Robot Learning

Basics of computer vision for robotics
Representation learning

USC University of Southern California

LiraLab

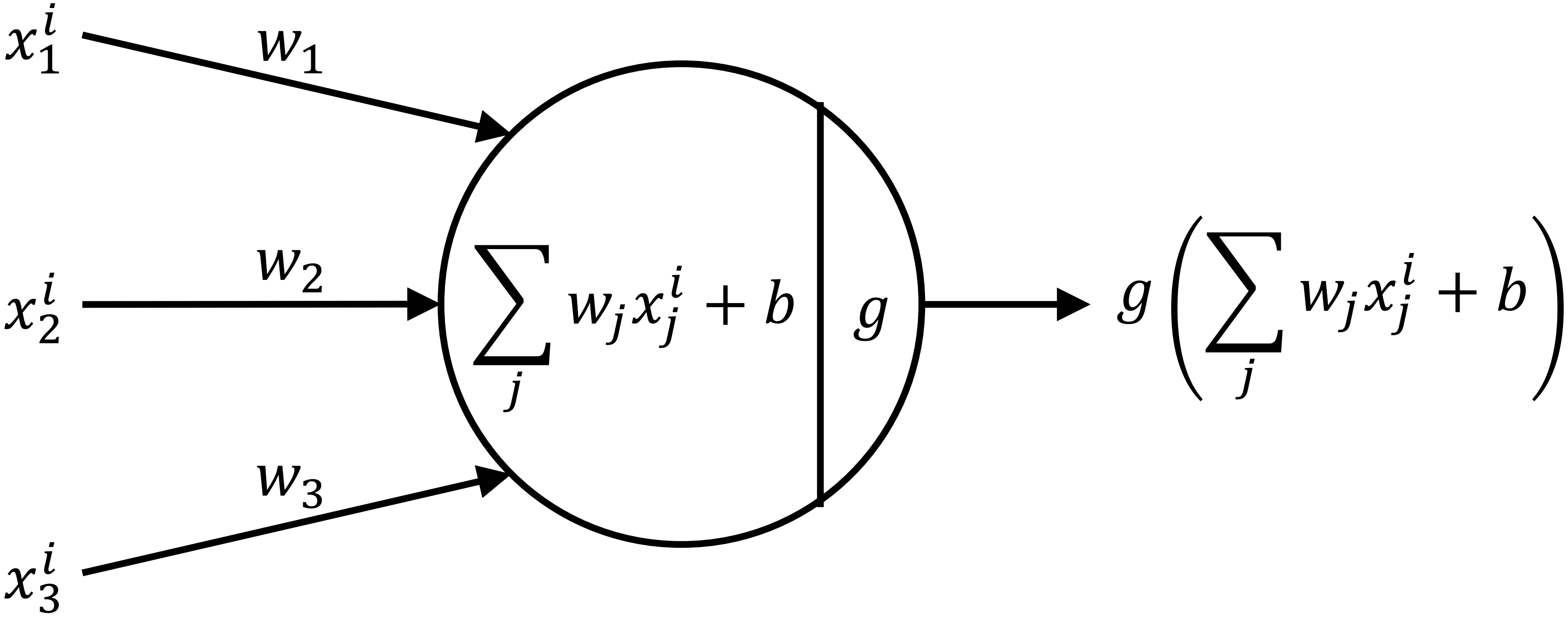
Logistics that I forgot to tell last time

- Project has multiple components.

Component	Contribution to Grade	
Project Proposal Report	5%	October 8th
Project Milestone Report	5%	November 5rd
Project Presentation (Possibly with Demo)	10%	December 1st
Final Project Report	15%	December 3rd
Peer Review	5%	December 10th
Total	40%	

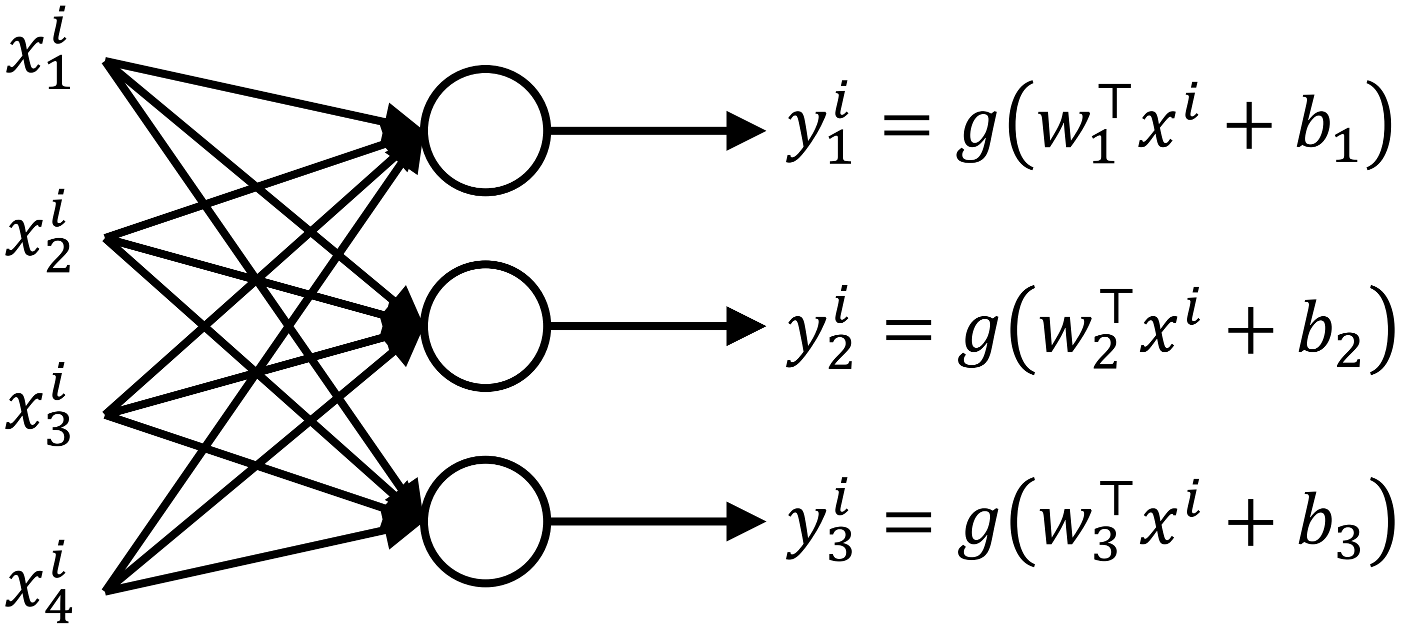
Last time…

1. A perceptron



Last time…

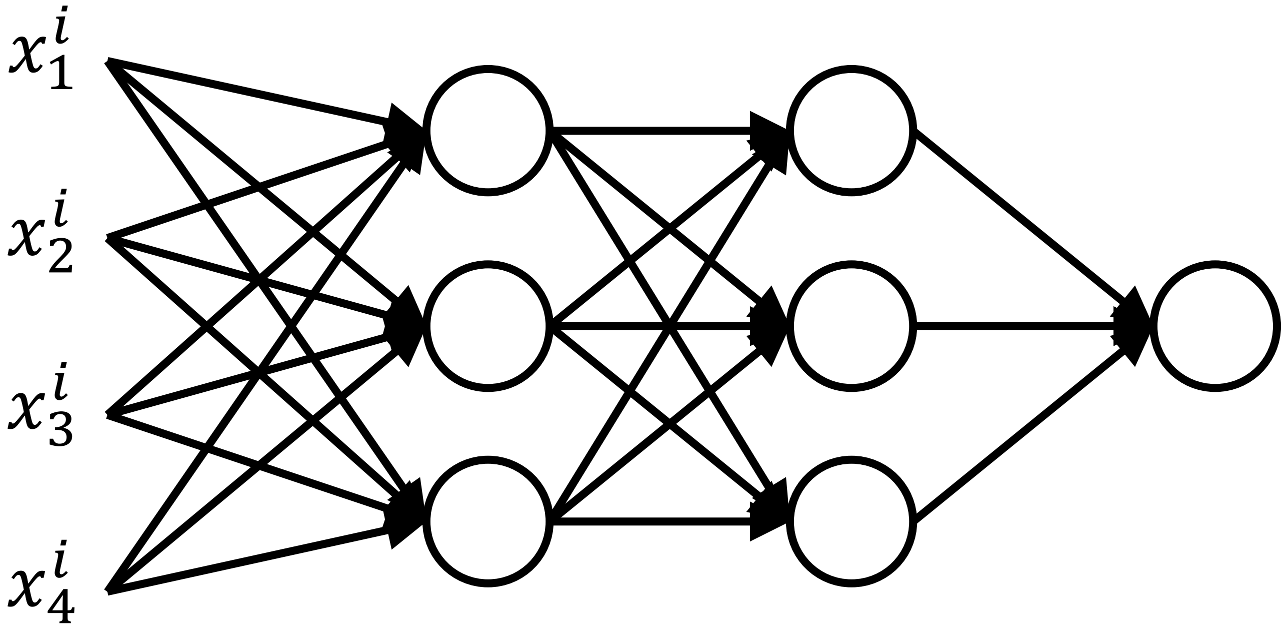
2. A single layer neural network



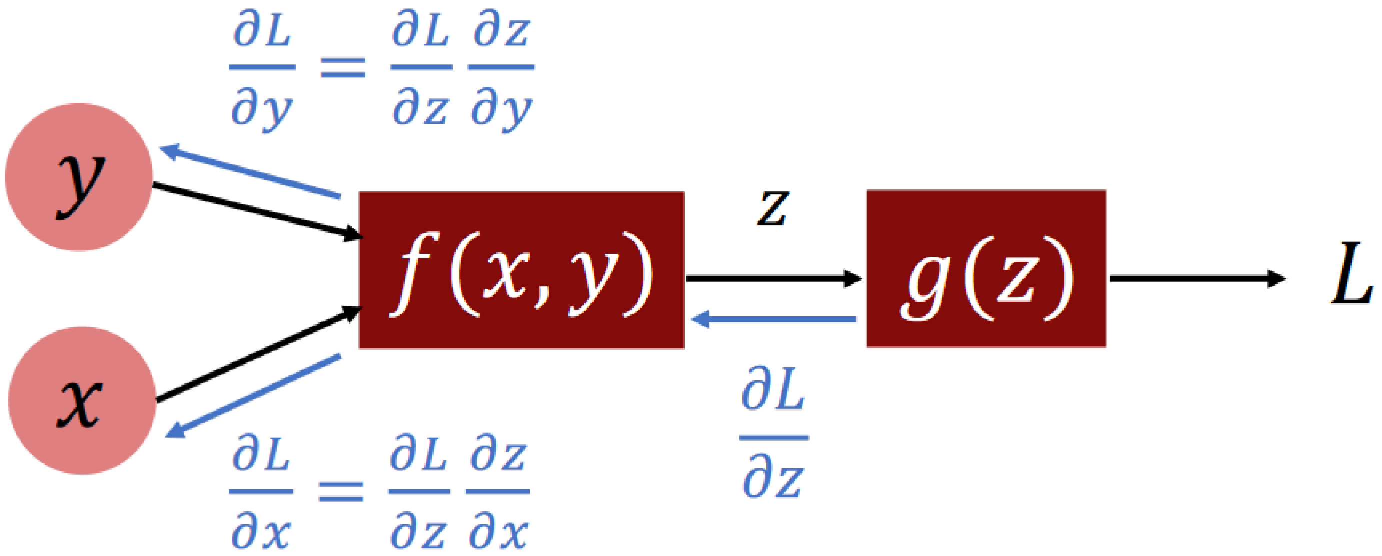
$$y = g(W^\top x + b)$$

Last time…

3. A deep neural network



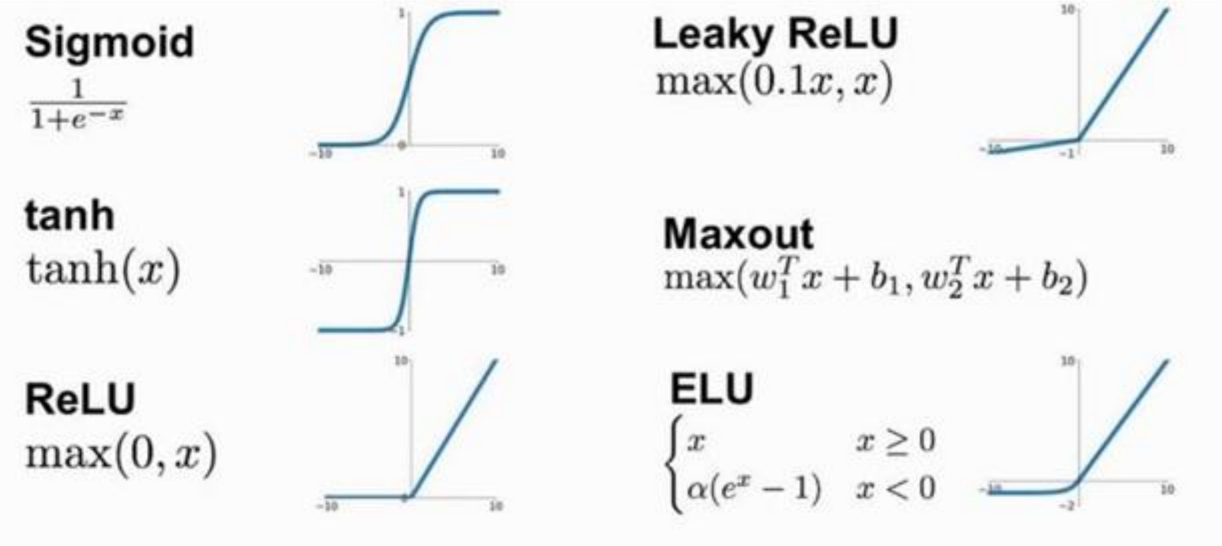
Last time…



From: Marco Pavone, Stanford University

Last time…

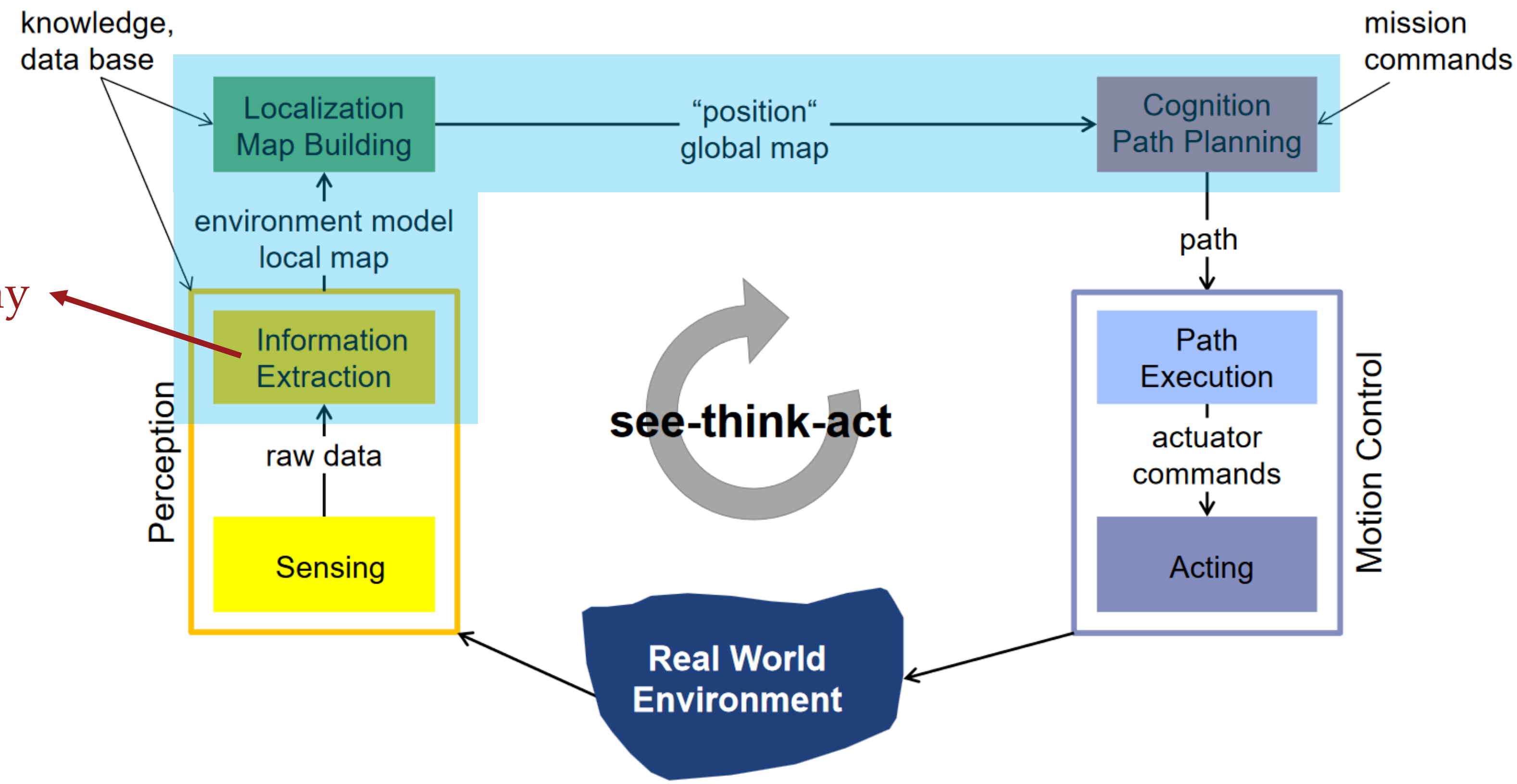
g should not be a linear function.



Analysis Of Optimizing Neural Networks And Artificial Intelligent Models For Guidance, Control, And Navigation Systems
Rahul Jayawardana, Thusitha Sameera Bandaranayake

Robot learning

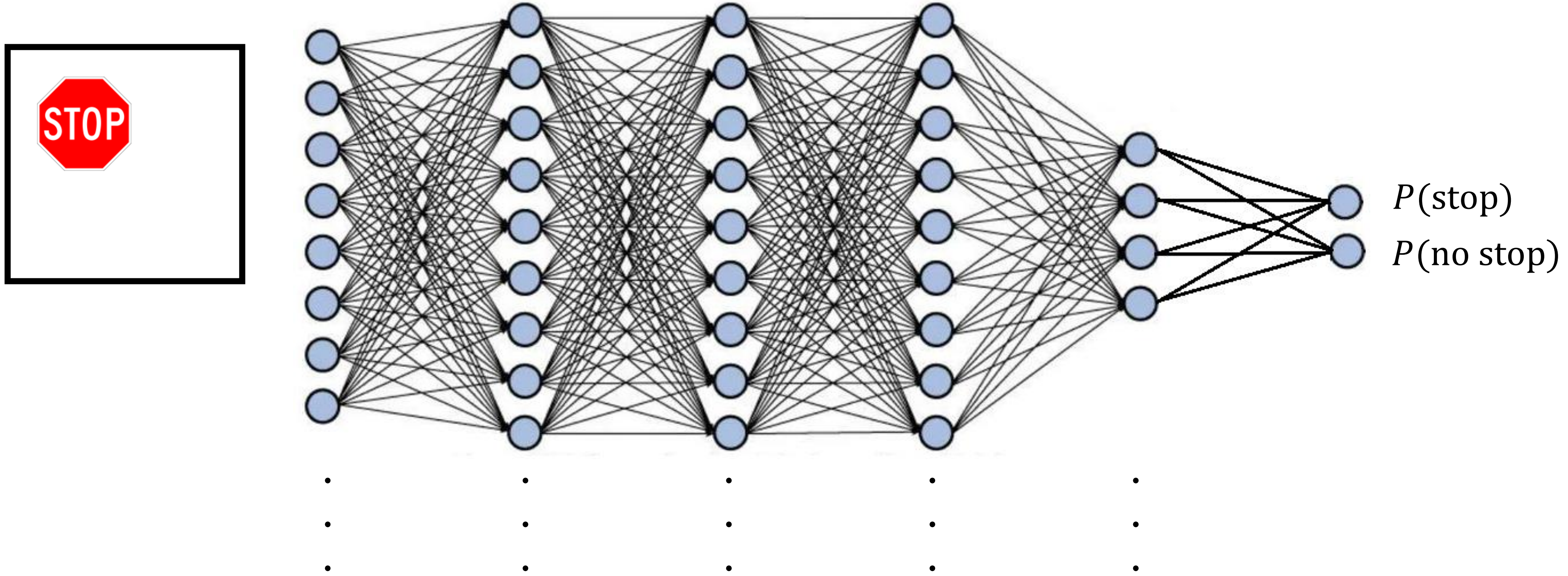
We are here today



Autonomous Mobile Robots
Siegwart et al.

Neural networks for images

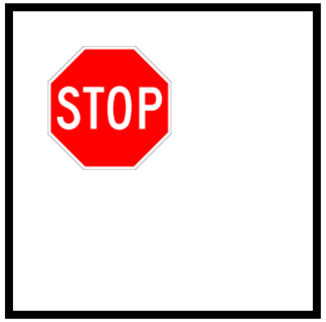
Easy "solution": flatten the image and make it a vector.

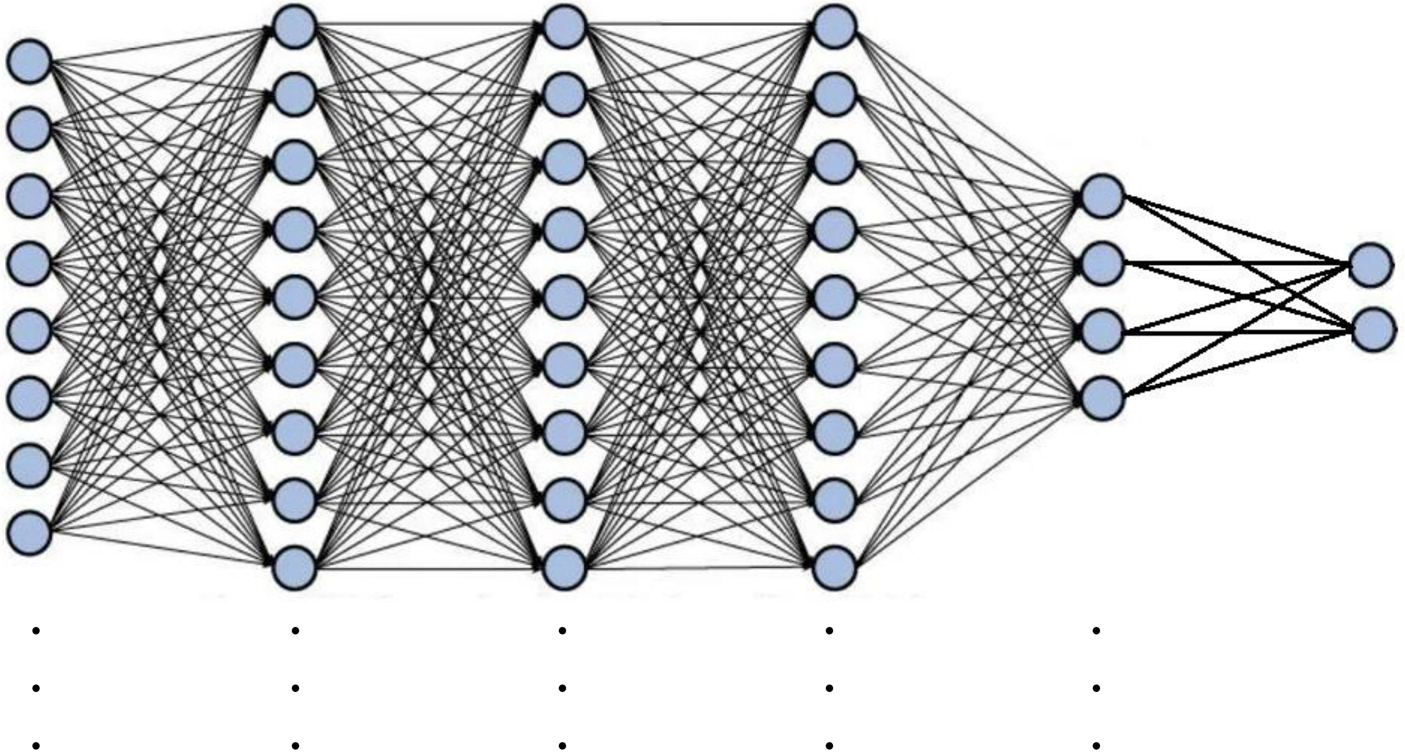


Neural networks for images

Easy “solution”: flatten the image and make it a vector.

This is a stop sign.

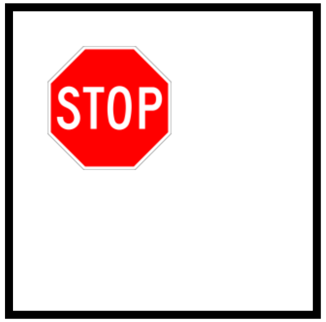




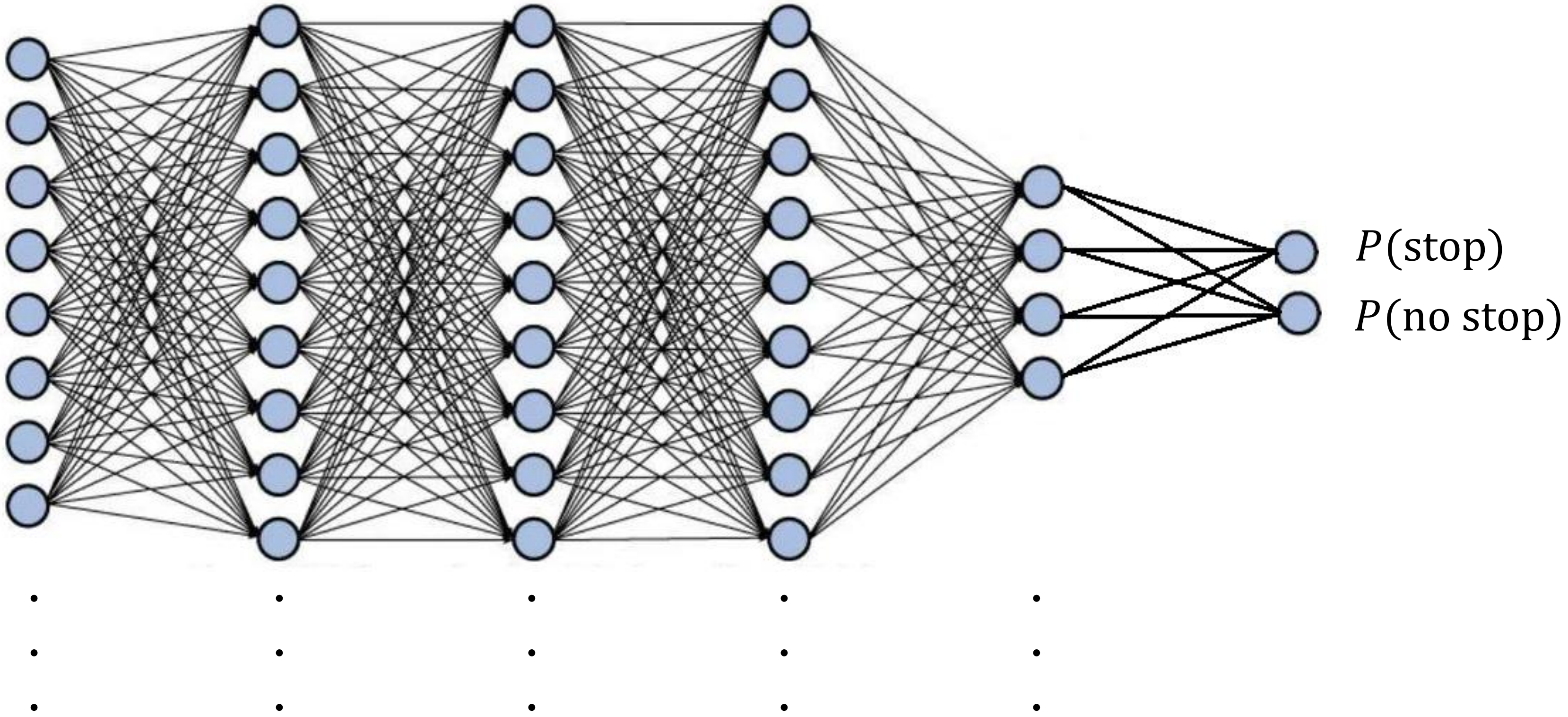
Neural networks for images

Easy "solution": flatten the image and make it a vector.

