# scai

SORBONNE CLUSTER FOR
ARTIFICIAL INTELLIGENCE

# Computer Vision: Understanding How Models See and Generate Images

**Raphael Cousin**
**Data Scientist**
raphael.cousin@sorbonne-universite.fr

raphaelcousin.com

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Plan

**Image Data Representation:**
- Definition
- Application
- History
- Context

***Duration : ~10 min***

**Essential Layers:**
- MLP
- CNN
- Attention

***Durée : ~10 min***

Applications:
- Classification
- Object Detection
- Segmentation
- Generation

***Duration : ~30 min***

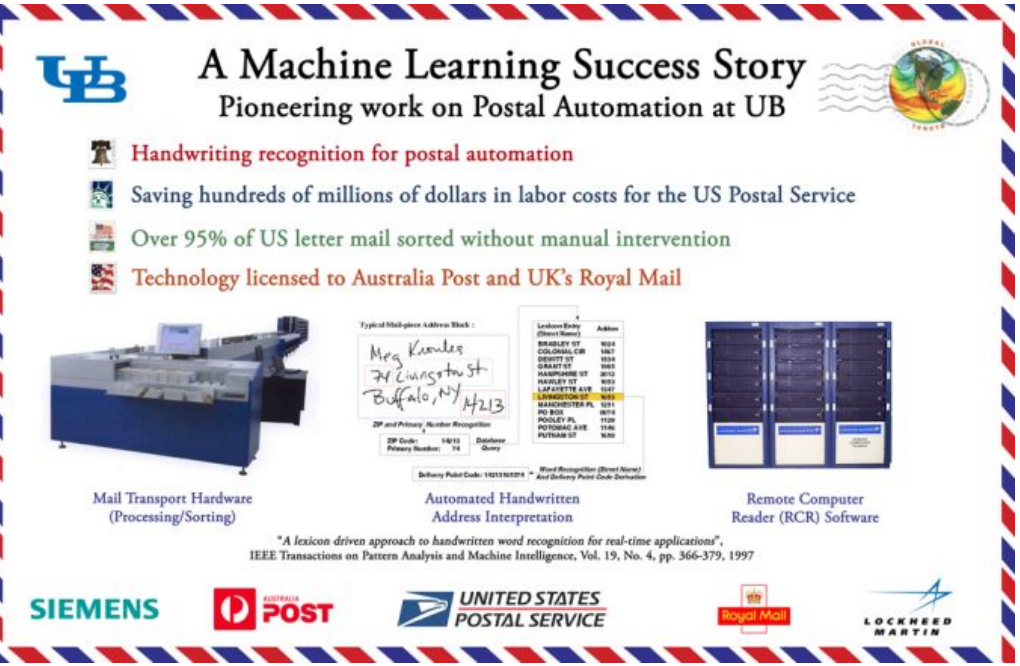
Ressources
- Fine tuning
- Labellisation

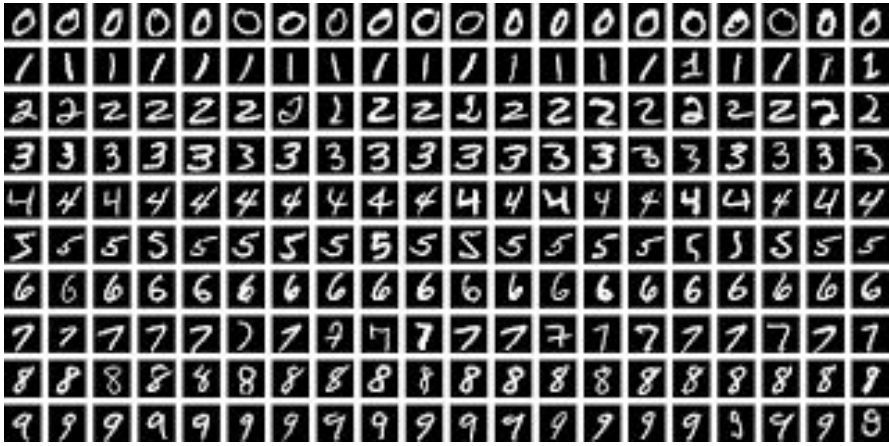
***Duration : ~10 min***

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Brief Introduction

# Optical Character Recognition (OCR)

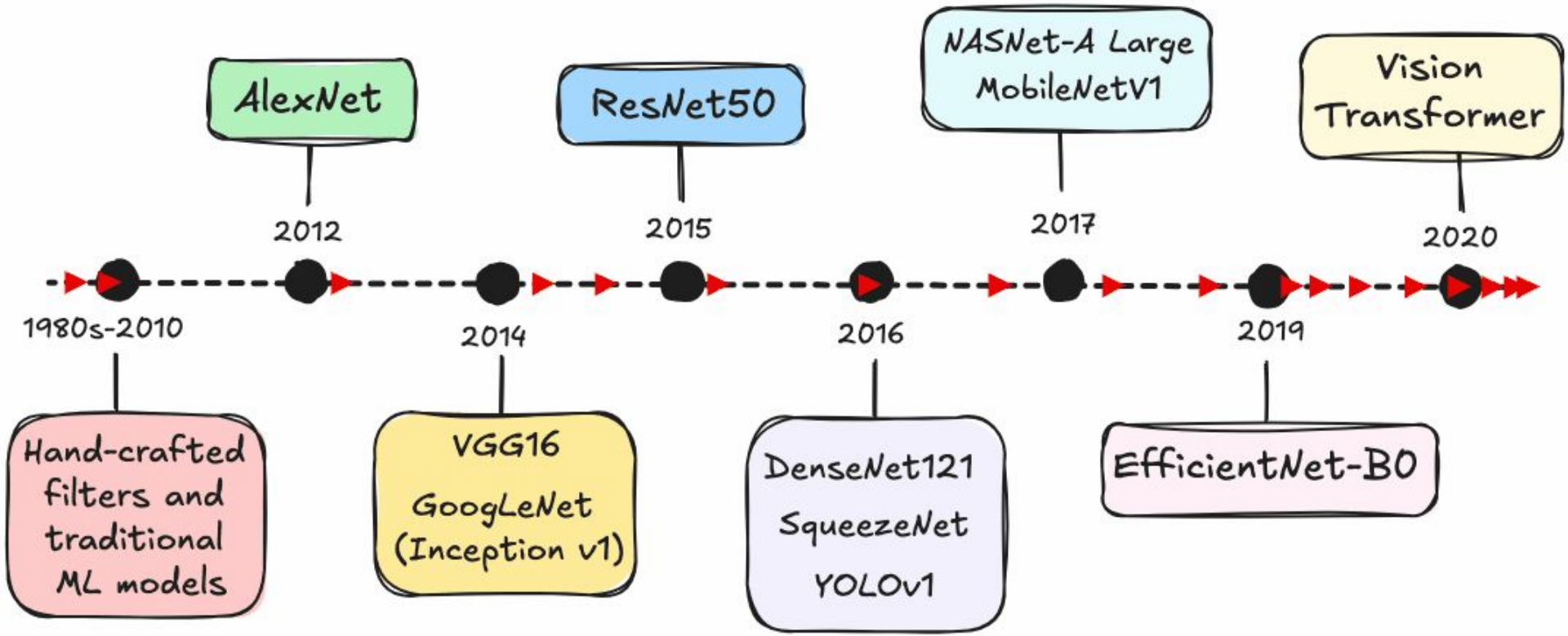


MNIST dataset contains 70,000 handwritten digits (0-9) as 28×28 grayscale images. It Simple, clean, and now essentially solved (>99.7% accuracy).

LeNet-5 (Yann LeCun 1998) deployed by the US Postal Service to automatically read handwritten ZIP codes on mail.

# History



# Image Data Representation

# Spatial and Hierarchical Nature of Images

**Local patterns form features**

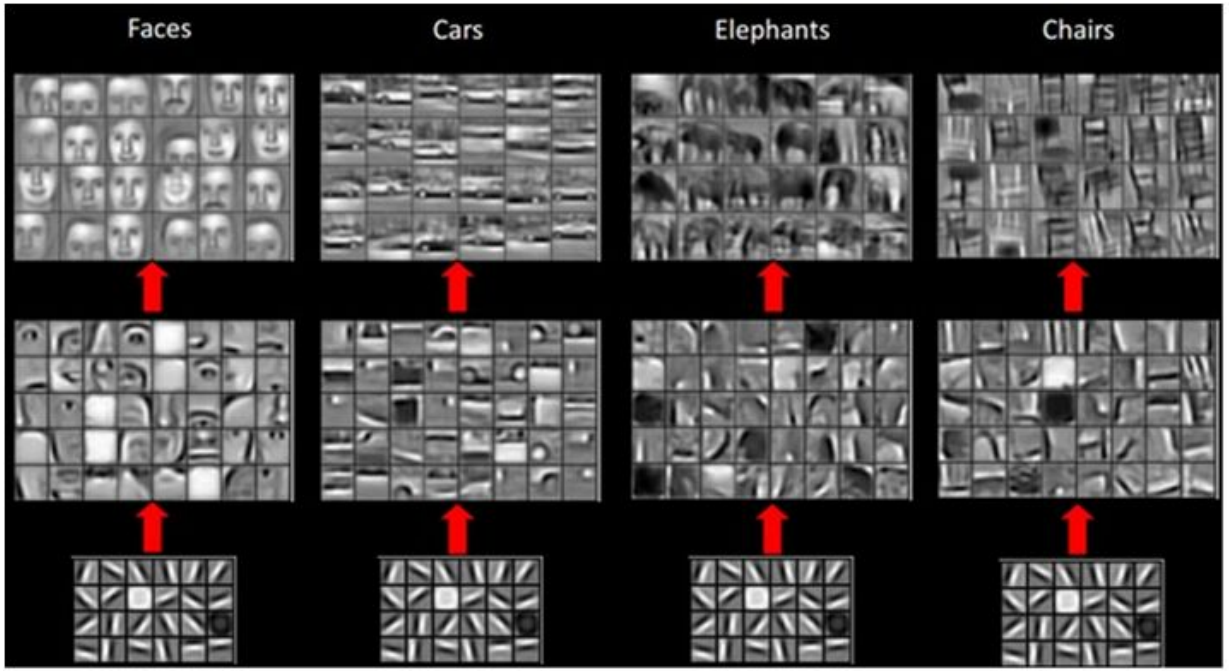
Edges, textures that combine into objects and scenes

**Spatial relationships matter**

A nose above a mouth signals a face, not random parts

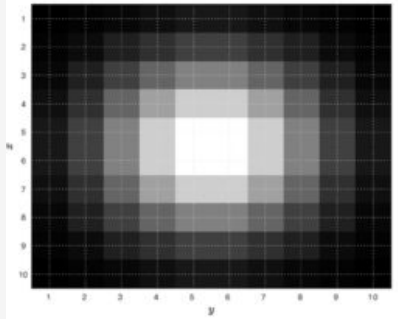
**Context is scale-dependent**

A patch of pixels might be fur, grass, or noise depending on surrounding regions

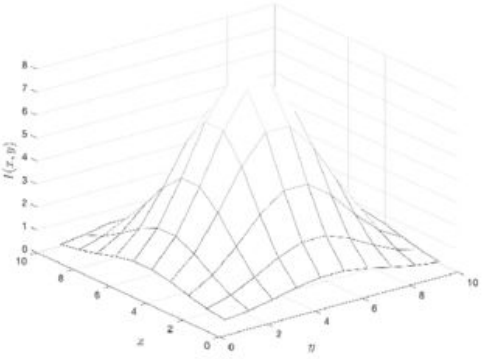


# Numerical Image Representation

A digital image is a **structured grid of numerical values**. Each point in this grid, called a pixel, contains information about color intensity.



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 0 |
| 0 | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 0 |
| 1 | 2 | 3 | 5 | 6 | 6 | 5 | 3 | 2 | 1 |
| 1 | 2 | 4 | 6 | 8 | 8 | 6 | 4 | 2 | 1 |
| 1 | 2 | 4 | 6 | 8 | 8 | 6 | 4 | 2 | 1 |
| 1 | 2 | 3 | 5 | 6 | 6 | 5 | 3 | 2 | 1 |
| 0 | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |



Single channel grayscale image: each pixel represents **intensity from 0 (black) to max value 8 (white)**

# Numerical Image Representation

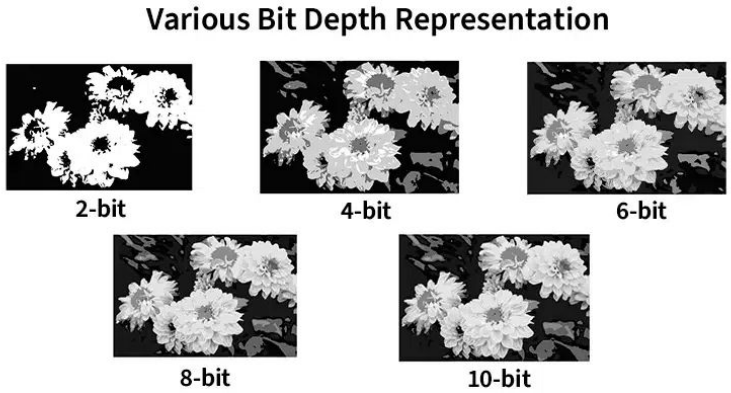
**Color Depth:**

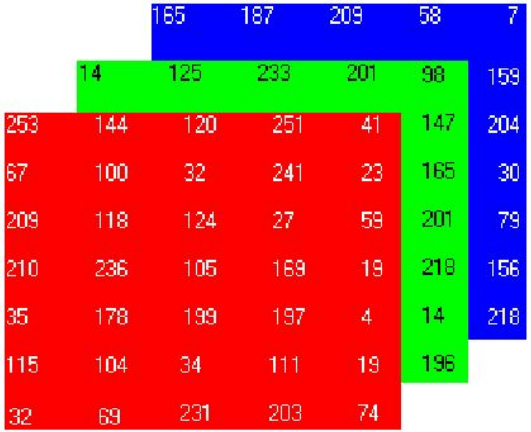
Bit depth determines the range of possible values:

2-bit: 2 levels (0-1)
8-bit: 256 levels (0-255)
16-bit: 65,536 levels (0-65535)

**Color Channels:**

RGB images consist of three channels, each representing the intensity of Red, Green, and Blue components (usually on 8-bit)



