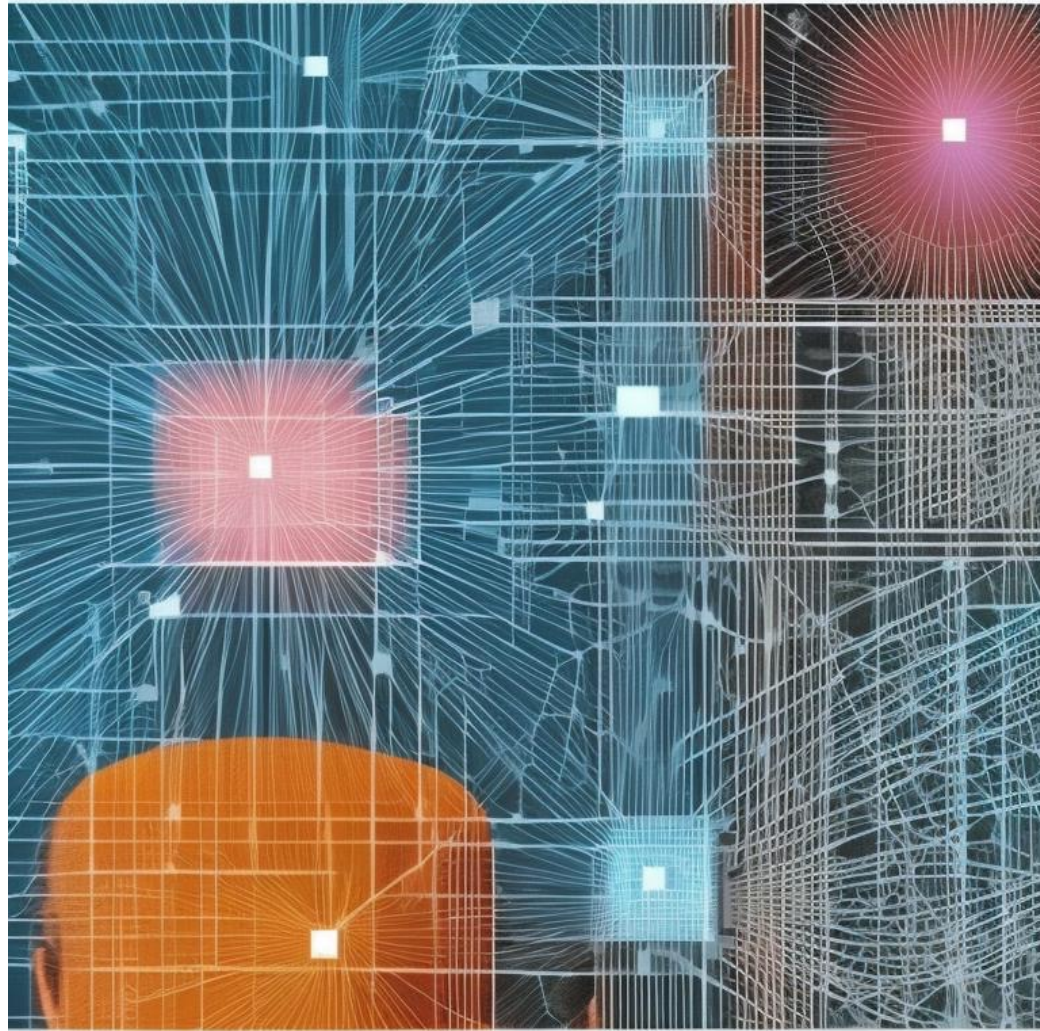


# CS 444: Deep Learning for Computer Vision

---



As generated using Stable Diffusion 2:  
<https://huggingface.co/stabilityai/stable-diffusion-2-1?text=Deep+Learning+for+Computer+Vision>

<https://saurabhg.web.illinois.edu/teaching/cs444/fa2023/>

# Lecture overview

---

- Computer Vision
- Learning for Computer Vision
- Deep Learning for Computer Vision
- Deep Learning Elsewhere
- Topics to be covered in class
- Logistics

# Computer Vision

---

To extract “meaning” from pixels



person, motorcycle, car, chair

*Meaning* can take different forms:

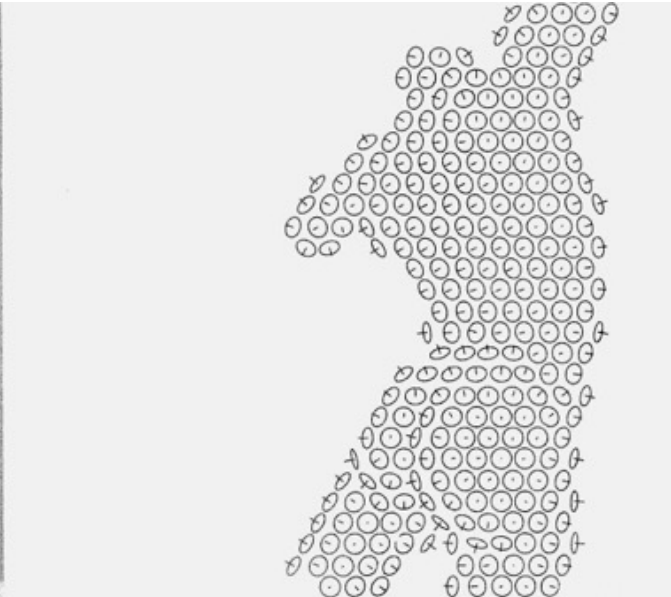
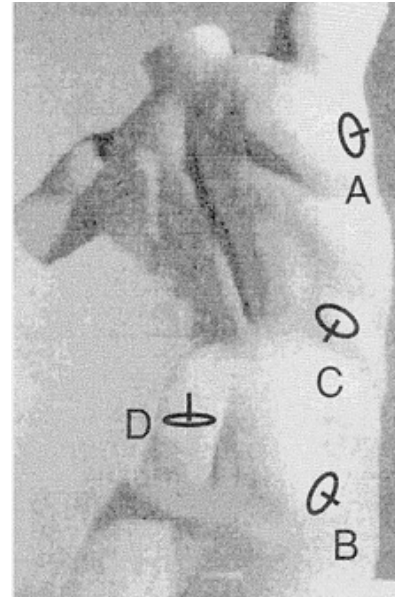
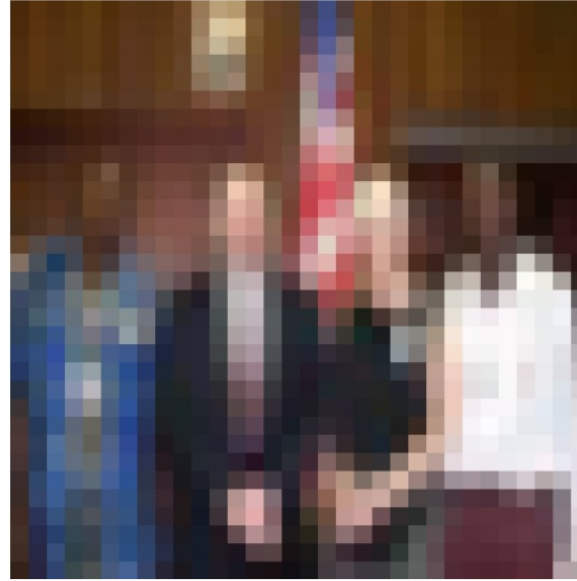
- Geometric Inferences
- Semantic Inferences
- Inferences about actions
- ...



# Computer vision is easy for humans

---

- Effortlessly analyze images for a variety of tasks
- Infer semantics even from severely ablated
- Can also make precise inference about certain geometric properties





# Yet has proven very hard for computers

- Computer vision research easily goes back 60 years ...

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
PROJECT MAC

Artificial Intelligence Group                      July 7, 1966  
Vision Memo. No. 100.

THE SUMMER VISION PROJECT  
Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".



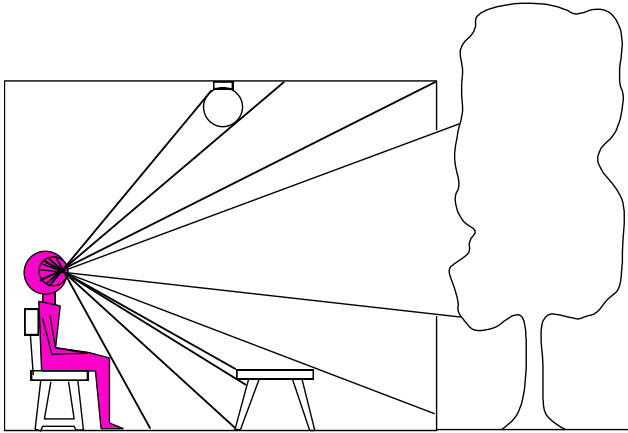
- Entirely true as of 2014 (or so) when this [xkcd](#) was published

# Why is computer vision hard?

---

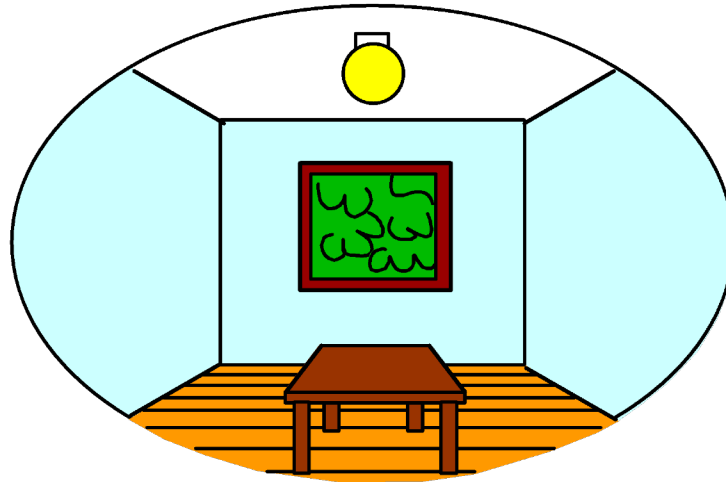
- Images are a lossy projection of the world

**3D world**



Point of observation

**2D image**



Geometry information is lost



# Why is computer vision hard?

---

- Images are a lossy projection of the world

What color is the dress?

- A) Black and blue
- B) White and gold?

