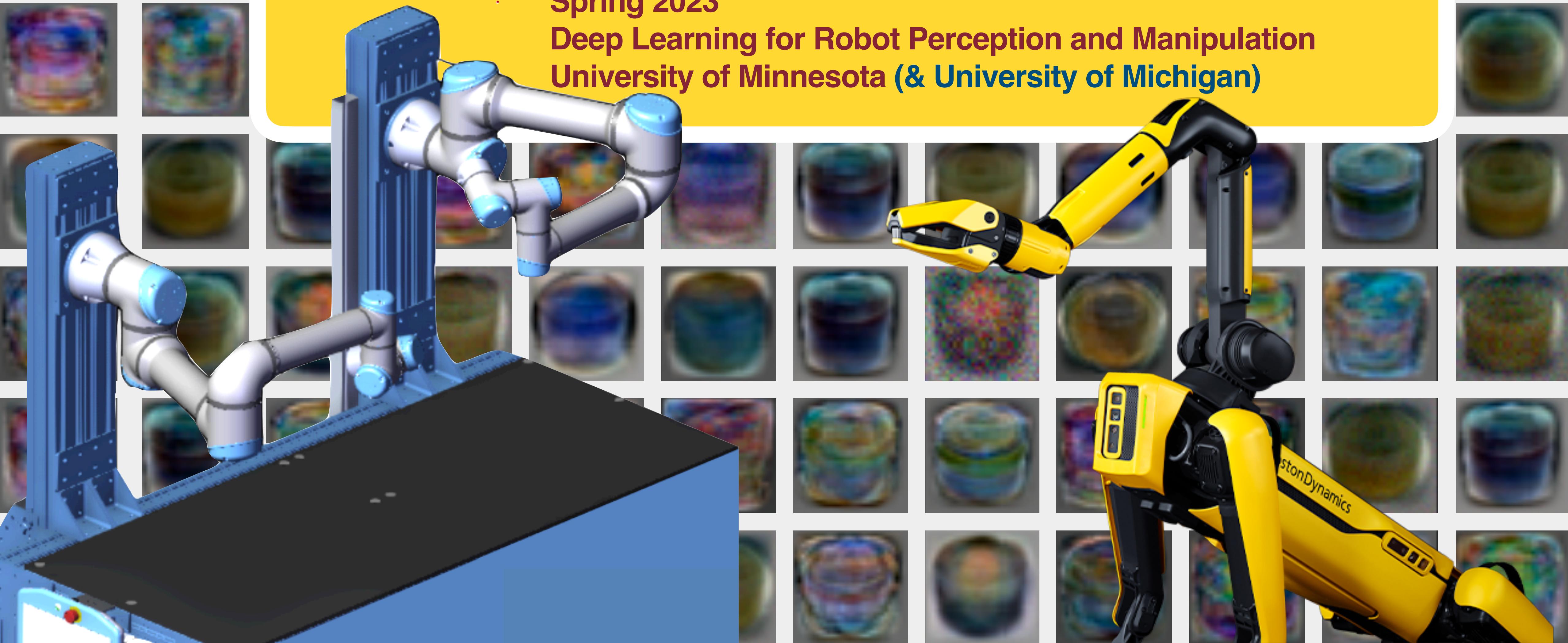


DR

DeepRob

Spring 2023

Deep Learning for Robot Perception and Manipulation
University of Minnesota (& University of Michigan)



DR

4Progress



4Progress

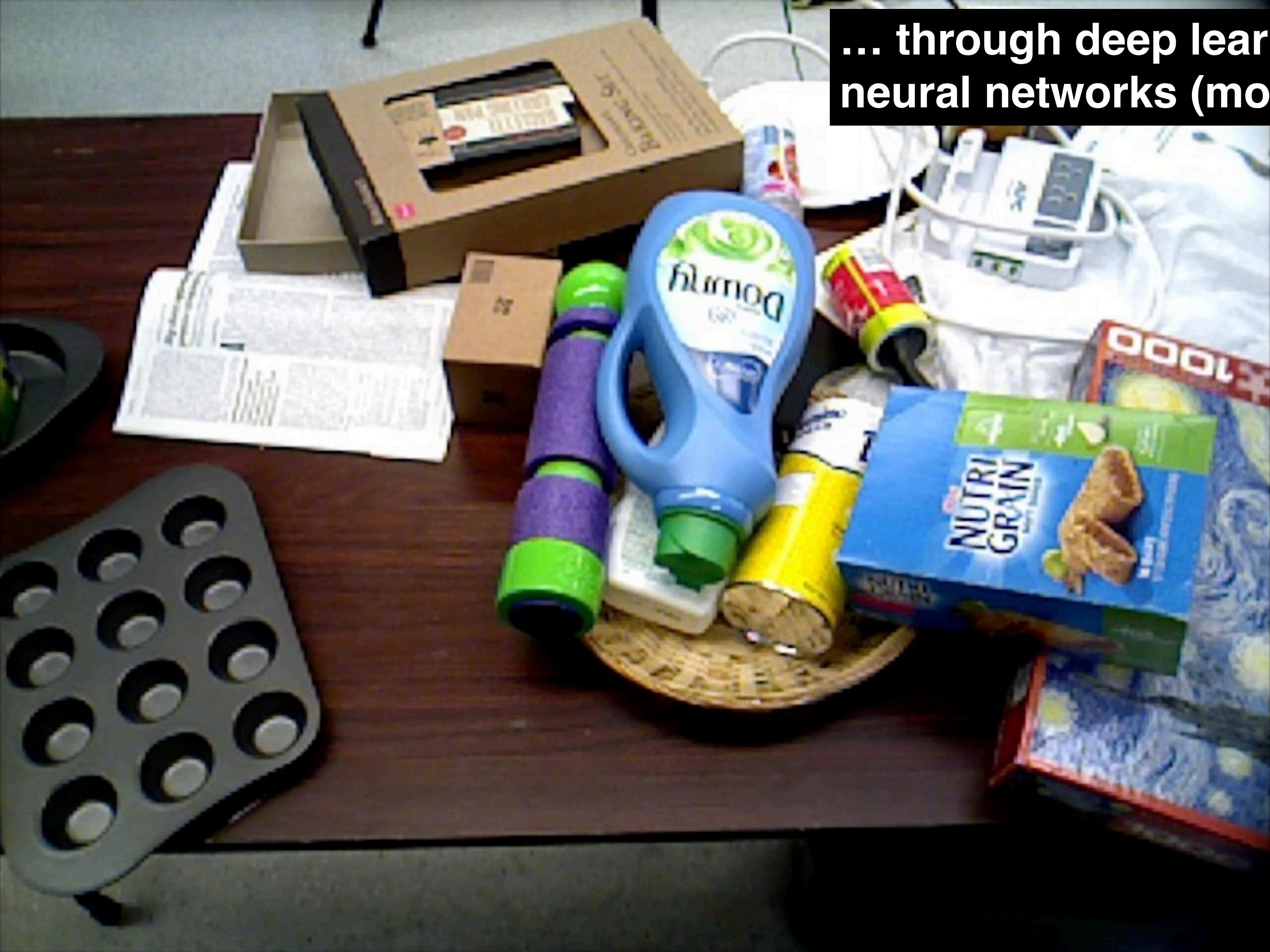
Sui, Zhou, Zeng, Jenkins 2017

Goal:
Perception and manipulation
of objects in clutter...



DR

... through deep learning with
neural networks (mostly) ...



Sample Efficient Grasp Learning Using Equivariant Models

Xupeng Zhu, Dian Wang, Ondrej Biza, Guanang Su, Robin Walters, Robert Platt

Khoury College of Computer Sciences
Northeastern University
Boston, Massachusetts

