NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2025; 14(4), 1559-1570

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Jigsaw puzzle solving with template matching

Şablon eşleştirme ile yapboz çözme

Kürşad Uçar[1,*]

[1] Selçuk University, Faculty of Technology, Department of Electrical and Electronics Engineering, 42250, Konya, Türkiye

Abstract

Reassembling fragmented objects is a crucial problem in fields like archaeology, often approached through jigsaw puzzle solutions. This study presents two novel template-matching-based methods for solving jigsaw puzzles. The first method employs a two-stage approach: Principal Component Analysis (PCA) determines the rotation of scattered pieces, followed by template matching to align and position them. The second method directly locates pieces using template matching. Three test puzzles were used to evaluate the effectiveness of these approaches. The results demonstrate that both methods accurately identified piece positions in all cases, proving their robustness and reliability. However, the proposed methods are currently limited to cases where the appearance of pieces is not heavily affected by noise, occlusion, or large-scale rotation.

Keywords: Puzzle reassembly, Template matching, Jigsaw puzzle, Puzzle solving

Öz

Parçalanmış nesneleri yeniden bir araya getirmek, arkeoloji gibi alanlarda sıklıkla yapboz bulmacası çözümleri yoluyla ele alınan önemli bir sorundur. Bu çalışma, yapboz bulmacalarını çözmek için iki yeni şablon eşleştirme tabanlı yöntem sunmaktadır. İlk yöntem iki aşamalı bir yaklaşım kullanır: Temel Bileşen Analizi, dağılmış parçaların dönüşünü belirler, ardından bunları hizalamak ve konumlandırmak için şablon eşleştirme yapılır. İkinci yöntem, şablon eşleştirmeyi kullanarak parçaları doğrudan bulur. Bu yaklaşımların etkinliğini değerlendirmek için üç test bulmacası kullanıldı. Sonuçlar, her iki yöntemin de tüm durumlarda parça konumlarını doğru bir şekilde belirlediğini ve sağlamlıklarını ve güvenilirliklerini kanıtladığını göstermektedir. Ancak önerilen yöntemler şu anda parçaların görünümünün gürültü, tıkanıklık veya büyük ölçekli rotasyondan çok fazla etkilenmediği durumlarla sınırlıdır.

Anahtar kelimeler: Yapboz yeniden birleştirme, Şablon eşleştirme, Yapboz, Yapboz çözme

1 Introduction

Automatic jigsaw puzzle (JP) solving is an intriguing research area that attracts interest from various fields such as pattern recognition, image processing, mathematics, and robotics [1]. A jigsaw puzzle consists of interlocking pieces that form a complete image when assembled. The goal is to disassemble the puzzle, shuffle the pieces, and then reassemble them to reconstruct the original image. The difficulty depends on factors such as the number of pieces, their shape, and the visual composition of the image [1, 2].

Solving the JP generally involves two main steps: understanding the puzzle visually and reassembling the pieces [3]. Several approaches exist for assembling puzzle pieces [4]. Traditionally, boundary knowledge has been utilized, with features such as contours [5, 6], shapes [7], and colors [8, 9] being the most commonly applied. More recently, deep learning-based solutions [1, 2, 10] have gained popularity in this field, as in many other areas.

This study presents two template matching (TM)-based approaches for automatically solving randomly placed JPs. TM is an image processing technique used to find a specific image part within a larger one [11]. For TM to work correctly, the searched image and the template must match in rotation, scale, and color. Applying TM to rotated images remains a challenging problem [12]. The first method utilizes TM for two purposes. Firstly, random JP pieces are detected, regardless of which one they belong to. These pieces are then isolated, and their correct positions are calculated using TM. Since the directions of the JP pieces are also random, TM is applied after rotating them. In the second method, TM directly searches for the final positions of the JP segments.

The novel contributions of this study are as follows Firstly, it performs detection and location finding while detects and localizes the JP pieces are in random positions and angles even when they randomly positioned and oriented. First, it detects and localizes pieces even when they are randomly positioned and oriented. Second, it uses a template matching that is robust to rotation. To the best of the author's knowledge, this is the first application of TM in jigsaw puzzle-solving.

The remainder of this paper is organized as follows: in the next section, the related works are summarized. Section 3 describes the proposed method in detail. Experimental results are examined in Section 4. Finally, Section 5 discusses future research directions and concludes the paper.

2 Related works

In one of the early studies [9], puzzle pieces were matched based on their shape and color. Similarly, [13] proposed three methods that leveraged geometric shapes and

* Sorumlu yazar / Corresponding author, e-posta / e-mail: kucar@selcuk.edu.tr (K. Uçar)
Geliş / Received: 17.02.2025 Kabul / Accepted: 25.09.2025 Yayımlanma / Published: 15.10.2025
doi: 10.28948/ngumuh.1641237

chromatic information. In [14], a Markov random field-based approach was introduced for solving square puzzles with unknown positions and orientations. In [5], both curve and color similarities were used, and rotationally invariant corner detection was employed to identify characteristic points like highly curved corners. To address large gaps between pieces and the lack of pattern or color continuity, [15] used a neural network to estimate the positions of pieces and then sought the optimal reconstruction graph based on these predictions.

Traditionally, puzzle solving has focused on identifying piece boundaries. However, missing or obscured boundaries present major challenges. To overcome this, [16] used a Generative Adversarial Network (GAN) for border completion and classifying adjacent pieces. Similarly, [17] applied a CNN-based encoder to represent each piece relative to its boundary in a hidden embedding space. In [1], a GAN architecture was proposed that combines geometric and semantic information for efficient puzzle solving.

In [18], the JP problem was tackled using a standard Vision Transformer. Alongside the usual classification process in end-to-end training, a puzzle flow was introduced to predict the absolute positions of input patches. A new method for solving larger puzzles using deep learning and Monte-Carlo Tree Search was proposed in [19]. This approach extracted visual features without access to rewards.

Doersch et al. [20] introduced a pretext task for classification and detection by employing a JP solver. Their network solved 3x3 puzzles by predicting the relative positions of surrounding patches with respect to a central patch. However, their experiments did not involve reassembling the patches.

Although these studies have significantly contributed to puzzle solving, they generally do not focus on assembling shuffled and rotated square puzzle pieces. Most of the works emphasize feature extraction from individual pieces rather than on full reassembly. Moreover, deep learning-based approaches often require high computational resources and are highly dependent on the success of the training process. In contrast, image-based approaches may offer a more practical and efficient alternative for solving such puzzles.

3 Methods

This study proposes performing automatic JP solving based on two simple but effective mathematical and image processing methods, PCA and TM. The puzzle solution was carried out with two different approaches based on the mentioned methods.

