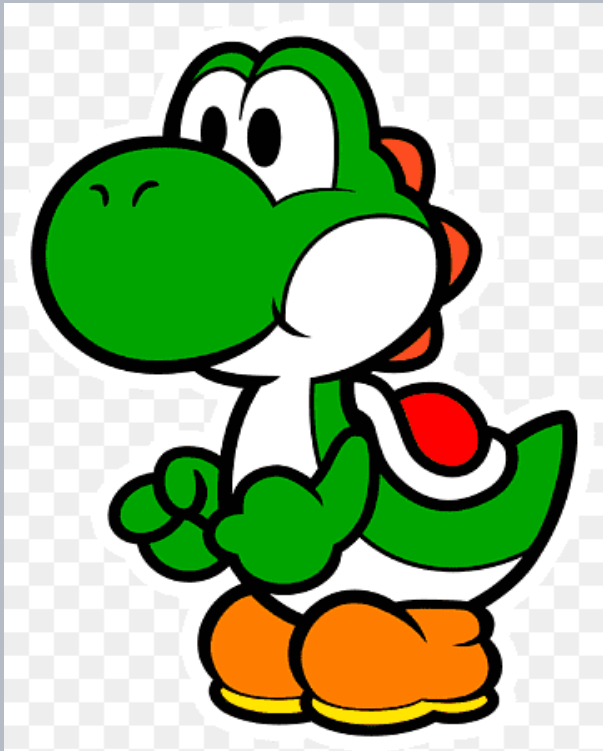
Esmiuçando CNN

Exemplos de kernel



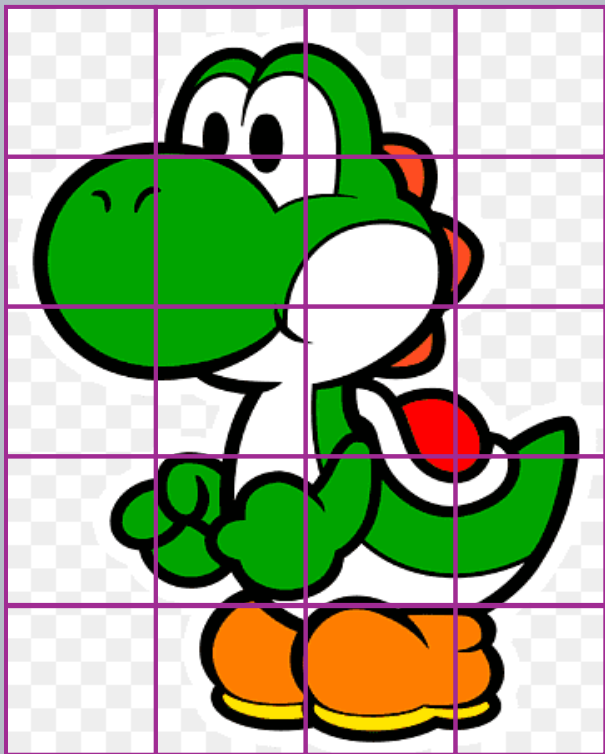
Esmiuçando CNN

Exemplos de kernel



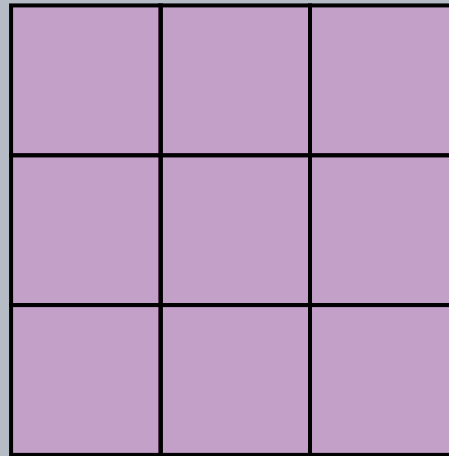
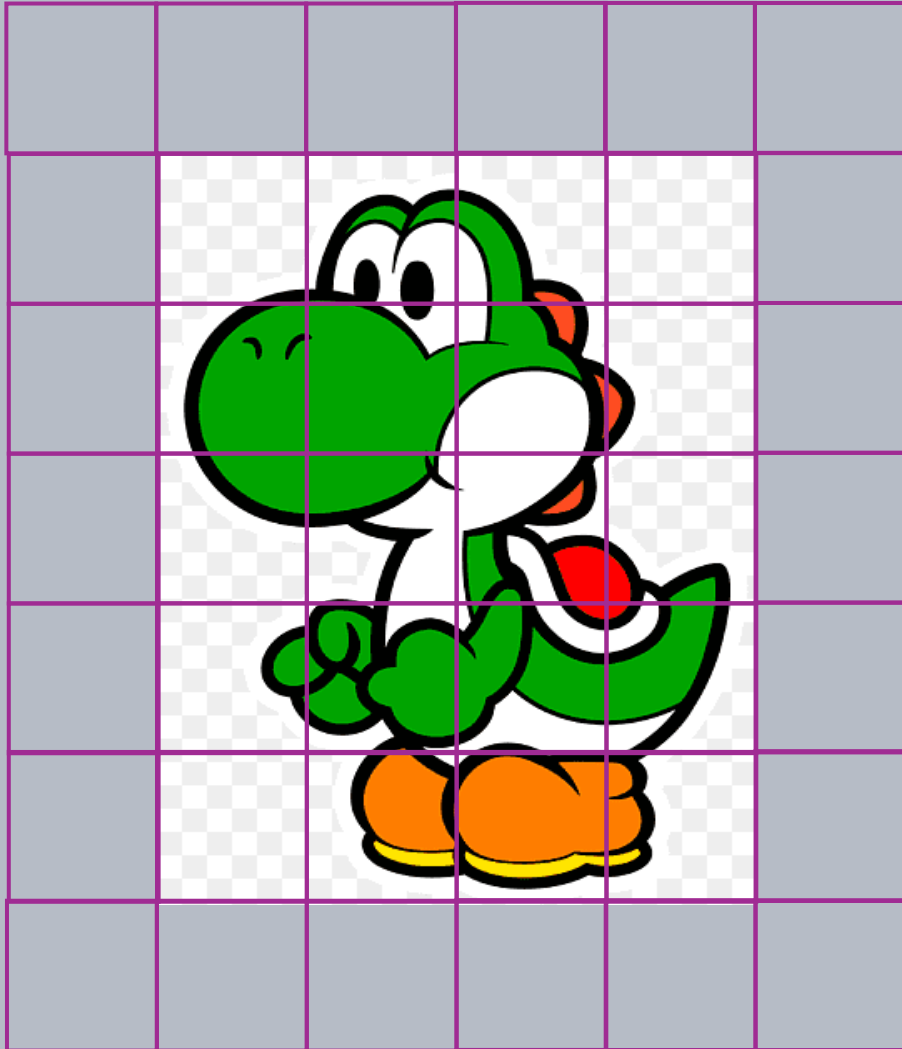
Esmiuçando CNN

Exemplos de kernel



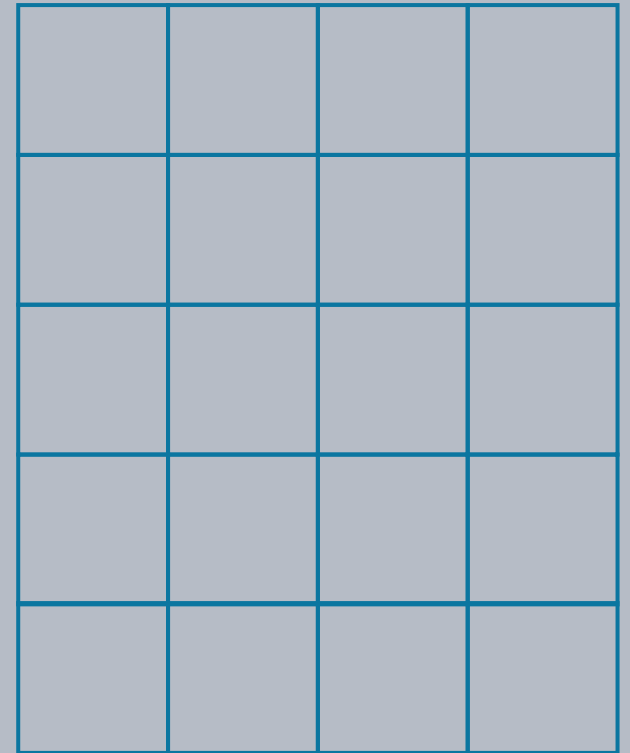
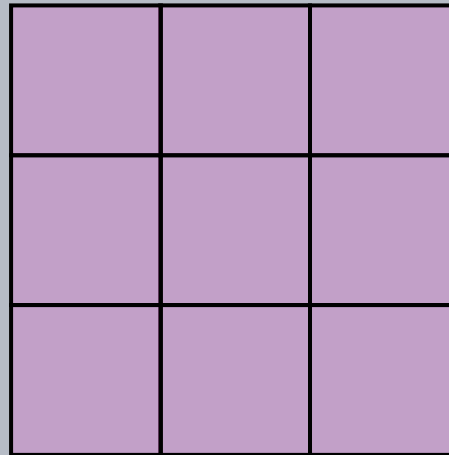
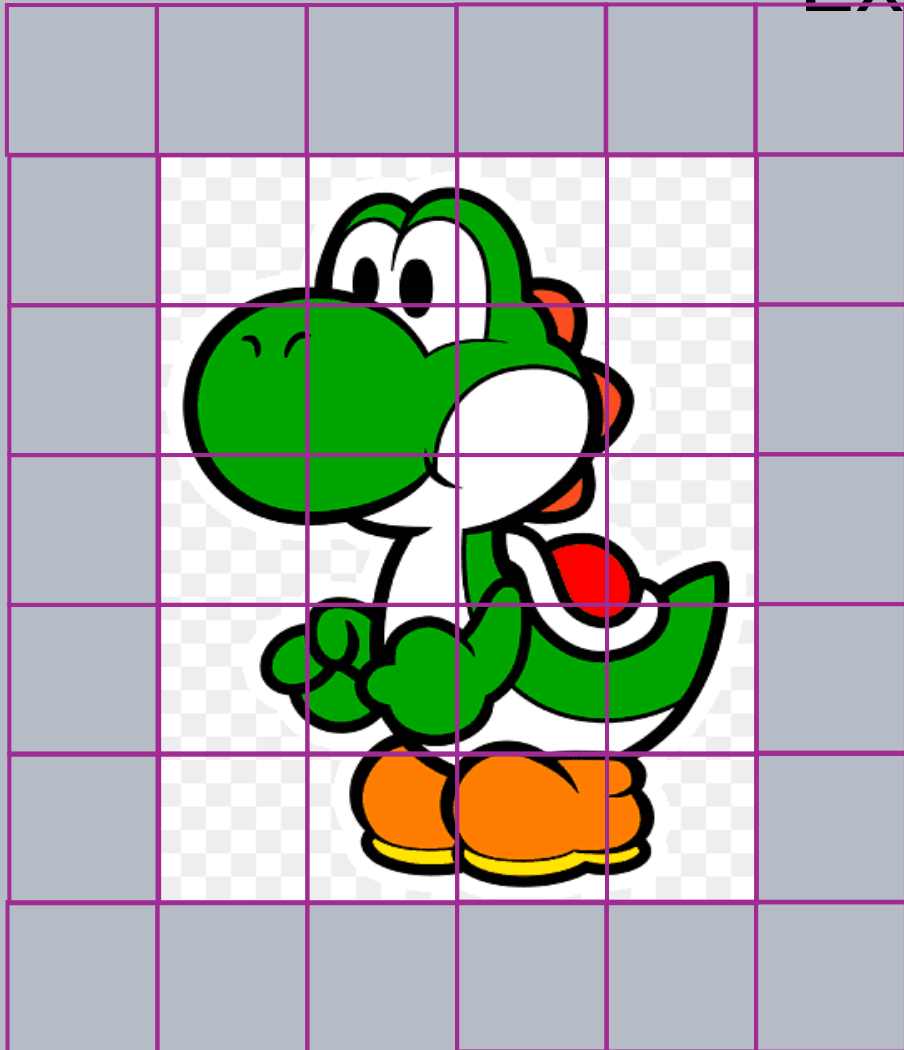
Esmiuçando CNN

Exemplos de kernel



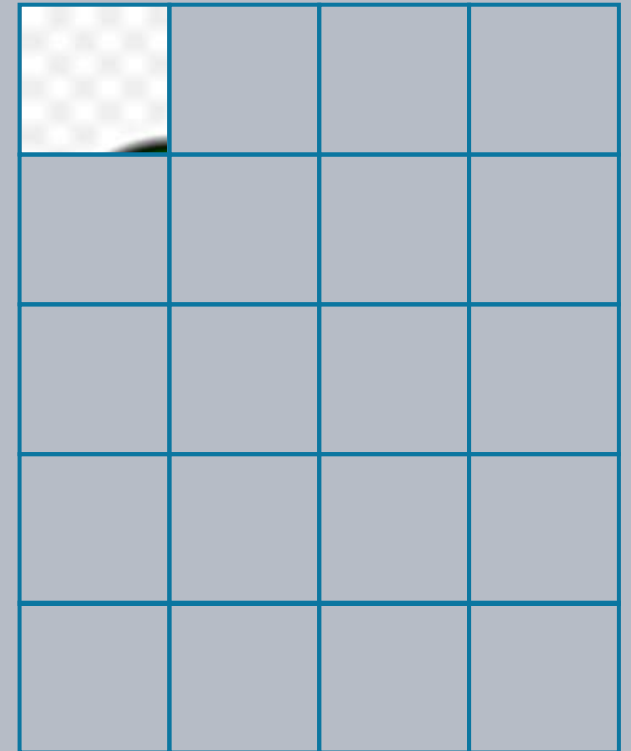
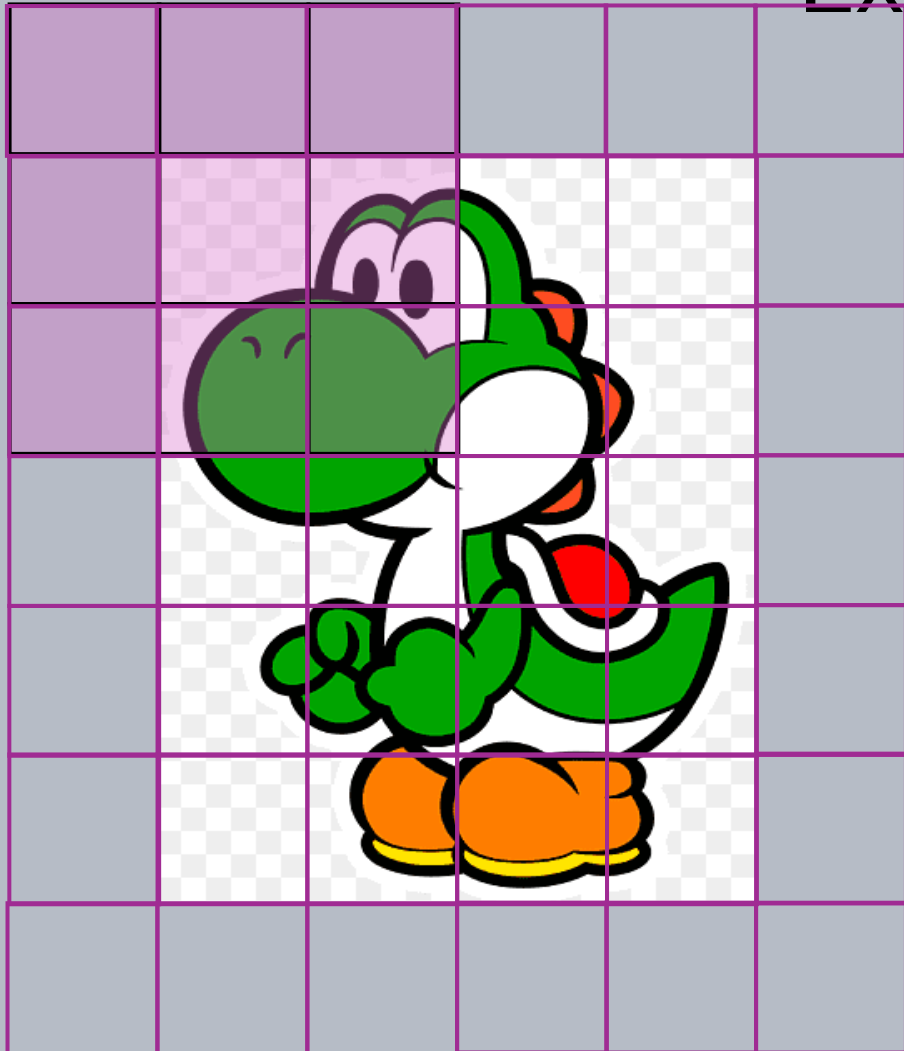
Esmiuçando CNN

Exemplos de kernel



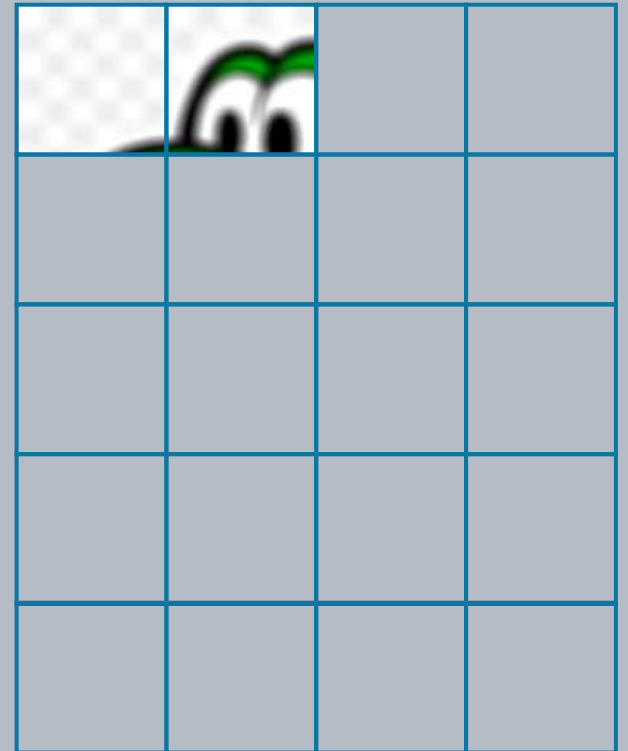
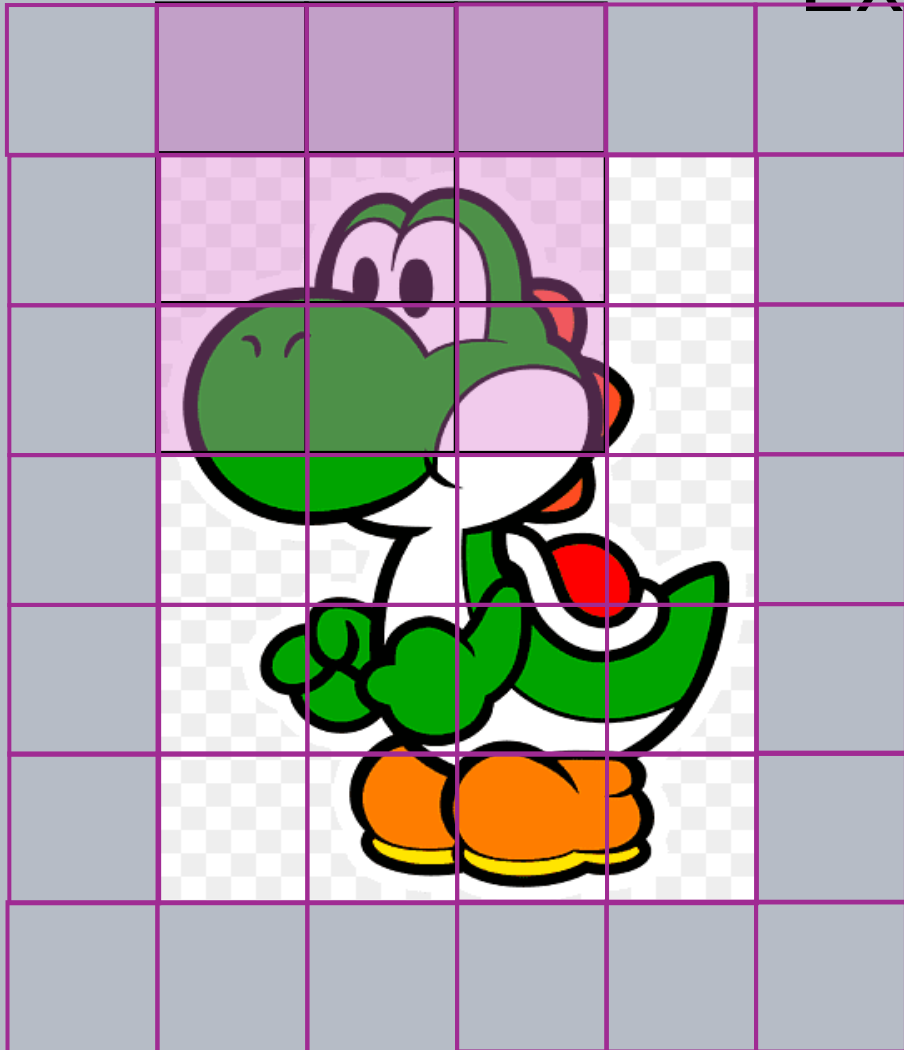
Esmiuçando CNN

