Data and Metadata. 2025; 4:54
doi: 10.56294/dm202554

ORIGINAL

A Two-stage Approach for Word Searching in Handwritten Document Images

Enfoque en dos etapas para la búsqueda de palabras en imágenes de documentos manuscritos

Ankur Goyal[1], Pronita Mukherjee[2], Dipra Mitra[3], Shiv Kant[4], Khalid Almalki[5], Suliman Mohamed Fati[6]

[1]Department of CSE, Symbiosis Institute of Technology, Symbiosis International Deemed University. Pune, Maharashtra, India.
[2]Department of CSE, Gargi Memorial Institute of Technology. Kolkata West Bengal, India.
[3]Department of CSE, Amity University Jharkhand. Ranchi, Jharkhand India.
[4]Department of Computer Science & Engineering (AI & DS), Greater Noida Institute of Technology (GNIOT). Greater Noida, Delhi/NCR, India.
[5]Department of Computer Science, College of Computing and Informatics, Saudi Electronic University. Riyadh.
[6]Chair of Information Systems Department, College of Computer and Information Sciences, Prince Sultan University. Riyadh-11586, Saudi Arabia.

Cite as: G Goyal A, Mukherjee P, Mitra D, Kant S, Almalki K, Mohamed Fati S. A Two-stage Approach for Word Searching in Handwritten Document Images. Data and Metadata. 2025; 4:54. https://doi.org/10.56294/dm202554

Submitted: 12-05-2024 **Revised:** 06-11-2024 **Accepted:** 15-03-2025 **Published:** 16-03-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding Author: Ankur Goyal

ABSTRACT

Introduction: despite the rise of electronic papers, handwritten paper documents remain important. Current technologies make document digitization, storage, compression, and transmission easy and affordable. But semi-automatic document image processing needs specific technology to extract document information accurately. Typed textual searches are used to get information from Digital Libraries.
Objective: generally, in a document, there exists a varying number of characters in different words. That is why searching a word in a whole document is incorporate mismatched word images in the fetched word image and also increases the time consumption to complete the task.
Method: keeping this idea in mind, the words having different number of characters with respect to the search word are discarded at the beginning as preprocessing.
Results: to confirm the outstanding words in the document page as probable search word, a voting-based approach has been used for doing this, a modified HOG feature descriptor is extracted from each word image, then 5 distance-matching metrics are calculated, fed to a voting schema with the help of threshold value of each metrics, calculated beforehand.
Conclusions: here 3 types of voting is performed, first 2, with the varying no of metrics vote for positivity of the search word and in the last one three distance metrics are used among which if more than one votes for the positivity the model will indicate the word as a search word.

Keywords: Feature Ex-Traction; Antlion Algorithm for Feature Section; Comparative Study with Existing Algorithm.

RESUMEN

Introducción: a pesar del auge de los documentos electrónicos, los documentos manuscritos en papel siguen siendo importantes. Las tecnologías actuales facilitan y abaratan la digitalización, el almacenamiento, la compresión y la transmisión de documentos. Pero el tratamiento semiautomático de imágenes de documentos necesita una tecnología específica para extraer la información del documento con precisión. Las búsquedas textuales se utilizan para obtener información de las Bibliotecas Digitales.

Objetivo: por lo general, en un documento existe un número variable de caracteres en las distintas palabras. Por ello, al buscar una palabra en un documento entero, se incorporan imágenes de palabras no coincidentes en la imagen de la palabra obtenida y también aumenta el consumo de tiempo para completar la tarea.
Método: teniendo en cuenta esta idea, las palabras que tienen un número diferente de caracteres con respecto a la palabra buscada se descartan al principio como preprocesamiento.
Resultados: para confirmar las palabras destacadas en la página del documento como palabra probable de búsqueda, se ha utilizado un método basado en votaciones. Se extrae un descriptor de características HOG modificado de cada imagen de palabra y, a continuación, se calculan 5 métricas de coincidencia de distancias, que se introducen en un esquema de votación con la ayuda del valor umbral de cada métrica, calculado de antemano.
Conclusiones: aquí se realizan 3 tipos de votaciones, las 2 primeras, con el nº variable de métricas votan por la positividad de la palabra buscada y en la última se utilizan tres métricas de distancia entre las cuales si más de una vota por la positividad el modelo indicará la palabra como palabra buscada.

Palabras clave: Feature Ex-Traction; Antlion Algorithm for Feature Section; Estudio Comparativo con Algoritmos Existentes.

INTRODUCTION

The main purpose of Document Image Analysis is the information retrieval properly from document images which contain either textual or pictorial or structural information the right understanding of this kind of information is a step toward closing the semantic gap between being able to recognize individual visual objects and being able to understand the whole paper in a given situation. It is not just typing up papers; it also involves getting semantic information from huge collections of image files kept in digital repositories and connecting it to other information. There are many historical handwritten papers that can be used for different projects and studies. To get the information, you can do one of two things:

- Word spotting.
- Transcribing documents (word-to-word)

The DIAR community wants to keep these papers safe and get as much useful information from them as possible. Office automation necessitates the digital storage, manipulation, and recovery of documents to accurately handle handwritten documents. Optical character recognition (OCR) systems can read digital files created from paper documents as a solution.(1) The present handwritten OCR work doesn't work well with big word lists.(2) The other option is to keep the papers digitally and tag them correctly.

Optical Character Recognition

OCR turns pictures of typed, scribbled, or printed text into machine-readable text. This can be done mechanically or electronically, and the image can be a scan or a picture of a document. Digitalizing papers in this way makes them easier to edit and store, and it's also used for machine translation, text-to-speech conversion, data and text mining. AI, pattern recognition, and computer vision are all studied in the area of OCR. The OCR system turns paper records into text that computers can read. Software that can automatically read scanned text files and change them into a format that computers can understand better.(3) Everything from handwriting analysis depends on OCR. People who have this soft copy can change the document's format and find it as if it were made with a word processor. Let's say there was only one letter, A. So, it's clear that OCR would have a tough time, since everyone writes the letter A a little differently. There is a problem even with printed writing because there are many typefaces (fonts) and the letter A can be printed in a lot of different ways.

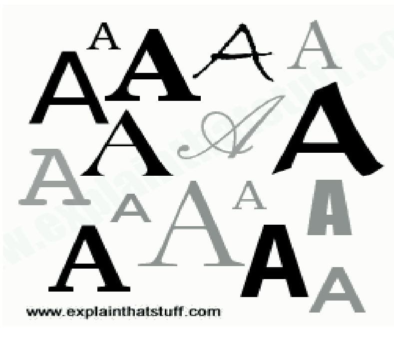


Figure 1. A fair bit of variation of a capital letter A, basic similarity: almost all of them are prepared from two angled lines that meet in the middle at the top with a horizontal line between

https://doi.org/10.56294/dm202554

Basically, there are two different ways to solve this problem: either by recognizing the symbols as a whole (pattern recognition) or by figuring out what lines and strokes make up each character (feature detection) and figuring out what they are. Pattern recognition: OCR programs are given examples of text in different fonts and formats. They then match the characters in a scanned document and figure out what they are.

