Learning-free pattern detection for manuscript research:

An efficient approach toward making manuscript images searchable

Hussein Mohammed[1] · **Volker Märgner[2]** · **Giovanni Ciotti[1]**

Abstract

Automatic pattern detection has become increasingly important for scholars in the humanities as the number of manuscripts that have been digitised has grown. Most of the state-of-the-art methods used for pattern detection depend on the availability of a large number of training samples, which are typically not available in the humanities as they involve tedious manual annotation by researchers (e.g. marking the location and size of words, drawings, seals and so on). This makes the applicability of such methods very limited within the field of manuscript research. We propose a learning-free approach based on a state-of-the-art Naïve Bayes Nearest-Neighbour classifier for the task of pattern detection in manuscript images. The method has already been successfully applied to an actual research question from South Asian studies about palm-leaf manuscripts. Furthermore, state-of-the-art results have been achieved on two extremely challenging datasets, namely the *AMADI_LontarSet* dataset of handwriting on palm leaves for word-spotting and the *DocExplore* dataset of medieval manuscripts for pattern detection. A performance analysis is provided as well in order to facilitate later comparisons by other researchers. Finally, an easy-to-use implementation of the proposed method is developed as a software tool and made freely available.

1 Introduction

Automatic pattern detection and recognition can facilitate research for scholars of manuscript studies and provide quantitative measurements as supporting information. Such methods are particularly important when dealing with a large number of manuscripts.

Over the last decade, considerable advances have been made in the tasks of object detection [1] and segmentation-free word-spotting [2]. Most of the state-of-the-art methods currently employed for these two tasks depend on the avail-

ability of a large number of training samples. These samples need to be annotated manually beforehand (e.g. marking the location and size of words, drawings, seals, etc.).

Although learning-based approaches can be useful when the training samples, annotations and computational resources for them are all available, the applicability of such methods is very limited in manuscript research. Scholars often deal with a small number of images within the scope of a specific research question. Even if a large number of images are available, most of the manuscripts that contain them do not contain any ground-truth information, such as related metadata or transcriptions. Annotations of this kind can only be created under the supervision of experts from the manuscript field in question, and even then some of these annotations are just a matter of subjective opinion. The aforementioned reasons render most of the learning-based methods inapplicable or at least unfeasible for most questions in manuscript research.

Furthermore, the images examined in manuscript research often contain different scripts, even on one page. Some of these scripts can only be read by a few experts from the humanities. In addition, manuscript images often suffer from

✉ Hussein Mohammed
  hussein.adnan.mohammed@uni-hamburg.de

  Volker Märgner
  maergner@ifn.ing.tu-bs.de

  Giovanni Ciotti
  giovanni.ciotti@uni-hamburg.de

1 Cluster of Excellence: Understanding Written Artefacts, Universität Hamburg, Hamburg, Germany

2 Technische Universität Braunschweig, Braunschweig, Germany

Fig. 1 Different detected patterns in manuscripts using the proposed learning-free method. To achieve better visibility, only parts of the images are shown

several types of degradation, such as low resolution, low contrast, varying background intensity and other factors caused by the poor state of preservation of the actual manuscripts or the nature of the writing support (e.g. bleed-through, a textured background, stains and water damage). Pre-processing steps such as segmentation, layout analysis, OCR and binarisation are therefore challenging, and in many cases they are not feasible at all.

This is why we developed a learning-free pattern detection method that does not require any pre-processing steps. It is a practical alternative to making digitised manuscripts searchable not only for text, but for visual patterns in general such as letters, seals or drawings. Different detected patterns in manuscripts are shown in Fig. 1 using the proposed method in order to demonstrate its general applicability. The pattern in (a) is a handwritten word in a manuscript from the École française d'Extrême Orient (EFEO), Pondicherry branch, the pattern in (b) is a seal in a manuscript from the British Library: Oriental Manuscripts (https://www.qdl.qa/archive/81055/vdc_100023410391.0x00003c), and the patterns in (c) and (d) are parts of a ship and a person's head in medieval manuscripts from the *DocExplore* dataset [3].

The work in [4] demonstrated a state-of-the-art classification rate for the task of writer identification on manuscript images using the learning-free NBNN-based classifier proposed in [5] without any pre-processing steps. In addition, the work in [6] proposed a category-level object detection method based on the Naïve Bayes Nearest-Neighbour (NBNN) algorithm with state-of-the-art performance on datasets of objects in complex scenes. We based our proposed method on the aforementioned methods in order to benefit from their strong points.

The method proposed in [6] has two free parameters. This can hinder its practicality as a research tool to be used on a wide variety of patterns with different quality and degradation levels. One of these parameters has therefore been eliminated in this work, while the other is calculated adaptively from the images of labelled patterns.

Consequently, we are presenting a learning-free method here that does not require any pre-processing steps at all and that can cope with the heavy degradation typically found in manuscript images. Furthermore, this method is a general detection algorithm that can be used to detect a wide variety of patterns in manuscripts.

The main achievements in this paper are the following:

– elimination of the two free parameters from the method presented in [6] in order to develop a practical solution (see Sect. 4 for more details);
– application of Features from Accelerated Segment Test (FAST) keypoints [7] with the adaptive threshold PCK (Percentage of Considered Keypoints) presented in [5] and application of the Normalised Local NBNN distance measure presented in [5] in order to enhance the performance of the method described in [6] when applied to manuscript images (see Sect. 4, steps 1 and 5 for details);
– application of the resulting learning-free method to an actual research question from the humanities about palm-leaf manuscripts (see Sect. 3 for details);
– providing state-of-the-art results on two extremely challenging datasets, namely the *AMADI_LontarSet* dataset [8] of handwriting on palm-leafs for word-spotting and the *DocExplore* dataset [3] of medieval manuscripts for pattern detection, with performance analysis in order to facilitate later comparisons.
– developing an easy-to-use implementation of the proposed method, and releasing it as a free software tool to the public.

The remainder of this paper is organised as follows: In Section 2, we will discuss some of the related works along with relevant public datasets. In Section 3, a use case from manuscript research is presented, followed by a discussion of the role and importance of our proposed method in answering the research question. In Sect. 4, the pattern detection method we have developed will be presented. In Sect. 5, a performance evaluation is provided using the research data from the use case and two relevant and very challenging public datasets. In Sect. 6, we will describe our implementation of the proposed method as a software tool. The final Section contains our conclusions.

2 Related work

Pattern detection can be considered as the general category that includes both object detection and segmentation-free word-spotting tasks as two of its special cases. The idea of automatically detecting patterns in manuscript images has been around for at least a decade [9], but no significant progress has been made so far due to the lack of standard and public datasets with ground-truth annotations. Furthermore, the fact that most of state-of-the-art methods depend on the availability of annotated training data has hindered progress.