Appearance  
information is  
also lost





# Why is computer vision hard?

---

- Images are a lossy projection of the world



Might cause  
objects to blend

# Why is computer vision hard?

---

- Images are a lossy projection of the world (geometry, appearance, ... are lost)
- Visual world is diverse



Viewpoint variation

Shape variation

# Why is computer vision hard?

---

- Images are a lossy projection of the world (geometry, appearance, ... are lost)
- Visual world is diverse



Background clutter



Occlusion



# Why is computer vision hard?

---

- Images are a lossy projection of the world (geometry, appearance, ... are lost)
  - need some priors to interpret what you are seeing
- Visual world is diverse
  - can't write down these priors by hand



*John's Diner with John's Chevelle, 2007*

Enter machine learning

# Why machine learning?

- Good old-fashioned AI (GOF AI) answer:  
Program expertise into the agent

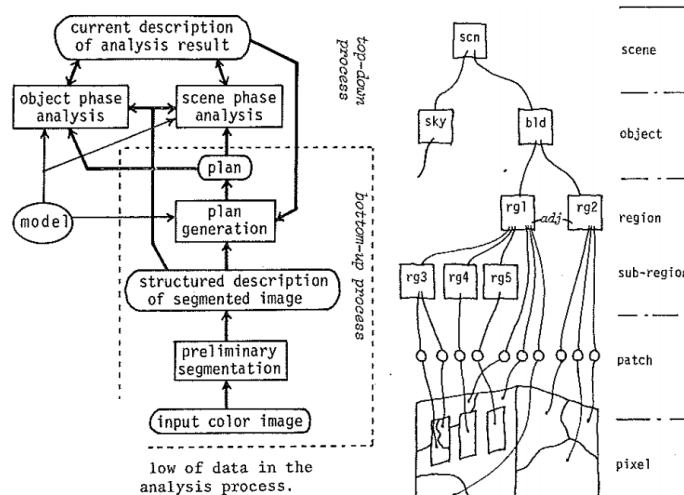


Figure 3. Structure of description built by the system.

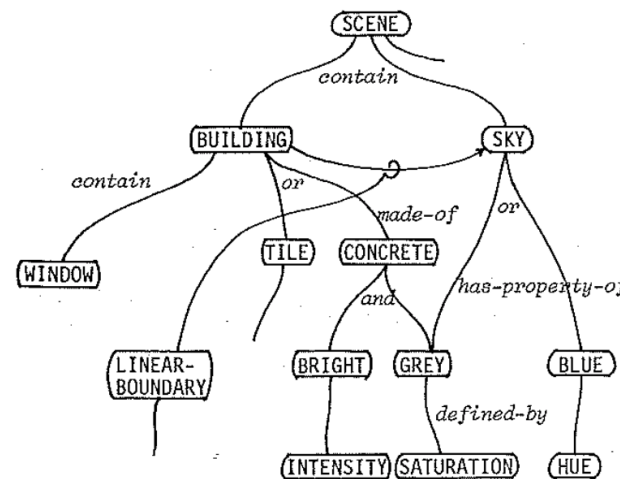


Figure 2. Semantic network for knowledge representation.

lock as a set of  
; it must satisfy  
The rules have

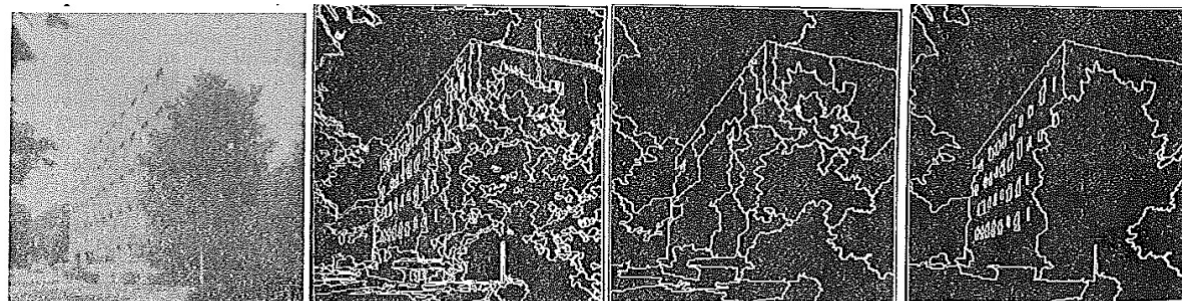


Figure 5-a. Digitized color scene.

5-b. Result of preliminary segmentation.

5-c. Plan image.

5-d. Result of semantic segmentation.

# Why machine learning?

- Good old-fashioned AI (GOF AI) answer:  
Program expertise into the agent

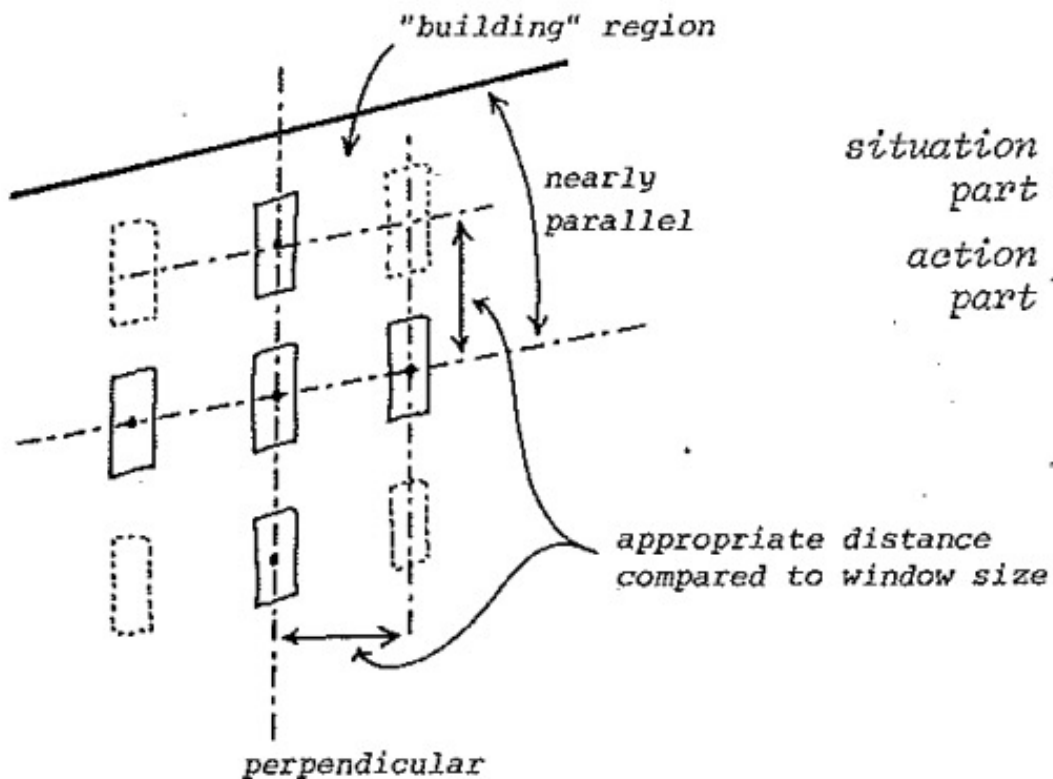


Figure 4-a. "Building" region and "windows".

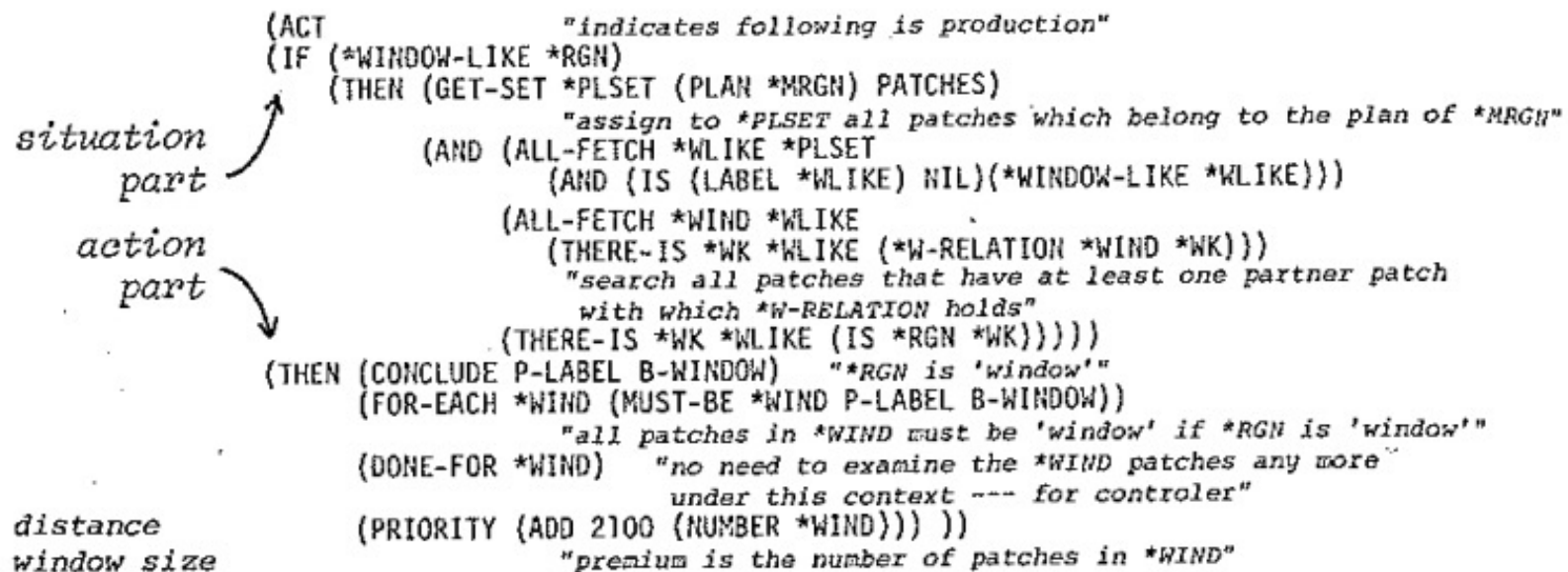


Figure 4-b. The production for analyzing "windows".