Feature detection—OCR programs use rules about the parts of a letter or number to figure out what they are in a scanned document. One feature that could be used to compare characters is the number of angled lines, crossed lines, or bends in each one. As an example, the letter "A" could be saved as two vertical lines that meet in the middle and form a straight line. It is possible for computers to do more things with characters after they have been discovered. People should be able to fix simple mistakes, check for spelling and grammar mistakes, and make sure that complicated plans were done correctly before they save the document for later use.

Problem with Optical Character Recognition

OCR works accurately for machine printed text. But, OCR performance is unstable, when documents contain either handwritten text or symbols or graphical structures. Though OCR is one of the techniques to swiftly collect and analyse massive volumes of physical (paper) data, it can still be extremely complex and time-consuming to use. It must be assured that the document is in a language, and the OCR software can recognize; not all engines are trained to recognize all languages.(4) Low contrast in documents can diminish OCR accuracy; contrast can be attuned in a photo manipulation tool. Text created earlier to 1850 or by a typewriter can be more challenging for OCR software to read. OCR software is unable to read handwriting; while the handwritten notes are digitized.

While most technologies have gotten better in the last few years, OCR technology hasn't changed much in over ten years. Many people haven't been motivated to use new technology because OCR works well enough for machine-printed text. Unfortunately, this has made a lot of people used to long wait times and think of them as "part of the process." Since there isn't much desire for faster OCR, there isn't much drive to make the technology better. This means that traditional OCR technology is slow and hard to predict. Poor accuracy and slow speeds can be big problems for OCR, a technology that is supposed to make document handling faster. When it comes to word spotting, OCR needs a lot of technical and human tools. It often needs a lot of memory and processing power, which slows down the system and makes it harder to scan a lot of papers. Most of the time, OCR is very wrong. This is especially true for low-quality papers like handwritten text, manuscripts, symbols, and graphical structures. As a result, OCR needs more and more human review to make sure the results are correct.

By considering the above issues with OCR, it is still a big scientific challenge to retrieve information from documents, which contain information in the form of either handwritten text or symbols or graphical structures.

Literature Survey

According to literature review, many researchers have used distance-based techniques(5) to probe a word from a document image using QbE. Word-spotting was developed to recognize speech words. Later, it matched and indexed terms in text documents. This was first suggested in (6) by Manmatha, and many words matching(7) methods were developed. In (8), Rath et al, proposed an automatic retrieval system for historical handwritten documents, which consists of two different statistical models to retrieve words (within a text query) from large collection of handwritten manuscripts. Both statistical models were using a collection of page images that have been transcribed in order to investigate a joint probability distribution between features that are calculated from word images and their transcriptions Later, this automatic retrieval system was used to retrieve handwritten documents unlabeled images as well. In generic the term image is regarded as a single shape rather than being divided into smaller components. As a result, the shape matching method is used for word matching recognition, which allows for the calculation of image features at specific points of interest. The Harris detector(9) can detect corners, according to recent comparative research between several interest detector sites; however, as is typically the case, this detector's sensitivity and response to noise are its drawbacks. In (9), a five-step query word detecting process was recommended for document pictures. Query word and document image were pre-processed for noise reduction. Quality, age, and scanning device imperfections cause noise in document pictures. In preprocessing, Adaptive Thinning Framework (ATF) reduced noise and normalized data. Adaptive Thinning Framework developed 1:1 picture representation that were more noise-resistant than conventional thinning techniques. Step two retrieved picture features from components as feature vectors. A feature vector was recovered for each query and document image component based on the shape points distribution within the form enclosing circle in polar coordinates using the Contour Points Distribution Histogram (CPDH) shape descriptor. A two-dimensional histogram showed the point distribution. A similarity matrix was produced by matching query component feature vectors and document picture component feature vectors in the third stage. Fourth, similarity matrix searches document image for candidate image occurrences. After filtering out extraneous patterns, relevant occurrences were found and kept. It was reported as an application-independent

https://doi.org/10.56294/dm202554

and segmentation-free multipart query method for document image detection. The 5-step feature matching method removed irrelevant pixels to detect query occurrences in document images. Experiments showed promising performance and room for development. In [10], 'Column based features' and 'Slit style Histogram of Oriented Gradient (SSHOG) based features' extracted features. 'Column-based features' extraction computes eight statistical features from left to right on each pixel column of an N-pixel image. SSHOG-based features extraction extracts HOG[10] features for each slit by sliding a fixed-sized window horizontally over the image. Classical word spotting was utilized for experiments. Due to handwriting quality diversity, words in the same category can vary in length and size. This was resolved using two pruning methods. One used bounding box area and another estimated word image characters. Simple image attributes for GW-90 and Bentham dataset were used to reduce irrelevant word images (based on query size) before performing. To avoid threshold fine tuning, primitive and simple threshold values were chosen. Finally, dynamic time warping determined the best warping path between two time series. This warping path meets boundary criteria, continuity, and monotonicity. The two-stage strategy of pre-selection of target words and confirmation of pre-selected word(s) as search word was introduced in document page pictures to search a word.[7] HOG, a validated texture-based feature descriptor, computes gradient information for a cell (a primitive sub-block) in multiple directions. Second, each block undergoes cell normalization to generate a pattern. For each cell orientation measurement, total orientation angle (0°-360° or 0°-180°) is divided into ranges. The angle ranges are called "range" or "bin". A 2c × 2c block size was used to extract HOG feature into b bins from a c × c picture cell size. Zeros were used to pad images to multiples of c before applying HOG. Next, extracted information mean and standard deviation for each b bin were used as feature values, resulting in a 2 * b feature vector for each image. This update preserved image size to construct an equal-length feature vector, reducing feature vector dimension. Two features were retrieved from an image's upper and bottom sections and separated by principal and non-principal diagonal. Later, these features were retrieved from all sub-images to collect local information. To find meaningful words in a document image, a length 3 feature vector was extracted from all words. Using a holistic word recognition approach, a feature vector with topological features and a modified HOG feature descriptor was extracted from each word image to classify with a Multi-Layer Perceptron classifier to confirm the relevant filter in words as expected search words. Seven features-based systems were presented in [11] to extract all feasible word similarities and remove noise or font style variances. The seven features were width-to-height ratio, word area density, center-of-gravity, vertical-projection, top-bottom form projections, upper grid features, and down grid features. In the first set, word outline width-to-height ratio matters for word shape. Word area density represents black pixels percentage in word bounding box in second set. In the third set, center of gravity is the Eucli-dean distance from word's C.G. to upper left c0rner of bounding box. Calculate this by finding the vertical and horizontal center of gravity. The fourth set's vertical projection feature is a 20-element vector recovered from word picture after smoothing and vertical projection normalization. These items matched the first 20 smoothed and normalized vertical projection coefficients of discrete cosine transform. Top-bottom shape projections were word shape signatures in fifth set. These signatures were 50-element feature vectors with 25 coefficients of "smoothed and normalized top shape projection" discrete cosine transform and 25 coefficients of "smoothed and normalized bottom shape projection discrete cosine transform". Top shape projection was calculated by scanning word image top-to-bottom. All subsequent pixels in the same column were black after the first black pixel was identified. Bottom shape projection was found by scanning word picture from bottom-to-top and converting all pixels to black until a black pixel was found. In six sets, upper grid features (UGF) were 10-element binary value vectors derived from word image uppers. After extracting the image's horizontal projection, the upper word component is calculated. Seven sets had down grid features that were comparable to upper grid features but extracted from the bottom word picture. We generated the down grid features using the upper grid features method, but we started from the bott-om of the horizontal projection histogram. The output was another 10-element binary vector.