# Extended Channel Applications

**A General image is represented** as a

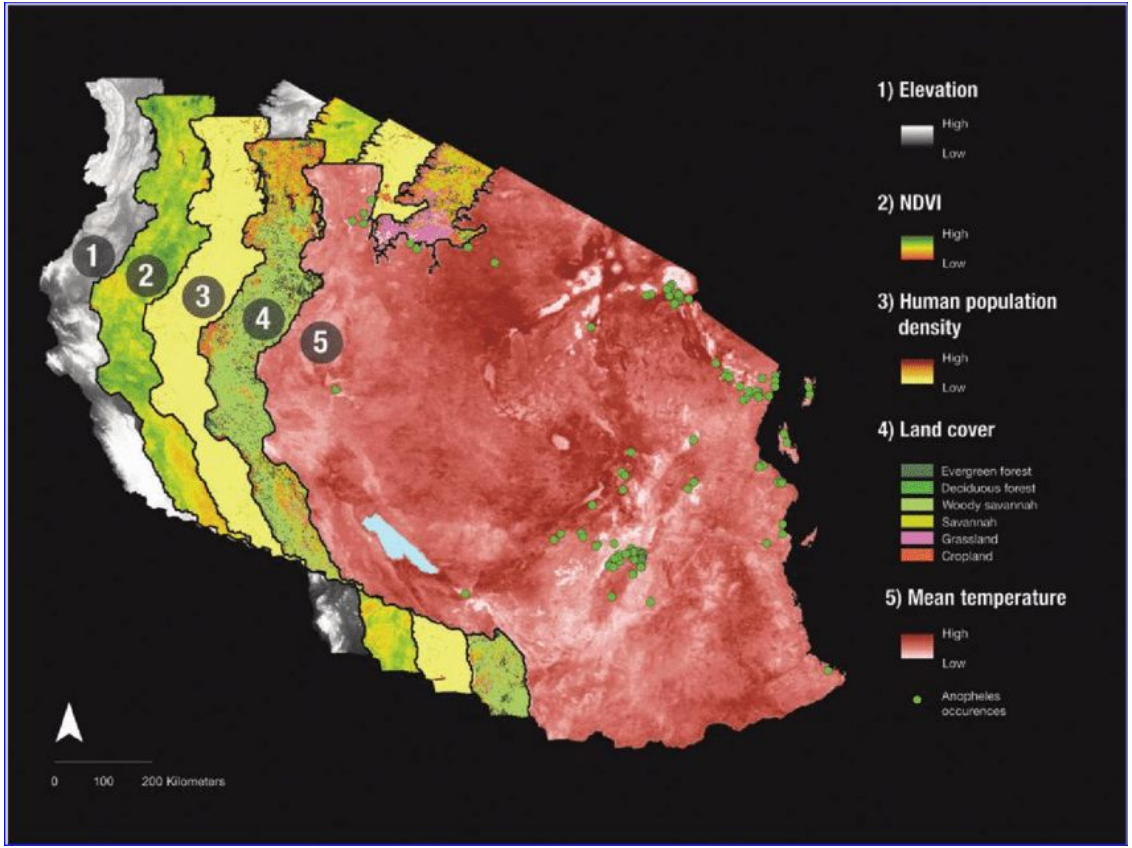
H × W × C tensor

H: Height in pixels

W: Width in pixels

C: Number of channels

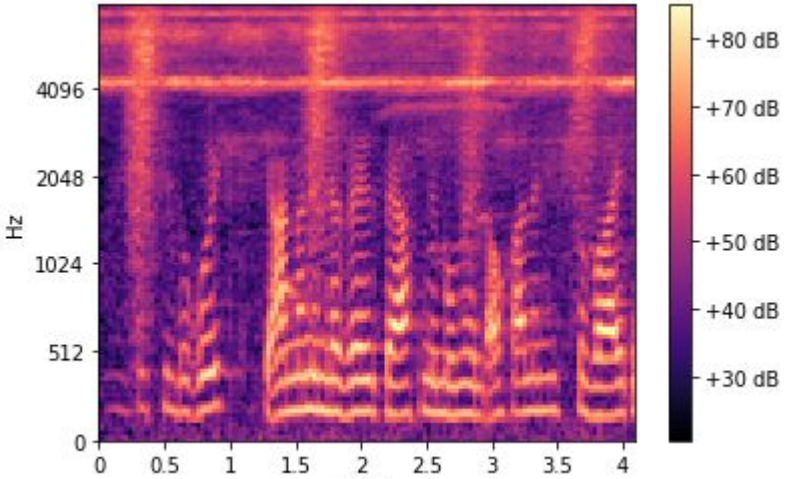
$$I \in \mathbb{R}^{H \times W \times C}$$



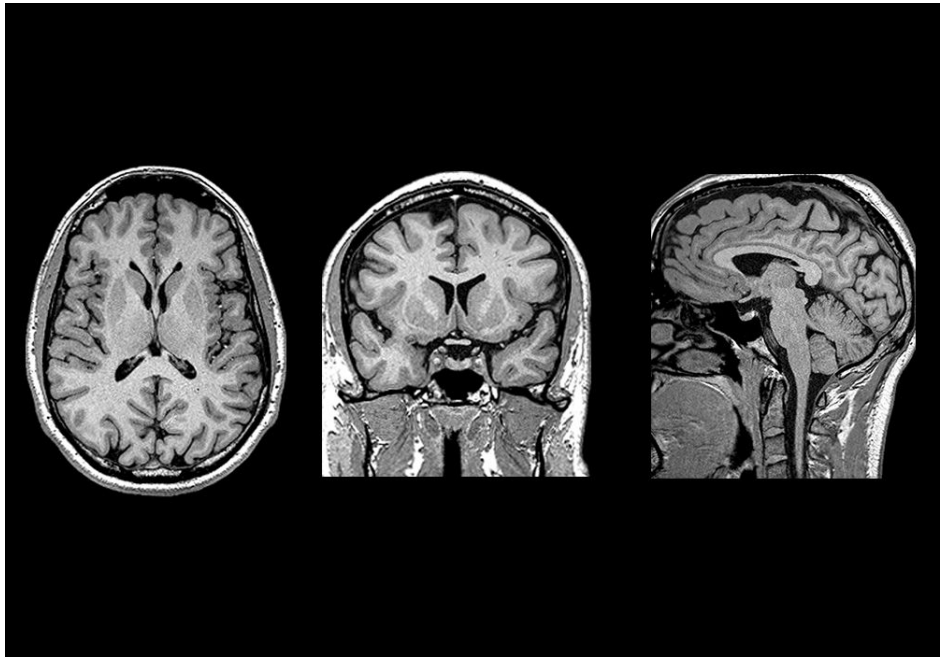

Satellite imagery: Near Infrared, Thermal, Altitude, etc.
Medical imaging: Different sensor readings

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Extended Channel Applications




**Signal processing**



**3D structure in medical imaging (MRI, ...)**

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Essential Layers

# MLP Fails with images

**No spatial awareness:**

Treating pixels as independent features

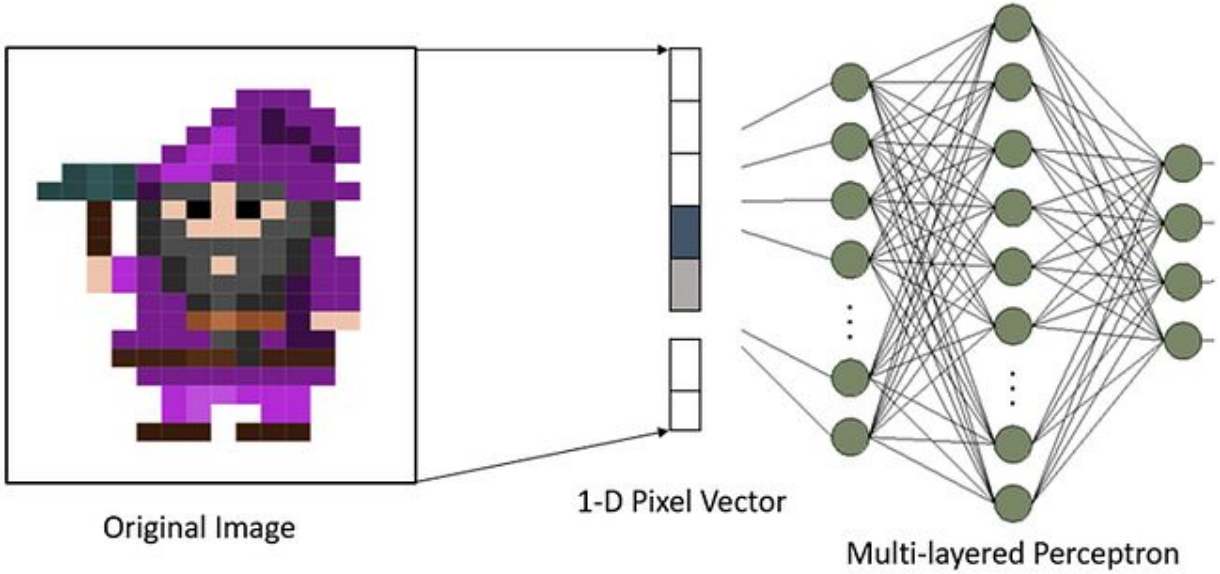
**Fixed input size:**

Can't generalize across resolutions

**No translation invariance:**

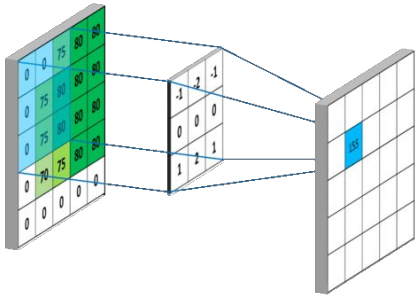
A cat in the top-left vs bottom-right looks completely different to an MLP

**Parameter explosion**

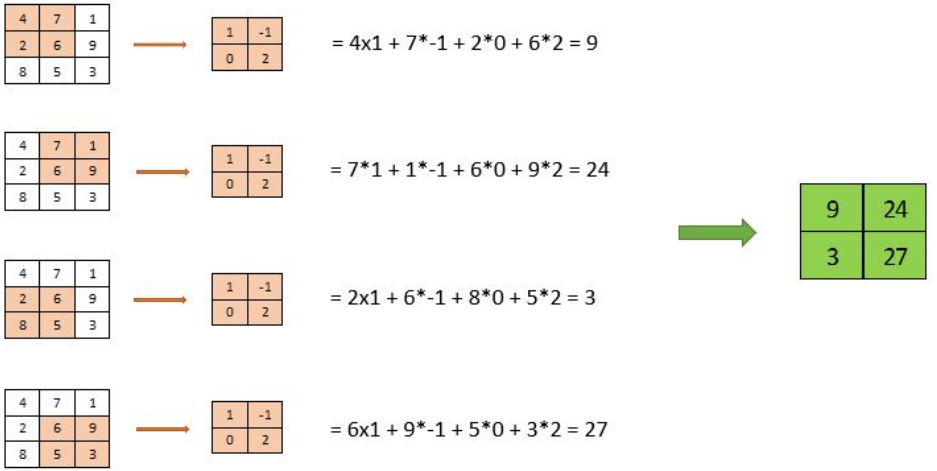


# Convolution

A convolution operation **slides a kernel** (small matrix) across the image, computing weighted sums of local regions.



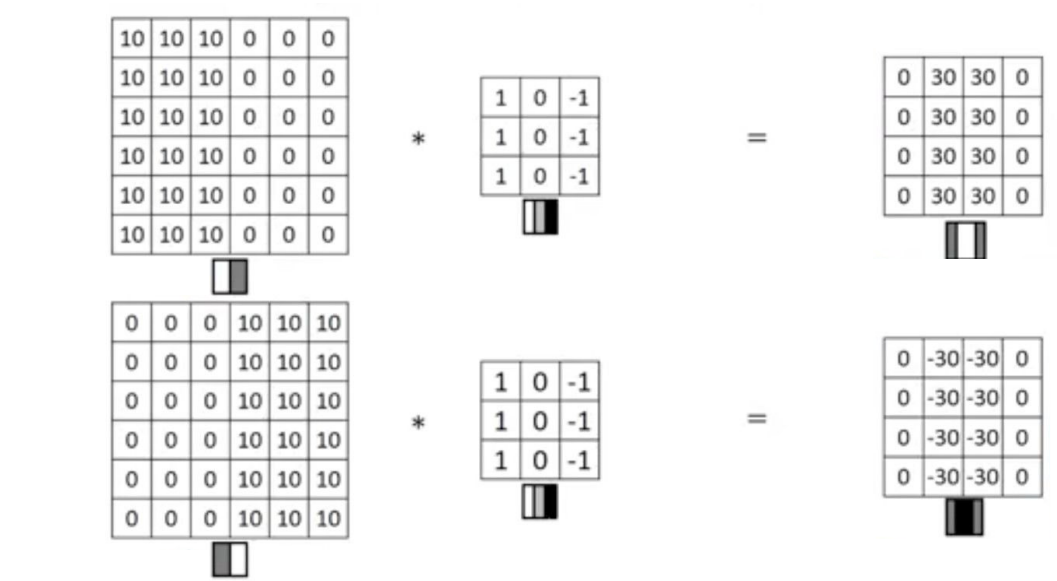
**Sliding kernel**

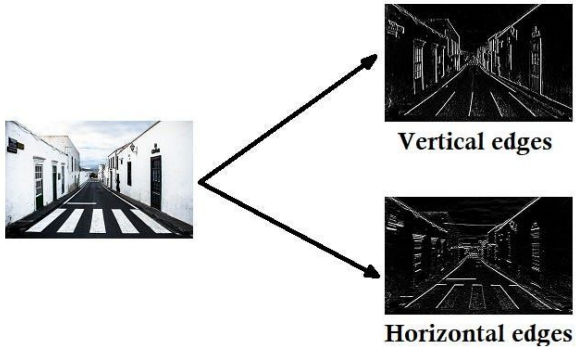


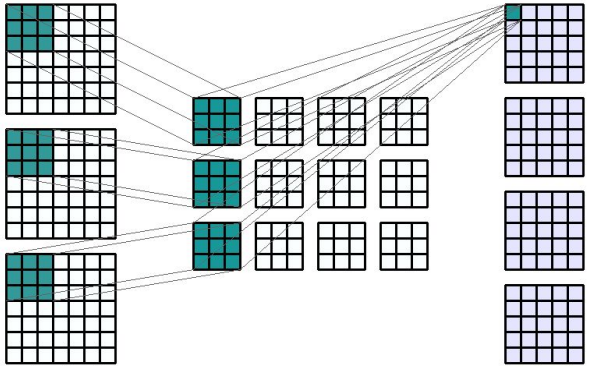
# Feature maps and learned representations

Different kernels produce different feature maps from the same image.

Convolution Layer. Let's just learn the kernel/filters!