Exemplos de kernel



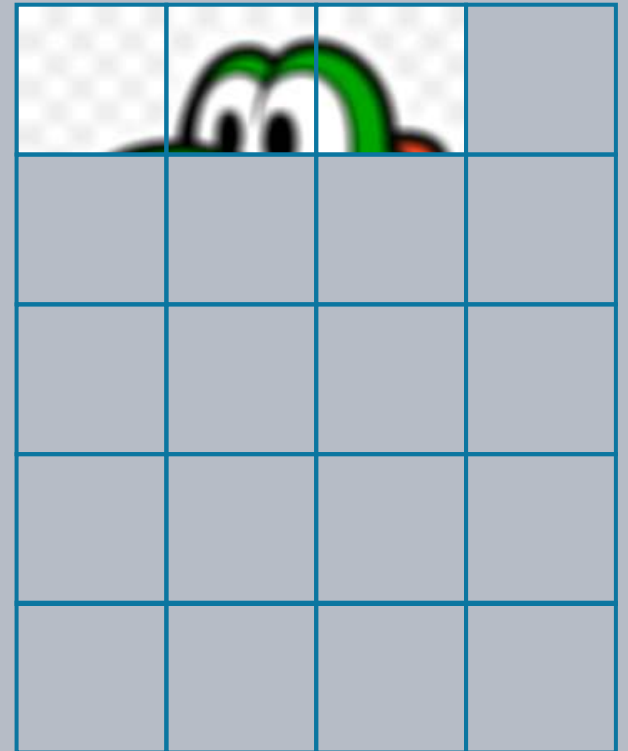
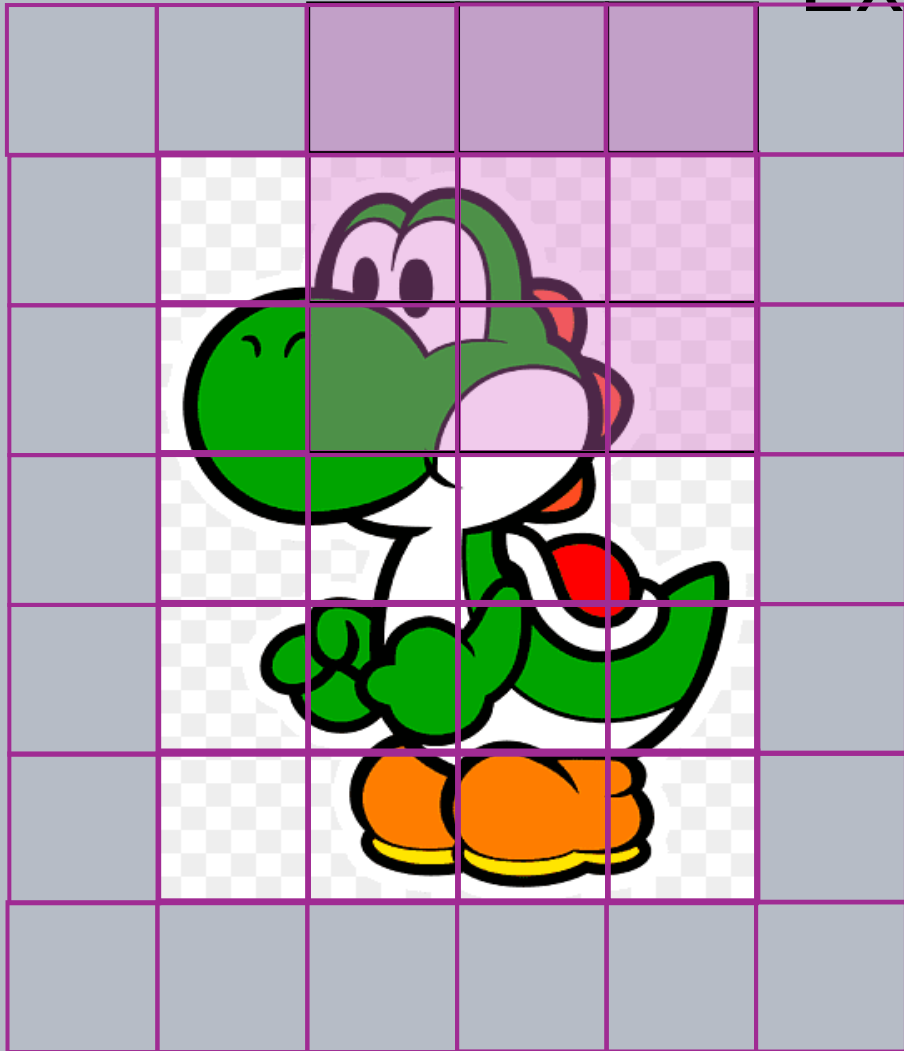
Esmiuçando CNN

Exemplos de kernel



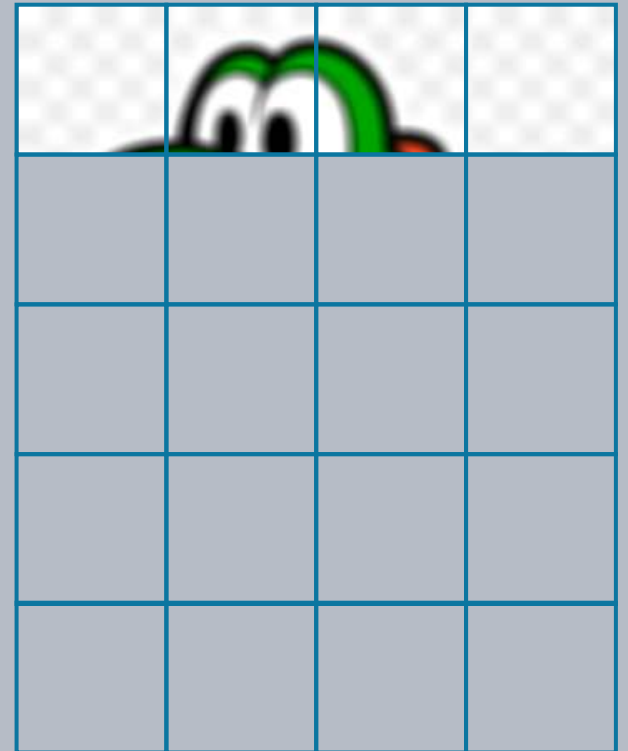
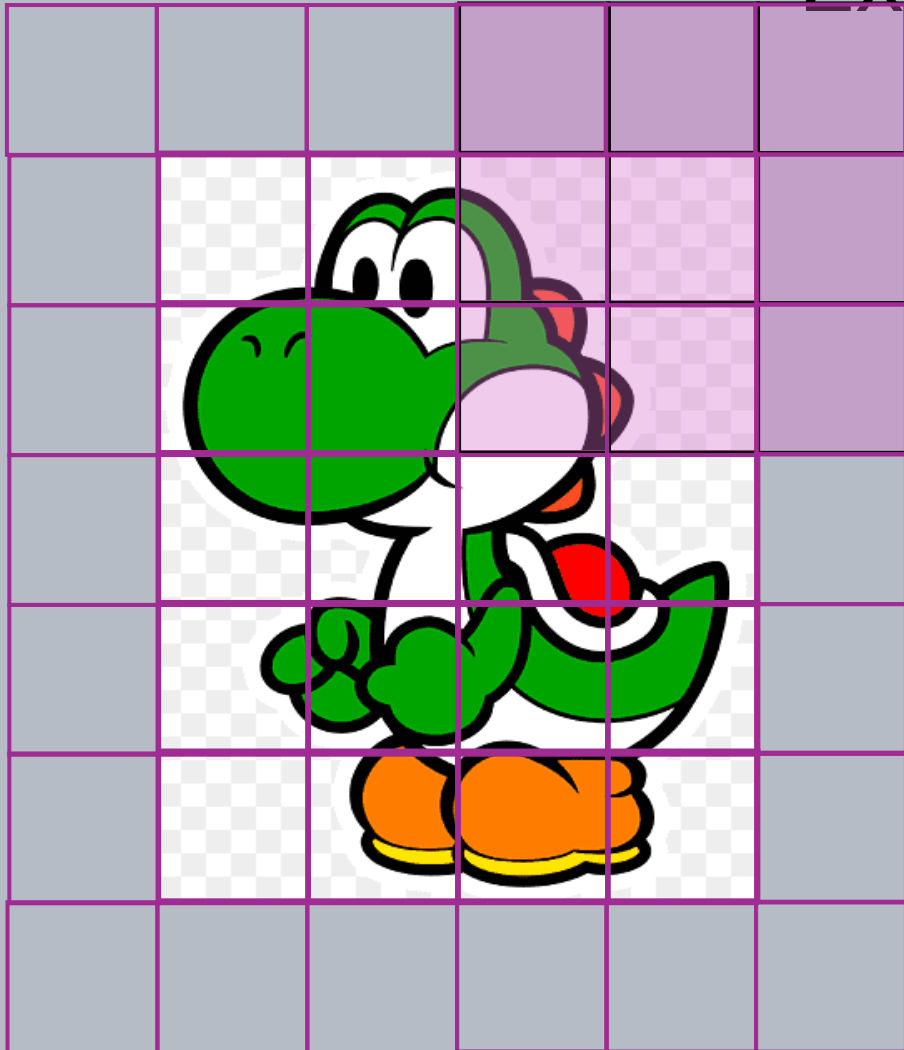
Esmiuçando CNN

Exemplos de kernel



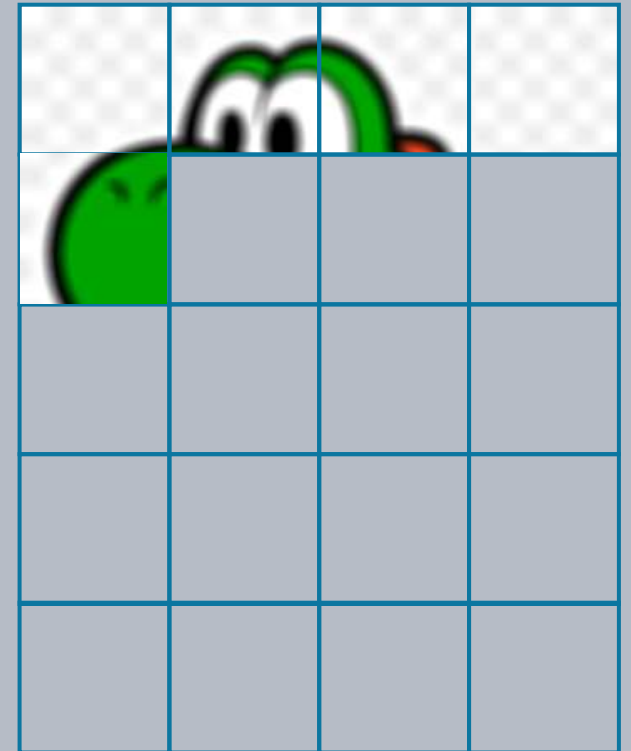
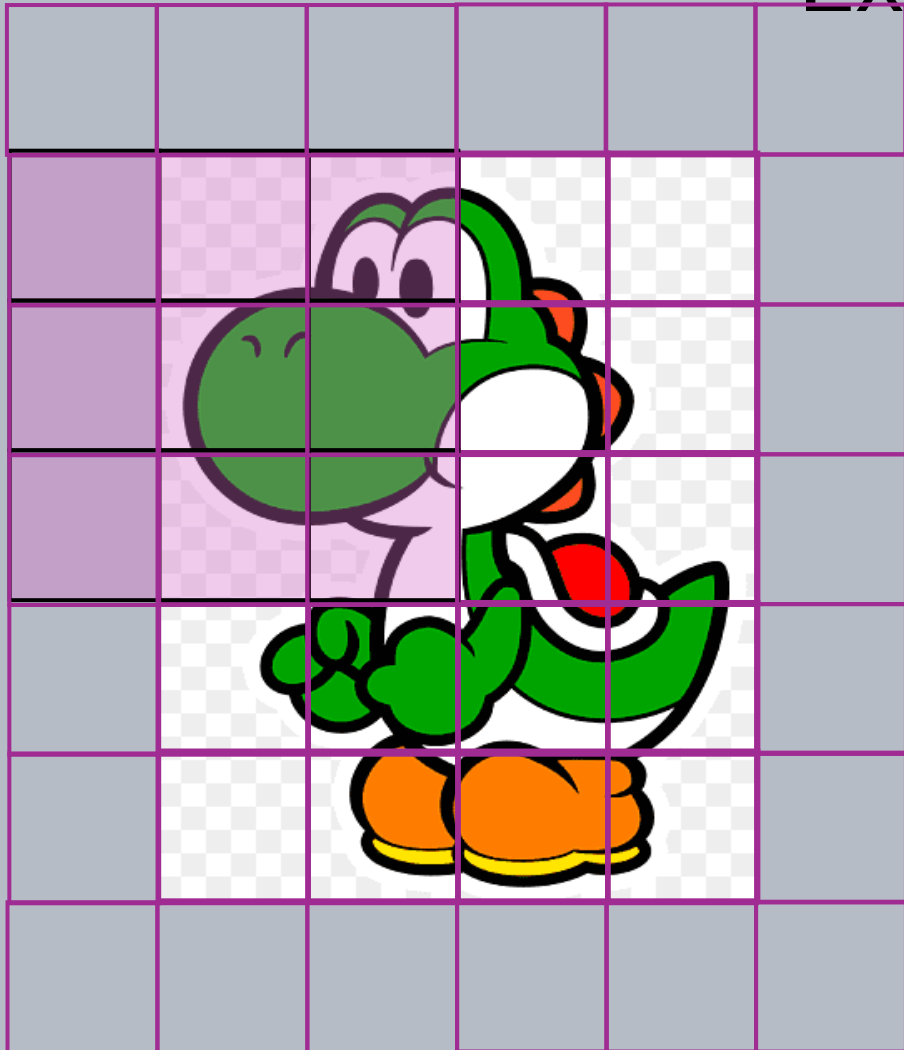
Esmiuçando CNN

Exemplos de kernel



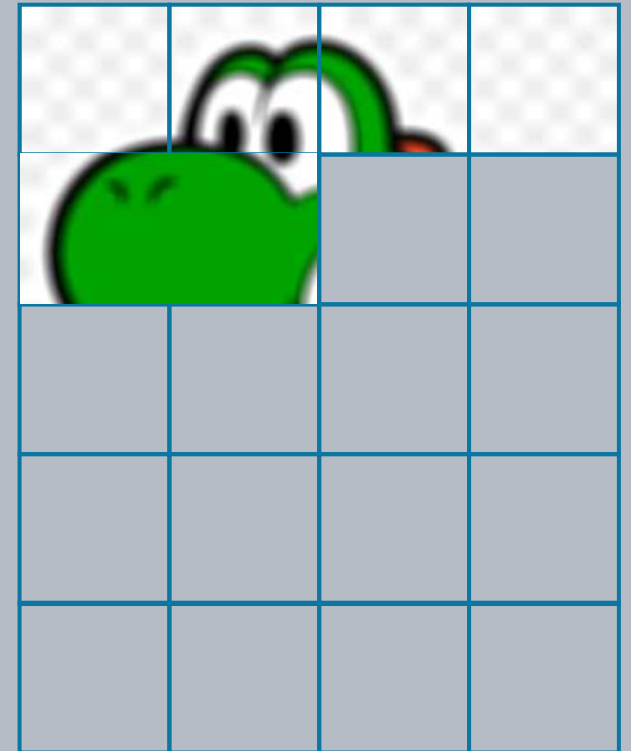
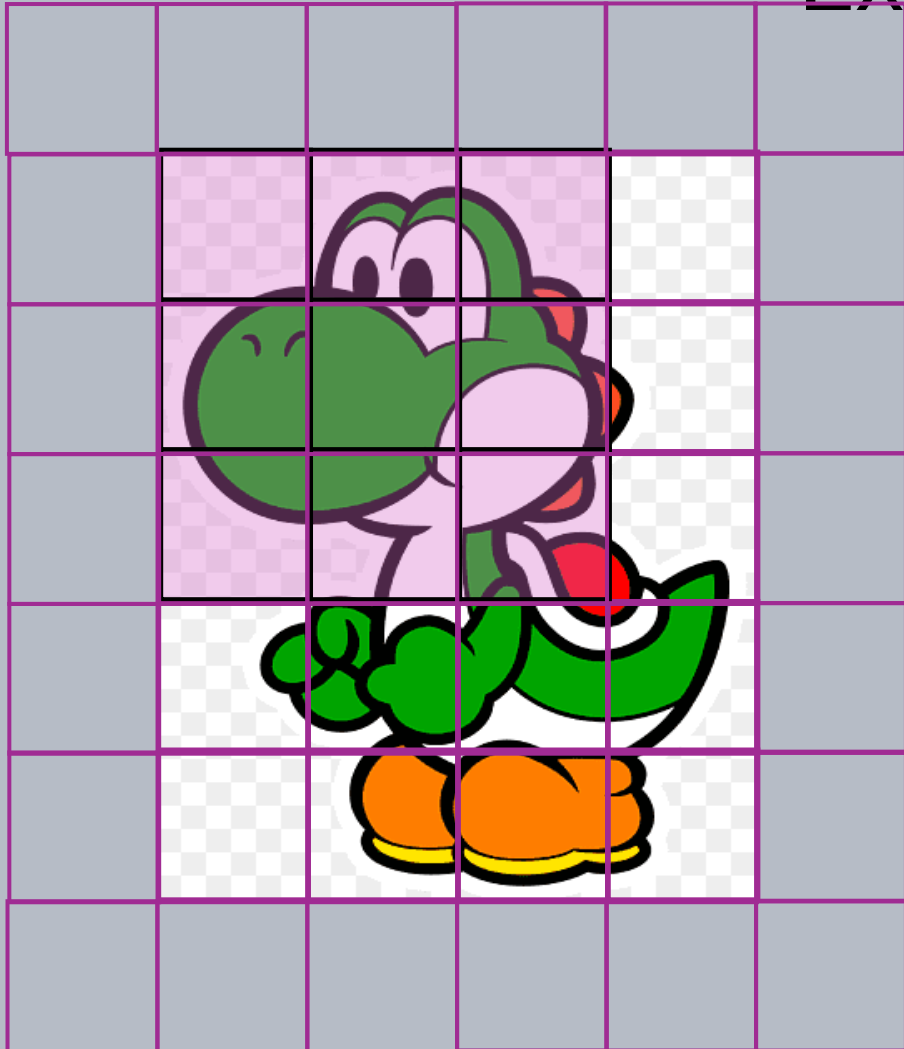
Esmiuçando CNN

Exemplos de kernel



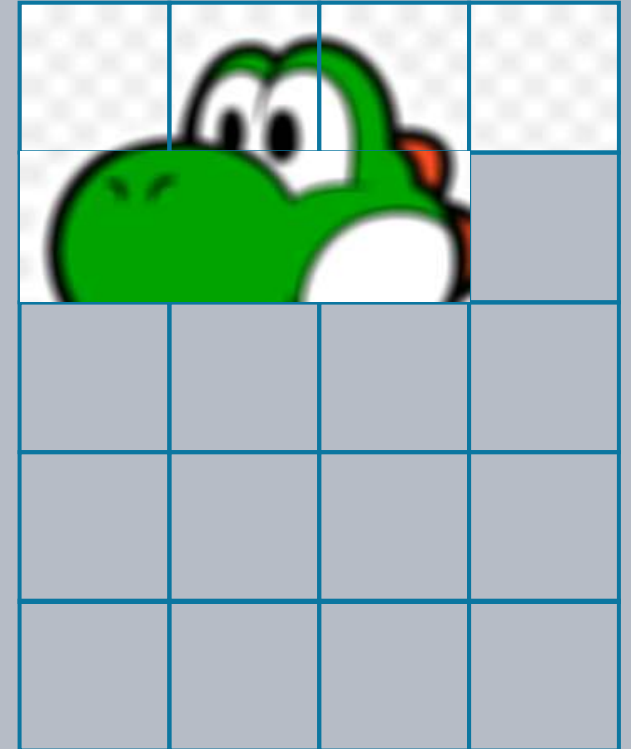
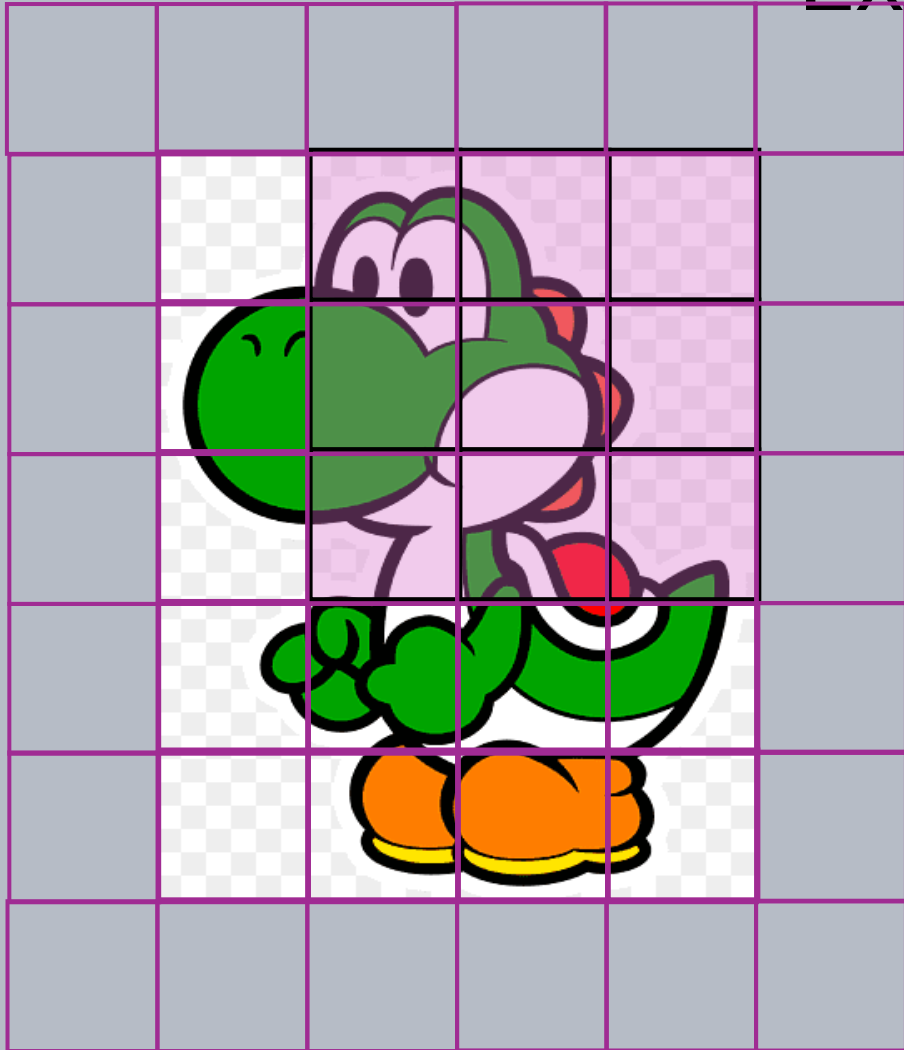
Esmiuçando CNN

