

“© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Writer Identification by Training on One Script but Testing on Another

Chandranath Adak\*, Bidyut B. Chaudhuri†, Michael Blumenstein\*‡

\*School of ICT, Griffith University, Gold Coast-4222, Australia

†CVPR Unit, Indian Statistical Institute, Kolkata-700108, India

‡School of Software, University of Technology Sydney-2007, Australia

{adak32, bbcisical}@gmail.com, michael.blumenstein@uts.edu.au

**Abstract**—This paper deals with identifying a writer from his/her offline handwriting. In a multilingual country where a writer can scribe in multiple scripts, writer identification becomes challenging when we have individual handwriting data in one script while we need to verify/identify a writer from handwriting in another script. In this paper such an issue is addressed with two scripts: English and Bengali. Here we model the task as a classification problem, where training data contains only Bengali handwritten samples and testing is performed on English handwritten texts. This work is based on the understanding that a writer has some inherent stroke characteristics that are independent of the script in which (s)he writes. In this work, some implicit structural and statistical features are extracted, and multiple classifiers are employed for writer identification. Many training sessions are run on a database of 100 writers and the performances are analyzed. We have obtained encouraging results on this database, which show the effectiveness of our method.

**Index Terms**—Handwritten document, Latent feature, Writer identification, Writer verification.

## I. INTRODUCTION

Handwriting analysis is considered as a challenging task in pattern recognition due to the wide variability of human handwriting. Such variation occurs between persons, over time, geographic locations and cultural aspects. Sometimes, the writing of an individual varies with time/age, mental state, speed, intensive use of allographs and so on. Therefore, writer identification and verification, which are sometimes required in computational forensics, are important and useful tasks of practical concern.

Writer verification can be viewed as a binary classification problem, where the classifier decides whether a manuscript is written by a particular writer or not. Writer identification is perceived as an  $n$ -class problem, where a handwritten specimen is provided and the task is to determine its writer from a set of  $n$  possible writers.

Automation of such solutions has been pursued by the document processing community during the last three decades. A survey on classical methods of writer identification is reported in [1], while recent advancements can be found in [2], [3], [4].

In [4], Balacu and Schomaker have divided the task of writer verification/identification into two general groups: text dependent and text independent. Text dependent [5] identification/verification is basically performed on known semantic

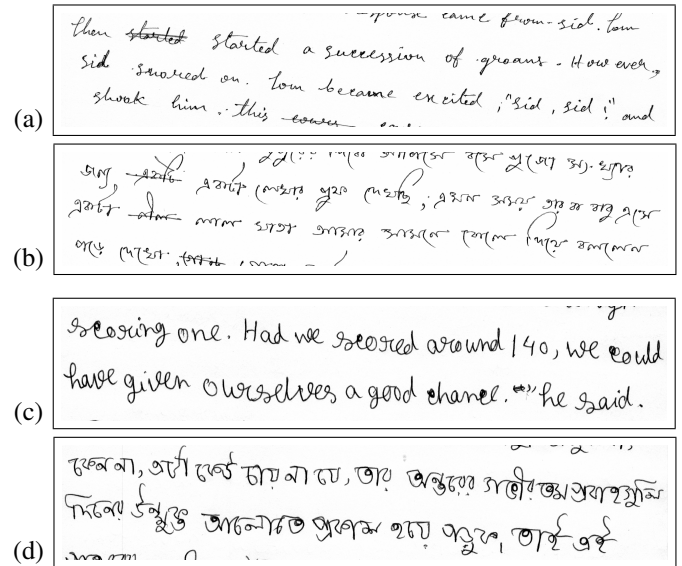


Fig. 1. Handwriting specimens of two separate writers in both (a),(c) English and (b),(d) Bengali script.

content, whereas a text independent method is, as its name suggests, unfettered by text content. This text independent [3], [4] identification is essential because at the time of handwriting matching, the same textual content written by the same person may not be available. It may also be a requirement to analyze some unknown texts written by a certain individual.

Recent offline writer identification techniques are based on allographs [4], textures [6], local binary patterns [7], contour/edge-based features [8], [9], and a combination of various features [4], [8]. In [10], a global descriptor is generated by a combination of local contour-Zernike moments, and used in writer identification. Schlapbach and Bunke [11] used an HMM (*Hidden Markov Model*)-based recognizer for writer identification/verification. SIFT (*Scale-Invariant Feature Transform*)-based techniques [3] are also used for such problems.