# Convolutional Layers

1. **Preserve spatial relationships between pixels**

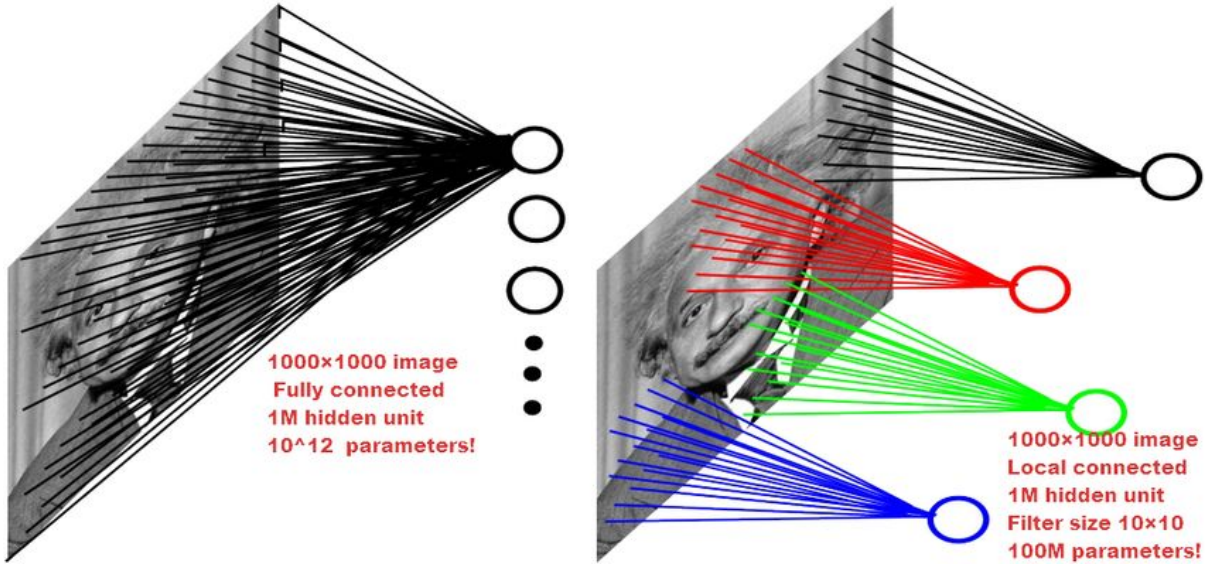
2. **Use much less Parameters**
Consider an image of size 224×224×3. A fully connected layer would require in first layer:

$$224 \times 224 \times 3 \times N_{neurons} = 150,528 \times N_{neurons}$$

In contrast, a convolutional layer with 64 filters/Channels of size 3×3 only needs in first layer:

$$3 \times 3 \times 3 \times 64 = 1,728$$

parameters, while maintaining the ability to detect features anywhere in the image.




scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Vision attention

**Image patching:**

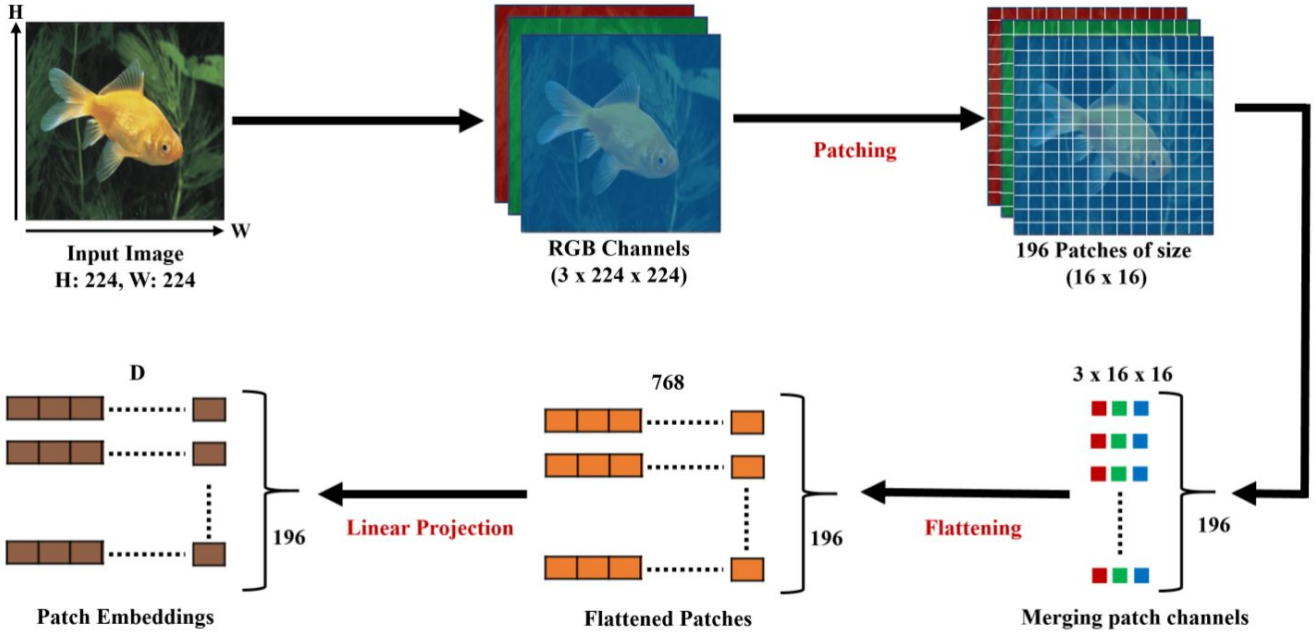
Treat your image as a sequence of patches

Attention can model global relationships: patch 1 can attend to patch 196 directly (no locality constraint like CNNs)

**Trade-off**

- Loses fine-grained pixel relationships within patches

- Quadratic complexity in number of patches: 196 patches → $196^2$ attention computations



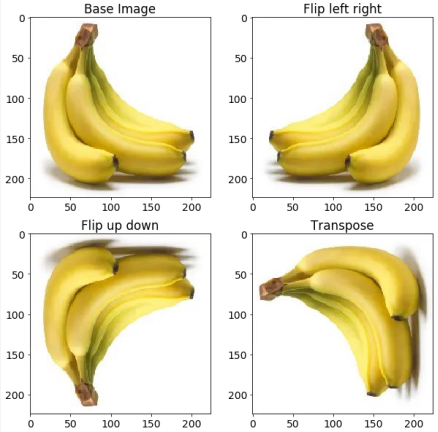

**2020 – Vision Transformer (ViT)**

# Data Augmentation

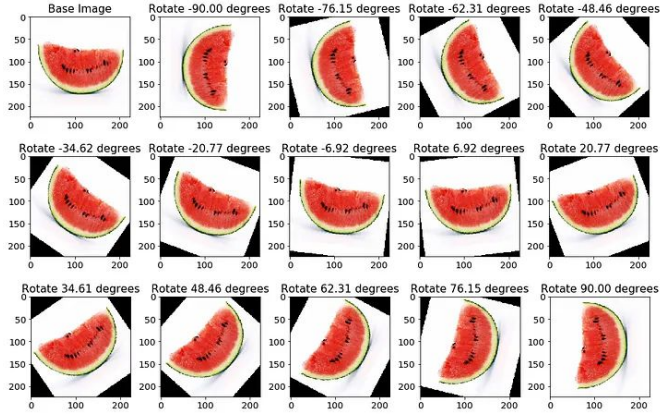
# Image Enhancement

Convolution and Attention **struggle with geometric invariance** — without augmentation, they fail to recognize the same object under rotation, scale changes, or viewpoint shifts.

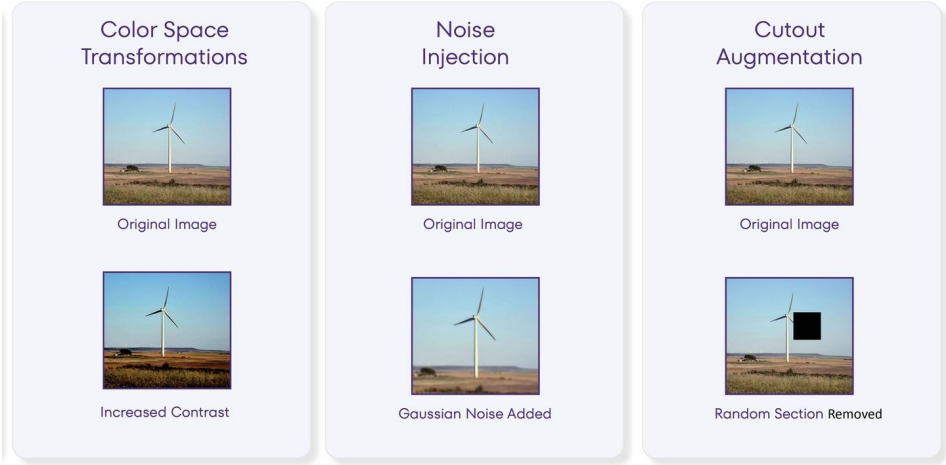
Deep learning projects often start with limited datasets. Image enhancement techniques allow us to **generate new, valid training samples from existing data**, effectively expanding our dataset.




**Flip**




**Rotate**




scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

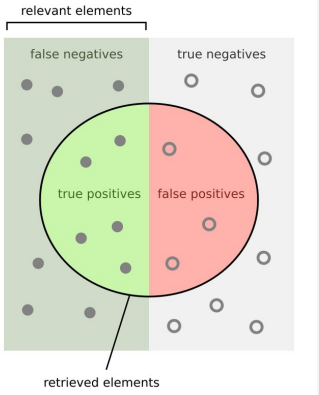
# Classification

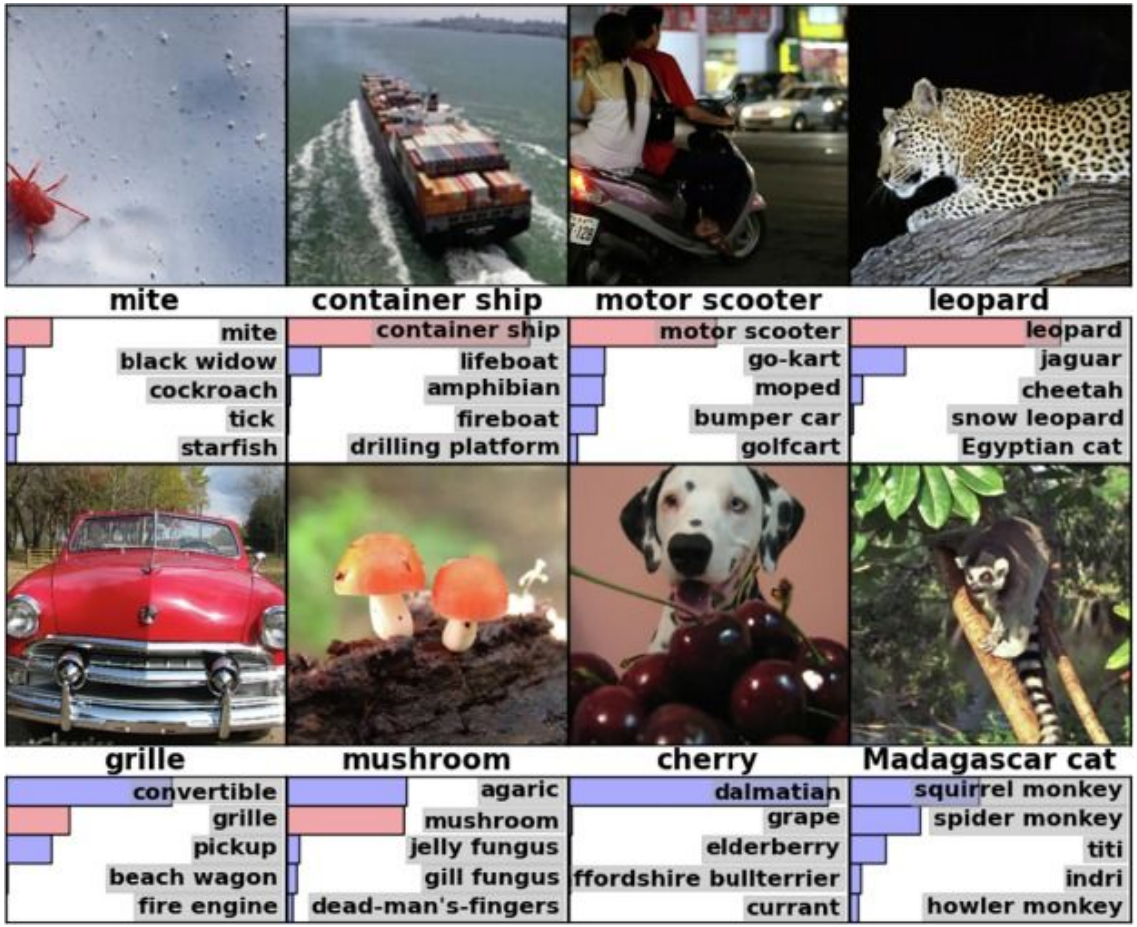
# Image Classification

$$f : \mathbb{R}^{H \times W \times C} \rightarrow \{1, \ldots, K\}$$

The model learns a probability distribution over all possible classes though loss cross entropy

$$\ell_{CE}(y, \hat{y}) = -\sum_{i=1}^{n} \sum_{c=1}^{C} y_{ic} \log(\hat{y}_{ic})$$




scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Convolutional Neural Networks

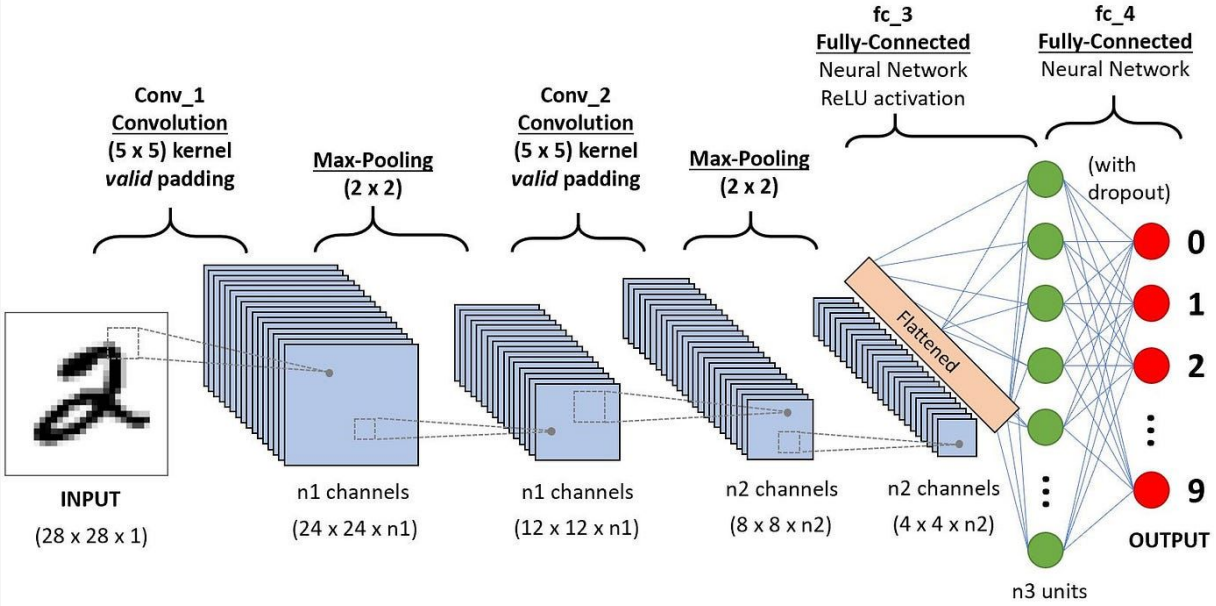
**Classic CNN architecture pattern**

**Channels grow (via conv filters):**
**3 → 64 → 128 → 256 → 512**

The intuition: you're trading spatial resolution for semantic richness.

