Siamese-Discriminant Deep Reinforcement Learning for Solving Jigsaw Puzzles with Large Eroded Gaps

Xingke Song[1], Jiahuan Jin[1], Chenglin Yao[1], Shihe Wang[1], Jianfeng Ren[12*], Ruibin Bai[12]

[1]School of Computer Science, University of Nottingham Ningbo China, China
[2]Nottingham Ningbo China Beacons of Excellence Research and Innovation Institute, University of Nottingham Ningbo China, China
{Xingke.Song, Jiahuan.Jin, Chenglin.Yao, Shihe.Wang, Jianfeng.Ren, Ruibin.Bai}@nottingham.edu.cn

Abstract

Jigsaw puzzle solving has recently become an emerging research area. The developed techniques have been widely used in applications beyond puzzle solving. This paper focuses on solving Jigsaw Puzzles with Large Eroded Gaps (JPwLEG). We formulate the puzzle reassembly as a combinatorial optimization problem and propose a Siamese-Discriminant Deep Reinforcement Learning (SD^{2}RL) to solve it. A Deep Q-network (DQN) is designed to visually understand the puzzles, which consists of two sets of Siamese Discriminant Networks, one set to perceive the pairwise relations between vertical neighbors and another set for horizontal neighbors. The proposed DQN considers not only the evidence from the incumbent fragment but also the support from its four neighbors. The DQN is trained using replay experience with carefully designed rewards to guide the search for a sequence of fragment swaps to reach the correct puzzle solution. Two JPwLEG datasets are constructed to evaluate the proposed method, and the experimental results show that the proposed SD^{2}RL significantly outperforms state-of-the-art methods.

Introduction

Jigsaw is a puzzle game of reassembling interlocking and inlaid fragments of irregular shapes. Automatic puzzle reassembly has been widely studied (Paumard, Picard, and Tabia 2020; Bridger, Danon, and Tal 2020). The developed techniques have been used beyond puzzle reassembly, e.g., self-supervised learning of visual representations (Noroozi and Favaro 2016; Ma et al. 2021; Yang et al. 2022).

Traditionally, puzzle solving often relies on the shape (Zhang and Li 2014; Gur and Ben-Shahar 2017; Zhang et al. 2015), contour (Huang et al. 2013) or color (Son, Hays, and Cooper 2014) of fragments, while only very few focus on the image content (Paikin and Tal 2015). Recently, Jigsaw puzzles are solved by utilizing deep learning techniques for image understanding and puzzle reassembly (Paumard, Picard, and Tabia 2020; Doersch, Gupta, and Efros 2015; Bridger, Danon, and Tal 2020; Li et al. 2022). In Deepzzle, Paumard, Picard, and Tabia (2020) designed a pair of VGG networks to predict the relative positions between fragments,

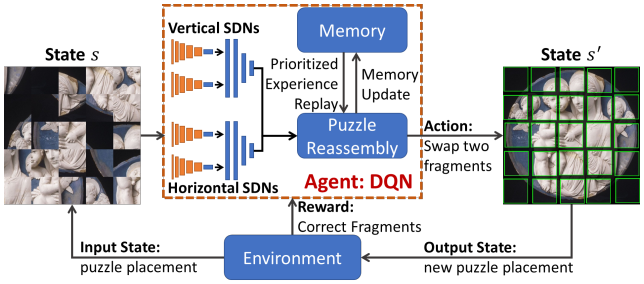

Figure 1: Proposed Siamese-Discriminant Deep Reinforcement Learning for puzzle solving. The agent aims to learn a Deep Q-learning Network combining visual understanding and puzzle reassembly based on the observed sequences of states, rewards, actions, and the past reassembly experience.

formulated the puzzle reassembly as the shortest-path problem and solved it using Dijkstra's algorithm with branch-cut. Bridger, Danon, and Tal (2020) designed a GAN-based method to find neighbors of each fragment by filling gaps in between and a greedy algorithm to reassemble the puzzles.

Solving puzzles often consists of two steps: 1) Image understanding through either handcrafted features (Gur and Ben-Shahar 2017; Li et al. 2022), or deep convolutional neural networks (Paumard, Picard, and Tabia 2020; Noroozi and Favaro 2016; Bridger, Danon, and Tal 2020; Li et al. 2022). 2) Puzzle reassembly by utilizing various strategies such as shortest-path optimization (Paumard, Picard, and Tabia 2020), genetic algorithm (Mirjalili 2019), greedy algorithm (Bridger, Danon, and Tal 2020) and nonconvex quadratic programming (Yan et al. 2021). A well-designed reassembly strategy could utilize all the perceived visual information to derive a globally optimal solution to puzzle solving. Previous methods often over-emphasize visual understanding, while neglecting the importance of the reassembly strategy. For example, Deepzzle utilizes a relatively simple Dijkstra's algorithm with branch-cut as the assembly strategy (Paumard, Picard, and Tabia 2020). For a small-scale puzzle, people could resort to the brute-force method (Paumard, Picard, and Tabia 2018a) or greedy method (Bridger, Danon, and Tal 2020) for reassembly. But for a large-scale puzzle, the number of puzzle permutations grows exponentially and a reassembly strategy is crucial to

*Corresponding Authors.

ensure the final success of puzzle solving.

To address the challenges of solving Jigsaw Puzzles with Large Eroded Gaps (JPwLEG), we propose a Siamese-Discriminant Deep Reinforcement Learning (SD^2RL) paradigm. As shown in Fig. 1, given an initial placement of puzzles, the proposed Deep Q-Network (DQN) consists of two sets of Siamese Discriminant Networks to visually perceive the pairwise relations between horizontal neighbors and vertical neighbors, respectively. The DQN is trained to estimate the Q value for an action of swapping a pair of fragments. The reward is calculated based on the number of fragments being correctly placed after taking the swapping action, and a greedy policy is used to find the action with the largest Q value. The process is repeated until certain stopping conditions are met. The proposed SD^2RL offers a unified reassembly strategy for solving puzzles. It is consistent with human's strategy for solving puzzles, e.g., examining the visual information of fragments to combine neighboring pieces first, and gradually merging them into large pieces. Humans also make use of a trial-and-error process for puzzle solving, which is well aligned with the exploration-exploitation principle of reinforcement learning.

