A Reinforcement Learning approach to Computer Vision problems

Rayan Samy Ramoul
AI Engineer

Quant AI Lab
Limitless Solution

17/10/2023

SICARA

PLAN

01. Why RL for CV

02. Reminders on RL

03. Deep RL

04. Applications

05. Observations & Challenges

06. Recent Advances

01.

Why Reinforcement Learning for CV?

Why Reinforcement Learning for Computer Vision ?

- Reinforcement learning augments the reinforcement learning framework and utilizes the powerful representation of deep neural networks.
- RL demonstrated remarkable successes in various domains including finance, healthcare, games, robotics, and computer vision.

OpenAI

Reinforcement performs well in other problematics.
Would it work on Computer Vision ?

DeepMind

02.

Reminders on RL

Foundations of reinforcement learning : Markov Decision Process

Environnement

Set of finite states *S*

Actions

Finite set of actions *A*

Probabilities of transitions and "policy"

$$P^{a}_{s's} = P(s_{t+1} = s' | s_t = s, A_t = a)$$

$$\pi(a|s) = P[A_t = a | S_t = s]$$

Reward

$$R : S \times A \rightarrow R$$

Return expectation

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Value function

$$Q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]$$

And then ?

When faced with the challenge of processing complex image data, traditional methods, including Reinforcement Learning (RL), often struggle to extract meaningful features, necessitating the integration of Deep Reinforcement Learning (DRL) to effectively navigate and make decisions in image-rich environments, as demonstrated in tasks such as autonomous driving and medical image analysis.

03.

Deep RL

Deep Reinforcement Learning

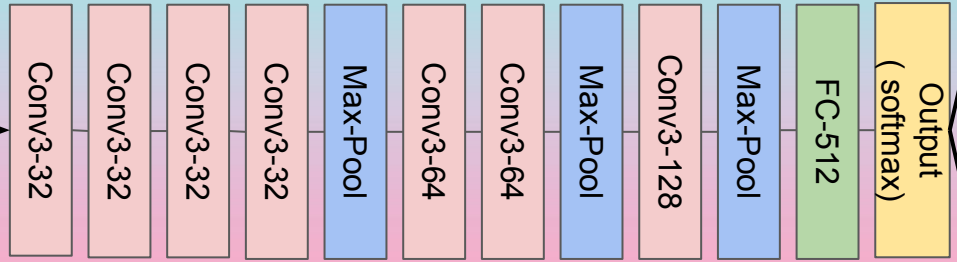
Deep RL = Reinforcement Learning + Deep neural networks.

It enables agents to learn complex behaviors and make decisions in high-dimensional state spaces. Deep RL has gained immense popularity due to its success in solving a wide range of challenging tasks.

Deep-Q-Learning (a DeepMind algorithm)

Environmental state representation

Q value Estimator



Neural network called "Q-Network" with convolutional layers for feature extraction and fully connected layers with softmax output

P = 0.489

P = 0.511

Network output after softmax: probability that each action is the best possible.

04.

Applications

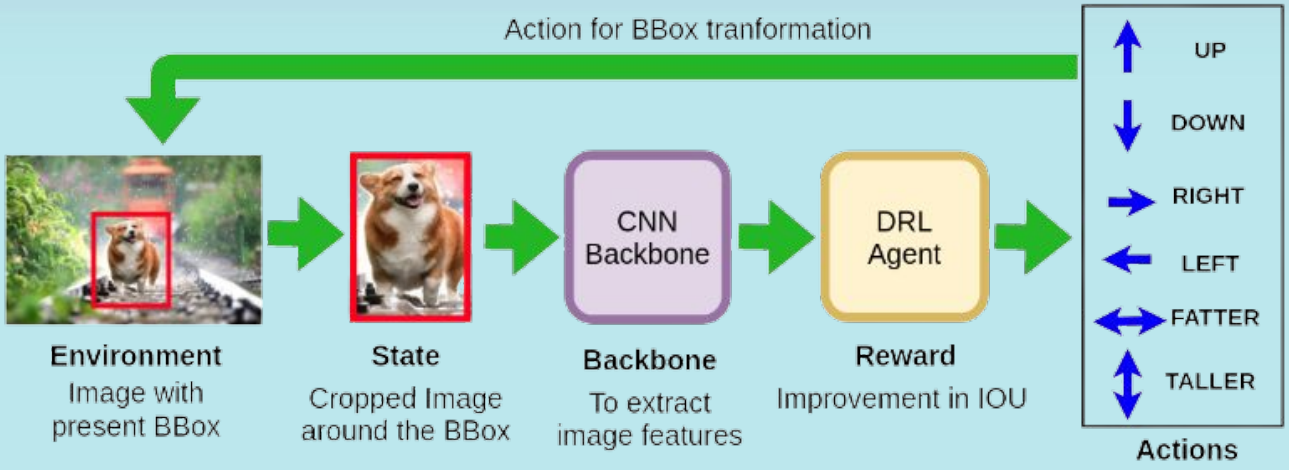
- Object Localization
- Object Tracking
- Landmark Detection
- Image Segmentation
- Image Registration
- Video Analysis

Object Localization

Object Localization

Active Object Localization with Deep Reinforcement Learning

- Use of the Replay-Memory algorithm (history sampling, periodic duplication of the q-network).
- Expert agent to optimize exploration during learning.
- Hyper-parameterization decisive for model efficiency.



Environnement : Bounding Box

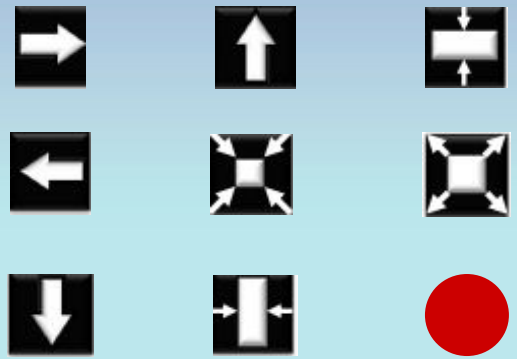
(x_{min}, y_{min})

(x_{max}, y_{max})

Actions : translations of bounding box

$$\alpha_w = \alpha.(x_{max} - x_{min})$$

$$\alpha_h = \alpha.(y_{max} - y_{min})$$



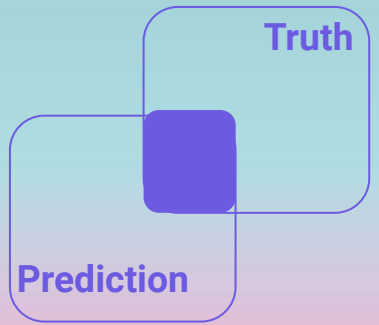
Datasets used :
Pascal VOC 2007 + 2012

Rewards

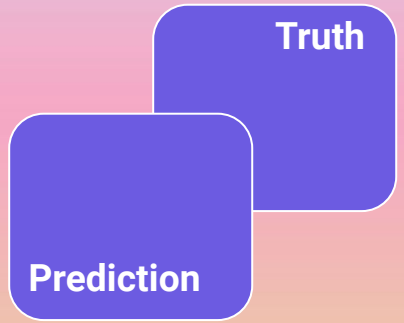
$$R_\omega(s, s') = \begin{cases} +\eta & \text{if } IoU(b, g) \geq \tau \\ -\eta & \text{otherwise} \end{cases}$$