Early layers detect edges and textures at precise locations

Deeper layers detect abstract concepts (eyes, wheels, faces) with less spatial precision but more "what" information.



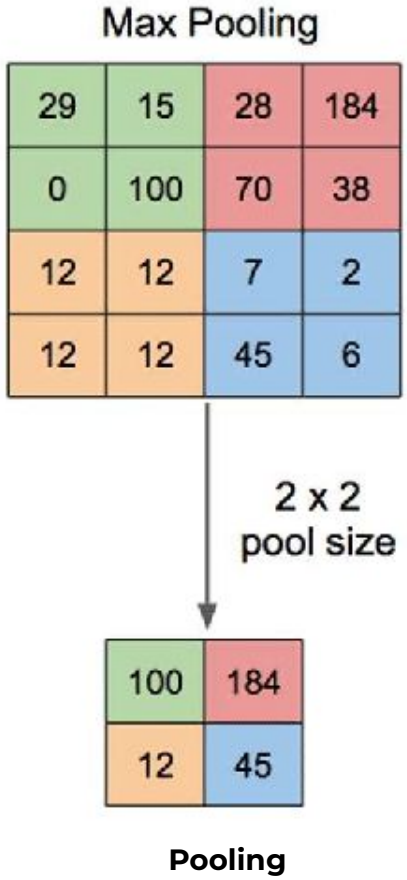
# Pooling and Stride Layers

**1. Spatial downsampling**

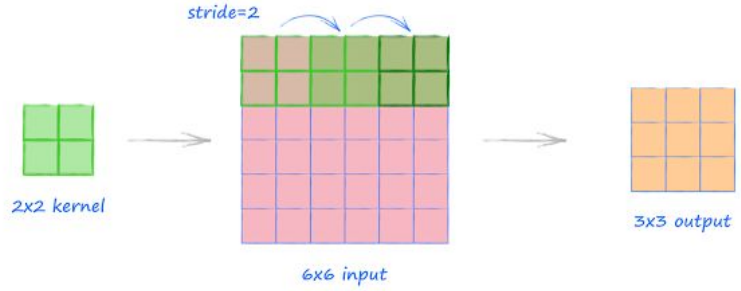
Reduces the feature map dimensions

**2. Translation invariance**

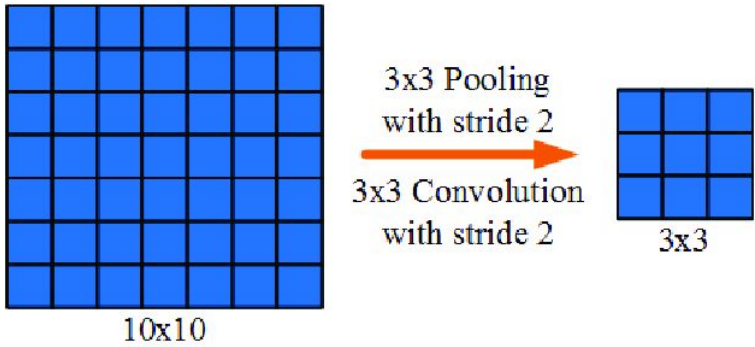
The network becomes slightly less sensitive to the exact position of a feature.

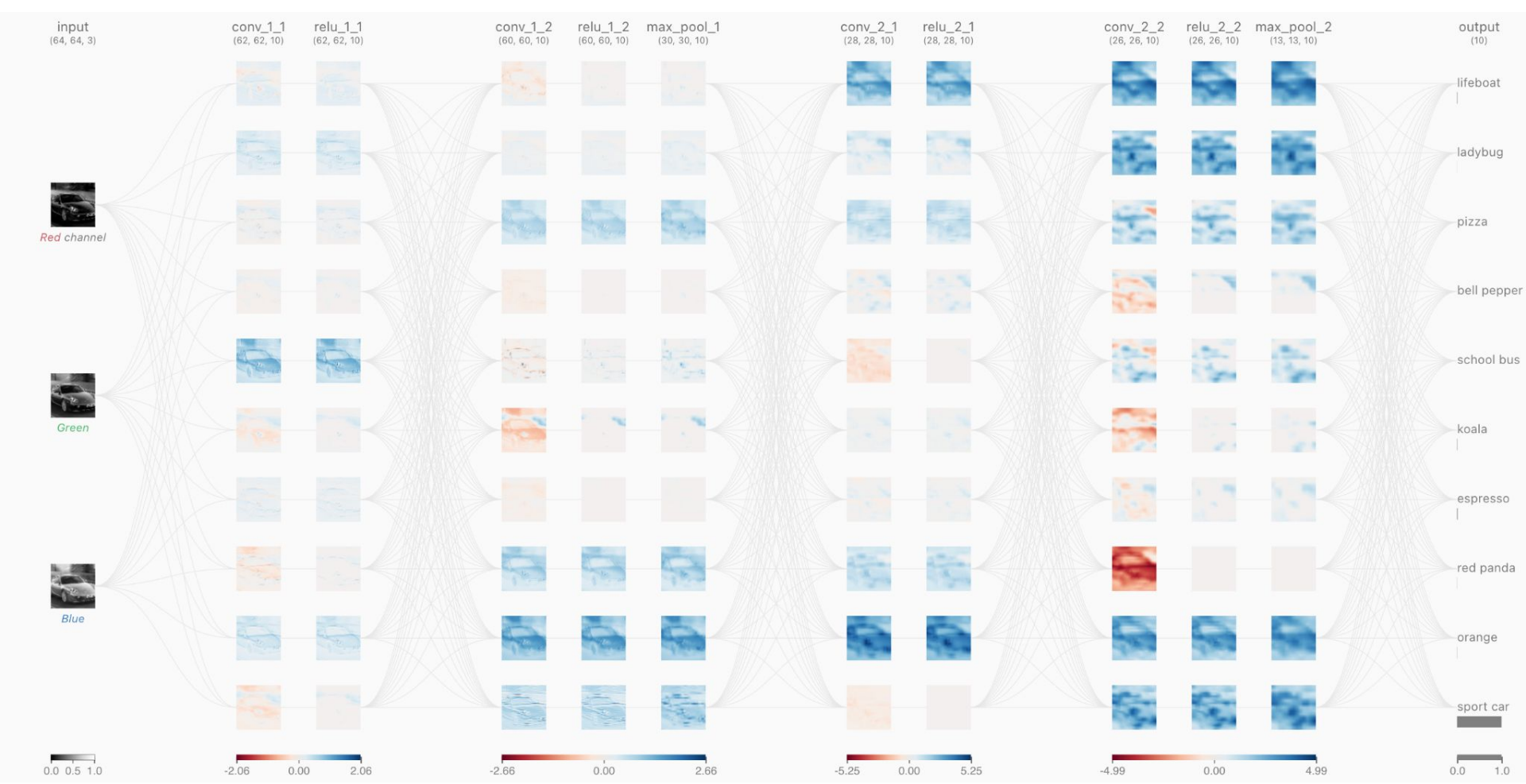


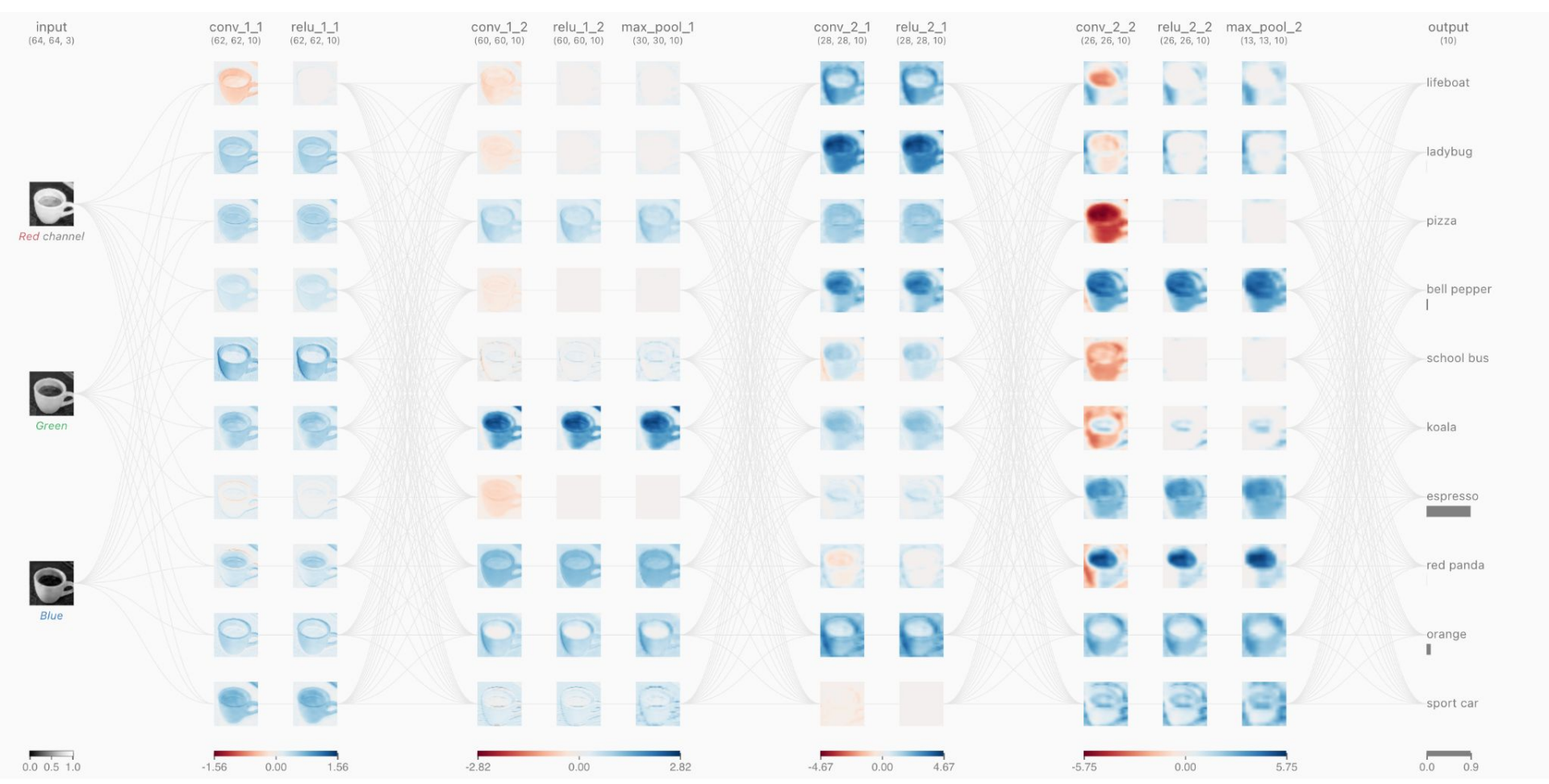

**Pooling**




**Stride**




input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-2.06 0.00 2.06
-2.66 0.00 2.66
-5.25 0.00 5.25
-4.99 0.00 4.99
0.0 1.0


input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.56 0.00 1.56
-2.82 0.00 2.82
-4.67 0.00 4.67
-5.75 0.00 5.75
0.0 0.9

# Convolutional Neural Networks

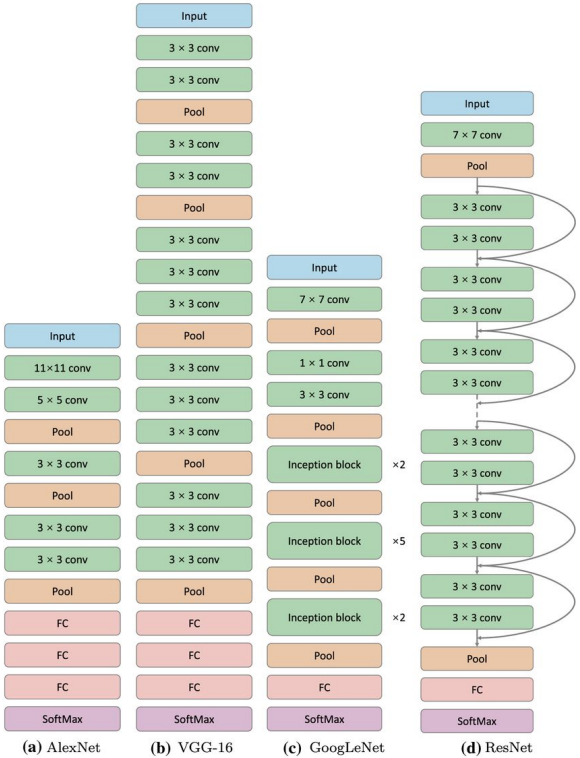
**Evolution of CNN Architectures**

**AlexNet :** 2012, First deep CNN to win ImageNet; ReLU, dropout, GPU training

**VGG-16 :** 2014, Simplicity: stack 3×3 convs; depth matters

**GoogLeNet :** 2014, Inception blocks: parallel filters of different sizes, 1×1 convs for efficiency

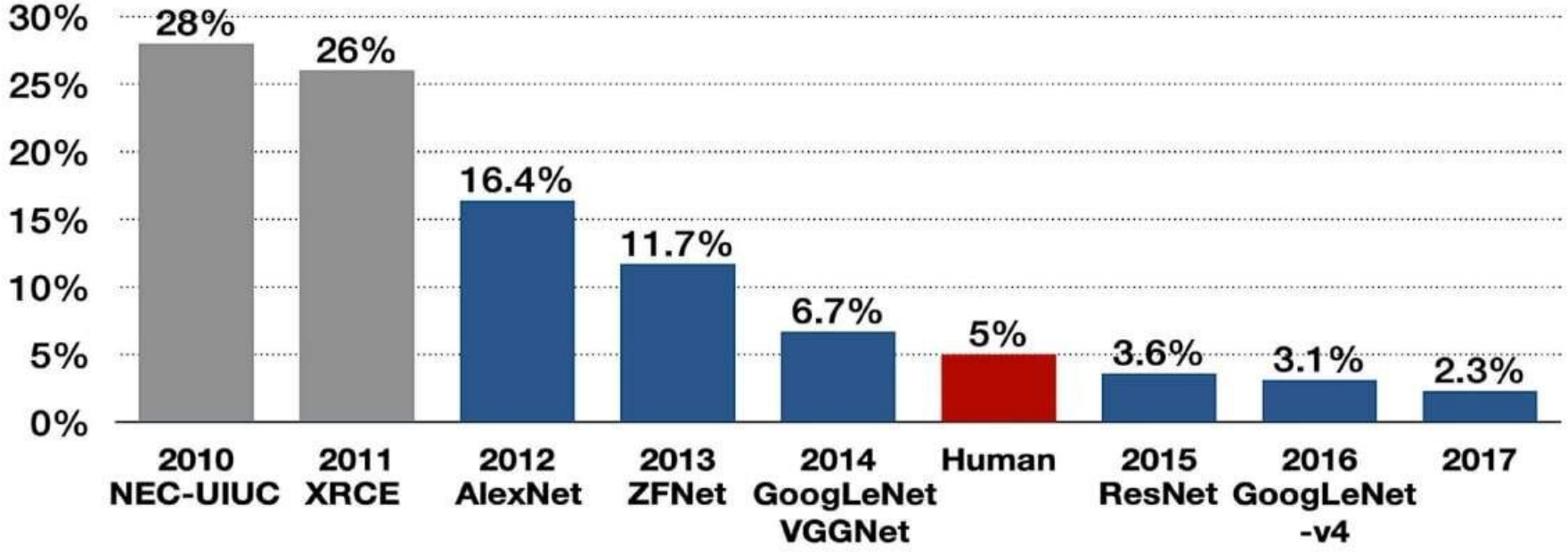
**ResNet** 2015, Skip connections: H(x) = F(x) + x; enabled 100+ layers