The proposed SD^2RL is significantly different from previous methods in the following aspects: 1) It tackles the JPwLEG problem using the reinforcement learning paradigm. To our best knowledge, this is the first attempt to introduce reinforcement learning into puzzle solving. 2) Previous methods such as Deepzzle (Paumard, Picard, and Tabia 2020) only make use of the relation between a fragment and the center piece, which is weak and less informative for fragments far away from the center. In contrast, the proposed method evaluates the pairwise relations to assess the likelihood of two pieces being neighbors. This framework could be extended to large-scale puzzles while Deepzzle could not. 3) The proposed SD^2RL makes use of not only the perceived visual information but also the past puzzle reassembly experiences guided by the rewards estimated from the ground-truth puzzle layout. The integration of visual inspection and deep reinforcement learning reassembly in SD^2RL could help solve the puzzles more effectively.

To validate the proposed SD^2RL, two benchmark JPwLEG datasets are constructed, JPwLEG-3 with 3×3 pieces and JPwLEG-5 with 5×5 pieces. The proposed method is compared with state-of-the-art methods on these two datasets. It consistently and significantly outperforms all the compared methods on these two datasets.

Our contributions can be summarized as follows: 1) We formulate the problem of puzzle reassembly as a combinatorial optimization problem, and propose a Siamese Discriminant Deep Reinforcement Learning method to find the best sequence of swapping actions for puzzle solving. 2) The proposed DQN integrates the visual understanding of puzzles from the Siamese Discriminant Networks with the reassembly strategy derived from the experience replay guided by the carefully designed rewards, to help each other to solve the puzzles mutually. 3) Two JPwLEG datasets are constructed to evaluate the proposed method, which will be made publicly available upon the acceptance of this paper.

Related Work

Jigsaw Puzzle Solving

Puzzle solving has recently attracted a lot of research attention. There are two main types of puzzles: regular-piece puzzles and irregular-piece puzzles. The geometric information of each fragment is often used to find the matches between fragments (Zhang and Li 2014; Zhang et al. 2015) for regular-piece puzzles. Many recent studies focus on solving square-piece puzzles (Paumard, Picard, and Tabia 2020; Bridger, Danon, and Tal 2020; Yan et al. 2021; Li et al. 2022). Besides the geometric information, the semantic relations between a pair of fragments have also been utilized (Paumard, Picard, and Tabia 2020).

Solving Jigsaw puzzles often consists of two steps: image understanding and puzzle reassembly. Existing image understanding methods for puzzle solving can be broadly categorized into methods based on handcrafted features and methods based on deep convolutional neural networks. Handcrafted features based on shapes (Gur and Ben-Shahar 2017), contours (Huang et al. 2013), colors (Paikin and Tal 2015), texture patterns (Son, Hays, and Cooper 2014) and boundary dissimilarities (Paikin and Tal 2015) are often extracted to visually understand the puzzle. For example, Son, Hays, and Cooper (2014) extracted the pairwise matching of colors and patterns of fragments as features. Paikin and Tal (2015) extracted the boundary dissimilarity measures and reassembled the puzzles with a greedy searching strategy.

Deep learning techniques have been widely used in puzzle solving and many other applications (He et al. 2023; Zhang et al. 2022; Yao et al. 2021; Liu et al. 2022). Doersch, Gupta, and Efros (2015) designed a pair of AlexNet-style architectures to learn the image representation for predicting the relative position of fragments. In Deepzzle, Paumard, Picard, and Tabia (2018b) improved the work by adding a combination layer to emphasize the co-occurrence features between the pair of fragments. Li et al. (2022) developed a GAN-based architecture for puzzle reassembly by utilizing both boundary and image semantic loss. To solve large-scale Jigsaw puzzles, Bridger, Danon, and Tal (2020) developed an image inpainting method with a U-Net architecture and a GAN model for visual perception. However, the eroded gaps between fragments are made relatively small in their experiments, e.g., only 2 or 4 pixels. This method may not work well on the JPwLEG problem studied in this work.

Puzzle Reassembly Strategies

In Deepzzle, Paumard, Picard, and Tabia (2020) implemented Dijkstra's algorithm with branch-cut to solve the puzzle. For large-scale Jigsaw puzzles with known relations between fragments, genetic algorithm (Mirjalili 2019) and simulated annealing (Delahaye, Chaimatanan, and Mongeau 2019) could achieve a good reassembly accuracy. Besides, the greedy method (Bridger, Danon, and Tal 2020), metaheuristic method (Sholomon, David, and Netanyahu 2013), and quadratic programming (Yan et al. 2021) have been utilized as the reassembly strategy. For the JPwLEG problem, the relation between two fragments with large gaps is much weaker, and hence these methods may not work well. In this

paper, we propose a deep reinforcement learning framework to improve the reassemble accuracy.

Reinforcement Learning (RL) has witnessed a great number of successes in combinatorial optimization (CO) (Bengio, Lodi, and Prouvost 2021; Woo, Lee, and Cho 2022; Chen and Tian 2019; Zong et al. 2022; Bai et al. 2021), robotics (Tomar, Sathuluri, and Ravindran 2019), natural language processing (Li, Kiseleva, and De Rijke 2019) and computer vision (Kim et al. 2021). Combination of RL and graph neural network also flourished in recent years (Vesselinova et al. 2020). In this paper, puzzle solving is first formulated as a CO problem, and an SD^{2}RL framework is designed to solve the problem.