3.1 Principal component analysis

In this study, Principal Component Analysis (PCA) was utilized to estimate the orientation angles of individual jigsaw puzzle pieces prior to template matching. The method was directly applied to the set of non-zero pixels in each binary mask corresponding to a puzzle piece. The spatial coordinates of these pixels were used as input data for PCA.

For each piece, PCA was used to calculate the first principal component, which indicates the direction of maximum variance in the pixel distribution. This dominant direction was assumed to correspond to the primary geometric axis of the puzzle piece. The angle between this axis and the horizontal reference axis was computed to determine the necessary rotation for alignment.

This angle estimation was used to rotate each piece before matching it to the reference image. The rotation step ensured that the pieces were oriented as close as possible to their correct positions, thereby improving the efficiency and accuracy of subsequent template matching.

PCA was selected for this task due to its robustness, computational efficiency, and ability to operate without training data. It performs reliably even under noisy or incomplete data conditions and adapts effectively to variations in puzzle piece shapes. Since puzzle pieces are generally asymmetric, the principal axis identified by PCA provides a stable and meaningful basis for rotation estimation.

In this work, PCA was not employed for feature dimensionality reduction but solely for orientation estimation. Therefore, no parameter tuning such as the number of components or any solver selection was required, since the eigenvectors of the covariance matrix are uniquely determined by the piece geometry.

All PCA computations were performed using the standard eigenvalue decomposition of the covariance matrix derived from the piece coordinates. The method does not require any domain-specific assumptions and can be generalized to various puzzle configurations.

3.2 Template matching

In this study, Template Matching (TM) is employed as the primary method for determining the correct placement of each puzzle piece within the complete image. Instead of providing a general theoretical overview, emphasis is placed on its practical application. The method involves sliding a given template—i.e., a puzzle piece—over the full image and calculating a similarity metric at each location to identify the best match. Among the available similarity measures in OpenCV, the TM_CCOEFF_NORMED method is chosen due to its robustness to lighting variations and its effectiveness in identifying structural similarity.

This approach calculates the normalized cross-correlation between the mean-adjusted template and the corresponding region of the image. A high positive score indicates strong alignment between the template and the image segment, while low or negative scores reflect poor matches. This metric is particularly beneficial in scenarios where pieces may be partially occluded or embedded in complex backgrounds, as it emphasizes relational structure rather than absolute pixel intensity.

The choice of TM as a core technique stems from both its computational simplicity and theoretical grounding in signal processing. Fundamentally, TM operates on the principle of spatial correlation, identifying regions where pixel intensity patterns closely align. This makes it well-suited for tasks where a known visual component—such as a puzzle piece—must be precisely located within a larger image, especially when edge information or distinct textures are present.

While basic TM is inherently sensitive to scale and rotation variations, this limitation is addressed in the proposed framework by integrating it with PCA-based orientation correction, which aligns the pieces prior to matching. This preprocessing step significantly improves TM's accuracy by ensuring that templates are already in a near-correct orientation before similarity evaluation.

The theoretical justification for using the TM_CCOEFF_NORMED variant lies in its ability to normalize matching scores, thereby mitigating the influence of global brightness and contrast differences between the template and the image. Moreover, it emphasizes structural similarity, which is crucial when matching pieces that may have similar colors but differ in local features or patterns.

Overall, TM provides a deterministic, interpretable, and training-free solution for puzzle piece localization. Its implementation within the proposed method yields accurate placement results across puzzles of varying complexity, especially when combined with orientation correction techniques.

3.3 Proposed method

In this study, two different methods based on TM and PCA are presented.

3.3.1 Method 1: Two stage JP solution

The first method is based on TM and consists of two primary stages. It uses three images of the JP pieces, as depicted in Figure 1. In the first stage, it determines which of the randomly placed pieces in Figure 1(a) match the individual pieces in Figure 1(b). In the second stage, it identifies the final positions of these pieces, shown in Figure 1(c).

In the initial setup, JP pieces are randomly placed on a white background, as depicted in Figure 1(a). To enable the TM-based method, care was taken to ensure that the pieces do not overlap and are spaced apart. In this stage, each individual JP piece in Figure 1(b) is matched with the randomly placed pieces in Figure 1(a). Additionally, PCA is used to calculate the rotation angles of the pieces.

The first step identifies the puzzle pieces within the randomly distributed layout. Color images are first converted to binary using an adaptive filter. A large kernel size, similar to the size of a puzzle piece, is used to reduce the effect of internal color variations. Figure 2 illustrates the differences between classical thresholding and adaptive filtering methods. In Figure 2(b), classical global thresholding is applied, which may not effectively separate puzzle pieces from the background under varying color conditions. Figure 2(c) demonstrates that adaptive filtering, which calculates thresholds based on local regions, provides better separation of puzzle pieces from the background. This improved separation facilitates more accurate detection of puzzle pieces in subsequent processing steps. Consequently, as shown in Figure 2, this helps to separate the pieces from the background. In the resulting binary image, puzzle pieces are detected by outlining regions within a specific size range that corresponds to the actual parts. In other words, regardless of the content of the puzzle piece, the piece can be segmented as a whole with the adaptive filter. Thus, when PCA is applied to the segmented point set, the direction of the piece is determined.

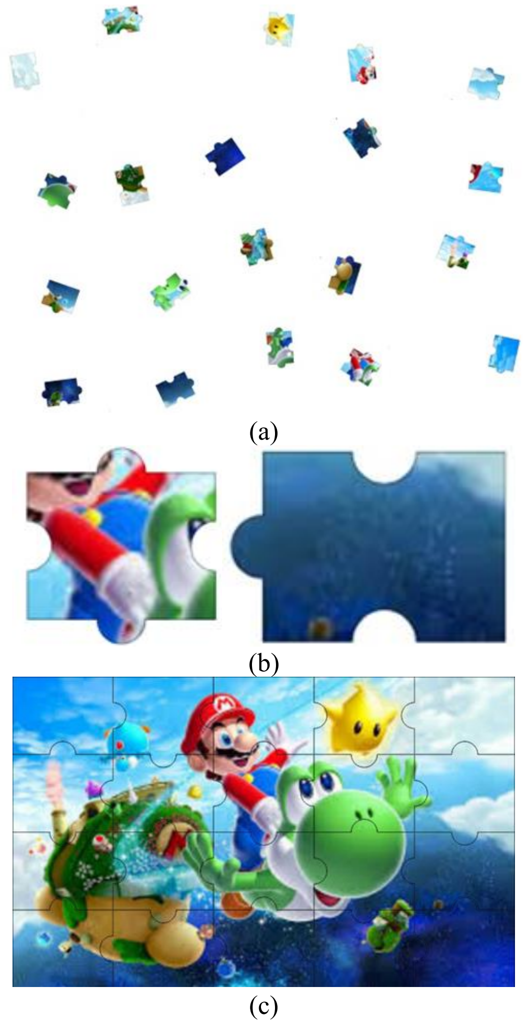


Figure 1. Input images (a) Randomly distributed pieces, (b) individual images of pieces, (c) completed puzzle

