Toward Autonomous LfO using Inverse Reinforcement Learning

Prashant Doshi (PI), Computer Science, University of Georgia
Yi Hong (Co-PI), Computer Science, University of Georgia
Kenneth Bogert (Co-PI), Computer Science, University of North Carolina at Asheville

Project URL: http://thinc.cs.uga.edu

Challenge

- Humans working with co-bots trained using demonstration in real-world scenarios
- Using incomplete, occluded, noisy observations to provide real-time predictions and resolve conflicts in human-robot interactions

Approach

- Develop incremental, multi-task IRL algorithms
- SA-Net – Detect expert trajectory(State-Action mapping) with **deep neural networks**
- Develop effective online IRL for real-time interactions

Scientific Impact:

- Generalize Inverse Reinforcement Learning to scenarios with imperfect data, limited resources, and multiple tasks
- Use IRL algorithms in realtime collaboration scenarios where storing and processing sensor data isn't practical

Broader Impacts:

- Expand capabilities of co-bots using customizable autonomy and enable spontaneous collaboration with humans
- Enrich courses in decision making and robotics
- Enhance understanding about HRI problems

SA-Net: State-Action Recognition using DNNs

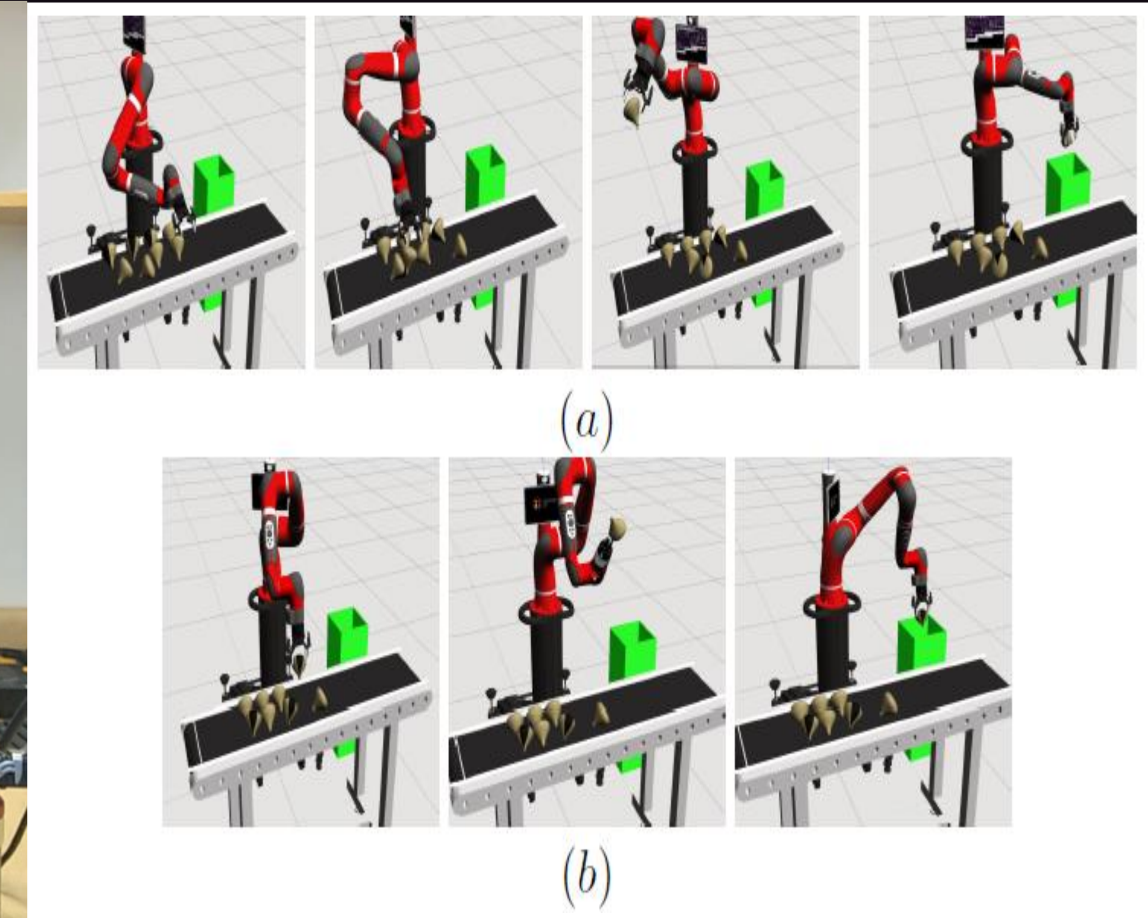
- ➢ A deep neural network architecture that recognizes state-action pairs from RGB-D data streams with high accuracy
- ➢ This supervised learning method offers a general deep learning alternative to the current adhoc techniques, which often rely on problem-specific implementations using OpenCV.
- ➢ The state is the 2D or 3D coordinates in a global reference frame and the orientation. SA Net architecture is shown.