This is a stop sign.



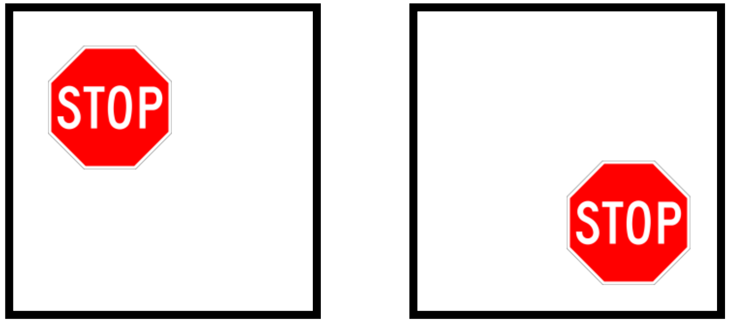
This is also a stop sign!



Neural networks for images

Regardless of it is location in the image, 

is a stop sign.



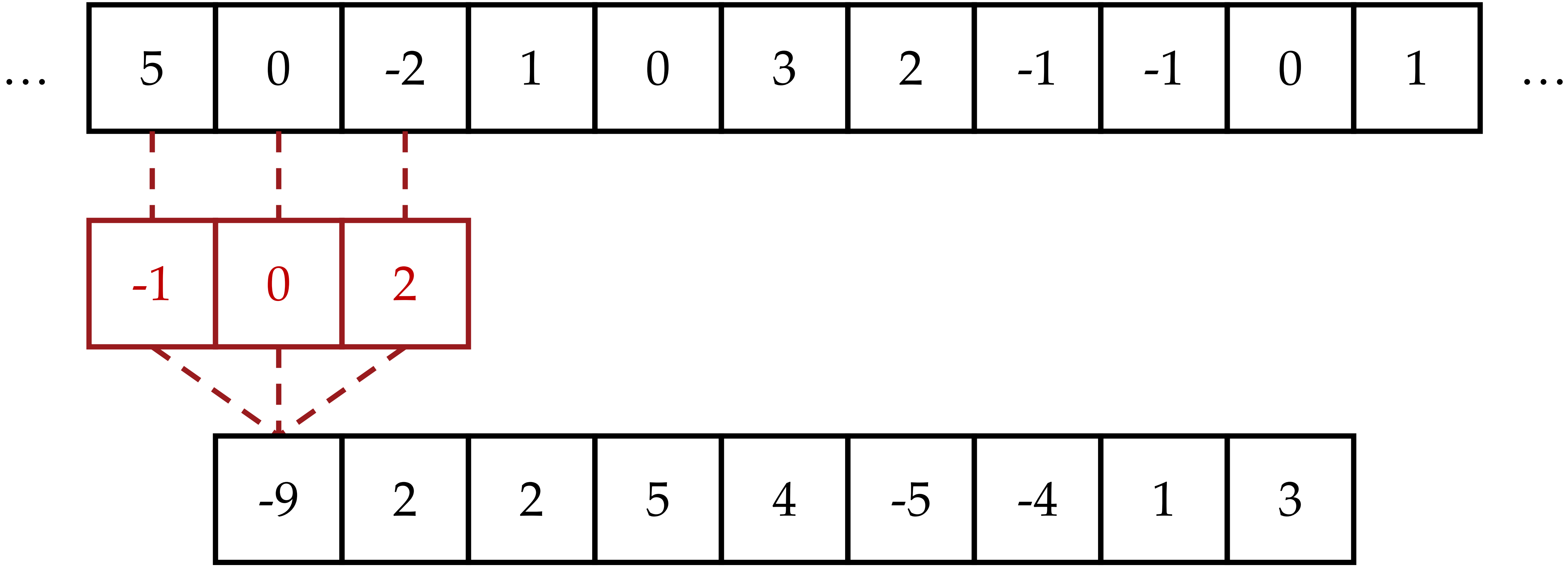
Spatial locality → weight sharing

Convolution

…	5	0	-2	1	0	3	2	-1	-1	0	1	…

2	0	-1

Convolution



Why is convolution important?

Many major technological developments in the past 50 years somehow relates to convolution.

- Used in: image processing, machine learning, telecommunications, audio processing, medical imaging, radar/sonar systems, seismology, quantum computing, optics, data compression, 3D graphics…

Why is convolution important?

Many major technological developments in the past 50 years somehow relates to convolution.

- Used in: image processing, machine learning, telecommunications, audio processing, medical imaging, radar/sonar systems, seismology, quantum computing, optics, data compression, 3D graphics…

Became especially popular after James Cooley and John Tukey developed Fast Fourier Transform (FFT) in 1965.

Reduces the computational complexity of convolution from $O(NM)$ to $O(N \log N)$

What's more?

FFT is *"the most important numerical algorithm of our lifetime."*

Gilbert Strang, 1994

Top 10 algorithms of the 20th century, according to IEEE's Computing in Science & Engineering magazine:

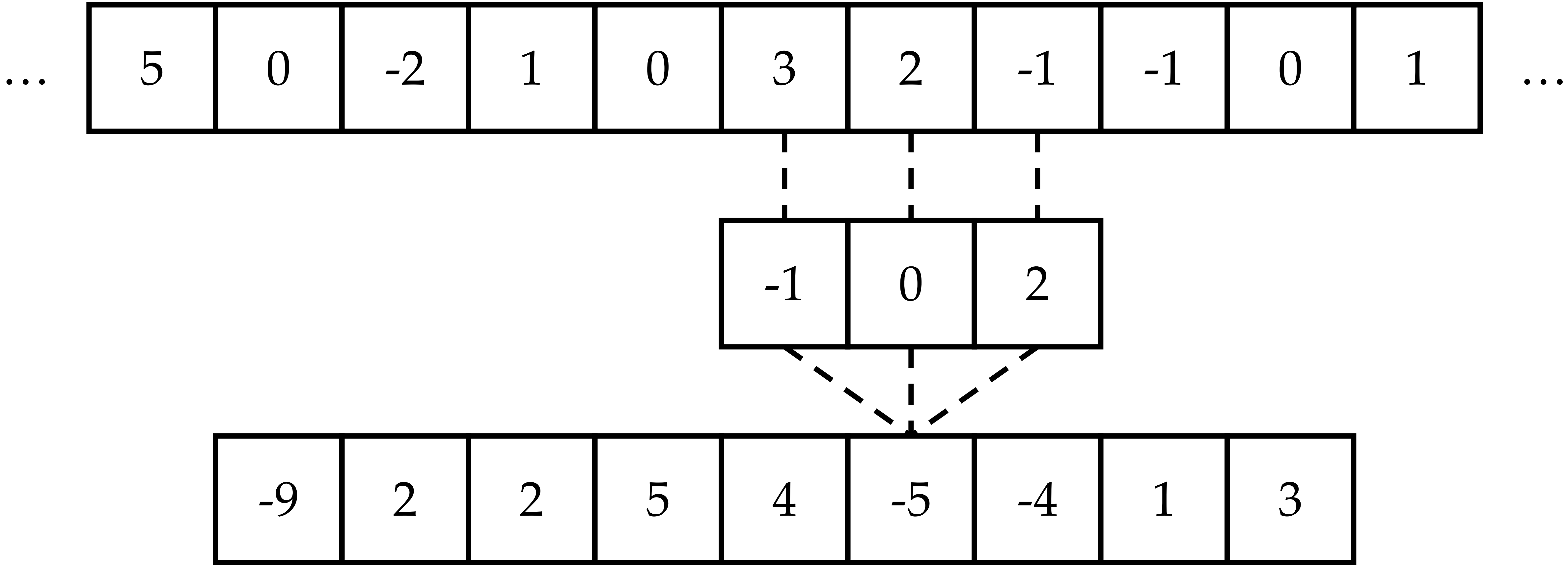
- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier Transform
- Integer Relation Detection
- Fast Multipole Method

Guest editors introduction to the top 10 algorithms
Dongarra and Sullivan, Computing in Science & Engineering, 2000

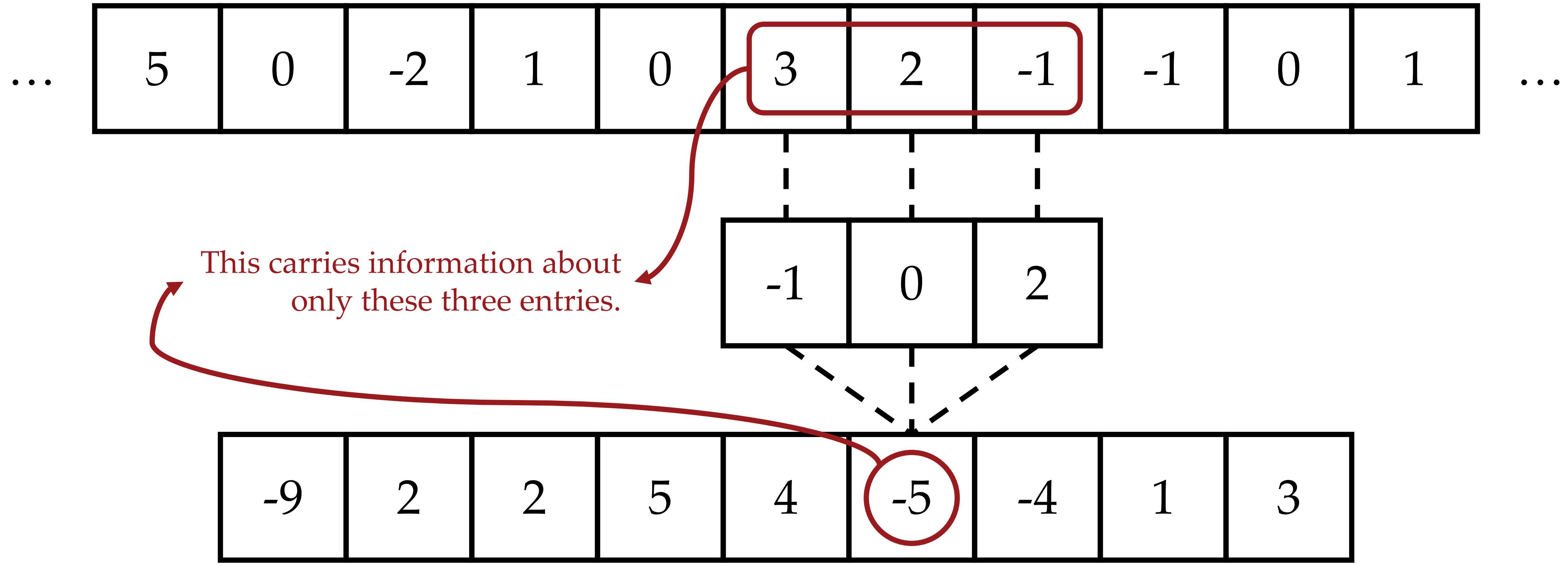
More on convolution and FFT

- EE 301L: Linear Systems
- EE 483: Introduction to Digital Signal Processing
- EE 434Lx: Digital Signal Processing Design Laboratory
- BME 413: Bioengineering Signals and Systems

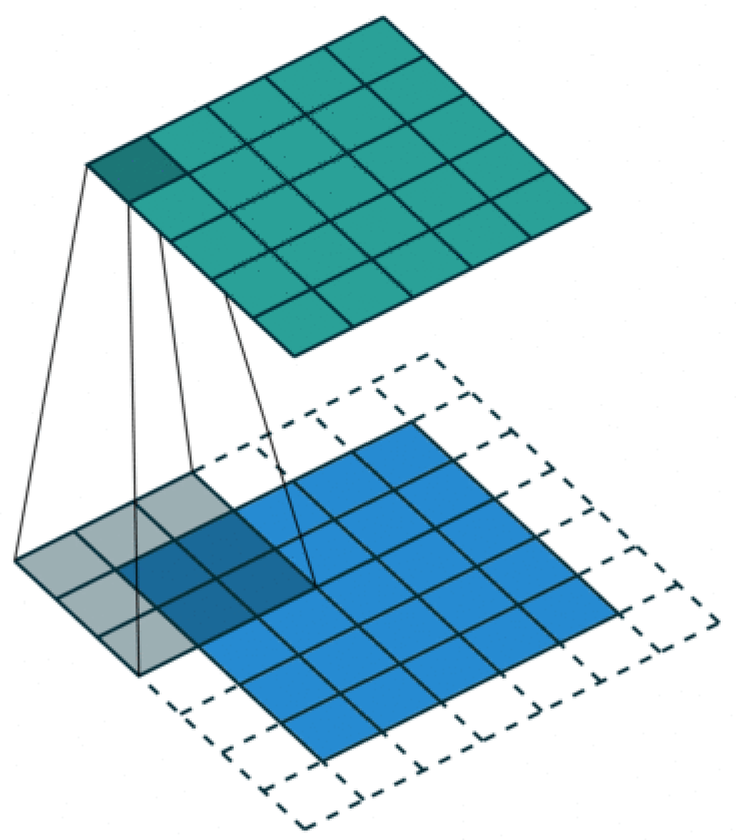
Convolution



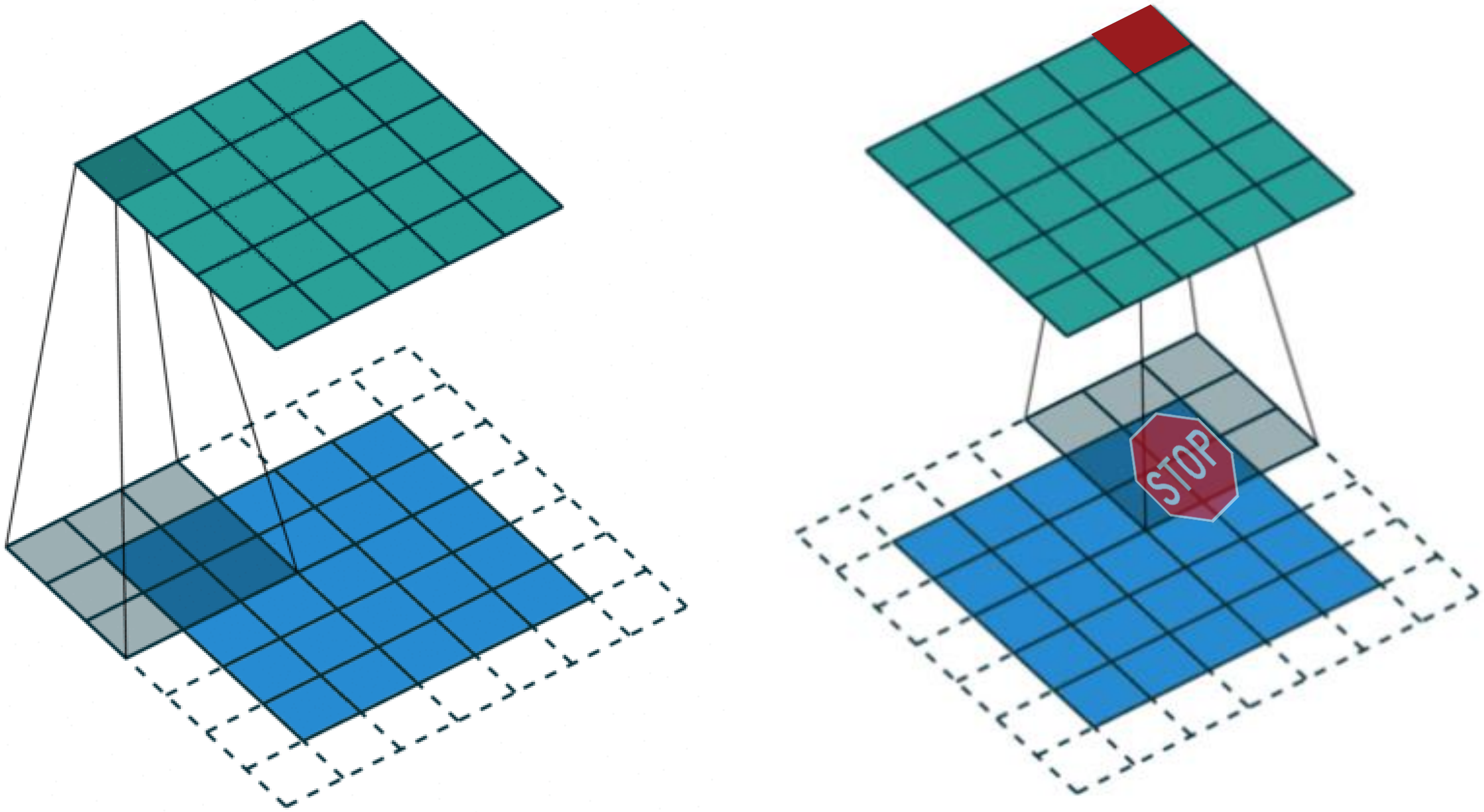
Convolution



Convolution in 2D

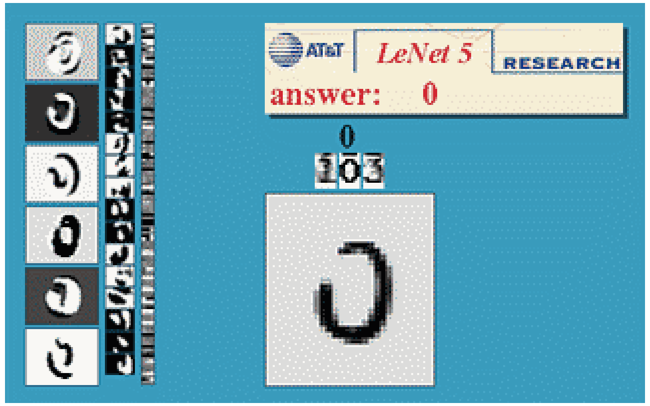


Convolution in 2D

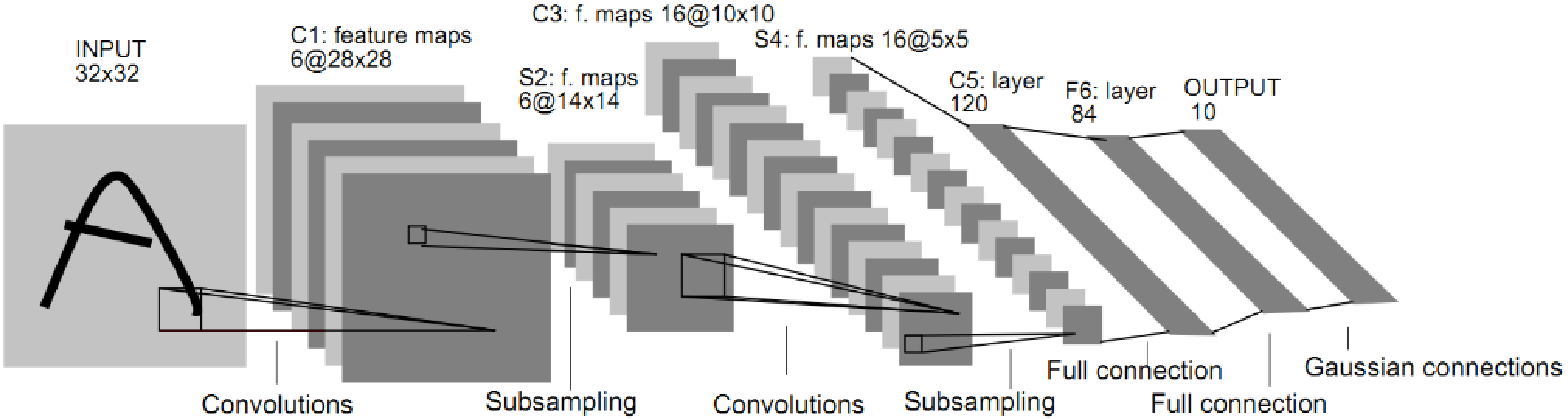


Convolutional neural networks (CNN)

- Traditionally consists of:
  - Convolutional layers
  - Nonlinearity layers
  - Pooling layers
  - Fully-connected layers

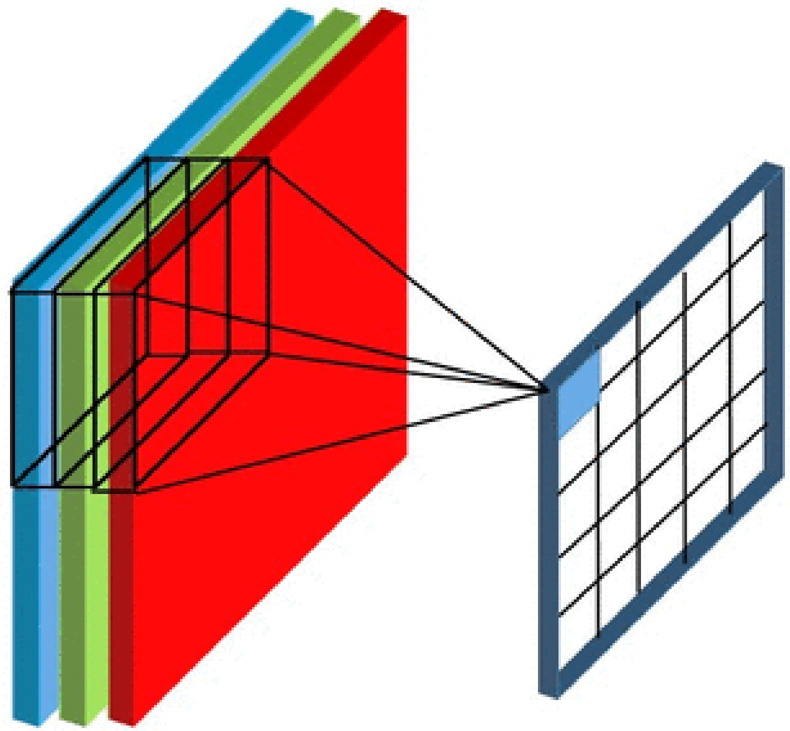


LeNet
(1998)



Gradient-based learning applied to document recognition
LeCun et al., Proceedings of the IEEE, 1998
Slide from: Marco Pavone (Stanford)

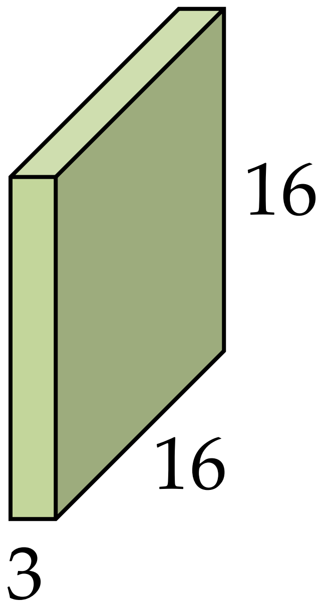
Convolutional layer



From: Bilel Guetarni, (École Centrale de Lille)

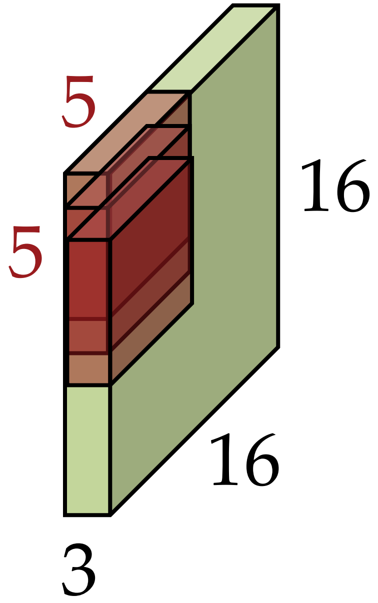
Convolutional layer

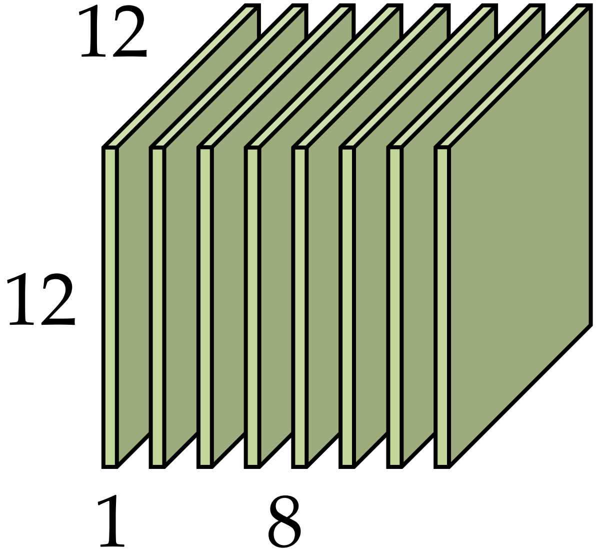
Let's say we have an image with size: (16, 16, 3) and 8 kernels where each kernel is (5, 5).



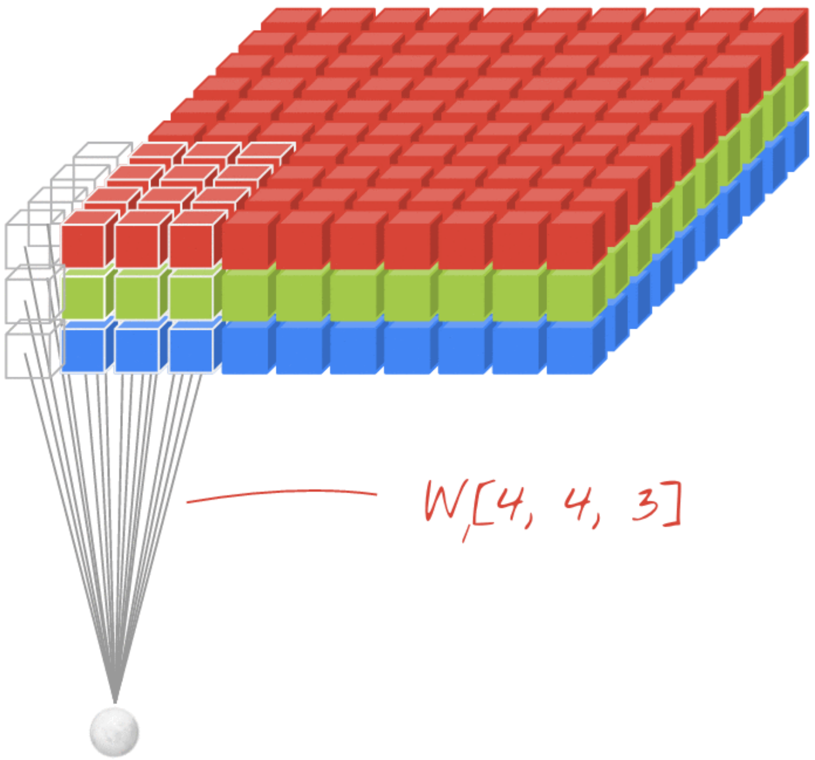
Convolutional layer

Let's say we have an image with size: (16, 16, 3) and 8 kernels where each kernel is (5, 5).