In a multilingual country, a writer may write in multiple (mostly two) scripts. Therefore, script independent writer identification is also essential. In ICDAR-2011, a contest [12] was organized on writer identification, where the participants were given handwriting of a writer in four different languages:

English, French, German and Greek. Wu et al. [3] used this ICDAR-2011 dataset in their SIFT-based writer identification approach and obtained fairly good results. In that work [3], samples from all four languages were used for training.

Now, a situation may arise when we have the knowledge of a writer's handwriting only in one script (say,  $S_{c_1}$ ) and we want to identify/verify the writer using test data in another script (say,  $S_{c_2}$ ). In this paper, we have taken this problem where the Bengali ( $S_{c_1}$ ) handwriting of a writer is available for training and we want to identify/verify this writer by testing on some English ( $S_{c_2}$ ) handwriting. This problem is interesting, since Bengali is an *Abugida* (Alpha-syllabary Indic) script, while English is an *Alphabetic* (Latin) script. The structures of these scripts [13] are dissimilar due to their difference in origin. However, we believe that there are some implicit personal characteristics in handwriting styles, which are script independent. In Fig.1(a),(b), the English and Bengali handwritings of an individual are shown. One more pair of writings by another writer is also shown in Fig.1(c),(d).

In West Bengal (India) and Bangladesh, the written medium of education is mostly Bengali and English. This region was under the British colonial power for about 200 years and hence the second language in school/college education is English. The same situation may occur in many countries of Europe, where writing in English is popular besides French, German, Greek, Italian, Spanish and so on. However, for such cases, the writer identification task is relatively easier since both English and native scripts are Roman-based only.

Ramaiah and Govindaraju [14] proposed an accent-based writer identification method of online handwriting which worked in two stages, namely "accent identification" followed by "writer identification". Although our work is focused on Bengali native writers, their work [14] is different from ours, because they have sufficient training data. However, in our approach training samples of a script (e.g. English) are considered unavailable. It uses some shared information from another script (e.g. Bengali) for training. We want to find out, given a training dataset (offline handwritten page of a writer) in Bengali script, whether it is possible to identify/verify the writer when we get his/her English writing for testing. Therefore, the main thrust of our study is to train our system with Bengali handwriting of an individual and to test it against the individual's English writing. Here we select some novel structural and statistical features based on the information from full text-pages, text-lines, words, characters, graphemes and even pixels. These features are employed for the writer classification (identification and verification) task.

In this paper, the contributions of our work are as follows:

1. Tackling a need-based relatively new problem, that is, identifying a writer from an unknown script (English) using prior knowledge about another script (Bengali).
2. Presenting a study on some moderately novel latent features, both structural and statistical, which have impact on writer identification while a different script is used for supervised training.
3. Analyzing the performance of various classifiers and their

combinations related to such a special type of writer identification/verification.

4. Generating a new handwritten database from 100 writers for investigation into such problems with which this paper is concerned.

The rest of the paper proceeds as follows:

Section II describes our proposed method in detail. Section III presents and analyzes the experimental results, and Section IV concludes the paper.

## II. PROPOSED METHOD

Bengali is an alpha-syllabary script and English is an alphabetic script. There exist a good degree of differences among characters of these two scripts. In this problem, we cannot employ character shape matching to identify the writer. However, a writer may imprint his latent identity in the margin/page-border, text-line skew, word slant, interline and inter-word gaps etc., which may be nearly identical when the individual writes in either script. Also, sub-segments of characters in two different scripts may have some identical shapes. In some computational analysis based on writing direction, curvature may be useful for our purpose. We start with extraction of various features of individual writing styles with the objective of writer identification and verification. The details are as follows.

### A. Feature Extraction

An elaborate description of the useful features computed for our problem is given below.

1) *Margin Space Width*: When an individual writes on a four-sided blank white page, the starting position and the end position of each text-line is bounded by two-sided margins with white space. When writing on a full page is completed, we can also find the upper and lower side margins. Now both Bengali and English scripts are written in a left-to-right horizontal direction. By inspecting a moderate number of handwritings, we note that more variation occurs at the left and top margins. Therefore, the left and top margins are more important compared to the right and bottom margins for writer identification.

