SAMSUNG

NOVA EXECUTIVE EDUCATION
NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Samsung Innovation Campus

Chapter 9. Deep Learning Module

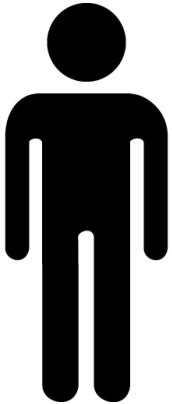
Chapter 9.

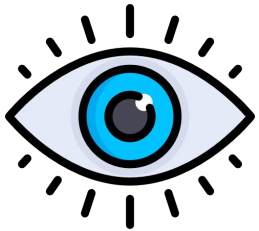
Convolutional Neural Networks

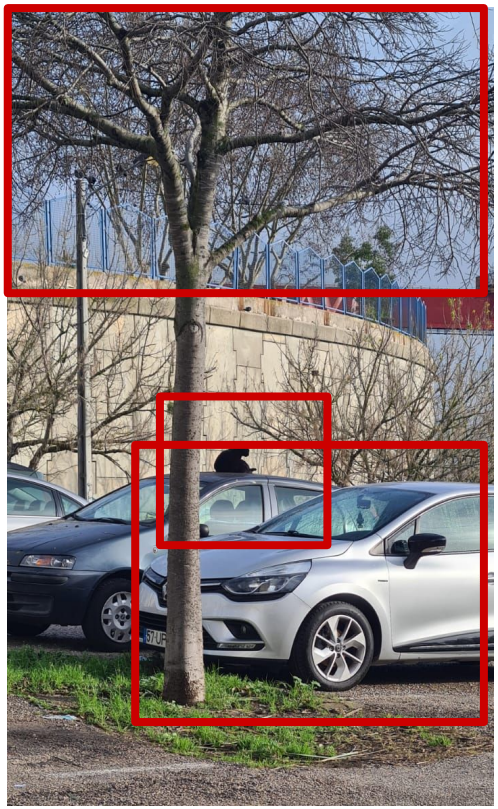
… or **CNNs**, for short!

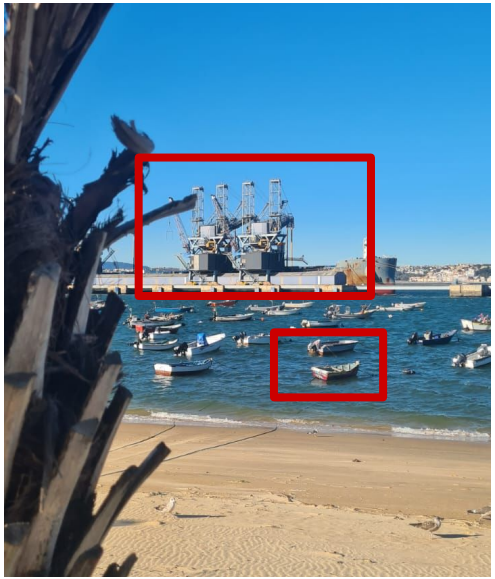
Let’s start from the beginning…

Vision









Let’s start from the beginning…

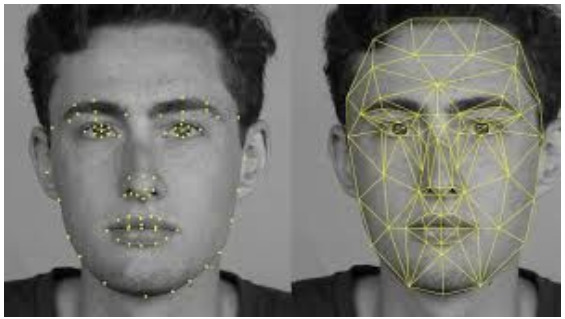
Computer Vision

Computer Vision: The Basics

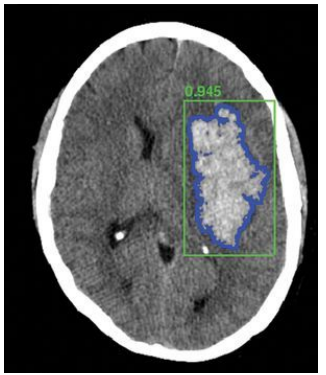
A field of CS that focuses on how computers can derive meaningful information from visual inputs (e.g., images, videos)

Self-driving cars

Facial Recognition

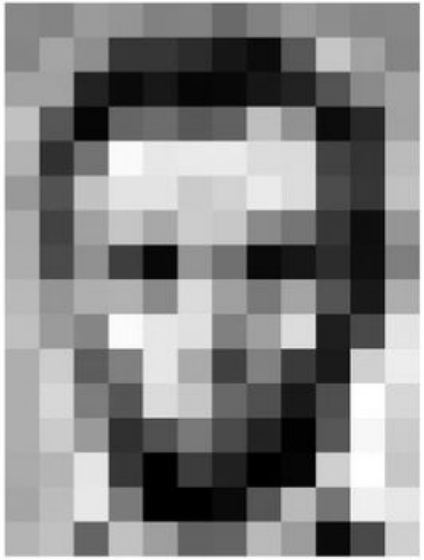


Biomedical Imagery



1 - IBM; 2 - American Journal of Neuroradiology

Computer Vision: Images are numbers



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

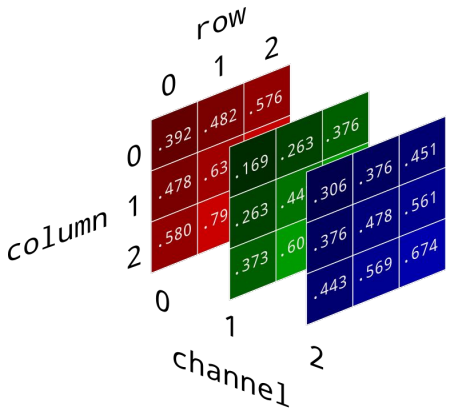
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An image is just a **matrix of numbers** [0,255]
(1080x1080x3 for an **RGB image**)

Computer Vision: Tasks

- **Regression:** predict a continuous value.
- **Classification:** output corresponds to the class label. Can produce the probability of belonging to a particular class.

