



Inteligência Artificial ao alcance de todos.

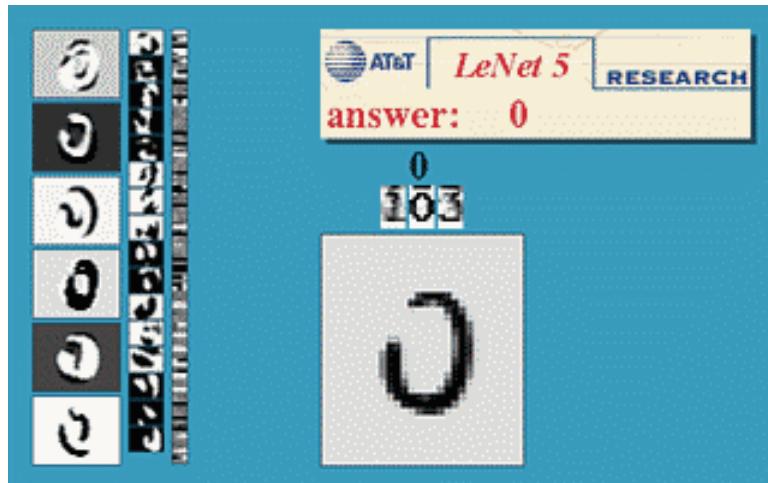
www.escolalivre-ia.com.br

O que é Visão Computacional?

- Desenvolve técnicas para ajudar computadores a “ver”
- Construção de sistemas artificiais que obtém informação provindas de imagens ou vídeos.

Quais são suas aplicações?

Reconhecimento Ótico de Caracteres



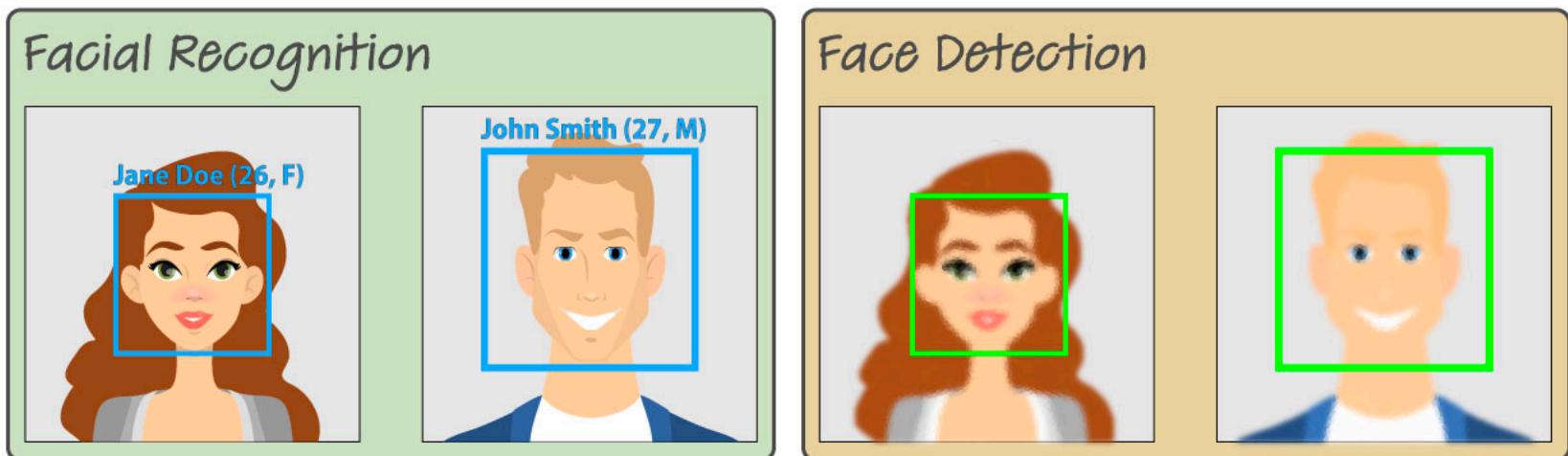
Reconhecimento de Dígito, AT&T



Reconhecimento de Placas veículares

Quais são suas aplicações?

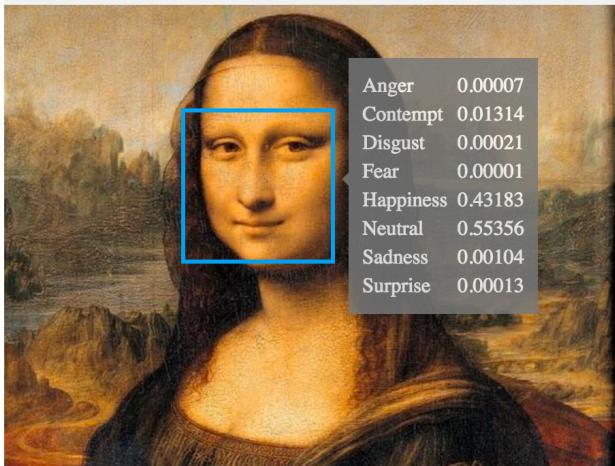
Reconhecimento e Detecção Facial



Fonte: <https://catarinoconsulting.com.au/articles/2019/6/13/facial-detection-vs-facial-recognition>

Quais são suas aplicações?