Proposed Siamese-Discriminant Deep Reinforcement Learning for Puzzle Solving

Revisit of Deepzzle Method

To solve the JPwLEG problem, Paumard, Picard, and Tabia (2020) developed a neural network to learn the probability of a fragment residing at its current location w.r.t. the fixed center piece. After deriving the probability matrix $\boldsymbol{P} \in \mathcal{R}^{8\times 8}$, where each entry $\boldsymbol{P}_{ij}$ is the probability of the i-th fragment in the j-th location, the Dijkstra's algorithm with branch-cut is utilized to find the puzzle placement. Deepzzle demonstrates good performance in solving puzzles of 3×3 pieces, but still faces two challenges. 1) The support evidence of a fragment residing at its current location w.r.t. the center piece may be weak when the fragment is far away from the center, especially when there are large eroded gaps between fragments. 2) The reassembly strategy in Deepzzle is relatively simple. It could not handle large-scale puzzles. As shown later, this will lead to a complicated combinatorial optimization problem, where much more efforts on reassembly strategy are needed.

Overview of Proposed SD^{2}RL

To tackle the challenges of puzzle solving, Siamese-Discriminant Deep Reinforcement Learning (SD^{2}RL) is proposed. For a sequence of fragment swapping actions, the target is to find the best swap sequence to solve the puzzle. As shown in Fig. 2, a Deep Q-Network (DQN) is designed with two sets of Siamese Discriminant networks, one set to estimate the likelihood probability that two fragments are horizontal neighbors, and another set for vertical neighbors. The visually perceived information is aggregated in the DQN through a set of fully connected layers to estimate the Q value of any swapping action over a given placement of fragments. The reward is calculated based on the number of correctly placed fragments after taking the action. A greedy policy is applied, which maximizes the Q value given a sequence of swapping actions. Such a formulation could make good use of the reward calculated based on the number of successfully assembled fragments. Such a reward is never exploited in other works, but could provide more accurate guidance than the roughly estimated visual evidence from neighboring fragments, as shown later in experiments.

Formulation of Combinatorial Optimization

To better understand the motivations of our work, we first formulate the puzzle solving as a Combinatorial Optimization problem. More specifically, denote Π as the set of all possible permutations of fragment indices $\{1, 2, ..., n\}$, where the indices encode the positions of fragments and n is the number of fragments. Given an initial permutation $\pi_0 \in \Pi$, the target is to find a mapping function $\mathcal{M}$ so that

$$\pi_G = \mathcal{M}(\pi_0), \tag{1}$$

where π_G is the ground-truth layout of puzzles. Assuming that the center piece is fixed, there are in total $(n-1)!$ different possible permutations. The optimal mapping $\mathcal{M}$ can be derived by maximizing the evidence E on all fragments,

$$\pi^* = \arg\max_{\pi\in\Pi} E, \tag{2}$$

where $E = \sum_{i=1}^{n} E_i$, and E_i is the evidence of a given fragment $\boldsymbol{f}_i$ being at its current location. Intuitively, E_i should consider the supporting evidence from its four neighbors as well, e.g. $\boldsymbol{f}_i^L, \boldsymbol{f}_i^R, \boldsymbol{f}_i^U$, and $\boldsymbol{f}_i^D$ with relative positional relations as *left, right, up*, and *down*, respectively. Denote the estimated evidence of a fragment based on its image content alone as E_i^C, and the evidence of its neighbors being at their current locations as E_i^L, E_i^R, E_i^U, and E_i^D, respectively. The pairwise probabilities are estimated as $P^H(\boldsymbol{f}_i^L, \boldsymbol{f}_i), P^H(\boldsymbol{f}_i, \boldsymbol{f}_i^R), P^V(\boldsymbol{f}_i^U, \boldsymbol{f}_i)$, and $P^V(\boldsymbol{f}_i, \boldsymbol{f}_i^D)$, respectively, where $P^H(\boldsymbol{f}_a, \boldsymbol{f}_b)$ and $P^V(\boldsymbol{f}_a, \boldsymbol{f}_b)$ denote the pairwise probability of $\boldsymbol{f}_a$ and $\boldsymbol{f}_b$ being horizontal and vertical neighbors, respectively. Then, the evidence after considering the support of neighbors is updated as:

$$\begin{aligned} E_i =& E_i^C + P^H(\boldsymbol{f}_i^L, \boldsymbol{f}_i)E_i^L + P^H(\boldsymbol{f}_i, \boldsymbol{f}_i^R)E_i^R \\ &+ P^V(\boldsymbol{f}_i^U, \boldsymbol{f}_i)E_i^U + P^V(\boldsymbol{f}_i, \boldsymbol{f}_i^D)E_i^D. \end{aligned} \tag{3}$$

This formulation is significantly different from Deepzzle (Paumard, Picard, and Tabia 2020), where only E_i^C is considered. It is almost infeasible to directly find the mapping function $\mathcal{M}$ where $\pi_G = \mathcal{M}(\pi_0)$. In this paper, we aim to find a sequence of mapping $\{\mathcal{M}_1, \mathcal{M}_2, \dots\}$,

$$\pi_{t+1} = \mathcal{M}_t(\pi_t), \tag{4}$$

so that the final permutation is the same as π_G.

Formulation of Reinforcement Learning

Given the objective function defined in Eqn. (2), one may resort to some algorithms to directly maximize it to find the solution. However, our study shows that the estimations of the probability of two fragments being neighbors may not be accurate due to the large eroded gaps. For example, the prediction accuracy is 82.6% when a Siamese Discriminant network is trained to determine whether two fragments are horizontal neighbors, and 84.8% for vertical neighbors for puzzles of 3×3 pieces. More importantly, due to the noisy estimation, maximizing the objective function defined in Eqn. (2) may not produce the correct reassembly order. In fact, for 43.4% of puzzles, the objective function E defined in Eqn. (2) calculated for the correct reassembly is not larger than