In the task of segmentation-free word-spotting, the pattern is typically a printed [10,11] or handwritten [12,13] word in a document. In manuscript research, it is often the case that words are parts of handwritten sentences on degraded writing supports such as parchment, palm leaves or papyri. Most segmentation-free word-spotting methods have been evaluated on texture-free paper with no or very limited degradation and a dedicated training set of annotated data [11,14–16].

The use of local features for the task of segmentation-free word-spotting has been a successful approach in many proposed methods [10–12]. These extracted features are typically clustered or used to train classifiers in most of these methods [12,14,15], or they are directly matched to the features of test images [10,11]. The need for "training-free" methods was recently highlighted [16] in order to cope with the lack of labelled samples for the task of segmentation-free word-spotting.

In contrast, several pattern-detection methods have been proposed to detect symbols, logos and other types of patterns found in documents [17–20]. Some of these methods have been dedicated to detecting patterns in historical documents and manuscripts [3,21,22]. This article is particularly concerned with detecting patterns in historical documents and aims to facilitate manuscript research. The focus of the paper is therefore on datasets which are relevant to research questions that manuscript scholars wish to address.

Recently, two extremely challenging datasets were published: the *AMADI_LontarSet* dataset [8] of handwriting on palm-leaves for word-spotting and the *DocExplore* dataset [3] of medieval manuscripts for pattern detection. No results have been reported on the first dataset so far. On the other hand, results of the authors of the second dataset showed clearly that there is room for improvement [21,22]. These two datasets are relevant to our own work and offer realistic scenarios in manuscript research, where very few labelled samples are available for each pattern to be detected (sometimes only one). These two datasets were used for performance evaluation in this article for the aforementioned reasons.
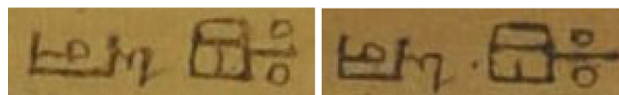

Fig. 2 The two occurrences of the invocation found by Giovanni Ciotti and Marco Franceschini [23]

3 Use case from manuscript research

The current research aims at contextualising the occurrence of a unique and hitherto unnoticed palaeographical feature that appears in some palm-leaf manuscripts hailing from the cultural area corresponding to Tamil Nadu today (in Southern India).

Out of the tens of thousands of manuscripts that are held in libraries across Tamil Nadu and contain texts mainly composed in Sanskrit and Tamil (the former mostly written in Tamilian Grantha script and the latter mostly in Tamil script), only a few thousand that are available for scholars of South Asian studies to scrutinise have been digitised so far (each manuscript consists of hundreds of folios). For the last few years, Giovanni Ciotti (University of Hamburg) and Marco Franceschini (University of Bologna) have been making a systematic study of colophons found in these manuscripts [23] and have identified several uncommon codicological and palaeographical features that await further investigation.

One such feature is a marginal invocation written in a rather unusual square style of Tamilian Grantha. This is a graphical variant of the widely attested invocation reading *hariḥ om*, very often found at the beginning of manuscripts.

So far, Ciotti and Franceschini have found two occurrences of this squared *hariḥ om* (see Fig. 2 for images of the word *hariḥ*) from the manuscripts belonging to the manuscript collection of the École française d'Extrême Orient (EFEO), Pondicherry branch. There are not enough occurrences to allow them to understand the context in which such a distinctively written invocation appears, however.

3.1 Research question

Collecting as many occurrences as possible of such a unique palaeographical feature can open a new window on the practices of traditional scribal activity in Tamil Nadu.

If more occurrences were available, it would be possible to link the squared *hariḥ om* to specific scribes or groups of scribes. It might even be possible to link them to specific literary genres, were they to appear in manuscripts containing a specific variety of texts, or to a well-defined time and place of production (if the colophons provided pertinent data), thus possibly corresponding to a particular scribal fashion that characterised a certain period or region.

If one or more of these assumptions were confirmed, it would be possible to make significant progress in the attempt

to divide manuscripts into subsets and thus reconnect them to their past. As it turns out, however, Indian libraries have not kept detailed records of their provenance. In other words, the ties between manuscripts and their past have been severed, so the individual history of each item needs to be reconstructed.

3.2 The importance of learning-free automatic pattern detection

The proposed method allows us to automate the search for specific palaeographical features we are interested in over hundreds of thousands of images of manuscripts in the EFEO collection. This procedure not only saves an enormous amount of time, but it enables us to answer our research question. Using a learning-free approach is critical in this case because only two instances of the pattern are available, as mentioned earlier. Furthermore, the possibility of providing annotated data is very limited due to the need for specialists in the field.

Moreover, producing such annotations would clash with the main reason for us using automated pattern detection in the first place, namely to save time and effort. Without a suitable form of pattern detection that is automated, it would take several years to go through each manuscript folio in the collection looking for occurrences of the squared *harih om*.

The proposed method can be applied to automate the search for the same palaeographical features over even larger sets of manuscript images. Furthermore, several other patterns could be looked for. For example, specific words that may appear in the margins of manuscripts and indicate the name of the literary genre of the texts contained there or symbols such as those used to indicate calendrical elements in colophons (the year, month and solar day).

4 The proposed method

As mentioned in the introduction, the proposed method is based on the work presented in [6] for category-level object detection and the work presented in [5] for writer identification. Several modifications and optimisations have been undertaken in order to have a practical pattern detection method for manuscript research. The resulting algorithm is shown in a simplified form in Fig. 3. A detailed description of the method involves the following steps:

Step 1: Since patterns in manuscript research are mostly the result of handmade marks on writing supports, the resulting features on the formed contours can be efficiently detected using the FAST [7] keypoints detector with the adaptive threshold PCK (Percentage of Considered Keypoints) after converting the coloured images
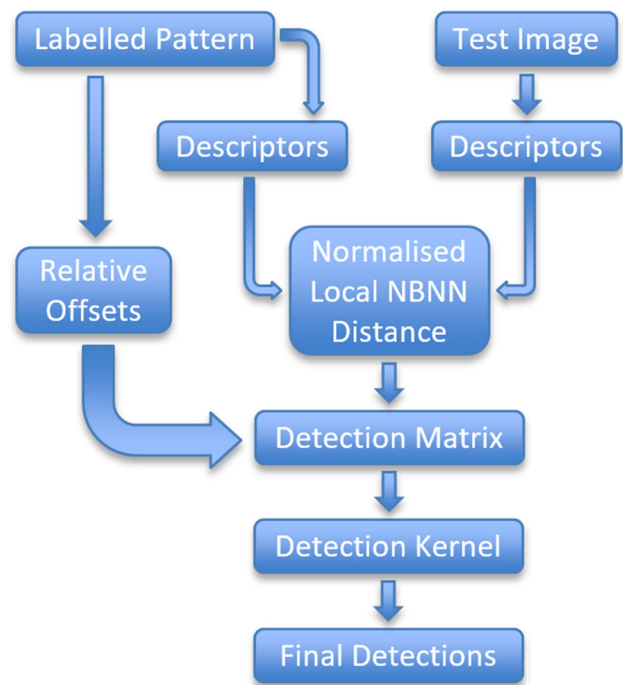