# Why machine learning?

- Good old-fashioned AI (GOF AI) answer: Program expertise into the agent

## Appendix-B Complete Listing of the Model

```

%SCENE knowledge-block-of-scene
(OBJECTS (%SKY %TREE %BUILDING %ROAD %WINDOW)
 SUB-OBJECTS (%B-WINDOW %CAR %C-SHADOW)
 KEY-PATCH-IS ((GREATERP (AREA %POCH 320) (%POCH)
 PLAN-IMAGE-GENERATION ((D1Y (BOUNDARY-LENGTH %POCH %POCH)
 (MULT (R-LS-DIFFERENCE %POCH %POCH)
 (BOUNDARY-CONTRACT %POCH %POCH))
 (%POCH %POCH))
 IF-PLAN-IS-MODIFIED (IF-DONE (
 rule-for-horizon-detection
 (FACT (IF (IS (OF HORIZON (SCENE)) NIL)
 (ALL-FETCH %RDN %PLAN-REGIONS
 (IF (AND (NOT (PROBABLY ROAD %RDN))
 (NOT (TOUCHING %RDN (LS-SIDE))
 (ALL-FETCH %RDN %PLAN-REGIONS
 (IF (AND (MAY-BE ROAD %RDN)
 (%RDN %RDN %RDN)
 (NOT (%RDN-COLOR %RDN %RDN))
 (%FACING HORIZONTALLY %RDN %RDN))
 (MULT (SUB (%FACING HORIZONTALLY %RDN %RDN) 0.5)
 (SUB (RDN (%ASK-VALUE ROAD %RDN) 0.6)
 (%ASK-VALUE ROAD %RDN) ))))
 (VALUE %RDN %RDN) ))
 (THEN (MEMO (SCENE) ROAD-ZONE
 (WITH (%R-LOU-SIDE %RDN 256) 1) 256))
 (MEMO (SCENE) HORIZON (%R-LOU-SIDE %RDN))
 (EXECUTE PLAN-EVALUATION) )) ))
 P-SELECT (TO-DO (
 rule-for-initial-start
 (FACT (AND (PROBABLY BUILDING %POCH (NOTFOUND BUILDING))
 rule-for-tree-occlusion
 (FACT (AND (%DARK %POCH (%UPPER %POCH)
 (OR (TOUCHING %POCH (UP-SIDE) (TOUCHING %POCH (SIDE))
 (THERE-IS %TR %REGIONS
 (AND (IS (LABEL %TR) TREE)
 (%BOVE %POCH %TR)
 (TOUCHING %TR (SIDE)
 (%WITH-IN2 %POCH (%V-ZONE %TR))))
 (THEN (CONCLUDE P-LABEL TREE)
 (CONCLUDE D-HERSE (WITH (OCCLUDE %TR FRAME))
 (SCORE-IS 1.0)) (%POCH) ))
 rule-for-tree-garbage
 (FACT (PROBABLY TREE %POCH)
 (THEN (CONCLUDE P-LABEL TREE)
 (SCORE-IS (%ASK-VALUE TREE %POCH)) (%POCH) ))
 P-LABEL (IF-DONE (
 if-done-rule-to-be-activated-when-keypatch-is-labeled
 (FACT (NOT (IS (OF PLAN %POCH NIL))
 (THEN (EXECUTE PLAN-EVALUATION)) (%POCH) )) ))
 %SKY knowledge-block-of-sky
 (PROPERTY-RULES (
 (GEN (NOT (%LARGER %RDN (1.0 , 0.5)) (%RDN)
 (GEN (%SHINING %RDN (1.0 , 0.2)) (%RDN))
 (GEN (OR (%BLUE %RDN (%GREY %RDN)) (1.0 , 0.2)) (%RDN)
 (GEN (NOT (%TEXTURAL %RDN) (1.0 , 0.7)) (%RDN)
 (STR (TOUCHING %RDN (UP-SIDE) (0.7 , 0.2)) (%RDN) ))
 RELATION-RULES (
 (STR (AND (%LINEAR-BOUNDARY %RDN %RDN)
 (IF (%LINEAR-BOUNDARY (POSITION (DAN %RDN %RDN))
 (0.0 , 0.5) FOR SKY) (%RDN %RDN))
 (STR (IF (NOT (IS (OF BUILDING-ZONE (SCENE)) NIL)
 (%FUZZY (D-RATIO %RDN (OF BUILDING-ZONE (SCENE))) 0.5 0.3)
 (0.0 , 0.5) FOR SCENE) (%RDN) ))

```

```

(THEN (CONCLUDE P-LABEL BUILDING)
 (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE %POCH))) (%POCH)
 (FACT (AND (PROBABLY ROAD %POCH (NOTFOUND ROAD))
 (THEN (CONCLUDE P-LABEL ROAD)
 (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE %POCH))) (%POCH)
 (FACT (AND (PROBABLY SKY %POCH (NOTFOUND SKY))
 (THEN (CONCLUDE P-LABEL SKY)
 (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE %POCH))) (%POCH)
 (FACT (AND (PROBABLY TREE %POCH)
 (NOT (THERE-IS %TR %REGIONS
 (AND (IS (LABEL %TR) TREE)
 (OR (TOUCHING (PLAN %POCH) (PLAN %TR)
 (%WITH-IN2 (PLAN %POCH)
 (%V-ZONE 30 (PLAN %TR))))))
 (THEN (CONCLUDE P-LABEL TREE)
 (SCORE-IS (ADD 4.0 (CONFIDENCE-VALUE %POCH))) (%POCH)
 rule-for-adjacent-wall-of-building
 (FACT (AND (MAY-BE BUILDING %POCH)
 (THERE-IS %BL %REGIONS
 (AND (IS (LABEL %BL) BUILDING)
 (NOT (IS (OF SHAPE VIEW (OBJECT %BL) ))
 (IS (OF ADJACENT (OBJECT %BL) NIL)
 (%DIFFERENT-ZONE %POCH %BL)))
 (THEN (CONCLUDE P-LABEL BUILDING)
 (CONCLUDE D-HERSE (WITH (ADJACENT %BL)
 (SCORE-IS (ADD 5.0 (%ASK-VALUE BUILDING %POCH)) (%POCH)
 rule-for-building-occlusion
 (FACT (AND (MAY-BE BUILDING %POCH)
 (THERE-IS %BL %REGIONS
 (AND (IS (LABEL %BL) BUILDING)
 (%SAME-ZONE %POCH %BL)
 (%SAME-COLOR %POCH %BL)
 (THERE-IS %TR (%KEYPATCHES
 (AND (%BETWEEN %TR %POCH %BL)
 (OR (IS (LABEL %TR) TREE)
 (AND (IS (LABEL %TR) BUILDING)
 (NOT (IS (OBJECT %BL)
 (OBJECT %TR))))))
 (THEN (CONCLUDE P-LABEL BUILDING)
 (CONCLUDE D-HERSE (WITH (OCCLUDE %BL (REGION %TR))
 (SCORE-IS (ADD 1.0 (%ASK-VALUE BUILDING %POCH)) (%POCH)
 P-SELECT (
 TO-DO (
 (FACT (MAY-BE SKY %POCH)
 (THEN (SCORE-IS (ADD 2.0 (%ASK-VALUE SKY %POCH)) (%POCH)
 (FACT (AND (IS-PLAN %POCH %RDN) (%RDN) (%RDN) (%RDN) (%RDN) (%RDN)
 (THEN (SCORE-IS 3.0)) (%POCH %RDN))
 (FACT (%BRIGHT %POCH) (THEN (SCORE-IS 0.85)) (%POCH) ))
 IF-DONE (
 (FACT %TR (THEN (CONCLUDE P-LABEL SKY)
 (CONCLUDE R-HERSE (%MASTER %POCH)) (%POCH) ))
 PRIORITY-VALUE-IS 0.1)
 %TREE knowledge-block-of-tree
 (MADE-OF (%LEAVES)
 PROPERTY-RULES (
 (GEN (%MIDDLE %RDN) (0.5 , 0.3)) (%RDN)
 (STR (%HEAVY-TEXTURE %RDN) (0.8 , 0.2)) (%RDN) ))
 P-SELECT (
 TO-DO (
 (FACT (MAY-BE TREE %POCH)
 (THEN (SCORE-IS (ADD 2.0 (%ASK-VALUE TREE %POCH)) (%POCH) (%POCH)
 (FACT (AND (IS-PLAN %POCH %RDN) (%RDN) (%RDN) (%RDN) (%RDN) (%RDN)
 (THEN (SCORE-IS 3.0)) (%POCH %RDN))
 IF-DONE (
 (FACT %TR (THEN (CONCLUDE P-LABEL TREE)
 (CONCLUDE R-HERSE (%MASTER %POCH)) (%POCH) ))
 PRIORITY-VALUE-IS 0.2)

```

```

%BUILDING knowledge-block-of-building
(MADE-OF (OR (%CONCRETE %TITLE %BRICK)
 SUB-OBJECTS (%B-WINDOW)
 PROPERTY-RULES (
 (GEN (%MIDDLE %RDN) (0.5 , 0.3)) (%RDN)
 (STR (%PANHOLE %RDN) (0.5 , 0.2)) (%RDN)
 (STR (%WINDIE %RDN) (0.4 , 0.2)) (%RDN)
 (GEN (%GLENE %RDN) (0.5 , 0.5)) (%RDN) ))
 RELATION-RULES (
 (GEN (AND (%LINEAR-BOUNDARY %RDN %RDN)
 (IF (%LINEAR-BOUNDARY (NOT (POSITION (UP %RDN %RDN))
 (0.0 , 0.4) FOR SKY) (%RDN %RDN))
 (STR (IF (NOT (IS (OF BUILDING-ZONE (SCENE)) NIL)
 (AND (D-RATIO %RDN (OF BUILDING-ZONE (SCENE))
 (%PROXIE %RDN))
 (0.5 , 0.3) FOR SCENE) (%RDN) ))
 P-SELECT (
 TO-DO (
 (FACT (AND (MAY-BE BUILDING %POCH (%SAME-ZONE %POCH %RDN)
 (THEN (CONCLUDE P-LABEL BUILDING)
 (CONCLUDE R-HERSE %RDN)
 (SCORE-IS (ADD 2.0 (%ASK-VALUE BUILDING %POCH)) (%POCH %RDN)
 (FACT (AND (NOT (IS-PLAN %POCH %RDN) (%SAME-ZONE %POCH %RDN)
 (MAY-BE BUILDING (PLAN %POCH))
 (THEN (CONCLUDE P-LABEL BUILDING)
 (CONCLUDE R-HERSE %RDN)
 (SCORE-IS (ADD 1.95 (%ASK-VALUE BUILDING (PLAN %POCH))
 (%POCH %RDN))
 rule-for-window-extraction
 (FACT (IF (AND (IS-PLAN %POCH %RDN) (%SAME-ZONE %POCH %RDN)
 (%VERTICALLY-LONG %POCH (%CONTACT %POCH (PLAN %RDN))
 (THEN (GET-SET (%LSET (PLAN %RDN) (%PATCHES)
 (AND (%L-FETCH %BL (%LSET
 (FACT (IS (LABEL %BL) %NIL)
 (%SAME-ZONE %BL (%LIVE %RDN)
 (%VERTICALLY-LONG %BL)
 (%CONTACT %BL (%LIVE (PLAN %RDN))
 (THERE-IS %AK %LIVE (%RELATION %POCH %AK)
 (ALL-FETCH %LIVE %LIVE
 (%RELATION %LIVE %LIVE))
 (THEN (CONCLUDE P-LABEL B-WINDOW)
 (FOR-EACH %LIVE (AND (%TEST-SE (%AND P-LABEL B-WINDOW)
 (%RDN %RDN) (%RDN) (%RDN) (%RDN) (%RDN) (%RDN)
 (SCORE-IS (ADD 2.1 (D1Y (NUMBER-OF %LIVE) (%RDN %RDN))
 (%POCH %RDN))
 (FACT (AND (IS-PLAN %POCH %RDN) (%SAME-ZONE %POCH %RDN)
 (THEN (CONCLUDE P-LABEL BUILDING)
 (CONCLUDE R-HERSE %RDN)
 (SCORE-IS 2.0)) (%POCH %RDN)
 D-HERSE (IF-DONE (
 (FACT %TR (%DESCRIBE-BUILDING (REGION %POCH)) (%POCH) ))
 D-CREATE (IF-DONE (
 (FACT %TR (THEN (EXTRACT-BUILDING-SHAPE (REGION %POCH)
 (%DESCRIBE-BUILDING (REGION %POCH)
 (EXECUTE PLAN-EVALUATION)) (%POCH) ))
 PRIORITY-VALUE-IS 0.2)
 %ROAD knowledge-block-of-road
 (MADE-OF (OR (%ASPHALT %CONCRETE)
 SUB-OBJECTS (%CAR %SHADOW)
 PROPERTY-RULES (
 (GEN (%LOSER %RDN) (0.8 , 0.4)) (%RDN)
 (GEN (%HORIZONTALLY-LONG %RDN) (0.7 , 0.2)) (%RDN)
 (STR (TOUCHING %RDN (%R-SIDE) (0.5 , 0.2)) (%RDN) ))
 RELATION-RULES (
 (STR (AND (%SAME-COLOR %RDN %RDN) (TOUCHING %RDN %RDN)
 (0.0 , 0.2) FOR ROAD (%RDN %RDN))
 (STR (IF (NOT (IS (OF HORIZON (SCENE)) NIL)
 (D-RATIO %RDN (OF ROAD-ZONE (SCENE)))
 (1.0 , 0.3) FOR SCENE) (%RDN) ))