Exemplos de kernel



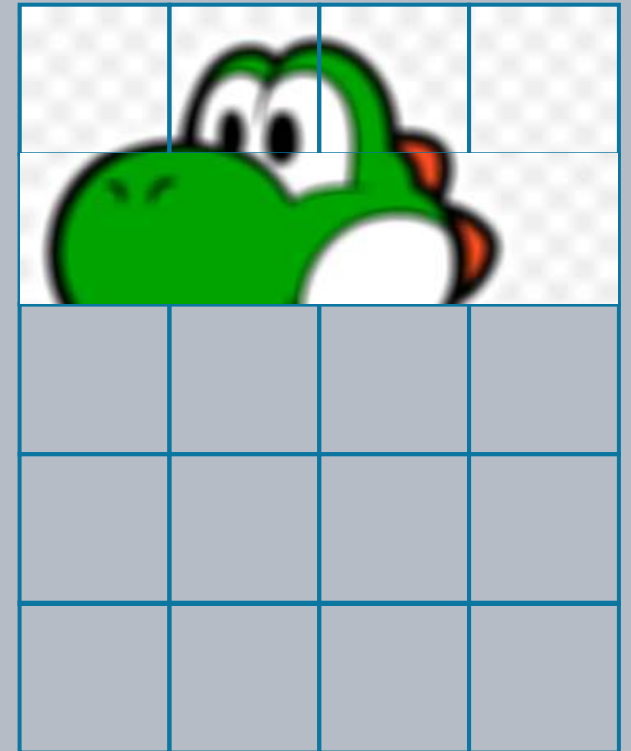
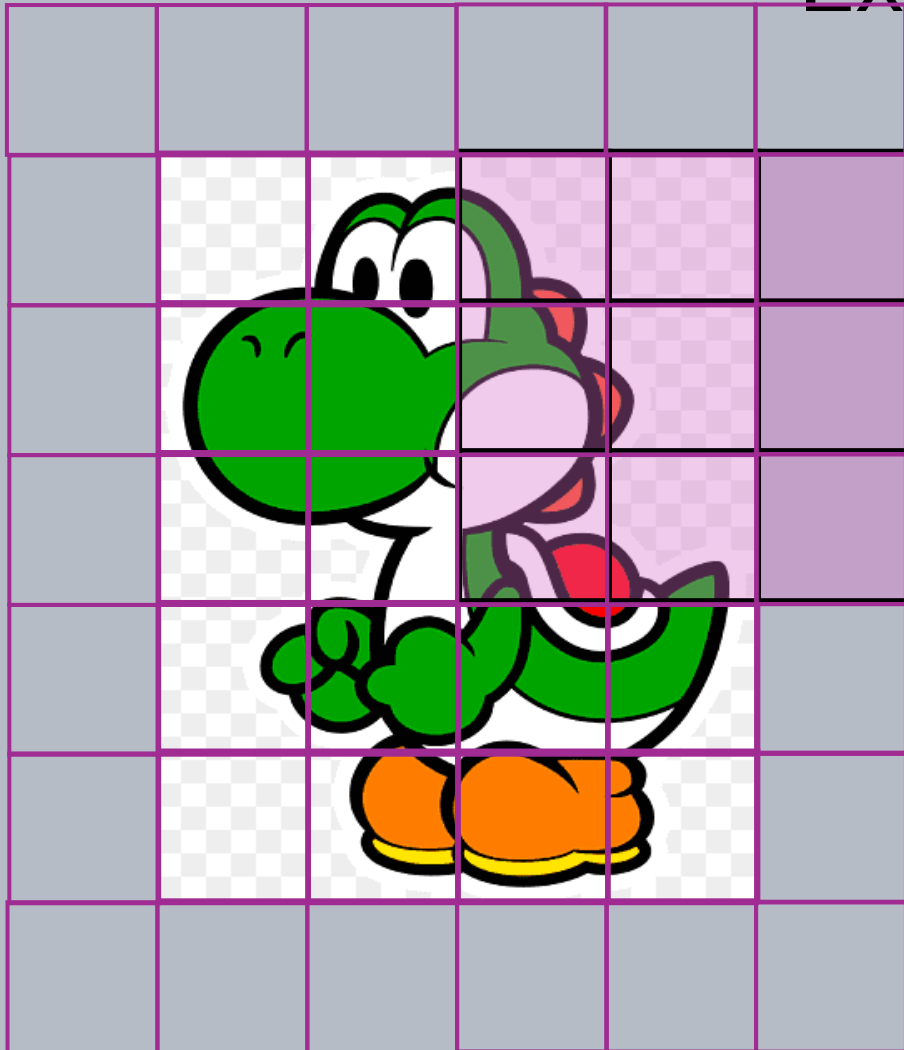
Esmiuçando CNN

Exemplos de kernel



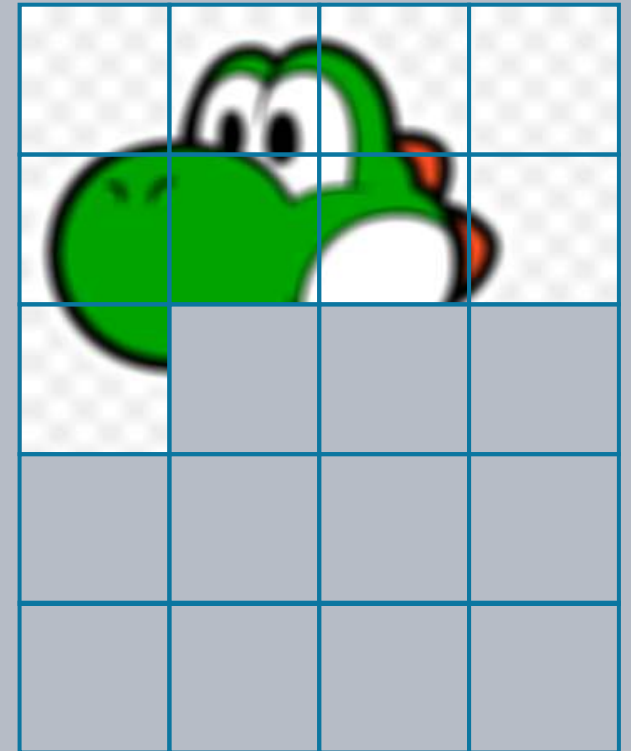
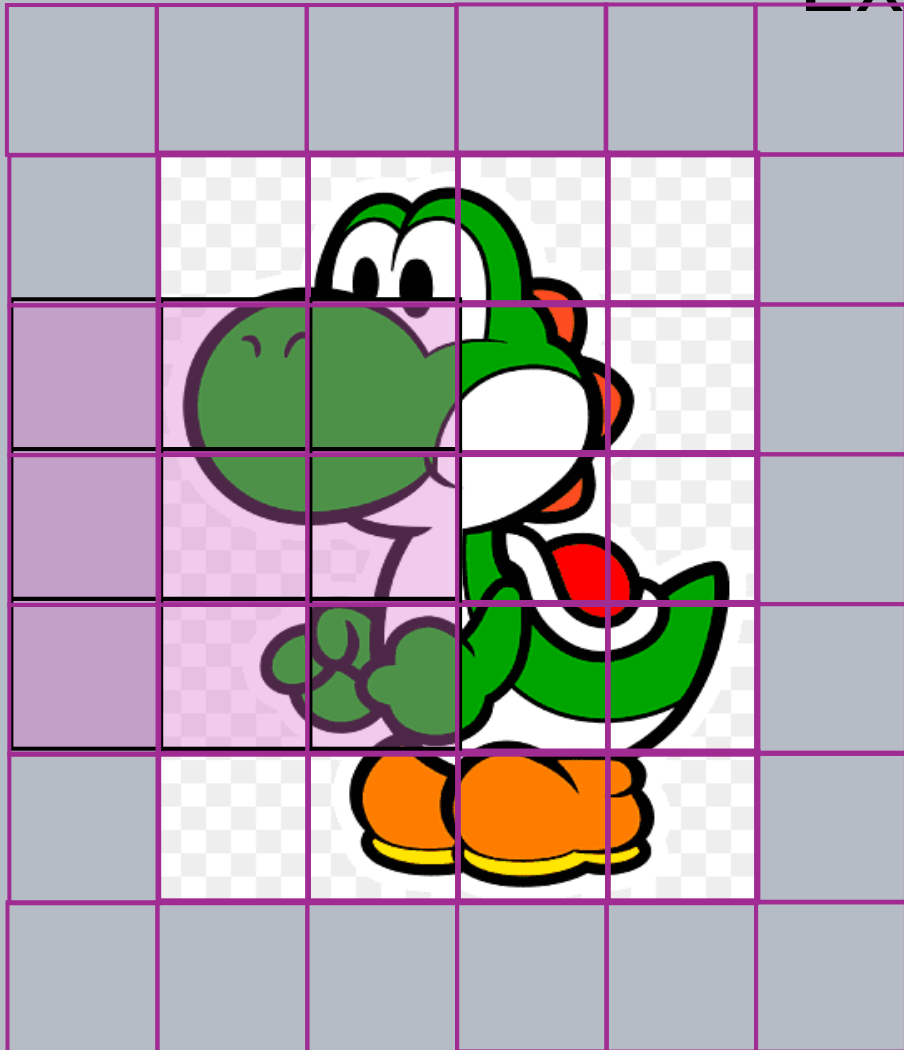
Esmiuçando CNN

Exemplos de kernel



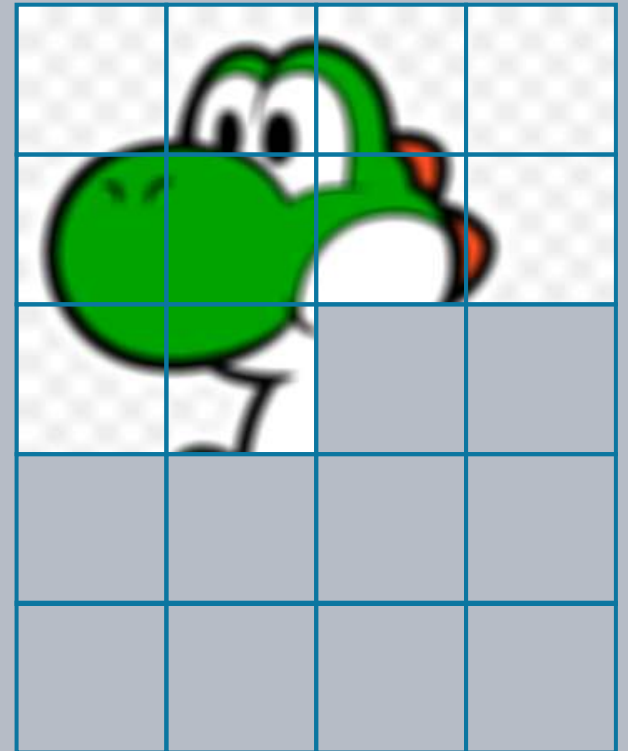
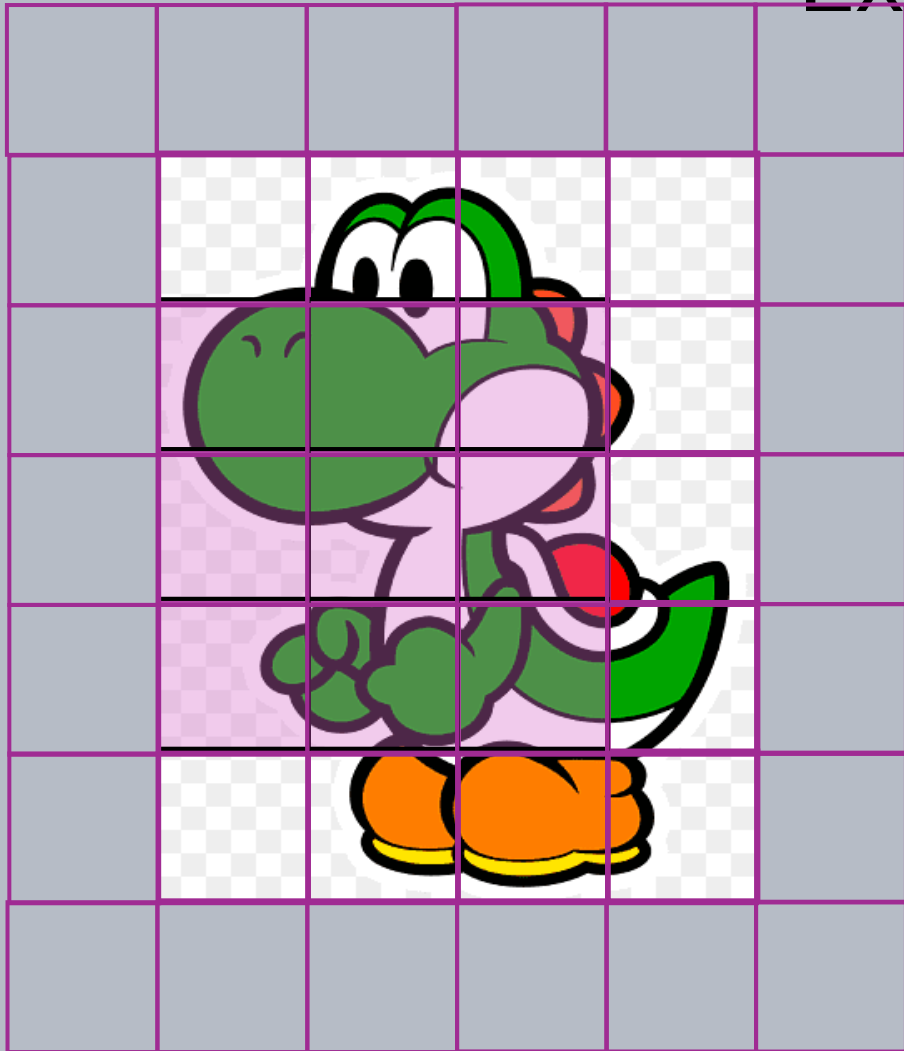
Esmiuçando CNN

Exemplos de kernel



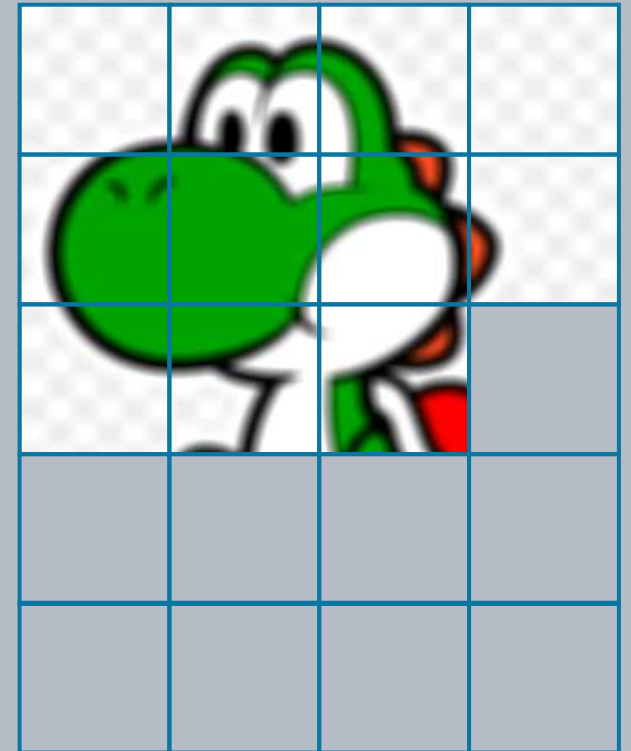
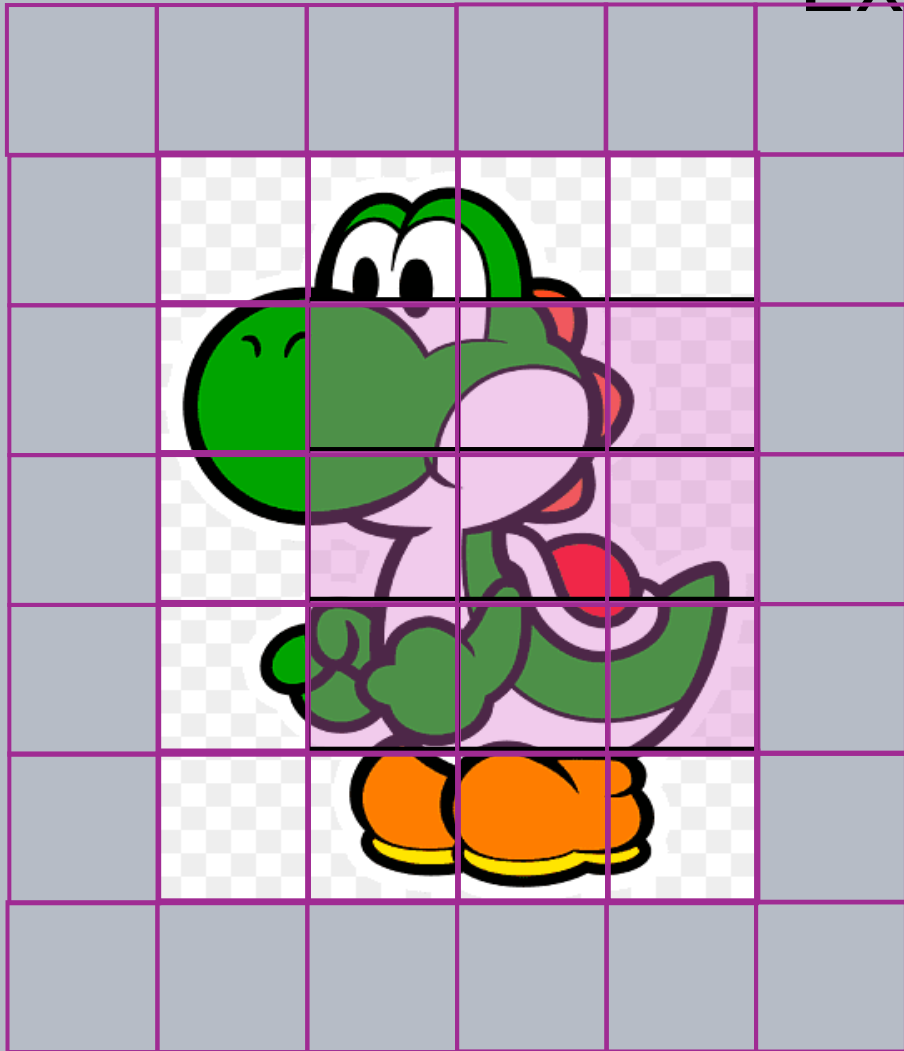
Esmiuçando CNN

Exemplos de kernel



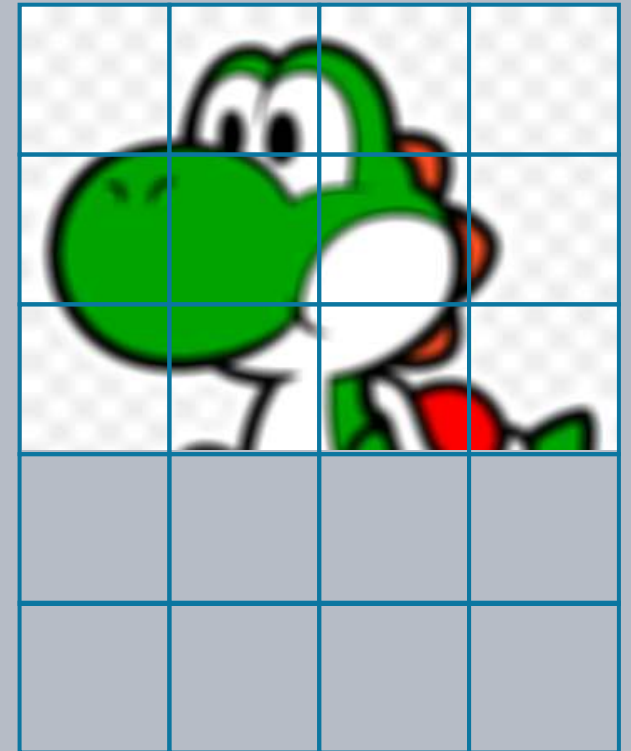
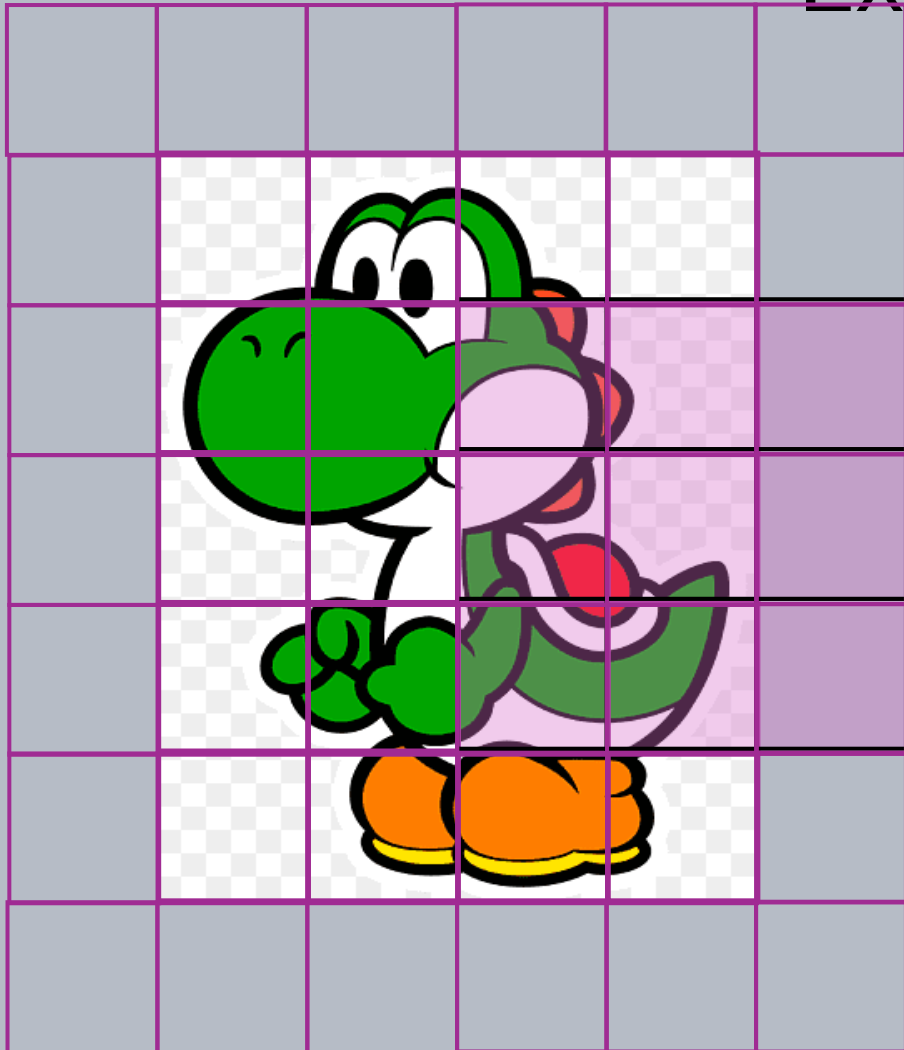
Esmiuçando CNN

