RESEARCH ARTICLE

Z-Transform-Based Profile Matching to Develop a Learning-Free Keyword Spotting Method for Handwritten Document Images

Debanshu Banerjee1 · Pratik Bhowal2 · Samir Malakar3 · Erik Cuevas4 · Marco Pérez-Cisneros5 · Ram Sarkar6

Received: 21 March 2022 / Accepted: 10 October 2022 / Published online: 2 November 2022 © The Author(s) 2022, corrected publication 2022

Abstract

For easy accessibility of the information from the digitized document images, optical character recognition (OCR)-based software can be used. But in the case of handwritten documents, the performance of the state-of-the-art OCR systems is not satisfactory owing to the complexity of the unconstrained handwriting. Hence, research affinity comes up with an alternative solution for this problem called keyword spotting (KWS) which is much more practical than an OCR-based solution. This work proposes a novel learning-free KWS method that can be applied to a heterogeneous collection of handwritten documents. In this work, we introduce a new way of profile matching to compare the query word profiles (i.e., both upper and lower) with the target words’ profiles. At first, both query and target words are binarized, and then two profiles from each such word are generated. Next, we formulate rules to filter out the irrelevant words concerning the query word and obtain the probable candidate query (i.e., target) words. Then we compare the profiles of the query and candidate query words in the Z-transform domain using the condition of resonance for the damped oscillator. However, before the match, we perform an affine transformation on the Bezier curve representation of the profiles of the candidate query words to reduce the effects like scaling, rotation, and shearing which might occur due to the variant writing styles of individuals. The proposed method achieves satisfactory performance compared to state-of-the-art learning-free methods when applied to four publicly available standard datasets namely ICFHR 2014 H-KWS competition Modern, IAM, ICFHR 2016 H-KWS competition Botany and ICFHR 2016 H-KWS competition Konzilsprotokolle datasets.

Keywords Keyword spotting · Z-transform · Handwritten document image · Affine transform

Abbreviations

| Abbreviation | Description |
|--------------|-------------|
| OCR          | Optical character recognition |
| KWS          | Keyword spotting |
| H-KWS        | Handwritten keyword spotting competition |
| ICFHR-2014   | 14th International Conference on Frontiers in Handwriting Recognition |
| QBE          | Query by example |
| QBS          | Query by string |
| DTW          | Dynamic time warping |
| HOG          | Histogram of oriented gradients |
| POG          | Projection of oriented gradients |
| mPOG         | Modified version of POG |
| LBP          | Local binary pattern |
| SVM          | Support vector machine |

1 Department of Metallurgical and Material Engineering, Jadavpur University, Kolkata, India
2 Department of Instrumentation and Electronics Engineering, Jadavpur University, Kolkata, India
3 Department of Computer Science, Asutosh College, Kolkata, India
4 Departamento de Electrónica, Universidad de Guadalajara, Guadalajara, Mexico
5 División de Tecnologías Para La Integración Ciber-Humana, Universidad de Guadalajara, Guadalajara, Mexico
6 Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
1 Introduction

Possibly the first way of documented communication was through handwriting. Even in this digital era, many of us still prefer to communicate through handwriting [1, 2]. In many places such as educational institutes, court premises, police stations, and administrative offices handwritten documents are widely used. These documents are prepared for interpersonal communication, for conveying messages, facts, etc. However, such documents are mostly kept for future use and thereby rarely destroyed. As a result, the volume of such documents is increasing with each passing day. Besides, due to a lack of proper management, the quality of the documents gets degraded as time passes.

In the light of the above facts, digitization of handwritten documents has become of paramount importance. Moreover, the recent trend of paperless offices demands digitization of such documents and hence in most cases documents are scanned and stored in image format. One such initiative is found in digital libraries where some old manuscripts are made available to common users. However, a document required in a rush, is not easily searchable from the online copies as the tagging of the scanned documents is made manually and mostly subject-specific. Hence, there is a pressing need to understand that the underlying texts present therein are automatically indexed by the computing devices and present the documents accordingly to a wide usage before the common users. An optical character recognition (OCR) system that can convert the underlying text in handwritten documents into their corresponding machine-encoded form is the best solution to this problem. In this context, it is worth mentioning that the existing OCR systems do not perform satisfactorily for handwritten texts due to the extreme writing style variation of the individuals [3, 4]. Besides, segmentation ambiguity becomes a prevalent issue when handwriting is cursive [5, 6]. Another notable problem with OCR-based solutions is that it tries to convert each word present in the document which is often unnecessary and makes the problem time-consuming and error-prone [7, 8]. Hence, a need for an alternative solution arises that can deal with the above-mentioned problem.

Handwritten document image processing research affinity comes up with an alternative technique for this problem called keyword spotting (KWS) which is much more practical and faster than that of an OCR-based solution [8, 9]. In this technique, one aims to retrieve only specific word(s) (called query word given by the user) from the document images without trying to decipher the entire document and hence it executes faster. More specifically, it compares the words of a document image (target words) with the query word and ranks the target words based on their similarity score with the example word image representing the query word, hereafter we call such image as query word only.

In the literature, KWS techniques are categorized based on three different aspects. First, based on the input type of the query word, KWS techniques are categorized into two approaches, namely query by example (QBE) [4, 8] and query by string (QBS) [8, 10]. If a user inputs the query word as an image to the KWS system then the process is following the QBE approach. On the contrary, if a query word is provided as a string (maybe feature string) to the KWS system then it is called to follow the QBS approach.

Second, based on the type of target from which a query word is to be searched one can classify the KWS system as segmentation-free [11–13] and segmentation-based [4, 7, 14, 15] systems. In the first case, the keyword is spotted on the entire document image, whereas in the latter case the query word is searched from the segmented parts of a document image i.e., in the segmentation-based method, segmentation of the document page images into characters, words or text lines is performed beforehand using some page segmentation techniques [16]. The segmentation-based methods are computationally more efficient than segmentation-free methods, as these methods focus on specific parts of the document rather than looking at the entire document. In this context, it is to be noted here that one can segment a document image into constituent text lines, words, and characters using techniques like [17–19].

Third, depending upon the underlying matching protocol used between query word and target word, we can classify a KWS technique as learning-free [4, 20, 21] and learning-based [22, 23]. As the name suggests, in a learning-based method, the model is first trained on a large image database and then searches the target word set. In learning-free methods, first, the same set of features are extracted from a query word as well as target words, and then a similarity score is calculated by some means. Finally, based on the similarity score a target word is confirmed as a query word. Learning-free methods are practical in a true sense as multiple copies for a query word are required while spotting. It is also faster compared to learning-based methodologies. Apart from these, it is also found that learning-based methods perform well when they are trained and tested on the same dataset but with the varying train and test datasets, their performances diminish [24]. These are the reasons behind choosing a learning-free approach in this work.

| HMM | Hidden Markov model |
| BoW | Bag of words |
| SIFT | Scale invariant feature transform |
| MLP | Multilayer perceptron |
| CNN | Convolutional neural network |
1.1 Motivation

The present work chooses a learning-free, segmentation-based, and QBE approach for KWS tasks. This work is motivated by several factors as described below.

- First, the KWS is considered by the researchers as this is much more practical and faster than the OCR-based solution for document retrieval [8, 9], and it has various real-life applications.
- Second, after analyzing the different KWS approaches found in the literature, we observe that the key objective of any KWS system is to come up with a better word matching technique following either learning-free or learning-based approaches. However, the most important pre-requisite of a learning-based method is a voluminous annotated training data [25–27], and preparing the training data is a cumbersome task. Moreover, their generalization is poor compared to learning-free approaches [4, 21, 28 i.e., a model trained on one dataset may perform poorly (generating poor results compared to the learning-free approach) on another dataset if not trained again [25, 28]. These are the reasons that guide us to develop a learning-free KWS system.
- Third, profile-matching-based methods [9, 20, 29, 30], theoretically and analytically sound, faster and perform well in a single-writer scenario, printed documents, etc. However, they fail to perform satisfactorily for multi-writer handwritten documents. These facts motivate us to accumulate the possible reasons behind this and provide solutions accordingly.
- Lastly, several KWS methods [7, 9, 27] used some fast filtering techniques to discard largely dissimilar words concerning a given query word from the target word set. These approaches reduced the number of comparisons required for KWS and consequently reduced the spotting time with improved performance [27]. Following these, we have designed two rule-based proposals to filter out irreverent word images.

1.2 Objectives and Contributions

The major objective of this work is to design a learning-free KWS method when resolving some intrinsic issues encountered by many profile-based handwritten KWS systems in the past. Certainly, one of our primary objectives is estimating finer upper and lower profiles as the success of the entire process is largely dependent on this. However, several issues appear during the profile generation process described in Sect. 3.2, and solving them suitably is another set of objectives of the present work. Consequently, some measures have been taken when extracting finer profiles (see Sect. 3.2) and such measures can be considered as some of the contributions of our work. The 2D Z-transform based profile matching technique is a bit time consuming process as its time complexity is $O(W^2)$, where $W$ is width of a word image (see Subsect. 3.5). Thus eliminating completely different target words that seem to have the same basic structure as a query word before employing 2D Z-transform based profile matching process would be of great help. Thus, the next objective is to design an effective and faster process to reject such target word images. To encounter this we have developed two novel pre-selection methods which is another contribution of the proposed work. This preselection stage reduces not only the overall execution time taken by the method to identify the instances of query words present in the target word set but also reduces the chance of selecting false instances for a given query word. We call the reduced target set a probable candidate query word set.

