# Word Spotting in cluttered environment

Divya Srivastava and Gaurav Harit

Indian Institute of Technology Jodhpur
{srivastava.5,gharit}@iitj.ac.in

**Abstract.** In this paper, we present a novel problem of handwritten word spotting in cluttered environment where a word is cluttered by a strike-through with a line stroke. These line strokes can be straight, slant, broken, continuous or wavy in nature. Vertical Projection Profile (VPP) feature and its modified version, which is the combinatorics Vertical Projection Profile (cVPP) feature is extracted and aligned by modified Dynamic Time Warping (DTW) algorithm. The dataset for the proposed problem is not available so we prepared our dataset. We compare our method with Rath and Manmath [6], and PHOCNET [17] for handwritten word spotting in the presence of strike-through, and achieve better results.

**Keywords:** Word Spotting · Dynamic Time Warping · Vertical Projection Profile · Combinatorics Vertical Projection Profile

## 1 Introduction

Digitization of a large number of documents raises the issue of indexing and retrieval in an efficient manner. For this reason, research has been emphasized on word spotting which refers to localization of word images of interest in the dataset without actually interpreting its content [3]. Some of the issues with handwritten word spotting in documents images are handwriting variability, ink bleed, language variability, degradation caused by aging, strains, repetitive use, etc. We are here considering a specific case where a handwritten word has been distorted accidentally or intentionally by a strike-through over it. The line used for striking can be straight, slant, broken, continuous or wavy. Especially in some scripts like, Devanagari, Sanskrit, Punjabi, Bengali, etc., the shirorekha can mistakenly be passed through the word leading to a clutter. In structured document images like form, cheques, etc., handwritten words are often overlapping with cell boundaries. To resolve this issue, we propose a word spotting technique that works in a cluttered medium where a word is cluttered by a strike-through over it. Vertical Projection Profile (VPP) feature is considered as a feature vector for uncluttered query word and a new feature called combinatorics Vertical Projection Profile (cVPP) feature is used for the cluttered candidate words. CVPP selects or discards the various combinatorics of possible strokes that contribute to VPP along each of its column. The similarity measure between the words is computed using a Dynamic Time Warping (DTW) approach which has been extensively used in word spotting. To the best of our knowledge word spotting for such handwritten words cluttered by strike is a novel problem which has not been addressed so far.

In the following section, related work for word spotting and DTW is discussed. Section 3 explains the VPP and its combinatorial version cVPP used as a feature descriptor. Section 4 explores the working and various constraints associated with DTW in our method. Section 5 discusses the matching procedure for the proposed word descriptor. Results and discussion are presented in section 6 followed by conclusion.

## 2 Related Work

### 2.1 Word Spotting

Analysis of off-line handwritten documents faces various challenges such as, inter and intra writer variability, ink bleed, language variability, document degradation, etc. Word spotting in off-line handwritten documents is a difficult task. The problem of touching and broken characters due to degradation is addressed in [12], where word spotting is done by decomposing text line into character primitives. A word image is encoded into a string of coarse and fine primitives chosen according to the codebook and then a string matching approach based on dynamic programming is used to retrieve a similar feature sequence in a text line from the collection.

HMM-based methods have been used extensively for this purpose. Lexicon-free keyword spotting using the filler or garbage Hidden Markov Models (HMM) has the issue of high computational cost of the keyword-specific HMM Viterbi decoding process required to obtain confidence score of each word. To deal with this issue [9] has proposed a novel way to compute this confidence score, directly from character lattice which is produced during a single Viterbi decoding process using only the "filler" model, thus saving the time consuming keyword-specific decoding step. Later they extended their work in [4] by using context-aware character lattices obtained by Viterbi decoding with high-order character N-gram models. In another approach by [13] lexicon-free keyword spotting system is presented which employs trained character HMMs. The statistical framework for word-spotting is proposed in [8] where HMMs model keywords and Gaussian Mixture Model (GMM) does score normalization.

Various approaches represent the handwritten word in graphical representation by their different notions and employ the various graph matching algorithms for spotting purpose. In [2] a handwritten word is represented as a graph such that graphemes are extracted from shape convexity that are used as stable units of handwriting. These graphemes are associated with graph nodes and spatial relationships between them are considered as graph edges. Spotting is done using bipartite-graph matching as error tolerant graph matching. In another approach using graph-based method [5], invariants which are the collection of writing pieces automatically extracted from the old document collection, are used as a descriptor to characterize the word and the graph edit distance is used as dissimilarity between the word images.

