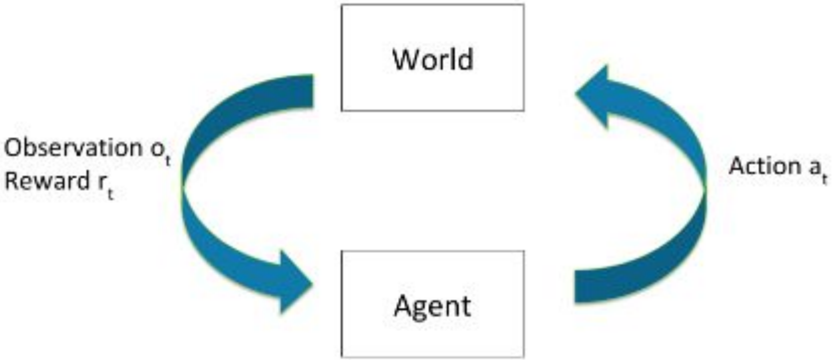
Q-Learning + Project 4 Introduction

TAs: Shimian Zhang, Keaton Kraiger
03/21/2023

First, let's talk about Q-learning

Q-Learning for RL

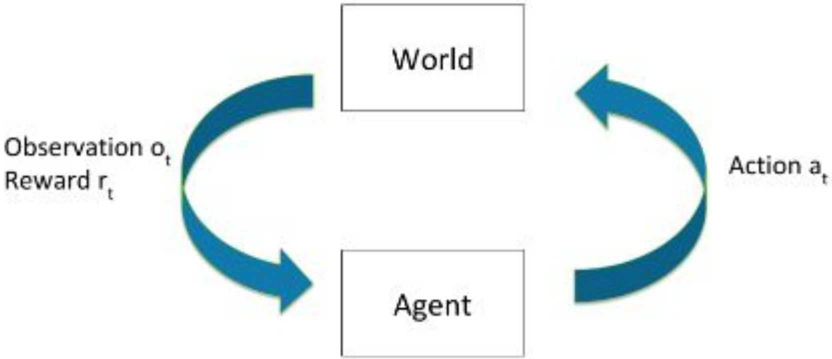
- Q-learning is a classical Reinforcement Learning (RL) approach



- Each time step t:
  - Agent takes an action a_t
  - World updates given action a_t, emits observation o_t and reward r_t
  - Agent receives observation o_t and reward r_t

Emma Brunskil, Stanford CS234: https://web.stanford.edu/class/cs234/slides/lecture1.pdf

History



- History $h_t = (a_1, o_1, r_1, \ldots, a_t, o_t, r_t)$
- Agent chooses action based on history
- State is information assumed to determine what happens next
  - Function of history: $s_t = (h_t)$

Emma Brunskil, Stanford CS234: https://web.stanford.edu/class/cs234/slides/lecture1.pdf

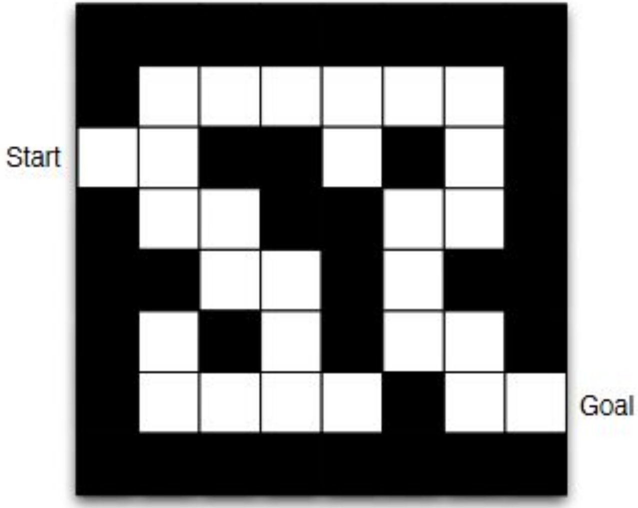
Policy

- Want to learn a policy (plan/strategy) for agent to use in environment
  - Policy π determines how the agent chooses actions
  - π: S→A, mapping from states to actions
- Deterministic policy:
  - $\pi(s) = a$
- Note you could instead have a stochastic policy

Q-Learning

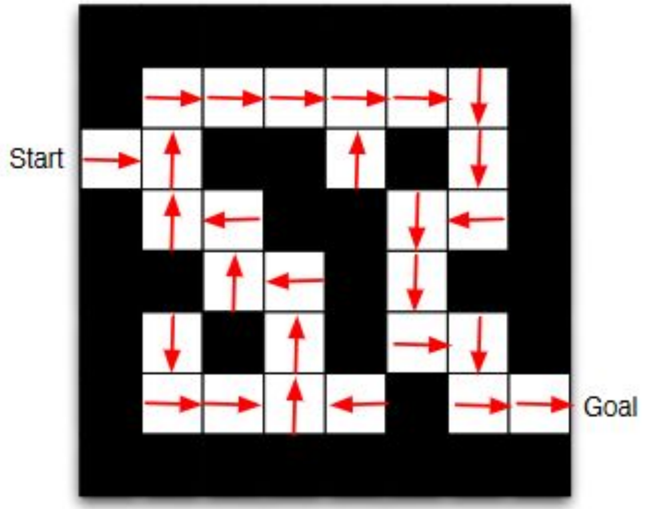
- Q-learning is a model-free based algorithm
  - Model-based
    - Explicit: Model
    - May or may not have a policy and/or value function
  - Model-free
    - Explicit: Value function and/or policy function
    - No model

