

강화학습을 이용한 레고 퍼즐 생성 기술 개발

박철성*, 양희경**, 민경하*

상명대학교 컴퓨터학과*, 상명대학교 SW융합학부**

tjd4987@naver.com, {yanghk, minkh}@smu.ac.kr

Reinforcement Learning-based Approach for Lego Puzzle Generation

Cheolseong Park*, Heekyung Yang**, Kyungha Min*

Dept. of Computer Science, Sangmyung Univ.*,

Div. of SW Convergence, Sangmyung Univ.**

요 약

2D 레고 퍼즐은 레고 브릭을 이용해서 다양한 영상을 완성하는 퍼즐로 많은 사람들의 사랑을 받고 있다. 본 연구에서는 입력된 픽셀 아트 영상으로부터 강화학습에 기반한 2D 레고 퍼즐을 구성하고 완성하는 방법을 제안한다. 먼저, 학습 단계에서는 바람직한 레고 퍼즐에 대한 휴리스틱을 안정성과 효율성으로 설정하고 이를 최대한 만족시키는 방향으로 DQN을 학습한다. 그리고, 레고화 단계에서는 이 DQN을 이용해서 실제 레고 브릭을 추가해가면서 퍼즐의 구성도를 완성하는 과정을 수행한다. 이 과정을 통해서 지금까지 기술로는 수행하기 힘들었던 매우 높은 해상도의 레고 퍼즐을 효율적으로 완성하는 기술을 제공한다.

ABSTRACT

We present a reinforcement learning-based framework for generating 2D Lego puzzle from input pixel art images. We devise heuristics for a proper Lego puzzle as stability and efficiency. We also design a DQN structure and train it to maximize the heuristics of 2D Lego puzzle. In legorization stage, we complete the layout of Lego puzzle by adding a Lego brick to the input image using the trained DQN. During this process, we devise a region of interest to reduce the computational loads of the legorization. Using this approach, our framework can present a very high resolutional Lego puzzle.

Keywords : Legorization (레고화), Reinforcement learning (강화학습), Heuristic (휴리스틱)

Received: Mar. 10. 2020 Revised: Apr. 03. 2020

Accepted: Jun. 08. 2020

Corresponding Author: Heekyung Yang and Kyungha Min
(Sangmyung University)

E-mail: {yanghk, minkh}@smu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

작은 조각을 맞추어서 영상을 완성하는 형태의 퍼즐은 오랫동안 사랑받아왔다. 널리 알려져있는 직소 퍼즐은 이러한 영상을 완성하는 형태의 퍼즐의 대표적인 예이다. 이러한 직소 퍼즐의 단점은 하나의 조각이라도 분실할 경우 전체 퍼즐을 완성할 수 없다는 점이다. 따라서, 많은 퍼즐 애호가들은 재사용이 가능한 조각을 이용하는 퍼즐에 대한 필요성을 느끼고 있다.

최근의 다양한 팝 아트의 영역 중에는 영상을 낮은 해상도의 픽셀화된 영상으로 변환하는 픽셀화 기법 (pixelization)을 적용한 작품들이 발표되고 있다. 또한, 비사실적 렌더링 분야의 여러 기술들 중에는 픽셀화된 영상을 자동으로 생성하는 기술에 연구되고 있다. 본 연구는 이러한 기술을 기반으로 낮은 해상도의 픽셀화 영상으로 변환된 영상에 대해서 레고의 기본 브릭을 이용해서 퍼즐을 개발하는 방법을 제안한다.

레고는 오랫동안 사랑받아온 완구로 제공되는 도면에 따라서 유사한 형태의 기본 조각들을 결합해서 다양한 물체를 완성한다. 본 연구에서는 레고의 기본 브릭들을 이용해서 픽셀화된 영상을 채워서 원하는 형태를 완성하는 2D 퍼즐을 생성하는 방법을 제안한다.

기존의 레고 퍼즐 생성 기술은 대부분 3D 레고 모델을 생성하는 방법을 제안하고 있으며, 3D 레고 모델의 계층을 하나씩 완성해서 최종적으로 전체적인 모델을 완성한다. 여기서 3D 레고 모델의 각 층을 완성하는 방법으로 바람직한 레고 모델에게 요구되는 성질 (heuristic)을 정의하고, 이를 목적 함수로 정의한 다음, 이 목적 함수를 최대한 만족시키는 구성도 (layout)을 찾는다. 이 목적 함수를 만족시키기 위해서 다양한 방법들이 제시되었지만, 이를 완전하게 만족시키는 방법에 대한 연구는 계속되고 있다.

