IEEE Transactions on Cybernetics

Handwritten Numeral Recognition integrating Start-End Points Measure with Convolutional Neural Network

Journal:	*IEEE Transactions on Cybernetics*
Manuscript ID	CYB-E-2020-04-0994
Manuscript Type:	Regular Paper
Date Submitted by the Author:	29-Apr-2020
Complete List of Authors:	Akhand, M. A. H. ; Khulna University of Engineering and Technology, Dept. of Computer Science and Engineering Rahat-Uz-Zaman, Md. ; Khulna University of Engineering and Technology, Dept. of Computer Science and Engineering Hye, Shadmaan ; Khulna University of Engineering and Technology, Dept. of Computer Sceince and Engineering Siddique, Nazmul; University of Ulster, School of Computing, Engineering and Intelligent Systems
Keywords:	Classification, Convolutional Neural Network, Handwritten Numeral Recognition, Start-End Writing Measure

Handwritten Numeral Recognition integrating Start-End Points Measure with Convolutional Neural Network

M. A. H. Akhand, *Member, IEEE,* Md. Rahat-Uz-Zaman, Shadmaan Hye, and N. Siddique, *Senior Member, IEEE*

Abstract—**Convolutional neural network (CNN) based methods have been very successful for handwritten numeral recognition (HNR) applications. However, CNN seems to misclassify similar shaped numerals (i.e., the silhouette of the numerals that look same). This paper presents an enhanced HNR system to improve the classification accuracy of the similar shaped hand written numerals incorporating the terminals points with CNN's recognition. In hand written numerals, the terminal points (i.e. the start and end positions) are considered as additional property to discriminate between similar shaped numerals. Start-End Writing Measure (SEWM) and its integration with CNN is the main contribution of this research. There are three major functional steps in the proposed SEWM-CNN: classification of a numeral image using a standard CNN; identification of start and end writing points from the silhouette of the numerals; and finally, system output integrating SEWM with the CNN decision. The proposed method is tested on rich benchmark numeral datasets of Bengali and Devanagari numerals. SEWM-CNN reveals itself as a suitable HNR method for Bengali and Devanagari numerals while compared with other existing methods.**

Index Terms— **Classification, Convolutional Neural Network, Handwritten Numeral Recognition, Start-End Writing Measure.**

I. Introduction

NUMERALS are an integral part of any language and play an important role in every day's life. At present, numerals are used in both printed and handwritten forms in everyday life and business. Handwritten numerals are used in postal code, bank cheque and in many other business [1]. Automatic recognition of handwritten numerals can release off tedious job at banks, post offices and registration departments. Therefore, handwritten numeral recognition (HNR) becomes an important research topic. HNR has significant challenges than character or word recognition. In case of character or word recognition, recognition errors made by a system can be verified using rules of grammar. But such error detection rule cannot be applied to numeral recognition since any numeral combination is technically valid. Thus, recognition of handwritten numerals is a sensitive task and system must be absolutely accurate for each individual digit.

Traditionally, HNR is considered as a pattern recognition task which includes pre-processing of handwritten numeral images, extraction of features and classification of the images into different numeral categories. Principal component analysis (PCA) [2], genetic algorithm (GA) [3], Bayes theorem, maximum a posteriori and k-means clustering [4], local binary patterns [5], histogram of oriented gradients [6], [7], convex hull [8], chain code and Fourier descriptor [9], wavelets [10]–[12], GIST descriptor [13] etc. are used for feature extraction. Then, different classification algorithms like support vector machine (SVM) [2], [3], [7], K-nearest neighbour (KNN) [14], naïve Bayes, random tree [15], random forest [16], etc. are applied to classify the images into different numeral categories.

Recently, convolutional neural network (CNN) is being used most frequently for image analysis, classification including HNR[17]–[21]. CNN does not require any separate feature extraction step as it is able to extract the inherent features from the image data through their deep layered structure. A number of CNN based studies are shown to outperform the other methods for HNR[20]–[22]. However, CNN based methods show unsatisfactory performance for similar shaped patterns [21]. In case of similar shaped numerals (i.e., the silhouette of the numerals that look same), CNN cannot distinguish between these hand written numerals.

Challenges of HNR are the language dependence based on shape, similarly and other complexities in the numeral sets. When a language contains similar shaped numerals, recognition task becomes difficult even for human because of the similarities turn out to be very close due to variation of writing patterns of people. On the other hand, the same numeral may look very different in size, shape, and orientation due to different writing patterns. Therefore, numerals having similar shaped patterns have lower recognition accuracy with respect to other languages. Among the major languages, Bengali and Devanagari suffer from low recognition accuracy due to the similar shaped numerals. Therefore, HRN system development with focus on similar shaped patterns is an open research challenge.