Maze Example



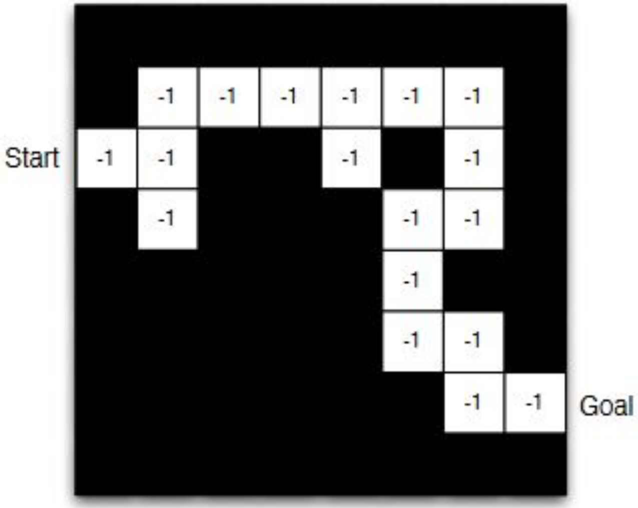
- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's Location

Maze Example - Policy



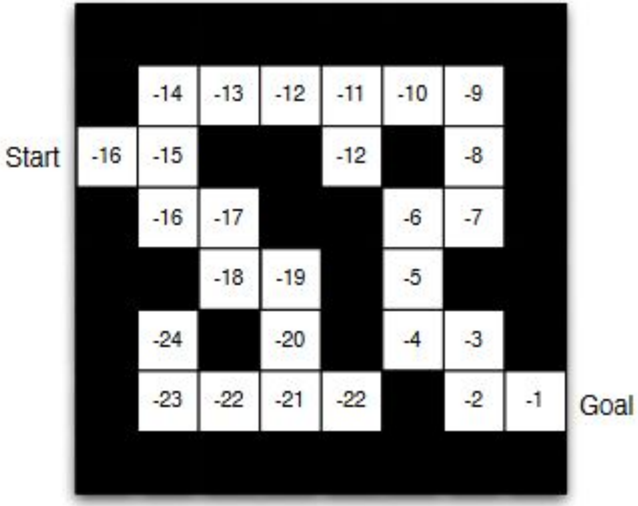
- Arrows represent policy $\pi(s)$ for each state s

Maze Example - Model



- Grid layout represents transition model
- Numbers represent immediate reward for each state s
  - Model-based approach

Maze Example: Value Function



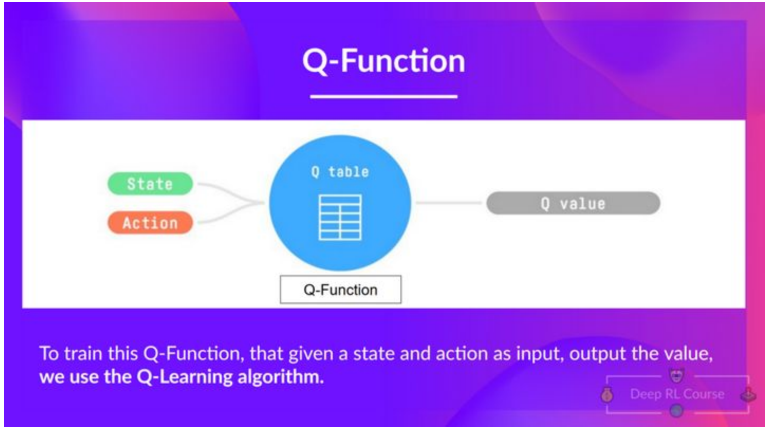
- Numbers represent value v_π(s) of each state s
  - Model-free approach

Quickly, On and Off-Policy Learning

- Off-policy
    - Learn to estimate and evaluate a policy using experience gathered from following a different policy
- On-policy
    - Direct experience
    - Learn to estimate and evaluate a policy form experience obtained from following that policy

Q-Learning Algorithm

- Q-learning is an off-policy method for training a Q-function
  - Finds policy indirectly by training a value or action-value function that will tell us the value of each state or each state-action pair