본 연구에서는 최근 발전하고 있는 심층 학습 기법에 기반한 강화 학습 모델을 이용하는 방법을

제안한다. 레고 모델 생성 과정을 게임으로 해석하여 레고 브릭을 하나씩 추가하는 행위를 게임의 한 행위로 처리한다. 따라서 레고의 구성도를 완성하게 되면, 게임이 종료되는 것으로 해석하여, 구성도의 보상 (reward)을 최대화시킴으로써 가장 완성도가 높은 레고의 구성도를 완성하는 목표를 추구한다.

본 논문은 다음과 같이 구성된다. 먼저, 2장에서 관련 연구를 요약하고 3장에서 연구의 전체적인 구성도를 제시한다. 4장에서 강화 학습을 이용한 학습 방법을 설명하고 5장에서 레고 생성 기법을 제시한다. 그리고 6장에서 결과를 보여주고 평가하며 7장에서 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 3D 레고 모델 연구

초기의 3D 레고 모델 생성 연구에서는 바람직한 레고 모델의 성질을 정의한 휴리스틱을 만족시키는 지의 여부를 판정하는 벌칙 함수 (penalty function)을 최소화하는 방법이 제안되었다 Gower 등은 처음으로 레고 모델에게 요구되는 6 개의 휴리스틱을 정의하였는데, 이 휴리스틱은 cover ratio, bigger bricks, perpendicularity, vertical boundary, T-shaped joining 등이다[1]. 이 연구에서 제시한 휴리스틱들을 최소화하는 방법으로 Petrobic은 유전 알고리즘을 이용하는 방법을 제시하였으며[2], van Zijl 과 Smal은 셀룰라 오토메타를 이용하는 방법을 제시하였다[3].

최근에는 Testuz 등이 그래프에 기반한 탐욕 알고리즘을 제안하였다[4]. 이 방법에서는 1x1 크기의 브릭으로 레고 모델을 생성하고, 이 브릭들을 결합해서 다양한 크기와 형태의 브릭으로 치환한다. 이 과정에서 그래프 모델을 이용해서 브릭의 연결성을 파악하고 레고 모델의 물리적 불안정성을 해소하는 방법을 구현하였다. Ono 등은 Testuz의 방법과 비슷한 그래프 모델을 도입해서 Gower 등

이 제안한 6 가지 휴리스틱을 만족시키는 방법을 제안하였다[5]. Lee 등은 탐욕 알고리즘과 유전 알고리즘을 결합해서 1x1 크기의 브릭을 조합하는 방법을 제안하였고[6], Zhang 등은 안정성과 대칭성을 고려하는 랜덤 결합 방법을 제안하였다[7]. 최근 Yun 등은 레고 모델의 휴리스틱을 안정성, 효율성 및 미학성으로 정리하고, 이를 만족하는 레고 모델의 구성도를 A* 탐색 기법을 이용해서 찾는 방법을 제시하였다[8].

3D 레고 모델의 물리적 안정성을 추구하기 위해서 Luo 등은 브릭들 사이의 마찰력과 한 브릭이 부담할 수 있는 최대 하중 등과 같은 물리적인 속성을 고려하는 자동화된 3D 레고 모델 생성 기법을 제안하였다[9]. 이 기법에서는 레고 모델을 그래프로 표현하고 그래프의 각 노드에서 부가되는 하중을 측정해서 안정성을 추구하는 방법을 구현하였다. Kosaki 등은 다양한 각도에서 촬영된 사진을 이용해서 레고 모델의 3D 형태를 복원하여 3D 모델을 생성하는 기술은 제안하였다[10]. 이 연구에서는 레고 모델의 물리적 안정성을 향상시키기 위해서 모멘텀 값을 측정해서 최소화하는 방법을 추가하였다.

2.2 2D 레고 퍼즐 연구

Zhang 등은 사용자의 개입을 받아서 2D 레고 퍼즐을 생성하는 시스템을 제안하였다[11]. 이 방법에서는 입력 영상으로부터 생성된 특징 맵에 특정한 패턴 타일을 적용해서 레고 퍼즐을 생성한다. 이 과정에서 사용자의 입력으로부터 D-feature라는 특징을 추출해서 이를 이용해서 패턴 타일을 수정하는 브릭 편집 도구를 개발한다. 이 방법은 사용자가 친숙하게 2D 레고 모델을 생성하는 환경을 제공하지만 자동화된 생성을 제공하지 못한다는 단점이 있다.

