



# **RBE474X/595-B01-ST: Deep Learning For Perception**

**Class 1: Introduction, Logistics And Sensors**

**Prof. Wei Xiao**

Special thanks to Prof. Nitin Sankit



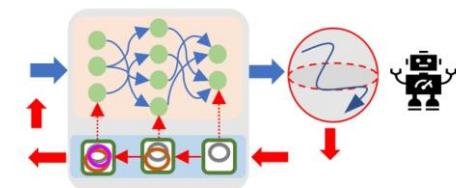


**Prof. Wei Xiao**

*he/him/his*

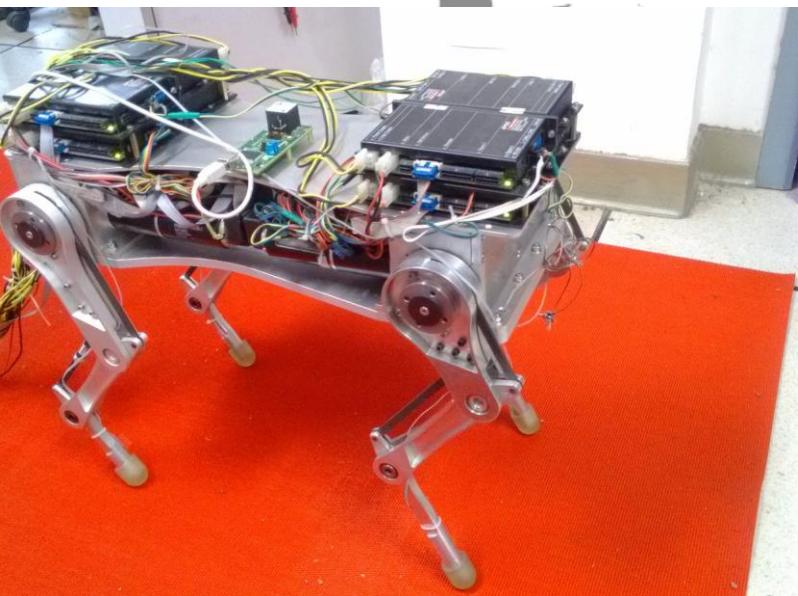


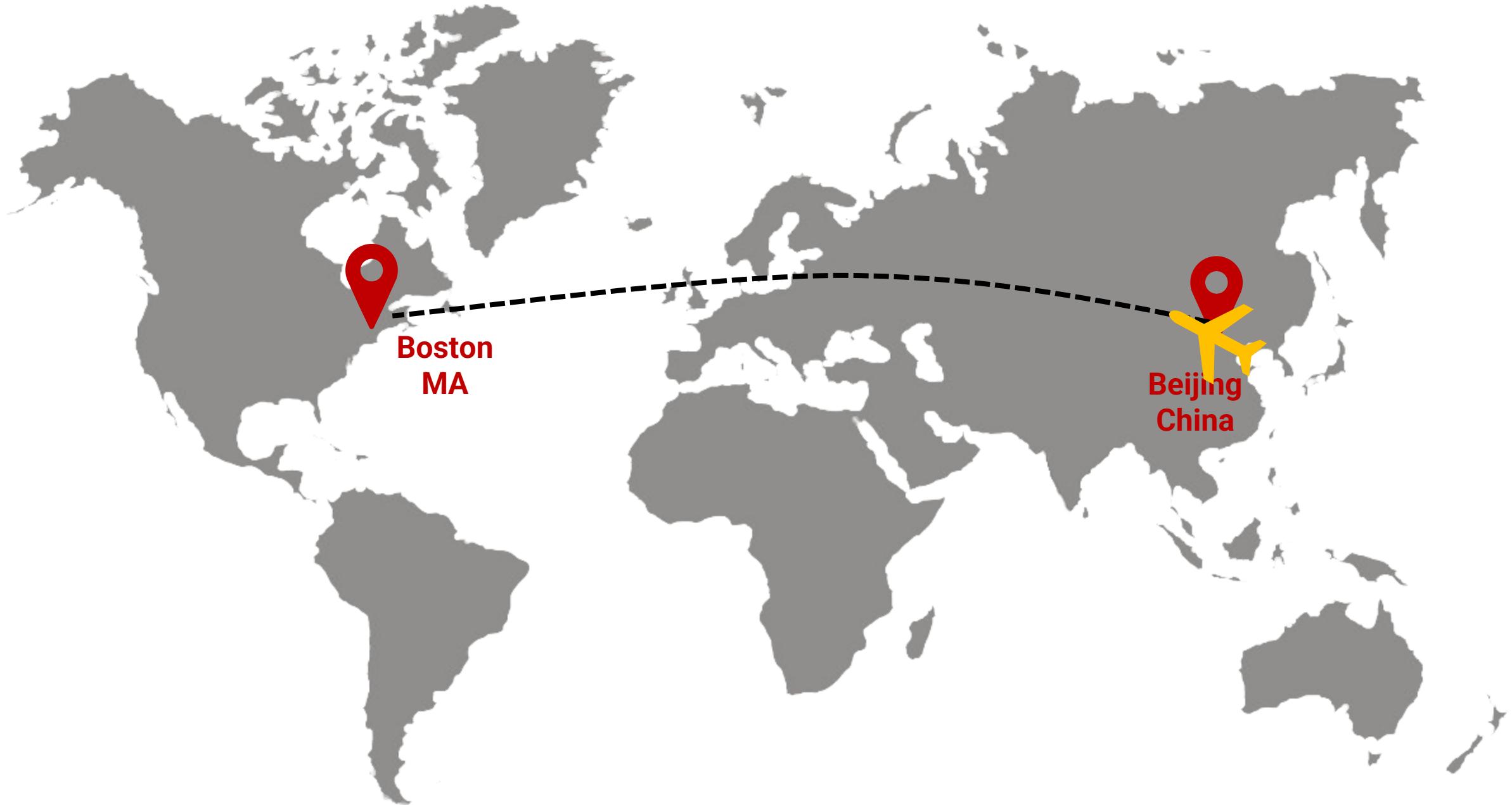
**WPI**



Assistant Professor  
**Safe Autonomy and Intelligence Lab (SAIL)**  
Robotics Engineering

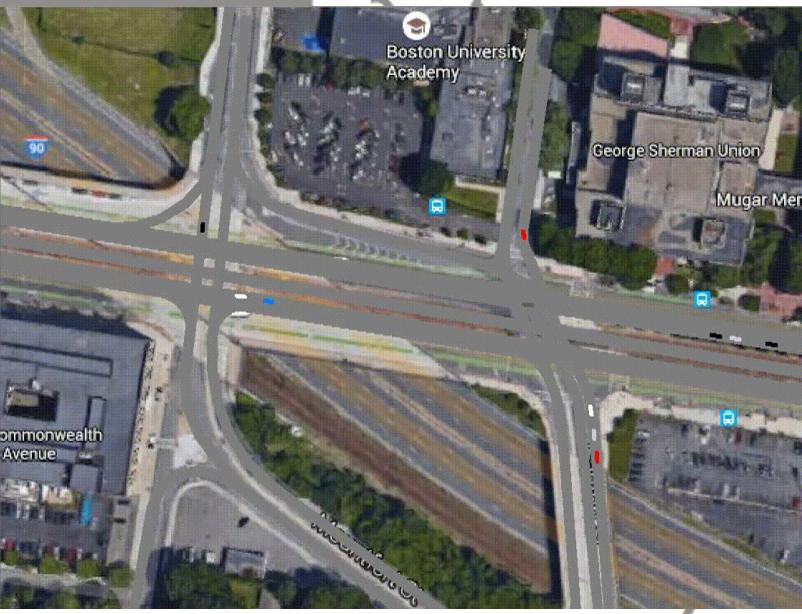
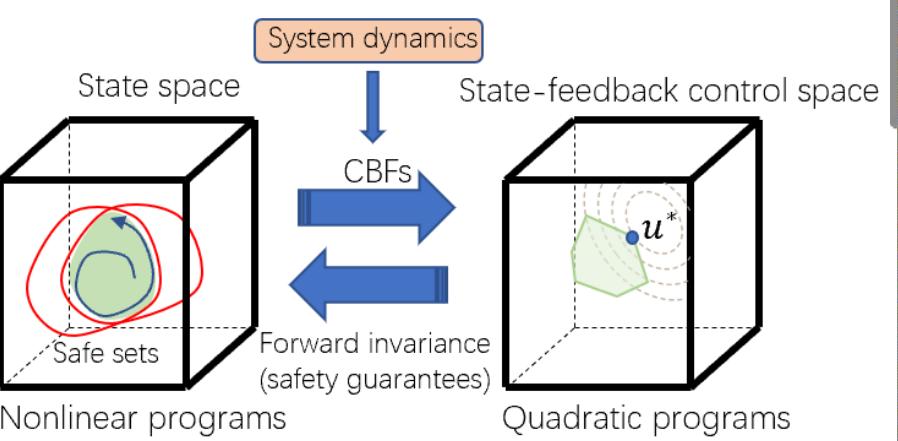
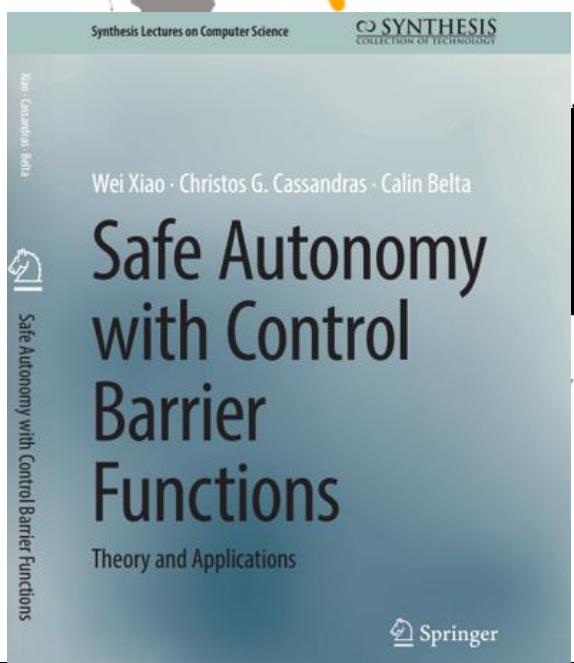
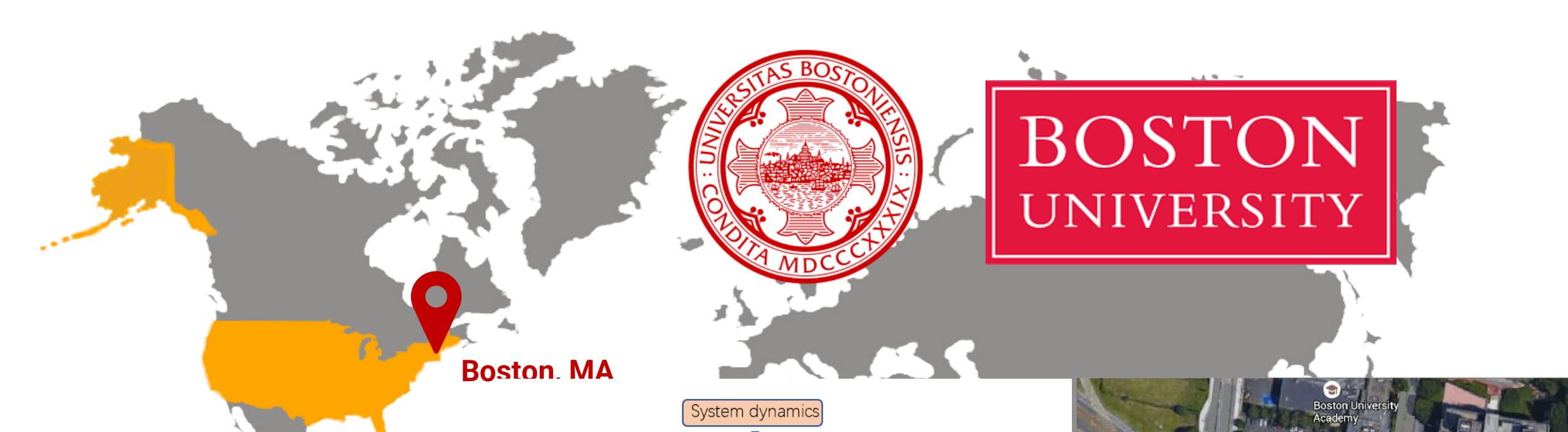






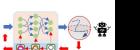
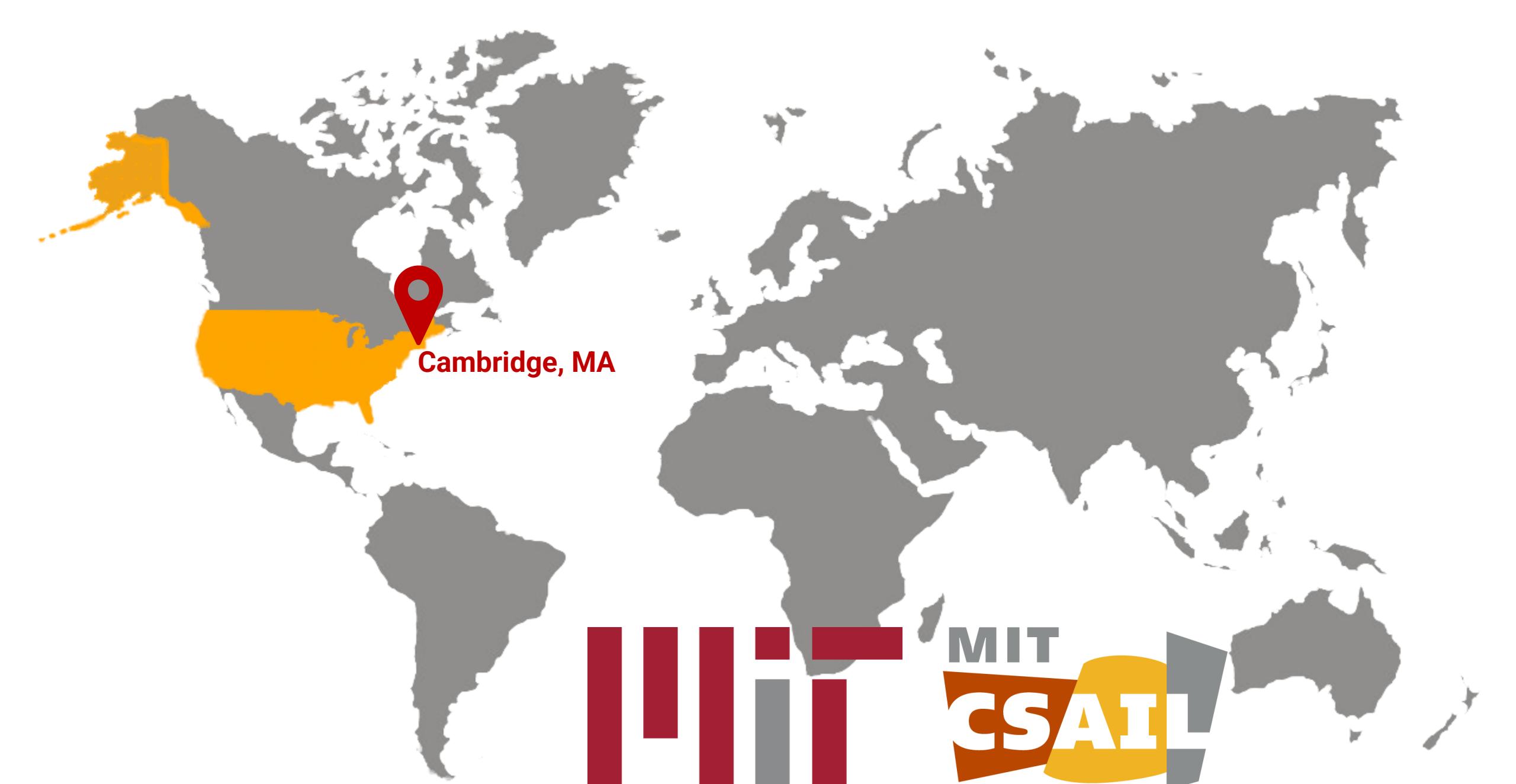
Boston  
MA