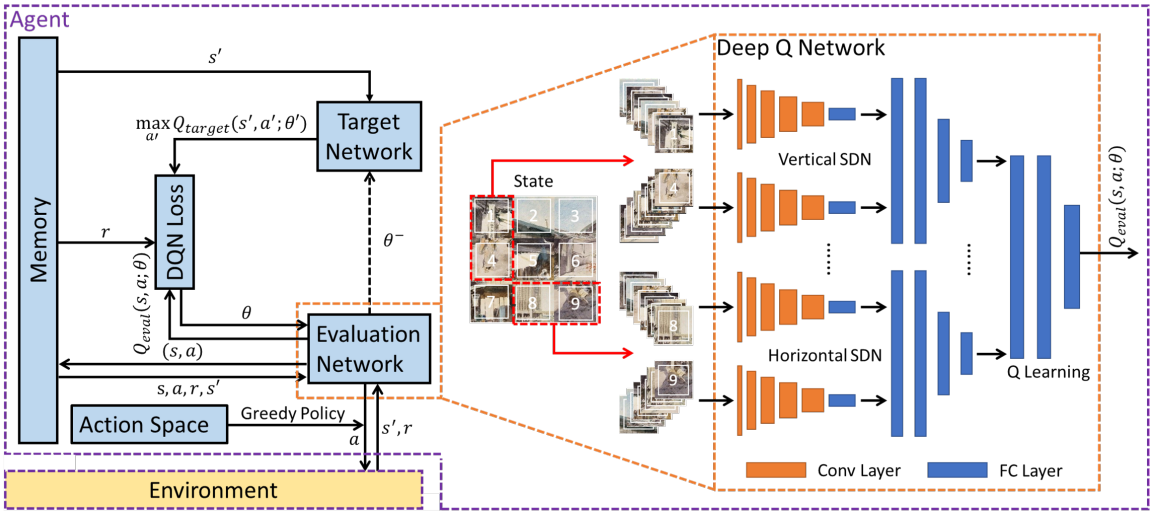

Figure 2: Block diagram of the proposed Siamese-Discriminant Deep Reinforcement Learning (SD^{2}RL) for puzzle solving. Given an initial state, the target is the find a sequence of fragment swaps, maximizing the Q value of an action till solving the puzzle. For the puzzles of $N \times N$ pieces, $N(N-1)$ horizontal Siamese Discriminant Networks and $N(N-1)$ vertical Siamese Discriminant Networks are incorporated with a set of fully connected layers to form the Deep Q-Network. The architecture of the target network is a sibling to the evaluation network.

that of the reassembly after a greedy search. Apparently, the objective function E could only provide an approximation of whether the placement is correct or not.

To tackle this problem, instead of maximizing E, the proposed SD^{2}RL maximizes the reward defined using the ground-truth placement, e.g. if the fragment is correctly placed at its location, the reward is 1 and 0 otherwise. Definitely, in this case, for the correct reassembly order, the reward is maximum, but the objective function E may not be maximum. This reward will guide the agent to choose a sequence of swap actions $\{a_1, a_2, \dots\}$ to finally achieve the maximal reward, i.e., to find the correct reassembly order. The problem is reformulated as a Markov Decision Process (MDP) with the following implementation details.

1. S: State space. A state $s_t \in S$ is an ordered sequence of fragments in Jigsaw puzzle problems. A vector $\boldsymbol{v}$ of length n is used to encode the state, where v_i^t denotes the fragment of index i at time step t.
2. A: Action space. $a_t \in A$ is an action from the determined action space. In our problem, $A = \{(i,j)|i,j \in F \backslash f_c\}, i \neq j\}$, where F is the set of fragments' indices and f_c is the immovable center fragment.
3. R: Reward function. The reward function $R(s_t, a_t)$ is defined as $R = \alpha L_t + (1-\alpha)H_t + b + C_t$, where L_t is the number of fragments being placed at their correct locations, H_t is the pairs of fragments whose relative position is correct, C_t is a large positive reward if all fragments are correctly placed and zero otherwise, b is a constant penalty for not correctly placing all fragments at this step, and $\alpha \in [0,1]$) balances the importance of L_t and H_t.
4. P is defined as the probabilities of state transitions which are denoted as $p(s'|s,a) = Pr\{S_t = s'|S_{t-1} = s, A_{t-1} = a\}$. In this problem, the state transition is a deterministic process.
5. $\gamma \in [0,1]$ is the discount rate. The agent aims at maximizing the discounted accumulated reward from time step t onward, which is denoted as $G_t = \sum_{k=0}^{T} \gamma^k R_{t+k+1}$, where T is the amount of time steps.

Siamese Discriminant Network of Deep Q-network

The proposed Deep Q-network consists of two sets of Siamese Discriminant networks (SDNs) to visually assess whether two fragments are neighbors, one set for vertical neighbors and the other for horizontal neighbors. To model the pairwise relations, as shown in Fig. 3, each Siamese Discriminant Network is built with two VGG-16 branches, one to extract the visual information from each fragment. SDN is trained to determine the likelihood of two fragments being neighbors. Besides VGG-16, other networks have also been evaluated, but the VGG-16 produces excellent performance while keeping the network lightweight. A plausible explanation is that by cutting puzzles into pieces with large eroded gaps, each fragment does not contain too much semantic information, so that simple architecture like VGG-16 is capable of capturing the discriminative information while maintaining the generalization ability.