Kuo 등은 픽셀 아트 영상으로부터 2D 레고 퍼즐을 생성하는 방법을 제안하였다[12]. 이 연구에서는 적절한 중심점을 얻기 위해서 픽셀 아트 영상의 형태를 변형하고 영상의 색을 사용가능한 색

팔레트의 색으로 변형한다. 그리고, 브릭의 면이 덮고 있는 다른 브릭의 면을 수정해서 모델에서 떨어져나가는 브릭이 없도록 모델을 수정하였다.

이 두 연구에서는 픽셀 아트 영상을 이용하는데, 두 연구 모두 레고 모델로 변형할 수 있는 픽셀 아트 영상의 해상도가 낮다는 단점이 있다. 그 이유는 두 연구 모두 탐욕 알고리즘에 기반한 방법으로 최적화를 수행한다는 점이다. 본 연구에서는 이러한 단점을 해소하기 위해서 강화 학습에 기반한 최적화 방법을 제안한다.

2.3 2D 강화 학습

강화 학습 (RL, Reinforcement Learning)은 행태 심리학에 기반한 기계 학습의 한 분야이며, 행위자 (agent)가 특정한 환경 (environment)에서 행위 (action)를 반복하면서 축적된 보상 (reward)를 최대화하는 기술을 연구한다[13]. 강화 학습 모델은 레이블이 적용된 훈련 모델을 사용하지 않고, 상태, 행위, 보상을 각 단계에서 기록하고 선택한 행위의 보상에 따라서 모델을 훈련한다.

Mnih 등은 기존의 강화 학습 모델에 심층 학습 모델을 적용한 심층 Q 네트워크 (DQN, Deep Q-Network)를 제안하고 이를 아타리 게임에 적용하였는데, 사람 플레이어보다 더 높은 성능을 보였다[14]. 이 방법은 기존의 Q 네트워크 기반의 강화 학습 모델을 개선하였는데, 지나치게 많은 연산으로 인한 과도 평가 문제 (overestimation problem)이 존재한다.

Hasselt 등은 보상의 최대값을 구하는 과정을 개별적인 네트워크에서 처리하는 방법으로 과도 평가 문제를 해소하는 이중 DQN (double DQN) 방법을 제안하였다[15]. 그리고, Schaul 등은 더 훈련 효율이 높은 행위를 선택하도록 우선 순위를 부여하는 방법으로 강화 학습의 성능을 향상시키는 방법을 제안하였다[16]. Bellemare 등은 distributional DQN 방법을 제안하였는데, 이 방법에서는 행위의 결과값이 갖는 분포를 근사적으로 표현하는 확률 모델을 찾고, 이 모델을 이용해서 행위를 선택한다[17].

Wang 등은 기존의 DQN보다 더 빠른 성능을 보이는 dueling DQN 방법을 제안하였다[18]. 이 방법에서는 행위의 결과를 측정하는 가치 함수 V 를 가치 함수 V 와 우선 함수 A 로 분리해서 처리함으로써 연산의 효율성을 추구하였다.

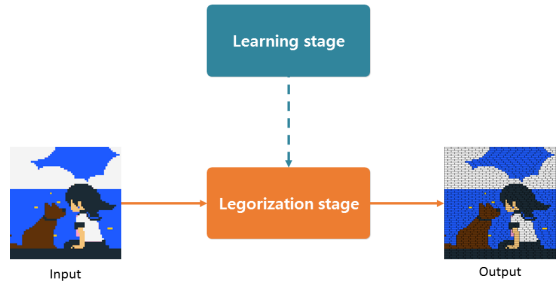
3. 제안하는 논문의 개요

본 연구에서는 낮은 해상도의 픽셀 아트 영상을 입력으로 받아서 강화학습의 하나인 DQN 모델을 적용해서 2D 레고 퍼즐을 생성하는 방법을 제안한다. 본 연구의 결과는 레고 브릭이 입력 영상과 유사한 형태를 갖도록 나열된 구성도이다. 우리 연구에서 레고화 과정은 가장 높은 보상을 획득하는 브릭의 구성도를 생성하는 게임으로 간주된다. 레고화 과정은 먼저 적절한 레고 브릭을 선택하는 규칙을 학습하는 학습 단계 (learning stage)와 이 규칙을 적용해서 레고 퍼즐을 완성하는 레고화 단계 (legorization stage)로 구성된다 [Fig. 1].

학습 단계에서는 주어진 상태에서 적절한 브릭을 선택하는 규칙을 DQN을 이용해서 학습한다.

레고 브릭들의 현재 상태의 구성도를 DQN의 상태 (state)로 간주하고 구성도의 픽셀을 채우는 브릭을 선택하는 연산을 행위 (action)으로 간주한다. 한 상태의 구성도에서 브릭을 추가함으로써 다음 상태로 이동하는데, 입력 영상의 모든 픽셀을 덮는 구성도를 마지막 상태 (final state)로 정의한다. 이를 위해서 DQN이 가장 높은 보상을 받는 브릭을 선택하는 규칙을 학습한다.