The next requirement of the work is an efficient profile matching technique which is the major objective since failure in fulfilling this objective means the selection of a large number of probable candidate query word images as a keyword. In this work, for the first time, we have applied the 2D Z-transform-based [31–33] string matching technique that calculates the similarity between two profiles (upper or lower) extracted from a query and a word from the probable candidate query word image set. The notable problem that is faced by the 2D Z-transform-based profile matching scheme is that it would return same word written differently as different word. Writing same word differently is very common since different writers (or even the same writer) write the same word in different ways. To deal with this problem we apply affine transform on the profiles extracted from the dense region of the word image before passing them to the Z-transform-based similarity measure. Intuitively, even if the same words are written differently, their basic structure would remain the same with minor variations in the dense region of the word image which is the major motivation behind choice of such region for profile extraction. Since the affine transform adds slight variations in curves, we use it to make two curves that seem different but have the same basic structure, look similar, so that the similarity score between these two curves becomes high. Such generalization is missing in most of the non-learning-based models. Lastly, it is to be noted here that combining all the corrective measures and designing an effective KWS system is another objective as well as contribution of the work.

In summary, the main highlights of our work are as follows:

- We have developed two novel pre-selection methods which remove different words having dissimilar structures.
• We have taken several measures, many of which are of the first kind, to extract finer upper and lower profiles of word images.
• We have developed an affine transform and 2D Z-transform-based method for evaluating the similarity between the upper and lower profiles generated from the query words and the target words.
• We have evaluated our model on four public datasets and our model performs better when compared with the state-of-the-art methods employed profile matching technique.

1.3 Organization of the Manuscript

The rest of the article is prepared as follows. Section 2 briefly describes the past methods where we focus mainly on learning-free methods, though few significant learning-based methods are also briefed here. Section 3 describes the proposed method, while Sect. 4 reports the results obtained by our method on four publicly available standard datasets. In Sect. 5, we conclude by referring to our scope for improvement.

2 Literature Survey

Many works could be found in the literature that dealt with KWS in handwritten documents. Looking at the nature of these works, we can categorize them differently as discussed earlier. However, we have categorized the existing works into two categories: learning-free methods and learning-based methods.

2.1 Learning-Free Methods

Learning-free methods do not consider any train samples to learn a model by which word matching could be performed. To develop a learning-free KWS method authors, in general, first extracted features like profile information, shape, structure, and texture and then employed some feature similarity measure techniques to perform KWS. Here we discuss some such initiatives from the past. Rath and Manmatha [20] first extracted the profiles (i.e., upper and lower) and then applied dynamic time warping (DTW) for finding the similarity among the profiles. In another work, Meshesha and Jawahar [29] combined structural features with the profile information for matching the target and query words using DTW. Howe [30] further extended this work by introducing a flexible inkball model that considers a set of connected nodes corresponding to text strokes. In this work, the computational effort is shifted to the matching procedure, which is performed using an iterative energy minimization algorithm. Recently, Majumder et al. [9] have proposed a voting-based technique for KWS using a QBE setup. In this work, the profiles (upper and lower) are partitioned into 1 (i.e., the original), 2, and 4 equal parts, and then each part is matched using DTW. The decisions from each level are combined using a voting strategy to make the final decision. The main problem of these techniques (i.e., profile matching-based methods) is that slight changes in handwriting due to skew and slant have a huge impact on the matching score it returns. Thus, these methods need a preprocessing technique like slant and skew corrections at the word level. But such a trial cannot handle the scaling effect on handwriting words and, the skew and slant corrections are error-prone [34]. Moreover, when a word image is formed off many connected components then it is critical to estimate the profiles [9, 21].

Due to these problems, researchers tried to search for better and more powerful features. Consequently, a good number of methods found in the literature that made use of directional features like the histogram of oriented gradients (HOG) [14, 35], slit style HOG [36], local binary pattern (LBP) [14], projection of oriented gradients (POG) [37] and modified version of POG (mPOG) [4], pyramid histogram of oriented gradients (PHOG) [38], angular features [21], oriented basic image features (oBIFs) [15], and document-oriented local features (DoLFs) [39] to perform KWS using the learning-free approach. Rodriguez-Serrano and Perronnin [35] used two feature similarity measure techniques viz., hidden Markov model (HMM) and DTW but found HMM as superior. Later, Terasawa and Tanaka [36] proposed the slit style HOG feature which is gradient-distribution-based with overlapping normalization and redundant expression, and observed better performance over the HOG feature when using DTW to measure feature similarity. In the work, Kovalchuk et al. [14] extracted HOG and LBP features first and then reduced the dimension of the feature vector using the max-pooling method. During feature reduction through the max-pooling method, the authors first portioned the feature vector into several equal-sized parts and then considered the index-wise maximum feature value. Finally, the reduced feature vectors were used for word matching for which the authors relied on Euclidian distance.

Observing the effective use of gradient-based features in many real-life applications Retsinas et al. [4, 37] designed POG [37] and mPOG [4] feature descriptors to perform KWS in historical document images in a segmentation-based approach. In the work [37], the authors extracted POG features from the entire word image and vertical word image segments to obtain global and local information about the query and target word images and named them gPOG and IPOG, respectively. They used Euclidian distance to measure similarity between features. Finally, they introduced a fusion mechanism using IPOG and gPOG that defined the similarity score between target and query words using the weighted distance method and called this method fPOG. However, in the work [4], the authors not only introduced mPOG but...
also designed a selective matching (SM) algorithm to measure similarity between two features to decide the number of vertical segments for query and target words. The SM algorithm was trained on a small trained set to fix the segment counts which varied from target word to query word. Finally, they performed the KWS task using the multi-instance SM (MISM) technique and augmenting the middle-zone segment called as PSeg. Bhunia et al. [38] employed zone-based foreground–background information for keyword spotting in handwritten documents written using Bangla and Devanagari scripts. In their work, they used the PHOG feature descriptor and HMM-based scoring technique for spotting the keywords in a text line. Recently, Kundu et al. [21] introduced Hough transform-based angular features while Youssfi et al. [15] used the oBIF descriptor. In the work [21], the authors relied on DTW-based feature matching whereas, the authors of the work [15] investigated several feature similarity measures like City-block distance, Euclidean distance, Hamming distance, Cosine similarity, correlation metric, Spearman index, and Chebyshev and found that City-block distance metric outperformed others. Although these works used directional features, the authors preferred to use slant and skew correction of the word images [34] before extraction of the features (e.g., [4, 21, 37]).

Apart from these, several bag-of-visual-words (BoVWs) models, a popular framework that is empowered by local gradient features like scale-invariant feature transform (SIFT) and HOG, were also used in the literature for building a KWS system in a learning-free way. Recently, Rothacker [26] used SIFT feature descriptor to design a BoVW model which was used to spot a keyword without any annotated samples. They built several hypotheses in a bottom-up approach that is they first employed the text hypothesis and then used the line hypothesis to perform keyword spotting in the document images. In another work, Konstantinios et al. [39] first extracted DoLFs from word images and then quantized them using BoVW methods. In other words, the authors used DoLFs instead of SIFT or HOG, generally used in BoVW models earlier. Next, they employed the quantitative near neighborhood search (QNNS) technique to cluster visually similar words present in the target word set. Finally, the distances of the cluster centers from a query word were used to perform KWS tasks. Aldavert et al. [40] provided a detailed survey consisting of the methods that employed the BoW framework-based method.

### 2.2 Learning-Based Methods

Although our technique is not directly comparable with the learning-based method, we describe some recently proposed learning-based methods to get a quick overview of the complexity of the problem. Malakar et al. [7] used a holistic word recognition approach to perform word spotting. In this work, the authors have relied on HOG and topological features and used a multilayer perceptron (MLP)-based classifier to retrieve query words from the target word sets. They introduced a filtering approach to reject largely dissimilar target words before employing exact matching-based keyword searching. In this work, only 15 keywords were considered which are very specific in practical scenarios. Recently, several researchers used graph-based methods [27, 41, 42] for KWS purposes. Stauffer et al. [41] used graph-edit distance to select some of the target words as the query word. In their work, they first converted each word image using four different word-graph representation techniques and then employed a graph ensemble method (they investigated five different graph ensemble methods) to obtain the final word-graph templates. In another work, Ameri et al. [42] used Hausdorff edit distance (HED) to estimate the similarity between target and query word graphs. Apart from these, recently, Stauffer et al. [27] used a fast and inexact graph similarity measure technique, known as the polar graph dissimilarity (PGD) metric, to filter out a sufficient number of target word-graphs before using exact graph edit distance (GED) based similarly measure technique likewise the methods reported in [7, 9]. They represented a polar graph using node-based histograms and edge-based histograms and PGD was calculated using $\chi^2$ distance. Later, they defined the similarity between a query word graph and a non-rejected word graph using bipartite graph edit distance. All these methods [27, 41, 42] used a small-sized training sample to optimize different parameters associated with their models.