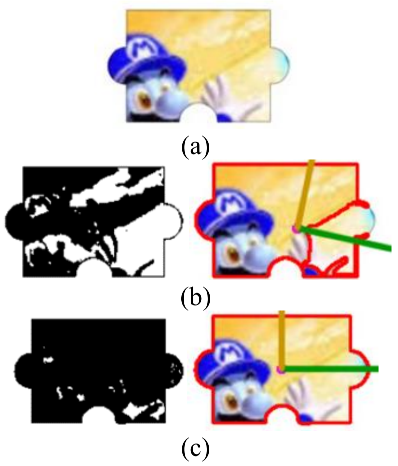


Figure 2. Comparison of classical thresholding and adaptive filtering methods: (a) Original grayscale image, (b) Left: Binary image obtained using classical global thresholding; Right: PCA result applied to the binary image from classical thresholding, (c) Left: Binary image obtained using adaptive filtering; Right: PCA result applied to the binary image from adaptive filtering

The identified regions are treated as 2D point clusters, and PCA [21] is applied to estimate the orientations of the puzzle pieces, as shown in Figure 3. However, the computed orientation lines are typically aligned with the top and right edges of the pieces, regardless of the actual rotation. As illustrated in Figure 4, these lines appear identical across rotations in the [0°, 360°] range. For this reason, the angles are calculated in the range of [0°, 90°] and the result can be 90°, 180° or 270° different from the real angle. This limitation arises from the inability of PCA to account for the specific angle by which a piece should be rotated. In other words, a piece rotated by an angle of 90xk+βº (where $k \in [0, 3]$ and $\beta \in [0°, 90º]$) is computed by PCA as β°. The intersection of the PCA lines is taken as the center of the piece.

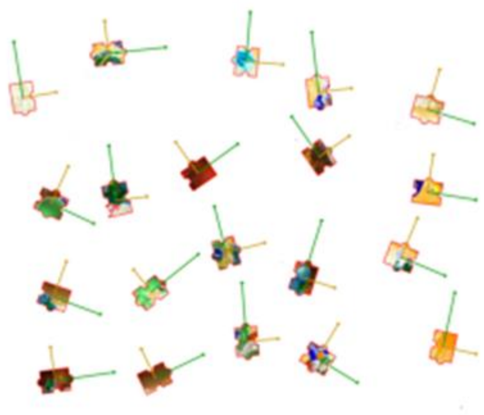


Figure 3. Calculation of the orientation of pieces with PCA

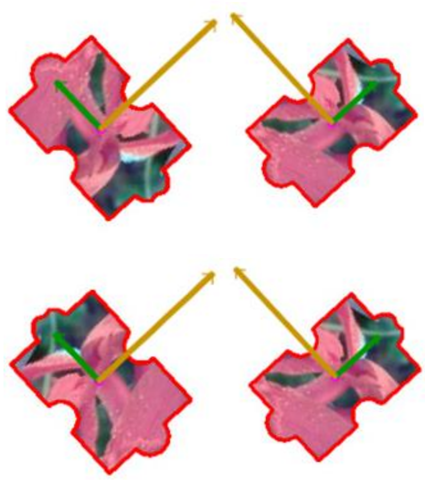


Figure 4. Displaying the orientations of the piece rotated at 90 intervals

TM is performed by sliding the template over the target image, but as the target image grows, so does the computation time. To improve efficiency, the search is limited to areas where JP pieces are located. For each piece, two images are prepared: its ideal form (Figure 1(b)) and its scattered version (Figure 1(a)). PCA is run to determine the midpoint in both images, and they are cropped to the same size with the midpoint at the center. TM is then applied to these images in various orientations. Since rotation introduces black padding that can lower the match score, both images are rotated together in 1° steps up to 90°, ensuring that similar black padding appears in both, as shown in Figure 5. This avoids score degradation due to mismatched borders.

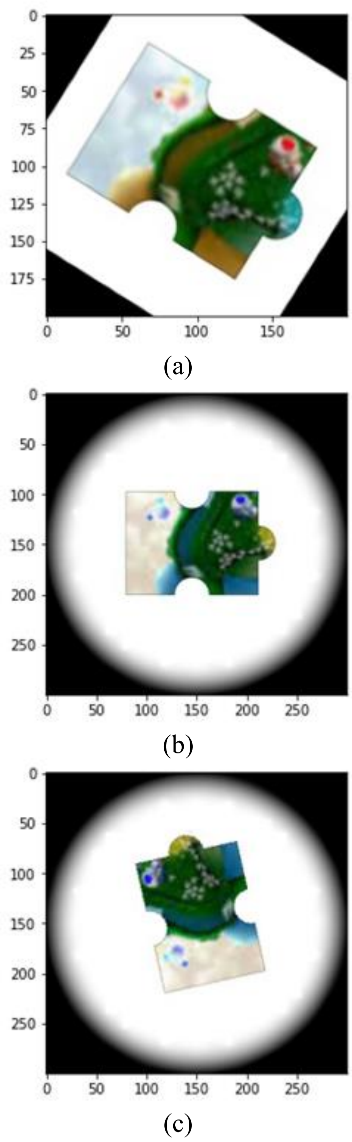


Figure 5. Same black structure is formed around both the target and the template image (a) The black padding in the rotated image, (b) The original state of the piece by one degree rotation, (c) The image at the random position of the piece rotated by one degree

For each fragment in Figure 1(a), the TM score is calculated against each fragment in Figure 1(b). For an n-piece puzzle, a total of n^2 matches are conducted.

Rotating images across the full 360º range for TM is time-consuming. To speed up the process, matching is limited to a ±2º range around the angle θ estimated by PCA. However, since PCA may yield incorrect results for rotations involving multiples of 90º, TM is applied at angles of $90 \times k + \theta \pm 2^{\circ}$ for different values of k. Each original piece is matched with its scattered version based on the highest score, and the correct rotation is determined from the angle that gives the best match.

In the second stage of the first method, the completed JP image (Figure 1(c) is matched with its individual pieces (Figure 1(b)). However, regions like protrusions, recesses, and white borders can lower the TM score. To address this, these regions are removed by cutting them out, as shown in Figure 6. Instead, a smaller template centered around the piece's midpoint is used for TM, allowing for more accurate location matching of the puzzle pieces.

