ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096300

SOLVING JIGSAW PUZZLE OF LARGE ERODED GAPS USING PUZZLET DISCRIMINANT NETWORK

Xingke Song[⋆] *Xiaoying Yang*[⋆] *Jianfeng Ren*[⋆†] *Ruibin Bai*[⋆†] *Xudong Jiang* [‡]

[⋆]School of Computer Science, University of Nottingham Ningbo China, China
[†]Nottingham Ningbo China Beacons of Excellence Research and Innovation Institute, University of Nottingham Ningbo China, China
[‡]School of Electrical & Electronic Engineering, Nanyang Technological University

ABSTRACT

Solving Jigsaw puzzles has recently become an emerging research topic. Traditionally, boundary similarities are utilized for puzzle reassembly. In this paper, we solve Jigsaw Puzzles of Large Eroded Gaps (JPLEG), where boundary similarities are weak and image semantics are the only feasible clues. Inspired by human strategy in solving a puzzle, we introduce the concept of puzzlet, where fragments are gradually combined to form puzzlets of different sizes until the completion of the puzzle. Two sets of Puzzlet Discriminant Networks are designed to visually perceive whether these puzzlets are correctly reassembled. The puzzle reassembly is then formulated as a combinatorial optimization problem, and solved using a genetic algorithm. The proposed method is evaluated on two large datasets, which shows that it significantly outperforms the state-of-the-art methods for puzzle solving.

***Index Terms*—** Puzzle Reassembly, Puzzlet Discriminant Network, Genetic Aglorithm, Combinatorial Optimization.

1. INTRODUCTION

Automatic puzzle reassembly has been studied in archaeology to assemble 2D artifact fragments such as paintings, textures and frescos, and 3D artifacts such as sculptures and ceramics [1, 2]. The developed techniques have been applied beyond puzzle reassembly, *e.g.*, self-supervised learning [3–5], unsupervised visual representation [6, 7], video spatial understanding [8, 9], and image super-resolution [10].

Puzzle assembly often consists of two steps. 1) Puzzle visual understanding. Traditionally, boundary information is often utilized for solving puzzles, *e.g.*, contour [11], shape [12, 13], and color [14, 15]. Only a few methods focus on the image content to infer the fragment relations [1–3, 16–19]. Recently, deep learning has been widely used in puzzle solving and many other applications [20, 21]. 2) Puzzle reassembly strategies, *e.g.*, greedy method [14, 18], Dijkstra's algorithm [1], meta-heuristic [22], genetic algorithm [23–25], simulated annealing [26], nonconvex quadratic programming [27], and deep reinforcement learning [2].

In this paper, we solve Jigsaw Puzzles of Large Eroded Gaps (JPLEG). The boundary similarity between fragments in this case is weak while fragments' semantic relations are the only feasible clues. Previous methods over-emphasize exploiting the correlation between two adjacent fragments [1–3, 18]. While this local pairing information is important, the image semantics of combined puzzle fragments of a large size is not fully exploited. Inspired by human process of solving puzzles by gradually merging small fragments into bigger ones until completing the puzzle, we propose the concept of puzzlet, which is a combination of spatially adjacent puzzle fragments. A puzzlet can be as small as one piece of fragment, or as big as the whole puzzle. Then, two sets of Puzzlet Discriminant Networks (PDNs) are proposed to visually perceive whether the puzzlet is correctly assembled or not. One separately extracts features from each fragment of the puzzlet to determine whether the puzzlet is correctly assembled, and the other checks the correctness directly based on the whole puzzlet. These two sets of PDNs could better extract the visual clues for puzzle reassembly than simple pairing clues.

The puzzle-reassembly is then formulated as a combinatorial optimization problem, aiming to find a permutation of puzzle fragments that maximizes the summation of evidence from all puzzlets of different sizes at different locations. The number of permutations increases exponentially with the number of fragments. Thus, a genetic algorithm is proposed to solve this problem. The proposed Puzzle Discriminant Network with Genetic Algorithm (PDN-GA) is evaluated on JPLEG-3 of 3×3 fragments and JPLEG-5 of 5×5 fragments. The proposed PDN-GA significantly and consistently outperforms all the compared methods on the both datasets.

Our contributions are two-fold: 1) The proposed two sets of Puzzlet Discriminant Networks better perceive the visual clues from puzzlets. 2) The proposed combinational opti-

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.
Corresponding author: jianfeng.ren@nottingham.edu.cn.