Detecção de Emoções



Fonte: <https://thenextweb.com/microsoft/2015/11/11/take-that-inside-out/>

Quais são suas aplicações?

Reconhecimento e Detecção de Objetos



Fonte: software.intel.com

Quais são suas aplicações?

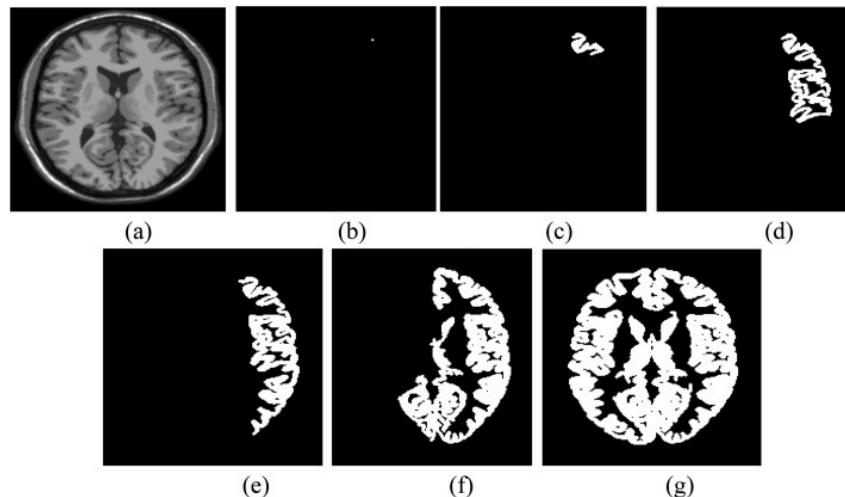
Estimação de Pose



Fonte: beyondminds.ai

Quais são suas aplicações?

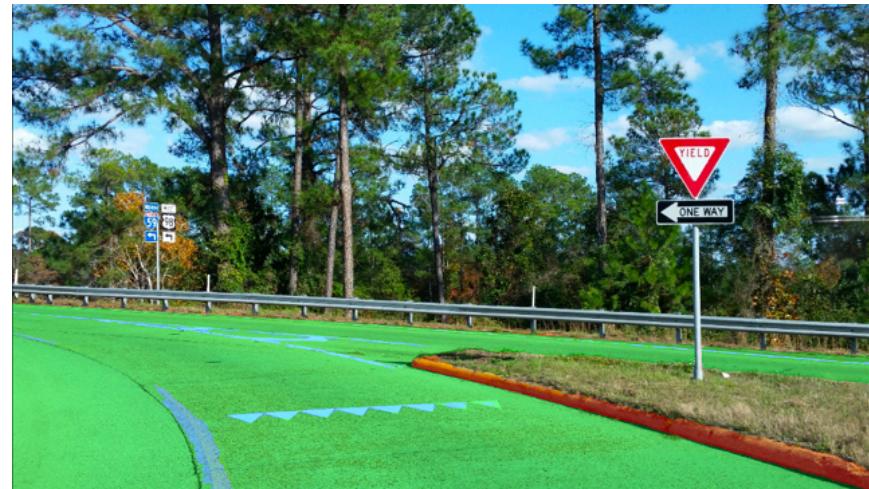
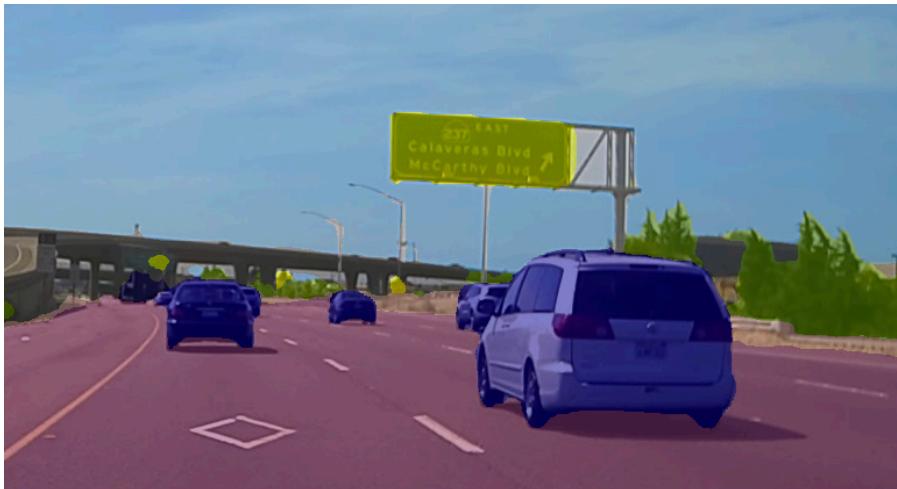
Processamento de Imagens Médicas