Figure 6. The template created for the second stage left image shows the indented and protruding state of the part and right image is the template created by cutting the edges of the part

3.3.2 Method 2: One stage JP solution

Unlike Method 1, the proposed second method searches for puzzle pieces directly on the completed puzzle image, reducing the number of operations and speeding up the solution. As in the first method, the directions and midpoints of the randomly placed puzzle pieces are calculated by PCA. Based on these midpoints, the puzzle pieces are separated in their current orientations. Then, each piece is rotated by $90 - 90 \times k + \beta^{\circ}$ ($k = 0, 1, 2, 3$), where β is the angle calculated by PCA. A template is generated from the midpoint to the edges of the rotated piece, and TM is used to search for its corresponding position in the completed puzzle image.

Since TM may occasionally produce mismatches, the accuracy of the matches is verified by calculating the distances between the matched positions.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

Here, $i=1,..n-1$ and $j=i+1,...n$ are the piece numbers, and x and y are the upper-left corner coordinates of the quadrilateral where the TM result is maximum. The distance between the matching positions of each piece is then calculated to detect possible mismatches.

$$Msc = \begin{cases} incorrect, & d_{ij} < w \\ correct, & d_{ij} \geq w \end{cases} \tag{2}$$

This allows detection of multiple matches occurring for the same region. In such cases, the matching scores of the pieces assigned to the same region are compared, and the piece with the highest score is accepted as the correct match. TM is then reapplied to find new positions for the incorrectly matched pieces. To prevent them from being matched to the same region again, the previously assigned area is masked by turning it white on the completed puzzle image, as formulated below.

$$Inew(x, y) = 255, x \in [x, x + w], y \in [y, y + h] \tag{3}$$

This process is repeated until all matches are sufficiently spaced apart, so that subsequent TM operations will target different, correct regions.

4 Experiments and results

In this study, Python 3 was utilized to solve three different puzzles. The "TM_CCOEFF_NORMED" method from the OpenCV library [22] was employed for template matching.

4.1 Results of method 1

In the first experiment, a relatively large 4-piece JP, shown in Figure 7 was solved. Because the pieces were quite large, it was sufficient to create templates from only a portion of the pieces. Additionally, due to the simplicity of this example, the matching scores showed clear differences.

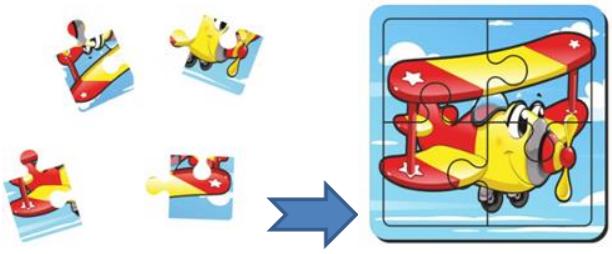


Figure 7. The puzzle solved the first experiment

The second JP, shown in Figure 8(a), consisted of 16 pieces and measured 500 x 375 pixels. The positions and rotation angles of the pieces were determined using the TM method. In this case, JP pieces shared similar colors, and some displayed visual resemblance when rotated (Figure 8(b)). Despite these challenges, the proposed method successfully matched all 16 pieces, achieving a maximum angular error of only 1.95º and an average angular error of 0.836º. Matching scores for selected parts are provided in Figure 9. Notably, the scores were often similar for rotations of 180º, with comparable results observed for $k = 0$ and $k = 2$, as well as for $k = 1$ and $k = 3$. This symmetry was generally consistent, except in cases of correct matching.

The third experiment involved a JP composed of 20 smaller pieces, with dimensions of 663 x 412 pixels, and limited pattern-matching information on the pieces. Despite these challenges, the proposed method successfully solved the puzzle, accurately matching all the pieces. The maximum angular error for this JP was 6.55º, with an average angular error of 1.65º. The matching result of the puzzle is shown in Figure 10. The only noticeable discrepancy occurred in the placement of the piece in the lower-right corner, likely due to the similarity in color characteristics in that region.