Input image



Pixel Representation

Marcelo R. de Sousa	0.75
Aníbal C. Silva	0.05
Jorge Sampaio	0.10
Mário Soares	0.10

1 - MIT 6.S191; 2 - Pixels, Arrays, and Images

Classification: Detecting Patterns!

1 - MIT 6.S191;

Classification Pipeline

We have two **major steps** when building a classifier:

1. What patterns **(features)** are we looking for?
2. How are we going to **detect** these patterns?

Invariant

Sensitive

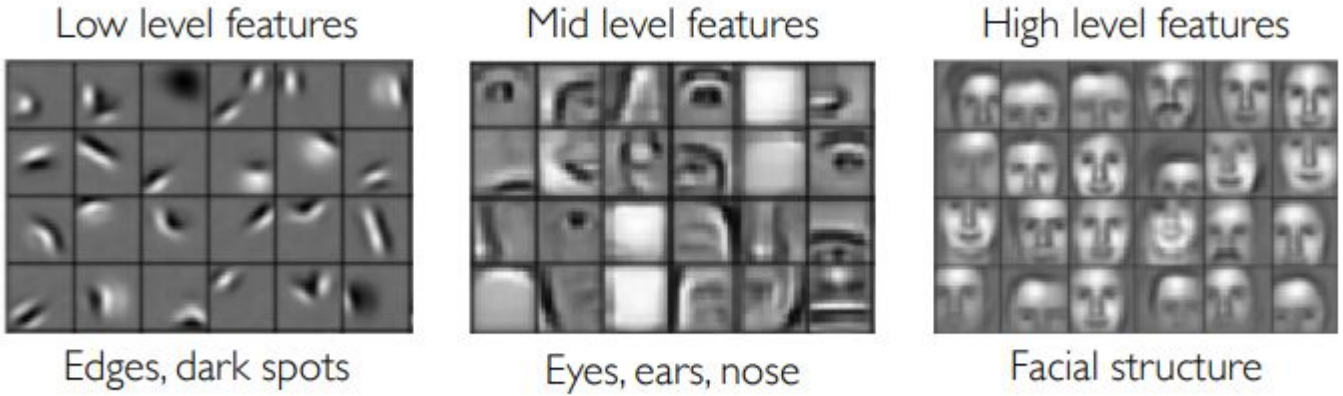
- Background
- Viewpoint
- Occlusion
- Rotations
- Scale variation
- Lighting
- Intra-class variation

Our system needs to be able to handle all of this!

1 - MIT 6.S191;

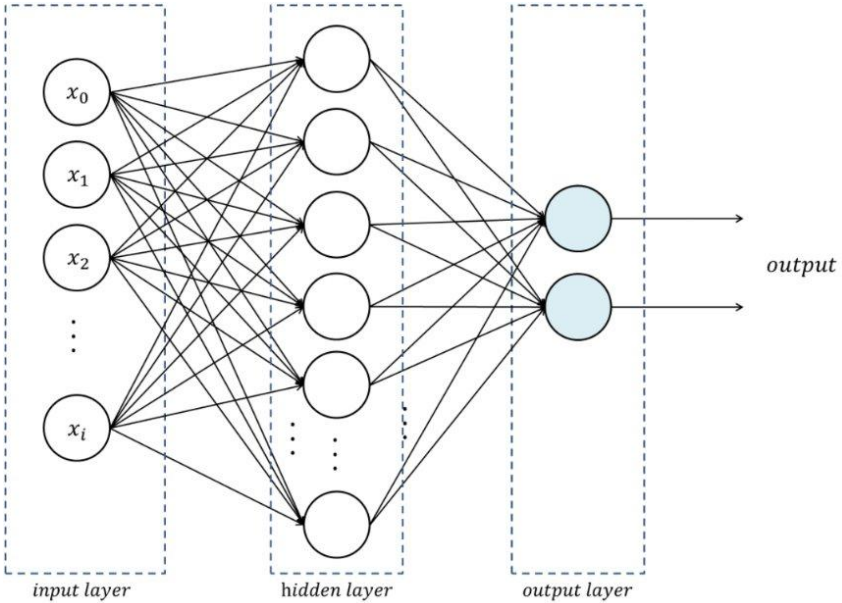
What do we need?

We want a CV algorithm that is capable of **automatically** extract the features in a **hierarchical-fashion**



1 - MIT 6.S191;

Traditional Neural Network



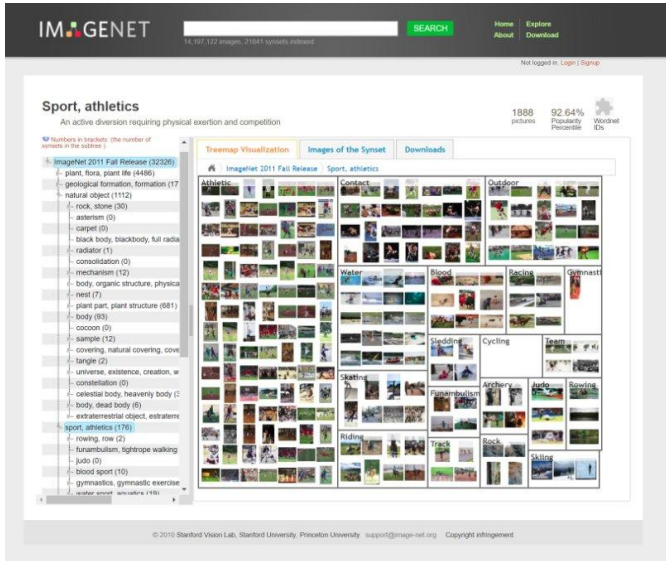
Problems:

- Receives input features
- **Fully connected** structure
- No spatial information preservation
- Number of parameters **explodes** with image size
- Does not account for **translational invariance** (the same object in a different position should be recognized the same way)

1 - MIT 6.S191;

A bit of history: ImageNet Competition

Annual competition for classifying images into **1000 categories**.