In [15], several types of margins and their impact on the writer's mental condition are considered. But, in this paper, we refrain from examining such graphological issues. We only calculate the average white-border width outside the margin. The margins are detected using a simple projection profile based technique [16], which is effective for our purpose. The left/right margin space widths are calculated using a vertical histogram, whereas top/bottom margins are computed by a horizontal histogram. These margin space widths (left, top, right and bottom) are used as features  $f_1$ ,  $f_2$ ,  $f_3$  and  $f_4$ .

The margin space width does not appear to be a robust feature, but inspired by [15], we employ margins as a latent feature for individual writing style. To investigate the impact of margins, we collect the text written on white A4 sized pages (Section III-A).

2) *Text-line Skewness*: When written on a white blank page (which does not contain any ruled line), the individual handwritten text-lines may not be horizontally aligned and skewed (oriented) at some position/negative angle.

Before skew detection, the document image should be segmented into text lines. For text-line segmentation, we use a state-of-the-art method [17], where a Gaussian window and a level set technique are used.

Now a reasonably straight line is considered horizontally through the middle of the segmented text-line. Next, the skew is measured as the angle between this line and the horizontal axis. For a text-line, the skew may be upward or downward direction, that is, making a positive or negative slope with the horizontal axis. We take the average and standard deviation of skewed angles of all upward directed text-lines. The average and standard deviation are used as features  $f_5$  and  $f_6$ , respectively. Similarly, for downward directed text-lines, we also obtain the average and standard deviation of the skew angles and call these as features  $f_7$  and  $f_8$ .

3) *Word Slant*: Word slant is like a shearing transformation over a straight word when it is transformed into up-right or up-left and rigidly attached to the baseline. Usually an individual writes by maintaining his/her characteristic slant irrespective of scripts. So, we use the slant-angle as a feature.

Before proceeding for word slant estimation, word segmentation is required. For this purpose we use a 2D Gaussian filter on the document image. The standard deviation ( $\sigma$ ) and block size ( $a \times b$ ) for this filter is chosen automatically by analyzing the connected components in a text-line. Since the words are arranged horizontally in Bengali (and English) script(s), the block size is chosen to be longer in width than in height. After blurring, an adaptive threshold is used for binarization. The connected components of this binarized image are considered as individual words.

The word slant can be calculated by the vertical parts of a word. In Bengali and English scripts, the characters having vertical parts help us to calculate the word slant. For this slant estimation we use the method given in [18]. This technique is based on vertical projection profiles and the Wigner-Ville distribution. The average and standard deviation of the word-slants are used as features  $f_9$  and  $f_{10}$ , respectively.

4) *Inter-textline and Inter-word Gap*: Between two consecutive text-lines, a writer leaves some white space to make a demarcation between the lines, called the inter-textline gap. Similarly, between two consecutive words an inter-word white gap is maintained. Such white gaps also vary with individual handwriting and reflect the characteristic of the writer.

From the text-line and word segmented image, we find the inter-textline and inter-word gaps. The average and standard deviation of inter-textline gaps are calculated and used as features  $f_{11}$  and  $f_{12}$ , respectively. For words also, the inter-word gap average and standard deviation are obtained to use as features  $f_{13}$  and  $f_{14}$ .

5) *Main-body Text Height*: We notice that when a person writes either in Bengali or in English, the text main body height remains almost the same. For text main body height

calculation, we analyze individual words obtained from the segmented word image.

To detect the main body text height, we partition a word image into three zones (upper, middle and lower) by two imaginary lines called upper and lower baselines. These baselines are obtained by simple horizontal histogram projections. Some details of the approach can be found in [19]. The maximum number of object (ink) pixels generally lies in the middle zone, where the horizontal histogram profile reaches the global maximum. This middle zone is actually the main body of a word, whose height is calculated for our requirement. The average and standard deviation of main body heights of all words are calculated and termed as features  $f_{15}$  and  $f_{16}$ , respectively.

6) *HOG at Keypoints*: We intend to capture some common aspects of writing styles of Bengali and English scripts. So, we try to find some keypoints (points of interest)  $p_i$  on the handwritten text image, since both the Bengali and English handwritten ink-strokes contain start/end-points, branch-points and curved-points [20]. Here we obtain some structural and SIFT keypoints on the text-strokes similar to the scheme used in [20].