Beijing  
China

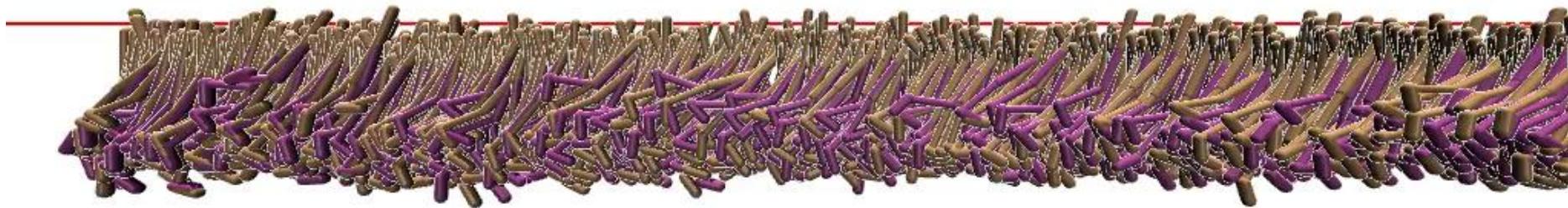
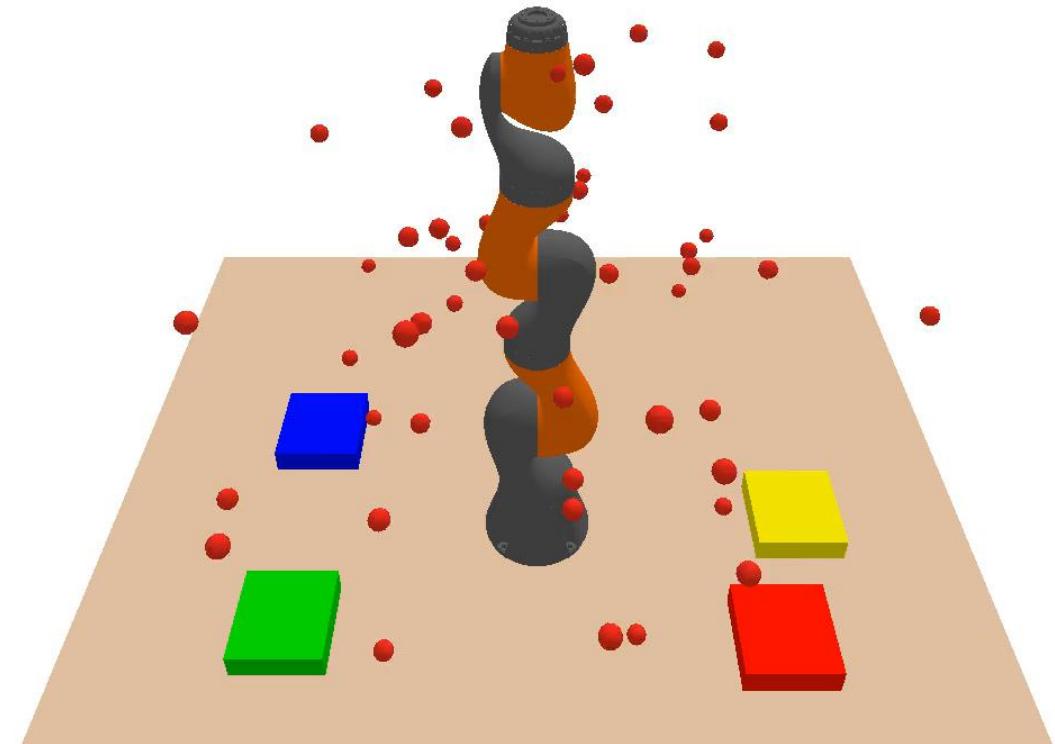
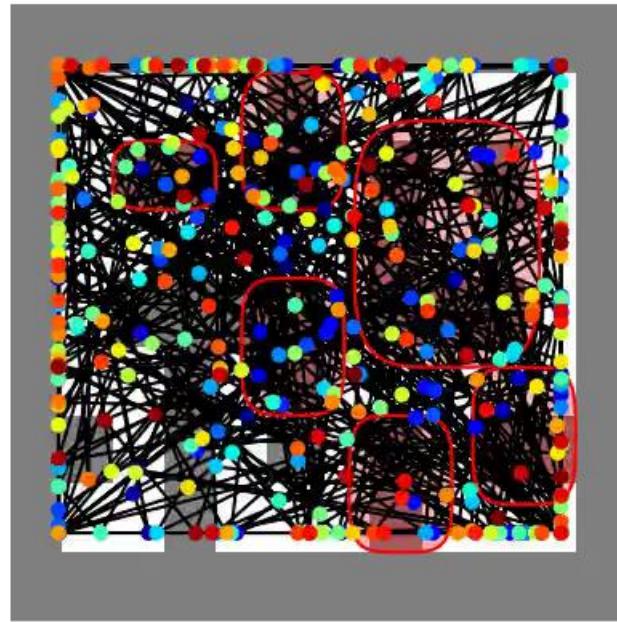


**Boston University Outstanding Dissertation Award  
2020 IEEE-CDC Outstanding Student Paper Award**



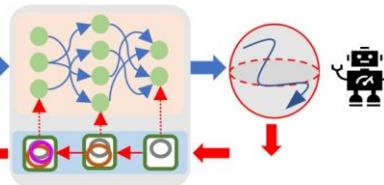








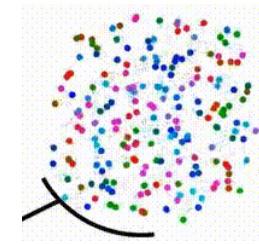
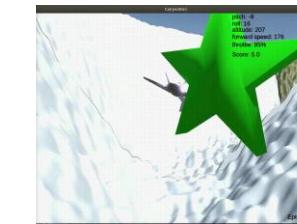
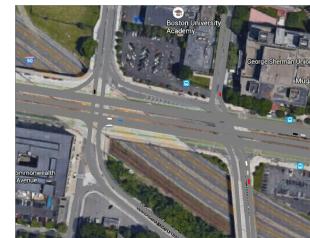
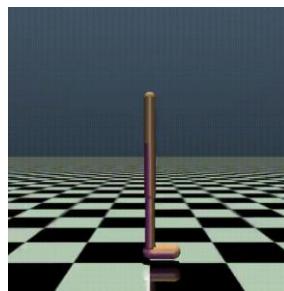
# WPI