Various fusion frameworks where more than one methods are used to solve word spotting with higher accuracy are have been proposed. To deal with the issue of large variation due to unconstrained writing style, a method based on Heat Kernel Signature (HKS) and triangular mesh structure are proposed in [20] for keyword spotting. HKS captures local features while triangular mesh structure is used to represent global characteristics. Notably, this method does not require pre-processing. In [10] both topological and morphological features are employed for word spotting. Skeleton-based graphs with shape context labeled vertices are constructed for connected components and each word is represented as a sequence of graphs. For fast retrieval, region of interest is found using graph embedding. For refined result graph edit distance based on DTW approach is used.

In [5] word spotting problem is addressed using a shape-based matching scheme where segmented word images are represented by local contour features. A segmentation free approach is given in [18] SIFT features are extracted and encoded into visual words which are accumulated in a set of local patches. These features are projected to a topic space using latent semantic analysis. Further compressing the feature with product quantization method leads to efficient indexing of document information both in terms of memory and time. Another segmentation free approach is proposed in [19] where again the spotting is done in two steps of selecting candidate zone followed by refining the results. Both the steps are based on the process of accumulation of votes obtained by application of generalized Haar-like features. Deep learning based approach for word spotting PHOCNET is proposed in[17]. CNN is used to extract PHOC features for word images which are then compared using Bray-Curtis dissimilarity [16].

### 2.2 Dynamic Time Warping

To overcome the shortcomings of Euclidean metric in measuring similarity measure between sequences with variation in stretches and compression, DTW was introduced as similarity metric by Berndt [7]. DTW works on the principle of Dynamic Programming to find the minimal distance between two sequences which are warped by stretching or shrinking in their subsection. DTW similarity measure finds its application whenever the feature vector is in a sequential format. It has been therefore widely used in speech processing, bioinformatics, online handwriting recognition, etc. DTW is used in offline handwriting recognition where the sequential features like projection profile are considered as a feature vector. In [14], DTW was used for handwritten word spotting for robustness against intra-writer variability. DTW was used for word spotting in noisy historical documents in [11] using projection profile, word profiles, and background/ink transitions as features. By comparing DTW with 6 matching techniques it was found that DTW performs better. DTW has quadratic complexity and therefore is computationally expensive for large-scale images. It's faster version was proposed in [21] which provides 40 times speed up. DTW based statistical framework is proposed in [8] for a challenging multi-writer corpus.

## 3 Features

Word spotting using DTW approach is carried out using sequential features and one among them is projection profile feature. Projection profile corresponds to the histogram of the number of black pixels accumulated along the horizontal and vertical axis of the image. Depending upon the axis considered they are termed as Horizontal Projection Profile (HPP) or Vertical Projection Profile (VPP) for the accumulation of black pixels along horizontal and vertical axis, respectively.

The proposed feature is a slight variation of VPP feature and uses combinatorics in VPP. Candidate word in our method is a set of cluttered and uncluttered word present in our dataset. As shown in Figure 1 the cluttered candidate word has a strike-through over it and the query word is an uncluttered word without any strike-through. VPP in the query word is computed by counting all the black pixels across each column in a black and white image with foreground black pixels and background white pixels. Several strokes can intersect with a column of pixels (i.e., the vertical axis). Here, a stroke refers to a run of black pixels along a column in an image. For a candidate word cluttered by a single strike-through line, we have an extra stroke which corresponds to the pixels along the column contributed by the strike-through. We consider summation of black pixels for various combinations of strokes which accounts for the profile feature of the candidate image.