In the neighborhood of each keypoint  $p_i$ , a small window of size  $n \times n$  is considered. In this window, we obtain a histogram of oriented gradients (HOG) [21] considering 8-bin angular ( $360^\circ/45^\circ$ ) information. Here the gradient strengths are normalized using  $L_2$ -norm. We use this HOG descriptor as feature  $f_{17}$ , since it stores significant information about text-stroke gradient change. We note that substantial change in gradient occurs for both Bengali and English handwritten strokes.

7) *Direction and Curvature at Keypoints*: On the ink-strokes, the above keypoints are used to find directional and curvature features. Here we take the idea of feature extraction for ‘‘The NPen++ Recognizer’’ [22] and use it according to our requirement in the offline situation as follows.

Between two connected (with ink-stroke pixels) keypoints  $p_i$  and  $p_{i+1}$ , we calculate its *writing direction* in Cosine and Sine values, and employ them as features  $f_{18}$  and  $f_{19}$ , respectively:

$$f_{18} \equiv \cos(\theta_i) = \frac{p_{i+1}.row - p_i.row}{d_i} \quad (1)$$

$$f_{19} \equiv \sin(\theta_i) = \frac{p_{i+1}.col - p_i.col}{d_i} \quad (2)$$

where,  $p_i.row$  and  $p_i.col$  are the row and column indices of  $p_i$ , and  $d_i = \sqrt{(p_{i+1}.row - p_i.row)^2 + (p_{i+1}.col - p_i.col)^2}$ .

The *curvature* of the ink-stroke is basically the angle (Cosine and Sine) generated by lines from  $p_{i-1}$  to  $p_i$  and  $p_i$  to  $p_{i+1}$ , which are used as features  $f_{20}$  and  $f_{21}$ :

$$f_{20} \equiv \cos(\theta_{i+1} - \theta_i) = \cos\theta_{i+1}\cos\theta_i + \sin\theta_{i+1}\sin\theta_i \quad (3)$$

$$f_{21} \equiv \sin(\theta_{i+1} - \theta_i) = \sin\theta_{i+1}\cos\theta_i - \cos\theta_{i+1}\sin\theta_i \quad (4)$$

These features are acquired from each keypoint and used to obtain separate histograms.

8) *Loop-like Shapes*: Some Bengali and English characters contain loop-like structures. Also, some extra loop-like patterns may arise due to individual cursive writing styles. Therefore, the loop feature plays an important role in our problem. In [23], Sutanto et al. used loop-based features for discrimination of the handwritten character ‘a’. Motivated by [23], we also use two types of loop feature.

In a loop region, let  $LP$  be the set of object pixels, and let  $LC$  be the set of pixels ( $p \in LC$ ) on the boundary of this loop region. The counts of object pixels in  $LP$  and  $LC$  are denoted by  $n(LP)$  and  $n(LC)$ , respectively. The center of gravity ( $CG$ ) inside the loop region is ( $CG.row, CG.col$ ) given by:  $CG.row = \frac{1}{n(LP)} \sum_{p \in LP} p.row$ ;  $CG.col = \frac{1}{n(LP)} \sum_{p \in LP} p.col$ .

We measure the standard deviation ( $\sigma_L$ ) of the loop as distinct from an ideal circle and employ it as feature  $f_{22}$  to fit our task.

$$f_{22} \equiv \sigma_L = \sqrt{\frac{1}{n(LC)} \sum_{p \in LC} (X_p - \mu_L)^2} \quad (5)$$

where,  $X_p = \sqrt{(p.row - CG.row)^2 + (p.col - CG.col)^2}$  and  $\mu_L = \frac{1}{n(LC)} \sum_{p \in LC} X_p$ .

Finally, we calculate another loop-feature ( $f_{23}$ ) based on the roundness of an object.

$$f_{23} \equiv roundness = \frac{4\pi \cdot n(LP)}{(\sum_{p \in LC} X_p)^2} \quad (6)$$

Thus, we have obtained these 23 types of features which are extracted from full page information, text-line/word/character level, and even semi-character/grapheme/point level information. All features are normalized with respect to the text height. The features  $f_1$ - $f_{16}$  are single-valued features, whereas  $f_{17}$ - $f_{23}$  are individual sets (histograms) of features. The  $f_{17}$ - $f_{23}$  feature sets are empirically downsampled to reduce computational costs.

### B. Classification

We formulate the writer verification as a 2-class classification problem and the writer identification as an  $n$ -class classification task. For such classification, we employ some well-known classifiers such as KNN (*K-Nearest Neighbors*), MLP (*Multi-Layer Perceptron*) and SVM (*Support Vector Machine*). Here, at the time of classifier training, only Bengali handwriting is available; whereas for testing, only the English handwritten specimen is provided.

