논문 19-44-12-13 The Journal of Korean Institute of Communications and Information Sciences '19-12 Vol.44 No.12
https://doi.org/10.7840/kics.2019.44.12.2269

스마트 팩토리 구축을 위한 동적인 환경에서의 DQN 강화학습 기반 로봇 최적 이동 경로 탐색

박 광 석◆, 박 진 만*, 윤 완 규*, 유 상 조°

DQN Reinforcement Learning: The Robot's Optimum Path Navigation in Dynamic Environments for Smart Factory

Kwang-Seok Park◆, Jin-Man Park*, Wan-Kyu Yun*, Sang-Jo Yoo°

요 약

스마트 팩토리에서 사용되는 로봇들은 대부분 사용자가 원격 조종하거나, 라인 트레이서 기법으로 대부분 출시되어 상용화 되었다. 이 로봇들은 지정된 공정 위치 또는 알람 경보가 작동된 위치로 이동하여 부속품을 전달하거나 수리를 하는 데 사용된다. 하지만 이러한 방법들은 수시로 변하는 공정 라인 배치와 고장 발생 위치에 의해 기존의 알고리즘을 적용하기 어렵다는 문제가 있다. 본 논문에서는 이러한 문제를 해결하기 위해 상황 변화에 능동적이고 유연하게 대처할 수 있는 강화학습 모델을 제안한다. 제안하는 모델은 강화학습 알고리즘의 하나인 심층 Q-네트워크를 이용한 모델로써 공장 내부 구조를 촬영한 이미지를 받아 현재 로봇의 위치에 대한 최적의 이동 경로를 도출해 낸다. 해당 방법을 통해 본 논문에서 제시하는 상황 변화에 능동적이고 유연하게 대처할 수 있는 시스템을 구축할 수 있다. 또한, 해당 시스템을 이용하면 에너지 절감 및 생산성 향상에 큰 기여를 할 수 있게 된다. 모의실험 결과 학습하지 않은 새로운 환경에서도 최적의 경로로 이동하는 결과를 보임으로써 제안된 방법이 공장 내부의 구조 및 상황들을 고려하여 효율적으로 최적 경로를 도출하는 것을 확인하였다.

Key Words : Deep Q-Network, optimum-path, image pre-processing, smart factory, reinforcement learning

ABSTRACT

Most of the robots used in smart factory are controlled by users or are commercially available with line-tracer techniques. These robots are used to carry out accessories or repair by moving them to a designated position or to a position where alarms are activated. However, there is a problem with these methods that it is difficult to apply the existing algorithms due to frequently changing process line layout and a position where alarms are activated. To solve these problems, this paper proposes a reinforcement learning model that can be actively and flexibly dealt with the change of situation. The proposed model is a model using Deep Q-Network, one of the enhanced learning algorithms, that receives images of the internal structure and derives the optimal path of movement for the current location of robots. This method allows the establishment of a system that can be

※ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2017R1A2B4003512)
◆ First Author : Inha University Department of Information and Communication Engineering, tpy7723@naver.com, 학생회원
° Corresponding Author : Inha University Department of Information and Communication Engineering, sjyoo@inha.ac.kr, 종신회원
* Inha University Department of Information and Communication Engineering, ye_in2@naver.com; zipell135@gmail.com, 학생회원
논문번호 : KICS2019-08-173, Received August 7, 2019; Revised November 6, 2019; Accepted November 6, 2019

proactive and flexible in responding to the changes in circumstances. In addition, the system will greatly contribute to energy savings and productivity gains. The simulation results showed that even in new unlearned environments it was found that the proposed method effectively elicited the optimal path taking into account the structure and conditions inside the plant.

Ⅰ. 서 론

스마트 팩토리는 제조업에 4차 산업혁명이 적용된 생산 시스템으로 생산의 전반적인 공정이 네트워크로 연결되어 다양하고 세분화된 고객의 요구에 대응할 수 있는 시스템이다. 기존 제조업체의 산업 기기와 ICT(Information and Communication Technologies) 융합을 통하여 제품 설계에서부터 생산, 품질, 유통, 고객 만족도까지 생산 활동 전반의 정보가 공유되는 지능형 생산 공장을 말한다[1,2]. 최근 산업현장에서는 생산성 향상 및 에너지 절감을 위해 스마트 팩토리에 대한 관심이 증가하고 있다. 생산 속도 향상 및 에너지 절감 측면에서, 공장 내부에서 사용되는 자율 주행 로봇은 부속품을 전달하거나 알람 경보가 발생한 위치로 이동할 때 효율적인 경로로 이동해야 한다. 현재 스마트 팩토리에서 사용하는 로봇들은 대부분 원격 조종 또는 라인 트레이서 기법으로 대부분 출시되었다[3]. 하지만 이러한 방법들은 수시로 변하는 공정 라인 배치와 알람 발생 위치에 적용될 때 경로 이탈, 장애물 충돌 등의 오작동 문제가 발생할 수 있기 때문에 스마트 팩토리에 적용되기에 적합하지 않다. 따라서 상황의 변화에 따라 능동적이고 유연하게 대처하는 시스템이 필요하다. 본 논문에서는 DQN(Deep Q-Network) 강화학습을 통해 동적인 환경에서 발생하는 오작동 문제를 해결하고, 더 나아가 로봇이 새로운 상황에서도 최단 시간으로 공장 내부의 알람 경보가 발생한 모든 경유지를 방문하도록 하는 최적 경로 탐색 방법을 제시한다.

강화학습(reinforcement learning)은 임의의 환경에서 학습주체인 에이전트의 행동 결과에 대한 보상 여부에 따라 행동을 변화시키고 발전시킨다는 이론이다. 즉, 정답이 정해져 있지 않은 상황에서 현재 행동에 따라 발생하는 보상을 통해 다음 행동에 영향을 주는 방식으로 학습한다.[4,5] 강화 학습은 대표적으로 Q-learning과 Deep Q-learning 등이 있다. 본 논문에서 다룰 Q-learning은 간단한 저 차원의 문제에서는 강력한 성능을 보이나 이미지 픽셀 입력 등 고차원의 문제로 갈수록 이들의 매개 변수들을 수용할 메모리 문제가 발생한다. 그러나 최근의 딥 러닝 연구의 발전과 더불어 Q-learning의 Q-table을 인공신경망으로 변형시켜 획기적으로 메모리 문제를 해결하였으며 이 변형으로 발생하는 여러 문제가 해결되어 심층 Q-learning, 즉 "Deep Q-Network" 알고리즘의 연구결과로 상당한 진전이 이루어졌다. 실제로 atari 비디오 게임에서 화면의 픽셀을 입력으로 사용하여 인간 수준의 성능을 구현하는데 깊은 신경망 함수 근사 식이 행동 가치 함수를 추정하는 데 사용되었다[6].