Consider an image $I$ with $M$ rows and $N$ columns, and let $I(x, y)$denote a pixel in the image such that $1 \leq x \leq M$ and $1 \leq y \leq N$. VPP value for the uncluttered query word at $y^{\text{th}}$ column can be computed as shown in equation 1, where $y$ indexes the column and $i$ indexes the row.

$$VPP(y) = \sum_{i=1}^{M} I(i, y) \tag{1}$$

A candidate word image can be cluttered with strike-through and therefore we compute the combinatorics VPP (cVPP) feature. Consider $n$ as the total number of strokes in a column $y$ and let $i$ be the row index. cVPP is a vector (indexed by $k$) computed for every column as follows:

$$cVPP(k, y) = \sum_{s_m \in S_k} \sum_{i=s_m^{x1}}^{s_m^{x2}} I(i, y) \tag{2}$$

Here, $S_k$ is one of the subsets of the set of strokes that cut across column $y$. $s_m$ is a member of the subset $S_k$. The $x$-limits (coordinates of the vertical black run) of the stroke $s_m$ are $s_m^{x_1}$ and $s_m^{x_2}$. Index $k$ varies over all possible subsets of strokes, including the null set. cVPP leads to a feature vector where for each column of strokes we have multiple entries depicting black pixel counts for various possible combinations of strokes. In further section by using modified DTW, selection will be done for each column of cVPP to find the best combination of strokes which matches with the query image.

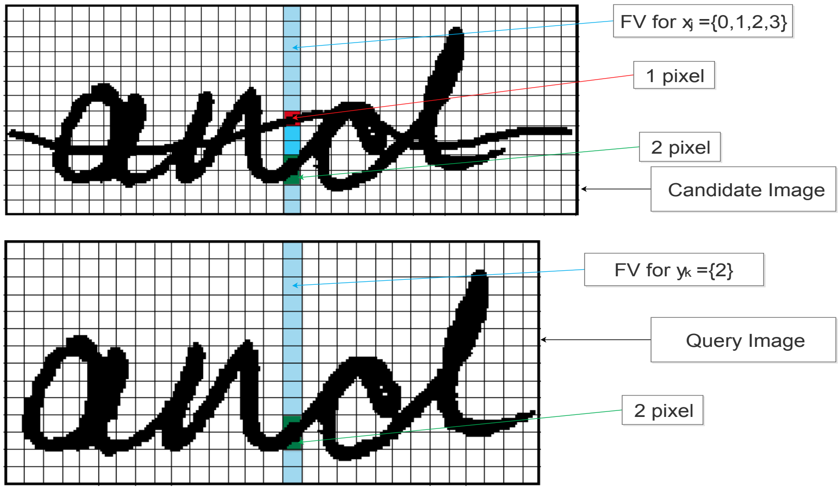


**Fig. 1.** Computation of feature vector for candidate and query image

## 4 Feature matching using DTW

Let X denote the feature sequence (VPP) for the query image and Y denote the feature sequence (cVPP) for the candidate image. The feature sequences X and Y are defined as $X = x_1, x_2, ...x_{M_1}$ of length $M_1$ and $Y = y_1, y_2, ...y_{M_2}$ of length $M_2$. As described in previous section, each entry in the cVPP feature vector denotes a vector whose length depends on the number of strokes in the corresponding column of the image. The alignment from X to Y is described in terms of a warp path $W = w_1, w_2, ...w_K$, where K is the warp path length whose range is $max\ (M_1, M_2) \leq k \leq M_1 + M_2$, where $w_k = (i_k, j_k)$ indicates the indices of feature sequences X and Y at $k^{\text{th}}$ instance of the matching. To find this alignment in terms of warp path consider an $(M + 1) \times (N + 1)$ cost matrix for the sequences X and Y. While doing the optimal alignment, it is assumed that the starting and ending indices of the two sequences align with each other and hence, we need to compute the path from lower left to top right corner of the cost matrix. Computing all possible paths between the two extremes of the cost matrix and finding out the minimum cost path among them is computationally very expensive. To overcome this, DTW distance measure which is based on dynamic programming makes use of recursive nature of distance function. Hence, the entries of the cost matrix need to be filled by distance function which is essentially based on dynamic programming approach.