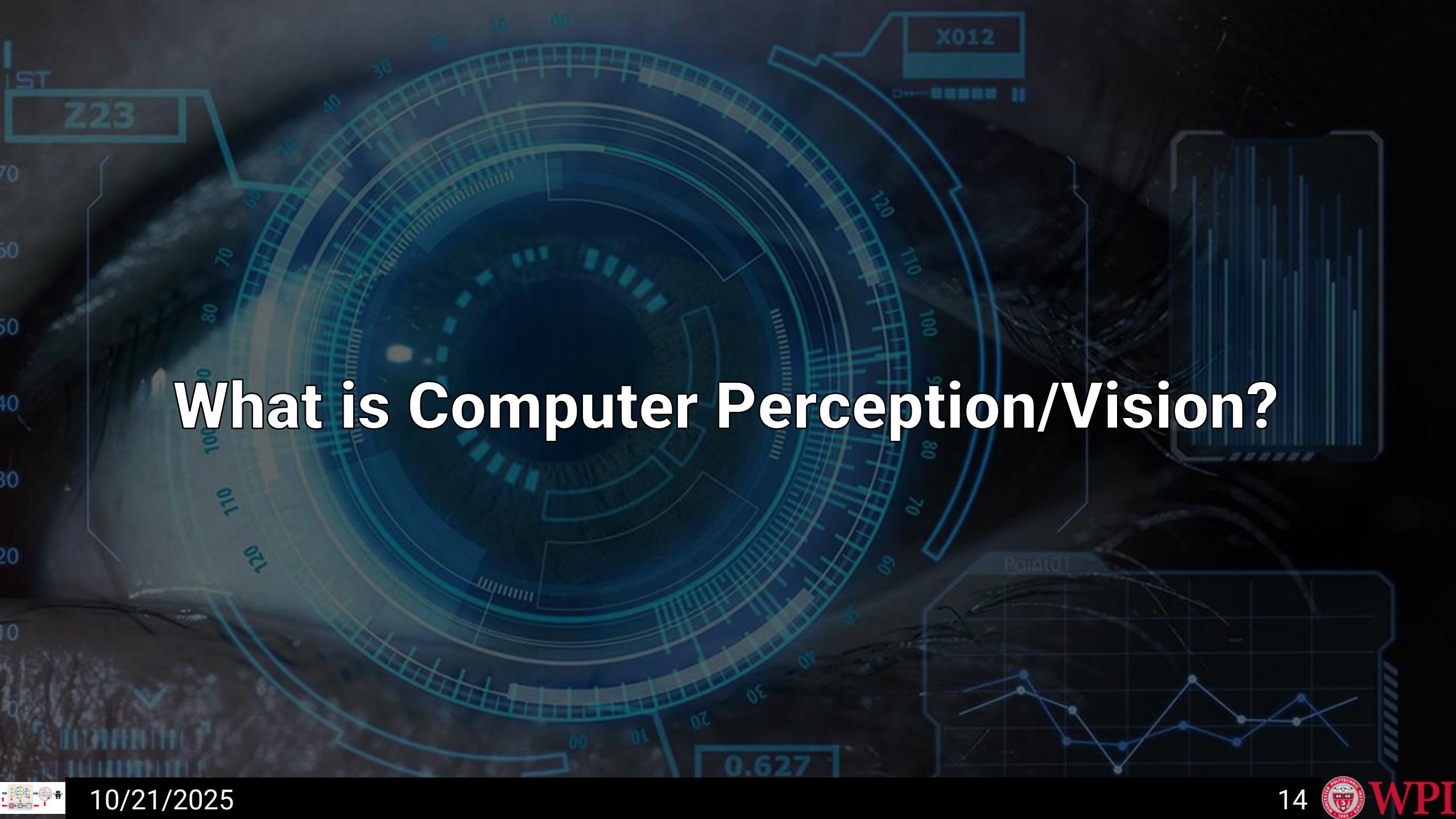


## Assistant Professor

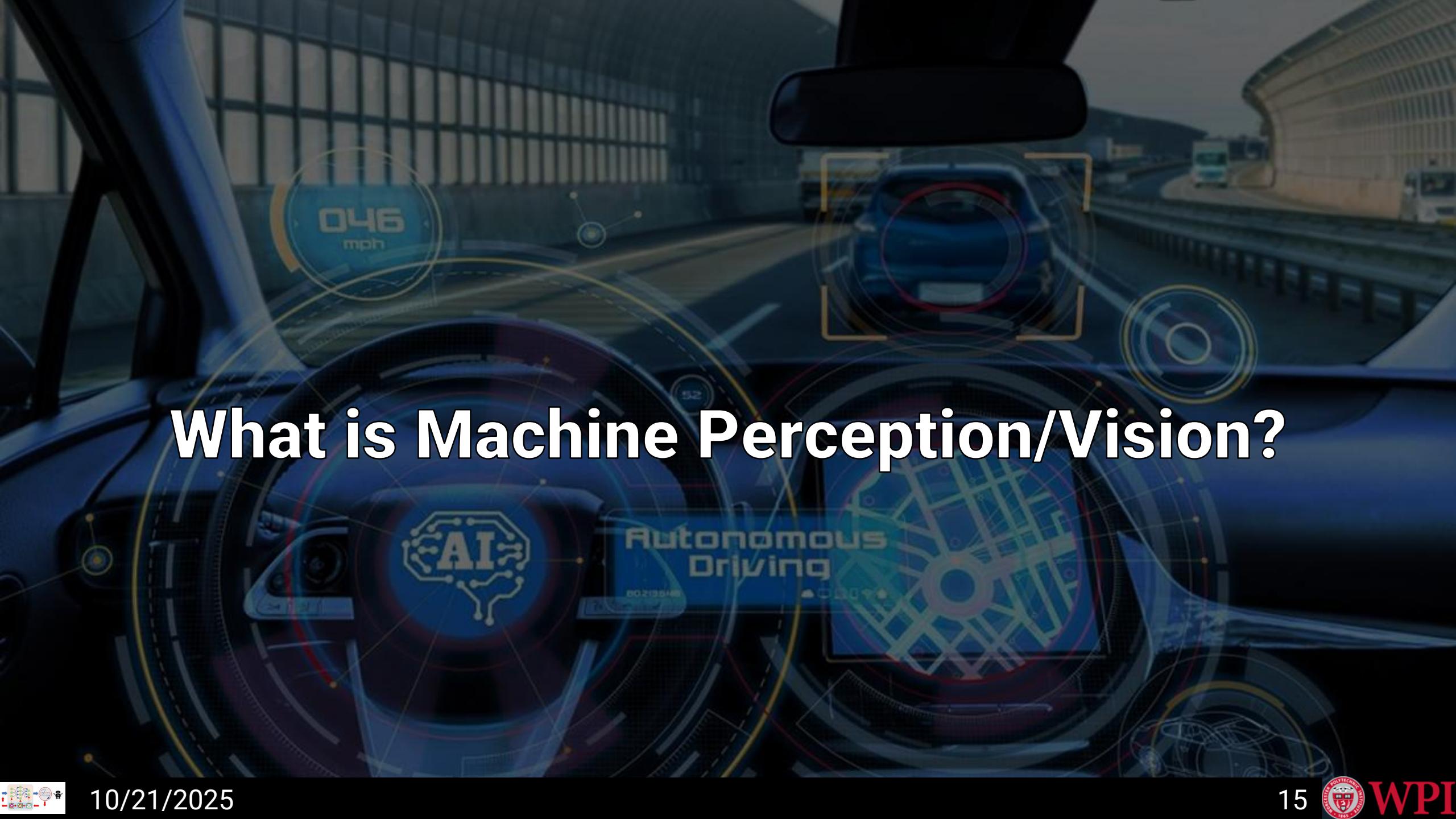


SAIL



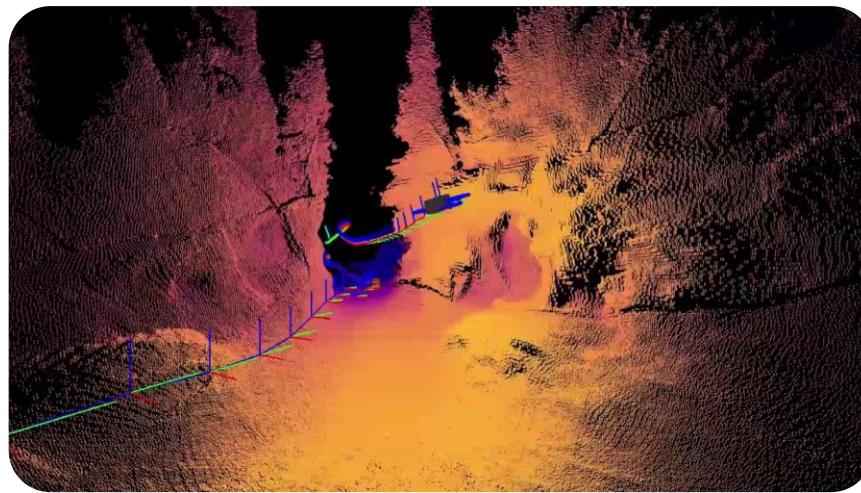
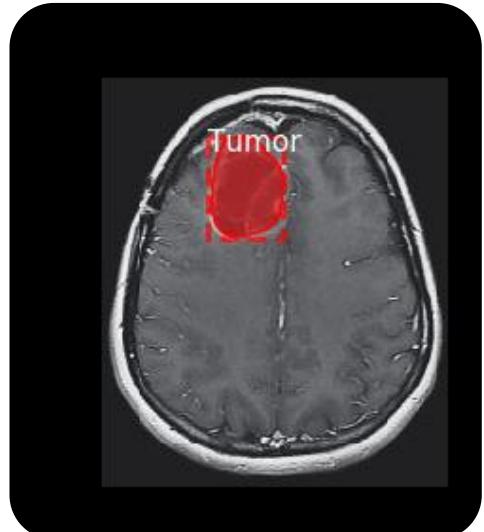
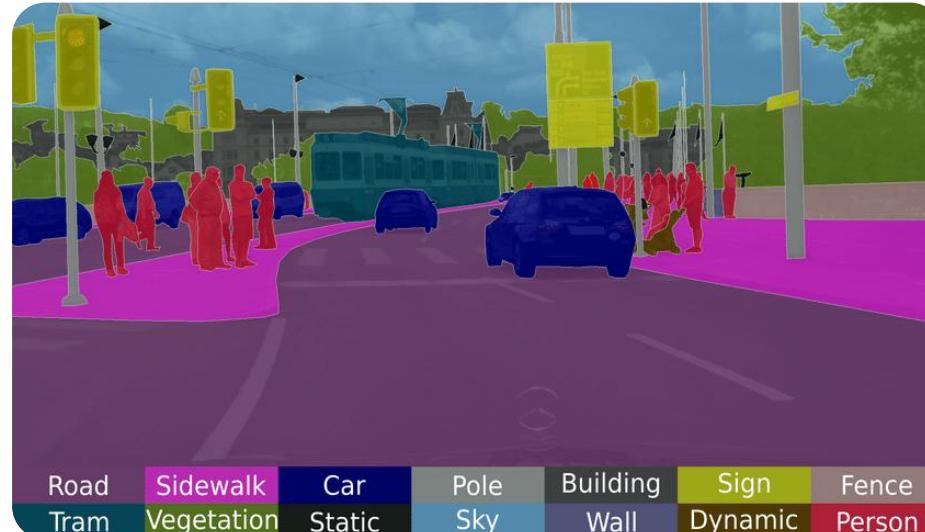
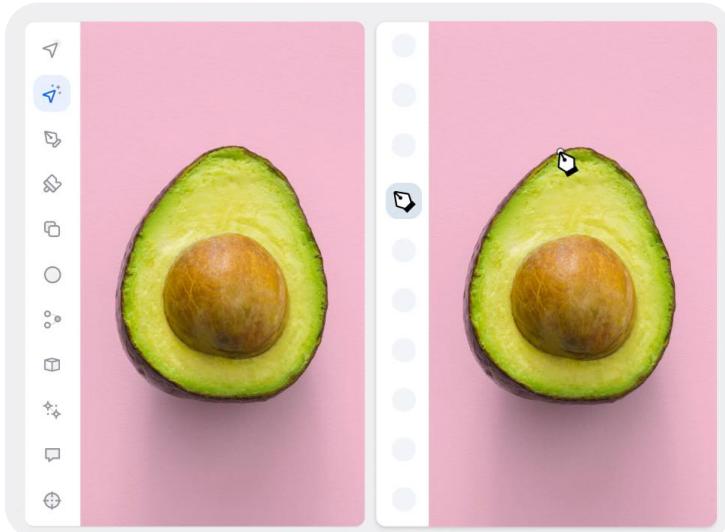


# What is Computer Perception/Vision?

A dark, futuristic-style image of a car's interior from the driver's perspective. The dashboard features a large, multi-layered heads-up display (HUD) with glowing blue and yellow graphics. The top layer shows a speedometer reading "046 mph". Below it is a circular interface with a red ring and a central white circle. The bottom layer displays a map of a city street grid. On the left side of the dashboard, there's a circular icon containing the letters "AI" with circuit board patterns around it. The overall atmosphere is high-tech and minimalist.

# What is Machine Perception/Vision?

# Some Applications



# What is this course about?

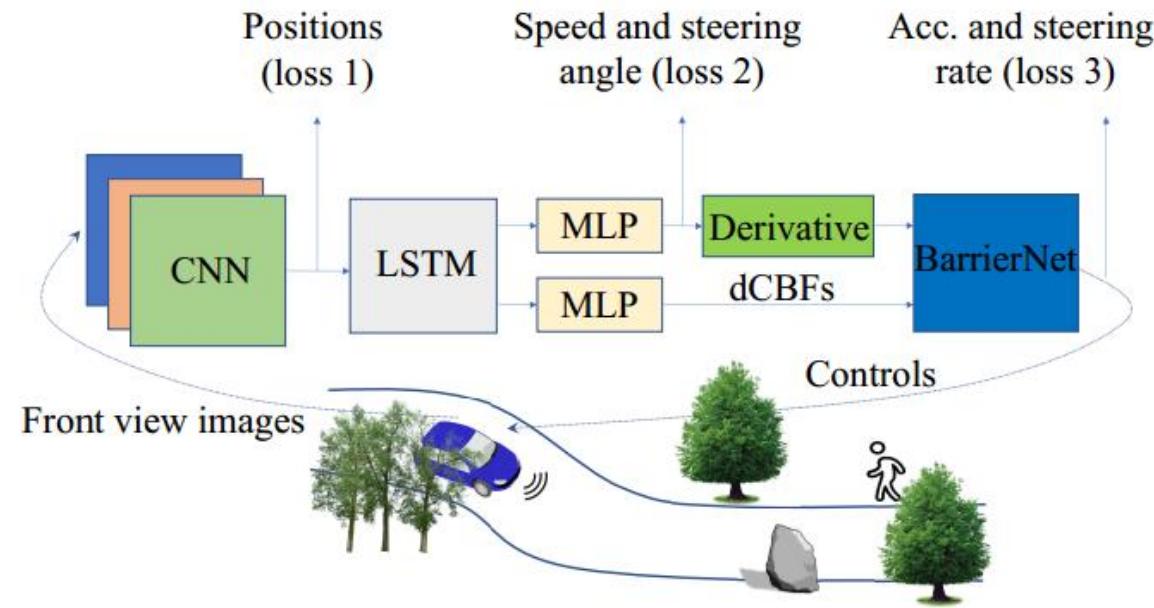


Magical  
“Box”



$p(\text{dog}|I)$   
Same as  $1 - p(\text{cat}|I)$

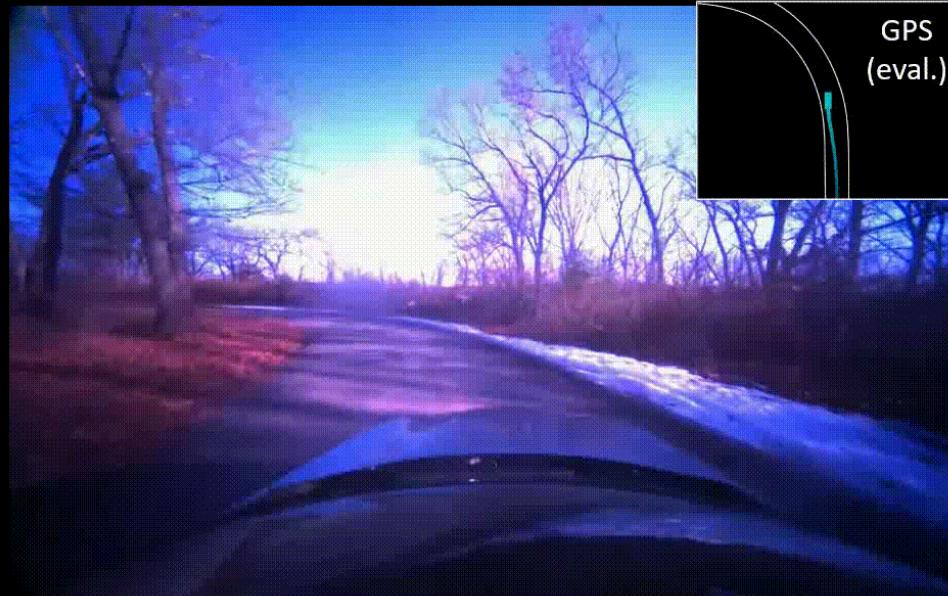
# What is this course about?



# What is this course about?

Challenging Case : Reflection & Shadow

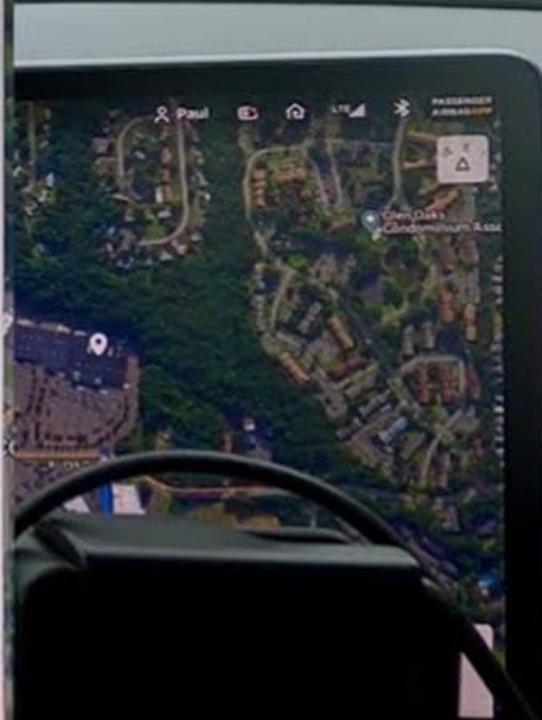
**Regular End-to-end learning**



**BarrierNet**



# What is this course about?



[Slight Street Sign Modifications Can Completely Fool Machine Learning Algorithms - IEEE Spectrum](#)  
[How Tesla Teaches Cars to Stop \(roboflow.com\)](#)

# What is this course about?



# Meet Your Awesome Grader/TA!

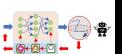


Khobragade, Shreyas Devdatta  
*he/him/his*

# **Class Locations**

**Washburn 323**

**T-F | 4:00 PM - 5:50 PM**



# COVID-19/Flu/Other Virus is Serious!

**Be safe and keep everyone safe!**

- Please stay back home if you are **sick** (don't come to class), contact me for accommodation
- **Please inform me** if you are sick
- Contact me via **Piazza** if you need extensions and you are out of your late days when you are sick
- Check WPI Resources: <https://www.wpi.edu/COVID-19-Resources>
- Masks are **optional**

# Class Resources



[https://github.com/  
Weixy21/RBX-  
474x595](https://github.com/Weixy21/RBX-474x595)



 [https://piazza.com/  
wpi/fall2025/rbe4  
74x595](https://piazza.com/wpi/fall2025/rbe474x595)

474x



[https://canvas.wpi.edu/  
courses/79479](https://canvas.wpi.edu/courses/79479)

595

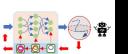


 <https://canvas.wpi.edu/courses/795>

16

# Class Resources

- Assignments, deadlines and other instructions will be released on the **Website (Syllabus)**
- Use **Piazza** for all communication, under extreme urgency such as COVID-19 contact or medical emergency use my email at **wxiao3[at]wpi[dot]edu. Use '[RBE474X]' tag to get through my email filters**
- **Canvas** will be used for all grades



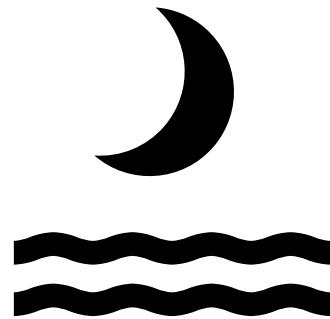
# Contacting Me

- Use **Piazza** for all communication, under extreme urgency such as COVID-19 contact or medical emergency use my email at **wxiao3[at]wpi[dot]edu**. Use **'[RBE474X]' tag to get through my email filters**
- **Do not contact me over email regarding grades: This is against the university policy**
- Feel free to contact me on **Piazza** to fix an appointment to meet me for discussions
- If you are interested in my research, **do email me!**
- My office location is Unity Hall 287. **My office hours are ONLY for Serious Issues with Appointment Over Piazza!**

# Home Rules!



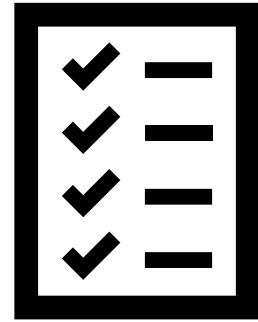
The entire class  
is your **family**,  
treat everyone  
with respect and  
help each other.



**4** late days  
**15%** late  
penalty per  
day.

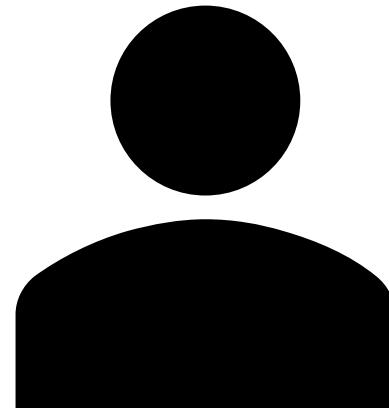


**No  
Cheating**  
<https://www.wpi.edu/about/policies/academic-integrity>



**Follow  
Submission  
Guidelines**  
**25% penalty** for  
not following  
them!

# Home Rules!



Strictly **In-person**

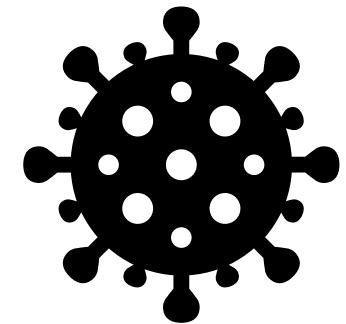


**Pro** ChatGPT and  
other LLMs  
**But NO**  
**CHEATING!**

# Home Rules!



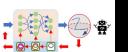
**Zero Tolerance**  
for plagiarism.



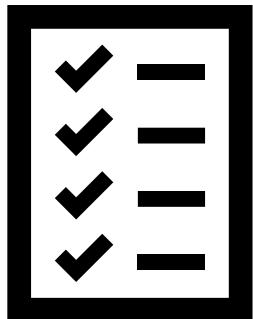
**Zero Tolerance**  
for not following  
health and  
safety protocols.

# Class Ideology

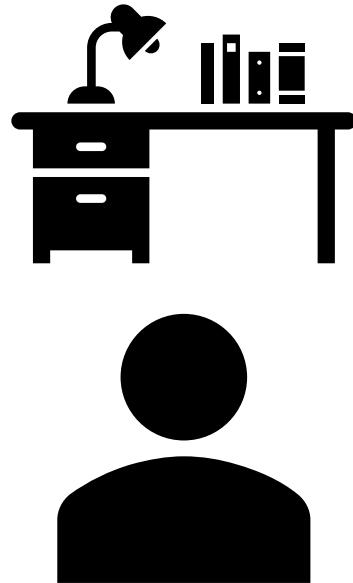
- Learning how to implement **concepts from scratch**
  - Math to code and code to math
  - Deeper understanding with extra reading outside class
- Failure is **common!**
  - It's not called failure, it's called "**path to success**"
  - **If life is without ups and downs, you have not tried hard enough!**
  - Giving up is **not**; leaving things till the **last minute** is **not** either!
- Ask for **conceptual** help!
  - Instructors **will not debug code**
  - Instructors **will not fix your computer**
- You need to **manage your time** throughout the term
  - It is easy to fall behind (submit assignments on time!)