관련 연구로써 Kwak은 Q-learning과 유전자 알고리즘을 이용하여 강화학습의 학습률(learning rate), 할인율(discount factor) 등 중요 파라미터를 최적화하는 방법을 적용하여 이동로봇이 현재 지점부터 목표지점까지 최적 경로를 탐색하는 기술을 설명했다[7]. Kim은 공간 압축과 실시간 강화학습(online reinforcement learning)을 이용하여 미로 환경에서 기존 강화학습 방식보다 더 빠르게 최단 경로를 탐색하는 것을 보였다[8]. Lee는 Lidar 센서 데이터를 DQN 입력으로 사용하여 장애물을 유동적으로 인식, 회피하는 것을 보였다[9]. 현재 운영 중인 스마트 팩토리의 내부 구조 및 상황은 유동적으로 바뀔 수 있고, 로봇의 신속성 및 정확성이 요구된다. 그러나 인공신경망을 사용하지 않은 Q-learning, 실시간 강화학습 등은 새로운 상황에 적용시키기에 오작동 문제가 발생한다. 본 논문에서는 메모리와 시간을 절약하고 다양한 상태를 처리하기 위해 Q-leaning 대신 Deep Q-learning을 사용함으로써 보다 많은 상황에 대처할 수 있고, 또한 학습하지 않은 새로운 상황에서도 최적 경로를 탐색하는 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 강화학습과 DQN에 대한 기본 이론을 설명한다. 3장에서는 구현한 강화학습 네트워크 모델과 DQN을 이용한 최적 경로 탐색하는 방법에 대해 설명한다. 4장에서는 시뮬레이션 결과 및 제안된 모델의 성능을 분석하고 5장에서 결론을 내린다.

Ⅱ. 관련 기술

2.1 강화학습

강화학습은 입력에 대한 답을 도출해내기 어렵거나 정답이 없는 상황에서 사용하는 학습 알고리즘이다. 어

떤 환경 안에서 정의된 에이전트가 현재의 상태를 인식하여, 선택 가능한 행동 중 보상을 최대화하는 행동 혹은 행동 순서를 선택하는 방법이다[10]. 강화학습은 마코프 의사결정 과정(MDP: Markov Decision Process)의 수학적 모델 정의에 의해 시작되었다. MDP의 기본 이론은 현재 상태는 과거의 모든 상태를 담고 있고, 다음 상태와 보상은 현재 상태와 행동에 의해 결정된다.

강화학습은 이러한 수학적 모델 정의를 기반으로, 어떤 환경을 탐색하는 에이전트가 현재의 상태를 인식하여, 어떤 행동을 취하면 환경으로부터 보상을 얻게 되며, 에이전트는 이를 통해 누적 보상을 최대화하도록 학습해 나가는 기계 학습(machine learning) 기법의 하나이다.

각 상태의 행동에 대한 현재 상태의 누적 보상은 수식 (1)와 같이 나타낼 수 있다.

$$Q(s,a) = r + \gamma \bullet \max_{a'} Q(s',a') \quad (1)$$

강화학습에서 사용하는 Q 값은 행동마다 그 행동을 취했을 때 얼마나 좋은지를 측정하는 요소로써 높은 Q 값을 가진 행동을 선택했을 때 더 좋은 보상을 받는다. 여기서 $Q(s,a)$는 현재 상태(s)에 대한 행동(a)에 대한 Q 값이며, r은 현재 상태를 기준으로 하여 다음 상태(s')에 받게 될 기대되는 보상이다. 또한 γ은 미래의 보상이 현재에 얼마나 가치 있는지를 표현하는 할인율로 0과 1 사이의 값을 갖는다. 즉, 현재의 행동이 미래에 더 좋은 상황을 가져다주는지에 대한 계수로 0에 가까울수록 현재 선택에 많은 영향을 받고, 1에 가까울수록 미래 선택에 많은 영향을 받게 된다. $\max_{a'} Q(s',a')$는 다음 상태에서의 가장 큰 Q 값이다. 즉, 현재 상태의 누적 보상은 다음 상태의 보상과 다음 상태의 행동에서 가장 큰 값으로 구해진다[11].

Q 학습의 최적 정책은 수식 (2)와 같이 나타낼 수 있다.

$$\pi^*(S) = \arg max_a Q(s,a) \quad (2)$$

여기서 $\arg max_a Q(s,a)$는 현재 상태(s)에 대해서 가장 Q값이 큰 행동(a)를 의미한다. 본 논문에서는 로봇이 움직일 수 있는 방향, 즉 상하좌우 네 가지 행동 중에서 Q 값이 가장 큰 값을 택하는 정책이 사용된다.

강화학습에서 사용하는 알고리즘 중 하나로 epsilon-greedy(ε-greedy) 알고리즘이 있다. 이 알고리즘은 1−ε의 확률로 현재 상태의 Q 값 중 최적 정책을 통해 행동을 선택(exploitation)하고, ε의 확률로 최적 정책과는 상관없이 무작위로 행동을 골라서 탐험(explore)하는 알고리즘이다. epsilon-greedy 방식을 사용하면 에피소드 초기에는 최적의 길을 탐험을 통해 발견해 내고, 후반에는 최적 정책을 이용하여 최적의 길을 강화할 수 있게 된다.

2.2 심층 Q-네트워크

DQN은 강화학습에 인공신경망이 결합한 형태의 알고리즘이며, 현재 상태가 입력될 때 가능한 모든 행동의 예측 Q 값이 출력되는 형태이다. 인공신경망을 학습시킬 때 예측값과 실제 값이 포함되는 손실 함수(loss function)를 사용한다. 하지만 두 값이 같은 파라미터를 사용하기 때문에 가중치를 업데이트할 때마다 실제 값이 함께 변경되어 loss가 수렴하지 않는 문제가 생긴다. 이를 딥 마인드에서 제안한 네트워크 분리(separation networks)와 경험 리플레이(experience replay)를 통해 방지한다. 따라서, DQN은 과거 경험을 저장할 경험 저장소(experience buffer)를 사용하고, 학습과 실제 값을 동시에 산출하는 하나의 네트워크와 학습을 위한 에이전트인 메인 네트워크와 실제 값을 산출하기 위한 타겟 네트워크 두 개의 네트워크로 구현한다. 여기서 메인 네트워크의 입력은 현재 상태이며, 출력은 상태에 대한 예측 Q 값이다. 타겟 네트워크의 입력은 다음 상태이며 실제 Q 값을 구하기 위해 사용된다[12].