The mathematical expression for DTW distance between the feature vector X and Y is given in equation 3:

$$D^{l,m}(X_i, Y_j) = d^{l,m}(X_i, Y_j) + \underset{\forall a,b}{min} \begin{cases} D^{a,b}(X_{i-1}, Y_j) \\ D^{a,b}(X_i, Y_{j-1}) \\ D^{a,b}(X_{i-1}, Y_{j-1}) \end{cases} \tag{3}$$

Where $d^{l,m}(X_i, Y_j)$ is the local cost of a cell at index $(i, j)$ by selecting the $l^{th}$ and $m^{th}$ combination for candidate and query image, respectively. Local cost is responsible for the selection of stroke combination in candidate feature vector such that it matches closely to the corresponding candidate feature vector. For this purpose, Manhattan distance is computed between VPP for a column in query image and various cVPP for a column in candidate image. Best matched stroke is selected and cVPP leading to minimum distance is selected and considered as the local distance for corresponding entry of the cost matrix for DTW algorithm.

Each entry in a cell of the cost matrix is computed by addition of local cost and minimum cost among the previously traversed adjacent cell. $D^{a,b}(X_i, Y_j)$ signifies the total distance for a cell at index $(i, j)$ of the cost matrix by selecting the $a^{th}$ and $b^{th}$ combination for candidate and query image respectively. $D^{l,m}(X_i, Y_j)$ has the same interpretation as $D^{a,b}(X_i, Y_j)$, difference lies in that $l$ and $m$ stands for the combination referring to the local cost of current cell while $a$ and $b$ stands for the combination referring to the total cost of the pre-traversed adjacent cell.

The total cost computed by the DTW approach is the summation of the cost incurred during the whole warp path which needs to be normalized to make it invariant to the variable length sequences. Therefore the DTW cost corresponding to the optimal warp path is given in equation 4, where $K$ is a normalizing factor equal to the optimal warp path length and $\sum_{k=1}^{K} D(X_k, Y_k)$ corresponds to the summation of all the entries in the DTW cost matrix along the optimal warp path.

$$D(X, Y) = \frac{1}{K} \sum_{k=1}^{K} D(X_k, Y_k) \tag{4}$$

The optimal warp path is identified by backtracking the cost matrix D along the minimum cost index pairs $(i_k, j_k)$ starting from $(M + 1, N + 1)$ to $(0, 0)$ in $O(MN)$ time.

Some constraints followed by the warp path [15] are mentioned below:

- Boundary Constraint: The starting point $(0, 0)$ and the ending point $(M + 1, N + 1)$ of the warping path are fixed, i.e., $w_1 = (0, 0)$ and $w_k = (M + 1, N + 1)$.

- Continuity Constraint: It ensures that the path is continuous and no break is permitted i.e., $i_k - i_{k-1} = 1$.
- Monotonicity Constraint: The sequence of characters in a word is constant. It requires the characters in a word to be clustered in monotonically increasing order, which means, $j_k - j_{k-1} = 1$
  Summarizing the continuity and monotonicity constraint. We have, $w_k - w_{k-1} \in (1, 0), (0, 1), (1, 1)$
- Global path Constraint: It is used to restrict the extent of expansion or compression of the features extracted from the word image. The spatial variability is considered to be limited which means that search space can be pruned. To ensure global path constraint, we have considered Sakoe-Chiba Band of window width 20.

## 5 Methodology

We are considering the segmented word as a grayscale image. The word image is binarized using the Otsu's thresholding method [1] which yields two classes of intensity values to have minimal intraclass variance and maximal interclass variance. VPP feature is computed for the query image and cVPP is computed for the candidate images as described in the previous section.

Our aim is to devise a matching procedure which is more likely to account for the correct set of strokes and is able to ignore the strokes arising due to strike-through. For this purpose, as illustrated in Figure 1, the cVPP is computed for the candidate image, where for each column VPP is computed for all the possible combinations of strokes present in that column. For each column, the objective is to find the combination of strokes in the candidate image which gives a projection value closest to that for the corresponding column in the query image. Applying Brute Force method for this combination and selection problem is computationally costly, therefore, we apply modified DTW algorithm as explained in Section 4. DTW cost is normalized two times. First it is normalized by the path length to account for the word length and then it is normalized by the cluttered word length after adding penalty to the DTW cost to make it invariant to the occurrence of blob induced due to clutter.

### 5.1 Computing the Penalty Term

Whenever a word is struck-through, the extra stroke of the line renders some extra pixels termed as blob at the intersecting region in a word due to binarization as shown in Figure 2. The strokes are horizontally oriented, therefore, a word containing more number of characters will have more intersection bleeds leading to more number of blobs getting formed in comparison to a word containing less number of characters. This results in misalignment and some non-diagonal steps in the DTW path.