The main objective of this work is to build a novel HNR system integrating features of human writing style with existing pattern recognition technique of CNN. A hypothesis behind the idea is that writing direction as well as style of a particular numeral in a language is common because people learn to write numerals practicing on the particular pattern. For example, writing direction of Bengali numeral ১ from top to bottom but it is bottom to up for the numeral ৯ in general even though they look similar. Therefore, when CNN confuses to classify a numeral image between ১ and ৯, inclusion of the writing direction would be a good choice for classification. In summary, the proposed method consists of three important stages: classification of a given numeral

M. A. H. Akhand, Md. Rahat-Uz-Zaman, and Shadmaan Hye are with Dept. of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh (e-mail: akhand@cse.kuet.ac.bd).

N. Siddique is with School of Computing, Engineering and Intelligent Systems, Ulster University, United Kingdom (e-mail: nh.siddique@ulster.ac.uk).

image using standard CNN; start-end points measure of the numeral and the final classification decision integrating start-end points measure with CNN's decision. Therefore, the significant contribution of the present study is the start-end points measure from a handwritten numeral image and integration of such measure for better HNR recognition. Efficiency of the proposed system is tested on Bengali and Devanagari numerals owing to achieve fair recognition performance because the HNR accuracy of these languages are relatively low due to inherited challenges.

The major contributions of this research are summarized in the following points:

- This study proposes an improved HNR method focusing similar shaped numerals;
- Start-end points measure of a handwritten numeral image is integrated the decision of CNN for HNR recognition.
- Enhanced recognition accuracy demonstrates the superiority of the proposed system over existing methods.

The rest of the paper is organized as follows: Section II describes the proposed system architectures along with the dataset description. Section III investigates the efficiency of the proposed method through experimental results and analysis. The section also compares performance of the proposed method with other related works. Finally, a brief conclusion of the work is given in Section IV.

II. HNR Integrating Start-End Writing Measure with CNN (SEWM-CNN)

Although CNN based methods outperformed other existing methods of HNR, CNN is shown to misclassify largely for similar shaped numerals [21] discussed in earlier section. According to a recent study [21], handwritten images of ১ (i.e., 1) and ৯ (i.e., 9) of Bengali numerals are interchangeably misclassified in large cases with respect to other numerals. The confidence level of CNN is found relatively low as well as similar for such numerals. In order to enhance the classification accuracy of similar shaped numerals, start-end points measure is introduced as a new feature for classification in addition to CNN. Fig. 1 demonstrates the proposed HNR system integrating Start-End Writing Measure with CNN (SEWM-CNN). Handwritten numeral images may look very different in size, shape, and orientation due to different writing patterns of people. Therefore, proposed SEWM-CNN considered a pre-processing step as like any image-based recognition system to make the images similar sized to put in the system.

There are three major functional steps to classify the pre-processed numeral images (I) in the proposed SEWM-CNN: classification using a CNN; identification of start and end writing points; and finally, system output integrating start-end points measure with the CNN decision. The outcome of CNN is the probability of the input numeral image into different numeral categories. Traditainaly, CNN's classification (i.e., $CL_{CNN} \in \{0, 1, \ldots, 9\}$) for highest probability is the outcome of CNN-based system. In the proposed system, along with classification in CL_{CNN}, its probability value (say CNN's confidence level σ_{CNN}) is also considered as a regulating element. Parallel to CNN's classification operation, SEWM measures start-end points of the numeral image, suggests CL_{SEWM} numeral category for which measured start-end points are found closed to reference start-end points of the numeral class. Finally, output label or system's classification (CL_{Sys}) of the given numeral image is provided comparing σ_{CNN} with predefined threshold value (σ_0): $CL_{Sys} = CL_{CNN}$ if $\sigma_{CNN} => \sigma_0$; otherwise, $CL_{Sys} = CL_{SEWM}$. The following subsections describe the pre-processing of data and the three functional steps of the system.

A. Dataset Selection and Pre-processing

Bengali and Devanagari numerals are chosen to verify the proposed method because both the scripts contain similar shaped numerals and hence accuracy is relatively low due to challenges in classifying such numerals. Another important factor for selection the scripts is the large usability; both the scripts are major scripts in south Asia region and a vast population worldwide use the scripts. Among several collections of handwritten numerals of both the scripts, datasets of Computer Vision and Pattern Recognition Unit, Indian Statistical Institute (ISI) [23] is the most prominent ones and used in several recent studies as benchmark. ISI datasets contain a relatively large number of training and test samples for both Bengali and Devanagari scripts. The samples are from postal codes written by different people with different sex, age and also educational level. Bengali dataset holds 19392 training image samples and 4000 testing image samples. Whereas, training and test sets of Devanagari dataset contain 18793 and 3763 image samples, respectively. In this study, 18000 (=1800×10) images are used for training purpose for each of Bengali and Devanagari. On the other hand, all the available test samples are employed for both Bengali and Devanagari. Table I shows several images from every numeral of both Bengali and Devanagari which reveals the level of ambiguity and challenges in recognition. It is easily visible from the presented images that several images from different numeral categories look very similar in shape including samples of ১ with ৯ in Bengali and samples of ४ with ५ in Devanagari.