Fonte: <https://www.sciencedirect.com/science/article/pii/S235291481730045X>

Quais são suas aplicações?

Carros Inteligentes (Segmentação de Imagem e Detecção de Objetos Máscara R-CNNs)



Fonte: <https://docs.openvino-toolkit.org/>

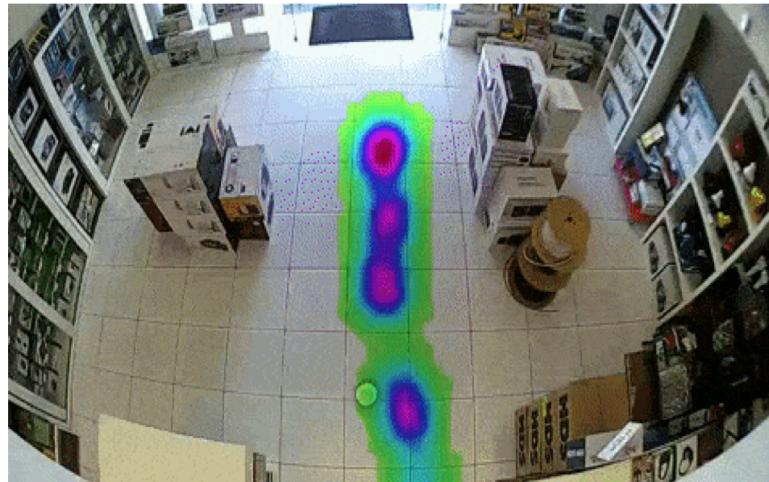
Quais são suas aplicações?

Marketing (Fluxo de Pessoas)



Quais são suas aplicações?

Marketing



Fonte: <https://dodvision.com/>

Matemática

- Processamento de sinal e imagem
- Cálculo Diferencial
- Otimização
- Álgebra Linear
- Probabilidade

Como os computadores veem?

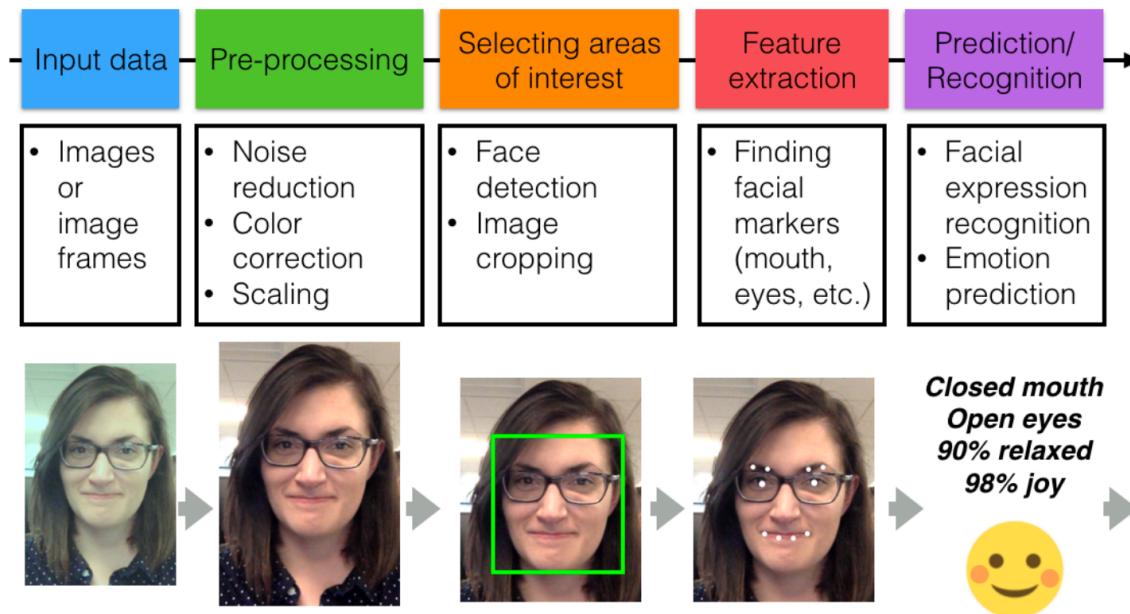
Representação de Imagens como Dados Númericos