A total of **1.43 million** annotated images

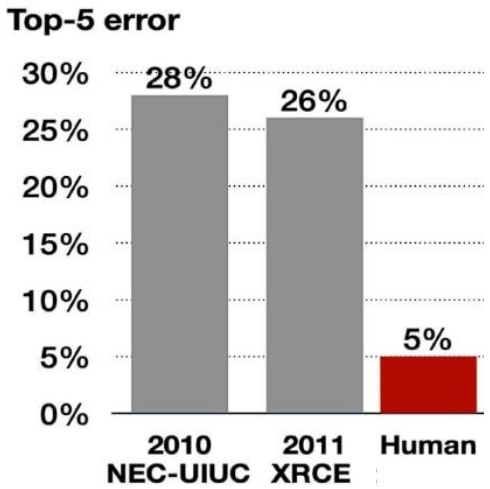
Goal: Annotate a test sample and be as accurate as possible

Instrumental in advancing the State-of-the-Art

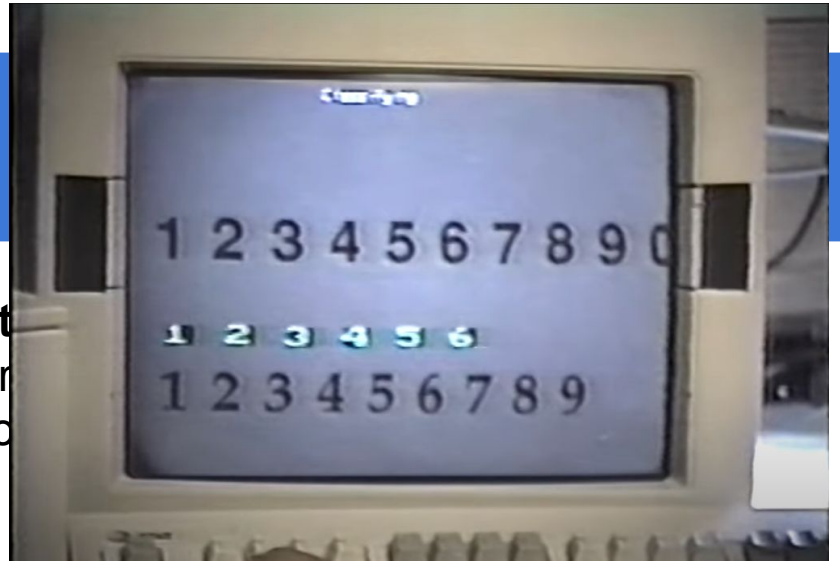
1 - ImageNet;

A bit of history: ImageNet Competition

Before 2012, traditional image classification approaches were based on **hand-crafted feature extraction** and **shallow neural networks**.

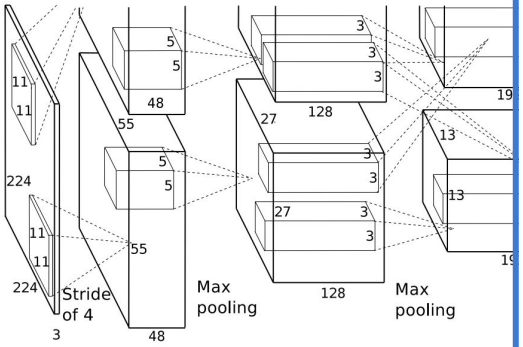


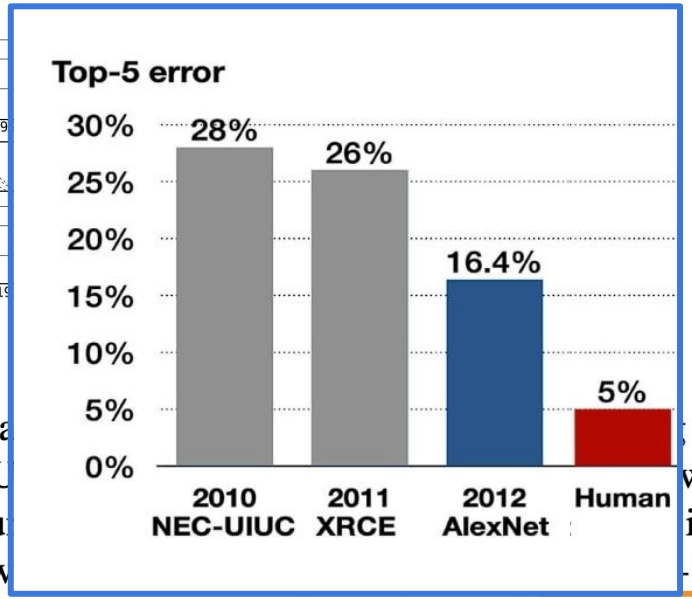
Aut
lear
thro



1 - ImageNet; 2 - CNNs in 1989

A bit of history: The 2012 Deep Learning Revolution





8 layers

Trained in the ImageNet dataset

Figure 2: An illustration of the a ; the delineation of responsibilities between the two GPUs. One GPU while the other runs the layer-parts at the bottom. The GPUs commu input is 150,528-dimensional, and the number of neurons in the netw -186,624–64,896–64,896–43,264–4096–4096–1000.

1 - AlexNet Paper;

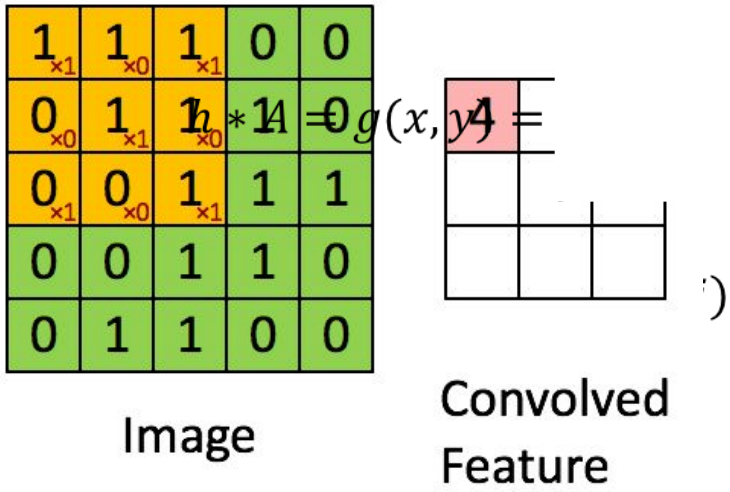
Convolutional Neural Networks

How do these models extract features directly from the input data?