Exemplos de kernel



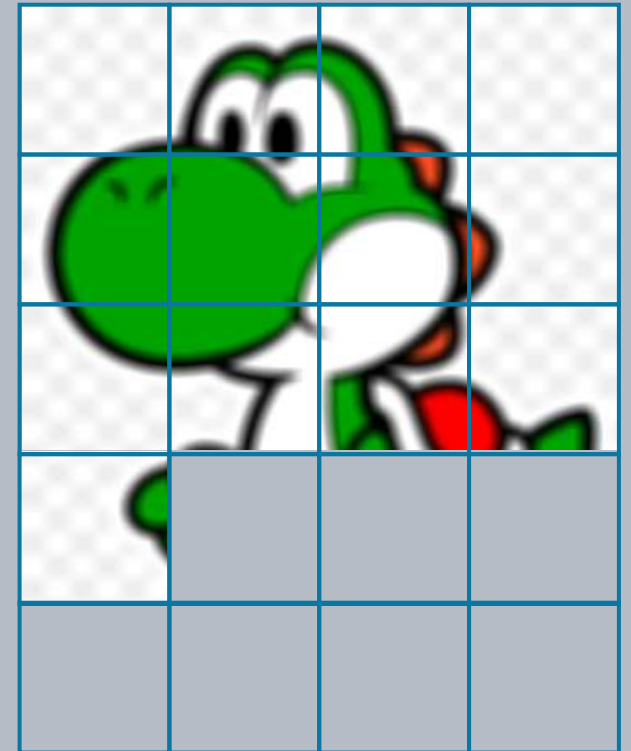
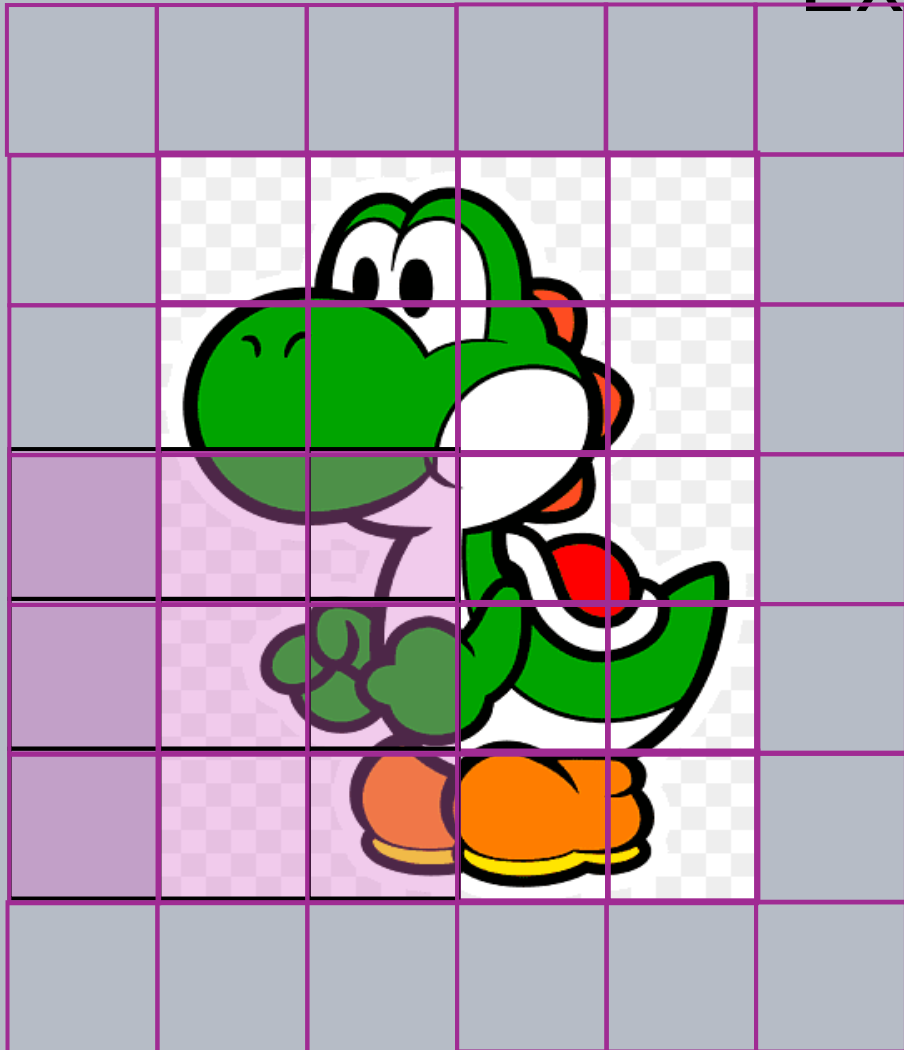
Esmiuçando CNN

Exemplos de kernel



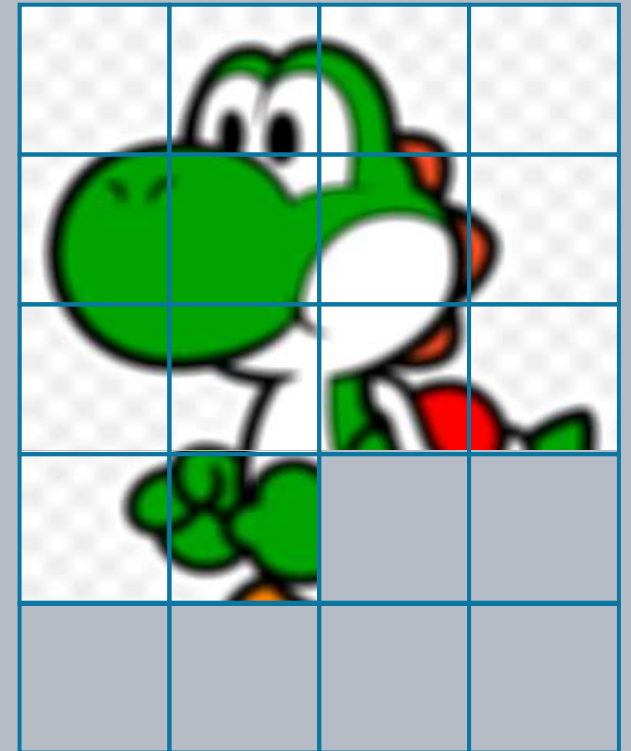
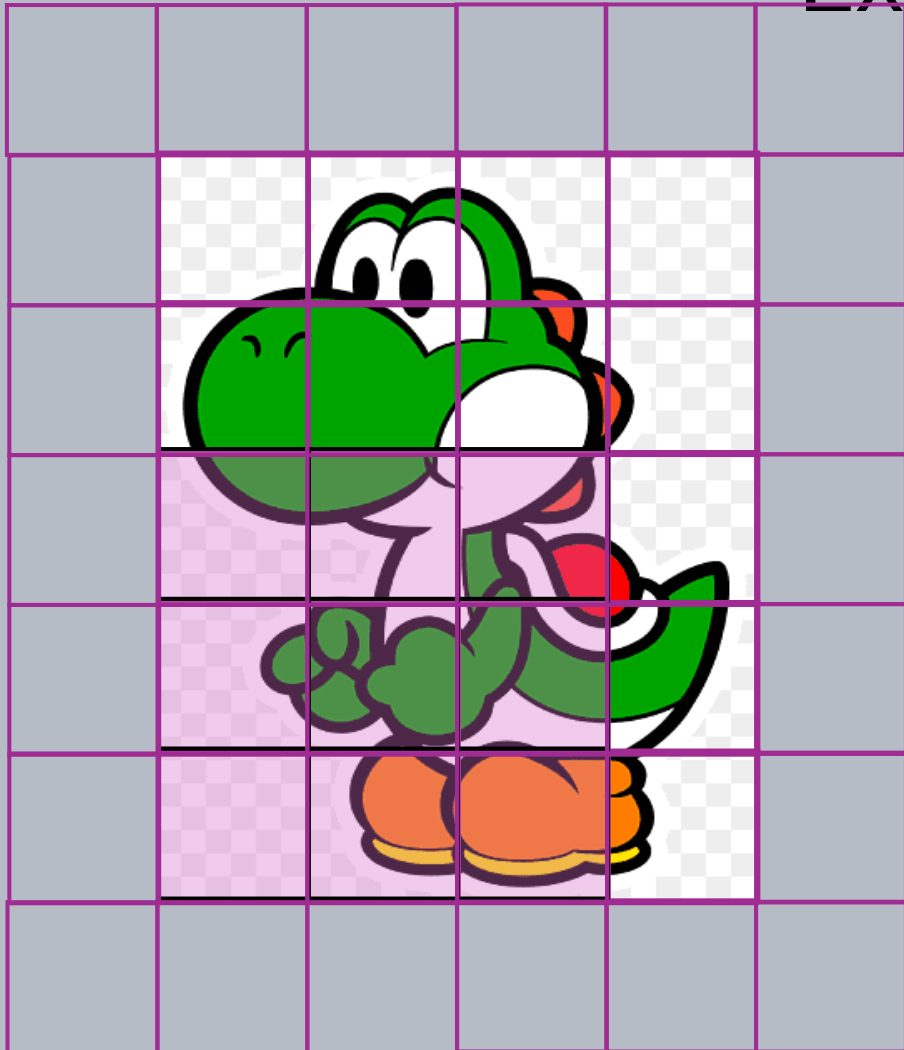
Esmiuçando CNN

Exemplos de kernel



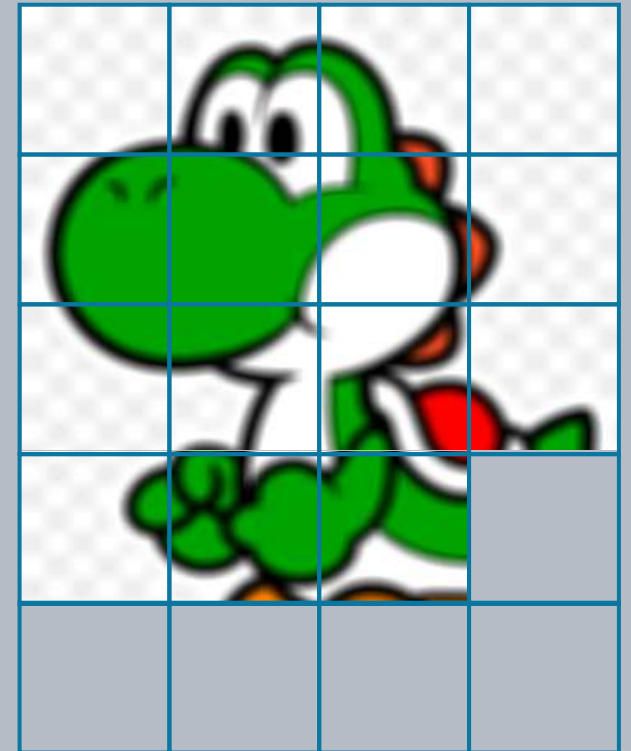
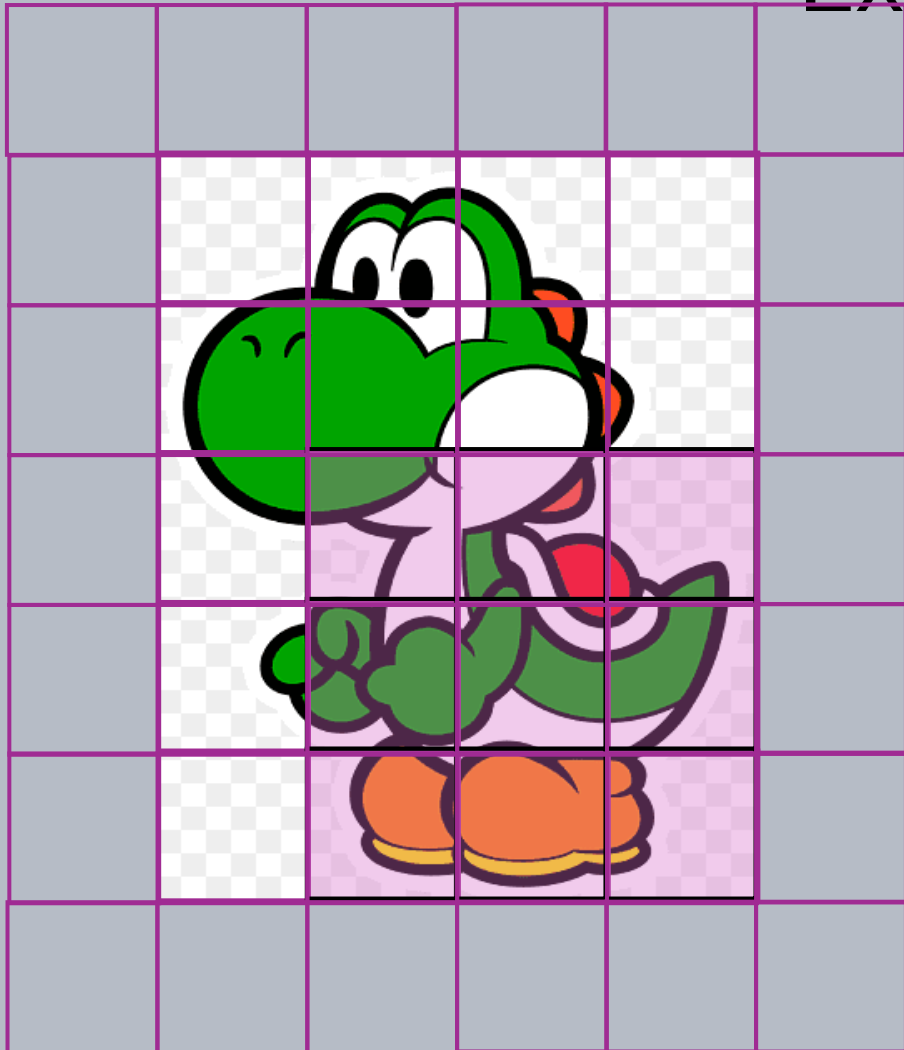
Esmiuçando CNN

Exemplos de kernel



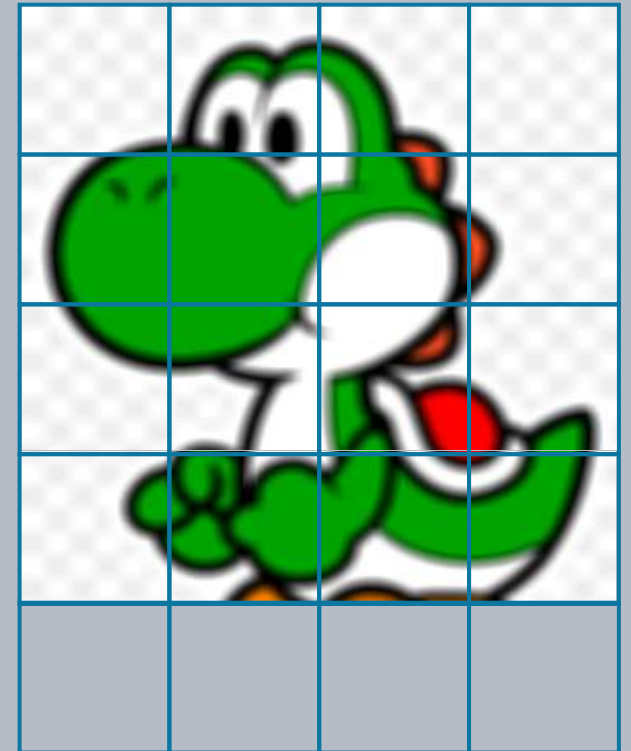
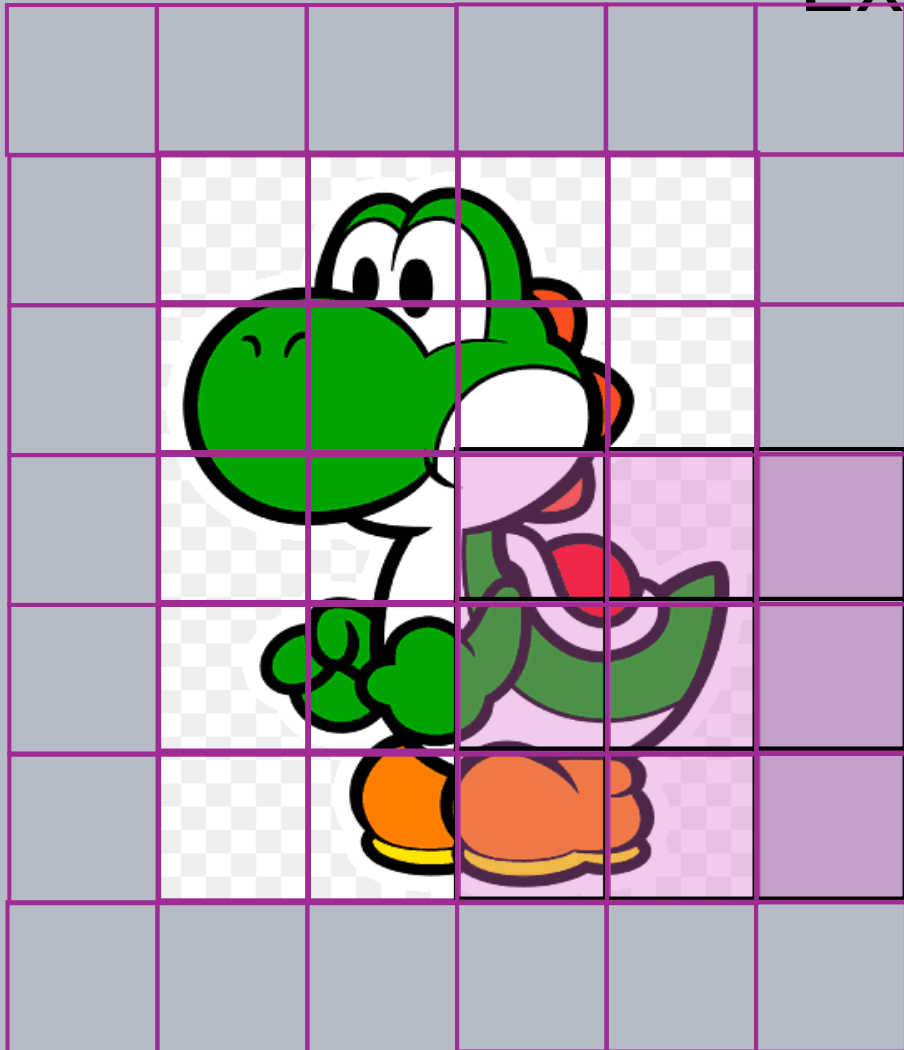
Esmiuçando CNN