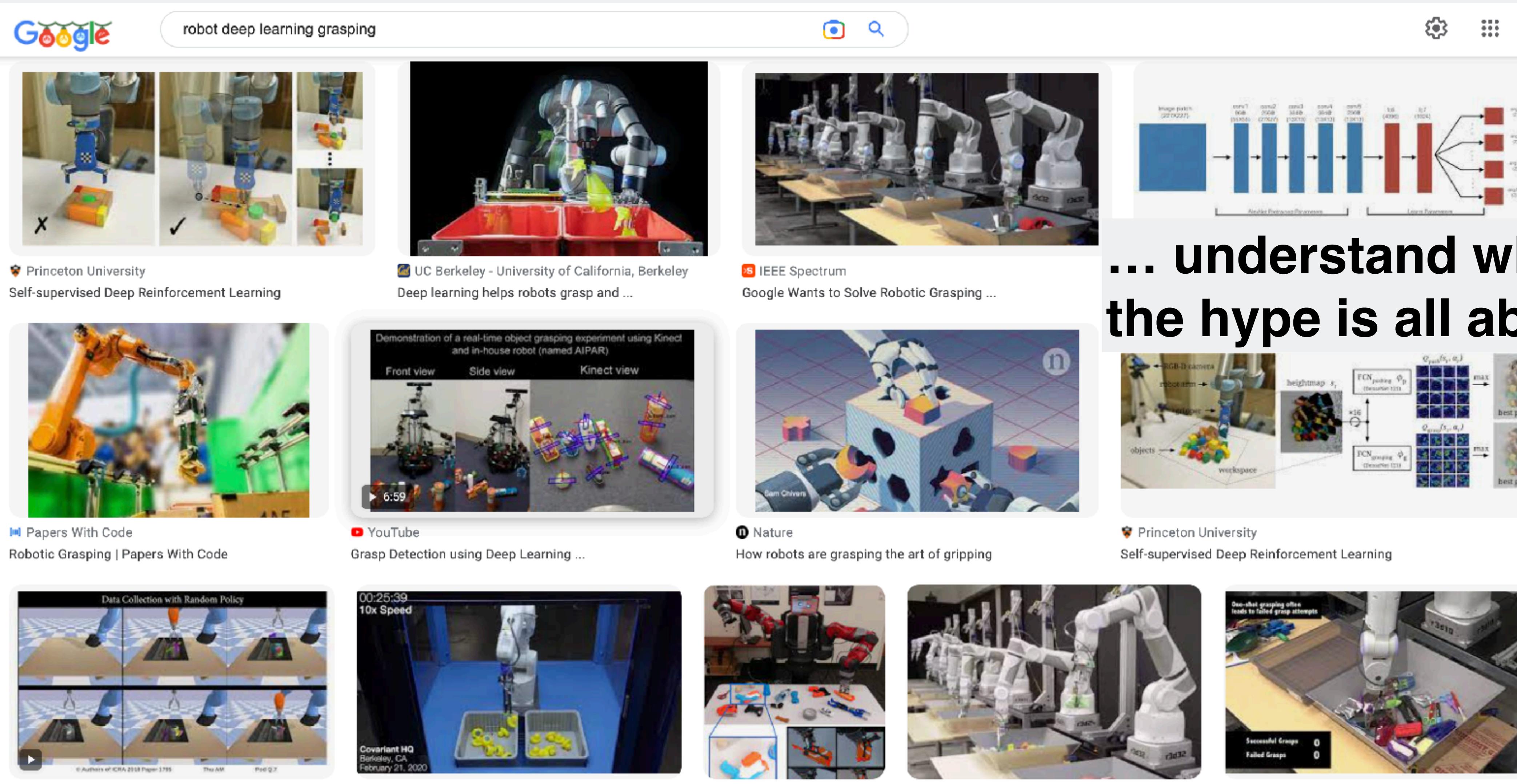




**Dual-arm setup in the Robotics:
Perception and Manipulation (RPM)
Lab @ Minnesota**

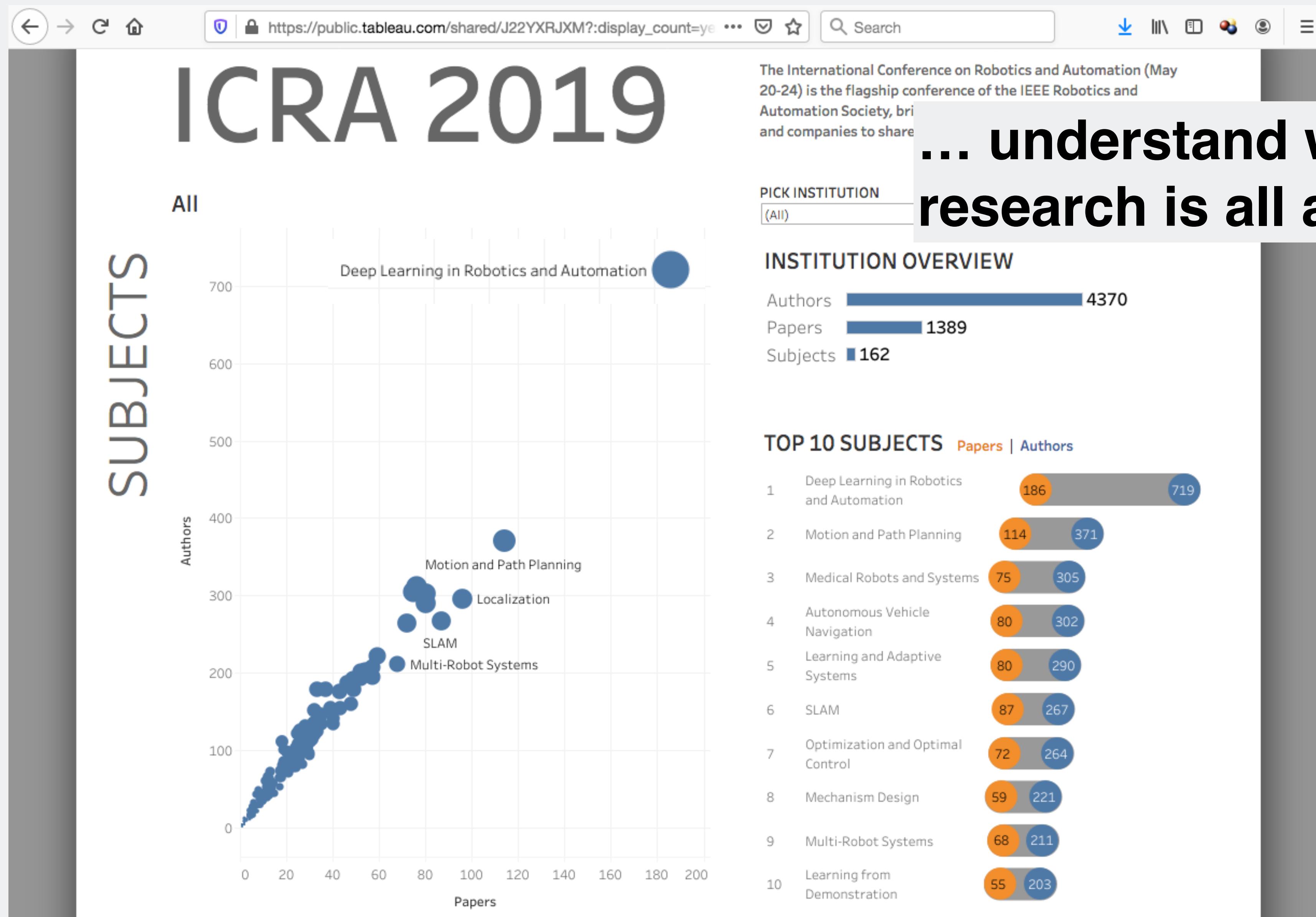


Carl Winge's demo



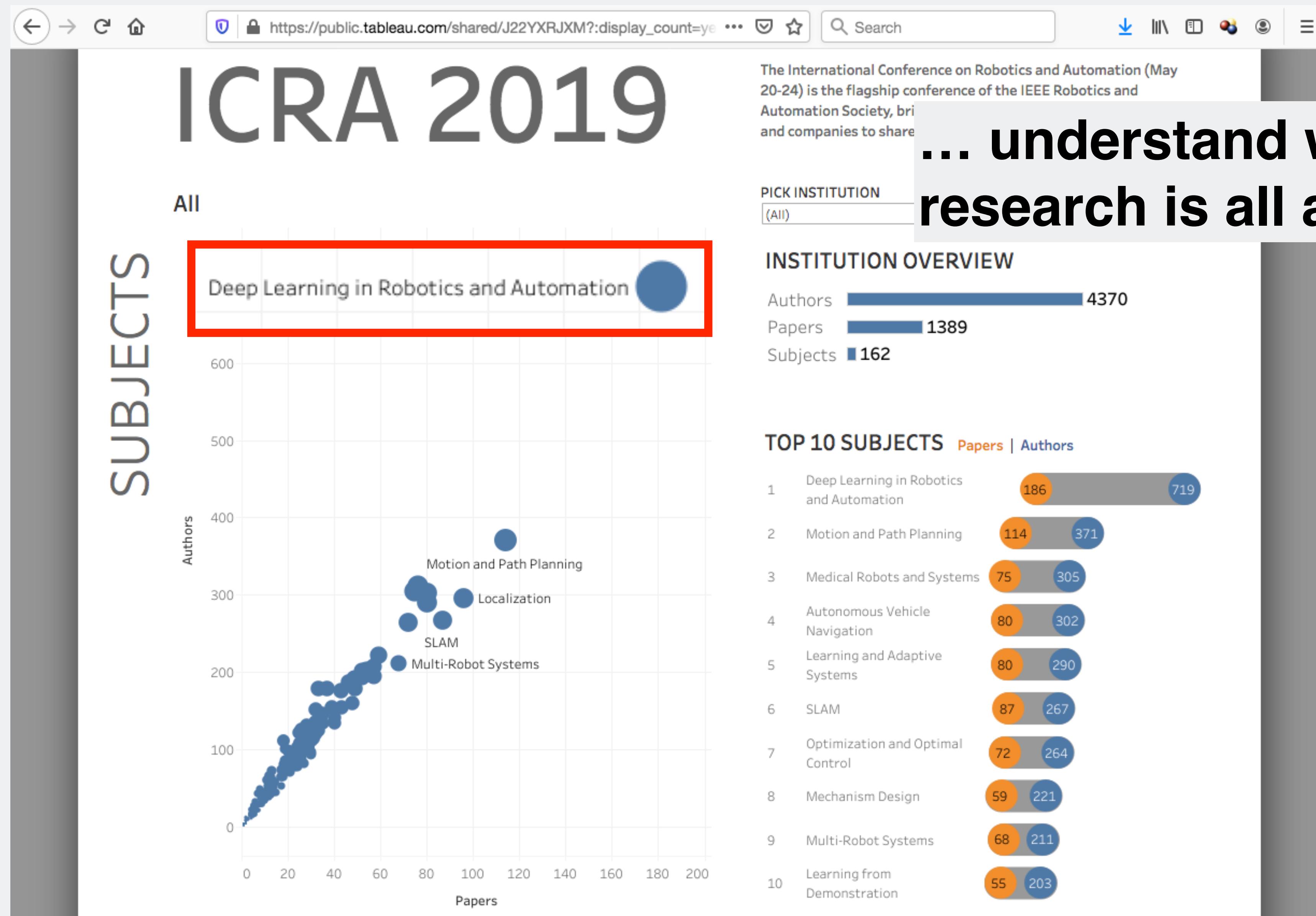
... understand what
the hype is all about ...





... understand what the research is all about ...





... understand what the research is all about ...





Welcome to DeepRob!





Welcome to DeepRob!



Anthony Opiari



Prof. Chad Jenkins



Prof. Karthik Desingh





DeepRob is a fork (and a star) of Deep Learning for Computer Vision

M
UNIVERSITY OF MICHIGAN

EECS 498.008 / 598.008
Deep Learning for Computer Vision
Winter 2022

Schedule

Lectures are Mondays and Wednesdays, 4:30pm to 6pm. Attendance is not required. Recordings will be posted after each lecture in case you are unable to attend the scheduled time.

We will hold lectures remotely over Zoom for the first few weeks. See Canvas for the Zoom meeting information.

Some lectures have reading drawn from the course notes of [Stanford CS 231n](#), written by [Andrej Karpathy](#).

Some lectures have optional reading from the book *Deep Learning* by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (GBC for short). The entire book is [available for free online](#) so you don't need to buy a copy.

Event	Date	Description	Course Materials
Lecture 1 [Remote]	Wednesday January 5	Course Introduction Computer vision overview Historical context Course logistics	[slides] [video (UMich only)] [FA2019 video (public)] [Python tutorial] [GBC Sec 1.2] [GBC Sec 6.6]
Lecture 2 [Remote]	Monday January 10	Image Classification Data driven approach K-Nearest Neighbor Hyperparameters Cross-validation	[slides] [video (UMich only)] [FA2019 video (public)] [231n Image Classification]
Lecture 3 [Remote]	Wednesday January 12	Linear Classifiers Algebraic / Visual / Geometric viewpoints Softmax / SVM classifiers	[slides] [video (UMich only)] [FA2019 video (public)] [231n Linear Classification]

Assignment 1 Due

**Similar introductory projects,
New projects for 3D estimation,
Build up to mini research project**

CS231n Home Course Notes Coursework Schedule Office Hours Final Projects Ed

CS231n: Deep Learning for Computer Vision

Stanford - Spring 2022

* This network is running live in your browser.

Course Description

Computer Vision has become ubiquitous in our society, with applications in search, image understanding, apps, mapping, medicine, drones, and self-driving cars. Core to many of these applications are visual recognition tasks such as image classification, localization and detection. Recent developments in neural network (aka "deep learning") approaches have greatly advanced the performance of these state-of-the-art visual recognition systems. This course is a deep dive into the details of deep learning architectures with a focus on learning end-to-end models for these tasks, particularly image classification. During the 10-week course, students will learn to implement and train their own neural networks and gain a detailed understanding of cutting-edge research in computer vision. Additionally, the final assignment will give them the opportunity to train and apply multi-million parameter networks on real-world vision problems of their choice. Through multiple hands-on assignments and the final course project, students will acquire the toolset for setting up deep learning tasks and practical engineering tricks for training and fine-tuning deep neural networks.