Fig. 3 A simplified illustration of the proposed learning-free algorithm for pattern detection. See method description of Steps 1 to 6 for more details


Fig. 4 An example of detected FAST keypoints in a handwritten pattern where $PCK = 10\%$. Each detected keypoint is represented by a circle of different colour. This pattern is part of an image from the École française d'Extrême Orient [EFEO], Pondicherry branch

Fig. 5 This figure shows five detected features in part of a test image and the corresponding centre of an expected pattern. This pattern is part of an image from the École française d'Extrême Orient [EFEO], Pondicherry branch

to grey-scale images, as demonstrated in [4]; an example is shown in Fig. 4. A circular neighbourhood of 16 pixels is used around every pixel p in the image to be classified as a keypoint if there are n contiguous pixels in the surrounding circle satisfying one of these conditions:

$$- \quad \forall i \in n : I_i > I_p + t,$$
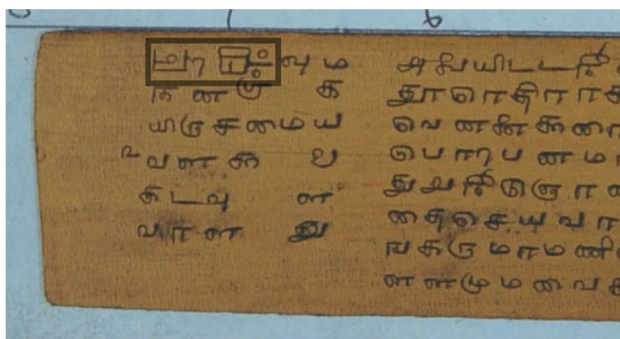
(a) Part of the original test image from the École française d'Extrême Orient [EFEO], Pondicherry branch.

(b) The corresponding detection matrix visualised after the smoothing step. The intensity values have been normalised for better visualisation.

Fig. 6 The detection matrix for part of a test image shown in (a). Each dark spot in (b) indicates one expected pattern centre voted for by one feature or more; see Fig. 5 for an illustration of the voting process

or

- $\forall i \in n : I_i < I_p - t$,

  where I_p is the intensity of the candidate pixel and I_i is the intensity of any pixel that belongs to the n contiguous pixels in the neighbourhood. t is a threshold to be selected manually. n is set to 9 following the recommendation in [7], and t is set to zero so that we initially consider all the detected keypoints before filtering them by strength using the PCK parameter as described below.

  The strength of a keypoint is the maximum value of t for which the segment test of that corner point is satisfied, and PCK is the percentage of considered FAST keypoints with the highest strength value; see Fig. 4.

  The detected keypoints using the FAST algorithm are obviously dependent on image resolution because of the fixed size of the circular neighbourhood. The detection performance is expected to drop gracefully as the scale difference between the queries and the pattern instances in the images increases; see the degradation analysis of FAST keypoints in [4]. Nev-

ertheless, limited scale invariance can be obtained by generating additional scales for each query sample. The descriptors of detected features are then calculated using the Scale-Invariant Feature Transform (SIFT) algorithm [24]. The relative location of detected features is stored as a scaled offset with respect to the spatial centre of the labelled pattern; the keypoint size can be used as a scaling factor when a multi-scale keypoints detection algorithm is used. Local features are detected and described in the test images following the same procedure for query images, but without storing any relative locations.

- *Step 2:* When coloured images are converted to greyscale images, pixels within the range of the red spectrum tend to have very low intensity values. As a consequence, the local contrast will be low compared with other spatial regions.

  Since our proposed method detects keypoints and extracts features from grey-scale images, the performance could be negatively affected if the query image contains red parts. Thus, the aforementioned issue has to be modified. This is particularly relevant when dealing with manuscript images because colours within the red range frequently appear in handwriting, decorations and drawings.

  The modification is done in the following way: First, the range of red colour is defined as a range of Hue values after converting the image from Red–Green–Blue (RGB) format to Hue–Saturation–Value (HSV) format. Then a mask is created to define the spatial location of red pixels in the image. Finally, the keypoints located within this spatial region are sorted separately. Once the strongest ten per cent of all the keypoints have been selected as described in Step 1, the strongest ten per cent of the spatial location of red pixels are added. This allows keypoints detected in low-contrast red regions to be included in the total number of Considered Keypoints (PCK).

- *Step 3::* The performance of the object detection method presented in [6] is sensitive to the Kernel Radius R, which is a free parameter of the method. Therefore, we propose to calculate it automatically using the image dimensions of labelled patterns. This parameter represents the radius of the kernel, which convolves with the detection matrix in order to generate the final detections; see Eq. 8. In our approach, the kernel size is adaptively calculated from the average value of all medians of width and height for all

the examples from a given labelled pattern (class) as follows:

$$R_c = 0.1 \times \left[\frac{(\text{Med}_c^w + \text{Med}_c^h)}{2} \right]; \qquad (1)$$

where R_c is the calculated parameter R for pattern (class) c, and Med_c^w and Med_c^h are the medians of widths and heights respectively, calculated from all the samples of a given labelled pattern (class) c; which are typically no more than a few samples, or even just one.

The average value of all medians are multiplied by a fixed value to calculate the final kernel radius. This fixed value has been set to 0.1 (10%) in all our experiments. Other values have been tested in our preliminary experiments with no significant difference in the overall performance, but the performance starts to drop once we exceed a value of 0.5.

– *Step 4:* Two detection matrices are used in [6] for each class with the same size of the test image. One matrix (M_c^v) accumulates the number of matched features for the corresponding class in a location calculated by Eq. 3. The other matrix (M_c^s) accumulates the distances calculated between the features in the test image and the labelled query. These two matrices are then combined, after being convolved with their corresponding kernels, in order to calculate the final detection matrix (M_c) using the parameter α, which has to be selected manually, as a weight:

$$M_c = M_c^s * K_{\text{mask}} + \alpha(M_c^v * K_{\text{dist}}); \qquad (2)$$

where K_{mask} and K_{dist} are the kernels to be convolved with the corresponding matrices.