Convolution

1. Applies a filter **(kernel)** to the input image by sliding it across the image
2. Processes small patches of the image by applying a **convolution operation**
3. Obtains a **convolved feature map**

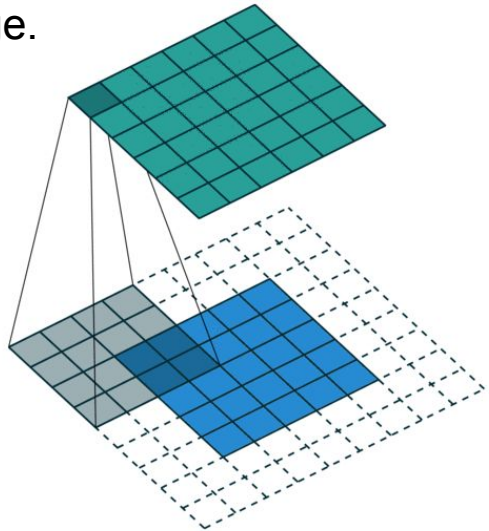
$$h(i,j) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$



Convolution: Basic Concepts

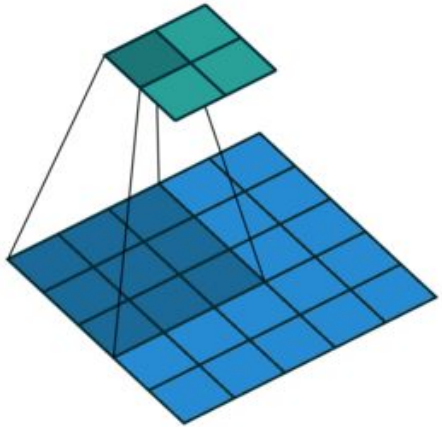
Padding

Extends the image with added **zeros*** to let the convolution center run over the image.



Stride

Defines the number of pixels the convolution moves in each direction.



1 - Based on WiSDoM slides by Professor João Magalhães.

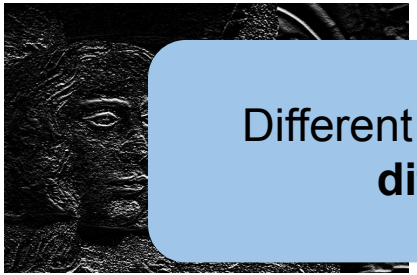
Exercise Time

Types of Convolution Filters

High-pass: Used to emphasize edges and fine details in the image

horizontal filter

$$h_h(i,j) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



vertical filter

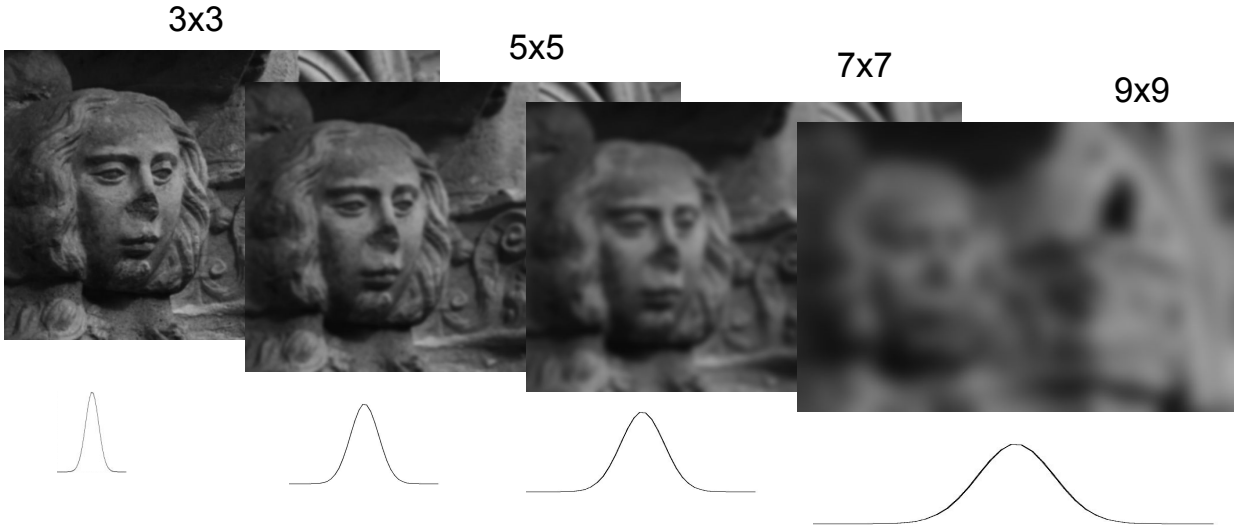
$$h_v(i,j) = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Different kernels are used to detect the edges at **different scales and orientations**

1 - Based on WiSDoM slides by Professor João Magalhães.

Types of Convolution Filters

Low-pass: Used to remove noise: blurs the image and removes edges

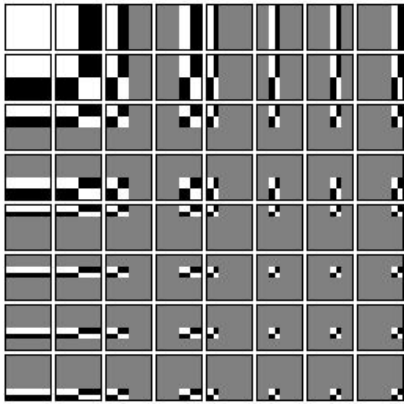


gaussian kernel

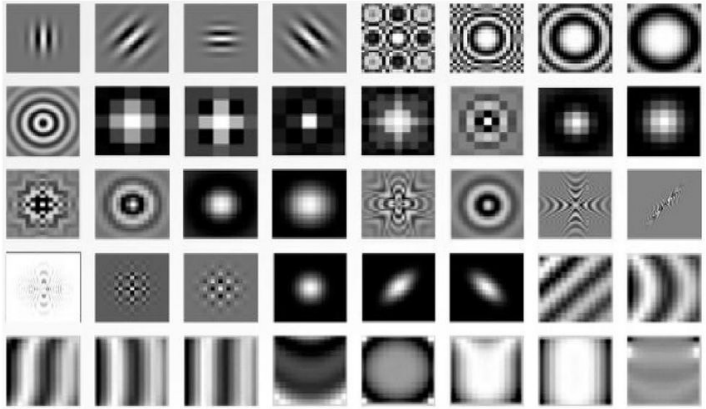
$$h_0(x,y) = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

1 - Based on WiSDoM slides by Professor João Magalhães.