Evaluation Metric for the Reward

Intersection



Union

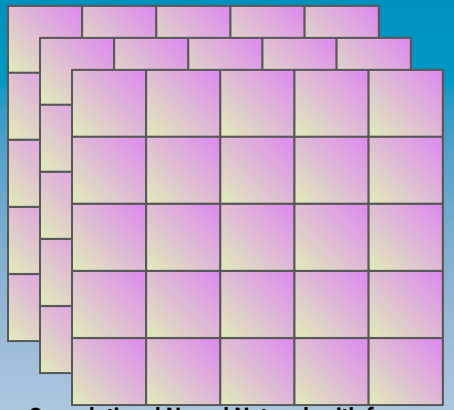


+1, *If IoU > threshold*
-1 *else*

$$AP = \sum_{n=1}^{datasetsize} I_{IoU(n)>threshold)}$$

Resize

Features Extraction



Convolutional Neural Network with frozen weights

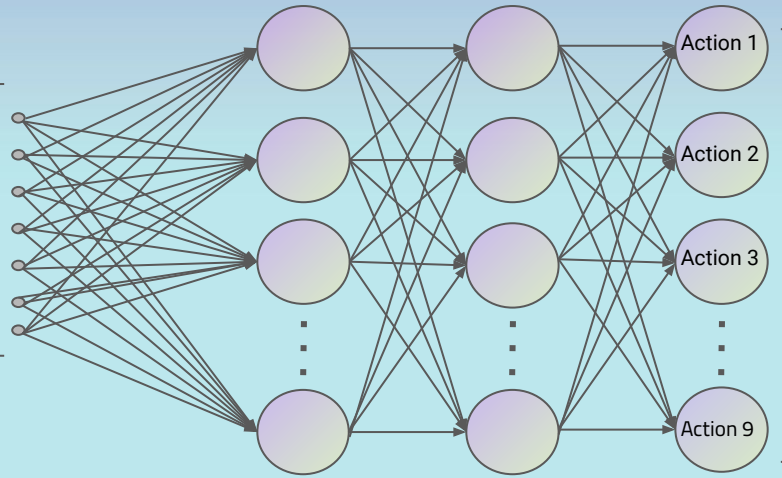
Feature Vector

History of 9 previous actions in one-hot

001000000
000010000
100000000
.
.
.
000100000
000100000
000000100

Feature Vector

X

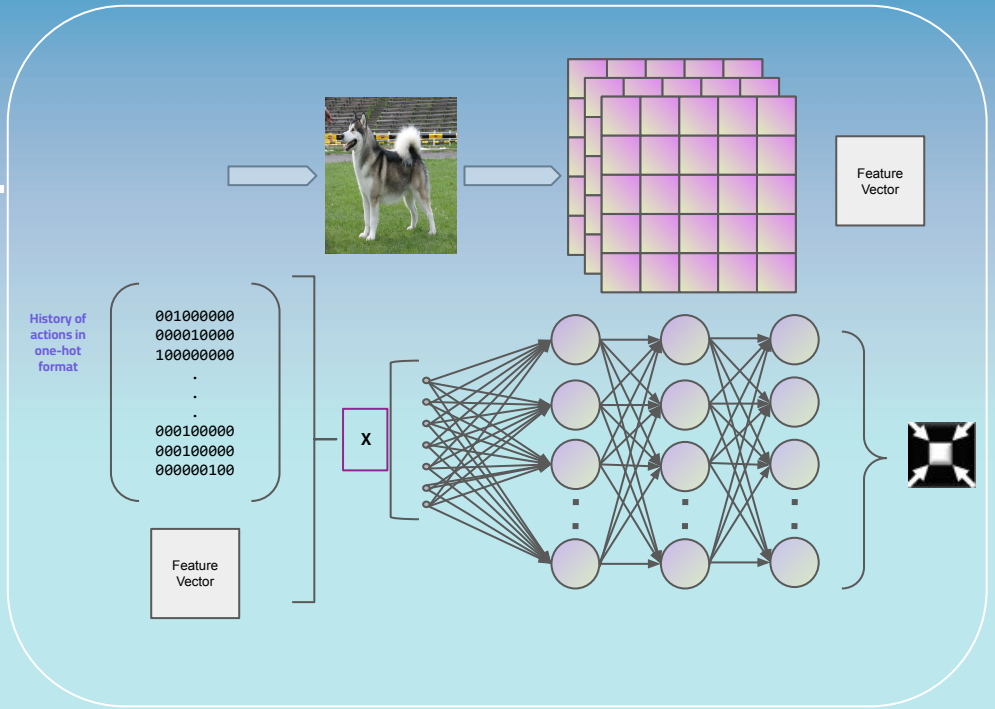


Max (output) = chosen action

The collected rewards are backpropagated through the network.

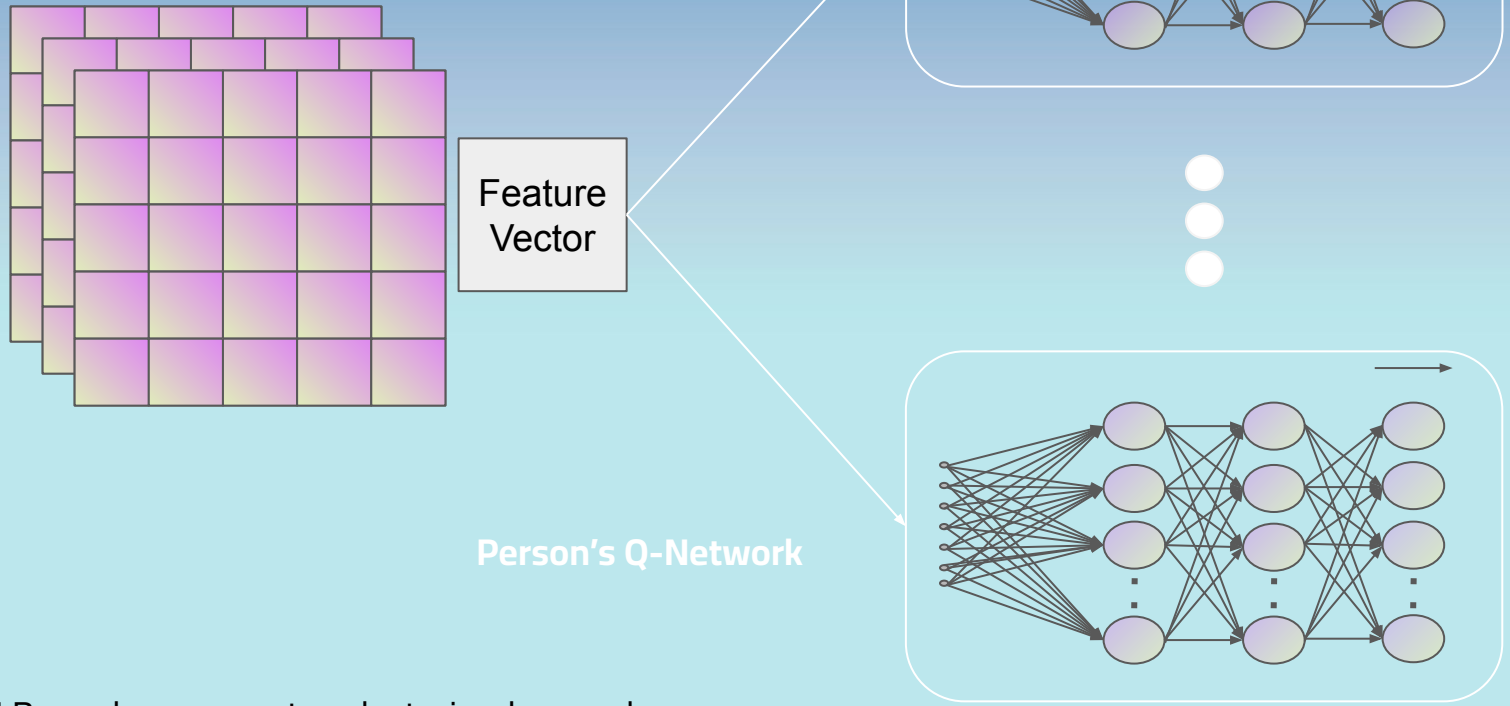
X (81+25088) | Layer 1 (1024) | Layer 2 (1024) | Output Layer (9)

Stopping at :
Or after 40 iterations.



