Toward Solving a Puzzle of Fragmented Archaeological Textiles

Davit Gigilashvili,[1] Casper Fabian Gulbrandsen,[1] Ha Thu Nguyen,[1] Margrethe Havgar,[2] Marianne Vedeler,[2] and Jon Yngve Hardeberg[1]

[1)]*Norwegian University of Science and Technology; Gjøvik, Norway*

[2)]*Museum of Cultural History, University of Oslo; Oslo, Norway*

(*Electronic mail: davit.gigilashvili@ntnu.no)

Archaeological textiles can provide invaluable insight into the past. However, they are often highly fragmented, and a puzzle has to be solved to re-assemble the object and recover the original motifs. Unlike common jigsaw puzzles, the archaeological fragments are highly damaged, and no correct solution to the puzzle is known. While automatic puzzle solving has fascinated computer scientists for a long time, this work is one of the first attempts to apply modern machine learning solutions to archaeological textile re-assembly. First and foremost, it is important to know which fragments belong to the same object. Therefore, features were extracted from digital images of the textile fragments using color statistics, classical texture descriptors, as well as deep learning methods. These features were used to conduct clustering and identify similar fragments. Four different case studies with increasing complexity are discussed in this article: from well-preserved textiles with available ground truth to an actual open problem of Oseberg archaeological tapestry with unknown solution. This work revealed significant knowledge gaps in current machine learning that helps us to outline a future avenue toward more specialized application-specific models.

I. INTRODUCTION

Archaeological artifacts tell interesting stories and provide invaluable insight into the past of the humankind. However, they are often found in a fragmentary state. This is especially true for archaeological textiles that are vulnerable to degradation and decomposition. A vivid example is the Viking Age tapestry collection from the Oseberg burial, Norway[1]. The tapestries depict the scenes that are hard to read and understand due to high degree of fragmentation. Therefore, solving a puzzle is needed to put the adjacent pieces together, re-assemble the object, and recover the motifs. Previously, this has been done manually by human experts, which is a time consuming and tedious process[2,3]. Besides, the fragments are usually very fragile, and physical interaction with them should be kept to a minimum for conservation considerations.

Re-assembly of the archaeological objects bears some similarities to jigsaw puzzle solving – a popular pastime activity. However, while a reference image exists as a correct solution for common jigsaw puzzles, no ground truth is usually available in real-world archaeological problems (see Fig. 1). Furthermore, often it is not obvious whether the fragments belong to the same object or not, which introduces an additional complexity.

The machine learning literature has addressed puzzle solving for artificially fragmented artwork imagery[4–6] or 3D archaeological artifacts[7–10], such as pottery fragments. However, the literature on a complex scenario of archaeological tapestry is very limited. The archaeological textile fragments are highly degraded and irregularly shaped, and the number of different objects before fragmentation is unknown. (cf. Fig. 1). While the state-of-the-art machine learning techniques face substantial challenges in relatively simpler scenarios, where the fragments are not degraded, have regular shapes, only small part of the fragments are missing, and all belong to the same object[2,4], we believe that fully automatic puzzle solving for archaeological textiles, where

FIG. 1. Jigsaw puzzle solving is a popular pastime activity (left). They usually include regularly shaped pieces with matching contours and a ground truth solution as a reference. Puzzle solving in actual archaeological applications (such as the Oseberg tapestry on the right) is a more challenging problem due to irregular shape of the fragments, their degradation, and the lack of ground truth. Reproduced from[12].

a machine automatically determines fragments' spatial location is currently infeasible. Instead, this work proposes a framework, where image processing and machine learning techniques are utilized to extract features from fragments' images and conduct clustering to identify fragments that are similar and hence, highly likely to belong to the same object. Based on this information, a human expert can finalize the puzzle solving on a digital canvas with substantially less effort. In our previous communication[11], the preliminary results of the clustering were reported. The work attempted to identify which fragments belonged to the same object. This article follows up and expands the previous communication by new results obtained in one of the authors' recent work[12].

II. RELATED WORKS

Automatization of puzzle solving has fascinated computer scientists for nearly six decades. The seminal work by Freeman and Garder[13] was a first one to propose such a solution. Their puzzle was composed of 9 gray pieces, and the solution relied on contour matching information. Substantial progress has been achieved since then, and the recent works also utilize color and semantic regularities present in the pieces[14–16]. Different machine learning methods have proved

efficient in solving puzzles of natural images[17–19]. For example, a Genetic Algorithm used by Sholomon *et al.*[19] can rely on pieces as small as 7×7 pixels and has been demonstrated to solve a 432-piece puzzle in 2 minutes. Another example is a Growing Consensus algorithm used by Son *et al.*[16]. Authors' objective was to solve puzzles with very high number of pieces, such as the one with 22 834 pieces that was solved in approximately 13 hours. The authors claim that this is the largest puzzle solved in a fully automatic fashion.

Paumard *et al.*[20] used deep learning to solve 3×3 puzzles of painting images. They used VGG-Net[21] for feature extraction and left 48 pixel-wide gaps among the fragments to simulate erosion. In their solution, ground truth information about the central fragment was known, which was utilized for classification, whether or not the fragment belonged to the same painting, as well as for determination of the position relative to the central piece. Many pieces, especially the homogeneous ones, were often misplaced. The authors followed up with several works using different deep learning implementations[4–6] and incorporation of the semantic content[15].

Puzzle solving has practical implications in many domains. A good example of this is re-assembly of shredded documents, for which, different solutions have been offered that used information such as paper color, font style, line spacing, and semantic content[22,23]. Puzzle solving plays an especially important role in cultural heritage, where re-assembly of fragmented heritage objects is needed. Different machine learning solutions have been proposed for pottery[7,24], mosaic tile panels[8,9,25], and frescoes[26], utilizing both 2D[24] as well as 3D[7] information. The works that use 3D information, usually rely on surface normal directions in addition to contour and color consistency[8,9,26].

Unlike rigid objects, such as pottery, textiles are flexible and prone to significant deformations, which especially complicates the matching and re-assembly process[27]. Few works have addressed virtual reconstruction of heritage textiles and are mostly limited to inpainting - reconstruction of

damaged parts by utilizing the information in the preserved regions[28–30]. However, the extent of degradation and fragmentation in archaeological textiles, such as the Oseberg tapestry, is often so large that application of inpainting techniques may not be feasible due to the lack of enough structural trends.

To the best of our knowledge, the first work that addressed puzzle solving specifically for archaeological textiles was our previous communication[11]. In that work, the fragments were split into square patches, features were extracted using classical texture descriptors as well as deep learning techniques and clustering was conducted to identify matching ones. The result was inconclusive. This was followed up by a recent work of Gulbrandsen[12] (one of the co-authors of this article), where different feature extraction and clustering techniques were explored. Furthermore, Gulbrandsen[12] developed a software for virtual puzzle solving on a virtual canvas. This article extends our previous communication with this new knowledge and summarizes the recent efforts toward solving a puzzle of fragmented archaeological textiles. A comprehensive literature review on this topic can be found in the work by Gigilashvili *et al.*[2].

III. METHODOLOGY

The contribution of this article is as follows:

- It investigates different feature extraction methods both from classical texture descriptors to the cutting-edge deep learning techniques to create a feature vector for each fragment based on its color image.
- Clustering is conducted using different clustering algorithms to identify potentially similar fragments.
- A novel dataset of well-preserved household textiles is created. The textiles in this dataset

were artificially fragmented by us to test the approach on it as a baseline study.

- The approach is also tested on well-preserved historical textile artifacts.
- Afterward, the approach is tested on highly degraded Oseberg tapestry imagery.
- Apart from that, a novel software with a digital canvas is introduced that can be used for digital puzzle re-assembly.
- Finally, the limitations of the current computational techniques are discussed and the knowledge gaps that need to be addressed in the future are outlined.

The work can be divided in four experiments with increasing degrees of complexity. For each experiment, a general approach can be divided into the following steps:

1. Data collection – either by image acquisition, or downloading existing datasets.
2. Pre-processing – optimizing the images for analysis: such as downsampling high-resolution images, sharpening and enhancement, segmentation to eliminate noisy regions, and dataset augmentation (e.g. rotation).
3. Feature extraction – using different texture descriptors, color moments, or pre-trained Convolutional Neural Networks (CNNs) to create a feature vector associated with each fragment.
4. Clustering – using different algorithms to group fragments based on the similarity of their feature vectors.
5. Evaluation – evaluating the clustering results, either based on the ground truth (when known) or the hypotheses of the human experts (when no ground truth is available).

FIG. 2. Manually cut textile fragments were photographed under controlled D65 illumination on a neutral gray background in a viewing booth, where the angle and the object-to-camera distance were fixed (to approximately 60 cm), and the D65 illuminant of the booth was the only source of light in the scene. Reproduced from[12].