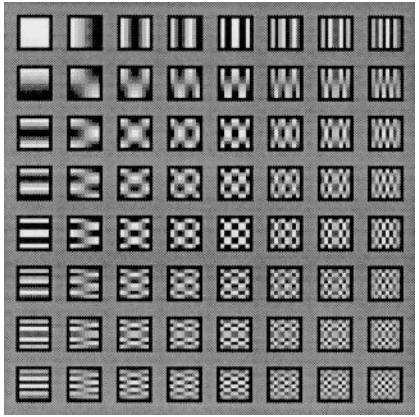
Kernels: The diversity!



Haar - edge detection



Gabor - texture analysis



DCT - compression

Instead of hand-crafting these kernels... CNNs can learn them automatically from data!

1 - Gabor; 2 - Based on WiSDoM slides by Professor João Magalhães.

What makes CNNs special?

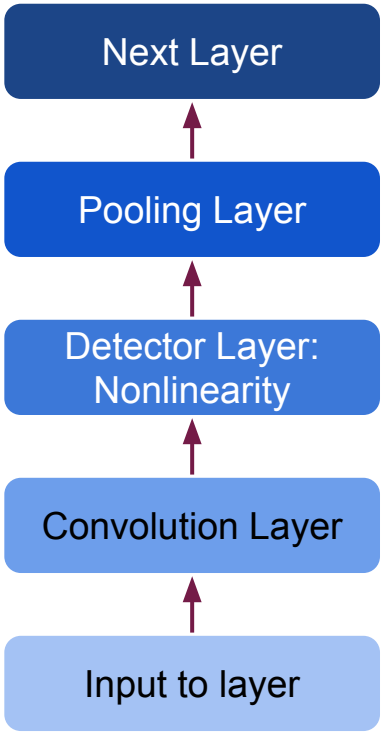
Problems:

- ~~Receives input features~~
- ~~Fully connected structure~~
- ~~No spatial information preservation~~
- ~~Number of parameters explodes with image size~~
- ~~Does not account for translational invariance (the same object in a different position should be recognized the same way)~~

1. **Efficient processing of Large Images:**
   - **Sparse connectivity**
   - **Parameter Sharing**
2. **The kernels are learned during training**
3. **Built-in Spatial Understanding:**
   - **Preserves spatial relationships**
4. **Works with any grid-structured data (1D, 2D, 3D)**

1 - Based on WiSDoM slides by Professor João Magalhães.

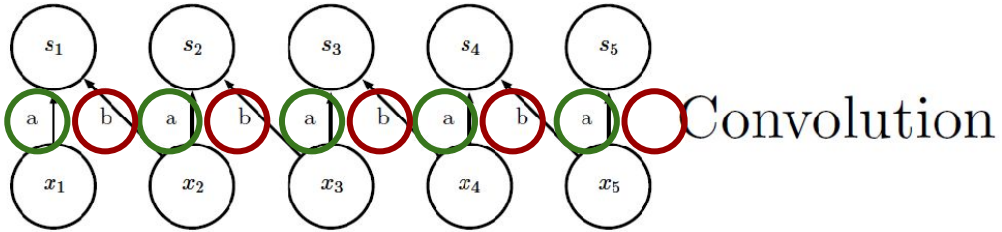
CNNs: The architecture



1. **Input to Layer:** An image!

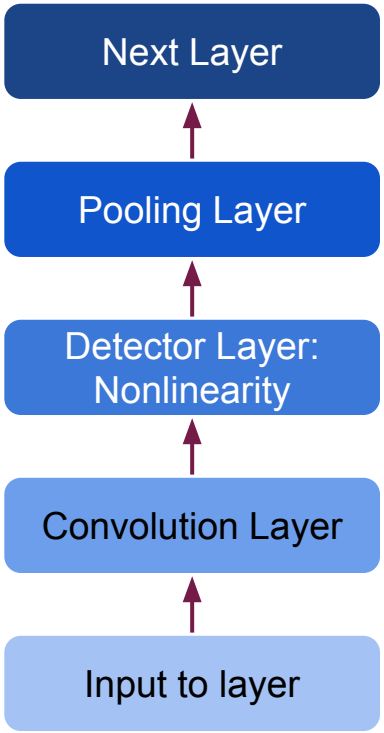
2. **Convolution Layer:** We obtain the convolved feature map.

The Secret:



1 - Based on WiSDoM slides by Professor João Magalhães.

CNNs: The architecture

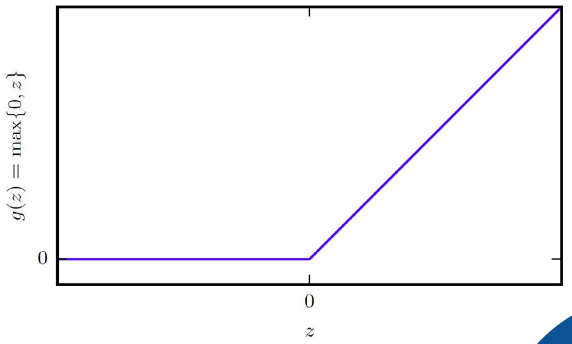


3. Detector Layer: Applies an **activation function** so the CNN can learn more complex patterns.

Common activation functions:

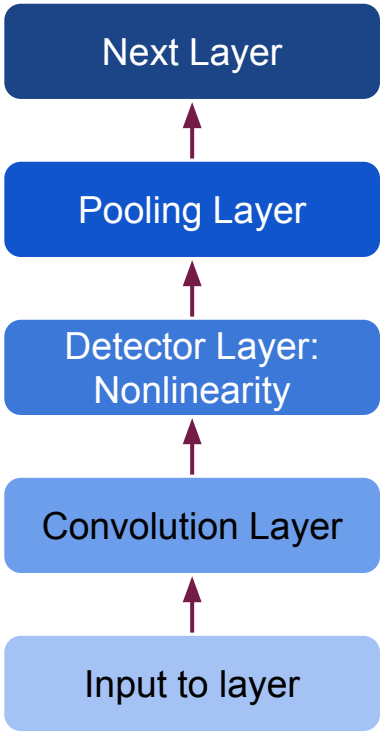
1. **ReLU**
2. Sigmoid
3. Tanh

$$g(z) = \max\{0, z\}$$



1 - Based on WiSDoM slides by Professor João Magalhães.