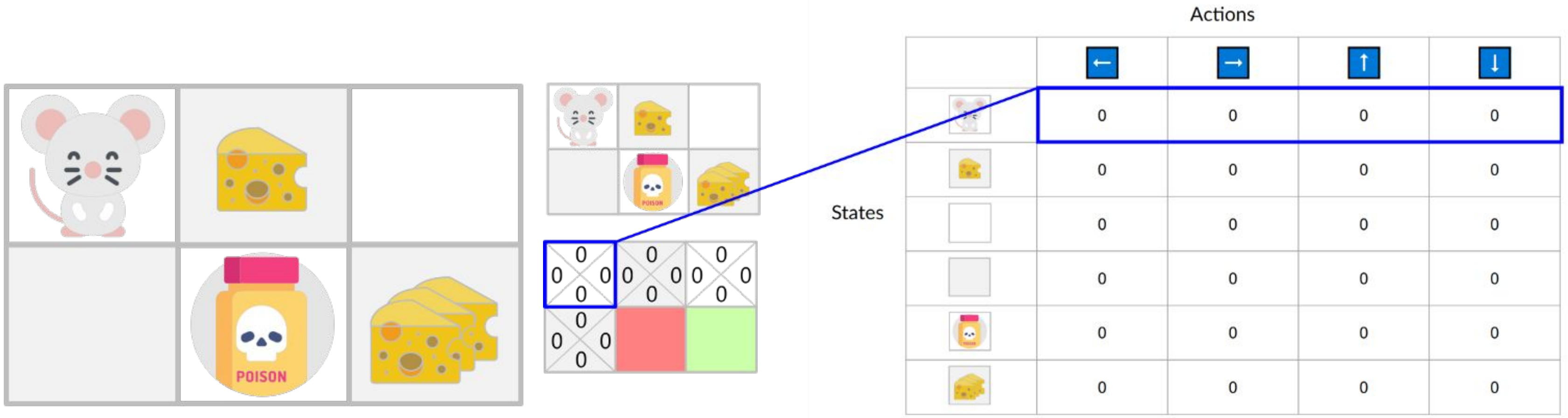
Given a state and action, our Q-function outputs a state-action value (or Q-value)

$$\underbrace{Q(S_t, A_t)}_{\text{New Q-value estimation}} \leftarrow \underbrace{Q(S_t, A_t)}_{\text{Former Q-value estimation}} + \underbrace{\alpha}_{\text{Learning Rate}}[\underbrace{R_{t+1}}_{\text{Immediate Reward}} + \underbrace{\gamma max_a Q(S_{t+1}, a)}_{\text{Discounted Estimate optimal Q-value of next state}} - \underbrace{Q(S_t, A_t)}_{\text{Former Q-value estimation}}]$$

Hugging face's deep RL course: https://github.com/huggingface/deep-rl-class

Q-Table

- Our Q-function has a Q-table where each cell corresponds to a state-action value pair value.



Hugging face's deep RL course: https://github.com/huggingface/deep-rl-class

Q-Table

- Initially, the Q-table is useless as it’s arbitrarily initialized.
- Need to explore to update it!

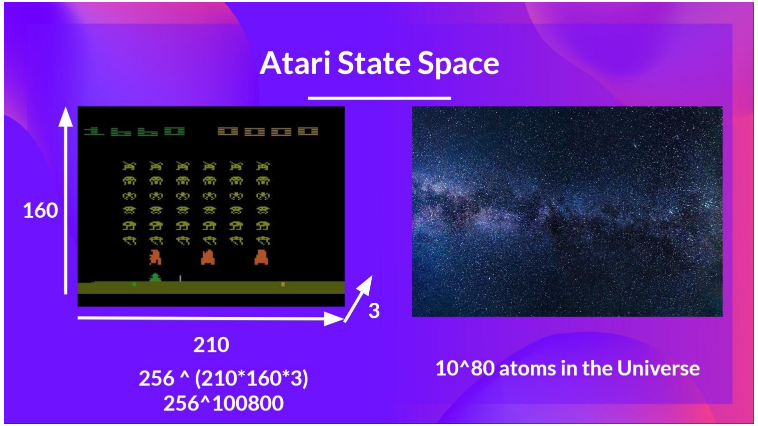
	←	→	↑	↓
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0

	←	→	↑	↓
	0	10.8	0	0
	0	9.9	0	-10
	0	0	0	10
	-10	0	0	0
	0	0	0	0
	0	0	0	0

Hugging face’s deep RL course: https://github.com/huggingface/deep-rl-class

Problem with (classical) Q-Learning

- Q-learning is a tabular method
- Internally, our Q-function has **a Q-table, a table where each cell corresponds to a state-action pair value.** Think of this Q-table as **the memory or cheat sheet of our Q-function.**



