Received 23 January 2024, accepted 17 February 2024, date of publication 20 February 2024, date of current version 27 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368004

RESEARCH ARTICLE

2D Irregular Fragment Reassembly With Deep Learning Assistance

YANGJIE CAO[1], (Member, IEEE), ZHENG FANG[1], HUI TIAN[1], AND RONGHAN WEI[1,2], (Member, IEEE)
[1]School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China
[2]School of Mechanics and Safety Engineering, Zhengzhou University, Zhengzhou 450002, China

Corresponding author: RongHan Wei (profwei@zzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52171193 and Grant 61972092.

ABSTRACT Fragment reassembly is widely used in fields such as archaeology and forensics. This paper introduces an algorithm for reassembling irregular fragments, enabling the reconstruction of arbitrarily segmented irregular fragments without any prerequisites. The reassembly procedure encompasses feature extraction, local pairwise matching, and global composition. We design a classification network to assess the compatibility of fragment pairings. In order to alleviate the impact of unavoidable errors in the local matching phase, we put forth an error assessment criterion calculated from the contour differences at the junctions, a two-stage reassembly strategy involving both initial and candidate phases, and a region completeness evaluation standard based on the size of the background region to gauge the ultimate composition outcomes. Additionally, due to the limited availability of publicly shared datasets at present, we created a dataset comprising 1000 sets of randomly divided irregular fragments for training and testing. The experimental results show the excellent performance of this algorithm in terms of accuracy and completeness in fragment reassembly for both public and self-constructed datasets.

INDEX TERMS Irregular fragments, classification network, error assessment, completeness evaluation.

I. INTRODUCTION

Fragment reassembly refers to rearranging and combining multiple fragments to restore an object to its original form. This technology is widely used in fields such as archaeology and forensics. In the past, it has been used to restore wall paintings [1], rejoin oracle bone fragments [2], reconstruct damaged skulls [3], and restore shredded documents [4]. These studies have not only conserved labor efforts but also averted additional harm to valuable artifacts.

The problem of fragment reassembly can be classified as a variant of the jigsaw puzzle problem. Teaching a computer to tackle jigsaw puzzles presents a formidable challenge, a concept first introduced by Freeman and Gardner [5]. Subsequent research has grappled with the complexity of leveraging extracted fragment features to accurately identify adjacent fragments and align them, which is inherently an NP-hard problem [6]. Conventionally, characteristics like geometric shapes, colors, and textures of fragment edges are employed to ascertain pairwise matches. However, these approaches are susceptible to noise, missing pieces, and edge approximations. Erroneous matches can ensnare the global reassembly process within local optima, consequently impacting the ultimate outcome of the global composition.

In recent years, deep learning technology has achieved remarkable progress and accomplishments across various image processing tasks due to continuous advancements in computer hardware and computing capabilities. Deep learning networks can autonomously learn image features and representations, enabling them to adapt effectively to diverse scenarios and data distributions. The Convolutional Neural Networks (CNNs) have gained extensive traction in computer vision applications, utilizing their convolutional layers to capture a range of image features, including edges, textures, and shapes. The Vision Transformer (ViT) [7] is a deep learning model based on the Transformer [8] architecture, specifically designed for computer vision tasks. Unlike CNNs, ViT processes images in a serialized manner

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

and leverages the self-attention mechanism intrinsic to the Transformer architecture to comprehend global relationships embedded within the image.

This paper employs a hybrid methodology integrating conventional image processing techniques with deep learning. It analyzes both the local and global features of pairwise matched fragments, resulting in a noteworthy enhancement in the precision of such pairings. Furthermore, a two-stage framework encompassing initial and candidate reassembly strategies is implemented to circumvent the risk of being ensnared in local optima during the composition procedure.

In summary, our contributions are as follows:

1) In the local matching stage, we introduce a lightweight, multi-scale classification network to evaluate the congruence of pairwise matches. An error assessment calculated from the contour differences at the junctions is put forth to gauge the precision of fragment pairings.
2) In the global composition stage, we propose a two-stage reassembly strategy, leveraging a region overlap-based pruning algorithm to facilitate initial and candidate fragment reassembly. We also introduce a region completeness evaluation criterion to appraise the final outcomes of composition.
3) we create a dataset comprising 1000 sets of irregular fragment images obtained from the publicly available ImageNet dataset [9] through random segmentation.

The experimental results, conducted on public and self-constructed datasets, amply demonstrate the superior performance of our proposed technique in terms of matching accuracy and completeness.

II. RELATED WORK

The computer-aided method to solving the fragment reassembly problem was first introduced by Freeman and Gardner [5]. Subsequent research in this area can be broadly categorized into two main methods: conventional image processing and deep learning networks.

A. CONVENTIONAL IMAGE PROCESSING

Yao et al. proposed a puzzle-solving method that combines shape matching and image feature fusion [10]. The method involves image extraction, corner detection, boundary shape matching, and image merging. Amigoni et al. extracted boundary contour curvature and color as feature descriptors for fragment images [11]. They first found the longest common part using curvature and then matched and aligned the fragments based on color to reassemble the image. Makridis et al. introduced another method that utilizes geometric features and color matching [12]. They used the IPAN99 algorithm and Kohonen self-organizing feature mapping color restoration technique to extract geometric and color features of contour feature points for fragment matching. Cho et al. proposed a probability-based solver for solving square jigsaw puzzles [13]. They measured various compatibility metrics and determined the optimal measurement criterion. Pomeranz et al. improved upon Cho et al.'s work and proposed a novel and superior compatibility metric. They designed a greedy strategy-based fully automatic jigsaw solver [14]. Paikin and Tal introduced an algorithm for the assembly of square jigsaw puzzles [15]. They employed a greedy approach that relies on prediction-based dissimilarity and the best buddies metric to address puzzles with missing pieces and those containing elements from multiple puzzles. Kamran et al. identified the optimal adjacency pairs by solving a Longest Common Subsequence problem [16]. A multi-piece alignment was employed to eliminate erroneous matches and globally compose the final image. Huroyan et al. designed a mathematical framework based on graph connectivity Laplacian, which allows the reconstruction of jigsaw puzzles even when the positions and orientations are unknown [17]. Derech et al. introduced a fully automatic general algorithm for solving jigsaw puzzles [1]. It simplifies the continuity problem of two fragments into a boundary-matching problem by inferring the outer contour region of the fragments. This method has good applicability for worn or faded fragments. Panagiotakis et al. proposed a graph-based approach for coastline matching [18]. They applied the CD-RCP algorithm for commonality detection to match pairs of coastlines.