In this work, only one detection matrix per pattern is created for each test image instead of the two matrices used in [6]. Our preliminary experiments showed that the matrix M_c^v does not contribute to the performance of the method in the used datasets of digitised manuscripts, yet it adds to the total computational cost. Only the matrix M_c^s is used from [6] and renamed $M_c^{d_i}$. As a result, the parameter α has been eliminated and there is no need to perform any further computations. The detection matrix $M_c^{d_i}$ is the same size as the corresponding test image.

– *Step 5:* One of the main contributions proposed originally by the NBNN algorithm [25] is measuring the image-to-class distance instead of image-to-image distance in order to generalise the image-matching to class-matching. The image-to-class distance is mea-

sured by calculating the overall distance of image features to the features of all the images in a given class instead of the features of one image (image-to-image distance). In this work, we measure the feature-to-class distance in order to estimate the distance of each detected feature in the test image to the class distributions estimated by their labelled features.

Each detected feature in the test image votes for a centre of an expected pattern in the detection matrix; see Fig. 5. The position of this expected centre is calculated using the relative location of nearest-neighbour feature in the corresponding labelled pattern as follows:

$$L_{i,c} = L_f(d_i) - \text{Offset}(NN_c(d_i)); \qquad (3)$$

where $L_{i,c}$ is the location of the expected centre by feature d_i in the detection matrix of class c. $L_f(d_i)$ is the location of feature d_i in the test image. $\text{Offset}(NN_c(d_i))$ is the scaled offset of the nearest-neighbour feature from the centre of the labelled pattern from the corresponding class.

An example in Fig. 5 shows five detected features. Each one in the test image votes for the centre of an expected (labelled) pattern (class) using relative offsets. Circles represent the detected features, and the dots indicate the expected centres. Colours are used to associate each detected feature with its expected centre. It is clear that the feature marked in pink has been mismatched with the wrong feature in this example. Only detected features in the second part of the word are used in this example, and PCK is set to one percent for better visibility.

The value of the vote is equal to the distance of each detected feature in the test image to features of the corresponding class (labelled pattern) using the Normalised Local NBNN distance calculation presented in [5] in order to consider the calculated priori of each class which is approximated by the number of detected features in each class:

$$M^d(L_{i,c}) = M^d(L_{i,c}) + \text{Dist}_N(d_i, c), \qquad (4)$$

$$\text{Dist}_N(d_i, c) = \frac{\text{Dist}(d_i, c)}{K_c}, \qquad (5)$$

where $M^d(L_{i,c})$ is the detection matrix of class c and $\text{Dist}_N(d_i, c)$ is the normalised distance between the detected feature d_i in the test image and class c using the distance calculation presented in [5]. K_c is the number of features from the labelled patterns in class

c, and Dist(d_i, c) is the Local NBNN [26], which has been reformulated in [5] as follows:

$$\text{Dist}(d_i, c) = \sum_{i=1}^{n} \left[\left(\parallel d_i - \phi(\text{NN}_c(d_i)) \parallel^2 \right. \right.$$
$$\left. \left. - \parallel d_i - \text{N}_{k+1}(d_i) \parallel^2 \right) \right], \qquad (6)$$

where

$$\phi(\text{NN}_c(d_i)) = \begin{cases} \text{NN}_c(d_i) & \text{if } \text{NN}_c(d_i) \leq \text{N}_{k+1}(d_i) \\ \text{N}_{k+1}(d_i) & \text{if } \text{NN}_c(d_i) > \text{N}_{k+1}(d_i), \end{cases}$$

and $\text{N}_{k+1}(d_i)$ is the neighbour $(k+1)$ of d_i. In a similar way to the work in [26], we used the distance to the $k + 1$ nearest neighbours ($k = 10$) as a "background distance" to estimate the distances of classes which were not found in the k nearest neighbours. According to Eq. 6, the larger the value of $Dist(d_i, c)$ the closer class c to feature d_i, because $Dist(d_i, c)$ measures the distance between class c and the background $(k + 1)$ relative to d_i. Therefore, the matrix $M^d(L_{i,c})$ is initialised with zeros in order to allow for the detection of local maximums.

Search indices are created for all the classes using the kd-trees implementation provided by the FLANN [27] (Fast Library for Approximate Nearest Neighbours) to have efficient nearest-neighbour search. An example of a detection matrix is shown in Fig. 6. It can be clearly seen that the darkest spot corresponds to the centre of the correct pattern annotated in part (a) of Fig. 6. The detection matrices are smoothed using a Gaussian filter. The kernel size of the filter is R_c x R_c, where R_c is the adaptive parameter calculated in Equ. 1.

– *Step 6*: Each detection matrix is convolved with a kernel in order to produce the final detections. The detection kernel can be described as follows:

$$K_c^{d_i}(x, y) = \begin{cases} 1 & \text{if } \text{Offset}_x^2 + \text{Offset}_y^2 < R_c \\ 0 & \text{otherwise}, \end{cases} \qquad (7)$$

where $K_c^{d_i}(x, y)$ is the detection kernel of class c for the detected feature d_i centred at location (x, y). Offset_x and Offset_y are the differences in the x- and y-axis between the kernel centre and the current location (x,y) respectively.

The final detections D_c are calculated as follows:

$$D_c = M_c^{d_i} * K_c^{d_i}; \qquad (8)$$

The size of a detected pattern is set to be equal to the median height and width of the corresponding labelled pattern samples.

5 Evaluation on relevant datasets

We applied the proposed method on the École française d'Extrême Orient [EFEO] dataset from the use case presented in Sect. 3 in order to demonstrate the applicability of this method on actual research questions from manuscript scholars. In addition, we evaluated the method using two different public datasets in order to demonstrate its generality and state-of-the-art performance. As mentioned above, the two extremely challenging datasets are: the *AMADI_LontarSet* dataset [8] of handwriting on palm-leaves for word-spotting and the *DocExplore* dataset [3] of medieval manuscripts for pattern detection. The first dataset was selected because of its relevance to the use case described in Sect. 3. The second dataset is the only available public dataset for pattern detection in digitised manuscripts to the best of our knowledge.

5.1 The École française d'Extrême orient [EFEO] dataset

The data used in this piece of collaborative research was a set of palm-leaf manuscripts from Tamil Nadu mostly ascribable to the $19th$ century, with a few exceptions from the $17th$, $18th$ and $20th$ century. The digitised manuscript collections are kept at the École française d'Extrême Orient, Pondicherry branch (there are 1625 manuscripts, 155,372 images in total). This valuable source of data was recognised as a UNESCO "Memory of the World Collection" in 2005. A few samples from the EFEO collection can be seen in Fig. 7.