End-to-End Process

A network per class



N.B : n-class = n-networks trained on each

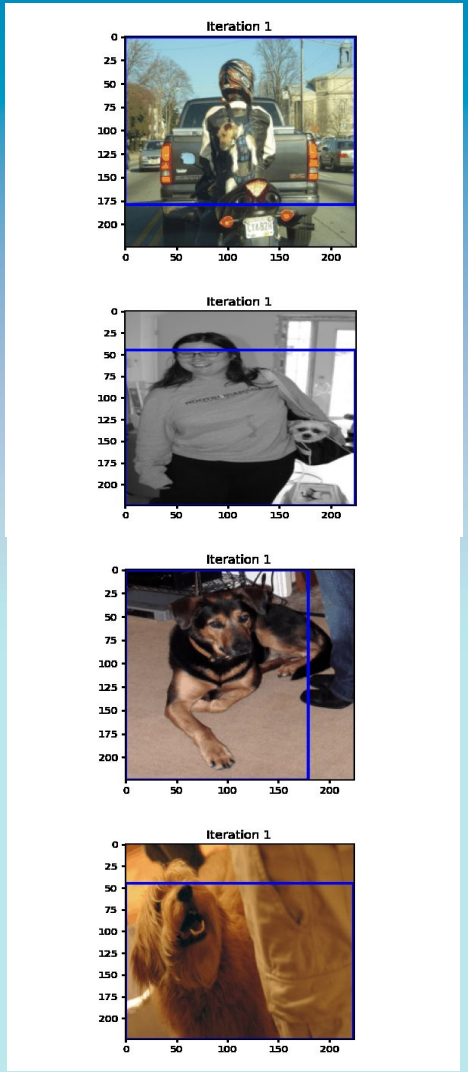
Benchmark vs State-of-the-art (at time)

Improvements

- Speed of inference.
- Processing of images with occlusion and truncation.
- Better in performance than approaches without object proposal.

Disadvantages

- 1 to 2% less accurate globally than Faster RCNN.

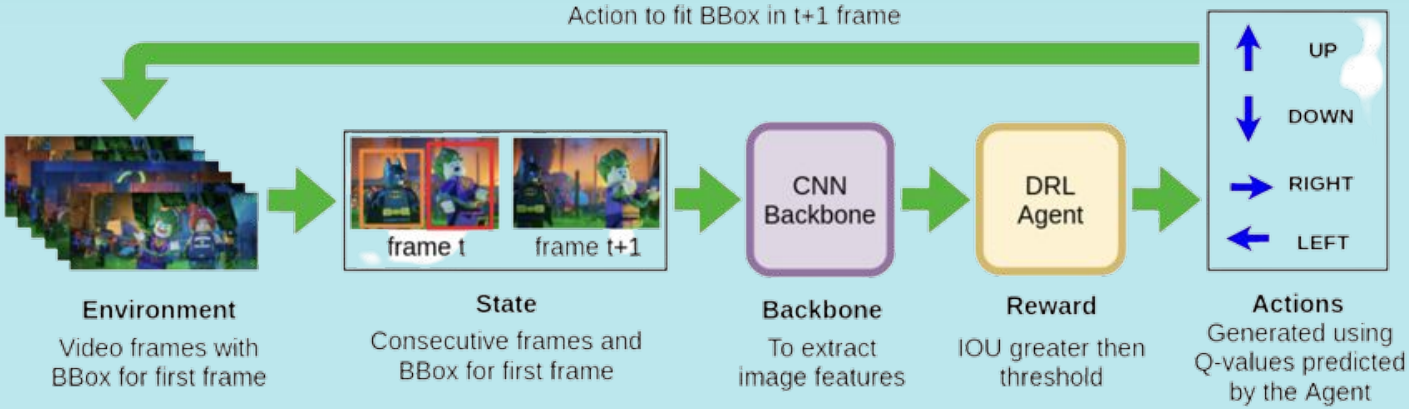


Object Tracking

Object Tracking

Multi-agent deep reinforcement learning for multi-object tracker

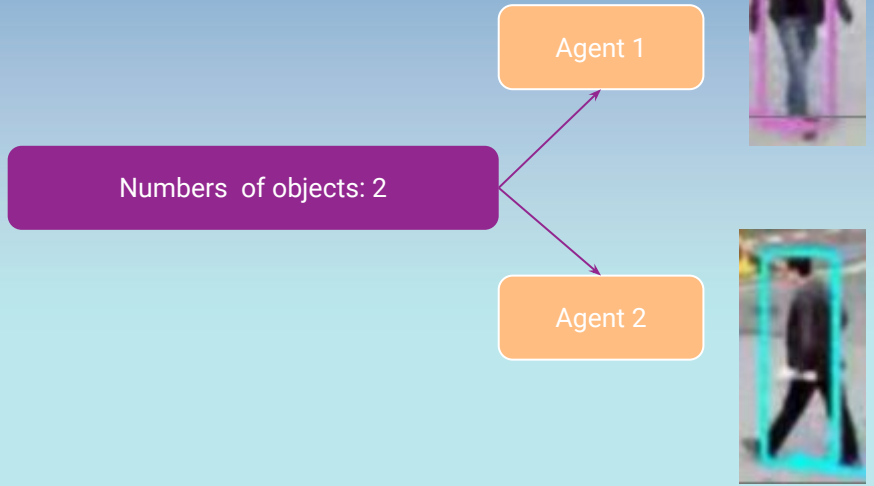
- Object tracking in computer vision is the process of continuously locating and following objects in video sequences, often involving object detection, feature extraction, and motion prediction. It is used in applications like surveillance, autonomous vehicles, and robotics to monitor and interact with objects in real-time.
- Using two consecutive frames (Ft, Ft+1) and a bounding box the agent adjust the bounding box to fit the object in frame Ft+1.



Object Tracking

First Frame of the video

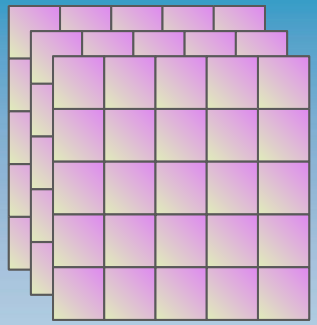
Object Localization using YOLO



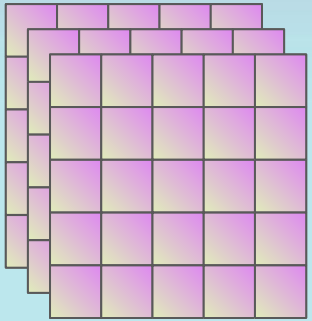
Create an agent associated to each bounding box

