Samir Malakar*, Manosij Ghosh, Ram Sarkar and Mita Nasipuri

Development of a Two-Stage Segmentation-Based Word Searching Method for Handwritten Document Images

https://doi.org/10.1515/jisys-2017-0384
Received July 30, 2017; previously published online July 4, 2018.

Abstract: Word searching or keyword spotting is an important research problem in the domain of document image processing. The solution to the said problem for handwritten documents is more challenging than for printed ones. In this work, a two-stage word searching schema is introduced. In the first stage, all the irrelevant words with respect to a search word are filtered out from the document page image. This is carried out using a zonal feature vector, called pre-selection feature vector, along with a rule-based binary classification method. In the next step, a holistic word recognition paradigm is used to confirm a pre-selected word as search word. To accomplish this, a modified histogram of oriented gradients-based feature descriptor is combined with a topological feature vector. This method is experimented on a QUWI English database, which is freely available through the International Conference on Document Analysis and Recognition 2015 competition entitled "Writer Identification and Gender Classification." This technique not only provides good retrieval performance in terms of recall, precision, and F-measure scores, but it also outperforms some state-of-the-art methods.

Keywords: Word searching, HOG feature, topological feature, holistic word recognition, handwritten documents, QUWI database.

1 Introduction

Handwriting is an age-old and standard way of communication. Before the invention of modern technologies for communication, like printing press, typewriter, and email, communication was predominantly made via handwriting. Even in this era of advanced technology, a large number of people still prefer the ancient way of communication, which leads to an exponential growth of handwritten documents. Due to mismanagement, the quality of the document degrades with each passing day, which implies that preservation of documents is a pressing need. In addition, manual searching for some important information from a huge repository is time consuming and at times also error prone. Sometimes, due to lack of proper maintenance, some important document(s) may be misplaced or lost. All these issues are posing genuine problems for proper storage and maintenance of handwritten document images. Apart from these facts, advanced office automation demands mechanized storage, manipulation, and retrieval of documents in electronic format, i.e. handwritten documents need to be managed properly. The undebatable solution of this is to convert document images into an electronic form and then process the same with an optical character recognition (OCR) engine [4]. However, current handwritten OCR engines work poorly for large lexicon sizes [7]. Therefore, the alternative solution is to keep the documents in well-indexed digital form with appropriate tagging. One of the possible ways of tagging is keyword-based document indexing, which is an enriched research problem. It has a large number of real-life applications [13, 15, 25].

***Corresponding author: Samir Malakar,** Department of Computer Science, Asutosh College, Kolkata, India, e-mail: malakarsamir@gmail.com
Manosij Ghosh, Ram Sarkar and Mita Nasipuri: Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

The pre-requisite for any keyword-based document indexing is identifying the keywords from documents (also known as word/keyword spotting in the literature). A word searching methodology tries to locate the occurrence(s) of some pre-defined words inside a document image. It is worthy to mention that searching a word in a handwritten document [35] poses more challenges than searching the word from its printed counterpart [23]. One of the key factors for this is the variation in writing styles, not only among different writers but also for the same writer. Even some typical problems related to handwriting, like skew, slant, overlapping, and/or joining of words/characters, add more complexity.

1.1 Categorization of Word Searching Methods

A typical way of categorizing word searching techniques is based on how query words are fed into the system. Depending on this, existing techniques are classified into two categories, namely query-by-example (QBE) [13, 15] and query-by-string (QBS) [11, 13]. In the former category, a word image is provided to the system and the system returns occurrence(s) of that given word in the document image. Methods of this approach have mostly relied on image matching techniques. On the other hand, the techniques that follow the QBS approach consider an arbitrary word as a string to be searched from a document image. Therefore, methods of this approach, in general, try to follow some word recognition model. Word searching techniques can also be categorized as segmentation based [38] and segmentation free [27]. Segmentation-based word spotting techniques have relied on pre-segmented text lines [12, 22, 38] or words [16, 22], which may be achieved by using text-line and/or word-level ground truth (GT) images. The other category of techniques [26, 27] tries to locate the words to be searched in document images without spending time for page segmentation. Use of word samples, collected in offline mode for preparing the training module, is found in recognition-based approaches [1]. On the other hand, recognition-free approaches [20, 21] try to spot search words using some matching techniques. The present work is an instance of a recognition-based approach that uses a holistic word recognition technique.

1.2 Literature Survey

From the literature survey, it is observed that, to date, many researchers have applied several distance-based methods [8, 22, 29, 36] for searching a word from a document image following the QBE way. In Ref. [22], in order to handle the word searching problem, gradient angle and its magnitude were extracted from word images to search query words using a disc-based matching schema. The work in Ref. [29] reports a word searching mechanism that uses scale-invariant feature transform (SIFT) features from word images, and search is confirmed by cosine and Euclidean distance measures. Whereas the work described in Ref. [8] extracted Gabor feature and then used Euclidean distance-based similarity check for word searching in the QBE way. Pixel value information was used in Ref. [36], where word searching in the QBE way was carried out by using a Bray-Curtis dissimilarity measure.

In another category of methods [20, 21, 30], several string matching approaches have been used for finding a solution to the said problem. The work described in Ref. [20] introduced a flexible sequence matching technique for comparing query word and target word. In this work, eight feature values, extracted from each column of the image, were used. The same set of feature values were used in Ref. [21] for investigating the effectiveness of some conventional time series matching techniques for word searching. A multi-angular feature descriptor was used in Ref. [30], where word images are represented by a variable length feature vector. Dynamic time warping (DTW) was used for matching the words.

Graph similarity-based methods [24, 37, 38] are also found in the literature for word spotting. Authors in Refs. [37, 38] used a graph edit distance-based similarity score for word spotting. In Ref. [38], each word image was represented as a sequence of skeleton-based graphs using context-labeled vertices for connected components (CCs). A similar approach is found in Ref. [37]. The only difference is the way the calculation of the graph edit distance is done, i.e. in Ref. [37] bipartite graph matching schema was used, while in Ref. [38] DTW alignment method was used. In Ref. [24], attributed graphs have been constructed using the graphemes that were extracted from words by a part-based approach. It employed an edit distance-based similarity measure.

Classical word recognition models were used in Refs. [11, 25, 36] for performing word spotting in QBS. A statistical framework for the word spotting problem, introduced in Ref. [25], has explored the use of two types of hidden Markov models (HMMs), namely continuous HMM and semi-continuous HMM. Geometrical features were extracted in Ref. [11] for performing word spotting. HMM-based character-level trained model was applied for finding the similarity score in this work. In Ref. [36], a pyramidal histogram of characters feature vector was extracted for performing word searching. The Bray-Curtis dissimilarity measure between the words was measured for performing word similarity tasks.

Researchers have also attempted to devise techniques [22, 26–28] where they segment a word into characters or character sub-parts. In Ref. [28], a patch-based segmentation-free framework was introduced. The framework used a bag-of-features (BoF)-based HMMs (BoF-HMM) model, whereas the authors of Ref. [27] used a SIFT feature descriptor for recognition purpose in their BoF-HMM model. An approach similar to that of Ref. [27] was proposed in Ref. [26]. In this case, the model was developed to search any arbitrary word and does not require any pre-segmentation of document pages. A semi-supervised handwritten word recognition-based technique using bi-directional long short-term (BLSTM) neural network (NN) model was presented in Ref. [22].