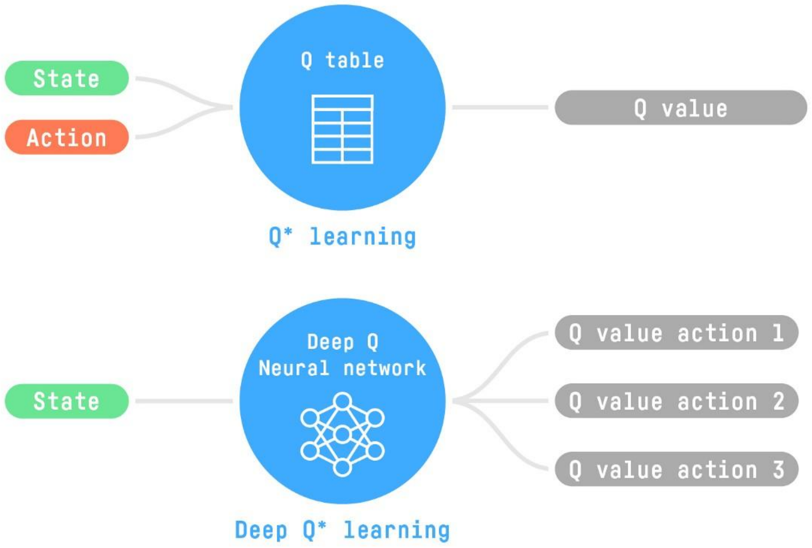
Difficult
to scale!

Hugging face's deep RL course: https://github.com/huggingface/deep-rl-class

Deep Q-Learning

- **Learn** to approximate Q-values

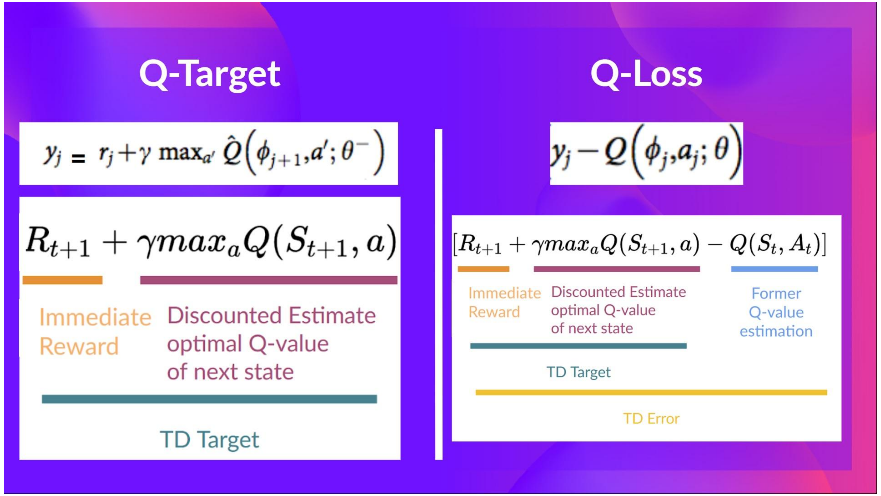
Original DQN Paper:
[Human-level control through deep reinforcement learning]



Hugging face's deep RL course: https://github.com/huggingface/deep-rl-class

Deep Q-Learning

- Deep Q-learning uses a deep neural network to approximate different Q-values for each possible action at a state (value estimation)



Hugging face’s deep RL course: https://github.com/huggingface/deep-rl-class

Let’s now discuss project 4!

Introduction

- This project will explore regularity discovery using PRML techniques.

- You will investigate an existing regularity-related method, work it a baseline method, and then further explore the topic

- Release Date: Tuesday, March 21

- Submission Due Date: **3, April 2023**

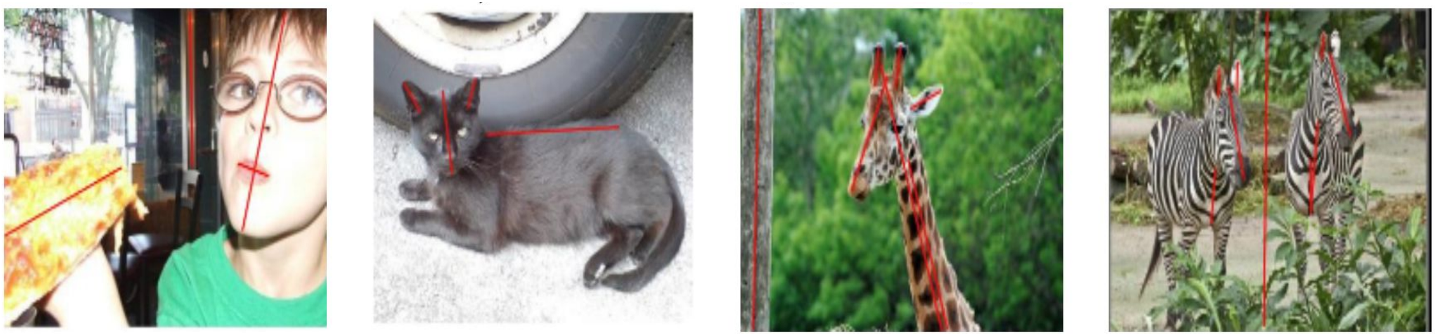
Regularity Discovery Topics

- You will choose to investigate one of six regularity-related topics from real data
  - Reflection Symmetry Detection
  - Rotation Symmetry Detection
  - Translation Symmetry Detection
  - Glide-Reflection Symmetry Detection
  - Recurring Pattern Detection
  - Reinforcement Learning-Based Regularity Understanding

- For each method you will...