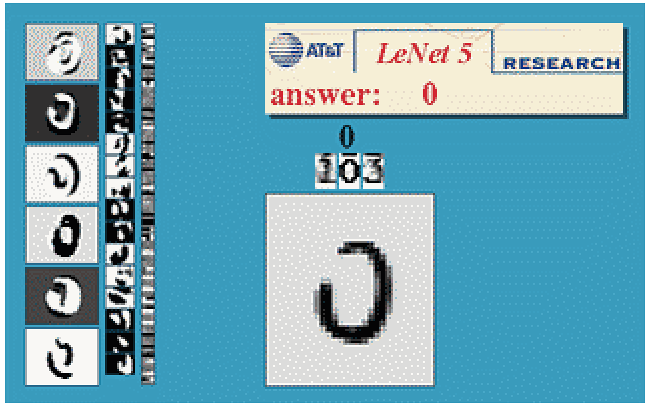
Padding and stride



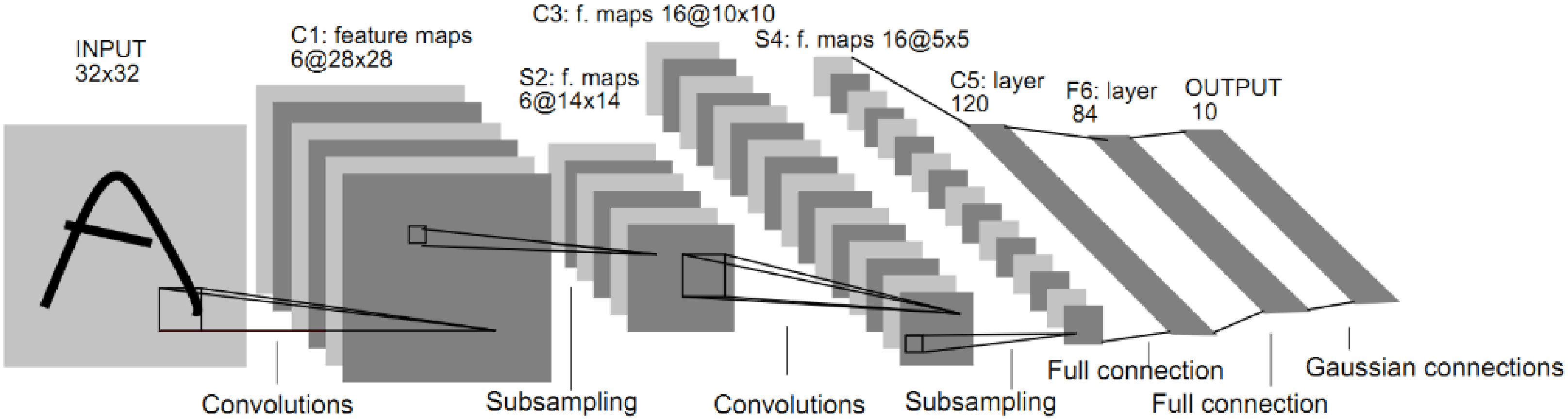
From: Mostafa Labib, StackOverflow

Convolutional neural networks (CNN)

- Traditionally consists of:
  - Convolutional layers
  - Nonlinearity layers
  - Pooling layers
  - Fully-connected layers

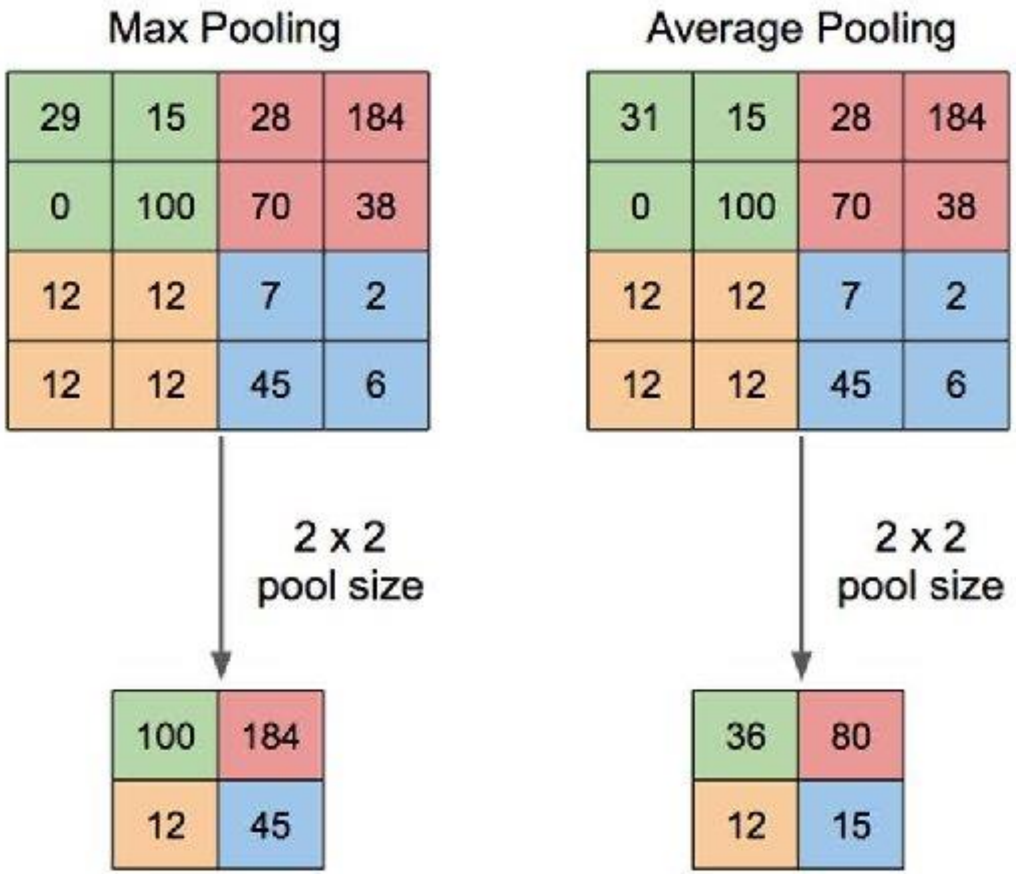


LeNet
(1998)



Gradient-based learning applied to document recognition
LeCun et al., Proceedings of the IEEE, 1998
Slide from: Marco Pavone (Stanford)

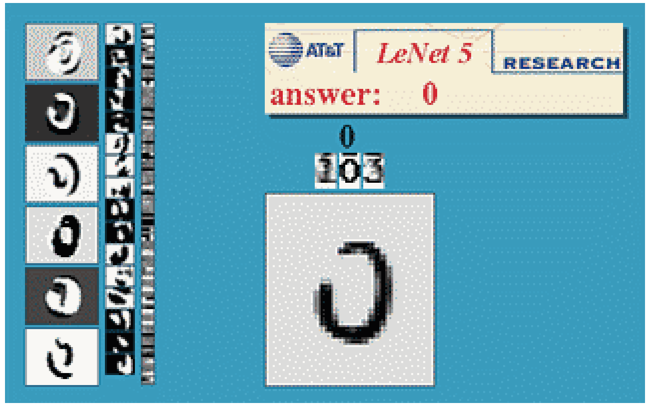
Pooling layer



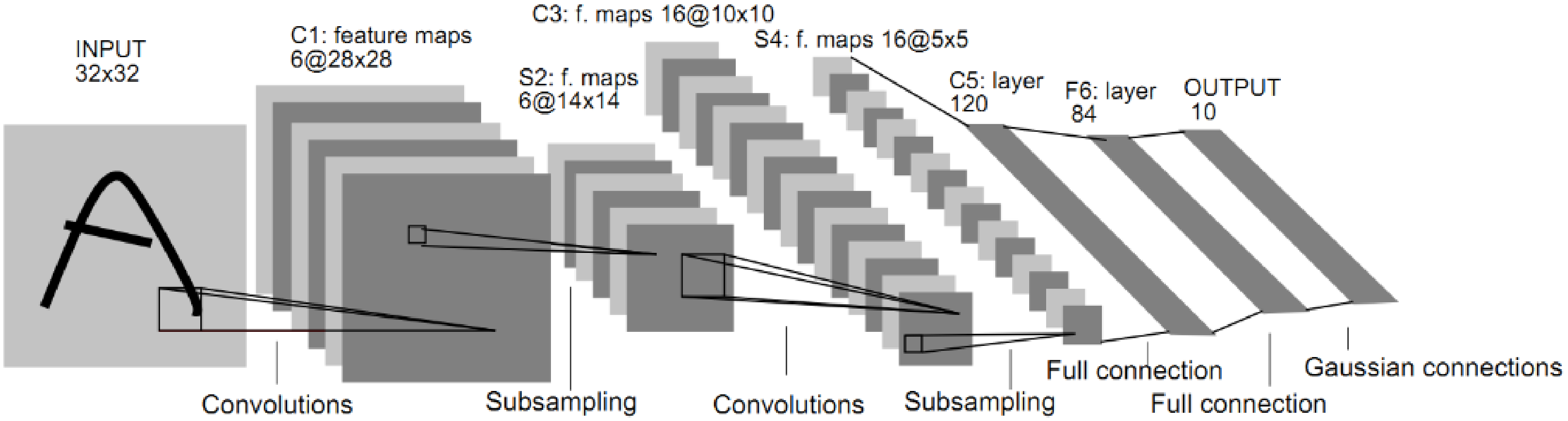
Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail
Yani et al., Journal of Physics Conference Series 2019.

Convolutional neural networks (CNN)

- Traditionally consists of:
  - Convolutional layers
  - Nonlinearity layers
  - Pooling layers
  - Fully-connected layers



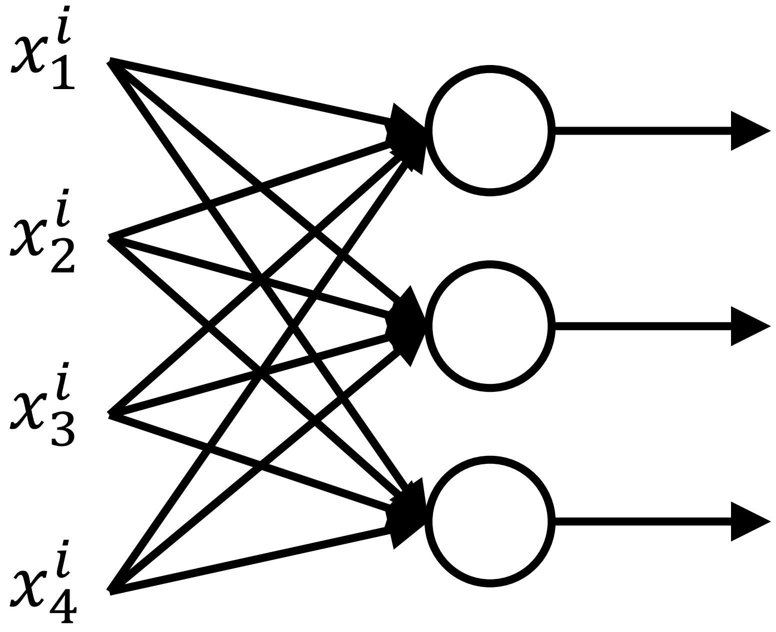
LeNet
(1998)



Gradient-based learning applied to document recognition
LeCun et al., Proceedings of the IEEE
Slide from: Marco Pavone (Stanford)

From lecture 1

A fully-connected layer



$$y_1^i = g\left(w_1^\top x^i + b_1\right)$$

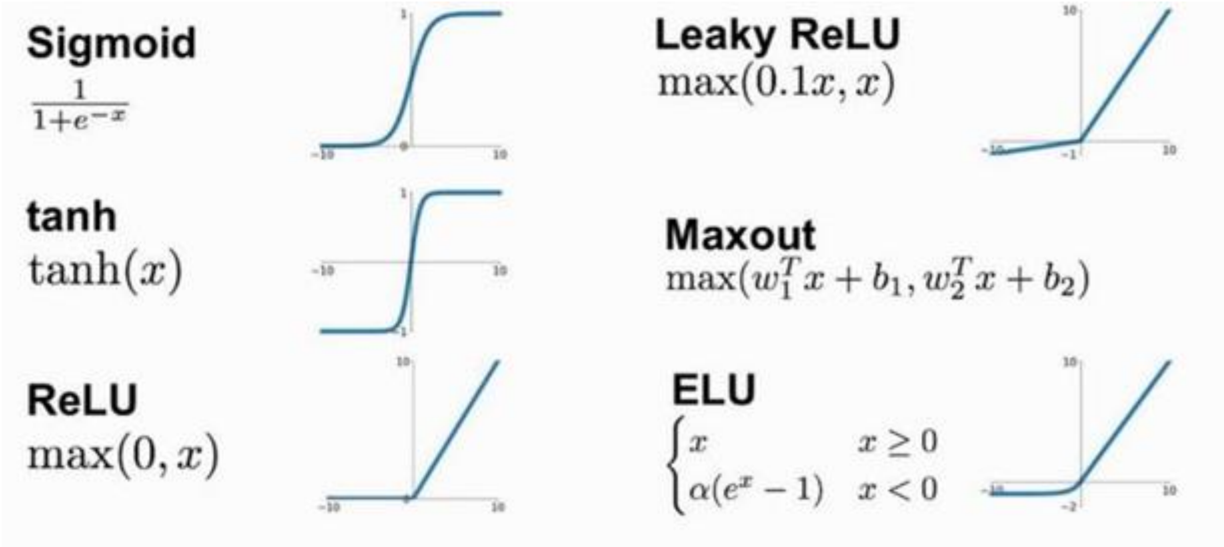
$$y_2^i = g\left(w_2^\top x^i + b_2\right)$$

$$y_3^i = g\left(w_3^\top x^i + b_3\right)$$

$$\Rightarrow \quad y = g(W^\top x + b)$$

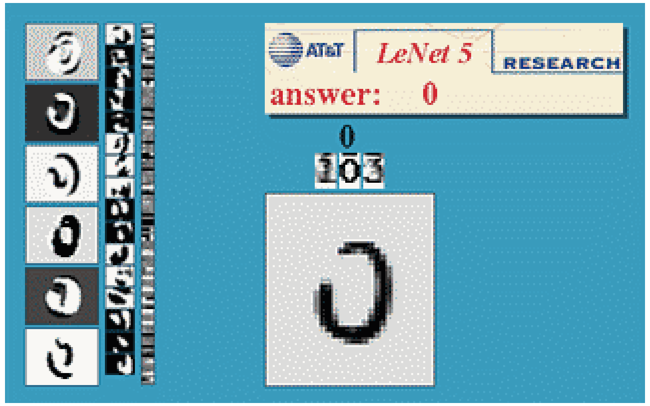
From lecture 1

Different types of nonlinearity layers:

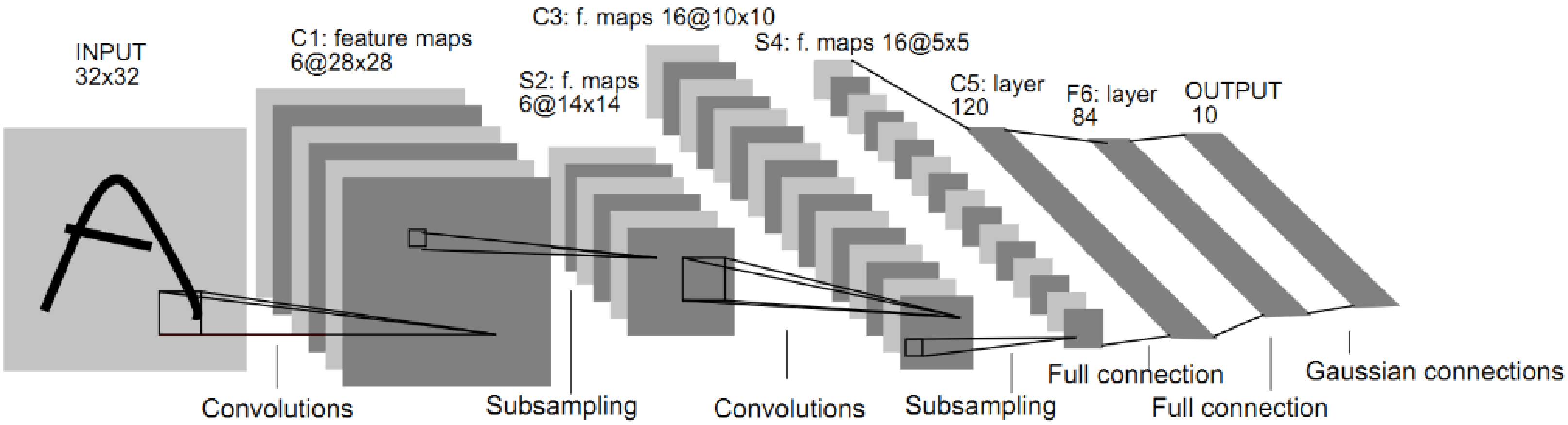


Convolutional neural networks (CNN)

- Traditionally consists of:
  - Convolutional layers
  - Nonlinearity layers
  - Pooling layers
  - Fully-connected layers



LeNet
(1998)



Gradient-based learning applied to document recognition
LeCun et al., Proceedings of the IEEE
Slide from: Marco Pavone (Stanford)

Classification with CNNs

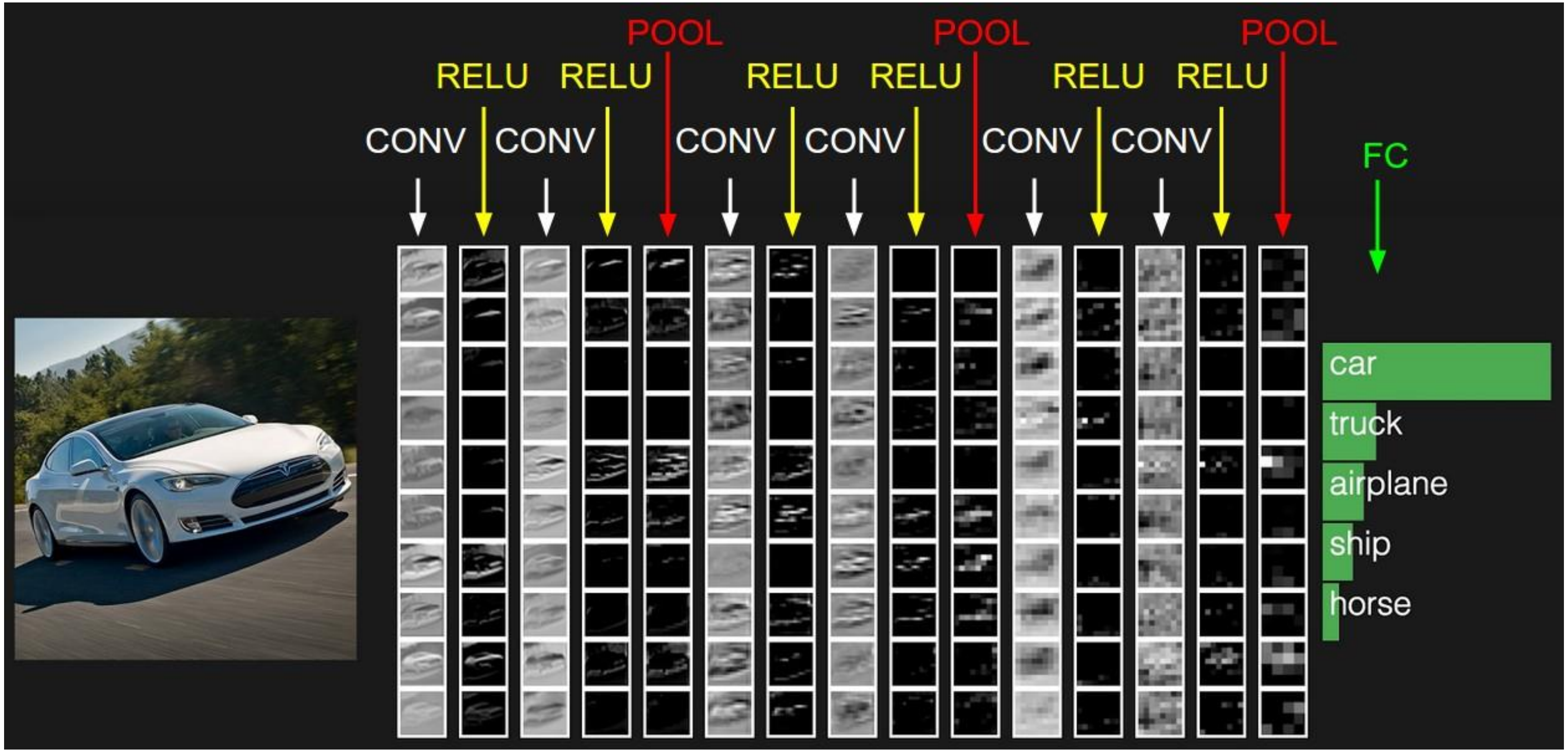
From: Richa Bhatia (Analytics India Magazine)

Classification via hand-crafted features

Uses image statistics and keypoints as features.

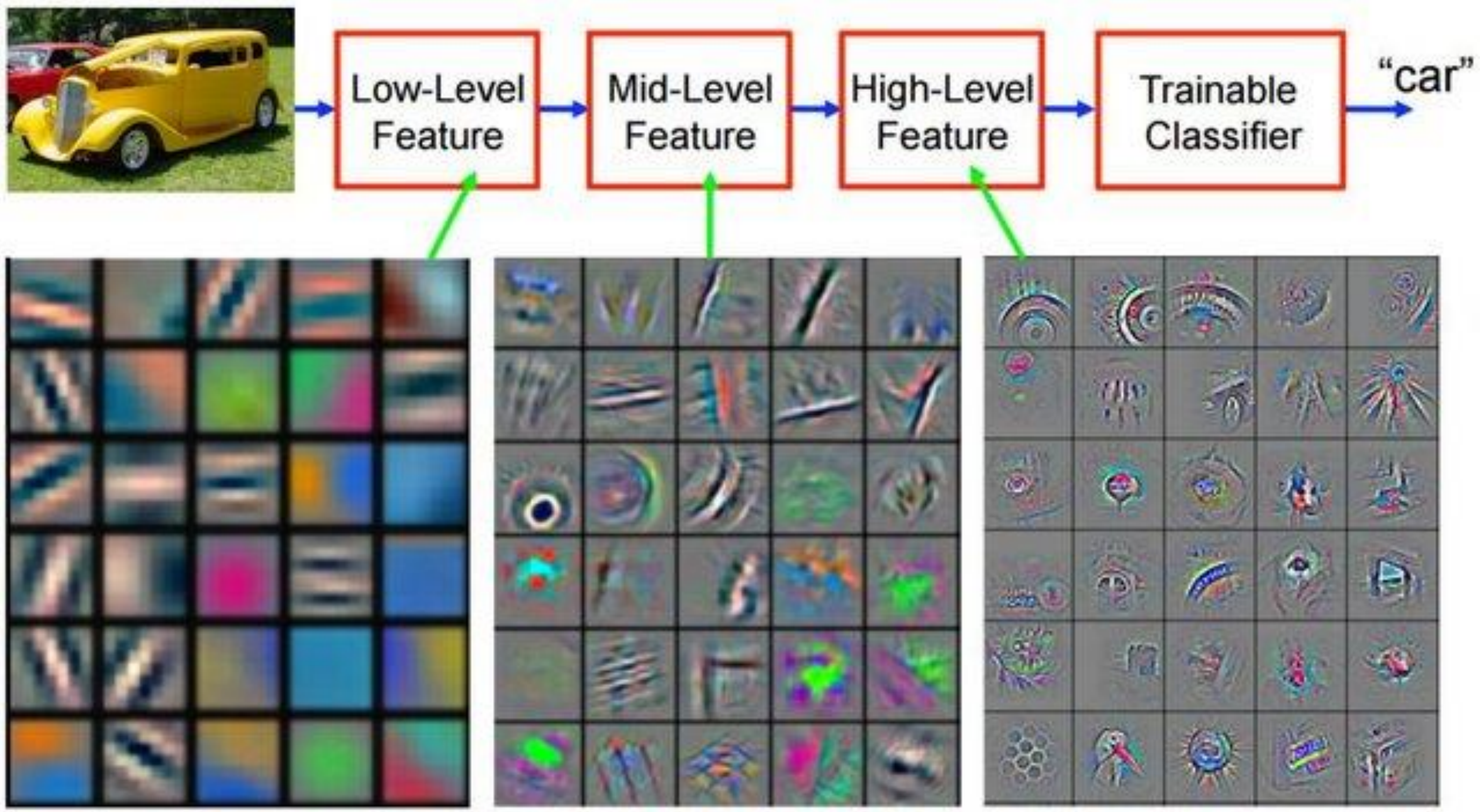
Trains a classifier over those features.