Previous Offerings





A brief history of AI



1956 Dartmouth Conference: The Founding Fathers of AI



John McCarthy



Marvin Minsky



Claude Shannon



Ray Solomonoff



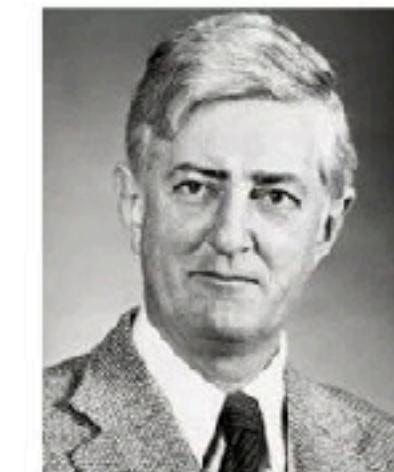
Alan Newell



Herbert Simon



Arthur Samuel



Oliver Selfridge



Nathaniel Rochester



Trenchard More

1956

1960

1970

1980

1990

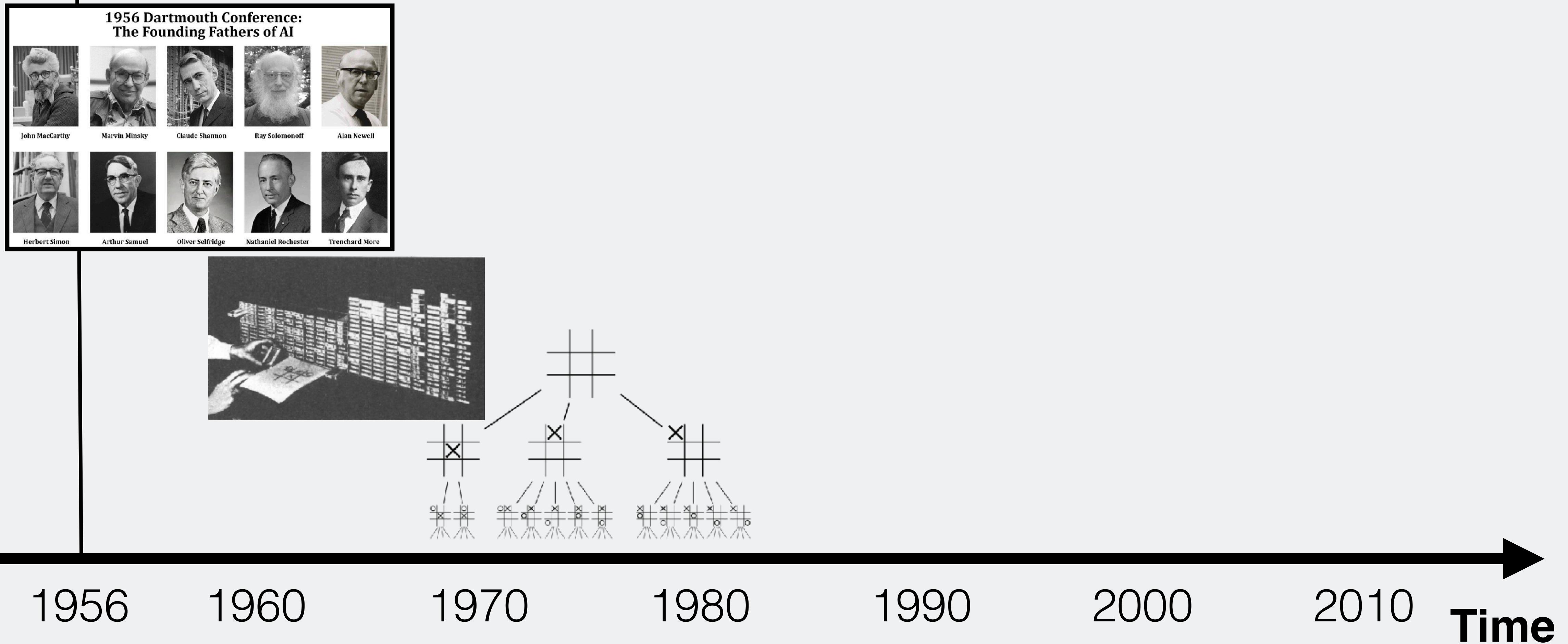
2000

2010

Time

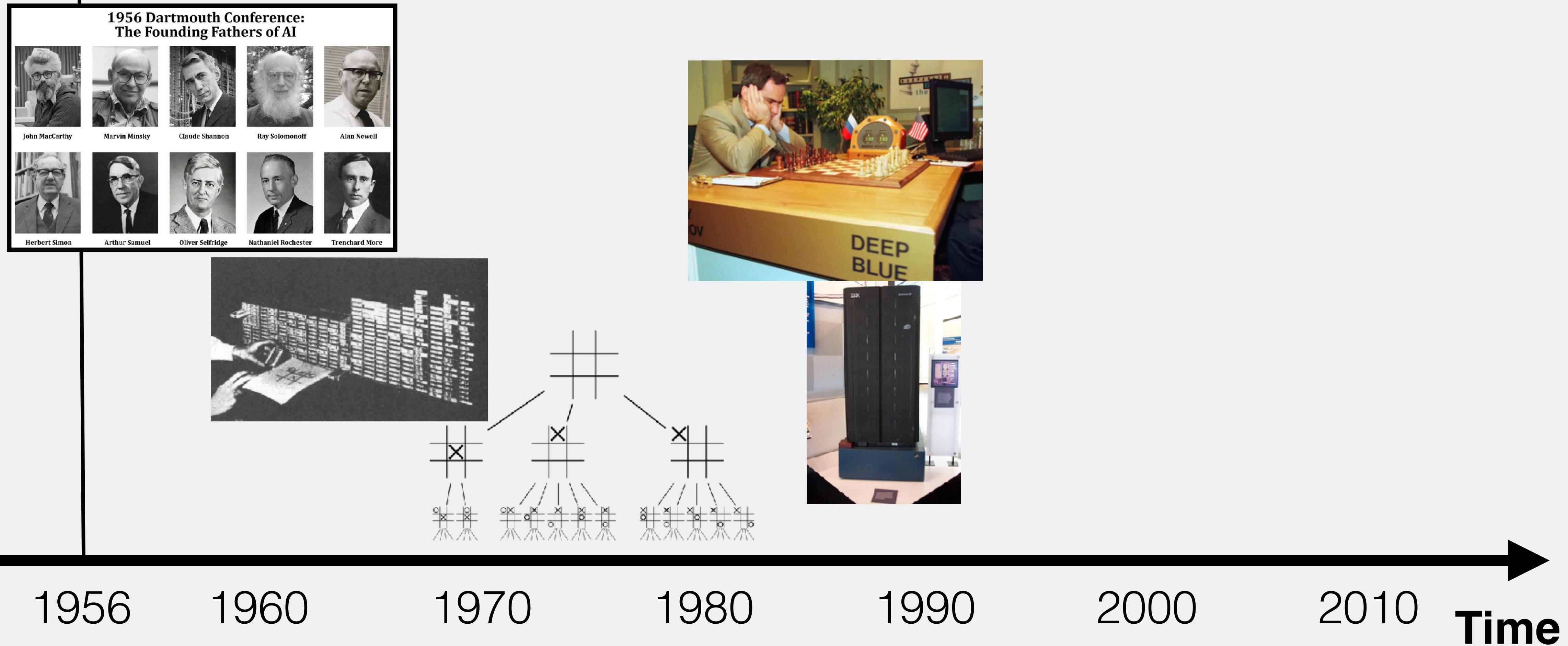
First wave AI: Model-based

“Think through the entire problem”



First wave AI: Model-based

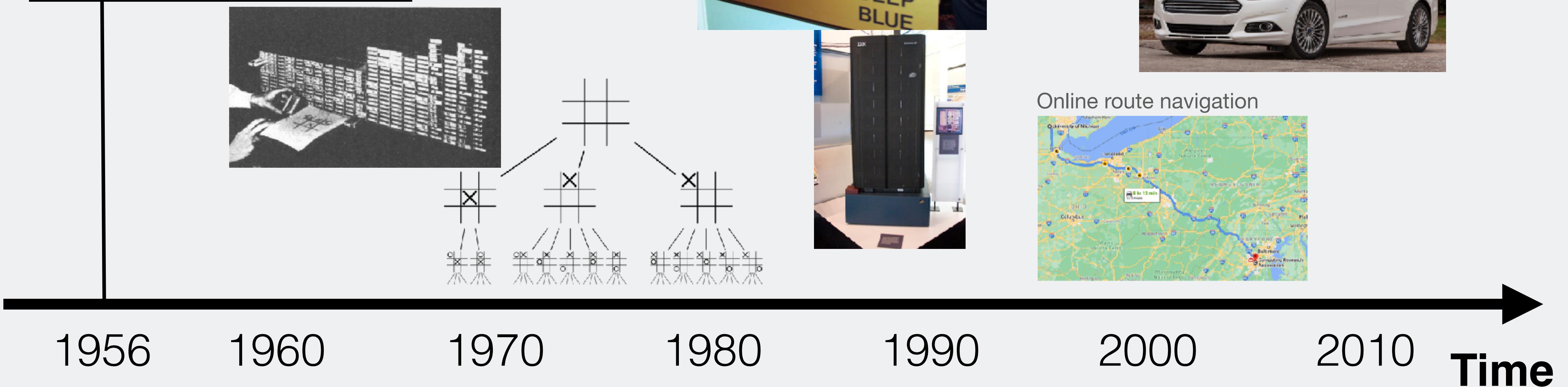
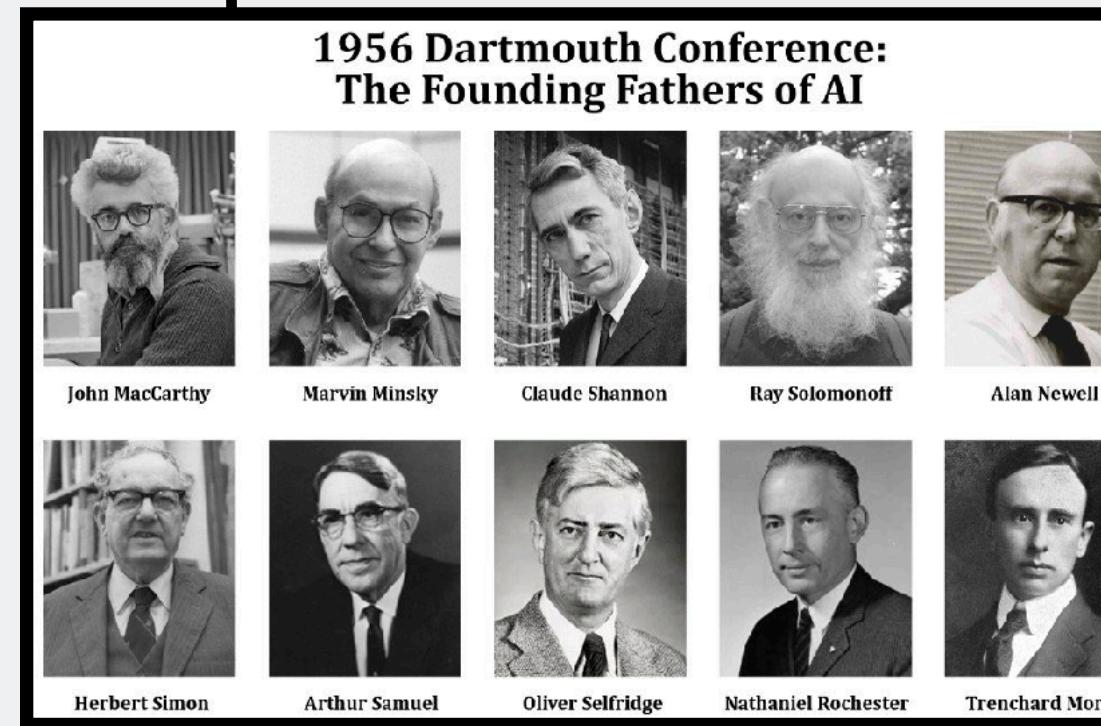
“Think through the entire problem”



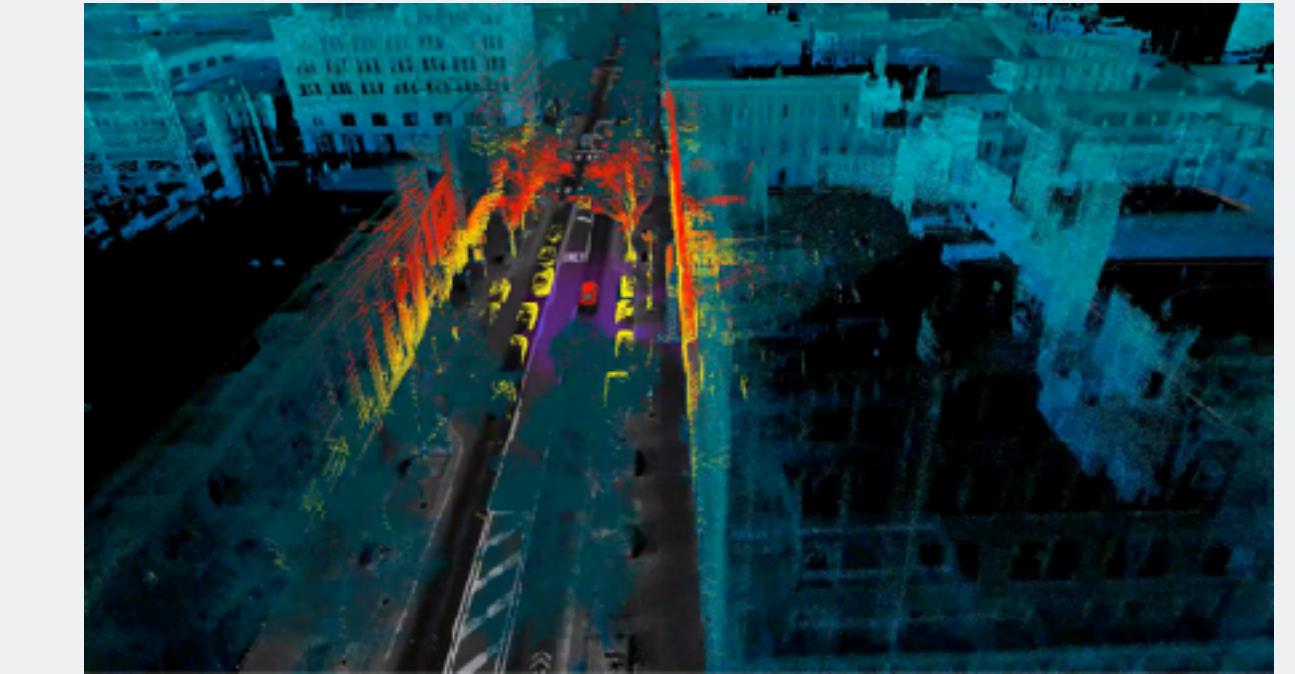


First wave AI: Model-based

“Think through the entire problem”



