An Implementation of Vision Based Deep Reinforcement Learning for Humanoid Robot Locomotion

Recep Özalp
Department of Mechatronics Engineering
Firat University,
Elazig, Turkey
recepozalp105@gmail.com

Çağrı Kaymak
Department of Mechatronics Engineering
Firat University,
Elazig, Turkey
ckaymak@firat.edu.tr

Özal Yıldırım
Department of Computer Engineering
Muznur University,
Tunceli, Turkey
yildirimozal@hotmail.com

Ayşegül Uçar
Department of Mechatronics Engineering
Firat University,
Elazig, Turkey
agulucar@firat.edu.tr

Yakup Demir
Department of Electrical and Electronics Engineering
Firat University,
Elazig, Turkey
ydemir@firat.edu.tr

Cüneyt Güzeliş
Department of Electrical and Electronics Engineering
Yaşar University,
Izmir, Turkey
cuneyt.guzelis@yasar.edu.tr

***Abstract*— Deep reinforcement learning (DRL) exhibits a promising approach for controlling humanoid robot locomotion. However, only values relating sensors such as IMU, gyroscope, and GPS are not sufficient robots to learn their locomotion skills. In this article, we aim to show the success of vision based DRL. We propose a new vision based deep reinforcement learning algorithm for the locomotion of the Robotis-op2 humanoid robot for the first time. In experimental setup, we construct the locomotion of humanoid robot in a specific environment in the Webots software. We use Double Dueling Q Networks (D3QN) and Deep Q Networks (DQN) that are a kind of reinforcement learning algorithm. We present the performance of vision based DRL algorithm on a locomotion experiment. The experimental results show that D3QN is better than DQN in that stable locomotion and fast training and the vision based DRL algorithms will be successfully able to use at the other complex environments and applications.**

***Keywords*— Deep reinforcement learning, humanoid robots, locomotion skills, control.**

I. INTRODUCTION

In recent decades, the field of humanoid robotics has demonstrated big developments [1], [2]. The robots are being designed to be used everywhere where humans live. These environments are unstructured and dynamic. Therefore, the robots have to recognize the changing environment and then do an appropriate action according to the new environment. Humans can react to environmental changes thanks to their eyes. Robots had equipped by visual sensors as to the humans since they can construct local representations of their surroundings, and sequentially can adapt their behavior. Many of the existing humanoid platforms such as Nao, Pepper, Asimo, and Robotis-op2 include the cameras. Cameras provide huge stacked data such as color geometry, and texture. To process and interpret the information, it is challenging problem in all robotics applications such as walking, localization, tracking, detecting, and grasping [2]-[9].

Previous control works on humanoid locomotion are done by conventional analytical methods such as Model-Predictive Control (MPC) of Zero Moment Point (ZMP) [9]-[14]. The methods require high mathematical computations and to know perfectly the dynamics of robots at both vision based works and the others. Moreover, they do not provide exact the humanoid locomotion that has stable, energy efficiency, a reasonable walking speed, and insensitive to disturbances at previously unknown environments and even known environments. Over the recent years, many machine-learning methods have proposed to control humanoid locomotion [15]-[20]. Especially, Reinforcement Learning (RL) algorithms attracted great attention since they can be applied as model-free and generates policies though trials [21]-[26]. The success of Markov decision Process (MDP) has get increased the usage of RL to provide the robots with locomotion skills [19]-[20]. RL algorithms using the information relating to sensors such as IMU, gyroscope, and GPS and/or the features extraction from camera image ware successfully applied [17]. However, after deep learning methods and Graphics Processing Units (GPU) are introduced to the research community, the Deep Reinforcement Learning (DRL) algorithms have become indispensable for robotics area thanks to their superhuman teaching capabilities. In these methods, the raw images were used for end-to-end learning aim. Convolutional Neural Networks (CNN), Long-Short Term Memory Networks (LSTM), Recurrent Neural Networks, and Generative Adversarial Network (GAN) were used to generate the network structure of DRL [21]-[38].

In this article, an implementation of the humanoid robot locomotion is carried out by relying on the raw camera image. The kinds of Double Dueling Q Networks (D3QN) and Deep Q Networks (DQN) of DRL algorithm are used for efficient humanoid locomotion. The contribution of this study is four-fold. The first is to generate a framework for implementing DRL algorithms in Webots simulator environment [39]. The second is to perform a line tracking operation of the Robotis-op2 with the D3QN using only the raw image. The third is that the information obtained from the simulation environment can be transferred smoothly to different scenarios in real life. Fourthly, the developed algorithm can be applied to new problems such as obstacle avoidance or road planning in real dynamic environments by changing rewards and actions easily.