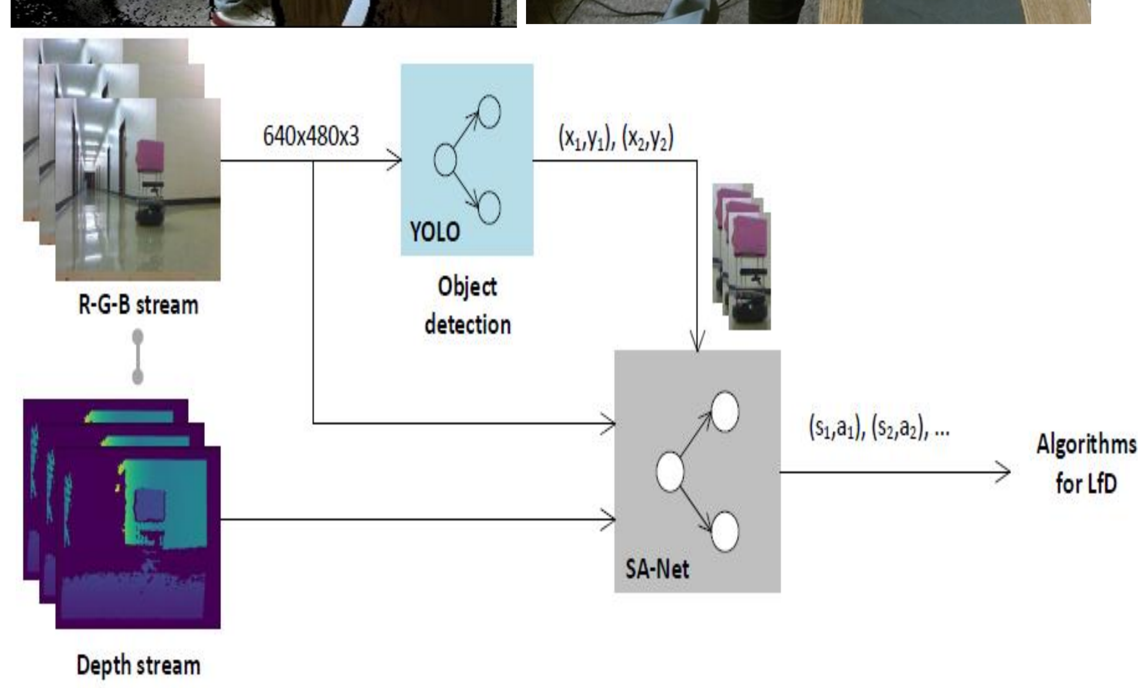
Task 1:

