Offline Bengali writer verification by PDF-CNN and siamese net

Offline Bengali Writer Verification by *PDF-CNN* and Siamese Net

Chandranath Adak*†, Simone Marinai‡, Bidyut B. Chaudhuri§, Michael Blumenstein†
**IIIS, School of ICT, Griffith University, Australia-4222*
†*CAI, School of Software, University of Technology Sydney, Australia-2007*
‡*DINFO, University of Florence, Italy-50139*
§*CVPR Unit, Indian Statistical Institute, India-700108*
adak32@gmail.com

***Abstract*—Automated handwriting analysis is a popular area of research owing to the variation of writing patterns. In this research area, writer verification is one of the most challenging branches, having direct impact on biometrics and forensics. In this paper, we deal with offline writer verification on complex handwriting patterns. Therefore, we choose a relatively complex script, i.e., Indic Abugida script Bengali (or, Bangla) containing more than 250 compound characters. From a handwritten sample, the probability distribution functions (PDFs) of some handcrafted features are obtained and input to a convolutional neural network (CNN). For such a CNN architecture, we coin the term "*PDF-CNN*", where handcrafted feature PDFs are hybridized with auto-derived CNN features. Such hybrid features are then fed into a Siamese neural network for writer verification. The experiments are performed on a Bengali offline handwritten dataset of 100 writers. Our system achieves encouraging results, which sometimes exceed the results of state-of-the-art techniques on writer verification.**

***Keywords*-CNN; Handwriting; Siamese neural network; Writer Verification.**

I. Introduction

Handwriting is one of the most powerful behavioral biometrics [1] which stores human individuality in forms of inter-variable writing patterns. Handwriting analysis is also commonly accepted in forensics to authenticate a person. Automated writer authentication has been established within a computerized civic society [2]. This field of research is still challenging due to ever-increasing and complex patterns. As a matter of fact, the handwritten samples of a person vary extensively with respect to the circumstances [3]. This makes the task even more challenging.

The handwriting can be captured in both offline and online modes [4]. For the capture of offline data, the writing is performed on a piece of paper by pen/pencil and the handwritten page is then scanned to obtain a digital image. Online data are captured by direct writing on a smart tablet, or by using a digital pen. The online data often have some advantages for their processing because they bring additional information such as the temporal writing sequence of strokes and the pen pressure.

A comprehensive survey related to handwriting authentication up to the year 1989 is reported in [2]. Recent advancements of writer identification and verification are discussed in [5], [6]. In this paper, we deal with the investigation of offline handwriting. A detailed survey dedicated to offline techniques is reported by Xiong et al. in [7].

In this paper, we focus on the handwriting of a complex Indic script, *Bengali* (endonym, *Bangla*). The Bengali script contains 50 basic characters and more than 250 compound characters [8]. A compound character is the conjunction of two or more (up to about five) basic characters. A basic character can be conjoined with another in four possible major neighborhood directions (top, bottom, left, and right), according to Bengali orthographic rules. After the conjunction, the shapes of the characters may change completely, or partially. Also, a particular character may be written in multiple ways (refer to Fig. 1). Moreover, the individual characters of a word are mostly joined by a headline-like stroke called "*matra*". Besides the above, the cursiveness of Bengali writing makes it even more complicated [9]. Some more complex characteristics of the Bengali script can be found in [8], [10].

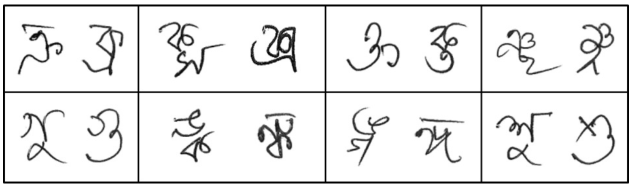

Figure 1. Every pair in a box comprises the same compound Bengali character written in two different styles [10].