METHOD

Any word searching model has two main steps: page segmentation (extracting text lines and/or words from document pictures) and word matching schema. Pré-selection and confirmation of target terms as search words are used to search document pictures. A 3-length feature vector was constructed from all document picture words to exclude extraneous terms for the search word.[7] Words in a document usually have varied character counts. That is why searching an entire document for a word returns mismatched word pictures and takes longer. Preprocessing discards terms with different character counts than the search word. Voting has been utilized to confirm outstanding terms on the document page as plausible search words. A modified HOG feature descriptor is extracted from each word picture, then 5 distance-matching metrics are produced and supplied to a voting schema using pre-calculated threshold values. First, two metrics vote for the positivity of the search word, and in the last one, three distance metrics are employed. If more than one vote for positivity, the model will designate the word as a search word.

https://doi.org/10.56294/dm202554

Pre-selection of Search Words for a given Search Word

Assume that a binarized word picture can be represented like Bp = {f (i, j) : 1 ≤ i ≤ h∧ 1 ≤ j ≤ w}, where h and w are height and width of Bp, respectively, and f (i, j) ∈ {0, 1} (as, 0 and 1 represent non-data pixels and data pixels, respectively).

For feature extraction, Bp is first split horizontally into three non-overlapping zones—that is, upper, middle, and lower zones as illustrated in figure 3. To mark the areas horizontally, this picture features four horizontal lines: R1, R2, R3, and R4. The lines R1 and R4 are computed in turn:

“$R1 = \min \{i : f(i, j) = 1 \wedge (i, j) \in [1, h] \times [1, w]\}$ (1)
$R4 = \max \{i : f(i, j) = 1 \wedge (i, j) \in [1, h] \times [1, w]\}$ “ (2)

The identification of R2 and R3 is complex. To identify these lines, the quantity of transition points between data and non-data pixels, as well as vice versa, is computed for each row of Bp.

“$Ti = | \{j: ((f(i, j) = 1 \wedge f(i, j + 1) = 0) \vee (f(i, j) = 0 \wedge f(i, j + 1) = 1)) \wedge j \in [1, w - 1]\} |$ “ (3)

The mean of all such transition point counts (μ.TP) of B is estimated by:

“$\mu.TP = 1/N\Sigma$, where $N = |\{i : Ti \neq 0\}|$, $i \in [1, H]$ “ (4)

Now, R2 and R3 are calculated as follows:

“$R\text{-}2 = \min_{i=1,2,...,H}\{i : T.i > \mu TP\}$ (5)
$R\text{-}3 = \max_{i=1,2,...,H}\{i : T.i > \mu TP\}$” (6)

Finally, the F-1(= (f 1, f 2, f 3)) is estimated as:

“$f1 = 1/M\Sigma\sum_{i=R2}^{R3} Ti$ (7)

Where $M = |\{i : T.i \neq 0\}|$, $\forall i \in [R.2, R.3]$.

$f2 = |\{C : \theta(C) = 1\}|$. (8)
$f3 = |\{C : \emptyset(C) = 1\}|$.” (9)

Now, let θ(.) and∅(.) be functions that represent the belongingness of a CC in upper/lower zone, respectively, which are defined by:

“$\theta(C) = \begin{cases} o, if \min\{i : f(i,j) \in C\} \leq (R1 + R2)\ 2 \text{ and } \max\{i : f(i,j) \in C\} = R2 - 1 \\ 1, otherwise \end{cases}$ (10)

$\emptyset(C) = \begin{cases} o, if \max\{i : f(i,j) \in C\} \leq (R3 + R4)/\ 2 \text{ and } \min\{i : f(i,j) \in C\} = R3 + 1 \\ 1, otherwise \end{cases}$

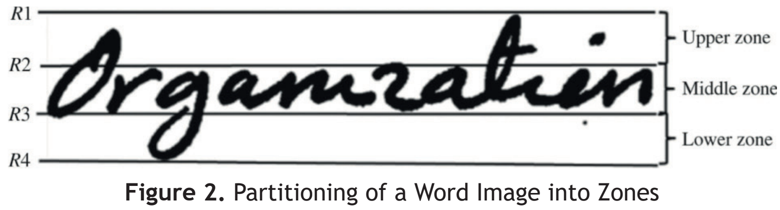


Figure 2. Partitioning of a Word Image into Zones

Decision Rule Creation

A decision rule removes extraneous terms from a search. For each extracted feature value, lower and upper decision limits are estimated. The decision limits are determined by analyzing the mean (μfi, i = 1, 2, 3) and standard deviation (σfi, i = 1, 2, 3) of feature values from manually selected N word picture examples for a search phrase. Define Lfi and Ufi, the lower and upper boundaries of feature value fi (i = 1, 2, 3).

“$L_{fi} = \mu_{fi} - \sigma_{fi}$, where i = 1, 2, 3” (11)
“$U_{fi} = \mu_{fi} + \sigma_{fi}$, where i = 1, 2, 3” (12)

https://doi.org/10.56294/dm202554

Decision rules pre-classify words as potential search word candidates. The first two steps of any word searching model are page segmentation (extracting text lines and/or words from document pictures) and word matching schema. Using document images, a two-stage strategy is used to search for a word: pre-selection and confirmation. A feature vector of length 3 was extracted from all document words to remove unnecessary terms for the search word.[7] Character counts vary by word in a document. Therefore, searching a word in a whole document returns mismatched word pictures and takes longer. With this in mind, preprocessing discards terms with differing character counts than the search word. Using voting, the document page's outstanding terms were confirmed as plausible search words. This is done by extracting a modified HOG feature descriptor from each word image, calculating 5 distance-matching metrics, and feeding them to a voting schema with their threshold values. The first two metrics vote for the positivity of the search word, and the last three distance metrics vote for it. If more than one is positive, the model will mark the word as a search word.