Exemplos de kernel



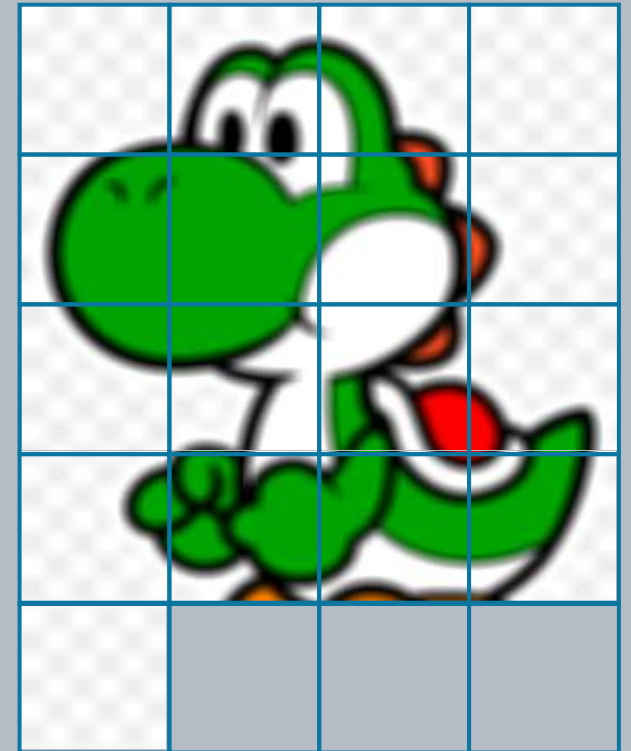
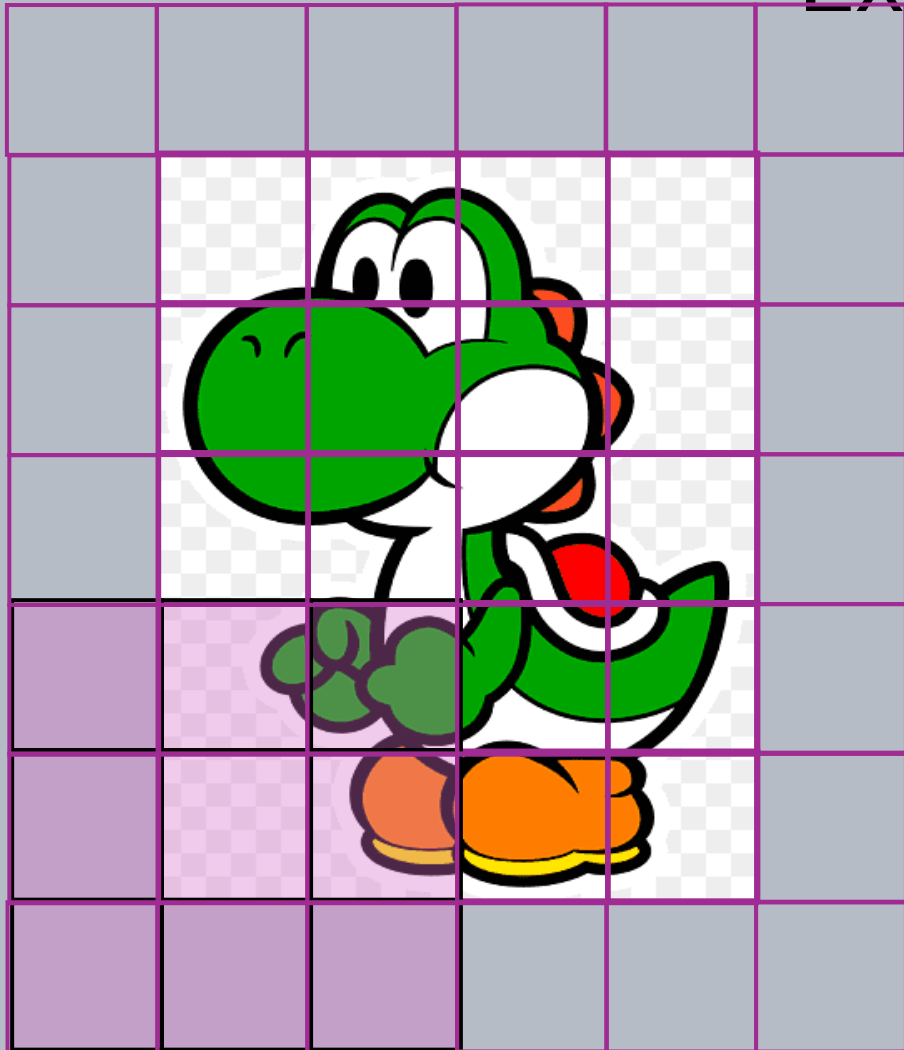
Esmiuçando CNN

Exemplos de kernel



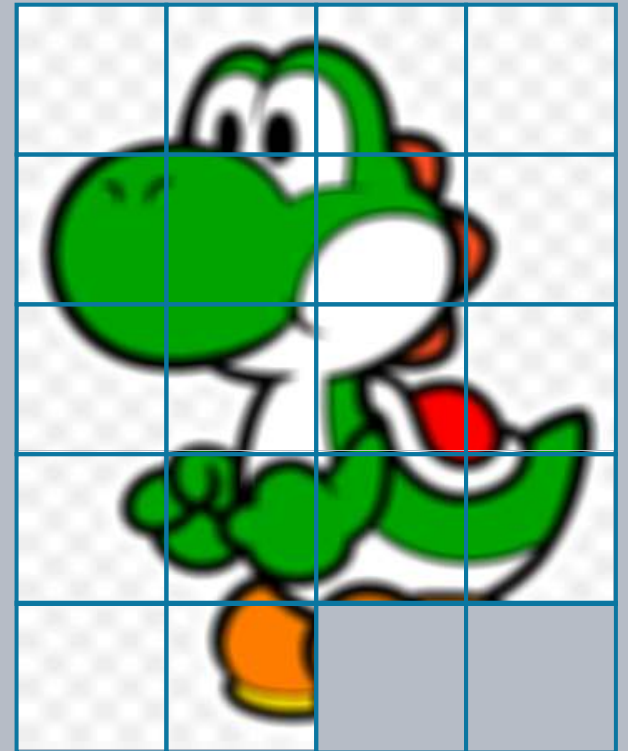
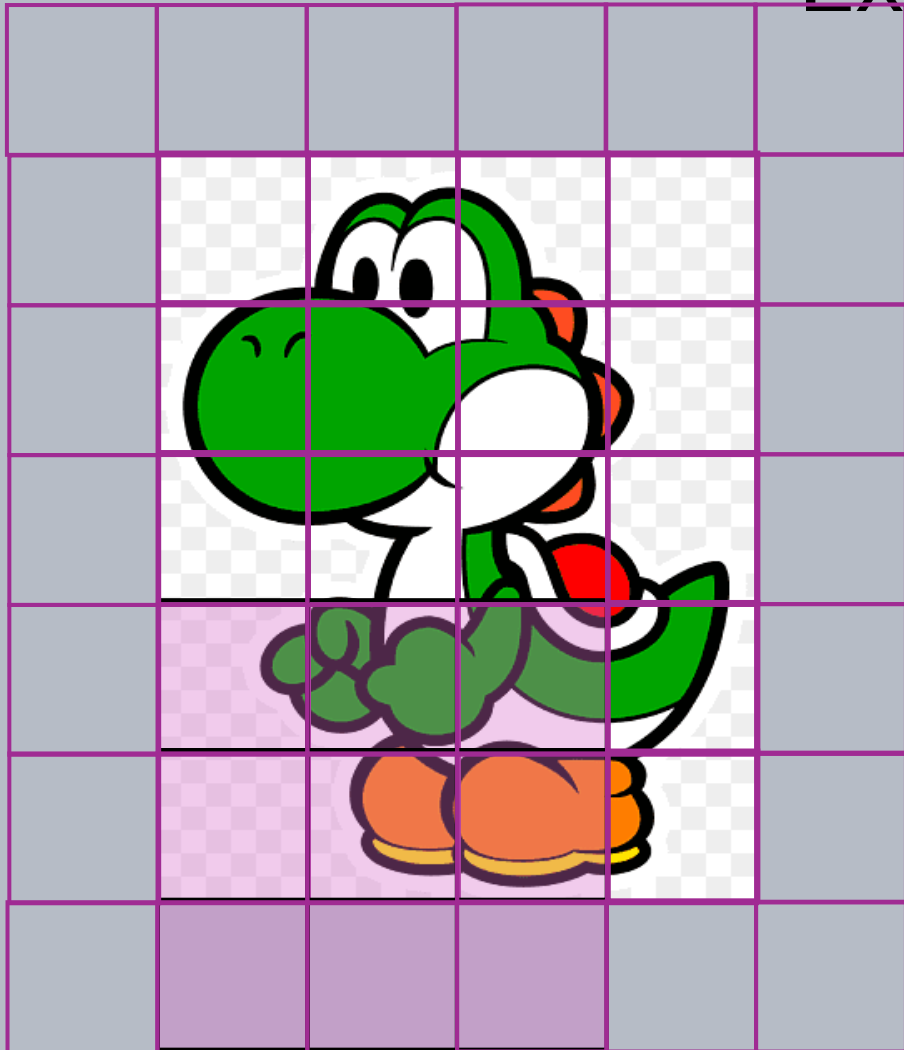
Esmiuçando CNN

Exemplos de kernel



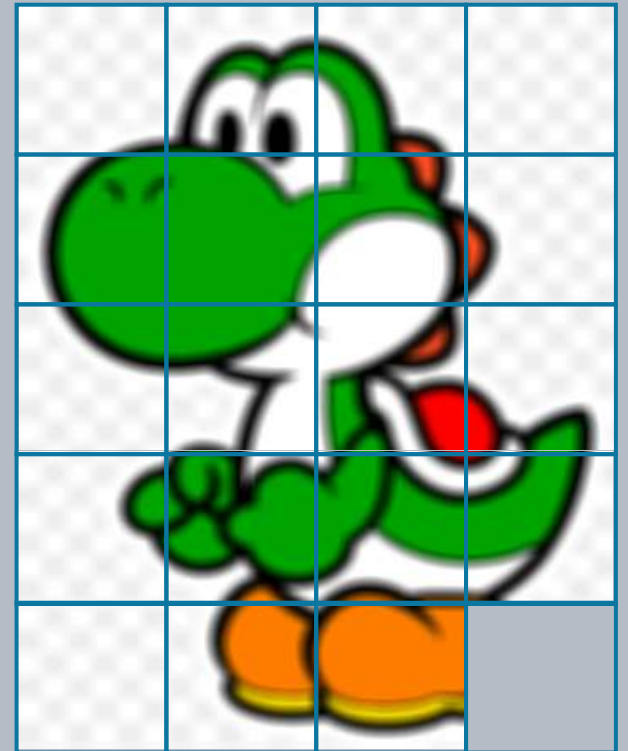
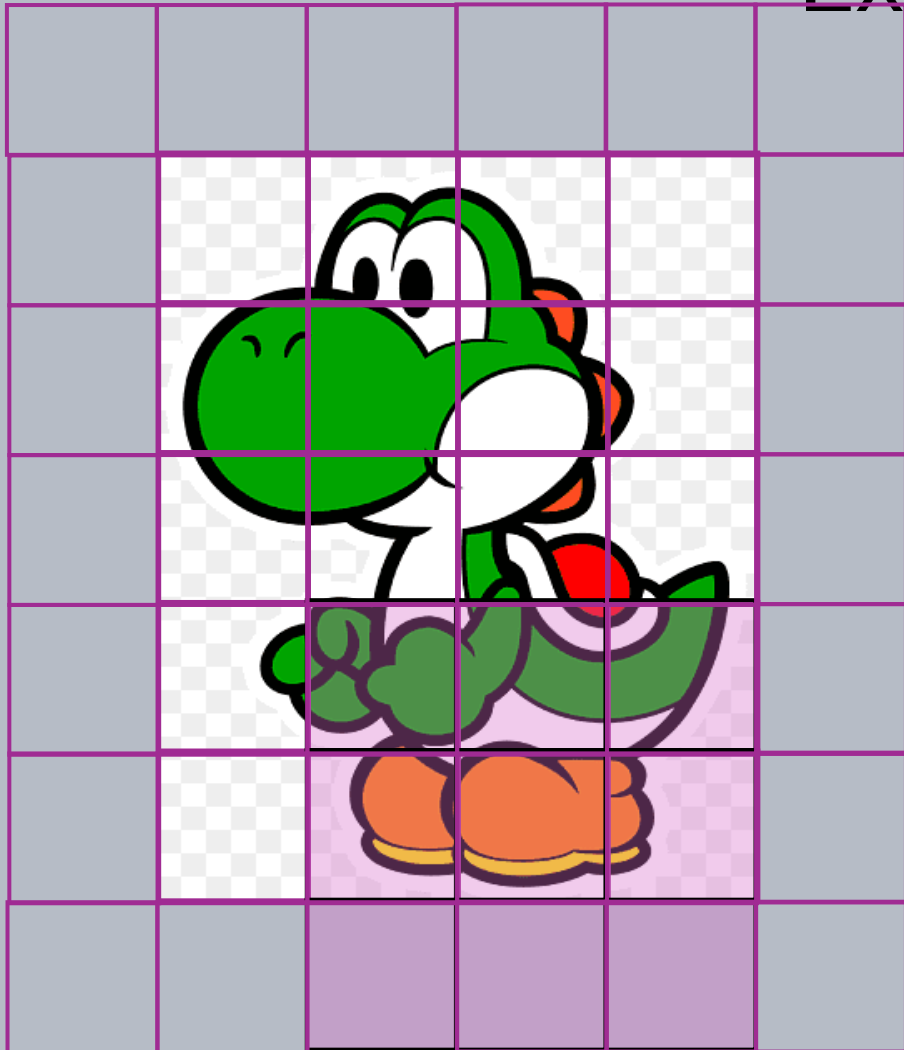
Esmiuçando CNN

Exemplos de kernel



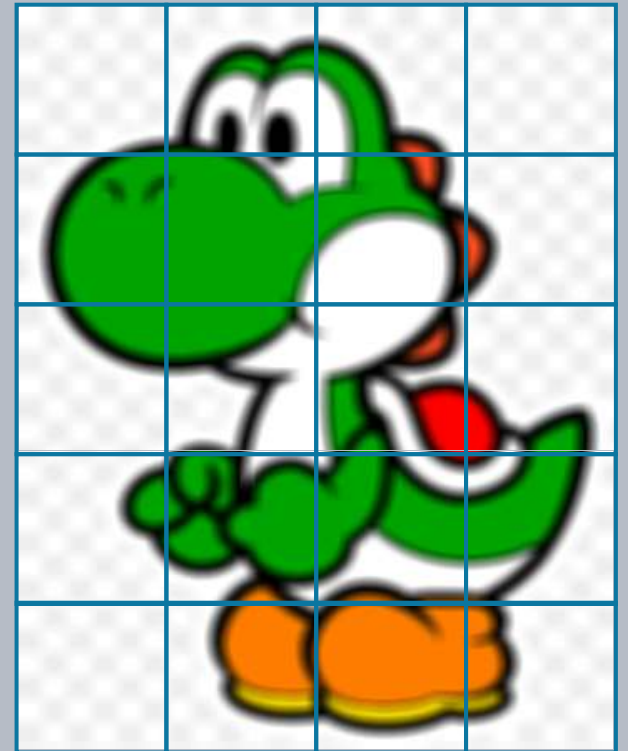
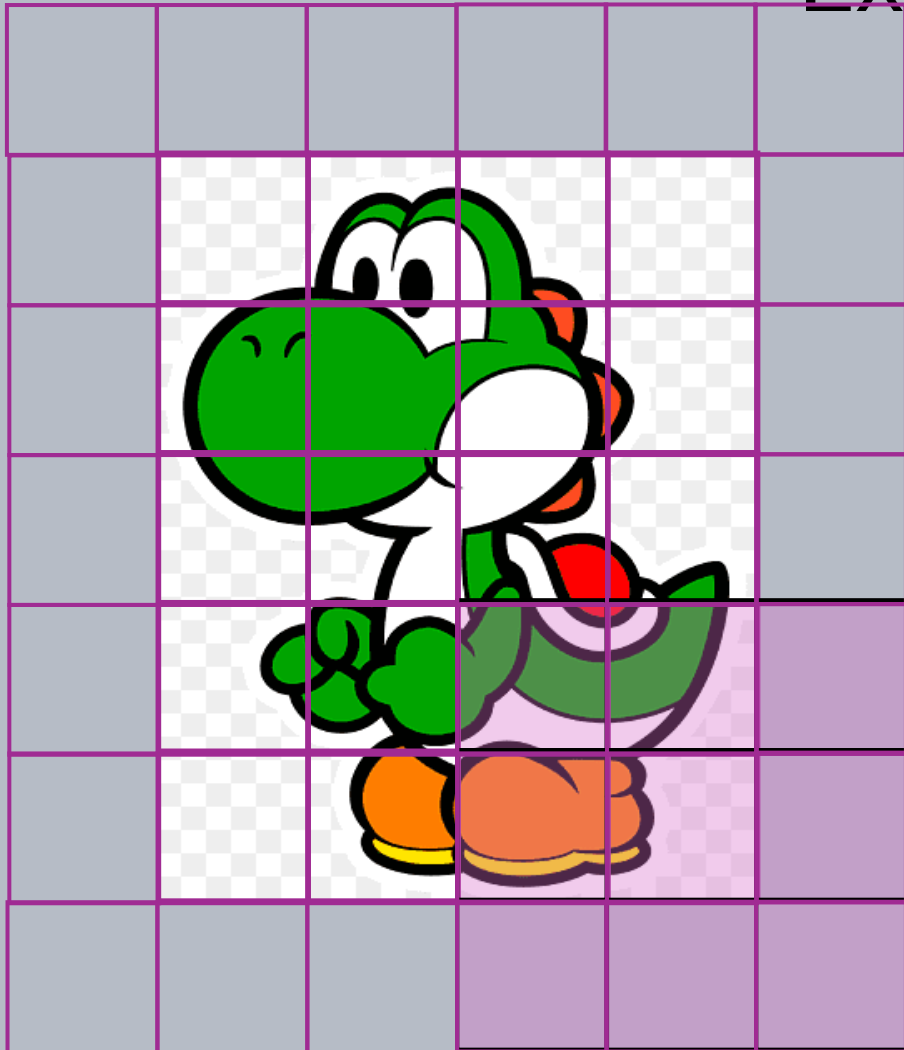
Esmiuçando CNN

Exemplos de kernel



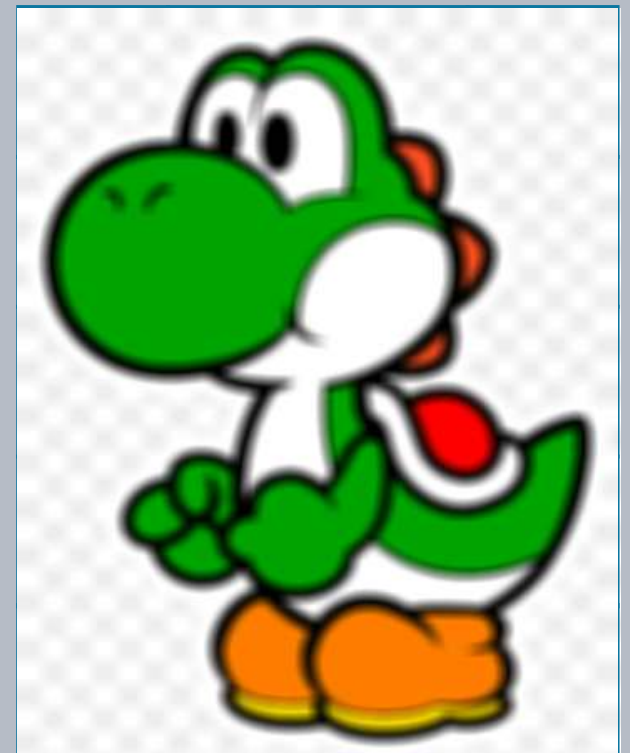
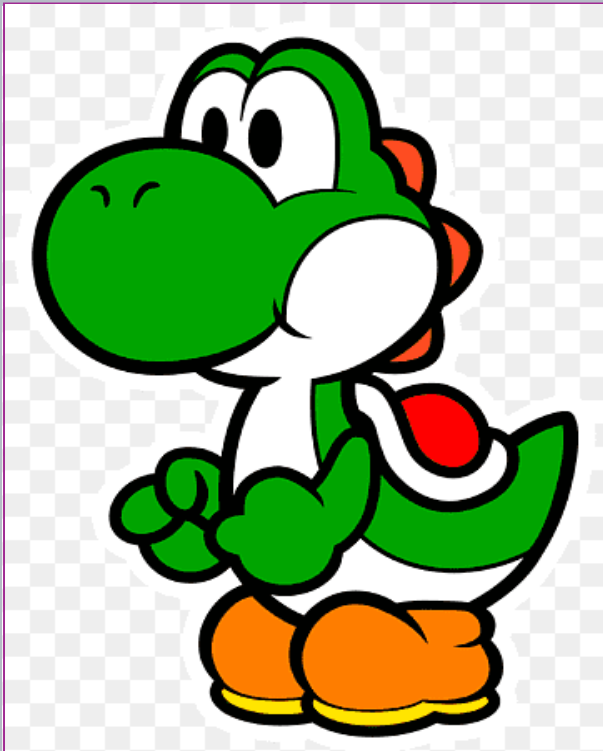
Esmiuçando CNN