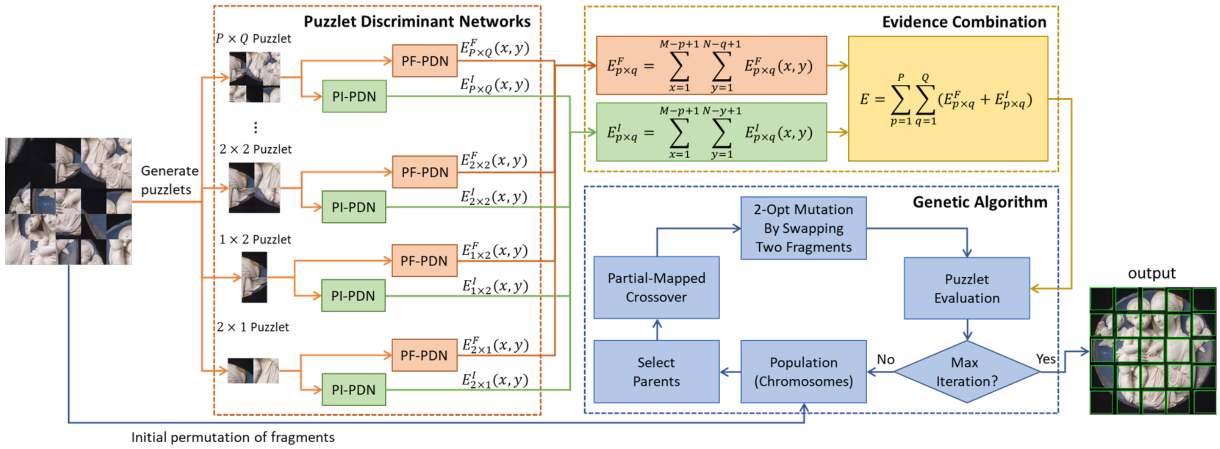

Fig. 1: Block diagram of the proposed method for puzzle solving. The target is to find a permutation of fragment placement that maximizes the evidence E aggregated from the visual clues perceived by two kinds of Puzzlet Discriminant Networks, PF-PDNs and PI-PDNs. A Genetic Algorithm is designed to find the optimal permutation by systematically swapping fragments.

mization formulation aggregates all perceived visual clues and facilitates the solution using genetic algorithm.

2. PROPOSED PDN-GA

2.1. Overview of Proposed PDN-GA

To solve the JPLEG problem, a Puzzle Discriminant Network with Genetic Algorithm is proposed. Formally, given a puzzle of sizes $M \times N$, a set of puzzlets $\boldsymbol{P}_{p\times q}(x, y)$ could be extracted, where $p \leq M, q \leq N$ are the size of puzzlet, and (x, y) denotes its location. As shown in Fig. 1, two sets of Puzzlet Discriminant Networks are designed to perceive the puzzlets visually. One set extracts features from each segment of the puzzlet using Siamese Networks to determine whether the puzzlet is correctly assembled, known as Per-Fragment Puzzlet Discriminant Networks (PF-PDN). The other decides the completion of the puzzlet by treating the whole puzzlet as an input image, known as Per-Image Puzzlet Discriminant Networks (PI-PDN). The visual clues from puzzlets of different sizes at different locations are aggregated to form the visual evidence. The target is to find the permutation of segments that maximizes the evidence. A genetic algorithm is designed to solve this problem.

2.2. Puzzlet Discriminant Networks

2.2.1. Per-Fragment Puzzlet Discriminant Network

The proposed Per-Fragment Puzzlet Discriminant Network consists of a set of shared-weight network branches, where each branch is adapted from VGG-16 to extract features from one fragment. The visual clues of all fragments are combined using a linear layer, and aggregated through three fully connected layers with batch normalization, to decide whether the puzzlet is correctly assembled or not. By utilizing PF-PDNs, we obtain the evidence $E^F_{p\times q}(x, y)$ for the puzzlet of size $p\times q$ at location (x, y) being correctly assembled. Fig. 2 shows an example of PF-PDN for puzzlets of size 2×2.