본 논문에서는 각 이미지를 한 개의 상태로 정의하고, 이를 네트워크 입력으로 사용하였다. 하지만 이미지 정보는 그 자체만으로도 데이터양이 많을 뿐만 아니라 이미지가 조금만 변화해도 새로운 상태로 정의되기 때문에 기존의 강화학습 알고리즘인 Q-learning 대신에 많은 상태를 처리하는데 적합한 DQN을 사용하였다.

네트워크 입력으로 이미지를 사용한 DQN 강화학습 순서도는 그림 1과 같이 나타낼 수 있다.

이미지 정보를 단순화하기 위해 전처리 과정(pre-processing)을 진행한 후 네트워크 입력으로 사용한다. 에이전트는 메인 네트워크를 거쳐서 나온 예측된 Q 값을 토대로 일정 확률로 탐험(exploration)을 하는 epsilon-greedy 방식 및 최적 정책 방식으로 행동을 선택한다. 선택된 행동을 기반으로 변화된 환경에 대한 행동의 보상 값을 DQN 모델에 전달하고 주고 변화된 환경을 다음 상태로 입력한다.

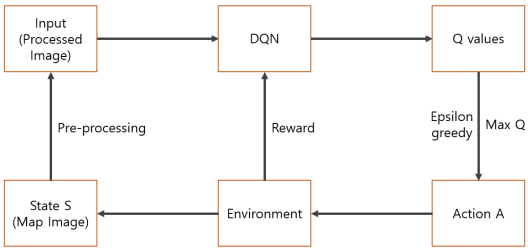

그림 1. 이미지 입력을 통한 강화학습
Fig. 1. Reinforcement learning through image input

예측값, 실제 값, 오차 함수는 수식 (3), (4), (5)와 같이 나타낼 수 있다. 식 (3)은 메인 네트워크에서 현재 상태에 대해 예측 Q 값을 산출하는 것이고 식 (4)는 다음 상태에서의 타겟 네트워크의 Q 값과 보상함수를 이용하여 실제 Q 값을 산출하는 것이다. 식 (5)는 메인 네트워크와 타겟 네트워크를 통해 얻은 실제 값과 예측 값의 오차 함수이며, 이를 이용하여 오차가 최소가 되도록 메인 네트워크 파라미터를 업데이트하여 네트워크를 학습한다.

$$W_s = Q(s,a|\theta) \tag{3}$$

$$Q^*(s,a|\bar{\theta}) = r + \gamma \bullet \max Q(s',a'|\bar{\theta}) \tag{4}$$

$$\cos t(W) = \frac{1}{m}\sum_{i=1}^{m}(W_s^i - Q^*(s,a|\bar{\theta})^i)^2 \tag{5}$$

여기서 θ는 예측값과 학습을 위한 메인 네트워크이며, $\bar{\theta}$는 실제 값을 산출하는 타겟 네트워크이다.

Ⅲ. 스마트 팩토리에서 최적의 이동 경로를 얻기 위한 DQN 모델 구현 및 학습방법

본 장에서는 내부 구조 및 알람 위치가 동적으로 바뀔 수 있는 스마트 팩토리에서 DQN 모델을 이용한 최적 경로를 탐색하는 방법과 해당 모델을 효율적으로 학습시키기 위한 환경 조성 방법에 대해 설명한다. 전체적인 작업 수행 과정은 그림 2와 같다.

먼저 천장에서 스마트 팩토리 내부 구조를 CCTV 등을 통해 촬영하고 중앙관리 시스템은 촬영된 이미지를 Wi-Fi와 같은 무선 네트워크를 통해서 전달받는다. 이후 관리 시스템은 이미지 프로세싱 과정을 진행한 뒤 DQN 강화학습을 수행하고 로봇에게 수행할 동작을 명령한다. 명령을 받은 로봇은 동작을 진행한 뒤 다

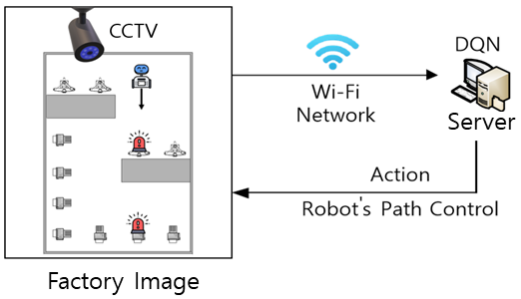

그림 2. 전체적인 작업 수행 과정
Fig. 2. The process of performing an overall task

음 명령을 기다린다. 로봇이 이동함에 따라 변화한 상태를 이용해 알고리즘을 다시 동작시킨다. 원하는 동작이 모두 수행될 때까지 이를 반복한다.

3.1 정확한 사물 인식과 학습 속도 향상을 위한 이미지 프로세싱 과정

상태 변수는 강화학습을 위해 에이전트에 입력되는 변수로서 예측값과 실제 값을 산출하는 요소이다. DQN에서 사용하는 상태 변수는 이미지 데이터이며, 원본 이미지는 데이터의 양이 많고 특정 사물을 탐지하기 어려워 전처리 과정이 필요하다. 이미지 전처리 후 네트워크에 입력되는 과정은 그림 3과 같다.

주어진 3차원 RGB 이미지의 크기를 일정한 비율 c로 나누어 down-sizing을 수행하고, 이미지의 특징과 모서리를 쉽게 찾기 위해 grayscale로 변환한다. 마지막으로 GPU 환경에 맞게 이미지를 정사각형 형태로 불필요한 부분을 삭제하고, 로봇의 이동 방향이나 속도 등을 알기 위해 last 4 frame을 스택으로 쌓아 네트워크 입력으로 사용한다. DQN network는 CNN

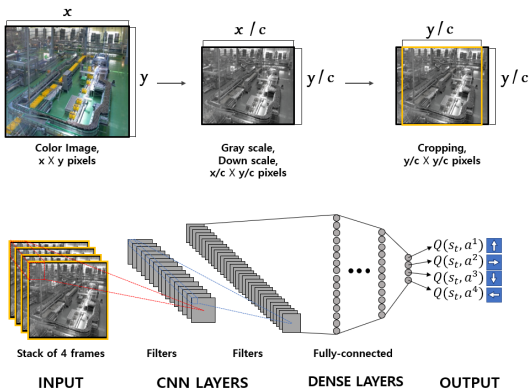

그림 3. 이미지 전처리 및 DQN 모델 구조
Fig. 3. Image pre-processing and DQN model architecture

(Convolutional network)을 통해 이미지의 특징을 찾고, FC (Fully-connected) 레이어를 이용하여 최종적으로 이동 방향에 대한 Q 값을 얻는다.