A. Case study 1: A baseline study using household textiles

1. Acquisition and Pre-processing

Well-preserved household textiles were purchased and manually cut with scissors into 6 or 9 irregularly shaped fragments. Then high-resolution photographs under controlled D65 illumination in a viewing booth (see Fig. 2 and 3) were acquired. The images were downsampled and segmented using MATLAB's Image Segmenter app and saved in the PNG format where the background was transparent. The textiles are in a good condition, all fragments are present, they have

FIG. 3. Examples of household textile fragments after segmentation. Reproduced from[11].

simple structure, and the ground truth information is available, which enables reliable evaluation. This helped us establish a baseline performance of the machine learning algorithms.

2. Feature Extraction

A broad range of features to capture color and texture information was used, as well as other visual patterns and deep features, as follows:

- Color-based features:
  - Color histograms
  - Color moments (mean, standard deviation, skew)
  - Color coherence vectors

- Texture-based features:
  - Local Binary Patterns (LBP)
  - Opponent Color LBP
- Deep learning-based features:
  - Pre-trained convolutional neural networks such as VGG, ResNet, and Inception

Color is a fundamental attribute of appearance, which is used by humans and machines alike[31]. Color histograms can provide an interesting insight into the overall distribution of colors in each fragment, and hence, help us identify similar fragments[32].

Color moments, such as mean, standard deviation, and skewness provide additional information about object's color properties, and are calculated as[33]:

$$E_i = \sum_{j=1}^{N} \frac{1}{N} p_{ij} \tag{1}$$

where E_i is a mean of specific property (e.g. color), p_{ij} is the value at a given pixel in the i-th row and j-th column of the image, and N is the total number of pixels.

$$\sigma_i = \sqrt{\left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^2\right)} \tag{2}$$

where σ_i is the deviation, E_i is the mean as in Eq. 1, p_{ij} is the value at the i-th row and j-th column, and N is the total number of pixels;

$$s_i = \sqrt[3]{\left(\frac{1}{N}\sum_{j=1}^{N}(p_{ij} - E_i)^3\right)} \tag{3}$$

where, s_i is skewness, which is a measure of the asymmetry. These calculations are done separately for each of the RGB channels.

Images with identical histogram may look substantially different, based on how the pixels are distributed in space. Color coherence vectors (CCVs) take into account not only the number of pixels with specific colors captured by spatially-blind histogram, but also their spatial coherence – i.e. the degree to which pixels of that color are members of large similarly-colored regions[34]. For instance, Fig. 4 illustrates the case, where histograms of two textures are identical, but they look different, which can be captured with CCVs instead of histogram statistics. A Jupyter notebook was used to add or remove different features to optimize the result.

Apart from color features, classical texture descriptors from computer vision, such as Local Binary Patterns (LBP) or Opponent Color Local Binary Patterns (OCLBP) were used. LBP captures local textures by creating binary codes that capture whether the intensity of a neighboring pixel is larger or smaller than that of a central pixel (see Fig. 5). It is applied to grayscale images and have proved efficient in computer vision applications[35,36]. OCLBP extends LBP from grayscale to color, and captures both color and texture information, by analyzing and then combining the patterns in R, G, and B channels [37,38].

Finally, pre-trained deep learning models that have revolutionized image classification and other computer vision problems were also used for feature extraction. Pre-trained models such

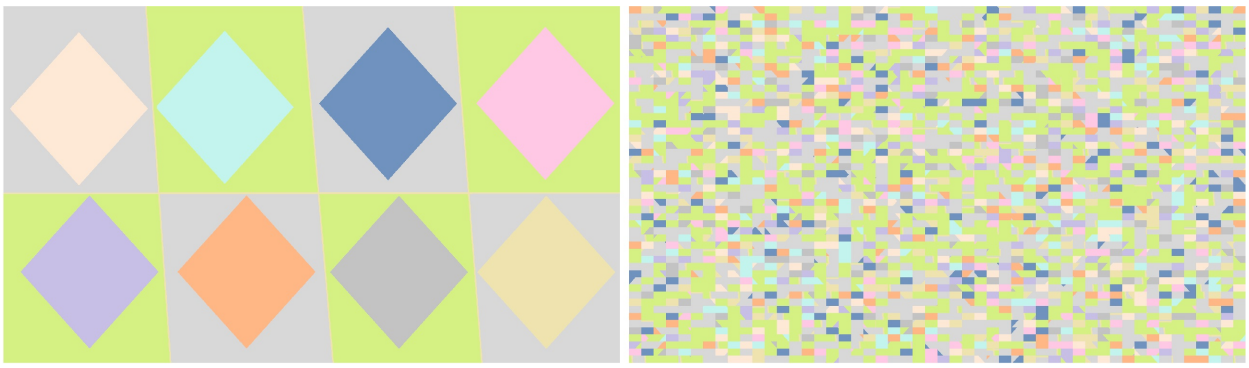


FIG. 4. The image on the right is a scrambled version of the image on the left. Their histograms are identical. However, their textures look substantially different, which can be captured by CCVs.

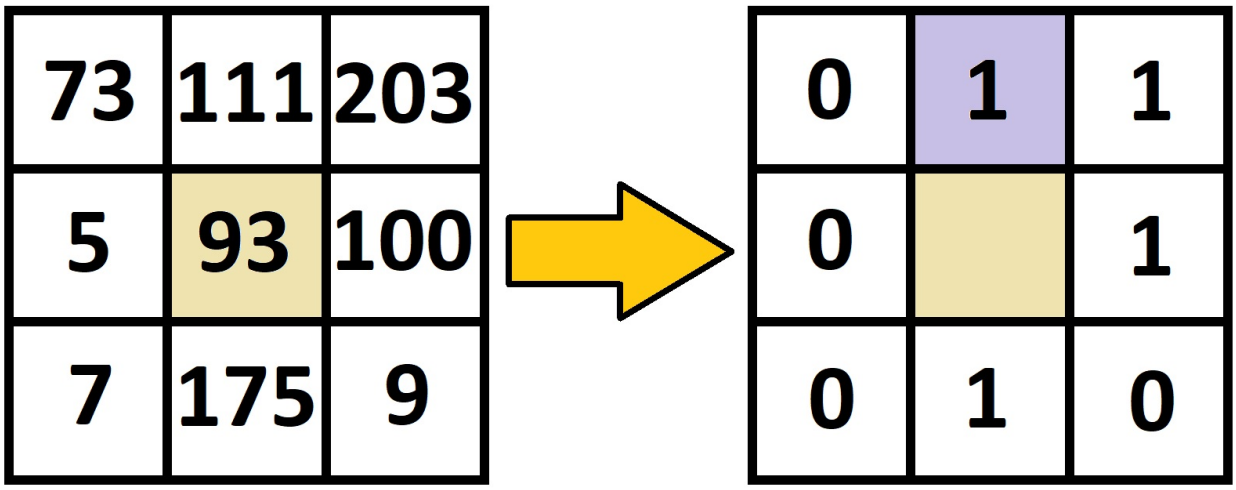


FIG. 5. The example of calculating an LBP. The value of the central pixel (beige cell) is compared with all of its eight neighbors. If the pixel value of the neighbor is higher, 1 is assigned to the respective cell in the matrix on the right; if it is lower, then 0. The binary code is generated by reading the binary values in the matrix on the right either clockwise, or counterclockwise. If the code is read counterclockwise from the lilac cell, the LBP code of the central cell will be 10001011. For convenience, the codes are converted to decimal, which in this case is 139. LBP code value for a given cell is 139. The statistics of all LBP codes can be then summarized with a histogram.

as VGG and ResNet have been used for feature extraction for classification purposes, since they are trained on very large datasets, such as ImageNet, and capture broad range of visual features from simple edges to more complex high-level attributes[39]. In this case, VGG16 and VGG19 deep convolutional neural networks were considered, where the number signifies the number of convolutional layers. It is a popular architecture for image recognition tasks[39].

3. Clustering

K-means and hierarchical clustering algorithms were used to group the fragments. K-means separates the data in a pre-defined number of clusters[40,41]. It randomly selects initial cluster cen-

troids and iteratively refines them by minimizing intra-cluster differences until convergence or for a predefined number of iterations. It is efficient, but depends on the number of clusters as well as the initialization of the centroids. Hierarchical clustering, on the other hand, does not need specifying the number of clusters in advance[42]. It builds a tree-like structure by successively merging or splitting the branches based on a distance measure[43]. It may deploy either agglomerative (bottom up) or divisive (top down) approach.

4. Evaluation

The results were compared against the ground truth. Besides, the results were evaluated using the Silhouette score, which is a measure of how similar an object is to its cluster compared to other clusters[44]. The silhouette score ranges from -1 to 1 with higher values indicating that the object matches well with its cluster.

B. Case study 2: A study on well-preserved heritage textiles

Unlike household textiles that usually have simple patterns, our approach was tested on the images of the heritage textiles. It is important to highlight that they are not archaeological textiles and are hence, well-preserved despite their age. The photographs of the Tingelstad, Överhogdal, and the Gudbrandsdalen cloth were used. They were virtually split into rectangular grids (see Fig. 6). The gaps were deliberately left to simulate degradation as it was done by Paumard[14]. Unlike Case study 1, they had a regular shape, which in part simplifies the problem. Ground truth, as in case of Case Study 1, was also available.