Reinforcement Learning Strategy

The proposed SD^{2}RL consists of an evaluation network and a target network, as shown in Fig. 2. The input to the network is the puzzle image at the current state, and the output is the Q value for a given action. Denote the Q value function as $Q(s_t, a_t; \theta)$, where θ is the parameters of the network. The evaluation DQN consists of two sets of SDNs capturing the

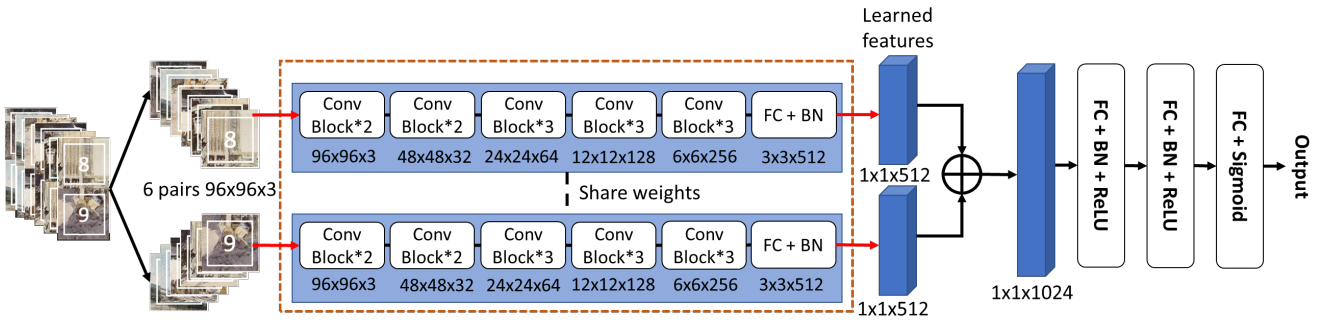

Figure 3: Siamese Discriminant Network to perceive the pairwise relations between horizontal/vertical neighbors.

vertical and horizontal relations of fragments. Then, fully connected layers are used to map the pairwise probabilities to Q values with the guidance of the rewards. The correct placement of two fragments in relative or absolute positions is rewarded positively. α is close to 1 because the absolute position is more important. Constant penalty b encourages the agent to solve a puzzle in fewer steps and avoids the agent being myopic for intermediate rewards. C_t represents the reward for correctly solving the puzzle.

A greedy policy $a^* = \arg\max_{a' \in A} Q(s_t, a'; \theta)$ is used to evaluate the current puzzle placement based on the estimated Q values, similarly as in (Mnih et al. 2015). During training, actions are selected through the ϵ-greedy policy. Random actions are selected with probability ϵ and greedy actions are selected with probability $1 - \epsilon$. ϵ is set to 1 at the beginning and gradually decreases until $\epsilon = \epsilon_{min}$. Such a mechanism ensures that adequate state space is explored.

In one episode, swapping actions will stop if all fragments are correctly placed or the maximum step T_{max} is reached. T_{max} is set to 20000 for training and 50 for evaluation, to guarantee that the experience of successfully solving a puzzle can be sampled during training and a puzzle needs to be solved in a short time during evaluation to avoid the lucky success for a long-term trial. Every W iterations, $\theta^- = \theta$, the model parameters of the evaluation network are copied to the target network. The priority replay mechanism (Schaul et al. 2016) is used to accelerate the convergence. The proposed Deep Q-learning is summarized in Algorithm 1.

Algorithm 1: Deep Q-learning for puzzle solving.

Input : Decay factor γ, mini-batch size K, number of episodes M, target network update step W, an evaluation network with parameters θ
Output: Target network with parameters θ^-
Data: A memory buffer B with capacity N, a data set D of training Jigsaw puzzles
for *episode=1 : M* **do**
  Randomly select a puzzle $\boldsymbol{p}$ from D and observe the initial state s_t **repeat**
    With probability ϵ select an action a_t, otherwise $a_t = \arg\max_{a \in A} Q(s_t, a; \theta)$
    Execute a_t on $\boldsymbol{p}$ and observe s_{t+1} and r_t
    Store the transition $\{s_t, a_t, r_t, s_{t+1}\}$ in memory buffer B
    if *(memory length $l \geq K$)* **then**
      Sample a batch of transitions by the prioritized experience $< s_i, a_i, r_i, s_{i+1} >$ from B
      $L(\theta) = (r_i + \gamma \max_a Q(s_{i+1}, a; \theta^-) - Q(s_i, a_i; \theta))^2$
      $\theta = Adam(\nabla L, \theta)$
    end
    Set $\theta^- = \theta$ every W steps
  until *time step $t = T_{max}$ or puzzle is correctly reassembled*;
  Update ϵ
end

Experimental Results

Construction of JPwLEG Datasets

To evaluate the proposed SD^{2}RL on the JPwLEG problem, two datasets are constructed based on the METropolitan (MET) Museum of Art open-source image and data resources (Paumard, Picard, and Tabia 2020). 12000 images of painting, engraving and artifacts are chosen. Each image is resized and square-cropped to 398×398 pixels. For the JPwLEG-3 dataset of 3×3 pieces, the image is divided into 9 parts separated by 48-pixel gaps, mimicking an erosion of the fragments. Each fragment has 96×96 pixels, and is randomly moved by ± 7 pixels horizontally or vertically.

To evaluate the robustness of the proposed method in solving puzzles of a larger scale, we construct a JPwLEG-5 dataset of 5×5 pieces from the MET dataset. Each image is resized and square-cropped to 534×534 pixels, and divided into 25 pieces separated by 12-pixel gaps. Each fragment has 96×96 pixels, and is randomly moved by ± 3 pixels horizontally or vertically. Following the same evaluation protocol as in Deepzzle (Paumard, Picard, and Tabia 2020), we randomly choose 9000 images for training, 1000 for validation, and 2000 for testing on both datasets.

Experimental Settings

The proposed SD^{2}RL is compared with Deepzzle (Paumard, Picard, and Tabia 2020), greedy search (Paikin and Tal 2015), Tabu search (Adamczewski, Suh, and Mu Lee 2015),

and Genetic Algorithm (Mirjalili 2019). Deepzzle serves as the baseline method and the other three are popular algorithms to solve CO problems. We implement these algorithms and apply them to the evidence used in Deepzzle, i.e., the pairwise probabilities between a fragment and the given center fragment, and the evidence defined in Eqn. (2) considering the pairwise relations between neighbors. The former is denoted by adding a suffix '-C' and the latter is denoted by adding a suffix '-P'. The proposed method is also applied to the visual perception between a fragment and the given center fragment as in (Paumard, Picard, and Tabia 2020), to evaluate the performance gain brought by the Siamese Discriminant Network, which is denoted as SD^{2}RL-C.