본 논문에서는 그림 4와 같이 실제 촬영된 이미지를 모델에 입력하기 전 전처리를 진행했다고 가정한다. 전처리를 통해 만들어진 이미지는 흰색, 파란색, 빨간색, 검은색으로 각각 길, 로봇, 경유지, 벽을 나타낸다.

실제 촬영된 이미지의 픽셀을 일정한 간격으로 그룹을 지어 픽셀 그룹 내부의 평균 RGB 값을 계산하여 길, 로봇, 경유지, 벽으로 분류한다. 예를 들어 공장 내부에서 빨간 알람등이 켜져 있는 곳은 빨간색의 평균값이 높고, 초록색의 평균값, 파란색의 평균값이 낮기 때문에 해당 구간을 경유지로 분류하게 된다. 픽셀 그룹의 사이즈를 작게 할수록 지형 및 구조에 대해 더 정확한 셀로 구분할 수 있다. 역할에 따라 특정 값을 부여한 후 이차원 배열 형태로 나타내고 모든 행을 연속된 데이터로 한 개의 행으로 나열한다. 최종적으로 강화학습을 위해 에이전트에 입력되는 입력층은 값들이 현재의 상태에 대한 정보를 의미하는 일차원 행렬 형태가 된다. 이러한 일차원 행렬의 길이를 MapSize로 정의하였다. 본 논문에서는 길, 로봇, 경유지, 벽 등 변동 요인이 되는 요소를 상태 변수로 사용하였다. 경유지는 벽이 아닌 곳에서만 무작위로 동시다발적으로 발생할 수 있고, 벽의 구조도 바뀔 수 있다. 또한, 로봇이 이동할 수 있는 방향은 상하좌우로 한정하였다.

에이전트의 상태 변수는 스마트 팩토리의 구조 및 역할을 파악하고 안정적인 학습하기 위해 현재 로봇의 위치, 길, 경유지, 벽 등 변동 요인이 되는 요소를 사용하였다. 표 1에 상태 변수에 대해서 나타냈다.

최종적으로 공장 내부 구조를 촬영한 이미지는 전처리 과정을 통해 각각 해당하는 위치에 대한 값으로 구성된 일차원 배열로 변환된다. 이를 통해 에이전트는 현재 환경에 대한 상태를 쉽게 파악할 수 있고, 이미지 전체가 입력층으로 들어가는 것보다 저장 공간이 절약되고 빠른 학습을 진행할 수 있게 된다.

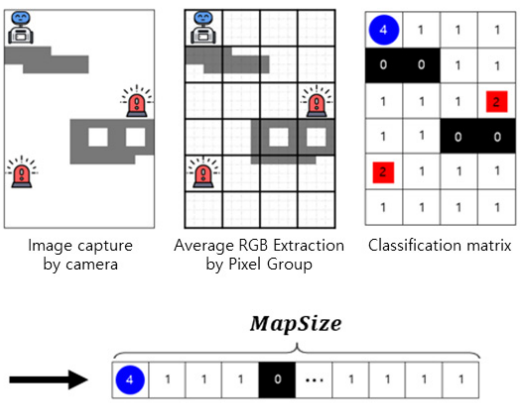

그림 4. RGB 데이터를 이용한 사물 인식
Fig. 4. Recognize objects using RGB data

표 1. 상태 변수
Table 1. State variable

type	wall	way	waypoint	robot's location
values	0	1	2	3

3.2 최적의 행동 Q 값 예측을 위한 DQN 심층 신경망 모델

그림 5는 이미지 프로세싱을 진행한 입력값에 대한 DQN 강화학습 모델을 나타낸 것이다. 강화학습 모델은 기본적으로 DQN 모델을 이용하며, 학습을 위한 에이전트인 메인 네트워크와 실제 값을 구하기 위한 타겟 네트워크 등으로 구성하였다. 또한, 과거 에피소드를 저장한 버퍼를 사용하였으며, 버퍼에서 일정 개수만큼 랜덤 샘플을 추출하여 학습하였다. Deep Q-Network는 에이전트 내부에 속하며 에이전트가 DQN을 거쳐서 나온 Q 함수의 값을 토대로 epsilon-greedy 방법을 사용해 행동을 선택하면 환경이 현재 상태와 행동에 따른 보상을 주고 다음 상태를 다시 에이전트에게 전달한다. <현재 상태, 행동, 보상, 다음 상태> 형태의 훈련 데이터(training set)가 되어 재현 저장소(replay memory)에 저장이 되고, 에이전트는 그중에서 지정된 배치(batch) 크기만큼 무작위로 훈련 데이터를 추출하여 손실 함숫값이 최소화되도록 타겟 네트워크 내부의 가중치(weight)와 바이어스(bias)를 업데이트한다. 다양한 상황에 대해 에피소드를 반복 학습하여 가중치와 바이어스의 과적합(overfitting)을 막고, 효율적인 학습을 진행한다. 이러한 경험 리플레이

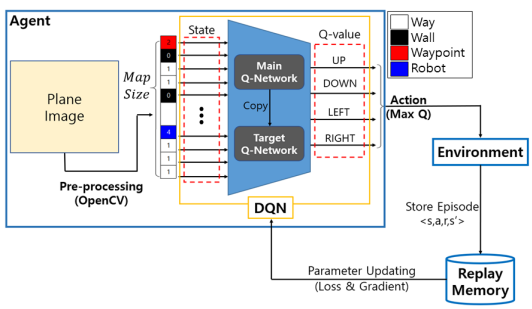

그림 5. 심층 Q-네트워크 모델
Fig. 5. Deep Q-Network model

와 네트워크 분리를 이용한 학습 과정은 알고리즘 1과 같다.