CNNs: The architecture

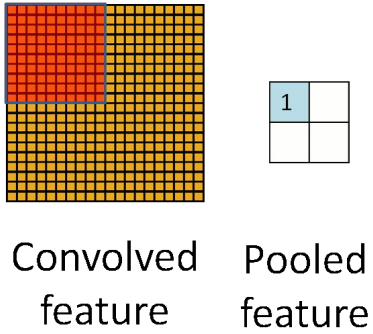


4. Pooling Layer: Reduce the spatial dimensions of convolved feature maps while retaining important information.

- Downsampling
- Feature Extraction
- Translation Invariant

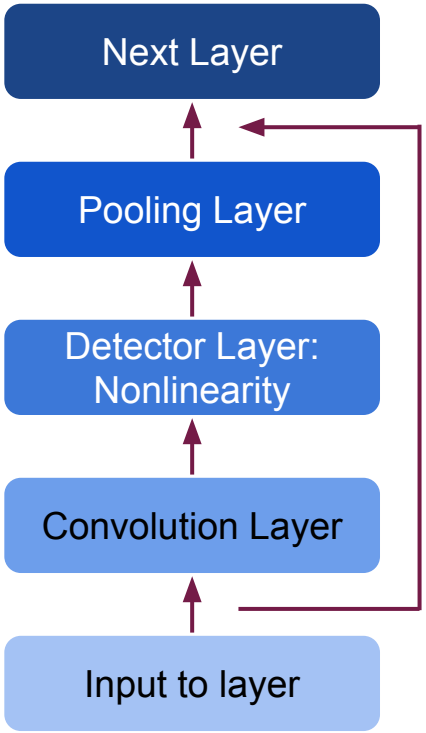
Pooling Strategies:

- Max pooling
- Min pooling
- Mean pooling

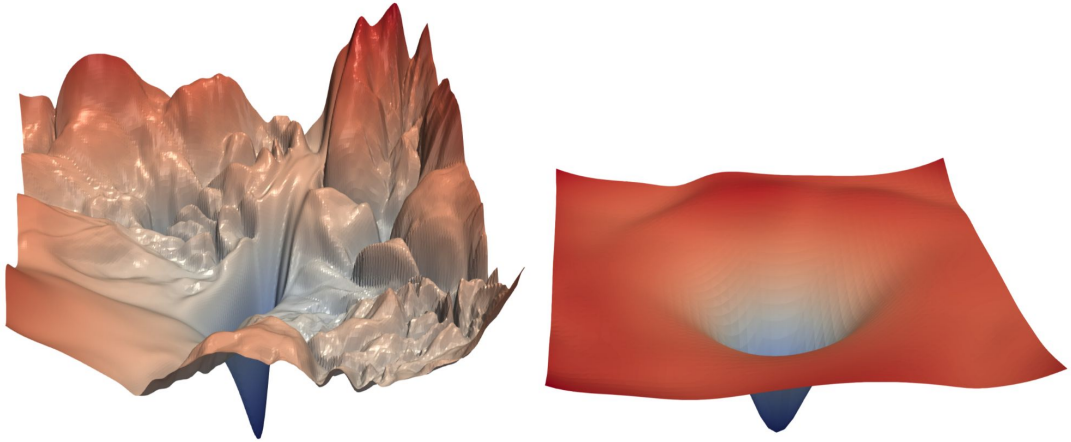


1 - Based on WiSDoM slides by Professor João Magalhães.

CNNs: The architecture



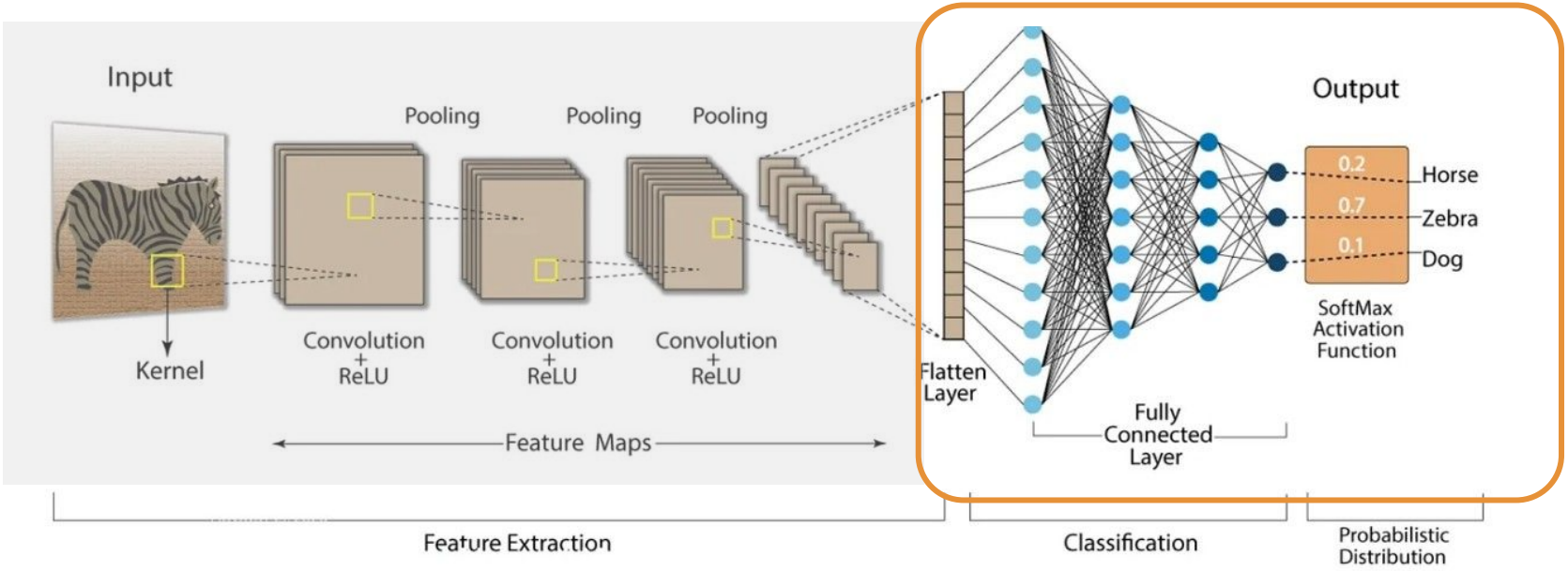
Residual Connection: Adds the output of a previous layer to the input of another layer.