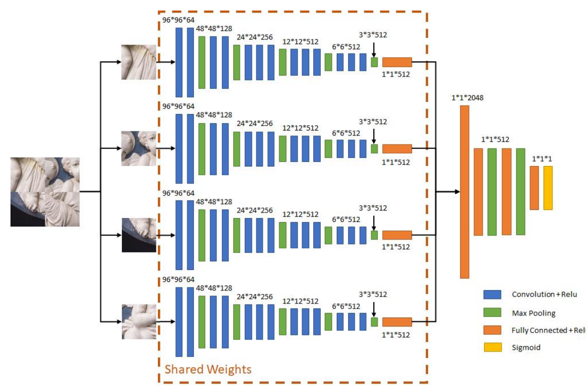

Fig. 2: An example of Per-Fragment Puzzlet Discrimant Network to perceive puzzlet of 2×2 fragments.

2.2.2. Per-Image Puzzlet Discriminant Network

The proposed PF-PDN aggregates the semantic relations between fragments in a puzzlet, but it lacks an overall perception of the puzzlet. Furthermore, as the number of fragments in the puzzlet grows, the PF-PDN will grow proportionally. Thus, a Per-Image Puzzlet Discriminant Network is proposed to globally inspect whether the puzzlet has been correctly assembled or not. Instead of extracting features from each segment of the puzzlet, PI-PDN treats the whole puzzlet as the input image to determine whether the puzzlet is correctly assembled. By utilizing PI-PDNs, we obtain the evidence $E^I_{p\times q}(x, y)$ for the puzzlet of size $p \times q$ at location (x, y) being correctly assembled. Fig. 3 shows an example of PI-PDN for 2×2 puzzlet, where the backbone is adapted from VGG-16.

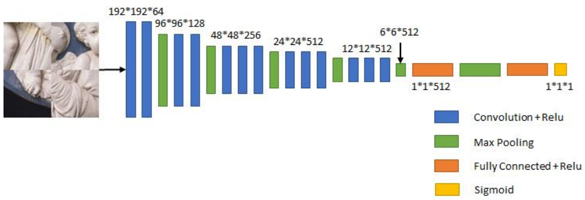

Fig. 3: An example of Per-Image Puzzlet Discrimant Network to perceive puzzlet of size 2×2.

2.3. Formulation of Combinatorial Optimization

To solve the puzzle reassembly problem, we formulate it as a Combinatorial Optimization problem. More specifically, denote Π as the set of all possible permutations of fragment indices $\{1, 2, 3, \ldots, M \cdot N\}$.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 puzzle layout. It is infeasible to directly find the mapping function $\mathcal{M}$.In this paper, we design a genetic algorithm to find a sequence of mapping $\{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$, $\pi_{t+1} = \mathcal{M}_t(\pi_t)$, so that the final permutation is the same as π_G. It is achieved by maximizing the evidence E of the puzzle being correctly assembled,

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

where E can be estimated by summing up the evidence of all puzzlets of different sizes at different locations.

$$E = \sum_{p=1}^{P} \sum_{q=1}^{Q} E_{p \times q}, \tag{3}$$

where $E_{p \times q}$ is the evidence of puzzlets of size $p \times q$.

$$E_{p \times q} = \sum_{x=1}^{M-p+1} \sum_{y=1}^{N-q+1} E^F_{p \times q}(x, y) + E^I_{p \times q}(x, y). \tag{4}$$

2.4. Genetic Algorithm for Puzzle Reassembly

Given the objective function defined in Eqn. (2), we resort to genetic algorithm (GA) [28] to derive the optimal mapping sequence. Specifically, N_a permutations of puzzle placement are encoded as the initial solution. N_p parents are selected from the population with the highest evidence E defined in Eqn. (3) obtained through the PF-PDNs and the PI-PDNs. For the crossover action in GA [28], we utilize tournament selection to select part of the sequence of fragments from both parent chromosomes. Different from traditional GA, the chromosome in our problem is a non-repeating sequence of fragments. To deal with the possible overlap in a crossover, we resort to the Partial-Mapped Crossover method [28] to generate N_c children. For the mutation action in GA, we use the 2-optimization [28] as an exchange mutation method that swaps two positions of fragments in the chromosome. The aforementioned steps repeat till convergence. Finally, the chromosome with the highest E is chosen as the optimal sequence of fragments. The proposed GA is summarized in Algorithm 1.

Algorithm 1 Genetic algorithm for puzzle solving.