A brief discussion on offline Indic as well as Bengali writer identification/verification is reported in [10]. The earliest attempt on Bengali writer recognition was performed by Tripathi et al. [11] using some simple word bounding-box level micro-features. In [12], a 2-D autoregressive (AR) model was used for Bengali and French writer identification. A directional and gradient feature-based SVM model was used by Chanda et al. [13] to identify Bengali writers. In [14], Biswas and Das used the occurrence of text component structure and a rejection schema to investigate Bengali writers. A Radon transform projection-based improved version of [14] was reported in [15]. In [10], some textural features were used to identify writers from isolated Bengali characters and numerals.

In one common approach, the writer verification task is addressed as a "one-to-one comparison" to decide whether two samples are written by the same writer [6]. In other words, the writer verification can be perceived as a binary classification problem to answer *yes/no* to the question whether Doc-A is written by Writer-B [3].

In this work, we perform this classification (for writer verification) on the basis of the *Probability Distribution Function* (PDF) of some handcrafted features. We intend to hybridize handcrafted features with auto-derived features to investigate their impact on verification performance. To this purpose, we feed the PDF into a *Convolutional Neural Network* (CNN) to produce auto-derived CNN features from the PDF information. Then, the CNN features are fed into two different classification modules: one MLP (*Multi-Layer Perceptron*) and one Siamese neural network for writer verification.

In *Section* II, we describe our proposed method in detail. Then, the experimental results are presented in *Section* III. Finally, *Section* IV concludes the paper.

II. Proposed Method

Once a digital image of a handwritten page is obtained, it undergoes a number of preprocessing steps. The scanned digital image of a handwritten page sample may be speckled with some noise. In the preprocessing stage, very small components are removed by a moderately fast single pass connected component labeling method [16]. If the document contains any doodle/drawing-like non-textual component, this non-textual component is removed using the technique presented in [17]. Some struck-out/crossed-out texts may exist in the document that also need to be removed due to their impeding effect on the writer analysis task [18]. We use the method in [19] to remove struck-out/crossed-out texts from the document. Subsequently, the preprocessed digital image of a handwritten page is finally used for writer verification.

As mentioned in *Section* I, the writer verification task can be expressed as a binary classification problem. To deal with the classification problem of writer verification, at first, we generate some handcrafted [20] features from the handwritten pattern. In the next subsections, we discuss the details of the features used in this paper and Subsequently describe the classification methods.

A. Handcrafted Feature PDF

Instead of employing the traditional single-valued feature to capture the facet of writing individuality, here we employ a probability distribution function (PDF) as an entire feature vector [6]. Two different types of handcrafted features namely *textural* and *allographic* are used and described in the following.

1) Textural feature PDF: The textural feature usually provides the information of pen-grip holding style and writing slant. It has been found experimentally that among the textural features such as directional, run-length, hinge, gray-level co-occurrence, the contour-hinge feature works well for writer verification [6]. We, therefore, focus on the contour-hinge feature (f_{ch}) here, as described below.

A hinge is created when two contour fragments are joined to a common junction-pixel [6]. Consider two contour fragments making angles ϕ_1 and ϕ_2 respectively with the horizontal axis. To avoid redundancy, we assume $\phi_2 \geq \phi_1$. The angles are spanned to all four quadrants (360^o) centering the junction-pixel. The entire quadrant is divided into $2n$ parts, where n is a given parameter. Based on this division, a normalized histogram is generated with a joint probability distribution $p_f(\phi_1, \phi_2)$, as demonstrated in [6]. Although the total number of combinations of two angles is $4n^2 = 2n \times 2n$, the final number of combinations is reduced to ${}^{2n}C_2 + 2n = n(2n+1)$, due to the assumption of non-redundancy. Here, ${}^{2n}C_2$ refers to the number of ways we can choose 2 elements out of $2n$ elements and the greater value is always assigned to ϕ_2 due to our assumption; $2n$ refers to the case when ϕ_1 and ϕ_2 lie in the same partition of the quadrant. In our experiment, empirically we choose $n = 12$, leading to $12(2 \times 12 + 1) = 300$ combinations. Though we have used the matrix of size 24×24 in our experiment, we consider only the non-redundant entries ($\phi_2 \geq \phi_1$). In other words, the employed feature map is in the form of a triangular matrix [6].