A holistic word recognition paradigm was used in Refs. [1, 14] for retrieval of search words from historical Arabic handwritten manuscripts. In Ref. [1], several structural and statistical features, extracted from connected parts of word images, are fed to a multi-layer perceptron (MLP)-based classifier for preparing a learning module. In this work, it was shown that features extracted from connected parts of the word perform well in comparison with features extracted from the entire word. The work reported in Ref. [14] introduced a hierarchical classifier, comprising support vector machine and regularized discriminant analysis for searching a word. A gradient-based feature vector is used for discriminating the words in feature space.

1.3 Word Recognition Methodologies

Word recognition is a process of realizing word images as a machine-editable form. Researches on word recognition, as found in the literature, follow either an analytical approach [4, 11, 27] or a holistic approach [6, 7, 18, 31]. In the analytical approach, a word image is represented as a collection of units known as characters and/or character sub-parts. These units are first recognized, and then the recognized units are compiled to form a machine-encoded word. It is certainly the best way of word recognition; however, it suffers from several segmentation ambiguities [7].

Consequently, researchers often use lexicon-driven character (or character-like shapes)-based model for word recognition, such as HMM [11, 25] and BLSTM-based NN [22]. These approaches allow us to obtain good performance when the learning set is well representative of the writing/font styles that are found in dataset. However, the main drawback of such approaches is the usual requirement of an enormous set of transcribed text line/word images for training. Such requirement could be costly and often needs manual effort. At times, this approach is designed focusing on some particular language. Apart from these, it takes a huge time to develop the learning module.

On contrary, in the holistic approach, a word is considered as a single and indivisible unit. This approach is, in general, segmentation free; hence, it can avoid the problems that occur while using a segmentation-based or analytical approach. As the features are extracted from the entire word image, this approach provides acceptable recognition accuracy irrespective of script [31]. Based on these facts, it can be concluded that a holistic approach can be a suitable choice for word matching in recognition-based word spotting schema, as the number of query words, in general, are pre-defined and smaller in size.

1.4 Motivation

Literature survey reveals that mainly two different approaches, viz. recognition free [22, 36] and recognition based [12, 28], are followed by researchers for providing solution to the word matching problem. The first category of works extracts geometrical/shape context characteristics of word image (e.g. [24, 29, 30, 36]) and then

applies some similarity measure for word matching. Although this mechanism is fast, it generally retrieves more irrelevant words with respect to the actual search word. On the contrary, matching techniques as applied in recognition-based methods (e.g. [1, 11, 14, 27]) have tried to identify a query word in the document images by recognizing all the words present therein. Therefore, these mechanisms not only suffer from a high time requirement but also retrieve more irrelevant words.

Considering the above facts, in the present work, a two-stage word searching technique is designed. This work employs a pre-selection schema that uses the benefits of the recognition-free word spotting approach and a final matching schema that relies on the recognition-based model. Here, a holistic paradigm is used for matching a query word with the target word. The experiment has been conducted on a handwritten English document page database, called "QUWI" database [2], which was also used in the writer identification competition in the International Conference on Document Analysis and Recognition (ICDAR) 2015 [10].

2 Present Work

Any word searching mechanism, in general, comprises two major steps, namely (i) page segmentation (text lines and/or words are extracted from document images) and (ii) word matching schema (i.e. confirming a target word present in a document image as the word to be searched). A number of works, found in the literature, have used GT images containing either pre-segmented text line [12] or words [22] for localizing words within a document page. Such use of GT images helps in avoiding several inherent errors occurring due to page segmentation [17, 32, 34]. Inspired by this fact, here, document-level GT images containing pre-segmented words have been prepared using GTgen 1.1 [33] software. The details of document-level GT image preparation are described in Section 2.2.

A two-stage approach, consisting of pre-selection of target words as words to be searched and confirmation of pre-selected word(s) as search word, has been introduced for searching a word from document page images. A feature vector (say $\mathfrak{F}_1$) of length 3 has been extracted from all the words present in a document image to filter out irrelevant words with respect to the word to be searched (refer to Section 2.3.1 for a detailed description). To confirm the remaining word(s) in the document page as expected search word(s), a holistic word recognition approach has been used. For this, a feature vector (say $\mathfrak{F}_2$), comprising topological features and a modified histogram of oriented gradients (HOG) feature descriptor, is extracted from each word image, which is then classified using an MLP classifier. The word searching module is described in Section 2.3, while the overall working procedure of the proposed word searching technique is depicted in Figure 1.

2.1 Data Preparation

The pre-requisite for developing a word searching algorithm is to build a collection of handwritten document images written by different persons. In the present work, the QUWI database [2], which is partially public through different competitions, has been preferred. Such preference is significant because of its number and diversity (e.g. nationality, age, background, etc.) of the writers and variations in writing materials (i.e. colors and thicknesses of pen/pencil): 300 handwritten document images for each of the scripts in English and Arabic, written by 300 different writers [≈14.75% and ≈29.5% of entire document images, with respect to number of categorical (script) document images and writers, respectively] are uploaded for public use through the ICDAR 2015 competition on multi-script writer identification and gender classification [10]. These document pages contain the same text with 117 words (in most of the cases). It is to be noted that some of these document images contain fewer number of words (the minimum number of words, as observed, is 99). Moreover, wrongly spelled words, use of abbreviated forms, and different spellings from the original words definitely have added huge complexities while considering searching some words therein. The publicly available document images written in English are considered here. The document pages are divided in a 1:5 ratio. The first set is used for evaluation of the present word searching algorithm, while the rest of the document page images have been used for confirming values of different parameters required to design the searching algorithm.