Deciding a Pre-selected Candidate Word as a Search Word

Figure 3. Fetched word images for the search word ASIA

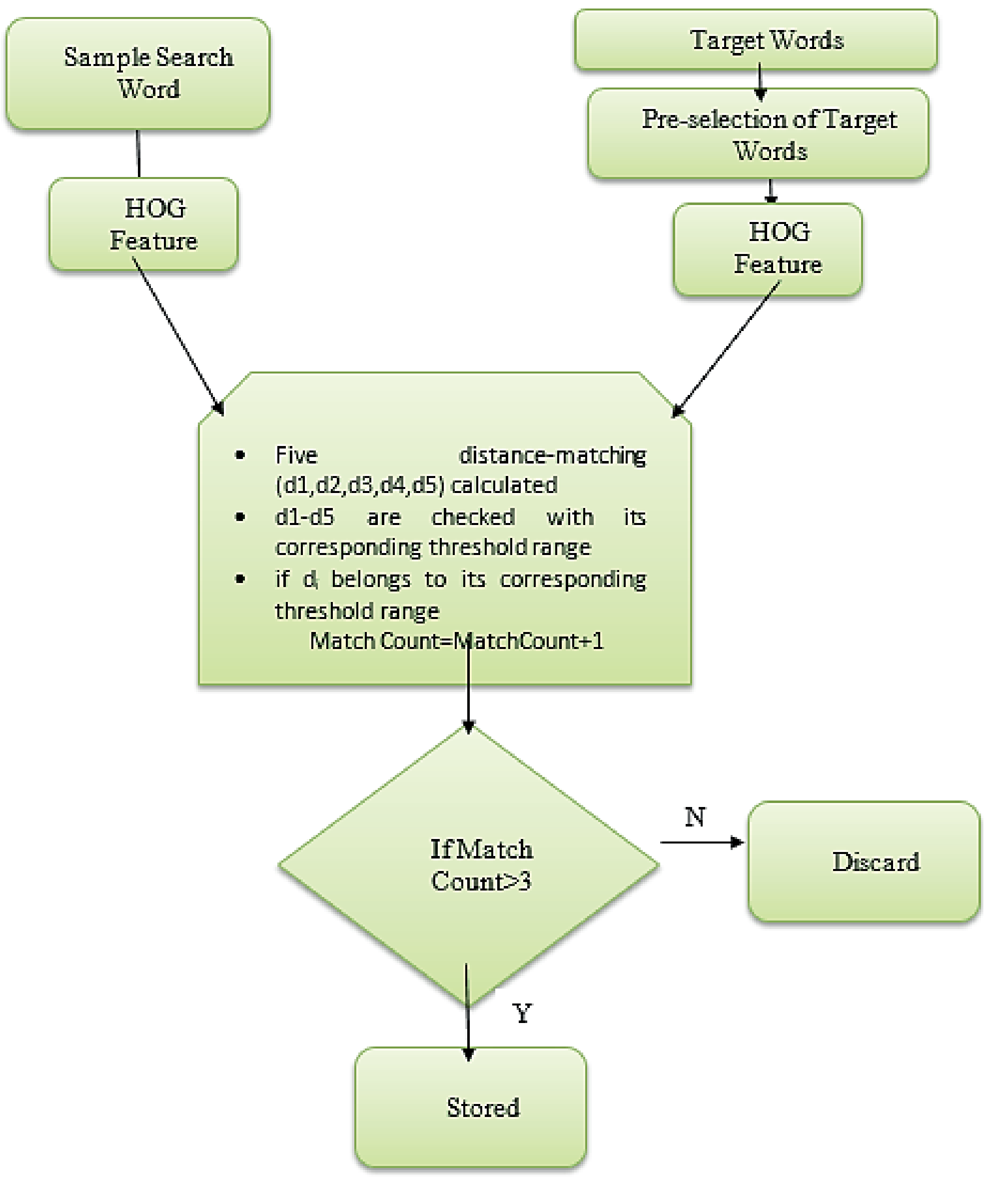


Figure 4. Block Diagram of the Present Word Spotting Model

https://doi.org/10.56294/dm202554

A texture-based feature descriptor; HOG is very useful in pattern-based recognition. HOG computes the gradient of a cell in various directions and then the values are normalized for the pattern description of each block. The total orientation angle (00-1800 or 00-3600) can be divided into different ranges (like 8, 9, etc.), each of which is known as bin. Here we used 9 bins over the total feature length extracted from HOG, in case of Hausdorff distance, Fréchet distance, DTW distance measures, as these distance matrices always check foremost corner to the last corner of all extracted feature values which may decries the required performance. For rest two matching metric (Waterman, LCS) total feature length is compared.

From all the search words, 15 samples are selected manually as the candidate word. For a candidate word, all pre-selected words are gone through the present module. At the beginning five distance-matching metrics for each of these words (d1, d2, d3, d4, d5) are calculated. Then, the obtained five scores are compared with the threshold range if the scores are in the corresponding threshold range, then that metric will increase the value of matching flag variable by 1. According to these, if M distance-matching metric votes for a search word as a target word then this model will store or load the word image in the designated folder. When M varies the performance of the model will change accordingly It is observed that, among the five metrics, distance metrics are providing better performance while compare with the matching metrics. The working principle of these matching metrics is based on the longest substring retrieval from the extracted features of the word images. At the time of matching score generation these metrics are producing huge matching scores, which belong to the threshold range in many cases though they are wrongly retrieved.

The matching metrics votes as the target word though they are not, which increases the false positive count as well as decreasing the performance of the model. To increase the performance of the present model, a voting system is designed with only 3 distance metrics, which is retrieving less no of wrong words i.e. the no of false positive is decrease, increasing the performance of the model. Let's say, when it is searched for word image ―ASIA‖, it picked the following two images along with others, where one image (figure 3a) is correct, though the handwritten version of this word is not good. But, it also picked the other image (figure 2) as word ‗ASIA'; despite the fact that, this word image is actually stands for ‗Also'.

RESULTS AND DISCUSSION

Database Description

Handwritten word spotting is a classification task to recognize pattern which is mainly used to detect specific keyword(s) within handwritten document images. In this current work, the QUWI database is preferred as it is available to a certain extent for public use in different forums. Such preference for QUWI database is significant as a huge number of diversified (e.g., age, sex, nationality, background etc.) of writers. A good number of writing materials variations are available in terms of colors and thicknesses of pen/pencil.

