Deep Reinforcement Learning for Robotic Grasping

Aykut Aykut, Arda Arslan

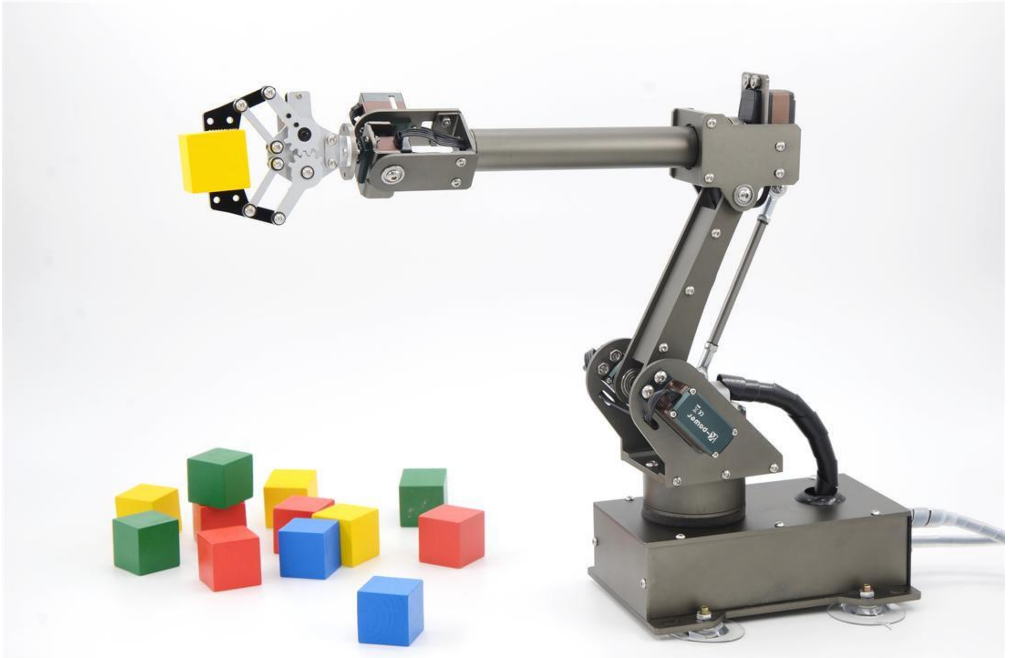
Advisor: Barış Akgün

Department of Computer Engineering

Koç University

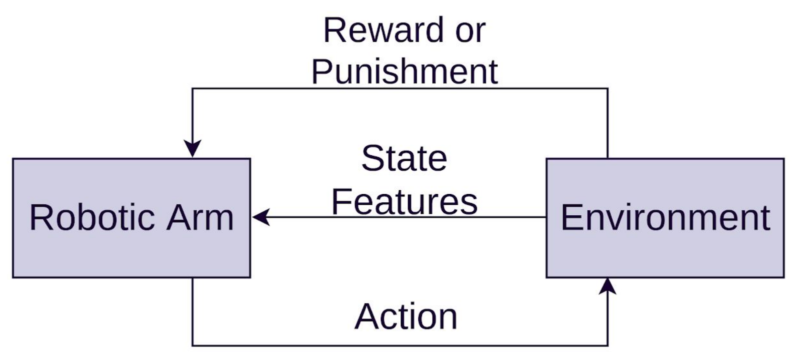
Project Description / Objectives

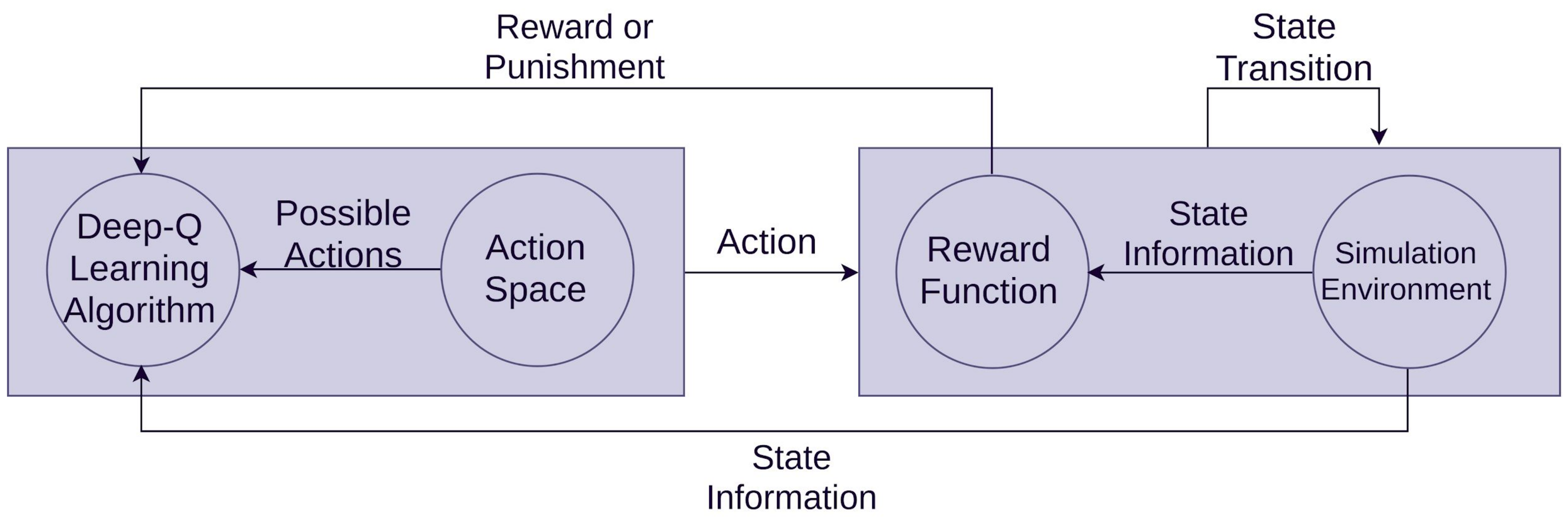
- Objective: Training a robotic arm so that it becomes able to grasp and raise objects.
- Motivation: Instead of designing a new robot software for the same industrial application on different materials, we can have a robotic arm which is capable of performing the same operation on different types of materials.