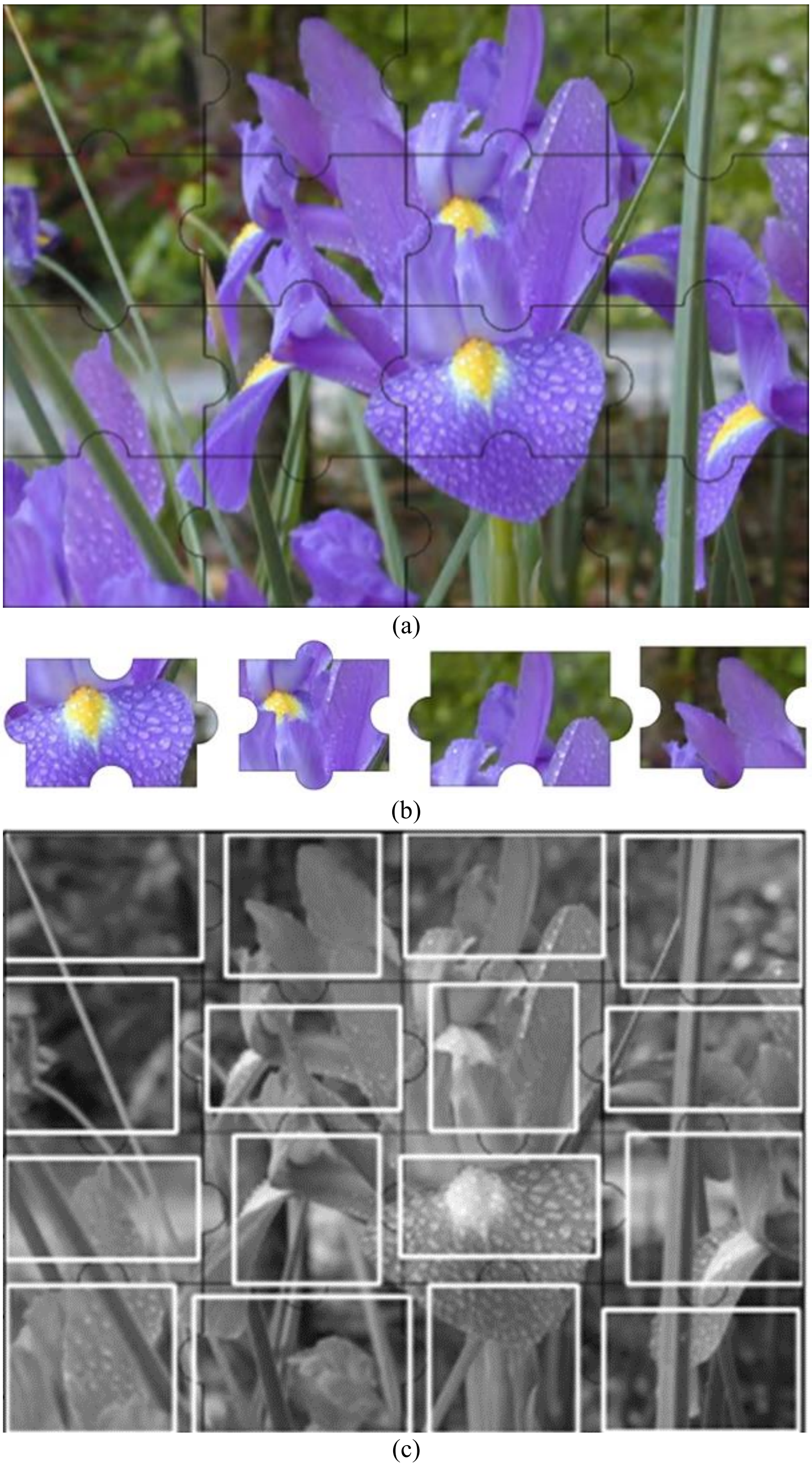


Figure 8. Experiment 2 JP (a) completed puzzle, (b) similar pieces, (c) reassembly of the puzzle

The matching results for this JP are presented in Figure 11, where some parts displayed very close match scores for each *k* value. This behavior was attributed to the similar properties of these parts when rotated, leading to minimal variation in the rotation angles. Despite this, the matching and angle determination were successfully performed.

To further evaluate the effectiveness of the proposed method, the incorrect placement rate in the initial matching step was calculated. This metric represents the proportion of puzzle pieces that were not correctly positioned in the first iteration of template matching. The incorrect placement rates for the second and third puzzles were found to be 0% (0 out

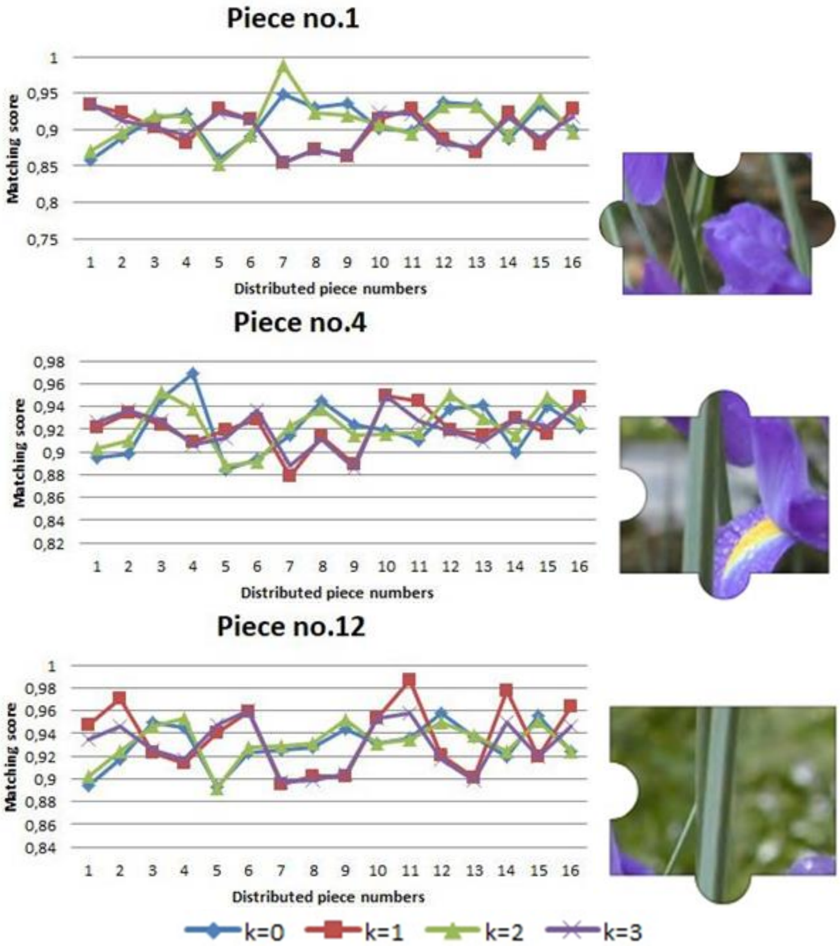
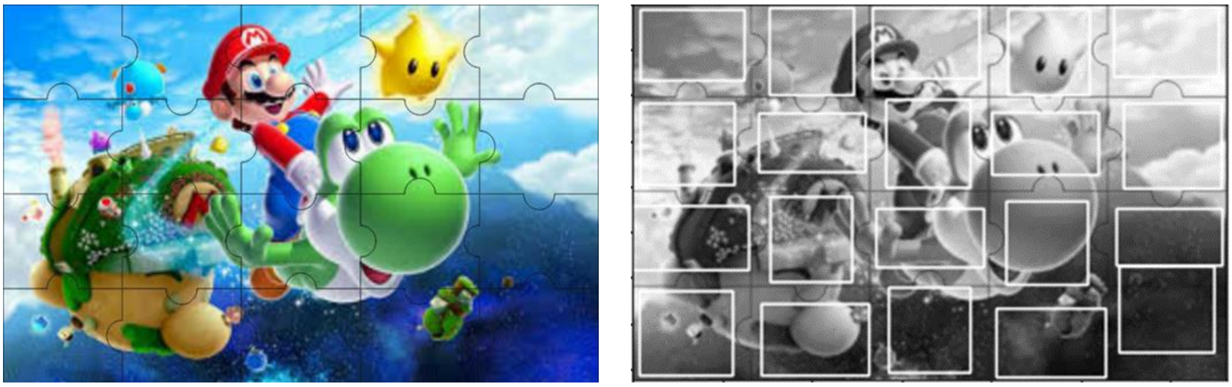


Figure 9. Matching scores of pieces in experiment 2 (piece no. is the random order in Figure 1(b), distributed piece numbers are the complex order in Figure 1(a))



(a) (b)

Figure 10. Third puzzle (a) completed puzzle, (b) reassembly of the puzzle

of 16 pieces) and 10% (2 out of 20 pieces), respectively. These misplacements were successfully corrected in the subsequent iterations using the refinement steps described earlier. The low initial error rates demonstrate the high reliability of the proposed matching approach, even before refinement.

As a result, all three puzzles were successfully reconstructed using the proposed method, and the corresponding rotation angles of the JP pieces were accurately calculated.

The results of this study demonstrate the effectiveness of TM as a viable approach for solving JPs. The proposed method successfully reassembled puzzles of varying complexity and accurately estimated the rotation angles of the individual pieces.