End-to-end training



From: Stanford CS231N

Visualizing CNN features



From: Stanford CS231N

Online CNN demo

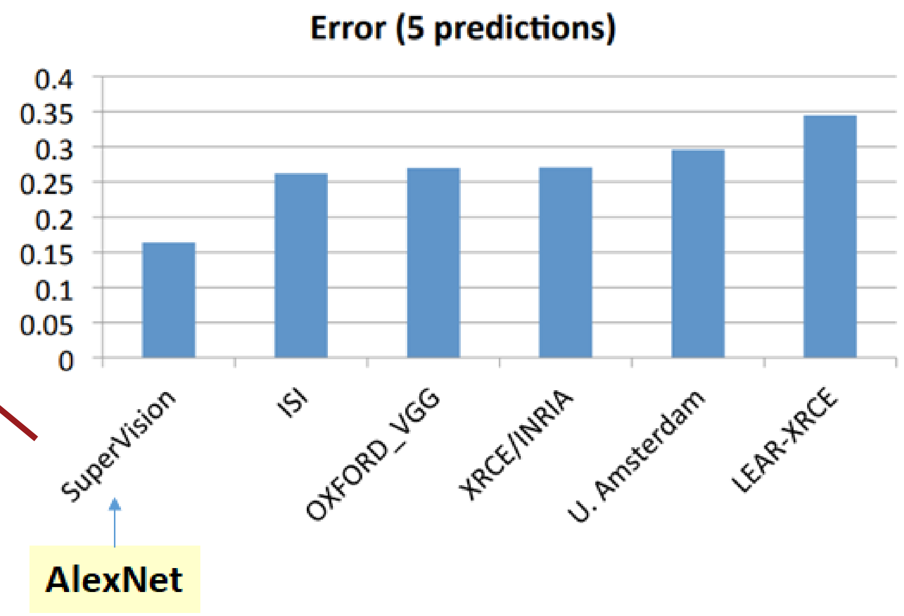
https://adamharley.com/nn_vis/

From: Adam Harley (Stanford)

Comparison

Team name	Filename	Error (5 guesses)	Description
SuperVision	test-preds-141-146.2009-131-137-145-146.2011-145f.	0.15315	Using extra training data from ImageNet Fall 2011 release
SuperVision	test-preds-131-137-145-135-145f.txt	0.16422	Using only supplied training data
ISI	pred_FVs_wLACs_weighted.txt	0.26172	Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
ISI	pred_FVs_weighted.txt	0.26602	Weighted sum of scores from classifiers using each FV.
ISI	pred_FVs_summed.txt	0.26646	Naive sum of scores from classifiers using each FV.
ISI	pred_FVs_wLACs_summed.txt	0.26952	Naive sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively.
OXFORD_VGG	test_adhocmix_classification.txt	0.26979	Mixed selection from High-Level SVM scores and Baseline Scores, decision is performed by looking at the validation performance

Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton



From: ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

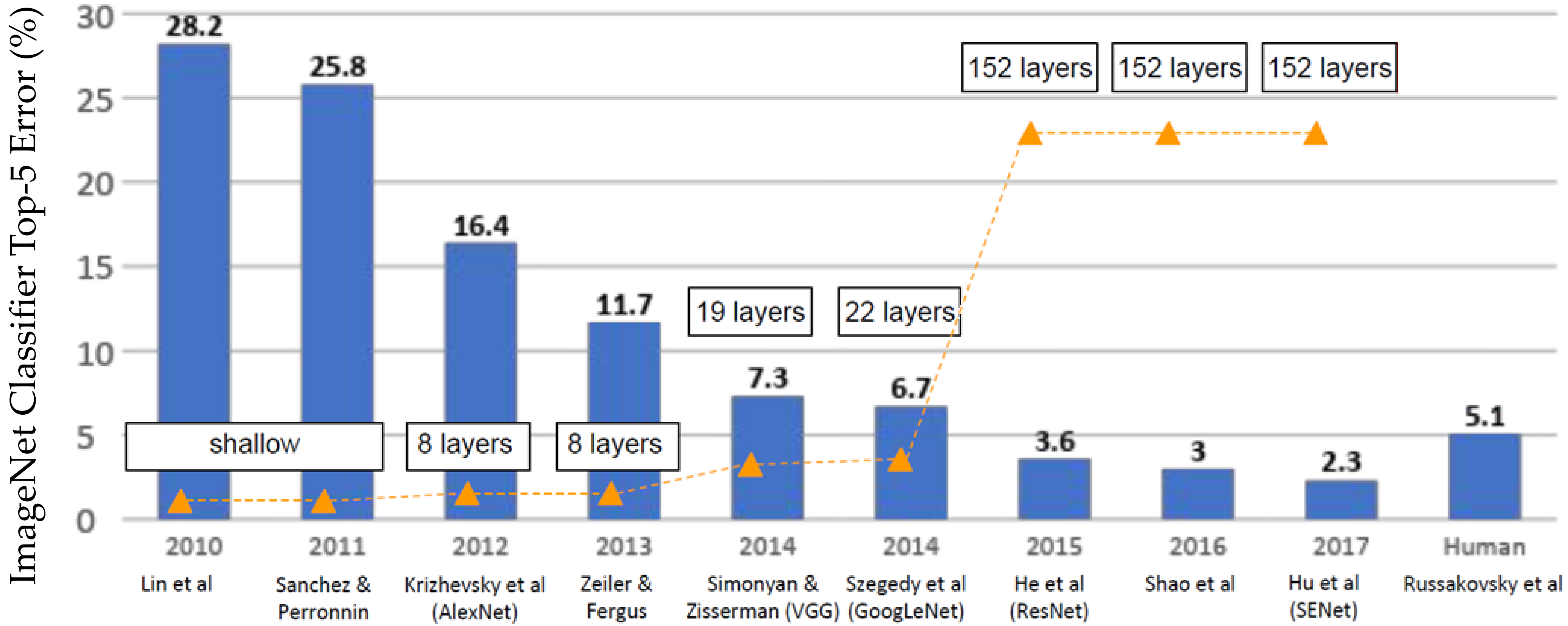
Comparison

The description of SuperVision:

*Our model is **a large, deep convolutional neural network** trained on raw RGB pixel values. The neural network, which has **60 million parameters and 650,000 neurons, consists of five convolutional layers**, some of which are followed by max-pooling layers, and three globally-connected layers with a final 1000-way softmax. It was trained on two NVIDIA GPUs for about a week. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of convolutional nets. To reduce overfitting in the globally-connected layers we employed hidden-unit "dropout", a recently-developed regularization method that proved to be very effective.*

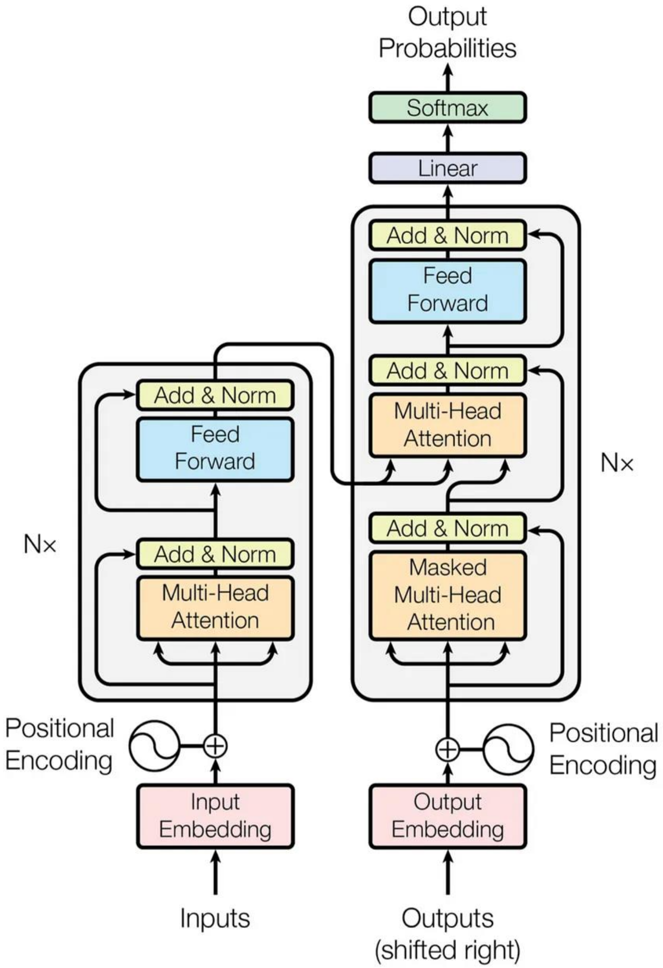
From: ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

Deeper and deeper



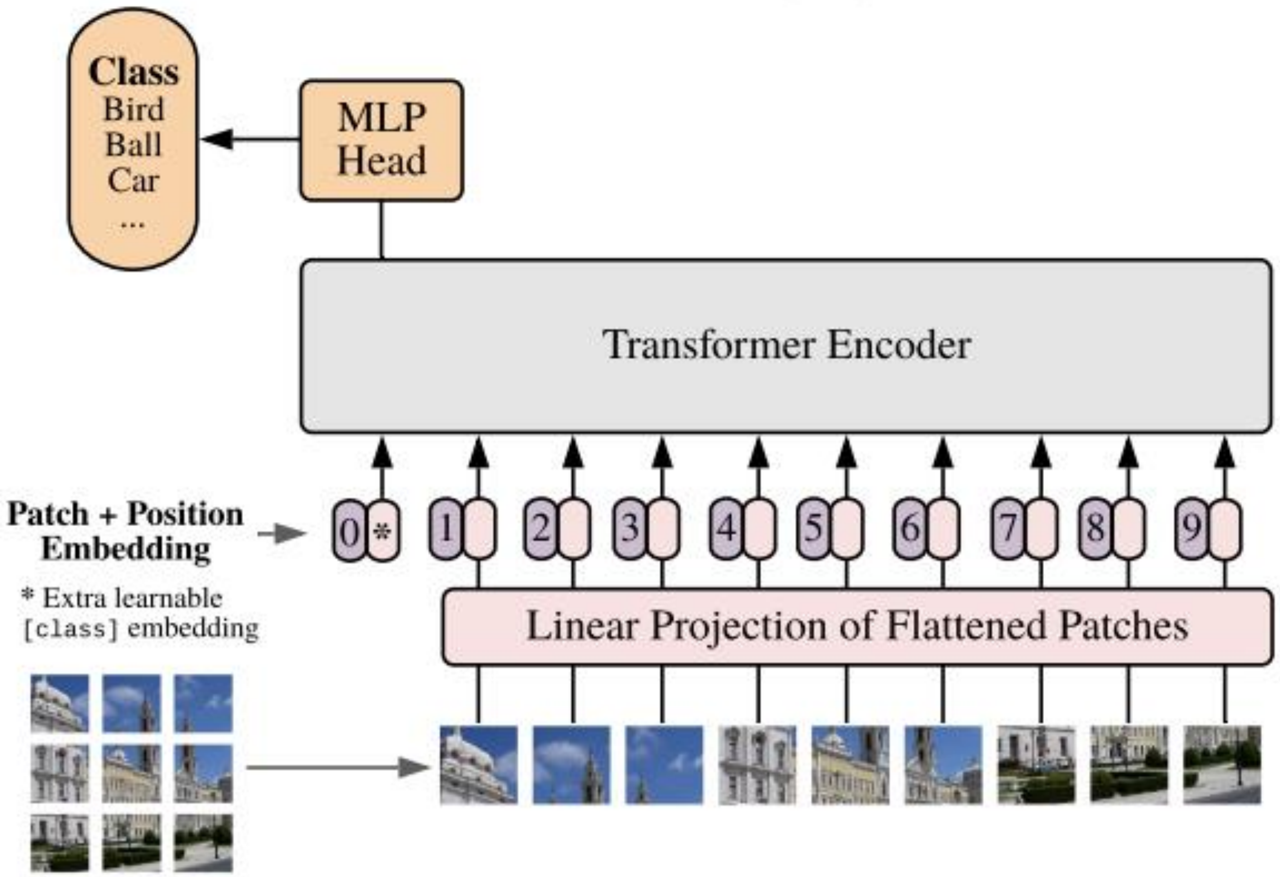
From: Stanford CS231N

Transformer



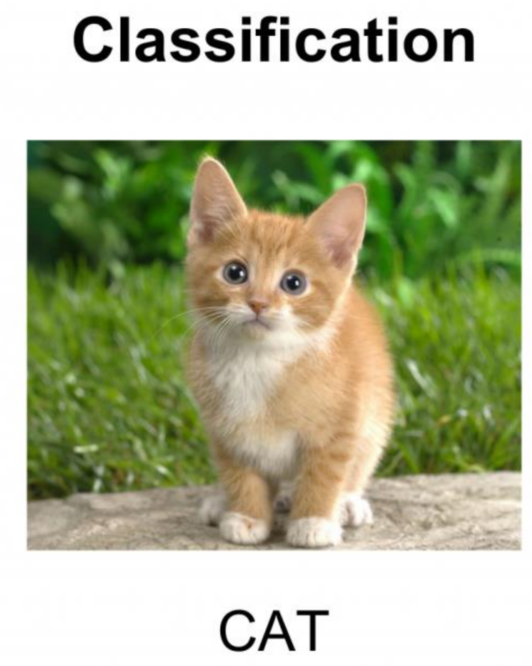
Attention is all you need
Vaswani et al., NeurIPS 2017

Vision transformers (ViT)



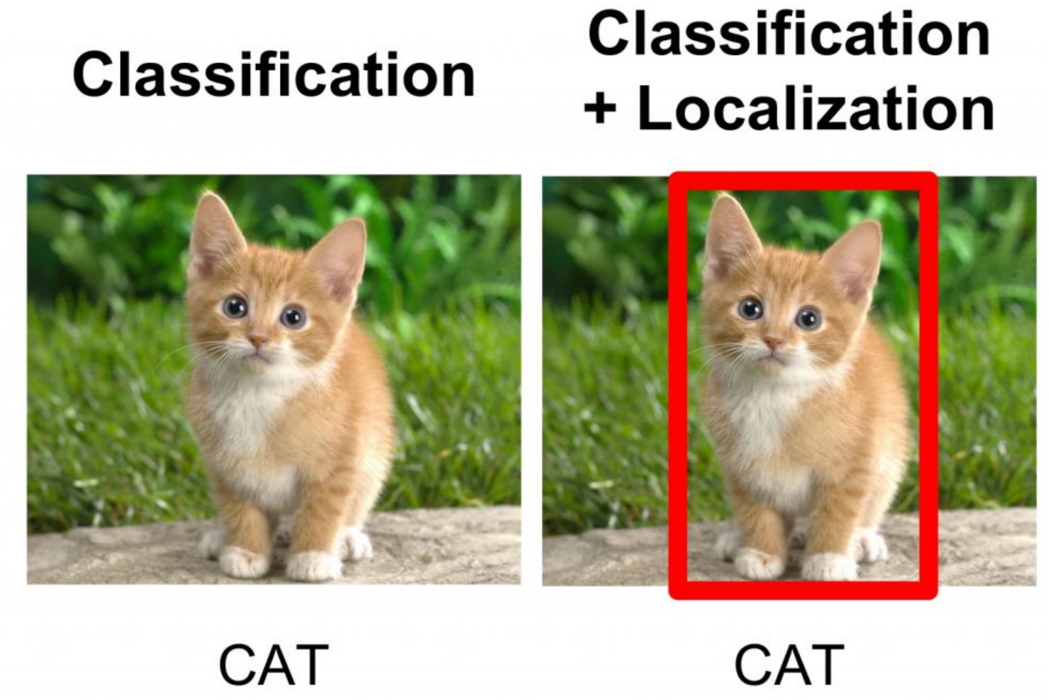
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale
Dosovitskiy et al., ICLR 2021

Robotics applications of vision



From: Richa Bhatia (Analytics India Magazine)

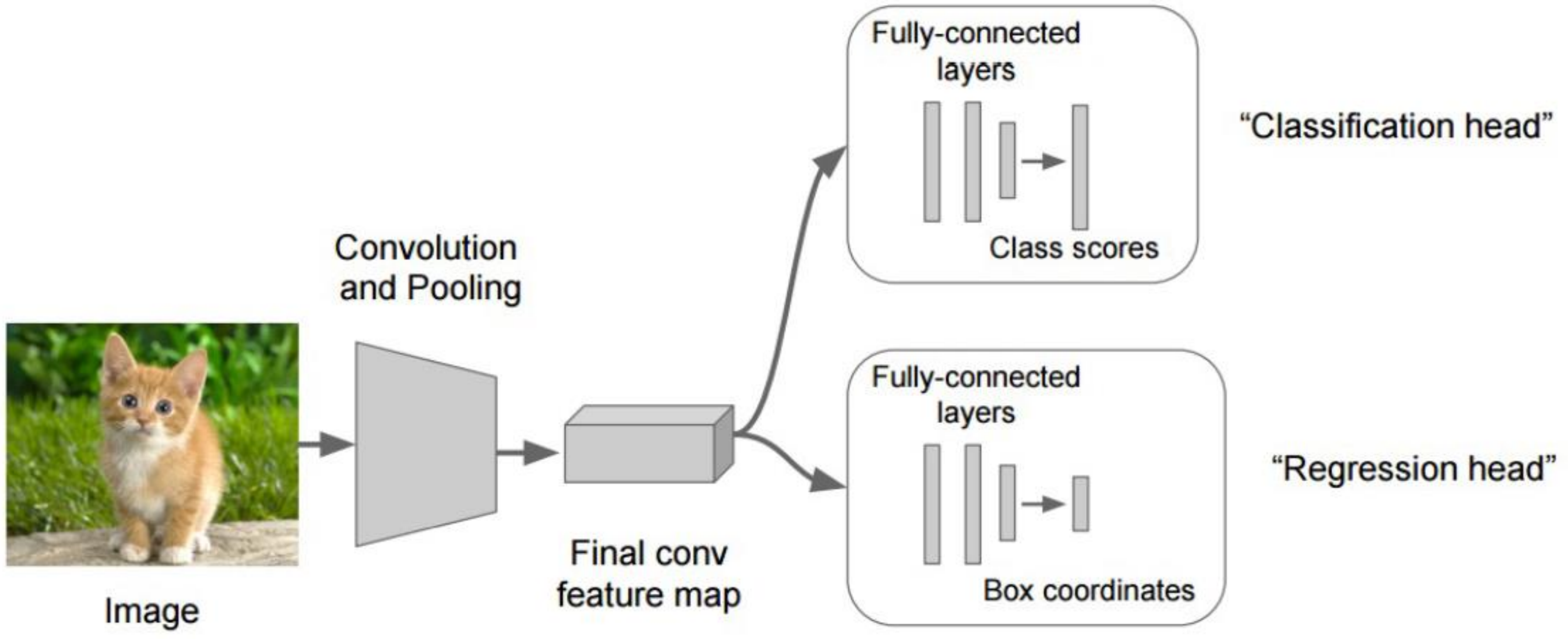
Robotics applications of vision



From: Richa Bhatia (Analytics India Magazine)

Classification + Localization

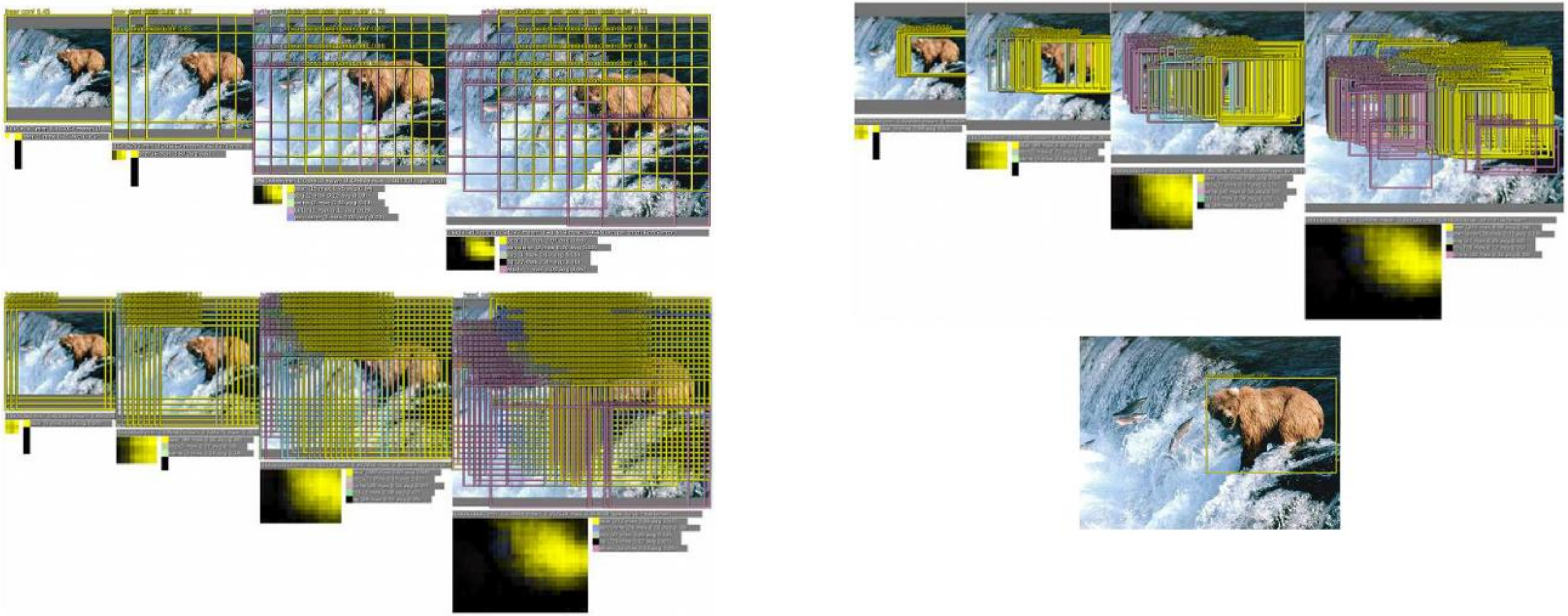
Do not just output the image class.
Output the class and 4 numbers, one for each corner of the bounding box.



From: Stanford CS231N

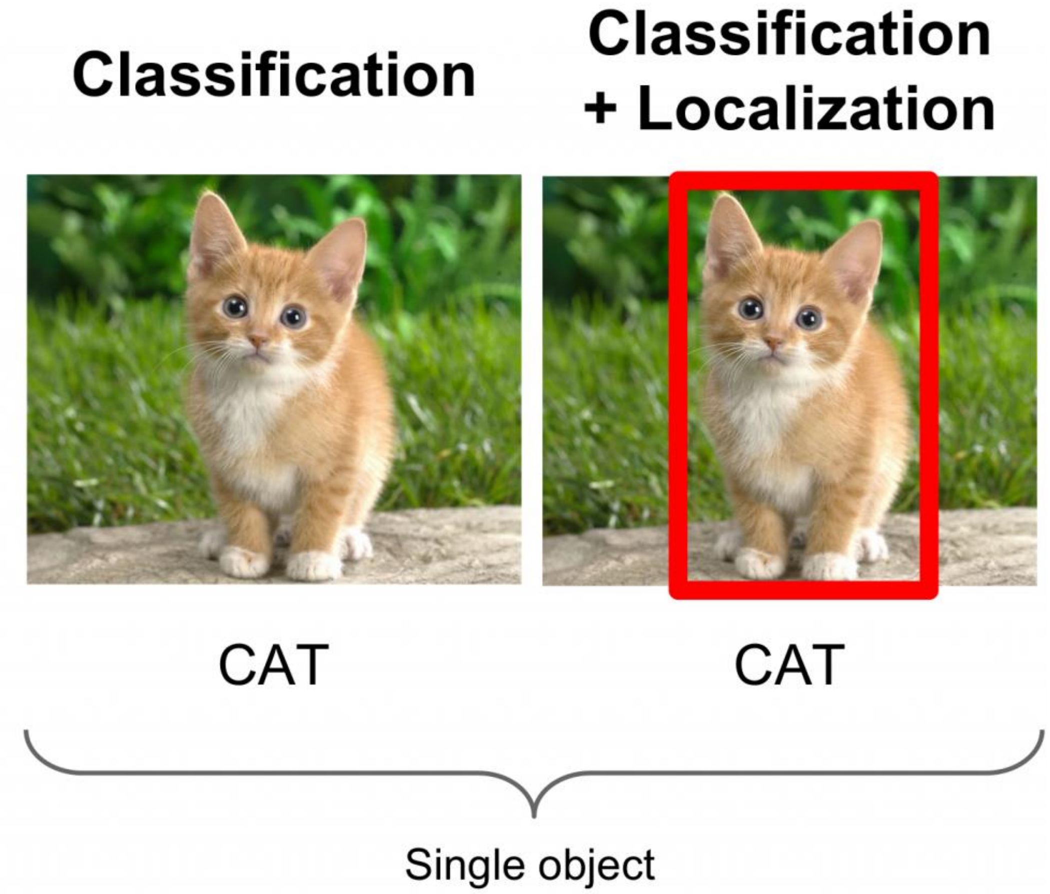
Classification + Localization

Or just use a classifier multiple times.