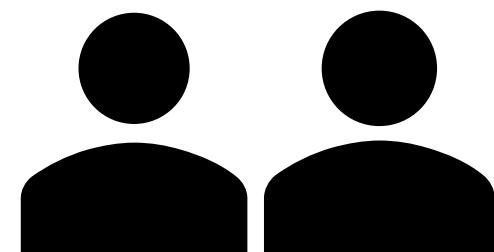
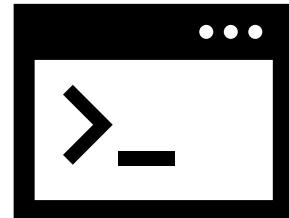
# Assignments



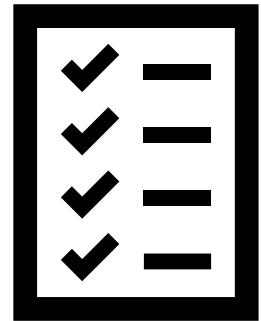
**Four**  
Assignments



**One**  
**Individual**  
Homework



**Two**  
**Homework**  
**Groups of 2 or 3**



**One**  
Course Project  
**Groups of 2 or 3**  
with oral presentation

# Reports

## CMSC733: Homework 0 - Alohomora

Kun Yu Phan  
Email: shawnpingyu@gmail.com  
UW.edu: ykun

### I. PLACE 1: SAVING MY BOUNDARY

In this section, I shall present a short sketch of my boundary detection algorithm.

The idea of the algorithm is to use little information I receive from the input image to detect the boundary of the image. The method consists of 3 main steps: 1) find the boundary; 2) remove the noise; 3) smooth the boundary.

### 2. Noise Reduction

The first step in this section is to generate a binary mask to the noise in those three filters. To do this, we can use the `threshold` function in MATLAB.

### 3. Edge Detection

The first step in this section is to generate a binary mask to the edges in those three filters. To do this, we can use the `edge` function in MATLAB.

### 4. Boundary Detection

The first step in this section is to generate a binary mask to the boundaries in those three filters. To do this, we can use the `im2bw` function in MATLAB.

### 5. Smooth Boundary

The first step in this section is to smooth the boundary. To do this, we can use the `imfilter` function in MATLAB.

### 6. Final Output

The final output of this section is a binary mask of the boundary.

### 7. Edge Detection

The first step in this section is to generate a binary mask to the edges in those three filters. To do this, we can use the `edge` function in MATLAB.

### 8. Boundary Detection

The first step in this section is to generate a binary mask to the boundaries in those three filters. To do this, we can use the `im2bw` function in MATLAB.

### 9. Smooth Boundary

The first step in this section is to smooth the boundary. To do this, we can use the `imfilter` function in MATLAB.

### 10. Final Output

The final output of this section is a binary mask of the boundary.

### 11. Edge Detection

The first step in this section is to generate a binary mask to the edges in those three filters. To do this, we can use the `edge` function in MATLAB.

### 12. Boundary Detection

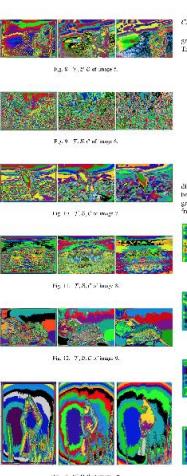
The first step in this section is to generate a binary mask to the boundaries in those three filters. To do this, we can use the `im2bw` function in MATLAB.

### 13. Smooth Boundary

The first step in this section is to smooth the boundary. To do this, we can use the `imfilter` function in MATLAB.

### 14. Final Output

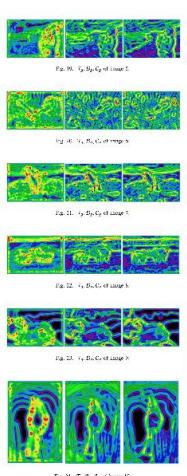
The final output of this section is a binary mask of the boundary.



**C. Feature Extraction, Edge Detection and Segmentation**

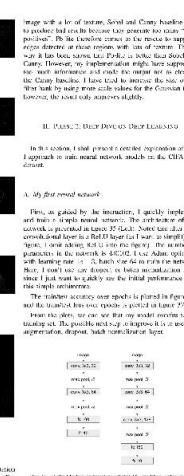
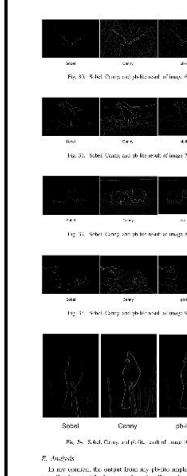
In order to segment the source, negative, and color gradient maps, we have to use the idea of feature maps. Label maps are captured as input  $\mathcal{L}$ .

Fig. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16



**D. Boundary Detection**

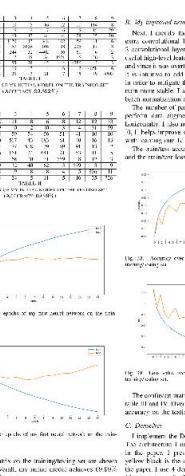
The final step is to extract the  $\mathcal{S}$ ,  $\mathcal{N}$ , and  $\mathcal{C}$  features and to perform the boundary detection test. The results of all 10 provided images are illustrated from Figure 23 to 30.



**E. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

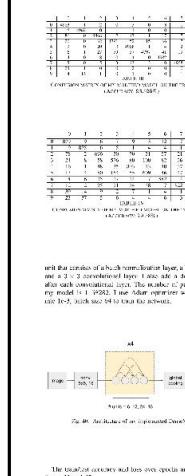
Fig. 21, 22, 23, 24, 25, 26, 27, 28, 29, 30



**F. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 31, 32, 33, 34, 35, 36, 37, 38, 39, 40



**G. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

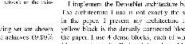
Fig. 41, 42, 43, 44, 45, 46, 47, 48, 49, 50



**H. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 51, 52, 53, 54, 55, 56, 57, 58, 59, 60



**I. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 61, 62, 63, 64, 65, 66, 67, 68, 69, 70



**J. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 71, 72, 73, 74, 75, 76, 77, 78, 79, 80

**K. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 81, 82, 83, 84, 85, 86, 87, 88, 89, 90



**L. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 91, 92, 93, 94, 95, 96, 97, 98, 99, 100



**M. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 101, 102, 103, 104, 105, 106, 107, 108, 109, 110

**N. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 141, 142, 143, 144, 145, 146, 147, 148, 149, 150

**O. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 151, 152, 153, 154, 155, 156, 157, 158, 159, 160

**P. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 161, 162, 163, 164, 165, 166, 167, 168, 169, 170

**Q. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 171, 172, 173, 174, 175, 176, 177, 178, 179, 180

**R. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 181, 182, 183, 184, 185, 186, 187, 188, 189, 190

**S. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 191, 192, 193, 194, 195, 196, 197, 198, 199, 200

**T. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 201, 202, 203, 204, 205, 206, 207, 208, 209, 210

**U. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 211, 212, 213, 214, 215, 216, 217, 218, 219, 220

**V. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 221, 222, 223, 224, 225, 226, 227, 228, 229, 230

**W. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 231, 232, 233, 234, 235, 236, 237, 238, 239, 240

**X. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 241, 242, 243, 244, 245, 246, 247, 248, 249, 250

**Y. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 251, 252, 253, 254, 255, 256, 257, 258, 259, 260

**Z. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 261, 262, 263, 264, 265, 266, 267, 268, 269, 270

**A. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 271, 272, 273, 274, 275, 276, 277, 278, 279, 280

**B. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 281, 282, 283, 284, 285, 286, 287, 288, 289, 290

**C. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 291, 292, 293, 294, 295, 296, 297, 298, 299, 300

**D. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 301, 302, 303, 304, 305, 306, 307, 308, 309, 310

**E. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 311, 312, 313, 314, 315, 316, 317, 318, 319, 320

**F. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 321, 322, 323, 324, 325, 326, 327, 328, 329, 330

**G. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 331, 332, 333, 334, 335, 336, 337, 338, 339, 340

**H. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 341, 342, 343, 344, 345, 346, 347, 348, 349, 350

**I. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 351, 352, 353, 354, 355, 356, 357, 358, 359, 360

**J. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 361, 362, 363, 364, 365, 366, 367, 368, 369, 370

**K. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 371, 372, 373, 374, 375, 376, 377, 378, 379, 380

**L. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 381, 382, 383, 384, 385, 386, 387, 388, 389, 390

**M. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 391, 392, 393, 394, 395, 396, 397, 398, 399, 400

**N. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 401, 402, 403, 404, 405, 406, 407, 408, 409, 410

**O. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 411, 412, 413, 414, 415, 416, 417, 418, 419, 420

**P. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 421, 422, 423, 424, 425, 426, 427, 428, 429, 430

**Q. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 431, 432, 433, 434, 435, 436, 437, 438, 439, 440

**R. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 441, 442, 443, 444, 445, 446, 447, 448, 449, 450

**S. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 451, 452, 453, 454, 455, 456, 457, 458, 459, 460

**T. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 461, 462, 463, 464, 465, 466, 467, 468, 469, 470

**U. Edge Detection**

In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 471, 472, 473, 474, 475, 476, 477, 478, 479, 480

**V. Edge Detection**

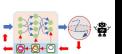
In this section, I shall present a brief expression of the Laplacian operator in each image.

Fig. 481, 482, 483, 484, 485, 486, 487, 488, 489, 490

# Evaluation

Assignment Name	Grade Percentage	Deadline
In-class quiz	5	
HW0: Alohomora ( <b>Individual</b> )	10	10/31/2025
HW1: Nifty Neural Networks! ( <b>Group of 2 or 3</b> )	15	11/14/2025
HW2: Dreaming Data! ( <b>Group of 2 or 3</b> )	20	12/05/2025
Course Project: Learning for End-to-End Robot Policies: From Perception to Control ( <b>Group of 2 or 3</b> ) – <b>coding, report, and oral presentations</b>	50	12/12/2025

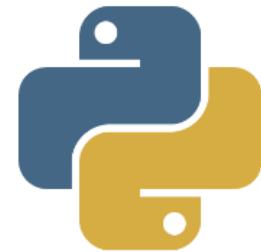
Instructions and Codebase - <https://github.com/Weixy21/RBX-474x595>



# Software Environment



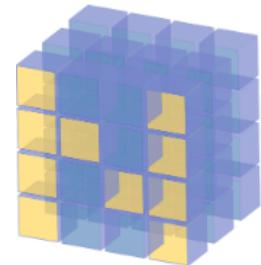
ubuntu



python



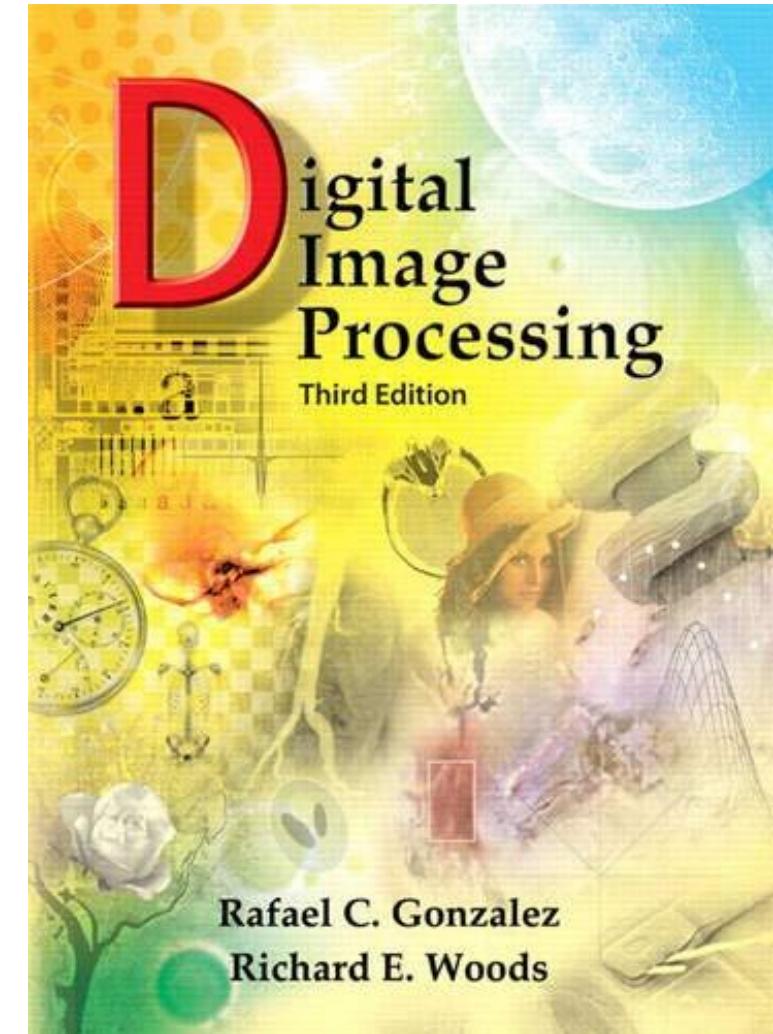
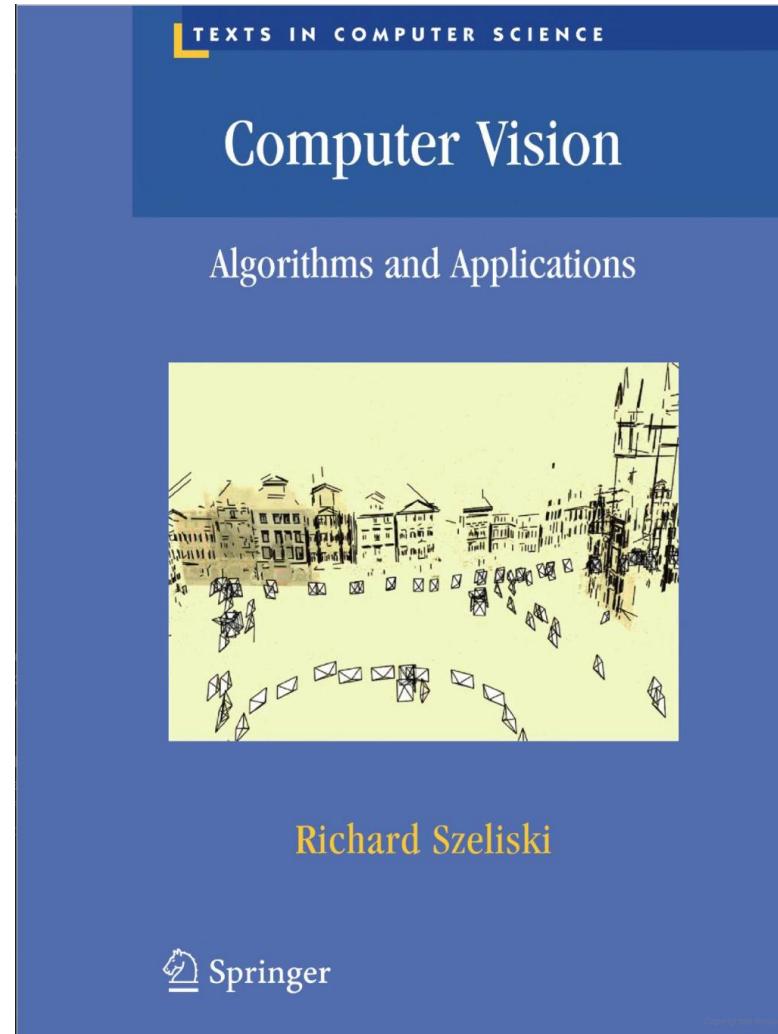
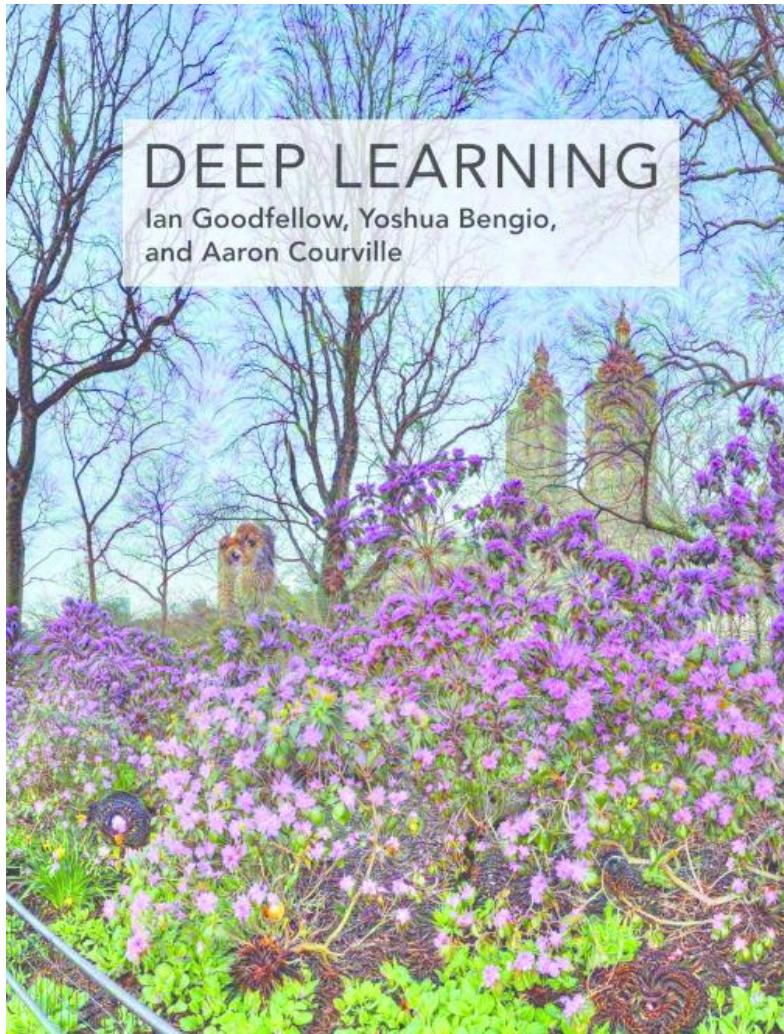
OpenCV