(a) without skip connections (b) with skip connections

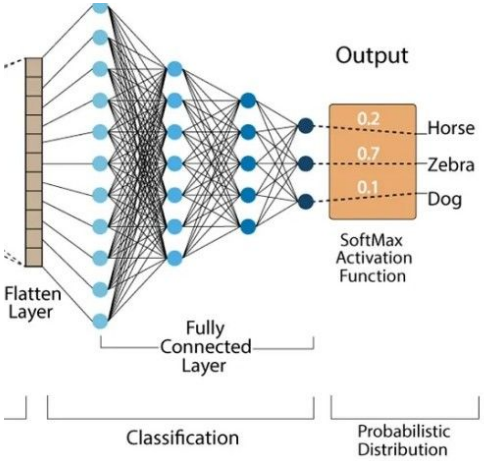
1 - Based on WiSDoM slides by Professor João Magalhães; 2- Deep Residual Learning for Image Recognition

CNNs: The overview



1 - Based on WiSDoM slides by Professor João Magalhães; 2- CNN Overview

CNNs: Linear Layer and Softmax Layer

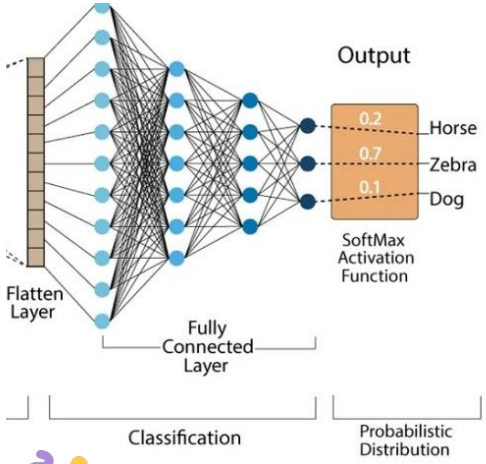


1. The output of the last pooling layer is **flattened** into a vector.
2. This vector is then fed into **fully connected** linear layer.
3. The output of the linear layer is fed into a **softmax function**.

$$\text{softmax}([z_1, z_2, \ldots, z_n]) = \left[\frac{e^{z_1}}{\sum_i e^{z_i}}, \frac{e^{z_2}}{\sum_i e^{z_i}}, \ldots \frac{e^{z_n}}{\sum_i e^{z_i}}, \right]$$

1 - Based on WiSDoM slides by Professor João Magalhães; 2- CNN Overview

CNNs: Classification



The softmax returns a **probability distribution** over all possible categories.

$$\text{softmax}([dog, cat, \dots, bird]) = \left[\frac{e^{dog}}{\sum_i e^{z_i}}, \frac{e^{cat}}{\sum_i e^{z_i}}, \dots \frac{e^{bird}}{\sum_i e^{z_n}},\right]$$

The result would be something like this…

Marcelo R. de Sousa	0.75
Aníbal C. Silva	0.05
Jorge Sampaio	0.10
Mário Soares	0.10

Congrats! We (finally) have a classification!

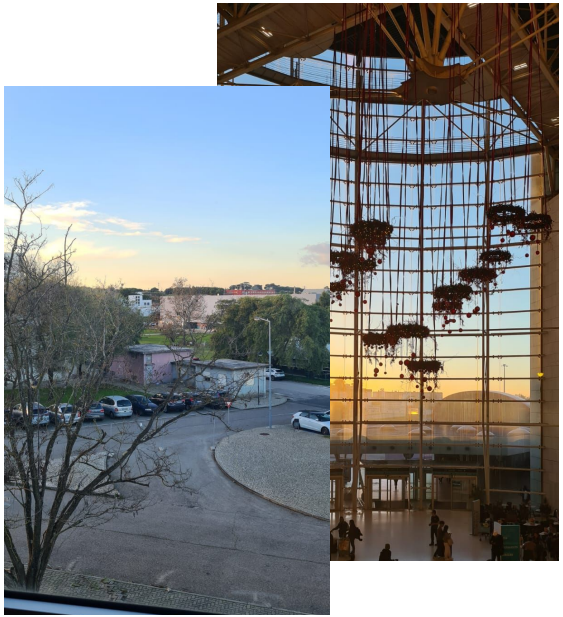
1 - Based on WiSDoM slides by Professor João Magalhães; 2- CNN Overview

Exercise Time

Chapter 9.

Visual Search

Visual Search: What? Why?