The article is organized as follows. In section II, the fundamentals of RL are introduced and followed by the descriptions of the DQN and D3QN algorithms. Section III describes experimental setup and shows the results of the simulation with the discussions. Section IV consists of the conclusions and future scope of research.

978-1-7281-1862-8/19/$31.00 ©2019 IEEE

II. Deep Reinforcement Learning

RL is a learning model that allows robots to learn the movements from the interactions with their environment. An intangible reward, which is obtained in response to an action in the environment without any labeling, achieves the learning process of the robot. For example, if a robot we have designed has to move inside the room, the robot first scans the motion space and performs a movement that it chooses. As a result of this movement, the robot receives a reward associated with the distance it has received if it is able to proceed in the environment. According to this reward function, the robot learns the process by determining the most appropriate strategy to increase the received reward at each time step. This learning model is briefly defined as learning through interaction to reach to a goal [22]. The most successful application of RL was carried out for Go, an ancient Chinese game. The Go game contains highly complex processes for many artificial intelligence models. However, the best Go player was defeated by the computer with the RL model [24].

A general RL structure is shown in Fig. 1. In this model, an agent called as learner and as decision maker and an environment in which this agent interacts is shown. The agent selects an action from the action space and the environment presents new states to the agent by giving it a reaction to the action. In addition, the environment returns to the agent the reward value in which the agent is trying to maximize. According to these definitions, there are different sub-components of a RL model with agent and environment. These are policy, reward function, value function and optional environment model.

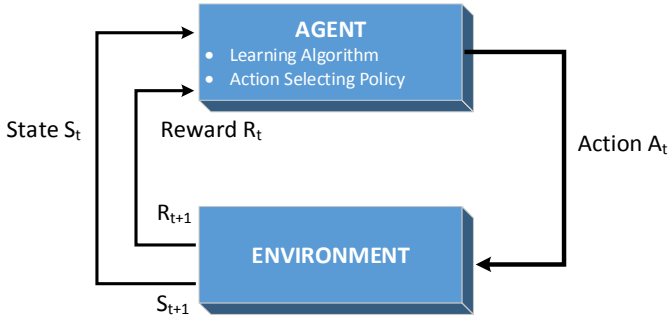

Fig. 1. A general reinforcement learning structure [22].

Robot selects an action of $a_t \in A$ according to the s_t image it receives from the camera at time $t \in [0, T]$. It then receives the r_t reward signal corresponding to this action. It starts to receive the new image, that is the next state, s_{t+1}.

The purpose of the algorithm is to maximize the accumulated feature rewards by

$$R_t = \sum_{i=t}^{T} \gamma^{i-t} r_i \quad (1)$$

where γ is discount factor. Given the policy $a_t = \pi(s_t)$, Q-value that is action-value function from a state-value pair (s_t, a_t) is calculated by

$$Q^{\pi}(s_t, a_t) = \mathrm{E}[R_t | s_t, a_t, \pi]. \quad (2)$$

Q-value function is calculated by using Bellman equation in

$$Q^{\pi}(s_t, a_t) = \mathrm{E}\big[r_t + \gamma \mathrm{E}[Q^{\pi}(s_{t+1}, a_{t+1}) | s_t, a_t, \pi]\big]. \quad (3)$$

By selecting optimal action $Q^*(s_t, a_t) = max_{a_{t+1}} \mathrm{E}[R_t | s_t, a_t, \pi]$ at each time step, the optimal Q-value function is calculate as fallows

$$Q^*(s_t, a_t) = \mathrm{E}_{s_{t+1}}\big[r + \gamma max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) | s_t, a_t\big]. \quad (4)$$

Eq. (1) means that the optimal Q-value obtained at time t consist of the accumulation of the discounted optimal Q-value at the time t+1 and the reward signal r_t. The fundamental of the DQN lies on that it is to approximate the optimal Q-value by using deep artificial neural networks, rather than calculating the Q-value over the larger state space. In the case of each s_t the calculation of all actions is unnecessary. In [26], the dueling model was proposed.

Given the present state in the conventional DQN, to calculate the Q-value of each action-state pair, a single branch of the Fully Connected (FC) layers is established after convolution layer. On the other hand, in dueling network, two separate branches of fully connected layers are set up to compute an advantage function and a value function and then it is fused to estimate Q-value. In this article, the two-arm structure examined is shown in the Fig.2. This structure has shown high success in many games both high success in many games both in terms of training speed and success [23-25].