What Computer Sees	
0	2 15 0 0 11 10 0 0 0 0 0 9 9 9 0 0 0
0	0 0 4 60 157 236 255 255 177 95 61 32 0 0 29
0	10 16 119 238 255 244 245 243 250 249 255 222 103 10 0
0	14 170 255 255 244 254 255 253 245 255 249 253 251 124 1
2	98 255 228 255 251 254 211 141 116 122 215 251 238 255 49
13	217 243 255 155 33 229 52 2 0 10 12 232 255 255 36
16	229 252 254 49 12 0 0 7 7 0 70 237 252 235 62
6	141 245 255 212 25 11 9 3 0 118 238 243 255 137 0
0	87 252 250 248 215 60 0 5 122 252 255 248 144 6 0
0	13 111 255 255 245 255 182 183 248 252 242 208 36 0 19
1	0 5 117 251 255 241 255 247 255 241 162 17 0 7 0
0	0 0 4 35 251 255 246 254 253 256 120 11 0 1 0
0	0 4 51 255 255 249 242 252 255 244 255 182 10 0 4
0	22 206 252 246 251 241 103 24 111 255 245 255 194 9 0
0	111 255 242 255 156 24 0 0 6 39 255 232 230 56 0
0	218 251 250 137 7 11 0 0 0 2 62 255 250 125 3
0	173 255 255 101 9 20 0 13 3 13 182 251 245 61 0
0	107 251 241 255 230 98 55 19 111 217 248 253 255 52 4
0	18 146 250 255 247 255 255 255 249 255 240 255 129 0 5
0	0 0 23 113 215 255 250 248 255 255 248 248 118 14 12 0
0	0 6 1 0 52 153 233 255 252 247 37 0 0 4 1
0	0 5 5 0 0 0 0 0 14 1 0 6 6 0 0 0

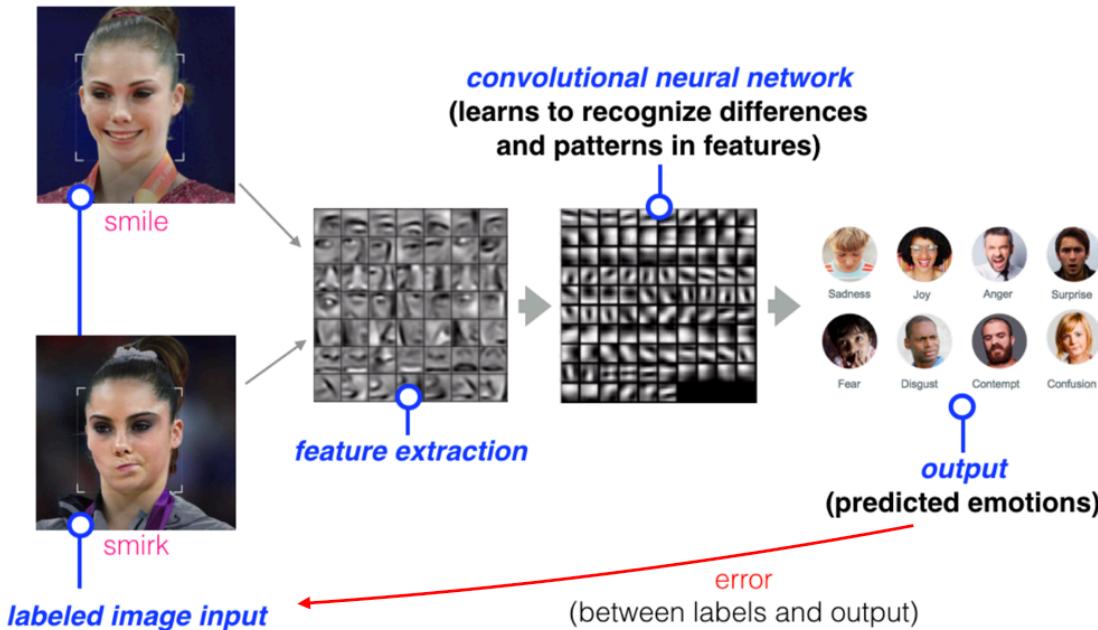
Fonte: <https://mozanunal.com/2019/11/img2sh/>

Pipeline de Visão Computacional



Fonte: <https://www.udacity.com>

Como treinar um modelo?



Fonte: <https://www.udacity.com/>

Extração de Características

- Podemos utilizar intensidade para detectar áreas de interesse.
- Por exemplo, detectar bordas de um objeto olhando por uma súbita mudança em intensidade.
- Para isto, utilizamos filtros que olham para grupos de pixels e detectam mudanças abruptas em intensidade em uma imagem.