B. DEEP LEARNING NETWORKS

Sholomon et al. proposed a deep neural network model called DNN-Buddies for measuring compatibility [19]. Le and Li introduced a CNN model named JigsawNet, founded on attention mechanisms within stitching regions and employing a boosted training strategy to assess pairwise compatibility [20]. They implemented a novel loop-closure algorithm to enhance the robustness of global composition. Rika et al. introduced a hybrid scheme incorporating a genetic algorithm and deep learning to address Portuguese tile panel problems [21]. They presented an improved genetic algorithm solver for tile placement and utilized a deep learning-based compatibility measure to differentiate between adjacent and non-adjacent pieces. Paumard et al. introduced a puzzle-solving method named Deepzzle [22]. Initially, they employed a CNN-based deep learning model to predict the adjacency relationships between fragments, followed by the application of an enhanced graph shortest path algorithm for the reassembly of fragments. Ostertag and Beurton used siamese neural networks to predict the matching relationship between fragments and then created a graph structure for global reassembly [23]. Li et al. proposed a self-supervised learning generative adversarial model called JigsawGAN [24]. It arranges fragments into categories and reconstructs images with correct fragment orders based on features. Zhang et al. proposed an internal similarity network for the reconstruction of Oracle Bone fragment images [2]. The model focuses on leveraging residual network architecture, gradient features, and internal similarity to enhance discrimination performance. Chen et al. proposed a classification model named Jigsaw-ViT [25]. They incorporated solving jigsaw puzzles as a self-supervised auxiliary loss, aiming to enhance both generalization and robustness. Hosseini et al. introduced an approach for spatial puzzle solving [26]. They employed an end-to-end neural architecture based on Diffusion Models named PuzzleFusion, capable of addressing Cross-cut jigsaw puzzles, Voronoi jigsaw puzzles, and room layout arrangements. Song et al. introduced a Siamese-Discriminant Deep Reinforcement Learning (SD^{2}RL) method to address Jigsaw puzzles with substantial eroded gaps [27]. They devised two sets of Siamese Discriminant networks to assess the compatibility of vertical and horizontal neighbors, along with a Deep Q-Network to identify the optimal swap sequence. Zhang et al. proposed an approach that leverages multimedia techniques for the restoration of ancient manuscripts [28]. They initially identified candidate locations among fragments using text-based localization. Subsequently, a Siamese network, utilizing contour similarity assessment, was employed to refine candidate matching pairs. Finally, reassembly was accomplished through a hierarchical closure loop method.

The pertinent literature reviewed above indicates that deep learning is not only viable but also exhibits exceptional performance in determining the positional relationships among fragments. In recent years, research on fragment reassembly has predominantly concentrated on square puzzles, regular polygons, and specific scenarios. However, this paper addresses the reassembly of irregular fragments, a scenario more reflective of the majority of real-life situations. Regarding the algorithm, we introduce an initial step in assessing adjacent relationships among fragments, where we redundantly store additional matching pairs, accommodating a certain number of errors in the matching process. This approach mitigates the risk of erroneously filtering out correctly matched pairs. In the subsequent assembly process, pruning is applied to eliminate erroneous matches, and through a two-stage reassembly, the accuracy and completeness of the assembly are enhanced. This method also introduces novel perspectives for future research in the field.

III. METHODS

This paper proposes a 2D irregular fragment reassembly algorithm consisting of three main parts: feature extraction, local pairwise matching, and global composition. The overall flow of the algorithm is illustrated in Fig. 1.

A. FEATURE EXTRACTION

In the experiments of this paper, each complete image is segmented into multiple irregular fragments. The segmented image is subjected to binary processing, contour extraction, and polygon approximation [29]. Data is extracted from these fragments, encompassing attributes such as polygon side lengths, centroids, bounding circle radii, and the mean color value of the contour edges. These collected features are then used for further analysis and to support the fragment reassembly algorithm.

1) POLYGON APPROXIMATION ALGORITHM.

This paper utilizes the Douglas-Peucker polygon approximation algorithm [29], which can simplify a curve containing a large number of vertices by iteratively preserving key points to obtain an approximate polygon representation. The specific steps are as follows:

1) Firstly, for a curve C, we can represent it as $C = \widetilde{p_i p_j}$, where p_i and p_j represent the starting and ending points of the curve, respectively;
2) Connect p_i and p_j to form a straight line $L_{(p_i,p_j)}$ and calculate the Euclidean distance from all other vertices to the line $L_{(p_i,p_j)}$. Find the vertex p_k that has the maximum distance from the line $L_{(p_i,p_j)}$. If this distance is greater than the threshold thr, consider p_k as a key point;
3) Use the key point p_k as a splitting point to divide the curve C into two sub-curves C_1 and C_2, where $C = C_1 + C_2$, $C_1 = \widetilde{p_i p_k}$, $C_2 = \widetilde{p_k p_j}$.
4) Repeatedly apply the same steps 1-3 through recursion until no key point can be found with a distance greater than the threshold thr.
5) Finally, all the key points obtained will constitute the approximate polygon representation after simplifying the curve.

In this paper, the threshold thr in the algorithm is dynamically set by performing contour length statistics on the fragments to be assembled. The minimum contour length is found and multiplied by a hyperparameter α, as shown in the following formula (1):

$$thr = \min(L_{C_1}, L_{C_2}, \ldots, L_{C_n}) * \alpha \tag{1}$$

By dynamically adjusting the threshold based on the minimum contour length, the algorithm can adapt to different fragment sizes and complexities, enhancing the effectiveness and accuracy of the reassembly process. The hyperparameter α allows fine-tuning the threshold based on specific requirements or characteristics of the input fragments.