3.3 DQN 강화학습 모델 학습 방법

본 논문에서 최적 경로 탐색 시스템의 목적은 로봇이 장애물을 피하여 모든 경유지를 최단 시간으로 방문하는 것이다. 로봇이 모든 경유지에 방문하거나 허용 가능한 최소 보상 값의 총합($R_{\min}$)보다 보상 값의 총합이 작으면 즉시 에피소드가 종료된다. 그 외에는 로봇이 최적 정책 π^*에 의해 상태별로 행동을 epsilon-greedy 방식으로 선택하게 된다. 최적의 경로를 최단 시간으로 이동하기 위해 행동에 대한 누적 보상을 최대화하기 위해서 셀의 역할에 따라 보상의 값이 다르게 산정된다. 행동에 대한 보상함수는 3장 4절에서 자세히 설명한다. 로봇은 경유지를 향해 이동하면서 허용 가능한 최소 보상 값의 총합보다 보상 값의 총합이 커야 하는 방향으로 학습하기에 에피소드가 진행될수록 목표지점까지 최단 거리로 도달하며 도달시간 또한 감소한다. 그러나 이산적인 환경과 그에 해당하는 행동 값 하나만을 도출하므로 바로 앞의 상태 정보 말고는 그 전에 어떠한 환경이 로봇 주위에 있었는지, 어떠한 행동을 취했는지 알 수가 없게 돼 같은 행동을 반복하는 문제가 발생할 수 있다[13]. 이를 해결하기 위하여 딥 마인드에서 제안한 알고리즘을 적용한다. 네트워크의 출력인 Q 값, 즉 로봇의 행동에 해당하는 행동의 값을 버퍼에 저장한다. 이 방법을 이용하면 새로운 학습 없이 이전의 행동 기억을 기반으로 다음 행동을 예측하여 현재의 행동을 정할 수 있다.DQN의 또 다른 문제점은 선택할 수 있는 행동을 과대평가한다는 것이다. 에이전트가 선택할 수 있는 이상적인 행동이 있음에도 불구하고, 발견하지 못하고 특정 행동만을 고집하는 현상이며, 더 좋은 해법에 다가가지 못하는 문제가 발생한다. 이 문제를 해결하기 위해 주 신경망이 행동을 선택하며 타겟 신경망은 행동에 대한 타겟 Q 값을 생성하고, 타겟 네트워크의 수치를 업데이트하는 방법을 사용하였다. 이를 통하여 특정 행동의 과대평가를 줄일 수 있고 빠른 학습을 진행할 수 있게 된다.

3.4 심층 신경망 모델 조절을 위한 활성화 함수

딥 러닝 네트워크에서는 현재 층에서 다음 층으로 전달하기 전에 비선형 활성화 함수를 통과시킨다. 활성화 함수를 사용하지 않으면 아무리 깊은 모델을 사용하더라도 1개의 레이어만 사용하는 모델과 같아지게 되어 모델의 효율이 떨어지게 된다[14].

활성화 함수 중 ReLU(Rectified Linear Unit) 함수는 수식으로 표현하면 식 (6)과 같다.

$$f(x) = \max(0, x) \tag{6}$$

알고리즘 1. 경험 리플레이와 네트워크 분리를 이용한 DQN 강화학습
Algorithm 1. DQN Reinforcement Learning Using Experience Replay and Separation Networks

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
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 $a_t = \arg max Q_a(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D
    Sample random minibatch of transitions $(\phi_t, a_t, r_t, \phi_{t+1})$ from D
    Set $y_i = \begin{cases} r_j & \text{if } episode\ terminates\ at\ step\ j+1 \\ r_j + \gamma \bullet \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^-) & otherwise \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters θ
    Every C steps reset $\hat{Q} = Q$
  End For
End For

x가 0보다 크면 기울기가 1인 직선이고, x가 0보다 작으면 함숫값이 0이 된다. 이 함수는 연산 비용이 크지 않고 구현이 매우 간단하여, 다른 활성화 함수와 비교했을 때, 학습이 더 빠르게 진행된다. 하지만 x가 0보다 작을 때 뉴런이 더 이상 학습하지 못하고 죽어버리는 Dying ReLU 문제가 발생한다. PReLU 함수는 ReLU의 뉴런이 죽는 Dying RuLU 현상을 해결하기 위해 사용되는 함수이다. 함수를 수식으로 나타내면 식 (7)과 같다.

$$f(x) = \max(\alpha x, x) \tag{7}$$

PReLU 함수는 ReLU와 거의 유사하지만 새로운 파라미터 α를 추가하여 x가 0보다 작을 때에서도 기울기를 학습할 수 있도록 한다. 음수의 x값에 대해 미분 값이 0이 되지 않는다는 점을 제외하면 ReLU와 같은 특성을 갖는다.

3.5 보상함수와 손실 함수

보상함수의 계산은 셀을 이동할 때 각 셀에서의 상태를 통해 계산되는 경우와 셀을 재방문하는 경우로 나눌 수 있다. 수식으로 표현하면 수식 (8), (9), (10), (11), (12) 와 같다.

$$R_{normal} = -\frac{1}{MapSize} \tag{8}$$

$$R^{i}_{waypoint} = \begin{cases} i \times \sqrt{MapSize} & \text{if } visit \text{ for the } 1^{st} time \\ R_{normal} & otherwise \end{cases} \tag{9}$$

$$R_{revisit} = -\frac{MapSize}{3} \tag{10}$$

$$R_{total} = \sum R_{normal} + \sum_{i=1}^{N} R^{i}_{waypoint} + R_{revisit} \tag{11}$$

$$R_{\min} = -\frac{MapSize}{2} \tag{12}$$

여기서 R_{normal}, $R^{i}_{waypoint}$는 차례대로 일반적인 셀, i번째 방문하는 경유지, 그리고 R_{dest}는 최종적으로 목적지를 방문했을 때의 보상 값이다. 그리고 R_{total}은 이 세 가지 보상 값을 모두 합한 값이며, $R_{\min}$은 보상 값이 증가하지 않고 진행되는 경우의 에피소드 종료 조건을 위한 상수, 즉 허용 가능한 최소 보상 값의 총합이다. 따라서 $R_{total} \le R_{\min}$조건이 성립하면 최적 경로 탐색 실패로 간주하여 에피소드를 종료하고 경로 탐색을 처음부터 다시 수행한다. N은 공장에 존재하는 경유지의 총 개수이다. 식 (8)은 로봇이 일반적인 셀을 방문할 때마다 보상 값이 감소한다는 것을 의미하고 식 (9)는 어느 한 경유지를 처음 방문했을 때는 해당 경유지에 대한 보상을 주는 반면, 경로를 찾는 도중 해당 경유지를 재방문한 경우 일반적인 셀을 방문한 보상 값을 할당하는 것을 나타낸다. 그리고 식 (9)에서 처음 방문하는 경유지에 대한 보상은 2차원 환경에서 경로 탐색을 하는 동안 식 (8)에 의해 차감되는 보상과 2차원 공장 면적을 고려하였으며 한 경유지 주위를 배회하지 않고 다른 경유지로 이동하도록 각 경유지에 선형적으로 증가하는 보상 값을 할당했다. 식 (10)은 로봇이 주행하면서 방문했던 셀을 재방문했을 때 차감되는 보상 값이다. 최적 경로를 찾지 못하고 어느 셀에 대해 재방문을 하게 되면 $R_{total} \le R_{\min}$조건이 성립하여 경로 탐색을 처음부터 다시 수행한다. 식 (12)는 에피소드 종료 조건인 $R_{\min}$을 정의한 것이다.