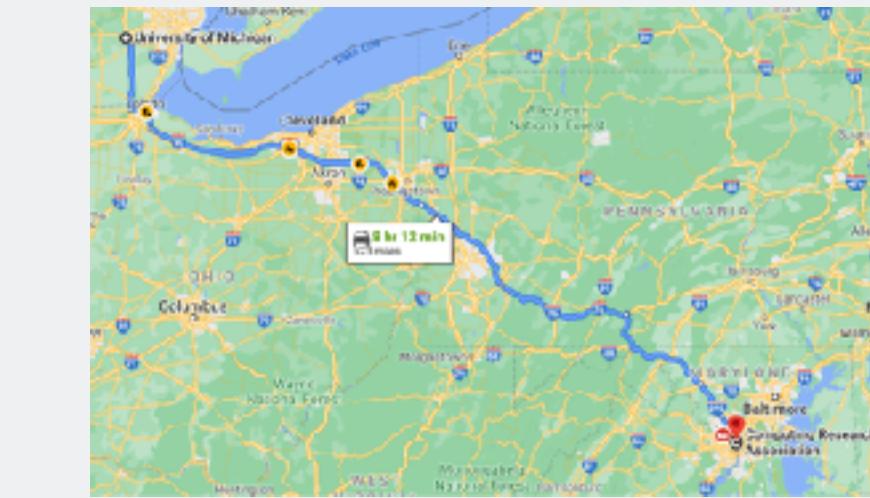
City-scale 3D mapping



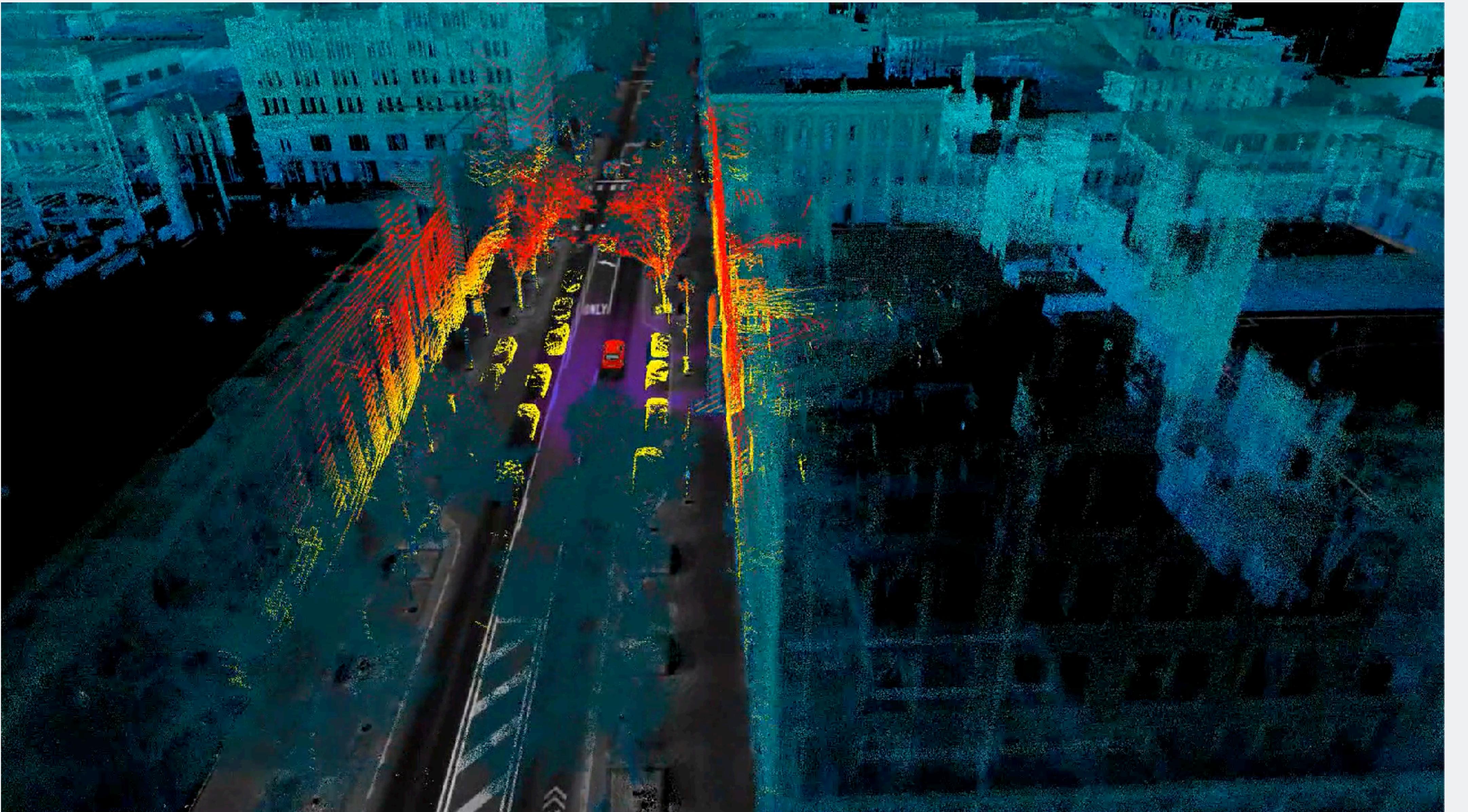
Self-driving cars



Online route navigation



DR

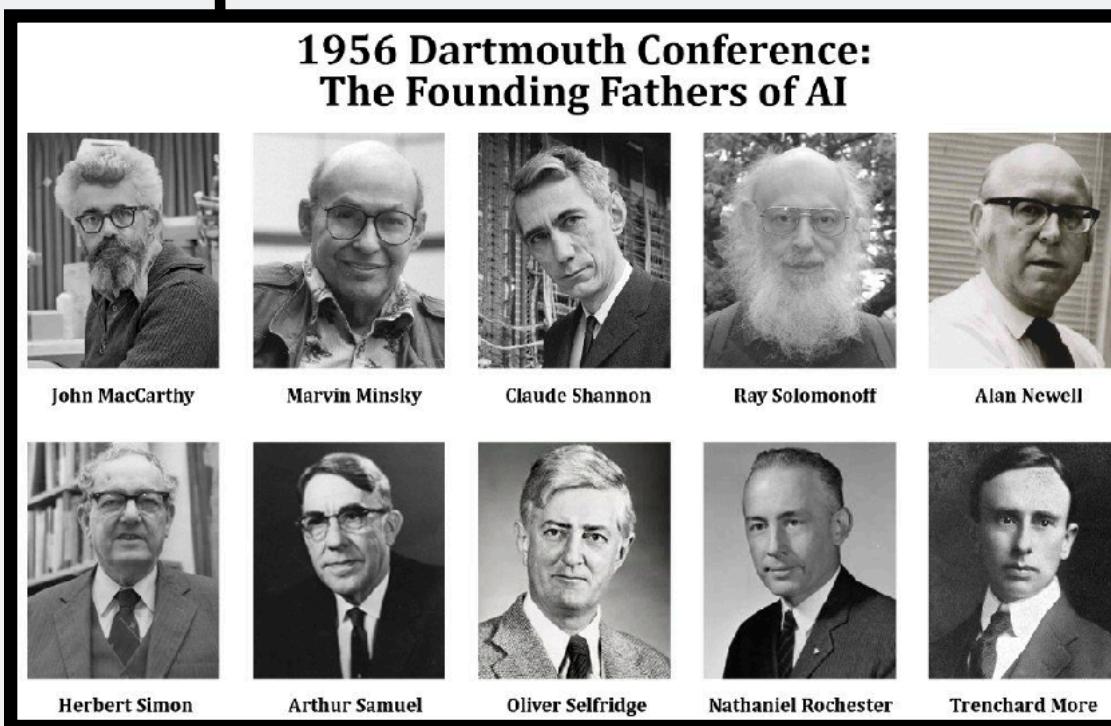


Let's watch this video closely



First wave AI: Model-based

“Think through the entire problem”

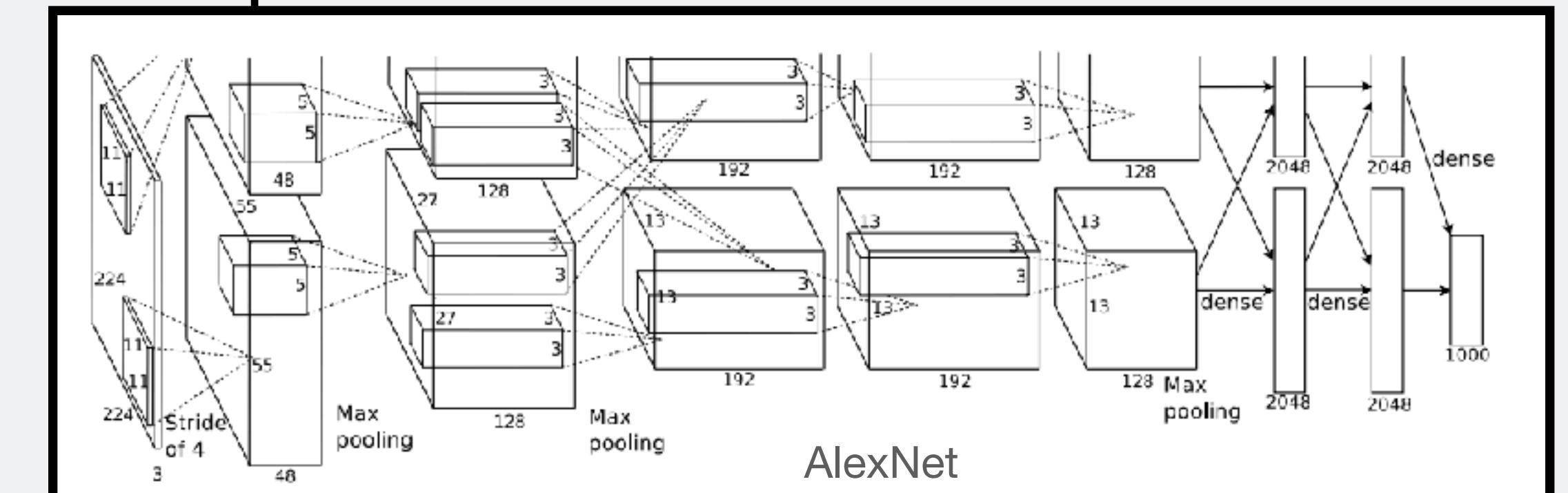


1956

Second wave AI: Data-driven

“Learn from lots of data”

Rise of “deep learning”



2011

Time



First wave AI: Model-based

The screenshot shows a news article from IEEE Spectrum. The title is "Deep Learning Makes Driverless Cars Better at Spotting Pedestrians". Below the title is a subtitle: "Pedestrian detection systems for cars could become faster and more accurate with help from deep learning algorithms". The author is listed as "By Jeremy Hsu". Below the author's name are two images showing pedestrian detection results. The first image is a street scene with several people and a bus, with red boxes highlighting detected pedestrians. The second image is a street scene with cars and a person, also with red boxes highlighting detected objects. At the bottom of the article is a credit: "Images: Statistical Visual Computing Lab/UC San Diego". The navigation bar at the top includes links for Engineering Topics, Special Reports, Blogs, Multimedia, The Magazine, Professional Resources, and Search.

1956



Second wave AI: Data-driven

"Learn from lots of data"



"deep learning"

2011

Time



Second wave AI: Data-driven

TEXT PROMPT
an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT PROMPT
an armchair in the shape of an avocado....

AI-GENERATED IMAGES

Edit prompt or view more images↓

DALL-E (OpenAI)

“Learn from lots of data”



“deep learning”

2011

Time





TEXT PROMPT
an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

[Start my free trial >](#) [Explore docs](#)