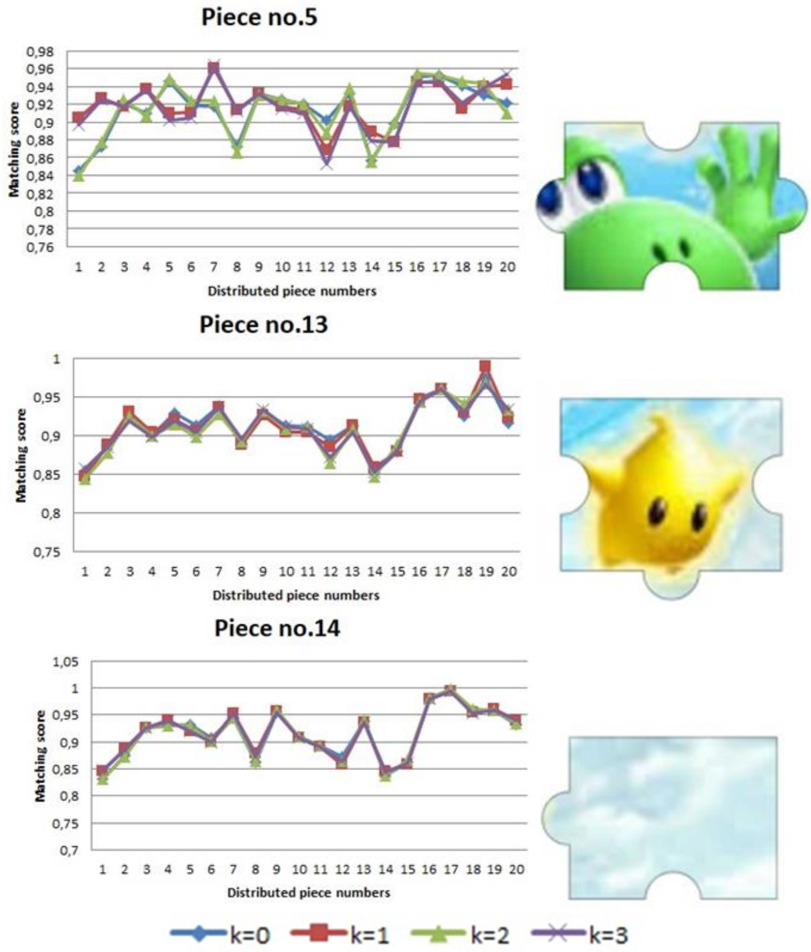


Figure 11. Matching scores of pieces in experiment 3 (piece no. is the random order in Figure 1(b), distributed piece numbers are the complex order in Figure 1(a))

4.2 Results of method 2

The second method was tested on the second and third puzzles. It is faster than the first method because it reduces the number of TM runs and does not require creating a black circle around the pieces. On the other hand, there are two drawbacks. First, the disadvantage is that since the method rotates the pieces by 90° based on the PCA-calculated angle, it cannot achieve accuracy within ±2° due to the impact of added black padding from small rotations. Second, the template size around the midpoint of each piece is fixed, which can lead to information loss from small angular errors. The TM results for the second puzzle are shown in Figure 12. The numbers indicate the order of operations, and the yellow circles indicate the position of the maximum matching score.

The solution steps of the other puzzle are shown in Figure 13. Mismatched pieces are highlighted in the yellow box. In the first step, three pieces matched in the same region, but the piece with the highest matching score was considered correct. To find the correct placement for the remaining pieces, the matched region was closed off, and TM was rerun. The correct positions of the pieces were then identified (see the bottom of Figure 13).

Although the proposed method has successfully reconstructed all test puzzles, certain limitations remain. First, the accuracy of the matching process can be affected when puzzle pieces have similar colors or patterns, particularly under rotation. Additionally, in the second method, using a fixed-size template around the midpoint may lead to information loss, especially in cases with minor angular errors. The rotation-induced padding during TM also restricts fine-tuning of the rotation angle beyond certain limits. For future work, integrating learning-based methods such as deep feature matching or orientation-invariant descriptors could improve robustness. Moreover, dynamic adjustment of the template region based on part geometry may help reduce matching errors. These enhancements could lead to faster and more accurate JP reconstruction.

The proposed method has several advantages. First, it does not require any prior training or large datasets, unlike deep learning-based approaches. This makes it lightweight and suitable for real-time or low-resource environments. Second, by combining PCA-based orientation estimation with template matching (TM), the method achieves accurate localization and angle estimation of puzzle pieces without using edge compatibility or color gradient information.