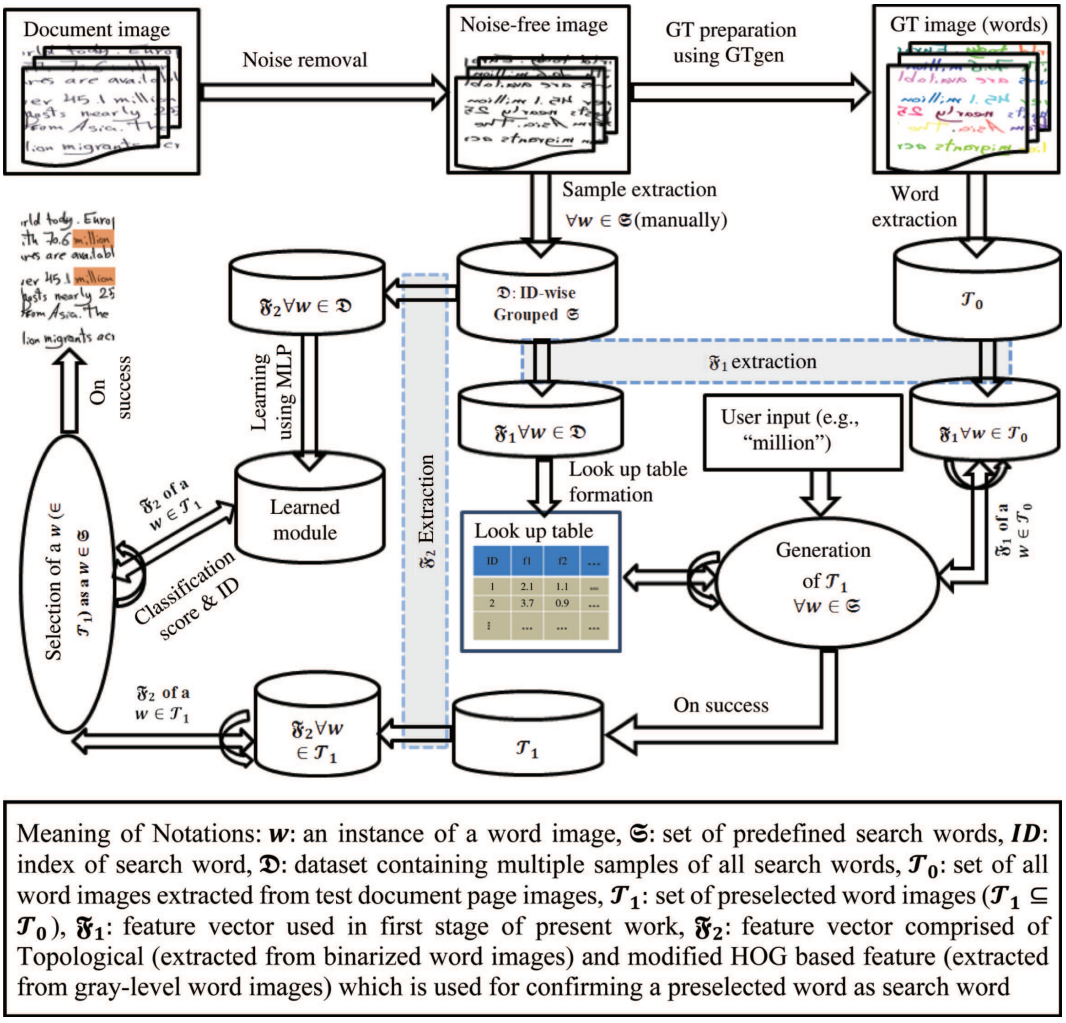


Figure 1: Block Diagram of the Present Word Searching Technique.

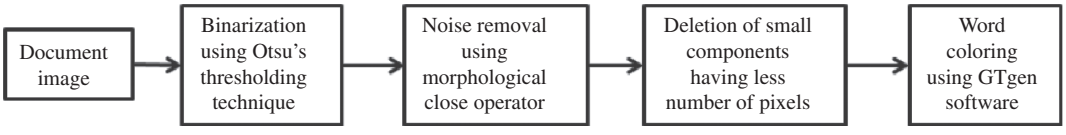


Figure 2: Block Diagram of GT Image Preparation.

2.2 GT Image Preparation

In the present work, instead of applying any word extraction technique, GT images representing ideal word boundaries are prepared. A block diagram representing the various stages of GT image generation for each document page is depicted in Figure 2. At first, document images are binarized using Otsu's thresholding technique. Next, a morphological close operator using a 5×5 mask is applied on binarized document images. Finally, noise-free document images are generated by deleting CCs having $\leq$5 data pixels (see Figure 3A,B).

To prepare a GT page image, all the words in it are colored using GTgen1.1 [33], a GT-generating tool. This tool can set uniform color for all data pixels (appearing as CCs or disconnected components) within a selected region. This feature helps in coloring words in a document page manually and takes less time than MS paint tool. An instance of GT images is shown in Figure 4.

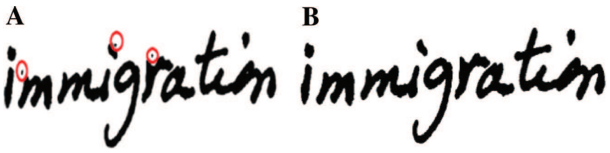


Figure 3: Word Instances (A) Before and (B) After Deletion of Small Components.

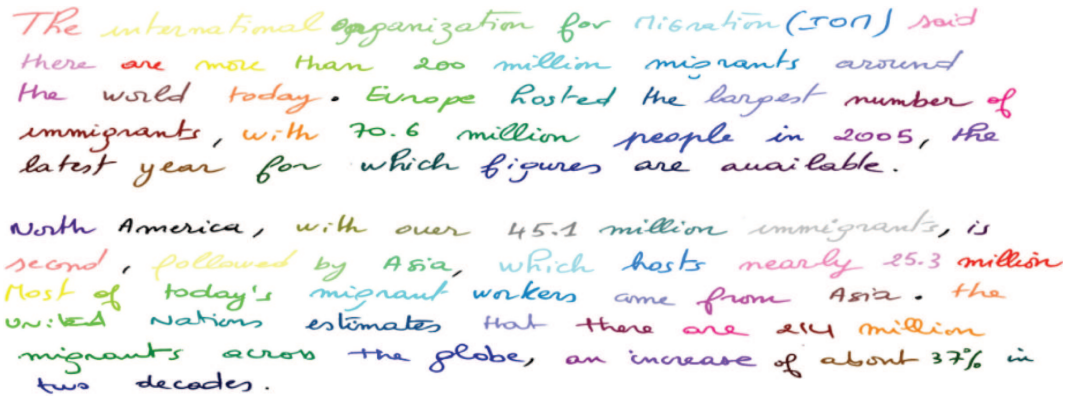


Figure 4: An Instance of GT Image Where Words are Colored Differently.
Here, neighboring words belonging to same text line, or words belonging to successive text line (below or above), are painted with different colors.

2.3 Word Searching

It has already been mentioned that, here, a two-stage approach is designed for searching a given word in a document page. In the first stage, irrelevant words (with respect to a search word) from a document image are filtered out. The remaining words present in a document page are termed as pre-selected words with respect to given query words. Such selection process is shown in Figure 5A. In the second stage, the pre-selected words are confirmed as the given query image (see Figure 5B). This process is carried out using a holistic word recognition method. The details of these stages are given in the following subsections.

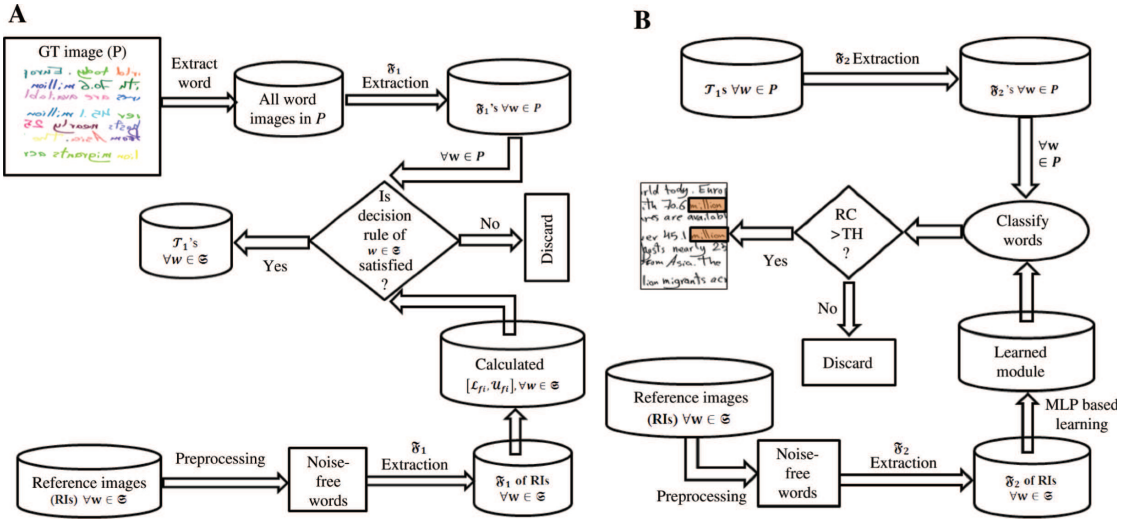


Figure 5: Schematic Diagrams.
(A) For pre-selecting probable candidate words in a document page that are relevant to a given search word and (B) for detecting instance(s) of the search word from its pre-selected candidate word set. Meaning of Notations: RC, Recognition confidence; TH, a predefined threshold value which is set here as 0.75. Other symbols are already described in Figure 1.