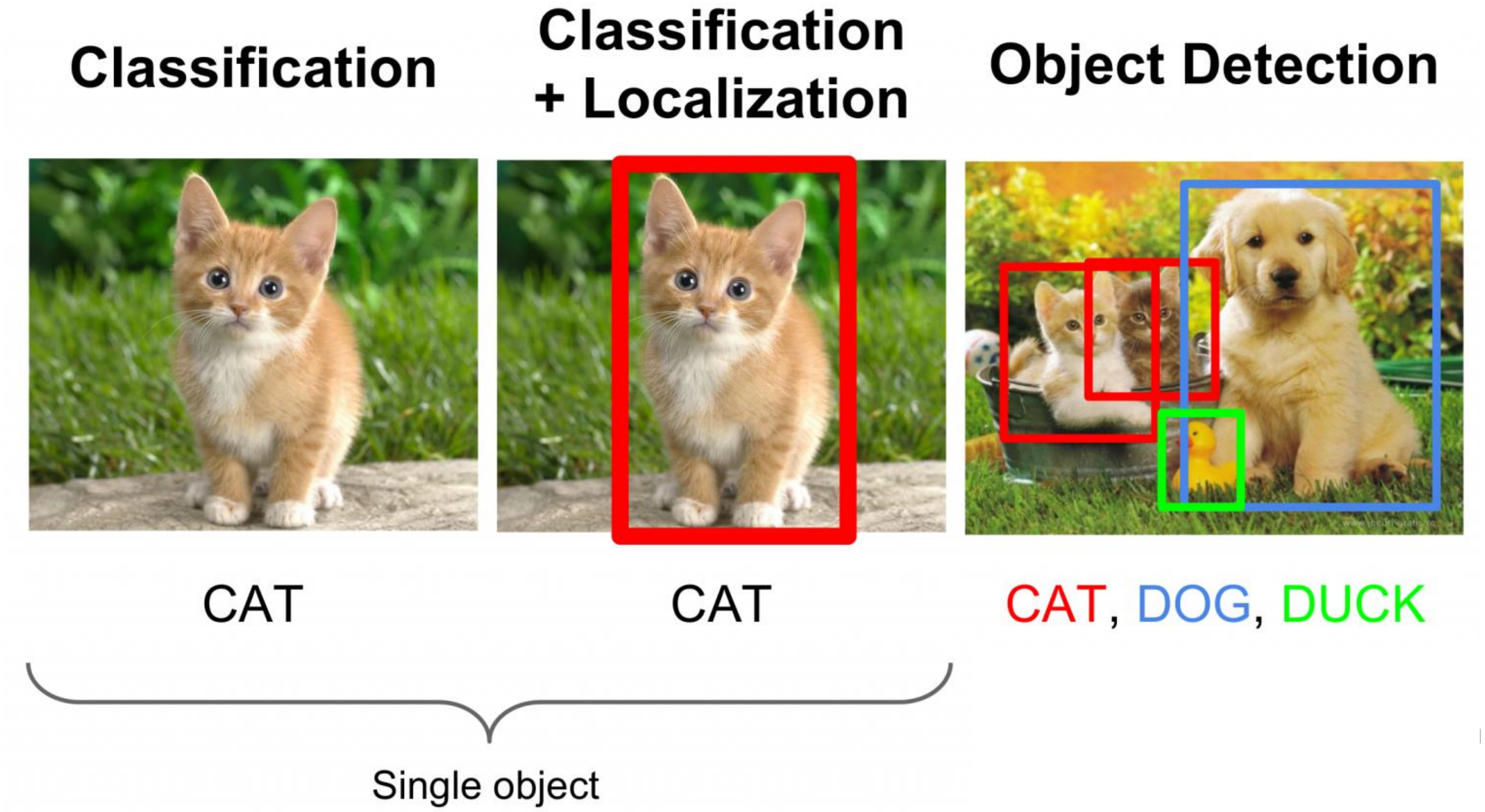
OverFeat: Integrated Recognition, Localization and
Detection using Convolutional Networks
Sermanet et al., ICLR 2014

Robotics applications of vision

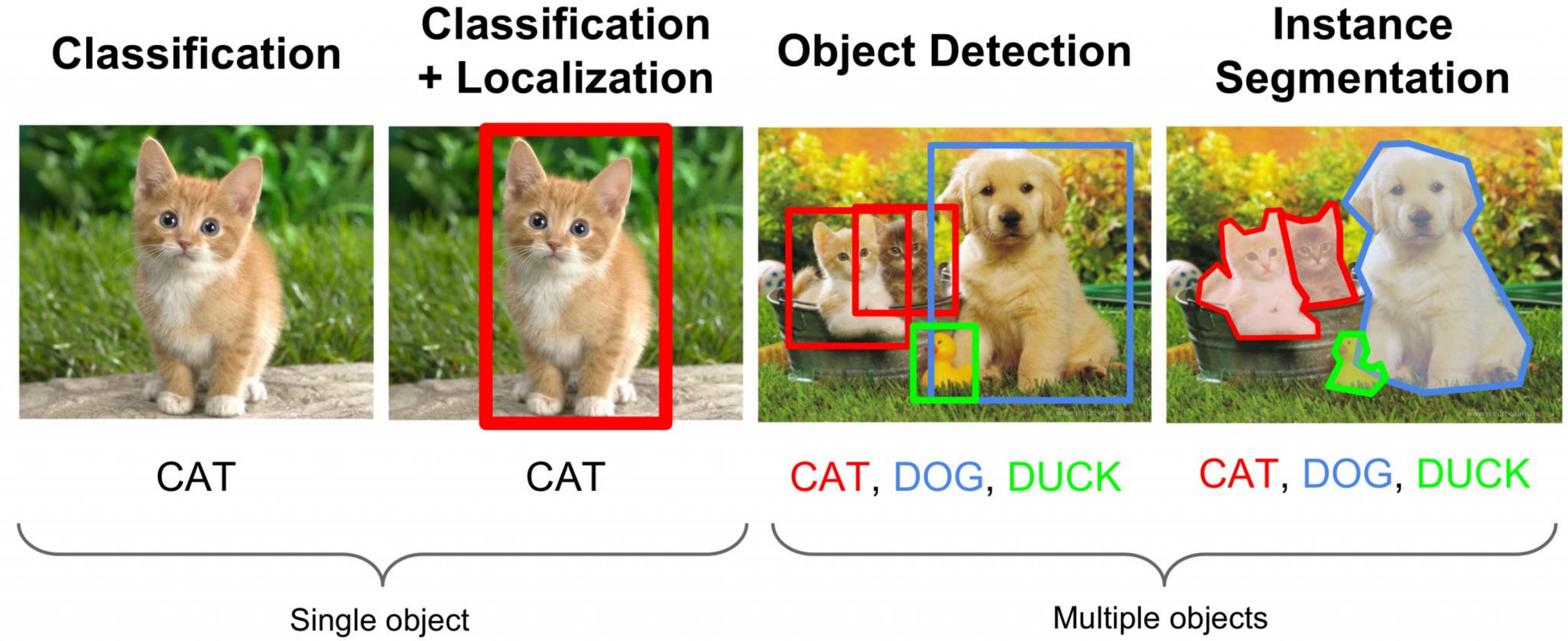


From: Richa Bhatia (Analytics India Magazine)

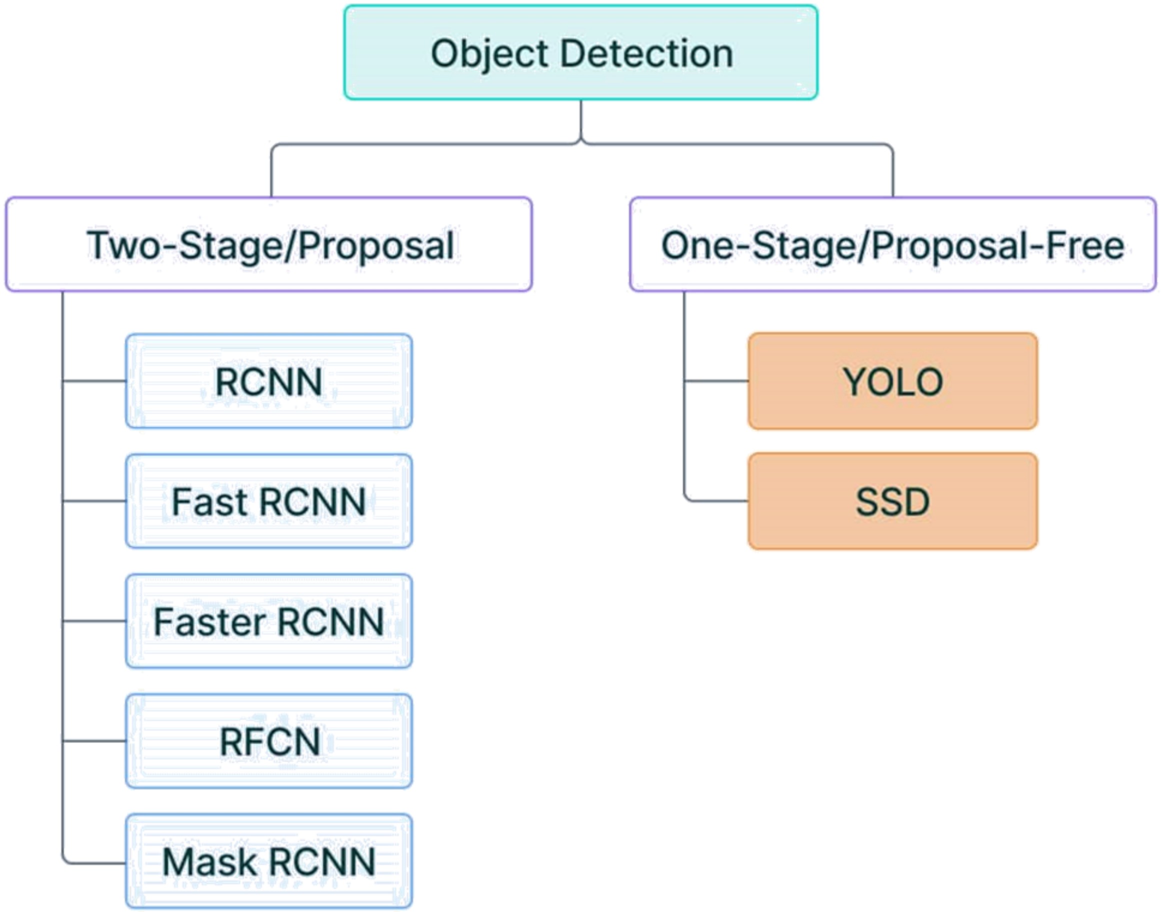
Robotics applications of vision



Robotics applications of vision



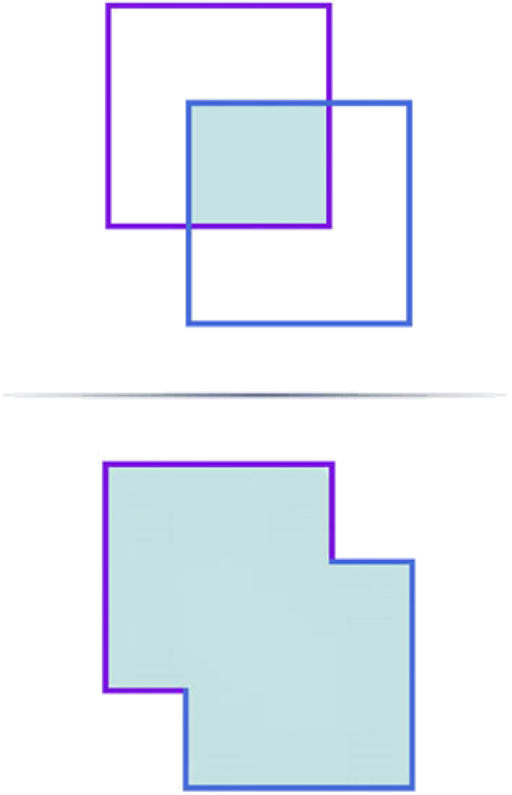
Object detection



From: Rohit Kundu (v7 Labs)

Metric for object detection

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} =$$



From: Rohit Kundu (v7 Labs)

Object detection with YOLO

Divide image into $S \times S$ grid. For each cell, predict:

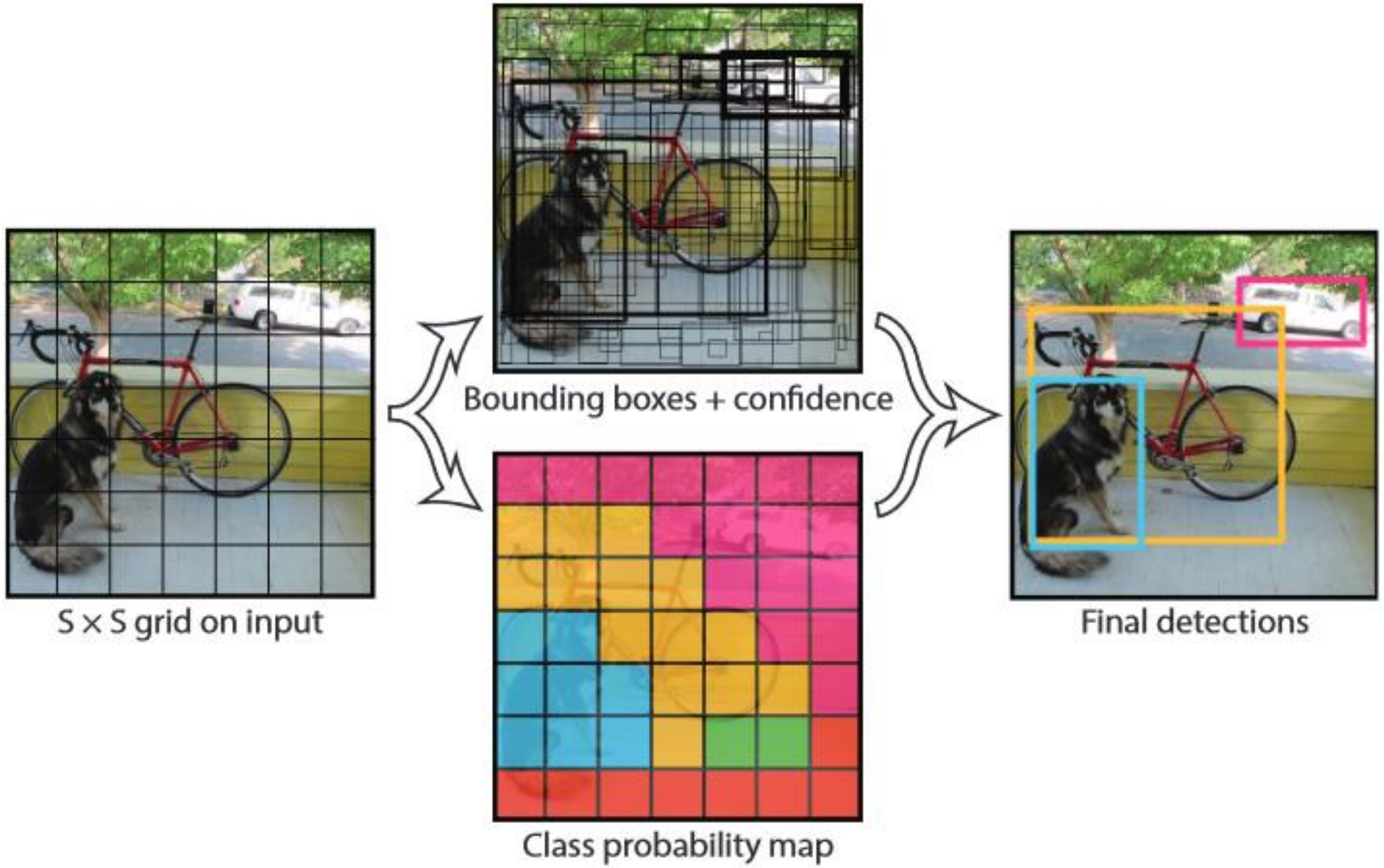
- B boxes with 4 coordinates and 1 confidence score each, and
- C class scores.

This gives a tensor of $S \times S \times (5B + C)$.

Apply non-maximum suppression to finalize.

You Only Look Once: Unified, Real-Time Object Detection
Redmon et al., CVPR 2016

Object detection with YOLO

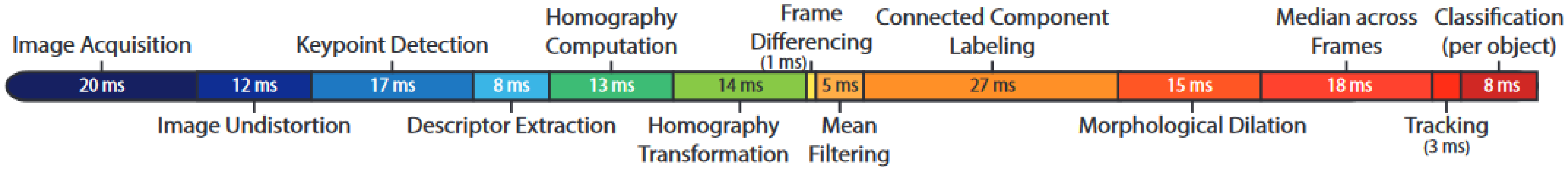


You Only Look Once: Unified, Real-Time Object Detection
Redmon et al., CVPR 2016

Object detection in robotics

Speed is crucial – we need to detect, track, and classify many objects with high frequency.

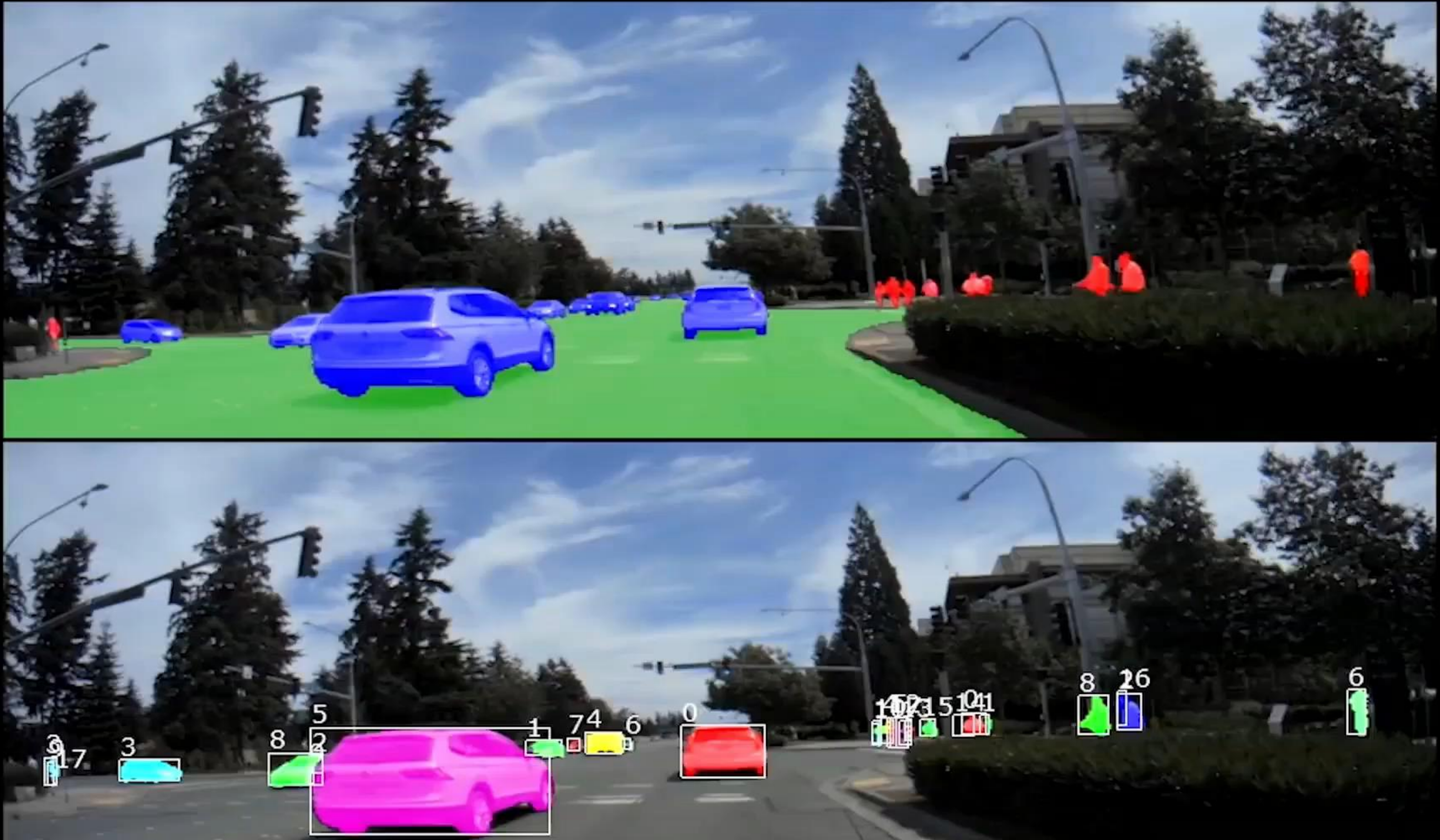
Real-Time Detection, Tracking and Classification of Multiple Moving Objects in UAV Videos
Hüseyin C. Baykara*, Erdem Bıyık*, Gamze Gül*, Deniz Onural*, Ahmet S. Öztürk*, İlkay Yıldız*
International Conference on Tools with Artificial Intelligence (ICTAI), November 2017
BibTeX PDF Video DOI
* denotes equal contribution.



Object detection in robotics

Speed is crucial – we need to detect, track, and classify many objects with high frequency.

Real-Time Detection, Tracking and Classification of Multiple Moving Objects in UAV Videos
Baykara et al., ICTAI 2017



How AI Helps Autonomous Vehicles See Outside the Box
NVIDIA

Today

- Basics of computer vision for robotics
- Representation learning

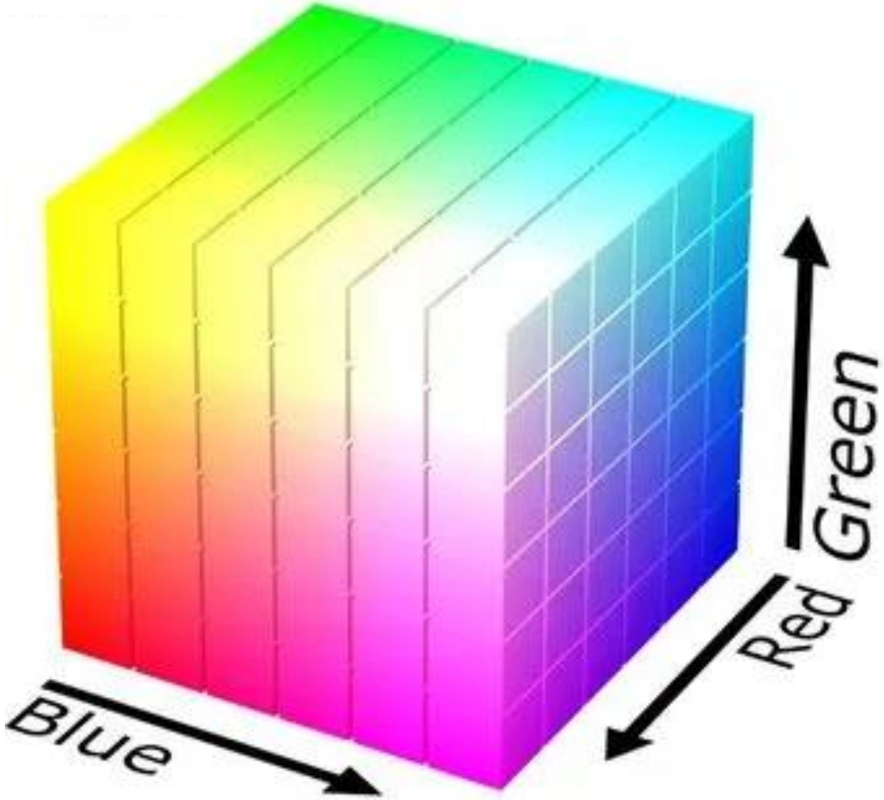
Representation learning

A good representation:

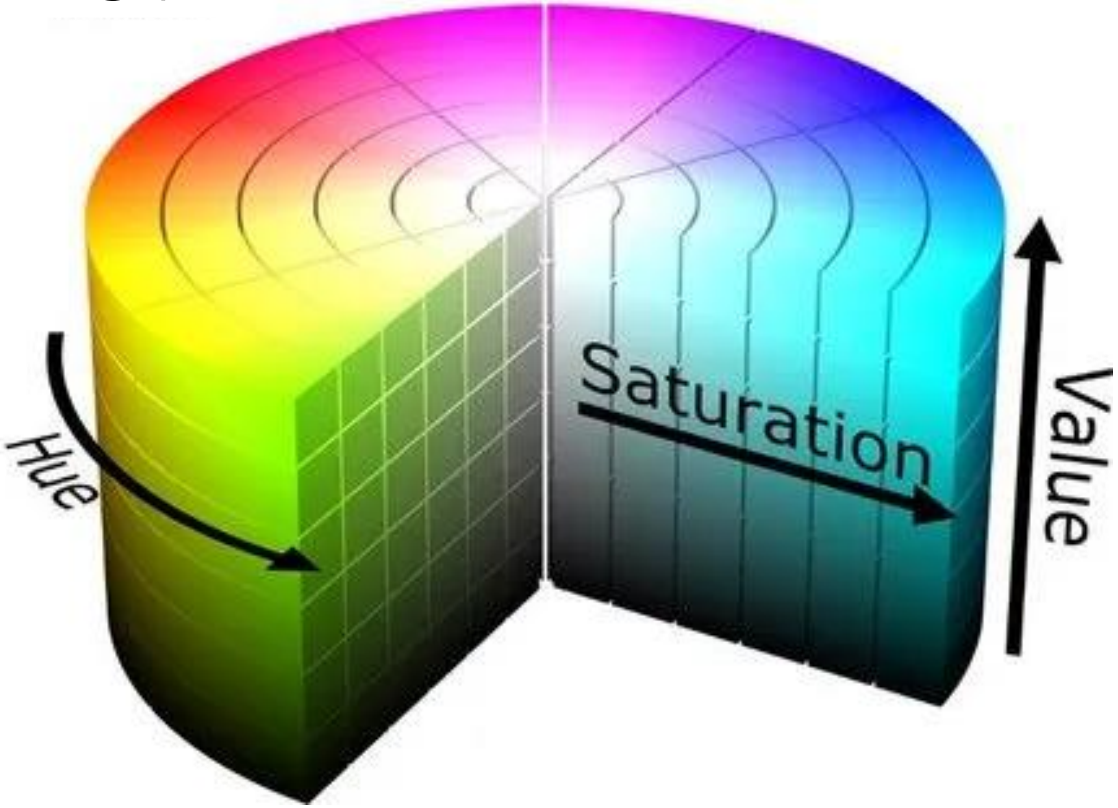
- contains the aspects of the raw data that are important for the downstream task
- is usually compact
- is ideally (but rarely) interpretable