```

# Why machine learning?

---

- Good old-fashioned AI (GOF AI) answer:  
Program expertise into the agent
  - Never worked (in general)

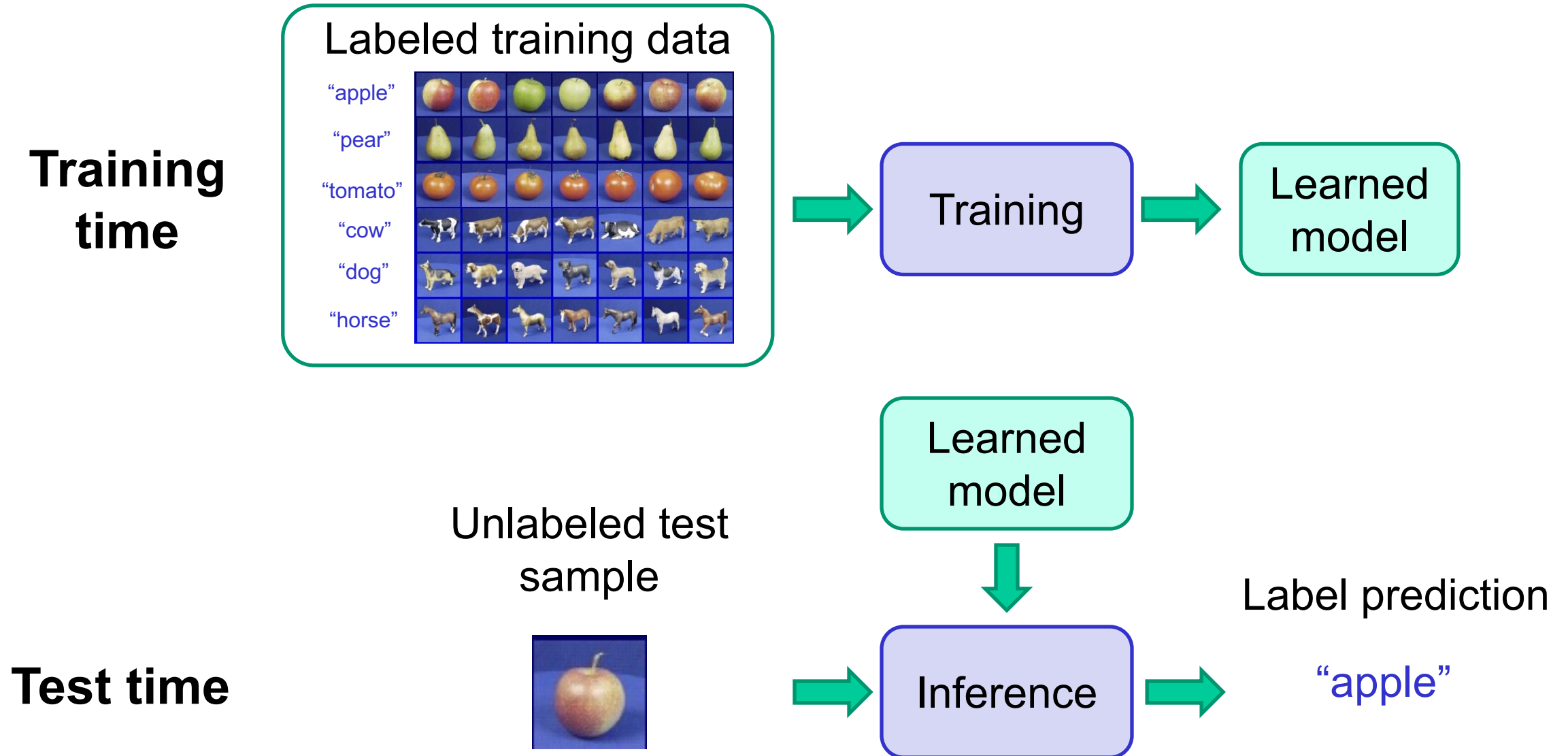
# Why machine learning?

---

- Good old-fashioned AI (GOF AI) answer:  
Program expertise into the agent
- Modern answer: Program into the agent the *ability to improve performance based on experience*
  - Experience should come from *training data* or *demonstrations*
  - We want to optimize the performance of the agent on the training data, with the hope that it will *generalize* to unseen inputs
  - This is the *statistical learning* viewpoint



# The basic ML framework (for supervised learning)



# The basic ML framework (for supervised learning)

---

$$y = f(x)$$

output                      prediction function                      input

$$y = f(x)$$

- **Training (or learning):** given a *training set* of labeled examples  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ , instantiate a predictor  $f$
- **Testing (or inference):** apply  $f$  to a new *test example*  $x$  and output the predicted value  $y = f(x)$

- **Training (or learning):** given a *training set* of labeled examples  $\{(x_1, y_1), \dots, (x_N, y_N)\}$ , instantiate a predictor  $f$
- **Testing (or inference):** apply  $f$  to a new *test example*  $x$  and output the predicted value  $y = f(x)$

- Rather than hand-defining how 2D projections of apples are different from pears,  $f$  will learn this from the data.



- Rather than hand-defining how 2D projections of apples are different from pears,  $f$  will learn this from the data.

# Deep Learning

---

- A general way to model function  $f$  as composition (layers) of simple functions, very loosely inspired by the brain.

# A few historical milestones

- 1958: [Rosenblatt's perceptron](#), aka linear classifier

## NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo  
of Computer Designed to  
Read and Grow Wiser

WASHINGTON, July 7 (UPI)—The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's \$2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of \$100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human be-

ings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

### Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

## 1958 New York Times...

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

### Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.



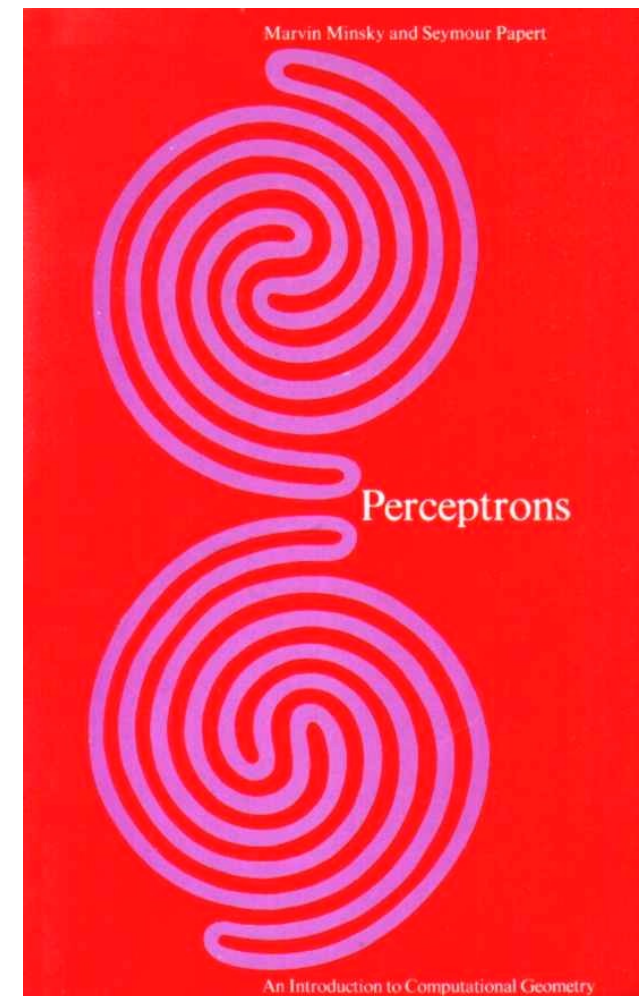
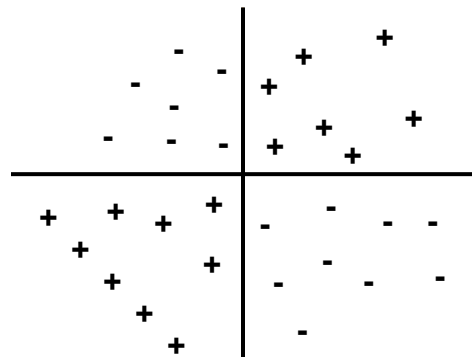
[Frank Rosenblatt](#) (1928-1971)



# A few historical milestones

---

- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
  - Made the case that perceptrons could not even learn the XOR function.

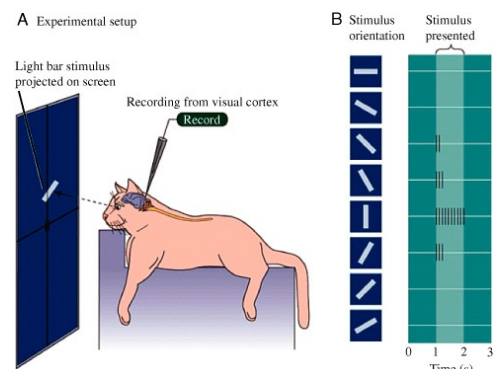


# A few historical milestones

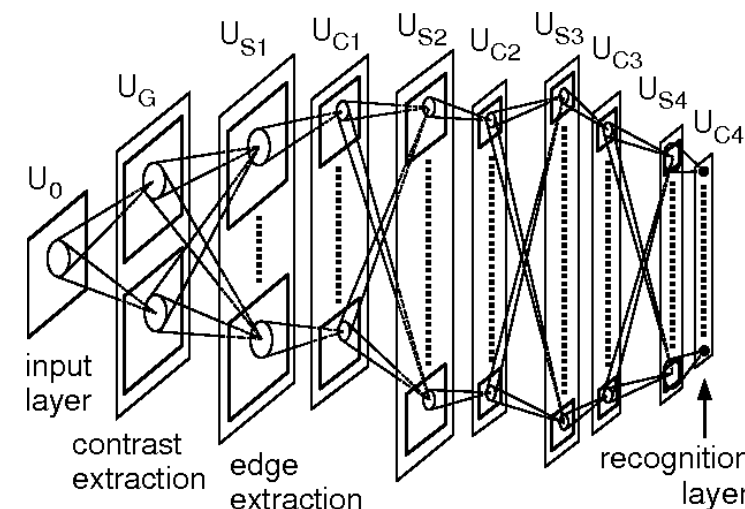
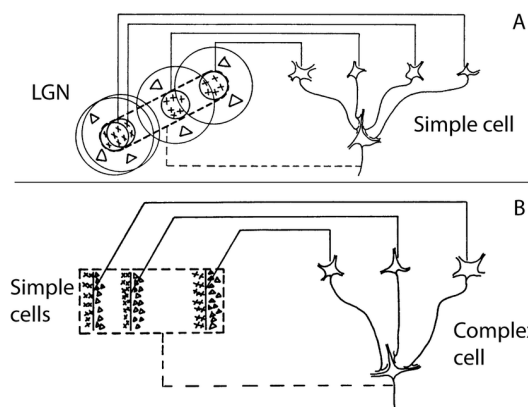
- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
- 1980: [Fukushima's Neocognitron](#)
  - [Video \(short version\)](#)
  - Inspired by the findings of [Hubel & Wiesel](#) about the hierarchical organization of the visual cortex in cats and monkeys (1959-1977)