2.3.1 Pre-selection of Probable Candidate Words for a Particular Search Word

Words in a document, in general, have varying numbers of characters. In this scenario, searching a word directly in a document image increases the chances of mismatch as well as enhances the computation cost. Sometimes, it leads to selection of unnecessary words as search word. That is why, in the present work, words having a different number of characters than the number of characters in the search word are not considered during the search process. Rather, a rule-based method is followed for the pre-selection of probable candidate words.

2.3.1.1 $\mathfrak{F}_1$ Extraction

Let a binarized word image be represented as $B = \{f(i,j) : 1 \leq i \leq H \wedge 1 \leq j \leq W\}$, where H and W are height and width of B, respectively, and $f(i,j) \in \{0, 1\}$ (0 and 1 represent non-data and data pixels, respectively). In this step, first, B is hypothetically segmented into three non-overlapping horizontal regions, viz. upper, middle, and lower zones (see Figure 6), for feature extraction. This figure also shows the four horizontal lines, viz. $R1$, $R2$, $R3$, and $R4$, to distinguish the zones horizontally. The lines $R1$ and $R4$ are estimated using Eqs. (1) and (2), respectively:

$$R1 = \min\{i : f(i,j) = 1 \wedge (i,j) \in [1,H]\times[1,W]\}. \tag{1}$$

$$R4 = \max\{i : f(i,j) = 1 \wedge (i,j) \in [1,H]\times[1,W]\}. \tag{2}$$

However, identification $R2$ and $R3$ is not straightforward. For identifying these lines, first, the number of horizontal transition points between data and non-data pixels or vice versa along each row of B is calculated. Such number along a horizontal row i ($\mathcal{T}_i$, $i \in [1, H]$) is calculated by

$$\mathcal{T}_i = |\{j : ((f(i,j) = 1 \wedge f(i,j+1) = 0) \vee (f(i,j) = 0 \wedge f(i,j+1) = 1)) \wedge j \in [1, W-1]\}|. \tag{3}$$

The mean of all such transition point counts (μ_{TP}) of B is estimated by

$$\mu_{TP} = \frac{1}{N}\sum_{i=1}^{H} \mathcal{T}_i, \text{ where } N = |\{i : \mathcal{T}_i \neq 0\}|, \ i \in [1, H]. \tag{4}$$

Now, $R2$ and $R3$ are calculated as follows:

$$R2 = \min_{i=1,2,\ldots,H} \{i : \mathcal{T}_i > \mu_{TP}\}. \tag{5}$$

$$R3 = \max_{i=1,2,\ldots,H} \{i : \mathcal{T}_i > \mu_{TP}\}. \tag{6}$$

Finally, the said $\mathfrak{F}_1(= (f1, f2, f3))$ is estimated by Eqs. (7)–(9):

$$f1 = \frac{1}{M}\sum_{i=R2}^{R3} \mathcal{T}_i, \text{ where } M = |\{i : \mathcal{T}_i \neq 0\}|, \ \forall i \in [R2, R3]. \tag{7}$$

Figure 6: Partitioning of a Word Image into Zones.

$$f2 = |\{C : \theta(C) = 1\}|. \tag{8}$$

$$f3 = |\{C : \varnothing(C) = 1\}|. \tag{9}$$

In Eqs. (8) and (9), C represents CC. Now, let $\theta(.)$ and $\varnothing(.)$ be functions that represent the belongingness of a CC in upper/lower zone, respectively, which are defined by

$$\theta(C) = \begin{cases} 0, \text{ if } \min\{i : f(i,j) \in C\} \leq \frac{R1+R2}{2} \text{ and } \max\{i : f(i,j) \in C\} = R2 - 1 \\ 1, \text{ Otherwise} \end{cases}. \tag{10}$$

$$\varnothing(C) = \begin{cases} 0, \text{ if } \max\{i : f(i,j) \in C\} \leq \frac{R3+R4}{2} \text{ and } \min\{i : f(i,j) \in C\} = R3 + 1 \\ 1, \text{ Otherwise} \end{cases}. \tag{11}$$

2.3.1.2 Decision Rule Formulation

To filter out words that are not relevant with respect to a given search word, here, a decision rule has been designed. For this, first, decision boundaries (lower and upper bounds) for each of the extracted feature values (i.e. $f1$, $f2$, and $f3$) are estimated. These decision boundaries are set by considering the mean (μ_{fi}, $i = 1, 2, 3$) and standard deviation (σ_{fi}, $i = 1, 2, 3$) of feature values extracted from manually chosen N samples for a search word. Let $\mathcal{L}_{fi}$ and $\mathcal{U}_{fi}$ be the lower and upper bounds of feature value fi ($i = 1, 2, 3$), respectively, which are defined by

$$\mathcal{L}_{fi} = \mu_{fi} - \sigma_{fi}, \text{ where } i = 1, 2, 3. \tag{12}$$

$$\mathcal{U}_{fi} = \mu_{fi} + \sigma_{fi}, \text{ where } i = 1, 2, 3. \tag{13}$$

Finally, a word is pre-classified as a probable candidate for the given search word by the decision rule as depicted in Figure 7.

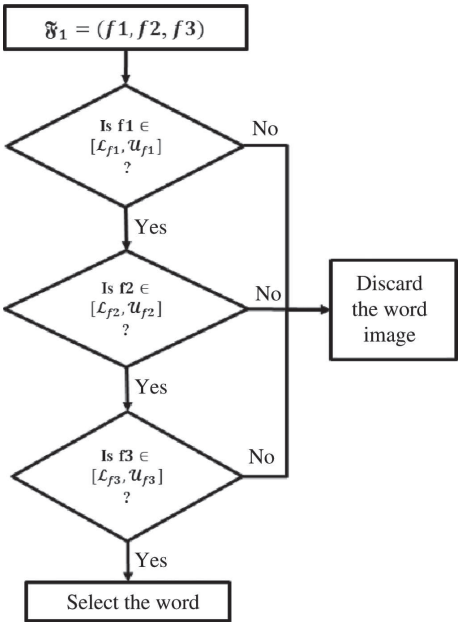


Figure 7: Diagrammatic Representation of the Selection Process of a Word Image as a Probable Candidate Word Using a Rule-Based Mechanism.

2.3.2 Confirming a Pre-selected Candidate Word as Search Word

For recognition of the keywords, a feature vector (i.e. $\mathfrak{F}_2$), comprising topological [18] and texture-based [3, 5] features, is extracted from both search and target word images.

2.3.2.1 Topological Feature

Topological features are extracted either from the entire word image (for extracting global information) or from sub-images (for extracting local information). For calculations of features from sub-images, word images are divided into an $m \times n$ grid.