NumPy



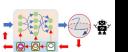
# Reference Books



# Acknowledgements

## More Resources

- [University of Maryland, College Park's CMSC733](#)
- [University of Minnesota, Twin Cities' CSCI5563](#)
- [Stanford University's CS231n](#)
- [University of Maryland, College Park's CMSC764](#)
- [WPI, RBE 474X/595 2024](#)



# Office Hours



**Prof. Wei Xiao**  
*he/him/his*  
Unity Hall 287  
Office Hours: For Serious  
Concerns ONLY



**Shreyas Devdatta  
Khobragade**  
*he/him/his*  
TBD  
Office Hours: **TBD**

# RBE474X vs RBE595-B01-ST

Undergraduate flavor

High Expectations

Graduate parts of the assignment are optional and extra credit

Gives a good coverage of the basics

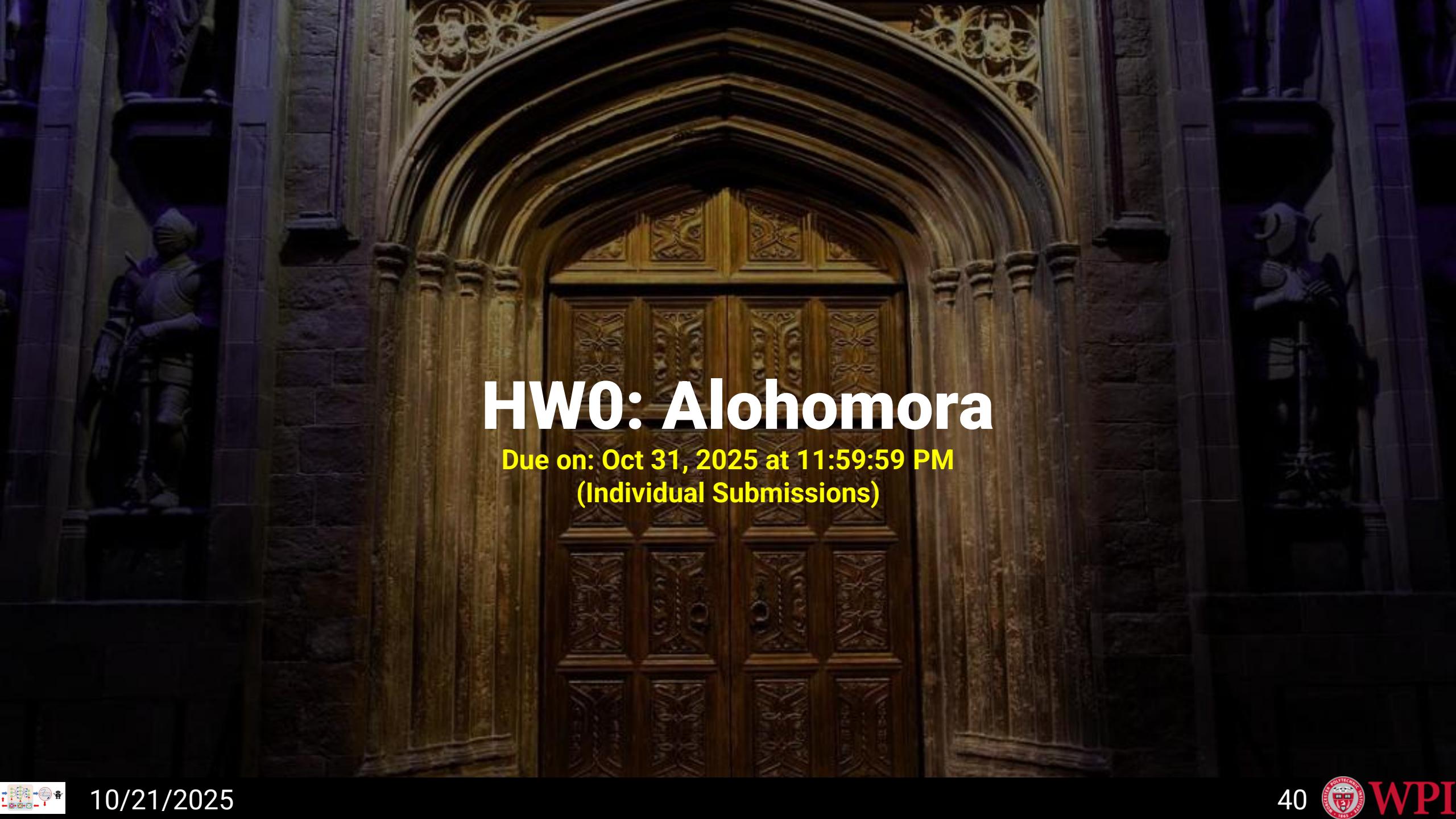
Graduate flavor

Higher Expectations

Graduate parts of the assignment are **REQUIRED**

Gives a good coverage of the basics + a little extra



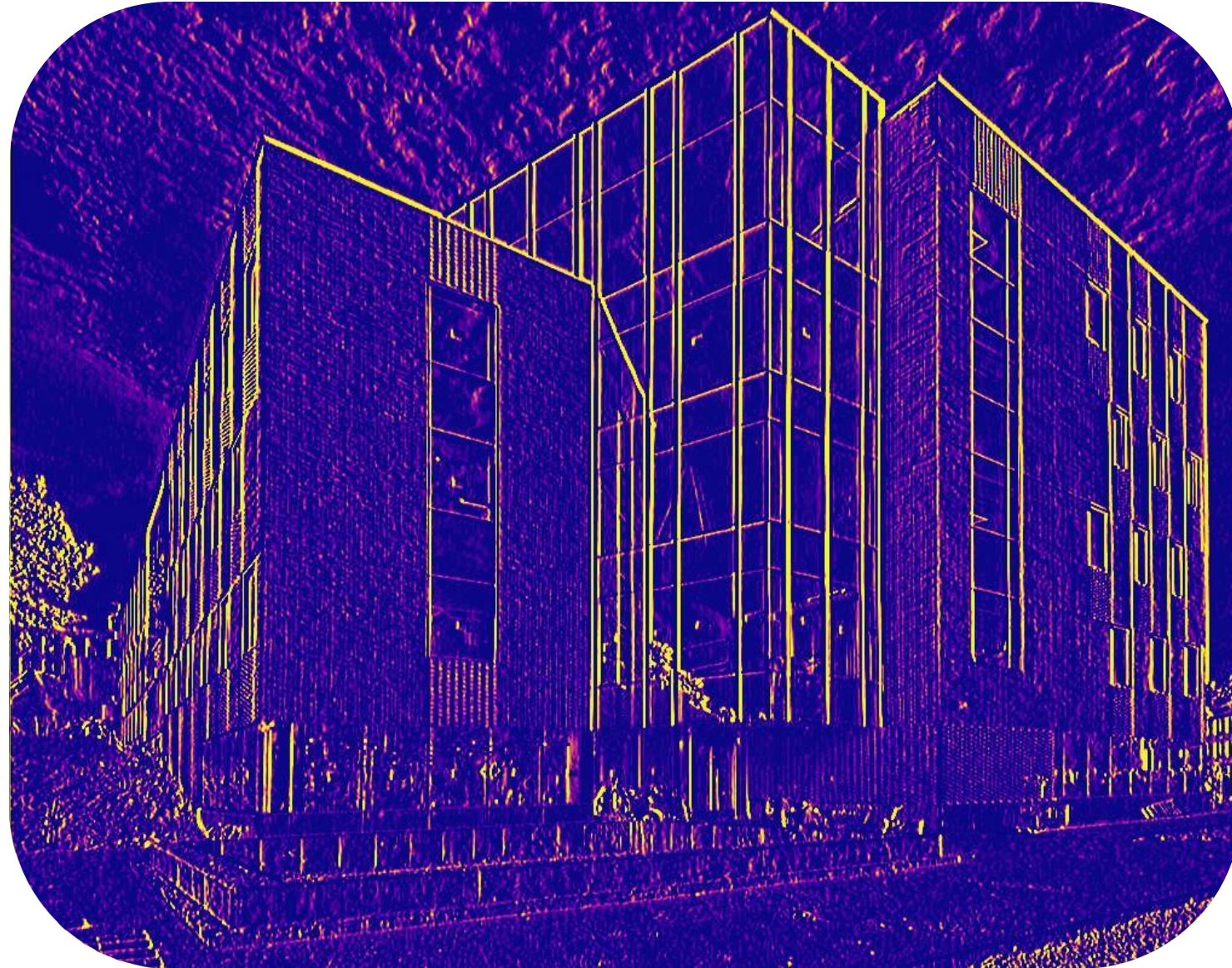


# HWO: Alohomora

**Due on: Oct 31, 2025 at 11:59:59 PM  
(Individual Submissions)**

# Part 1

*Convolution*



# Part 2

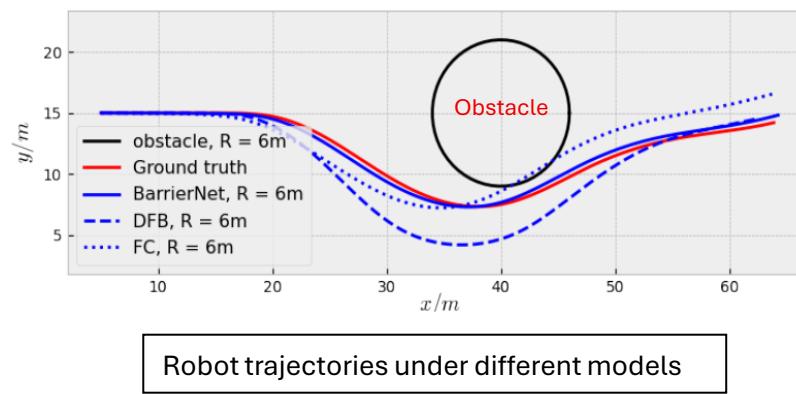
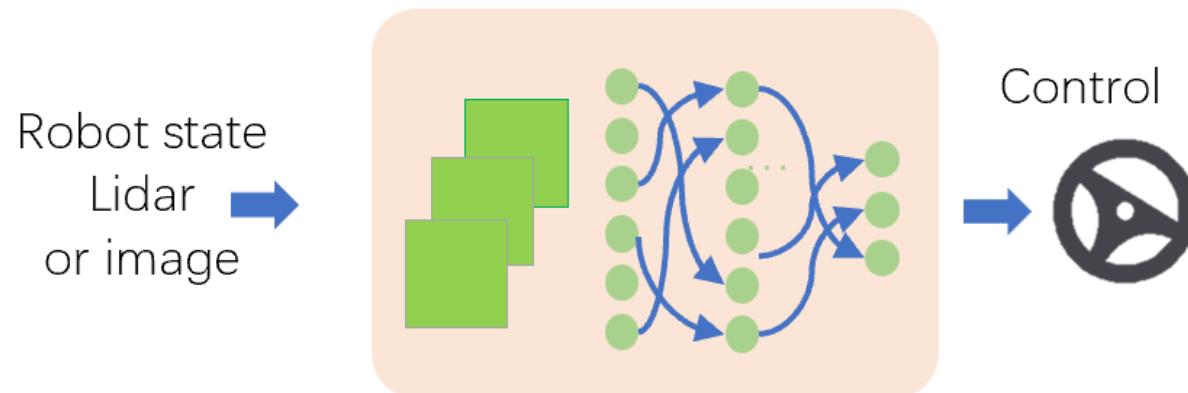
*Classification  
based on  
colors*



# Course Project

Learning for End-to-End Robot Policies: From Perception to Control (**Group of 2 or 3**)

An End-to-end learning system



Duration:  
From now to 12/12/2025

# Course Project

Learning for End-to-End Robot Policies: From Perception to Control (**Group of 2 or 3**)

1. Learning from robot state to control (option 1, **recommended for undergraduates**)
2. Learning from Lidar to control (option 2, **recommended for graduates**)
3. Learning from image to control (option 3, **advanced project**)

# Course Project

Learning for End-to-End Robot Policies: From Perception to Control (**Group of 2 or 3**)

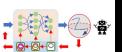
1. Complete all tasks (except optional safety filters): 30
2. Project reports with all necessary contents from all tasks: 30
3. Project presentations: 40
4. Add safety filters: 50 (bonus)