Feature extraction and clustering algorithms were the same as for Case Study 1. The results were compared against the ground truth. Besides, the results were evaluated using the Silhouette

FIG. 6. Well-preserved Tingelstad heritage textile was digitally cut into 3×3. The gaps were left deliberately to simulate degradation. Reproduced from[12].

score.

C. Case study 3: Archaeological textiles – part A: classical texture descriptors and AlexNet

This case study was reported in the previous communication[11]. In this case, highly fragmented and degraded Oseberg tapestry[1] imagery was used, which was obtained from the Museum of Cultural History. This is an open in-the-wild problem, and no ground truth is available for this dataset. The overall workflow is illustrated in Fig. 7.

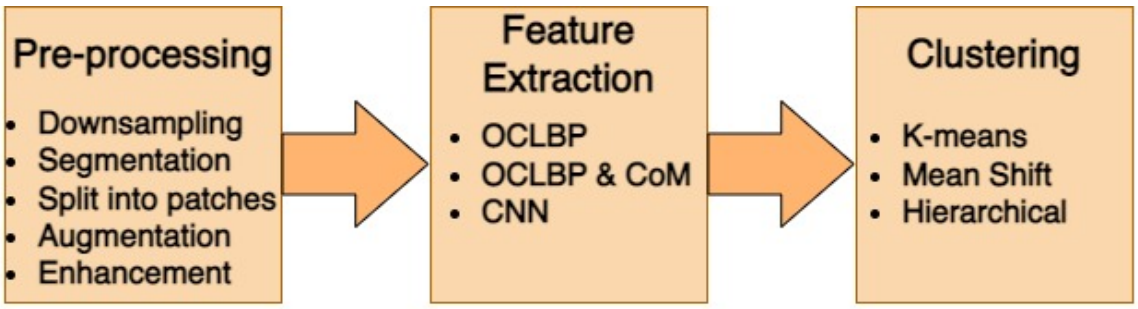
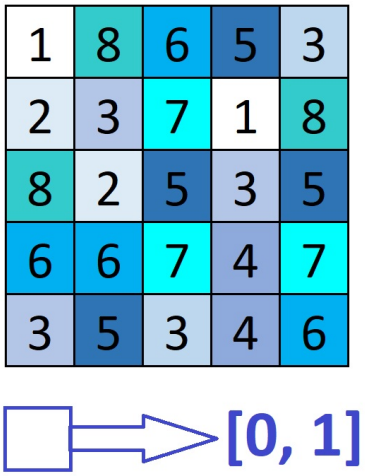


FIG. 7. The workflow in Case Study 3. Reproduced from[11].

FIG. 8. Examples of the patches used in Case Study 3. Reproduced from[11].

1. Acquisition and Pre-Processing

First of all, the ultra-high-resolution color photographs obtained from the Museum of Cultural History were downsampled due to memory limitations and computational efficiency. Afterward, the fragments were segmented, and each fragment was split into smaller 200×200 pixel patches, shown in Fig. 8. This had two objectives: first, to increase the number of images, since there are few fragments in the Oseberg dataset; second, to create a ground truth for evaluation purposes (it is known which patches come from the same fragment). The fragments with high degree of noise were discarded. For smaller fragments with less than 60 patches, the dataset was augmented by rotation to 90°, 180°, and 270°. If the patches were blurry, image sharpening was used. In total, 6650 200×200 px patches corresponding to 77 original Oseberg tapestry fragments were generated.



	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	2
2	0	0	1	0	1	0	0	0
3	0	0	0	1	2	0	1	0
4	0	0	0	0	0	1	1	0
5	0	0	3	0	0	0	0	0
6	0	0	0	0	1	1	1	0
7	1	0	0	1	0	0	0	0
8	0	1	0	0	0	1	0	0

FIG. 9. Co-occurrence Matrices indicate how often pixels with given values at a given spatial distance and direction (different offsets) occur. For the part of the image with pixel values shown on the left, a Co-occurrence Matrix for the [0, 1] offset is given on the right. For instance, 2 at the 1st row and 8th column (right matrix) means that there are two instances where 1 and 8 are horizontally adjacent and 1 is on the left. 5 and 3 are horizontally adjacent three times, and so on. Similar matrices can be constructed for different offsets – different directions (vertical adjacency, diagonal adjacency) and distances.

2. Feature Extraction

Three different methods for feature extraction were used: classical feature descriptors, such as OCLBP[37] and Co-occurrence Matrices (CoM)[45]. CoM captures the relative positions of the pixels and represents them as a matrix of probabilities of co-occurrence of certain pixels within a certain distance (see Fig. 9). As proposed in[46], OCLBP features were combined with CoM features. Apart from this, features using pre-trained AlexNet convolutional neural networks (using ImageNet weights)[47] were extracted. This feature vector has 507 elements.

3. Clustering

After extracting the features, they were fed into clustering algorithms to group the patches with similar textures together. K-means[40], Mean-Shift[48], and Agglomerative Hierarchical clustering[42] were used. Similarly to K-means, Mean-Shift is also a centroid-based method, but it determines the optimal number of clusters without the user input. The user, however, defines the radius (the "bandwidth") parameter. For each data point, a local mean is found within this pre-defined radius. Given point is shifted to this local mean, and then the new mean is found toward which the points should be shifted. The process continues iteratively, and the points shifted toward the same final centroid are concluded to be in the same cluster.

4. Evaluation

The accuracy of the clustering result was measured as follows: the number of clusters k was varied from 2 to 77 for patches that come from 77 fragment images, which are considered the pseudo-classes. Because this is an unsupervised problem, all 6650 patches were used to fit the clusters. At each class j, the number of patches that are assigned to each cluster (denoted as c) is calculated, which is represented by $p_{c,j}$. The accuracy for each class j is then calculated as:

$$acc_j = \frac{max(p_{c,j})}{N_i} \tag{4}$$

where $c \in [1;k]$, and N_j is the number of patches in a given class (from the same fragment). Finally, mean accuracy among all 77 classes and its standard deviation were calculated. It is worth highlighting that this method captures false negatives (patches that are known to be from the same fragment and end up in different clusters), and it does not capture false positives (patches from different fragments end up in the same cluster). This decision is intentional, because our objective is not clustering patches to 77 original classes. Any false positive may, in fact, indicate that the

fragments that those patches come from belonged to the same original.

The results were also evaluated by the archaeologists who work with the Oseberg Tapestry. Although the ground truth is not known, the professionals hypothesize which fragments may belong together based on the motif and weaving technique analysis.

D. Case study 4: Archaeological textiles – part B: Color Moments and VGG features

The same ultra-high-resolution Oseberg Tapestry imagery was used as in Case study 3. The original images were as large as 15000×12000 pixels. Therefore, they were cropped into 1200×1200 pixel regions that contained informative parts with less noise and degradation. Afterward, the image was split into four patches (2×2) to create the ground truth. The process is shown in Fig. 10. Feature extraction and clustering algorithms were the same as for Case Study 1.

IV. RESULTS

This section presents the clustering results. Each subsection presents the results for each case study with visual illustrations and numerical evaluations. Overall observations are discussed in the subsequent section.

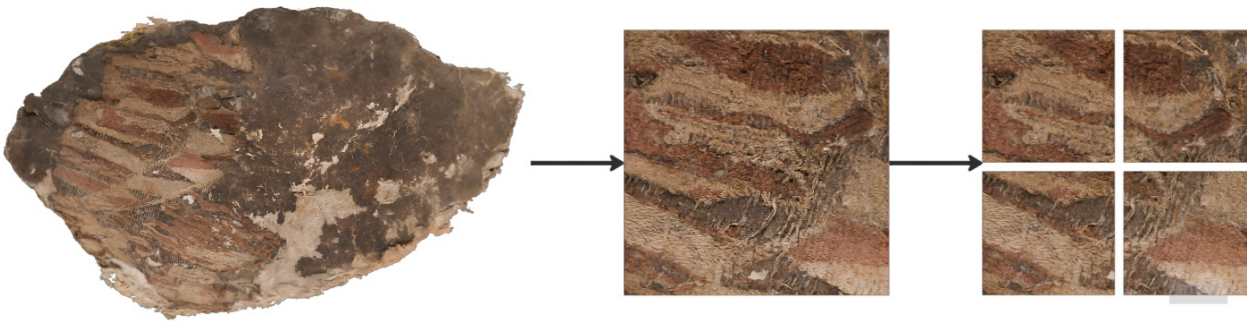


FIG. 10. Pre-processing example of the Oseberg tapestry images in Case study 4. Reproduced from[12].