Jupyter Notebook: Canny Edge Detector

- Algoritmo detector de bordas amplamente utilizado e preciso
1. Filtra ruídos utilizando desfoque gaussiano
 2. Encontra a força e direção das bordas utilizando filtro Sobel
 3. Aplica supressão não máxima para isolar as bordas mais fortes e finas
 4. Usa histerese para isolar as melhores bordas

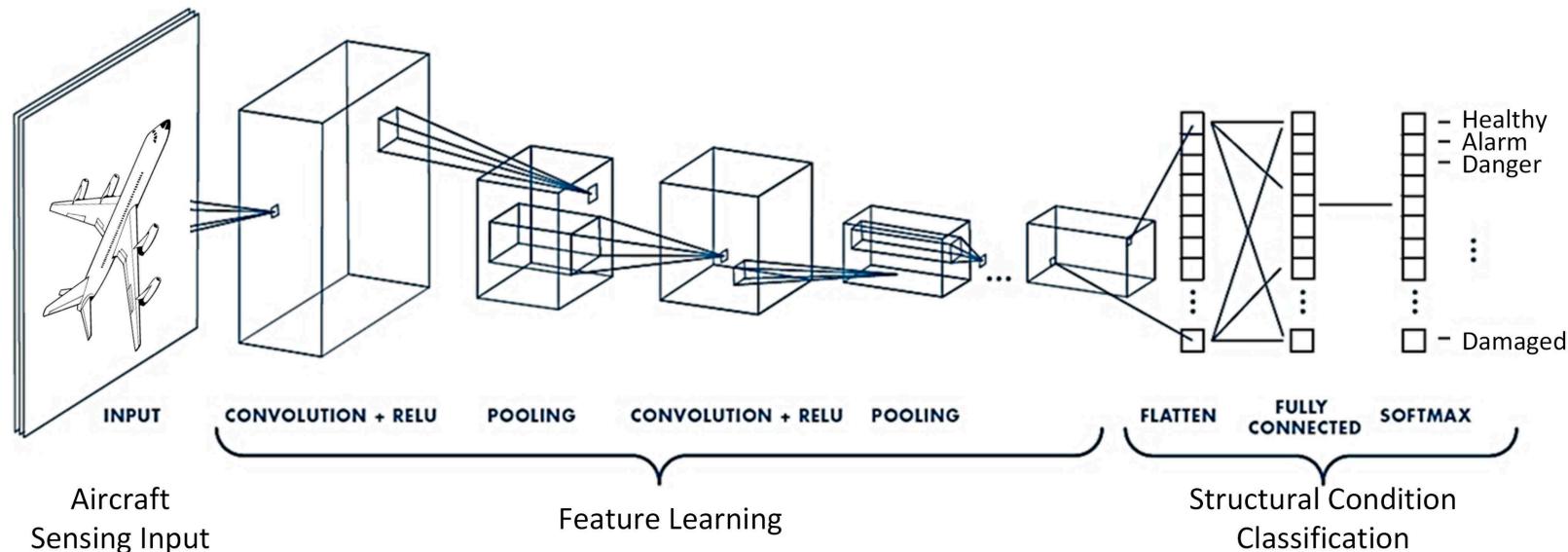
Open Source Computer Vision Library (OpenCV)

- É uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial para o desenvolvimento de aplicativos na área de visão computacional.



Fonte: <https://opencv-python-tutroals.readthedocs.io/en/latest/index.html>

Redes Neurais Convolucionais para Classificação de Imagens



Fonte: <https://www.mdpi.com/1424-8220/19/22/4933/htm>

Redes Neurais Convolucionais

- Tipo de rede neural que encontra e representa padrões no espaço da imagem.
- Conseguem aprender padrões espaciais olhando para uma área de pixels em uma imagem

Redes Neurais de Convolucionais

- Cada Rede Neural de Convolução é feita por múltiplas camadas.
- Principais:
 1. Camada de Convolução
 2. Camada de Pooling
 3. Camada totalmente conectada

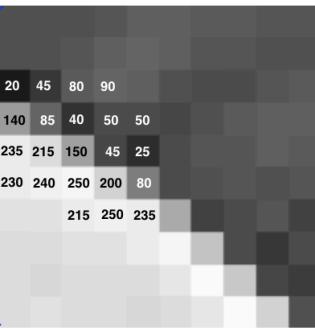
Camada Convolucional

- Pode ser vista como uma extrator de características da Rede Neural.
- Aprende a encontrar características espaciais da imagem de entrada
- É feita aplicando uma série de filtros de imagens, conhecidos como kernels de convolução, a uma imagem.