TEXT PROMPT
an armchair in the shape of a green pear

AI-GENERATED IMAGES

```
ts sentiments ts-write_sql.go parse_expenses.py addressscr
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

DALL-E (OpenAI)

Copilot (OpenAI)

Second wave AI: Data-driven

“Learn from lots of data”



“deep learning”

2011

Time





Second wave AI: Data-driven

TEXT PROMPT
an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in

In the following sample, ChatGPT is able to understand the reference ("it") to the subject of the previous question ("fermat's little theorem").

Sample 3 of 4 Previous Next

User what is the fermat's little theorem

ChatGPT Fermat's Little Theorem states that if p is a prime number and a is an integer not divisible by p , then a^p (a to the power p) is congruent to a modulo p . In other words, if we divide a^p by p , the remainder is always a .

For example, if $p = 5$ and $a = 2$, then $2^5 = 32$, and 32 divided by 5 has a remainder of 2. Therefore, 2^5 is congruent to 2 modulo 5, and Fermat's Little Theorem holds for this case.

Fermat's Little Theorem is often used in cryptography and other applications where it is necessary to perform modular arithmetic operations quickly and efficiently. It is also a useful tool for proving other theorems in number theory

User how is it used in cryptography?

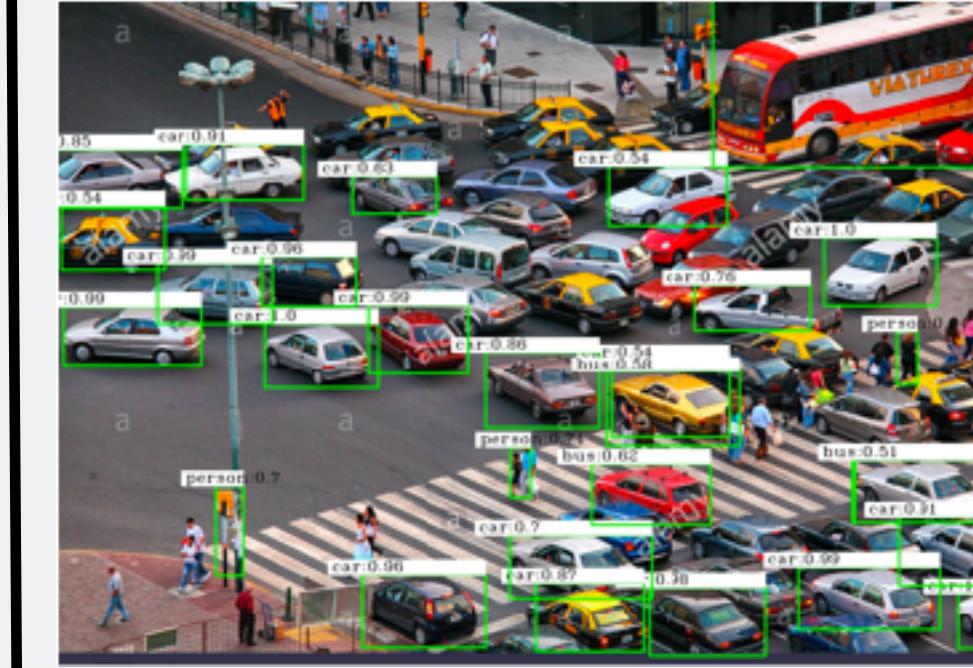
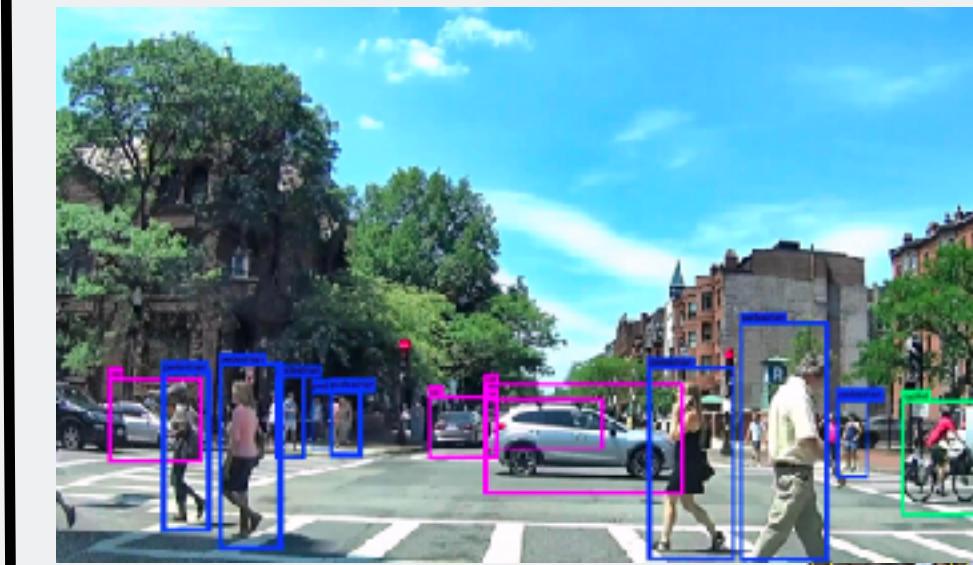
ChatGPT Fermat's Little Theorem is used in cryptography in several ways. One of the most common applications is in the generation of so-called "public-key" cryptography systems, which are used to securely transmit messages over the Internet and other networks.

In a public-key cryptography system, each user has a pair of keys: a public key, which is widely known and can be used by anyone to encrypt a message intended for that user, and a private key, which is known only to the user and is used to decrypt messages that have been encrypted with the corresponding public key.

DALL-E (OpenAI)

Copilot (OpenAI)

ChatGPT (OpenAI)



“deep learning”

2011

Time





The AI of today needed decades of investment and research



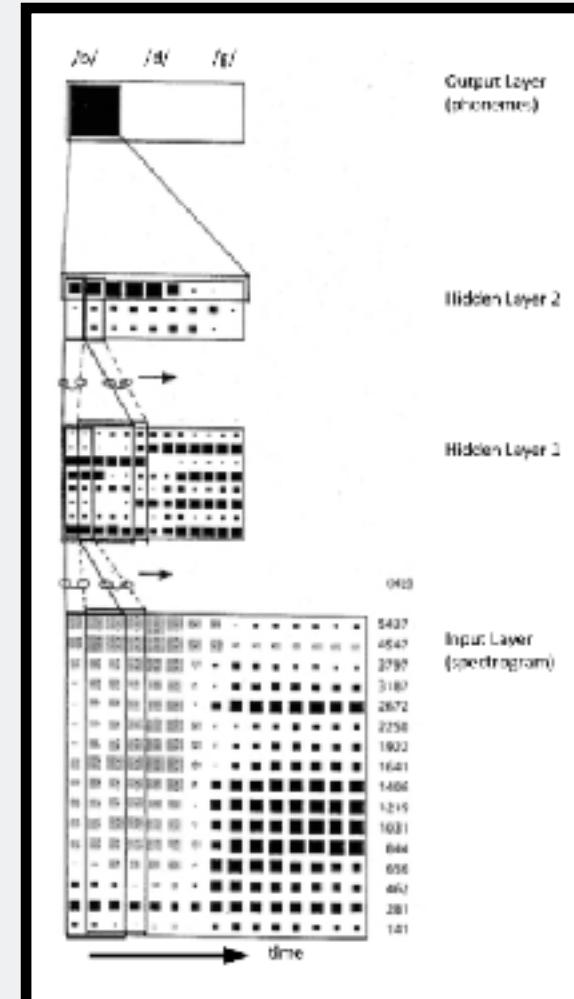
Rosenblatt's Perceptron

1956

1958



Speech recognition by neural networks

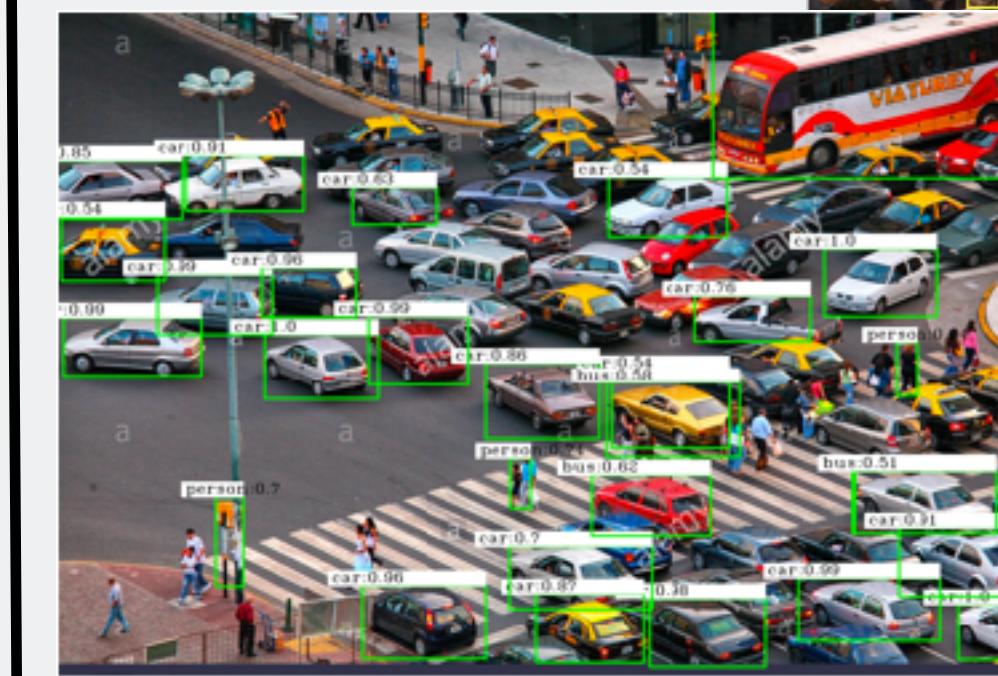


Waibel et al.

1989

Second wave AI: Data-driven

"Learn from lots of data"



"deep learning"

2011

Time





What is this?



Search

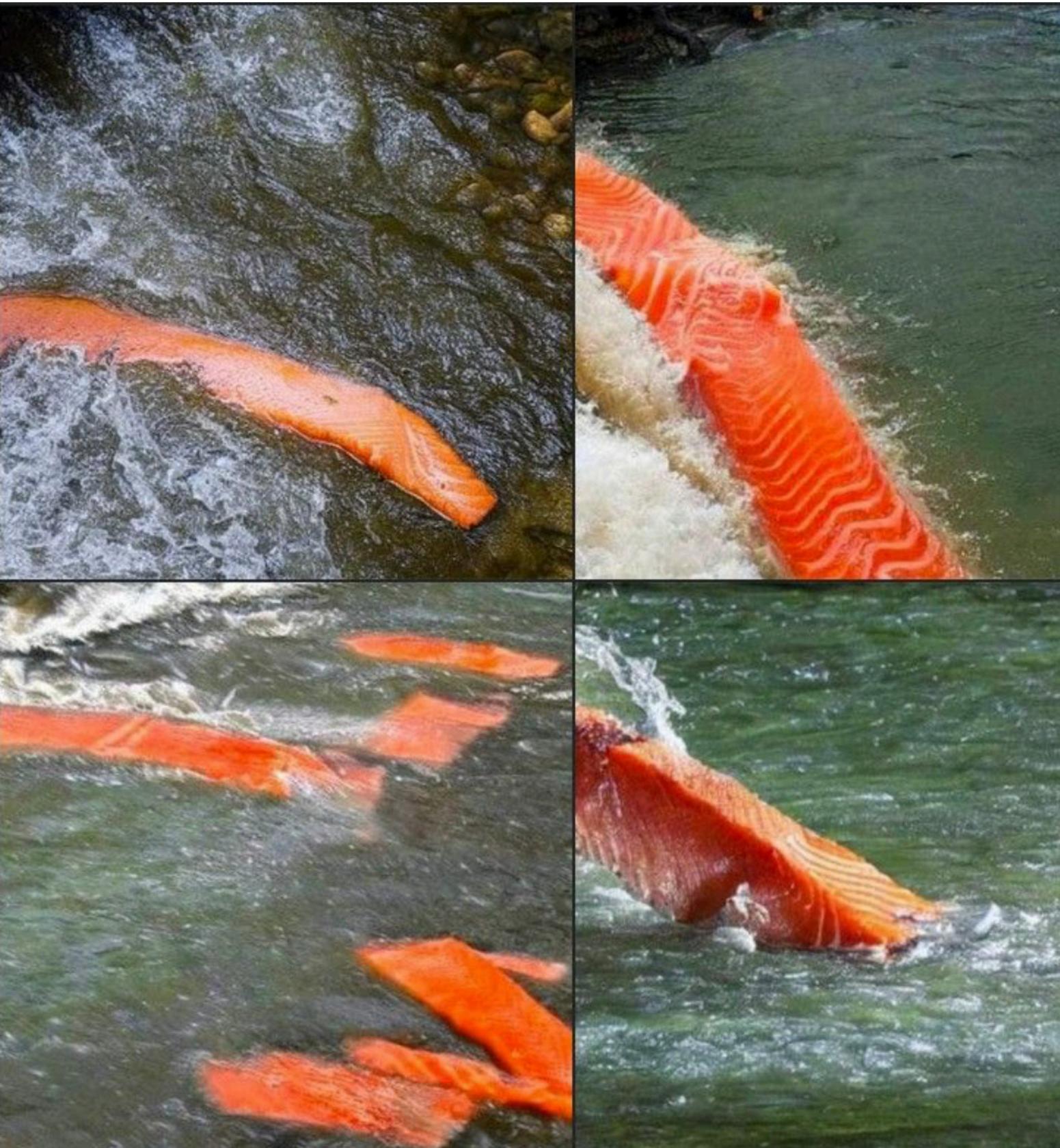
Carlos la Orden Tovar · 3rd+
Trainer and speaker at international events in
the language industry
2d · 0

+ Follow

They asked an AI engine to recreate "a salmon swimming down a river", and here's what it guessed it would look like.

It's exactly the same thing that happens to your texts when you leave them in the hands of a raw machine translation engine and fail to plan a proper post-editing, proofreading and—above all—put a discerning human eye in the loop to look for context and read between the lines.

#IA #MT #MachineTranslation



Be mindful about how
you use AI

“a salmon swimming
down a river”





Posted by u/Callec254 25 days ago

4 Fluff

I asked the new Open AI GPT chat bot for advice...

How do you counter mass Protoss air as a Terran player in Starcraft II?

As a Terran player facing a mass Protoss air army in Starcraft II, one potential counter strategy is to use missile turrets. These turrets can be built on the Terran player's own base, and are effective at shooting down enemy air units.

In addition to missile turrets, the Terran player can also use ground units such as marines, marauders, and thors to attack the Protoss air units. These units are effective at dealing damage to air units, and can be used to weaken the enemy army before it reaches the Terran player's base.

Another potential counter strategy is to use Vikings, which are Terran air units that are specifically designed to counter other air units. These units can be used to attack the Protoss air army directly, or to provide support for other ground units.

Overall, the key to effectively countering a mass Protoss air army as a Terran player is to use a combination of defensive structures, ground units, and air units to weaken the enemy army and protect the Terran player's base.

I mean... it's not *wrong*...

About Community

r/starcraft

All about the StarCraft games and professional scenes surrounding them. Please read the rules before submitting content.

Created Dec 10, 2008

291k Members • 324 Online

[Join](#)

meta_subliminal · 25 days ago