# Image Net


**Top-5 error**

**1.2 million training images** across **1,000 classes**



# Vision Transformer

**Architecture:**

1. Split the image into fixed-size patches (e.g., 16×16 pixels)

2. Flatten each patch into a vector (16×16×3 = 768 values for RGB)

3. Project each flattened patch through a linear layer → patch embedding

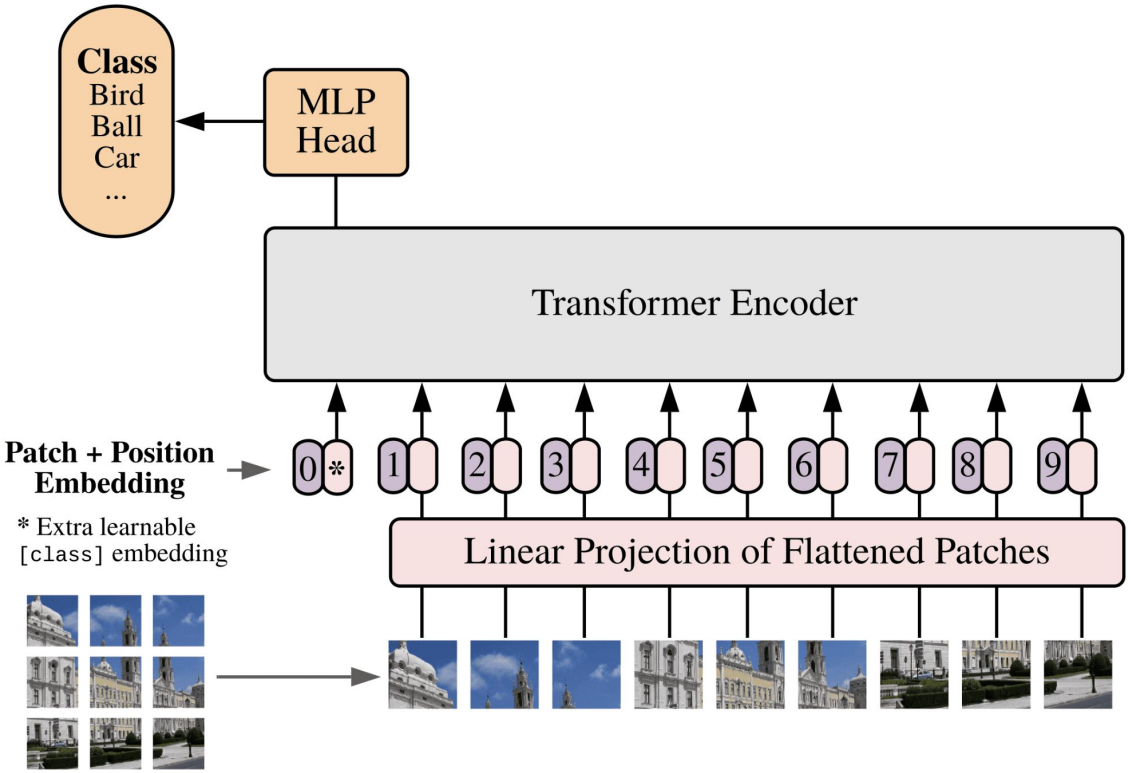
4. Add positional encoding so the model knows where each patch was located + a [CLS] token

Pass through standard Transformer encoder

**Key differences from CNNs:**

No translation invariance by design — must learn it from data

Needs more data to match CNNs (or use pretraining)



# Coca state of the art classification model

**CoCa (2022) —** Contrastive Captioners, combines contrastive + generative learning — ImageNet-1K — Top-1: 91.0%

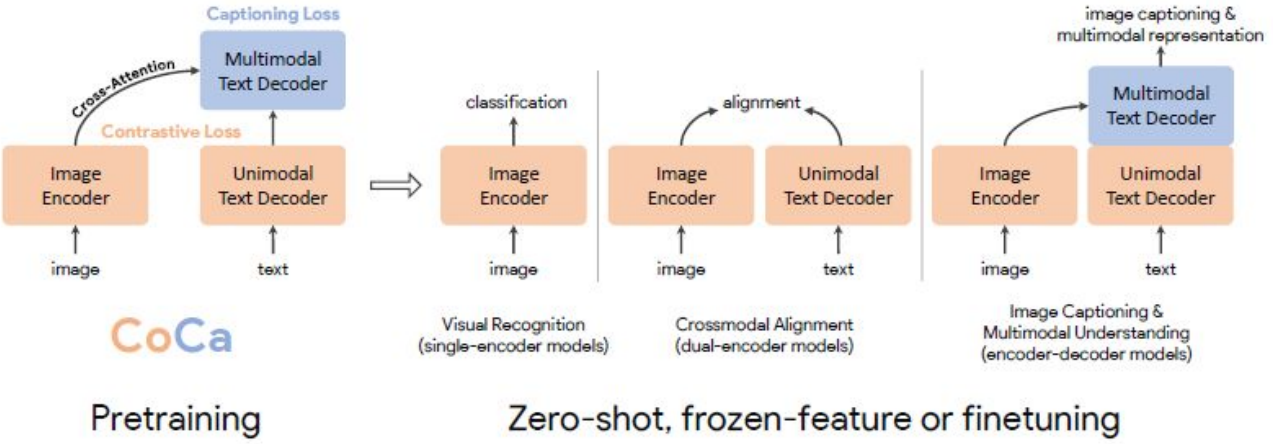



Figure 1: Overview of Contrastive Captioners (CoCa) pretraining as image-text foundation models. The pretrained CoCa can be used for downstream tasks including visual recognition, vision-language alignment, image captioning and multimodal understanding with zero-shot transfer, frozen-feature evaluation or end-to-end finetuning.

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Object detection

# Regression And Classification

$$f : \mathbb{R}^{H \times W \times C} \rightarrow \{(b_i, c_i, p_i)\}_{i=1}^{N}$$

$$b_i \in \mathbb{R}^4$$

Bounding box coordinates (x_min, y_min, x_max, y_max)

$$c_i \in \{1, \ldots, K\}$$

Class probability from K object categories

$$p_i \in [0, 1]$$

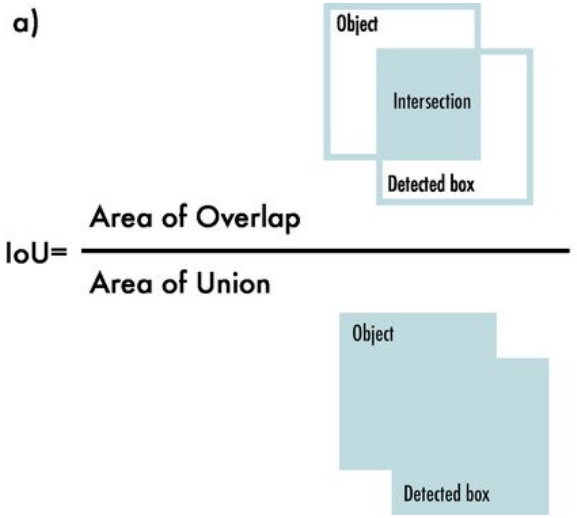
Confidence score (is anything here?)

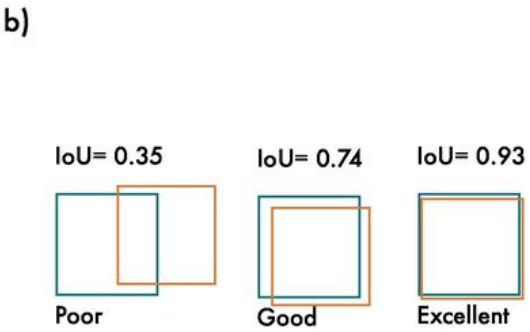
# Metric

**Intersection over Union (IoU)**
Measures how well the predicted box matches the ground truth.

$$\text{IoU}(B_{pred}, B_{gt}) = \frac{\text{Area}(B_{pred} \cap B_{gt})}{\text{Area}(B_{pred} \cup B_{gt})}$$

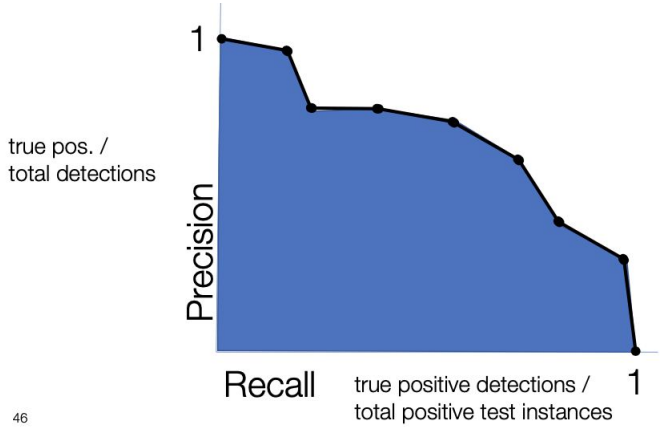
IoU = 0 → no overlap
IoU = 1 → perfect match
IoU ≥ 0.5 → typically considered a "correct" detection




**Average Precision (AP)**

AP@0.5: A detection is correct if IoU ≥ 0.5 (lenient)






scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Two-Stage Detectors (R-CNN Family)

**Stage 1: Region Proposal Network**

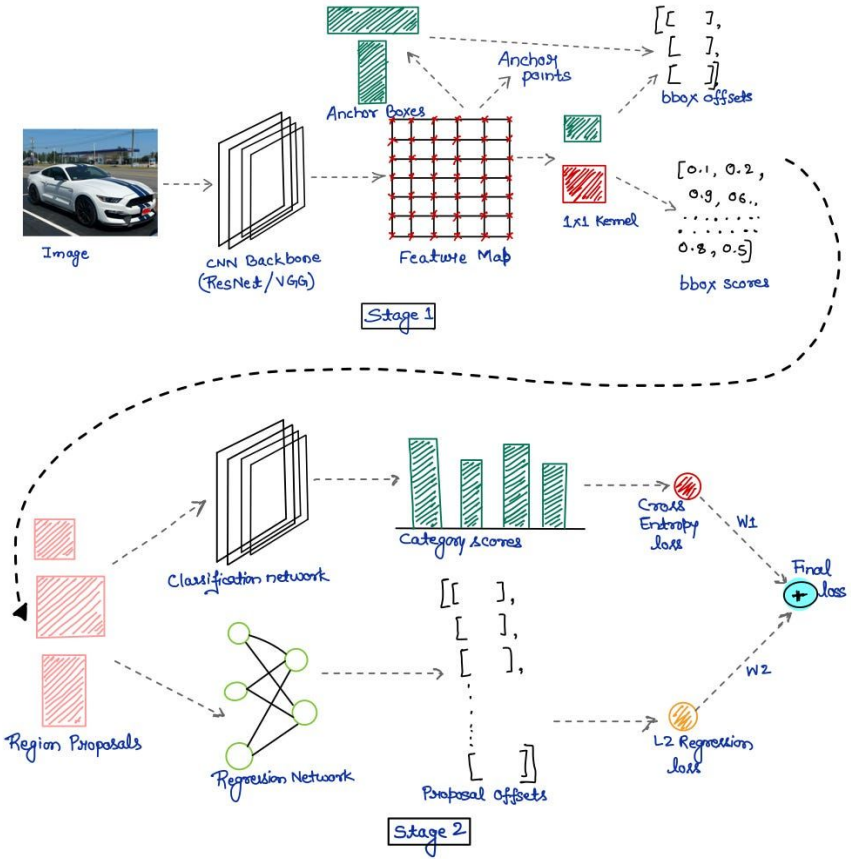
Predicts: "object or not?" + box offsets to refine anchor

Outputs ~2000 candidate regions

**Stage 2: Classification & Refinement**

Classification head: which class

Regression head: refine box coordinates



# One-Stage Detectors : YOLO

**How it works:**

Divide image into grid cells

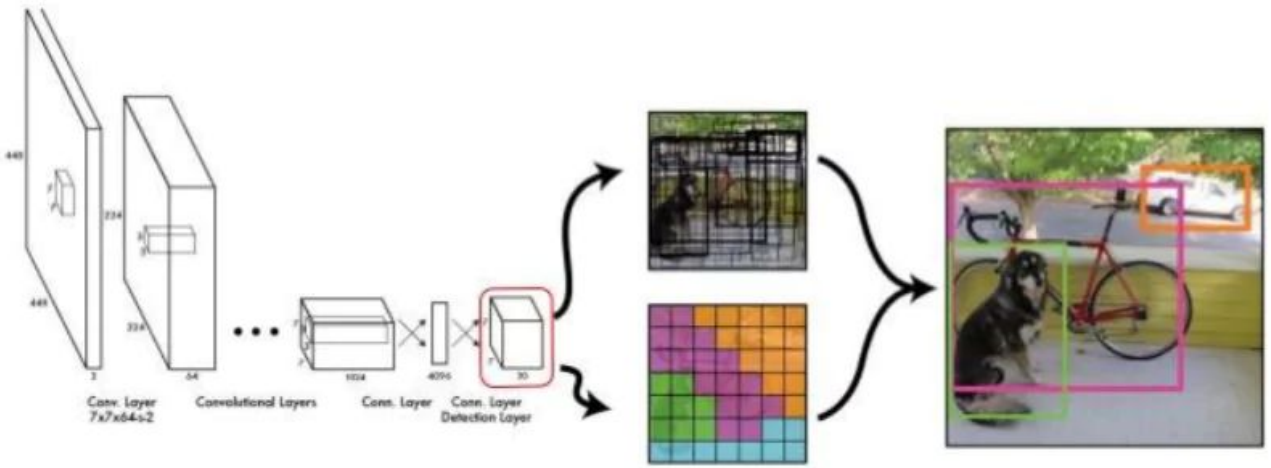
Each cell predicts, for each anchor (10-50)
- Bounding box offsets

- Objectness score

- Class probabilities per anchor

**Non-Maximum Suppression** (NMS):

Keep highest-confidence box, remove boxes with IoU > threshold (Repeat until no overlapping boxes remain)