The detection process resulted in 86 images which were saved automatically to a folder along with a rectangular annotation for each detection hypothesis. A manual inspection by an expert from the field of Tamil studies confirmed seven correct detections in the saved images. The process of manual inspection only took a few minutes due to the low number of hypotheses and the clear annotations around each one. The clear visual differences (inter-class variation) between the detected instances and the labelled patterns demonstrate the ability of the proposed method to generalise beyond the labelled patterns; see Fig. 8.

In addition, some of the false positives that were detected are also pertinent to the aims of the current case study. In fact, they present features that are in between those of the standard way of writing *hariḥ om* and its squared version.

Fig. 7 A few examples of manuscripts from the École française d'Extrême Orient, Pondicherry branch. The samples have been cropped for better visualisation
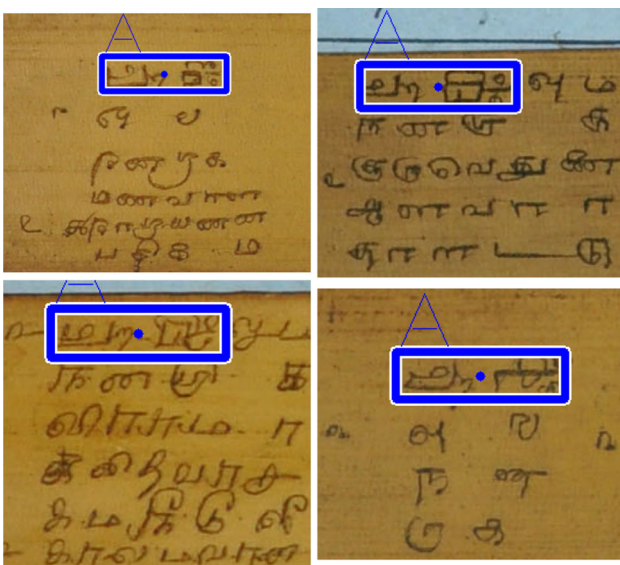

Fig. 8 Examples of the detections generated automatically by the proposed method. The samples have been cropped for better visualisation. Note the visual variations between the detected patterns and the labelled patterns in Fig. 2

The possibility of making such an observation thanks to the detections produced by our method indicates that the scribal activity we are investigating was more articulated than we thought initially since scribes had the possibility of modulating the graphic rendition of *hariḥ om* in more than just two ways.

Retrieving as many correct patterns as possible is more desirable in most of the cases, but it is done at the expense of precision because detected patterns can be inspected with very little effort. In other words, the recall rate is often more important than the precision for most questions in manuscript research.

This automatic pattern detection test was carried out using a standard office computer (with an Intel i5 core, 3.3 GHz)

in about three seconds per image. The test took up less than 1GB of RAM.

5.2 The *AMADI_LontarSet* dataset

The manuscript samples used in the *AMADI_LontarSet* dataset [8] are sample images of palm-leaf manuscripts from Bali, Indonesia. In order to obtain a fair representation of palm-leaf manuscript images, the sample images were collected from 23 different collections coming from five different regions: two museums and three private collections.

The dataset is partitioned into training and test subsets. Since the proposed method is a learning-free approach, the training subset is not used for training phase in this performance evaluation. A hundred original images and 36 word-level annotated query images were provided for the test subset. This means that only one example (labelled pattern) was used per query.

To the best of our knowledge, no word-spotting results have been published for this particular dataset, which makes this the first published result. Several standard performance measurements are provided in order to facilitate later comparisons with other methods and provide a thorough performance evaluation.

The performance evaluation of the proposed method is presented in Table 1 using standard metrics for object detection and word-spotting, namely mean Average Precision (mAP), average F-score, and the average recall rate at 0.3 False Positives Per Image (Recall at 0.3 FPPI). In order for the detection hypothesis to be considered as a true positive, the Intersection over Union ratio (*IoU*) must be more than 0.5 following the standard detection criteria. The same *IoU* condition was applied in all our experiments.

It is worth noting here that the performance of the method varies greatly across different patterns (queries) in this dataset, as Table 1 shows. In general, its performance is comparable to the state-of-the-art results even on much less challenging datasets used for word-spotting in historical handwritten manuscripts [28]. Nevertheless, the mAP is very low for a few queries. One possible explanation of the big difference in mAP across different queries is the complexity of the query pattern itself; see Figs. 9 and 10. The more complex the labelled pattern is, the more unique it is in terms of its visual features. Furthermore, the quality of the query image was an additional factor that influenced the quality of the calculated descriptors.

This method provides automatic, learning-free pattern detection that can save a significant amount of time and effort in the field of manuscript research. In the case of word-spotting, the method is a segmentation-free approach that can cope with the typical degradation found in manuscript images.

Table 1 Performance analysis of the proposed method on the *AMADI_LontarSet* dataset [8]

Queries	mAP	average F-score	average Recall at 0.3 FPPI
All 36 queries	0.476	0.707	0.732
The best performing 30 queries	0.560	0.780	0.810
The worst performing 6 queries	0.053	0.344	0.343
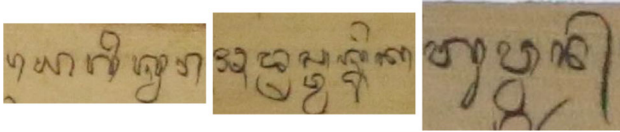
Fig. 9 Examples of the 30 queries with the highest mAP from the *AMADI_LontarSet* dataset [8]
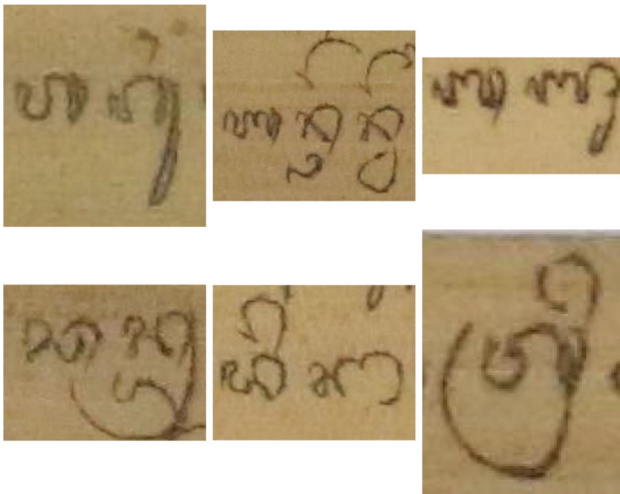
Fig. 10 The six queries with the lowest mAP from the *AMADI_LontarSet* dataset [8]

The test on the *AMADI_LontarSet* dataset was also performed using a standard office computer (with an Intel i5 core, 3.3GHz) taking an average of 13 seconds per image for all the 36 queries combined (thus making an average of 0.36 second per query). Only 1.8 GB of RAM was needed.