Bounding Box from F_t-1

Frame F_t

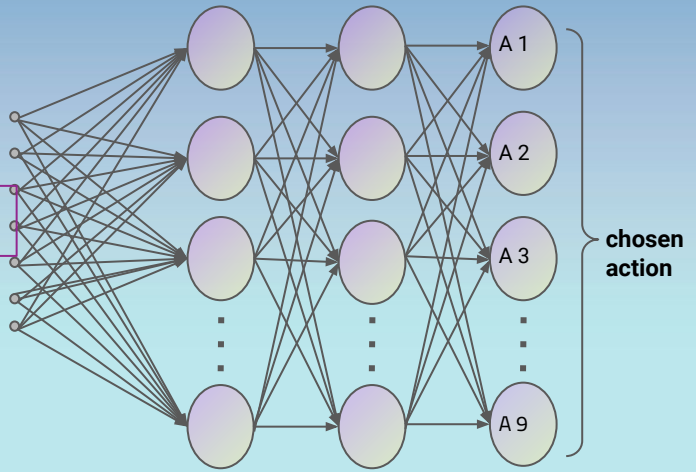


Feature Vector



Feature Vector

X



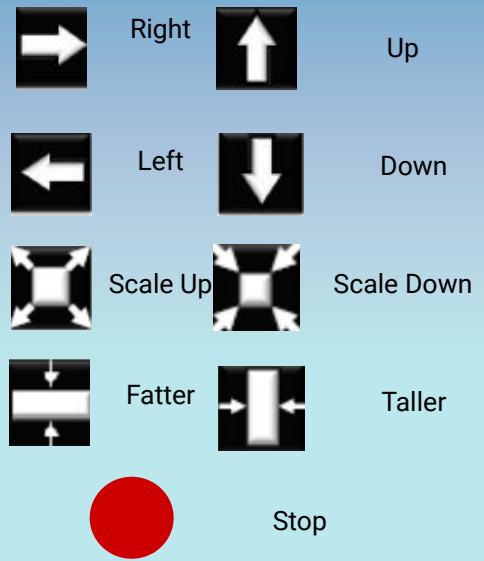
chosen action

State Representation

For each agent

For each frame F_t

Actions : translations of the bounding box



Rewards

$$r_T = \begin{cases} 1 & if\ IoU(p_T, g) > \tau \\ -1 & otherwise \end{cases}$$

Based on IOU (Intersection over Union) exceeding a specified threshold.

Benchmark vs State-of-the-art (at time)

Improvements

- **Speed** of inference.
- Powered by YOLOv3 effectiveness.
- Better performance in **robustness**.

Disadvantages

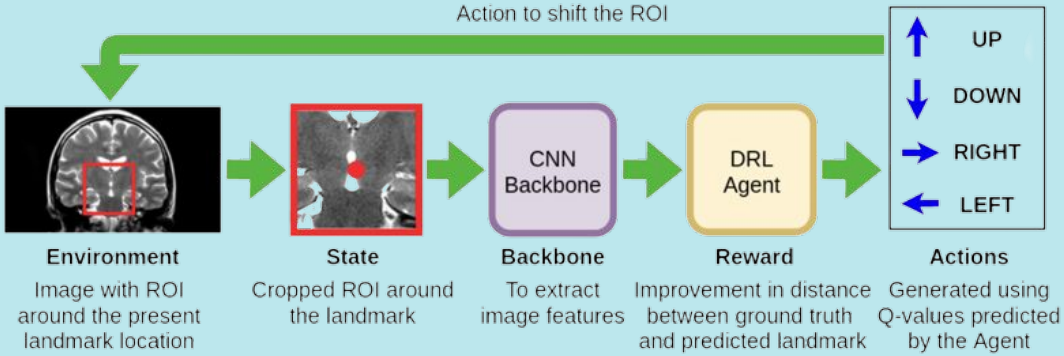
- Top 2 in mAP globally

Landmark Detection

Landmark Detection

Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans

- Landmark detection involves the localization of an object within a 3D acquisition, such as in the case of CT scans.
- Using the region of Interest (ROI) centered around the current landmark location cropped from the image the agent shifts the ROI across the image, forming a new state.



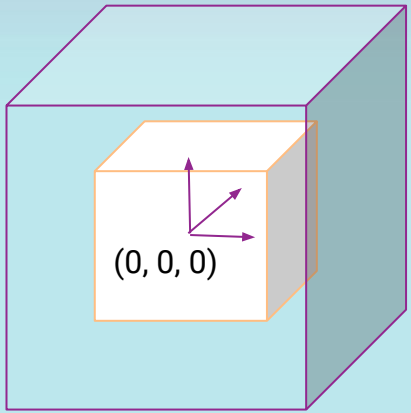
Rewards

Distance-based feedback, which is positive if the agent gets closer to the target structure and negative otherwise.

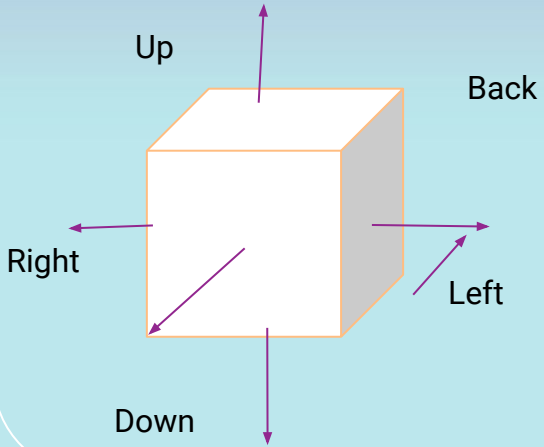
$$\mathcal{R}_{s,a}^{s'} = \|\vec{p}_t - \vec{p}_{GT}\|_2^2 - \|\vec{p}_{t+1} - \vec{p}_{GT}\|_2^2.$$

Environnement

3D system re-center on voxel position



Actions : moves of 3D box



Benchmark vs State-of-the-art (at time)

Improvements

- Achieves a **0% failure rate on all considered landmarks** and improves the average accuracy of reference methods by 20-30%.
- **Speed** of algorithm is 2**-3 orders of magnitude faster**, reaching real-time performance on high-resolution 3D-CT volumes
- **Robust** against outliers

Disadvantages

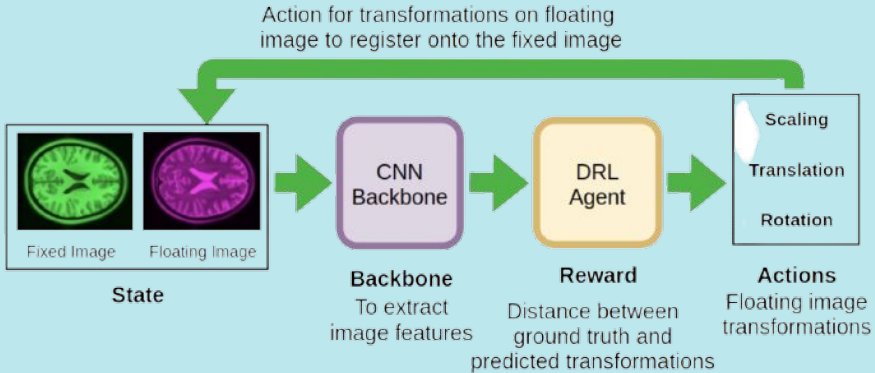
- Difficulties with **absent objects**.
- Difficulties with detection of **multiple objects.**

Image Segmentation

SeedNet

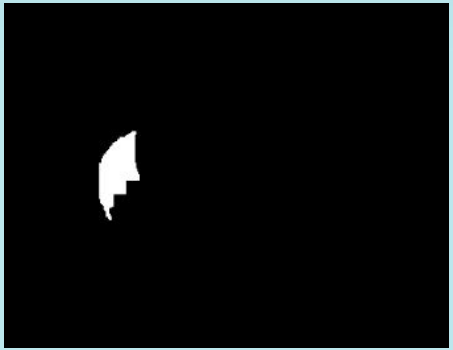
Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation

- In a medical context, image segmentation refers to the process of partitioning a medical image, such as an MRI or CT scan, into specific regions or structures, enabling precise identification and analysis of anatomical or pathological features. This technique is essential for tasks like organ delineation, tumor localization, and disease assessment in healthcare.
- Early work in medical image segmentation using Q-Matrix.
- The image and the segmentation mask are the input of the DQN. The seed set is updated
- Using the newly created seed from the DQN, and the mask is generated using the revised seed set. The obtained mask is used to calculate



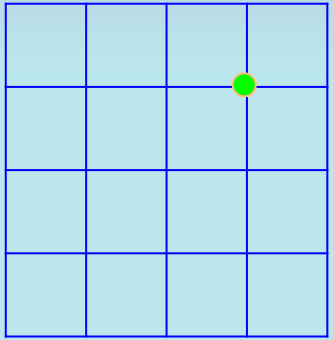
Environnement

Original Image + Segmentation Result



Actions : Seed point

- Action : is defined as a positioning new seed point.
- The agent decides the label (foreground/background) and position of the seed in the 2D grid given the states.
- Transform image to bigger grid to reduce action space.