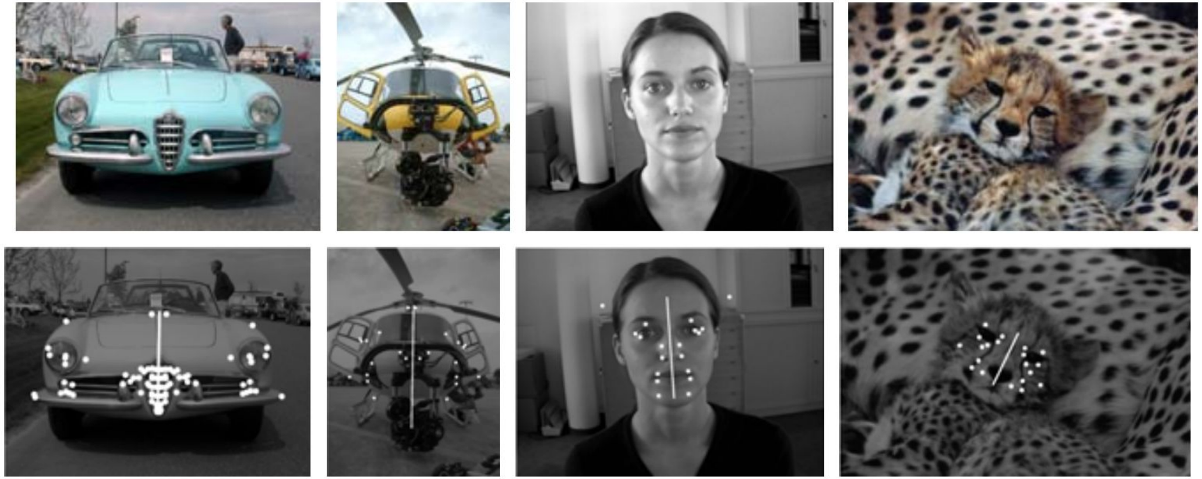
Reflection Symmetry Detection

The goal of reflection symmetry detection aims to discover a reflection axis/plane that remains invariant under reflection.



Baseline Method

Loy, Gareth, and Jan-Olof Eklundh. "Detecting symmetry and symmetric constellations of features." Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9. Springer Berlin Heidelberg, 2006.

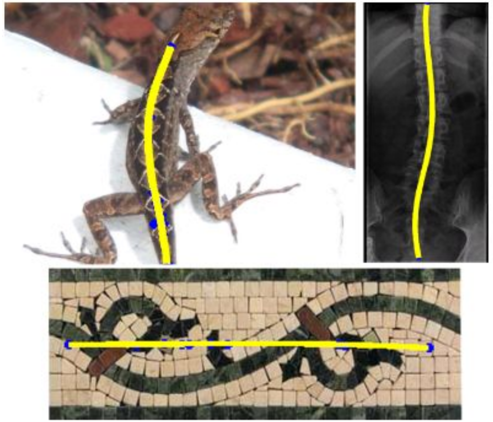


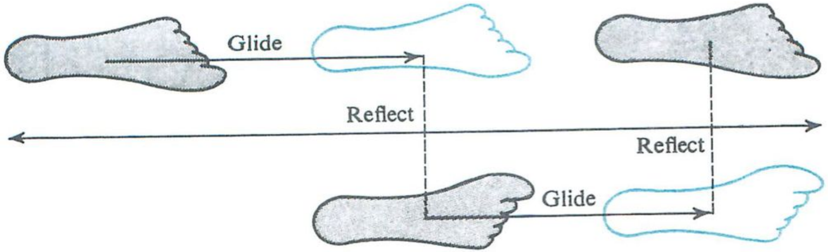
Methods for Investigation

Seo, Ahyun, Byungjin Kim, Suha Kwak, and Minsu Cho. "Reflection and Rotation Symmetry Detection via Equivariant Learning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9539-9548. 2022.

Glide-Reflection Symmetry Detection

Glide-reflection is another symmetry primitive. Glide-reflection is a combination of two symmetries that consists of an axis of reflection as well as translation along that axis.

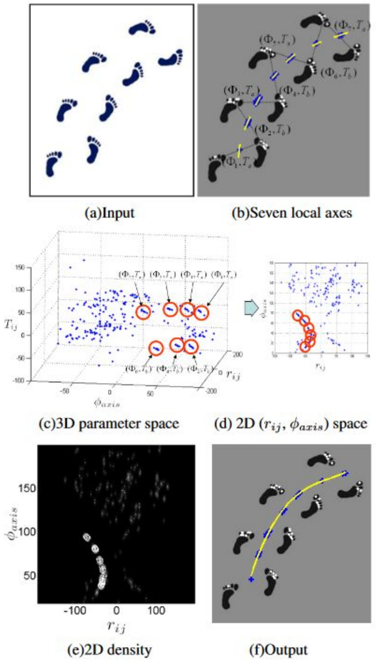




Lee, Seungky, and Yanxi, Liu. "Curved Glide-Reflection Symmetry Detection". Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011.