B. LOCAL PAIRWISE MATCHING

In this section, we identify the longest common sub-contour by establishing edge length thresholds and corresponding mean value thresholds for contour colors. This approach helps determine the optimal matching positions between fragments. A sequence of candidate images is generated by matching contour points. Subsequently, a classification network is applied to evaluate the adjacency of these fragments. The key points for assembly and their corresponding error scores are recorded.

1) ERROR SCORE

Given two matching curves C_i and C_j, $C_i = (C_{i1}, C_{i2}, \ldots, C_{in})$, $C_j = (C_{j1}, C_{j2}, \ldots, C_{jn})$. The error score S, which evaluates

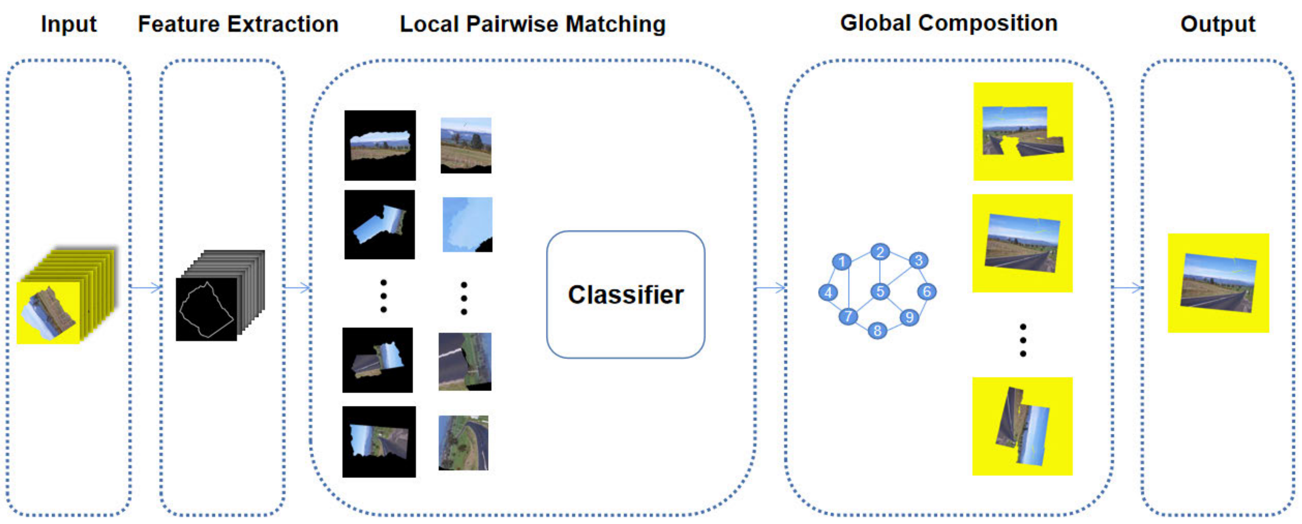

FIGURE 1. The pipeline of our reassembly algorithm.

the mismatch between two matched fragments, is calculated using the following formula:

$$\omega 1 = \sum \frac{L_{C_{ik}} + L_{C_{jh}}}{2} * M_{(C_{ik},C_{jh})} \tag{2}$$

$$\omega 2 = \sum \frac{D_{(C_{ik},C_{jh})}}{N_{(C_i,C_j)}} \tag{3}$$

$$\omega 3 = \sum \frac{L_{C_{ik}} + L_{C_{jh}}}{2} \tag{4}$$

$$\omega 4 = N_{(C_i,C_j)} \tag{5}$$

$$S = \frac{\omega 1 * \omega 2}{\omega 3 * \omega 4} + \beta * \omega 2 \tag{6}$$

where:

- L represents the length of the corresponding contour in C_i or C_j.
- M represents the mean color value the corresponding contour in C_i or C_j.
- N is the number of corresponding contours matched between C_i and C_j.
- D is the length error between the corresponding contour in C_i and C_j.
- β is a hyperparameter.

A lower error score indicates higher compatibility between the matched fragments. The algorithm aims to minimize this error score during fragment reassembly to achieve more accurate and reliable results.

2) CLASSIFICATION NETWORK

MobileViT [30] is an optimized model based on ViT [7], which combines the strengths of ViT and CNN to create a lightweight network suitable for mobile devices. It leverages the advantages of both architectures to achieve efficient and accurate processing. This work employs a multi-scale MobileViT to determine the compatibility of fragment-matching regions. The specific network structure is depicted in Fig. 2

The network takes a series of 224 × 224 × 3 images as input. The feature extraction process begins with standard convolutional layers, followed by lightweight convolutional blocks MobileNetv2 (MV2) [31] and MobileViT blocks [30]. After feature extraction, 1 × 1 convolutions are applied to modify the channel dimensions of feature maps that vary in size. These feature maps, sharing the same channel count but varying in size, are subsequently upsampled and fused. The fused feature map is further expanded using 1 × 1 convolutions for channel augmentation. Finally, the globally pooled feature map is fed into fully connected layers to accomplish classification tasks.

a: MOBILENETV2 BLOCK

As is shown in Fig. 3, the network architecture includes tow forms of MobileNetV2 block: regular and down-sampling. The regular block is a bottleneck depth-separable convolution with residuals. For a given input feature map, The block applies a 1 × 1 convolutional layer to expand channels, a 3 × 3 depthwise separable convolution layer to encode information and a 1 × 1 convolutional layer to change channels. Swish is chosen as the non-linearity due to its robust empirical performance [32], and batch normalization is applied during training.

b: MOBILEVIT BLOCK

The Fig. 4 describes the MobileViT block's operation on a given input tensor $X \in \mathbb{R}^{H\times W\times C}$. The block applies a $n \times n$ convolutional layer followed by a 1 × 1 convolutional layer, resulting in $X_L \in \mathbb{R}^{H\times W\times d}$. To capture global representations with spatial inductive bias, X_L is unfolded into N non-overlapping flattened patches $X_U \in \mathbb{R}^{P\times N\times d}$, where $P = wh$ and $N = \frac{HW}{P}$. Inter-patch relationships