Well it's is wrong about the marauder part. It sounds like an elementary student being introduced to the 5 paragraph essay structure haha

6 Reply Share Report Save Follow

jiraiyaperv · 24 days ago

play like maru

1 Reply Share Report Save Follow

Comment removed by moderator · 21 days ago





First wave AI: Model-based

The Economist Subscribe

Menu Weekly edition Search

Technology Quarterly Automobiles

Jun 13th 2020 edition >

Driverless cars show the limits of today's AI

They, and many other such systems, still struggle to handle the unexpected

A cartoon illustration by Tom Gauld shows a blue driverless car with a person inside looking at a tablet. Above the car are three white boxes with question marks. The car is approaching a road sign that has been spray-painted with a large yellow 'Z'. The background is a pinkish-purple gradient.

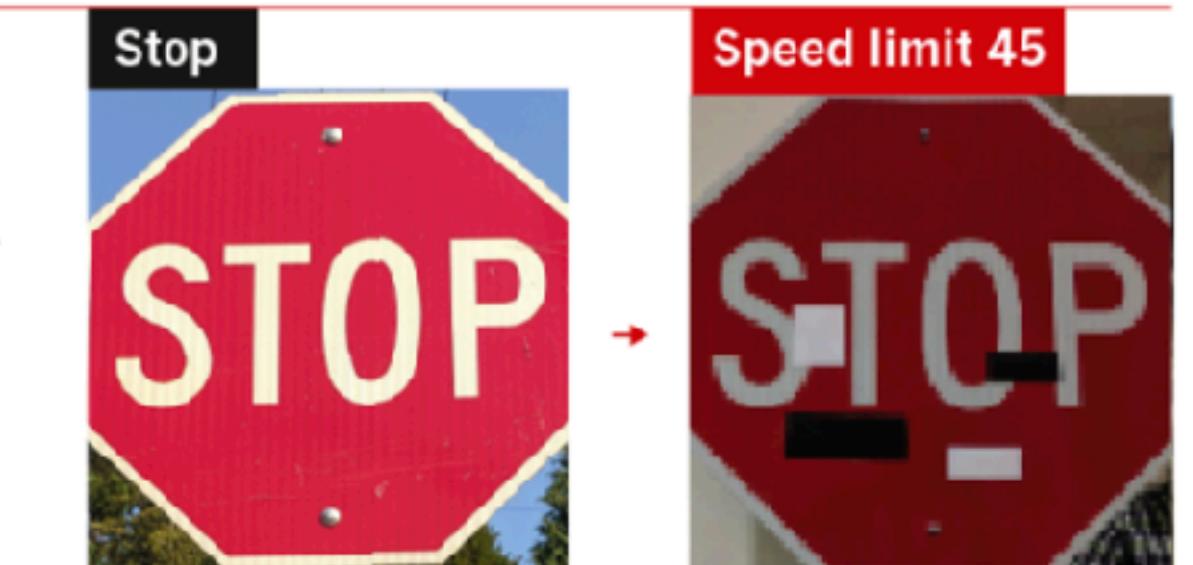
Second wave AI: Data-driven

“Learn from lots of data”

FOOLING THE AI

Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

These stickers made an artificial-intelligence system read this stop sign as 'speed limit 45'.



1956

2011

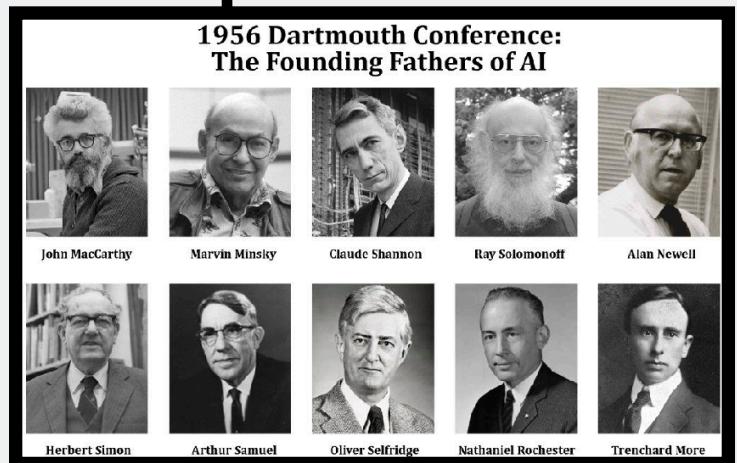
Time





First wave AI: Model-based

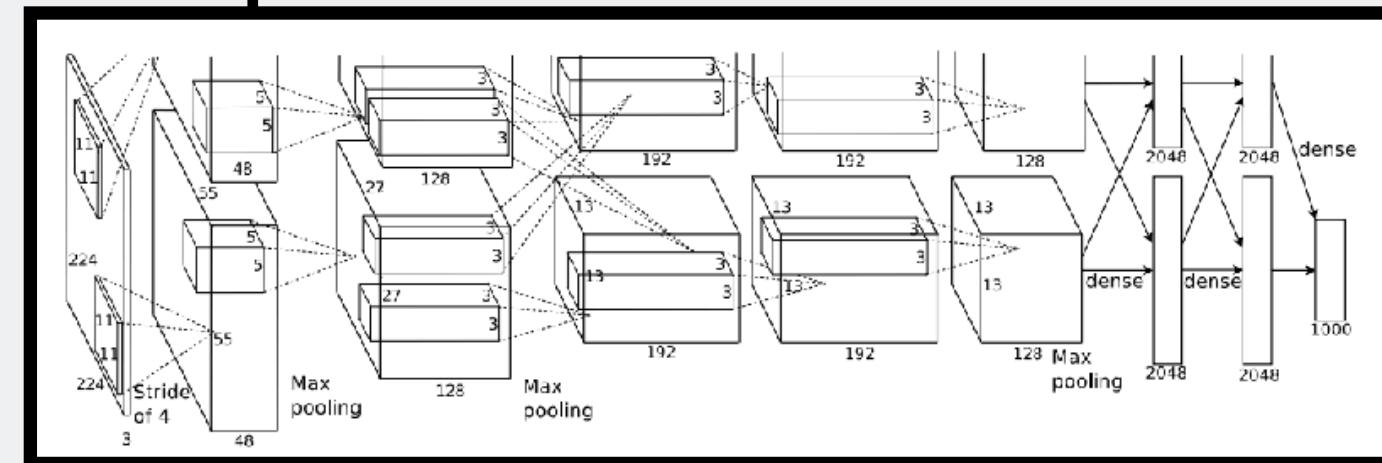
“Think through the entire problem”



1956

Second wave AI: Data-driven

“Learn from lots of data”



2011

Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



20??

Time

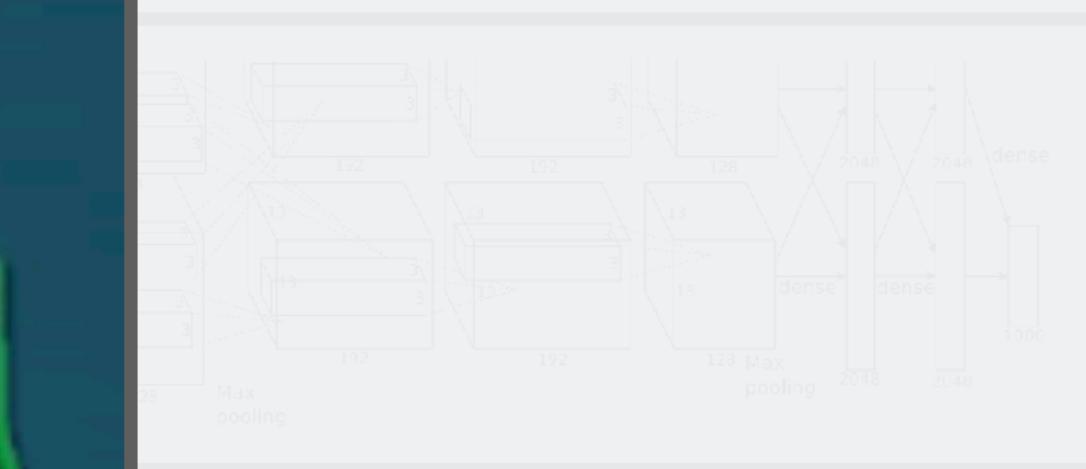




First wave AI: Model-based

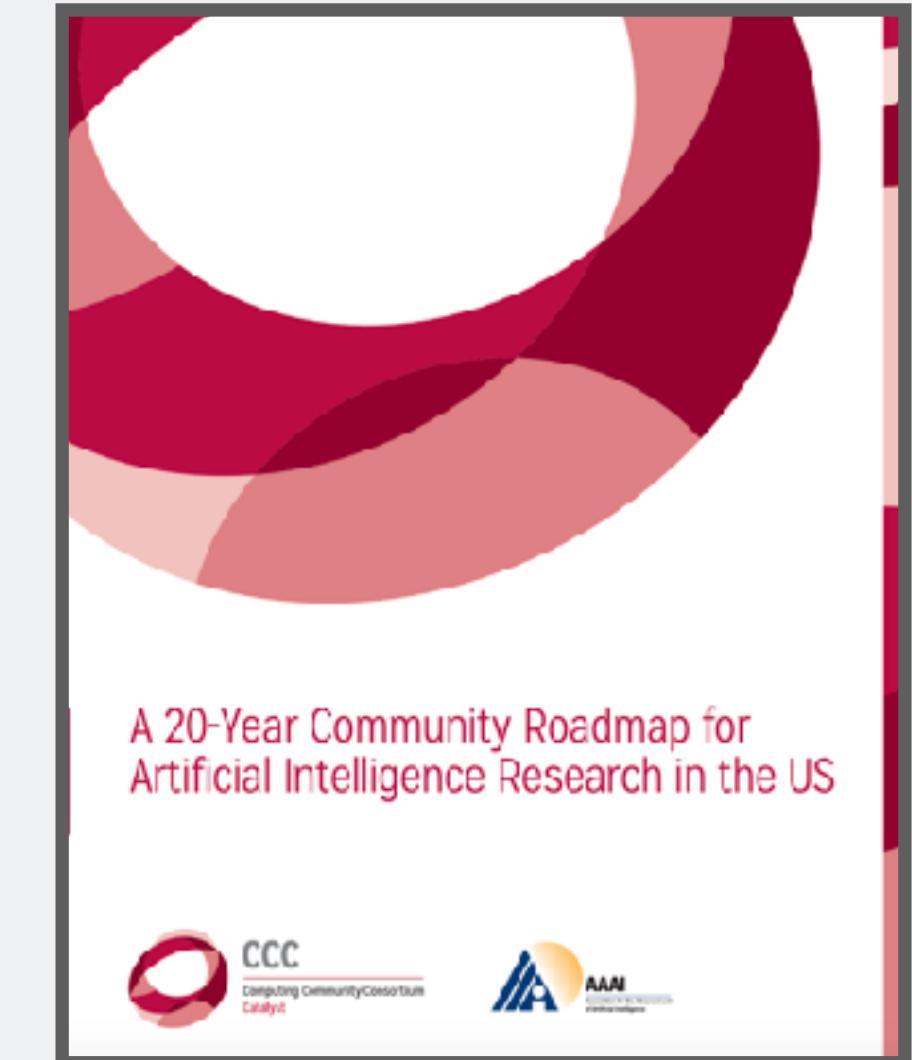
“Think through the entire problem”

Is this handwritten character a ‘9’ or a ‘4’?



Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



1956 2011 20?? Time



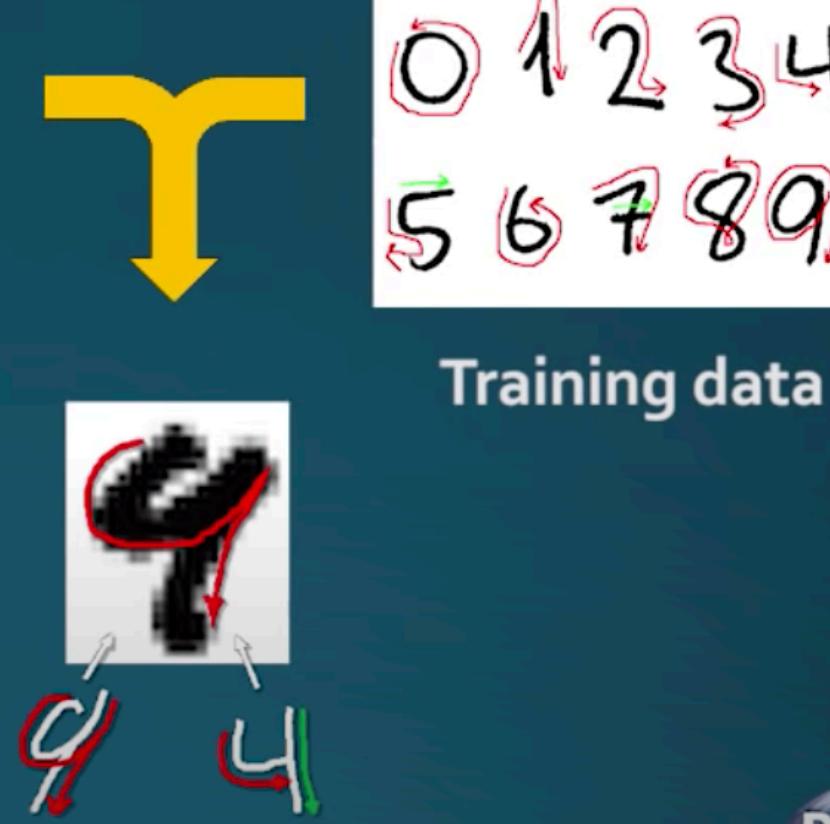


Models to drive decisions

Probable number of strokes: 1 - 4
Each stroke: probable trajectory
Each trajectory: probable shift in shape and location

Seed model

Generative model
Generates explanations of how a test character might have been created



▶ ▶ 🔍 14:54 / 16:11

▶ CC ⚙ □ □

A DARPA Perspective on Artificial Intelligence

241,472 views • Feb 15, 2017

5.5K 130 SHARE SAVE ...

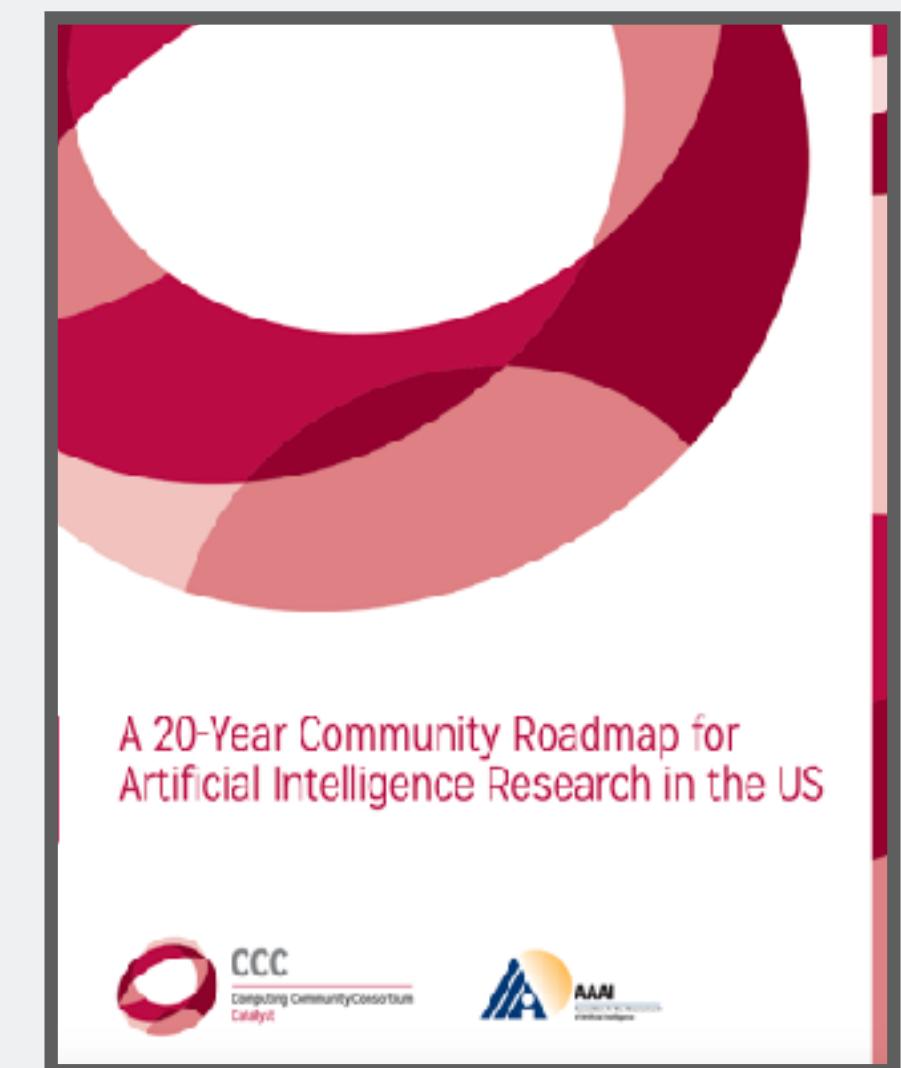


DARPAtv ✓
143K subscribers

SUBSCRIBE

Third wave AI: Explainable

"Combine first and second wave AI to generate explanations"



1956

2011

20??

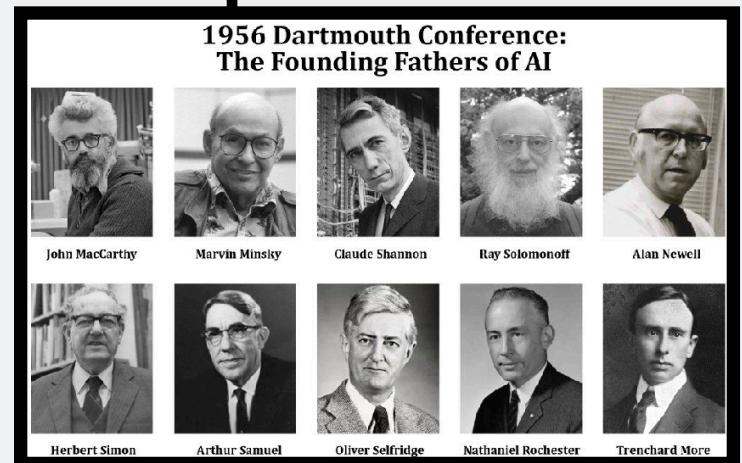
Time





First wave AI: Model-based

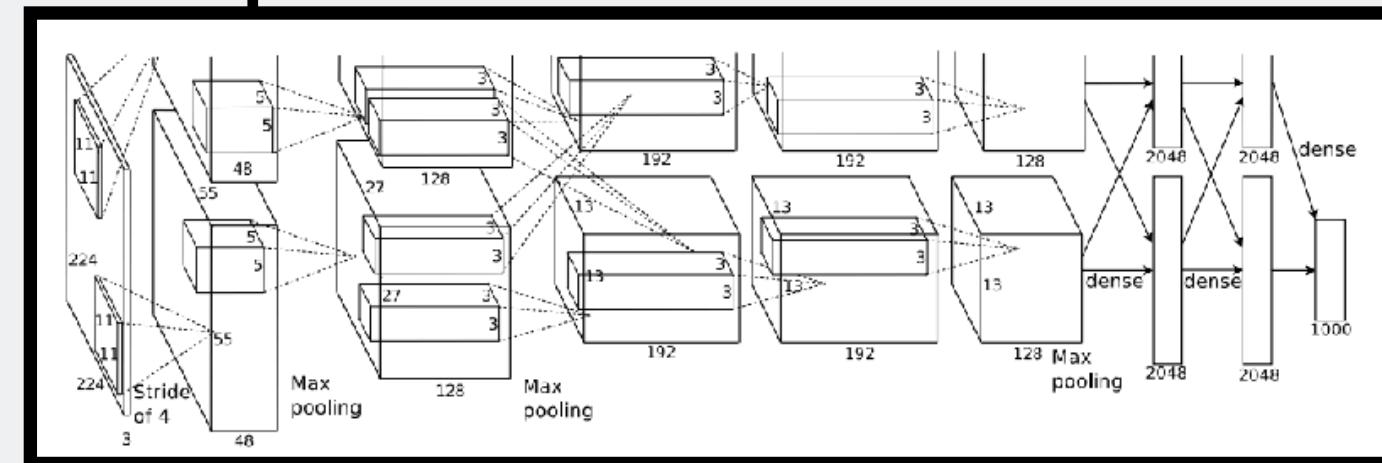
“Think through the entire problem”



1956

Second wave AI: Data-driven

“Learn from lots of data”



2011

Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



20??

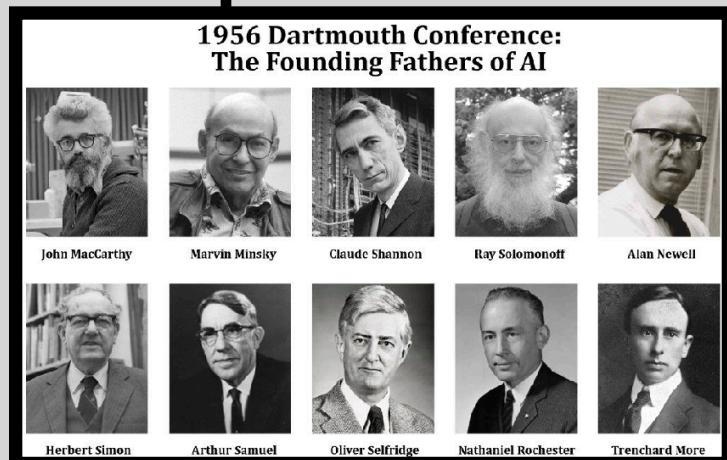
Time



Other Robotics and AI courses

First wave AI: Model-based

“Think through the entire problem”

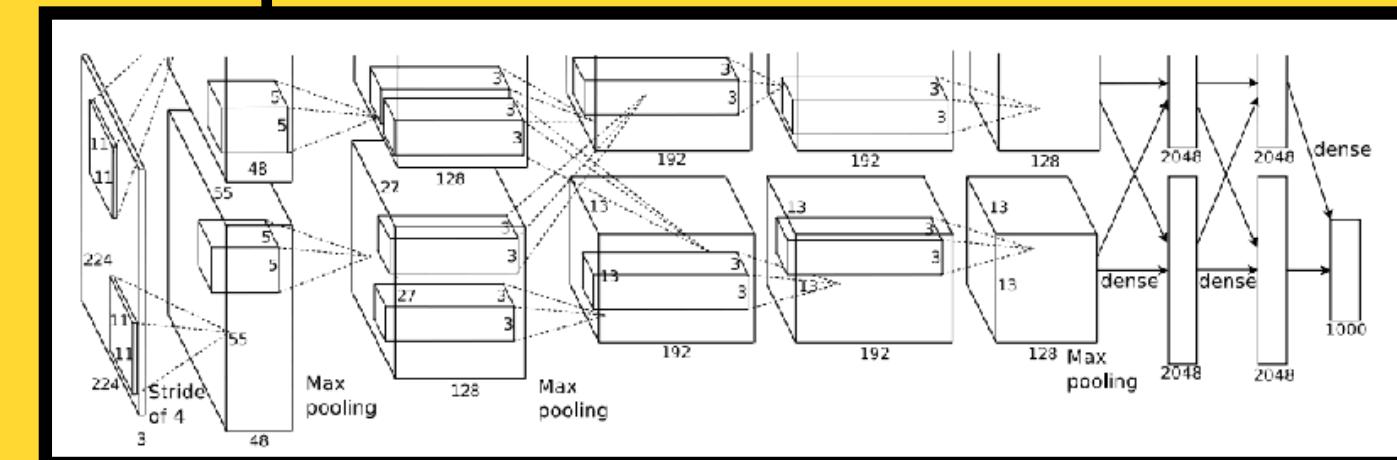


1956

DeepRob is a step into modern robot learning

Second wave AI: Data-driven

“Learn from lots of data”

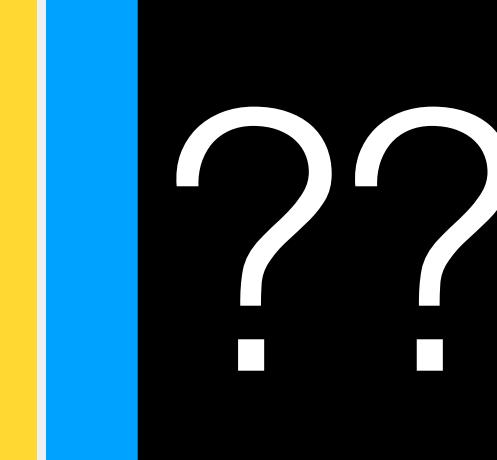
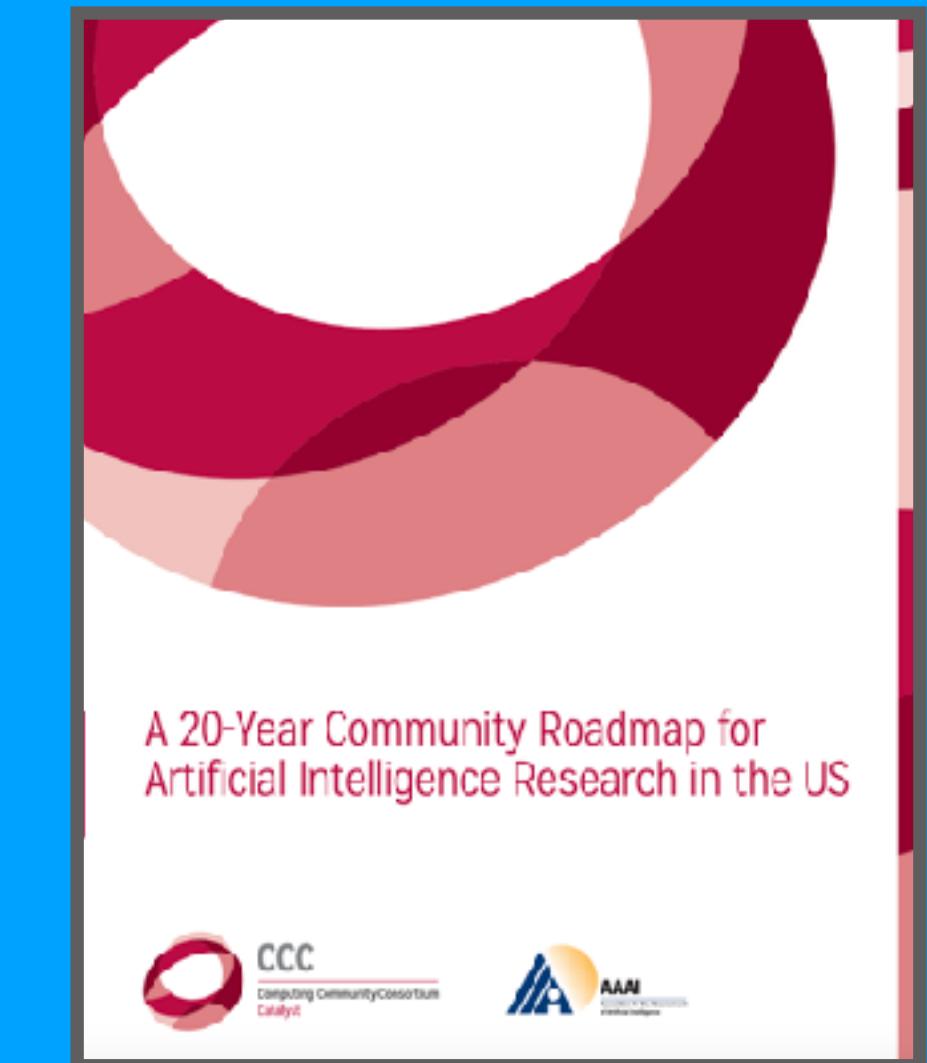


2011

Research for future AI

Third wave AI: Explainable

“Combine first and second wave AI to generate explanations”



20??

Time



Course Resources

- Course Website: <https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/>
 - Syllabus, calendar, project files, slides, links, etc.
- Ed Stem: <https://edstem.org/us/courses/31008/discussion/>
 - Forum for communication and question answering
 - Optional to join and shared across course offerings





Course Website:

<https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/>

The screenshot shows a web browser window displaying the course website. The URL in the address bar is <https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/>. The page title is "Deep Learning for Robot Perception and Manipulation". The header includes the "Deep Rob" logo, a search bar, and links for "Forum" and "Gradescope". The Minnesota Robotics Institute logo is also present. The left sidebar contains links for "Home", "Syllabus", "Calendar", "Projects", "Papers", and "Staff". The main content area describes the course as CSCI5980-02 Spring 2023 at The University of Minnesota - Twin Cities, meeting Tu, Th 2:30PM-3:45PM CT - Rapson Hall 58. It details the course's focus on neural-network-based deep learning for robot perception and manipulation, mentioning the exploration of recent research publications and implementation of projects. A footer note states the site uses "Just the Docs" theme for Jekyll. The bottom right corner shows the University of Minnesota "M" logo.

DR Deep Rob

Search Deep Rob

Forum Gradescope