As deep learning becomes more and more popular nowadays, we can observe a paradigm shift in the domain of pattern recognition and image processing. Following the same convention, recently, several different convolutional neural network (CNN) architectures [3, 23, 24, 28, 43] were proposed that embedded deep features into different textual embedding spaces defined by the pyramidal histogram of characters (PHOC) encoding. For example, Sudholt and Fink [24] proposed an architecture that directly embeds image features to PHOC attributes and has a sigmoid activation in the final layer, referred to as PHOCNet. Sudholt and Fink [23] introduced a temporal pooling layer in PHOCNet and called the new model as TPP-PHOCNet, whereas deep features were embedded with PHOC representation in the works [3, 22]. Krishnan et al. [3] used HWnet while HWnet v2, an improved version of HWNet, was used by Krishnan and Jawahar [22] to extract deep features. Recently, Boudraa et al. [44] designed a new CNN architecture using U-Net and PHOCNet (PUNet) architectures employed one after another. First, the output layers of U-Net architecture were improved by introducing transfer learning, and then these features were fed to the actual PHOCNet model. A semi-supervised learning scheme was introduced by Wolf and Fink [25] for KWS tasks. In their work, they used the deep
features extracted using TPP-PHOCNet fine-tuned using synthetically generated word samples. Thus, they called the model annotation-free method. They introduced a confidence measure scheme to avoid inaccurate attributes in their semi-supervised learning strategy.

There exist some recent works that did not include PHOC encoding in their model. For example, Sfikas et al. [45] used a pre-trained CNN model to extract features from vertically segmented zones. The features extracted from the zones were then concatenated to form a single feature vector and then the nearest neighbor searching method was used to retrieve instances of a query word. In another work, Cheikhrouhou et al. [46] proposed a bi-stage script-independent KWS system using a hybrid model consisting of bi-directional long-short term memory (BiLSTM) and HMM (BiLSTM-HMM). In the first stage, they performed script identification where four different HMM models were introduced to understand writing directions (left–right and right–left) and writing styles (handwritten and printed). Finally, in the second stage script and writing style, specific KWS techniques were employed. In another work, Daraei et al. [47] used the Monte-Carlo dropout network to avoid the uncertainty that may arise during the training process due to the use of fewer training samples. During the training time, they estimate the certainty quotients among the query and target words, which was later used for word spotting purposes. Recently, Kumari and Sharma [48] described many such deep learning-based methods and their effect on the KWS.

3 Present Method

The present KWS method follows a learning-free and segmentation-based approach. It has four key steps: preprocessing, profile extraction, pre-selection of target words as probable candidate query words, and matching of a query word with the reduced set of probable candidate query words. The image datasets are first preprocessed. The dense region of the word is estimated to generate the profiles of the word. In the following subsections, we first describe the dense region estimation method, and then we describe the profile extraction technique.

3.1 Preprocessing

Our profile extraction technique works for binarized word images. To obtain the noise-free binarized word images, we use the noise removal techniques [49, 50]. Next, we enclose all the data pixels in a word image using a minimal bounding box and let, such word image (say, $I$) is represented using Eq. (1).

$$I = \{ f(i, j) : i \in [1, H] \land j \in [1, W] \}$$

In Eq. (1), the variables: $H$ and $W$ represent the height and width of the minimal bounding box containing all the data pixels of the word image respectively, and $f(i, j) \in \{0, 1\}$ where ‘1’ and ‘0’ represent data and non-data pixels of $I$, respectively.

3.2 Estimating the Upper and Lower Profiles

In this work, we use the upper and lower profiles of the word images to perform KWS tasks. Before extracting the profiles, we determine the dense region of a word image, i.e., the region with the maximum concentration of data pixels of the word. In the following subsections, we first describe the dense region estimation method, and then we describe the profile extraction technique.

3.2.1 Estimation of the Dense Region

We use the dense region of a word image for the profile generation and the pre-selection of target word images as probable candidate keywords purposes. Such choice is guided by some factors like (i) the dense region is the most informative region of any word image [4, 21], (ii) it can minimize the effect of elongated portions of middle zone characters appearing in the writing style of many individuals [9], and (iii) it can avoid the misleading peak and valley
in the profiles (upper and lower) due to the presence of different letters having upper and lower zone parts [2] at the same position.

The idea behind dense region estimation is that the number of times a line drawn parallel to the length of the image cuts the word segments more the probability of the line being in a dense region is more. It is usually the central position of the image. The number of such cutting points is calculated for each row of the word image. Let, $c_{si}$ be the number of cutting points for a specific row $i \in [1, H]$. We calculate $c_{si}$ using Eq. (2).

$$c_{si} = |\{ j | f(i, j+1) - f(i, j) \neq 0 \}|, \forall i \in [1, H] \quad (2)$$

Next, we calculate the mean crossover ($\mu_{cs}$) which is the average of the crossovers for all lines drawn parallel to the length of the image using Eq. (3). The number of such lines is equal to the height of the image under consideration.

$$\mu_{cs} = \frac{1}{H} \sum_{i=1}^{H} c_{si} \quad (3)$$

Now we define the dense region with the help of the upper (say, $dub$) and the lower (say, $dlb$) boundary of the dense region. $dub$ and $dlb$ divide the image into three parts, namely the upper region, the dense region, and the lower region. Word image that contains no elongated part in the upper or lower portion of the word image is entirely confined within the dense region. The dense region is confined within two parallel straight lines lying along the length of the image, one differentiating the elongated data pixels protruding upwards from the dense region while the other differentiates the elongated data pixels of the word protruding downwards from the dense region. To estimate the $dub$ and $dlb$, we traverse the word image from the top of the image towards the bottom and list down all the row numbers for which the number of crossovers is greater than the average crossover (i.e., $\mu_{cs}$) and then we use the formula defined in Eqs. (4) and (5), respectively, to estimate them. We show examples of two estimated boundaries in Fig. 2.

$$dub = \min \{ i | c_{si} > \mu_{cs} \} \quad (4)$$

$$dlb = \max \{ i | c_{si} > \mu_{cs} \} \quad (5)$$

### 3.2.2 Profile Generation

In this section, we describe different issues that occur during the estimation of profiles (both lower and upper) of a handwritten word image. While estimating a profile, we observe four different cases.

- **Observation 1:** The entire word is singly connected (see Fig. 3(a)).
- **Observation 2:** A word image contains two or more connected components (see Fig. 3(b)) and if we look from
the top of the word image then we can see the bottom line of the word images through the gap between the two horizontally placed consecutive connected components.

- **Observation 3**: A word image contains two or more connected components (see Fig. 3(c)) and if we look from the top of the word image then we cannot see the bottom line of the word image through the gap between two horizontally placed consecutive connected components due to overlapping parts of the connected components in the upper zone.

- **Observation 4**: A word image contains two or more connected components (see Fig. 3(d)) and if we look from the top of the word image then we cannot see the bottom line of the word image through the gap between two horizontally placed consecutive connected components due to overlapping parts of the connected components in the lower zone.

For the first and the third observations, we do not face any error while estimating the upper profile. But, the second and fourth cases give us erroneous results, as while estimating the upper profile we are not interested in how the lower half of the image varies. To deal with this situation, we start scanning from the top of the image to the bottom, and the first point where we encounter a data pixel is considered the coordinate of the profile provided it is above the lower boundary of the dense region of the image. But if we find that the first data pixel that we encounter is in the lower part of the dense region then we take the row value of the last data pixel which is above the lower boundary of the dense region. After we have traversed the entire image in this way from left to right, we get the upper profile of the image.

Let, $L_j$ is the set of data pixels in the $j^{th}$ column of a word image and $w_{up,j}$ is the smallest value in $L_j$. Therefore, the coordinate of such a point is $(w_{up,j}, j)$. $L_j$ and $w_{up,j}$ are estimated using Eqs. (6) and (7), respectively. Equation (8) refers to the fact that the point is itself part of the upper profile if it lies above the lower boundary (i.e., $dlb$) of the dense region, else the y coordinate of the last point which is located above the $dlb$ is taken along with $j$ and so the new point included in the upper profile becomes $(w_{up,j-1}, j)$. In Eq. (8), $F_{up}(i,j)$ represents the upper profile of the word image. Examples of the estimated upper profiles of the example word images are shown in Fig. 4(a-b).

\[
L_j = \{ i | l(i,j) \neq 0 \land j \in [1, W] \} \quad (6)
\]

\[
up_j = \min \{ L_j \} \quad (7)
\]

\[
F_{up}(i,j) = \begin{cases} 
    (w_{up,j}, j), & \text{if } w_{up} \leq db \\
    F_{up}(i,j-1), & \text{otherwise}
\end{cases} \quad (8)
\]

We find the same set of observations while estimating the lower zone of the word image. This time we traverse the image from the bottom to the top along the line parallel to the breadth of the image and continue to do so from the left to the right of the image. Let, $L'_j$ is the set of data pixels in the $j^{th}$ column of a word image, and $l_{pb,j}$ is the largest value in $L'_j$. Therefore, the coordinate of such a point is $(l_{pb,j}, j)$. $L'_j$ and $l_{pb,j}$ are estimated using Eqs. (9) and (10), respectively. Equation (11) refers to the fact that the point is itself part of the upper profile if it lies below the upper boundary (i.e., $dup$) of the dense region, else the y coordinate of the last point which is located above the $dup$ is taken along with $j$ and so the new point included in the lower profile becomes $(l_{pb,j-1}, j)$. In Eq. (11), $F_{lp}(i,j)$ represents the upper profile of the word image. We show examples of estimated lower profiles of the example word images in Fig. 5(a–b).
Pre-selection of Probable Candidate Query Word Images

Pre-selection of the images is important because of the following reasons:

(a) It removes the images which are different but seem to have the same basic structure and applying affine transform on the profiles may lead to a high similarity score when calculating the similarity score using the 2D Z-transform.