2) Allographic feature PDF: The allographic feature divulges the character shape and formation [6]. It is assumed that a writer acts as a stochastic generator of grapheme/fragment (small parts of character) shapes. The probability distribution of such grapheme shapes may be used as a feature. In this work, the handwritten text is segmented into graphemes/fragments by means of the water reservoir technique [21] and by partitioning the ink-trace at the minima of the lower contour vertically thorough the ink-stroke-width of the upper contour [6].

The grapheme codebook generation is obtained by using Kohonen's 2-Dimensional *Self-Organizing Map* (SOM) [22]. Here, we use a 25×25 SOM-2D map. Some alternatives for codebook generation are k-means and one-dimensional SOM. However, we choose the SOM-2D because of the spatial organization of codebooks that has been used in some document image analysis tasks [23], [24] and that can be taken into account by local receptive fields in convolutional layers of the CNN architecture.

The feature vector in this case is a probability distribution function of graphemes organized into the 2D topology of the SOM. A histogram of 625 ($= 25 \times 25$) bins is therefore generated considering every grapheme in one bin. A sample grapheme is matched with the nearest codebook prototype and is put in a histogram bin, similarly to [6]. From this histogram, we obtain a normalized feature distribution map of size 25×25.

B. Handcrafted Feature PDF Hybridized with Auto-derived CNN Features (PDF-CNN)

In a CNN architecture, generally, an image is used as an input to extract some auto-derived features. Here, the handcrafted feature PDF is fed into the CNN, since it is still one 2D matrix of non-negative values. We call this CNN architecture as a *PDF-CNN.* Fig. 3 shows the model diagram of a PDF-CNN.

The PDF-CNNs used for writer verification are discussed below.

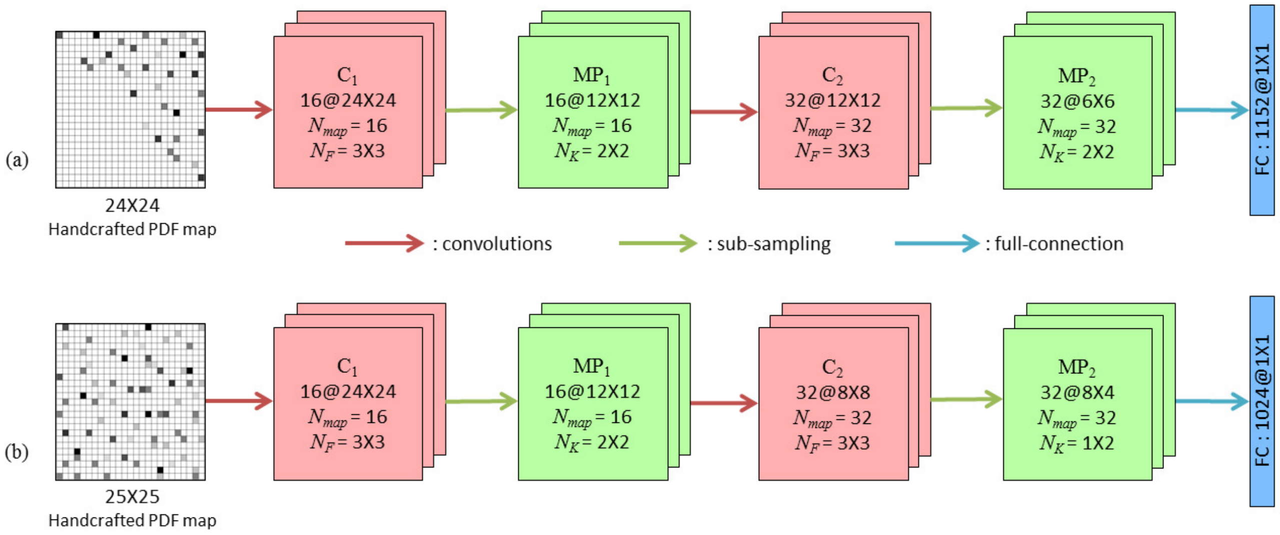

Figure 2. (a) Textural PDF-CNN and (b) Allographic PDF-CNN architectures.

1) Textural PDF-CNN: The above textural contour-hinge PDF, as demonstrated in *Section* II-A1, is the input to the CNN for deriving an automated feature vector. As discussed earlier, our square textural hinge map is of size 24×24, and half of the matrix (below its main diagonal) is left unused to avoid redundancy.

The CNN architecture contains two convolutional layers, each followed by a sub-sampling layer. The overall architecture is shown in Fig. 2(a).

The convolutional layer C_1 has 16 feature maps of size 24×24. Each feature map is connected to a 3×3 (N_F) neighborhood in the input textural PDF map. The MP_1 subsampling (max-pooling) layer contains 16 feature maps of size 12×12, each connected to a 2×2 (N_K) neighborhood in the corresponding feature map in C_1.

The following C_2 convolutional layer consists of 32 feature maps of size 12×12. Here, $N_F = 3 \times 3$, therefore, each feature map is connected to a 3×3 neighborhood of corresponding MP_1's feature map. The MP_2 sub-sampling (max-pooling) layer contains 32 feature maps of size 6×6. Here, $N_K = 2 \times 2$.

The following layer FC has 1152 feature maps each sized 1×1 and fully connected with MP_2.

The ReLU (*Rectified Linear Unit*) is used as the activation function for all the convolutional layers. A learning rate of 10^{-3} with a momentum term of 0.9 is used for training up to 500 epochs. We have obtained an 1152-dimensional feature vector from this textural PDF-CNN.

2) Allographic PDF-CNN: The handcrafted allographic PDF feature map of size 25×25 (refer to *Section* II-A2) is fed to the CNN for automated feature extraction.

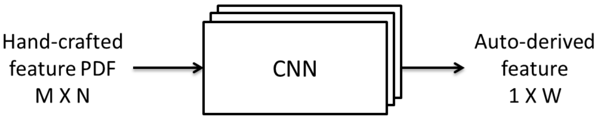

Figure 3. PDF-CNN model diagram.

This CNN architecture is also constructed with two convolutional layers, each followed by a sub-sampling layer, as shown in Fig. 2(b). The first two layers (C_1 and MP_1) of CNN architecture are analogous to the Textural PDF-CNN (refer to *Section* II-B1).

The following C_2 convolutional layer consists of 32 feature maps of size 8×8 and again $N_F = 3 \times 3$. The MP_2 sub-sampling (max-pooling) layer contains 32 feature maps of size 8×4 and now $N_K = 1 \times 2$ is used.

In this case, the following layer FC has 1024 feature maps, each of size 1×1 and fully connected with MP_2.

Also, for this CNN, the ReLU is used as activation function for the convolutional layers. The learning rate is set to 10^{-3} with a momentum of 0.9. The training epochs are increased up to 500. This allographic PDF-CNN produces a 1024-dimensional feature vector.

C. Classification / Writer Verification

The output of a PDF-CNN is fed into a classifier for writer verification. It is perceived as a binary classification task to classify/authenticate a writer in the "same" (*yes*) or "not same" (*no*) class. We choose MLP (*Multi-Layer Perceptron*) and Siamese neural network separately for this verification task.

1) CNN-MLP: The output of PDF-CNN, as discussed in *Section* II-B, is fed into a MLP classifier with one hidden layer and two neurons as MLP output. One different network is trained for each writer and the target output has two different outcomes for samples of the corresponding writer and for remaining texts.

The training is made with the scaled conjugate gradient backpropagation, since it requires less memory and the mean squared error is used as the performance parameter. The number of neurons in the hidden layer have been empirically set from 512 to 100 on the basis of preliminary experiments. The activation function for the MLP is the hyperbolic-tangent sigmoid. The training epochs are increased from 1,000 up to 10,000.

a) Textural CNN-MLP: As discussed in *Section* II-B1, the handcrafted textural feature PDF is fed to the CNN. From the textural PDF-CNN, we have obtained a 1152-dimensional feature vector. This feature vector is fed to the above MLP (refer to *Section* II-C1) for writer verification.

b) Allographic CNN-MLP: Similarly, the handcrafted allographic feature PDF is sent as input to the CNN to obtain a 1024-dimensional feature vector (refer to *Section* II-B2). This feature vector is input to the above MLP as discussed in *Section* II-C1 for writer verification.

2) CNN-Siamese: Siamese neural network is a well-known architecture for ranking the similarity between two inputs [25], [26] and for weakly supervised learning. Here, we use the Siamese net for writer verification to know whether a handwritten sample is written by a particular writer. The generated PDF-CNN, as discussed in *Section* II-B, is employed in the Siamese neural network.

The usual Siamese neural network contains two identical subnetworks. Our PDF-CNN depicts both subnetworks of the Siamese twins, individually. The Siamese twin subnets are joined with a loss function to calculate the similarity between the feature representations on each subnet. The contrastive loss function [27] is used here for training the whole network according to the similarity of the two handwritten samples.

a) Textural CNN-Siamese: The handcrafted textural feature PDF is fed into the CNN and the textural PDF-CNN is the architecture of both Siamese twins. The contrastive loss function is used here as mentioned earlier (refer to *Section* II-C2).

b) Allographic CNN-Siamese: The handcrafted allographic feature PDF is fed into the CNN (*Section* II-B2). This allographic PDF-CNN is one subnet of the Siamese twins, whereas the other one is exactly identical. Here, also, the contrastive loss function is employed as mentioned earlier (refer to *Section* II-C2).

III. Experimental Results

Before evaluating the performance of our system, we discuss the database used in our experiments.

A. Database Employed

We employed an in-house dataset of 100 writers being in the age group of 9 to 67 years old with different academic backgrounds from primary school to university level. Each writer produced 3 pages of Bengali handwriting. Therefore, a total of 300 document pages were stored in this database. The writers freely wrote a piece of Bengali text independent of any content. The dataset was made in an uncontrolled way. The writers were free to write anytime without any time-gap restriction between two writing samples. The A4 sized 75 GSM (g/m^2) blank white pages without any ruled-lines and a 0.5 mm ball-point black inked pen of the same model were provided to the writers for maintaining consistent writing strokes.

We split each page roughly into two parts, which is an approach commonly used in the literature [28] for dividing the samples among training, validation and test set. Since every person wrote 3 pages, 6 handwritten sub-samples per person were obtained.

B. Results and Evaluation

In this subsection, we present the experimental results and analyze the performance of our methodology.

We had 6 handwritten sub-samples for every writer. The training and validation sets contained 3 and 1 samples of each writer, respectively. The remaining portion of the dataset was used for testing. In other words, the training, validation and test sets were in the ratio of 3:1:2. All parameters of our system were tuned on the validation set.

The system performance was evaluated in terms of *False Acceptance Rate* (FAR) and *False Rejection Rate* (FRR) [29]. The *Balanced Error Rate* (BER) is the arithmetic mean of FAR and FRR. In other words, BER $=$ (FAR+FRR)$/2$. The *balanced accuracy* can be calculated as $(100-\text{BER})\%$.

The *Equal Error Rate* (EER) can also be obtained where the FAR is equal to FRR in the plot of error rate vs. decision threshold (T). In the case of the CNN-MLP, the threshold T is used to decide whether one unknown handwritten fragment belongs to the class for which the CNN-MLP network has been trained. In the case of the CNN-Siamese, the T value is used to decide whether two handwritten fragments belong to the same writer. We present these error rate curves with respect to different values of T in Fig. 4 - Fig. 7.

The system performances with respect to EER and balanced accuracy are shown in Table I.

Testing on our Bengali offline dataset, overall the CNN-Siamese performed better than the CNN-MLP. The textural feature worked better than the allographic ones. In particular, the textural PDF-CNN with Siamese net provided best EER of 2.36% on our test set. The lowest performance, obtained from allographic PDF-CNN with MLP, was even well, i.e. 4.07% of EER (see Table II).

It was interesting to notice that we did not use very deep neural architecture because of the small sized PDF feature

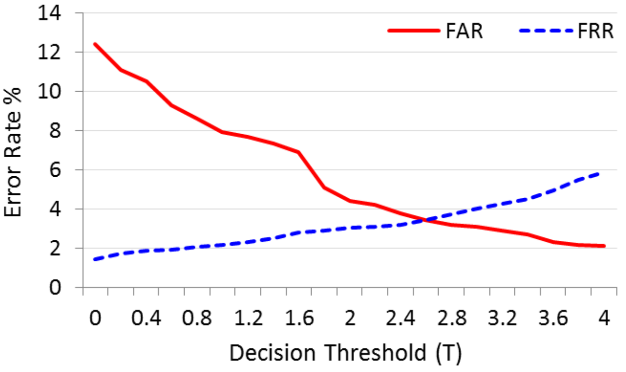

Figure 4. Textural CNN-MLP performance evaluation: FAR and FRR plot. EER = 3.42.

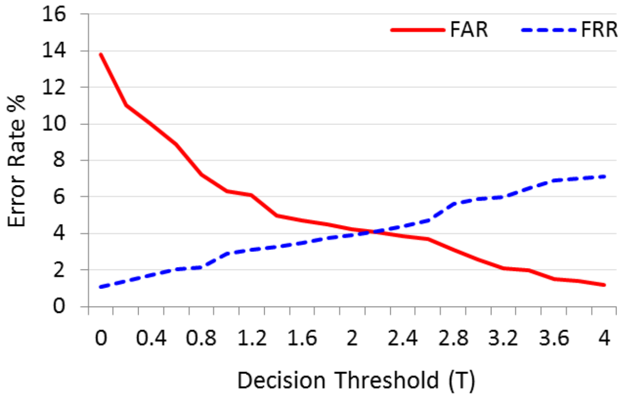

Figure 5. Allographic CNN-MLP performance evaluation: FAR and FRR plot. EER = 4.07.

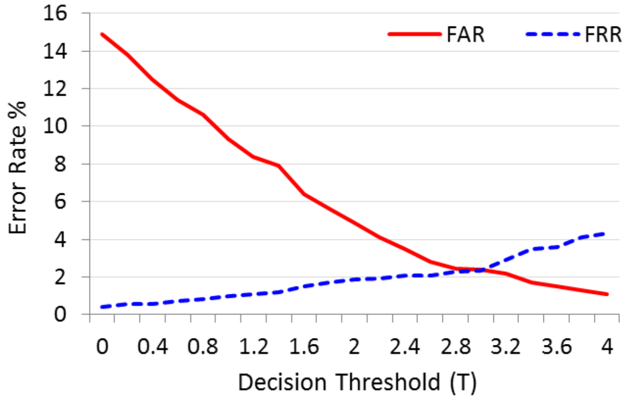

Figure 6. Textural CNN-Siamese performance evaluation: FAR and FRR plot. EER = 2.36.

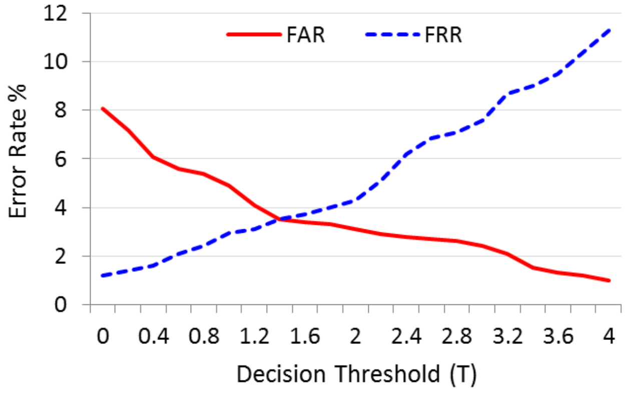

Figure 7. Allographic CNN-Siamese performance evaluation: FAR and FRR plot. EER = 3.51.

maps. However, our system performed satisfactorily with respect to the computation and space cost efficiency.

C. Comparative Analysis

To the best of our knowledge, the number of research papers on offline Bengali writer *identification* [10]-[15] is quite limited and we have not found any comparable work on Bengali writer verification.

Table I
WRITER VERIFICATION PERFORMANCE

Method	EER (%)	Balanced Accuracy (%)
Textural CNN-MLP	3.42	96.49
Allographic CNN-MLP	4.07	95.92
Textural CNN-Siamese	2.36	97.64
Allographic CNN-Siamese	3.51	96.47

The benchmark writer verification system like [6] produced 4.1% of the best EER on Firemaker-uppercase Dutch dataset using the textural contour-hinge feature. Bulacu et al. [6] also obtained 2.6% best EER on the Large-lowercase (English+Dutch) database by combining textural (contour-hinge, run-length) and allographic features. In [6], the system performance is reported on the Dutch and English script. In this paper, our emphasis is on Bengali script which is more complex owing to more than 250 compound characters. Here, we have obtained 2.36% of the best EER using a textural PDF-CNN with a Siamese net.

In [30], the overall error rate was 3.9% on the BFL (*Brazilian Forensic Letter*) database by employing texture-based features. Bensefia et al. [31] obtained 95.66% balanced accuracy on the English IAM database. In our current Bengali writer verification task, the lowest balanced accuracy is 95.92%, obtained using an allographic PDF-CNN with an MLP.

Pal et al. [32] obtained 33.82% of EER using uniform local binary pattern (LBP) on the Bengali part of the *BHSig260* signature verification dataset. Although we are not focusing on signatures, this *BHSig260* database can be treated as a word database containing mostly full signatures.In our case, we worked at the paragraph level writer verification and obtained the best result of 2.36% EER on our Bengali database.

To perform some additional comparisons, we executed some of these state-of-the-art methods [6], [30], [32] on our Bengali database (refer to *Section* III-A). Their performances in terms of EER and balanced accuracy are shown in Table II.

Table II
COMPARISON WITH SOME STATE-OF-THE-ART METHODS

Method	EER (%)	Balanced Accuracy (%)
Contour-hinge [6]	5.68	94.28
Texture [30]	6.39	93.60
LBP [32]	6.12	93.84
Ours inferior (Allographic CNN-MLP)	4.07	95.92

Here, our lowest performing method (Allographic CNN-MLP) even worked better than some earlier methods in the literature [6], [30], [32].

IV. CONCLUSIONS

In this paper, we propose a method for offline writer verification on a fairly complex script, Bengali. Here, we hybridize handcrafted feature (both textural and allographic)-

PDF with auto-derived CNN features. These hybrid features are finally fed into an MLP and a Siamese-twin net separately. The textural contour-hinge PDF induced CNN with a Siamese net (textural CNN-Siamese) has performed best by achieving an EER of 2.36% on our Bengali database. This database contains 300 Bengali pages written by 100 volunteers. The textural CNN-MLP, allographic CNN-MLP, and allographic CNN-Siamese have produced 3.42%, 4.07%, and 3.51% EER, respectively.

In future, we plan to undertake further experiments on the different datasets. We also plan to hybridize different neural networks for a comparative study.

ACKNOWLEDGMENT

We heartily thank all the volunteers for their contribution to our database generation. Support by the *IEEE Computational Intelligence Society: 2017 Graduate Student Research Grants*, to one co-author (C. Adak), is also gratefully acknowledged.

REFERENCES

[1] J. Chapran, “Biometric Writer Identification: Feature Analysis and Classification”, IJPRAI, vol. 20, no. 4, pp. 483-504, 2006.

[2] 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.

[3] C. Adak, B.B. Chaudhuri, M. Blumenstein, “Writer Identification and Verification from Intra-variable Individual Handwriting”, arXiv.org: CoRR abs/1708.03361, 2017.

[4] C. L. Liu et al., “CASIA Online and Offline Chinese Handwriting Databases”, Proc. ICDAR, pp. 37-41, 2011.

[5] L. Schomaker, “Advances in Writer Identification and Verification”, Proc. ICDAR, pp. 1268-1273, 2007.

[6] M. Bulacu, L. Schomaker, “Text-Independent Writer Identification and Verification Using Textural and Allographic Features”, IEEE T. PAMI, vol. 29, no. 4, pp. 701-717, 2007.

[7] Y.-J. Xiong et al., “Off-line Text-Independent Writer Recognition: A Survey”, IJPRAI, vol. 31, no. 5, 2017.

[8] B.B. Chaudhuri, U. Pal, “A Complete Printed Bangla OCR System”, Pattern Recognition, vol. 31:5, pp. 531-549, 1998.

[9] C. Adak, B.B. Chaudhuri, M. Blumenstein, “Offline Cursive Bengali Word Recognition using CNNs with a Recurrent Model”, Proc. ICFHR, pp. 429-434, 2016.

[10] C. Adak, B.B. Chaudhuri, “Writer Identification from Offline Isolated Bangla Characters and Numerals”, Proc. ICDAR, pp. 486-490, 2015.

[11] P. Tripathy et al., “Writer Recognition by Analyzing Word Level Features of Handwritten Documents”, Proc. ICAPR, World Scientific, pp. 73-78, 2007.

[12] U. Garain et al., “Off-Line Multi-Script Writer Identification Using AR Coefficients”, ICDAR, pp. 991-995, 2009.

[13] S. Chanda, K. Franke, U. Pal, T. Wakabayashi, “Text Independent Writer Identification for Bengali Script”, Proc. ICPR, pp. 2005-2008, 2010.

[14] S. Biswas, A.K. Das, “Content Independent Writer Identification Using Occurrences of Writing Styles for Bangla Handwritings”, Proc. NCVPRIPG, pp.154-157, 2011.

[15] S. Biswas, A.K. Das, “Writer Identification of Bangla Handwritings by Radon Transform Projection Profile”, Proc. DAS, pp.215-219, 2012.

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

[17] C. Adak, B.B. Chaudhuri, “Extraction of Doodles and Drawings from Manuscripts”, Proc. Int. Conf. on Pattern Recognition and Machine Intelligence (PReMI), LNCS #8251, pp. 515-520, 2013.

[18] C. Adak, B.B. Chaudhuri, M. Blumenstein, “Impact of Struck-out Text on Writer Identification”, Proc. IJCNN, pp. 1465-1471, 2017.

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

[20] L. Nanni, S. Ghidoni, S. Brahnam, “Handcrafted vs. Non-handcrafted Features for Computer Vision Classification”, Pattern Recognition, vol. 71, pp. 158-172, Nov. 2017.

[21] U. Pal, A. Belad, C. Choisy, “Touching Numeral Segmentation using Water Reservoir Concept”, Pattern Recognition Letters, vol. 24, no.(1-3), pp. 261-272, 2003.

[22] T. Kohonen, “Self-Organization and Associative Memory”, Springer Verlag, ISBN: 978-3-642-88163-3, 1989.

[23] S. Marinai, B. Miotti, G. Soda “Bag of Characters and SOM Clustering for Script Recognition and Writer Identification”, Proc. ICPR, pp. 2182-2185, 2010.

[24] S. Marinai, E. Marino, G. Soda, “Self-Organizing Maps for Clustering in Document Image Analysis”, Machine Learning in Document Analysis and Recognition, Springer, pp. 193-219, 2008.

[25] J. Bromley et al., “Signature Verification using A Siamese Time Delay Neural Network”, IJPRAI, vol. 07, #669, 1993.

[26] G. Koch, “Siamese Neural Networks for One-Shot Image Recognition”, M.S. Thesis, University of Toronto, 2015.

[27] S. Chopra, R. Hadsell, Y. LeCun, “Learning a Similarity Metric Discriminatively, with Application to Face Verification”, Proc. CVPR, pp. 539-546, 2005.

[28] X. Wu, Y. Tang, W. Bu, “Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform”, IEEE TIFS, vol. 9, no. 3, pp. 526-536, 2014.

[29] J. Neyman, E. Pearson, “On the Problem of the Most Efficient Tests of Statistical Hypotheses,” Phil. Trans. of the Royal Society of London, vol. A:231, pp. 289-337, 1933.

[30] R.K. Hanusiak, L.S. Oliveira, E. Justino, R. Sabourin, “Writer Verification using Texture-based Features”, IJDAR, vol. 15, no. 3, pp. 213-226, 2012.

[31] A. Bensefia, T. Paquet, L. Heutte, “A Writer Identification and Verification System”, Pattern Recognition Letters, vol. 26, no. 13, pp. 2080-2092, 2005.

[32] S. Pal et al., “Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large Indic-Script Signature Dataset”, Proc. DAS, pp. 72-77, 2016.