Baseline Method

Lee, Seungky, and Yanxi, Liu. "Curved Glide-Reflection Symmetry Detection". Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011.



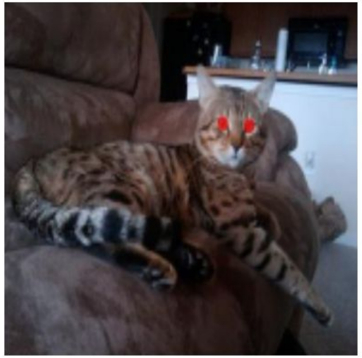
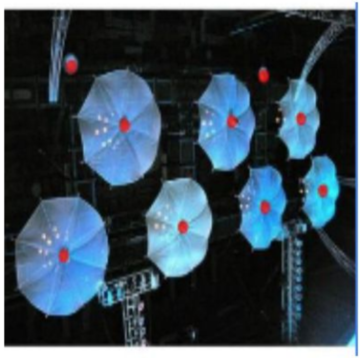
Methods for Investigation

Seo, Ahyun, Byungjin Kim, Suha Kwak, and Minsu Cho. "Reflection and Rotation Symmetry Detection via Equivariant Learning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9539-9548. 2022.

Rotation Symmetry Detection

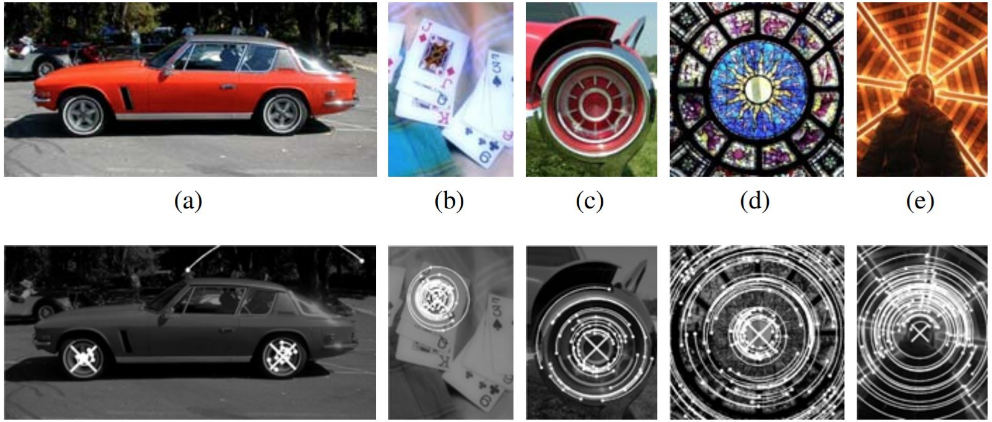
The goal of rotation symmetry detection is to find a rotation center invariant of rotation.



Baseline Method

Loy, Gareth, and Jan-Olof Eklundh. "Detecting symmetry and symmetric constellations of features." Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9. Springer Berlin Heidelberg, 2006.

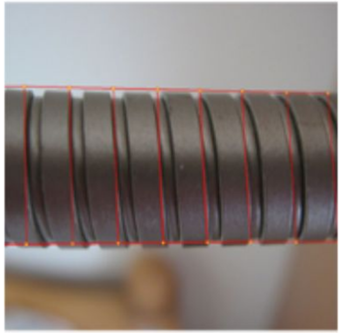
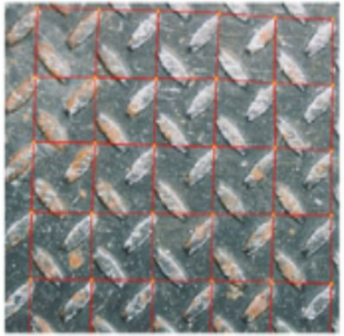


Methods for Investigation

Seo, Ahyun, Byungjin Kim, Suha Kwak, and Minsu Cho. "Reflection and Rotation Symmetry Detection via Equivariant Learning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9539-9548. 2022.

Translation Symmetry Detection

There are two types of translation (1) Translation with repetition in one direction, also known as **Frieze Pattern**. (2) Translation with repetition in two directions, also known as **Wallpaper Pattern**.