To resolve this issue, we use penalty term in DTW cost. The blobs produce large projection values in their respective columns. A penalty term is computed on the basis of possible number of occurrences of intersection points present in the candidate word. This helps to minimize the effect of blobs. We find the mean of VPP values and the count of VPP values exceeding their mean are computed as penalty term. The penalty term is subtracted from the DTW cost which is then again normalized by the optimal warp path length $K$.

Given $K$ as the DTW path length and $p$ as the penalty term, the aggregate match cost is given as follows.

$$D(X, Y) = \frac{1}{K}\left[\sum_{k=1}^{K} D(X_k, Y_k) - p\right] \quad (5)$$

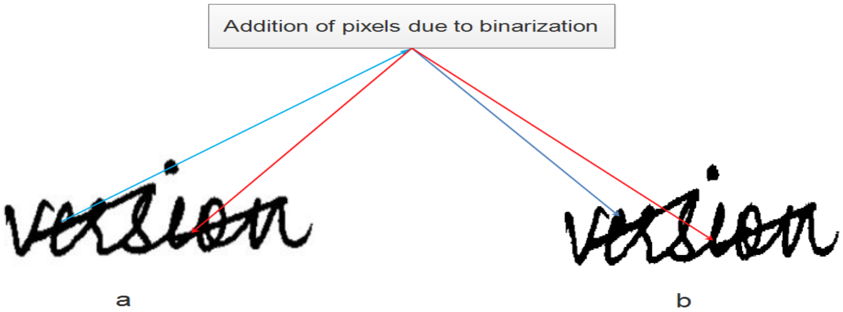


**Fig. 2.** a. Grayscale cluttered image without blob. b. Presence of blob in binarized image

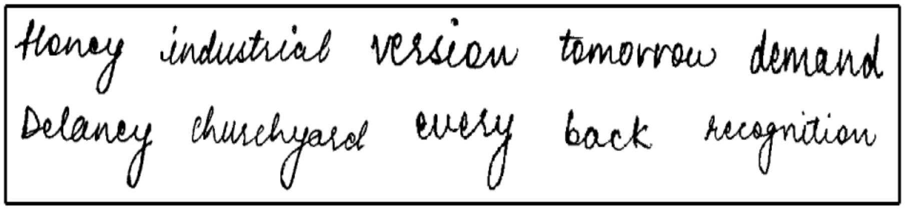
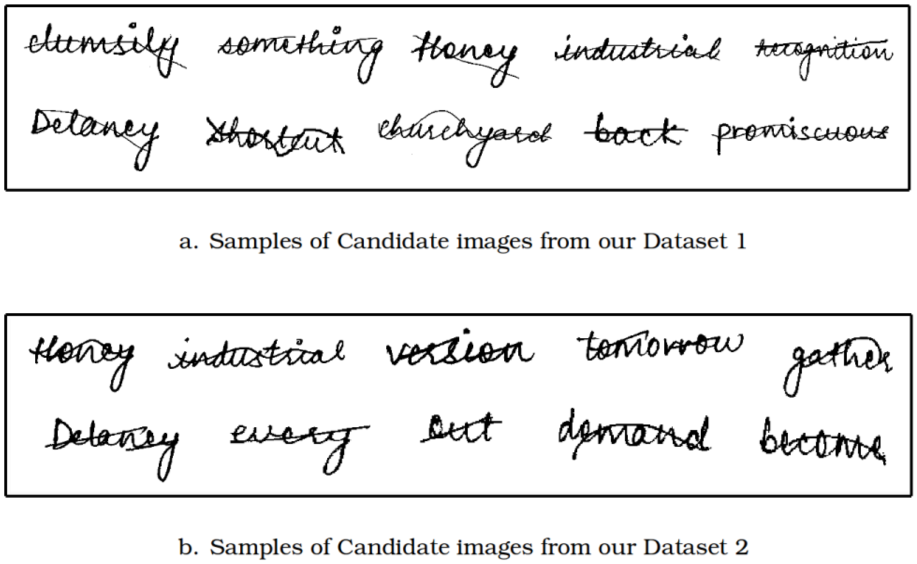

Honey industrial version tomorrow demand
Delaney churchyard every back recognition

**Fig. 3.** Samples for Query images from our dataset

## 6 Dataset

The performance of the proposed approach is evaluated in the retrieval step, where for each of the uncluttered query word, its best possible match among candidate words is found. To test our approach we need a dataset, where some words are rewritten and cluttered by strike-through or line segments. We have developed our own dataset in which word images are written and then the same image is cluttered with the strike-through line. These are type 1 images. Words are also rewritten and cluttered with the line, i.e., strike-through on a different instance of the word. We refer to them as type 2 images.