A. Case study 1: A baseline study using household textiles

One textile was split into 6 fragments, while the other two into 9 pieces each – totaling to 24 pieces in total. Two textiles were lighter in color, while the third one was relatively dark. Clustering was first run on these 24 fragments. Afterward, each fragment was split into 4 sub-fragments, increasing the dataset to 96 fragments. For both cases, color moments, color histogram, and LBP essentially separated them into light and dark, placing many fragments from two lighter textiles in the same cluster (see Fig. 11). On the other hand, VGG19 provided perfect (without mistake) performance for 24 fragments with hierarchical clustering (see Fig. 12), and making just 1 mistake with k-means clustering. However, when the number of the fragments was increased, the performance deteriorated, and it placed all lighter fragments in one cluster, while wrongly distributing the darker ones into two clusters (see Fig. 13).

The Silhouette scores are summarized in Table I. The scores show no difference between the clustering methods. However, the difference is apparent among the features that were used for clustering. The best results were produced by LBP and color moments. The scores are lower for histograms, and they drop substantially when VGG19 is used as a feature extractor, both alone, as well as in combination with other descriptors. This is consistent with the results shown in Fig. 13.

B. Case study 2: A study on well-preserved heritage textiles

When using color histogram and LBP features for K-means clustering, the results were perfect, without mistake, as illustrated in Fig. 14. Interestingly, when VGG19 features were used instead, several false positives were detected. For example, one Tingelstad cloth was grouped with the Överhogdal images (Fig. 15). The Silhouette scores are given in Table II. The table shows that

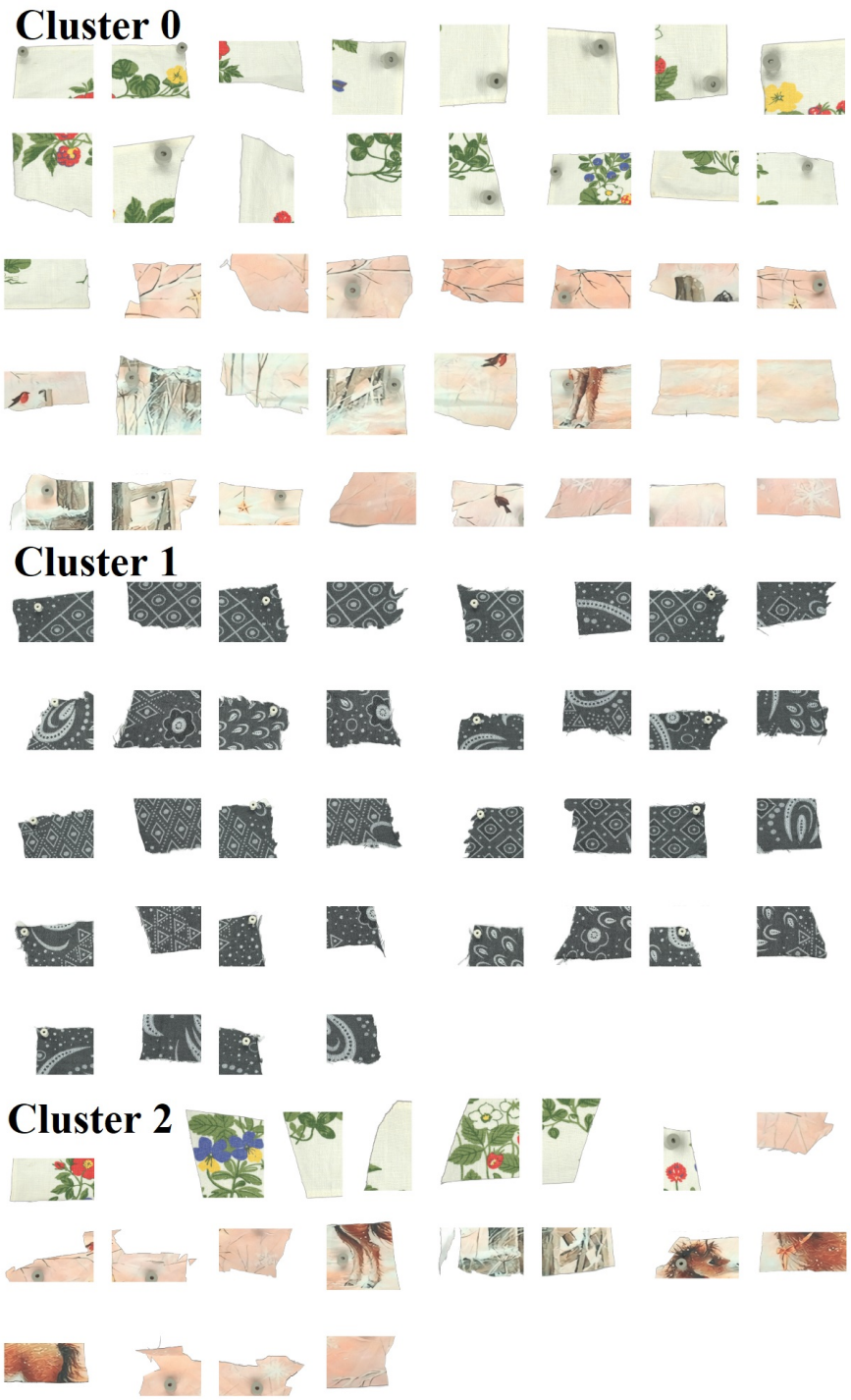


FIG. 11. Clustering results using 96 fragments. Features were extracted using color moments, color histogram, and LBP. Lighter fragments are mixed up and grouped together. Reproduced from[12].

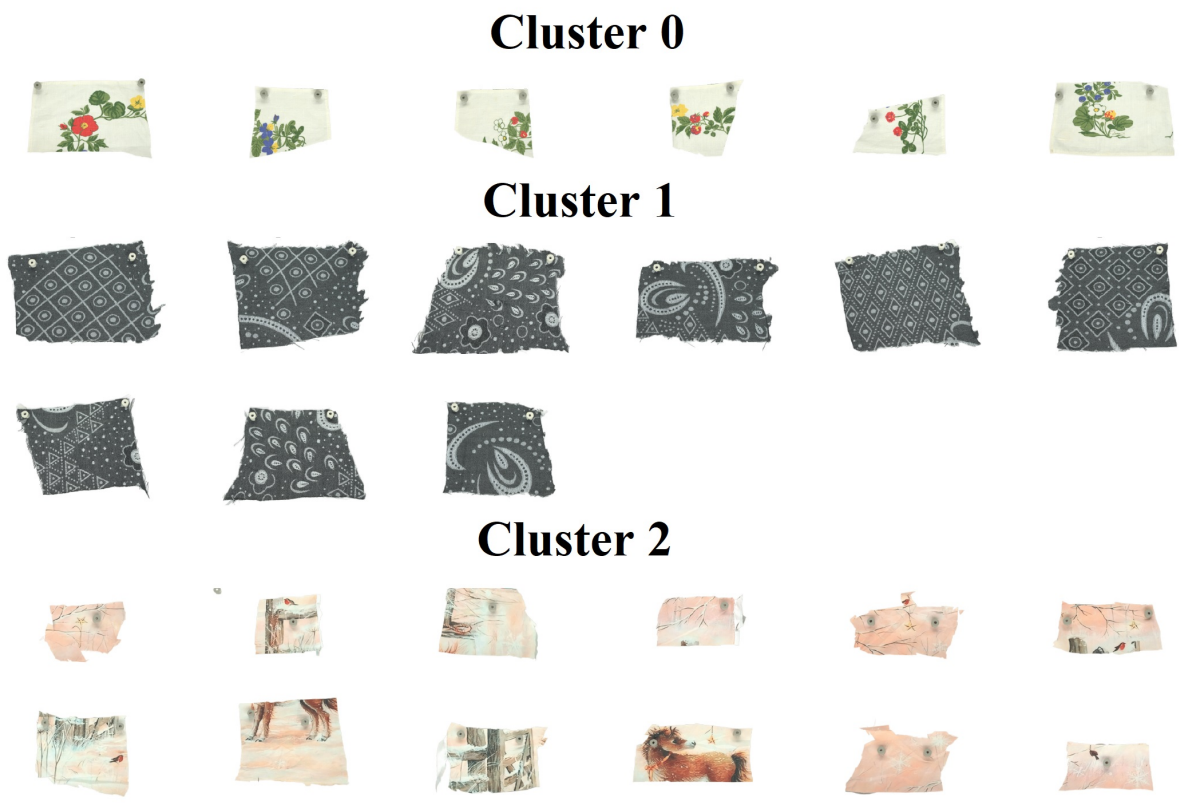


FIG. 12. Clustering results for 24 fragments using the VGG19 model as a feature extractor and hierarchical clustering, where the clustering was 100% accurate. K-means performs similarly with only 1 mistake.

unlike Case Study 1 (cf. Table I), the clustering method slightly affects the Silhouette scores. Hierarchical clustering led to higher Silhouette scores when LBP and VGG19 features were used, while it was outperformed by K-means for other features. The best results were produced for color histograms and LBP. It is worth mentioning that the Silhouette score is below 0.6 even when the clustering result was 100% accurate. Generally, color statistics and classical texture classifiers produced higher Silhouette scores (above 0.5) that dropped to near 0 when CNNs, such as VGG19 and AlexNet were used. Interestingly, using all features at the same time does not improve the results.