Area: Areas covered by the different patterns (here, query and target words) vary a lot because of the presence of different numbers of characters and also for shape variations of these characters therein. The area of B is calculated by

$$\text{Area} = ((R4 - R1 + 1) * (C2 - C1 + 1)). \tag{14}$$

Values of $R1$ and $R4$ are calculated using Eqs (1) and (2), respectively. Values of other parameters, i.e. $C1$ and $C2$, are estimated by

$$C1 = \min\{j : f(i,j) = 1 \wedge (i,j) \in [1,H] \times [1,W]\}. \tag{15}$$

$$C2 = \max\{j : f(i,j) = 1 \wedge (i,j) \in [1,H] \times [1,W]\}. \tag{16}$$

Area-based feature values are extracted from sub-images only. Therefore, the total number of area features ($\mathcal{L}_A$) extracted from an image can be represented by

$$\mathcal{L}_A = m * n. \tag{17}$$

Pixel density (PD): PD represents the number of data pixels per unit area. It is calculated by

$$\text{PD} = \frac{|\{(i,j) \ : \ f(i,j) = 1 \wedge (i,j) \in [1,H] \times [1,W]\}|}{\text{Area}}. \tag{18}$$

PD is extracted from sub-images only. Therefore, a feature vector of dimension $m*n$ is generated, i.e.

$$\mathcal{L}_{PD} = m * n. \tag{19}$$

Here, $\mathcal{L}_{PD}$ is the length of pixel density feature vector, extracted from an image.

Aspect ratio (AR): The AR of an image is defined as ratio of width to height that generally differs when word images contain different number of characters. So, this feature is also considered here and the feature value is calculated as

$$\text{AR} = \frac{(C2 - C1 + 1)}{(R2 - R1 + 1)}. \tag{20}$$

AR is extracted from the original image and sub-images. Therefore, the total number of ARs ($\mathcal{L}_{AR}$) extracted from a word image is represented by

$$\mathcal{L}_{AR} = m * n + 1. \tag{21}$$

Longest run: The longest run feature is considered here as another topological feature. This feature consists of two feature values, viz. lengths of the longest run along X and Y directions, respectively. Let B has $N_i(\geq 1)$ number of runs (occurrence of continuous data pixels) along i^{th} row and HL_{ij} represents the length of j^{th} run along i^{th} row. The horizontal longest run (HLR) is calculated as

$$\text{HLR} = \max_i\{\max_j\{HL_{ij}\}\}, \ 0 \leq j \leq N_i, \ R1 \leq i \leq R2. \tag{22}$$

The vertical longest run (VRL) feature is computed by identifying runs in a column-wise manner. Let B has $N_i(\geq 1)$ number of runs along i^{th} column and VL_{ij} represents the length of j^{th} runs along i^{th} column. Now, the VLR of B is measured as

$$\text{VLR} = \max_i\{\max_j\{VL_{ij}\}\},\ 0 \leq j \leq N_i,\ C1 \leq i \leq C2\ . \tag{23}$$

These feature values are extracted from actual word image and sub-image structures. Therefore, a feature vector of length $2*(m*n+1)$ is generated, which means the total number of longest run features ($\mathcal{L}_{LR}$) is defined by

$$\mathcal{L}_{LR} = 2*(m*n+1)\,. \tag{24}$$

Centroid: The centroid feature is based on the center of gravity of an image. This position varies depending on the shape of the pattern. Centroid coordinates (C_X, C_Y) are calculated as

$$C_X = \frac{\sum^i}{|\{(i,j) : f(i,j) = 1 \wedge (i,j) \in [1,H]\times[1,W]\}|}. \tag{25}$$

$$C_Y = \frac{\sum^j}{|\{(i,j) : f(i,j) = 1 \wedge (i,j) \in [1,H]\times[1,W]\}|}. \tag{26}$$

These two features are extracted (in a global way) from an entire image and from upper and lower parts of an image, which are separated by principal and non-principal diagonals (see Figure 8A,B). Therefore, the number of features generated from an image is $2*(1+2+2) = 10$. Also, all these features are again extracted from all the sub-images for obtaining local information. Therefore, the total number of centroid features ($\mathcal{L}_C$) can be formulated as

$$\mathcal{L}_C = 10*(m*n+1)\,. \tag{27}$$

Projection length: Projection lengths of word image on the principal axes, i.e. on X and Y axes, are considered here as feature values. Let $\mathcal{R}$ and $\mathcal{C}$ be sets of row index and column index, respectively, that contain at least 1 data pixel and are defined as

$$\mathcal{R} = \{i : f(i,j) = 1 \wedge (i,j) \in [1,H]\times[1,W] \text{ for some } j \in [C1,\ C2]\}\,. \tag{28}$$

$$\mathcal{C} = \{j : f(i,j) = 1 \wedge (i,j) \in [1,H]\times[1,W] \text{ for some } i \in [R1,\ R4]\}\,. \tag{29}$$

Now the horizontal projection length (HPL) and vertical projection length (VPL) are defined as

$$\text{HPL} = |\mathcal{R}|\,. \tag{30}$$

$$\text{VPL} = |\mathcal{C}|\,. \tag{31}$$

These two features are extracted from the entire image and from upper and lower parts of the image, which are separated by principal and non-principal diagonals (see Figure 8A,B). Therefore, using this projection length-

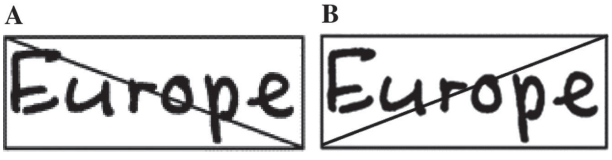


Figure 8: Showing Hypothetically Generated Sub-images by Segmenting an Image Along (A) Principal and (B) Non-principal Diagonals.

based concept, the total number of global features generated from such images is $2*(1+2+2)=10$, i.e. the total number of projection length features ($\mathcal{L}_{PL}$) can be written as

$$\mathcal{L}_{PL}=10. \tag{32}$$

By accumulating all of the above-mentioned features, the length of the topological feature becomes

$$\begin{aligned}&\mathcal{L}_S=\mathcal{L}_A+\mathcal{L}_{PD}+\mathcal{L}_{AR}+\mathcal{L}_{LR}+\mathcal{L}_C+\mathcal{L}_{PL}\\&\Rightarrow \mathcal{L}_S=m*n+m*n+(m*n+1)+2\,(m*n+1)+10*(m*n+1)+10\\&\therefore \mathcal{L}_S=15*m*n+23.\end{aligned} \tag{33}$$

2.3.2.2 Texture-Based Feature

HOG [9] is a texture-based feature descriptor that was proven to be useful in many pattern recognition problems. It firstly computes gradient information of a cell (a primitive sub-block) in different directions and, secondly, normalization of cell information is done for each block to describe a pattern. The orientation measurement on each cell is performed by dividing the total orientation angle (0°–360° or 0°–180°) into different ranges (commonly it is 8, 9, 16, etc.). Each of these angle ranges is called a "bin." It needs a size-normalized image to obtain the fixed length of a feature vector. Such resizing not only strives to compromise with resolution but also destroys the AR of images, which is a discriminative feature while considering the word recognition problem.

The HOG feature, as described in Ref. [19], is extracted into b bins from an image by considering $c\times c$ cell size and $2c\times 2c$ block size. Before extracting this feature, images are padded with zeros to convert their dimension in multiples of c. Next, the mean and standard deviation of the extracted information in each of the b bins are considered here as feature values. This implies that a feature vector of length $2*b$ is generated for every image. This modification not only helps to generate a feature vector of equal length by preserving the actual size of the image but is also capable of generating feature vectors of lesser dimension. Here, to get local information of a word image, these HOG feature values are again extracted from sub-images, generated by diving an image into a $p\times q$ grid. Therefore, the total length of HOG feature vector ($\mathcal{L}_H$) is formulated as

$$\mathcal{L}_H=2*b*(p*q+1)\,. \tag{34}$$

Therefore, the length of $\mathfrak{F}_2$ ($\mathcal{L}$) is calculated as below:

$$\begin{aligned}\mathcal{L}=\mathcal{L}_S+\mathcal{L}_H&=15*m*n+23+2b*(pq+1)\\&=2*b*p*q+15*m*n+2*b+23.\end{aligned} \tag{35}$$

Inspired by the work in Ref. [19], the value of b is set as 9. Also, considering the work as reported in Ref. [18], values of m and n are kept the same, i.e. $m=n$. Therefore, Eq. (35) can be written as

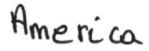
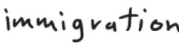
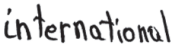
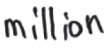
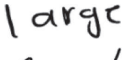
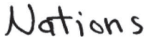
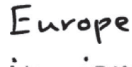
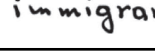
$$\mathcal{L}=15*m^2+18*p*q+41. \tag{36}$$

The values of other parameters (i.e. m, p, and q) are decided experimentally.