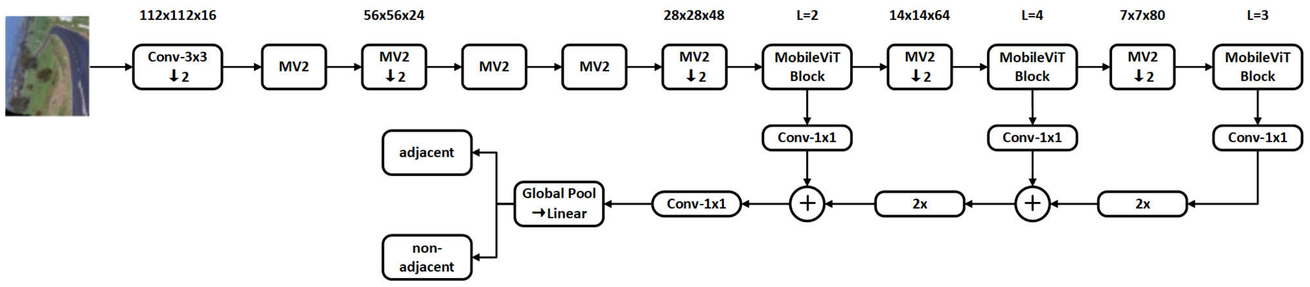

FIGURE 2. The architecture of the classification network.

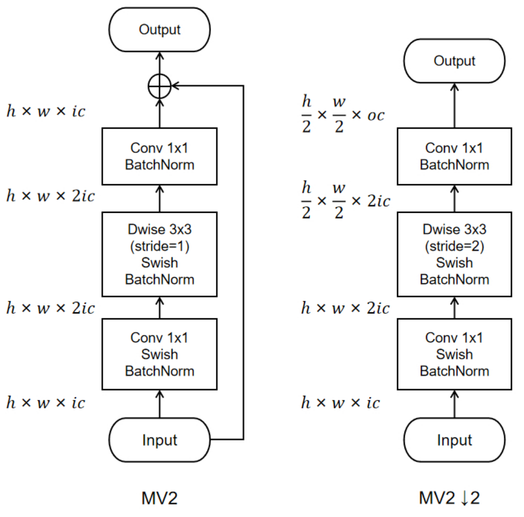

FIGURE 3. The architecture of MobileNetV2 block.

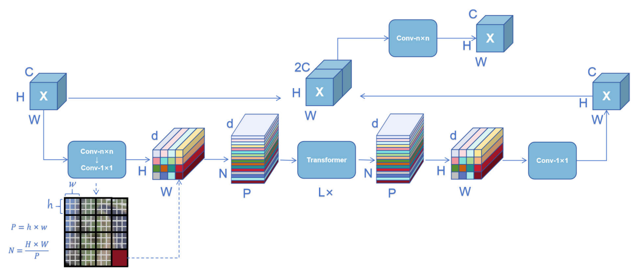

FIGURE 4. The architecture of MobileViT block.

are encoded using transformers, producing $X_G \in \mathbb{R}^{P \times N \times d}$ for each $p \in \{1, .., P\}$. X_G is then folded to obtain $X_F \in \mathbb{R}^{H \times W \times d}$, which is projected to a low C-dimensional space through a 1×1 convolution. This result is combined with X through concatenation. Finally, another $n \times n$ convolutional layer fuses these concatenated features. The MobileViT block can model the local and global information in an input tensor with fewer parameters [30].

C. GLOBAL COMPOSITION

In this section, we construct an undirected weighted graph, where the weight of each edge represents the error score of the corresponding fragment pair. The composition process proceeds in two stages: initial reassembly and candidate reassembly. The outcomes of assembly are assessed and produced via the utilization of the region completeness evaluation metric.

1) INITIAL REASSEMBLY

We represent the pairwise matching relationships between fragments using an undirected weighted graph G, $G = \{V, E\}$, $v_i \in V$, $e_{i,j,s} \in E$. The v_i represents the i-th fragment, and the $e_{i,j,s}$ represents a matching between the i-th and j-th fragments, with an error score s indicating the mismatch between them. To find the initial vertex for the reassembly process, we calculate the degree of each vertex and select the vertex with the highest degree as the initial vertex. The degree of a vertex is the number of edges connected to it. Starting from the initial vertex, we use the Prim algorithm to obtain the Minimum Spanning Tree (MST) of the graph. To determine the order of fragment reassembly, we use three different methods: greedy search (GS), breadth-first search (BFS), and depth-first search (DFS). During the composition process, We record the fragments deemed incompatible matches due to excessively large overlapping regions using GS_cut, BFS_cut, and DFS_cut, respectively. We also record the paired fragments that have been successfully assembled using GS_cor, BFS_cor, and DFS_cor. The resulting assembled images are stored in $Init\_imgs$. Finally, we take the union of GS_cut, BFS_cut, and DFS_cut to obtain the set of cut-off vertices, denoted as $Cut\_vertices$. Similarly, we take the intersection of GS_cor, BFS_cor, and DFS_cor to obtain the set of paired fragments, denoted as Cor_edges. Algorithm 1 outlines the steps involved in the initial reassembly process.

2) CANDIDATE REASSEMBLY

In the candidate reassembly process, we start by setting the error scores of the edges in Cor_edges to zero in graph G. Then, we iterate through the list of vertices in $Cut\_vertices$, using each vertex as the initial vertex. We apply the Prim algorithm for each initial vertex to find the MST of the graph G. Subsequently, we use the greedy search method based on the error scores to perform reassembly, starting from each initial vertex. We generate a series of candidate-reassembled images by repeating this process for

Algorithm 1 Initial Reassembly

```
Input: $G \leftarrow (V, E), v_{start}$
Output: $Init\_imgs, Cut\_vertices, Cor\_edges$
1: Staring from vertex $v_{start}$, obtain MST using the Prim's algorithm.
2: $GS\_img, GS\_cut, GS\_cor \leftarrow GS(MST)$
3: $BFS\_img, BFS\_cut, BFS\_cor \leftarrow BFS(MST)$
4: $DFS\_img, DFS\_cut, DFS\_cor \leftarrow DFS(MST)$
5: $Init\_imgs \leftarrow (GS\_img, BFS\_img, DFS\_img)$
6: $Cut\_vertices \leftarrow GS\_cut \cup BFS\_cut \cup DFS\_cut$
7: $Cor\_edges \leftarrow GS\_cor \cap BFS\_cor \cap DFS\_cor$
```

all vertices in *Cut_vertices*. These candidate results are evaluated based on our specified region completeness criteria to determine the final reassembled image with the best overall quality and compatibility. Algorithm 2 outlines the steps involved in the candidate reassembly process.