레고 생성 단계에서는 학습된 DQN을 적용해서 입력된 픽셀 아트 영상의 픽셀들을 레고 브릭으로 채우는 과정을 반복한다. 이 과정에 선택할 브릭의 후보로 모든 브릭을 고려하는 동적 프로그래밍 기법을 적용할 경우, 매우 긴 연산 시간이 요구된다. 따라서, 본 연구에서는 다음의 두 가지 전략을 적용해서 연산 시간을 절약한다. 첫째, DQN은 모든 브릭을 후보로 고려하지 않고, 가장 적당한 두



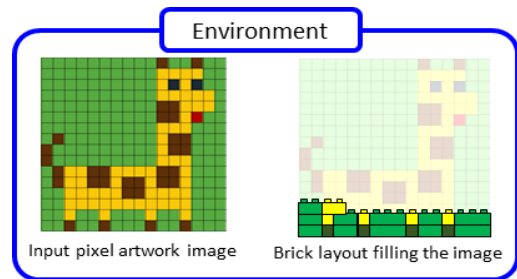
[Fig. 1] The overview of our framework

개의 브릭만을 후보로 고려한다. 두번째, 입력 영상의 모든 픽셀들을 고려하지 않고, 적절한 관심 영역 (ROI, Region of Interest)을 설정하고, 이 관심 영역 내의 레고화만 고려한다. 이러한 두 가지 전략을 통해서 탐색 공간을 축소함으로써 효율적으로 레고 생성 단계를 수행한다.

4. 제안하는 논문의 학습 단계

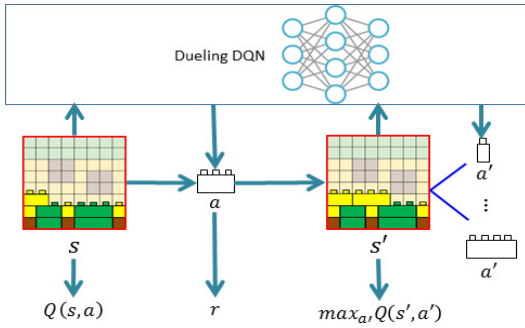
4.1 강화 학습

강화 학습의 환경은 상태 (s), 행위 (a), 보상 (r) 등 학습에 필요한 모든 요소들을 포함하는 개념이다. 이 환경 내에서 어떤 상태에서의 행위를 통해서 새로운 상태로의 전이가 이루어진다. 레고화를 위한 환경을 입력된 픽셀 아트 영상과 영상을 채우고 있는 브릭의 구성도의 조합으로 표현한다 [Fig. 2].



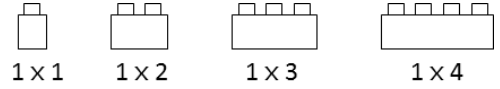
[Fig. 2] Environment of our framework

상태 s 는 입력된 영상을 채우는 레고 브릭의 구성도로 영상의 좌측 하단에 있는 픽셀부터 수평으로 브릭을 배치하면서 영상을 채운다. 따라서 영상에 브릭이 하나도 배치되지 않은 상태가 초기 상태이며, 영상의 모든 픽셀을 브릭이 가득 채운 상태가 최종 상태이다.



[Fig. 3] The process of DQN

각 상태에서 DQN 알고리즘은 가장 큰 보상 r 을 줄 것으로 기대되는 행위 a 를 선택한다. 이 환경에서 행위는 현 상태에서 다음 픽셀을 채우는 브릭을 선택하는 것이다. 이 브릭의 선택을 통해서 현재 상태 s 는 다음 상태 s' 으로 전이된다. 이 과정은 [Fig. 3]에서 설명되어 있다. DQN은 레고화가 실패하거나 최종 상태에 도달할 때까지 반복적으로 행위를 수행하면서 훈련된다. 최종 상태에 도달하면 그때까지 적용한 보상의 합을 계산하여 손실 함수를 측정하고 이를 역전파하는 방식으로 DQN을 훈련한다. 각 상태에서 보상을 고려하는 대상을 전체 픽셀로 설정할 경우 지나치게 많은 시간이 소요되기 때문에 $k \times k$ 크기의 관심 영역 (ROI)을 설정하고, 이 영역에 대한 보상을 고려한다. 행위는 하나의 브릭을 선택하는데, 본 연구에서는 [Fig. 4]에서 제시되는 4 개의 브릭만을 대상으로 설정한다. 이는 이 브릭들이 가장 흔하게 사용되는 브릭이기 때문이며, 실제 레고 모델을 완성하기 위해서 구입하는 브릭의 가격 및 구입의 용이성을 고려할 때, 이 브릭들 이외의 브릭은 구입이 힘들거나 가격이 비싸기 때문이다.