Exemplos de kernel



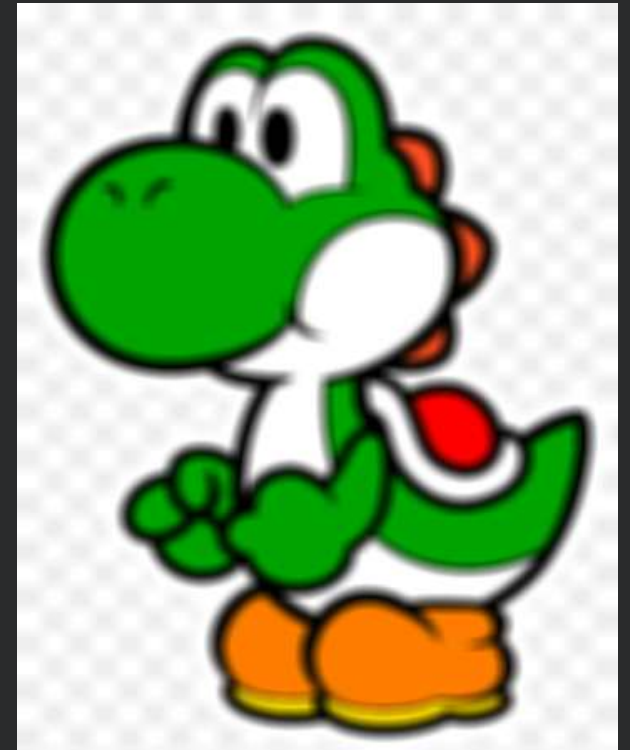
Esmiuçando CNN

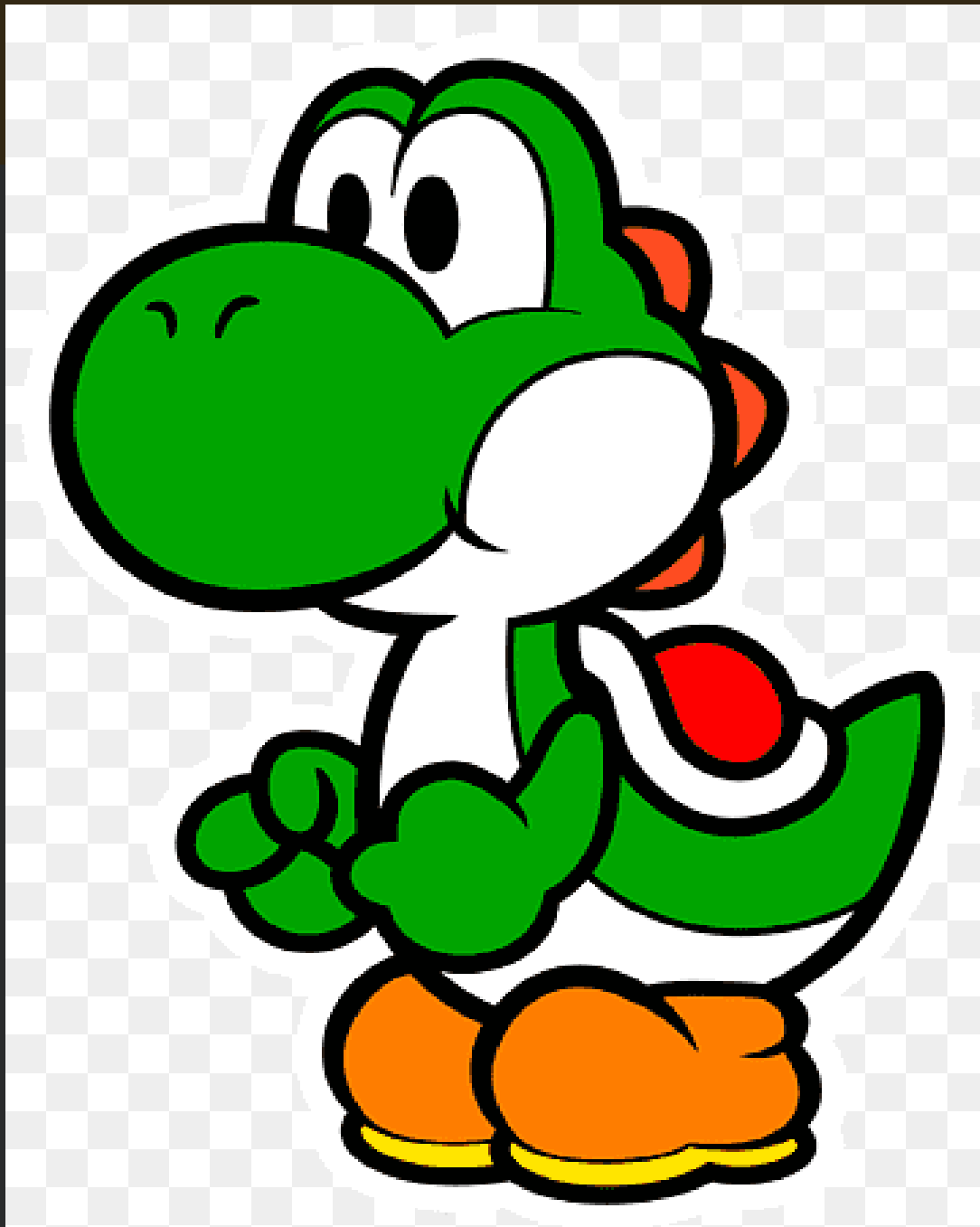
Exemplos de kernel



Esmiuçando CNN

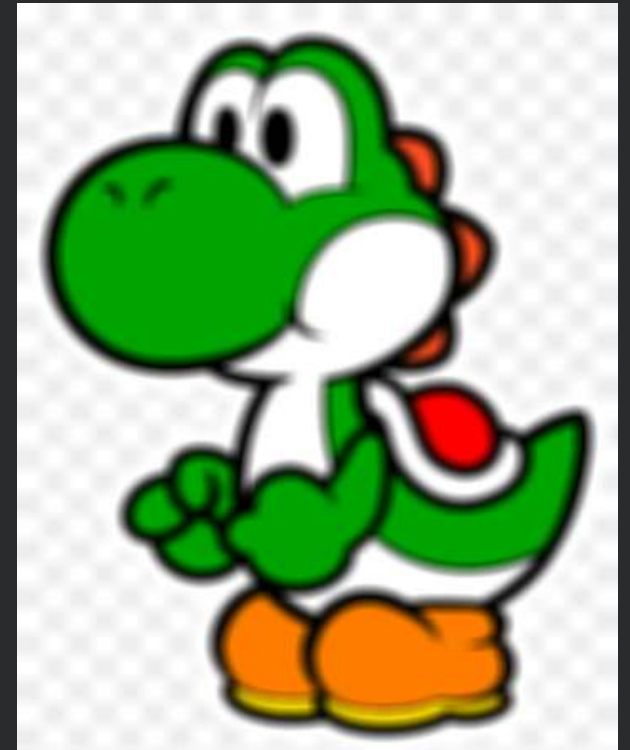
Exemplos de kernel





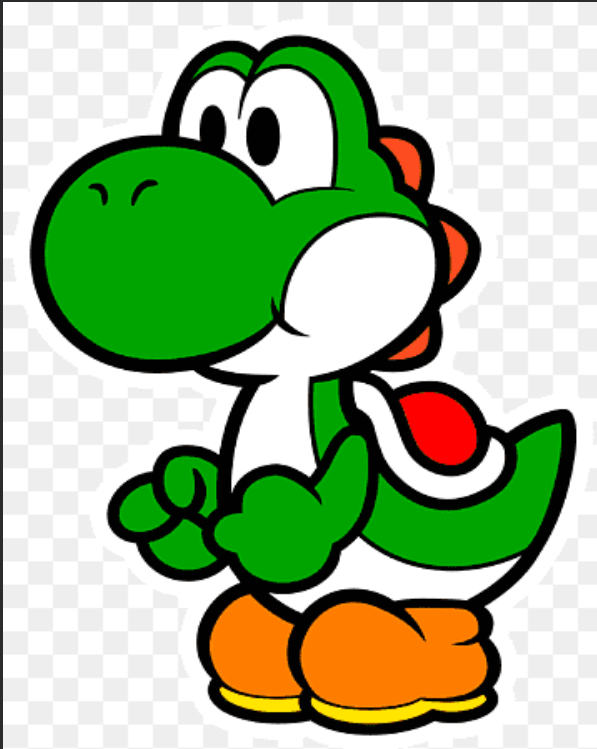
do CNN

de kernel



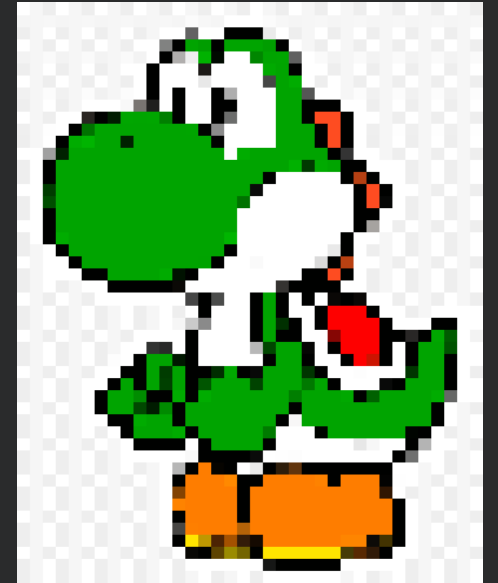
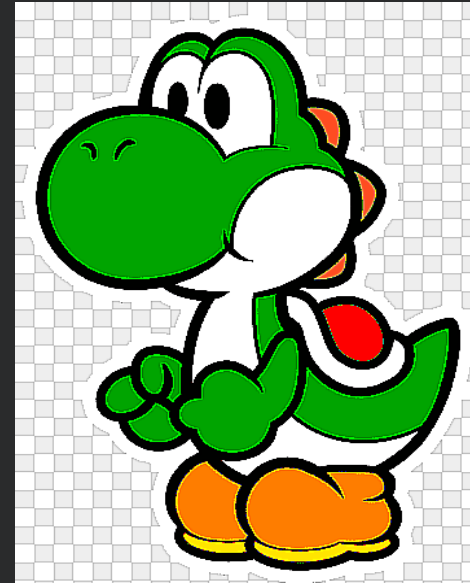
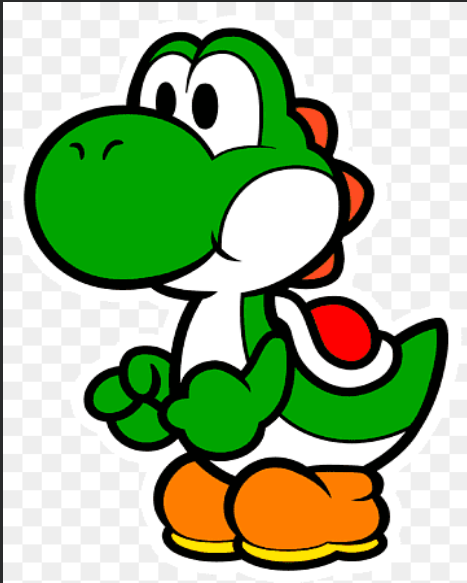
Esmiuça

Exemplos

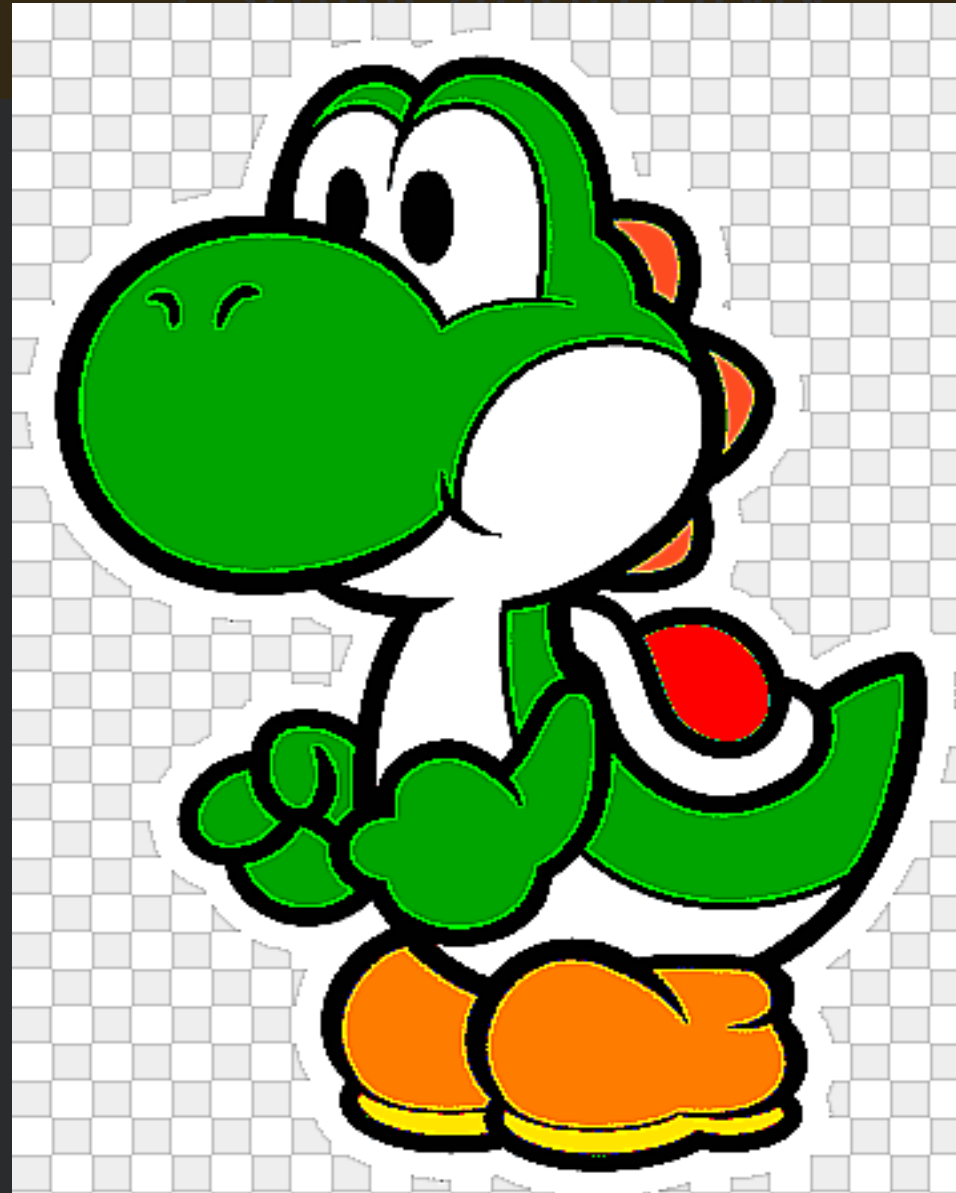


Esmiuçando CNN

Exemplos de kernel

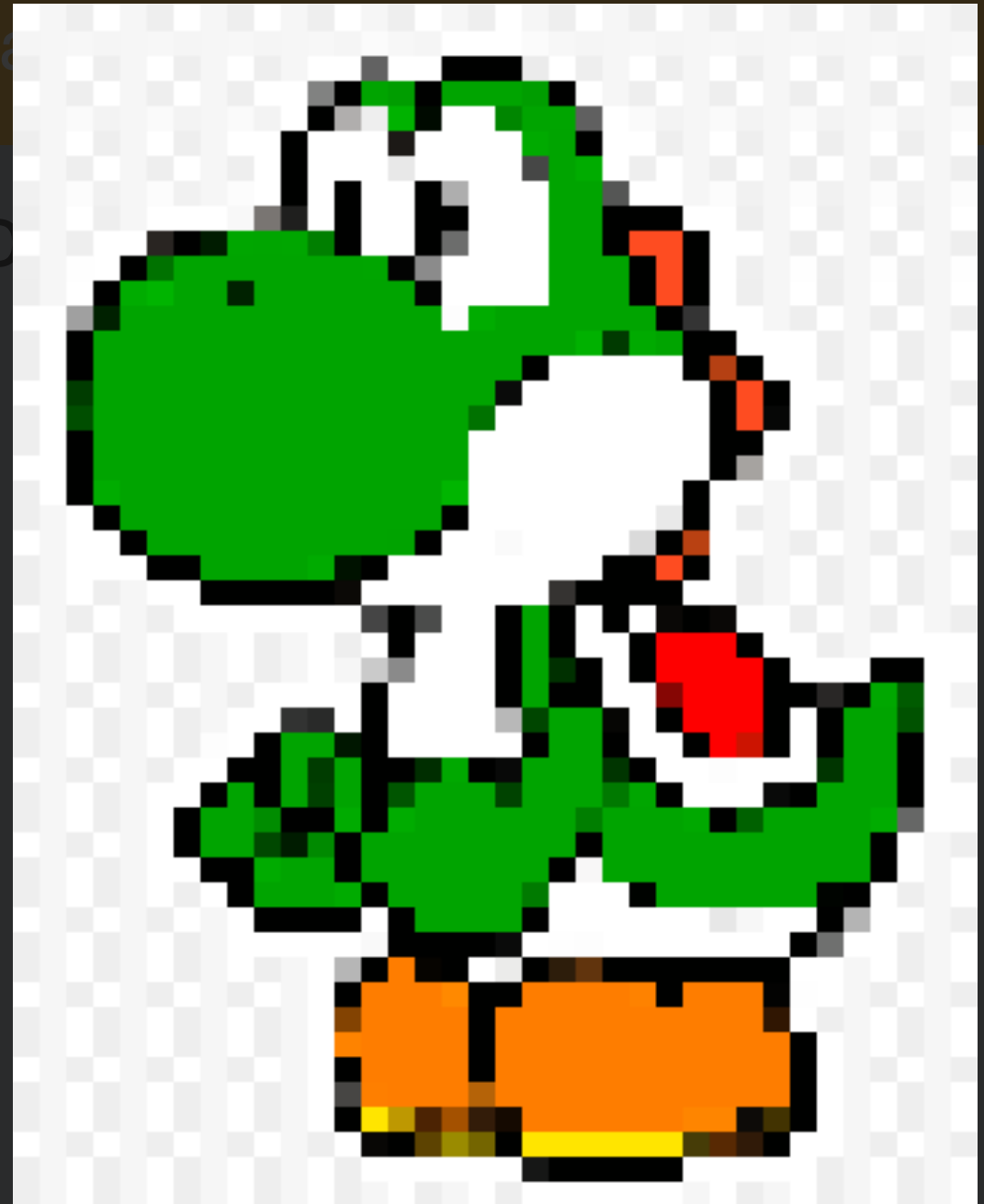
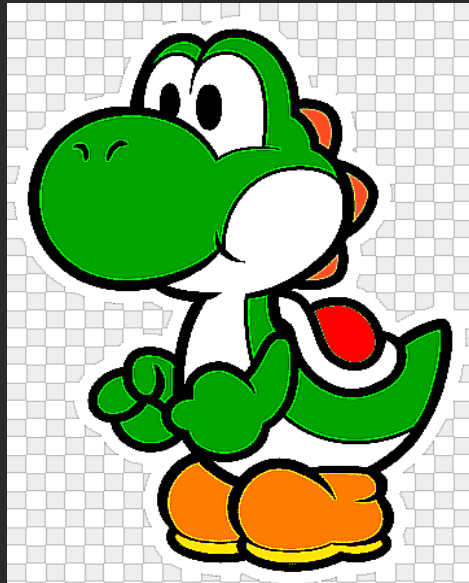


Esmiucando CNN



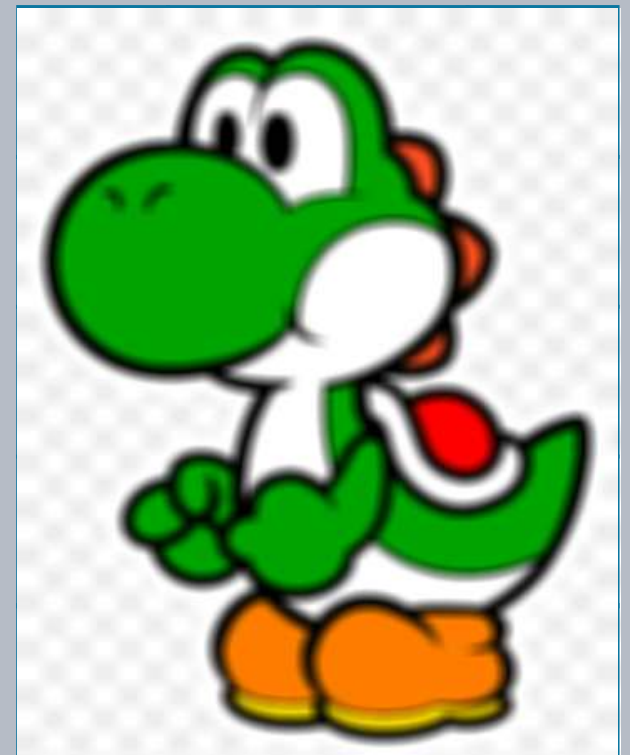
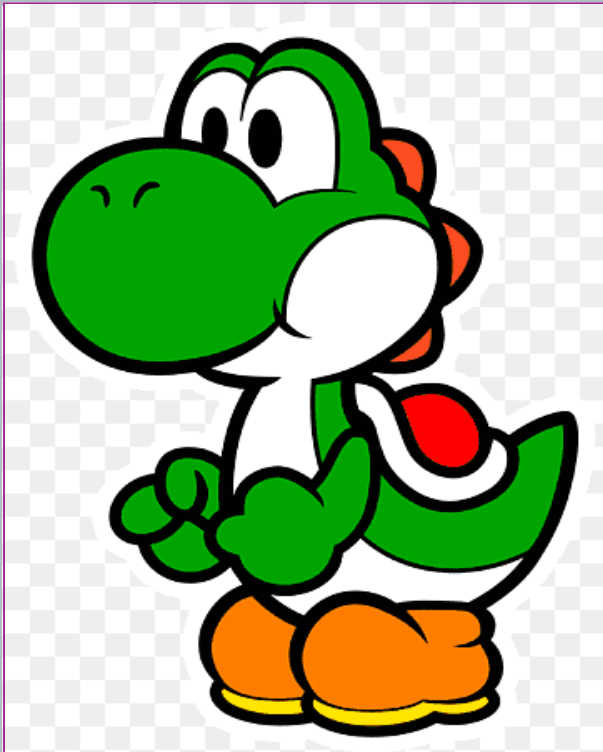
Esmiuça