Input: N_a random permutations as the population.
Output: The optimal permutation maximizing E in Eqn. (3).
Initialize: L: max number of iterations;
Generate N_a permutations as the initial population;
for iteration $1, 2, \ldots, L$ **do**
    Select N_p parents that maximize E in Eqn. (3).
    Use Partial-Mapped Crossover to generate N_c children;
    Mutation by using the 2-optimization [28] to swap the positions of two fragments;
    Evaluate E for the population using Eqn. (3);
end for
Select the permutation with maximal E.

3. EXPERIMENTAL RESULTS

3.1. Experimental Settings

To evaluate the proposed PDN-GA on the JPLEG problem, two datasets are constructed based on the METropolitan (MET) Museum of Art open-source image and data resources [1]. 12,000 images of painting, engraving, and artifacts are chosen to create the puzzle, among which 10,000 are randomly selected as the training set, and the remaining 2,000 are used for testing. For the JPLEG-3 dataset of puzzles with 3×3 fragments, each image is resized and square-cropped to 398×398 pixels. 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 proposed method in a more challenging scenario, we develop a JPLEG-5 dataset of Jigsaw puzzles with 5×5 fragments 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.

To validate the proposed PDN-GA, we design two sets of PF-PDNs for 1×2 puzzlets and 2×1 puzzlets respectively, and one set of PI-PDNs for 2×2 puzzlets. The proposed PDN-GA is compared with the following state-of-the-art methods. Suffix '-Pair' denotes methods only utilizing the pairwise relations between adjacent neighbors [18]. Suffix '-Puzzlet' denotes methods utilizing the visual evidence from puzzlets.

Deepzzle [1] solves JPLEG problems using Dijkstra algorithm with branch cut on pairwise relations between a center fragment and its neighbors. 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 [14, 18] is an efficient method to reassemble Jigsaw puzzles by greedily selecting fragments that maximize the visual evidence.

Tabu Search [22] 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.

Proposed Genetic Algorithm updates the permutation of fragments to maximize the evidence, with $N_a = 64$, $N_p = 4$, $N_c = 60$, a crossover rate of 90%, a mutation rate of 10%, and up to 50 iterations. To avoid premature convergence, the fitness of duplicated solutions is decayed by 10%.

The results are evaluated 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.

3.2. Comparisons on JPLEG-3 Dataset

The experimental results on the JPLEG-3 dataset are shown in Table 1. The proposed method achieves the best performance over all the compared methods under four evaluation metrics. Compared with the previous best method, Deepzzle [1], the performance gain is 13.3% for Perfect measure. Compared with Tabu search [22], the performance gain of GA is 0.3% for Perfect measure when using pairwise relations as clues, and 1.7% when using puzzlets. Comparisons of execution time and model size are shown in Table 2. The training time and the number of parameters of our method are comparable to others while the testing time is a bit higher than others.

Table 1: Comparison results on the JPLEG-3 dataset. The proposed method significantly outperforms Deepzzle [1] and other compared methods in all four evaluation metrics.

Method	**Perfect**	**Absolute**	**Horizontal**	**Vertical**
Deepzzle [1]	44.9%	74.0%	67.2%	59.0%
Greedy-Pair [18, 29]	55.2%	79.5%	74.0%	74.2%
Tabu-Pair [22]	55.2%	79.0%	73.8%	73.7%
GA-Pair [28, 30]	56.2%	79.7%	74.6%	74.6%
Greedy-Puzzlet	56.5%	80.1%	74.8%	74.9%
Tabu-Puzzlet	56.5%	81.0%	75.6%	75.6%
Our method	**58.2%**	**81.3%**	**76.5%**	**76.4%**

Table 2: Comparisons of execution time and the number of model parameters on the JPLEG-3 dataset.

Method	Training time	Test time per image	#Params
Deepzzle [1]	3.2 h	1.1 s	15.8M
Greedy-Pair [18, 29]	5.9 h	0.6 s	31.5M
Tabu-Pair [22]	5.9 h	0.8 s	31.5M
GA-Pair [21,27]	5.9 h	1.5 s	31.5M
Greedy-Puzzlet	11.6 h	1.0 s	46.8M
Tabu-Puzzlet	11.6 h	1.3 s	46.8M
Our Method	11.6 h	2.8 s	46.8M

Table 3: Comparison results on the JPLEG-5 dataset. Our method significantly outperforms the compared methods.