MINNESOTA ROBOTICS INSTITUTE
UNIVERSITY OF MINNESOTA

Home Syllabus Calendar Projects Papers Staff

Deep Learning for Robot Perception and Manipulation

CSCI5980-02 Spring 2023 at The University of Minnesota - Twin Cities

Tu, Th 2:30PM-3:45PM CT - Rapson Hall 58

This course covers the necessary background of neural-network-based deep learning for robot perception – building on advancements in computer vision that enable robots to physically manipulate objects. During the first part of this course, students will learn to implement, train and debug their own neural networks. During the second part of this course, students will explore recent emerging topics in deep learning for robot perception and manipulation. This exploration will include analysis of research publications in the area, building up to reproducing one of these publications for implementation as a final course project.

This course is being offered through a Distributed Teaching Collaborative between faculty at [the University of Michigan](#) ([Anthony Opipari](#), [Chad Jenkins](#)) and [the University of Minnesota](#) ([Karthik Desingh](#)).

This site uses [Just the Docs](#), a documentation theme for Jekyll.



Meeting Logistics

- In-person Lectures
 - Tue & Th 2:30-3:45 PM CT - Rapson Hall 58
 - Zoom link access will be made under special circumstances per lecture basis. Request by emailing to the instructor.
- Office Hours
 - Wed, 2:00-3:30pm, Fri, 9:00-10:30am, 161 Shepherd Labs & Zoom (Meeting ID: 942 1850 7156)





Office Hours Queue

The screenshot illustrates the process of managing an office hours queue. It shows a web browser window with the URL rpm-lab.github.io/CSCI5980-Spr23-DeepRob/. The page features a yellow header with the "Deep Rob" logo. Below the header are navigation links for "Forum" and "Gradescope". Two buttons, "Join OH-Q" and "View OH-Q", are highlighted with red boxes and arrows pointing to their corresponding functions.

CSCI5980 OH Queuing Form

Current Queue

Switch account

Full name *

Karthik

Email *

dfafad

Mode of communication *

In person @ 161 Shepherd

Zoom (Meeting ID: 942 1850 7156, Passcode: deeprob23)

What do you want to discuss? *

adfad

Submit Clear form

CSCI5980 OH Queuing Form (Responses)

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

	A	B	C	D	E
1	Timestamp	Full name	Email	Mode of communication	What do you want to discuss?
2	1/17/2023 6:46:45	First Last	firstlast@umn.edu	In person @ 161 Shepherd	Project
3	1/17/2023 6:47:10	First last	some@umn.edu	Zoom (Meeting ID: 942 1850 7156, Passcode: deeprob23)	Presentation
4	1/17/2023 6:47:32	First last	someone@umn.edu	In person @ 161 Shepherd	Course content
5					
6					
7					
8					

M M

Course Structure

- **Objective:** Give you the computational skills to understand and reproduce emerging applications of deep learning for robot perception
- **Project focused class**
 - 6 total projects: building in complexity from basic linear layers to state-of-the-art research topics

Course Structure

First half: Fundamentals

- How to implement, train and debug neural networks
- Classifiers, fully-connected networks, convolutional networks
- Weeks 1-7 and Projects 0-2

Second half: Emerging topics

- Seminar style coverage of active research areas in robotics
- 3D perception, pose estimation, tracking, NeRFs, transformers, grasp learning, tactile perception
- Weeks 7-14 and Projects 3 onward

January

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

February

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

March

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

April

Su	Mo	Tu	We	Th	Fr	Sa
			1			
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						



Schedule – <https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/calendar/>

The screenshot shows a calendar page for the Deep Rob course. The left sidebar has links for Home, Syllabus, Calendar (which is selected), Projects, Papers, and Staff. The main content area is titled "Calendar" and shows a weekly schedule:

- Week 1:** Jan 17: LEC 1 Course Introduction (PROJECT 0 OUT)
- Jan 19: LEC 2 Image Classification
- Week 2:** Jan 24: LEC 3 Linear Classifiers (PROJECT 0 DUE, PROJECT 1 OUT)
- Jan 26: LEC 4 Regularization + Optimization
- Week 3:** Jan 31: LEC 5 Neural Networks
- Feb 02: LEC 6 Backpropagation
- Week 4:** Feb 07: LEC 7 Convolutional Neural Networks (PROJECT 1 DUE, PROJECT 2 OUT)
- Feb 09: LEC 8 CNN Architectures

- Lecture slides
- Discussion resources
- Suggested readings
- Project schedule & downloads



Project Topics

- Project 0