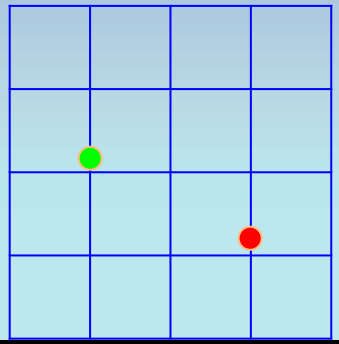
Rewards

Using the accuracy of the segmentation mask as a score concept. the reward value by comparing with the GT mask, and this process is repeated.

$$R_{\text{exp}} = \frac{exp^{k*IoU(M,G)} - 1}{exp^{k} - 1},$$

SeedNet

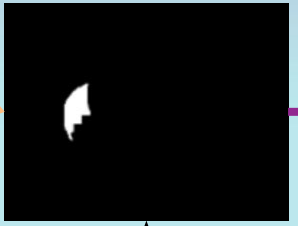
Image



Seed

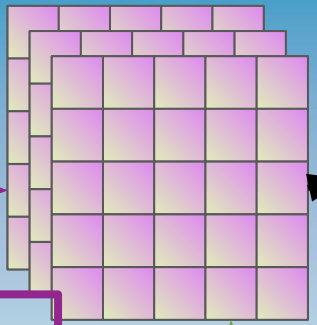
Ground Truth

Segmentation



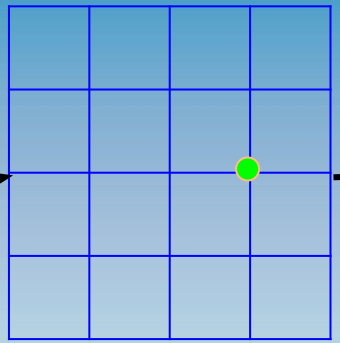
Observation Update

Q-Network



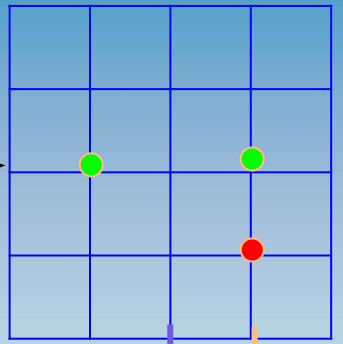
Reward

Seed Generation



New Segmentation

Seed Update



Compute Reward

Benchmark vs State-of-the-art (at time)

Improvements

- Achieved **improved** results compared to initial seed points.
- **Best seed generation algorithm** (to prepare for next segmentation algorithms)
- Compared with supervised methods (FCN and iFCN) and outperformed them.
- Improved ratio **pixel-wise labeling / computation**.

Disadvantages

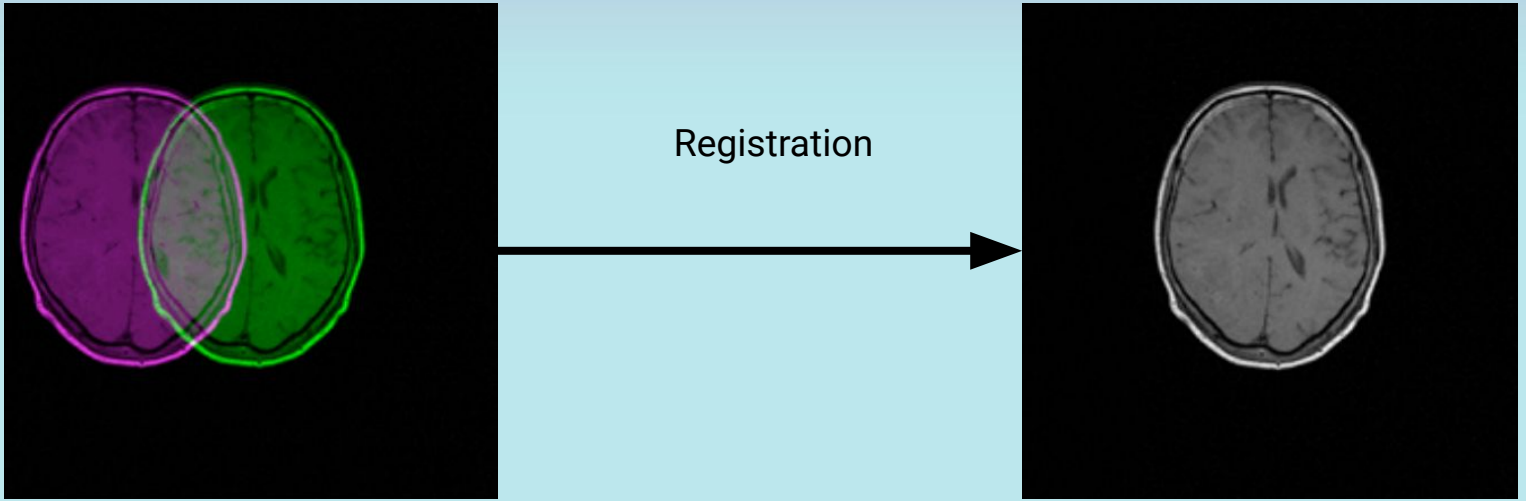
- Weaknesses on some specific datasets (lack of generalization).

Image Registration

Image Registration

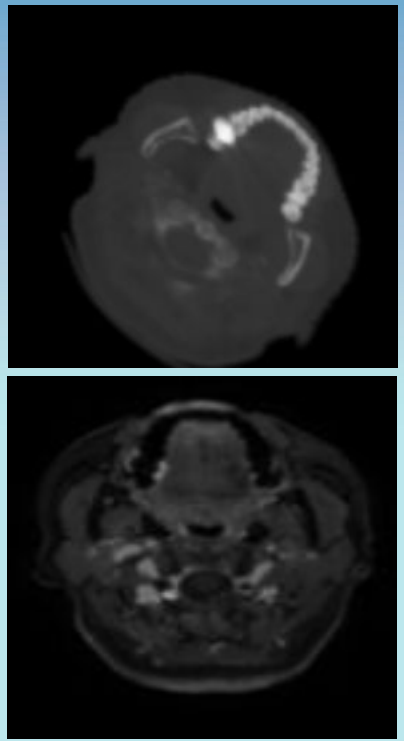
Robust multimodal image registration using deep recurrent reinforcement learning

- Image registration is a basic yet important pre-process in many applications such as remote sensing, computer-assisted surgery and medical image analysis and processing.
- In the context of brain registration, for instance, accurate alignment of the brain boundary and corresponding structures inside the brain such as hippocampus are crucial for monitoring brain cancer development



Environnement

Fixed Image + Image to Register



Actions : Rotations/Translations

The set of actions comprises all the rotation and resizing operations that can be performed on the object within the image.