2.3.2.3 Confirmation Process

For identifying the search word from pre-selected candidate words, $\mathfrak{F}_2$ is extracted from the manually chosen N samples of all the search words. Next, these features are fed to an MLP-based classifier, which returns a learned module. Next, all the pre-selected candidate words for a given search word are passed through this learned module. The system provides a recognition confidence (say, RC) for all of these words. Finally, a pre-selected candidate word is confirmed as the said search word if RC $\geq$ TH. TH is a pre-defined threshold value, which is here chosen as 0.75.

Table 1: Instances of Search Words.

SW ID	Sample Image	WF	SW ID	Sample Image	WF	SW ID	Sample Image	WF
SW01		2	SW06		2	SW11		2
SW02		2	SW07		1	SW12		5
SW03		1	SW08		1	SW13		1
SW04		2	SW09		1	SW14		1
SW05		2	SW10		1	SW15		1

WF represents the number of occurrences of the corresponding word in a document page, and SW ID stands for search word index.

3 Results and Discussion

In the present work, a process of word searching from handwritten document images is reported. For this purpose, a two-stage approach, discussed earlier, is introduced. In the following subsection, issues related to the experimental outcome are described.

3.1 Selection of Search Word Set

It has already been mentioned that the QUWI database has been chosen for carrying out the experiment. For evaluation of the designed word searching technique, a set of 15 words is selected from document images as query words (see Table 1). The total number of occurrences of these words in a document page is 25 (around 21% of the total words in an ideal document page). The choice of such words to be searched from documents is mainly based on multiple occurrences of words in the document pages. Along with this, word pairs that are almost indistinguishable in terms of shape (e.g. “today” and “today’s”), stemming words (e.g. “Asia” and “Asian” or “migrant” and “migrants”), derived forms (e.g. “large” and “largest”), and arbitrary words (e.g. “international” and “nations”) are also included here in searching word set. Use of uppercase or lowercase letters at the beginning (in a few cases in between) of the same search word is also found in the document pages. All these cases add more complexities while searching words from handwritten document images.

3.2 Training Sample Preparation

As mentioned earlier, 250 document pages out of 300 publicly available pages of the QUWI database are kept aside for confirming different parameters that are required to build the present word searching method. A total of 250 word images, per search word, are randomly extracted from 250 document pages that are used for the evaluation of the present technique. Therefore, a database containing $15 * 250 = 3750$ word images is prepared. This word set helps in selecting optimal parameter values, used for selection of the probable candidate search words and finalizing the search word from those probable candidate search words.

3.3 Parameters for Detecting Pre-selected Candidate Words

The feature values $f1$, $f2$, and $f3$ [see Eqs. (7)–(9)] are first extracted from each of the word images, and then the bounds of the same [see Eqs. (12)–(13)] are also calculated. These bound values for each of the query word images are reported in Table 2. Using these values and selection process, described in Figure 7, the irrelevant target words with respect to a search word are removed successfully. It has been observed that around 20% of the words in a document page are passed onto the next stage.

Table 2: Search Word-Wise Lower and Upper Bound Values of $f1$, $f2$, and $f3$ of $\mathfrak{F}_1$.

SW ID	$\mathcal{L}_{fi}$			U_{fi}		
	$i = 1$	$i = 2$	$i = 3$	$i = 1$	$i = 2$	$i = 3$
01	17.05	1	0	22.65	3	0
02	9.43	1	0	11.84	3	0
03	12.07	1	0	16.04	3	0
04	13.45	0	0	17.77	2	2
05	22.17	2	0	30.10	4	2
06	22.56	2	0	32.20	6	2
07	26.62	4	0	36.22	6	0
08	10.76	0	0	14.55	2	2
09	14.08	1	0	18.47	3	2
10	16.20	1	0	22.52	3	2
11	18.03	1	0	24.27	3	2
12	15.94	3	0	20.90	5	0
13	16.78	2	0	21.95	4	0
14	11.70	1	1	16.54	3	1
15	13.10	2	0	18.03	4	2

SW ID indicates search word index.

Table 3: Top 10 Recognition Results with Different Sets of Parameter Values Applied in a Five-Fold Cross-validation Mechanism.

Experiment No.	Parameter Value			Feature	No. of Neurons	Recognition
	m	p	q	Dimension	in the Hidden Layer	Accuracy (in %)
1	2	2	4	245	60	91.54
2	2	2	5	281	60	90.74
6	2	3	5	371	65	93.77
7	2	3	6	425	65	96.21
8	2	3	7	479	70	93.74
19	3	3	6	500	70	93.17
20	3	3	7	554	70	93.47
30	4	3	5	555	70	94.12
31	4	3	6	605	75	94.21
36	4	4	7	785	75	90.21

3.4 Parameter Optimization for Holistic Word Recognition

Selection of optimal parameter values for the holistic word recognition [i.e. as m, p, and q in Eq. (36)] is carried out here. This selection is performed based on recognition accuracy. For this purpose, a five-fold cross-validation schema is carried out using the said isolated word samples. Here, values of m and p are varied between 2 and 4. Such choice preserves the nature of zonal variation inside a word image. Whereas, the value of q is varied between 4 and 7. It is done to include more local information of the word images. Therefore, this experimental structure leads to $3 \times 3 \times 4 = 36$ number of experiments. All such experiments are conducted and the best recognition accuracies in five-fold cross validations are found to vary between 90.17% and 96.21% (see Table 3). In Table 3, the results of 10 such experiments are summarized that include the boundary values of the parameters. From this table, it is clear that the best recognition accuracy is achieved when values of m, p, and q are chosen as 2, 3, and 6, respectively. The learned module that provides the best recognition accuracy is used for confirming a pre-selected candidate word as search word (refer to Section 2.3.2.3).