[Fig. 4] Four candidate bricks

4.2 휴리스틱

각 행위의 보상은 배치되는 브릭이 얼마나 레고 구성도의 완성도를 높이는지에 따라서 결정된다. 많은 연구자들이 Gower et al. 등이 제시한 6 가지 휴리스틱을 만족시키는 레고 구성도를 추구하는데, 이는 3D 레고에 대한 휴리스틱이기 때문에 본 연구와는 일치하지 않는다. 따라서, 본 연구에서는 2D 레고 퍼즐의 특성을 고려해서 효율성 (efficiency) 및 안정성 (stability)을 측정하는 휴리스틱을 고안한다. 이 휴리스틱에 기반한 보상은 [Fig. 5]에서 제시되어 있다.

4.2.1 효율성

첫번째 휴리스틱은 효율성으로 가능하면 큰 브릭을 이용할 경우에 효율성이 높아지는 것으로 판단한다. 따라서 각각의 브릭에 대해서 그 크기에 따라서 다음과 같은 가중치를 부여한다.

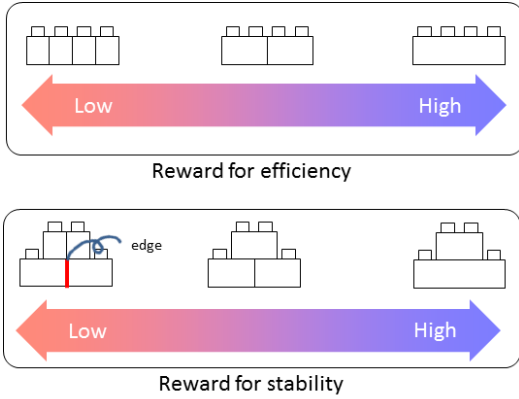
$$brick_weight = \begin{cases} 0, & \text{if brick is } 1 \times 1 \\ 1/2, & \text{if brick is } 1 \times 2 \\ 3/4, & \text{if brick is } 1 \times 3 \\ 1, & \text{if brick is } 1 \times 4 \end{cases}$$

따라서 브릭 구성도의 효율성에 대한 보상 (r_e)는 다음과 같이 정의된다 (n 은 brick.의 수).

$$r_e = \frac{1}{n} \sum brick_weight$$

4.2.2 안정성

두번째 휴리스틱은 안정성으로 레고 구성도의 각 브릭들의 수직 단면 (edge)이 서로 겹치지 않



[Fig. 5] Rewards for efficiency and stability

도록 배치됨을 목적으로 한다. 따라서, 모든 수직 단면에 대해서 겹치는지 겹치지 않는지를 판단하여 서로 겹치도록 배치되지 않은 수직 단면의 경우 1/2을 부여하고, 이 값의 평균을 구해서 안정성에 대한 보상 (r_s)로 정의한다 (m 은 수직 단면의 수).

$$r_s = \frac{1}{m} \sum_{i=1}^m e_i, \quad e_i = \begin{cases} 1/2, & \text{if } edge_i \text{ is covered} \\ 0, & \text{otherwise} \end{cases}$$

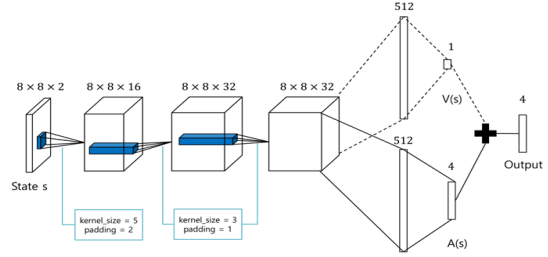
따라서 완성된 레고 구성도의 보상은 다음과 같이 측정된다.

$$r = w_e r_e + w_s r_s$$

4.3 DQN 학습

4.3.1 네트워크의 구조

Wang 등이 제안한 dueling DQN의 구조를 이용한다. 입력은 $k \times k \times 2$ 크기의 배열로 입력된 영상과 레고 퍼즐의 초기 상태를 나타낸다. 이에 대해서 세번의 연속된 합성곱 연산을 수행한다. 첫 번째 합성곱에서 사용되는 필터는 5×5 크기이며, 입력 영상의 크기가 감소하지 않도록 두께 2의 padding을 적용한다. 두번째와 세번째 합성곱에서는 3×3 크기의 필터와 두께 1의 padding을 사용한다.