The input metrics of the classifiers are to be tuned and some pre-arrangements of classifiers are to be considered, such as the distance function of KNN, the training function and number of hidden layers (and number of neurons) for MLP, and also the kernel function and parameters of the SVM.

We also use different sized subsets of training data in multiple sessions to design multiple classifiers. The classifier outputs are combined using a fusion technique to improve the performance.

More details and performance of the classifiers are provided in Section III-B.

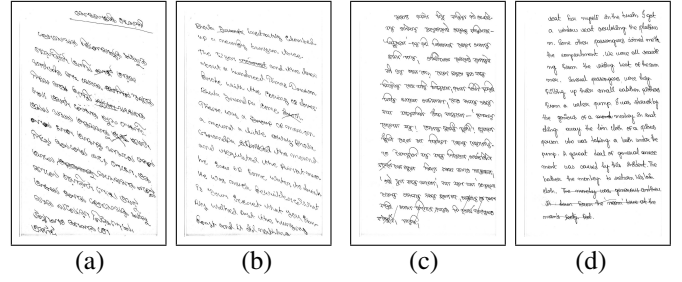


Fig. 2. Image samples from our database of (a)-(b) Writer-001 and (c)-(d) Writer-064. (a), (c) Bengali and (b), (d) English handwritings.

## III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, at first we discuss the database employed for the experiments.

### A. Database Generation

For our experiments, we needed such a database where an individual had written some pages in Bengali (required for training) and English (for testing). Although several open-source handwritten Bengali and English databases were available separately, we did not find any Bengali/English datasets with proper ground-truth stating the writer information as per our requirements. Hence, we generated our own database by inviting 100 volunteers. Those volunteers were native Bengali writers from West Bengal, India, and were of both genders with different ages and academic backgrounds. Each volunteer wrote 3 full pages (A4 sized) of Bengali and 1 page of English from any book/article or from their memory, in their regular handwriting. The A4 sized 75 GSM ( $g/m^2$ ) white pages and black-inked pens with 0.5 mm ball-point of the same brand/model were provided to all writers to obtain prominent ink-strokes of similar color intensity and thickness.

A total of 300 pages of Bengali and 100 pages of English handwriting with writer information are present in our database. This scenario is quite realistic while the labelled dataset (e.g. English here) is absent, and we want to see the impact of learning by sharing information (from Bengali script). Here, each page contains about 16 text-lines, and 10 words per text-line. Thus, approximately 4,800 ( $= 300 \times 16$ ) Bengali and 1,600 ( $= 100 \times 16$ ) English text-lines are present. Also, the total numbers of Bengali and English handwritten words in this database are almost 48,000 and 16,000, respectively. Typical handwriting specimens from our database are shown in Fig.2, where text-line skew variation can be seen.

### B. Results and Evaluation

In this subsection, at first we discuss the arrangements/settings for the classifiers.

In our KNN classifier, the exhaustive nearest neighbor search with the ‘Euclidean’ distance function was used. The value of ‘K’ was set to the square root of the number of data points in the training set.

We used two variants of MLP (say MLP<sub>1</sub> and MLP<sub>2</sub>). For MLP<sub>1</sub>, the training function employed was “scaled conjugate

gradient backpropagation”. This training function took less memory. Here the performance function was “*Mean Squared Error*” (MSE). Only one hidden layer was used here. The number of neurons in this hidden layer was set empirically from 500 to 100. The transfer function used for activation of neurons, was “log-sigmoid”. The number of training epochs was increased from 1,000 up to 30,000.

In the MLP<sub>2</sub>, we used the “Levenberg-Marquardt back-propagation” algorithm as a training function. It was relatively faster but required more memory. Here also MSE was used as the performance function. Two hidden layers with neurons from 500 to 200 were used here. The “hyperbolic-tangent sigmoid” transfer function was applied to activate the neurons. The count of training epochs was from 1,000 to 30,000.

In the SVM classifier, we used the RBF (*Radial Basis Function*) as the kernel, where the hyper-parameters ( $\gamma$  and  $C$ ) were evaluated from a tuning set in the logarithmic scale, for  $\gamma$  in  $\{2^{-3}, 2^{-2}, \dots, 2^4\}$  and  $C$  in  $\{2^{-2}, 2^{-1}, \dots, 2^7\}$ . From experiments in the training phase, the best performance was obtained for  $\gamma = 2^2$  and  $C = 2^5$ , respectively. These values were used throughout the experiments on the test data.