Around 300 writers' handwritten manuscripts in English and Arabic have been posted for public usage in the "ICDAR 2015 competition on multi-script writer identification and gender classification". using QUWI database. Mostly same text with117 words are kept in for each script. A page can have minimum of 99 words, as observed. Following constrains have been added with these document pages to increase searching complexity:

- Wrongly spelled words.
- Use of different-abbreviated forms.
- Different spellings of same word.

The document is alienated into 1:5 ratio. The first set of document-pages are for the-evaluation of present word searching-algorithm, while the rest-of-document pages are useful to confirm values require to design the searching algorithm different parameters.

In this current work, a word searching model for hand-written document images is presented. A two-stage technique to word searching that discussed earlier is introduced for this purpose. In this sub-section experimental outcomes are described in details.

It has been mentioned at the start of this section that, the QUWI database is used to carry out the experiment. A set of 15 words are selected from document page images as query words, as shown in table 1, for the evaluation of designed word searching technique. The choices of selecting word(s) to be searched are mainly depended on multiple occurrences of the searching word(s) within the respective document page.

Additionally, following word pairs that are almost impossible to differentiate are also included here in searching word set:

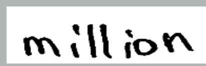
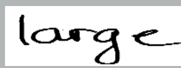
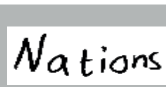
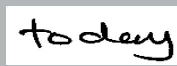
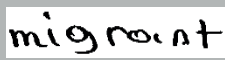
- In-terms of shape (e.g., "today" and "today's")
- Ste-mming words (e.g., "Asia" and "Asian" or "migrant" and "migrants")
- D-erived forms (e.g., "large" and "largest")
- Arbitrary words (e.g., "international" and "nations")

Distinguishing same words, which are starting with upper case or lower-case increases searching complexity within handwritten document images.

https://doi.org/10.56294/dm202554

To identify the search word within pre-selected candidate words is extracted from manually chosen set of fifteen words among all in our experimental setup. The word instances are:

Table 1. Search words instances

SW	Image Instance	SW	Image Instance	SW	Image Instance
SW01		SW06		SW11	
SW02		SW07		SW12	
SW03		SW08		SW13	
SW04		SW09		SW14	
SW05		SW10		SW15	

SW stands for "Index number of the particular Search Word image instances".

Evaluation Metrics

Results of present work are measured in terms of accuracy in searching process. The evaluation of the present searching model has been done using recall, precision and F-measure scores.

- Recall is the ratio of the relevant document that are retrieved successfully, can be depicted as:

$$Recall = \frac{\{Relevantwordimage\} \cap \{Retrivedwordimage\}}{\{Relevantwordimage\}}$$

- Precision is the ratio of retrieved document with respect to relevant query, can be depicted as:

$$Precision = \frac{\{Relevantwordimage\} \cap \{Retrivedwordimage\}}{\{Retrivedwordimage\}}$$

- F-measure or balanced F-score is harmonic mean of recall and precision, can be depicted as:

$$F - Measure = \frac{2 \times recall \times precision}{recall + precision}$$

Experimental Outcomes and Analysis

In this section, the experimental results are carried out using this model. Among two stages, the first stage, pre-processing is carried out in.

In the present work, 15-word images are selected as search word of this model, and then threshold range is calculated for DTW, Fréchet, Hausdorff, Waterman and LCS methods, in table 2.

Table 2. Search Word-based Threshold Range of Word Searching Model using DTW, Fréchet, Hausdorff, Waterman, LCS

SW	DTW	Fréchet	Hausdorff	Waterman	LCS
America	0-0,2128	0-0,1097	0-0,0518	7,8704-100	47,2222-100
Asia	0-0,2730	0-0,1359	0-0,0399	6,2500-100	40,9722-100
Asian	0-0,2797	0-0,1396	0-0,0438	6,9444-100	43,0556-100
Europe	0-0,1621	0-0,0834	0-0,0576	9,7222-100	41,6667-100
Immigrants	0-0,2577	0-0,1300	0-0,0576	9,9537-100	49,3056-100
Immigration	0-0,2314	0-0,1190	0-0,0498	13,889-100	46,5278-100
International	0-0,1541	0-0,0919	0-0,0444	13,789-100	59,0278-100
Large	0-0,2426	0-0,1177	0-0,0288	6,4815-100	39,5833-100
Largest	0-0,2744	0-0,1357	0-0,0548	6,4815-100	33,3333-100
migrant	0-0,2916	0-0,1776	0-0,0647	9,0278-100	45,8333-100

https://doi.org/10.56294/dm202554

migrants	0-0,2783	0-0,1703	0-0,0589	8,9935-100	46,3261-100
million	0-0,1827	0-0,0984	0-0,0327	9,0278-100	52,7778-100
Nations	0-0,1658	0-0,0897	0-0,0453	11,111-100	53,4722-100
today	0-0,3204	0-0,1488	0-0,0428	6,7130-100	40,2778-100
Today's	0-0,2344	0-0,1232	0-0,0444	6,2500-100	38,1944-100

Note: "SW" stands for search word index.

The testing outcomes of this work are reported separately into eight tables, among those table 3 contains the values of Recall, Precision and F-Measure of the 15 search words passed through DTW.

Table 3. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using DTW

SW	Recall	Precision	F-Measure
SW01	0,7100	0,1224	0,2088
SW02	0,3069	0,0920	0,1416
SW03	0,6458	0,0286	0,0548
SW04	0,1020	0,7692	0,1802
SW05	0,4950	0,1707	0,2539
SW06	0,5200	0,2321	0,3210
SW07	0,4600	0,7667	0,5750
SW08	0,3878	0,07567	0,1267
SW09	0,7500	0,0540	0,1007
SW10	0,7347	0,0350	0,0667
SW11	0,5657	0,1000	0,1699
SW12	0,3036	0,4573	0,3650
SW13	0,4222	0,0526	0,0936
SW14	0,7083	0,0865	0,1542
SW15	0,5400	0,0415	0,0770
Average	0,5101	0,2056	0,1926

In table 4, the values of Recall, Precision and F-Measure of the 15 search words passed through Hausdorff Distance technique are reflected.