 - Introduction to Python, PyTorch and Google Colab
- Project 1
 - Classification using K-Nearest Neighbors and Linear Models
- Project 2
 - Classification using Fully-Connected and Convolutional Neural Networks
- Project 3
 - Detection using convolutional neural networks
- Project 4
 - State estimation with deep networks
- Final Project
 - Published paper review, presentation, reproduction, extension and report

Final Project

- Published paper review, presentation, reproduction, extension and report
- Work as collaborative teams up to 3 people
- Deliverables:
 - Written review of a published paper
 - Paper presentation during lecture
 - Result reproduction as code
 - Network extension as code
 - Project report as <= 2-page paper



Project Grading

- Projects 0-4
 - 2 total late days available
 - 25% daily penalty after deadline and late days
- Final project graded manually by course staff



Overall Grading Policy

- Projects 0-4: 12% each
- Final Project: 24%
 - Written Review: 3%
 - Presentation: 3%
 - Reproduction: 6%
 - Extension: 6%
 - Written report: 6%
- 16 Pre-Lecture Quizzes: 1% each

This will start from Lecture 4. An announcement will be made.



Collaboration Policy

- All work submitted must be your own
 - All code submitted must comply with College of Engineering Honor Code
- No code can be communicated, including verbally
 - Explicit use of external sources must be clearly cited
- Free flow of discussion and ideas is encouraged



Discussion Forum

- Ed Stem available for course discussion and questions
- **Forum is shared across UMich and UMinn students**
 - UMich lectures are 2 weeks ahead of UMinn.
 - Participation and use is not required
 - Opt-in using this Google form
 - **Discussion of quizzes and verbatim code must be private**





Project 0

- Instructions and code available on the website
- Released today: <https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/projects/project0>

<https://rpm-lab.github.io/CSCI5980-Spr23-DeepRob/projects/project0>

- Intro at this week's discussion - link will be posted on Thursdays
- Due next Tuesday, January 24th 11:59 PM CT





Project 0

The screenshot shows a Google Colab notebook titled "pytorch101.ipynb". The left sidebar contains a "Table of contents" with sections like "ROB 498-002/599-009 Project 0-1: PyTorch 101", "Setup Code", "Google Colab Setup", "Introduction", "Python 3", "PyTorch", "Tensor Basics", and various tensor indexing and reshaping operations. The main content area displays the "Setup Code" section, which includes instructions for putting name and UMID, a placeholder for "Your Answer" (Your NAME, #XXXXXXX), and a code cell for running `%load_ext autoreload` and `%autoreload 2`. A note at the bottom states that the autoreload extension is already loaded.

ROB 498-002/599-009 Project 0-1:
PyTorch 101

Setup Code

Google Colab Setup

Introduction

Python 3

Print is a function

Floating point division by default

No xrange

PyTorch

Tensor Basics

Creating and Accessing tensors

Tensor constructors

Datatypes

Tensor indexing

Slice indexing

Integer tensor indexing

Boolean tensor indexing

Reshaping operations

View

ROB 498-002/599-009 Project 0-1: PyTorch 101

Before we start, please put your name and UMID in following format
: Firstname LASTNAME, #00000000 // e.g.) Anthony OPIPARI, #12345678

Your Answer:
Your NAME, #XXXXXXX

Setup Code

Before getting started we need to run some boilerplate code to set up our environment. You'll need to rerun this setup code each time you start the notebook.

First, run this cell load the [autoreload](#) extension. This allows us to edit .py source files, and re-import them into the notebook for a seamless editing and debugging experience.

```
[ ] %load_ext autoreload
%autoreload 2
```

The autoreload extension is already loaded. To reload it, use:
%reload_ext autoreload





Research Topics

RGB-D Architectures

Pointcloud Processing

Object Pose, Geometry, SDF, Implicit Surfaces

Dense object descriptors, Object category-level representations

Recurrent Networks and Object Tracking

Visual Odometry and Localization

Semantic Scene Graphs and Explicit Representations

Neural Radiance Fields and Implicit Representations

Datasets

Self-supervised Learning

Grasp Pose Detection

Tactile Perception for Grasping and Manipulation

Transformer Architectures

More frontiers





Next Lecture: Classification

Image Classification and Linear Classifiers



DR

DeepRob

Spring 2023

Deep Learning for Robot Perception and Manipulation
University of Minnesota (& University of Michigan)