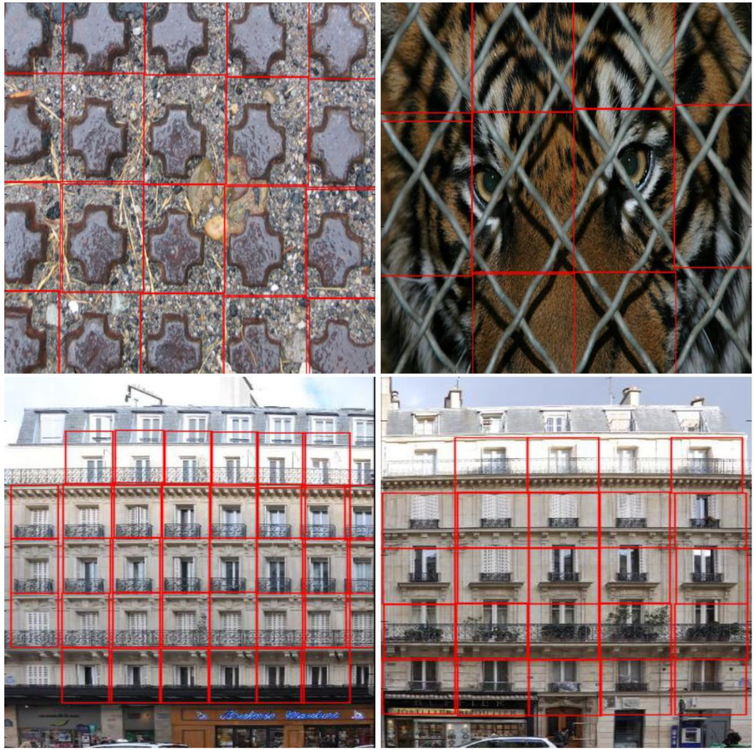

Baseline Method

- Leordeanu, Marius, et al. "Discovering texture regularity as a higher-order correspondence problem." (2006).
- Park, Minwoo, et al. "Translation-symmetry-based perceptual grouping with applications to urban scenes." Computer Vision–ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III 10. Springer Berlin Heidelberg, 2011.

Fig. 4. From left to right: input, final lattice, warped texture, and extracted texels.

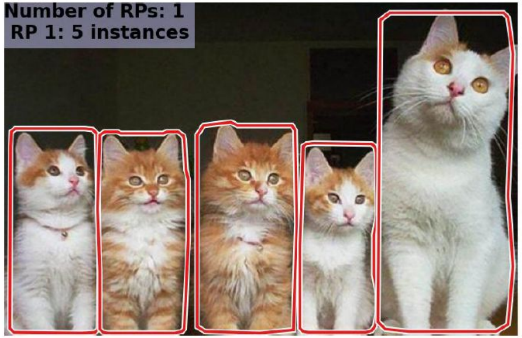
Methods for Investigation

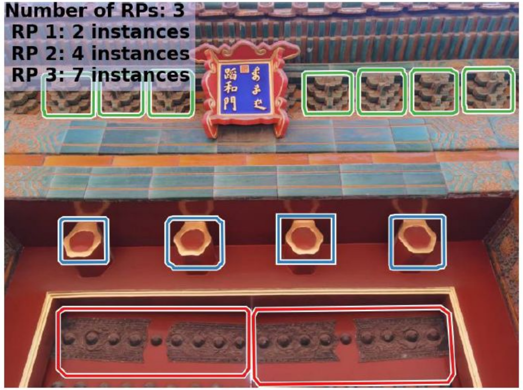
Rodriguez-Pardo, Carlos, et al. "Automatic extraction and synthesis of regular repeatable patterns." Computers \& Graphics 83 (2019): 33-41.



Recurring Pattern Discovery

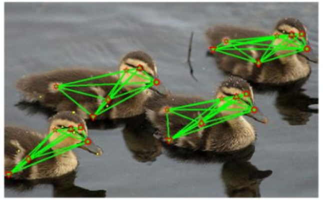
Recurring pattern (RP) detection refers to the detection of ”things that recur” in images, ranging from perfect symmetry to real-world similar objects.

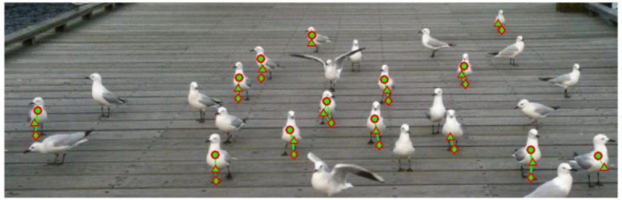


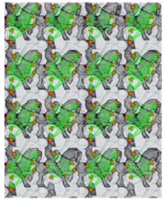


Baseline Method

Liu, Jingchen, and Yanxi Liu. "Grasp recurring patterns from a single view." Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.