We collected 50 query word images and 100 candidate word images of type 1 (referred to as dataset 1), and 100 candidate word images of type 2 (referred to as dataset 2). Candidate word images contain 50 cluttered and 50 uncluttered word images. The line drawn to induce the clutter is of free form, it can be straight, slant, waveform, broken, etc. In the dataset, we have considered clutter by a single line so that along with a column there is only one extra stroke. Sample images from our dataset are shown in Figure 3 and 4.



a. Samples of Candidate images from our Dataset 1

b. Samples of Candidate images from our Dataset 2

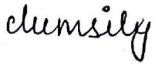
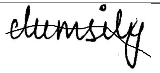
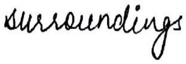
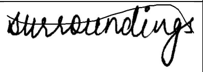
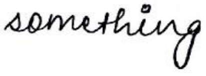
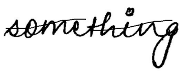
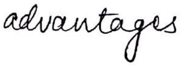
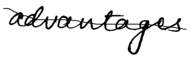
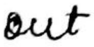
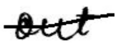
**Fig. 4.** Samples for Candidate images from our Dataset

## 7 Experiments

Experiments are conducted on our dataset, where for each word in the query image set, its best match in the dataset is retrieved based on the dissimilarity measure computed as the matching cost obtained from DTW algorithm. To verify our approach, results are compared to the word spotting results obtained from the DTW approach given by Rath et.al. in [6]. Also, we compare our method with PHOCNET [17]. It is a pre-trained model on George Washington dataset[2]. We use this model and test it on our images.

There are three sets of experiments conducted. Firstly, DTW cost is computed using [6] and our approach on word images from dataset 1. Secondly, DTW cost is computed using [6] and our approach on word images from dataset 2. In the third set of experiments, precision at rank 1 is computed for [6], PHOCNET [17], our approach for both the types of our dataset.

**Table 1.** Matching cost for 5 examples from Dataset 1

| Query image | Candidate image | Proposed method | DTW approach [6] |
|---|---|---|---|
|  |  | **1.26** | 6.07 |
|  |  | **0.74** | 6.46 |
|  |  | **1.05** | 8.91 |
|  |  | **0.85** | 4.88 |
|  |  | **0.37** | 1.48 |

## 8 Results

Table 1 and Table 2 show the matching cost computed using our approach and that developed by [6] for 5 example images from dataset 1 and dataset 2 respectively. It can be seen that the matching cost is smaller in our approach. In Table 3 the average precision at rank 1 is computed for our approach and two other approaches [6] and [17]. The average precision at rank 1 computed for our approach is more than that computed for the approaches [6] and [17]. PHOCNET is an attribute-based approach where for each word, PHOC (Pyramidal Histogram of Characters) is computed. This network has been trained for clean images with minimal distortion to the structure of the word. In our approach cluttering the word distorts its structure affecting the performance of PHOCNET.

## 9 Conclusion

In this paper we have proposed a novel problem of handwritten word spotting in the cluttered environment where a word is cluttered by line/lines. These lines can be straight, slant, broken, continuous or wavy in nature. VPP feature and its modified version cVPP feature are considered as feature vectors. cVPP considers the possible combinations of VPP in a cluttered word which lead to the selection of appropriate projection value for a column and, hence the rejection of cluttered pixels, thereby retrieving the matched word. DTW is used for aligning the two features and obtaining the cost of matching. A penalty term helps to overcome

**Table 2.** Matching cost for 5 examples from Dataset 2

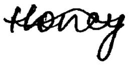
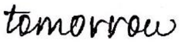
| Query image | Candidate image | Proposed method | DTW approach [6] |
|---|---|---|---|
| |  | **2.5** | 12.04 |
| | | **1.57** | 9.82 |
|  | | **1.38** | 11.13 |
| | | **0.62** | 13.46 |
| | | **0.68** | 12.87 |