Algorithm 2 Candidate Reassembly

```
Input: $G \leftarrow (V, E)$, $Cut\_vertices$, $Cor\_edges$
Output: $Can\_imgs$
1: while $Cor\_edges$ is not empty do
2:     Extract the edge$(u, v)$ from $Cor\_edges$
3:     Set $s_{(u,v)} \leftarrow 0$ in $E$
4: end while
5: Initialize an empty set $Can\_imgs$ to store the final results.
6: while $Cut\_vertices$ is not empty do
7:     Extract the vertex $v_{start}$ from $Cut\_vertices$
8:     Staring from vertex $v_{start}$, obtain MST using the Prim's algorithm.
9:     $GS\_img \leftarrow GS(MST)$
10:    Add $GS\_img$ to $Can\_imgs$
11: end while
```

3) REGION OVERLAP-BASED PRUNING

During the reassembly phase, fragment pairs are systematically combined one by one until the ultimate fragment is reconstructed. In an ideal reassembly, fragments should exhibit no overlapping sections, a criterion that can serve as a gauge for evaluating the precision of the reassembly process. Nonetheless, owing to the inevitable pixel misalignments between fragments, errors may arise and accumulate with the increased number of assembled fragments during composition. This can result in correctly matched fragments overlapping with already assembled regions, as is shown in Fig. 5. When deliberating the inclusion of a new fragment in the reassembly process, the extent of overlap during the amalgamation is assessed by analyzing the proportion of the overlapping region in relation to the freshly incorporated fragment's area. If this overlap ratio surpasses a predefined threshold denoted as *ol_thr*, the fragment is deemed incompatible with the assembled region. Consequently, the respective edge connecting the vertices corresponding to these two fragments in the graph G is pruned ($E - \{e_{i,j,s}\}$).

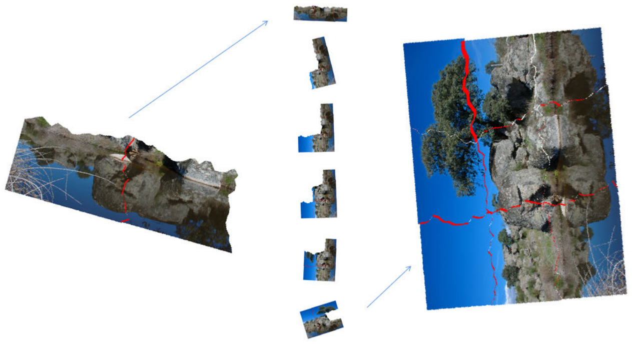

FIGURE 5. Pairwise Fragment Reassembly Process, with the red region indicating the overlapping area during the reassembly process.

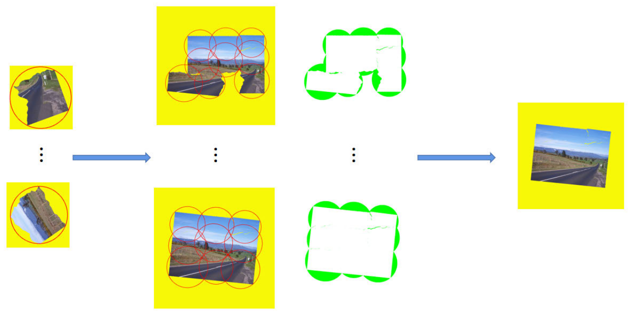

FIGURE 6. The corresponding process of obtaining the final result through region completeness evaluation.

4) REGION COMPLETENESS EVALUATION

After a two-stage (initial-candidate) reassembly, we obtain a series of reassembled images $I = \{I_{gs}, I_{bfs}, I_{dfs}, I_{C_1}, \ldots, I_{C_n}\}$. Based on the centroids and minimum bounding circle radii of each fragment obtained in the feature extraction stage, each image I_X is divided into regions $R_{I_X} = \{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$. The size of the background area A_{I_X} contained in each partitioned region is calculated, and the image with the smallest background area I^* is identified as the final output, as shown in Fig. 6. The corresponding formula is as follows:

$$A(I_X) = \sum_{x,y}^{R_{I_X}} P(I_X(x, y)) \tag{7}$$

$$P(I_X(x, y)) = \begin{cases} 0 & if\ I_X(x, y) \neq background \\ 1 & if\ I_X(x, y) = background \end{cases} \tag{8}$$

$$I^* = \arg\min_X A(I_X) \tag{9}$$

IV. EXPERIMENTS

We conducted experiments on public datasets, including the ImageNet dataset [9], MIT dataset [13], and BGU dataset [14]. We randomly selected images for each dataset

TABLE 1. The performance of different classification networks.

Network	TP	FP	FN	TN	Precision	Recall	F1 Score	Params(M)
Vgg16 [35]	11777	274	223	16607	0.977	0.981	0.979	134.2
ResNet34 [36]	10959	281	1041	16600	0.975	0.913	0.943	21.29
MobileNetV3 [37]	11827	411	173	16470	0.966	0.985	0.975	1.52
ViT [7]	10265	2761	1735	14120	0.788	0.855	0.82	85.8
Swin-T [38]	11753	110	247	16771	0.99	0.979	0.985	27.52
MobileViT [30]	11490	159	510	16722	0.986	0.957	0.971	0.95
Ours	11824	159	176	16722	0.986	0.985	0.986	0.93

and divided them into 9, 36, and 100 fragments. From the ImageNet dataset, a subset of 1000 images was chosen to generate training and validation datasets for the classification network. These datasets included a total of 17,000 negative samples and 12,000 positive samples.