Table 4. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using Hausdorff

SW	Recall	Precision	F-Measure
SW01	0,9400	0,1414	0,2458
SW02	0,7327	0,0846	0,1516
SW03	0,5625	0,0254	0,0486
SW04	0,6735	0,0565	0,1042
SW05	0,5455	0,1709	0,2602
SW06	0,5200	0,1831	0,2708
SW07	0,5800	0,6304	0,6042
SW08	0,3673	0,0387	0,0700
SW09	0,8333	0,0330	0,0635
SW10	0,8571	0,0398	0,0762
SW11	0,5960	0,1021	0,1743
SW12	0,4291	0,3072	0,3582
SW13	0,4667	0,0332	0,0620
SW14	0,8125	0,0400	0,0762
SW15	0,6400	0,0287	0,0549
Average	0,6371	0,1277	0,1747

https://doi.org/10.56294/dm202554

In table 5, the values of Recall, Precision and F-Measure of the fifteen-search word passed through Fréchet Distance technique is reflected.

Table 5. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using Fréchet

SW	Recall	Precision	F-Measure
SW01	0,6500	0,1451	0,2372
SW02	0,3069	0,1076	0,1594
SW03	0,6250	0,0309	0,0588
SW04	0,1224	0,8000	0,2124
SW05	0,4848	0,1868	0,2697
SW06	0,4700	0,2186	0,2984
SW07	0,4800	0,7742	0,5926
SW08	0,3265	0,1391	0,1951
SW09	0,6458	0,0555	0,1021
SW10	0,6531	0,0304	0,0581
SW11	0,7071	0,1094	0,1894
SW12	0,3036	0,4601	0,3658
SW13	0,4000	0,0627	0,1084
SW14	0,6458	0,1308	0,2175
SW15	0,5200	0,0384	0,0715
Average	0,4894	0,2193	0,2091

In table 6, the values of Recall, Precision and F-Measure of the fifteen-search word passed through LCS matching technique is reflected.

Table 6. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using LCS

SW	Recall	Precision	F-Measure
SW01	0,7400	0,1080	0,1885
SW02	0,6238	0,0809	0,1432
SW03	0,5625	0,0288	0,0548
SW04	0,5918	0,0599	0,1088
SW05	0,4041	0,1270	0,1932
SW06	0,5200	0,1751	0,2620
SW07	0,1800	0,7500	0,2903
SW08	0,6531	0,0217	0,0421
SW09	0,7500	0,0264	0,0510
SW10	0,5306	0,0273	0,0519
SW11	0,5051	0,0806	0,1391
SW12	0,1822	0,3309	0,2350
SW13	0,4444	0,0359	0,0664
SW14	0,6250	0,0294	0,0561
SW15	0,5000	0,0162	0,0313
Average	0,5208	0,1265	0,1276

In table 7, the values of Recall, Precision and F-Measure of the 15 search words passed through Waterman matching technique are reflected.

In table 8, 9 y 10, voting technique is performed using the present model by applying DTW, Hausdorff, Fréchet, where we assume that if more than two processes vote for yes for a certain word then that will be considered as the target word.

Table 7. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using Waterman

SW	Recall	Precision	F-Measure
SW01	0,7300	0,1059	0,1850
SW02	0,6040	0,0546	0,1001
SW03	0,6042	0,0243	0,0467
SW04	0,1939	0,1418	0,1638
SW05	0,5354	0,1715	0,2598
SW06	0,5100	0,2267	0,3138
SW07	0,6400	0,3299	0,4354
SW08	0,6939	0,0274	0,0528
SW09	0,6042	0,0247	0,0475
SW10	0,4082	0,0456	0,0820
SW11	0,7071	0,1165	0,200
SW12	0,4089	0,3146	0,3556
SW13	0,4222	0,0341	0,0631
SW14	0,5625	0,0265	0,0507
SW15	0,5800	0,0184	0,0356
Average	0,5470	0,1108	0,1595

Table 8. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using voting among five metrics (match count>2)

SW	Recall	Precision	F-Measure
SW01	0,7300	0,1148	0,1984
SW02	0,6040	0,1871	0,2857
SW03	0,6042	0,0322	0,0612
SW04	0,1939	0,7600	0,3089
SW05	0,5051	0,1629	0,2463
SW06	0,5100	0,2267	0,3138
SW07	0,5400	0,8438	0,6585
SW08	0,3673	0,0428	0,0766
SW09	0,7292	0,0311	0,0597
SW10	0,6531	0,0309	0,0589
SW11	0,5859	0,0853	0,1489
SW12	0,1903	0,2670	0,2222
SW13	0,4444	0,0424	0,0774
SW14	0,7500	0,0729	0,1328
SW15	0,5200	0,0232	0,0449
Average	0,5285	0,1949	0,1929

Table 9. Search Word-based Recall, Precision, F-Measure Values of present Word Searching Model using voting among five metrics (match count>3)

SW	Recall	Precision	F-Measure
SW01	0,6800	0,1331	0,2226
SW02	0,3168	0,0982	0,1499
SW03	0,5625	0,0329	0,0622
SW04	0,1225	0,8000	0,2124
SW05	0,4646	0,1631	0,2415

SW06	0,4800	0,2233	0,3048
SW07	0,4200	0,8077	0,5526
SW08	0,2653	0,0684	0,1088
SW09	0,5208	0,0383	0,0714
SW10	0,4694	0,0241	0,0458
SW11	0,4949	0,0851	0,1452
SW12	0,1822	0,4369	0,2571
SW13	0,3778	0,0547	0,0956
SW14	0,7083	0,1411	0,2353
SW15	0,4800	0,0344	0,0642
Average	0,4363	0,2094	0,1846

Table 10. Search Word-wise Recall, Precision, F-Measure Values of present Word Searching Model using voting among DTW, Fréchet and Hausdorff (match count>1)

SW	Recall	Precision	F-Measure
SW01	0,7200	0,1309	0,2215
SW02	0,4158	0,1170	0,1826
SW03	0,6250	0,0287	0,0549
SW04	0,2143	0,0732	0,1091
SW05	0,5051	0,1742	0,2591
SW06	0,5800	0,2589	0,3581
SW07	0,5000	0,8065	0,6173
SW08	0,5102	0,1202	0,1946
SW09	0,7500	0,0534	0,0997
SW10	0,7347	0,0439	0,0829
SW11	0,6667	0,0629	0,1149
SW12	0,3239	0,4124	0,3628
SW13	0,6000	0,0520	0,0957
SW14	0,7708	0,1000	0,1770
SW15	0,5800	0,0408	0,0762
Average	0,5664	0,1650	0,2004

Comparison with State-of-the-art Methods

Current approaches are compared to state-of-the-art.[12] Recognition-based[6] and recognition-free searching methods are shown. Several time series matching methods were compared in. Table 11 shows average performance of state-of-the-art approaches, including the present one.