Exemplo



Esmiuçando CNN

Exemplos de kernel

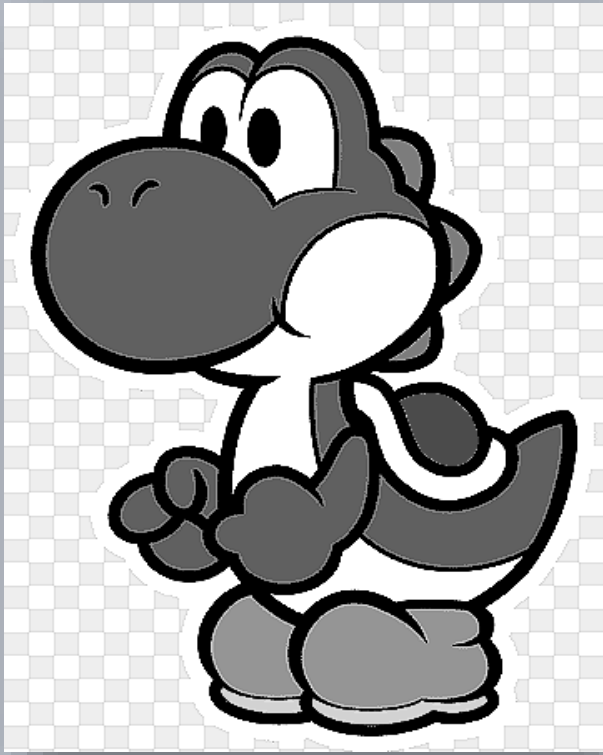


Esmiuçando CNN



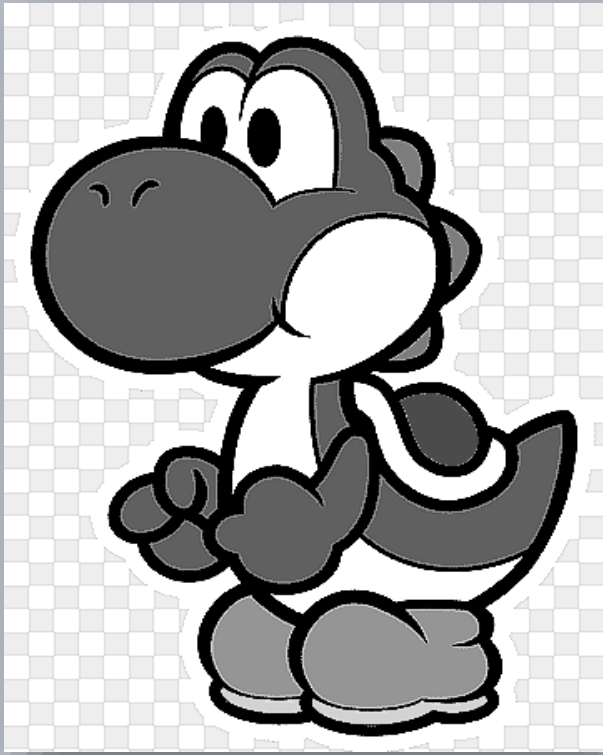
Esmiuçando CNN

Entrada



Esmiuçando CNN

Entrada

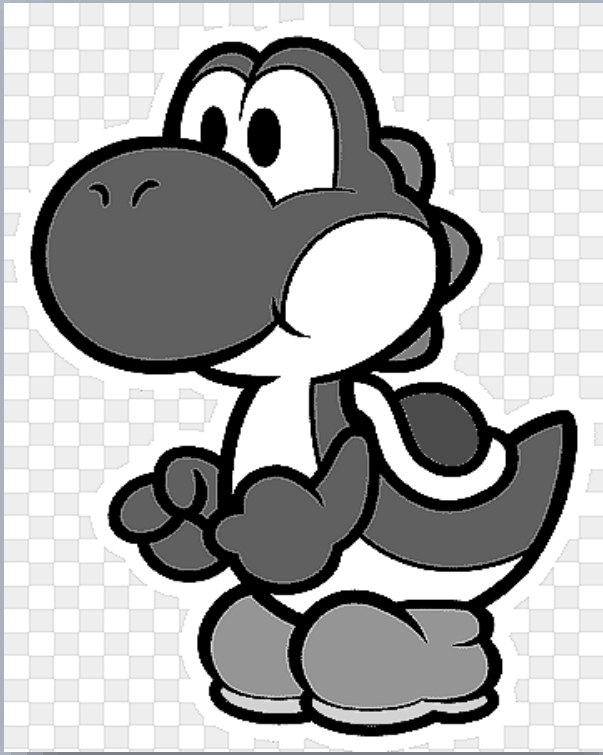


Aplica
filtro



Esmiuçando CNN

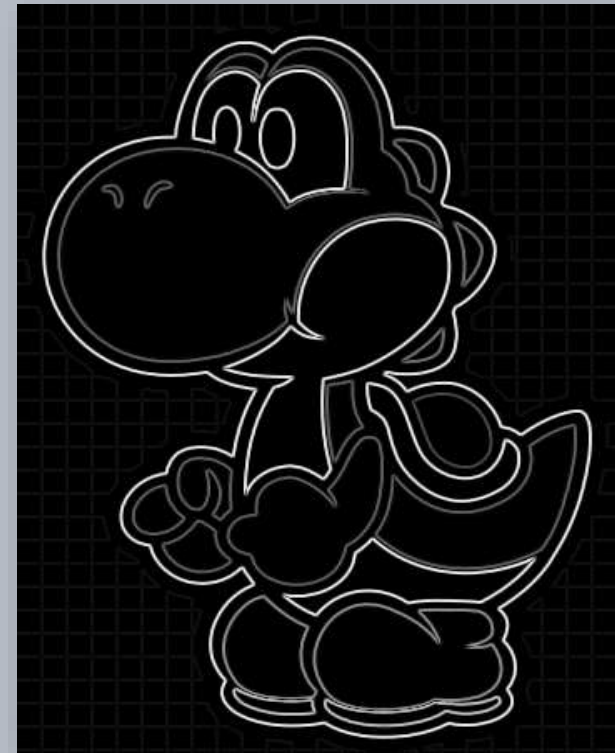
Entrada



Aplica
filtro

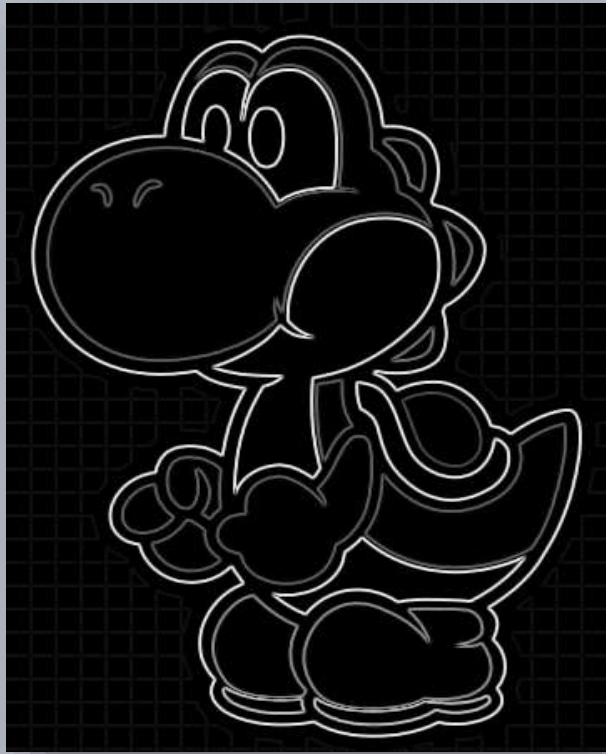


Camada Convolutucional



Esmiuçando CNN

Camada Convolutional

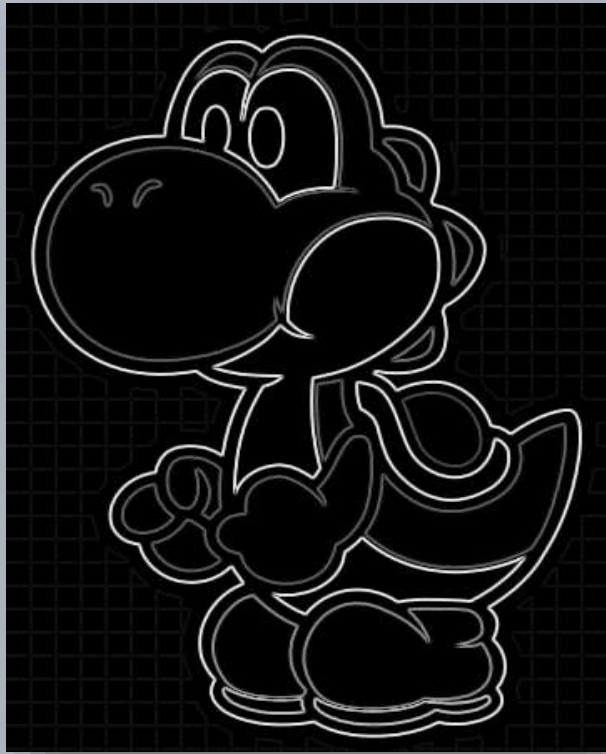


Camada de Polling

Resumir os pixels

Esmiuçando CNN

Camada Convolucional



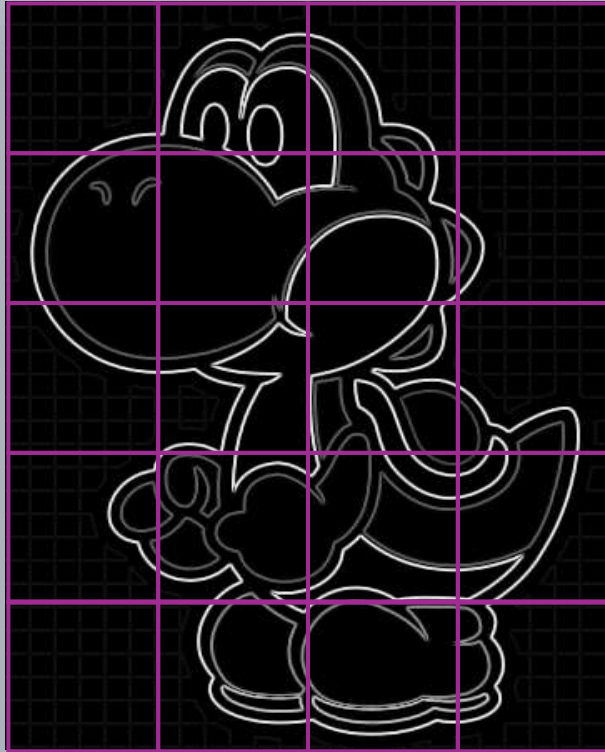
**Aplica
polling**



Camada de Polling

Esmiuçando CNN

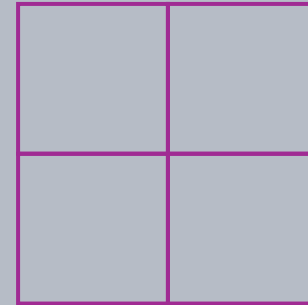
Camada Convolutional



Aplica
polling

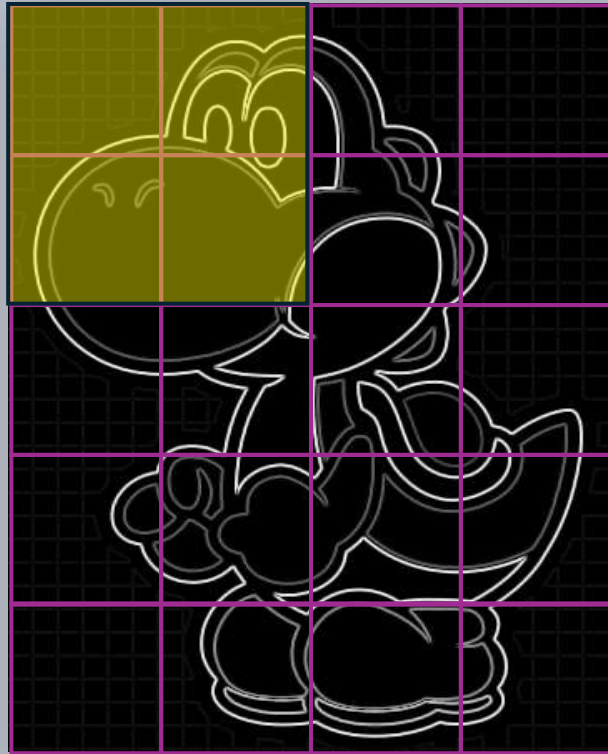


Camada de Polling



Esmiuçando CNN

Camada Convolucional



Aplica
pooling

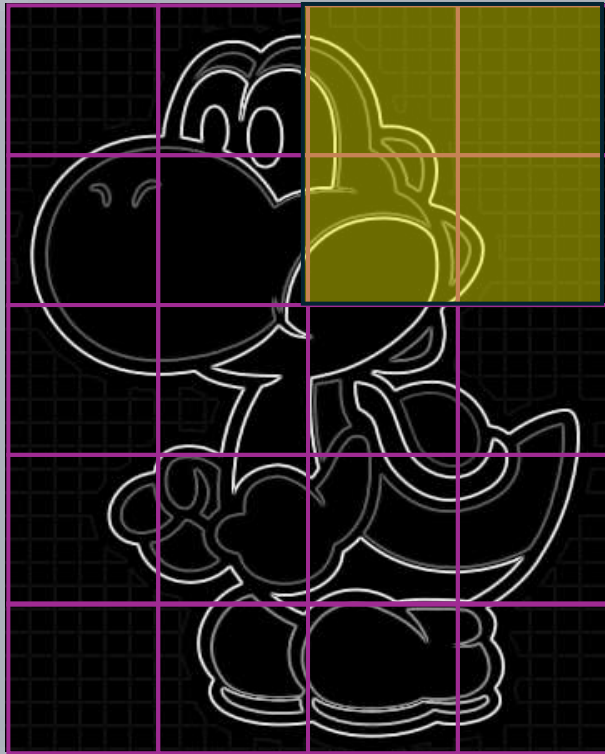


Camada de Polling



Esmiuçando CNN

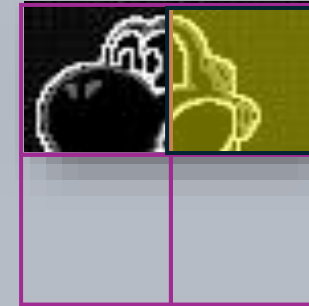
Camada Convolucional



Aplica
polling

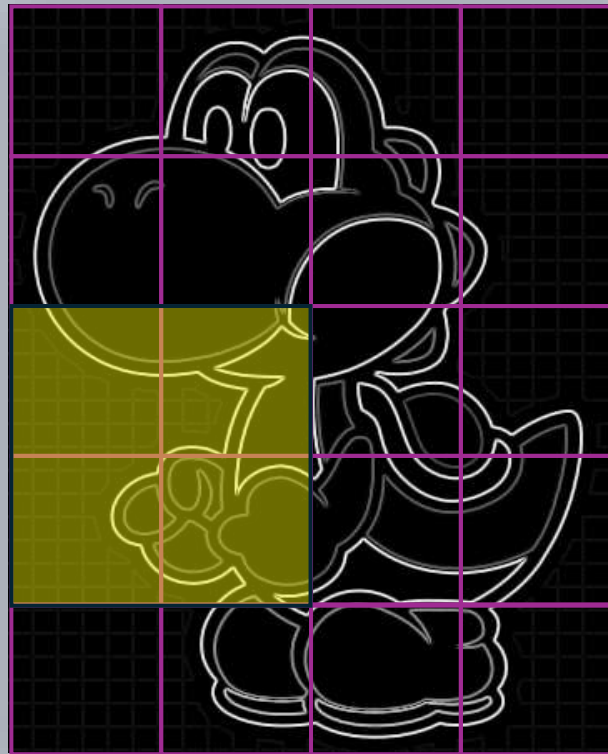


Camada de Polling



Esmiuçando CNN

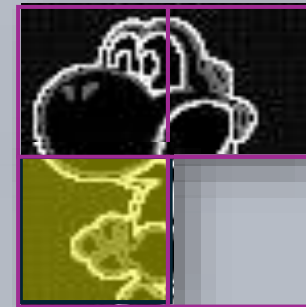
Camada Convolucional



Aplica
polling

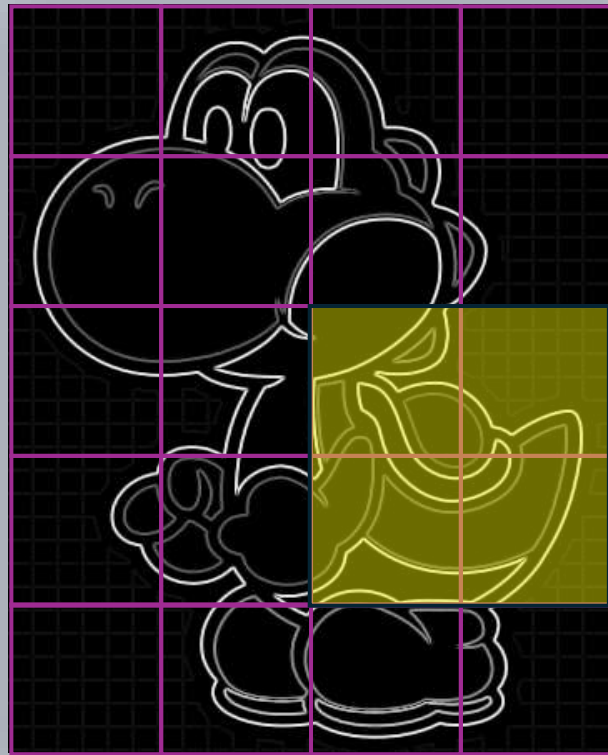


Camada de Polling



Esmiuçando CNN

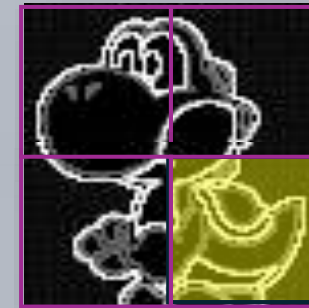
Camada Convolutacional



Aplica
polling

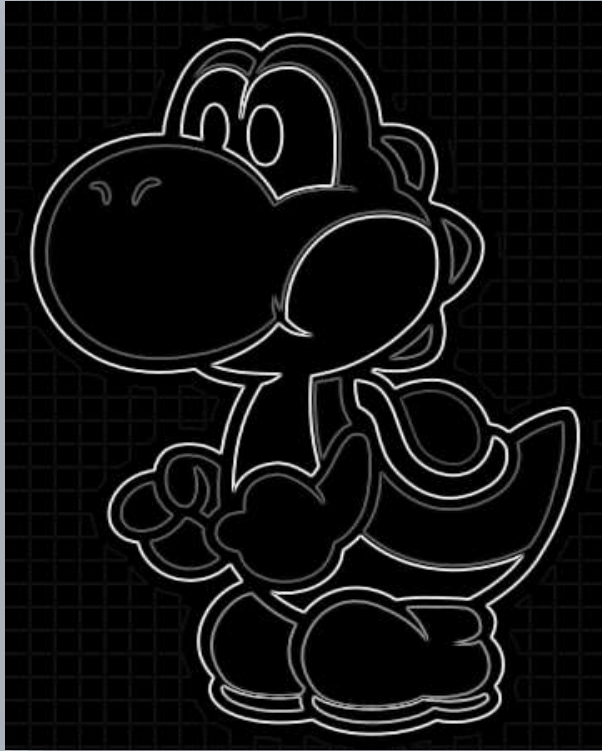


Camada de Polling



Esmiuçando CNN

Camada Convolucional



Aplica
polling



Camada de Polling



Esmiuçando CNN

Camada de Polling



**Monta mais um bloco de
Convolução + Polling**

**Planifica e manda para
classificação**

Esmiucando CNN

Camada c



Esmiuçando CNN

Camada de Polling



Monta mais um bloco de
Convolução + Polling

Camada de
convolução

Camada de
Polling

Planifica e manda para
classificação

Esmiuçando CNN

Camada de Polling



Monta mais um bloco de
Convolução + Polling

Camada
CNN

Planifica e manda para
classificação

Esmiuçando CNN

Camada de Polling



Camada
CNN

Camada
CNN

Camada
CNN

Monta mais um bloco de
Convolução + Polling

Camada
CNN

Planifica e manda para
a próxima camada

Camada
CNN

Camada
CNN

Animal

Cachorro

Beagle

Esmiuçando CNN

Camada de Polling



Camada
CNN

Camada
CNN

Camada
CNN

Monta mais um bloco de
Convolução + Polling

Camada
CNN

Planifica e manda para
a próxima camada

Camada
CNN

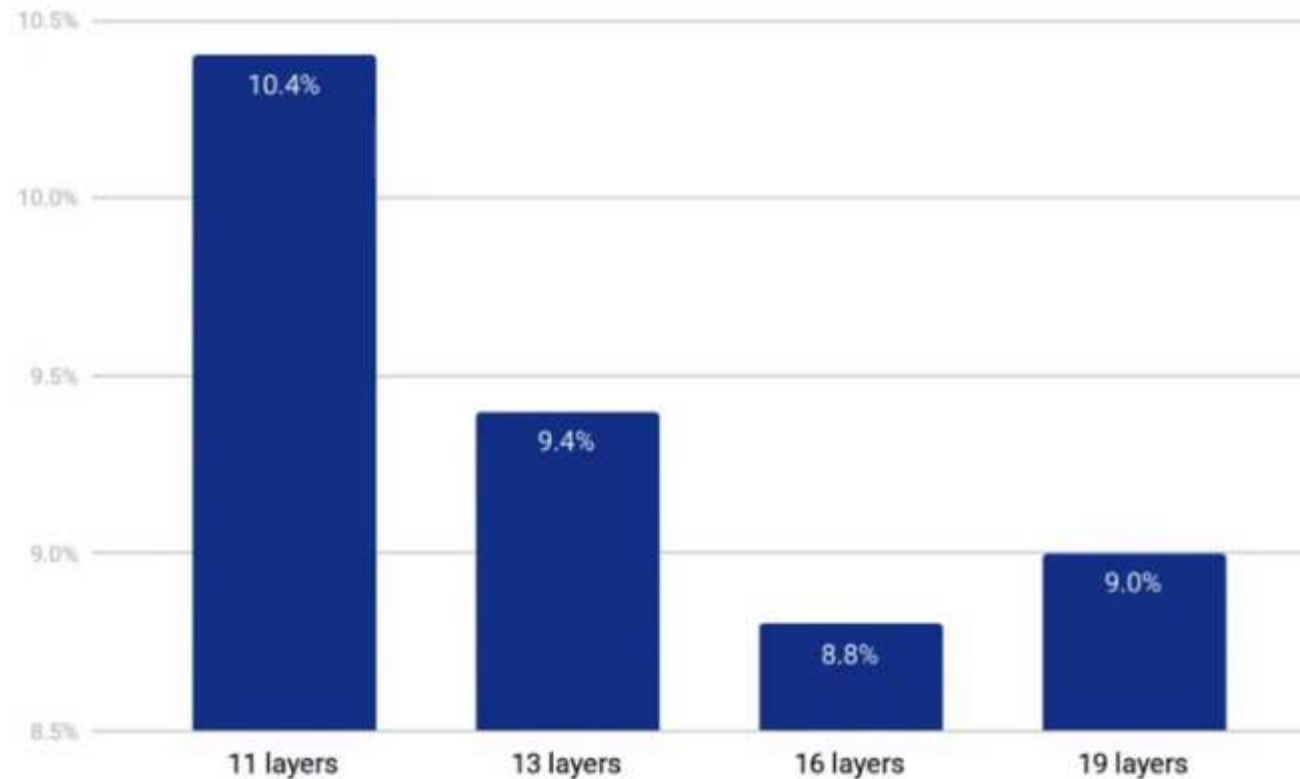
Camada
CNN

Animal

Cachorro

Beagle

VGGNet - 2014



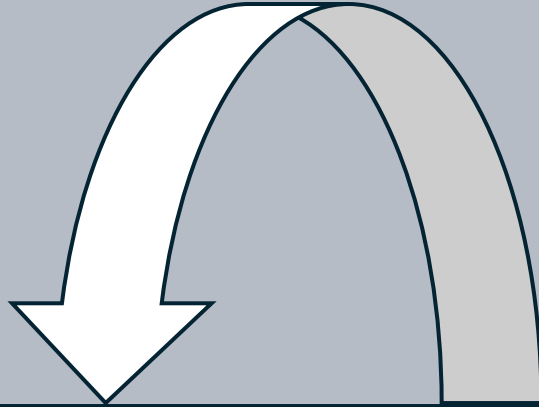
Fonte: Convolutional Neural Networks for Image Recognition, Sander Dieleman, DeepMind