A. TRAINING DETAILS

We implement the classification network in PyTorch [33], and all the experiments are performed on an RTX 3090 GPU with 300 epochs. The AdamW optimizer [34] was used for training the model. The batch size was set to 40. The learning rate was set to 0.0002. The weight decay rate was set to 0.01. We use the CrossEntropyLoss function as the loss function. To mitigate overfitting, the label_smoothing parameter was set to 0.1. The training loss and validation accuracy curves is shown in Fig. 7.

B. COMPARISON WITH EXISTING CLASSIFICATION NETWORKS

In this study, the proposed classification dataset was used to evaluate various classification networks. The tested classification networks include: CNN-based Networks [35], [36], [37] and Transformer-based Networks [7], [30], [38]. The networks include light-weight and heavy-weight architectures. We present a comprehensive analysis of the performance metrics obtained from various classification networks in the table 1. The key metrics considered include True Positives(TP), False Positives(FP), False Negatives(FN), True Negatives(TN), Precision, Recall, F1 Score, Params.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

The experimental results demonstrate that the proposed classification network achieves impressive performance, even when utilizing fewer parameters. The network exhibits remarkable accuracy and efficiency in addressing the classification task of pairwise matching. This indicates that the proposed model strikes a favorable balance between model complexity and performance, making it a promising choice for scenarios where computational resources may be limited or efficiency is a key consideration. The ability of the network to achieve competitive performance with fewer parameters is a significant advantage, as it enables faster training and inference times while maintaining high accuracy on the classification task.

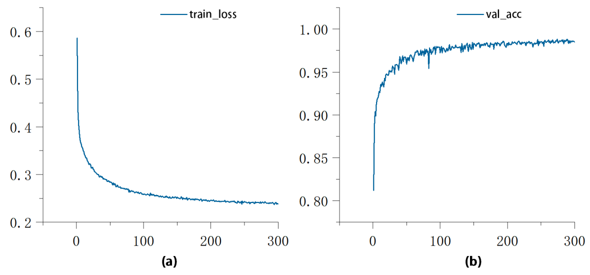

FIGURE 7. Training Loss and Validation Accuracy.

C. REASSEMBLY RESULTS EVALUATION

1) EVALUATION METRICS

As mentioned in [39], there are currently unresolved issues in the field of puzzle problems. One prominent challenge is the absence of a commonly accepted evaluation criteria. To quantitatively evaluate the ultimate reassembly performance, we employed several metrics in [13], [20], and [40]:

Neighbor Correctness (NC): The ratio of correct pairwise matching adjecencies.

Largest Component (LC): The size of the largest correct part in each image.

Perfect Reconstruction (PR): The number of images that were perfectly reconstructed to their original state.

In the context of irregular fragment reassembly, especially when lacking predefined conditions, our focus is on ensuring the accuracy and completeness of the reassembly process. The Neighbor Correctness (NC) metric assesses the accuracy of relative positions between fragments. Additionally, the Largest Component (LC) metric evaluates the overall completeness of the reassembly, while the Perfect Reconstruction (PR) metric counts the number of completely reconstructed images after the final assembly.

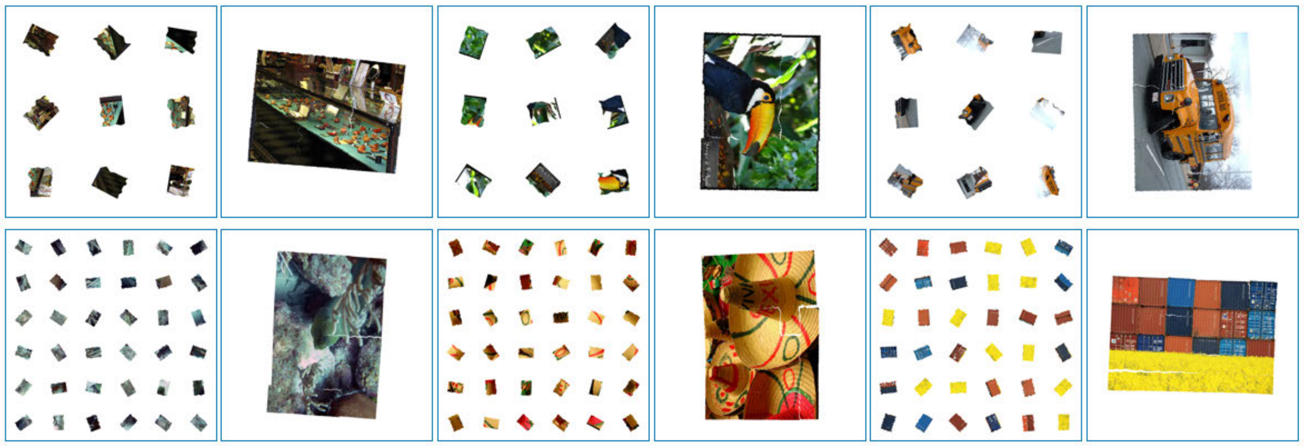
FIGURE 8. Some reassembly results on the ImageNet data. The first row shows a reassembly of three 9-piece puzzles. The second row shows a reassembly of three 36-piece puzzles.