Method	**Perfect**	**Absolute**	**Horizontal**	**Vertical**
Deepzzle [1]	0.0%	21.9%	10.9%	10.7%
Greedy-Pair [14, 18]	0.1%	24.1%	12.6%	12.3%
Tabu-Pair [22]	2.5%	39.2%	25.2%	25.1%
GA-Pair [28, 30]	3.7%	40.5%	26.4%	26.2%
Greedy-Puzzlet	0.0%	30.7%	17.4%	18.2%
Tabu-Puzzlet	3.9%	42.9%	29.2%	29.0%
Our Method	**6.1%**	**44.3%**	**30.8%**	**30.6%**

3.3. Comparisons on JPLEG-5 Dataset

The experimental results on the JPLEG-5 dataset are shown in Table 3. The proposed method achieves significantly better performance than all the compared methods under the four evaluation metrics. Our method obtains the best Perfect measure of 6.1% while Deepzzle [1] perfectly assembles 0 puzzle. It is indeed challenging to reassemble puzzles with large eroded gaps. Compared with Tabu search, GA achieves significant performance gains on the all four metrics. Sample results for puzzle reassembly for Deepzzle [1] and proposed PDN-GA are presented in Fig. 4. It is clear that many pieces are not correctly reassembled by Deepzzle [1], while the proposed PDN-GA could perfectly reassemble the puzzle.

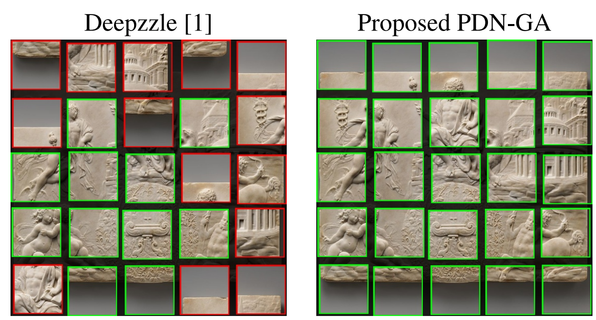

Fig. 4: Visualization of the reassembling results on sample image from the JPLEG-5 dataset.

4. CONCLUSION

In this paper, the puzzle is systematically divided into puzzlets to aggregate the discriminant information of whether the puzzle is correctly reassembled. Two sets of Puzzlet Discriminant Networks are proposed to visually perceive the puzzlets, where the proposed PF-PDN locally extracts the visual clues each fragment, and the proposed PI-PDN extracts the global clue on the completion status of the puzzlet. All visual clues are aggregated into the evidence of the puzzle being correctly reassembled. The puzzle is then solved by finding a sequence of optimal mappings that maximizes the visual evidence. A genetic algorithm is proposed to solve the problem. The proposed PDN-GA significantly outperforms all the compared methods on both JPLEG-3 and JPLEG-5 datasets.

5. REFERENCES

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

[2] Xingke Song, Jiahuan Jin, Chenglin Yao, Wang Shihe, Jianfeng Ren, and Ruibin Bai, "Siamese-discriminant deep reinforcement learning for solving jigsaw puzzles with large eroded gaps," in *AAAI*, 2023.

[3] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[4] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Haoming Zhou, Minghui Qiu, and Ling Shao, "Kaleido-BERT: Vision-language pre-training on fashion domain," in *CVPR*, 2021, pp. 12647–12657.

[5] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu, "Ms2l: Multi-task self-supervised learning for skeleton based action recognition," in *ACM International Conference on Multimedia*, 2020, p. 2490–2498.

[6] Jinyu Yang, Chunyuan Li, Weizhi An, Hehuan Ma, Yuzhi Guo, Yu Rong, Peilin Zhao, and Junzhou Huang, "Exploring robustness of unsupervised domain adaptation in semantic segmentation," in *ICCV*, 2021, pp. 9194–9203.

[7] Pengguang Chen, Shu Liu, and Jiaya Jia, "Jigsaw clustering for unsupervised visual representation learning," in *CVPR*, 2021, pp. 11526–11535.

[8] Xuhui Yang, Yaowei Wang, Ke Chen, Yong Xu, and Yonghong Tian, "Fine-grained object classification via self-supervised pose alignment," in *CVPR*, 2022, pp. 7399–7408.