FIG. 13. Clustering results using 96 fragments. Features were extracted using VGG19. Lighter fragments are grouped together, while the dark ones are distributed in two clusters. Reproduced from[12].

TABLE I. The resulting Silhouette scores for the household textiles.

Features	**Clustering**	**Silhouette Score**
LBP	K-means	0.689
LBP	Hierarchical	0.689
Color moments	K-means	0.686
Color moments	Hierarchical	0.686
Histogram	K-means	0.529
Histogram	Hierarchical	0.529
LBP, Histogram, Moments, VGG19	K-means	0.106
VGG19	K-means	0.106
VGG19	Hierarchical	0.106

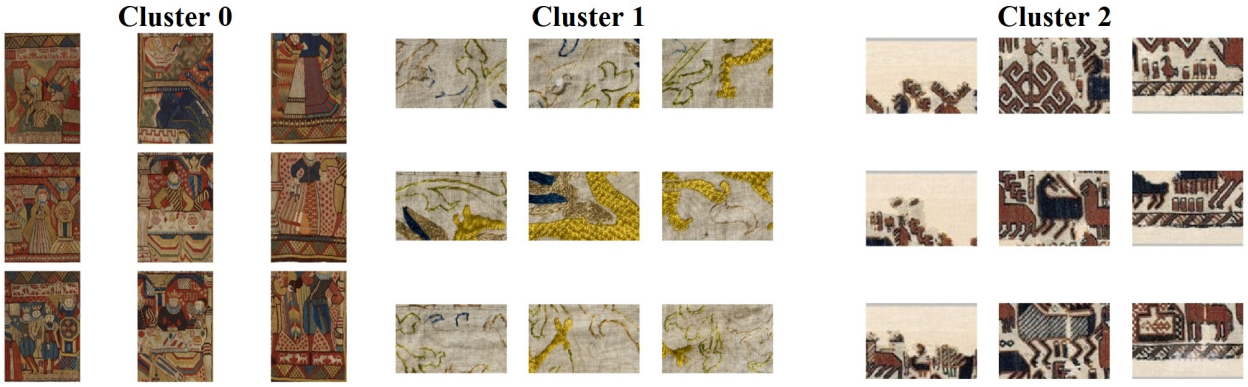


FIG. 14. When using color histogram and LBP, Gudbrandsdalen (left), Tingelstad (middle), and Överhogdal (right) textile were clustered correctly. Reproduced from[12].

FIG. 15. One Tingelstad fragment was incorrectly clustered with Överhogdal textile fragments for VGG19. Reproduced from[12].

C. Case study 3: Archaeological textiles – part A: classical texture descriptors and AlexNet

The results for OCLBP and its combination with CoM are illustrated in Fig. 16. All clustering algorithms demonstrated similar accuracy, as well as high correlation among their cluster compositions. Mean-Shift and Hierarchical clustering determined the optimal number of clusters to be two. K-means clustering with 2 to 77 clusters was tested, and the accuracy was the highest when it was 2. However, considering that the accuracy criterion does not penalize for false positives, this result need to be taken with great care. With this method, it is possible to evaluate whether patches from the same fragment end up in the same cluster, but it is unclear whether placing the patches from different fragments in a same cluster is necessarily wrong – the two fragments could actually be part of the same object. Therefore, if all patches are put in the same cluster, the accuracy will

TABLE II. The resulting Silhouette scores for well-preserved historical textiles.

Features	**Clustering Method**	**Silhouette Score**
Histogram	K-means	0.597
LBP	Hierarchical	0.597
Moments	K-means	0.509
LBP, Histogram, Moments	K-means	0.509
LBP	K-means	0.507
Moments	Hierarchical	0.504
LBP, Histogram, Moments	Hierarchical	0.504
Histogram	Hierarchical	0.503
All	Hierarchical	0.101
VGG19	Hierarchical	0.101
AlexNet	K-means	0.099
AlexNet	Hierarchical	0.093
All	K-means	0.047
VGG19	K-means	0.016

be 100% (all patches from the same fragment will be together).

Fig. 17 illustrates that the fragments in one cluster seem to have thicker threads and lower spatial frequency texture, while in the second cluster, the patches with smaller thread size and higher spatial frequency variation are grouped. This could be an indication that thread thickness is an useful cue to fragment similarity, but as the small patches do not capture global motifs, it can also be misleading if two different artworks were weaved with a similar technique. This result was evaluated by the archaeologists, who pointed out many potential false positives, as they

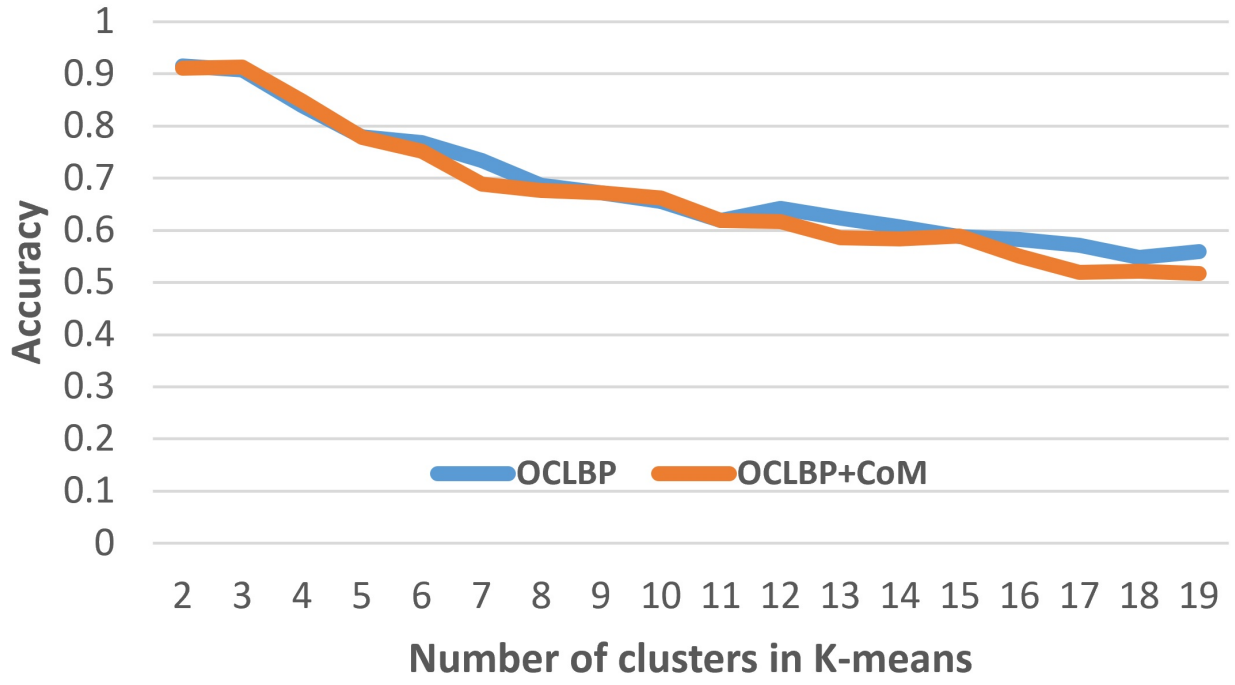


FIG. 16. The accuracy as a function of number of clusters in K-means. The blue curve corresponds to OCLBP features, and the orange curve corresponds to a combination of OCLBP+CoM. Reproduced from[11].

FIG. 17. The left and right pair of images were grouped in different clusters, respectively; while the left two patches have thicker threads and relatively low spatial frequency texture, the opposite is true for the right ones. Reproduced from[11].

hypothesize that the number of clusters should be larger.

AlexNet feature vectors were 507-dimensional. Due to a low number of samples, K-means failed due to the curse of dimensionality. Hierarchical clustering detected two clusters, where one

was substantially larger than the other. First cluster contained most of the patches (6450), while the second one had just 200 extremely degraded homogeneous patches with barely visible variation.

D. Case study 4: Archaeological textiles – part B: Color Moments and VGG features

Since no ground truth is available, it was first decided to compare the clustering results based on archaeologists' hypotheses. Fragments from 5 hypothesized clusters were selected that archaeologists considered similar based on their motifs and weaving techniques. The distribution of these fragments using different feature extraction techniques is shown in Tables III and IV. The name of each fragment starts with a letter (A-E) signifying hypothesized cluster, while the numbers are their unique identifiers. In other words, if the clustering results are perfectly aligned with the hypotheses, all fragments whose name starts with "A" should be clustered together; those with "B" – together, and so on, respectively. Tables III and IV demonstrate that there is a substantial misalignment between the clustering result and hypothesized clusters. Full fragment images contain a lot of noise. Therefore, the same procedure was repeated with 1200×1200 px close-ups, as well as four smaller patches produced from each of them (2×2, as shown in Fig. 10). The results do not improve substantially, and in some cases, the patches from the same fragment end up in different clusters. A closer look at the clustering results using these fine-scale patches (see Fig. 18) illustrates that the grouping is primarily based on a color shade.