(b) In actual implementation, we need to compare an example query word with a large collection of target word samples, which in turn increases the overall execution time. However, in a practical scenario, many target words are very irrelevant to the query word (e.g., comparing the words “Asia” with “International” or “Hello” with “a”). Hence, this step lessens the need for unnecessary comparison, by removing the irrelevant words, we perform during matching of a query word image with a collection of target word images described in Sect. 3.4. This not only saves an ample amount of execution time during matching but also helps in retrieving the correct set of target word images for a given query word image.

To illustrate the point just mentioned about the importance of pre-selection, we use Fig. 6. In the figure, it can be seen that since the upper part of the letter “p” has not been extended properly, its profile would look very similar to that of the letter “q”. When affine transform would be applied to the profiles, they would look very similar. Here, we use pre-selection to remove such similar words from the target dataset. Although it is not always possible to remove such target words completely from the target dataset, pre-selection increases the accuracy considerably. In this work, we perform a pre-selection process following two rules described hereafter.

3.3.1 Rule 1: Comparison Between the Average Number of Crossovers in Query and Target Words

Under this rule, we compare the average number of crossovers (i.e., $\mu_{cs}$) present in the target and query words to decide whether the target word under consideration is a probable candidate query word or not. We denote the number of average crossovers for query word and target word by $\mu_{cs,query}$.
and \( \mu_{cs,\text{target}} \), respectively. Now we set the rule as, value of \( \mu_{cs,\text{target}} \) must be in the range of \( \pm 3 \) that of \( \mu_{cs,\text{query}} \) for being the target word under consideration as a probable candidate query word. To set the threshold \( (i.e., \pm 3) \) we study the common pattern of average crossover counts of the different letters of the English alphabet. From our study, we notice that, in general, variation in writing styles for a letter affects the \( \mu_{cs} \) value by an amount of 3. In Fig. 7, we show some examples of letters ‘a’ and ‘l’ with \( \mu_{cs} \) value written in different styles. Accordingly, we set the rule that is defined in Eq. (12).

\[
\mu_{cs,\text{query}} - 3 \leq \mu_{cs,\text{target}} \leq \mu_{cs,\text{query}} + 3
\] (12)

### 3.3.2 Rule 2: Comparison Between Numbers of Character Parts Outside the Dense Region

Theoretically, for each part of the word protruding upwards outside the dense region the upper profile cuts the upper boundary of the dense region twice. The same is true in the case of the lower profile also. But this is usually not the case as often texts are written in such a manner that letters that should ideally be confined within the dense region protrude outside it (sometimes protrude outside the dense region considerably) and the profile cuts the boundaries of the dense region unnecessarily (see Fig. 2(a–b)). To handle this issue, we modify the previously estimated \( \mu_{cs} \) (say, \( \mu_{lb} \)) and \( \mu_{ub} \) using Eqs. (13) and (14), respectively.

The idea behind such modification is that words that do not have any protruding upwards have the upper boundary of the dense region within \( \frac{\text{length}}{8} \) (we set the value 8 using a locally conducted study) of the image. Ideally, they should not be in the image at all but the length/8 margin takes care of the practical situation. Similarly, words that do not have any parts or connected components protruding downwards have the lower boundary of the dense region below \( 7 \times \frac{\text{length}}{8} \) part of the image. Similar is the case for words that have parts protruding both upwards and downwards or neither upwards nor downwards. To handle both the practical problems mentioned here, we use Eqs. (13, 14).

\[
\text{mdub} = \begin{cases} \text{um, if } \text{dub} \leq \frac{n}{8} \\ \text{dub} - \left( \frac{n}{8} \right), \text{ otherwise} \end{cases}
\] (13)

\[
\text{mdlb} = \begin{cases} \text{lm, if } \text{dlb} \geq \frac{7\text{en}}{8} \\ \text{dlb} + \left( \frac{n}{8} \right), \text{ otherwise} \end{cases}
\] (14)

Now we calculate the number of times the upper (say, \( U \)) and lower (say, \( L \)) profiles cut the modified upper and lower boundaries of the dense region, respectively. The values of \( U \) and \( L \) are calculated using Eqs. (15) and (16), respectively. We illustrate the value of \( U \) and \( L \) by taking two-word images in Fig. 8.

\[
U_{\text{word}} = |F_{\text{up}}(i, j)| i \geq \text{mdub} \text{ and } i + 1 < \text{mdub, } \forall i \in \text{up}_j | (15)
\]

\[
L_{\text{word}} = |F_{\text{lp}}(i, j)| i \leq \text{mdlb} \text{ and } i + 1 < \text{mdlb, } \forall i \in \text{lp}_j | (16)
\]

Next, using the formulas of Eqs. (15–16), we calculate the number of times \( \text{mdub} \) cuts upper profile a target (say, \( U_{\text{target}} \)) and query (say, \( U_{\text{query}} \)) word images and \( \text{mdlb} \) cuts the lower profile of a target (say, \( L_{\text{target}} \)) and query (say, \( L_{\text{query}} \)).
In this pre-selection stage, we should select only those target words as probable candidate query words which have an equal number of characters protruding upwards (like ‘b’, ‘l’ etc.) and downwards (like ‘y’, ‘p’ etc.) like query word (i.e., \( U_{\text{target}} = U_{\text{query}} \) and \( L_{\text{target}} = L_{\text{query}} \)). However, this ideal case is not always true for real-life data. Therefore, we leave a hard margin of ±2 for taking some soft decisions. The selection rules are defined in Eqs. (17, 18).

\[
U_{\text{query}} - 2 \leq U_{\text{target}} \leq U_{\text{query}} + 2 \tag{17}
\]

\[
L_{\text{query}} - 2 \leq L_{\text{target}} \leq L_{\text{query}} + 2 \tag{18}
\]

### 3.4 Profile Matching

As already mentioned, in this work we use the similarity score between the profiles (upper and lower) of target and query word images. However, in the case of freely written word images, the profiles of the same words rarely match (see Fig. 9) due to variation of individual adapted writing styles. To be more specific, profiles of the same word written by different individuals are variants in translation, rotation, scaling, and shearing. Consequently, a normal similarity measurement technique like DTW [21, 51], Euclidian distance [37], Graph-based Distance [41], and Hausdorff Edit Distance [42] fails to provide the expected similarity score in such cases.

Therefore, in this work, we match the affine transformed profiles for the target word images in the Z-transform space. The idea behind this selection is that the major variations that occur in the word profiles due to different writing styles of individuals can be affected by some composite affine transformation. Therefore, if we can match the profiles in the Z-transformed domain preceded by affine transformation then it is expected to obtain a higher similarity score for the same word (written with different constraints) while this score will remain dissimilar for unlike words. In this section, we first describe the prerequisites for matching two curves (here, either upper or lower profiles of target and query words at one instance) under affine transformation and then the matching technique.

#### 3.4.1 Bezier Curve Generation

Sometimes, word profiles that are generated using the method mentioned in Sect. 3.2.2 contain one or more cusps [52] as evident from Fig. 9 where the curve is not differentiable. In this figure, we mark some of the points (marked within red-colored circles) where the function (i.e., profiles) is not differentiable. But a necessary condition for arc length reparameterization (see Sect. 3.4.2) is that it must be differentiable everywhere in its domain. Hence, we transform the
profiles into Bezier curves that eliminate the issue of non-differentiability. The two discrete functions as obtained from Eqs. (8) and (11) are the mathematical representations corresponding to the upper and lower profiles of a particular word image, respectively, for which Bezier curves are generated.

The Bezier curve is based on the concept of Bernstein basis polynomials [53]. Any \( n+1 \) Bernstein basis polynomial is defined by a binomial distribution as defined in Eq. (19).

\[
b_{v,n}(x) = \binom{n}{v} x^v (1-x)^{n-v}, \text{ where } v = 0, \ldots, n \text{ and } x \in [0,1] \tag{19}
\]

The Bezier curve thus generated from the Bernstein polynomial parameterized by some arbitrary real number \( t \in [0,1] \) with control points \( P_i \in F_{up}, \forall i \in \{0,1, \ldots, n\} \) in case of the upper profile and \( P_i \in F_{lp}, \forall i \in \{0,1, \ldots, n\} \) in case of the lower profile is shown in Eq. (20).

\[
B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i = \sum_{i=0}^{n} b_{v,n}(t) P_i \tag{20}
\]

The degree of the Bezier curve is one less than the number of points in the profiles. Therefore, for a profile having \( n+1 \) points as in \( F_{up} \) or \( F_{lp} \), a Bezier curve of degree \( n \) is formed. The Bezier curves corresponding to the profiles shown in Figs. 4 and 5 are depicted in Fig. 10.

Considering these curves to be time-domain signals, the derivative of any Bezier curve of degree \( n \) concerning time \( t \) is expressed in Eq. (21).

\[
B'(t) = n \sum_{i=0}^{n-1} b_{v,n-1}(t) (P_{i+1} - P_i) \tag{21}
\]

Now once the Bezier curves corresponding to both profiles for each image present in datasets are generated, arc length reparameterization is performed. Parameterization with respect to arc length takes into account the intrinsic geometric properties such as its curvature implicitly and reduces the cost of feature extraction.