Examples of representations

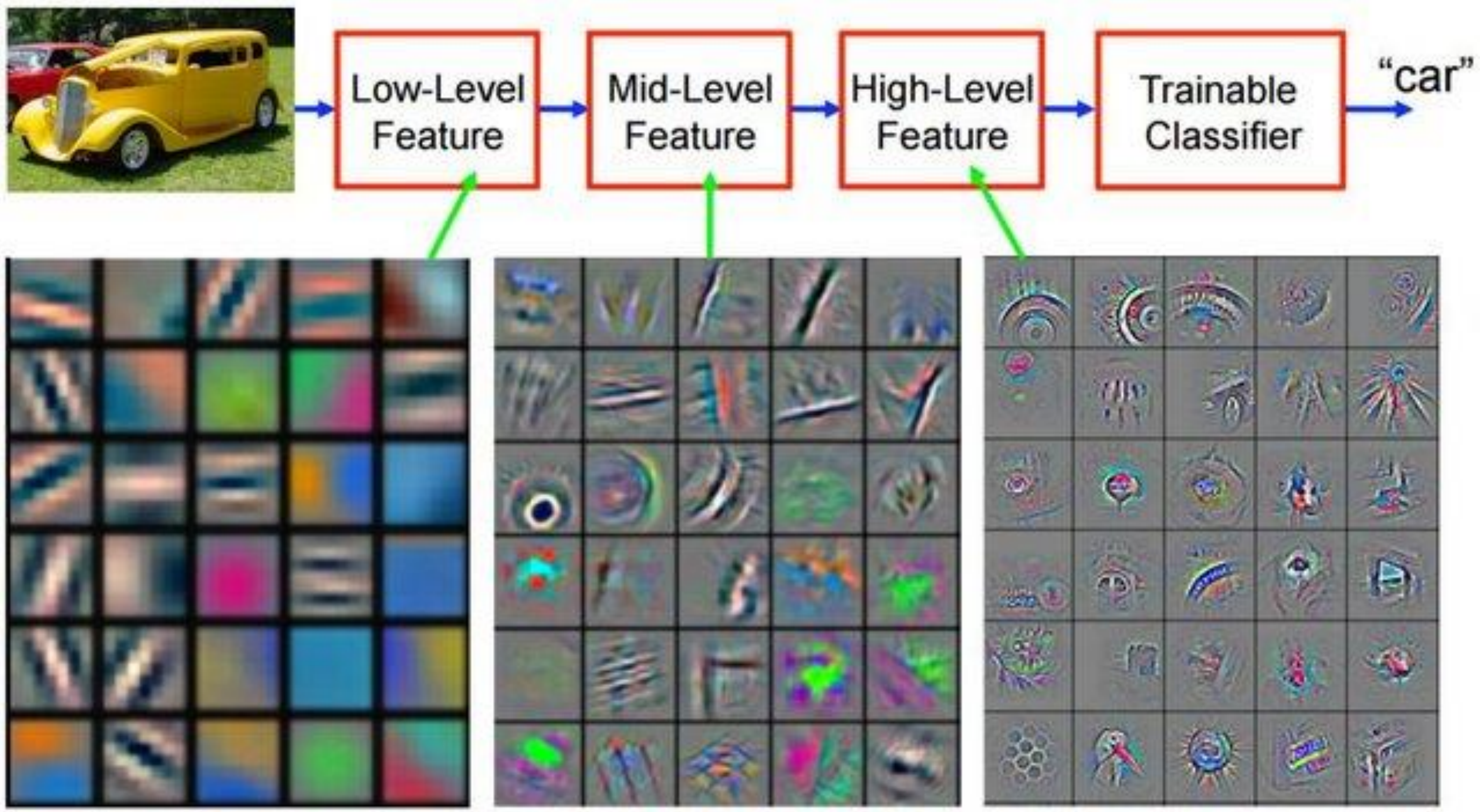
RGB



HSV



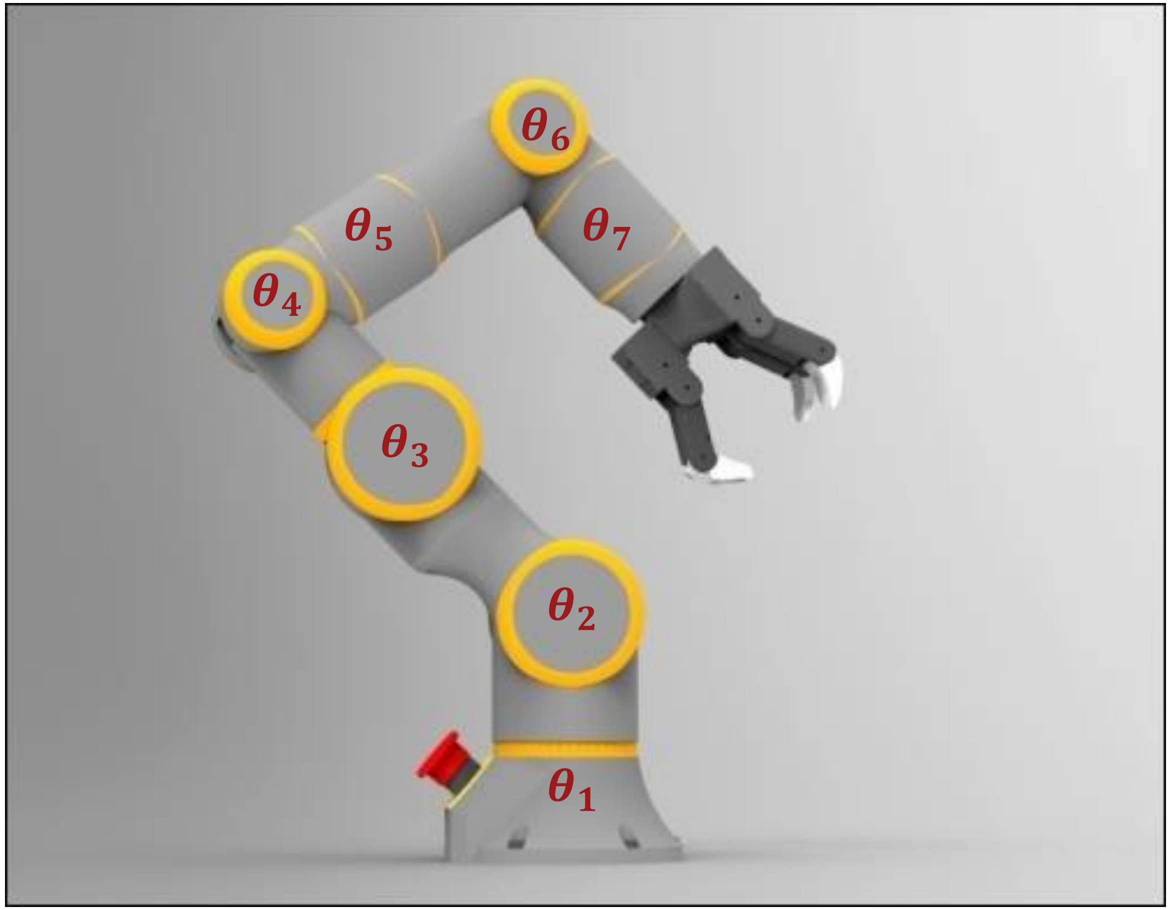
Examples of representations

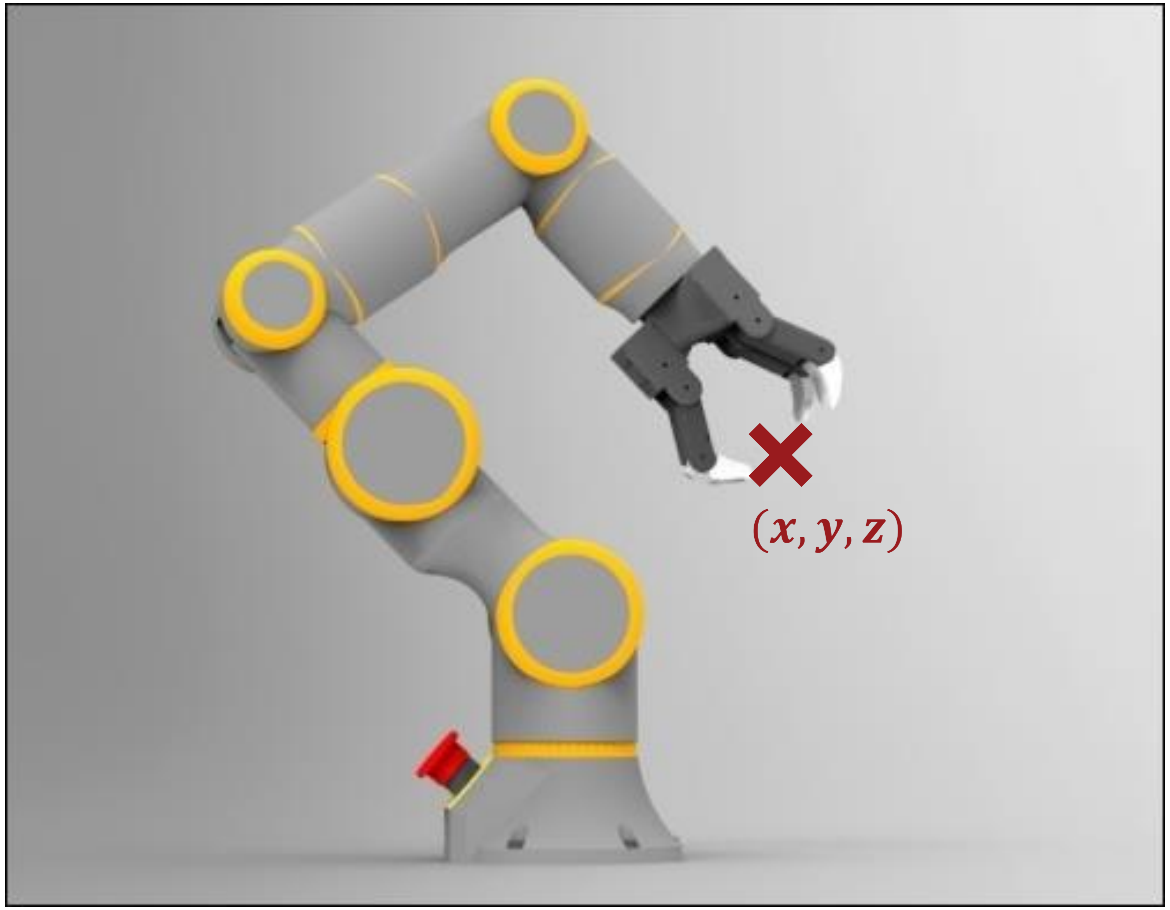


From: Stanford CS231N

Examples of representations

Examples of representations





Robot: COOL arm-5000 (ASIMOV Robotics)
Images from: Packt Publishing

How do we learn representations?

So many methods that it could be a course on its own.

Dynamics of Representation Learning							
30117D	048	4.0	Lecture	12:00-1:50pm	Mon, Wed	23 of 28	Greg Ver Steeg
Advanced Topics in Representation Learning for NLP							
30111D	048	4.0	Lecture	2:00-5:20pm	Tuesday	16 of 18	Xiang Ren
Representation Learning: Theory and Practice							
30220D	048	4.0	Lecture	10:00-11:50am	Mon, Wed	51 of 52	Aram Galstyan, Greg Ver Steeg

How do we learn representations?

So many methods that it could be a course on its own.

We will cover only two:

- Autoencoders
- Variational autoencoders

Principal component analysis (PCA)

PCA is a special case of autoencoders.

Raw data:	$X \in \mathbb{R}^{n \times d}$
Top k eigenvectors:	$V \in \mathbb{R}^{d \times k}$
Representation:	$Z = XV \in \mathbb{R}^{n \times k}$
Reconstruction:	$\hat{X} = XVV^\top \in \mathbb{R}^{n \times d}$

Principal component analysis (PCA)

PCA is a special case of autoencoders.

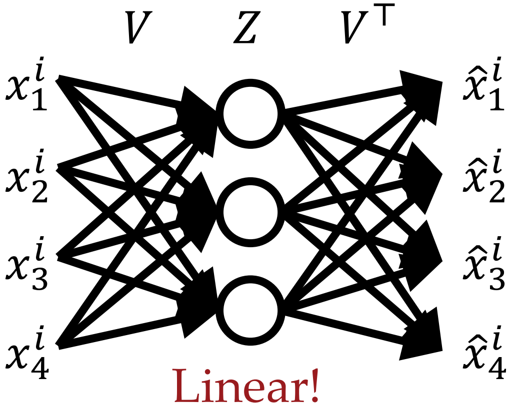
This is not typical in a neural network

Raw data: $X \in \mathbb{R}^{n \times d}$

Top k eigenvectors: $V \in \mathbb{R}^{d \times k}$

Representation: $Z = XV \in \mathbb{R}^{n \times k}$

Reconstruction: $\hat{X} = XVV^\top \in \mathbb{R}^{n \times d}$

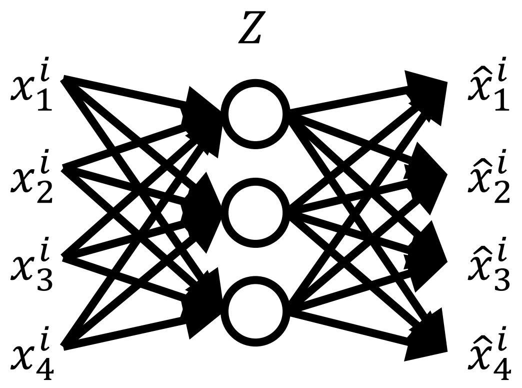


Principal component analysis (PCA)

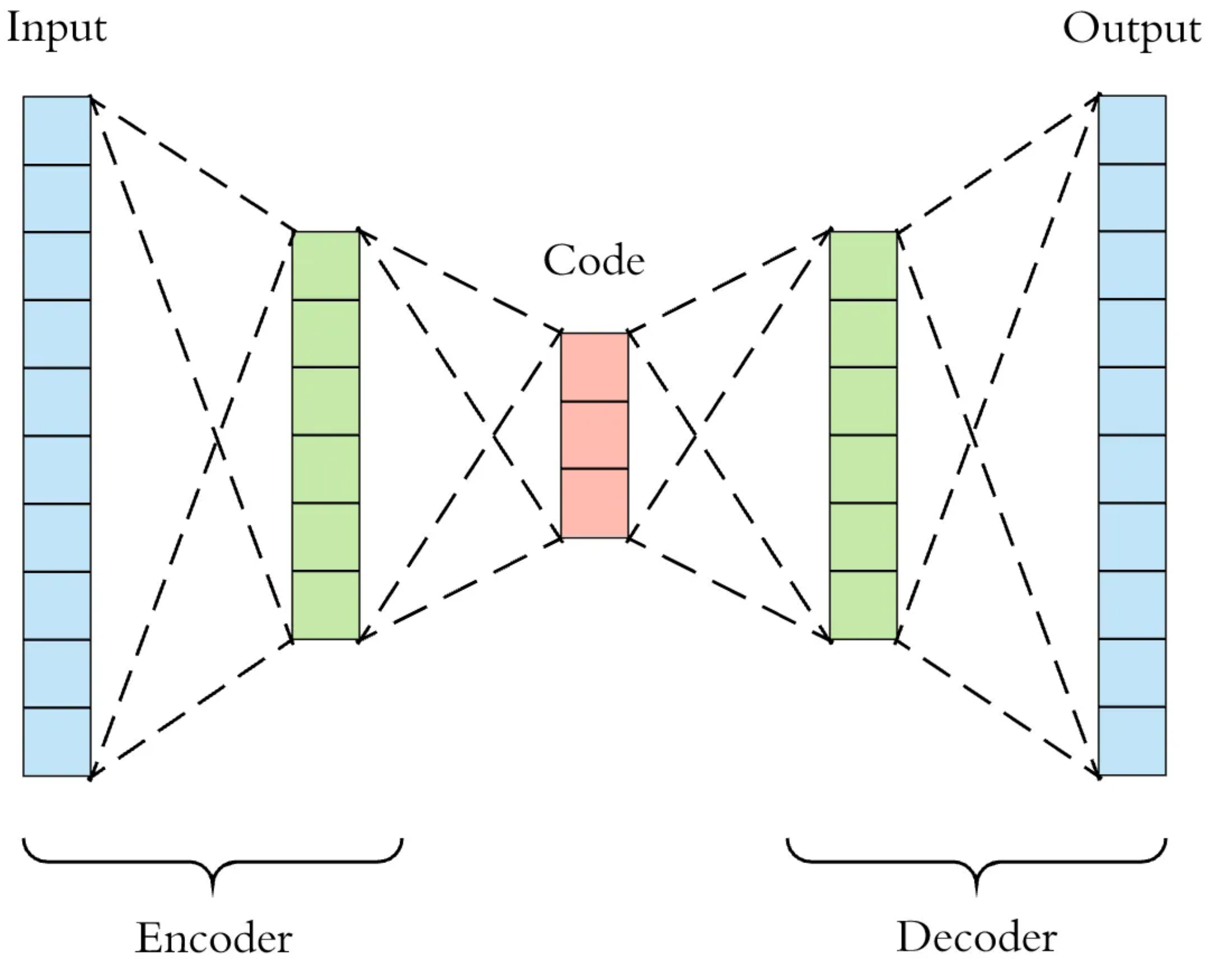
PCA is a special case of autoencoders.

1. Add nonlinearity
2. Remove the dependence between layers
3. Possibly add more layers

Now you have a more powerful autoencoder.

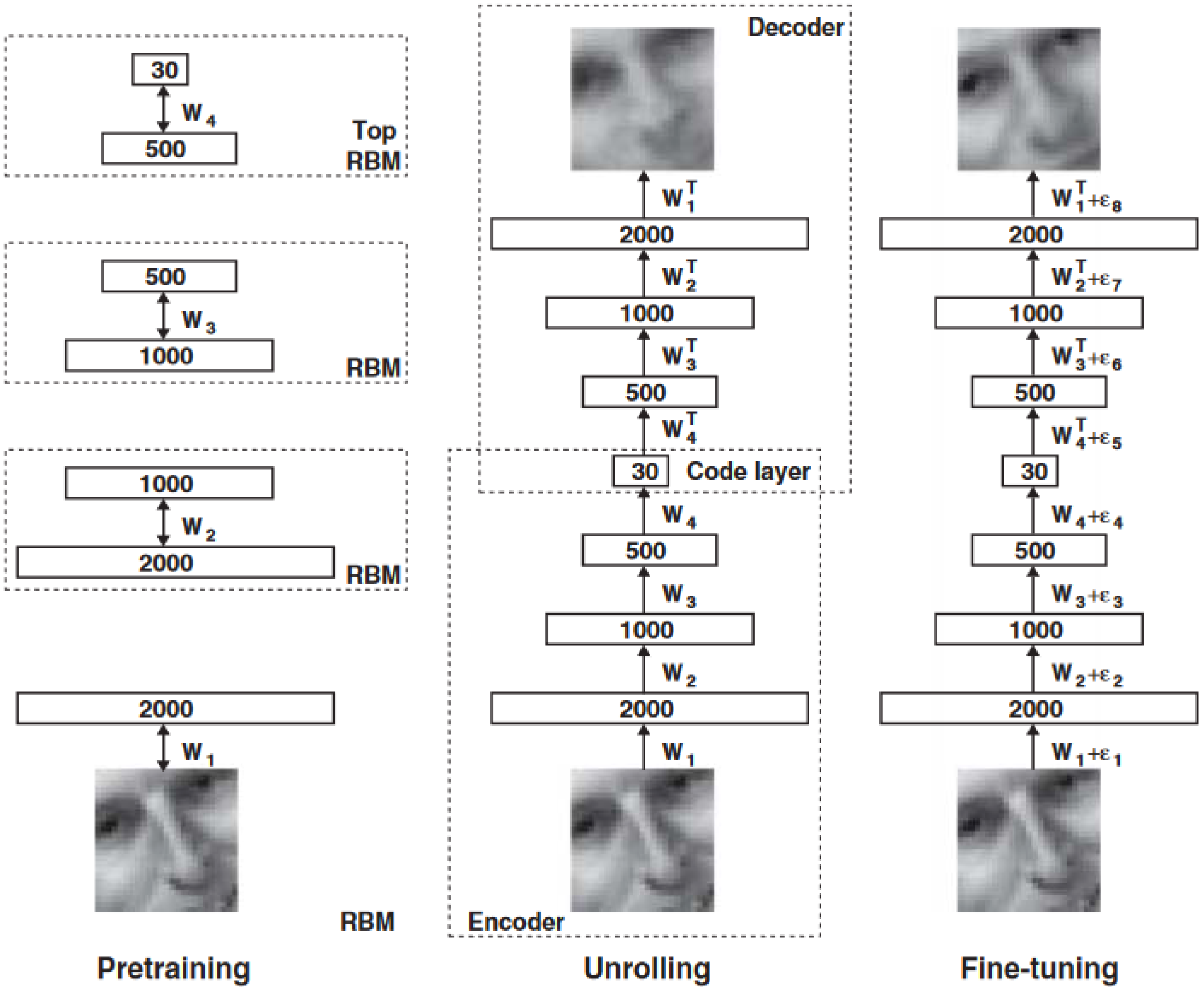


Autoencoders



A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA
Alkhayrat et al., Journal of Big Data 2020

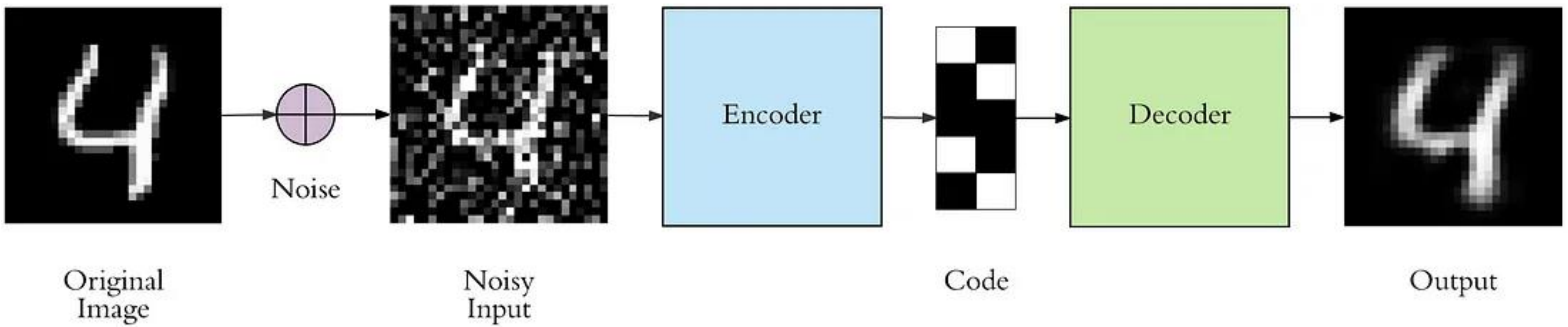
Autoencoders



Reducing the dimensionality of data with neural networks
Hinton and Salakhutdinov, Science 2006

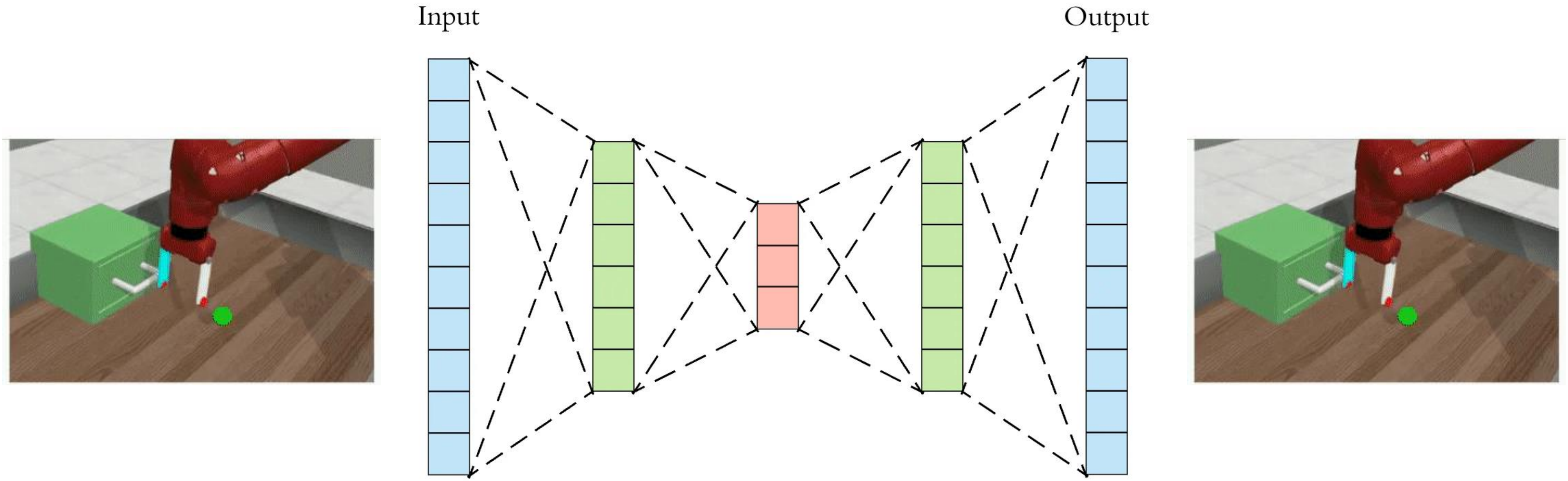
Applications of autoencoders

- Denoising



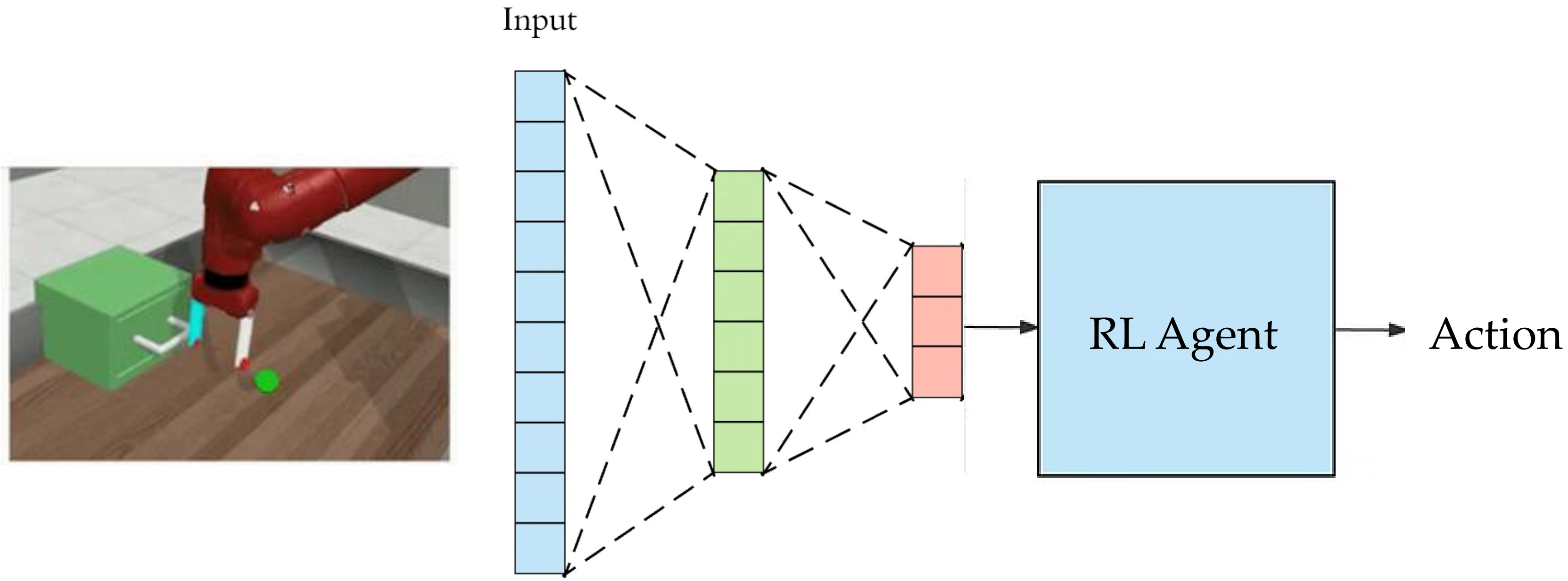
Applications of autoencoders

- Reinforcement learning



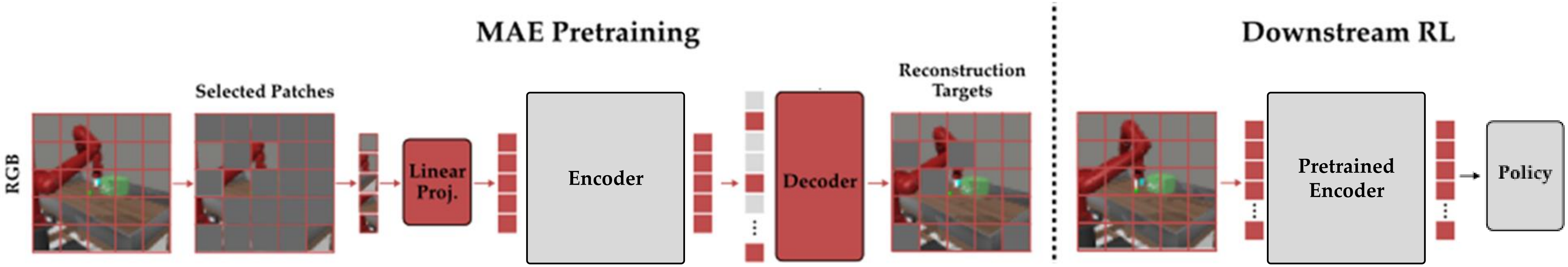
Applications of autoencoders

- Reinforcement learning



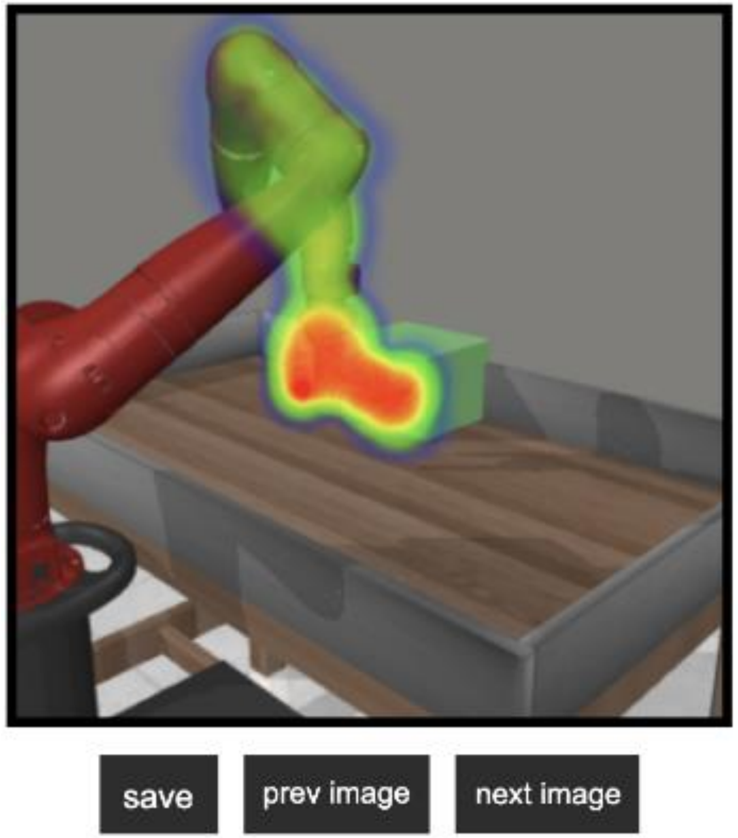
Applications of autoencoders

- Reinforcement learning



ViSaRL: Visual Reinforcement Learning Guided by Human Saliency
Liang et al., 2023