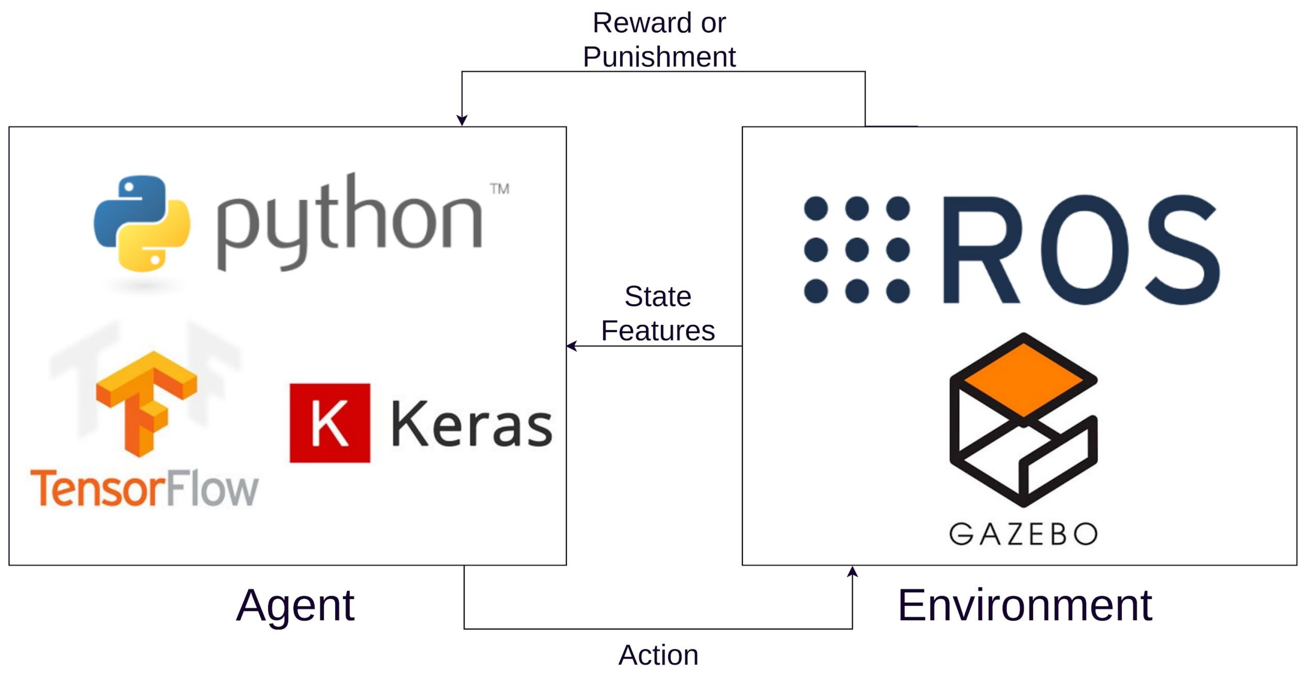
System Design

- A physics engine and simulation environment for simulating actions taken by the robotic arm.
- A visual input processor to create state features.
- Definition of reward function, state space and action space.
- A python script for developing a deep reinforcement learning algorithm and training a model.





System Architecture



- The reinforcement learning algorithm was implemented in Python programming language. While processing the visual features Keras and Tensorflow were used.
- An environment which consists of an object, a table and a robotic arm was designed using ROS and Gazebo. Gazebo includes the physics engine which handles collisions, gravity, frictions and the actions were simulated using Gazebo.

Methods

Algorithm 1: deep Q-learning with experience replay.

Initialize replay memory D to capacity N

Initialize action-value function Q with random weights θ **1**

Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$

For episode $= 1, M$ **do**

  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequence $\phi_1 = \phi(s_1)$

  For $t = 1$,T **do**

    With probability ε select a random action a_t

    otherwise select $a_t = \mathrm{argmax}_a Q(\phi(s_t), a; \theta)$ **2**

  3 Execute action a_t in emulator and observe reward r_t and image x_{t+1} **4**

    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

    Store transition $\left(\phi_t, a_t, r_t, \phi_{t+1}\right)$ in D

    Sample random minibatch of transitions $\left(\phi_j, a_j, r_j, \phi_{j+1}\right)$ from D

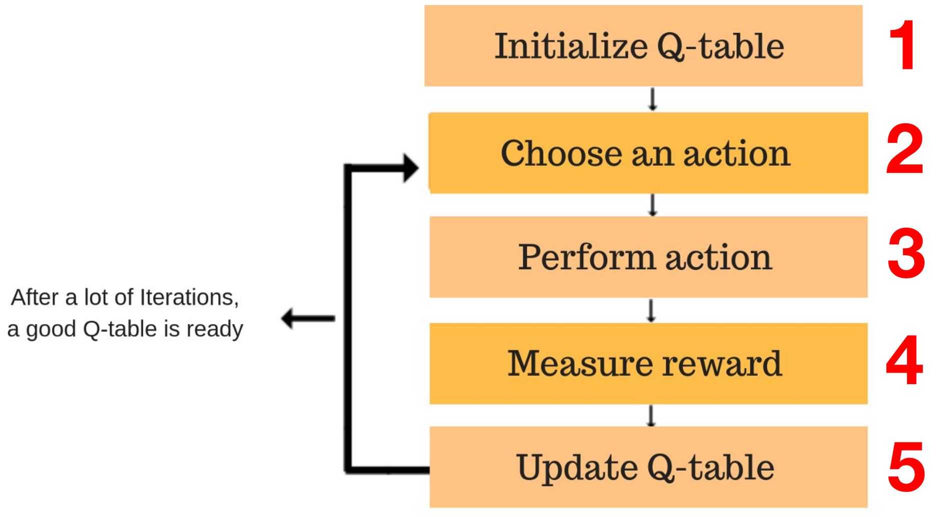
    Set $y_j = \begin{cases} r_j & \text{if episode terminates at step j+1} \\ r_j + \gamma \max_{a'} \hat{Q}\left(\phi_{j+1}, a'; \theta^-\right) & \text{otherwise} \end{cases}$