**Total: full credits (100/100), credits with bonus (150/100)**

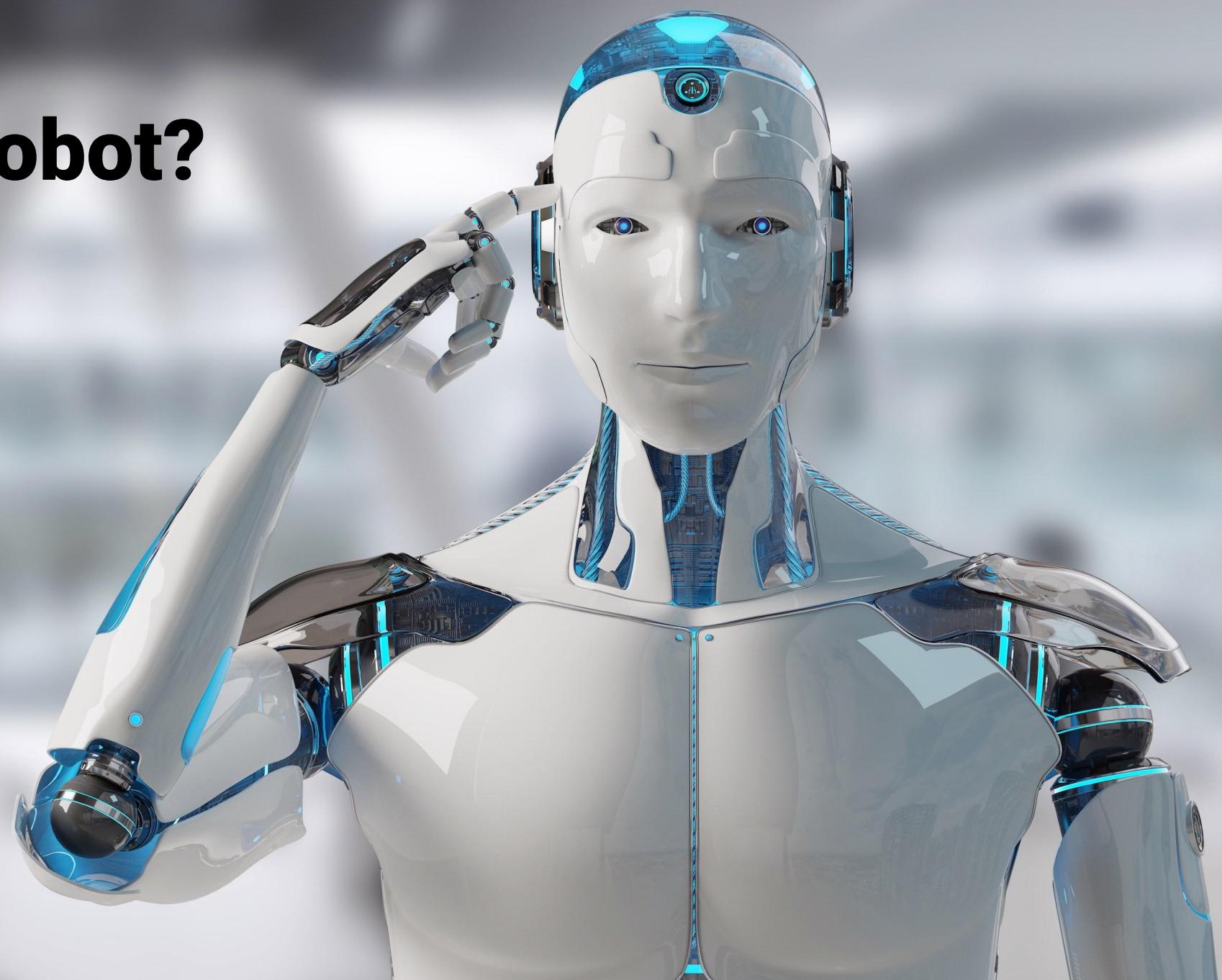
Note: You may complete both task options 1 and 2, and there will also be a bonus based on the performance (determined by the lecture and TA).

# Course Schedule

- 10.21 Class 1: Introduction, Logistics And Sensors
- 10.24 Class 2: Multi Layer Perceptron And Backpropagation
- 10.28 Class 3: NN Tuning, Image Filtering And Convolutional Neural Networks
- 10.31 Class 4: Advanced CNN Architectures And Image Warping
- 11.4 No class
- 11.7 Class 5: Simulation for Data Generation And Sim2Real
- 11.11 Class 6: Object Detection And Segmentation
- 11.14 Class 7: Learned Depth: Monocular + Stereo
- 11.18 Class 8: Vision Transformers, Can We Trust Neural Networks?
- 11.21 Class 9: Single Pixel Attacks, Patch Based Attacks
- 11.25 Class 10: Generative Models: VAEs, GANs, Attacking GANs
- 11.28 No class
- 12.2 Class 11: Advanced Generative Models: Diffusion Models
- 12.5 Class 12: Advanced Generative Models++: Multi-modal Generative Deep Learning
- 12.9 Presentation 1 (Group-based)
- 12.12 Presentation 2 + Course Summary (Group-based)



# What is a Robot?



# Robot

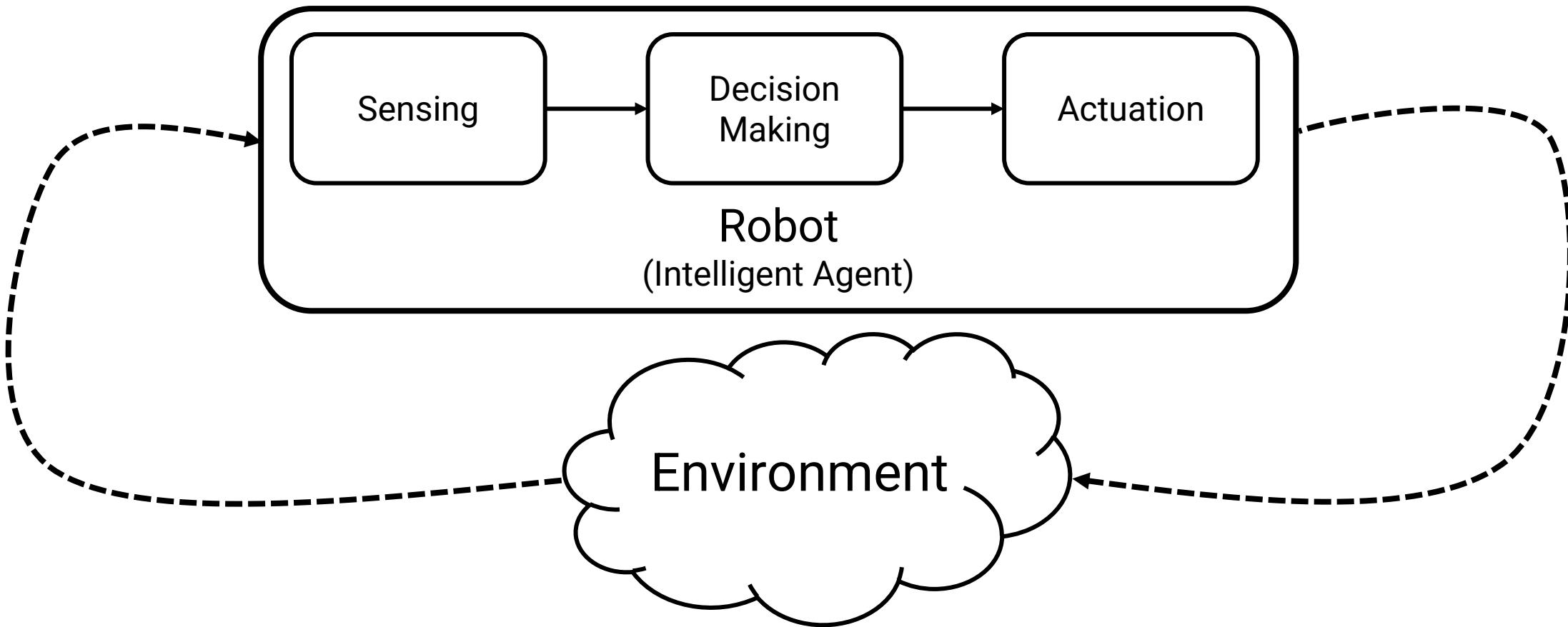
## High-school Version

Robot  
(Intelligent Agent)



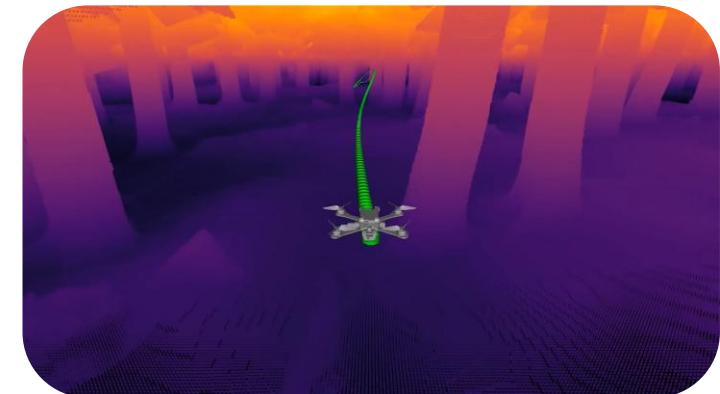
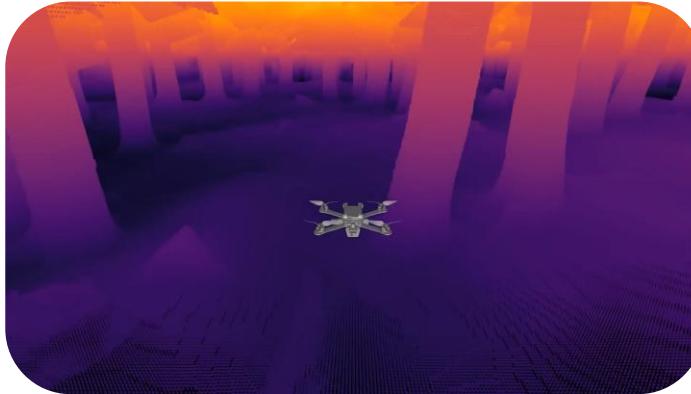
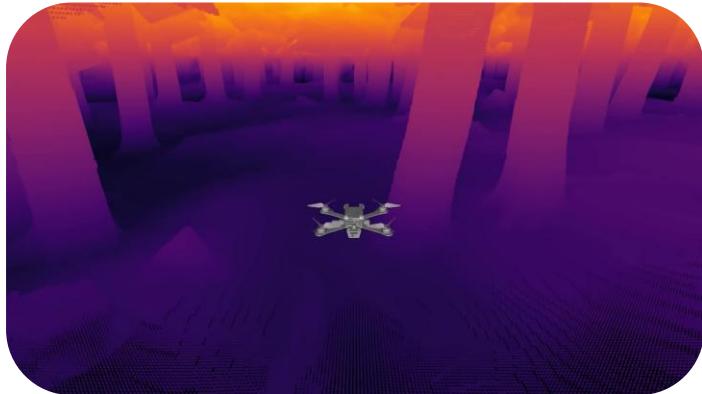
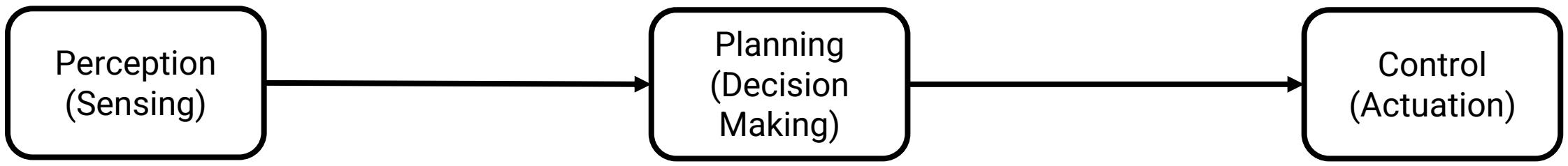
# Robot

## High-school Version



# Robot

## Ph.D. Version



Loquercio, Antonio, Elia Kaufmann, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. "Learning high-speed flight in the wild." *Science Robotics* 6, no. 59 (2021).

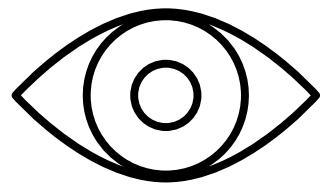
# Sensors!

## What is a sensor?

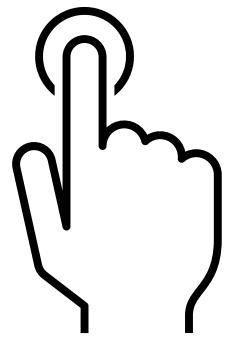
A device which detects or measures a physical property and records, indicates, or otherwise responds to it.

**AKA How we perceive the world?**

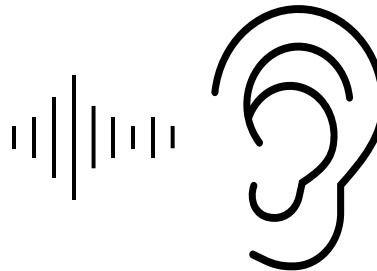
# Types of Sensors!