→	t_x + 1	⤢	s - 0.05
↑	t_y + 1	↻	alpha - 1
←	t_x - 1	↺	alpha + 1
↓	t_y - 1	⤡	s + 0.05

Rewards

The Euclidean distance D between the transformed landmarks and the corresponding ground truth is used to define the reward for action.

$$r_t = -D = -\frac{1}{\#\left\{\mathbf{p}_G\right\}} \sum_i \left\| p_i - \tilde{p}_i \circ T_{t+1} \right\|_2, p_i \in \mathbf{p}_G, \tilde{p}_i \in \tilde{\mathbf{p}}_G$$

Where pi and ˜pi are the landmark points, ◦ denotes the align operator, # {} calculates the number of points. In addition, if D is smaller than a threshold, we assume that the terminal is triggered, and a terminal reward is set in this situation.

Benchmark vs State-of-the-art (at time)

Improvements

- **Best performance** at time.
- Results indicated that RL with all three components contribute to the method's performance.

Disadvantages

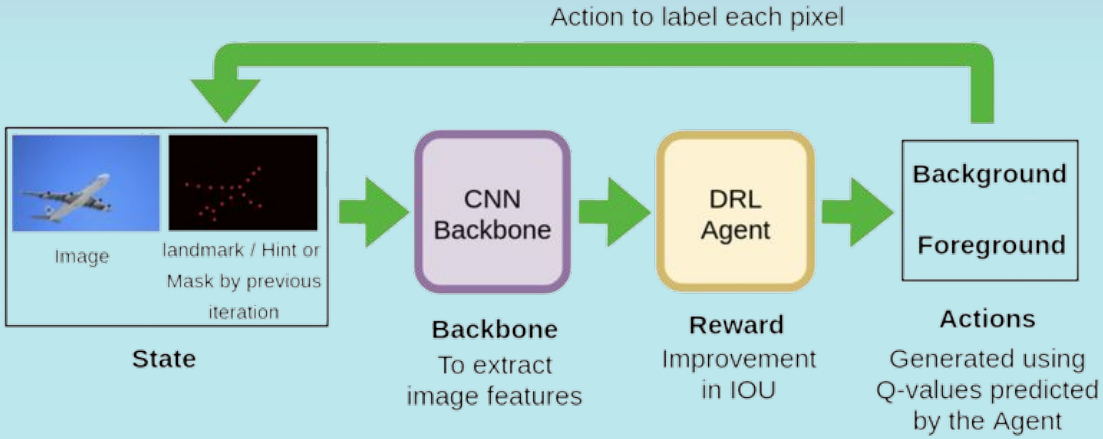
- **Weak scalability to 3D** image registration in cases of high dimensionality.
- Huge impact/sensitivity of reward function can have a significant impact on the method performance.
- No benchmark on images with **large deformations**.

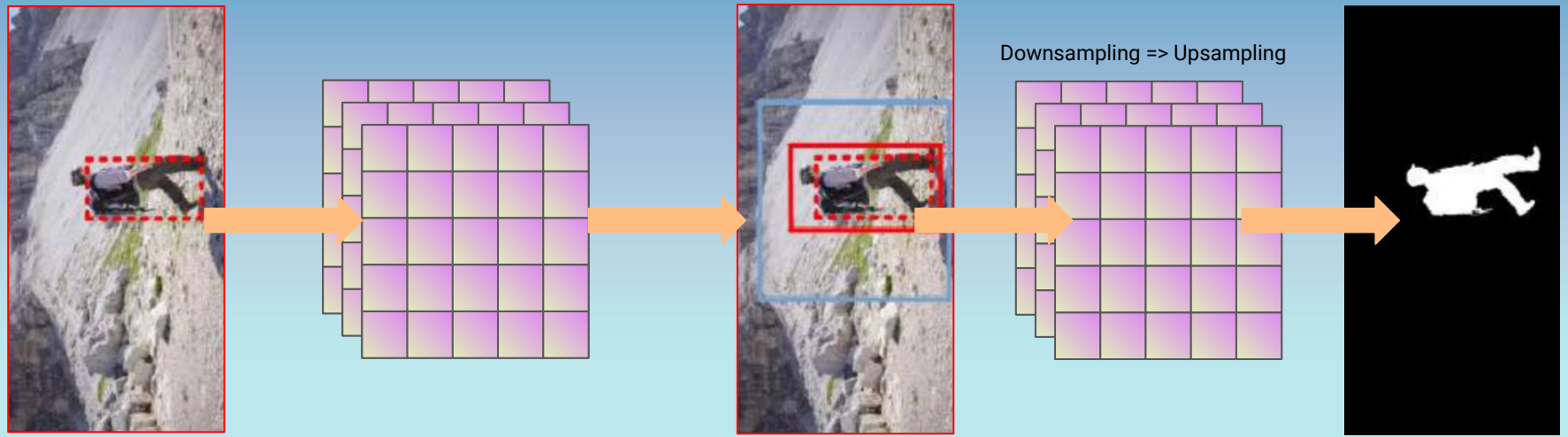
Video Analysis

Video object segmentation

Reinforcement Cutting Agent Learning

- **Video Object Segmentation:** It's a challenging computer vision task involving the segmentation of objects in video sequences.
- **Complexity:** Segmenting objects in videos is complex due to the continuous decision-making process, involving numerous agents (pixels or superpixels) and action steps.
- **Challenge:** The sheer volume of agents and actions required for video segmentation makes it a nontrivial problem.





Input Image

Cutting-Policy Network (CPN)

object-context box pair
1box foreground +
1 box object of interest

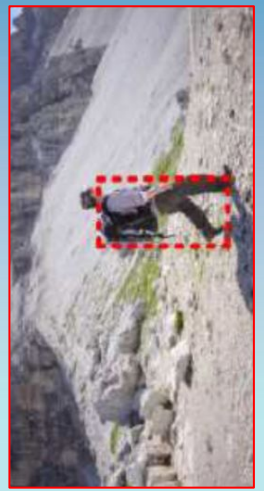
Cutting-Execution Network (CEN)

generating object masks based on the inferred object-context box pair. It involves learning segmentation-aware representations and discriminative functions to accurately separate the desired foreground object from the background.

Mask

For each frame F_t

Environnement

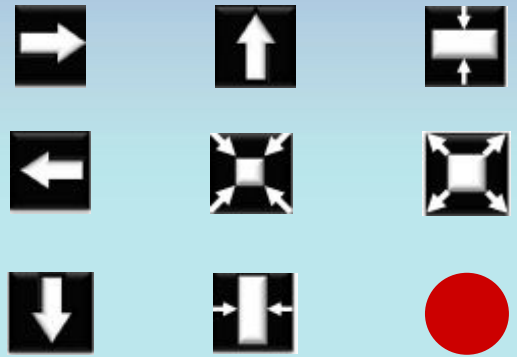


Input Frame

+

actions history

Actions : adjustments of bounding box



Rewards

Reflects the positive and negative variations of the segmentation mask

$$r(s_{t,k}, s_{t,k+1}) = \begin{cases} +\alpha \cdot 1, & \Delta > +0.1 \\ 10 \quad \alpha \quad \Delta, & -0.1 \leq \Delta \leq +0.1 \\ -\alpha \quad 1, & \Delta < -0.1 \end{cases}$$

where,

$$\Delta = IoU(m_{t,k+1}, y_t) - IoU(m_{t,k}, y_t)$$

$$\alpha = \begin{cases} 1, & a^o_{t,k} \neq stop \\ 3, & a^o_{t,k} = stop \end{cases}$$

Benchmark vs State-of-the-art (at time)

Improvements

- **Best stability through time.**
- **Effective even with** motion blur, occlusions, and appearance changes.

Disadvantages

- **Limited Dataset Evaluation.**
- **Segmentation Challenges Not Discussed.**

05.

Observations and Challenges

Observations

Does Reinforcement Learning in Computer Vision works ?