Pre-processing is an essential task of feeding numeral images into the recognition system and a simple pre-process is employed in this study. The image samples in both the

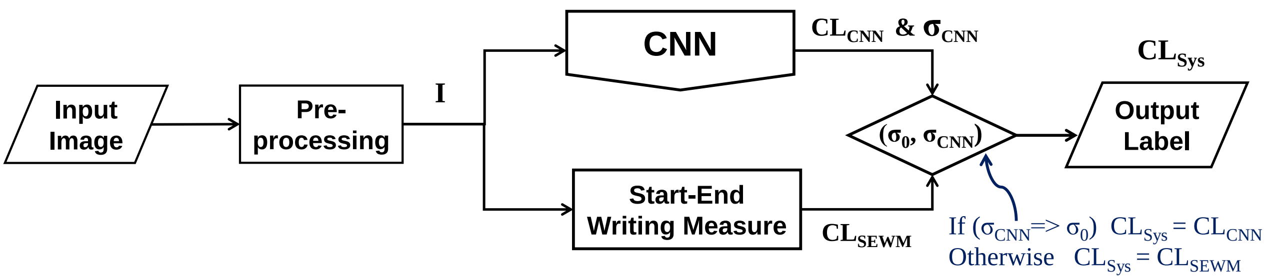
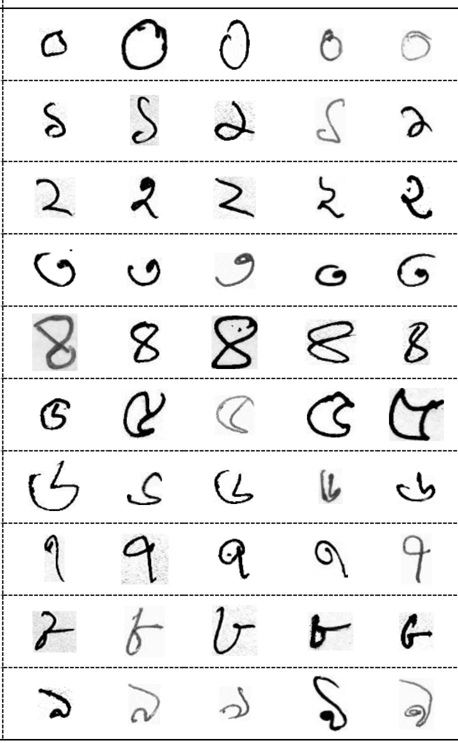
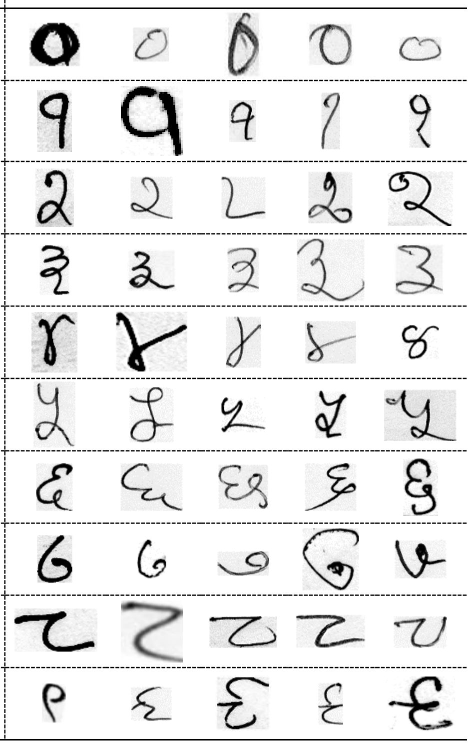


Fig. 1: Structure of the proposed HNR method integrating Start-End Writing Measure with CNN.

TABLE I: HANDWRITTEN NUMERAL SAMPLES FROM ISI BENGALI AND DEVANAGARI DATASETS.

English Numeral	Bengali Numeral	Sample Bengali Handwritten Numeral Images	Devanagari numeral	Sample Devanagari Handwritten Numeral Images
0	০		०	
1	১		१	
2	২		२	
3	৩		३	
4	৪		४	
5	৫		५	
6	৬		६	
7	৭		७	
8	৮		८	
9	৯		९	

datasets are of various sizes, resolutions, and shapes. Automatic thresholding is applied to the images to generate binary images. Foreground and background are interchanged to reduce the numeral values because of the original numerals are written in black on white background leading to more values of 1. Once the foreground-background interchange is performed, there are fewer ones and more zeros, which reduces the computational overhead. Then all the images are resized to the size of 28×28.