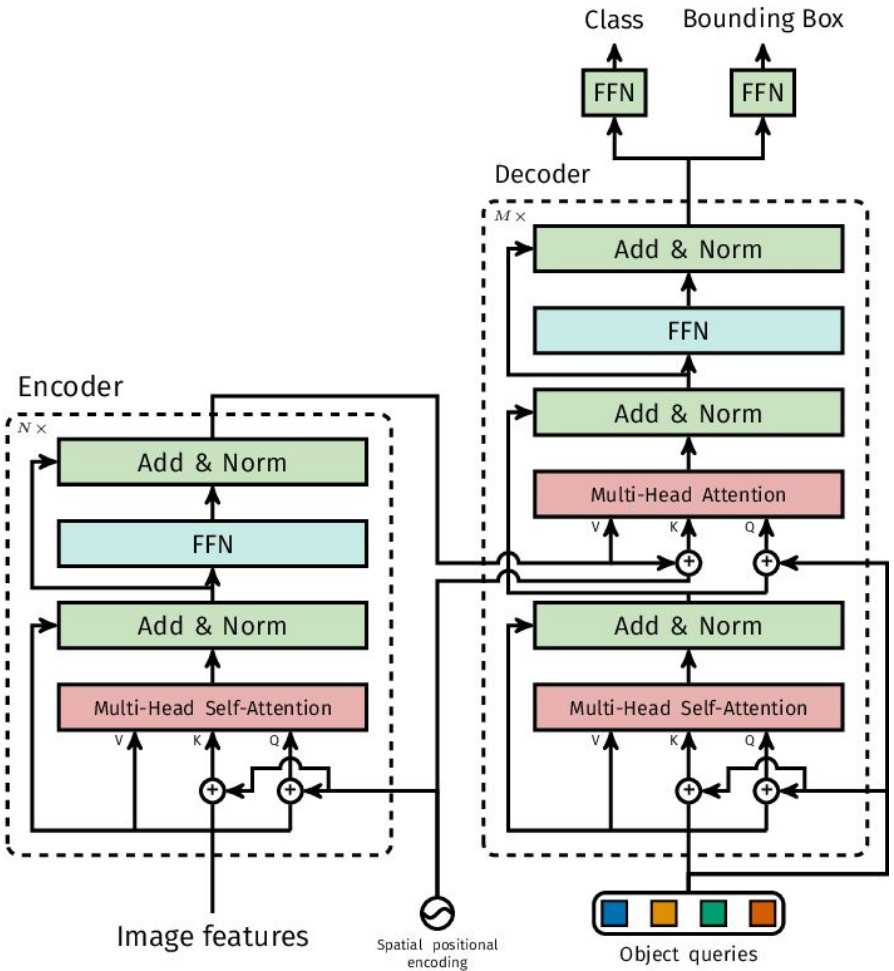

# Detection Transformer

Deter architectures eliminate predefined anchor boxes, no NMS, predicting object centers and sizes directly.

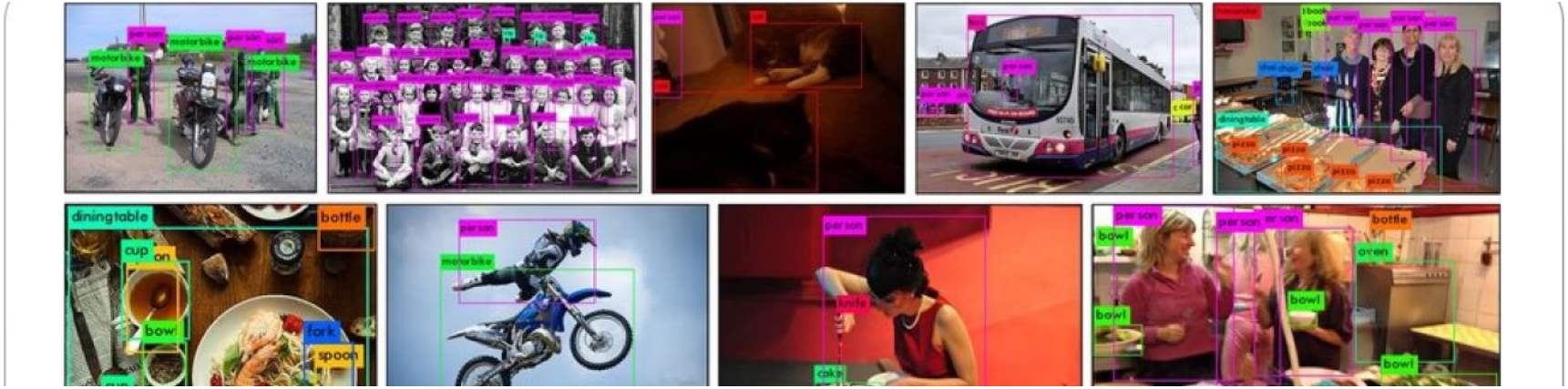
**Object queries:** Learnable embeddings that each "specialize" in detecting objects at certain positions/scales — no explicit anchors needed.

**Centerness** (from FCOS): Predicts how centered the detection is within the box; downweights edge predictions.

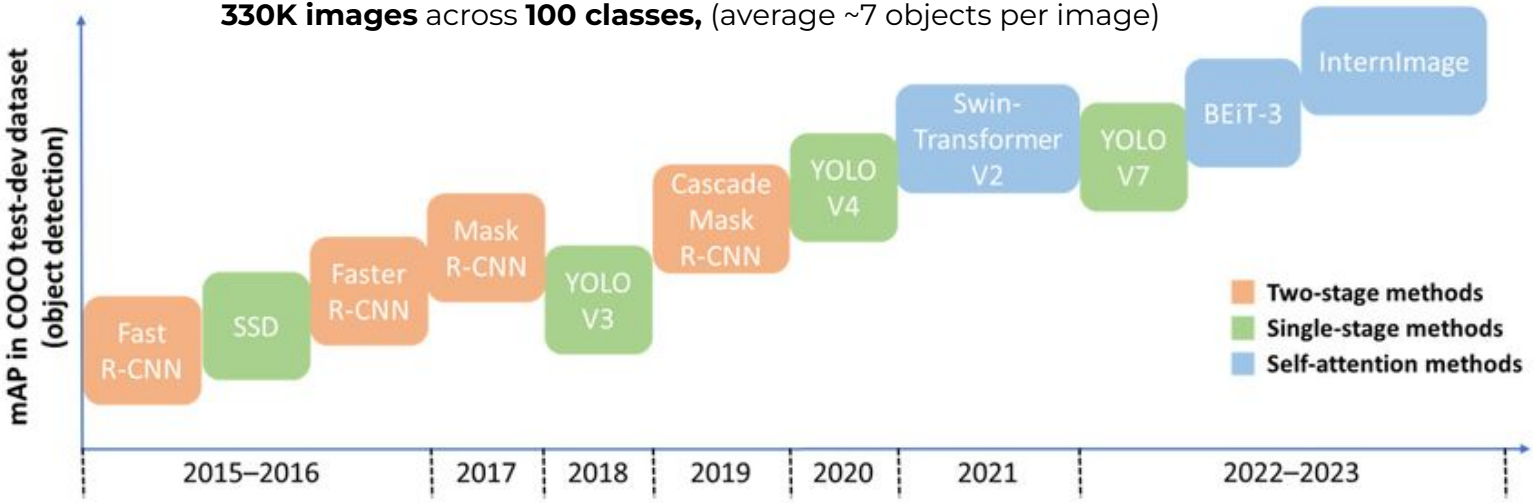
**Cons**: Slower convergence, needs more training.



# CoCo (Common Objects in Context)




**330K images** across **100 classes,** (average ~7 objects per image)

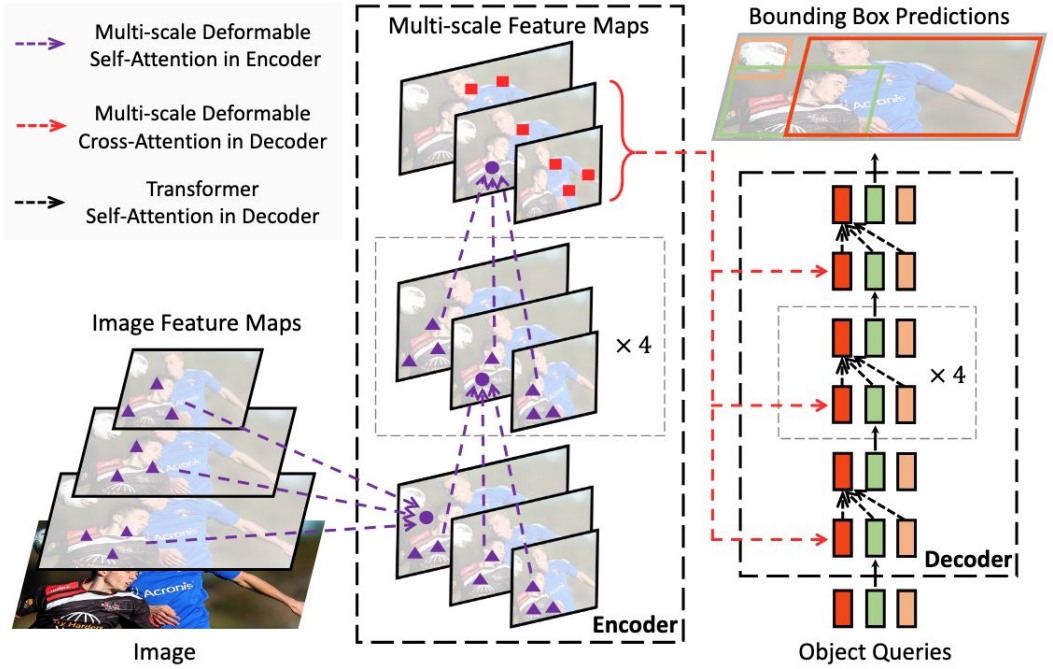



scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Deformable DETR state of the art object detection

**Deformable DETR** (Mar 2025) — Neural architecture detection transformers — COCO — AP: 60.6

**YOLOv12 (Feb 2025)** — Attention-centric YOLO with area attention — COCO — AP: ~54 (real-time)



# Segmentation

# Pixel Classifier

Predict one label per pixel.

$$f : \mathbb{R}^{H \times W \times C} \rightarrow \{0, \ldots, K\}^{H \times W}$$

**Applications:**

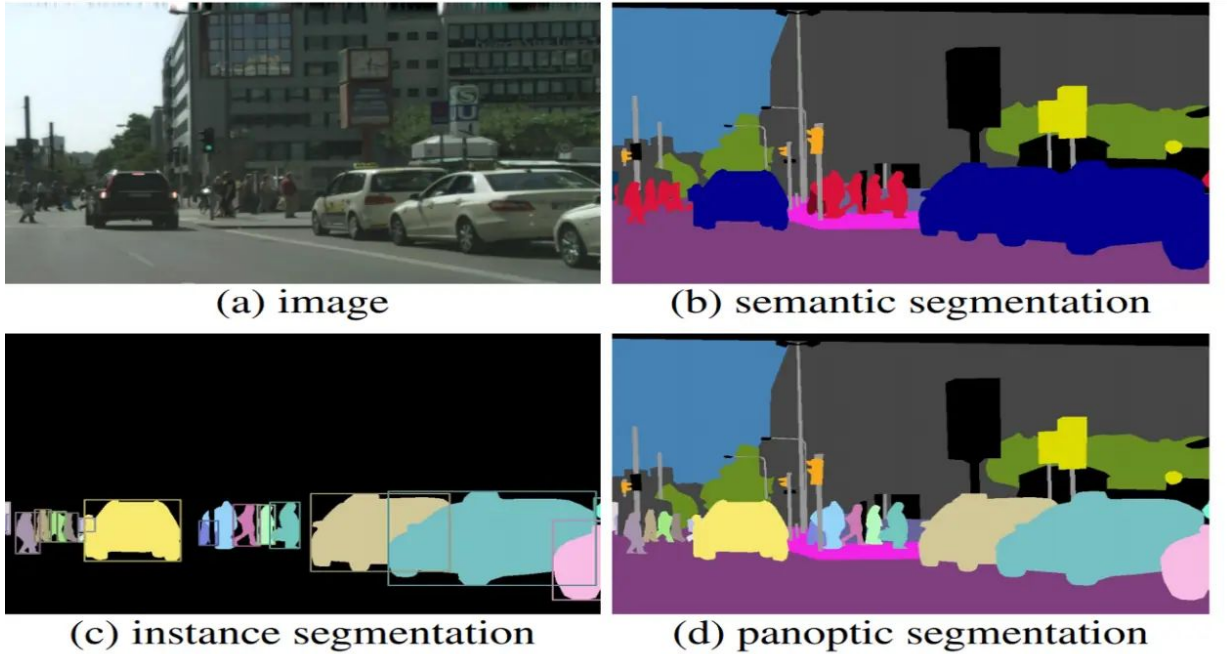
**Medical imaging**: tumor delineation, organ segmentation

**Autonomous driving**: road, vehicle, pedestrian segmentation

**Satellite imagery**: land use classification, building footprints

**Video editing**: background removal, object isolation

**Agriculture**: crop monitoring, disease detection



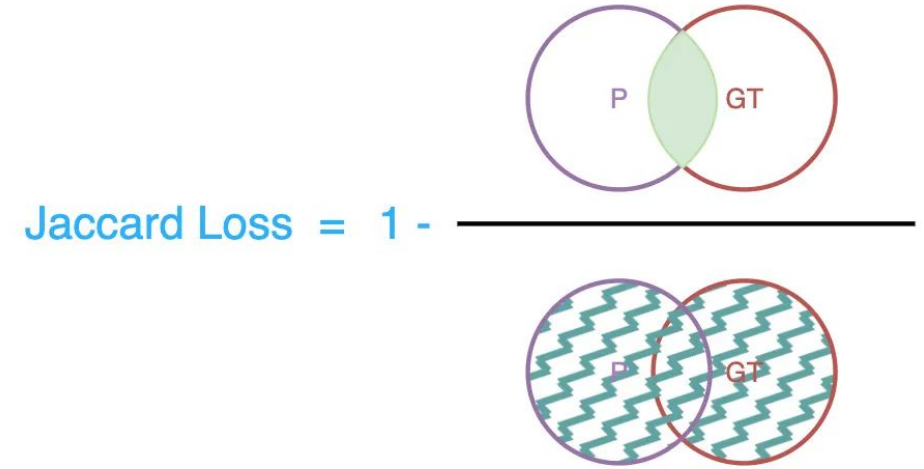
# Segmentation losse

**IoU (Intersection over Union)**

Penalizes both false positives and false negatives equally

Ranges from 0 (no overlap) to 1 (perfect match)

**Mean IoU** (mIoU): average IoU across all classes




$$\mathcal{L}_{\text{Jaccard}} = 1 - \frac{\sum_i p_i g_i}{\sum_i p_i + \sum_i g_i - \sum_i p_i g_i}$$

$p_i \in [0, 1]$ are the predicted probabilities

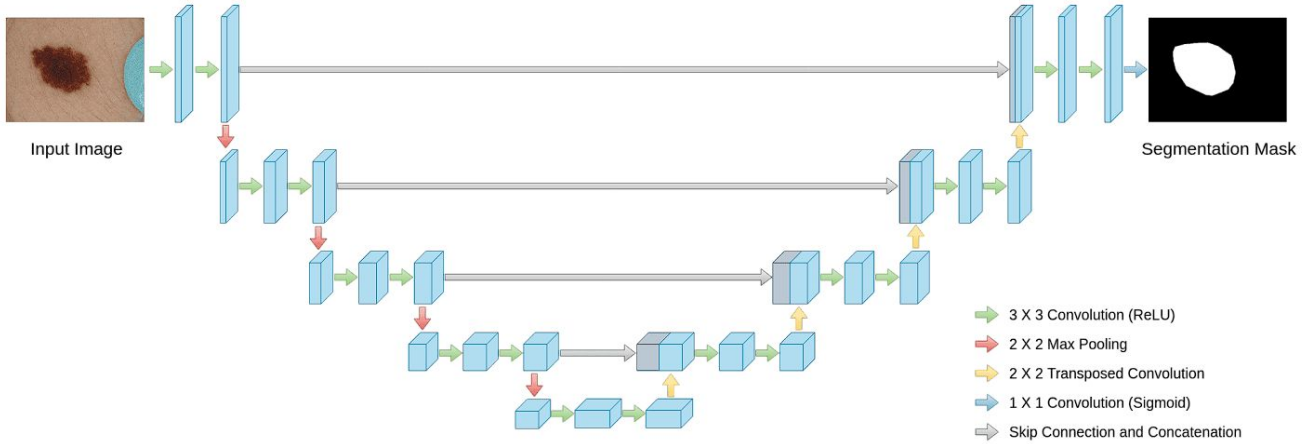
$g_i \in \{0, 1\}$ are the ground truth labels.