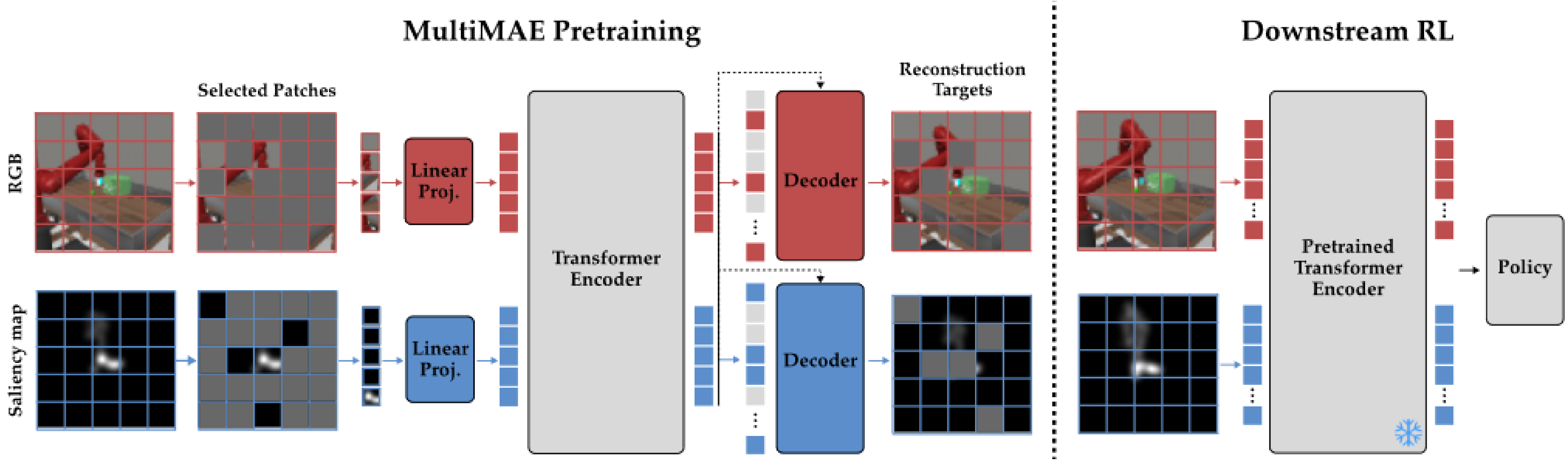
Applications of autoencoders



ViSaRL: Visual Reinforcement Learning Guided by Human Saliency
Liang et al., 2023

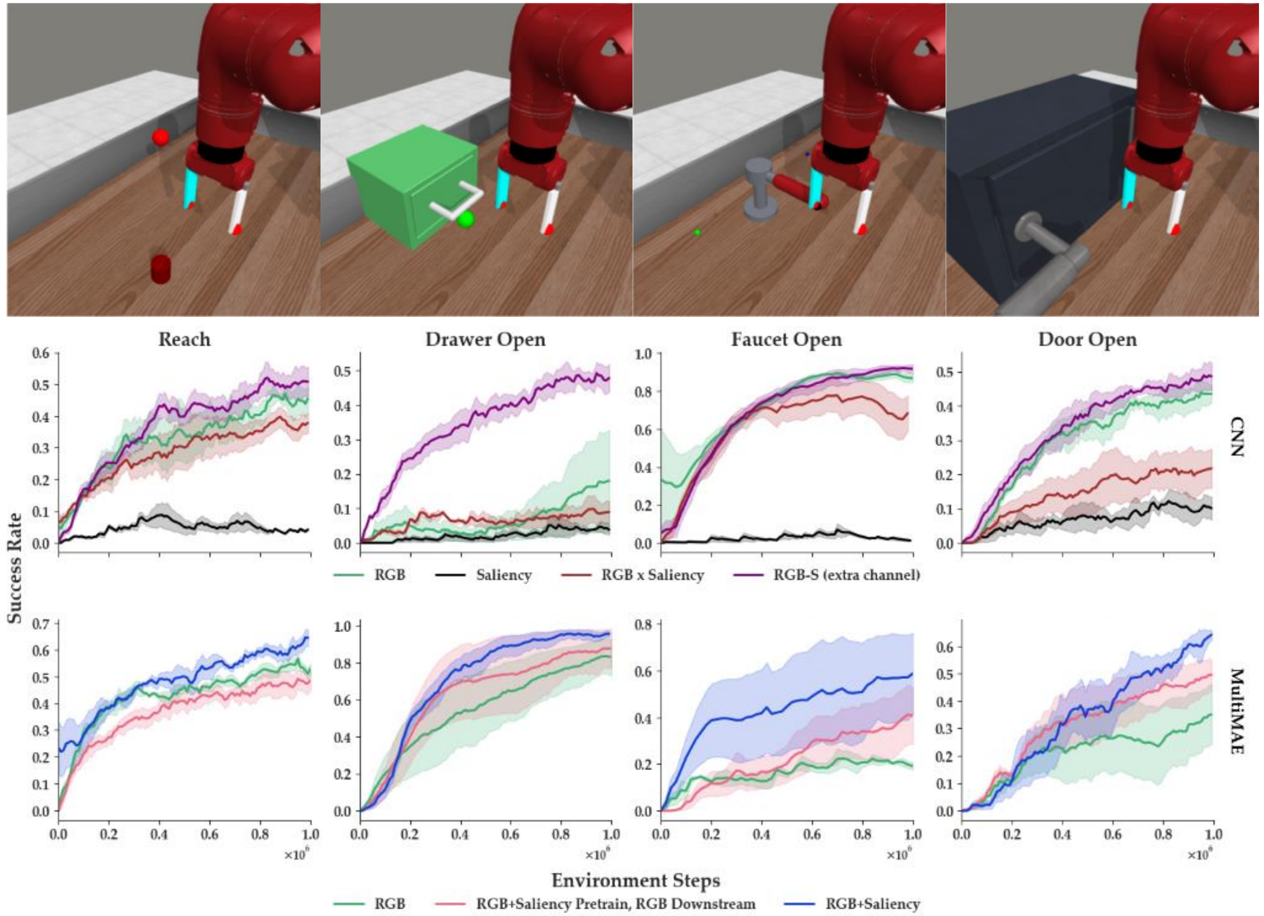
Applications of autoencoders

- Reinforcement learning (ViSaRL)



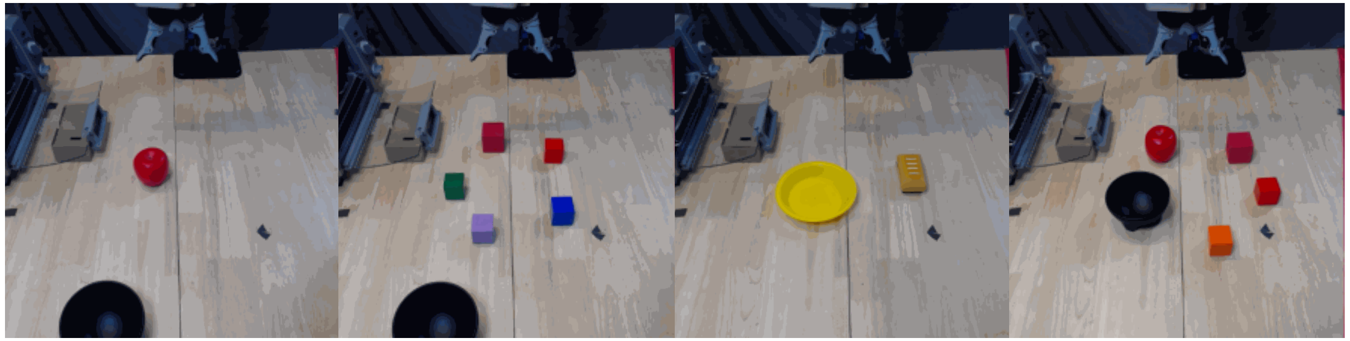
ViSaRL: Visual Reinforcement Learning Guided by Human Saliency
Liang et al., 2023

ViSaRL results



ViSaRL: Visual Reinforcement Learning Guided by Human Saliency
Liang et al., 2023

ViSaRL results

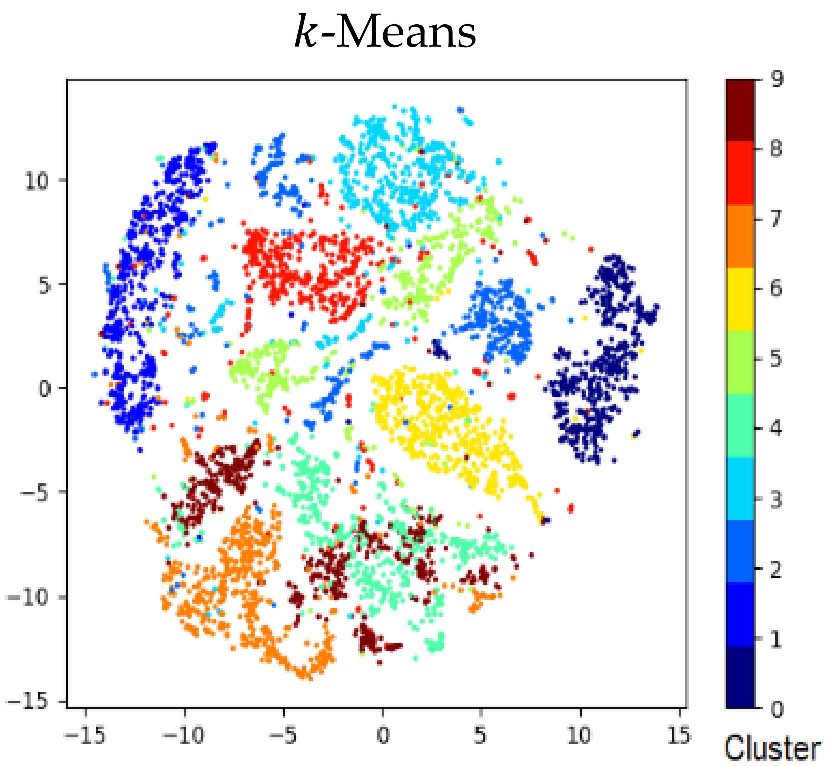


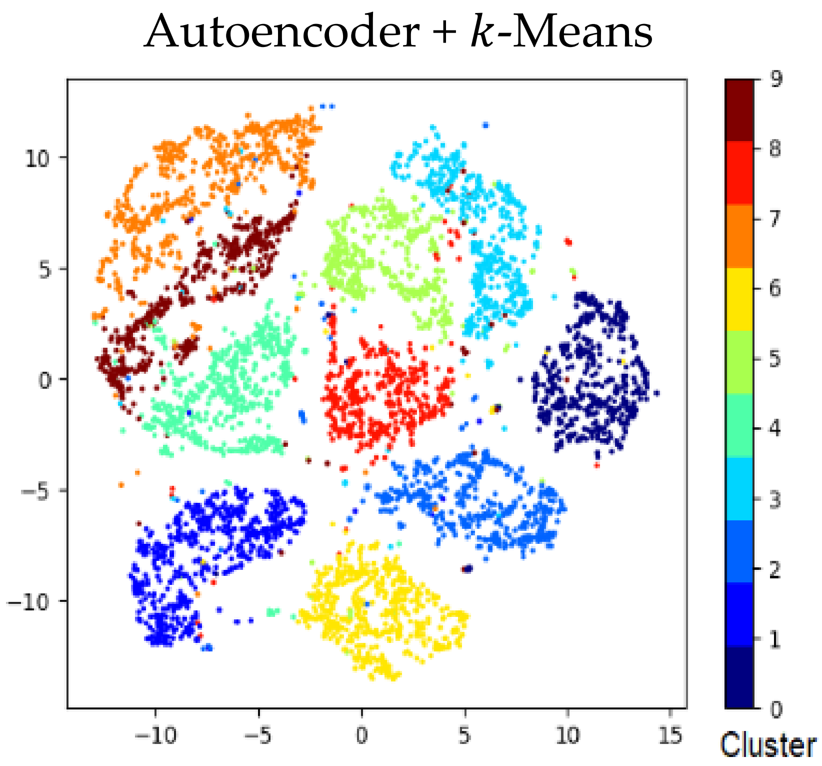
Model	Apple	Red Block	Bread → Plate	Apple → Bowl
MultiMAE (RGB-only)	6/10	4/10	3/10	1/10
MultiMAE (RGB + Saliency)	8/10	7/10	6/10	6/10

ViSaRL: Visual Reinforcement Learning Guided by Human Saliency
Liang et al., 2023

Applications of autoencoders

- Clustering

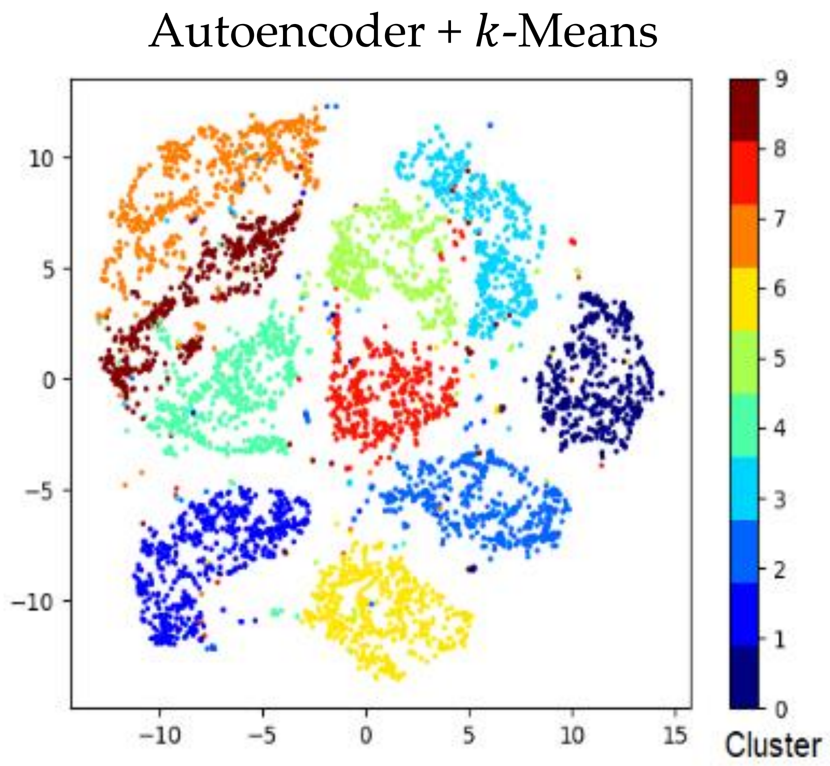




Clustering with Deep Learning: Taxonomy and New Methods
Aljalbout et al., 2018

Applications of autoencoders

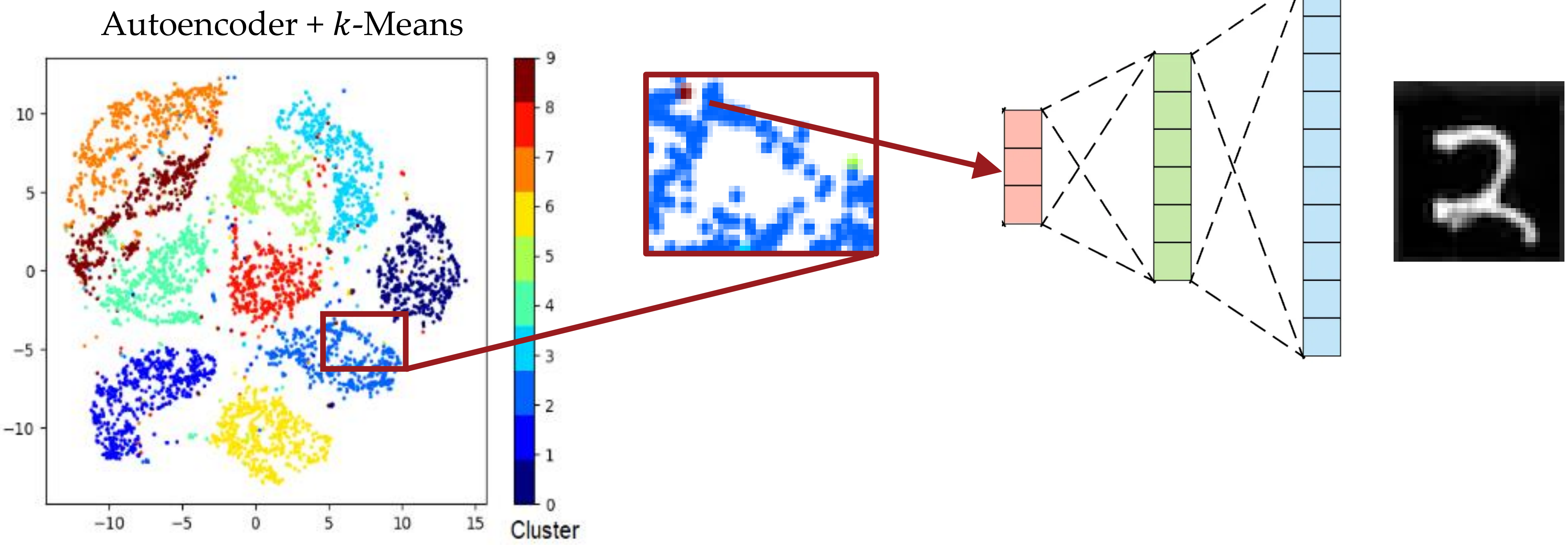
- Generate new data??



Clustering with Deep Learning: Taxonomy and New Methods
Aljalbout et al., 2018

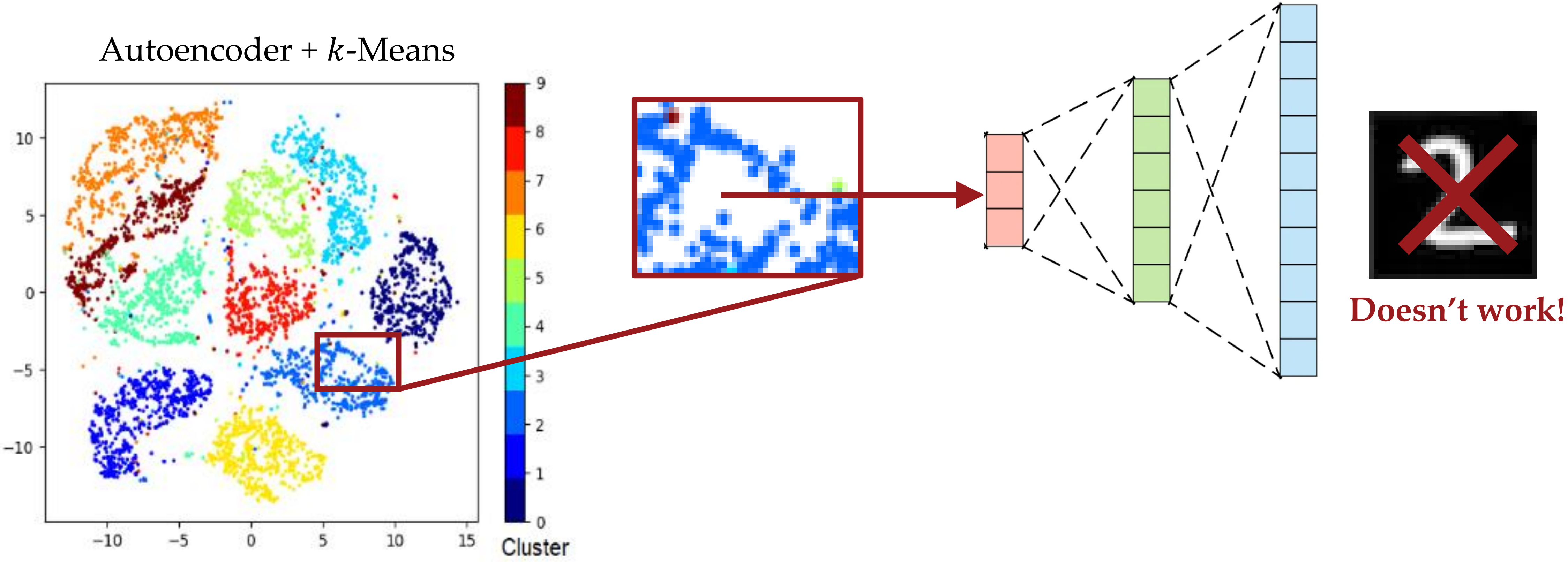
Applications of autoencoders

- Generate new data??



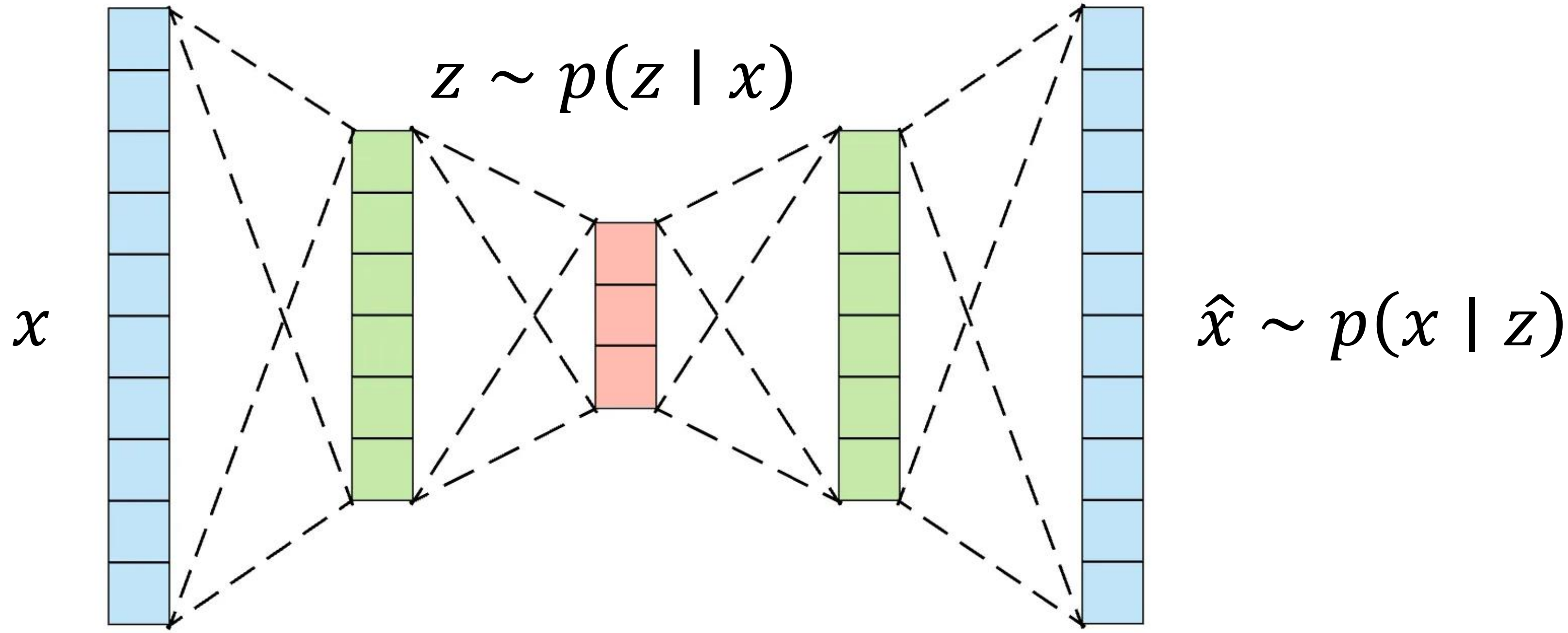
Applications of autoencoders

- Generate new data??



Variational autoencoders (VAE)

Same as autoencoders, but everything is a distribution now.