Deepzzle (Paumard, Picard, and Tabia 2020) is a state-of-the-art method for solving JPwLEG problems. The branch-cut threshold is set to 0.05 for a balance of iteration times and accuracy. The number of search steps is limited to 10^6.

Greedy Search (Paikin and Tal 2015) is often used to solve CO problem. It is implemented as the reassembly strategy and compared with the proposed SD^{2}RL.

Tabu Search (Adamczewski, Suh, and Mu Lee 2015) is a metaheuristic that enhances the local search by using a short-term memory called Tabu List. The tabu size is set to 10 and the number of iterations is limited to 100.

Genetic Algorithm (Mirjalili 2019) simulates the natural selection process of candidate solutions, including the multi-point crossover operator and mutation operator for evolution. The population size is 256, with the crossover rate of 80%, the mutation rate of 20%, and up to 50 iterations. To avoid the premature convergence, the fitness values of duplicated solutions are decayed by 10%.

Proposed SD^{2}RL utilizes a batch size of 32, a memory buffer of size 200,000, $W = 100$, $\lambda = 0.995$, $\epsilon_{max} = 1$ and $\epsilon_{min} = 0.05$, learning rate of 0.0003, $\alpha = 0.8$, $b = 1$ and $C_t = 1000$. The Adam optimizer is used.

The results are reported in terms of four evaluation metrics: **Perfect**, **Absolute**, **Horizontal** and **Vertical**, which show the percentage of puzzles that are correctly reassembled, in their correct absolute positions, in correct horizontal and vertical pairwise relations, respectively.

Comparison Results on JPwLEG-3 Dataset

As shown in Table 1, the experimental results on the JPwLEG-3 dataset are summarized into two groups, the methods utilizing the pairwise relations in (Paumard, Picard, and Tabia 2020), denoted by the suffix '-C', and the methods utilizing the pairwise relations between neighbors, denoted by the suffix '-P'. Based on the experimental results, we have the following observations: 1) By utilizing the pairwise relations between neighbors, all the methods in the second group achieve a significant performance gain over the corresponding methods in the first group. This clearly demonstrates the benefits of utilizing the pairwise relations between neighbors, which also justifies the design choice in the proposed DQN by including two sets of Siamese Discriminant Networks for visual support. 2) The performance of the compared methods such as Deepzzle (Paumard, Picard, and Tabia 2020), greedy search (Paikin and Tal 2015), Tabu search (Adamczewski, Suh, and Mu Lee 2015), and Genetic Algorithm (Mirjalili 2019) does not vary significantly within each group. Given a relatively small-scale search problem, different optimization strategies applied on the noisy estimations of visual evidence do not significantly affect the final reassembly performance. 3) In contrast, the proposed SD^{2}RL significantly improves the performance compared with other reassembly strategies. The underlying reason is that the proposed SD^{2}RL could utilize not only the visual perception information of the current puzzle for reassembly, but also the past reassembly experience learned from the experience replay for other puzzles guided by the carefully designed reward function. The reward function is derived using the ground-truth placement of the puzzle, which is much more accurate than the estimated pairwise relations. 4) The proposed SD^{2}RL significantly outperforms all the compared methods on all four evaluation metrics. Compared to the baseline method, Deepzzle-C (Paumard, Picard, and Tabia 2020), the proposed method achieves the performance gain of 14.8%, 7.6%, 9.2%, and 17.6% on Perfect, Absolute, Horizontal, and Vertical metrics respectively, in which the improvement on Vertical metric is most significant.

Method	Perfect	Absolute	Horizontal	Vertical
Deepzzle-C	44.9%	74.0%	67.2%	59.0%
Greedy-C	40.0%	71.5%	64.0%	64.7%
Tabu-C	44.8%	73.8%	66.9%	67.4%
GA-C	44.9%	73.9%	66.8%	67.4%
SD^{2}RL-C	47.8%	75.1%	68.6%	69.1%
Deepzzle-P	52.3%	73.8%	69.3%	70.0%
Greedy-P	55.2%	79.5%	74.0%	74.2%
Tabu-P	55.2%	79.0%	73.8%	73.7%
GA-P	55.5%	79.6%	74.3%	74.2%
SD^{2}RL	**59.7%**	**81.6%**	**76.4%**	**76.6%**

Table 1: Evaluation results on the JPwLEG-3 dataset. The first group of methods utilizes the pairwise relations between a fragment and the given center as in (Paumard, Picard, and Tabia 2020), denoted by the suffix '-C'. The second group utilizes the pairwise neighboring relations, denoted by the suffix '-P'. The large performance gains of the second group over the first demonstrate the effectiveness of utilizing the neighboring relations. The proposed SD^{2}RL significantly and consistently outperforms all the compared methods in the same group in terms of all evaluation metrics.

The JPwLEG datasets consist of puzzles from three main sources: painting, engraving, and artifact. The evaluation results for Perfect and Absolute measures on these different types of images are summarized in Table 2. It can be seen that the proposed SD^{2}RL outperforms all the compared methods on puzzles from different sources using both measures. Painting puzzles are relatively more challenging because they contain diverse image content. Fig. 4 shows the visual results of all the compared methods. Deepzzle (Paumard, Picard, and Tabia 2020), greedy search (Paikin and Tal 2015), Tabu search (Adamczewski, Suh, and Mu Lee 2015) and Genetic Algorithm (Mirjalili 2019) often make mistakes for pieces at corners, whereas the proposed method could find the correct reassembly order.