Table 11. Comparative study of the Present Method with other state-of-the-art Word Searching Methods

Methods along with Publication year	Feature Extracted from	Average		
		Recall	Precision	F-measure
M1: Mondal et al., 2016	Each column of word image	0,7045	0,0504	0,0901
M2: Mondal et al., 2018	Each column of word image	0,5721	0,06324	0,1057
M3:Present Study with 5 metrics having Match Count>2	Entire word	0,528481	0,194867	0,192935
M4:Present Study with 5 metrics having Match Count>3	Entire word	0,436347	0,209409	0,184615
M5:Present Study with 3 metrics having Match Count>1	Entire word	0,566431	0,164999	0,200428

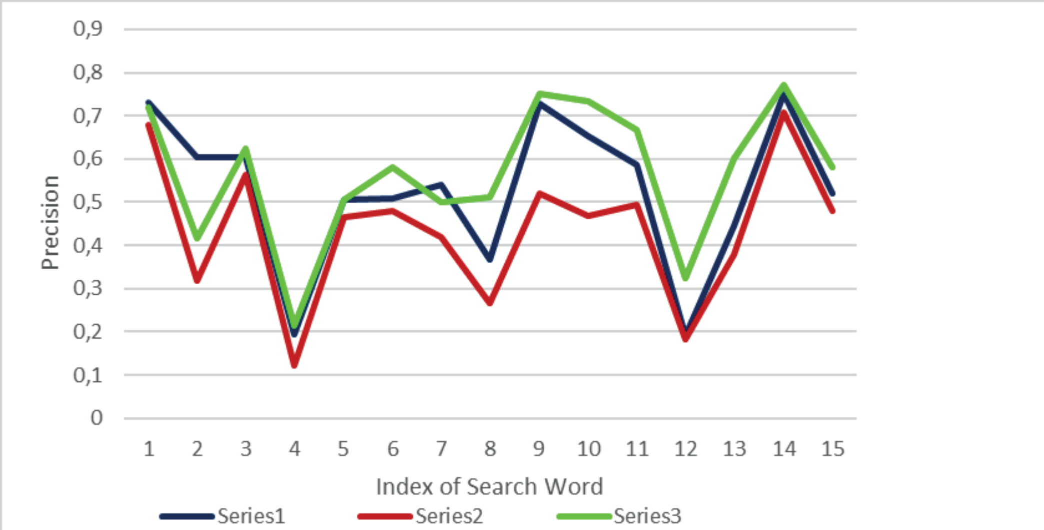


Figure 5. Comparative study of the Precision values of three Methods (M1:Series1, M2:Series2, M3:Series3) of words searching techniques

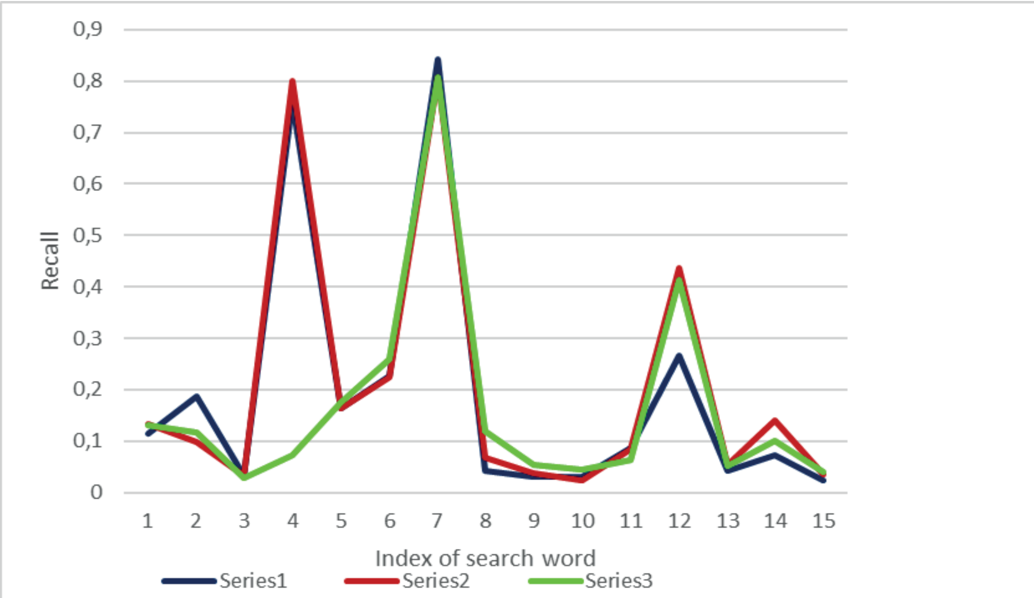


Figure 6. Comparative study of Recall values of three Methods (M1:Series1, M2:Series2, M3:Series3) of word searching techniques

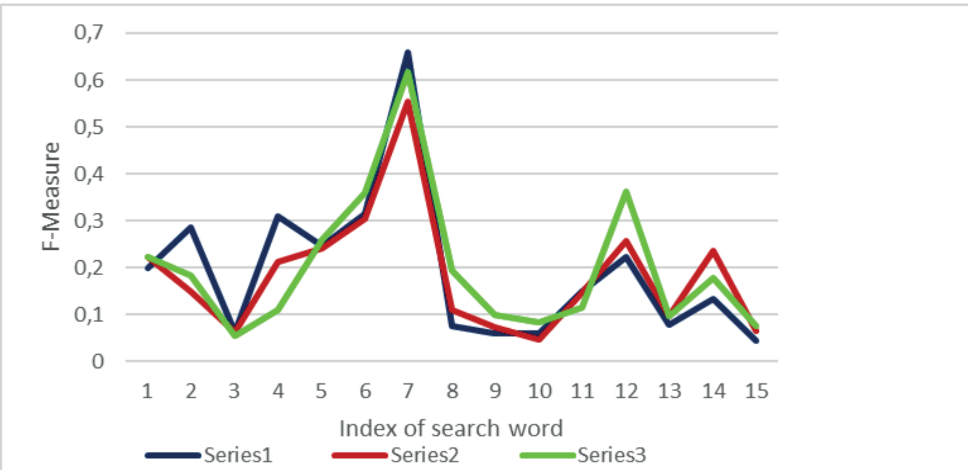


Figure 7. Comparative study of F-Measure values of three Methods (M1:Series1, M2:Series2, M3:Series3) of word searching techniques

https://doi.org/10.56294/dm202554

After comparing it with existing state-of-the-art, figure 4, figure 5, and figure 6 show good results compared to existing algorithms.

Error Analysis

Let, n_1:= the actual number of search words in the document.

And, n_2:= the number of fetched target words.

Error $\Delta = n_1 - n_2$

Ideally $\Delta=0$ (i.e., $n_1=n_2$), when the accuracy level is 100 %. In other words, the model can recognize all the words successfully. If $n_1>n_2$ we call it as positive error and if $n_1<n_2$ it is known as negative error.