Variational autoencoders (VAE)

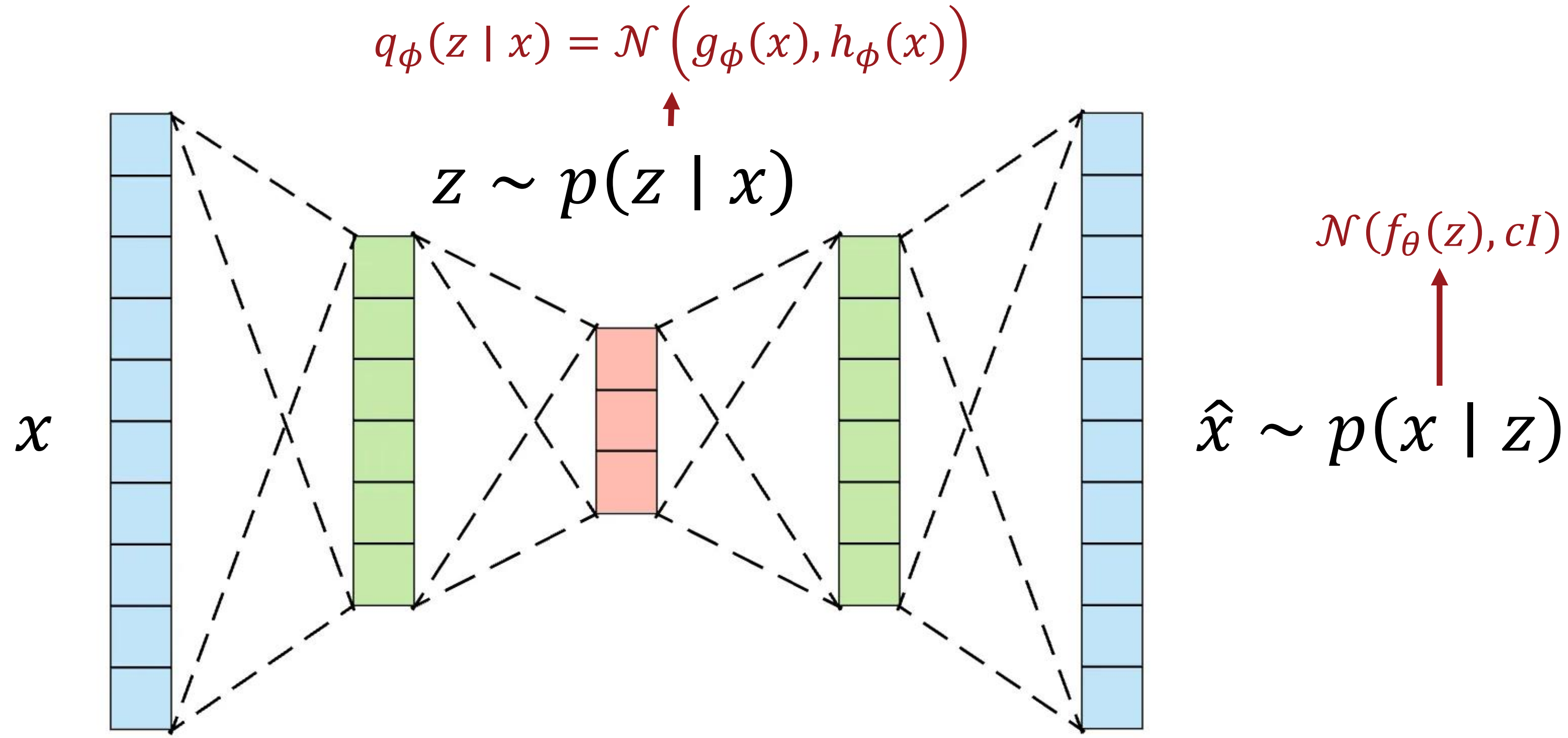
How does a neural network output a distribution?

We assume a parameterized distribution, and the network outputs the parameters.

Most commonly a Gaussian distribution. Then, the network outputs the mean vector and the covariance matrix.

Variational autoencoders (VAE)

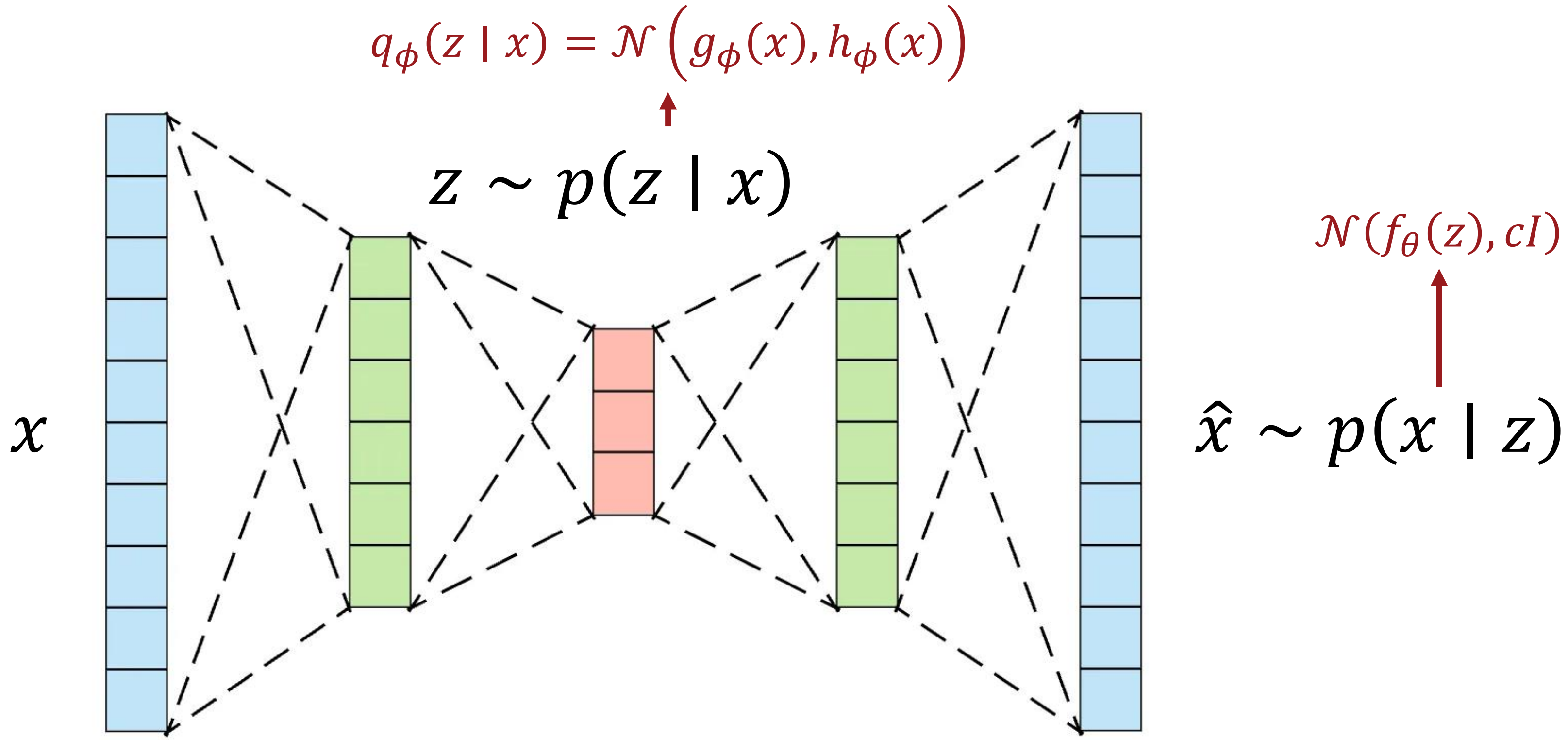
Same as autoencoders, but everything is a distribution now.



Variational autoencoders (VAE)

Just minimize $(x - \hat{x})^2$?

No.



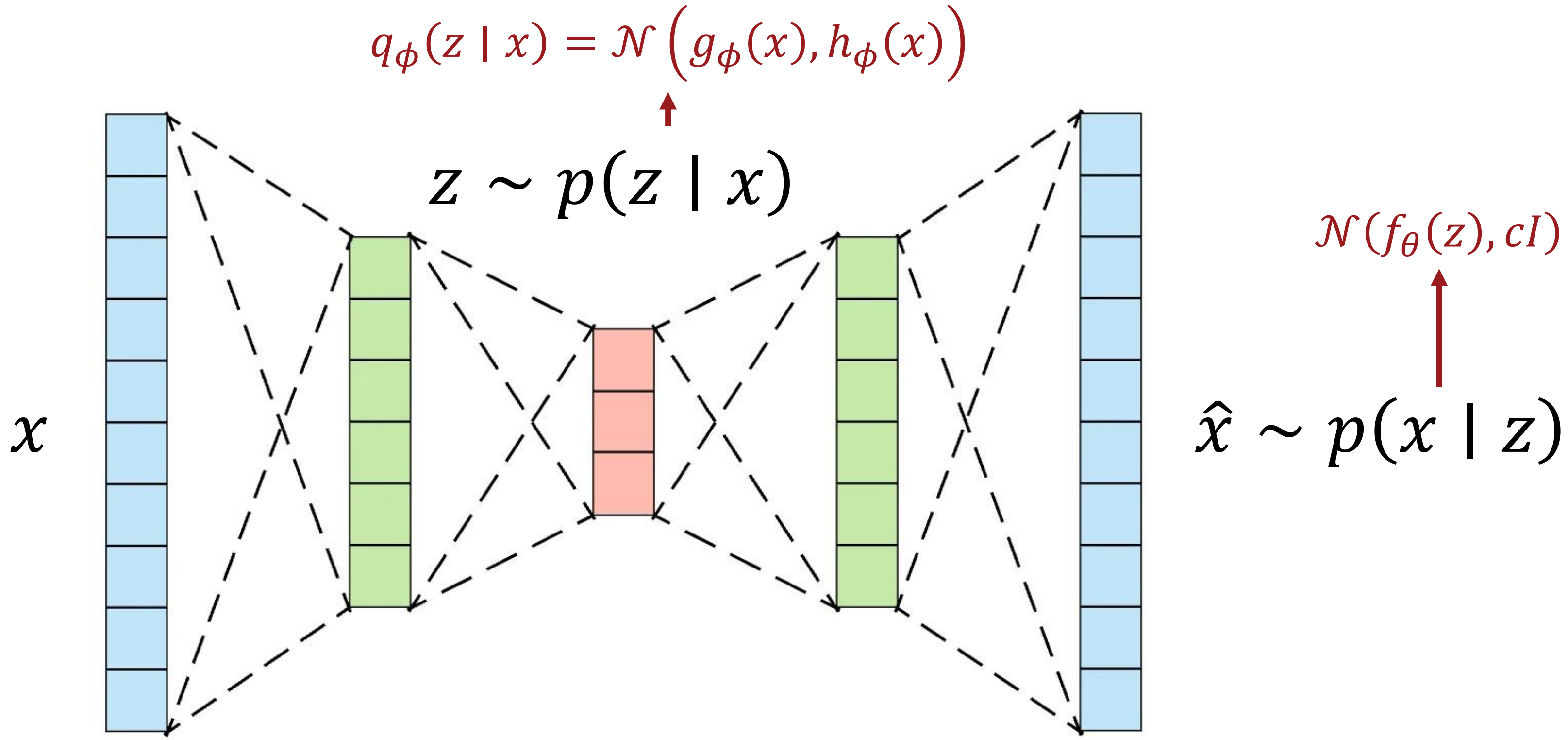
Variational autoencoders (VAE)

Just minimize $(x - \hat{x})^2$?

No.

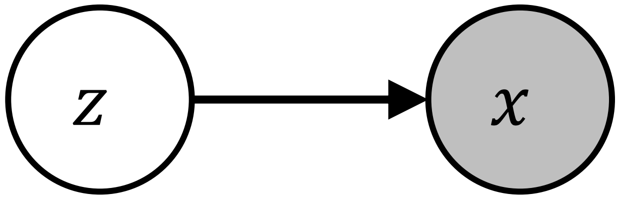
The network would learn Gaussians for z that are far from each other.

We need to regularize.



Variational autoencoders (VAE)

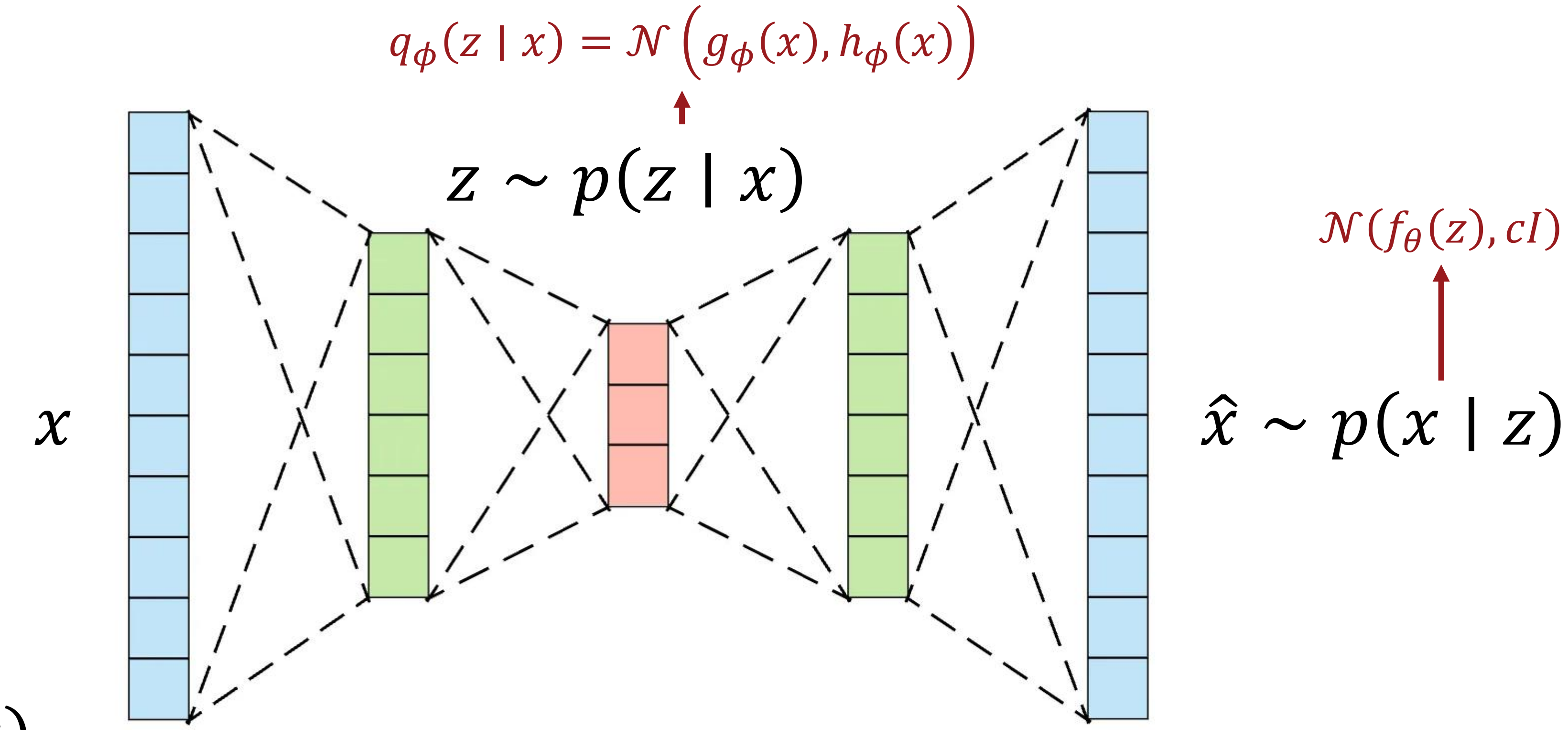
<u>Graphical model</u>



x is observed.

We want to maximize the likelihood of the observed data: $p(x)$.

But we can't compute $p(x)$.



Variational autoencoders (VAE)

$$p(x)$$

Distribution of raw data.

MNIST example:

All samples from $p(x)$ is a drawing of a 0-9 digit. For all such drawings $p(x) > 0$.

We don't have this.

Variational autoencoders (VAE)

$p(x)$

Distribution of raw data.

MNIST example:

All samples from $p(x)$ is a drawing of a 0-9 digit. For all such drawings $p(x) > 0$.

We don't have this.

$p(z)$

Distribution of the latent code.

We can assume anything.

This is what makes VAEs work!

We will assume $\mathcal{N}(0, I)$ so that the network will not cheat by learning Gaussians that are far.

Variational autoencoders (VAE)

Our goal is to maximize $p(x)$, or equivalently $\log p(x)$.

We will use variational inference.

Variational inference

$$D_{KL}\left(q_\phi(z \mid x) \parallel p(z \mid x)\right) = -\int q_\phi(z \mid x) \log\left(\frac{p(z \mid x)}{q_\phi(z \mid x)}\right) dz \geq 0$$

$$-\int q_\phi(z \mid x) \log\left(\frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x) p(x)}\right) dz \geq 0$$

$$-\int q_\phi(z \mid x) \left[\log\left(\frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)}\right) - \log p(x)\right] dz \geq 0$$

$$\log p(x) \int q_\phi(z \mid x) dz - \int q_\phi(z \mid x) \log\left(\frac{p_\theta(x \mid z) p(z)}{q_\phi(z \mid x)}\right) dz \geq 0$$

Variational inference

$$\log p(x) \geq \int q_\phi(z \mid x) \log\left(\frac{p_\theta(x \mid z)p(z)}{q_\phi(z \mid x)}\right) dz$$

$$\log p(x) \geq \int q_\phi(z \mid x) \left[\log p_\theta(x \mid z) + \log\left(\frac{p(z)}{q_\phi(z \mid x)}\right)\right] dz$$

$$\log p(x) \geq \mathbb{E}_{z \sim q_\phi(\cdot|x)}[\log p_\theta(x \mid z)] + \int q_\phi(z \mid x) \log\left(\frac{p(z)}{q_\phi(z \mid x)}\right) dz$$

$$\log p(x) \geq \mathbb{E}_{z \sim q_\phi(\cdot|x)}[\log p_\theta(x \mid z)] - D_{KL}\big(q_\phi(z \mid x) \parallel p(z)\big)$$

Variational inference

$$\log p(x) \geq \underbrace{\mathbb{E}_{z \sim q_\phi(\cdot|x)}[\log p_\theta(x \mid z)] - D_{KL}\big(q_\phi(z \mid x) \parallel p(z)\big)}_{\text{Evidence Lower Bound (ELBO)}}$$

Variational inference

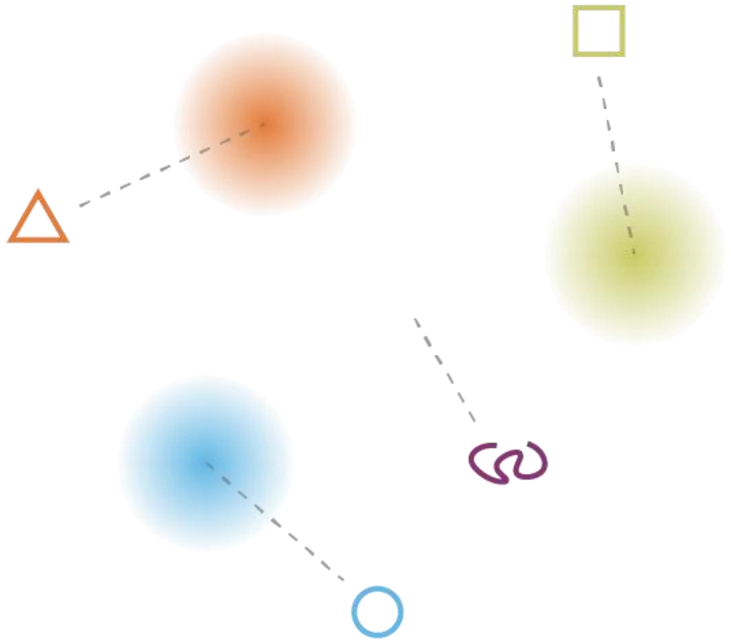
$$\log p(x) \geq \underbrace{\mathbb{E}_{z\sim q_\phi(\cdot|x)}[\log p_\theta(x \mid z)] - D_{KL}\big(q_\phi(z \mid x) \,\|\, p(z)\big)}_{\text{Evidence Lower Bound (ELBO)}}$$

Practical version:

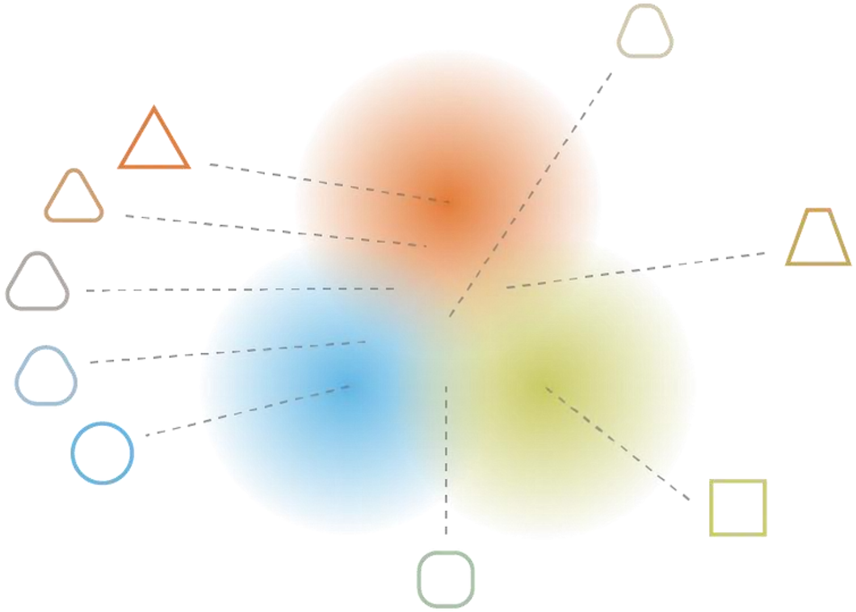
$$\log p(x) \geq \mathbb{E}_{z\sim q_\phi(\cdot|x)}\big[\log p_\theta(x \mid z) + \log p(z) - \log q_\phi(z \mid x)\big]$$

Comparison

Autoencoder, or
VAE without regularization

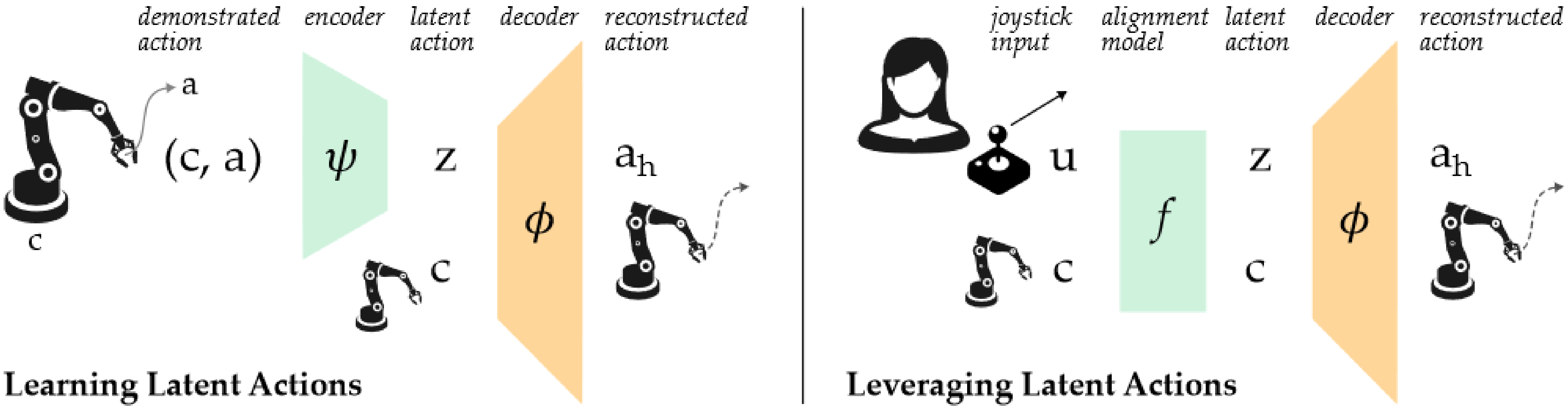


VAE with regularization



From: Joseph Rocca (Meta)

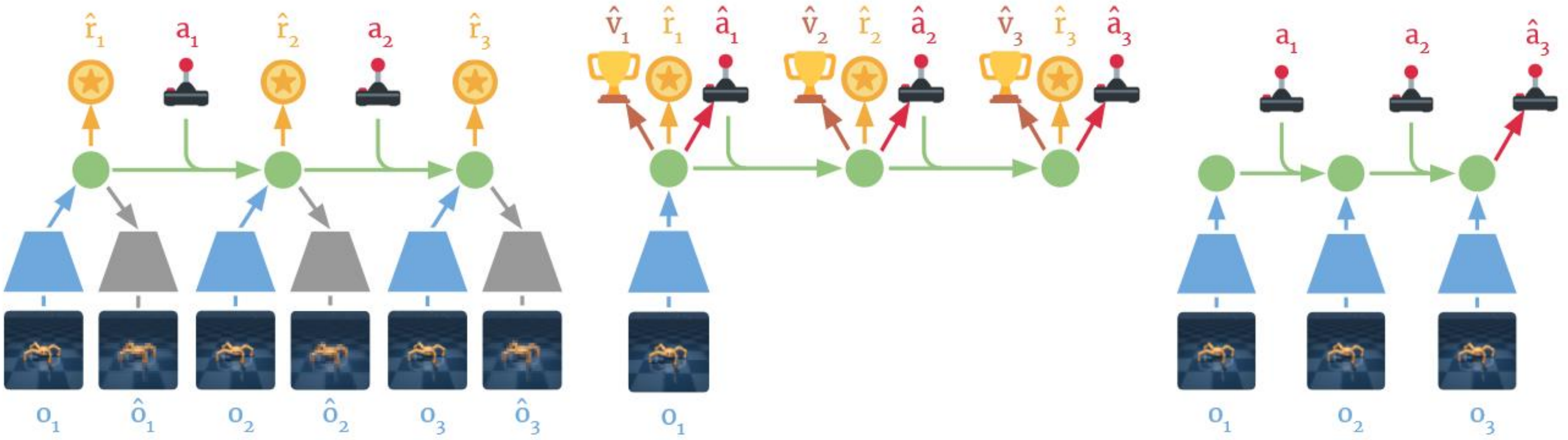
Variational autoencoders in robotics



More on this in a few weeks!

Learning Latent Actions to Control Assistive Robots
Losey et al., AURO 2022

Variational autoencoders in robotics

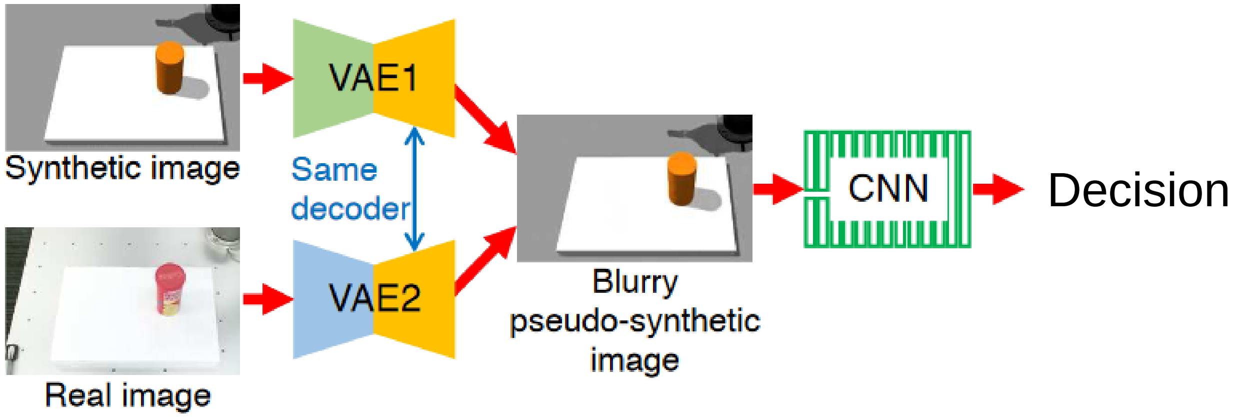


(a) Learn dynamics from experience (b) Learn behavior in imagination (c) Act in the environment

More on this in a few weeks!

Dream to control: learning behaviors by latent imagination
Hafner et al., ICLR 2020

Variational autoencoders in robotics



More on this in a few weeks!

Transfer learning from synthetic to real images using variational autoencoders for precise position detection
Inoue et al., ICIP 2018

Today

- Basics of computer vision for robotics
- Representation learning

Next time…

- Presentations on representation learning

- Reinforcement learning