5.3 The *DocExplore* dataset

The manuscript images in the *DocExplore* dataset [3] are from the Municipal Library of Rouen, France, and they have been dated to between the 10*th* and the 16*th* century. A total of 1464 objects in 35 different graphical categories ranging from ornate initial letters to human faces and decorative objects in paintings were annotated for the task of pattern detection. Each object in a category was used as a query. The remaining objects in that category were kept as correct detections.

The number of annotated objects per category ranges from 2 to 409, with an average of 42. The query size can be very

small (about 10 x 20 pixels), but the average size is 77 x 77 pixels, which still only occupies 0.7% of the average document image size, which is 1024 x 768 pixels.

The mean Average Precision (mAP) was selected as the only possible performance measure for the task of pattern detection. The authors of this dataset did not provide any ground-truth information or annotation data, but they did develop a command-line tool which runs under Linux to generate mAP values as a performance measure for a given input file with a pre-defined format. We were therefore unable to perform any further performance analysis. As an additional consequence, we were not able to do a proper parameter analysis in order to determine the best possible settings for this dataset. The results provided were generated using the same parameter values used in the other datasets for FAST keypoint detection, the Normalised Local NBNN classifier and the adaptive kernel size K_C^d.

Large variations in the performance can be observed across different pattern categories in this dataset as well; see Table 2. The very low mAP values for a few query categories can be attributed to the lack of visual complexity in the queries compared to the queries in other categories from the same dataset; see Figs. 11 and 12. In addition, some categories in this dataset are visually identical to parts of patterns in other categories; "Ship hull L" can be detected in a "Ship" instance, for example, and both "Simple Separator" and "Double Separator" can be detected in a "Triple Separator" instance; see parts (c), (d) and (e) in Fig. 12. This can result in many false positives which are in fact correct detections in terms of visual features.

The final detection result in Table 2 represents the average value of mAP for all 35 pattern categories (mAP per category). This measurement approach allows the impact of each pattern category on the overall performance metrics to be evaluated. However, calculating the mean value of the Average Precision (AP) for every query (mAP per query) can be extremely misleading, especially for this particular dataset. The number of queries varies considerably across different categories, and only six categories contain around 70% of all the queries in the dataset. As a consequence, the mAP per query mainly represents the results from a very small number of categories rather than providing a valid estimation of the overall pattern detection performance in all categories. This fact can easily be verified in this dataset by comparing the results shown in Tables 2 and 3. We calculated the

Table 2 Performance analysis of the proposed method using the *Doc-Explore* dataset [3] for the task of pattern detection

Queries	mAP per Category
All 35 categories	0.587
The best performing 29 categories	0.700
The worst performing 6 categories	0.041

Table 3 Performance comparison for the state-of-the-art results on the *DocExplore* dataset [3] for the task of pattern detection

Method	mAP per Query
Proposed Method	**0.251**
En *et al.* 2016 [3]	0.111
En *et al.* 2016 [22]	0.157

Fig. 11 Examples of the 29 queries with the highest mAP from the *DocExplore* dataset [3]

Fig. 12 The six queries with the lowest mAP from the *DocExplore* dataset [3]

mAP per query as well in order to provide a fair comparison with the existing state-of-the-art results, but we encourage other researchers to evaluate their methods using the mAP per category for this dataset.

The proposed method achieved a state-of-the-art result for the task of pattern detection as shown in Table 3. We expect the result would be significantly higher if the ground-truth information were publicly available, meaning that a thorough performance analysis can be performed and the method can be optimised even further. The reported result has been achieved without any training or pre-processing. However, the result (mAP per query = 0.272) in [21] was obtained by using a subset of the test images to train a classifier in order to classify each page into text and non-text regions after manually annotating and labelling non-textual regions in 79 images from the test set, so this result is not considered a valid state-of-the-art outcome in the comparison in Table 3.

The aforementioned discussions and results demonstrate the generality and efficiency of our proposed method and maintain its high performance using very different datasets. These attributes exemplify the potential of our learning-free method for use as a pattern-detection tool in manuscript research.

6 Software tool implementation

An efficient and easy-to-use software tool for pattern detection has been developed, which is based on the proposed method. It provides a suitable environment for scholars to carry out tests independently and can help make many digitised manuscripts searchable. Known as the *Visual-Pattern Detector v1.0* (or VPD v1.0) [29], the software tool has already been released and made freely available for non-commercial use, similar to the software tools previously published by our research centre [30–32]. The main goal of VPD is to automatically recognise and allocate visual patterns such as words, drawings and seals in digitised manuscripts.
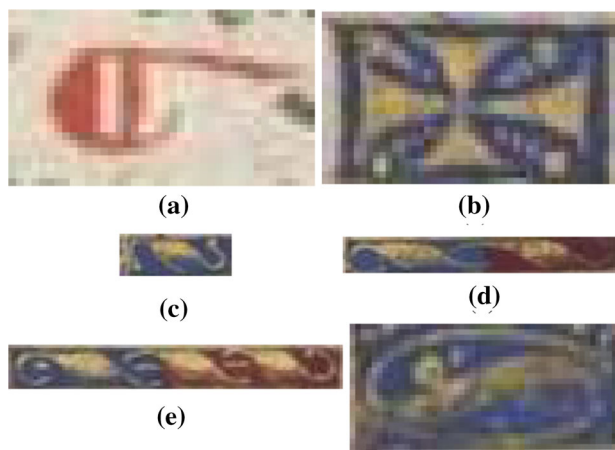
The VPD was developed as an offline Razor Pages web application using the .NET CORE platform from Microsoft (https://dotnet.microsoft.com/download/dotnet-core). It is a free software tool published under the Creative Commons Attribution-NonCommercial 4.0 International Public License. The VPD has been tested by researchers concerned with document analysis and scholars from manuscript research. A brief description of the main features is provided here, but please refer to the description in the VPD itself for more details.