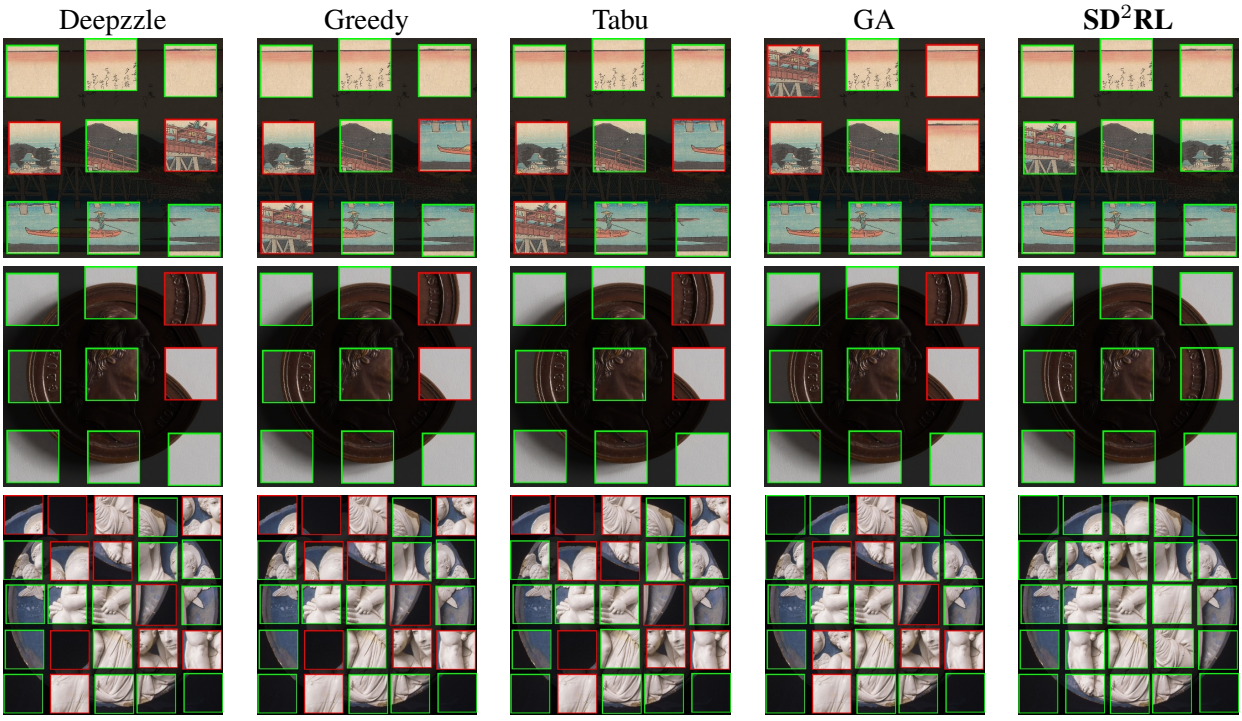

Figure 4: Visualization of the reassembling results on sample images from the JPwLEG-3 and JPwLEG-5 datasets.

Method	Perfect			Absolute		
	Pnt.	Eng.	Art.	Pnt.	Eng.	Art.
Deepzzle-P	30.6%	58.9%	67.4%	61.6%	75.7%	84.2%
Greedy-P	33.7%	61.9%	70.0%	68.1%	81.6%	88.9%
Tabu-P	33.1%	63.9%	68.6%	66.9%	82.1%	88.1%
GA-P	33.9%	63.1%	70.4%	67.7%	82.2%	89.3%
SD^{2}RL	**37.3%**	**67.5%**	**74.3%**	**70.0%**	**84.2%**	**90.5%**

Table 2: Evaluation results on different types of images of the JPwLEG-3 dataset. The proposed method outperforms all the compared methods for all three types of images.

Method	**Perfect**	**Absolute**	**Horizontal**	**Vertical**
Deepzzle	0.0%	21.9%	10.9%	10.7%
Greedy	0.1%	24.1%	12.6%	12.3%
Tabu	0.0%	24.6%	12.8%	12.8%
GA	0.0%	25.1%	12.4%	12.3%
SD^{2}RL	**5.1%**	**40.3%**	**26.5%**	**26.2%**

Table 3: Comparison results on the JPwLEG-5 dataset. The proposed method significantly outperforms other methods.

Comparison Results on JPwLEG-5 Dataset

The experimental results on the JPwLEG-5 dataset are shown in Table 3. The proposed method achieves the best performance among all the compared methods under four evaluation metrics. It is indeed difficult to solve large-scale JPwLEG problems, as evidenced in (Bridger, Danon, and Tal 2020), where only 1 puzzle is correctly assembled for 4-pixel gaps between fragments, while the gap in the JPwLEG-5 dataset is 12 pixels. Compared to Deepzzle, the performance gains are 5.1%, 18.4%, 15.6%, and 15.5% on Perfect, Absolute, Horizontal and Vertical measures, respectively. Our method utilizes the rewards estimated using the ground-truth puzzle placement to guide the search for the best swap action, which greatly helps the puzzle reassembly. As shown in the last row of Fig. 4, many pieces are wrongly assembled by the compared methods, which are mainly due to the weak semantic relations between small pieces, while the proposed method could exploit the rewards and the perceived visual information to reassemble the puzzle correctly.

Conclusion

In this paper, a Siamese-Discriminant Deep Reinforcement Learning is proposed to solve the Jigsaw puzzle with large eroded gaps. Two sets of Siamese Discriminant Networks are designed as part of the Deep Q-network to visually understand the puzzles. The developed Deep Q-network derives the Q values based on not only the perceived current image content of puzzles, but also the past experience with puzzle reassembly. A greedy policy maximizing the Q value is applied to determine the best swapping action given the current puzzle placement. The proposed SD^{2}RL makes use of both the perceived visual information from SDNs and the exploration-exploitation strategy of RL guided by the carefully designed rewards estimated using the puzzle placement. Two JPwLEG datasets are created to evaluate the proposed method. The proposed SD^{2}RL significantly outperforms all the compared methods on both datasets.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 72071116, and in part by the Ningbo Municipal Bureau Science and Technology under Grants 2019B10026 and 2022Z173.

References

Adamczewski, K.; Suh, Y.; and Mu Lee, K. 2015. Discrete tabu search for graph matching. In *IEEE International Conference on Computer Vision*, 109–117.