[Fig. 6] The structure of DQN ($k = 8$).

다. 이 합성곱 연산 후에 두 개의 완전 연결 계층을 배치한다. 이 구조는 [Fig. 6]에서 제시되어 있다. 최종 보상은 $V(s)$ 와 $A(s, a)$ 의 합을 계산해서 측정된다.

4.3.2 Q 함수

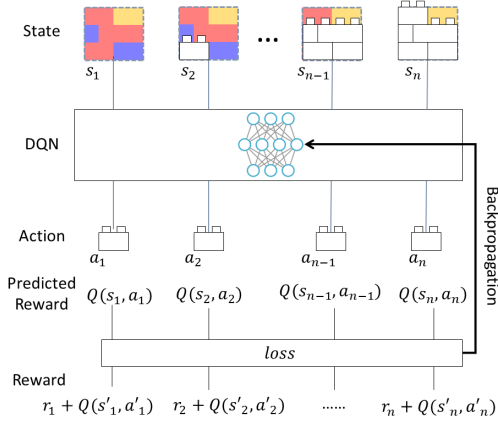
$Q(s, a)$ 로 표현되는 Q 함수는 최종 보상에 대한 측정값을 계산한다. 따라서, 상태 s 에서 행위 a 를 통해서 r 이라는 보상을 받고 s' 이라는 상태로 전이된다면, 이 관계는 앞으로의 예측값에 대한 가중치 γ 를 적용해서 다음과 같이 정의된다.

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

4.3.3 훈련 과정

DQN에 대한 훈련은 역전파 과정을 이용한다. 역전파를 위해서 우리는 우리가 원하는 값과 계산된 값 사이의 오차를 측정한다. 우리가 원하는 값은 레고화 단계를 통해서 계산된 Q 함수를 통해서 기록된 값이다. $Q(s, a)$ 가 최종 상태에서 계산된 값이기 때문에 우리는 현재의 보상 r 에 대해서 다음 단계의 보상의 예측값인 $\gamma \max_{a'} Q(s', a')$ 의 차이를 오차로 측정한다. 이 오차는 다음과 같이 정의되며, 역전파를 수행하는 과정은 [Fig. 7]에서 제시된다.

$$loss = \frac{1}{2} (r + \gamma \max_{a'} Q(s', a') - Q(s, a))^2$$



[Fig. 7] The process of learning through backpropagation

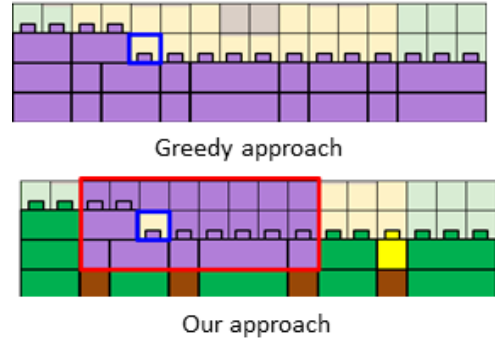
5. 제안하는 논문의 레고화 단계

5.1 입력

레고화 단계의 입력은 낮은 해상도의 픽셀 아트 영상이다. 우리는 사용할 수 있는 브릭의 색으로부터 컬러 팔레트를 생성하고, 입력된 픽셀 아트 영상의 색을 이 컬러 팔레트의 색들 중에서 가장 유사한 색으로 교체한다. 이 과정에서 컬러 팔레트의 색과 픽셀 아트 영상의 색의 비교는 $Lu \cdot v^*$ 컬러 공간에서 수행되어 색의 왜곡을 최소화한다.

5.2 ROI 설정

우리는 레고화 과정에서의 탐색 공간을 적절하게 제약하기 위해서 우리가 브릭을 놓으려고 하는 픽셀 주변에 관심 영역인 ROI를 설정한다. 이 ROI는 3×8 크기의 픽셀들의 집합으로 우리가 관심을 보이는 픽셀 주변에 배치된다. 우리가 (x, y) 에 픽셀을 놓으려고 하면 ROI는 $(x - 2, y - 1) \sim (x + 5, y + 1)$ 인 영역으로 설정된다 [Fig. 8].



[Fig. 8] The comparison of our ROI-based approach and greedy approach

5.3 레고화 과정

레고화 과정은 입력된 픽셀 아트 영상의 모든 픽셀에 브릭이 배치될 때까지 수행된다. 이를 위해서 우리는 Legorization 함수와 ROI_Legorization 함수를 정의한다. Legorization 함수는 각 픽셀에서 ROI_Legorization 함수를 호출하는데, ROI_Legorization 함수는 우리가 브릭을 배치하려는 픽셀에서의 가장 최적의 브릭을 반환한다.