**Table 3.** Average precision at rank 1

| Dataset | Proposed Method | DTW approach [6] | PHOCNET [17] |
|---|---|---|---|
| Dataset 1 | **0.92** | 0.86 | 0.36 |
| Dataset 2 | **0.80** | 0.76 | 0.24 |

the problem of addition of extra pixels due to the intersection of a word with clutter. The dataset for the proposed problem is not present so we prepared our dataset. We have compared our method with the method proposed in [6] and in [17] and have achieved improved performance over both the methods. Also, the comparison of matching cost has been made between our method and [6]. Our method performs better and we have obtained smaller matching cost for struck-through words.

## References

1. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD workshop. vol. 10, pp. 359–370. Seattle, WA (1994)
2. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character hmms. Pattern Recognition Letters **33**(7), 934–942 (2012)
3. Ghorbel, A., Ogier, J.M., Vincent, N.: A segmentation free word spotting for handwritten documents. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. pp. 346–350. IEEE (2015)
4. Giotis, A.P., Gerogiannis, D.P., Nikou, C.: Word spotting in handwritten text using contour-based models. In: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. pp. 399–404. IEEE (2014)
5. Lavrenko, V., Rath, T.M., Manmatha, R.: Holistic word recognition for handwritten historical documents. In: Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on. pp. 278–287. IEEE (2004)
6. Manmatha, R., Croft, W.: Word spotting: indexing handwritten manuscripts, intelligent multimedia information retrieval (1997)
7. Manmatha, R., Han, C., Riseman, E.M.: Word spotting: a new approach to indexing handwriting. In: Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 631–637 (June 1996). https://doi.org/10.1109/CVPR.1996.517139
8. Nagendar, G., Jawahar, C.: Efficient word image retrieval using fast dtw distance. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. pp. 876–880. IEEE (2015)
9. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1), 62–66 (1979)
10. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. vol. 2, pp. II–II. IEEE (2003)
11. Rath, T.M., Manmatha, R.: Word spotting for historical documents. International Journal of Document Analysis and Recognition (IJDAR) **9**(2-4), 139–152 (2007)
12. Rodríguez-Serrano, J.A., Perronnin, F.: Handwritten word-spotting using hidden markov models and universal vocabularies. Pattern Recognition **42**(9), 2106–2116 (2009)
13. Roy, P.P., Rayar, F., Ramel, J.Y.: Word spotting in historical documents using primitive codebook and dynamic programming. Image and Vision Computing **44**, 15–28 (2015). https://doi.org/https://doi.org/10.1016/j.imavis.2015.09.006, http://www.sciencedirect.com/science/article/pii/S0262885615001122
14. Rusiñol, M., Aldavert, D., Toledo, R., Lladós, J.: Efficient segmentation-free keyword spotting in historical document collections. Pattern Recognition **48**(2), 545–555 (2015)

15. Shanker, A.P., Rajagopalan, A.: Off-line signature verification using dtw. Pattern recognition letters **28**(12), 1407–1414 (2007)
16. Sudholt, S., Fink, G.A.: A modified isomap approach to manifold learning in word spotting. In: German Conference on Pattern Recognition. pp. 529–539. Springer (2015)
17. Sudholt, S., Fink, G.A.: Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In: Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on. pp. 277–282. IEEE (2016)
18. Toselli, A.H., Puigcerver, J., Vidal, E.: Context-aware lattice based filler approach for key word spotting in handwritten documents. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. pp. 736–740. IEEE (2015)
19. Toselli, A.H., Vidal, E.: Fast hmm-filler approach for key word spotting in handwritten documents. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. pp. 501–505. IEEE (2013)
20. Wang, P., Eglin, V., Garcia, C., Largeron, C., Lladós, J., Fornés, A.: A coarse-to-fine word spotting approach for historical handwritten documents based on graph embedding and graph edit distance. In: Pattern Recognition (ICPR), 2014 22nd International Conference on. pp. 3074–3079. IEEE (2014)
21. Zhang, X., Tan, C.L.: Handwritten word image matching based on heat kernel signature. Pattern Recognition **48**(11), 3346–3356 (2015)