- Deep Reinforcement Learning (DRL) has proven its effectiveness in various computer vision applications, from landmark detection to video analysis with competitive results.
- It offers promising solutions to complex optimization problems such as parameter tuning and neural architecture search.
- Usually faster inference on most tasks.

Challenges

What difficulties are we facing when using it in computer vision ?

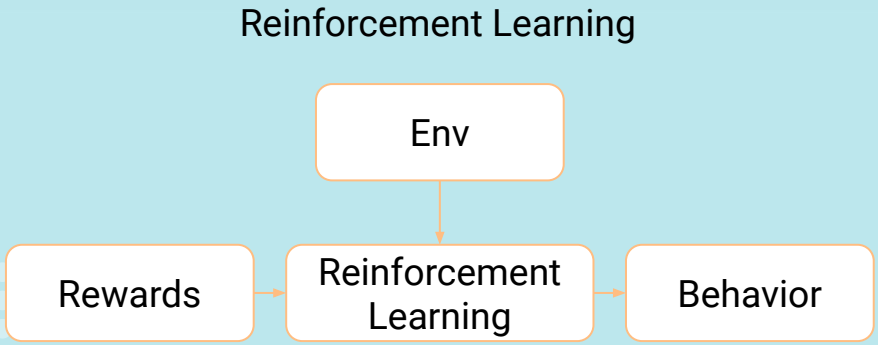
- Despite its successes, DRL faces significant challenges when applied to real-world computer vision systems.
- Defining a precise reward function often requires interdisciplinary knowledge and may not always be feasible.
- Handling continuous, high-dimensional state and action spaces remains a challenge, often necessitating discretization.
- Real-world environments are rarely stationary, posing challenges for DRL models built on assumptions of stability.
- Training DRL agents demands extensive data, which can be scarce in real-world scenarios.

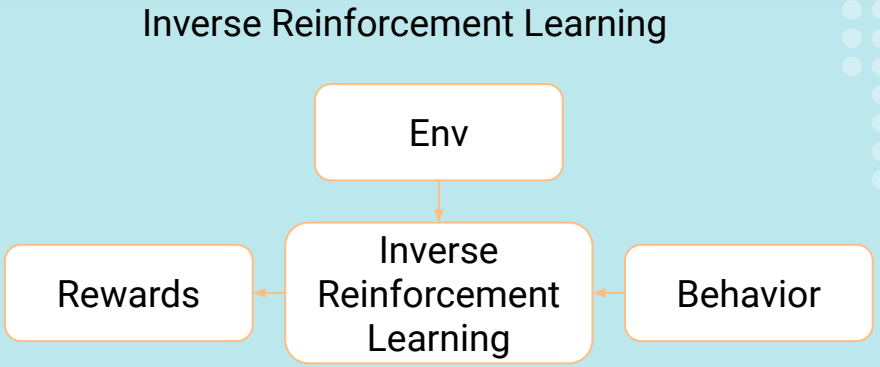
06.

Promising advances

Inverse Deep Reinforcement Learning

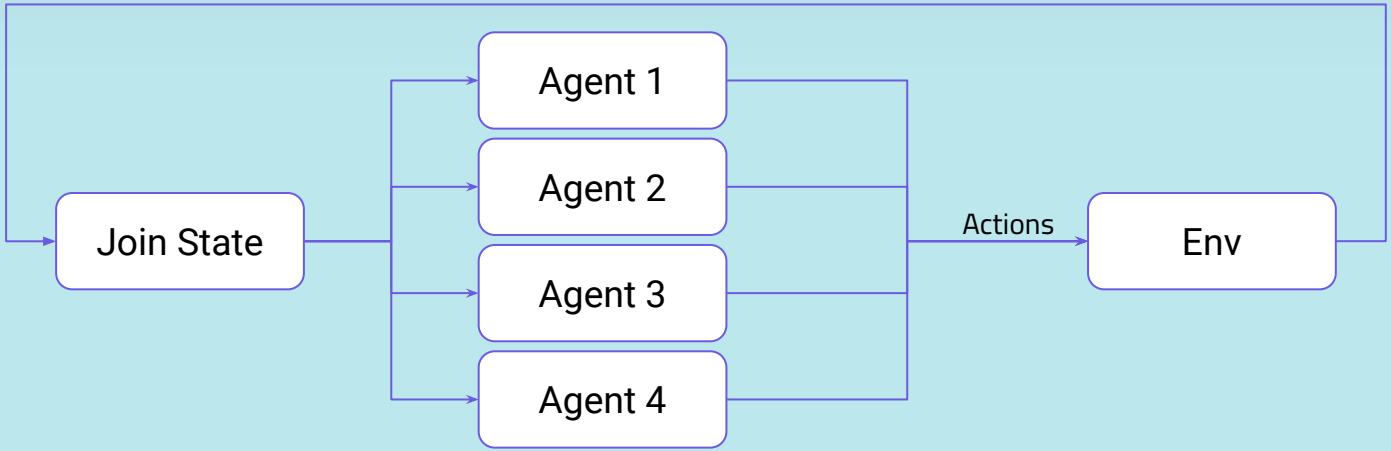
- Particularly useful in models like the Landmark Detection's one where reward is very impactful.
- Inverse DRL is an approach within Deep Reinforcement Learning.
- It differs from traditional DRL by inferring reward functions instead of relying on manually defined ones.
- Useful in scenarios like autonomous driving, where creating precise reward functions is challenging.
- Inverse DRL uses observed behavior to approximate rewards.
- Applied successfully in autonomous driving and complex movement analysis.
- Promising for tasks where traditional DRL struggles with reward specification.





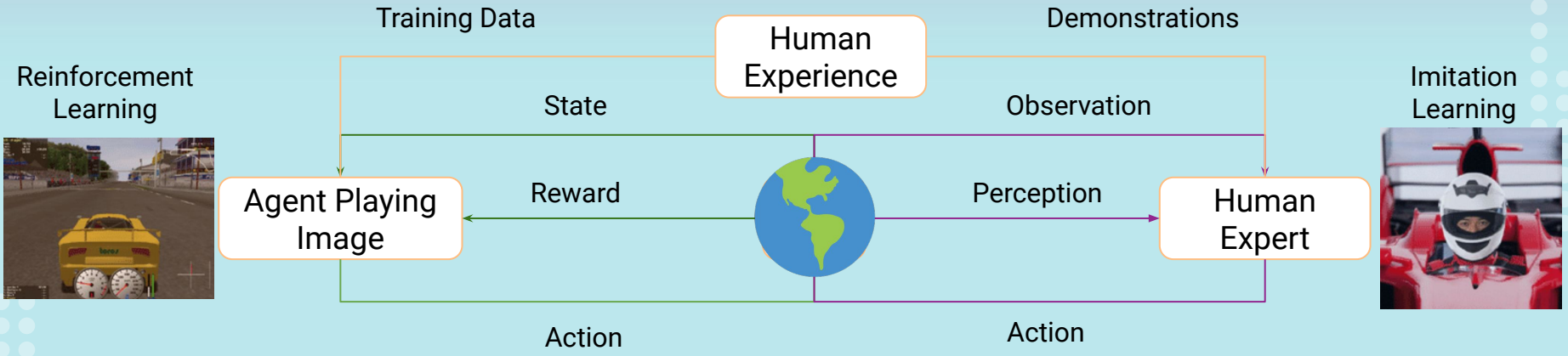
Multi-Agent Deep Reinforcement Learning

- Used in approaches like the one we have seen for Object Tracking.
- Widely used in successful DRL applications, e.g., games, robotics, autonomous driving, stock trading, social science.
- Addresses sequential decision-making with multiple autonomous agents, each optimizing its utility.
- Complexity arises from non-stationarity, multi-dimensionality, and credit assignment.
- Agents can cooperate for long-term gain or compete for a total utility of zero.
- Recent research focuses on developing new criteria and setups for Multi-Agent RL.