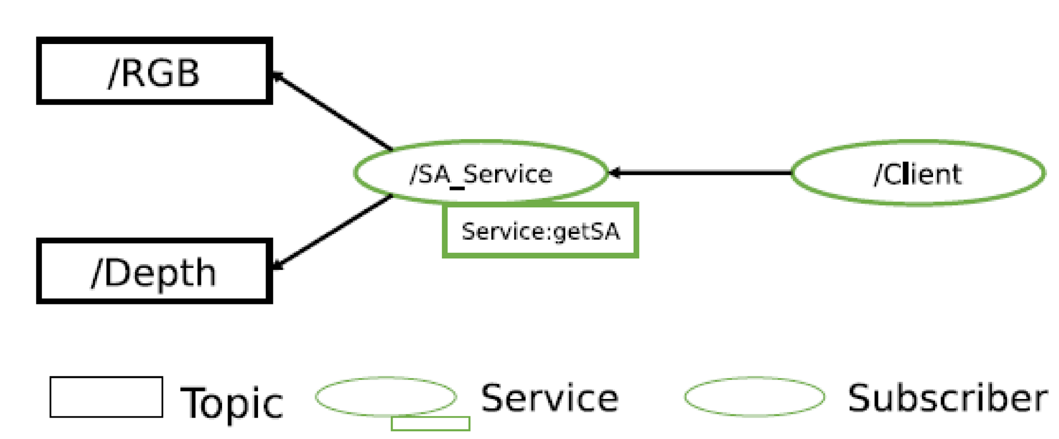
- On a TurtleBot tasked with penetrating cyclic patrols by two other TurtleBots in a hallway.

Task 2:

- This involves observing a PhantomX arm mounted on a TurtleBot, which is performing a pick-and-place task.







	Method	(TP,FP,FN,TN)	P%,R%
Expert	Pick-inspect-place Roll-pick-place	(4,0,8,12) (8,4,4,8)	100, 33 66, 66
Learned (ME-MTIRL)	Pick-inspect-place Roll-pick-place	(3,0,9,12) (6,4,6,8)	100, 25 60, 50
Learned (DPM-BIRL)	Pick-inspect-place Roll-pick-place	(2,0,10,12) (5,5,7,7)	100, 16.7 50, 41.7
Learned (EM-MLIRL)	Pick-inspect-place Roll-pick-place	(3,1,9,11) (5,4,7,8)	75, 25 55.6, 41.7

SA-Net	X	Y	θ	Action
Run 1	98.853	99.96	99.99	99.97
Run 2	98.853	99.96	99.99	99.95
Run 3	98.853	99.95	100	99.95
Run 4	98.885	99.97	99.97	99.94
Run 5	98.81	99.99	100	99.97
Mean ± SD	**98.85±0.02**	**99.97±0.014**	**99.99±0.01**	**99.74±0.01**

SA-Net	X	Y	Z	θ	Action
Run 1	97.63	95.23	96.54	98.19	99.12
Run 2	97.65	95.19	96.56	98.12	99.14
Run 3	97.62	95.2	96.58	98.23	99.1
Run 4	97.63	95.22	95.59	98.17	99.14
Run 5	97.66	95.21	95.55	98.15	99.16
Mean ± SD	**97.64±0.02**	**95.22±0.01**	**96.16±0.49**	**98.17±0.04**	**99.13±0.02**

Online IRL: Incremental Latent MaxEnt:

- ➢ Introduced I2RL, framework for online IRL, which decomposes batch IRL into sessions and defines stopping criteria
- ➢ A new method that generalizes latent maximum entropy optimization (LME) to online settings
- ➢ It offers the capability to perform online IRL in contexts where portions of the observed trajectory may be occluded.

Task:

- ▪ Patrolling scenario where the learner must observe two patrolling robots and learn the pattern to eventually penetrate the patrol without getting noticed by either of the patrolling robots.

Implementation:

- Done using physical robots (Turtlebots) to test how well I2RL can be implemented and extended in real world scenarios.
- The robots were deployed in the corridor shown and the learner robot only observes about 30% of the patrol trajectory. Results shown below are promising.

Multi-Task MaxEnt IRL:

- ✓ Due to the locality of the action probability computation, the distribution over trajectories is impacted by the number of action choice points (branching) encountered by a trajectory
- ✓ But MaxEnt technique is free from this bias
- ✓ New method combines the non-parametric clustering of trajectories and learning multiple reward functions by finding trajectory distributions of maximum entropy

Task:

- ▪ To learn and execute the most optimal sorting behavior out of the two behaviors demonstrated by the expert

Implementation:

- The image displayed shows the two behaviors performed by Sawyer Robot in simulation(ROS Gazebo).
  - (a) Sawyer robotic arm rolls its gripper over the onions thereby exposing more of their surface area. Possibly blemished onions are then picked and placed in the bin
  - (b) Sawyer picks an onion, inspects it closely to check if it is blemished, and places it in the bin on finding it to be blemished