타겟 네트워크에서의 Q-value값을 이용하여 실제 값을 산출하기 위해 식 (13)의 Bellman 수식을 사용한다.

$$Q^{*}(s,a;\theta^{-}) = \begin{cases} R(s,a) & \text{if } episode \, terminates \, at \, state \, s' \\ R(s,a) + \gamma \bullet \max_{a'} \hat{Q}(s',a';\theta^{-}) & otherwise \end{cases} \tag{13}$$

$R(s,a)$ 는 현재 상태(s)에서 행동(a)을 수행했을 때 보상 값이며, 그 이후 다음 상태는 s' 로 정의한다. $\max_{a'} \hat{Q}(s',a';\theta^{-})$는 네트워크 파라미터 θ^{-} 를 갖는 타겟 네트워크에서 다음 상태(s')일 때 해당하는 최대 Q 값이다. 할인 요소 γ에 의해 다음 상태에서 얻는 최대 Q 값이 점점 감소하여 더해지므로 로봇은 최단 경로로 이동하게 된다. 이를 이용하여 최종적으로 손실 함수는 식 (14)으로 정의된다.

$$L = \frac{1}{2}\{Q^{*}(s,a;\theta^{-}) - Q(s,a;\theta)\}^{2} \tag{14}$$

여기서 $Q(s,a;\theta)$는 메인 네트워크에서 예측값을 산출한 것이며, 실제 값과 예측값의 오차를 감소시키는 방향으로 경사 하강법(gradient descent)을 진행하고 주기적으로 메인 네트워크를 타겟 네트워크로 복제하며 메인 네트워크를 학습한다.

Ⅳ. 모의 실험 및 성능 평가

4.1 시뮬레이터 구현

스마트 팩토리에서 자율 주행 로봇이 최적 방향을 예측하는지 확인하기 위해 시뮬레이터를 구현하였다. 시뮬레이터는 경로 탐색에 영향을 미칠 수 있는 파라미터들을 이용하여 표 2의 값을 사용해 구현하였다. 시뮬레이터는 파이썬을 이용하여 구현하였으며, 그래픽 유저 인터페이스(GUI)를 이용하여 동작을 확인할 수 있도록 시각화하였다.

시뮬레이터를 통해 가로 셀의 개수가 X, 세로 셀의 개수가 Y인 셀 집합이 형성되고, 각 셀은 구조물 역할에 따라 색으로 구분된다. 공장 내부를 촬영하고 전처리 과정을 진행하면 그림 6과 같이 그래픽 유저 인터페이스에서 길, 로봇, 경유지, 벽을 포함한 가상 환경이 생성된다. 로봇은 상하좌우 네 방향으로 이동할 수 있으며, 이동 후에 공장 내부 촬영을 반복한다. 전체 내부 상황이 한 상태가 되고, 로봇이 이동한 뒤에는 새로운 상태가 된다. 로봇이 경유지를 방문하게 되면 수리가 완료되었다고 판단하여 경유지 아이콘이 사라지고 일반적인 길의 형태로 가상 환경이 업데이트된다. 보상 값의 총합이 허용 가능한 최소 보상 값의 총합보다 크고 모든 경유지를 방문했을 때, 해당 에피소드를 성공이라고 판단한다. 또한, 배치에 대한 에피소드 성공률에 따라 epsilon 값을 변경하여 효율적인 강화학습을 진행하였다. 모든 에피소드는 버퍼에 저장되고 버퍼의 공간이 부족하면 무작위로 한 개의 에피소드를 삭제하고 새로운 에피소드가 추가되도록 하여 버퍼 크기를 유지한다. 버퍼에서 일정 개수만큼 랜덤 샘플을 추출하여 DQN 모델을 학습하였다.

그림 6(a)는 촬영 이미지를 이미지 전처리 과정을 통해 가상 환경에 반영한 모습이다. 그림 6(b)는 현재 위치에서 벽을 피해 모든 경유지를 최단 경로로 방문하는 것을 나타낸다. 그림 6(c)는 벽과 경유지의 위치

표 2. 학습 매개 변수 및 시뮬레이션 파라미터
Table 2. Learning parameters and simulation parameters

Parameter	Values	Parameter	Values
Epsilon(ϵ)	0.1	Minimum Reward(R_{min})	-12
Discount Factor(γ)	0.95	Replay Buffer Size (N)	50000
Batch Size	64	Random Sampling	32
X, Y Size	4, 6	MapSize	24

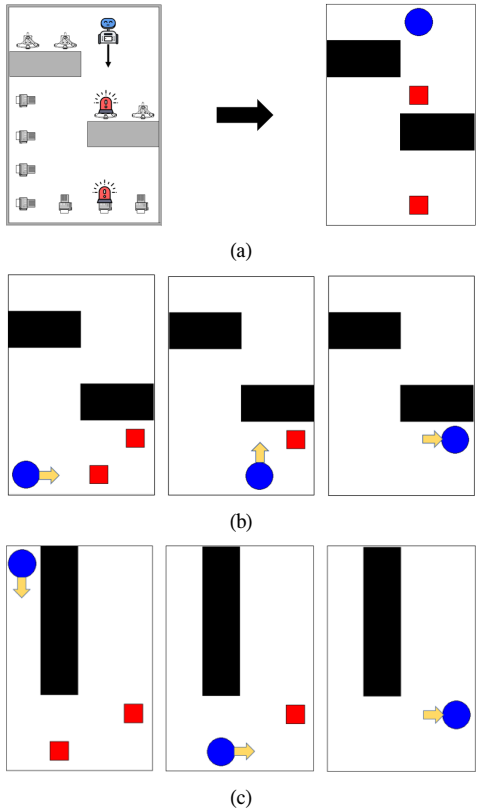

그림 6. 로봇 최적 경로 예측 시뮬레이션, (a) 가상 시뮬레이션 환경 생성, (b) 시뮬레이션 테스트, (c) 지형 및 경유지 변화
Fig. 6. Simulation of robot's optimal-path prediction, (a) creating a virtual simulation environment, (b) simulation test, (c) changing the structure and waypoints

가 바뀌어 공장 구조가 변형되었을 때 최단 경로를 탐색한 것을 나타낸다. 결과적으로, 다양한 학습 데이터를 훈련했을 때 정확도가 증가하였으며, 학습시키지 않은 새로운 상황에서도 최적 경로를 탐색하는 것을 확인하였다.