[Kunihiko Fukushima](#)



[Image source](#)



# A few historical milestones

---

- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
- 1980: [Fukushima's Neocognitron](#)
- 1986: [Back-propagation](#) aka chain rule
  - Origins in control theory and optimization: Kelley (1960), Dreyfus (1962), Bryson & Ho (1969), Linnainmaa (1970)
  - Application to neural networks: Werbos (1974)
  - Popularized by Rumelhart, Hinton & Williams (1986)

# A few historical milestones

- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
- 1980: [Fukushima's Neocognitron](#)
- 1986: [Back-propagation](#)
- 1989 – 1998: [Convolutional neural networks](#)
  - LeNet to LeNet-5



[Yann LeCun](#)  
[2018 ACM Turing Award winner](#)  
(with Hinton and Bengio)

PROC. OF THE IEEE, NOVEMBER 1998

7

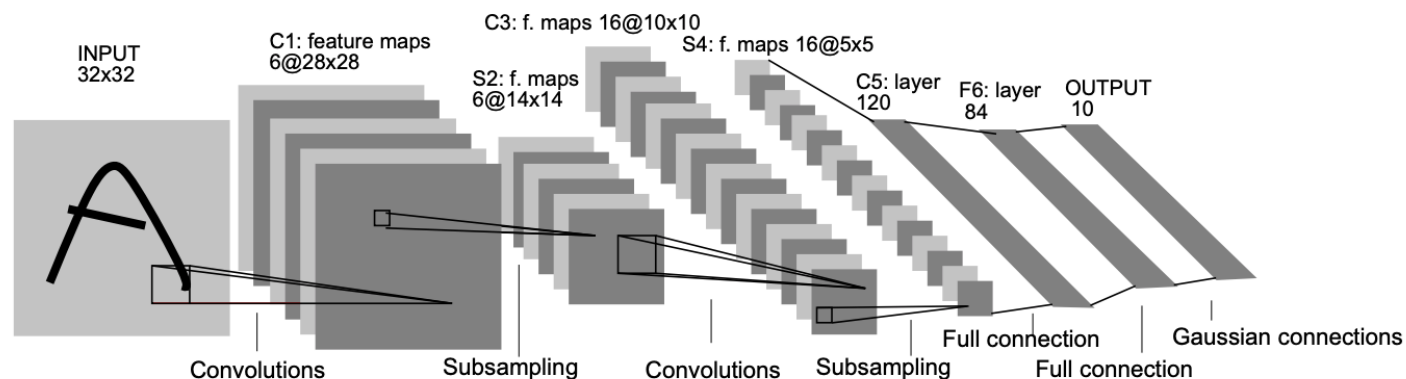


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

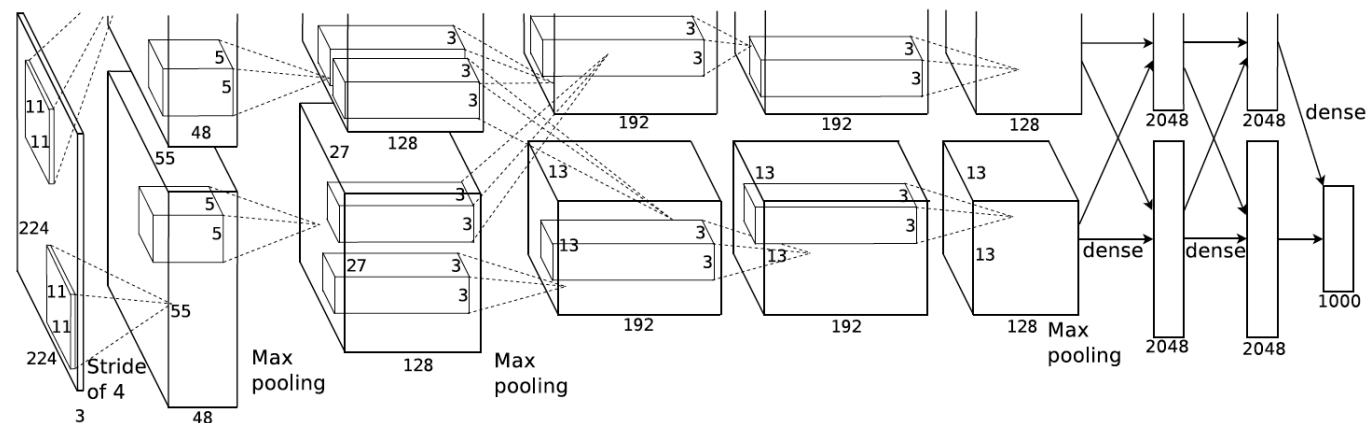


# A few historical milestones

- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
- 1980: [Fukushima's Neocognitron](#)
- 1986: [Back-propagation](#)
- 1989 – 1998: [Convolutional neural networks](#)
- 2012: [AlexNet](#)

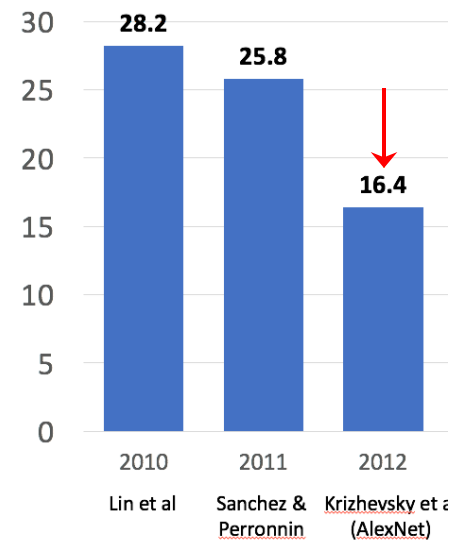


[Photo source](#)



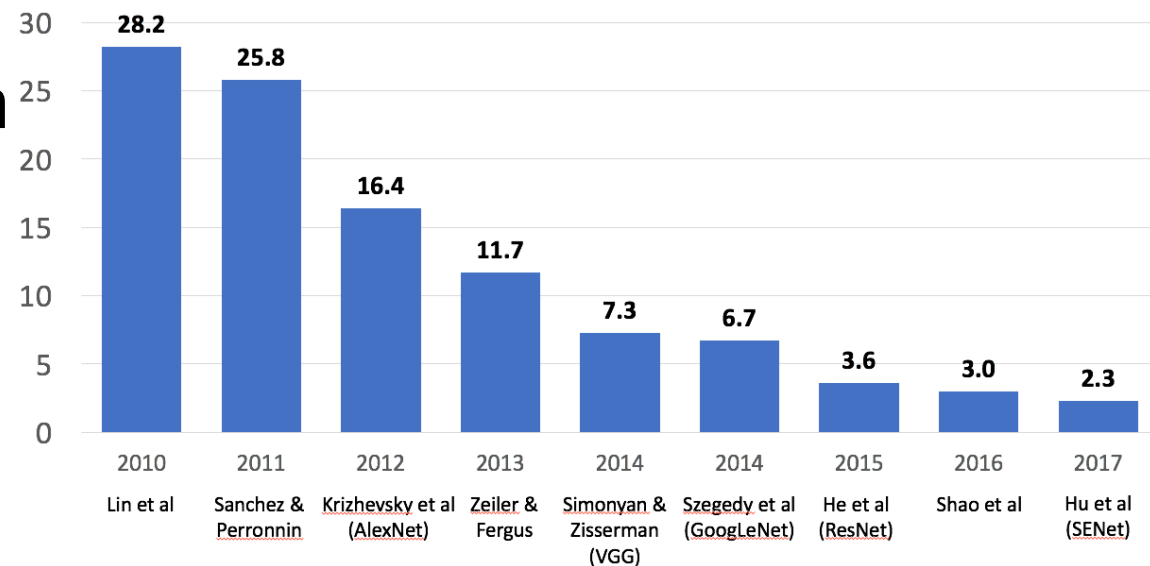
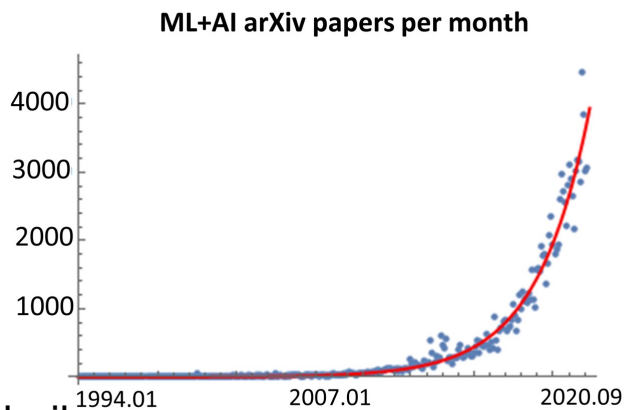
# A few historical milestones

- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
- 1980: [Fukushima's Neocognitron](#)
- 1986: [Back-propagation](#)
- 1989 – 1998: [Convolutional neural networks](#)
- 2012: [AlexNet](#)

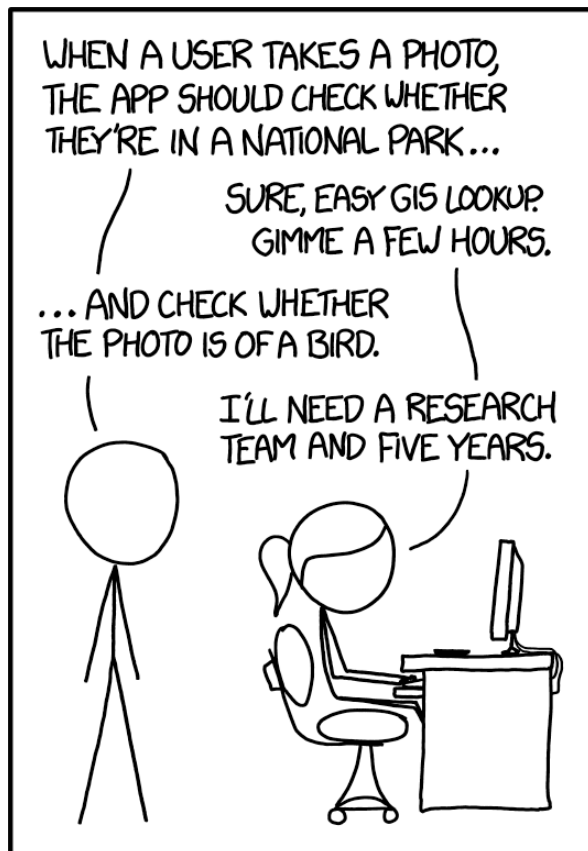


# A few historical milestones

- 1958: [Rosenblatt's perceptron](#)
- 1969: [Minsky and Papert Perceptrons book](#)
- 1980: [Fukushima's Neocognitron](#)
- 1986: [Back-propagation](#)
- 1989 – 1998: [Convolutional neural networks](#)
- 2012: [AlexNet](#)
- 2012 – : deep learning explosion