ROI_Legorization 함수는 DQN을 이용해서 4개의 브릭들 중에서 2개의 후보 브릭을 선택하는 재귀 함수이다. 이 함수는 최적의 브릭을 선택하기 위해서 브릭을 배치하는 과정을 시뮬레이션한다. 이 두 함수는 다음과 같다.

Algorithm 1: ROI Legorization

```

if Legorization is failed
    return -1, -1
if Legorization is completed
    return  $w_e r_e + w_s r_{s'} - 1$ 
    
```

```

DQN -> action1, action2
todoList <- [action1, action2]
    
```

```

for brick in todoList
    add brick to brick_set
    r, last_act := ROI_Legorization()
    remove brick from brick_set
    if opt_r < r
        opt_r := r
        opt_act := brick
    
```

```

return r, opt_act
    
```

```

ALGORITHM 2: Legorization
while true
  get ROI
  r,opt_act=ROI_Legorization()
  brick<- opt_act
  add brick to brick_set

if Legorization is completed
  return brick set
    
```



[Fig. 9] 16x16 pixel art images for training

6. 구현 및 결과

이 연구의 하드웨어 구현 환경은 Intel Pentium i7 CPU, 16 GByte의 메인 메모리 및 nVidia TitanX GPU를 갖춘 워크스테이션이며, 소프트웨어 환경은 Linux Ubuntu에서 설치된 Python 2.7 과 Python과 호환되는 심층 학습 라이브러리인 Pytorch 0.3.1이다.

훈련을 위해서 [Fig. 9]에서 제시한 5 장의 16 x 16 크기의 픽셀 아트 영상을 선택하였고, 각 영상에 대해서 50,000회의 훈련을 수행하였다. 훈련 단계를 통해서 훈련된 DQN을 [Fig. 10]에서 제시하는 픽셀 아트 영상에 제시하였고, 그 결과를 [Fig. 11]에서 보여준다. 본 연구에서는 특히, [Fig. 10]에서 제시된 150 x 75 해상도의 영상에 대해서도 성공적으로 레고 퍼즐을 생성하는 결과를 보여준다.

[Table. 1]에서 본 연구의 결과를 동적 프로그래밍 및 Yun 등의 연구 결과에 픽셀과 브릭의 색 일치를 적용하지 않은 결과 및 색 일치를 적용한 결과와 비교하였다. 이를 통해서 본 연구의 결과는



(a) 45 x 26 (b) 45 x 32 (c) 100 x 100 (d) 150 x 75

[Fig. 10] Input pixel art images

기존의 방법들보다 더 빠르고 높은 보상을 받을 수 있음을 증명하였다.

7. 결론 및 향후 연구 방향

본 연구에서는 강화 학습에 기반한 2D 레고 퍼즐 생성 기술을 제안하였다. 이를 위해서 dueling DQN에 기반한 학습 모델을 설계하고, 단순한 픽셀 아트 영상을 이용해서 훈련하였다. 성공적인 레고 퍼즐 생성을 위해서 레고 생성에서 요구되는 휴리스틱을 안정성과 효율성으로 설정하였으며, 이를 최적화하는 방향으로 2D 레고 퍼즐을 생성하였다. 본 연구에서 제안하는 방법은 기존의 연구에서는 가능하지 않았던 높은 해상도의 픽셀 아트 영상을 레고 퍼즐로 변환하는 기술을 제공한다.

본 연구를 통해서 개발된 2D 레고 퍼즐 생성 기술을 더욱 발전시켜서, 단순한 평면상에 구현된 퍼즐에 간단한 높이 맵을 적용하는 입체 2D 레고 퍼즐 생성 기술을 연구하고자 한다. 또한, 이 기술을 더욱 발전 시켜서 8K 이상의 초고해상도 영상에 대한 레고 퍼즐 생성 기술을 개발하고자 한다.