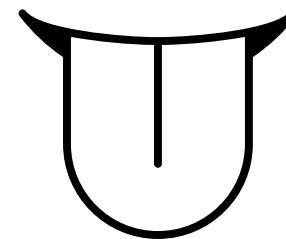
Vision



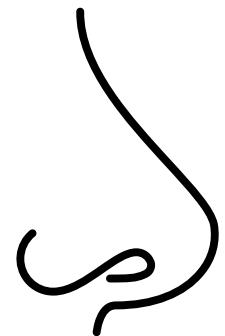
Touch



Sound



Taste



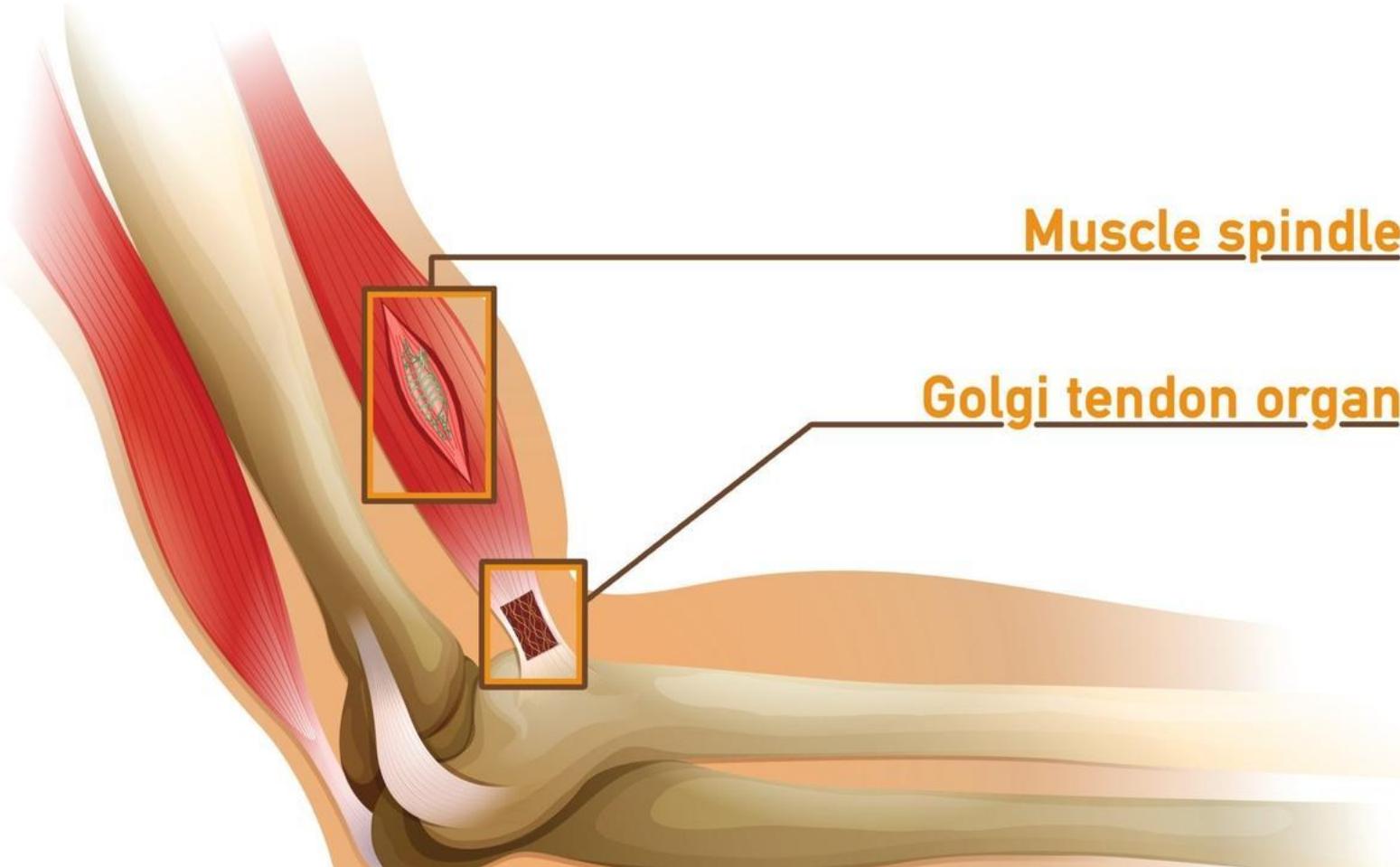
Smell

**What kind of sensor enables you to know how much your knee is rotated?**

# ANOTHER ONE



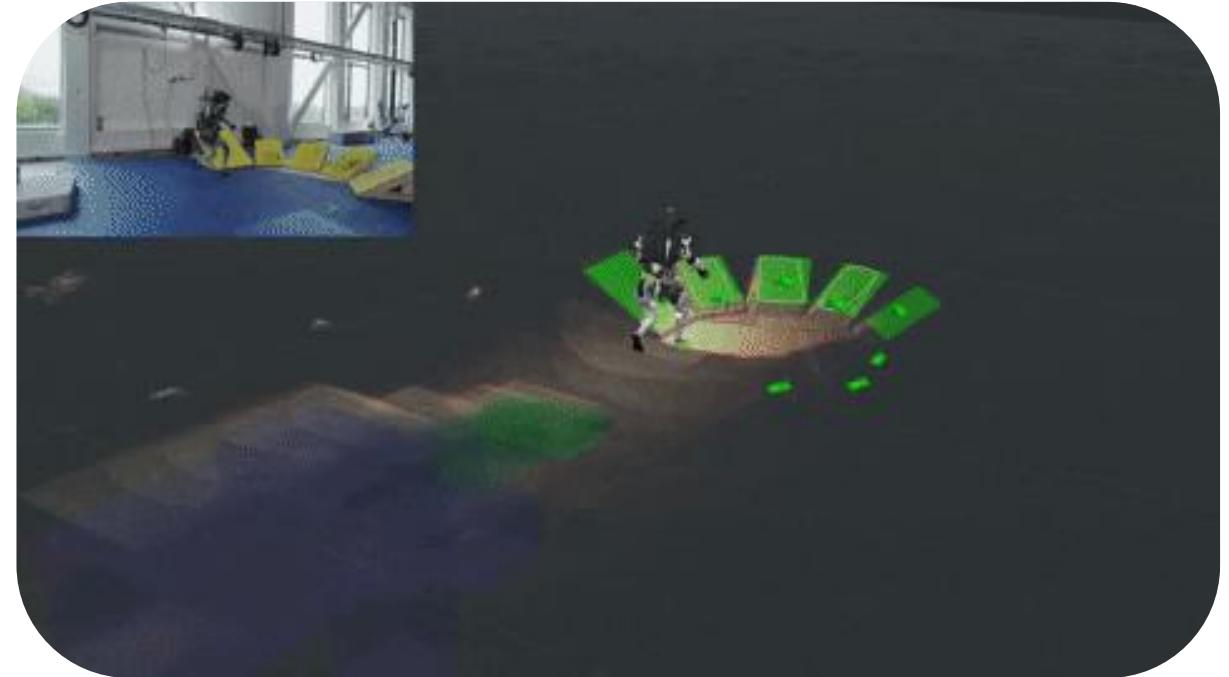
# Internal Perception: Proprioception



# Let's Play A Game!



Boston Dynamics Atlas Humanoid Robot



What sensors does the Atlas have?  
What does the perception/vision stack have?  
What does the planning stack have?  
What does the control stack have?  
What does the AI stack have?

# Let's Play A Game!



PRG Husky



The Husky doesn't perceive depth!  
How do you think it's finding the gap?

# Let's Play A Game!



Boston Dynamics Spot Mini



What considerations should the Spot have to open the door and go in?

# Let's Play A Game!



Tesla Model S

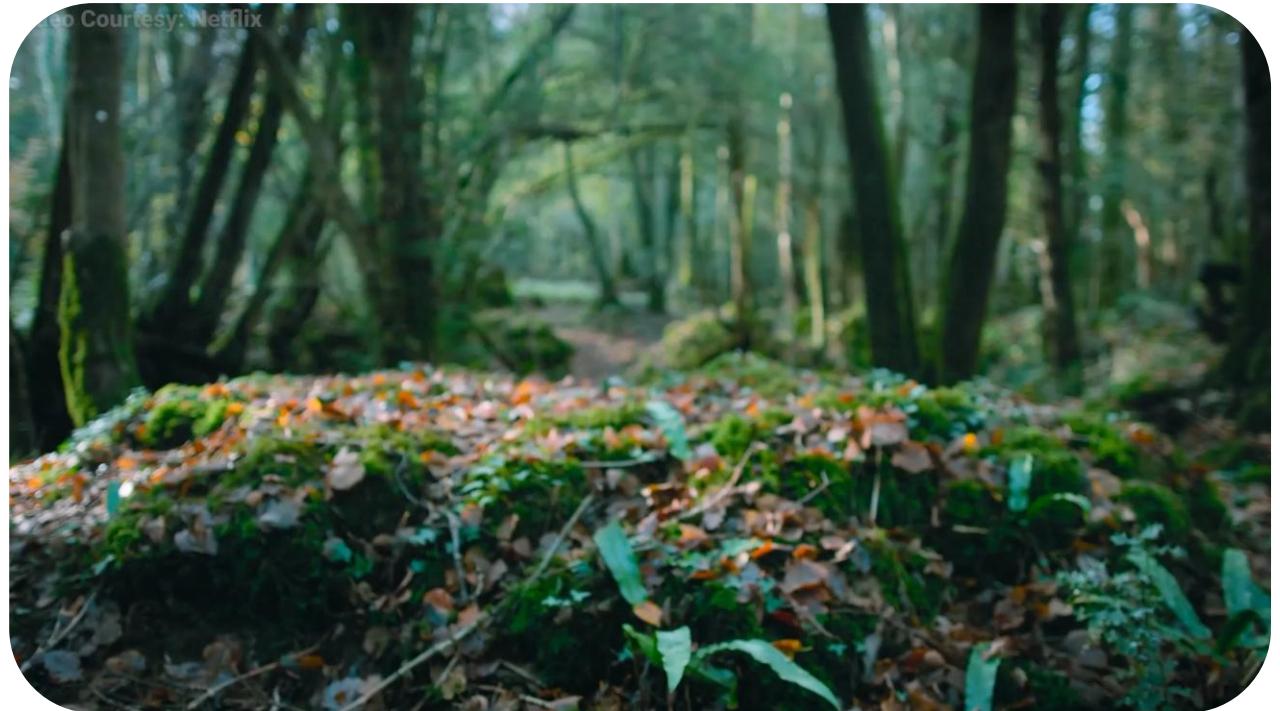


What design considerations should the car have?  
What scenarios should it work in?

# Let's Play A Game!



PRG Labrador 500 α



The labrador doesn't sense depth!  
How do you think it can navigate through various scenarios?

# Most Ubiquitous Robotics Sensor!

Camera/Imaging Sensor

- Cheap
- Dense information
- Mimics one of our senses
- Multi-functional





**Elon Musk** 

@elonmusk · [Follow](#)

Vision became so good that radar actually reduced SNR,  
so radar was turned off.

Humans drive with eyes & biological neural nets, so  
makes sense that cameras & silicon neural nets are only  
way to achieve generalized solution to self-driving.

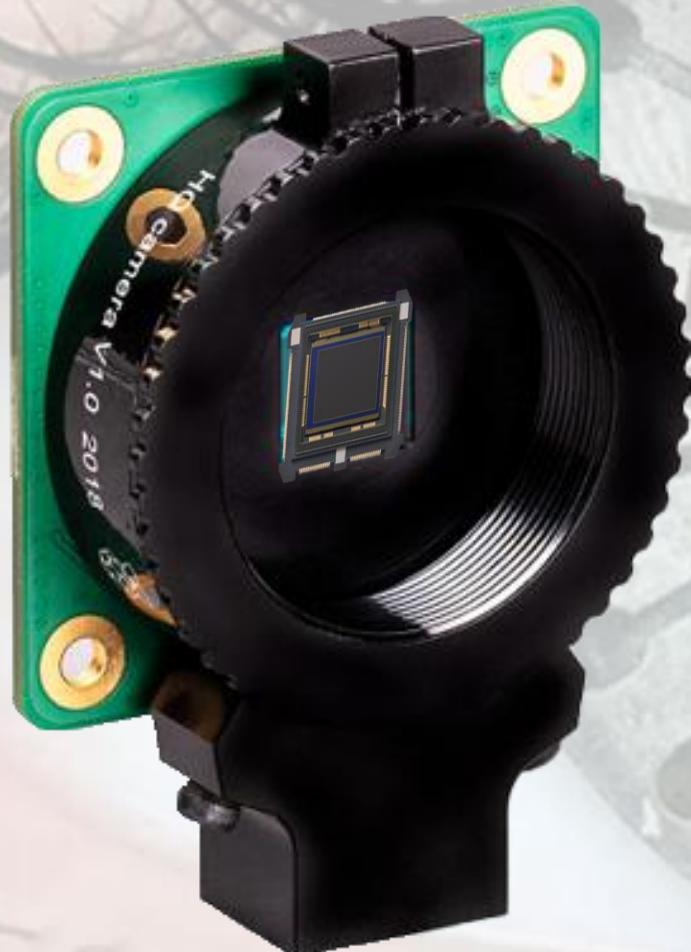
11:44 AM · Oct 11, 2021





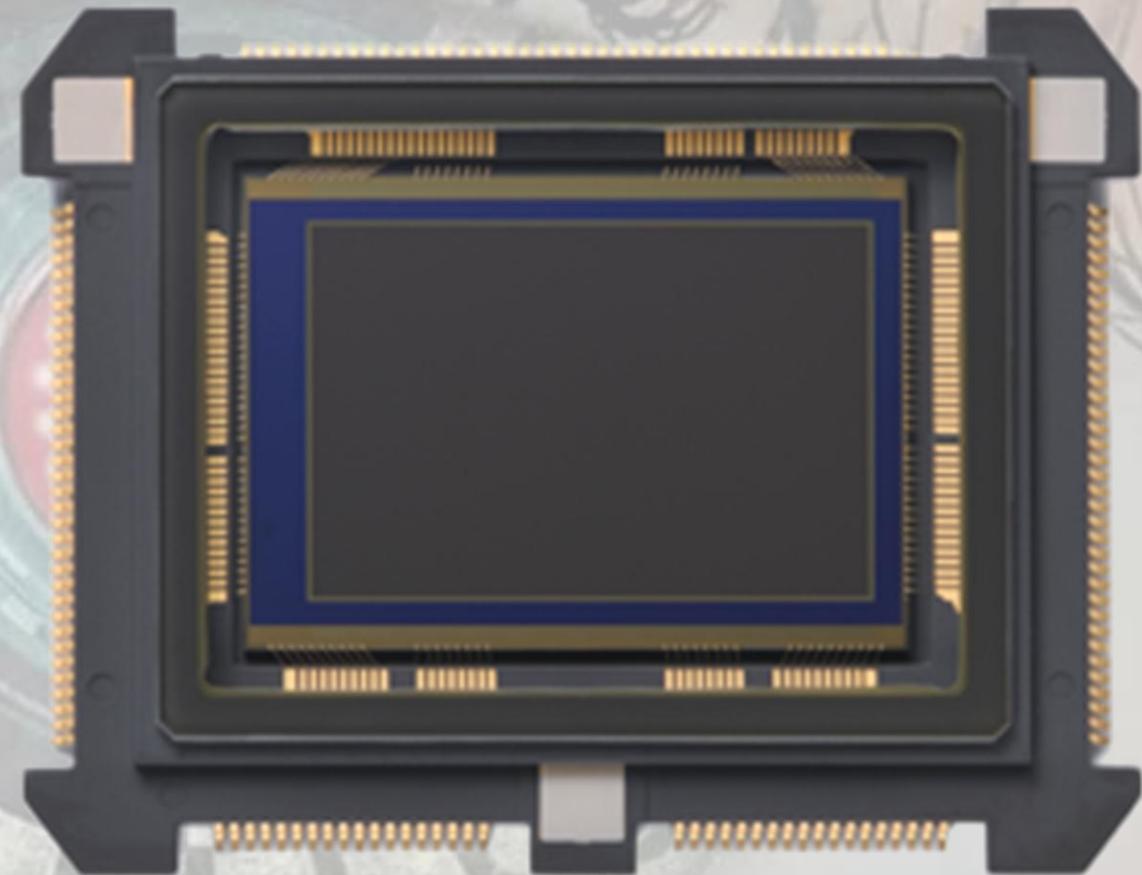
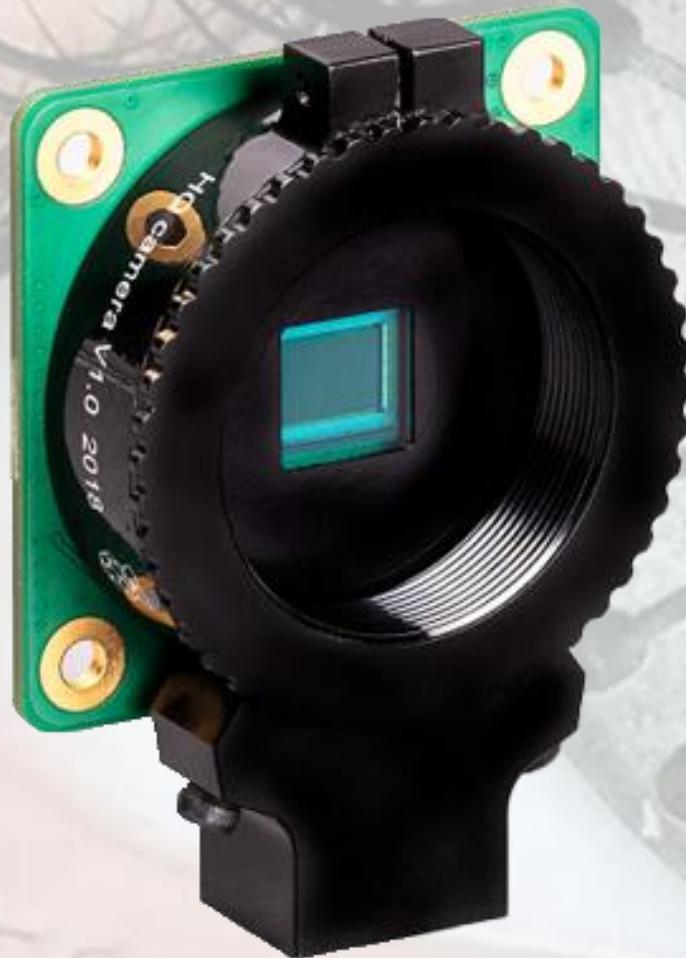
# Vision Sensors