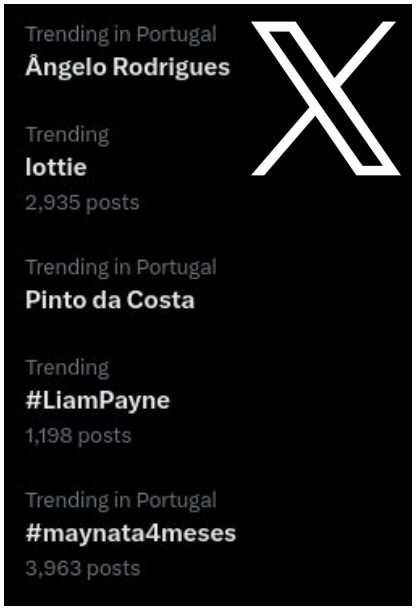
Photographs

TV Shows/Movies

1 - Based on WiSDoM slides by Professor João Magalhães

Visual Search: Types

Keyword



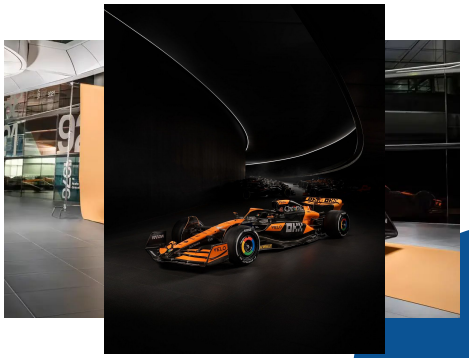
Text

Captioning Model

Picture of the 25th of April bridge, in Lisbon, on a rainy day.

Similarity

User Query



1 - Based on WiSDoM slides by Professor João Magalhães

Visual Search: Similarity Search

Similarity search involves finding images in a database that are similar to a given image. Okay, we got that. But…

what do CNNs have to do with this?

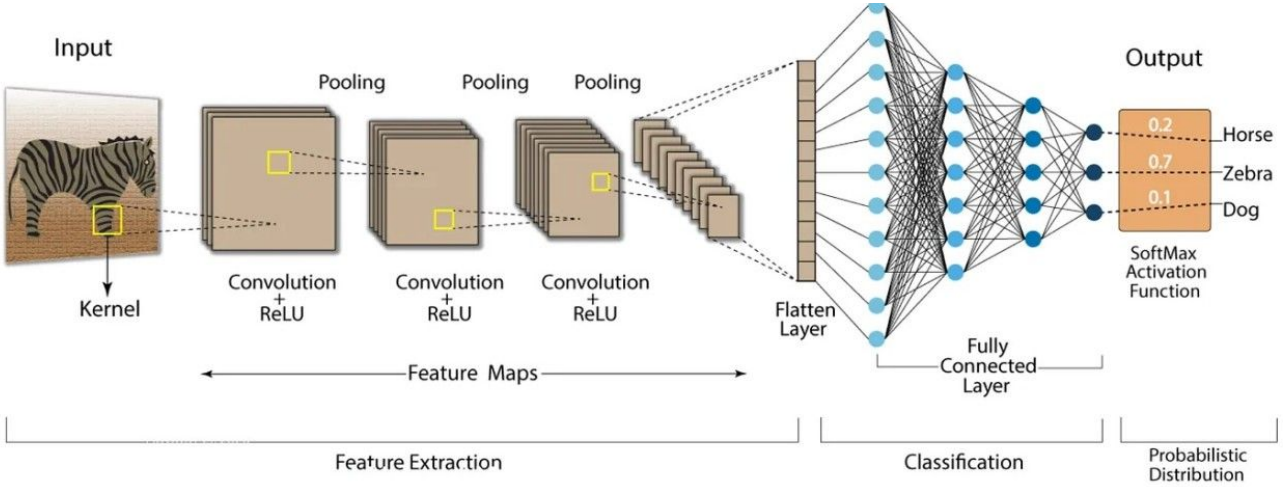
How do we find **similar images**?

Obviously not!

We (can) use CNNs!

Visual Search & CNNs

CNNs can be leveraged as **feature extractors:** These models transform images in a high-dimensional representation - a **feature vector** - that **encodes the image's characteristics**.



1 - CNN Overview

Visual Search: Similarity Metrics

How do we find **similar images**?

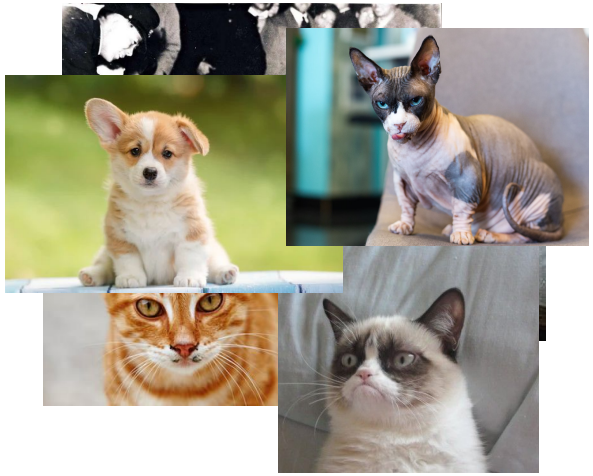
Okay we use feature vectors… but **how?**

We compare them using **similarity/distance metrics**

$$\text{Cosine Similarity}(A, B) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\|\|\vec{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

$$\text{Euclidean Distance}(A, B) = d(A, B) = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}$$

Visual Search: How it (usually) works



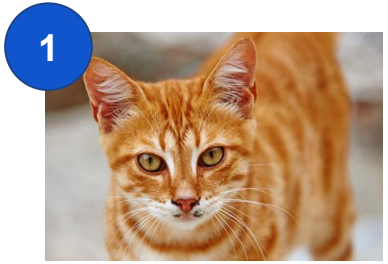
CNN

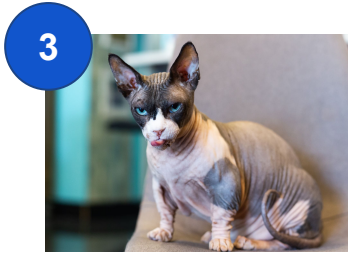
Ima ..., 0.504, 0.507]
Ima ..., 0.870, 0.008]
Ima ..., 0.321, 0.278]
Ima ..., 0.841, 0.313]
..

Query_Image: [0.253, 0.518, ..., 0.321, 0.278]

Visual Search: How it (usually) works

The results are going to be **ordered by similarity!** (from highest to lowest)





...

Congrats! We (finally) have a retrieval result!

Exercise Time