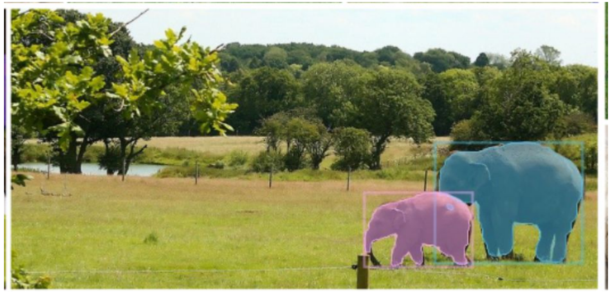


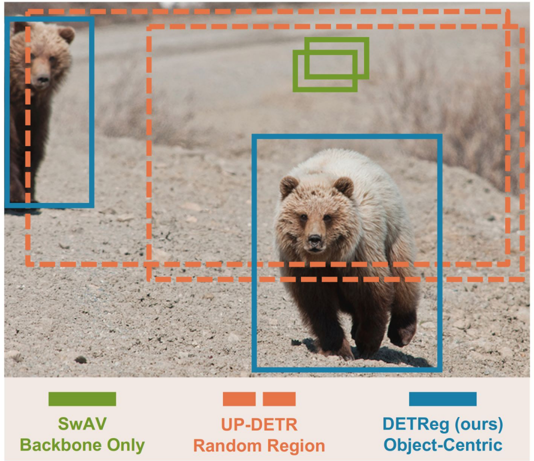


Methods for Investigation

Unsupervised Object Detection:

- Wang, Xudong, et al. "Cut and Learn for Unsupervised Object Detection and Instance Segmentation." arXiv preprint arXiv:2301.11320 (2023).
- Bar, Amir, et al. "Detreg: Unsupervised pretraining with region priors for object detection." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

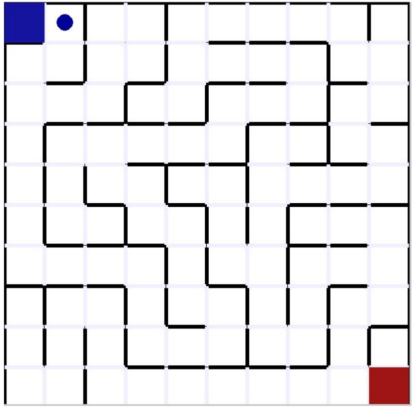




Regularity in Reinforcement Learning

We will be working with Q-learning, where an agent attempts to learn an optimal policy when acting in an environment.

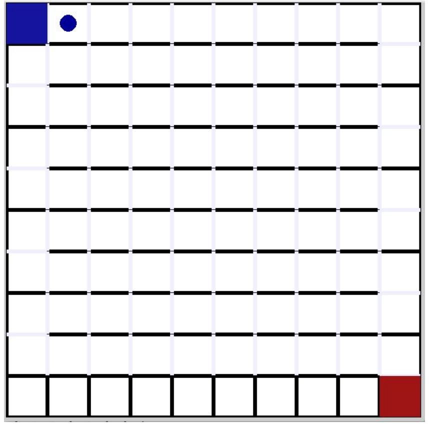
"Normal maze"



How does an agent perform

in a maze with regularity?

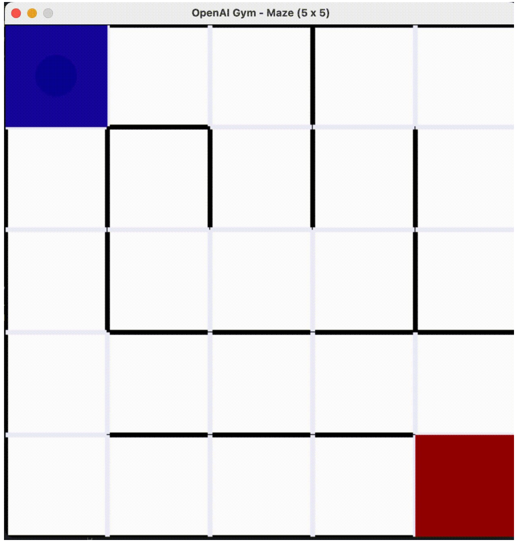
Maze with regular elements



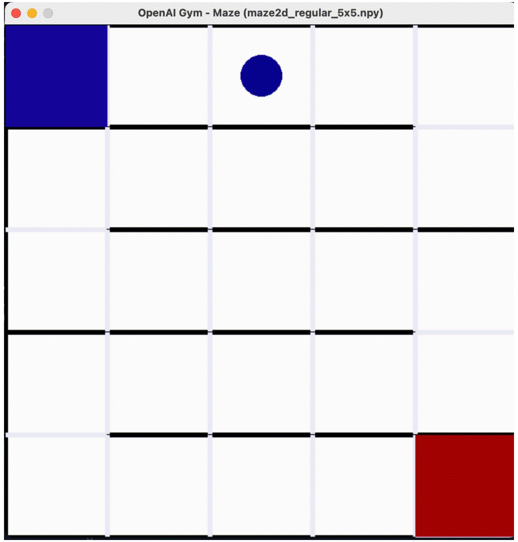
Regularity in RL

- We will be evaluating agent performance on
  - Random/non-regular maps
  - Maps with regular elements

Random Layout



Regular Layout



Grading Criteria

- Part one (all methods):
  - Summarize the baseline
  - Try it out
  - Report on it
- Part two (option **one**) Symmetry or RP detection:
  - Apply new/different method
  - Report on it and compare it
- Part two (option **two**) RL:
  - Compare classical Q-learning to deep Q-learning
  - Measure performance on maps with regular elements
  - Report on results

Q & A