### 3.4.2 Arc Length Reparameterization

Intuitively a particle considered moving through space is considered and the Bezier curves as the locus of that particle when it moves through space. If \( B_x(t) \) and \( B_y(t) \) are the \( x \) and \( y \) components, respectively, of \( B'(t) \), then a notation based on L2-norm is expressed using Eq. (22). Here the Bezier curves are anticipated as time-domain signals. The length traversed by the particle in time \( t \) (here, \( i^{th} \) point on the profile) while traveling along the curve is calculated using Eq. (23).

\[
\| X(t) \|_2 = \sqrt{(B'_x(t))^2 + (B'_y(t))^2} \tag{22}
\]

\[
s(t) = \int_{a}^{t} \| X(t) \|_2 \, dt \tag{23}
\]

The length (i.e., \( s(t) \)) is normalized by the total arc length (say, \( L_{arc} \)), calculated using Eq. (24).

\[
L_{arc} = \int_{a}^{b} \| X(t) \|_2 \, dt \tag{24}
\]

Here \( dt \) is an infinitesimally small period and \( a, b \) are coordinates in \( \mathbb{R}^2 \) indicating the starting and terminating points of the Bezier curves. Now a parameterized curve (say, \( B(s(t)) \)) can be formed using Eq. (25).

\[
B(s(t)) = \frac{s(t)}{L_{arc}}, \forall t \in \{0,1,2, \ldots, n\} \tag{25}
\]

Now each word image is represented by two sets of transformed points (i.e., \( B(s(t)) \)), one corresponding to each profile. Similar words vary according to one’s handwriting due to which the words may be written with a tilt also they may vary in size. Now owing to differences in writing styles, a single word can have several representations as shown in Fig. 9. The Bezier curves corresponding to different profiles for different representations of a single word need to be brought in a similar shape to obtain the best matching score. In other words, the reparameterized Bezier curves (i.e., \( B(s(t)) \)) of a target word image are rotated, sheared or scaled i.e., affected by some affine transformation to the corresponding curves of a query word. The details of affine transformation applied here are discussed in the next section.

### 3.4.3 Affine Transform

For every target word image, a matrix \( P \) is defined as Eq. (26).

\[
P = \begin{bmatrix}
m_{11} & m_{12} & m_{13} \\
m_{21} & m_{22} & m_{23} \\
0 & 0 & 1
\end{bmatrix} \tag{26}
\]
Here $m_i$ s’ are the parameters of affine transformation. An iterative operation is performed on the reparametrized Bezier curves (i.e., $B(s(t))$ s) corresponding to both profiles and at each iteration. Let a point of a curve $(x_{\text{target}}, y_{\text{target}})$ be modified to $(x', y')$ using Eq. (27) at each iteration. Convergence of the iteration is achieved by minimizing the cost function (CF) at each iteration. Let $f(t)$ and $g(t)$ represent the $B(s(t))$ of target and query profiles, respectively. Now, CF is expressed as the point-wise Manhattan distance between the transformed target profile and query profile as shown Eq. (28).

$$[x']^t P [y'] = [x_{\text{target}}]$$ (27)

$$CF = \sum_i |f(mx'_i) - g(x_i)| + |f(my'_i) - g(y_i)|$$ (28)

In this problem, we need to find the parameters $m_{ij}$ such that the CF is minimized. Since there are six $m_{ij}$ (see Eq. (26)), we need to minimize the CF not to one parameter but with six independent parameters. However, the CF is written as a summation of two independent functions, namely, $f(mx'_i)$ and $f(my'_i)$. It is clear that when we differentiate CF to any of the six $m_{ij}$, $g(x_i)$ and $g(y_i)$ would not have any effect on the derivative, rather the terms $f(mx'_i)$ and $f(my'_i)$ would be the only terms to affect the derivative.

To solve this problem, we define a function as shown in Eq. (29).

$$p : R^6 \rightarrow R^2 \text{ such that } p(m_{ij}) = (f(mx'_i), f(my'_i))$$ (29)

Thus, to minimize $p(m_{ij})$, we cannot use the Gradient descent method as it requires the minimization of both $f(mx'_i)$ and $f(my'_i)$ concerning $m_{ij}$. The alternative that we use is the Jacobian Matrix. We express $p(m_{ij})$ in matrix form as

$$P' = \left[\begin{array}{c} f(mx) \\ f(my) \end{array} \right]$$

The Jacobian Matrix of the matrix $P'$, is defined by Eq. (30).

$$J = \begin{bmatrix} \frac{\partial f(mx)}{\partial m_{11}} & \cdots & \frac{\partial f(mx)}{\partial m_{16}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(my)}{\partial m_{61}} & \cdots & \frac{\partial f(my)}{\partial m_{66}} \end{bmatrix}$$ (30)

The dimension of the matrix $J$ is $(2 \times 6)$. We define another matrix $m'$ (see Eq. (31)), with a dimension of $(6 \times 1)$, which contains the values of the parameters $m_{ij}$ after $e$ iterations.

$$m' = \begin{bmatrix} m_{11} \\ \vdots \\ m_{23} \end{bmatrix}$$ (31)

The parameters for the affine transformation are modified according to Eq. (32) with $\delta$ satisfying Eq. (30). $\delta$ is a matrix of dimension $(6 \times 1)$ that is used to modify the matrix $m'$ during every iteration.

$$m^{e+1} = m' + \delta$$ (32)

$$(J^TP)\delta = -JTP'$$ (33)

### 3.4.4 2D Z-Transform

To get the frequency response, $\theta_Z$ of the curves, Z-transformation [31] using unimodular complex numbers, $Z_1 = e^{i\varphi_1}$ and $Z_2 = e^{i\varphi_2}$ is applied as shown in Eq. (34). $\varphi_1$ and $\varphi_2$ represent the angular frequency in radians such that both $Z_1$ and $Z_2$ lie within the region of convergence (ROC).

$$\theta_Z(Z_1, Z_2) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} B(s(t))Z_1^{-n_1}Z_2^{-n_2}$$ (34)

The calculated result in Eq. (35) is a measure of similarity (correlation) between two signals. Now for a single word image, we have two such values, which are convoluted. The convolution operation is expressed as $\theta_{e, u}(Z_1, Z_2)\theta_{e, d}(Z_1, Z_2)$ where $\theta_{e, u}$ and $\theta_{e, d}$ are the similarity measures of the upper and lower reparametrized curves of a word image.

The graphical representation corresponding to the Z-transformation as shown in Fig. 11, resembles the damping nature of an oscillator. Now it is assumed that the
convoluted frequency response of the query word images represents the natural frequency of the signal (reparametrized curve in this case). A situation is assumed where the reparametrized curves for a particular query word image is considered to be the state of resonance. While matching a particular target word image with a query, the affine transformation is performed such that the curves corresponding to the target word images are brought as close to the resonating state as possible.

Resonating condition is defined by the fact that the frequency of a periodically applied external force (i.e., the convoluted frequency response of a target in this case) needs to be in a harmonic proportion with the convoluted frequency response of a query. If $\theta_{Zt}$ and $\theta_{Zq}$ represent the frequency responses of target and query word images, respectively, after convolution, we calculate a score based on the difference between the Golden ratios $\frac{\theta_{Zt}}{\theta_{Zq}}$ and $\frac{\theta_{Zt}+\theta_{Zq}}{\theta_{Zq}}$ for every target word image against a single query word image such that $\frac{\theta_{Zt}}{\theta_{Zq}} - \frac{\theta_{Zt}+\theta_{Zq}}{\theta_{Zq}} < \epsilon$, where $0 < \epsilon \ll 1$. The difference is the measure according to the Golden ratio rule used to measure harmonic proportion shown in Eq. (35).

This difference is sorted in ascending order implying that the target word with the least value of this difference is the best match for the concerned query word image. The lower the value of the difference, the higher is the harmonic proportion between the two signals.

### 3.5 Computation of Time Complexity

Here, we discuss the time complexity of the proposed method when deciding whether a target word is the input keyword or not. The time complexity for the profile generation process is $O(H \times W)$, where $H$ and $W$ represent height and width of the image, respectively. Similarly, for the pre-processing steps, the overall complexity is $O(H \times W)$. This is because the entire image is being traversed during pre-processing and profile generation process. For the Bezier curve generation, the algorithm we use has a complexity of $O(n \log(n))$, where $n$ is the length of the string. In our case, the profile gets generated is of length $W$ and hence, the complexity of Bezier curve generation is $O(W \times \log(W))$. Again, the complexity of affine transformation step is $O(W)$ since, in our case, we have applied affine transformation on a string, i.e., on profiles. For 2D Z-transform, the complexity is $O(W \times W)$, due to the double summation (see Eq. (35)). Hence, the overall complexity of the method would be either $O(W \times W)$ or $O(H \times W)$, depending on weather $H > W$ or vice versa. However, for almost all practical situations and practical words $W > H$ and so the overall complexity may be considered as $O(W \times W)$.