Bai, R.; Chen, X.; Chen, Z. L.; Cui, T.; Gong, S.; He, W.; Jiang, X.; Jin, H.; Jin, J.; Kendall, G.; et al. 2021. Analytics and machine learning in vehicle routing research. *International Journal of Production Research*, 1–27.

Bengio, Y.; Lodi, A.; and Prouvost, A. 2021. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421.

Bridger, D.; Danon, D.; and Tal, A. 2020. Solving Jigsaw puzzles with eroded boundaries. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3526–3535.

Chen, X.; and Tian, Y. 2019. Learning to perform local rewriting for combinatorial optimization. *Advances in Neural Information Processing Systems*, 32: 6278–6289.

Delahaye, D.; Chaimatanan, S.; and Mongeau, M. 2019. Simulated annealing: From basics to applications. In *Handbook of Metaheuristics*, 1–35. Springer.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *IEEE International Conference on Computer Vision*, 1422–1430.

Gur, S.; and Ben-Shahar, O. 2017. From square pieces to brick walls: The next challenge in solving Jigsaw puzzles. In *IEEE International Conference on Computer Vision*, 4029–4037.

He, W.; Zhang, J.; Ren, J.; Bai, R.; and Jiang, X. 2023. Hierarchical ConViT with attention-based relational reasoner for visual analogical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Huang, H.; Yin, K.; Gong, M.; Lischinski, D.; Cohen-Or, D.; Ascher, U. M.; and Chen, B. 2013. "Mind the gap": Tele-registration for structure-driven image completion. *ACM Transactions on Graphics*, 32(6): 174:1–174:10.

Kim, H. G.; Park, M.; Lee, S.; Kim, S.; and Ro, Y. M. 2021. Visual comfort aware-reinforcement learning for depth adjustment of stereoscopic 3D images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1762–1770.

Li, R.; Liu, S.; Wang, G.; Liu, G.; and Zeng, B. 2022. JigsawGAN: Auxiliary learning for solving Jigsaw puzzles With generative adversarial networks. *IEEE Transactions on Image Processing*, 31: 513–524.

Li, Z.; Kiseleva, J.; and De Rijke, M. 2019. Dialogue generation: From imitation learning to inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6722–6729.

Liu, J.; Ren, J.; Lu, Z.; He, W.; Cui, M.; Zhang, Z.; and Bai, R. 2022. Cross-Document Attention-based Gated Fusion Network for Automated Medical Licensing Exam. In *Expert Systems With Applications*, volume 205, 117588.

Ma, C.; Rao, Y.; Lu, J.; and Zhou, J. 2021. Structure-preserving image super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–14.

Mirjalili, S. 2019. Genetic algorithm. In *Evolutionary algorithms and neural networks*, 43–55. Springer.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving Jigsaw puzzles. In *European Conference on Computer Vision*, 69–84.

Paikin, G.; and Tal, A. 2015. Solving multiple square Jigsaw puzzles with missing pieces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4832–4839.

Paumard, M. M.; Picard, D.; and Tabia, H. 2018a. Image reassembly combining deep learning and shortest path problem. In *European Conference on Computer Vision*, 153–167.

Paumard, M. M.; Picard, D.; and Tabia, H. 2018b. Jigsaw puzzle solving using local feature co-occurrences in deep neural networks. In *IEEE International Conference on Image Processing*, 1018–1022.

Paumard, M. M.; Picard, D.; and Tabia, H. 2020. Deepzzle: Solving visual Jigsaw puzzles with deep learning and shortest path optimization. *IEEE Transactions on Image Processing*, 29: 3569–3581.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2016. Prioritized experience replay. In *International Conference on Learning Representations*.

Sholomon, D.; David, O.; and Netanyahu, N. S. 2013. A genetic algorithm-based solver for very large Jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1767–1774.

Son, K.; Hays, J.; and Cooper, D. B. 2014. Solving square Jigsaw puzzles with loop constraints. In *European Conference on Computer Vision*, 32–46. Springer.

Tomar, M.; Sathuluri, A.; and Ravindran, B. 2019. MaMiC: macro and micro curriculum for robotic reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 10053–10054.

Vesselinova, N.; Steinert, R.; Perez-Ramirez, D. F.; and Boman, M. 2020. Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access*, 8: 120388–120416.

Woo, H.; Lee, H.; and Cho, S. 2022. An efficient combinatorial optimization model using learning-to-rank distillation. *optimization*, 3(5.1): 0–9.

Yan, F.; Zheng, Y.; Cong, J.; Liu, L.; Tao, D.; and Hou, S. 2021. Solving Jigsaw puzzles via nonconvex quadratic programming with the projected power method. *IEEE Transactions on Multimedia*, 23: 2310–2320.

Yang, X.; Wang, Y.; Chen, K.; Xu, Y.; and Tian, Y. 2022. Fine-grained object classification via self-supervised pose alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 7399–7408.

Yao, C.; Wang, S.; Zhang, J.; He, W.; Du, H.; Ren, J.; Bai, R.; and Liu, J. 2021. rPPG-based spoofing detection for face mask attack using efficientnet on weighted spatial-temporal representation. In *IEEE International Conference on Image Processing*, 3872–3876.

Zhang, J.; Zhang, Q.; Ren, J.; Zhao, Y.; and Liu, J. 2022. Spatial-context-aware deep neural network for multi-class image classification. In *2022 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2022 - Proceedings*, 1960–1964.

Zhang, K.; and Li, X. 2014. A graph-based optimization algorithm for fragmented image reassembly. *Graphical Models*, 76(5): 484–495.

Zhang, K.; Yu, W.; Manhein, M.; Waggenspack, W.; and Li, X. 2015. 3D fragment reassembly using integrated template guidance and fracture-region matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2138–2146.

Zong, Z.; Zheng, M.; Li, Y.; and Jin, D. 2022. MAPDP: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9980–9988.