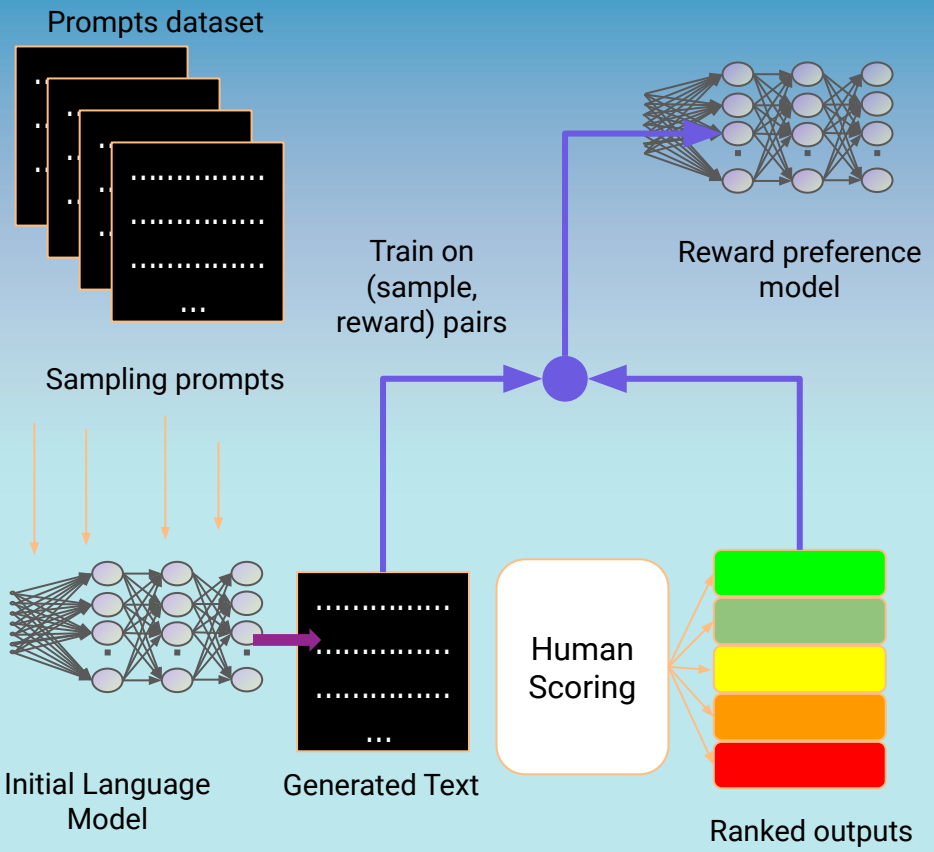
Imitation Learning

- Imitation learning is akin to learning from demonstrations.
- It trains a policy to replicate expert behavior using samples from the expert.
- An alternative to RL/DRL for solving sequential decision-making problems.
- Behavior cloning, a form of imitation learning, trains policies through supervised learning.
- Notable methods include third-person imitation learning and behavior cloning Loss.
- Generative Adversarial Imitation Learning (GAIL) integrates imitation learning into policy gradient frameworks, considering smoothness and causal entropy.



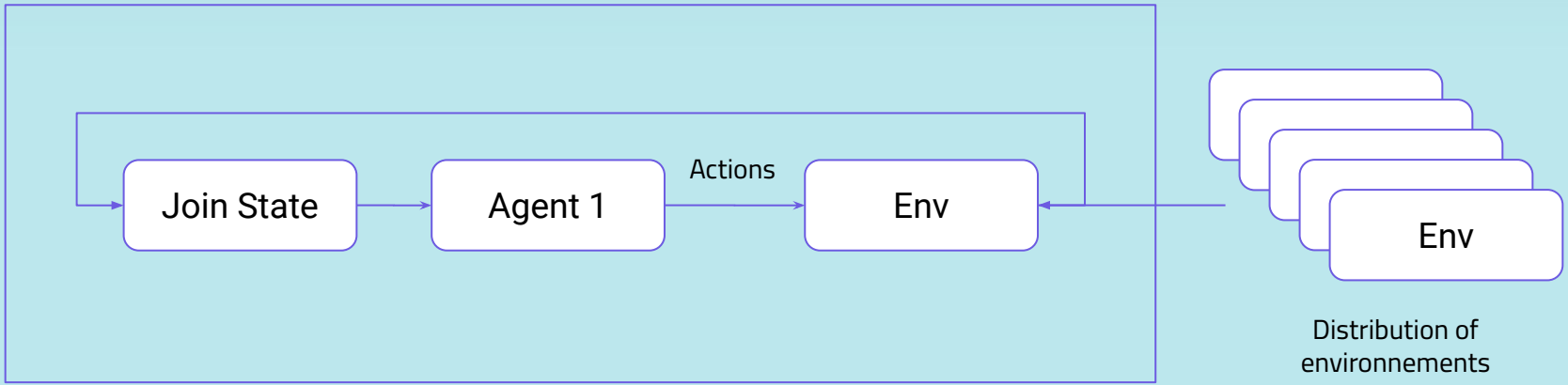
Reinforcement Learning from human-feedback

- RLHF (Reinforcement Learning from Human Feedback) trains AI agents through human interaction and feedback.
- It's ideal for tasks like customer service, negotiation, and creative writing where explicit rules are hard to define.
- RLHF uses a "reward model" to predict the quality of agent outputs and enhance RL agent exploration.
- Applications include Natural Language Processing tasks like conversational agents and text summarization.
- Challenges include scalability issues, potential undesirable behaviors, overfitting risks, and manipulation concerns.
- Would be particularly useful for edge cases in computer vision.



Meta Deep Reinforcement Learning

- Traditional DRL relies on extensive experience for task learning but struggles to generalize to new problems.
- Meta-RL addresses this by enabling agents to acquire new skills from limited experience.
- Growing interest in meta RL with various approaches explored.
- For benchmarking and evaluation, Meta-world offers an open-source simulator with 50 distinct robotic manipulation tasks.



World Models : DreamerV3

https://danijar.com/project/dreamerv3/

- DreamerV3 is a reinforcement learning algorithm based on world models.
- It outperforms previous methods in a wide range of domains with fixed hyperparameters.
- Domains include continuous/discrete actions, visual/low-dimensional inputs, 2D/3D worlds, and different data budgets, reward frequencies, and scales.
- DreamerV3 is the first algorithm to collect diamonds in Minecraft from scratch without human data (no imitation learning).
- It reduces the need for extensive tuning, making reinforcement learning more broadly applicable.

Conclusion

- Reinforcement Learning looks like a viable way to approach computer vision.
- Already multiple papers and approaches successfully implementing it
- Lot of challenges in terms of modeling and architecture.
- Advantages in terms of speed and interpretability which are needed to be studied

Thank you !

Questions Time

References

- *Active Object Localization with Deep Reinforcement Learning*.
- Reinforcement Cutting Agent Learning.
- Robust multimodal image registration using deep recurrent reinforcement learning.
- Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation.
- *Multi-agent deep reinforcement learning for multi-object tracker*.
- Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans.

2 Types of RL algorithms : Model-based & Model-free

Model-Based	Model-Free
These models are used for simulating possible scenarios, planning, and optimizing actions.	It directly learns optimal policies or value functions through trial and error.
Model-based methods aim to learn a representation of how the environment behaves.	Model-free methods focus on interacting with the environment and improving decision-making without building a detailed model.