As stated earlier, we used Bengali handwriting for training. To obtain multiple classifiers ( $C_1$ ,  $C_2$  and  $C_3$ ), we trained the classifiers by using 3 subsets ( $s_1$ ,  $s_2$  and  $s_3$ ) of the training data. These subsets  $s_1$ ,  $s_2$  and  $s_3$  contained 1, 2, 3 Bengali handwritten pages of each writer, respectively. So, for a total of 100 writers, in  $s_1$ ,  $s_2$  and  $s_3$ , there were approximately 1600, 3200 and 4800 text-lines, respectively. The classifier outputs were combined using the ‘*sum rule*’, i.e., addition of the confidence measure of each classifier. This fusion strategy works better than some other fusion schemes such as the *max*, *min*, *median*, *majority vote* and *product rule* [24]. We also performed fusion of KNN, MLP<sub>1</sub>, MLP<sub>2</sub> and SVM classifier outputs to analyze the performance.

The test set contained an English handwritten page having approximately 16 text-lines and 160 words for each writer. Writer identification and verification tasks were executed on this test set. For writer identification we used the Top-N criterion, where the possible writer was extracted in a reduced set of ‘N’ ( $\ll$  total number of writers) elements. Here, we choose the Top-1, Top-2 and Top-5 criteria, and present their performance (basis of F-Measure) in Table I, II and III, respectively. The bar chart (Fig.3) also depicts the performance of writer identification. The results of writer verification are shown in Table IV. The last row and last column of Table I-IV show the results after combining classifiers. It is observed that the combined classifiers consistently outperform the standalone classifiers.

In Fig.4, we present the impact of different features used by various classifiers for Top-1 writer identification, while training on Bengali dataset ( $s_3$ ). Here the  $x$ -axis denotes the cumulative feature types, e.g., ‘1’ represents feature  $f_1$ , ‘2’ represents  $f_1 + f_2$ , ‘3’ represents  $f_1 + f_2 + f_3$ , and so on. The performance of all classifiers correlates positively with the increase in the number of feature types from 1 to 23.

TABLE I  
WRITER IDENTIFICATION PERFORMANCE (TOP-1)

Classifiers	F-Measure (%)			
	$C_1$	$C_2$	$C_3$	$C_1+C_2+C_3$
KNN	54.62	57.13	58.74	59.27
MLP <sub>1</sub>	55.18	57.87	62.43	64.74
MLP <sub>2</sub>	57.26	61.34	65.37	68.59
SVM	61.84	63.39	66.56	70.02
KNN+MLP <sub>1</sub> +MLP <sub>2</sub> +SVM	62.33	64.45	67.92	<b>71.19</b>

TABLE II  
WRITER IDENTIFICATION PERFORMANCE (TOP-2)

Classifiers	F-Measure (%)			
	$C_1$	$C_2$	$C_3$	$C_1+C_2+C_3$
KNN	54.78	58.38	59.10	59.66
MLP <sub>1</sub>	56.27	59.33	63.17	65.82
MLP <sub>2</sub>	59.73	64.45	66.30	70.74
SVM	63.58	66.07	69.67	72.58
KNN+MLP <sub>1</sub> +MLP <sub>2</sub> +SVM	64.25	67.48	70.53	<b>73.81</b>

TABLE III  
WRITER IDENTIFICATION PERFORMANCE (TOP-5)

Classifiers	F-Measure (%)			
	$C_1$	$C_2$	$C_3$	$C_1+C_2+C_3$
KNN	56.63	59.52	60.98	61.28
MLP <sub>1</sub>	57.31	61.35	65.60	67.52
MLP <sub>2</sub>	60.72	66.69	68.48	72.37
SVM	65.43	69.40	71.56	74.52
KNN+MLP <sub>1</sub> +MLP <sub>2</sub> +SVM	66.12	69.66	72.75	<b>75.68</b>