DQN 모델은 표 3과 같은 구조를 이용하여 구현하였다. 모델 작성에는 파이썬(python)을 이용하였으며, 딥 러닝 라이브러리(deep learning library)로는 케라스(keras)를 사용하였다. 최적 경로 추정 모델은 3개의 은닉 층(hidden layer)과 1개의 출력 층(output layer)으로 구성이 되어있다. 은닉 층은 모두 FC 레이어로 구성하였으며 과적합 문제를 해결하기 위해 드랍아웃(dropout) 레이어를 사용하여 일부 노드를 사용하지 않도록 하였다[15]. 또한, FC 레이어마다 활성화 함수 PReLU를 추가하여 네트워크 모델의 성능을 개선하였다. 학습 시에는 표 3의 비용 함수(cost function)를 이용하였으며, Global Minimum을 찾기 위해 최적화 알

표 3. 최적 경로 예측 모델
Table 3. Optimal path prediction model

DNN Model			
Layers		Nodes	Rate
Input Layer	FC Layer	24	-
Hidden Layer	FC Layer	72	-
	Dropout	-	0.2
	FC Layer	72	-
	Dropout	-	0.2
	FC Layer	72	-
	Dropout	-	0.2
Output Layer	FC Layer	4	-
Optimizer	Cost Function	Error Rate Function	
Adam Optimizer	$\frac{1}{m}\sum_{i=1}^{m}(\hat{Y}_i - Y_i)^2$	$\frac{1}{m}\sum_{i=1}^{m}(\frac{\hat{Y}_i - Y_i}{Y_i})\times 100(\%)$	

고리즘으로 gradient decent optimizer에 momentum 알고리즘을 추가한 adam optimizer을 사용하였다[16]. 마지막으로 오차율 함수(error rate function)를 사용하여 성능 측정을 진행하였다.

4.2 로봇 최적 경로 예측 모델 성능 평가

최적 경로 탐색 시뮬레이션은 그림 6과 같이 진행한다. 촬영한 이미지가 사용자가 보기 쉬운 그래픽 유저 인터페이스 형태로 변환되고 시뮬레이션이 진행된다. 로봇은 벽을 피해 모든 경유지에 방문하도록 이동하며, 행동에 의해 바뀐 새로운 상태에서 얻게 되는 보상은 3장 5절에서 소개한 수식 (8), (9), (10)을 사용하여 얻는다. 최종적으로 가장 높은 보상 값의 총합을 갖도록 로봇이 최적 경로로 이동하게 된다.

epoch에 따른 손실과 보상의 변화는 그림 7, 8과 같이 나타낼 수 있다.

차례대로 각 epoch에 해당하는 손실 값, 전체 경로

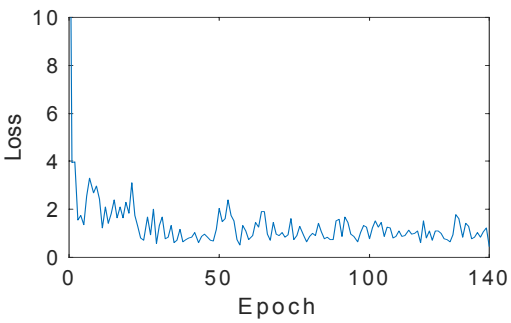

그림 7. Epoch에 따른 손실 변화
Fig. 7. Loss variation by epoch

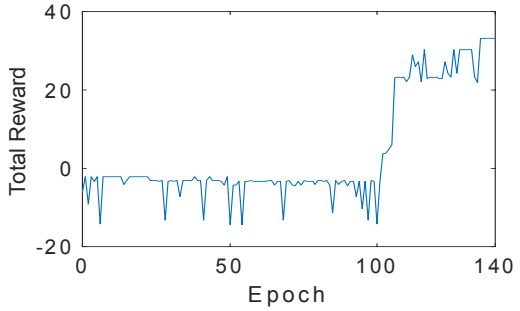

그림 8. Epoch에 따른 보상 값의 총합 변화
Fig. 8. Variation of compensation value according to epoch

에서 얻을 수 있는 보상 값의 총합이며, 학습에 사용된 맵의 구조 및 변수는 표 2와 같다.

학습 초반에는 로봇의 탐험률이 높아서 손실 값이 크지만, 학습이 진행되면서 보상 값이 큰 셀을 인지하고 탐험률도 감소하기 때문에 손실 값이 줄어드는 결과를 보인다. 반면 셀을 이동하면서 얻은 보상 값의 총합은 학습이 진행될수록 증가하게 된다. 손실 값과 전체 경로에서 얻을 수 있는 보상 값의 총합을 통해 로봇이 강화학습을 통해 최적의 경로를 도출해 낸다는 것을 알 수 있다.

그림 9는 모든 경유지를 방문하는 경로를 발견하였을 때부터 평균 이동 거리를 나타낸 것이다. 모든 경유지를 방문하는 최단 경로를 찾아가는 것을 확인하기 위해 50개의 환경에서 평균적인 이동 거리를 확인하였다. 맨 처음 모든 경유지를 발견하였을 때부터 20번의 에피소드를 더 진행하여 최단 거리를 찾는 학습을 진행하였다. epsilon-greedy 방식과 사전에 정의된 손실 함수와 최적 정책을 통해 네트워크를 업데이트하며, 학습이 진행될수록 처음의 모든 경유지를 방문하는 경로보다 더 짧은 이동 거리를 갖는 경로를 찾는 것을 확인할 수 있다.