The DQN in [23] uses the target network throughout the online network to make the overall network success determined. The target network is a copy of the online network. However, unlike the online network that updates the weights with the back propagation algorithm at each step, the weights of the target network are kept constant for a short period and then updated from the online network. Based on these two network structures, [25] claimed that the online network should choose actions and the target network should be used to solve the problem of over-optimistic calculation in Fig. 2.

More specifically, the state at time t+1, s_{t+1} is used to calculate the optimal Q'^* for at time t+1 at both target and online network. Then, the target value is calculated with the discount factor γ and the r_t reward at time t. Finally, the error is calculated by subtracting the target value with the optimal Q^* value estimated by the online network and then back propagated to update the weights when the current state, s_t is given. In this article, it is fused two methods and called as D3QN for learning humanoid locomotion in fast way [3], [26].

III. Experimental Setup

In this experimental setup, we generated a platform consisting of white line on black background in three-dimensional simulation environment, Webots [39]. This is the first study on the training of humanoid robots by using vision-based DRL. Therefore, this specific simulation environment was selected. The gained experience in this environment will continue with the applications at different simulation environments and real environments. We used small-sized humanoid robot platform, Robotis-op2 in Fig.3 and Fig.4 [40].

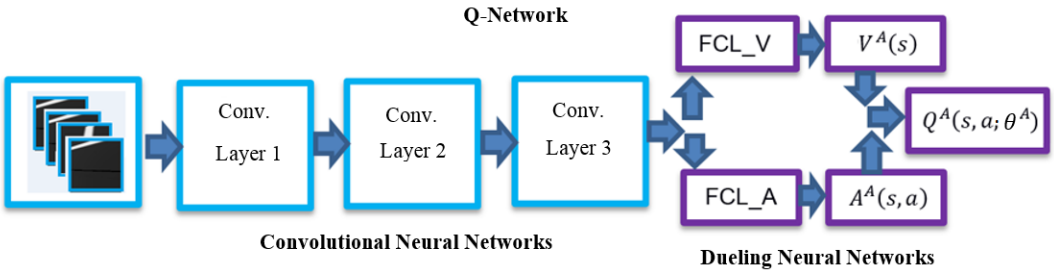

Fig. 2. The structure of D3QN model.

TABLE I. LAYER STRUCTURE OF D3QN

Layers	Filter Number-Stride	Neuron Number
Inputs	160x128x4	-
Convolutional Layer 1	10,14 -8	32
Convolutional Layer 2	4x4 -2	64
Convolutional Layer 3	3x3 -1	64
FCL-1 for Advantage Branch	1x1	512
FCL-1 for Value Branch	1x1	512
FCL-2 for Action	1x1	4
FCL-2 for Advantage Branch	1x1	2
FCL-2 for Value Branch	1x1	1

TABLE II. LAYER STRUCTURE OF DQN

Layers	Filter Number-Stride	Feature Map
Inputs	160x128x4	-
Convolutional Layer 1	8x8-4	32
Convolutional Layer 2	4x4-2	64
Convolutional Layer 3	3x3-1	64
Maximum Pooling	-	
FCL-1	-	512

We generated DQN and D3QN frameworks using Keras and Tensorflow in Python. DQN network structure in [23] and D3QN network structure in [3] were used in this study. The used network structures of DQN and D3QN are given in Table 1 and 2. The models were trained and tested in Webots simulation environment on a workstation including Nvidia Titan XP. The performances of DQN and D3QN structures were evaluated. We generated the states of environment by using the last four historical camera images. We used the four actions consisting of right, left, and straight walking at two different speeds. We selected that

reward is 1 if the robot is on right or left side of line,

reward is 0.5 if the robot is on right or left side of line,

reward is -10 for D3QN if the robot falls down,

reward is -1 for DQN if the robot falls down,

reward is -1 for the other cases

The results of D3QN and DQN are given in Fig. 5 and Fig 6. As can be seen from Fig. 5, D3QN learns its way after just 57 training epochs and it receives high rewards.

On the other hand, Fig. 6 shows that DQN is not able perfectly learns and receives low rewards.

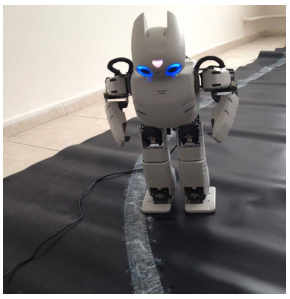

Fig. 3. Humanoid robot, Robotis-op2 on line.