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Architecture

**U-net architecture**.

Originally designed for biomedical image segmentation (2015), now the backbone of many segmentation and diffusion models.




scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# CoCo

**Seg-VAR (Nov 2025) —** Visual autoregressive modeling for segmentation — ADE20K — mIoU: 64.8

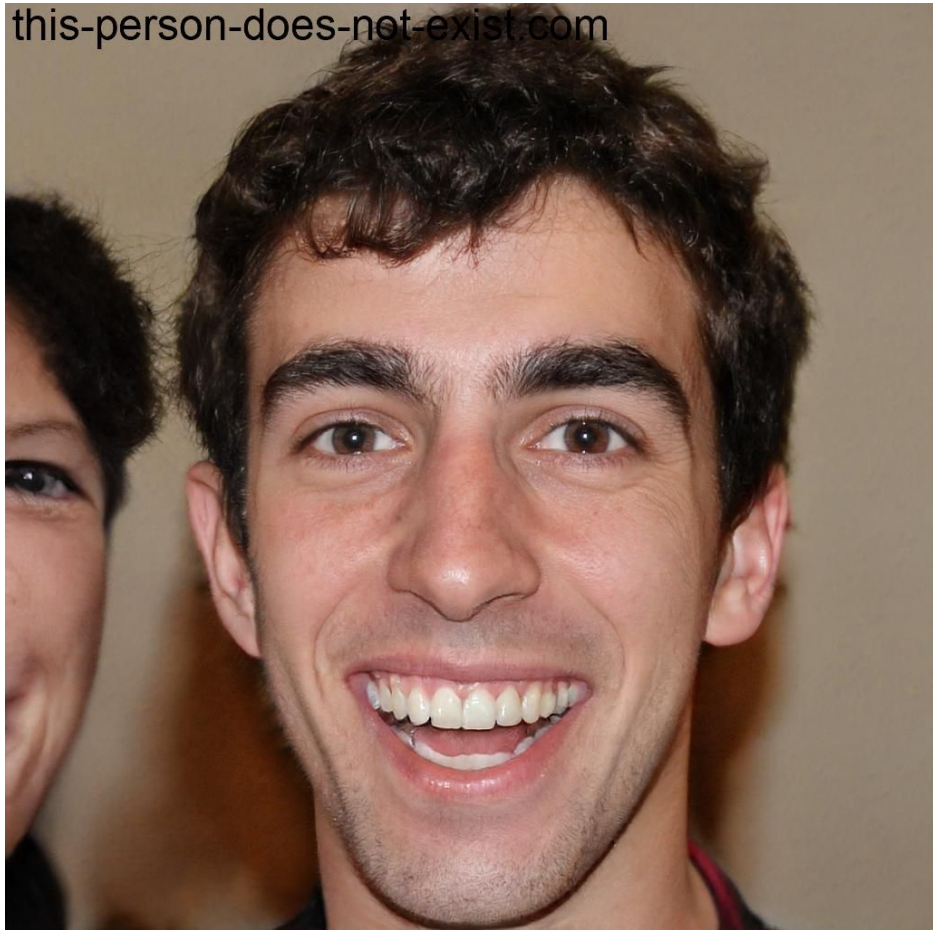
scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Generation

# Examples

https://thispersondoesnotexist.com/

https://thisxdoesnotexist.com/



# Multiple technics

Map between a **simple distribution** (z ~ N(0,I)) and **complex data distribution** p(x).

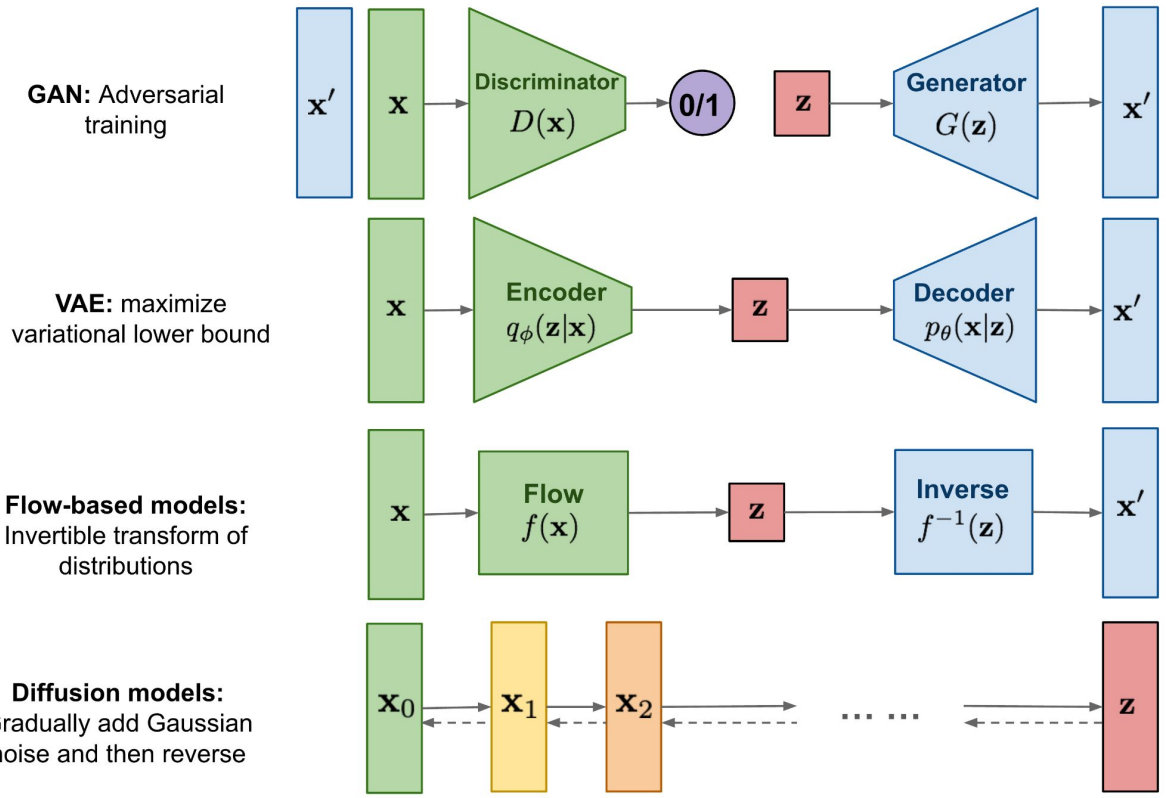
**Key differences:**

**GAN**: implicit density (can sample, can't compute p(x))

**VAE**: approximate density (ELBO is lower bound)

**Flow**: exact density (but architecture constraints)

**Diffusion**: approximate density, iterative sampling

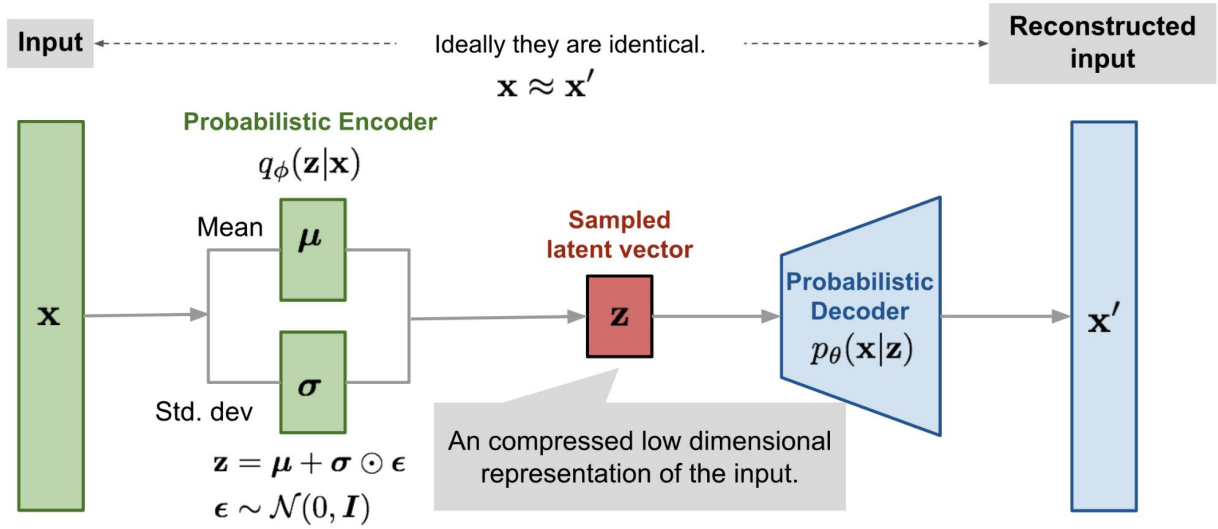


# Variational Autoencoders (VAE)

**Key idea:** Learn a compressed latent representation z that captures the essential structure of x, with probabilistic encoding.

**Loss function (ELBO):**

- **Reconstruction term:** Decoder should reconstruct x well
- **KL term:** Encoder distribution should stay close to prior p(z) = N(0, I)




$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p(z))$$

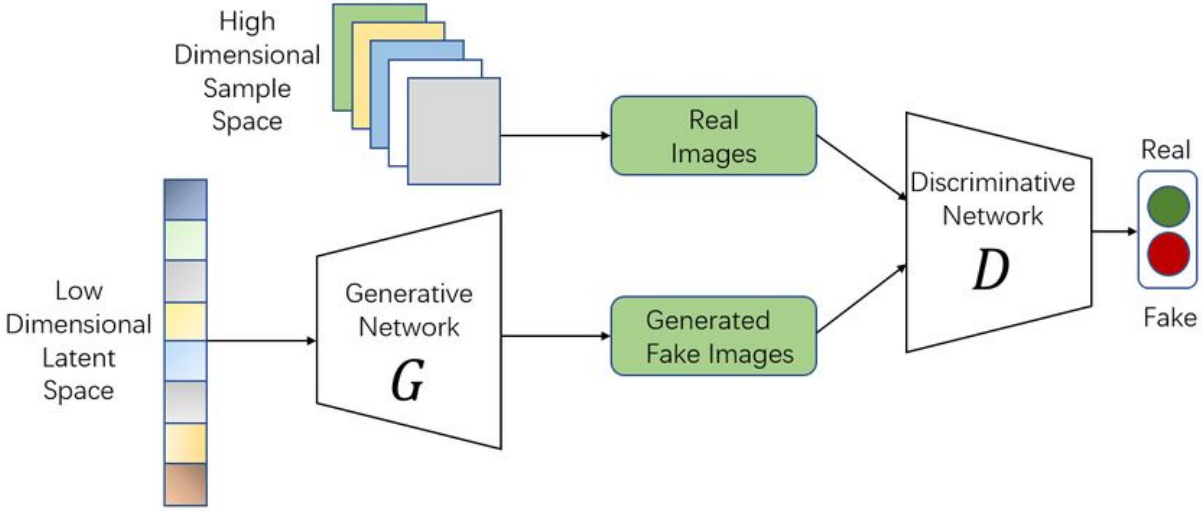
scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Generative Adversarial Networks

**GANs** learn to generate data through adversarial training between two networks: a generator G that creates fake samples and a discriminator D that distinguishes real from fake samples.

**Discriminator** maximizes its ability to classify real vs fake

**Generator** minimizes discriminator's ability to detect fakes




$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$$

# Diffusions

Diffusion models learn to generate data by reversing a gradual noising process.

**Forward process (fixed):**

1. Start with data x_0
2. Gradually add Gaussian noise over T
3. After T steps: x_T ≈ N(0, I) (pure noise)

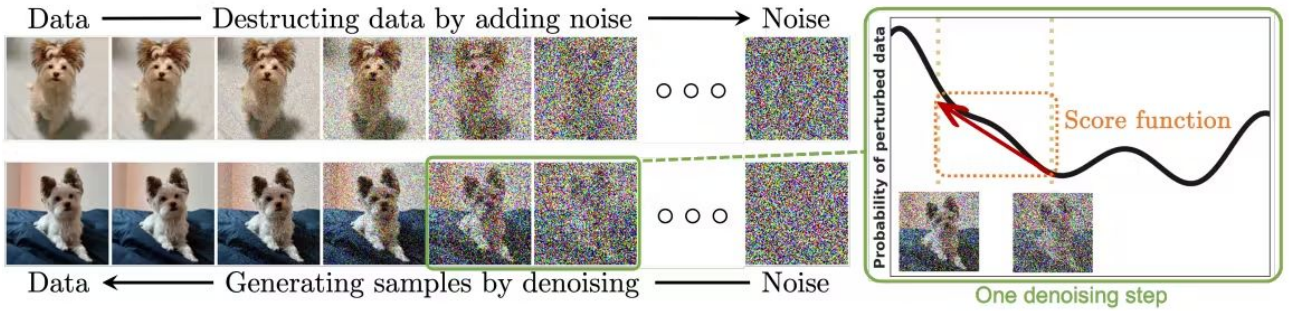
**Reverse process (learned):**

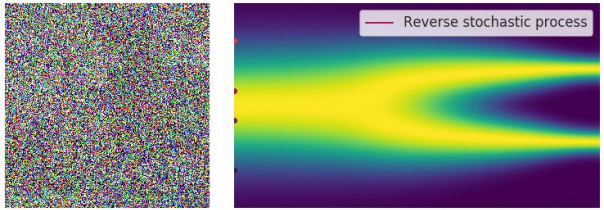
1. Start from noise x_T ~ N(0, I)
2. Learn to denoise step by step: x_{t-1} = f_θ(x_t, t)
3. After T steps: recover clean sample x_0

**Training objective:**

At each step, predict the noise ε that was added

Trade-off: High quality but slow sampling (typically 20-1000 steps).