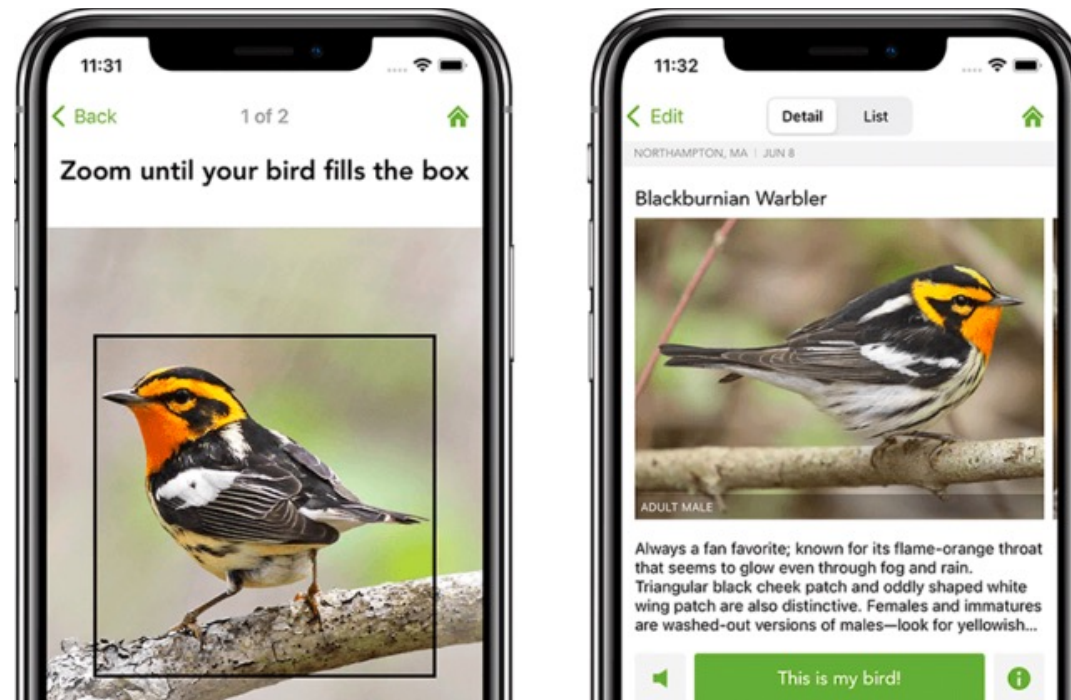


# What can current deep learning CV systems do?



TheCornellLab

## Merlin®



- Entirely true as of 2014 (or so) when this [xkcd](#) was published

- Today, phone apps that can do a reasonable job



# What can current deep learning CV systems do?

---



K. He, G. Gkioxari, P. Dollar, and R. Girshick, [Mask R-CNN](#),  
ICCV 2017 (Best Paper Award)

# What can current deep learning CV systems do?

---

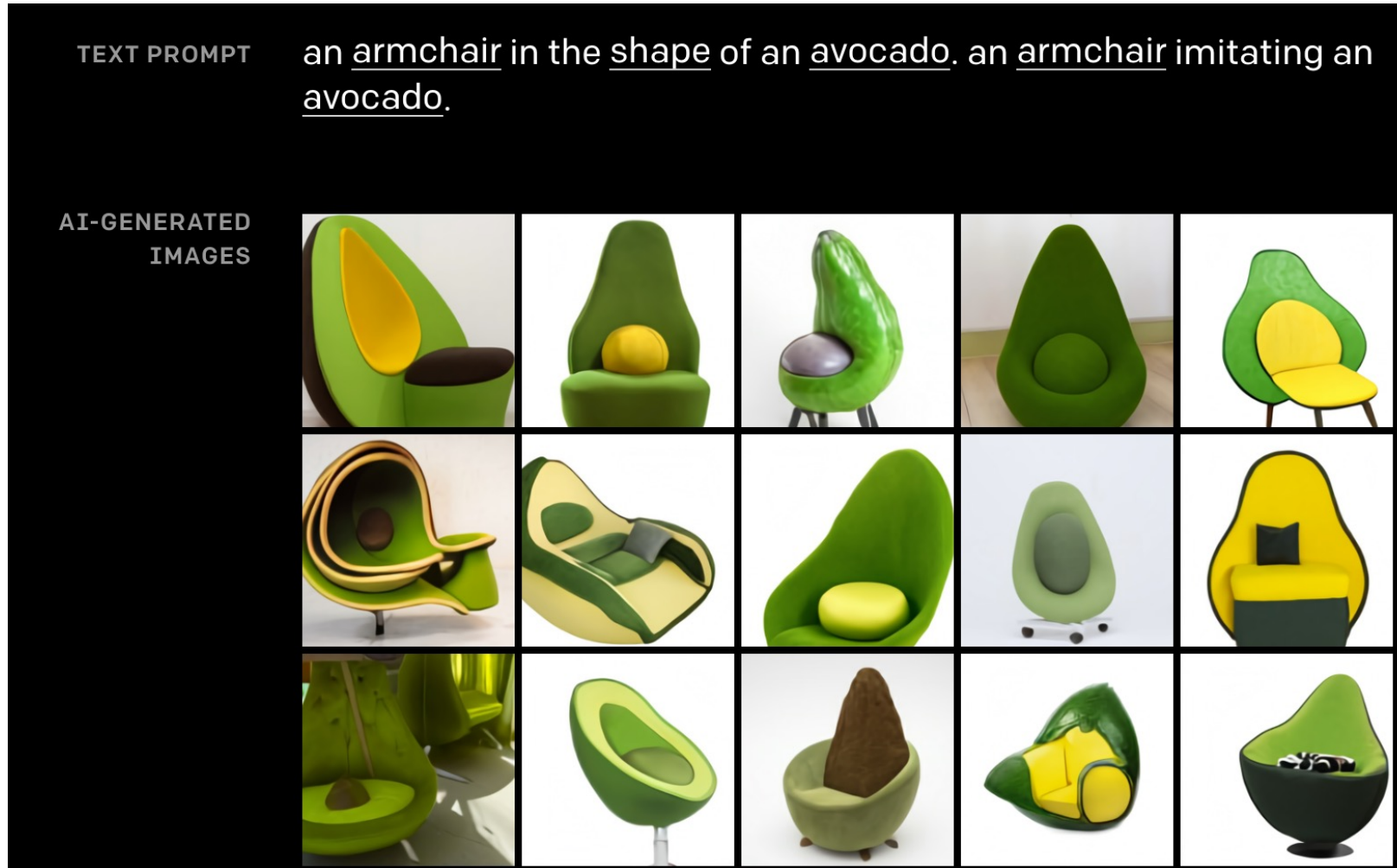
## Shape and Pose Estimation for Objects and Humans





# What can current deep learning CV systems do?

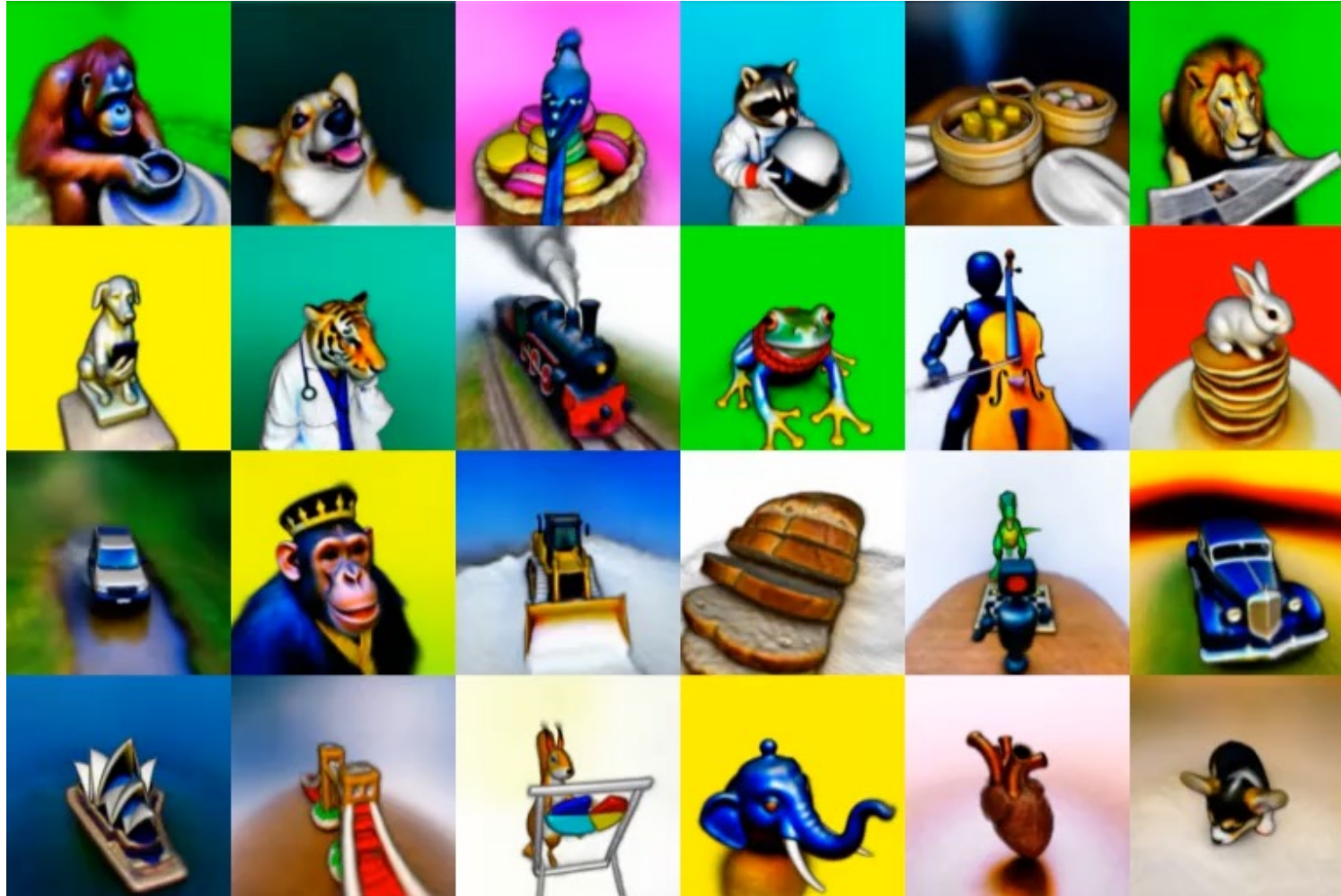
## Image Generation



# What can current deep learning CV systems do?

---

## 3D Generation



B. Poole, A. Jain, J. Barron, B. Mildenhall. [DreamFusion: Text-to-3D using 2D Diffusion.](#) arXiv 2022



# What can current deep learning CV systems do?

---

Video Generation

A teddy bear washing dishes

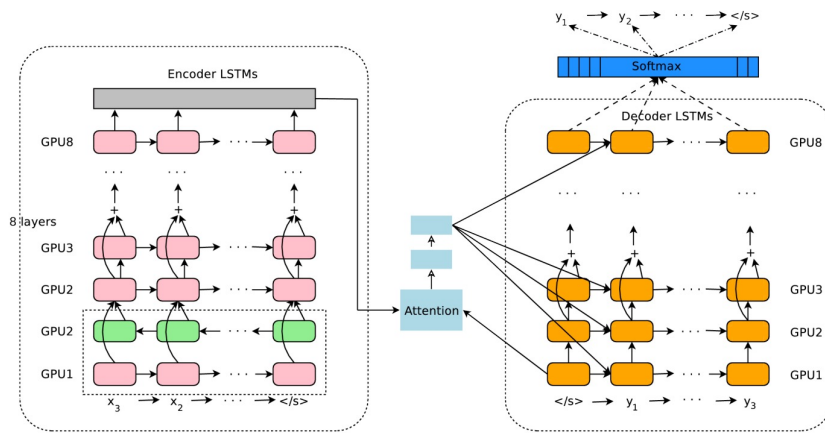


Jonathan Ho\*, William Chan\*, Chitwan Saharia\*, Jay Whang\*, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David Fleet, Tim Salimans\*. [Imagen Video: High-definition video generation with diffusion models](#) arXiv 2022