Camada Convolucional

- Kernels de Convolução são grades pequenas com valores que deslizam sobre uma imagem, pixel por pixel
- Produzem uma imagem de saída que serão quase o mesmo tamanho da imagem de entrada.
- Múltiplos Kernels irão produzir múltiplas imagens filtradas de saída, conhecidos como mapas de características, ou mapa de ativação.

Camada Convolucional



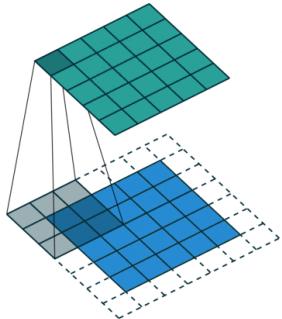
0	-1	0
-1	4	-1
0	-1	0



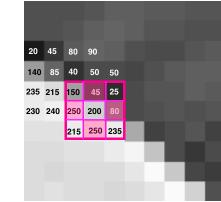
Fonte: <https://www.udacity.com>

Camada Convolucional

- Os valores dentro do kernel, os quais são multiplicados pelo valor do pixel correspondente são chamados de **pesos**.
- Eles determinar o quanto importante (ou pesados) um pixel é formando uma nova imagem de saída.



<i>weights</i>
0 -1 0
-1 4 -1
0 -1 0



$$\begin{aligned}
 & 0 * 150 + -1 * 45 + 0 * 25 + \\
 & -1 * 250 + 4 * 200 + -1 * 80 + \\
 & 0 * 215 + -1 * 250 + 0 * 235 \\
 & = 175
 \end{aligned}$$

Camada Convolucional

- Bordas da Imagem
- A convolução não funciona nos pixels das bordas de uma imagem.
- Ex: Um kernel 3x3 não pode ser perfeitamente colocado sobre um pixel central nas bordas.
- Solução: Padding!

Camada Convolucional

- Padding
- É um processo que alguns pixels (pesos 0), são adicionados ao redor da imagem antes da convolução com a finalidade de manter a dimensionalidade na imagem resultante.

0	-1	0
-1	4	-1
0	-1	0



What to do at the edges?

Normalização

- Os valores de imagens cinzas vão em valores de 0-255.
- Redes Neurais trabalham melhores com valores entre 0 a 1.
- Antes de avançarmos, precisamos normalizar o dados da faixa de 0-255 para 0 a 1.

Função de Ativação

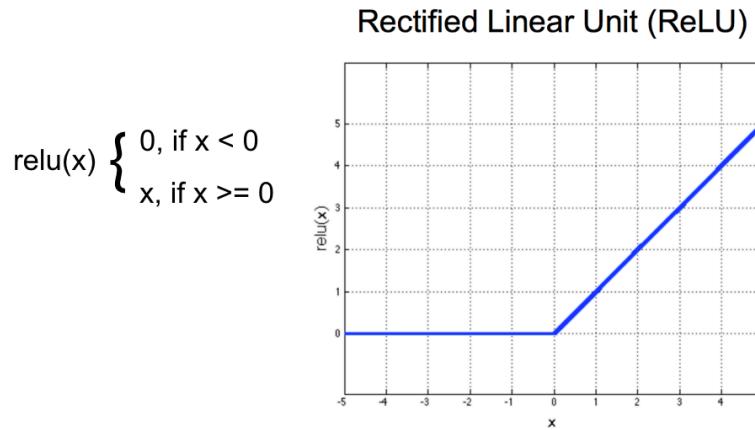
- Levando em consideração, a mudança de valor de cada pixel no processo de normalização, a Rede Neural aplica uma Função de Ativação que transforma o valor de cada pixel.
- A mais comum é a função de ativação ReLu (Rectified Linear Unit)
- Esta função transforma todos os valores negativos em 0s (preto) e também introduz não-linearidade no modelo.

Função de Ativação

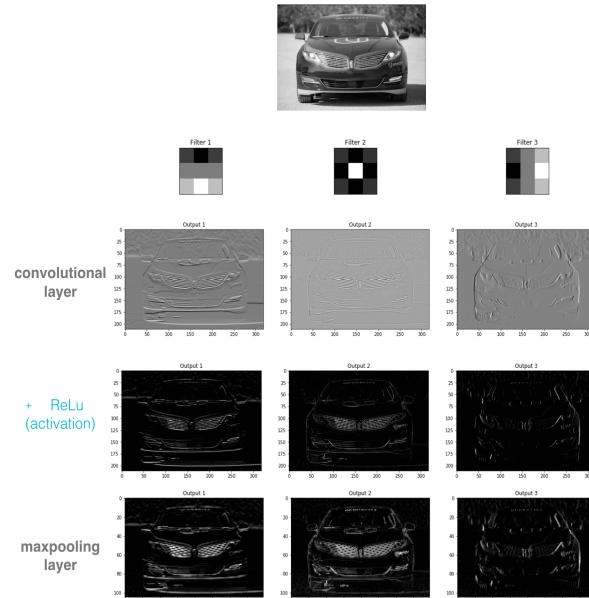
- Levando em consideração, a mudança de valor de cada pixel no processo de normalização, a Rede Neural aplica uma Função de Ativação que transforma o valor de cada pixel.
- A mais comum é a função de ativação ReLu (Rectified Linear Unit)

Função de Ativação ReLu

- Esta função transforma todos os valores negativos em 0s (preto) e também introduz não-linearidade no modelo.