V. DISCUSSION

This article reported the first step toward puzzle solving of highly degraded and fragmented archaeological textiles. This work reveals the significant knowledge gaps in the state-of-the-art machine learning. Color statistics, classical texture descriptors, as well as the recent deep learning

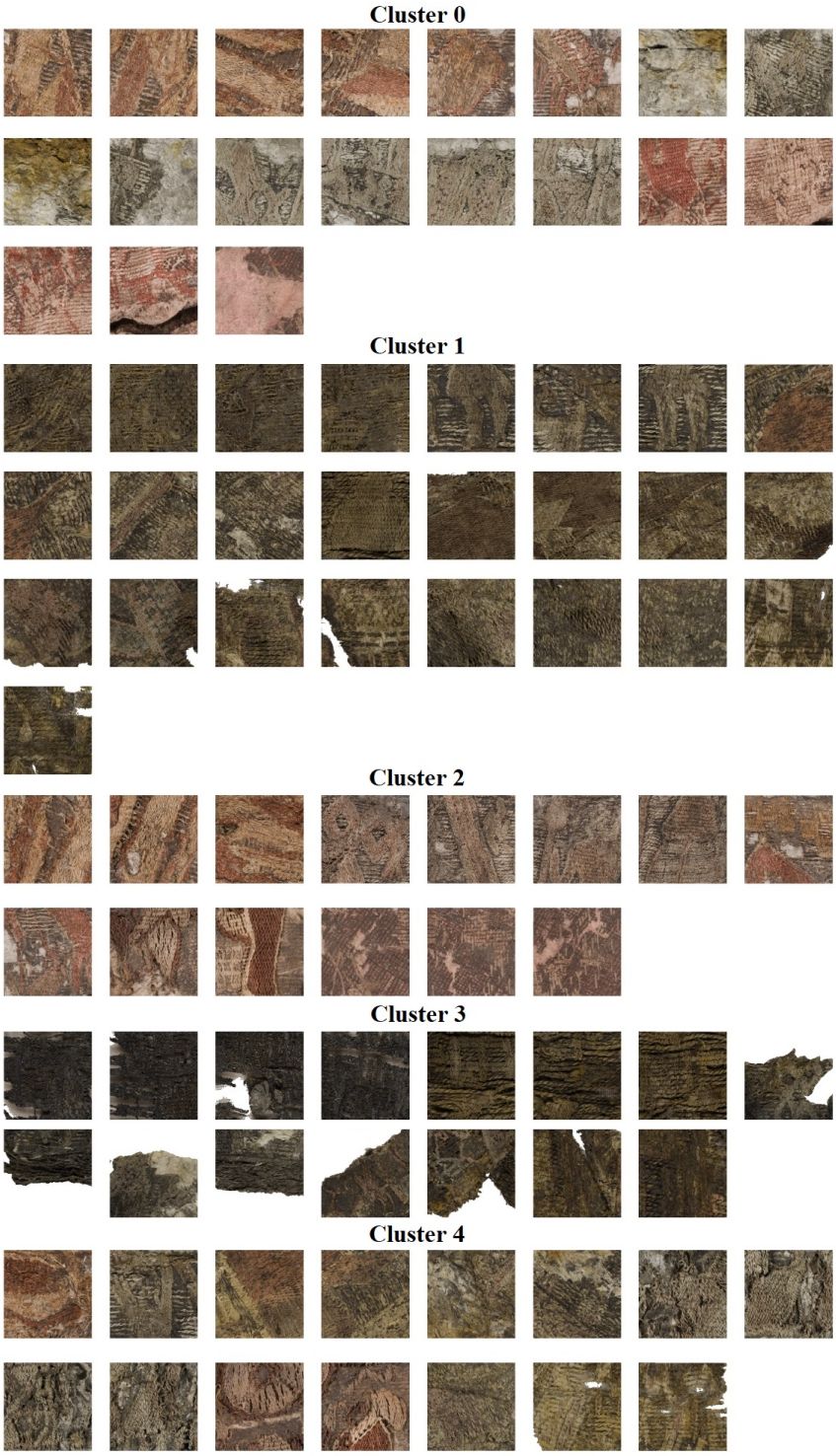


FIG. 18. The results from the clustering of 2×2 ROI-focused Oseberg textiles using color and texture features and hierarchical clustering. Reproduced from[12].

TABLE III. The initial clustering results when using color and texture feature extractors with hierarchical clustering. The result was the same for K-means.

Cluster	**Fragments**
0	A_008, B_010, B_090, D_067, D_071
1	B_066, B_082, B_086, B_088, D_072, E_102, E_103, E_104
2	A_008-1, B_012, B_056, C_001, C_029, C_057
3	B_083, B_087
4	B_016, B_053, B_055-1, B_131, D_068

TABLE IV. The clustering results using the VGG19 model with k-means clustering.

Cluster	**Images**
0	A_008, B_010, B_012, B_016, B_053, B_055-1, B_066, C_057
1	B_082, B_083, B_086, B_087, B_088, B_131, D_067, D_068, D_071, D_072, E_102, E_103, E_104
2	A_008-1
3	B_090, C_001, C_029
4	B_056

techniques manage to correctly cluster the textile artifacts when they are in good conditions and substantially differ from one another; however, the performance deteriorates when the textiles are similar in shade or are highly degraded. The state-of-the-art techniques show that the results are not perfect even when 3×3 puzzle of high quality photographs need to be solved if a small gap mimics an erosion[5,20]. The actual complexity that the researchers face in real-world archaeological

problems is substantially higher, as the fragments are highly degraded, come in irregular shapes, many fragments are missing, and the number of objects they belong to is unclear.

Color statistics are not informative enough to differentiate the fragments, as many of them may have similar color distributions after long degradation and decomposition process. The sophisticated deep learning models are, on the other hand, trained on natural images for classification tasks, to learn, for example, how to distinguish cats from dogs. However, they cannot capture the subtle differences among archaeological textiles, which may require specialized training on the application-specific dataset, which itself is a fundamental problem, since no big enough dataset of archaeological textiles exists, to the best of our knowledge. For instance, refer to Fig. 19. The pairs of fragments shown in Fig. 19 (a) and (b) are believed to belong together after manual inspection and analysis. The fragments in (a) have similar motifs: processions with visible horses and men with spears. Such motifs are more challenging to be detected by a machine, and specialized training with a labeled dataset, where such fragments could be labeled as *horse* could benefit the model, which is trained on natural images of horses. The similarities in (b) that were detected by the experts are difficult not only for machines but for non-expert humans as well. These fragments are also grouped together because they have been part of the same "textile cake" that was later split up. Such information is not only useful for manual analysis, but it can also be a valuable metadata supplemented to the images that can be utilized by computational algorithms. One of the cues that the human experts rely on are similar patterns, such as the rhombi shown in Fig. 19 (c). Even though the two fragments where these patches are extracted from are believed to bear similarities with one another, the deep learning model failed to capture this, and the clustering algorithm placed the fragments into different clusters. Specialized training on close-up images of popular patterns could teach the model detecting such significant cues. Furthermore, human experts rely on specific technical features (such as the number threads per centimeter, fiber thickness)

as well as high-level semantics and metadata (such as the local grave context where each fragment was found)[1,3,28]. For example, Fig. 19 (d) illustrates the case, where two patches were mistakenly clustered together despite apparent difference in the size of the threads as well as the number of threads per centimeter. This is expected, since the threads play a negligible role in the datasets that the neural networks are trained on. Training the networks on close-up images of textiles can facilitate classification by the fiber diameter, thread count per cm, and twisting direction (S or Z). For instance, Fig. 19 (e) illustrates two different twists – S and Z, which is a significant cue for human experts when analyzing the textiles[3]. Training on labeled datasets of S and Z twists can easily automatize twist detection, which is apparently overlooked in existing pre-trained deep learning models. Future works should attempt to mimic the humans, automatize the technical analysis, and also consider the user-provided metadata. For example, automatic solutions have been proposed for canvas thread counting for art forensics applications[49]. A similar approach may be used to characterize the thread count of the archaeological tapestries. The challenges arise at the different stages of the pipeline. At the acquisition stage, large amount of dimensionality reduction takes place when the reflected spectrum is decreased to R, G, and B channels of the color image. Future works should try more sophisticated imaging techniques, such as hyperspectral imaging (HSI) and reflectance transformation imaging (RTI). The former may reveal more information about fiber's state and chemical composition, while the latter can shed more light on structural patterns. Artificial aging to learn the inversion of the degradations, as well as inpainting in specific cases, can be used to improve the quality of the dataset.