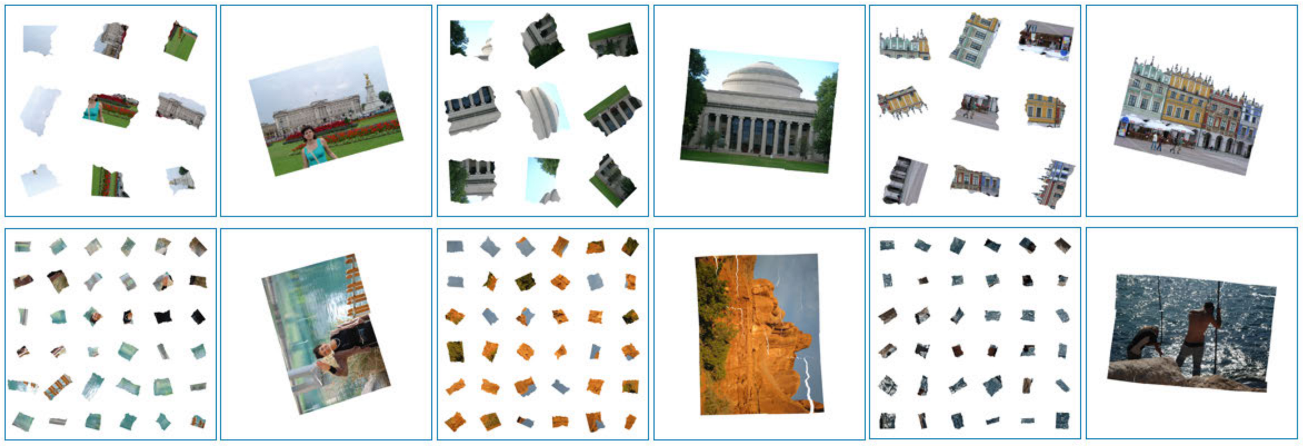
FIGURE 9. Some reassembly results on the MIT and BGU data provided by JigsawNet. The first row shows a reassembly of three 9-piece puzzles. The second row shows a reassembly of three 36-piece puzzles.

TABLE 2. The overall reassembly results on MIT9 and BUG36 datasets provided by JigsawNet [20].

Dataset	Method	NC	LC	Avg.time
MIT9	JigsawNet [20]	**99.2%**	100%	0.76 min
	Ours	84.6%	**100%**	**0.19 min**
BGU36	JigsawNet [20]	89.5%	99.1%	18.75 min
	Ours	**92.5%**	**100%**	**6.6 min**

TABLE 3. The overall reassembly results on 9, 36 and 100 pieces. Single indicates the strategy of single-stage (greedy search). Two indicates the strategy of tow-stage (initial-candidate).

	LC		PR	
	single	two	single	two
3x3	100%	100%	20	20
6x6	97.6%	99.1%	6	8
10x10	60.1%	84.3%	1	3

2) COMPARISON WITH EXISTING METHOD

We conducted tests on the MIT9 and BGU36 datasets provided by JigsawNet [20], and a comparison was made between our method and theirs, as illustrated in Table 2. The results indicate that despite some initial errors in adjacent matching, our approach exhibits outstanding performance in both reassembly completeness and speed.

3) ABLATION STUDY

We compared the single-stage (greedy search) and the two-stage (initial-candidate) reconstruction algorithm using 20 sets of 9-piece fragments, 10 sets of 36-piece fragments, and 10 sets of 100-piece fragments. The results are summarized in Table 3.

Drawing from the findings of our experiments, we can deduce the following insights: When dealing with a limited

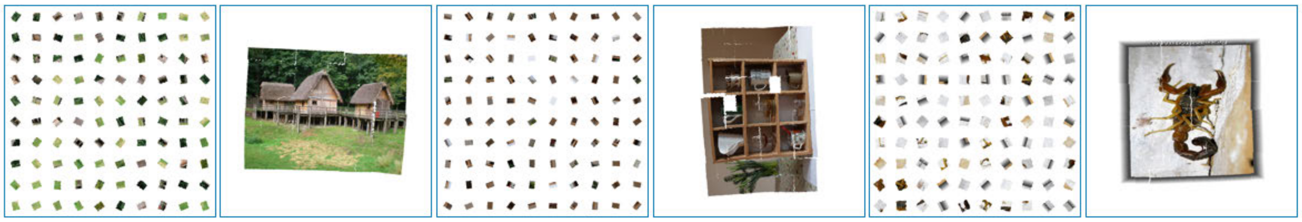

FIGURE 10. The reassembly results of some 100-piece puzzles.

number of fragments, a greedy search strategy for reassembly proves viable. This is because, with fewer fragments, their relative placements exhibit a higher concentration, facilitating the swift elimination of erroneous matches. However, as the fragment count increases and their positions become more dispersed, relying solely on a greedy search approach might inadvertently eliminate accurate matches and lead to being trapped in local optimal solutions. In scenarios characterized by a larger fragment count and greater dispersion of positions, adopting a two-stage (initial-candidate) reassembly method can rectify this issue and yield improved outcomes.

4) REASSEMBLY RESULTS

The performance on a subset of the self-created ImageNet [9] dataset with 9 and 36 fragments is shown in Fig. 8. The performance on the MIT dataset [13] and BGU dataset [14] provided by JigsawNet [20] with 9 and 36 fragments is shown in the in Fig. 9. The partial performance on 100 fragments is shown in Fig. 10.

V. CONCLUSION

In this paper, we introduce an algorithm for reassembling irregular fragments, integrating conventional image processing with deep learning to reconstruct arbitrarily fragmented pieces without requiring prior conditions. The method proposed here introduces several innovations, including a multi-scale MobileViT classification network for evaluating the compatibility of fragment matches, an error assessment criterion calculated from the contour differences at the junctions, a two-stage (initial-candidate) reassembly approach, and a region completeness evaluation standard based on the size of the background region to gauge the ultimate composition outcomes. These contributions collectively enhance the precision and coherence of fragment reassembly. The strategies presented in this study exhibit the potential for broader applicability to reassembly tasks in diverse domains.

REFERENCES

[1] N. Derech, A. Tal, and I. Shimshoni, "Solving archaeological puzzles," *Pattern Recognit.*, vol. 119, Nov. 2021, Art. no. 108065.

[2] Z. Zhang, A. Guo, and B. Li, "Internal similarity network for rejoining Oracle bone fragment images," *Symmetry*, vol. 14, no. 7, p. 1464, Jul. 2022.

[3] Z. Yin, L. Wei, X. Li, and M. Manhein, "An automatic assembly and completion framework for fragmented skulls," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2532–2539.

[4] L. Zhu, Z. Zhou, and D. Hu, "Globally consistent reconstruction of ripped-up documents," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 1–13, Jan. 2008.

[5] H. Freeman and L. Garder, "Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-13, no. 2, pp. 118–127, Apr. 1964.

[6] E. D. Demaine and M. L. Demaine, "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity," *Graphs Combinatorics*, vol. 23, no. S1, pp. 195–208, Jun. 2007.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[10] F.-H. Yao and G.-F. Shao, "A shape and image merging technique to solve jigsaw puzzles," *Pattern Recognit. Lett.*, vol. 24, no. 12, pp. 1819–1835, Aug. 2003.