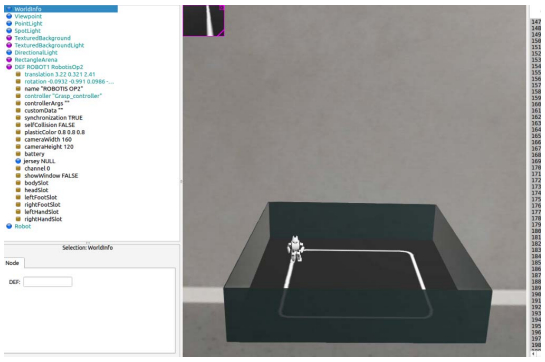

Fig. 4. The experimental setup constructed at Webot environment.

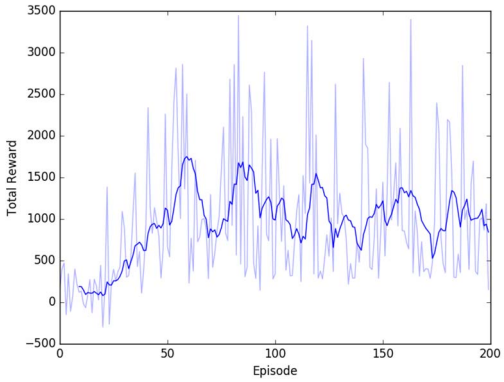

Fig. 5. Total reward and average reward with respect to trainig episode for D3QN

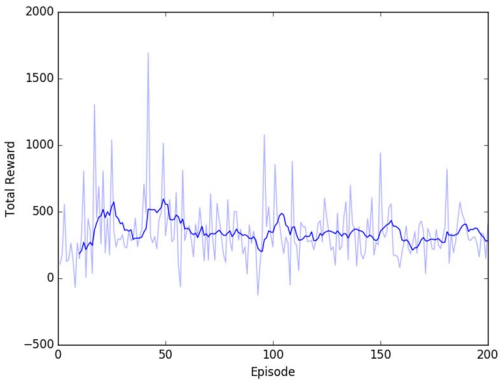

Fig. 6. Total reward and average reward with respoct to training episode for DQN.

IV. Conclusion

In this article, we proposed the usage of a D3QN and DQN models for locomotion of Robotis-op2 humanoid robot. We used just raw images from camera as the input of D3QN and DQN models. The models ware trained and tested in Webots simulator. The experimental results showed that the vision-based DRL provides a promising for the future and the performance of D3QN is much better than that of DQN in that training speed and stable locomotion. The feature work will include to use different algorithms for same problem and to transfer the DRL algorithms to real robot. Moreover, the same algorithms will be able to use to walk by avoiding from obstacles for both real and simulation environment.

Acknowledgment

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) grant numbers 117E589. In addition, GTX Titan X Pascal GPU in this research was donated by the NVIDIA Corporation.

References

[1] SPENKO, Matthew; BUERGER, Stephen; IAGNEMMA, Karl (ed.). The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue. Springer, 2018.

[2] RUSSELL, Ben. Robots: The 500-year Quest to Make Machines Human. Scala Arts Publishers, Inc., 2017.

[3] XIE, Linhai, et al. Towards monocular vision based obstacle avoidance through deep reinforcement learning. *arXiv preprint arXiv:1706.09829*, 2017. *"RSS 2017 workshop on New Frontiers for Deep Learning in Robotics"*

[4] CHEN, Xi, et al. Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018. p. 3110-3116.

[5] KAHN, Gregory, et al. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018. p. 1-8.

[6] LEVINE, Sergey, et al. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 2016, 17.1: 1334-1373.

[7] ZHANG, Fangyi, et al. Towards vision-based deep reinforcement learning for robotic motion control. *arXiv preprint arXiv:1511.03791*, 2015.

[8] KAYMAK, Cagri; UCAR, Aysegul. Implementation of Object Detection and Recognition Algorithms on a Robotic Arm Platform Using Raspberry Pi. In: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP). IEEE, 2018. p. 1-8.

[9] GARCÍA VÁZQUEZ, Mauricio Josafat. Vision-Based Locomotion for Humanoid Robots. 2013.

[10] DALLALI, Houman. *Modelling and dynamic stabilisation of a compliant humanoid robot, CoMan*. 2012. PhD Thesis. The University of Manchester (United Kingdom).

[11] Vukobratovi, M. and Borovac, B., 2004. Zero-moment pointthirty five years of its life. International journal of humanoid robotics, 1(01), pp.157-173.