While no substantial difference was observed among clustering techniques, feature extraction seems to be the crucial step in the puzzle solving problem. Machine learning models need to be trained on specialized datasets. In addition to CNNs, alternative deep learning architectures, such as vision transformers[50] and diffusion models[51], have demonstrated promising performance for

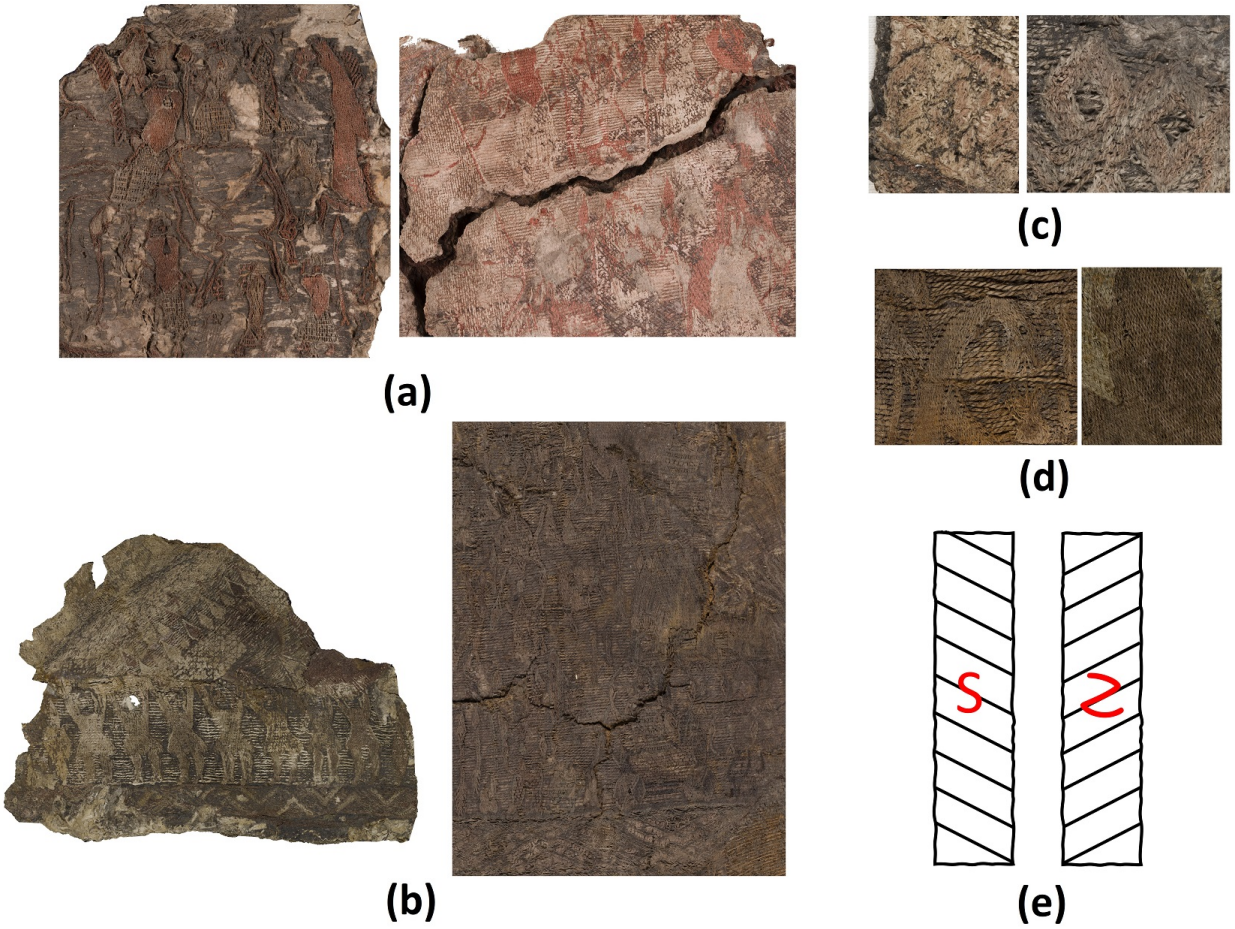


FIG. 19. The motifs provide invaluable information for puzzle solving. The horses and warriors with spears in (a) indicate that the two fragments may belong to the same item. The fragments in (b) originally belonged to the same "textile cake" that was later split up. Their motifs are, however, difficult to be detected by the machine. (c) illustrates the rhombus pattern, which is also an additional indication on similarity and which the machine also failed to capture. Close-up textile images, such as those in (d) can help the neural networks to learn classifying textiles by their technical characteristics, such as thread count. Additional criterion that is widely used by humans and can be automatically detected if proper dataset exists for training is the twist direction, S or Z, as shown in (e). Photographs (a)-(d) by George Alexis Pantos.

puzzle solving that may merit the interest in the future.

Nevertheless, considering the limitations of the state-of-the-art as well as the complexity of the

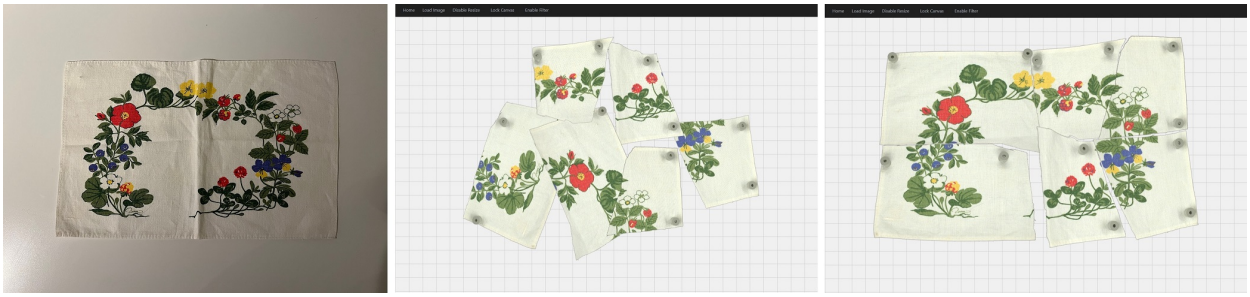


FIG. 20. The physical textile (left) need to be digitized and segmented. Afterward, its fragments can be loaded into the *Artifact Assembly* software (middle), where they can be assembled freely without a fear of damage to the physical object (left). Reproduced from[12].

problem, we believe that no fully automatized solution will be achieved in a foreseeable future. The computational techniques should be, hence, considered supplementary to human expertise and not its substitute.

VI. ARTIFACT ASSEMBLY

Manual solutions to archaeological textile puzzles are also limited due to the fragility of the artifacts. Therefore, Gulbrandsen[12] developed a software *Artifact Assembly* that enables users to virtually assemble puzzles on a digital canvas. The software lets the users scale the fragments, move them freely around the canvas, zoom in and zoom out. This will facilitate manual puzzle solving as it removes the stress and care associated with handling the physical fragments. An example of such assembly is illustrated in Fig. 20. Besides, the visibility of the motifs can be enhanced by color enhancement filters that enable hue rotation, as well as saturation and contrast manipulation for better visibility, as illustrated in Fig. 21. If robust similarity metrics are found in the future, they will be incorporated into the software, so that the software could suggest potential solutions to a human user.

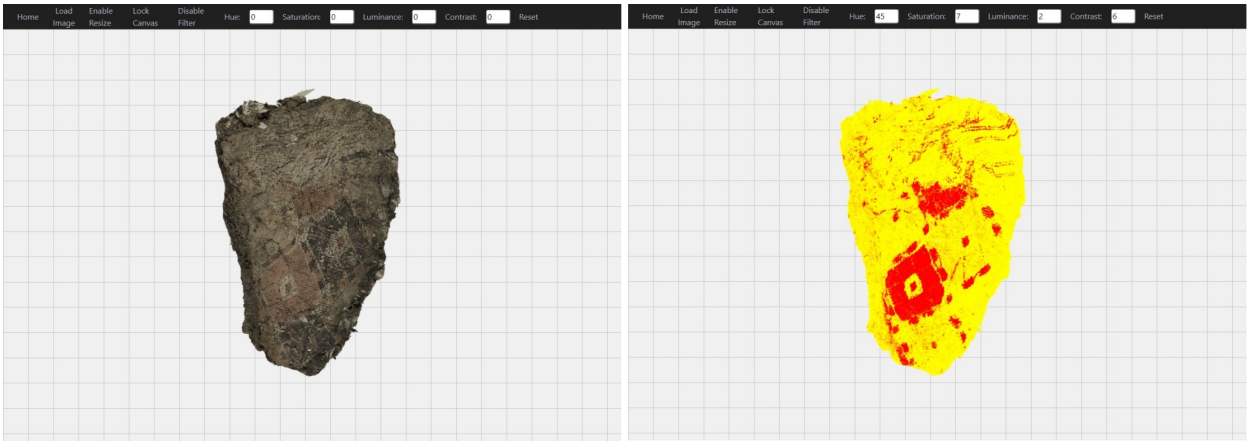


FIG. 21. Color enhancement may facilitate manual puzzle solving. Reproduced from[12].