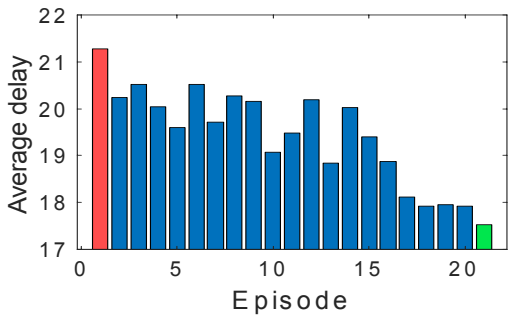

그림 9. 에피소드에 따른 평균 시간 변화
Fig. 9. Average time to reach according to episode

V. 결 론

본 논문에서는 강화 학습의 한 종류인 DQN을 통해 공장 내부의 구조 및 상황들을 고려하여 효율적으로 최적 경로를 도출하는 방법을 제안하였다. 수시로 변하는 공정 라인 배치와 고장 발생 위치에서도 시간을 최소화하고 경유지를 모두 방문하는 최적의 경로 탐색을 목표로 하였으며, 시뮬레이션을 통해 이를 확인하였다. 다양한 공장 라인의 변화와 상황 변화를 수용할 수 있도록 강화학습 중 DQN을 이용하여 수많은 상태에 대해 처리가 가능한 네트워크를 구현하였으며, 학습 시 epsilon-greedy 방식으로 탐험하여 효율적인 방향으로 Q 값과 네트워크 파라미터를 업데이트하는 것을 확인했다. 최종적으로 스마트 팩토리 환경의 다양하고 새로운 상황에서 로봇이 강화 학습된 데이터를 기반으로 즉각적으로 최적 경로를 탐색할 수 있는 DQN 시스템을 구현했다. 본 논문에서 제안하는 상황 변화에 능동적이고 유연하게 대처하는 시스템을 이용하여 스마트 팩토리에서의 에너지 절감 및 생산성 향상에 크게 기여할 수 있음을 보였다. 하지만 주어진 환경은 임의로 생성한 환경이기 때문에 추후 실제 공장에서 사용하는 이미지를 통해 모델을 보완하고 연구할 필요가 있다.

References

[1] S. Wang, J. Wan, D. Li, and C. Zhang "Implementing smart factory of industrie 4.0: An outlook," *Int. J. Distrib. Sensor Netw.,* vol. 12, no. 1, Jan. 2016.

[2] F. Shrouf, J. Ordieres, and G. Miragliotta, "Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm," *2014 IEEE Int. Conf. Ind. Eng. and Eng. Management*, pp. 697-701, Bandar Sunway, 2014.

[3] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," in *IEEE Access*, vol. 6, pp. 6505-6519, 2018.

[4] S. J. Jang and S. J. Yoo, "Q-learning-based dynamic joint control of interference and transmission opportunities for cognitive radio," *EURASIP J. Wireless Commun. and Netw.*, Jun. 2018.

[5] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning : A survey," *The J. Artificial Intell. Res.*, vol. 4, pp. 237-285, May 1996.

[6] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 26, pp. 529-541, Feb. 2015.

[7] H. J. Kwak and G. T. Park, "Optimal path-finding for mobile robots using reinforcement learning and genetic algorithm," *Inf. and Control Symp.,* pp. 139-140, Apr. 2010.

[8] B. C. Kim, S. G. Kim, and B. J. Yun, "Online reinforcement learning to search the shortest path in maze environments," *KIPS Trans.*, vol. 9, no. 2, pp. 155-162, Feb. 2002.

[9] S. H. Lee, D. H. Yeom, and P. I. Kim, "Local path planning and obstacle avoidance system based on reinforcement learning," in *Proc. Korean Soc. Comput. Inf. Conf.*, vol. 27, no. 1, pp. 59- 60, Jan. 2019.

[10] J. S. Moon, S. J. Jang, and S. J. Yoo "UAV route search based on reinforcement learning for efficient sensing data collection," in *Proc. Symp. KICS*, pp. 898-889, Jan. 2018.

[11] H. S. Yi, E. S. Park, and S. G. Kim, "Deep reinforcement learning for autonomous vehicle driving," *Korea Inf. Sci. Soc.,* pp. 784-786, Dec. 2017.

[12] I. Y. Hwang, S. H. Wang, S. M. No, and D. S. Eom, "Comparison and analysis of autonomous flight drones algorithm based on deep reinforcement learning," in *Proc. Symp. KICS,* pp. 630-631, Nov. 2018.

[13] K. S. Kim, J. W. Moon, D. K. Kim, S. B. Jo, and J. M. Lee, "Natural behavior learning based on deep reinforcement learning for autonomous navigation of mobile robots," *J. Inst. Control Robotics and Syst.*, vol. 24, no. 3, pp. 256-262, Mar. 2018.

[14] J. S. Han and K. C. Kwak, "Image classification using convolutional neural network and extreme learning machine classifier based on ReLU function," *The J. Korean Inst. Inf. Technol.*, vol. 15, no. 2, pp. 15-23, Feb. 2017.

[15] W. K. Yun, Y. J. Song, J. S. Moon, S. J. Jang,

and S. J. Yoo, “Deep learning based temperature sensor data and wildfire propagation prediction in duty cycled wireless sensor network,” *J. KICS*, vol. 44, no. 6, pp. 1092-1104, Apr. 2019.

[16] K. Diederik and J. Ba, “Adam: A method for stochastic optimization,” *arXiv* preprint, arXiv:1412.6980v9 [cs.LG], 2017.

박 광 석 (Kwang-Seok Park)

2014년 2월~현재 : 인하대학교 정보통신공학과 학사

<관심분야> IoT, 네트워크, 딥 러닝

박 진 만 (Jin-Man Park)

2015년 2월~현재 : 인하대학교 정보통신공학과 학사

<관심분야> 딥 러닝, 영상 처리

윤 완 규 (Wan-Kyu Yun)

2019년 2월 : 인하대학교 정보통신공학과 학사 졸업

2019년 2월~현재 : 인하대학교 정보통신공학과 석사

<관심분야> 이동통신, IoT, 무선 센서 네트워크, 머신러닝

유 상 조 (Sang-Jo Yoo)

1988년 2월 : 한양대학교 전자통신학과(공학사)

1990년 2월 : 한국과학기술원 전기 및 전자 공학과(공학석사)

2000년 8월 : 한국과학기술원 전자전산학과(공학박사)

1990년 3월~2001년 2월 : KT 연구 개발 본부

1990년 3월~2000년 11월 : NIST(미국 표준기술연구원) 초빙연구원

2001년 3월~현재 : 인하대학교 정보통신공학과 교수

<관심분야> 무선 네트워크 프로토콜, Cognitive Radio Network, 무선 센서 네트워크, 지능형 사물인터넷

[ORCID:0000-0003-1533-0814]