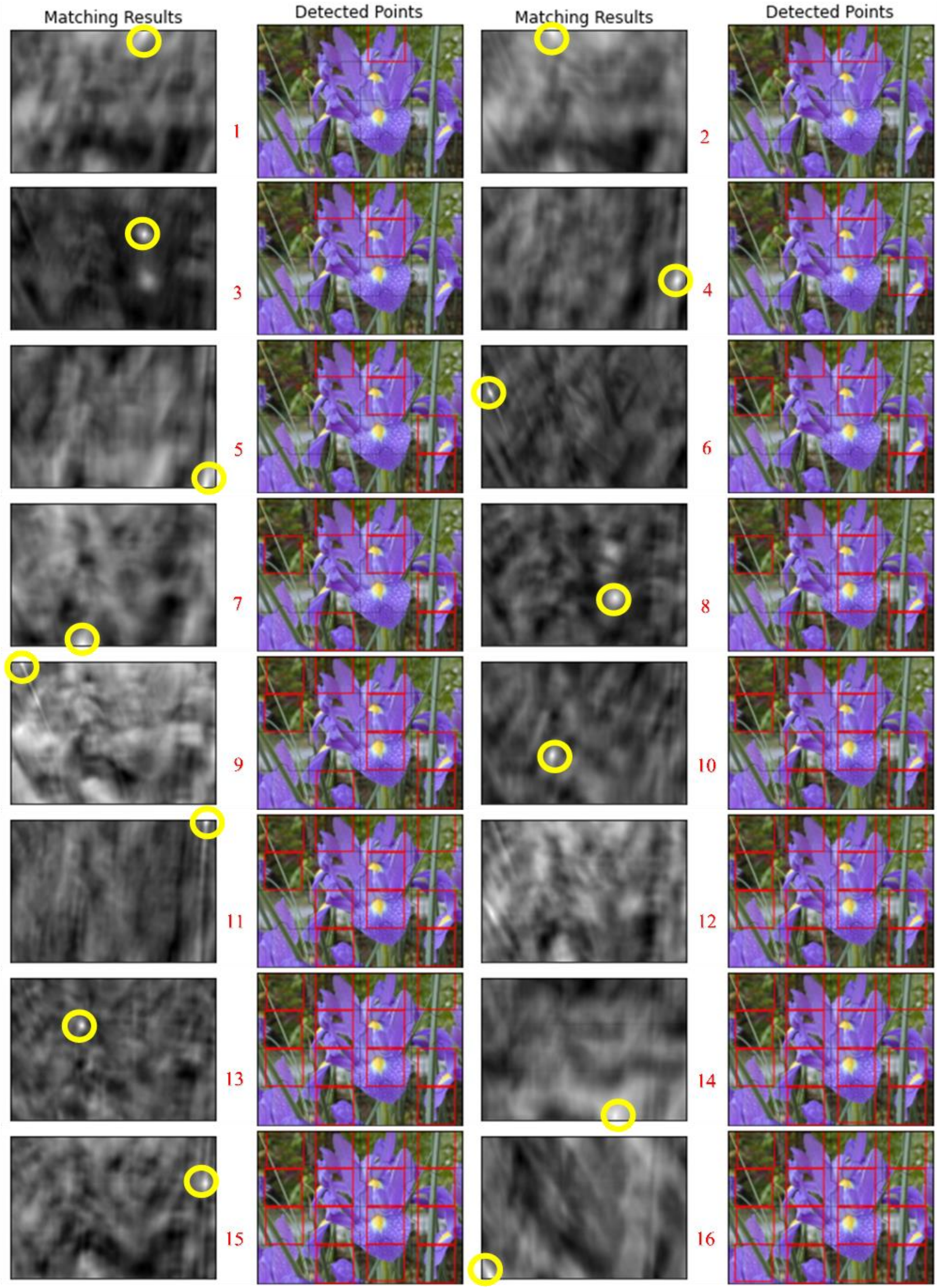


Figure 12. Matching the results of pieces in puzzle 2 step by step

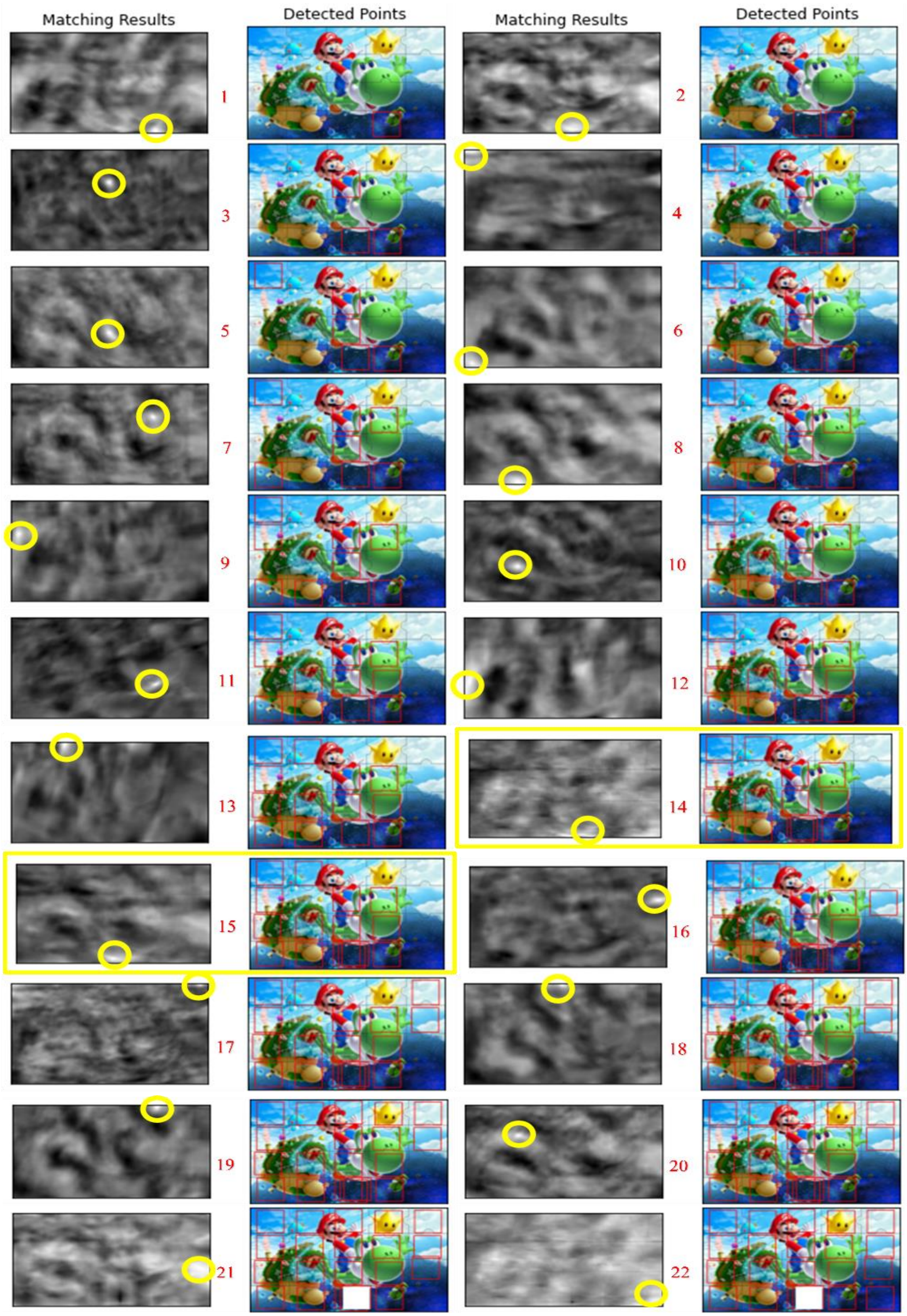


Figure 13. Matching the results of pieces in puzzle 3 step by step

However, there are also limitations. TM may yield false positives when puzzle pieces have similar patterns, especially in symmetric regions. Additionally, due to the ±2° error margin in PCA-based rotation estimation, small angular inaccuracies may occur. To overcome these, a repeated matching process with white-masked regions was proposed.

Compared to state-of-the-art methods that rely on learning-based or complex feature extraction pipelines, our method is simpler and more interpretable. While it may not outperform deep models on large-scale datasets, it is highly effective for moderate-sized puzzles and cases where pre-trained models are not available. The results from all three experiments demonstrated high accuracy, low average angular error, and successful placement of all pieces, even under challenging visual conditions.

Future improvements may include hybridizing the method with machine learning for better handling of ambiguous patterns and exploring alternative similarity measures instead of basic TM.