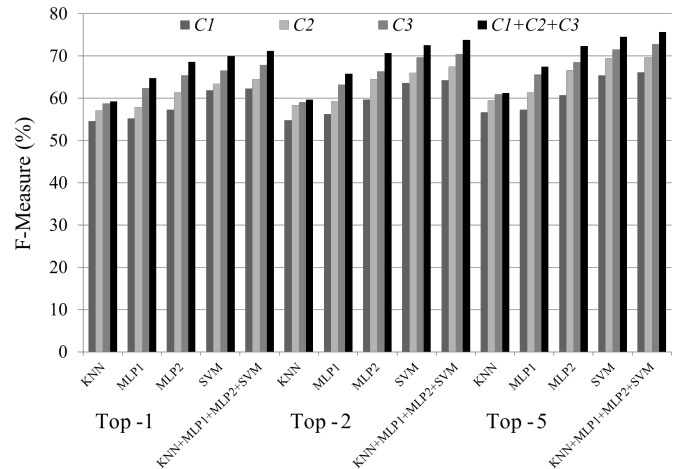


Fig. 3. Bar chart: Top-1, Top-2 and Top-5 writer identification performance.

TABLE IV  
PERFORMANCE OF WRITER VERIFICATION

Classifiers	F-Measure (%)			
	$C_1$	$C_2$	$C_3$	$C_1+C_2+C_3$
KNN	64.93	66.23	70.98	72.54
MLP <sub>1</sub>	65.75	68.15	72.57	75.62
MLP <sub>2</sub>	68.93	74.49	75.34	80.28
SVM	72.61	75.54	78.76	81.76
KNN+MLP <sub>1</sub> +MLP <sub>2</sub> +SVM	74.35	76.60	79.53	<b>83.66</b>

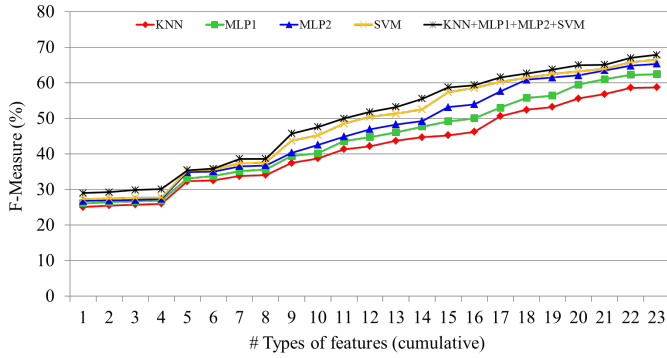


Fig. 4. Impact of different features used by various classifiers for English writer identification (Top-1), while training on Bengali dataset ( $s_3$ ).

For *comparison* of our work with others, we have not found any work where the training is performed on Bengali handwriting and the testing is undertaken on English data.

We have analyzed possible reasons behind such fairly encouraging outcomes of our work, which are as follows:

1. Although Bengali and English scripts have no similarity in overall character structures, but there are a few shape-wise resemblances between Bengali and English character curvatures, e.g., the English character ‘c’ and Bengali grapheme ‘ঔ’. The main difference is that the curvature direction for ‘c’ is left-to-right, while for ‘ঔ’, it is right-to-left.

2. Both English and Bengali text-lines are written in a left-to-right direction. So, there are similarities in writing flow and direction of text-lines.

3. India was under British rule for approximately 200 years. For many Bengali writers, their second language is English in school-level education. So, the Bengali writing style may have been influenced by English writing for a long period of time.

Our work may be extended to investigate the impact of writing in various scripts and to study whether there is any similarity among multiple handwritten manuscripts.

#### IV. CONCLUSION

In this paper, we deal with writer identification and verification on English script while supervised learning is performed on Bengali offline handwriting. Here we use some latent features with multiple classification techniques for identification/verification. To evaluate the efficacy of our system, a database of 100 writers scribing 300 Bengali and 100 English pages, containing 4,800 and 1,600 text-lines, respectively, has been generated. This database will be freely available for academic research purposes upon receipt of an e-mail request. Writer identification on the Top-1, Top-2 and Top-5 criteria have produced an overall F-Measure of 71.19%, 73.81% and 75.68%, respectively. Also, for writer verification an 83.66% F-Measure has been obtained. Our future work will focus on obtaining higher accuracy by introducing some more useful features for discriminating different types of handwriting. Our next endeavor will be a collaborative study with researchers of various multilingual countries to collect more datasets for further experimental analysis of this work.