3.5 Evaluation Metrics

The assessment of the present word searching mechanism is carried out in terms of recall, precision, and F-measure scores [18]. Let $\mathcal{W}_S$ and $\mathcal{W}_R$ represent the set of valid instances of a search word inside a document page image and the set of word(s) retrieved after applying the present method on said document image, respectively. Recall and precision and F-measure scores can be formulated as

$$\text{Recall} = \frac{|\mathcal{W}_S \cap \mathcal{W}_R|}{|\mathcal{W}_S|}. \tag{37}$$

$$\text{Precision} = \frac{|\mathcal{W}_S \cap \mathcal{W}_R|}{|\mathcal{W}_R|}. \tag{38}$$

$$\text{F} - \text{measure score} = \frac{2 \times \text{ recall } \times \text{ precision}}{\text{recall } + \text{ precision}}. \tag{39}$$

Let, $\mathcal{W}_S^{ij}$ and $\mathcal{W}_R^{ij}$ represent the set of actual and retrieved words while i^{th} ($i = 1, 2, \ldots, 15$) indexed word is searched in j^{th} ($j = 1, 2, \ldots, N$) document image. Now, measures such as recall ($\mathcal{R}_i$), precision ($\mathcal{P}_i$), and F-measure score ($\mathcal{F}_i$) for i^{th} indexed search word can be defined as follows:

$$\mathcal{R}_i = \frac{1}{N}\sum_{j=1}^{N} \frac{\left|\mathcal{W}_S^{ij} \cap \mathcal{W}_R^{ij}\right|}{\left|\mathcal{W}_S^{ij}\right|}, \text{where } i = 1,\ 2,\ \ldots,\ 15. \tag{40}$$

$$\mathcal{P}_i = \frac{1}{N}\sum_{j=1}^{N} \frac{\left|\mathcal{W}_S^{ij} \cap \mathcal{W}_R^{ij}\right|}{\left|\mathcal{W}_R^{ij}\right|}, \text{ where } i = 1,\ 2,\ \ldots,\ 15. \tag{41}$$

$$\mathcal{F}_i = \frac{1}{N}\sum_{j=1}^{N} \frac{2 \times \left|\mathcal{W}_S^{ij} \cap \mathcal{W}_R^{ij}\right|}{\left|\mathcal{W}_S^{ij}\,|+|\,\mathcal{W}_R^{ij}\right|}, \text{ where } i = 1,\ 2,\ \ldots,\ 15. \tag{42}$$

In Eqs. (40)–(42), $N = 50$, which is the number of document pages considered for evaluation. Search word-wise recall, precision, and F-measure scores are summarized in Table 4. From this table, it is clear that the present work has achieved reasonably good retrieval accuracy. This table also reveals that the best recall is achieved while searching the word "million" and overall satisfactory recall value (lowest recall value = 0.68) indicates better retrieving capability of the present work. Whereas, lower precision values ($\leq$0.5) are observed for the search words "Asia," "Asian," "today," and "Nations." The reason behind such lower precision rate is the similar $\mathfrak{F}_1$ information with other words in the document page, which, in turn, retrieves more words in the pre-selection stage.

3.6 Comparison with State-of-the-Art Methods

The performance of the present technique is compared with some state-of-the-art methods [36, 37]. Among these, the method reported in Ref. [36] is recognition based and another two methods [30, 37] are recognition-free word searching methods. It is to be noted that the authors of Ref. [37] compared several variations of time series matching techniques. However, here, the popularly used DTW method is used. The average performances of the said state-of-the-art methods along with the present one are reported in Table 5. The search word-wise performances of these techniques are shown in Figure 9A–C. From the results, it is clear that the present technique performs better than the other methods.

Table 4: Search Word-Wise Recall, Precision, and F-measure Values of Present Word Searching Technique.

SW ID (i)	Recall ($\mathcal{R}_i$)	Precision ($\mathcal{P}_i$)	F-measure ($\mathcal{F}_i$)
1	0.8462	0.7586	0.8000
2	0.8846	0.3108	0.4600
3	0.9167	0.3929	0.5500
4	0.7200	0.9205	0.8000
5	0.6800	0.8095	0.7391
6	0.7692	0.7692	0.7692
7	0.8462	0.8462	0.8462
8	0.8462	0.9167	0.8800
9	0.8333	0.9091	0.8696
10	0.9167	0.7333	0.8148
11	0.7600	**0.9830**	0.8636
12	**0.9194**	0.9344	**0.9268**
13	0.9000	0.3600	0.5143
14	0.9091	0.3125	0.4651
15	0.6923	0.9000	0.7826
Average	0.8349	0.7235	0.7388

SW ID stands for search word index. Bold numbers indicate the best scores.

Table 5: Comparison of the Present Method with Some State-of-the-Art Word Searching Techniques.

Methods with Year of Publication	Feature Extracted from	Average		
		Recall	Precision	F-measure
M1: Al Aghbari and Brook [1], 2009	Entire word	0.7090	0.4994	0.5638
M2: Al Aghbari and Brook [1], 2009	Connected parts of word image	0.6777	0.5460	0.5694
M3: Mondal et al. [20], 2016	Each column of word image	0.7045	0.0504	0.0901
M4: Mondal et al. [21], 2018	Each column of word image	0.5721	0.06324	0.1057
M5: Present study		**0.8349**	**0.7235**	**0.7388**

Bold numbers indicate the best scores.

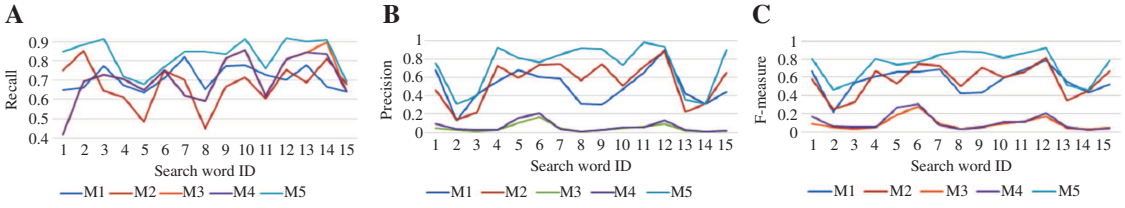


Figure 9: Search Word-Wise Evaluation Metrics of Different Methods.
Refer to Table 5 for the notations M1, M2, . . . , M5. (A) Recall, (B) Precision, and (C) F-measure.

4 Conclusion

A two-stage approach toward solving the problem of word spotting in a handwritten document page is reported here. In the pre-selection stage, irrelevant words in terms of number of characters present in the word image are removed. In the confirmation stage, pre-selected candidate words from a document image are passed through a holistic word recognition-based system. Based on recognition, confidence words are tagged as search words. The experimental results indicate that the proposed two-stage approach yields satisfactory retrieval performance. In spite of this success, there is still some room for improvement. First and foremost, a high-dimensional feature vector (dimension as reported in the best case recognition result is 425) is used for word recognition purpose. Therefore, application of a feature selection algorithm (either wrapper or filter) would be a good choice to improve the retrieval performance of the proposed approach. Also, some more context-sensitive features in the first stage of the current work could be applied in the future.

Acknowledgments: This work was partially supported by the CMATER research laboratory of the Computer Science and Engineering Department, Jadavpur University, India, and PURSE-II and UPE-II Jadavpur University projects. Dr. Ram Sarkar is partially funded by a DST grant (EMR/2016/007213).