### 4 Experimental Results

The proposed method has been developed for the task of segmentation-based and learning-free QBE KWS. For the evaluation purpose, we use four standard datasets: ICFHR 2014 H-KWS competition Modern [54], IAM dataset [55], ICFHR 2016 H-KWS competition Botany dataset, and ICFHR 2016 H-KWS competition Konzilsprotokolle dataset [56]. Initially, we generate binarized word images using Otsu’s method and then remove the noises that appeared on the word images. Subsequently, the proposed method has been evaluated on these preprocessed datasets. Finally, the outcome of the proposed method has been compared with some state-of-the-art methods that reported results on all the considered datasets. The performance of the word spotting methods is recorded in terms of widely used evaluation metrics, viz., precision at top 5 retrieved words (i.e., P@5) as well as the mean average precision (MAP). The experiments are performed on an Intel® core™ i5-8265U CPU at 1.60 GHz with 8 GB of RAM. In the following subsections, we give a detailed description of the datasets used, evaluation metrics, and the performance comparison with the state-of-the-art methods including the proposed one and error analysis.

#### 4.1 Dataset Description

The proposed method is applied to four publicly available handwritten word datasets that are ICFHR 2014 H-KWS competition Modern dataset [54], IAM dataset [55], ICFHR 2016 H-KWS competition Botany dataset, and ICFHR 2016 H-KWS competition Konzilsprotokolle dataset [56]. A summary of the target and query word sets is provided in Table 1 and three examples of word images

| Dataset                                      | Number of Target words | Number of Query words |
|----------------------------------------------|------------------------|-----------------------|
| IAM                                          | 19,654                 | 580                   |
| ICFHR 2014 H-KWS competition Modern          | 14,727                 | 300                   |
| ICFHR 2016 H-KWS competition Botany          | 16,686                 | 3318                  |
| ICFHR 2016 H-KWS competition Konzilsprotokolle | 9102                  | 3891                  |
from this dataset are shown in Fig. 12. These datasets are briefly described hereafter in this subsection.

4.1.1 ICFHR 2014 H-KWS Competition Modern Dataset

This dataset consists of two variants of images: one contains modern handwriting samples and the other contains samples from historical manuscripts. In our case, we consider the modern part that consists of a total of 100 handwritten document pages from the ICDAR 2009 handwritten segmentation contest. Non-text elements such as lines and drawings have been excluded from these documents. The documents are prepared using four languages: English, French, German, and Greek. The variation of the same word across these 100 documents involves differences in writing style, font size, noise, or a combination of all three. Segmentation of the 100 handwritten documents gives a target set of 14,727 word images and 300 query word images. The query set of each dataset is provided in XML format and it contains word image queries of length greater than 6 and frequency greater than 5.

4.1.2 IAM Dataset

The IAM handwriting database contains forms of handwritten English text that are used for different applications like handwritten text recognition, writer identification and verification, word recognition, form processing, and KWS in the past. In total 657 writers contributed samples of their handwriting to prepare the dataset with 1539 form images. These form images contain isolated and labeled text line images (in total, 13,353), and word (in total, 115,320) images. In addition to this, it also proves a standard division for train, test, and two validation sets of text lines for benchmarking text recognition and writer classification problems. However, no such division is provided for KWS although it has been widely used for experimenting the several KWS methods [4, 8, 37, 45]. Therefore, in our work, we use our division using the train and test text lines labeling.

The train set contains 6,482 text lines having 55,081 number isolated words while the test set contains 2,915 text lines that contain 25,920 word samples. We first list the words coming from the test set (i.e., from 25,920 word samples in the test set) that have multiple copies (here, we consider 10 or more) for the same word. Next, to form the example query word set, we randomly select sample images (2–6 samples for each listed word) from the isolated word set that belong to the text lines of the train set (i.e., from 55,081 number isolated words). We also remove the isolated words in the test set that have been labeled as erroneous segmentation in actual IAM word labeling to form the target word sets. Thus, we use a list of 580 query word images and 19,654 target word images.

4.1.3 ICFHR 2016 H-KWS Competition Datasets

(a) Botany: This dataset is from the Indian Office Records and provided by the British Library. The collection covers the following topics: botanical gardens; botanical collecting; useful plants (economic and medicinal).

(b) Konzilsprotokolle: This dataset belongs to the University Archives Greifswald and it has around 18,000 pages. The collection contains copies of the minutes, written during the formal meetings held by the central administration between the years 1794–1797. The documents belong to the university archives and were digitized and provided by the University Library in Greifswald. Transcripts were provided by the University archives.

For both datasets, a set of page images and two XML files containing the word-level and line-level transcription

---

1 http://vc.ee.duth.gr/H-KWS2014/#Datasets.
2 https://fki.tic.heia-fr.ch/databases/iam-handwriting-database.
3 https://zenodo.org/record/52994#.YpvFLHZByHs.
and segmentation ground truth are given. The word-level bounding boxes of the training pages are obtained using segmentation, which is performed manually by human operators. Each test dataset comprises 20 pages wherein the bounding boxes of all words are manually prepared. The query set of each dataset is provided in UTF-8 plain text format for QBS-based word spotting and word image queries for QBE setup of various lengths and frequencies.

4.2 Evaluation Metrics

It is already mentioned that we follow the most used evaluation metrics in the literature of KWS. The metrics are, viz., precision at top 5 retrieved words (i.e., P@5) as well as the mean average precision (MAP) \[4, 54\] which we describe in the following subsections.

4.2.1 Precision at Top k (i.e., P@k)

In any retrieval system, precision is defined as the fraction of the number of retrieved information that is relevant to the information we need and the total number of retrieved information while \(P@k\), defined in Eq. (36), is the precision value while we retrieve top-k information based on some similarity score. In particular, in the proposed evaluation, \(P@5\) is used which is the precision for top-5 retrieved words. This metric defines how successfully the algorithms produce relevant results to the first 5 positions of the ranking list.

\[
P@K = \frac{|\{\text{relevant words}\} \cap \{\text{top} - k \text{ retrieved words}\}|}{|\{\text{top} - k \text{ retrieved words}\}|}.
\]

4.2.2 Mean Average Precision

The second metric used for the evaluation of proposed methods is the Mean average precision (MAP) score which is a typical measure of the performance of information retrieval systems. The above metric is defined as the average of the precision value obtained after each relevant word is retrieved. The formula of MAP score is defined in Eq. (37).

\[
MAP = \frac{\sum_{k=1}^{n} (P@k \times \text{rel}(k))}{|\{\text{relevant words}\}|}.
\]

\(\text{rel}(k)\), in Eq. (37), is a function that is defined in Eq. (38).
4.3 Comparison with State-of-the-Art Methods

In this section, we compare performances of the proposed method on the four datasets described earlier with performances of the state-of-the-art KWS methods [4, 15, 25–27, 30, 37, 41, 45, 56]. Methods used here for comparison can be grouped as (a) strictly learning-free methods where not a single annotated training sample is used (e.g., [15, 26, 30, 37, 45, 56]), (b) tunable-based models where a few annotated samples are used to optimize the model parameters (e.g., [4, 27, 41, 42]), and (c) strictly learning-based models where a large scale annotated training samples are used for learning model hyperparameters (e.g., [25]). The present work belongs to the first-category of work and is directly compared with strictly learning-free methods.

The comparative results of the proposed method along with the state-of-the-art KWS models are shown in Tables 2 and 3. In Table 2, we compare the methods in terms of MAP score, whereas in Table 3, methods are compared in terms of P@5 score. From these results, it is observed that the present method outperforms all the strictly learning-free methods on IAM and ICFHR 2016 Botany datasets, whereas ranks second based on MAP score among strictly learning-free methods for the other two datasets. Here, we would like to mention that Yousfi et al. [15] reported the performance using only 6 query words, and therefore, it is not a kind of fair comparison since the proposed method is evaluated using 300 query words for the ICFHR 2014 H-KWS competition Modern dataset. However, in the case of the ICFHR 2016 H-KWS competition Konzilsprotokolle dataset, the present method becomes second to the method proposed by Rothacker et al. [26] because our preprocessing techniques performs a bit poorly while generating noise-free binarized word images in many cases. If we compare the performance of the proposed method, irrespective of the model type, then it can be noted that our method is comparable with state-of-the-art methods.

More specifically, from the comparison shown in Tables 2 and 3, the following observations are made.

- The performance on the IAM dataset is quite promising as the proposed method can achieve a gain in performance by MAP score evaluation. The method proposed by Wolf and Fink [25] provides a better MAP value, however, the difference is marginal and this method is strictly learning-based.
- The ICFHR 2014 H-KWS Modern competition dataset is also tested using the proposed method along with the ones considered for comparison. Unlike the IAM dataset here the proposed algorithm fails to surpass the results of tunable-/strictly learning-based methods, however, provides satisfactory results when compared with learning-free methods in terms of both P@5 and MAP evaluation scores. MISM and m-POG feature descriptor-based proposals by Retsinas et al. [4] are marginally ahead of the current method in terms of P@5 metrics. However, almost all the learning-based methods include a parameter tuning/training step.
- In the case of the ICFHR 2016 H-KWS competition Botany dataset, the proposed method outperforms most of the methods in terms of MAP and P@5 scores.
ever, KWS methods based on the graph-based distance calculation method by Stauffer et al. [41] and Wolf and Fink [25] provided better results as compared to the proposed method. Also, the proposed learning-free method is not at par with the state-of-the-art learning-based methods.

- Like the previous dataset, the proposed method fails to compete with the learning-based methods in ICFHR 2016 H-KWS competition Konzilsprotokolle dataset. The results are very much similar to the ICFHR 2016 Botany dataset, where graph-based methods seem to be more efficient than the proposed method.

We also show some examples of retrieved words as search words in Fig. 13.