VII. CONCLUSION

This work is one of the first steps toward automatic re-assembly of highly fragmented archaeological textiles. Different color statistics, texture descriptors, and pre-trained convolutional neural networks were used to extract features from the textile images that were subsequently fed to clustering algorithms to identify similar fragments. Four case studies were illustrated: from simpler scenarios, where the textiles were in a good condition and ground truth solution was known, to a highly complex in-the-wild problem of the actual archaeological tapestry with high degree of degradation and unknown solution. The work revealed a significant knowledge gap in the state-of-the-art and highlighted the need for more specialized computational techniques for archaeological textile classification and re-assembly.

REFERENCES

[1]M. Vedeler, *The Oseberg Tapestries* (Scandinavian Academic Press, Oslo, Norway, 2019).

[2]D. Gigilashvili, H. Lukesova, C. F. Gulbrandsen, A. Harijan, and J. Y. Hardeberg, "Computational techniques for virtual reconstruction of fragmented archaeological textiles," Heritage

Science **11:259**, 1–16 (2023).

[3]D. Gigilashvili, H. Lukesova, and J. Y. Hardeberg, "Criteria for matching fragmented archaeological textiles: A survey," Currently under review.

[4]M.-M. Paumard, D. Picard, and H. Tabia, "Jigsaw puzzle solving using local feature co-occurrences in deep neural networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)* (IEEE, 2018) pp. 1018–1022.

[5]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).

[6]M.-M. Paumard, H. Tabia, and D. Picard, "Alphazzle: Jigsaw puzzle solver with deep monte-carlo tree search," arXiv preprint arXiv:2302.00384 , 1–20 (2023).

[7]N. A. Rasheed and M. J. Nordin, "A survey of computer methods in reconstruction of 3D archaeological pottery objects," International Journal of Advanced Research **3**, 712–714 (2015).

[8]R. de Lima-Hernandez, S. Vincke, M. Bassier, L. Mattheuwsen, J. Derdeale, and M. Vergauwen, "Puzzling engine: A digital platform to aid the reassembling of fractured fragments," International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences , 563–570 (2019).

[9]R. de Lima-Hernandez and M. Vergauwen, "A hybrid approach to reassemble ancient decorated block fragments through a 3D puzzling engine," Remote Sensing **12**, 1–19 (2020).

[10]R. de Lima-Hernandez and M. Vergauwen, " A generative and entropy-based registration approach for the reassembly of ancient inscriptions ," Remote Sensing **14**, 1–27 (2021).

[11]D. Gigilashvili, H. T. Nguyen, C. F. Gulbrandsen, M. Havgar, M. Vedeler, and J. Y. Hardeberg, "Texture-based clustering of archaeological textile images," in *The Proceedings of Archiving 2023 Conference*, Vol. 20 (IS&T, 2023) pp. 139–142.

[12]C. F. Gulbrandsen, *Reconstructing the Original: Machine Learning Puzzle Assembly for Matching Archaeological Textile Fragments.* (Master Thesis. Norwegian University of Science and Technology, 2023).

[13]H. Freeman and L. Garder, "Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition," IEEE Transactions on Electronic Computers , 118–127 (1964).

[14]M.-M. Paumard, *Solving Jigsaw Puzzles with Deep Learning for Heritage*, Ph.D. thesis, PhD Thesis. CY Cergy Paris Université (2020).

[15]M.-M. Paumard, "Semantic-based automated reassembly of heritage fragments," in *CeROArt. Conservation, exposition, Restauration d'Objets d'Art*, 12 (Association CeROArt asbl, 2020) p. n.p.

[16]K. Son, J. Hays, D. B. Cooper, *et al.*, "Solving small-piece jigsaw puzzles by growing consensus," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016) pp. 1193–1201.

[17]G. Paikin and A. Tal, "Solving multiple square jigsaw puzzles with missing pieces," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (2015) pp. 4832–4839.

[18]K. Son, J. Hays, and D. B. Cooper, "Solving square jigsaw puzzles with loop constraints," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13* (Springer, 2014) pp. 32–46.

[19]D. Sholomon, O. David, and N. S. Netanyahu, "A genetic algorithm-based solver for very large jigsaw puzzles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013) pp. 1767–1774.

[20]M.-M. Paumard, D. Picard, and H. Tabia, "Image reassembly combining deep learning and shortest path problem," in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018) pp. 153–167.

[21]Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision* (2017) pp. 2736–2744.

[22]A. Ukovich and G. Ramponi, "Feature extraction and clustering for the computer-aided reconstruction of strip-cut shredded documents," Journal of Electronic Imaging **17**, 013008:1–013008:13 (2008).

[23]S. Chanda, K. Franke, and U. Pal, "Clustering document fragments using background color and texture information," in *Document Recognition and Retrieval XIX*, Vol. 8297 (SPIE, 2012) pp. 230–237.

[24]N. A. Rasheed and M. J. Nordin, "A survey of classification and reconstruction methods for the 2d archaeological objects," in *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)* (IEEE, 2015) pp. 142–147.

[25]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 *Proceedings of the Genetic and Evolutionary Computation Conference* (2019) pp. 1319–1327.

[26]C. Toler-Franklin, *Matching, archiving and visualizing cultural heritage artifacts using multi-channel images* (Princeton University, 2011).

[27]K. Liu, J. Zhao, and C. Zhu, "Research on digital restoration of plain unlined silk gauze gown of mawangdui han dynasty tomb based on ahp and human–computer interaction technology," Sustainability **14**, 1–19 (2022).

[28]M. M. Kodrič Kesovia, Ž. Penava, and D. Jemo, "The story of two historical textile fragments: Technical analysis and reconstruction of the lost textile pattern," Textile Research Journal **91**, 2859–2871 (2021).

[29]C. Stoean, N. Bacanin, R. Stoean, L. Ionescu, C. Alecsa, M. Hotoleanu, M. Atencia, G. Joya, *et al.*, "On using perceptual loss within the u-net architecture for the semantic inpainting of textile artefacts with traditional motifs," in *SYNACS Conference Publishing Service (CPS)* (2022) p. 8 pages.

[30]X. Sun, J. Jia, P. Xu, J. Ni, W. Shi, and B. Li, "Structure-guided virtual restoration for defective silk cultural relics," Journal of Cultural Heritage **62**, 78–89 (2023).

[31]CIE, *CIE 175:2006 A framework for the measurement of visual appearance* (International Commission on Illumination., 2006) p. 92 pages.

[32]R. Chakravarti and X. Meng, "A study of color histogram based image retrieval," in *2009 Sixth International Conference on Information Technology: New Generations* (IEEE, 2009) pp. 1323–1328.

[33]D. Ping Tian *et al.*, "A review on image feature extraction and representation techniques," International Journal of Multimedia and Ubiquitous Engineering **8**, 385–396 (2013).

[34]G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the fourth ACM International Conference on Multimedia* (1997) pp. 65–73.

[35]T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," Pattern Recognition **29**, 51–59 (1996).

[36]T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8* (Springer, 2004) pp. 469–481.

[37]T. Mäenpää, *The local binary pattern approach to texture analysis—extensions and applications* (University of Oulu, 2003).

[38]C.-H. Chan, J. Kittler, and K. Messer, "Multispectral local binary pattern histogram for component-based color face verification," in *2007 First IEEE International Conference on Bio-*

metrics: Theory, Applications, and Systems (IEEE, 2007) pp. 1–7.

[39]S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Vol. 1 (IEEE, 2021) pp. 96–99.

[40]S. Lloyd, "Least squares quantization in pcm," IEEE transactions on information theory **28**, 129–137 (1982).

[41]J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," Journal of the royal statistical society. series c (applied statistics) **28**, 100–108 (1979).

[42]F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**, 86–97 (2012).

[43]F. Nielsen and F. Nielsen, "Hierarchical clustering," Introduction to HPC with MPI for Data Science , 195–211 (2016).

[44]K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)* (IEEE, 2020) pp. 747–748.

[45]R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," IEEE Transactions on Systems, Man, and Cybernetics **6**, 610–621 (1973).

[46]L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods-a review," arXiv preprint arXiv:1904.06554 (2019).

[47]A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM **60**, 84–90 (2017).

[48]D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Transactions on pattern analysis and machine intelligence **24**, 603–619 (2002).

[49]D. H. Johnson, C. R. Johnson, A. G. Klein, W. A. Sethares, H. Lee, and E. Hendriks, "A thread counting algorithm for art forensics," in *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop* (IEEE, 2009) pp. 679–684.

[50]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).

[51]S. Hosseini, M. A. Shabani, S. Irandoust, and Y. Furukawa, "Puzzlefusion: Unleashing the power of diffusion models for spatial puzzle solving,".