[9] Jialu Zhang, Qian Zhang, Jianfeng Ren, Yitian Zhao, and Jiang Liu, "Spatial-context-aware deep neural network for multi-class image classification," in *ICASSP*, 2022, pp. 1960–1964.

[10] Cheng Ma, Yongming Rao, Jiwen Lu, and Jie Zhou, "Structure-preserving image super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2021.

[11] H. Huang, K. Yin, M. Gong, D. Lischinski, D. Cohen-Or, U. M. Ascher, and B. Chen, ""Mind the gap": Tele-registration for structure-driven image completion.," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 174:1–174:10, 2013.

[12] K. Zhang and X. Li, "A graph-based optimization algorithm for fragmented image reassembly," *Graphical Models*, vol. 76, no. 5, pp. 484–495, 2014.

[13] K. Zhang, W. Yu, M. Manhein, W. Waggenspack, and X. Li, "3D fragment reassembly using integrated template guidance and fracture-region matching," in *CVPR*, 2015, pp. 2138–2146.

[14] A. C. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," in *CVPR*, 2012, pp. 382–389.

[15] K. Son, J. Hays, and D. B. Cooper, "Solving square Jigsaw puzzles with loop constraints," in *ECCV*. Springer, 2014, pp. 32–46.

[16] Wentao He, Jialu Zhang, Jianfeng Ren, Ruibin Bai, and Xudong Jiang, "Hierarchical convit with attention-based relational reasoner for visual analogical reasoning," in *AAAI*, 2023.

[17] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving Jigsaw puzzles," in *ECCV*, 2016, pp. 69–84.

[18] D. Bridger, D. Danon, and A. Tal, "Solving Jigsaw puzzles with eroded boundaries," in *CVPR*, 2020, pp. 3526–3535.

[19] Ru Li, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng, "Jigsawgan: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks," *IEEE Transactions on Image Processing*, vol. 31, pp. 513–524, 2022.

[20] Chenglin Yao, Shihe Wang, Jialu Zhang, Wentao He, Heshan Du, Jianfeng Ren, Ruibin Bai, and Jiang Liu, "rPPG-based spoofing detection for face mask attack using EfficientNet on weighted spatial-temporal representation," in *ICIP*, 2021, pp. 3872–3876.

[21] Jiandong Liu, Jianfeng Ren, Zheng Lu, Wentao He, Menglin Cui, Zibo Zhang, and Ruibin Bai, "Cross-document attention-based gated fusion network for automated medical licensing exam," in *Expert Systems With Applications*, 2022, vol. 205, p. 117588.

[22] Kamil Adamczewski, Yumin Suh, and Kyoung Mu Lee, "Discrete tabu search for graph matching," in *IEEE International Conference on Computer Vision*, 2015, pp. 109–117.

[23] D. Sholomon, O. David, and N. S. Netanyahu, "A genetic algorithm-based solver for very large Jigsaw puzzles," in *IEEE CVPR*, 2013, pp. 1767–1774.

[24] Smaragda Markaki and Costas Panagiotakis, "Jigsaw puzzle solving techniques and applications: a survey," *The Visual Computer*, pp. 1–17, 2022.

[25] Ijaz Ahmad, Suk-Seung Hwang, and Seokjoo Shin, "Determining jigsaw puzzle state from an image based on deep learning," in *International Conference on Artificial Intelligence in Information and Communication*, 2022, pp. 030–032.

[26] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *Handbook of Metaheuristics*, pp. 1–35. Springer, 2019.

[27] Fang Yan, Yuanjie Zheng, Jinyu Cong, Liu Liu, Dacheng Tao, and Sujuan Hou, "Solving jigsaw puzzles via nonconvex quadratic programming with the projected power method," *IEEE Transactions on Multimedia*, vol. 23, pp. 2310–2320, 2021.

[28] Seyedali Mirjalili, "Genetic algorithm," in *Evolutionary algorithms and neural networks*, pp. 43–55. Springer, 2019.

[29] G. Paikin and A. Tal, "Solving multiple square Jigsaw puzzles with missing pieces," in *CVPR*, 2015, pp. 4832–4839.

[30] Dror Sholomon, Omid E David, and Nathan S Netanyahu, "An automatic solver for very large jigsaw puzzles using genetic algorithms," *Genetic Programming and Evolvable Machines*, vol. 17, no. 3, pp. 291–313, 2016.