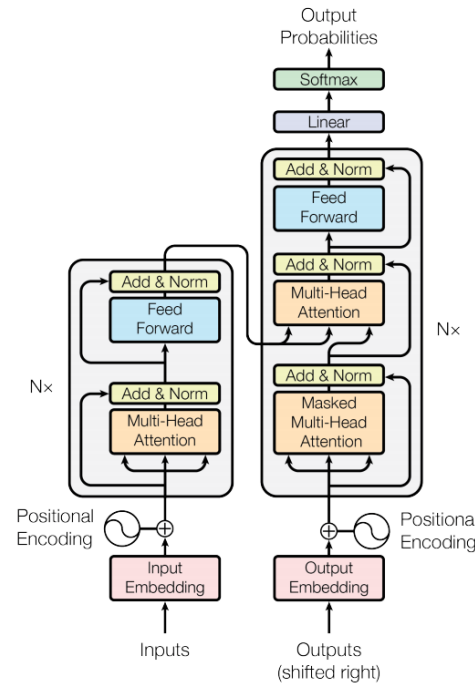
# What else can deep learning do?

## Natural Language Processing (e.g. machine translation)

### Google Neural Machine Translation (GNMT)



### Transformers



### English French Translation Quality (BLEU score)

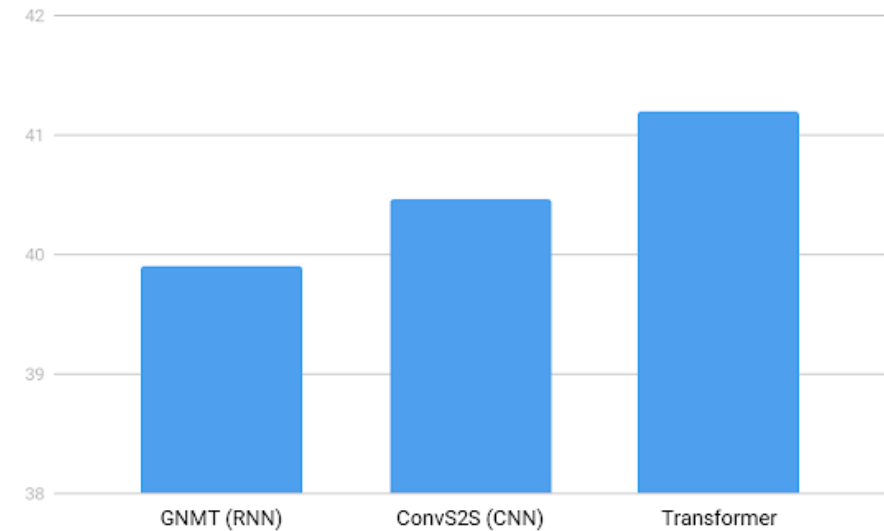


Figure source

Y. Wu et al. [Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](#). arXiv 2016

<https://mobile.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html>

A. Vaswani et al. [Attention is all you need](#). NeurIPS 2017

Previous system (before deep learning):  
PBMT (2014): 37 BLEU

# ChatGPT

---

SL

Write a poem explaining ChatGPT for an Introduction to Deep Learning class



ChatGPT is a model that's deep and grand,  
Built with machine learning in hand,  
It processes language with such ease,  
Giving answers that often please.

It's based on the transformer architecture,  
Which allows for seamless interaction,  
It's trained on vast amounts of data,  
Giving it knowledge that's second to none, one could say.

It can generate text that's brand new,  
Or complete the sentence you started, too,  
It can answer questions with great accuracy,  
And even write poetry with a certain fluidity.

But ChatGPT is more than just a tool,  
It's a glimpse into the future, that's cool,  
With each improvement and update,  
We inch closer to machines that think and debate.

So let ChatGPT be your guide,  
As you explore the world of deep learning inside,  
And who knows, maybe one day you'll build,  
A model that's even greater still.

Generated on 1/10/2023

# ChatGPT

---

SL

Explain ChatGPT to an eight-year-old



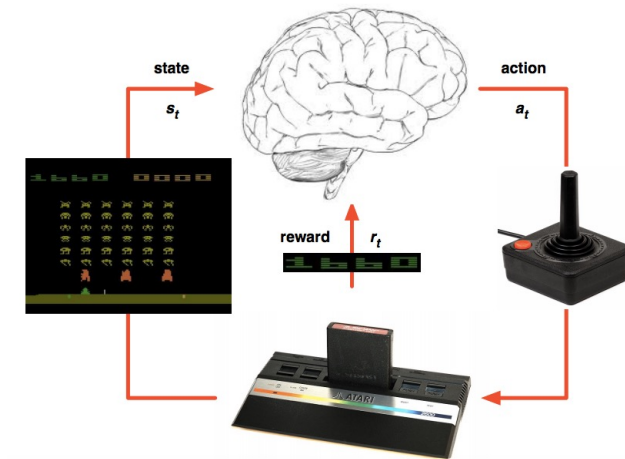
ChatGPT is a computer program that can understand and write in the English language. It's like having a super smart friend who can help you write stories, answer questions and even make up new words. It works by looking at a lot of examples of how people talk and write and then it can use what it has learned to help you. It's kind of like how you learn new words and sentences by listening to people talk and reading books. And it's fun to use because you can ask it to write stories or make up jokes and it can come up with some pretty funny stuff!



# What else can Deep Learning do?

## Decision Making

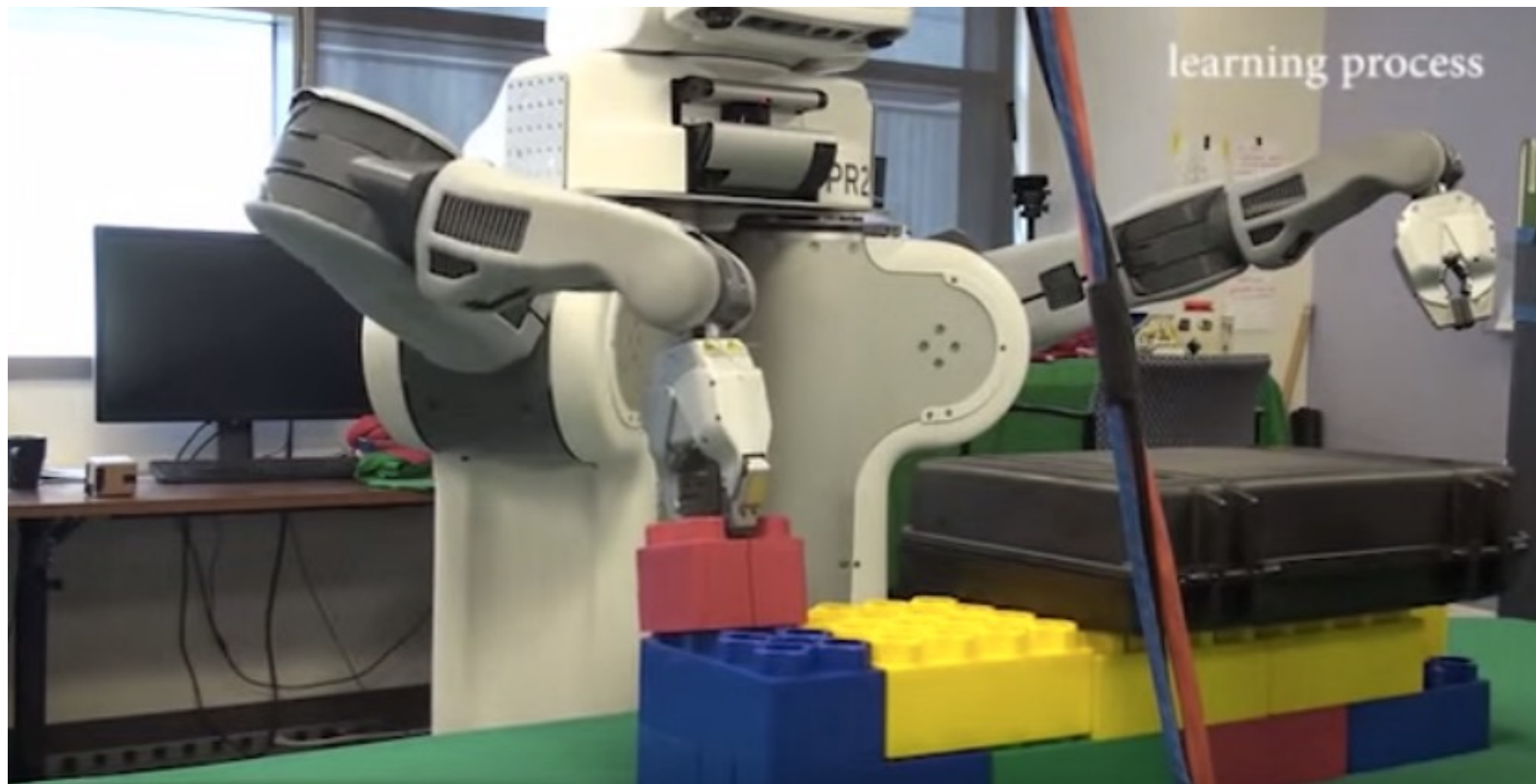
- 2013: [DeepMind uses deep reinforcement learning to beat humans at some Atari games](#)
- 2016: [DeepMind's AlphaGo system beats Go grandmaster Lee Sedol 4-1](#)
- 2017: [AlphaZero learns to play Go and chess from scratch](#)
- 2019: [DeepMind's StarCraft 2 AI is better than 99.8 percent of all human players](#)



# What else can Deep Learning do?

---

## Sensorimotor Control



[Overview video](#),  
[training video](#)