Esmiuçando CNN

Camada de Polling



**Monta mais um bloco de
Convolução + Polling**



**Planifica e manda para
classificação**

Esmiuçando CNN

Camada de Polling



**Planifica e manda para
classificação**

Esmiuçando CNN

Camada de Polling

Camada totalmente conectada



**Planifica e manda para
classificação**

Esmiuçando CNN

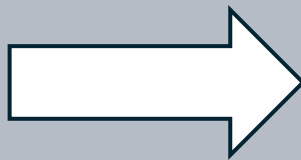
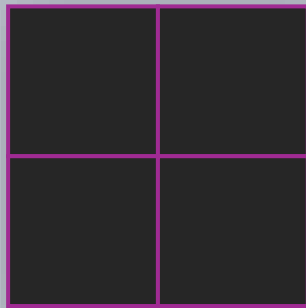
Camada de Polling



Camada totalmente conectada

Esmiuçando CNN

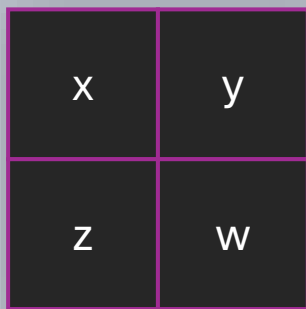
Camada de Polling



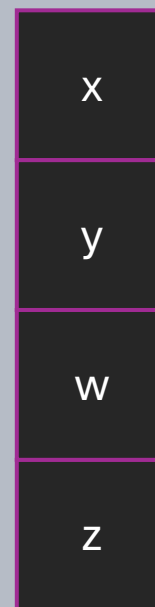
Camada totalmente conectada

Esmiuçando CNN

Camada de Polling

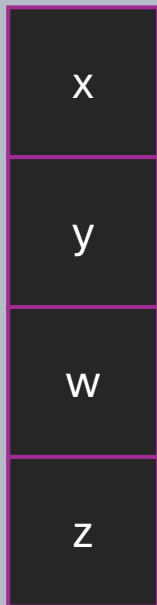


Camada totalmente conectada



Esmiuçando CNN

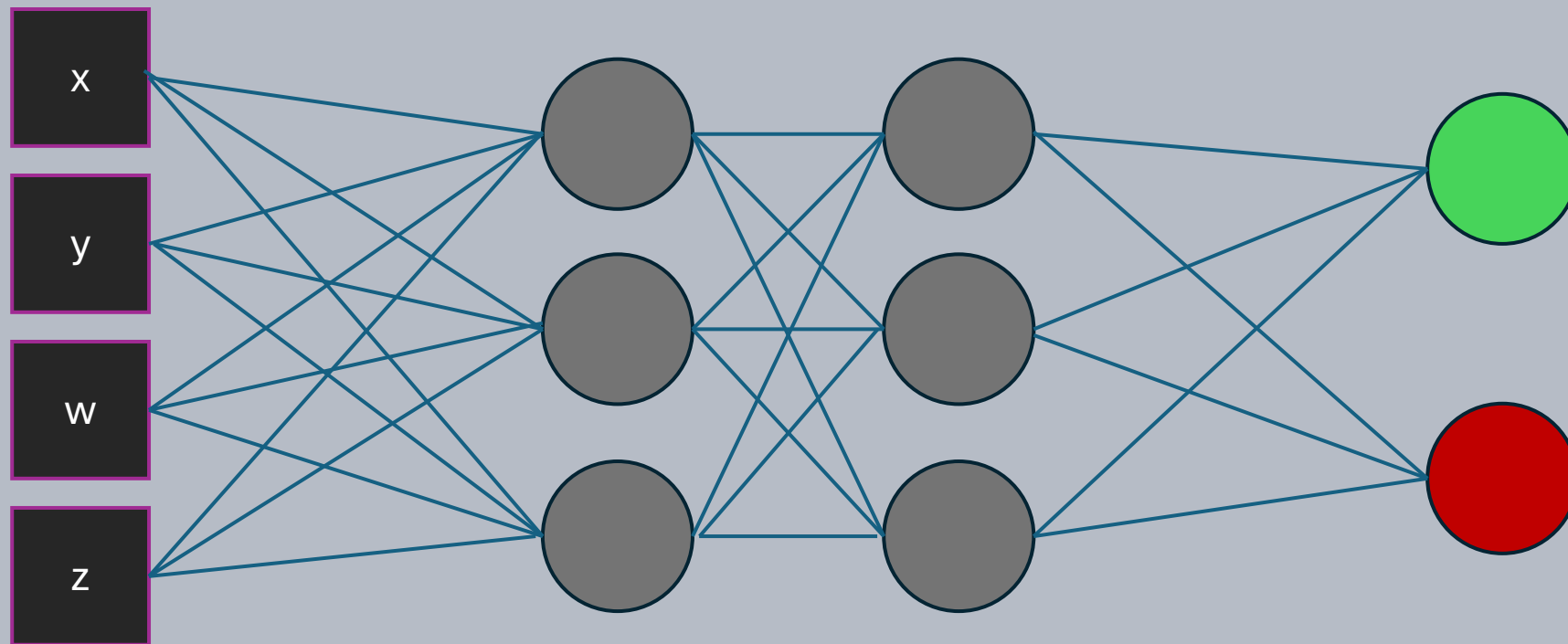
Camada totalmente conectada



Esmiuçando CNN

Camada totalmente conectada

Camada de saída



Esmiuçando CNN





Trajetória até o tema

01



O que é uma CNN?

02



Esmiuçando CNN

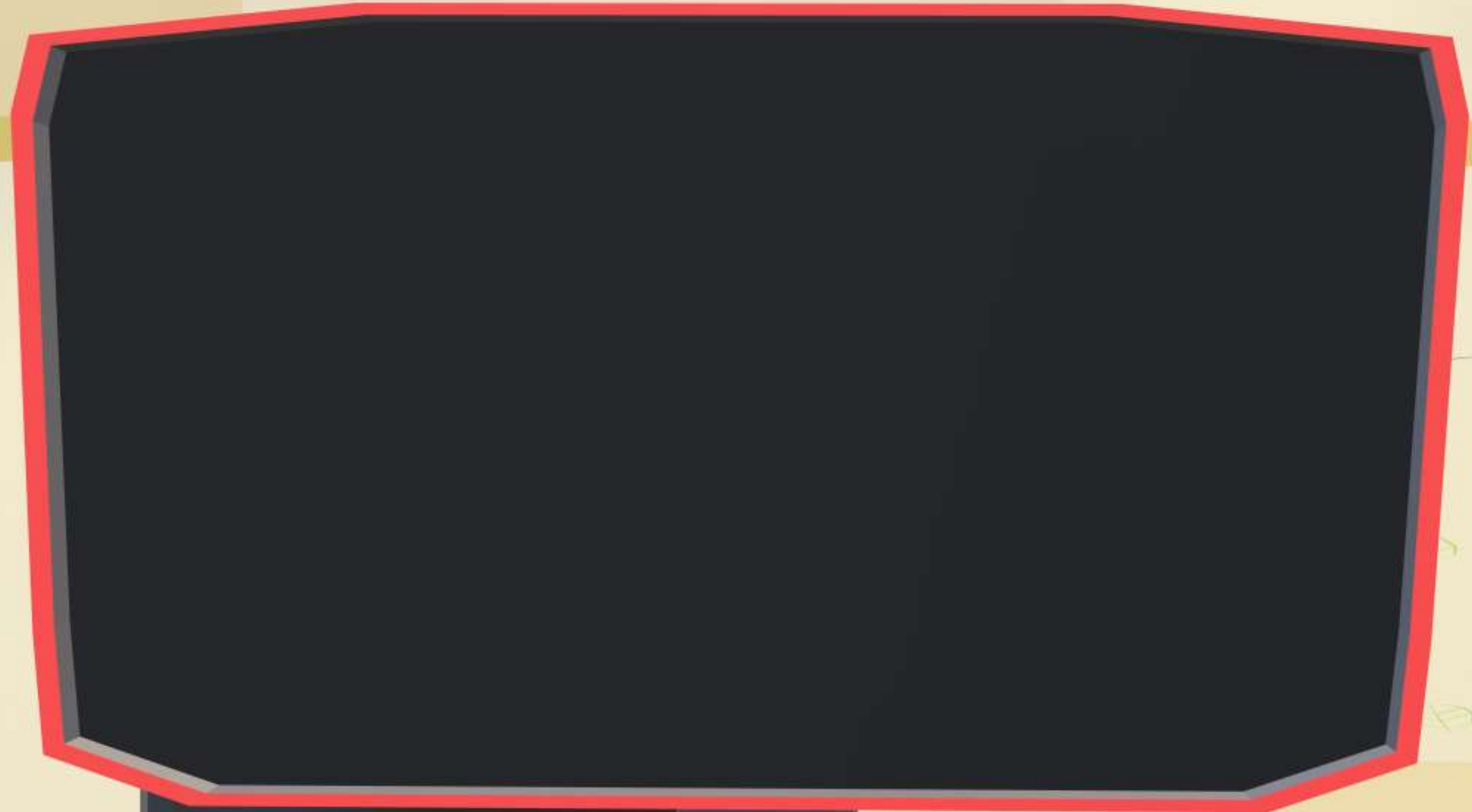
03



CNN na prática

04

CNN na prática



CNN na prática

CNN na prática

Detector de Números interativo

CNN na prática

Detector de Números interativo

MNIST – Treino



CNN na prática

Detector de Números interativo

CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo



CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```


CNN na prática

Detector de Números interativo



CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo



CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo



CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

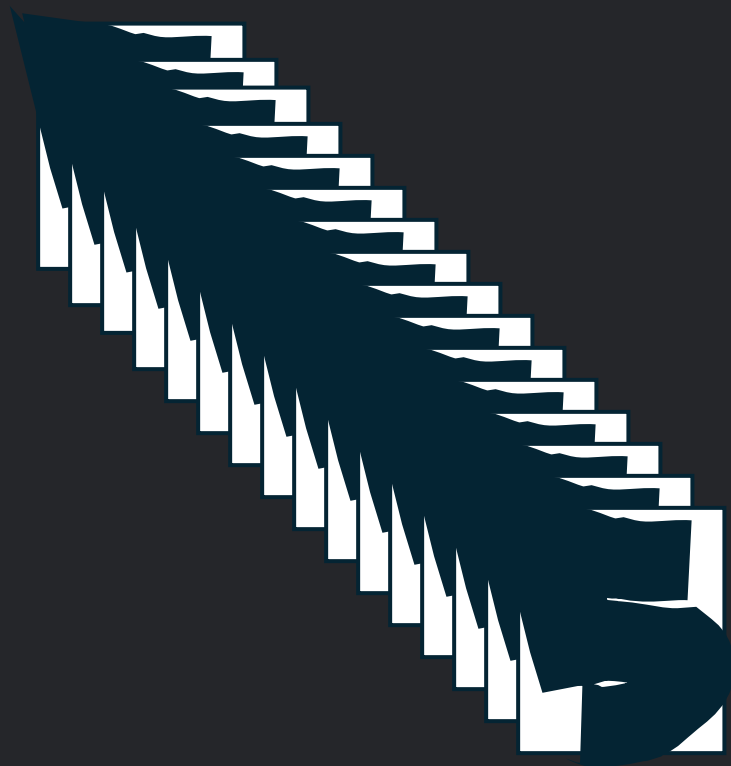
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo

CNN



```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()

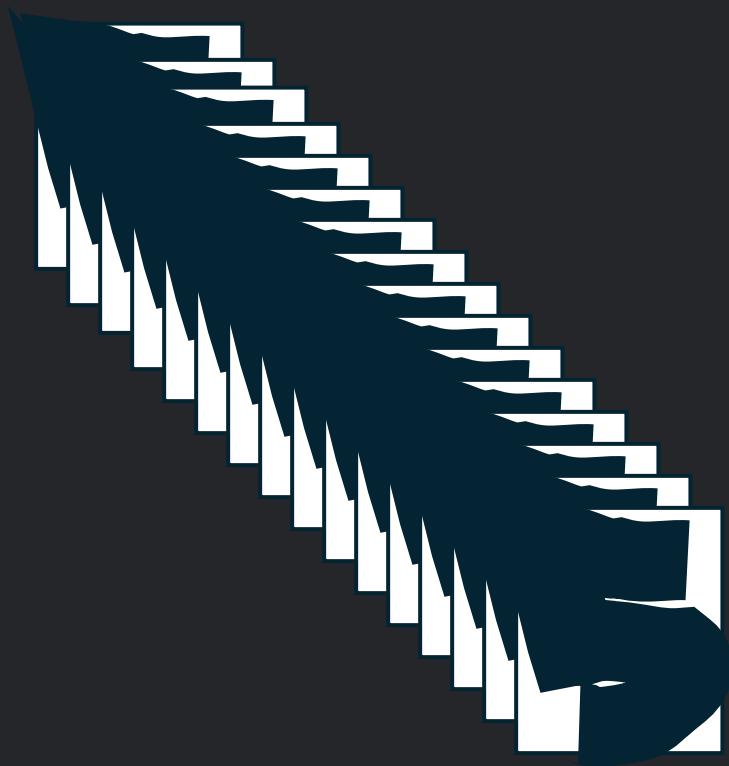
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo

CNN



```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)

        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

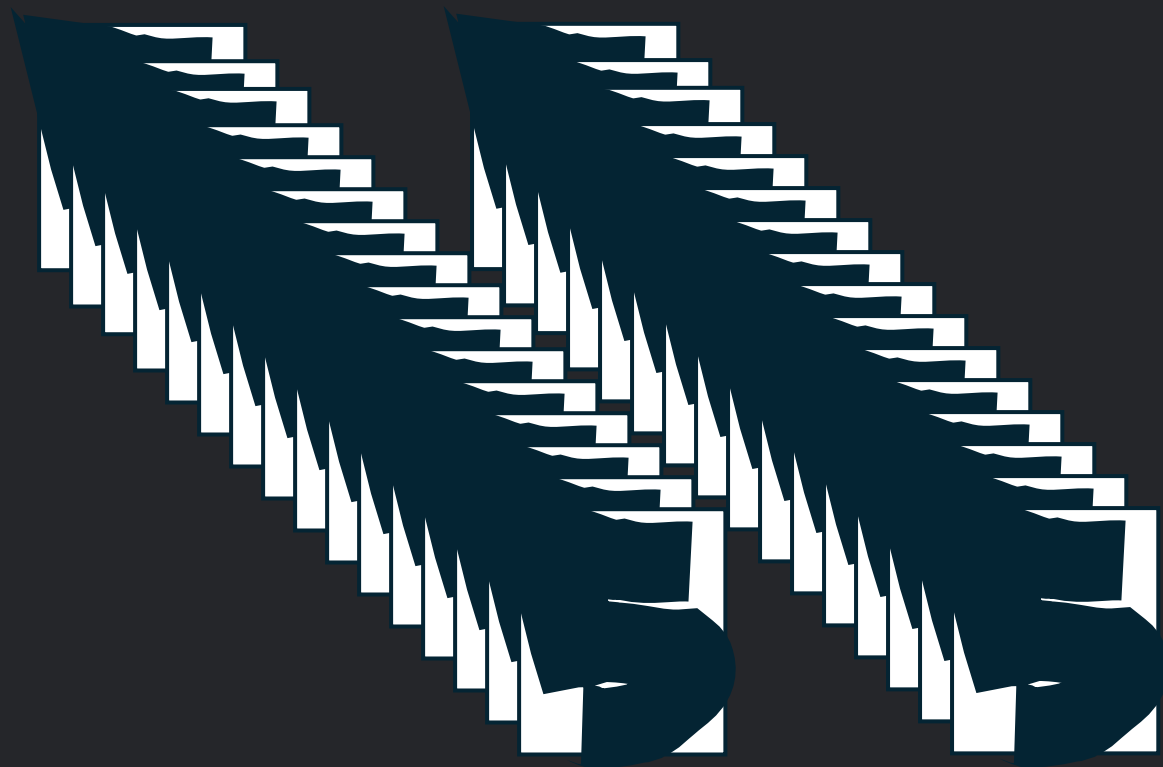
    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))

        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo

CNN



```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)

        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

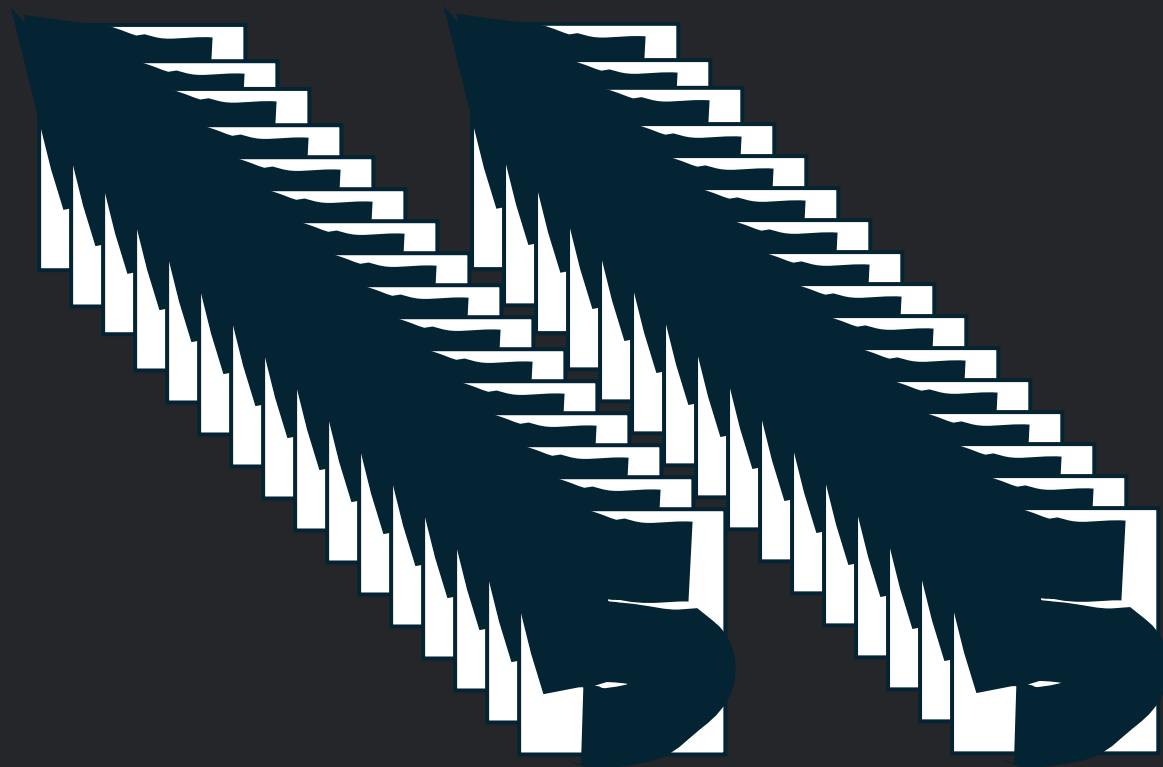
    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))

        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo

CNN



```
# Modelo da CNN
```

```
class CNN(nn.Module):
```

```
    def __init__(self):
```

```
        super(CNN, self).__init__()
```

```
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
```

```
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
```

```
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
```

```
        self.fc2 = nn.Linear(128, 10)
```

```
        self.relu = nn.ReLU()
```

```
        self.pool = nn.MaxPool2d(2)
```

```
    def forward(self, x):
```

```
        x = self.relu(self.conv1(x))
```

```
        x = self.pool(self.relu(self.conv2(x)))
```

```
        x = x.view(x.size(0), -1)
```

```
        x = self.relu(self.fc1(x))
```

```
        x = self.fc2(x)
```

```
        return x
```

CNN na prática

Detector de Números interativo

CNN



Vetor = [9216]

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```


CNN na prática

Detector de Números interativo



CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo



CNN

```
# Modelo da CNN
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3)
        self.fc1 = nn.Linear(64 * 12 * 12, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()
        self.pool = nn.MaxPool2d(2)

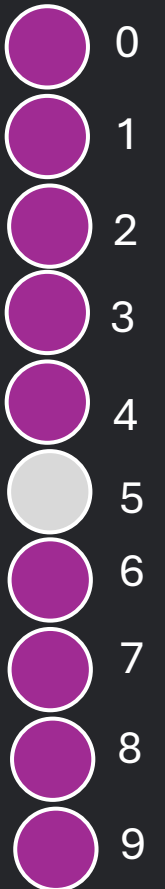
    def forward(self, x):
        x = self.relu(self.conv1(x))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(x.size(0), -1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

CNN na prática

Detector de Números interativo



CNN



CNN na prática

Detector de Números interativo



nn-digit-recognize.vercel.app



Trajetória até o tema

01



O que é uma CNN?

02



Esmiuçando CNN

03



CNN na prática

04