[Table. 1] The comparison of the results

		Ours	Dynamic program	Yun et al.'s [8] w/o color match	Yun et al.'s [8] with color match
(a)	Time (sec)	11.81	2981.53	16.24	17.85
	No. of bricks	1042	1019	985	991
	Reward (r)	0.4357	0.4402	0.4231	0.4108
(b)	Time (sec)	11.92	3018.82	17.41	19.03
	No. of bricks	1210	1148	1103	1092
	Reward (r)	0.4813	0.4908	0.4719	0.4649
(c)	Time (sec)	82.69	9633.30	130.02	133.04
	No. of bricks	3709	3659	3865	3972
	Reward (r)	0.4439	0.4339	0.4212	0.4032
(d)	Time (sec)	129.66	9829.9	154.14	159.28
	No. of bricks	4492	4465	4612	4661
	Reward (r)	0.3804	0.3756	0.3678	0.3605



[Fig. 11] Result Lego puzzles assembled by tiny bricks

REFERENCES

- [1] R. Gower, A. Heydtmann, and H. Petersen, "Lego: Automated model construction," in Proc. of 32nd European Study Group with Industry 1998, 1998, pp. 81 - 94.
- [2] P. Petrovic, "Solving lego brick layout problem using evolutionary algorithms," in Proc. of Norwegian Conference on Computer Science, 2001, pp. 87 - 97.
- [3] L. van Zijl and E. Smal, "Cellular automata with cell clustering," in Proc. of Automata 2008, 2008, pp. 425 - 440.
- [4] R. Testuz, Y. Schwartzburg, and M. Pauly, "Automatic generation of constructable brick sculptures," in Proc. of Eurographics (Short Papers) 2013, 2013, pp. 81 - 84.
- [5] S. Ono, A. Andre, and Y. Chang, "Automatic generation of lego from the polygonal data," in Proc. of International workshop on advanced image technology 2013, 2013, pp. 262 - 267.
- [6] S. Lee, J. Kim, J. W. Kim, and B. R. Moon, "Finding an optimal lego brick layout of voxelized 3d object using a genetic algorithm," in Proc. of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 1215 - 1222.
- [7] M. Zhang, Y. Igarashi, Y. Kanamori, and J. Mitani, "Designing miniblock artwork from colored mesh," in Proc. of Smart Graphics 2015, 2015, p. 2.
- [8] G. Yun, C. Park, H. Yang, and K. Min, "Legerization with multiheight bricks from silhouette-fitted voxelization," in Proc. of Computer Graphics International 1986 (Short paper), 2017, p. 40.
- [9] S. J. Luo, Y. Yue, C. K. Huang, Y. H. Chung, S. Imai, T. Nishita, and B. Y. Chen, "Legolization: Optimizing lego designs," ACM Transactions on Graphics, vol. 34, no. 6, p. 222, 2015.
- [10] T. Kosaki, H. Tedenuma, and T. Maekawa, "Automatic generation of lego building instructions from multiple photographic images of real objects," Computer-Aided Design, vol. 70, no. C, pp. 13 - 22, 2016.
- [11] M. Zhang, J. Mitani, Y. Kanamori, and Y. Fukui, "Blocklizer: Interactive design of stable mini block artwork," in Proc. of ACM SIGGRAPH 2014 Posters, 2014, p. 18.
- [12] M. Kuo, Y. Lin, H. Chu, R. Lee, and Y. Yang, "Pixel2brick: Constructing brick sculptures from pixel art." Computer Graphics Forum, vol. 34, no. 7, pp. 339 - 348, 2015.
- [13] Reinforcement Learning, https://en.wikipedia.org/wiki/Reinforcement_learning.
- [14] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou,

- H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, Vol. 518, No. 7540, pp. 529 - 533, 2015.
- [16] H. Hasselt, A. Guez and D. Silver, "Deep reinforcement learning with double Q-learning," Proc. of AAAI conference on Artificial Intelligence 2016, pp. 2094-2100, 2016.
- [17] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," Proc. of International Conference on Learning Representation (ICLR) 2016.
- [18] Bellemare, M., Dabney, W., and Munos, R., "A distributional perspective on reinforcement learning," Proc. of International Conference on Machine Learning 2017, pp. 449-458, 2017.



박철성
(Park, Cheolseong)

약 력 : 2014년 2월 상명대학교 학사
2017년 8월 상명대학교 석사
2017년-현재 삼성전자

관심분야 : 컴퓨터 그래픽스, 인공 지능



양희경
(Yang, Heekyung)

약 력 : 2010년 2월 상명대학교 학사
2012년 8월 상명대학교 석사
2019년 2월 상명대학교 박사
2019년 3월 - 2020년 2월 상명대학교
박사후 연구원
2020년 3월-현재 상명대학교 SW융합학부 교수

관심분야 : 컴퓨터 그래픽스, 컴퓨터 비전, 심층 학습



민경하
(Min, Kyungha)

약 력 : 2000년 2월 포항공과대학교 박사
2002년 9월 - 2006년 2월 Rutgers 대학교
박사후 연구원
2006년 9월-현재 상명대학교 컴퓨터과학과교수

관심분야 : 컴퓨터 그래픽스, 컴퓨터 비전, 심층 학습