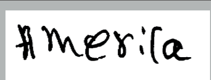
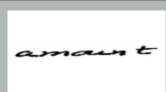
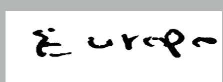
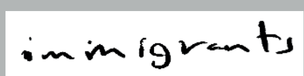
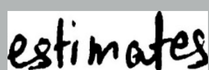
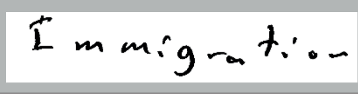
In our experimental setup we get the value of Δ for different SWs in different metrics, among those the result of DTW is shown in table 12.

Table 12. Error calculation (Δij) table over DTW for different SWs

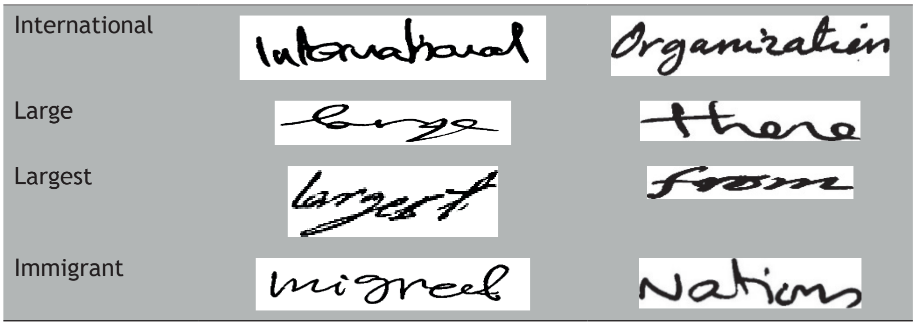
	DTW			
	Actual in PSW	**No. of Retrieved**	**No. of Correctly Retrieved**	**No. of Wrongly Retrieved**
SW1	816	580	71	509
SW2	1245	337	31	306
SW3	1529	1084	31	1053
SW4	1192	13	10	3
SW5	339	287	49	238
SW6	303	224	52	172
SW7	100	30	23	7
SW8	1582	251	19	32
SW9	1391	667	36	631
SW10	1077	1030	36	994
SW11	740	560	56	514
SW12	680	164	75	89
SW13	838	361	19	342
SW14	1854	393	34	359
SW15	1637	651	27	624

In case of searching a word image, some wrong word image as a searched word as well as some unbelievably good result are retrieved which is deformed badly. In table 13 we can observe some of these examples.

Table 13. Retrieved (correct, wrong) word images

Search Word	Successful Retrieval	Unsuccessful Retrieval
America		
Asia		
Asian		
Europe		
Immigrants		
Immigration		

https://doi.org/10.56294/dm202554



CONCLUSIONS

Information retrieval from historical, ancient, or current handwritten document images may be better with word spotting than recognize-then-retrieve. Matching word pictures quickly and accurately could enable word collections with reasonable accuracy and minimal human interaction. Word spotting can automatically identify found words, allowing for more cautious use of expensive human labor for transcription. This article shows a two-stage approach to word searching in handwritten document photos. We remove irrelevant word images in pre-selection. Pre-selected candidate words from a document image are voted on while confirming. The current model generates satisfactory results. Validated words are placed in a folder as search words after voting. Trials show that the two-stage word searching method for handwritten document images works well. Despite this success, there are still challenges. Word searching can be done with high-dimensional feature extraction. Implementing a feature selection technique would improve the suggested approach's retrieval performance. Context-sensitive features from the first stage of this study may be used later. These findings suggest future investigation in several areas. Improvements to this work may help overcome scalability issues like various handwriting patterns, document structures, and structure differences within words. It will make digitizing handwritten documents easier in the future.

BIBLIOGRAPHIC REFERENCES

1. Bhowmik S, Malakar S, Sarkar R, Basu S, Kundu M, Nasipuri M. Off-line Bangla handwritten word recognition: a holistic approach. Neural Comput Appl. 2019;31:5783–98.

2. Basu S, Das N, Sarkar R, Kundu M, Nasipuri M, Basu DK. A hierarchical approach to recognition of handwritten Bangla characters. Pattern Recognit. 2009;42(7):1467–84.

3. Rath TM, Manmatha R. Word spotting for historical documents. Int J Doc Anal Recognit. 2007;9:139–52.

4. Begum N, Goyal A. Analysis of legal case document automated summarizer. In: 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC). IEEE; 2021. p. 533-8.

5. Sharma S, Choudhary S, Sharma VK, Goyal A, Balihar MM. Image watermarking in frequency domain using Hu's invariant moments and firefly algorithm. no April. 2022;1–15.

6. Rath TM, Manmatha R. Word image matching using dynamic time warping. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003 Proceedings. IEEE; 2003. p. II-II.

7. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee; 2005. p. 886–93.

8. Zagoris K, Ergina K, Papamarkos N. A document image retrieval system. Eng Appl Artif Intell. 2010;23(6):872–9.

9. Retsinas G, Louloudis G, Stamatopoulos N, Gatos B. Efficient learning-free keyword spotting. IEEE Trans Pattern Anal Mach Intell. 2018;41(7):1587–600.

10. Pantke W, Dennhardt M, Fecker D, Märgner V, Fingscheidt T. An historical handwritten arabic dataset for segmentation-free word spotting-hadara80p. In: 2014 14th International Conference on Frontiers in Handwriting Recognition. IEEE; 2014. p. 15–20.

11. Rusiñol M, Aldavert D, Toledo R, Lladós J. Efficient segmentation-free keyword spotting in historical document collections. Pattern Recognit. 2015;48(2):545–55.

12. Malakar S, Ghosh M, Sarkar R, Nasipuri M. Development of a two-stage segmentation-based word searching method for handwritten document images. J Intell Syst. 2019;29(1):719–35.

FINANCING

No financing.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Data curation: Ankur Goyal, Pronita Mukherjee, Dipra Mitra, Shiv Kant, Khalid Almalki, Suliman Mohamed Fati.

Methodology: Ankur Goyal, Pronita Mukherjee, Dipra Mitra, Shiv Kant, Khalid Almalki, Suliman Mohamed Fati.

Software: Ankur Goyal, Pronita Mukherjee, Dipra Mitra, Shiv Kant, Khalid Almalki, Suliman Mohamed Fati.

Drafting - original draft: Ankur Goyal, Pronita Mukherjee, Dipra Mitra, Shiv Kant, Khalid Almalki, Suliman Mohamed Fati.

Writing - proofreading and editing: Ankur Goyal, Pronita Mukherjee, Dipra Mitra, Shiv Kant, Khalid Almalki, Suliman Mohamed Fati.