5 Conclusion

This study introduced two novel, template matching (TM)-based methods for solving jigsaw puzzles, both of which demonstrated high accuracy in aligning pieces of varying sizes and complexities. The first method employs a two-stage framework that integrates Principal Component Analysis (PCA) for orientation correction, followed by TM for positional alignment. In contrast, the second method combines orientation and position estimation in a single step, offering a faster yet still effective alternative.

In comparison to deep learning-based approaches that typically require extensive annotated datasets and significant training times, the proposed methods offer a lightweight, unsupervised, and computationally efficient solution. Furthermore, unlike traditional greedy or graph-based algorithms—which are often susceptible to local minima and combinatorial explosion—the PCA-based orientation correction significantly reduces placement ambiguity by aligning pieces according to their dominant structural direction.

Despite these strengths, the proposed methods have certain limitations. They assume that puzzle pieces are relatively intact and not heavily occluded or distorted. Consequently, their performance may degrade under conditions involving severe illumination changes, texture variations, or physical deformation of the puzzle components.

Nonetheless, the presented approach highlights the potential of combining simple yet effective preprocessing techniques with classical matching algorithms to achieve robust and interpretable puzzle reconstruction. This methodology holds promise for broader applications in computer vision, image analysis, and digital content reconstruction.

Future work may focus on improving robustness against real-world challenges by incorporating adaptive preprocessing techniques, exploring alternative similarity measures, or hybridizing the method with machine learning-based techniques. Such enhancements could further increase the scalability, accuracy, and applicability of the proposed approach in diverse visual domains.

Conflict of interest

The author declares that there is no conflict of interest.

Similarity rate (iThenticate): %11

References

[1] R. Li, S. Liu, G. Wang, G. Liu, and B. Zeng, JigsawGAN: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks. IEEE Transactions on Image Processing, 31, 513–524, 2021. https://doi.org/10.1109/TIP.2021.3120052

[2] I. Ahmad, S.-S. Hwang, and S. Shin, Determining jigsaw puzzle state from an image based on deep learning. Proceedings of 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 30–32, Jeju Island, Korea, 2022. https://doi.org/10.1109/ICAIIC54071.2022.9722672

[3] X. Song, X. Yang, J. Ren, R. Bai, and X. Jiang, Solving jigsaw puzzle of large eroded gaps using puzzlet discriminant network. Proceedings of ICASSP 2023–IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5, Rhodes Island, Greece, 2023. https://doi.org/10.1109/ICASSP49357.2023.10096300

[4] S. Markaki and C. Panagiotakis, Jigsaw puzzle solving techniques and applications: a survey. The Visual Computer, 39, 4405–4421, 2023. https://doi.org/10.1007/s00371-022-02598-9

[5] M. Makridis and N. Papamarkos, A new technique for solving puzzles. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 40, 789–797, 2009. https://doi.org/10.1109/TSMCB.2009.2029868

[6] H. Wolfson, E. Schonberg, A. Kalvin, and Y. Lamdan, Solving jigsaw puzzles by computer. Annals of Operations Research, 12, 51–64, 1988. https://doi.org/10.1007/BF02186360

[7] K. Zhang, W. Yu, M. Manhein, W. Waggenspack, and X. Li, 3D fragment reassembly using integrated template guidance and fracture-region matching. Proceedings of the IEEE International Conference on Computer Vision, pp. 2138–2146, Santiago, Chile, 2015.

[8] K. Son, J. Hays, and D.B. Cooper, Solving square jigsaw puzzles with loop constraints. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 383–390, Portland, USA, 2013.

[9] D.A. Kosiba, P.M. Devaux, S. Balasubramanian, T.L. Gandhi, and K. Kasturi, An automatic jigsaw puzzle solver. Proceedings of the 12th International Conference on Pattern Recognition (ICPR), pp. 616–618, Jerusalem, Israel, 1994. https://doi.org/10.1109/ICPR.1994.576377

[10] M.-M. Paumard, D. Picard, and H. Tabia, Jigsaw puzzle solving using local feature co-occurrences in deep neural networks. Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), pp. 1018–1022, Athens, Greece, 2018. https://doi.org/10.1109/ICIP.2018.8451094

[11] P. Nagaraj, V. Muneeswaran, K. Muthamil Sudar, S. Hammed, D.L. Lokesh, and V.S. Simha Reddy, An exemplary template matching techniques for counterfeit currency detection. Proceedings of the Second International Conference on Image Processing and Capsule Networks (ICIPCN), pp. 370–378, Madurai, India, 2021. https://doi.org/10.1007/978-3-030-84760-9_32

[12] A. Goshtasby, Template matching in rotated images. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-7(3), 338–344, 1985. https://doi.org/10.1109/TPAMI.1985.4767663

[13] M.G. Chung, M.M. Fleck, and D.A. Forsyth, Jigsaw puzzle solver using shape and color. Proceedings of the Fourth International Conference on Signal Processing (ICSP), pp. 877–880, Beijing, China, 1998. https://doi.org/10.1109/ICOSP.1998.770951

[14] A.C. Gallagher, Jigsaw puzzles with pieces of unknown orientation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 382–389, Providence, USA, 2012. https://doi.org/10.1109/CVPR.2012.6247699

[15] 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, 29, 3569–3581, 2020. https://doi.org/10.1109/TIP.2020.2963378

[16] D. Bridger, D. Danon, and A. Tal, Solving jigsaw puzzles with eroded boundaries. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3526–3535, Seattle, USA, 2020.

[17] D. Rika, D. Sholomon, E. David, and N.S. Netanyahu, TEN: Twin embedding networks for the jigsaw puzzle problem with eroded boundaries. arXiv preprint, arXiv:2203.06488, 2022. https://doi.org/10.48550/arXiv.2203.06488

[18] Y. Chen, X. Shen, Y. Liu, Q. Tao, and J.A. Suykens, Jigsaw-ViT: Learning jigsaw puzzles in vision transformer. Pattern Recognition Letters, 166, 53–60, 2023. https://doi.org/10.1016/j.patrec.2022.12.023

[19] M.-M. Paumard, H. Tabia, and D. Picard, Alphazzle: Jigsaw puzzle solver with deep Monte-Carlo tree search. arXiv preprint, arXiv:2302.00384, 2023. https://doi.org/10.48550/arXiv.2302.00384

[20] C. Doersch, A. Gupta, and A.A. Efros, Unsupervised visual representation learning by context prediction. Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1422–1430, Santiago, Chile, 2015.

[21] W. Yi and S. Marshall, Principal component analysis in application to object orientation. Geo-spatial Information Science, 3, 76–78, 2000. https://doi.org/10.1007/BF02826615

[22] G. Bradski and A. Kaehler, OpenCV. Dr. Dobb's Journal of Software Tools, 2000. https://www.drdobbs.com