Jupyter Notebook: Visualizando a camada de Convolução



Fonte: Udacity

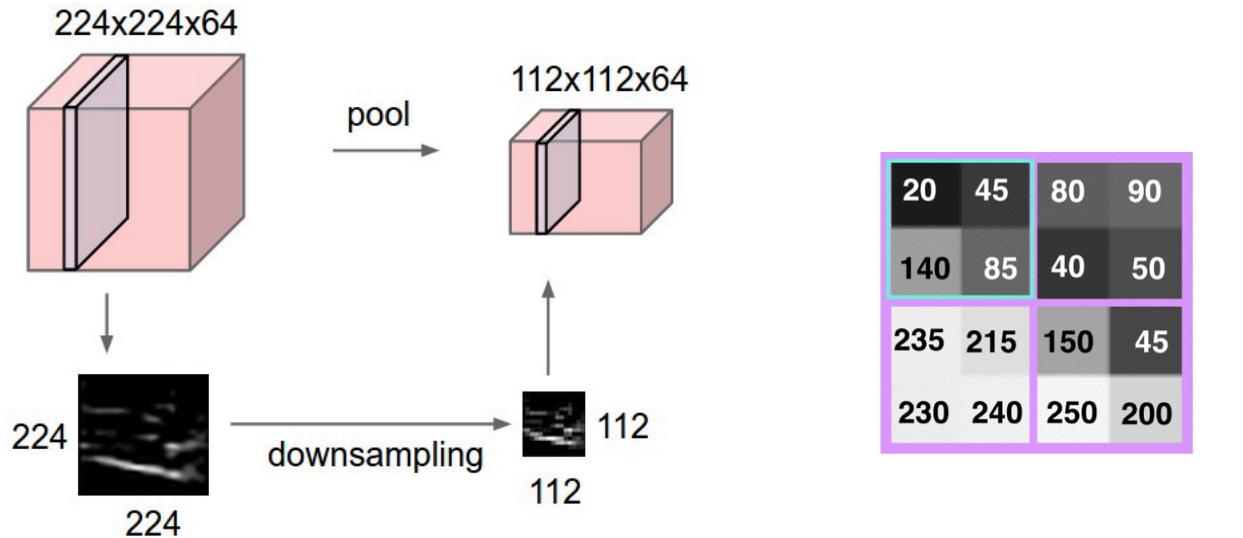
Camada de Pooling

- Pooling é um processo de redução da dimensionalidade/mapas de características.
- De modo simples, podemos entender essa transformação como uma redução do tamanho da imagem.
- Objetivo: Reduzir quantidade de parâmetros treinados pela rede e reduzir sua variância a pequenas alterações.

Camada de Pooling

- Existem 3 operações de Pooling: MaxPooling, SumPooling e AveragePooling.
- A mais utilizada é a **MaxPooling**
- Retira o maior elemento de determinada região da matriz e após isso, é feito um deslizamento considerando um parâmetro de stride, para a aplicação da próxima operação.