**Small denoising steps are easy to learn but slow to produce**

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Conditional Generation

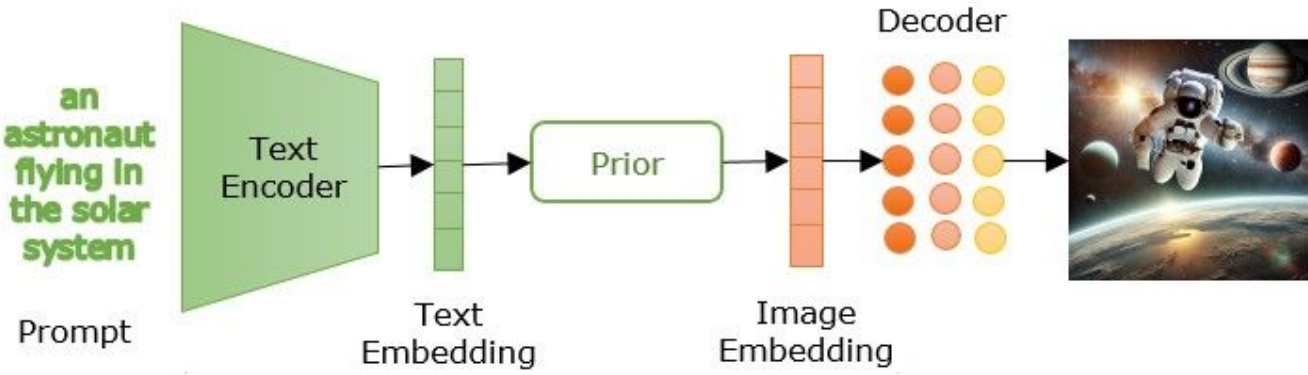
All generative model families can be **conditioned** on additional information (class labels, text, images) to control generation.

Condition encoder and decoder on label

**Diffusion dominates text-to-image** (Stable Diffusion, FLUX)

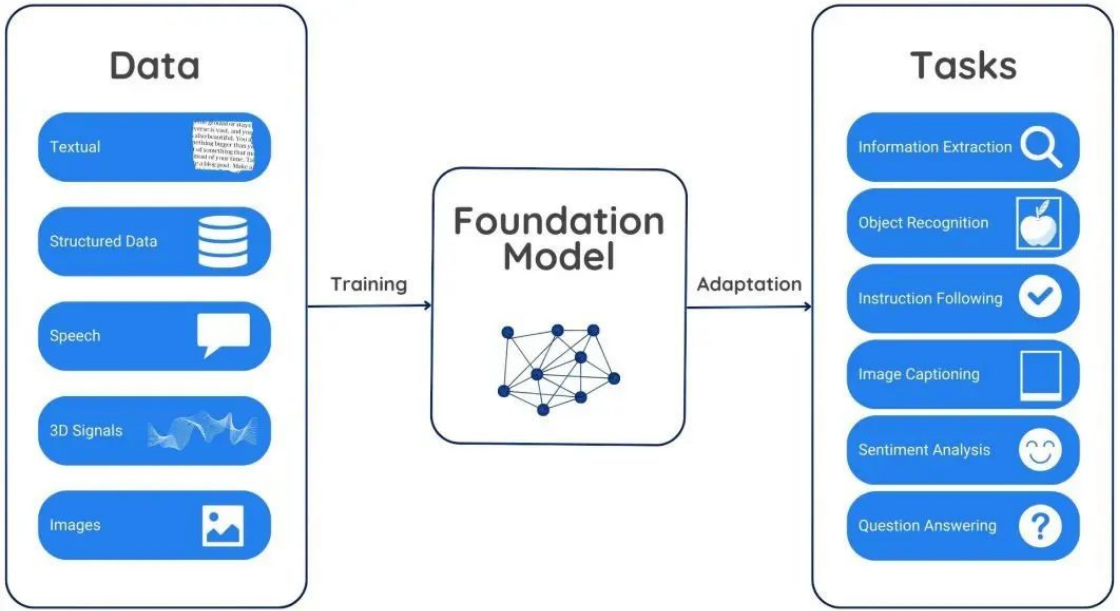
**GANs remain competitive for fast inference** (real-time applications, video)

**VAEs often used as latent space compressors** within diffusion pipelines



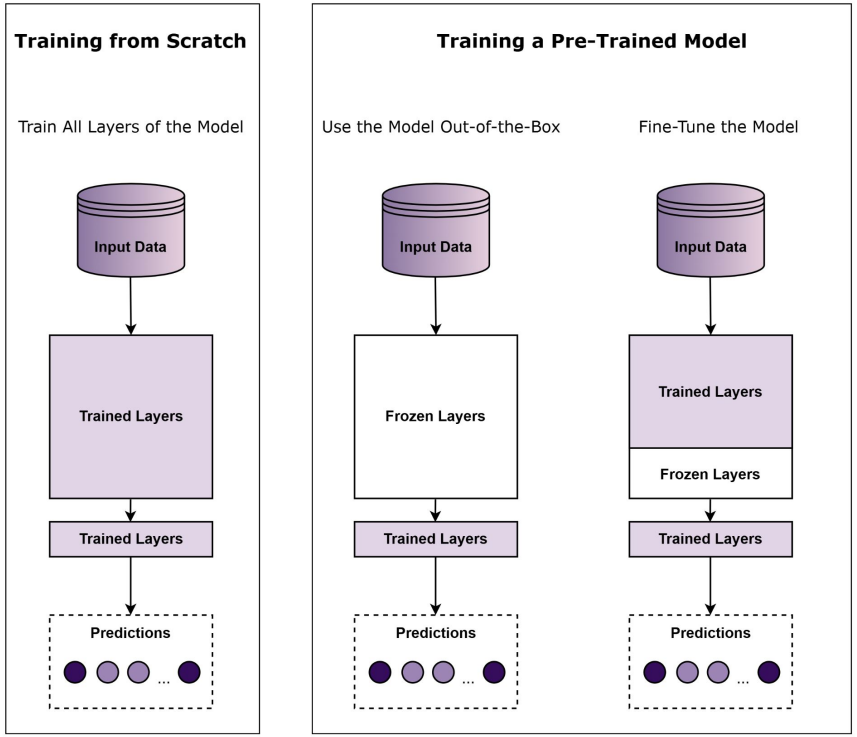

# Ressources

# Foundation Model




Foundation models are large AI models pre-trained on vast amounts of diverse data that can be adapted for a wide range of downstream tasks without training from scratch.

# Transfer Learning

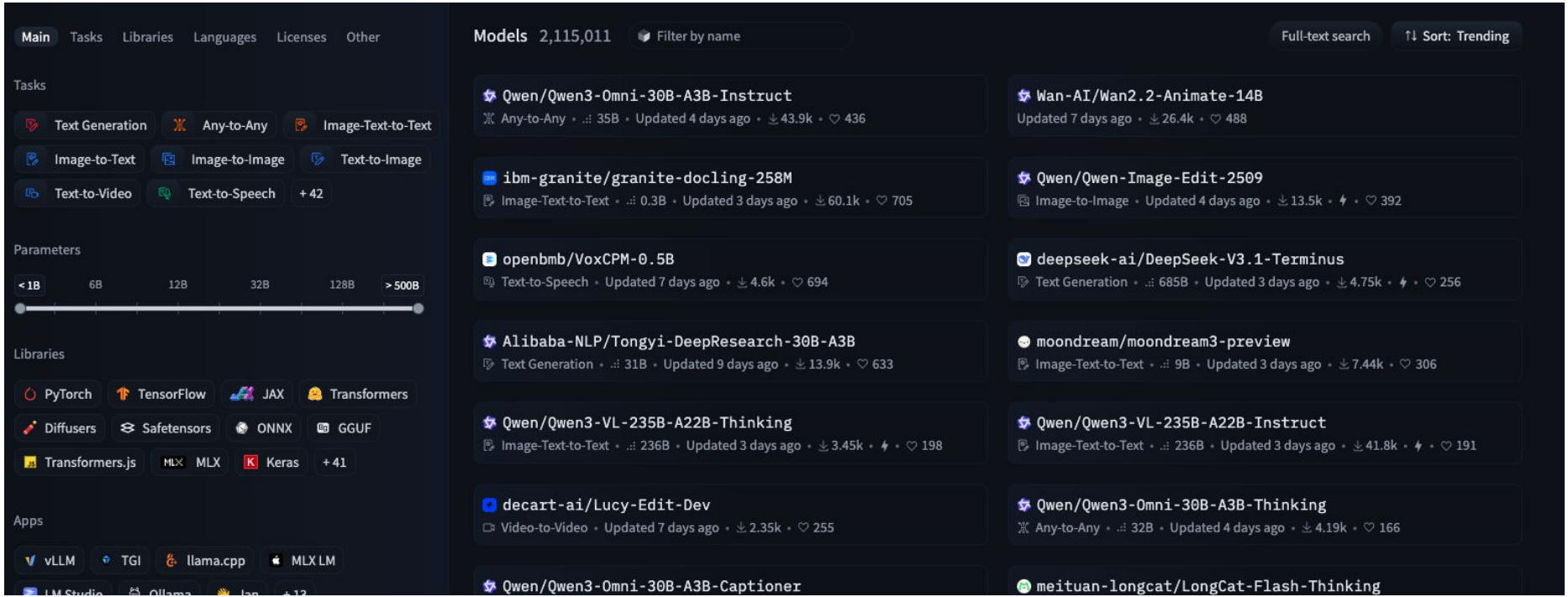



Instead of training a model from scratch (starting with randomly initialized parameters), we can start with a pre-trained model (one that has fine-tuned its parameters efficiently on a large dataset) and then fine-tune it in just a few iterations on our own (smaller) data.

scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

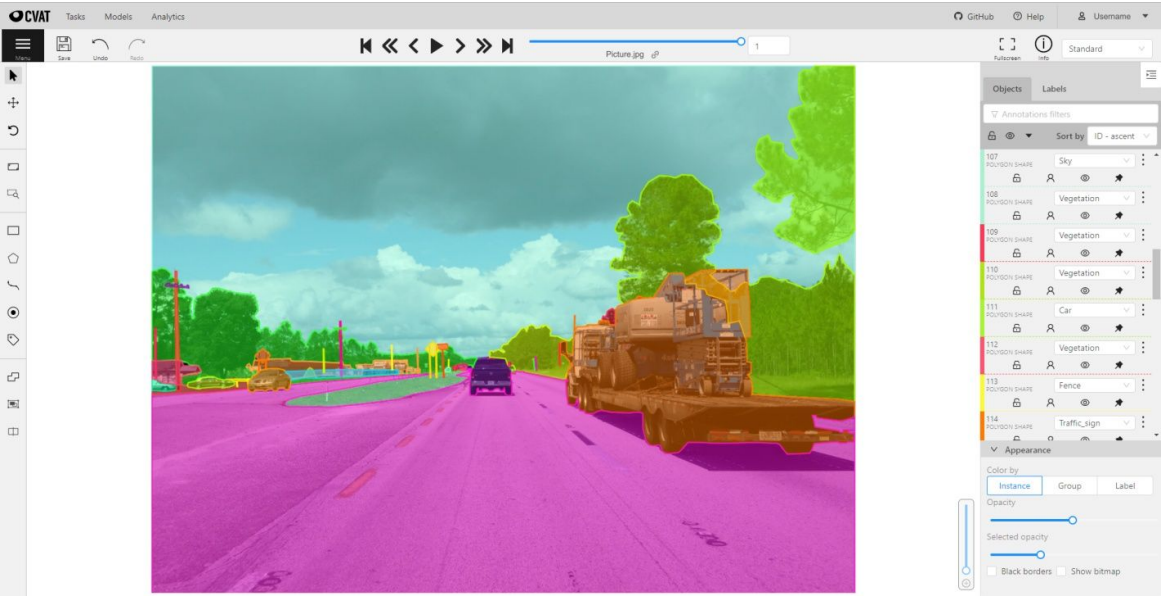
# Open Source Pre-trained Models

**Hugging Face**




scai
SORBONNE CLUSTER FOR ARTIFICIAL INTELLIGENCE

# Labelized your data



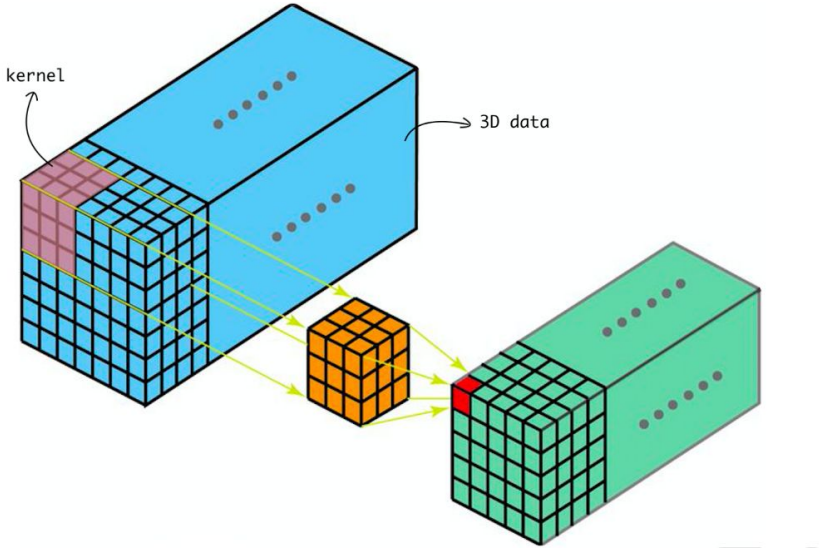

**CVAT**


**Label Studio**

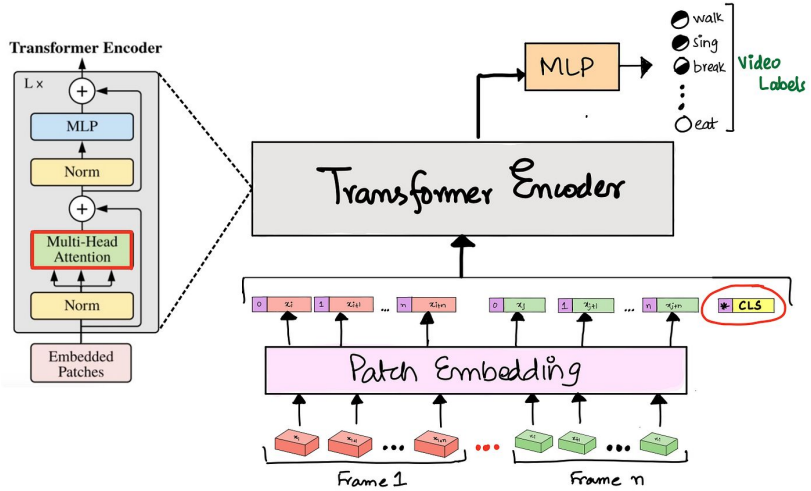
These tools support pre-annotation (importing model predictions for human validation/correction), which dramatically speeds up labeling

# Thanks

# Video Application




3D Cnn




Video Transformers