The Graphical User Interface (GUI) of the VPD allows the user to perform the detection process in individual steps: selecting patterns to be detected, the images to be searched
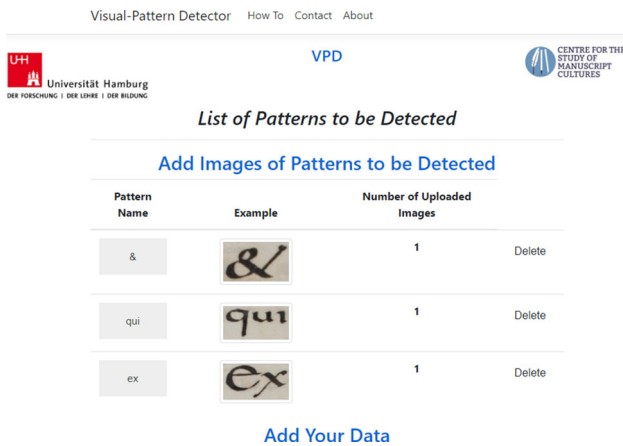
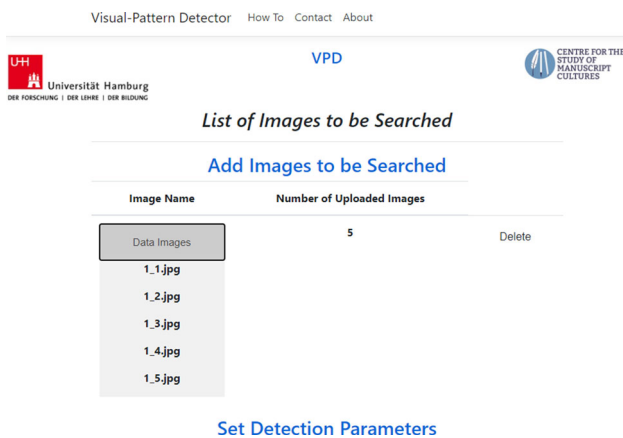**Fig. 13** The user can select pattern images from multiple folders to be detected using the VPD

Fig. 14 In the VPD, users can select images from multiple folders to be searched for the pre-selected patterns
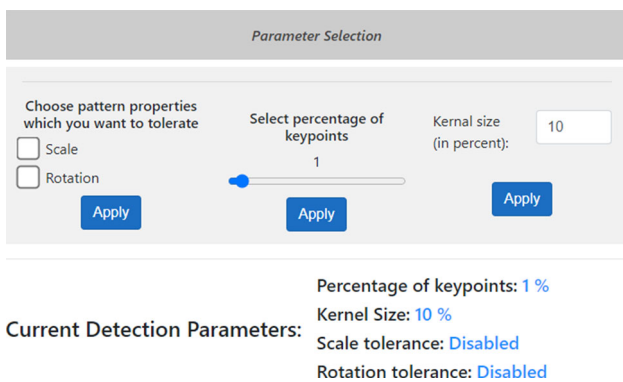

Fig. 15 In the VPD, users can change the main parameters of the method if the initial results are unsatisfactory

and finally the detection parameters; see Figs. 13, 14 and 15. The instructions for each step can be found at the bottom of the corresponding pages in the software tool. Furthermore, a general guideline is provided in the "How To" Section of the VPD.
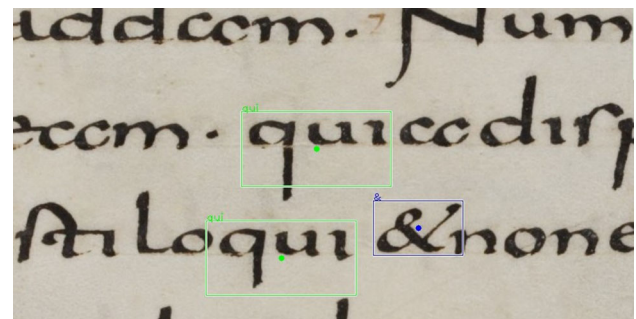
Fig. 16 Example of the results that can be generated by the VPD. The images are reproduced from the St. Gall collection kept by the "Stiftsbibliothek" library [33]
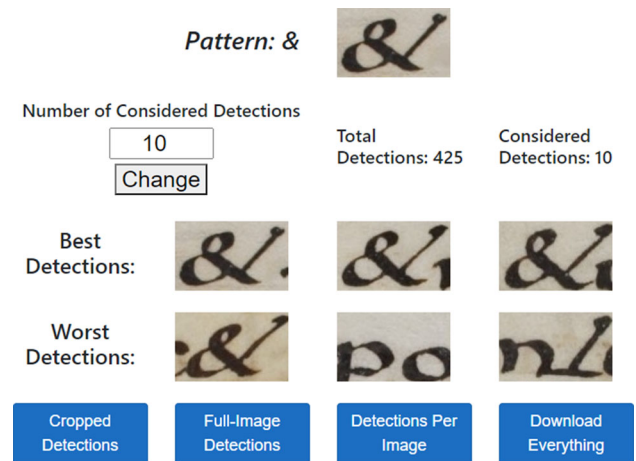
Fig. 17 A visual inspection of the best and worst detection results from the VPD can be used to determine a suitable detection threshold

The current version of the software allows users to change the main parameters of the proposed method. In addition, a limited scale and rotation invariance can be provided by creating scaled and rotated versions of the uploaded pattern images; see Fig. 15.

Finally, the detection results can be generated in a wide range of formats so that different requirements that scholars may have can be met. In addition, all detected patterns in an image can be annotated concurrently; see Fig. 16. The detection threshold can be controlled intuitively by visually inspecting the three best and worst detection results from the considered detections; see Fig. 17.

7 Conclusion

In this article, we have presented a novel, learning-free pattern-detection method for manuscript research. The proposed method is efficient and very fast, and it performs very well on very challenging manuscript images. Furthermore, this method can cope with a very wide range of degradation

in manuscript images without the need for any customised pre-processing steps.

A use case from South Asian studies was outlined in order to demonstrate the applicability of the approach to actual questions from manuscript research. In this use case, we presented a typical scenario where training data and annotations cannot be provided and a high recall rate is required.

In addition, a performance evaluation is provided in which state-of-the-art results were achieved using two relevant but very challenging datasets, namely the *AMADI_LontarSet* dataset of handwriting on palm leaves for word-spotting and the *DocExplore* dataset of medieval manuscripts for pattern detection. Since our results are the first one to be published on the first dataset, we provided three different standard evaluation metrics in order to facilitate later comparisons. As for the second dataset, we presented a comparison with the state-of-the-art results.

Achieving such high performance on very different datasets and patterns without the need for any training or fine-tuning of parameters demonstrates both the generality and feasibility of the proposed method for manuscript research.

This method was developed in order to provide a practical, automated, high performance tool that can help make many digitised manuscripts searchable for patterns such as words, seals and drawings. Therefore, the VPD software tool is developed as an easy-to-use implementation of the method and made publicly available for free.

The next step in our research is to develop an interactive learning-based method that is capable of enhancing its performance after every correct detection. Since this method requires no more than one labelled sample, the detected patterns can be employed, after being interactively validated by scholars, to further enhance the performance. Once multiple instances of the same pattern are detected, they can be used to build a generic model of that pattern.