MaxPooling



maxpool

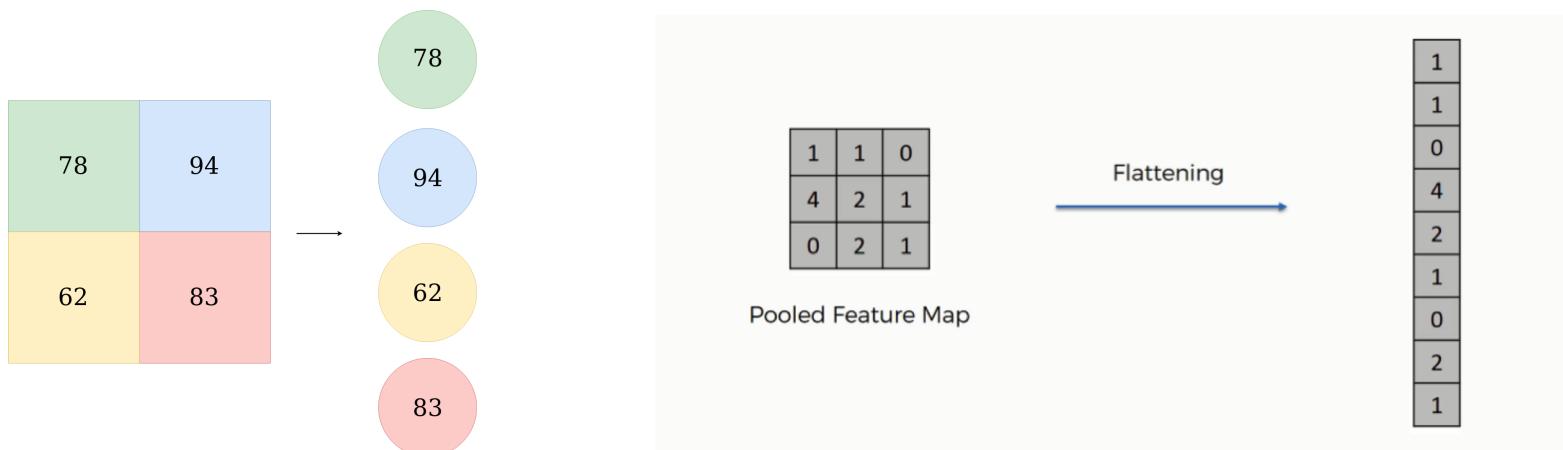
- filter/patch size = 2×2
- stride = 2

Fonte: <https://cs231n.github.io/convolutional-networks/#conv>

Camadas Flattened

- O processo da Rede começa com camadas de convolução e pooling, quebrando a imagem em características e analizando-as independentemente.
- O resultado deste processo será “Achatado” (a saída das camadas de convolução e pooling são matrizes 3D). Acharatar significa transformar esses valores em um vetor

Camadas Flattened



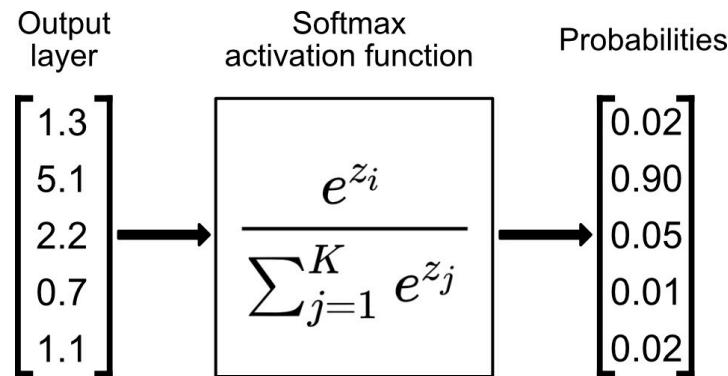
Fonte: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>

Camada Totalmente Conectada

- Após o achatamento dos valores em um vetor, este alimentará uma camada totalmente conectada. O papel desta camada é produzir uma **lista de notas de classes** e realizar a classificação baseado nas características que foram extraídas das camadas anteriores.

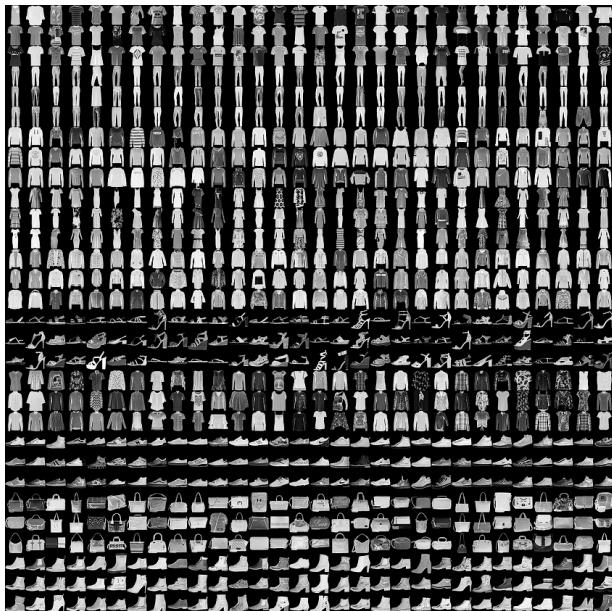
Camada Softmax

- Por fim, a ultima camada utiliza uma **função de ativação softmax**
- Transforma a lista de notas da camada anterior e probabilidades



Fonte: <https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60>

Jupyter Notebook: Classificar FashionMNIST



Fonte: <https://github.com/zalandoresearch/fashion-mnist>

Muito obrigado!



Inteligência Artificial ao alcance de todos.

LinkedIn: Lucas Takara

www.escolalivre-ia.com.br