    Perform a gradient descent step on $\left(y_j - Q\left(\phi_j, a_j; \theta\right)\right)^2$ with respect to the network parameters θ

    Every C steps reset $\hat{Q} = Q$ **5**

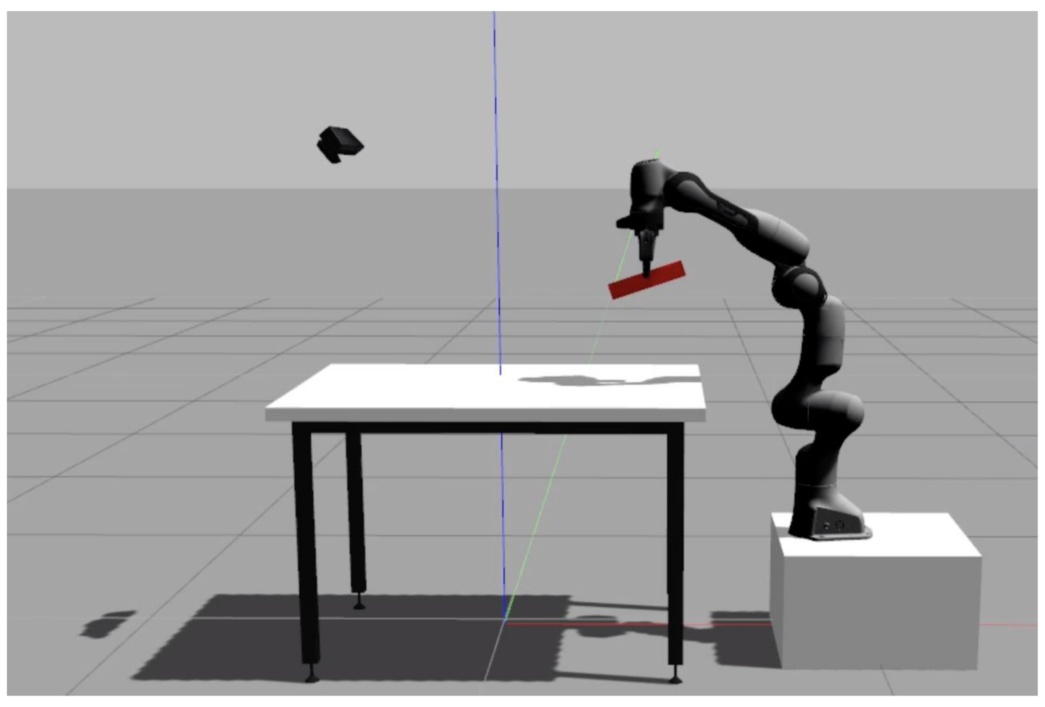
  End For

End For



Results

- Trained a robotic arm which can grasp and raise simple shaped objects.
- As a future work, a robotic arm can be trained so that it will be able to grasp and raise more complex shaped objects.



Acknowledgement

Special thanks to Barış Akgün, Cem Eteke, Öznur Özkasap, Beakal Assefa, Karim Sonbol and Faizan Safdar Ali.

References

Sanchit Jain. 2018. Pick and place robotic arm. Retrieved May 14, 2019 from https://engineering.eckovation.com/pick-place-robotic-arm/

Jonathan Hui. 2018. RL — DQN Deep Q-network. Retrieved May 14, 2019 from https://medium.com/@jonathan_hui/rl-dqn-deep-q-network-e207751f7ae4

Freecodecamp.org. 2018. An introduction to Q-Learning: reinforcement learning. Retrieved May 14, 2019 from https://medium.freecodecamp.org/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc