Impact of Struck-out Text on Writer Identification

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*—The presence of struck-out text in handwritten manuscripts may affect the accuracy of automated writer identification. This paper presents a study on such effects of struck-out text. Here we consider offline English and Bengali handwritten document images. At first, the struck-out texts are detected using a hybrid classifier of a CNN (Convolutional Neural Network) and an SVM (Support Vector Machine). Then the writer identification process is activated on normal and struck-out text separately, to ascertain the impact of struck-out texts. For writer identification, we use two methods: (a) a hand-crafted feature-based SVM classifier, and (b) CNN-extracted auto-derived features with a recurrent neural model. For the experimental analysis, we have generated a database from 100 English and 100 Bengali writers. The performance of our system is very encouraging.**

***Index Terms*—CNN; Crossed-out text; Recurrent neural network; Struck-out text; SVM; Writer identification.**

I. Introduction

Writer identification is a challenging task in the field of handwriting analysis owing to the intensive variation of human writing styles over space and time. However, promising results with acceptable accuracy have been obtained in the state-of-the-art methods of identifying a writer by his/her free-form running handwriting in various Roman-based western [1], [2] as well as Oriental scripts [3], [4]. A detailed survey on writer identification up to the year 1989 can be found in [1]. Some advanced information on this topic is also available in [2].

Writer identification can be seen as a classification problem, where the task is to detect a class (writer) among n classes (writers), given a handwriting specimen. This specimen may be a full page of writing. Even a paragraph/text-line [5]/word [6]/character [4], [7] only may be available for writer identification.

The recent writer identification techniques are built upon on edge/contour-based features [8], [9] of writing strokes, texture patterns [10], projection profiles [11], allographs [12], and combinations of various micro and macro features [13].

However, most published papers on writer identification consider ideal writing as the input, i.e. texts containing no writing error. In reality, a writer may strike-out/cross-out inappropriate words. Therefore, a handwritten document may contain such struck-out texts. Such examples are shown in Fig. 1. The struck-out texts may have some impact on the automatic writer identification process, which is the primary focus of this work. For this purpose, we analyze the task on struck-out text and normal text separately.

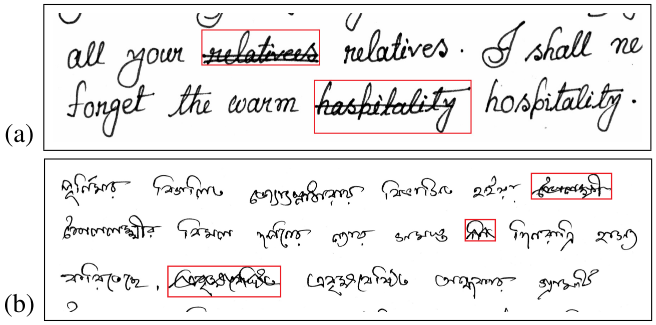
(a)
all your ~~relatives~~ relatives. I shall ne
forget the warm ~~haspitality~~ hospitality.
(b)

Fig. 1. Examples of (a) English and (b) Bengali handwritten documents containing struck-out texts (marked by red boxes).

To perform the above, the detection of struck-out/crossed-out text is necessary. In the literature, a few papers deal with struck-out text detection. An early report by Arlandis et al. [14] mentioned "crossing-outs", "scribbles", "isolated strokes", without giving any clear solution. Tuganbaev and Deriaguine [15] filed a US patent for their crossed-out character recognizer using a feature-based classifier. A work on HMM (*Hidden Markov Model*)-based crossed-out word recognition was reported in [16]. A graph-based approach for finding struck-out (or, "strike-through") words from handwritten manuscripts was shown in [17]. A modified version of [17] was presented in [18], where a feature-based SVM classifier was used for struck-out text detection.

In this paper, we consider offline handwriting of an alphabetic Latin script *English* and an alpha-syllabary Indic script *Bengali* (or, *Bangla*). Recent advances in writer identification on Indic scripts are mentioned in [4]. Here, for struck-out text detection, we use CNN (*Convolutional Neural Network*) as a feature extractor and SVM (*Support Vector Machine*) as a classifier. For writer identification, we use two separate systems: (i) hand-crafted feature-based SVM classifier, (ii) CNN with a recurrent neural network. We use BLSTM (*Bidirectional Long Short-Term Memory*) neural net architecture in this study.

The main contributions of this paper are as follows:
(i) Analyzing the impact of struck-out text on writer identification.
(ii) Detecting struck-out text by a hybrid classifier (CNN and SVM).
(iii) Identifying the writer using CNNs followed by a recurrent neural network.
(iv) Experimenting on alphabetic English and alpha-syllabary

Bengali scripts as well as generating a database of handwritten specimens containing struck-out texts of 100 English and 100 Bengali writers.

The rest of the paper is arranged as follows. Section II describes the proposed method. Then, the experiments and performance evaluation are presented in Section III. Finally, Section IV concludes the paper.

II. Proposed Method

As stated before, the impact of struck-out text on writer identification can be analyzed by identifying such text in the document. The work-flow of the proposed method for undertaking this is shown in Fig. 2.

In the preprocessing stage, a handwritten page is segmented into text components using single-pass connected component labeling [19]. This single-pass algorithm is relatively faster than classical two-pass connected component labeling methods. Very small-sized components such as dots, dashes, commas, colons etc. and noise are filtered out. We also segment the words using an off-the-shelf 2D Gaussian filter-based method, called *GOLESTAN-a* and described in [20].

A. Normal and Struck-out Text Separation

In handwritten documents, the ideal text without any writing-error is considered as "*normal*" text and the erroneous text containing a strike-out stroke is grouped as "*struck-out*". Here, the task is perceived as a binary classification problem, where we have to classify a word into *normal* versus *struck-out* class.

In recent days, the deep neural networks have created a benchmark for many machine vision applications [21]. The researchers have obtained outstanding results using CNNs as a feature extractor [22]. Sometimes a hybrid model performs better than stand-alone methods. Niu and Suen [23] proposed a CNN-SVM hybrid classifier for recognizing English handwritten numerals and obtained impressive results. In that work, they used a simplified CNN model instead of a more complex LeNet-5 [22].

1) Feature extraction: In our approach, we adapt the common LeNet-5 CNN architecture as per our requirement for feature selection. Our CNN model, employed as a feature extractor, is schematically shown in Fig. 3.

Here, the preprocessed word level text components are fed into the CNN as inputs. These components are normalized to a fixed size of 32×92 pixels. The normalized image passes through convolutions followed by subsampling operations.

Here, the first convolution layer C1 contains 6 feature maps, whereby a 5×5 convolutional filtering kernel is used. Thus, each unit of the feature map is connected with a 5×5 area of the input image, called a "receptive field". All units in the feature map share the same kernels and the same set of weights. The size of this C1 feature map is 28×88.

The following subsampling layer S2 is comprised of 6 feature maps of size 14×44. Each feature map is connected with non-overlapping 2×2 receptive fields of C1. Here a *max-pooling* operation with 2×2 filters and a stride of 2 is used for the subsampling from C1 to S2.

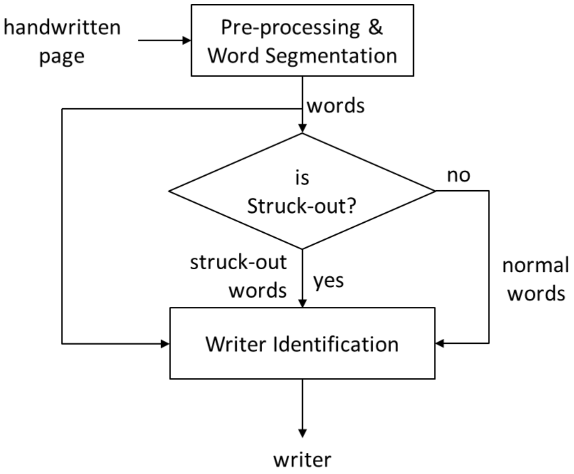

Fig. 2. Work-flow of the proposed method.

The second convolution layer C3 is of 16 feature maps with size of 10×40. Here also a 5×5 convolution filter is used, while each unit of the feature map is connected with 5×5 receptive fields of S2.

Similarly, C3 is followed by another subsampling layer S4. This S4 layer contains 16 feature maps of size 5×20. As before, the max-pooling operation is employed with a 2×2 filter size and a stride of 2.

Adopting the idea of LeNet-5, we add another layer F5 after S4, quite similar to the C5 of LeNet-5. In C5, LeCun et al. [22] used 120 feature maps. In LeNet-5, there was the full connection between S4 and C5, since S4 contained 16 feature maps of size 5×5 and the convolution kernel size was also 5×5. For our case, the S4 feature map size is 5×20. We assume it as 4 horizontally segmented non-overlapping feature maps of size 5×5 each. Therefore, we consider the F5 layer as 4×120 feature maps, where each 120 feature map is fully connected with the 5×5 segmented feature maps of S4.

In this way, our CNN model extracts the feature vector of dimension 480 (= 120×4).

2) SVM classifier: The feature vector extracted from the CNN is to be fed into a classifier for separation of normal and struck-out texts.

We use the SVM with an RBF (*Radial Basis Function*) kernel [24] as a classifier, since it works better than MLP (*Multi-Layer Perceptron*), MQDF (*Modified Quadratic Discriminant Function*), k-NN (k-*Nearest Neighbors*) and SVM-linear for handwriting analysis [18], [25]. Tuning of the SVM-RBF hyper-parameters (γ and $\mathcal{C}$) is required to avoid over-fitting and to control the decision boundary [26]. Such parameters are selected from a tuning set for the optimal performance of the classifier. This process is called "model selection". We use a traditional *grid-searching* technique for model selection.

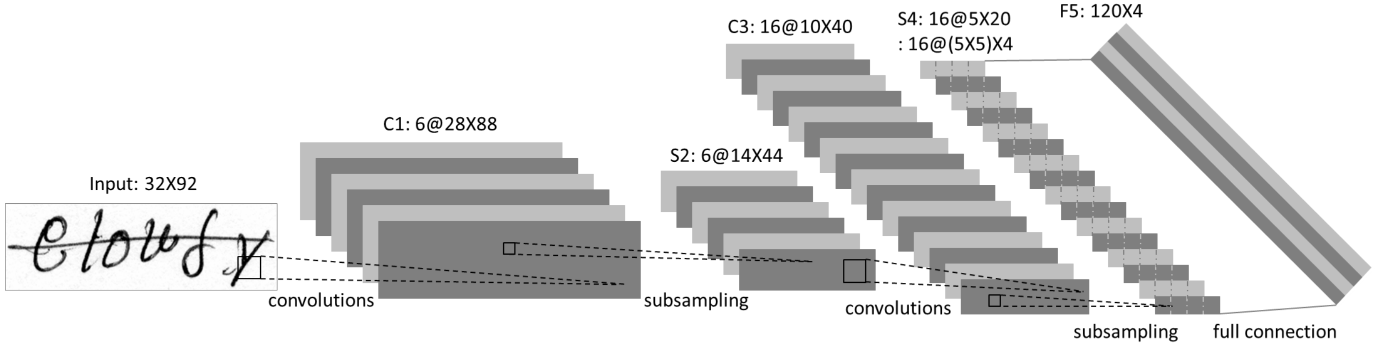

Fig. 3. Our CNN architecture as a feature extractor.

Here k-fold cross-validation is also used on the training set. The choice of grid-searching range, selection of γ and $\mathcal{C}$, and the choice of k are discussed in Section III-B.

B. Writer Identification

To see the impact of struck-out text on writer identification, we execute our experiments on three types of texts: (i) all handwritten texts, i.e. normal and struck-out, (ii) only normal texts, (iii) only struck-out texts.

The writer identification task can be perceived as an n-class classification problem, where the job is to map an unknown class (handwritten specimen) to its class-id (writer).

In the pre-processing stage, we segment the handwritten page into words. We feed these words, due to their discriminatory power [6], for writer identification.

A set of words per writer is used for training. At the testing phase, we also provide a set of words of a particular unknown writer. Each word is tested by the classifier separately and the classifier output is combined using a *majority voting* scheme. The choice of the number of words in the training and test set depends on the user with respect to the available data and the tool (classifier) used. It may be of a pre-determined fixed size or the number of words present in a full handwritten page. The details of the training and test sets are discussed in Section III-B2.

1) Hand-crafted features with SVM classifier: The traditional hand-crafted features are not derived automatically. In the literature, it has been seen that among such hand-crafted features, the contour-based features [8], [9], [12] work quite well for writer identification. For this reason, we choose the "contour-hinge" feature, proposed by Bulacu and Schomaker [12]. This feature tracks the orientation and curvature of the ink-strokes scribed by individual writers. In [12], the authors used the number of histogram bins $n_b = 12$ in a 360^o orientation span and obtained $n_b(2n_b + 1) = 300$ dimensional feature vector. We set $n_b = 16$, leading to a feature vector of dimension 528.

This feature vector is sent to an SVM classifier with an RBF kernel for classification (writer identification). We have discussed the SVM-RBF classifier in Section II-A2. The hyper-parameter tuning and classifier training details are described in Section III-B2.

2) CNN extracted features with Recurrent Neural Network: We extract the word level features automatically derived from the CNN as described in Section II-A1.

This feature vector of dimension 480 is provided as an input in a bi-directional *Recurrent Neural Network* (RNN) [27]. The number of nodes in the RNN input layer is the dimension of the feature vector, i.e. 480. The number of individual writer classes is the number of nodes of the output layer. One ϵ node at the output layer is extra and remains as null. We use two distinct hidden layers for forward and backward sequences, separately. Here, LSTM (*Long Short-Term Memory*) [28] blocks are used as hidden units. These two hidden layers contain 256 and 128 LSTM memory cells, respectively. This BLSTM recurrent net prevents the occurrence of so-called "vanishing gradient problem" [29]. All the meta-parameters are tuned and optimized using a tuning set, discussed in Section III-B2.

III. Experimental Results and Discussion

In this section, we present the experimental results and performance of our system. At first, we discuss the database employed for our experiments.

A. Database Generation

For experimental analysis, we required a database of handwritten documents containing struck-out texts. Among the publicly available databases, only the dataset used in [18] (henceforth called *NewISIdb: SoT*) contained a fair amount of such texts. However, for this study, we needed a reasonable amount of handwritten specimens of a particular writer for training and testing. The *NewISIdb: SoT* did not contain multiple copies of handwriting of each individual and sometimes the writer information was absent due to the difference in motivation of their work. Therefore, it was necessary to generate a new database containing multiple copies of handwriting.

We collected English handwriting from 100 writers and Bengali handwriting from 100 individuals. Those writers were of both genders in the age group of 10 to 66 years with various academic backgrounds (from elementary school to university level). Each writer provided 2 full pages of handwriting samples. The handwritten content was independently chosen by the writer from any piece of article. The writers were requested to strike-out some words in their running handwriting styles. It was guaranteed that the handwritten pages contained some struck-out texts. We supplied $A4$ sized 75 GSM (g/m^2) blank white pages (without any ruling-lines) and a 0.5 mm ball-point black-ink pen of the same brand/model in order to maintain the consistency with respect to color and stroke-width.

Our database contained a total of 200 English and 200 Bengali handwritten pages. Here a handwritten page contained approximately 16 text-lines and a text-line contained about 10 words. Therefore, a total of almost 3200 (= 200 × 16) text-lines and 32000 words were available for both English and Bengali scripts. For each writer, about 320 handwritten word samples were present. Precisely, the total numbers of struck-out words and normal words were 2152 (2317) and 30045 (29341), respectively, for the English (Bengali) script in our database. Here, the *strike-out rates*[1] [18] were 6.68% for English and 7.31% for Bengali script.

B. Results and Evaluation

In this section, we present the experimental results and performance of our method for struck-out text detection and their effect on writer identification.

1) Normal and struck-out text separation using CNN-SVM: This step was like a pre-processing stage of our main task of analyzing the impression of struck-out text on writer identification. However, we also wanted to evaluate the performance of our CNN-SVM hybrid classifier for such normal versus struck-out class detection.

Besides our generated database (Section III-A), we also tested our method on the *NewISIdb: SoT* database [18]. The *NewISIdb: SoT* contained a total of 1395 (1432) struck-out and 30670 (32762) normal English (Bengali) words.

For training purposes, we used 50% data of our generated database and 20% of the *NewISIdb: SoT*. For the CNNs, the training epochs were increased up to 500. The CNN was trained with stochastic gradient descent. We employed a learning rate of 10^{-3} with a momentum term of 0.9 for this CNN. The SVM-RBF hyperparameters (γ and $\mathcal{C}$) were also tuned by this training set. The grid searching range for γ and $\mathcal{C}$ were $[2^3, 2^2, \ldots, 2^{-7}]$ and $[2^7, 2^6, \ldots, 2^{-4}]$, respectively. The best performance was obtained for $\gamma = 2^{-3}$ and $\mathcal{C} = 2^6$. Here we used 5-fold cross-validation.

The performance for struck-out text detection is calculated in terms of *Precision*, *Recall* and *F-Measure*.
True Positive (TP) := # genuine struck-out words detected by our system,
False Negative (FN) := # genuine struck-out words incorrectly recognized as normal words,
False Positive (FP) := # normal words detected as struck-out words,
Precision (P) = $TP/(TP + FP)$,
Recall (R) = $TP/(TP + FN)$,
F-Measure (FM) = $(2 \times P \times R)/(P + R)$.

The quantitative performance is shown in TABLE I.

TABLE I
STRUCK-OUT TEXT DETECTION PERFORMANCE

Script	English (Bengali)		
Database	*Precision* %	*Recall* %	*F-Measure* %
Generated	98.45 (98.16)	98.93 (98.67)	98.69 (98.41)
NewISIdb: SoT [18]	98.63 (98.25)	99.08 (98.84)	98.85 (98.54)

For English (Bengali) struck-out text detection, we obtained 98.69% (98.41%) and 98.85% (98.54%) of *F-Measure* on our generated database and *NewISIdb: SoT*, respectively.

2) Struck-out text impression on writer identification: As stated earlier, our writer identification method was tested on three categories of data: (i) full handwritten page (normal + struck-out text), (ii) normal texts only, (iii) struck-out texts only.

For writer identification, we employed the Top-N criterion, where the possible writer was a member of a reduced set of 'N' ($\ll$ total number of writers) individuals. Here, we chose Top-1, Top-2 and Top-5 criteria and provided the performance in terms of *F-Measure*.

a) Writer identification using hand-crafted feature-based SVM: In our generated database, each writer wrote two full pages with some struck-out texts in their natural handwriting. We used one page for training and the other page for testing. We did not use any fixed size word vocabulary for training, since we intended to make the system quite independent of word segmentation. Therefore, if the words were not properly segmented, then our method would still perform well. Here, one page having approximately 160 words was used for training, per writer. The hyper-parameters (γ and $\mathcal{C}$) for SVM-RBF were tuned using this training set. The best performance was obtained for $\gamma = 2^{-4}$ within the range $[2^3, 2^2, \ldots, 2^{-8}]$ and $\mathcal{C} = 2^7$ within the range $[2^8, 2^7, \ldots, 2^{-2}]$. Here 5-fold cross-validation was used.

The writer identification performance employing a hand-crafted feature-based SVM in terms of *F-Measure* (*FM*) is presented in TABLE II.

In TABLE II, the Top-1 criterion produces an overall 74.89% (74.15%) *FM* on full page English (Bengali) writing, while it is 80.32% (79.07%) when tested on the normal text. It can also be seen that removal of struck-out text increases the performance of writer identification. Top-1 performance has increased by 5.43% for English and 4.92% for Bengali script in terms of *F-Measure* (*FM*).

[1] $strike\text{–}out\ rate = \frac{\#struck\text{–}out\ words}{\#total\ words}$.

TABLE II
WRITER IDENTIFICATION PERFORMANCE USING SVM

Script	English (Bengali)		
Handwritten	*F-Measure %*		
Text	*Top-1*	*Top-2*	*Top-5*
Normal + Struck-out	74.89 (74.15)	75.95 (75.03)	77.28 (76.64)
Normal	80.32 (79.07)	82.53 (81.83)	85.26 (84.92)
Struck-out	26.43 (24.94)	26.91 (25.78)	27.39 (26.92)

b) Writer identification using auto-derived CNN feature-based RNN: Here we also used one page per writer for training as in the procedure from Section III-B2a. We employed a learning rate of 10^{-3} with a momentum term of 0.9 for this neural net. The training epochs were increased up to 500.

In TABLE III, we present the writer identification performance in terms of *F-Measure* using an automatically extracted CNN feature-based recurrent net.

TABLE III
WRITER IDENTIFICATION PERFORMANCE USING CNN-RNN

Script	English (Bengali)		
Handwritten	*F-Measure %*		
Text	*Top-1*	*Top-2*	*Top-5*
Normal + Struck-out	89.75 (88.89)	90.16 (89.62)	91.75 (91.10)
Normal	92.83 (92.12)	93.75 (93.28)	95.26 (94.87)
Struck-out	30.65 (29.43)	31.73 (30.56)	32.85 (31.96)

TABLE III shows that our writer identification method performs 3.08% (3.23%) better with respect to the Top-1 *F-Measure* on normal English (Bengali) text than the mixed-up text.

In Fig. 4, we present the writer identification performance in the form of a bar chart.

The writer identification performance on struck-out text alone is very poor. The possible reasons for such poor performance are as follows:

(i) The dataset contains a very small amount of struck-out texts per writer. Since our dataset is quite realistic and contains natural running-form of handwriting, the struck-out rate becomes 6.68% (7.31%) for English (Bengali) pages. Therefore, the reduced amount of data for training struck-out text may have reflected the performance while identifying writers on struck-out texts only.

(ii) Sometimes, the strike-out strokes damage the writing strokes heavily. Therefore, the discriminative power of writing strokes diminishes and performs poorly for writer identification.

Our method confirms that struck-out text impedes the writer identification performance, while the normal text plays a more important role in this task.

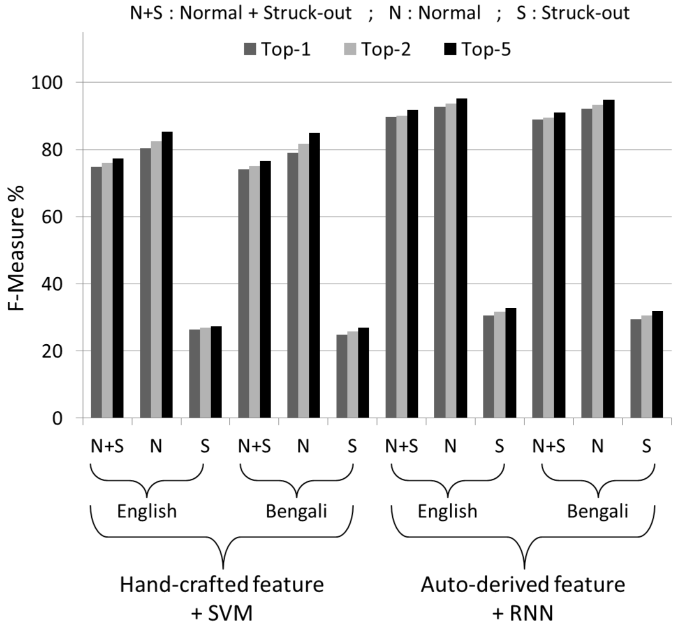

Fig. 4. Bar chart of writer identification performance.

C. Comparison with Other Works

Among previous works for struck-out text detection, the method of [14] did not provide any solution. Tuganbaev and Deriaguine [15] dealt with struck-out (or, “stricken-out”) characters only. Being a US patent [15], proper technical detail for implementation was missing there. In [16], the straight and wavy strokes used to strike-out were machine-generated and superimposed on offline words. In our case, the strike-out strokes were generated by humans at the time of writing. Only the methods of [17], [18] were related to our task of struck-out word detection. The system of [17] worked with only straight-line type strike-out strokes, while the updated version [18] worked with various forms of strike-outs. Therefore, we compare our struck-out text detection method with the scheme of [18] on the same database (*NewISIdb: SoT*) as employed in [18]. This quantitative comparison in terms of *Precision*, *Recall* and *F-Measure* (*FM*) is presented in TABLE IV. On the *NewISIdb: SoT*, for struck-out text detection, our method acquires 98.85% (98.54%) of *FM* on English (Bengali), whereas the method of [18] produces an *FM* of 91.56% (91.06%).

TABLE IV
COMPARISON OF STRUCK-OUT TEXT DETECTION

Script	English (Bengali)		
Method	*Precision %*	*Recall %*	*F-Measure %*
Method of [18]	90.94 (90.19)	92.18 (91.94)	91.56 (91.06)
Proposed method	98.63 (98.25)	99.08 (98.84)	98.85 (98.54)

We found that the work of Brink et al. [30] was related to our task. To separate the struck-out/“crossed-out” text, they employed a *decision tree* classifier using two simple hand-crafted features based on crossing counts of writing strokes (*"branching"*) and ink-pixel counting (*"size"*). Their method removed 47.5% of struck-out text retaining 99.1% of the normal text, while tested on a real forensic dataset of the *Netherlands Forensic Institute* (NFI). We contacted the group leader of [30] who informed that the NFI biometric data were taken from real criminal suspects and it could not be shared with us. Therefore, we were not able to compare our results on the NFI dataset. However, our method detected 98.69% (98.41%) and 98.85% (98.54%) struck-out English (Bengali) text in terms of F-Measure while testing on our generated database and *NewISIdb: SoT*, respectively. After removal of struck-out text, the writer identification performance of [30] deteriorated by about 1%, while in our case, this performance improved by approximately 3%–5%. According to the discussion and results of Brink et al. [30], their method removed some good normal texts as struck-out/"crossed-out", whereas our method was very precise about normal and struck-out text separation.

A slightly related work in [31] discussed the effect of ruling-line deletion on writer identification. They reported that by retaining rather than by deleting the ruling-lines increased the writer identification performance. However, the strike-out strokes were hand-drawn irregular patterns (straight horizontal, slanted, crossed, wavy, zigzag etc.), whereas ruling-lines were of a regular pattern, i.e. underline-like, pre-printed straight-lines. The ruling-line affected most of the words of a page, while the struck-out rate of our database was 6.68% (7.31%) for English (Bengali). Therefore, retaining the strike-out strokes, like retaining the ruling-line, had no positive impact on writer identification. On the other hand, deleting the strike-out stroke and modifying the writing stroke using the method of [18] were computationally costly, and also were not so effective on the writer identification performance.

IV. Conclusion

In this paper, we investigate the effect of struck-out texts on the automated writer identification process. We show that the presence of struck-out texts impedes writer identification performance. A CNN-SVM hybrid model is used to detect struck-out texts. We employ both hand-crafted and auto-derived features for writer identification. The hand-crafted features are fed into an SVM classifier and the auto-derived features are supplied to an RNN model. We generate a handwritten document database containing some struck-out texts from 100 English and 100 Bengali writers, which will be available for academic research purposes by receiving an e-mail request. We have obtained a 98.85% (98.54%) of *F-Measure* for struck-out text detection on English (Bengali) handwriting while testing on our generated database. Here, for English (Bengali) handwriting, the presence of struck-out texts degrades the writer identification *F-Measure* by 5.43% (4.92%) and 3.08% (3.23%) while employing the hand-crafted and auto-derived features, respectively.

Although the struck-out text removal and processing only on normal texts improves the writer identification a little (about 3%–5% *F-Measure*), our method analyzes that the same features used for normal texts and struck-out texts drop the performance. For a page containing more struck-out texts (with a higher strike-out rate) and less normal texts, the writer identification performance on normal texts becomes lower. Therefore, for such cases, the general writer identification technique does not work well. However, auto-derived features may perform better for struck-out texts on the availability of more struck-out training data. Our next endeavor will be to analyze such cases where a page will contain approximately 15%–20% of struck-out texts/writing errors. Therefore, our future plan is to collect more struck-out data and perform the experiments again. Moreover, we believe that a writer retains his/her latent identity in strike-out strokes. We will also try to explore this in the near future.

Acknowledgment

We are very thankful to the volunteers for their help in generating the database. Support by SERB and INAE, New Delhi, to one co-author (B. B. Chaudhuri) is also gratefully acknowledged.

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] Z. He, X. You, Y.Y. Tang, "Writer Identification of Chinese Handwriting Documents using Hidden Markov Tree Model", Pattern Recognition, vol.41, no.4, pp.1295-1307, 2008.

[4] 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.

[5] Z.A. Daniels, H.S. Baird, "Discriminating Features for Writer Identification", Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), pp.1385-1389, 2013.

[6] C.I. Tomai, B. Zhang, S.N. Srihari, "Discriminatory Power of Handwritten Words for Writer Recognition", Proc. Int. Conf. on Pattern Recognition (ICPR), vol.2, pp. 638-641, 2004.

[7] B. Zhang, S.N. Srihari, S. Lee, "Individuality of Handwritten Characters", Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), pp.1086-1090, 2003.

[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] 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.

[11] E.N. Zois, V. Anastassopoulos, "Morphological Waveform Coding for Writer Identification", Pattern Recognition, vol.33, no.3, pp.385-398, 2000.

[12] 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.

[13] S.N. Srihari, S.-H. Cha, H. Arora, S. Lee, "Individuality of Handwriting", Journal of Forensic Sciences, vol.47, no.4, pp.1-17, 2002.

[14] J. Arlandis, J.C.P.-Cortes, J. Cano, "Rejection Strategies and Confidence Measures for a k-NN Classifier in an OCR Task", Proc. Int. Conf. on Pattern Recognition (ICPR), vol.1, pp. 576-579, 2002.

[15] D. Tuganbaev, D. Deriaguine, "Method of Stricken-out Character Recognition in Handwritten Text", Patent: US 8472719 B2, 2013.

[16] L.L.-Sulem, A. Vinciarelli, “HMM-based Offline Recognition of Handwritten Words Crossed Out with Different Kinds of Strokes”, Proc. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pp.70-75, 2008.

[17] C. Adak, B.B. Chaudhuri, “An Approach of Strike-through Text Identification from Handwritten Documents”, Proc. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pp. 643-648, 2014..

[18] B.B. Chaudhuri, C. Adak, “An Approach for Detecting and Cleaning of Struck-out Handwritten Text”, Pattern Recognition, vol.61, pp.282-294, January 2017.

[19] F. Zhao, H.Z. Lu, Z.Y. Zhang, “Real-time Single-pass Connected Components Analysis Algorithm”, EURASIP Journal on Image and Video Processing-2013:21, pp.1-10, 2013.

[20] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal A. Alaei, “ICDAR 2013 Handwriting Segmentation Contest”, Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), pp.1402-1406, 2013.

[21] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview”, Neural Networks, vol.61, pp.85-117, 2015.

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based Learning Applied to Document Recognition”, Proceedings of the IEEE, vol.86, no.11, pp.2278-2324, 1998.

[23] X.-X. Niu, C.Y. Suen, “A Novel Hybrid CNN-SVM Classifier for Recognizing Handwritten Digits”, Pattern Recognition, vol.45, no.4, pp.1318-1325, 2012.

[24] V.N. Vapnik, “The Nature of Statistical Learning Theory”, ISBN: 0-387-94559-8, Springer-Verlag, 1995.

[25] C.L. Liu, C.Y. Suen, “A New Benchmark on the Recognition of Handwritten Bangla and Farsi Numeral Characters”, Pattern Recognition, vol.42, pp.3287-3295, 2009.

[26] K. Duan, S.S. Keerthi, A.N. Poo, “Evaluation of Simple Performance Measures for Tuning SVM Hyperparameters”, Neurocomputing, vol.51, pp.41-59, 2003.

[27] M. Schuster, K.K. Paliwal, “Bidirectional Recurrent Neural Networks”, IEEE Transactions on Signal Processing, vol.45, no.11, pp.2673-2681, 1997.

[28] S. Hochreiter, J. Schmidhuber, “Long Short-Term Memory”, Neural Computation, vol.9, no.8, pp.1735-1780, 1997.

[29] A. Graves, M. Liwicki, S. Fernndez, R. Bertolami, H. Bunke, J. Schmidhuber, “A Novel Connectionist System for Unconstrained Handwriting Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.31, no.5, pp.855-868, 2009.

[30] A. Brink, H. van der Klauw, L. Schomaker, “Automatic removal of crossed-out handwritten text and the effect on writer verification and identification”, Proc. Document Recognition and Retrieval (DRR) XV, #68150A, 2008.

[31] J. Chen, D. Lopresti, G. Nagy, “Conservative Preprocessing of Document Images”, International Journal on Document Analysis and Recognition (IJDAR), vol.19, no.4, pp.321-333, 2016.