---

| Query word | Top-5 retrieved words from the target word images |
|------------|--------------------------------------------------|
| Nachruhm   | Nachruhm Nachruhm Nachruhm Nachruhm Nachruhm |
| anopptou    | anopptou anopptou anopptou anopptou anopptou |
| nichts      | nichts nichts nichts nichts nichts           |

Fig. 13 Top-5 search results for three queries from a ICFHR 2014 H-KWS competition Modern dataset, b IAM dataset c ICFHR 2016 H-KWS competition Botany dataset, and d ICFHR 2016 H-KWS competition Konzilsprotokolle dataset
4.4 Error Case Analysis

Even though the proposed method outperforms most of the learning-free KWS techniques, there are cases when it does not produce a satisfactory result. The reasons behind such failures have been discussed below.

- Due to the variations in handwriting, different words are observed to take similar profile (both upper and lower) shapes and consequently these different target words appear in the search results with a better score whereas similar words are pushed back in the search results. Some instances of error cases are shown in Fig. 14.

Fig. 14 Examples of top-5 retrieved target word images for given query word images. The word images marked within the red-colored bounding boxes indicates erroneous retrieval. The first, second, third, and fourth rows show the retrieval results on ICFHR 2014 H-KWS competition Modern, IAM, ICFHR 2016 H-KWS competition Botany and ICFHR 2016 H-KWS competition Konzilsprotokolle datasets, respectively.

Fig. 15 The green marked portions are results of faulty word segmentation using the ground truth information of the ICFHR 2014 H-KWS competition Modern dataset.

Table 4 Some choices of the arguments $\emptyset_1$ and $\emptyset_2$, and the corresponding effects on the final spotting result.

| Arguments | Performance on Dataset |
|-----------|------------------------|
|          | MD14 | IAM | BO16 | KZ16 |
| $(\emptyset_1, \emptyset_2)$ | P@5 | MAP | P@5 | MAP | P@5 | MAP | P@5 | MAP |
| (20,70) | 0.54 | 0.45 | 0.24 | 0.33 | 0.40 | 0.43 | 0.45 | 0.50 |
| (63,23) | 0.38 | 0.20 | 0.10 | 0.47 | 0.55 | 0.22 | 0.57 | 0.45 |
| (57,23) | 0.21 | 0.13 | 0.10 | 0.56 | 0.57 | 0.50 | 0.59 | 0.56 |
| (16,74) | 0.56 | 0.42 | 0.20 | 0.49 | 0.60 | 0.55 | 0.60 | 0.52 |
| (70,20) | 0.56 | 0.42 | 0.22 | 0.53 | 0.63 | 0.61 | 0.65 | 0.67 |
| (30,60) | 0.59 | 0.65 | 0.51 | 0.62 | 0.68 | 0.65 | 0.66 | 0.71 |

Here MD14, BT16, and KZ16 represent ICFHR 2014 Modern, ICFHR 2016 Botany, and ICFHR 2016 Konzilsprotokolle datasets, respectively.
• ICFHR 2016 H-KWS competition Botany and ICFHR 2016 H-KWS competition Konzilsprotokolle datasets are noisy and require a competent binarization technique for pre-processing the query and the target word images as mentioned in Sect. 3.1. Due to improper noise removal and binarization, the retrieved words for some query words of the aforesaid datasets are incorrect as shown in Fig. 14.

• Segmentation error using ground truth information of the ICFHR 2014 H-KWS competition Modern dataset gives incorrect results. Some examples of segmentation errors are shown in Fig. 15.

The choice of arguments $\emptyset_1$ and $\emptyset_2$ (described in Sect. 3.4.4) is very crucial. It is observed that different values of these arguments provide different results. Outcomes using some variation in $\emptyset_1$ and $\emptyset_2$ are recorded in Table 4.

4.5 Shortcomings of the Proposed Work

The performance of the proposed work is comparable with state-of-the-art techniques. However, the present work has some limitations that are listed below.

• Use of Ostu’s binarization technique fails to generate quality binarized images when the input is from historical manuscripts.

• Relying only on profiles of the dense region of a word image restricts its generalization capability.

• The proper choice of arguments $\emptyset_1$ and $\emptyset_2$ is very crucial to return expected solutions.

5 Conclusion

In this work, we present a profile matching-based learning-free KWS technique that can be applied to a heterogeneous collection of handwritten documents. To do this, first, we extract the upper and lower profiles from the binarized version of the query and the target words. These profiles are not only used to find a matching score in the Z-transform domain between query and target words but also help in the selection of probable candidate query words by diminishing a substantial number of words from the actual target word set. In the matching stage, before projecting the profiles in the Z-transform domain, we apply the affine transformation on the Bezier curve representation of the profiles to deal with the variations like rotation, shear, and scale that might occur due to the individual’s writing style. The similarity score is calculated using the condition of resonance for the damped oscillator. The proposed method achieves satisfactory performance compared to state-of-the-art learning-free KWS methods for ICFHR 2014 H-KWS competition, and IAM datasets.

Although the performance of the proposed method is at par with many state-of-the-art methods, yet there is still some room for improvement. The performance of the present work is largely affected by failure at binarization and noise removal levels. Hence, the use of some state-of-the-art binarization and noise removal methods could improve the performance. Also, fusing some state-of-the-art texture-based features like mPOG, oBIFs, and DoLFs with profile information might help in designing a better KWS system. Certain transformations performed on images take significant time resulting in a limiting factor of this process which would be taken care of in the future. Finally, concerning resource optimization, we may aim to further compress the extracted features to make the algorithm memory efficient.

Acknowledgements We would like to thank the Center for Microprocessor Applications for Training Education and Research (CMATER) research laboratory of the Computer Science and Engineering Department, Jadavpur University, India, for providing us the infrastructural support.

Author Contributions All authors, D.B., P.B., S.M., E.C., M.P. and R.S., discussed the results and contributed all equally to the final manuscript that is submitted to consideration.

Funding Thereby, the authors confirm that there are no organizations that may gain or lose financially through the publication of this manuscript.

Availability of Data and Material The datasets employed by the current study are publicly available standard datasets, namely ICFHR 2014 H-KWS competition Modern and IAM datasets. However, should any other information related to the work presented in this manuscript be required, they are available from the corresponding author on reasonable request.

Declarations

Conflict of interest None of the authors of this paper has a financial or personal relationship with other people or organizations that could inappropriately influence or bias the content of the paper. Therefore, it is to expressly state that “No competing interests are at stake” and there is “No conflict of interest” with other people or organizations that could inappropriately influence or bias the content of the paper. The authors declare that there is no conflict of interest.

Ethics Approval and Consent to Participate The authors confirm that this manuscript has not been submitted to any other journal for simultaneous consideration. This is an original work that has not been published elsewhere in any form or language.

Consent for Publication All authors confirm that they all agreed with the content and gave explicit consent to submit the manuscript for consideration.
Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Malakar, S., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M.: An image database of handwritten Bangla words with automatic benchmarking facilities for character segmentation algorithms. Neural Comput. Appl. 33, 449–468 (2021). https://doi.org/10.1007/s00521-020-04981-w
2. Bhattacharya, R., Malakar, S., Schwenker, F., Sarkar, R.: Fuzzy-Based Pseudo Segmentation Approach for Handwritten Word Recognition Using a Sequence to Sequence Model with Attention. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part II. pp. 582–596. Springer International Publishing (2021)
3. Krishnan, P., Dutta, K., Jawahar, C. V.: Deep feature embedding for accurate recognition and retrieval of handwritten text. In: International Conference on Frontiers in Handwriting Recognition, ICFHR. pp. 289–294 (2016)
4. Retsinas, G., Louloudis, G., Stamatopoulos, N., Gatos, B.: Efficient Learning-Free Keyword Spotting. IEEE Trans. Pattern Anal. Mach. Intell. 41, 1587–1600 (2019)
5. Malakar, S., Sharma, P., Singh, P.K., Das, M., Sarkar, R., Nasipuri, M.: A holistic approach for handwritten Hindi word recognition. Int. J. Comput. Vis. Image Process. 7, 59–78 (2017). https://doi.org/10.4018/ICVIP.2017010104
6. Malakar, S., Paul, S., Kundu, S., Bhowmik, S., Sarkar, R., Nasipuri, M.: Handwritten word recognition using lottery ticket hypothesis based pruned CNN model: a new benchmark on CMATERdb. 2. 1.2. Neural Comput. Appl. 32, 15209–15220 (2020)
7. 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. 29, 719–735 (2020). https://doi.org/10.1515/jisyys-2017-0384
8. Giotis, A.P., Stikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognit. 68, 310–332 (2017). https://doi.org/10.1016/j.patcog.2017.02.023
9. Majumder, S., Ghosh, S., Malakar, S., Sarkar, R., Nasipuri, M.: A voting-based technique for word spotting in handwritten document images. Multimed. Tools Appl. (2021). https://doi.org/10.1007/s11042-020-10363-0
10. Rothacker, L., Fink, G.A.: Segmentation-free query-by-string word spotting with Bag-of-Features HMMs. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. pp. 661–665. IEEE (2015)
11. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Segmentation-free word spotting with exemplar SVMs. Pattern Recognit. 47, 3967–3978 (2014). https://doi.org/10.1016/j.patcog.2014.06.005
12. Ghosh, S., Valveny, E.: Text box proposals for handwritten word spotting from documents. Int. J. Doc. Anal. Recognit. 21, 91–108 (2018). https://doi.org/10.1007/s10032-018-0300-7
13. Rothacker, L., Sudholt, S., Rusakou, E., Kasperidus, M., Fink, G.A.: Word Hypotheses for Segmentation-Free Word Spotting in Historic Document Images. In: International Conference on Document Analysis and Recognition, ICDAR. pp. 1174–1179 (2017)
14. Kovalchuk, A., Wolf, L., Dershowitz, N.: A simple and fast word spotting method. In: 2014 14th International Conference on Frontiers in Handwriting Recognition. pp. 3–8. IEEE (2014)
15. Yousfi, D., Gattal, A., Djeddi, C., Siddiqi, I., Bensefta, A.: Keyword Spotting in Modern Handwritten Documents Using eBIFs. In: Mediterranean Conference on Pattern Recognition and Artificial Intelligence. pp. 240–250. Springer (2022)
16. Singh, P.K., Mahanta, S., Malakar, S., Sarkar, R., Nasipuri, M.: Development of a page segmentation technique for Bangla documents printed in italic style. In: Proceedings of the 2nd International Conference on Business and Information Management (ICBIM 2014) (2014)
17. Sarkar, R., Malakar, S., Das, N., Basu, S., Kundu, M., Nasipuri, M.: Word extraction and character segmentation from text lines of unconstrained handwritten Bangla document images. J. Intell. Syst. 20, 227–260 (2011). https://doi.org/10.1515/JISYS.2011.013
18. Sarkar, R., Halder, S., Malakar, S., Das, N., Basu, S., Nasipuri, M.: Text line extraction from handwritten document pages based on line contour estimation. In: 2012 3rd International Conference on Computing, Communication and Networking Technologies, ICCCNT 2012. pp. 1–8. IEEE (2012)
19. Malakar, S., Ghosh, P., Sarkar, R., Das, N., Basu, S., Nasipuri, M.: An improved offline handwritten character segmentation algorithm for Bangla script. In: Proceedings of the 5th Indian International Conference on Artificial Intelligence (IICAI) (2011)
20. Rath, T.M., Mannmatha, R.: Word image matching using dynamic time warping. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2003)
21. Kundu, S., Malakar, S., Geem, Z.W., Moon, Y.Y., Singh, P.K., Sarkar, R.: Hough transform-based angular features for learning-free handwritten keyword spotting. Sensors. 21, 4648 (2021)
22. Krishnan, P., Jawahar, C. V.: HWNet v2: an efficient word image representation for handwritten documents. Int. J. Doc. Anal. Recognit. 22, 387–405 (2019). https://doi.org/10.1007/s10032-019-00336-x
23. Sudholt, S., Fink, G.A.: Attribute CNNs for word spotting in handwritten documents. Int. J. Doc. Anal. Recognit. 17, 199–218 (2018). https://doi.org/10.1007/s10032-018-0295-0
24. Sudholt, S., Fink, G.A.: PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents. In: International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 277–282. IEEE (2016)
25. Wolf, F., Fink, G.A.: Annotation-free learning of deep representations for word spotting using synthetic data and self labeling. In: International Workshop on Document Analysis Systems. pp. 293–308. Springer (2020)
26. Rothacker, L., Wolf, F., Fink, G.A.: Annotation-free word spotting with bag-of-features HMMs. Int. J. Pattern Recognit. Artif. Intell. 35, 2153001 (2021)
27. Stauffer, M., Fischer, A., Riesen, K.: Filters for graph-based keyword spotting in historical handwritten documents. Pattern Recognit. Lett. 134, 125–134 (2020)
28. Sudholt, S., Fink, G.A.: Evaluating Word String Embeddings and Loss Functions for CNN-Based Word Spotting. In: International Conference on Document Analysis and Recognition, ICDAR. pp. 493–498. IEEE (2017)
29. Meshesha, M., Jawahar, C. V.: Matching word images for content-based retrieval from printed document images. Int. J. Doc.
30. Howe, N.R.: Part-structured inkball models for one-shot handwritten word spotting. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR (2013)

31. Sullivan, D.M.: Z-transform theory and the FDTD method. IEEE Trans. Antennas Propag. 44, 28–34 (1996)

32. Balashkov, M.V., Bogachev, V.M.: Z-transform exponential approximation of one-dimensional functions: theory and applications. Radioelectro. Commun. Syst. 64, 189–203 (2021)

33. Sundararajan, D.: The z-Transform. In: Digital Signal Processing, pp. 155–188. Springer (2021)

34. Bera, S.K., Kar, R., Saha, S., Chakrabarty, A., Lahiri, S., Malakar, S., Sarkar, R.: A one-pass approach for slope and slant estimation of tri-script handwritten words. J. Intell. Syst. 29, 688–702 (2018). https://doi.org/10.1515/jissys-2018-0105

35. Rodríguez-Serrano, J.A., Perronnin, F.: Local gradient histograms features for word spotting in unconstrained handwritten documents. In: International Conference on Frontiers in Handwriting Recognition. pp. 7–12 (2008)

36. Terasawa, K., Tanaka, Y.: Slit style HOG feature for document image word spotting. In: 2009 10th International Conference on Document Analysis and Recognition. pp. 116–120. IEEE (2009)

37. Retsinas, G., Louloudis, G., Stamatopoulos, N., Gatos, B.: Keyword spotting in handwritten documents using projections of oriented gradients. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS). pp. 411–416. IEEE (2016)

38. Bhunia, A.K., Roy, P.P., Sain, A., Pal, U.: Zone-based keyword spotting in Bangla and Devanagari documents. Multimed. Tools Appl. 79, 27365–27389 (2020)

39. Zagoris, K., Amanatiadis, A., Pratikakis, I.: Word Spotting as a Service: An Unsupervised and Segmentation-Free Framework for Handwritten Documents. J. Imaging. 7, 278 (2021)

40. Aldavert, D., Rusiñol, M., Toledo, R., Lladós, J.: A study of bag-of-visual-words representations for handwritten keyword spotting. Int. J. Doc. Anal. Recognit. 18, 223–234 (2015)

41. Stauffer, M., Fischer, A., Riesen, K.: Keyword spotting in historical handwritten documents based on graph matching. Pattern Recognit. 81, 240–253 (2018)

42. Ameri, M.R., Stauffer, M., Riesen, K., Bui, T.D., Fischer, A.: Graph-based keyword spotting in historical manuscripts using Hausdorff edit distance. Pattern Recognit. Lett. 121, 61–67 (2019)

43. Wilkinson, T., Brun, A.: Semantic and verbatim word spotting using deep neural networks. In: International Conference on Frontiers in Handwriting Recognition, ICFHR (2016)

44. Boudraa, O., Michelucci, D., Hidouci, W.K.: PUNet: Novel and efficient deep neural network architecture for handwritten documents word spotting. Pattern Recognit. Lett. 155, 19–26 (2022)

45. Sfikas, G., Retsinas, G., Gatos, B.: Zoning aggregated hypercolumns for keyword spotting. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 283–288. IEEE (2016)

46. Cheikhrouhou, A., Kessentini, Y., Kanoun, S.: Hybrid HMM/BLSTM system for multi-script keyword spotting in printed and handwritten documents with identification stage. Neural Comput. Appl. 32, 9201–9215 (2020)

47. Daraeef, F., Moazzafari, S., Razavi, S.M.: Handwritten keyword spotting using deep neural networks and certainty prediction. Comput. Electr. Eng. 92, 107111 (2021)

48. Kumari, L., Sharma, A.: A Review of Deep Learning Techniques in Document Image Word Spotting. Arch. Comput. Methods Eng. 1–22 (2021)

49. Bhattacharyya, A., Roy, S., Sarkar, N., Malakar, S., Sarkar, R.: Circuit component detection in offline hand-drawn electrical/electronic circuit diagram. In: IEEE Calcutta Conference (CALCON 2020). IEEE, Kolkata (2020)

50. Banerjee, D., Bhowal, P., Bera, S.K., Sarkar, R.: Rectification of Camera Captured Document Images using Component Analysis. In: 2020 IEEE Calcutta Conference (CALCON). pp. 421–425. IEEE (2020)

51. Mondal, T., Ragot, N., Ramel, J., Yves, Pal, U.: Comparative study of conventional time series matching techniques for word spotting. Pattern Recognit. 73, 47–64 (2018). https://doi.org/10.1016/j.patcog.2017.07.011

52. Cusp (singularity), https://en.wikipedia.org/wiki/Cusp_(singularity)

53. Woźny, P., Lewanowicz, S.: Multi-degree reduction of Bézier curves with constraints, using dual Bernstein basis polynomials. Comput. Aided Geom. Des. 26, 566–579 (2009)

54. Pratikakis, I., Zagoris, K., Gatos, B., Louloudis, G., Stamatopoulos, N.: ICFHR 2014 competition on handwritten keyword spotting (H-KWS 2014). In: 2014 14th International Conference on Frontiers in Handwriting Recognition. pp. 814–819. IEEE (2014)

55. Marti, U.V., Bunke, H.: The IAM-database: an English sentence database for offline handwriting recognition. Int. J. Doc. Anal. Recognit. 5, 39–46 (2002)

56. Pratikakis, I., Zagoris, K., Gatos, B., Puigcerver, J., Toselli, A.H., Vidal, E.: ICFHR2016 handwritten keyword spotting competition (H-KWS 2016). In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 613–618. IEEE (2016)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.