[11] F. Amigoni, S. Gazzani, and S. Podico, "A method for reassembling fragments in image reconstruction," in *Proc. Int. Conf. Image Process.*, vol. 3, Sep. 2003, pp. 1–10.

[12] M. Makridis and N. Papamarkos, "A new technique for solving a jigsaw puzzle," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 2001–2004.

[13] T. S. Cho, S. Avidan, and W. T. Freeman, "A probabilistic image jigsaw puzzle solver," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 183–190.

[14] D. Pomeranz, M. Shemesh, and O. Ben-Shahar, "A fully automated greedy square jigsaw puzzle solver," in *Proc. CVPR*, Jun. 2011, pp. 9–16.

[15] G. Paikin and A. Tal, "Solving multiple square jigsaw puzzles with missing pieces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4832–4839.

[16] H. Kamran, K. Zhang, M. Li, and X. Li, "An LCS-based 2D fragmented image reassembly algorithm," in *Proc. 13th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Aug. 2018, pp. 1–6.

[17] V. Huroyan, G. Lerman, and H.-T. Wu, "Solving jigsaw puzzles by the graph connection Laplacian," *SIAM J. Imag. Sci.*, vol. 13, no. 4, pp. 1717–1753, Jan. 2020.

[18] C. Panagiotakis, S. Markaki, E. Kokinou, and H. Papadakis, "Coastline matching via a graph-based approach," *Comput. Geosci.*, vol. 26, no. 6, pp. 1439–1448, Dec. 2022.

[19] D. Sholomon, O. E. David, and N. S. Netanyahu, "DNN-buddies: A deep neural network-based estimation metric for the jigsaw puzzle problem," in *Proc. Int. Conf. Artif. Neural Netw.*, Barcelona, Spain. Cham, Switzerland: Springer, 2016, pp. 170–178.

[20] C. Le and X. Li, "JigsawNet: Shredded image reassembly using convolutional neural network and loop-based composition," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4000–4015, Aug. 2019.

[21] D. Rika, D. Sholomon, E. David, and N. S. Netanyahu, "A novel hybrid scheme using genetic algorithms and deep learning for the reconstruction of Portuguese tile panels," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2019, pp. 1319–1327.

[22] M.-M. Paumard, D. Picard, and H. Tabia, "Deepzzle: Solving visual jigsaw puzzles with deep learning and shortest path optimization," *IEEE Trans. Image Process.*, vol. 29, pp. 3569–3581, 2020.
[23] C. Ostertag and M. Beurton-Aimar, "Matching ostraca fragments using a Siamese neural network," *Pattern Recognit. Lett.*, vol. 131, pp. 336–340, Mar. 2020.
[24] R. Li, S. Liu, G. Wang, G. Liu, and B. Zeng, "JigsawGAN: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks," *IEEE Trans. Image Process.*, vol. 31, pp. 513–524, 2022.
[25] Y. Chen, X. Shen, Y. Liu, Q. Tao, and J. A. K. Suykens, "Jigsaw-ViT: Learning jigsaw puzzles in vision transformer," *Pattern Recognit. Lett.*, vol. 166, pp. 53–60, Feb. 2023.
[26] S. Hosseini, M. A. Shabani, S. Irandoust, and Y. Furukawa, "PuzzleFusion: Unleashing the power of diffusion models for spatial puzzle solving," 2023, *arXiv:2211.13785v3*.
[27] X. Song, J. Jin, C. Yao, S. Wang, J. Ren, and R. Bai, "Siamese-discriminant deep reinforcement learning for solving jigsaw puzzles with large eroded gaps," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 2, pp. 2303–2311, Jun. 2023.
[28] Y. Zhang, Z. Fang, X. Yang, S. Zhang, B. He, H. Dou, J. Yan, Y. Zhang, and F. Wu, "Reconnecting the broken civilization: Patchwork integration of fragments from ancient manuscripts," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 1157–1166.
[29] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geographic Inf. Geovisualization*, vol. 10, no. 2, pp. 112–122, Dec. 1973.
[30] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.
[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[32] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.
[33] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.
[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[37] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
[39] S. Yılmaz and V. V. Nabiyev, "Comprehensive survey of the solving puzzle problems," *Comput. Sci. Rev.*, vol. 50, Nov. 2023, Art. no. 100586.
[40] A. C. Gallagher, "Jigsaw puzzles with pieces of unknown orientation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 382–389.

YANGJIE CAO (Member, IEEE) received the M.Sc. degree in computer science from Zhengzhou University, Zhengzhou, China, in 2006, and the Ph.D. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2012. He is currently a Professor with Zhengzhou University. His current research interests include computer vision and intelligent computing, artificial intelligence, and high-performance computing.

ZHENG FANG received the B.S. degree in information management and information system from Xinyang Normal University, China, in 2015. He is currently pursuing the master's degree with Zhengzhou University. His research interests include computer vision, image processing, and artificial intelligence.

HUI TIAN received the Ph.D. degree in instrument science and technology from Xi'an Jiaotong University, China, in 2020. He is currently with the School of Cyberspace Security, Zhengzhou University, China. His research interests include AI-based biochemical analysis in vitro diagnostics, intelligent sensing and detection microsystems, and bioinformation security.

RONGHAN WEI (Member, IEEE) received the bachelor's degree in life science from National Taiwan University, in 1995, and the master's and Ph.D. degrees in physics from National Taiwan University, in 1997 and 2001, respectively. He is currently a Distinguished Professor with the Hanwei Institute of Internet of Things, School of Cyber Science and Engineering, and School of Mechanics and Engineering Science, Zhengzhou University. At the same time, he is also the Dean of the School of Mechanics and Engineering Science. He has published over a 100 SCI articles and hold more than 20 granted patents. He has undertaken nearly 30 research projects, many of which involve interdisciplinary research. His main research interests include microelectromechanical systems, bio-microfluidic systems and biosensors, intelligent sensing and the Internet of Things, and precision instrument manufacturing.