Declaration

References

1. Han, J., Zhang, D., Cheng, G., Liu, N., Xu, D.: Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Process. Mag. **35**(1), 84–100 (2018)

2. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognit. **68**, 310–332 (2017)

3. En, S., Nicolas, S., Petitjean, C., Jurie, F., Heutte, L.: New public dataset for spotting patterns in medieval document images. J. Electron. Imaging **26**(1), 1–15 (2016). https://doi.org/10.1117/1.JEI.26.1.011010

4. Mohammed, H., Märgner, V., Stiehl, H. S.: "Writer identification for historical manuscripts: Analysis and optimisation of a classifier as an easy-to-use tool for scholars from the humanities," in 2018 16th international conference on frontiers in handwriting recognition (ICFHR), Aug 2018, pp. 534–539

5. Mohammed, H., Märgner, V., Konidaris, T., Stiehl, H. S.: "Normalised local naïve bayes nearest-neighbour classifier for offline writer identification," in 2017 14th IAPR international conference on document analysis and recognition (ICDAR). IEEE, 2017, pp. 1013–1018

6. Terzić, K., du Buf, J. H.: "An efficient naive bayes approach to category-level object detection," in 2014 IEEE international conference on image processing (ICIP). IEEE, 2014, pp. 1658–1662

7. Rosten, E., Porter, R., Drummond, T.: Faster and better: a machine learning approach to corner detection. IEEE Trans. Pattern Anal. Mach. Intell. **32**(1), 105–119 (2010)

8. Burie, J.-C., Coustaty, M., Hadi, S., Kesiman, M. W. A., Ogier, J.-M., Paulus, E., Sok, K., Sunarya, I. M. G., Valy, D.: "ICFHR2016 competition on the analysis of handwritten text in images of balinese palm leaf manuscripts," in 2016 15th international conference on frontiers in handwriting recognition (ICFHR). IEEE, 2016, pp. 596–601

9. Yarlagadda, P., Monroy, A., Carque, B., Ommer, B.: "Recognition and analysis of objects in medieval images," in Computer Vision – ACCV 2010 Workshops, R. Koch and F. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 296–305

10. Gatos, B., Pratikakis, I.: "Segmentation-free word spotting in historical printed documents," in 2009 10th international conference on document analysis and recognition, 2009, pp. 271–275

11. Konidaris, T., Kesidis, A.L., Gatos, B.: A segmentation-free word spotting method for historical printed documents. Pattern Anal. Appl. **19**(4), 963–976 (2016)

12. Frinken, V., Fischer, A., Manmatha, R., Bunke, H.: A novel word spotting method based on recurrent neural networks. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 211–224 (2012)

13. Ahmed, R., Al-Khatib, W.G., Mahmoud, S.: A survey on handwritten documents word spotting. Int. J. Multim. Inf. Retr. **6**(1), 31–47 (2017)

14. Rusiñol, M., Aldavert, D., Toledo, R., Lladós, J.: Efficient segmentation-free keyword spotting in historical document collections. Pattern Recognit. **48**(2), 545–555 (2015)

15. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character hmms. Pattern Recognit. Lett. **33**(7), 934–942 (2012)

16. Vats, E., Hast, A., Fornés, A.: "Training-free and segmentation-free word spotting using feature matching and query expansion," in 2019 international conference on document analysis and recognition (ICDAR). IEEE, 2019, pp. 1294–1299

17. Dutta, A., Lladós, J., Pal, U.: A symbol spotting approach in graphical documents by hashing serialized graphs. Pattern Recognit. **46**(3), 752–768 (2013)

18. Le, V.P., Nayef, N., Visani, M., Ogier, J.-M., Tran, De, C.: "Document retrieval based on logo spotting using key-point matching," in, : 22nd international conference on pattern recognition. IEEE **2014**, 3056–3061 (2014)

19. Rusiñol, M., Lladós, J.: "Word and symbol spotting using spatial organization of local descriptors," in, : The Eighth IAPR international workshop on document analysis systems. IEEE **2008**, 489–496 (2008)

20. Wiggers, K. L., Britto, A. S., Heutte, L., Koerich, A. L., Oliveira, L. S.: "Image retrieval and pattern spotting using siamese neural network," in 2019 international joint conference on neural networks (IJCNN). IEEE, 2019, pp. 1–8

21. Úbeda, I., Saavedra, J.M., Nicolas, S., Petitjean, C., Heutte, L.: Improving pattern spotting in historical documents using feature pyramid networks. Pattern Recognit. Lett. **131**, 398–404 (2020)

22. En, S., Petitjean, C., Nicolas, S., Heutte, L.: A scalable pattern spotting system for historical documents. Pattern Recognit. **54**, 149–161 (2016)

23. Ciotti, G., Franceschini, M.: Certain times in uncertain places: a study on scribal colophons of manuscripts written in Tamil and Tamilian Grantha scripts. Studi. Manuscr. Cult. **7**, 59–129 (2016)

24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)

25. Boiman, O., Shechtman, E., Irani, M.: "In defense of nearest-neighbor based image classification," 2008 IEEE Conf. on computer vision and pattern recognition, pp. 1–8, 2008

26. McCann, S., Lowe, D. G.: "Local Naive Bayes Nearest Neighbor for image classification," 2012 IEEE Conf. on computer vision and pattern recognition, pp. 3650–3656, Jun. 2012

27. Muja, M., Lowe, D. G.: "Fast approximate nearest neighbors with automatic algorithm configuration." in VISAPP (1), 2009, pp. 331–340

28. Ahmed, R., Al-Khatib, W.G., Mahmoud, S.: A survey on handwritten documents word spotting. Int. J. Multim. Inf. Retr. **6**(1), 31–47 (2017)

29. Mohammed, H.: "Visual-Pattern Detector v1.0 (VPD V1.0)," Feb. 2021. [Online]. Available: https://doi.org/10.25592/uhhfdm.8832

30. Mohammed, H.: "Handwriting Analysis Tool v3.0 (HAT3)," Feb. 2020. [Online]. Available: https://doi.org/10.25592/uhhfdm.902

31. Mohammed, H.: "X-Ray Fluorescence Data Analysis Tool v1.3 (XRF-DAT1.3)," Jun. 2020. [Online]. Available: https://doi.org/10.25592/uhhfdm.1037

32. Mohammed, H.: "Line Detection Tool v1.0 (LDT1)," Jun. 2020. [Online]. Available: https://doi.org/10.25592/uhhfdm.1043

33. e-codices Virtual Manuscript Library of Switzerland. St. gallen, stiftsbibliothek. [Online]. Available: http://www.e-codices.ch