Bibliography

[1] Z. Al Aghbari and S. Brook, HAH manuscripts: a holistic paradigm for classifying and retrieving historical Arabic handwritten documents, *Expert Syst. Appl.* **36** (2009), 10942–10951.
[2] S. Al Maadeed, W. Ayouby, A. Hassaïne and J. M. Aljaam, QUWI: An Arabic and English handwriting dataset for offline writer identification, in: *Proceedings of International Conference on Frontiers in Handwriting Recognition*, pp. 746–751, IEEE, 2012.
[3] S. Barua, S. Malakar, S. Bhowmik, R. Sarkar and M. Nasipuri, Bangla handwritten city name recognition using gradient-based feature, in: *Proceedings of International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pp. 343–352, Springer, 2017.
[4] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri and D. K. Basu, A hierarchical approach to recognition of handwritten Bangla characters, *Pattern Recogn.* **42** (2009), 1467–1484.
[5] S. Bhowmik, M. G. Roushan, R. Sarkar, M. Nasipuri, S. Polley and S. Malakar, Handwritten Bangla word recognition using HOG descriptor, in: *Proceedings of Fourth International Conference of Emerging Applications of Information Technology*, pp. 193–197, IEEE, 2014.
[6] S. Bhowmik, S. Polley, M. G. Roushan, S. Malakar, R. Sarkar and M. Nasipuri, A holistic word recognition technique for handwritten Bangla words, *Int. J. Appl. Pattern Recogn.* **2** (2015), 142–159.
[7] S. Bhowmik, S. Malakar, R. Sarkar, S. Basu, M. Kundu and M. Nasipuri, Off-line Bangla handwritten word recognition: a holistic approach, *Neural Comput. Appl.* (2018), 1–16 (in press).
[8] S. Cao and V. Govindaraju, Template-free word spotting in low-quality manuscripts, in: *Proceedings of International Conference on Advances in Pattern Recognition*, pp. 45–53, 2007.
[9] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.
[10] C. Djeddi, S. Al-Maadeed, A. Gattal, I. Siddiqi, L. Souici-Meslati and H. El Abed, ICDAR2015 competition on multi-script writer identification and gender classification using 'QUWI' database, in: *Proceedings of International Conference on Document Analysis and Recognition*, pp. 1191–1195, IEEE, 2015.
[11] A. Fischer, A. Keller, V. Frinken and H. Bunke, Lexicon-free handwritten word spotting using character HMMs, *Pattern Recogn. Lett.* **33** (2012), 934–942.
[12] V. Frinken, A. Fischer, M. Baumgartner and H. Bunke, Keyword spotting for self-training of BLSTM NN-based handwriting recognition systems, *Pattern Recogn.* **47** (2014), 1073–1082.
[13] A. P. Giotis, G. Sfikas, B. Gatos and C. Nikou, A survey of document image word spotting techniques, *Pattern Recogn.* **68** (2017), 310–332.
[14] M. Khayyat, L. Lam and C. Y. Suen, Learning-based word spotting system for Arabic handwritten documents, *Pattern Recogn.* **47** (2014), 1021–1030.
[15] K. Khurshid, C. Faure and N. Vincent, A novel approach for word spotting using merge-split edit distance, in: *Computer Analysis of Images and Patterns*, pp. 213–220, Springer, 2009.
[16] Y. Liang, M. Fairhurst and R. Guest, A synthesised word approach to word retrieval in handwritten documents, *Pattern Recogn.* **45** (2012), 4225–4236.
[17] S. Malakar, P. Ghosh, R. Sarkar, N. Das, S. Basu and M. Nasipuri, An improved offline handwritten character segmentation algorithm for Bangla script, in: *Proceedings of Indian International Conference on Artificial Intelligence*, pp. 71–90, 2011.
[18] S. Malakar, P. Sharma, P. K. Singh, M. Das, R. Sarkar and M. Nasipuri, A holistic approach for handwritten Hindi word recognition, *Int. J. Comput. Vis. Image Process.* **7** (2017), 59–78.
[19] S. Mallick, Histogram of oriented gradients, Available at http://www.learnopencv.com/histogram-of-oriented-gradients/, accessed 31 March, 2018.
[20] T. Mondal, N. Ragot, J. Y. Ramel and U. Pal, Flexible sequence matching technique: an effective learning-free approach for word spotting, *Pattern Recogn.* **60** (2016), 596–612.
[21] T. Mondal, N. Ragot, J. Y. Ramel and U. Pal, Comparative study of conventional time series matching techniques for word spotting, *Pattern Recogn.* **73** (2018), 47–64.
[22] W. Pantke, M. Dennhardt, D. Fecker, V. Margner and T. Fingscheidt, An historical handwritten Arabic dataset for segmentation-free word spotting – HADARA80P, in: *Proceedings of International Conference on Frontiers in Handwriting Recognition*, pp. 15–20, IEEE, 2014.
[23] R. Pintus, Y. Yang, E. Gobbetti and H. Rushmeier, An automatic word-spotting framework for medieval manuscripts, in: *Digital Heritage*, vol. 2, pp. 5–12, 2015.

[24] P. Riba, J. Lladās and A. Fornés, Handwritten word spotting by inexact matching of grapheme graphs, in: *Proceedings of International Conference on Document Analysis and Recognition*, pp. 781–785, IEEE, 2015.
[25] J. A. Rodriguez-Serrano and F. Perronnin, Handwritten word-spotting using hidden Markov models and universal vocabularies, *Pattern Recogn.* **42** (2009), 2106–2116.
[26] L. Rothacker and G. A. Fink, Segmentation-free query-by-string word spotting with bag-of-features HMMs, in: *Proceedings of the 13th International Conference on Document Analysis and Recognition*, pp. 661–665, 2015.
[27] L. Rothacker, M. Rusiñol and G. A. Fink, Bag-of-features HMMs for segmentation-free word spotting in handwritten documents, in: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 1305–1309, 2013.
[28] P. Roy, J. Ramel and N. Ragot, Word retrieval in historical document using character-primitives, in: *Proceedings of International Conference on Document Analysis and Recognition*, pp. 678–682, 2011.
[29] M. Rusinol, D. Aldavert, R. Toledo and J. Llados, Efficient segmentation-free keyword spotting in historical document collections, *Pattern Recogn.* **48** (2015), 545–555.
[30] R. Saabni and A. Bronstein, Fast keyword searching using boostmap based embedding, in: *International Conference on Frontiers in Handwriting Recognition*, pp. 734–739, IEEE, 2012.
[31] S. Sahoo, S. Nandi, S. Barua, P. Priyam, S. Bhowmik, S. Malakar and R. Sarkar, Handwritten Bangla word recognition using negative refraction based shape transformation, *J. Intell. Fuzzy Syst. Appl. Eng. Technol.* (2018) (in press).
[32] R. Sarkar, S. Malakar, N. Das, S. Basu, M. Kundu and M. Nasipuri, Word extraction and character segmentation from text lines of unconstrained handwritten Bangla document images, *J. Intell. Syst.* **20** (2011), 227–260.
[33] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri and D. K. Basu, CMATERdb1: a database of unconstrained handwritten Bangla and Bangla-English mixed script document image, *Int. J. Doc. Anal. Recogn.* **15** (2012), 71–83.
[34] R. Sarkar, S. Halder, S. Malakar, N. Das, S. Basu and M. Nasipuri, Text line extraction from handwritten document pages based on line contour estimation, in: *Proceedings of International Conference on Computing Communication & Networking Technologies*, pp. 1–8, IEEE, 2012.
[35] S. N. Srihari, C. Huang and H. Srinivasan, Search engine for handwritten documents, in: *Proceedings of International Society for Optics and Photonics Electronic Imaging*, pp. 66–75, 2005.
[36] S. Sudholt and G. A. Fink, PHOCNet: a deep convolutional neural network for word spotting in handwritten documents, in: *Proceedings International Conference on Frontiers in Handwriting Recognition*, pp. 277–282, 2016.
[37] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Llados and A. Fornes, A novel learning-free word spotting approach based on graph representation, in: *Proceedings of International Workshop on Document Analysis Systems*, pp. 207–211, IEEE, 2014.
[38] P. Wang, V. Eglin, C. Garcia, C. Largeron, J. Llados and A. Fornes. A coarse-to-fine word spotting approach for historical handwritten documents based on graph embedding and graph edit distance, in: *Proceedings of International Conference on Pattern Recognition*, pp. 3074–3079, 2014.