#### REFERENCES

- [1] R. Plamondon, G. Lorette, “Automatic Signature Verification and Writer Identification - The State of the Art”, *Pattern Recognition*, vol.22, no.2, pp.107-131, 1989.
- [2] L. Schomaker, “Advances in Writer Identification and Verification”, *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, vol.2, pp.1268-1273, 2007.
- [3] X. Wu, Y. Tang, W. Bu, “Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform”, *IEEE Trans. on Information Forensics and Security*, vol.9, no.3, pp.526-536, 2014.
- [4] M. Bulacu, L. Schomaker, “Text-Independent Writer Identification and Verification Using Textural and Allographic Features”, *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol.29, no.4, pp.701-717, 2007.
- [5] E.N. Zois, V. Anastassopoulos, “Morphological Waveform Coding for Writer Identification”, *Pattern Recognition*, vol.33, no.3, pp.385-398, 2000.
- [6] D. Bertolini, L.S. Oliveira, E. Justino, R. Sabourin, “Texture-based Descriptors for Writer Identification and Verification”, *Expert Systems with Applications*, vol.40, no.6, pp.2069-2080, 2013.
- [7] L. Du, X. You, H. Xu, Z. Gao, Y. Tang, “Wavelet Domain Local Binary Pattern Features For Writer Identification”, *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pp.3691-3694, 2010.
- [8] L. Schomaker, M. Bulacu, “Automatic Writer Identification Using Connected-Component Contours and Edge-Based Features of Uppercase Western Script”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.26, no.6, pp.787-798, 2004.
- [9] R. Jain, D. Doermann, “Writer Identification Using an Alphabet of Contour Gradient Descriptors”, *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp.550-554, 2013.
- [10] V. Christlein, D. Bernecker, E. Angelopoulou, “Writer Identification Using VLAD Encoded Contour-Zernike Moments”, *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp.906-910, 2015.
- [11] A. Schlapbach, H. Bunke, “A Writer Identification and Verification System Using HMM Based Recognizers”, *Pattern Analysis and Applications*, vol.10, no.1, pp.33-43, 2007.
- [12] G. Louloudis, N. Stamatopoulos, B. Gatos, “ICDAR 2011 Writer Identification Contest”, *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp.1475-1479, 2011.
- [13] B.B. Chaudhuri, U. Pal, “A Complete Printed Bangla OCR System”, *Pattern Recognition*, vol.31, no.5, pp.531-549, 1998.
- [14] C. Ramaiah, V. Govindaraju, “A Hierarchical Framework for Accent Based Writer Identification”, *Proc. IAPR Int. Workshop on Document Analysis Systems (DAS)*, pp.21-25, 2014.
- [15] S. Lowe, “The Complete Idiot’s Guide to Handwriting Analysis”, ISBN-13: 978-1592576012, Publisher-ALPHA, 2<sup>nd</sup> Edition, 2007.
- [16] N. Stamatopoulos, B. Gatos, A. Kesidis, “Automatic Borders Detection of Camera Document Images”, *Proc. Int. Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pp.71-78, 2007.
- [17] L. Yi, Z. Yefeng, D. Doermann, “Detecting Text Lines in Handwritten Documents”, *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pp.1030-1033, 2006.
- [18] E. Kavallieratou, N. Fakotakis, G.K. Kokkinakis, “Slant Estimation Algorithm for OCR Systems”, *Pattern Recognition*, vol. 34, no.12, pp.2515-2522, 2001.
- [19] U.-V. Marti, H. Bunke, “Using A Statistical Language Model to Improve the Performance of An HMM-based Cursive Handwriting Recognition Systems”, *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol.15, no.1, pp.65-90, 2001.
- [20] C. Adak, B.B. Chaudhuri, “Writer Identification from Offline Isolated Bangla Characters and Numerals”, *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp.486-490, 2015.
- [21] N. Dalal, B. Triggs, “Histograms of Oriented Gradients for Human Detection”, *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol.1, pp.886-893, 2005.
- [22] S. Jaeger, S. Manke, J. Reichert, A. Waibel, “Online Handwriting Recognition: The NPen++ Recognizer”, *Int. Journal on Document Analysis and Recognition*, vol.3, no.3, pp.169-180, 2001.
- [23] P.J. Sutanto, G. Leedham, V. Pervouchine, “Study of the Consistency of Some Discriminatory Features Used by Document Examiners in the Analysis of Handwritten Letter ‘a’”, *Proc. Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp.1091-1095, 2003.
- [24] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, “On Combining Classifiers”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.20, no.3, pp.226-239, 1998.