S. Levine, C. Finn, T. Darrell, P. Abbeel, [End-to-end training of deep visuomotor policies](#), JMLR 2016



# What else can Deep Learning do?

## Sensorimotor Control

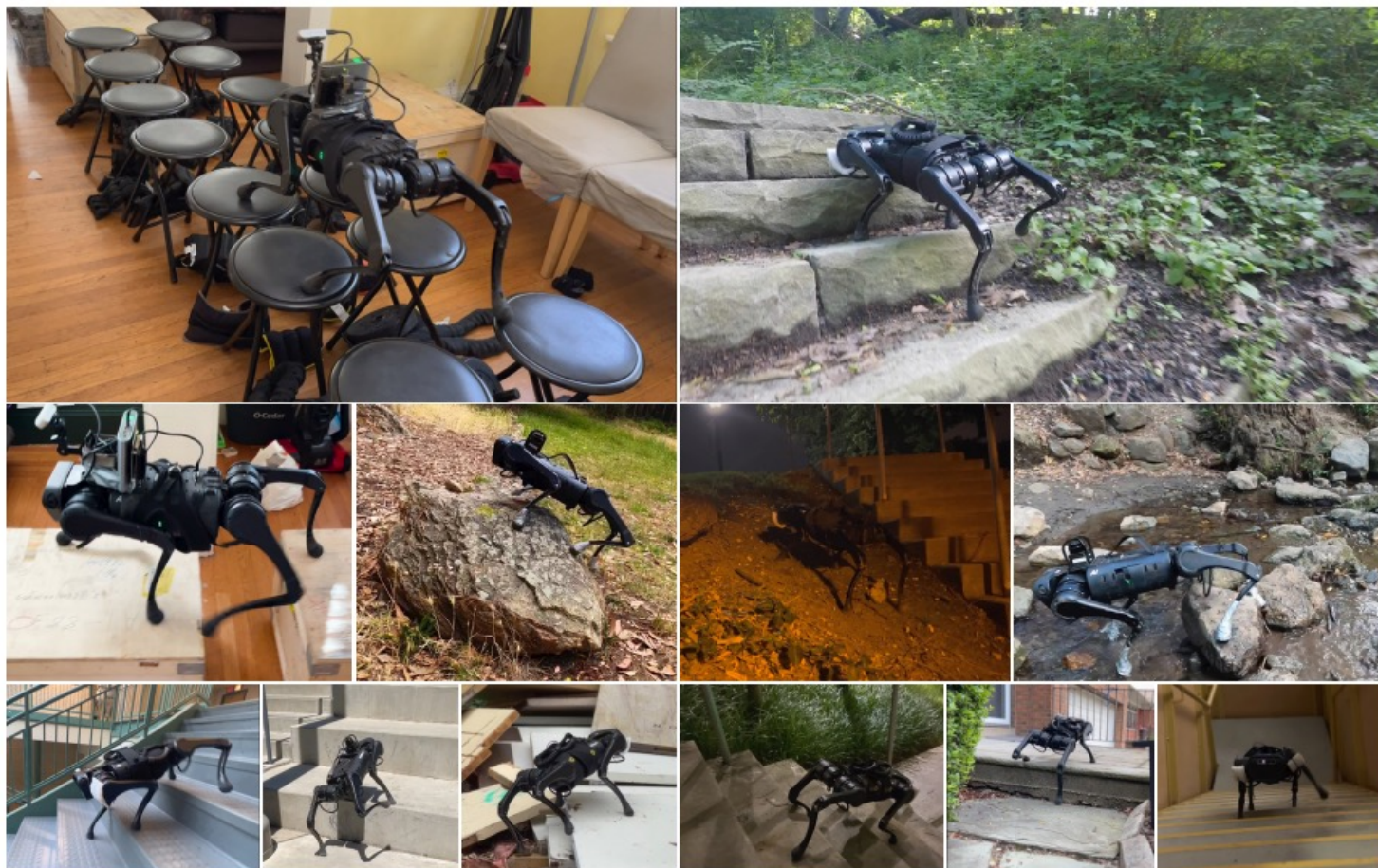
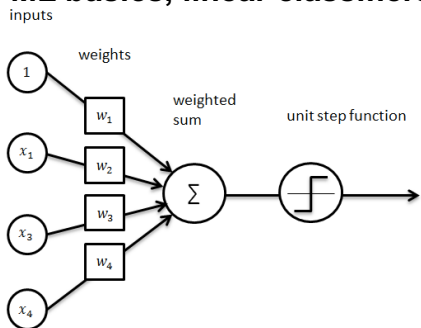


Figure 1: Our robot can traverse a variety of challenging terrain in indoor and outdoor environments, urban and natural settings during day and night using a single front-facing depth camera. The robot can traverse curbs, stairs and moderately rocky terrain. Despite being much smaller than other commonly used legged robots, it is able to climb stairs and curbs of a similar height. Videos at <https://vision-locomotion.github.io>

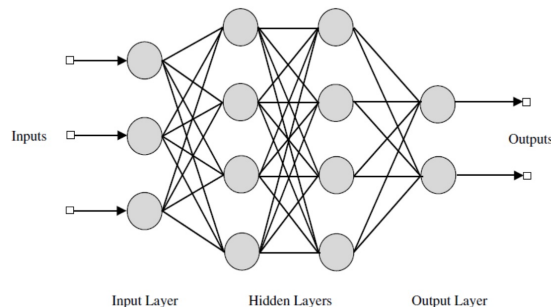
A. Agarwal, A. Kumar, J. Malik, and D. Pathak. [Legged Locomotion in Challenging Terrains using Egocentric Vision](#). CoRL 2022

# Topics to be covered in class

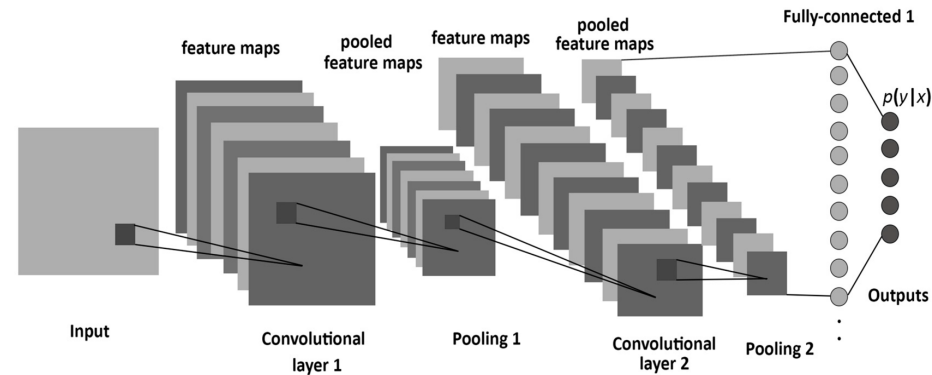
## ML basics, linear classifiers



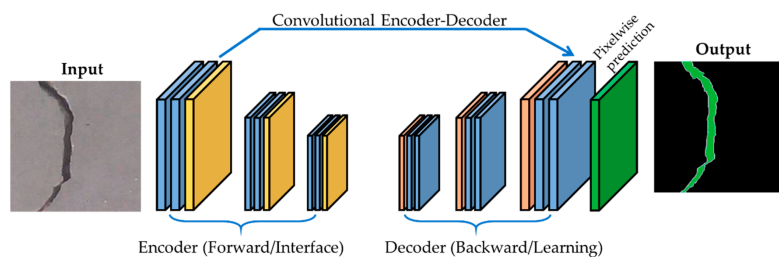
## Multilayer neural networks, backpropagation



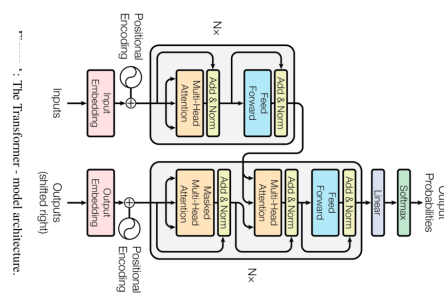
## Convolutional networks for classification



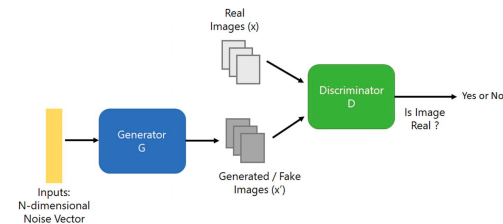
## Networks for detection, dense prediction



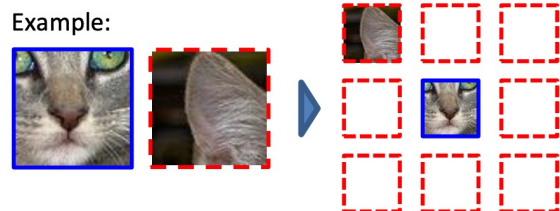
## Sequence Models: Recurrent models, Transformers



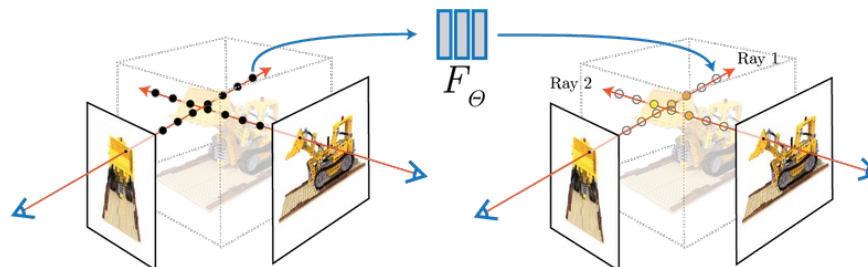
## Generative models (GANs, variational auto-encoders, diffusion models)



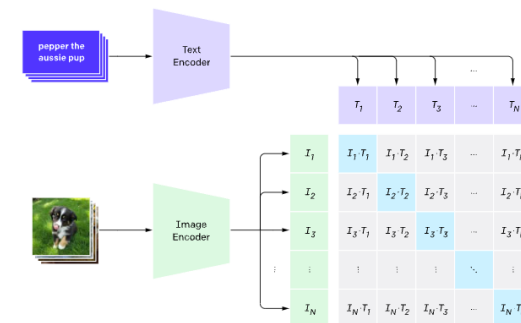
## Self-supervised learning



## Neural Radiance Fields



## Large-language Models, Vision and Language, ...





# Course Staff

---



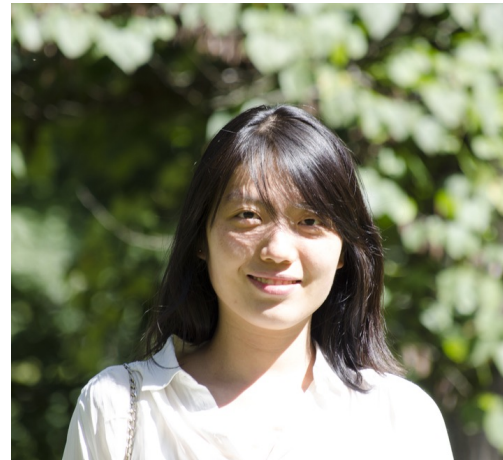
Saurabh Gupta



Xiaodan Hu



Chaoran Cheng



Yana Zhao



Anshul Bheemreddy