[12] MARTÍNEZ, Pablo A.; CASTELÁN, Mario; ARECHAVALETA, Gustavo. Vision based persistent localization of a humanoid robot for locomotion tasks. *International Journal of Applied Mathematics and Computer Science*, 2016, 26.3: 669-682.

[13] VAN DER NOOT, Nicolas; BARREA, Allan. Zero-Moment Point on a bipedal robot under bio-inspired walking control. In: MELECON 2014-2014 17th IEEE Mediterranean Electrotechnical Conference. IEEE, 2014. p. 85-90.

[14] SHAN, Jiang; NAGASHIMA, Fumio. Neural locomotion controller design and implementation for humanoid robot HOAP-1. In: *20th annual conference of the robotics society of Japan*. 2002.

[15] SILVA, Pedro, et al. Automatic generation of biped locomotion controllers using genetic programming. *Robotics and Autonomous Systems*, 2014, 62.10: 1531-1548.

[16] TAN, Jie, et al. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*IEEE., 2018.

[17] DANEL, Marek. Reinforcement learning for humanoid robot control. *POSTER, May*, 2017.in POSTER 2017, PRAGUE MAY 23.

[18] SONG, Doo Re, et al. Recurrent Deterministic Policy Gradient Method for Bipedal Locomotion on Rough Terrain Challenge. In: *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018. p. 311-318.

[19] PENG, Xue Bin; BERSETH, Glen; VAN DE PANNE, Michiel. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 2016, 35.4: 81

[20] PENG, Xue Bin, et al. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 2017, 36.4: 41.

[21] DUAN, Yan, et al. Benchmarking deep reinforcement learning for continuous control. In: *International Conference on Machine Learning*. 2016. p. 1329-1338.

[22] SUTTON, Richard S.; BARTO, Andrew G. Reinforcement learning: An introduction. MIT press, 2018.

[23] MNIH, Volodymyr, et al. Human-level control through deep reinforcement learning. *Nature*, 2015, 518.7540: 529.s

[24] SILVER, David, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 2016, 529.7587: 484.

[25] VAN HASSELT, Hado; GUEZ, Arthur; SILVER, David. Deep reinforcement learning with double q-learning. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[26] WANG, Ziyu, et al. Dueling network architectures for deep reinforcement learning. arXiv preprint arXiv:1511.06581, 2015.

[27] XIE, Zhaoming, et al. Iterative Reinforcement Learning Based Design of Dynamic Locomotion Skills for Cassie. *arXiv preprint arXiv:1903.09537*, 201

[28] CLARY, Patrick, et al. Monte-Carlo Planning for Agile Legged Locomotion. In: *Twenty-Eighth International Conference on Automated Planning and Scheduling*. 2018.

[29] QUILLEN, Deirdre, et al. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018. p. 6284-6291.

[30] PENG, Xue Bin; VAN DE PANNE, Michiel. Learning locomotion skills using deeprl: Does the choice of action space matter?. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2017. p. 12.

[31] SIRAVURU, Avinash, et al. Deep visual perception for dynamic walking on discrete terrain. In: *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017. p. 418-424.

[32] LOBOS-TSUNEKAWA, Kenzo; LEIVA, Francisco; RUIZ-DEL-SOLAR, Javier. Visual navigation for biped humanoid robots using

deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2018, 3.4: 3247-3254.

[33] HA, Sehoon; KIM, Joohyung; YAMANE, Katsu. Automated deep reinforcement learning environment for hardware of a modular legged robot. In: *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, 2018. p. 348-354.

[34] KUMAR, Visak CV; HA, Sehoon; YAMANE, Katsu. Improving Model-Based Balance Controllers using Reinforcement Learning and Adaptive Sampling. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018. p. 7541-7547.

[35] UÇAR, Ayşegül; DEMIR, Yakup; GÜZELIŞ, Cüneyt. Moving towards in object recognition with deep learning for autonomous

[39] MICHEL, Olivier. Cyberbotics Ltd. Webots™: professional mobile robot simulation. International Journal of Advanced Robotic Systems, 2004, 1.1: 5.

[40] http://emanual.robotis.com/docs/en/platform/op2/getting_started/

driving applications. In: 2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA). IEEE, 2016. p. 1-5.

[36] CHEN, Yu Fan, et al. Socially aware motion planning with deep reinforcement learning. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017. p. 1343-1350.

[37] KUMAR, Arun; PAUL, Navneet; OMKAR, S. N. Bipedal Walking Robot using Deep Deterministic Policy Gradient. *arXiv preprint arXiv:1807.05924*, 2018.

[38] HEESS, Nicolas, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.