## Imaging Sensors



# Vision Sensors

## Imaging Sensors

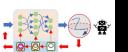


# Grayscale Imaging

Measures  $\frac{\text{Number of Photons}}{\text{Time}}$

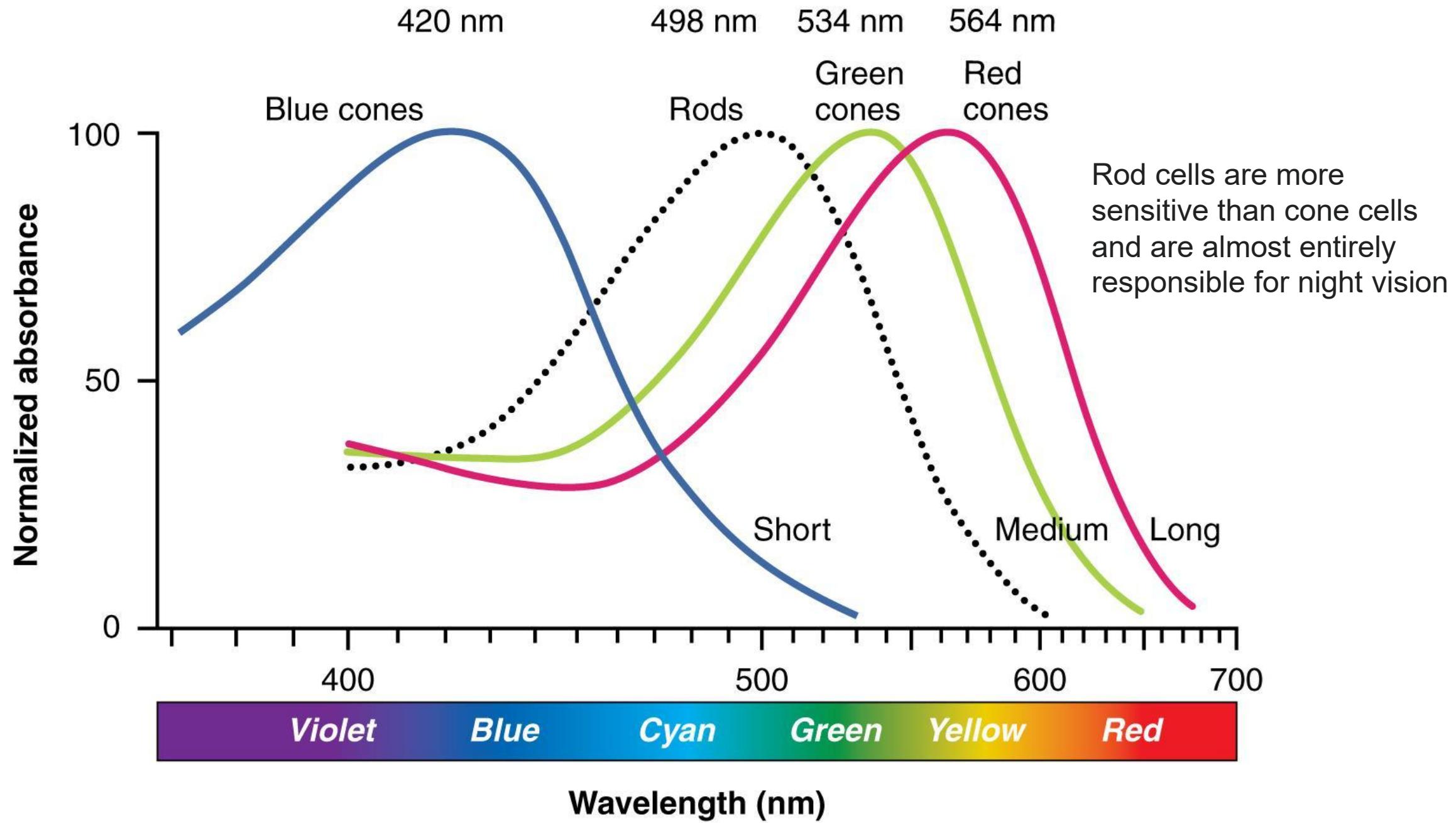
# RGB Color Imaging

Why **RGB**?

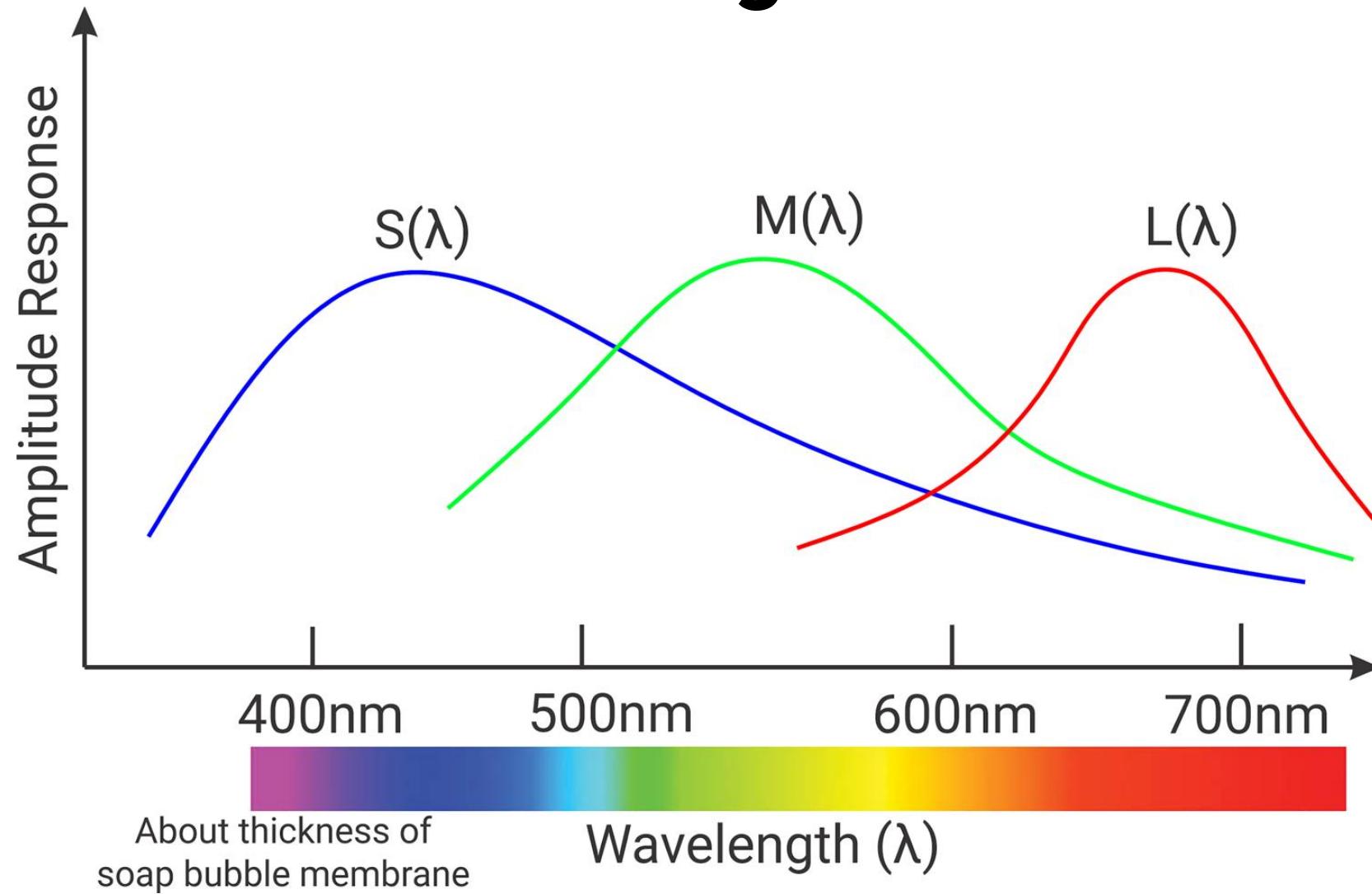




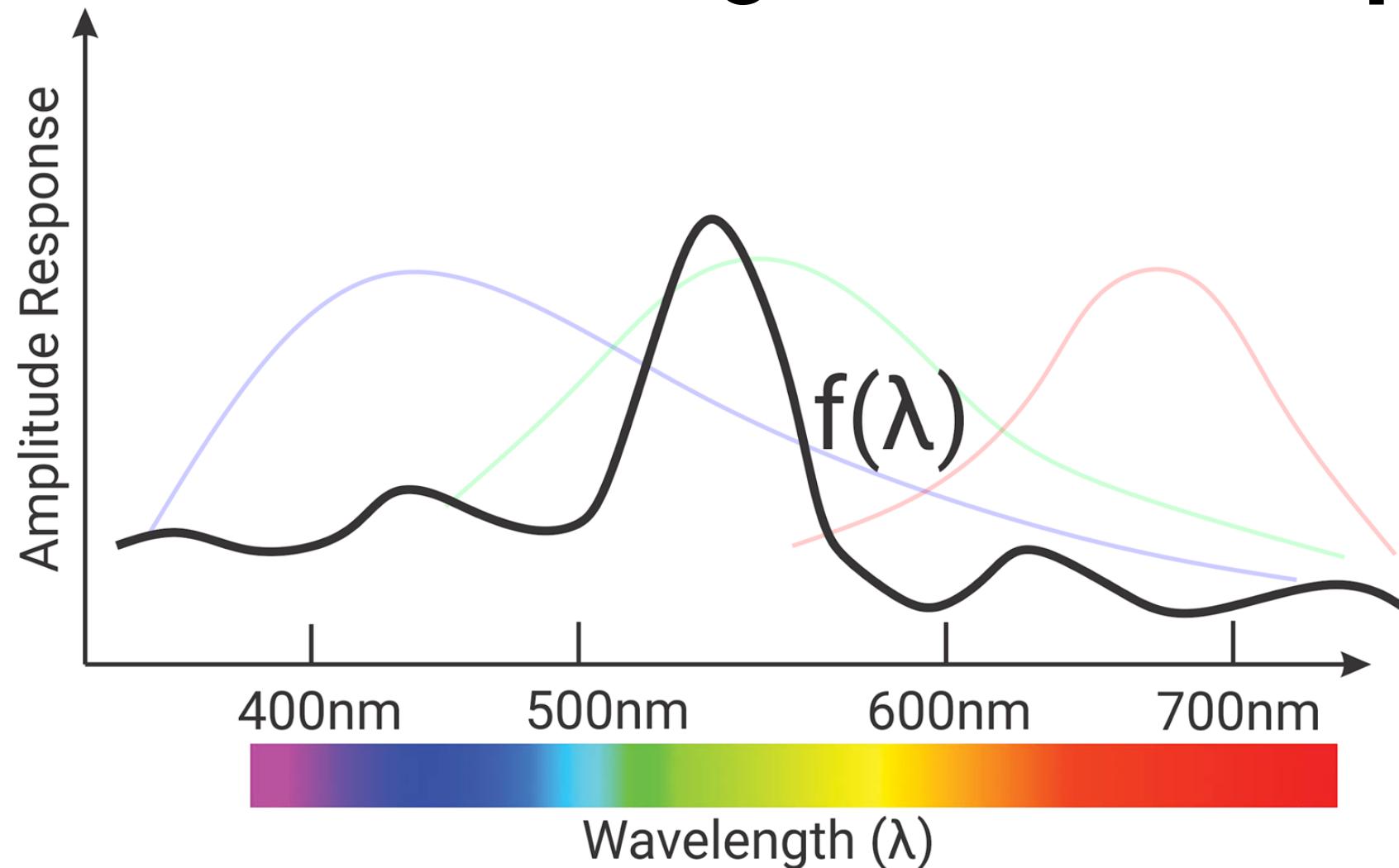
# How do we see color?



# Mathematical Modelling of Color Perception



# Mathematical Modelling of Color Perception



# Mathematical Modelling of Color Perception

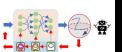
$$S_{res} = \int S(\lambda) f(\lambda) d\lambda$$

$$M_{res} = \int M(\lambda) f(\lambda) d\lambda$$

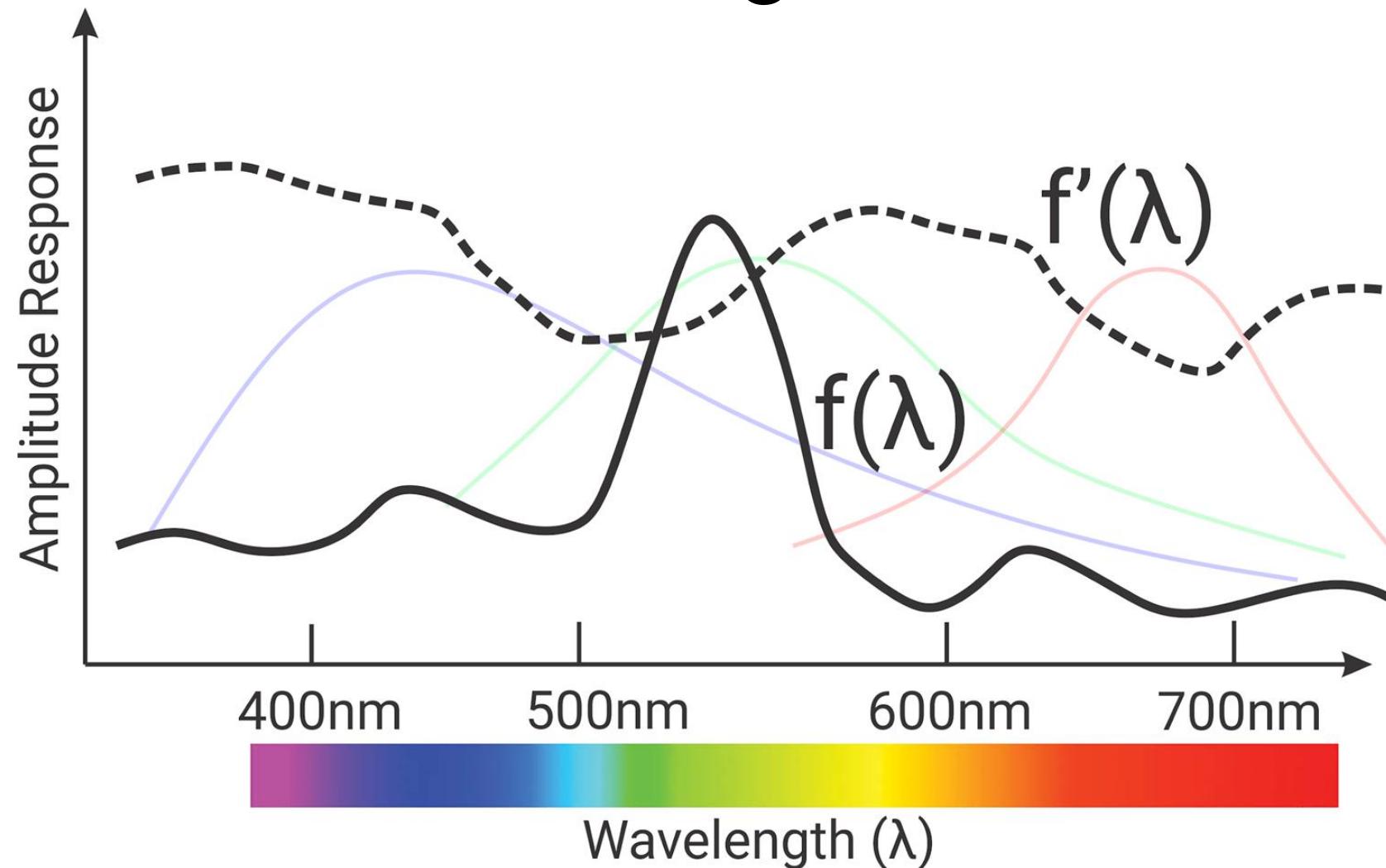
$$L_{res} = \int L(\lambda) f(\lambda) d\lambda$$

$$\mathbb{R}^\infty \rightarrow \mathbb{R}^3$$

how a color will be perceived under specific viewing conditions



# Mathematical Modelling of Color Perception



# Color Illusion



**ORIGINAL**

(Blue and Gold)  
+0% brightness, +0%  
contrast



**BRIGHTER**

(White and Gold)  
+40% brightness, +40%  
contrast



**DARKER**

(Blue and Black)  
-30% brightness, +40%  
contrast

# Next Class!



Multi Layer Perceptrons And Backpropagation