There are some well-known HNR works available using ISI datasets. The pioneering work with the datasets is reported in [24] where wavelet filter-based selected features are used in cascade of four multi-layer perceptions (MLPs) for classification. For Devanagari HNR, the works [25] [26] and [27] utilized samples from the ISI Devanagari dataset. Recently, Guha et al. [28] developed HNR system using Memory-Based Histogram with GA for feature selection and KNN for classification; and the method is tested on selected samples from both Bengali and Devanagari datasets. On the other hand, several CNN-based methods have also used the datasets as benchmark. Akhand et al. [22] reported a pioneering work with CNN for Bengali and Bengali-English mixed HNR using ISI Bengali dataset. Shopon et al. [29] also investigated auto-encoder (AE) with CNN for Bengali HNR. Recently, Akhand et al. [21] developed two different CNN based models with rotation based generated patterns from available numeral images and tested on both Bengali and Devanagari datasets.

B. Classification with CNN

The purpose of a classifier is to assign each of its admissible inputs to one of the finite number of classes by computing a set of decision functions. For two-dimensional data such as image, CNN [30] performs well with its convolution and subsampling mechanisms which can capture rotations, shifts invariance and scaling. A standard CNN model of HNR [21] with the following architecture is used in the proposed SEWM-CNN.

$$I_{28\times28} \rightarrow \{6K1_{5\times5}C1_{24\times24} - S_{2\times2}6S1_{12\times12}\} \rightarrow \{12K2_{5\times5}C2_{8\times8} - S_{2\times2}12S2_{4\times4}\} \rightarrow \{Wo_{192\times10}\} \rightarrow O_{10}$$

The CNN has two convolution-subsampling layers and a fully connected layer and Fig. 2 depicts the structure. The input (*I*) is a pre-processed image with size 28×28. First convolution operation with six *5×5* sized kernels (*K1*) on *I* produces 24×24 sized six different convolved feature maps (CFMs) of C1. Then subsampling is used to half the width and height of each CFM with pooling area 2×2, outcomes are six 12×12 sized sub-sampled feature maps (SFMs) in S1. The second convolution operates 12 kernels (*K2*) of size 5×5 on S1 and

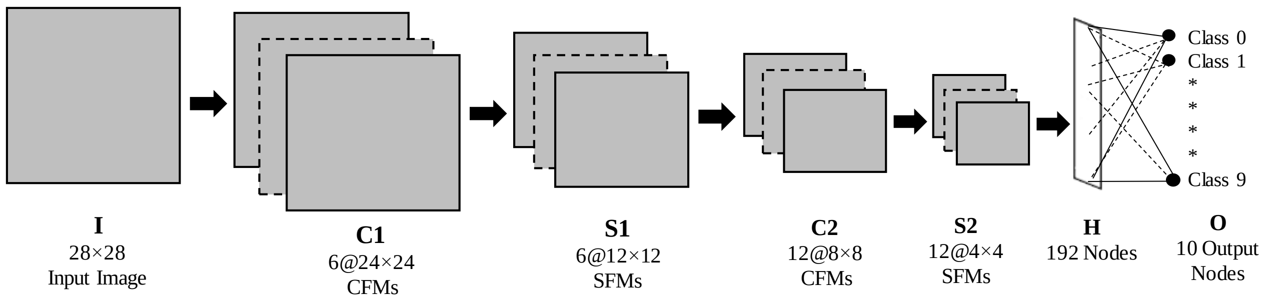


Fig. 2: Structure of CNN used in the proposed SEWM-CNN method for HNR.

outcomes are 8×8 sized 12 CFMs in C2. Again, the second subsampling operation with 2×2 pooling area produces 4×4 sized 12 SFMs in S2. The 192 (=12×4×4) values of these 12 SFMs are placed linearly as hidden layer (*H*) with 192 individual nodes. Finally, nodes of *H* are fully connected to output layer (*O*) through *Wo*. Output layer contains 10 nodes and a particular node represents a particular numeral class. The desired value in a particular node is 1 (and other 9 output nodes value as 0) for the input of a particular numeral category. Training of CNN is performed to get appropriate values of kernels (*K1* and *K2*) and weight (*Wo*) values so that it correctly recognizes a test image generating 1 (or close to 1) in right output node. Output values generated in 10 output nodes are normalized (to sum as 1) to get the classification probability into individual numeral classes. The highest probability value (as confidence level σ_{CNN}) and its class category (CL_{CNN}) are the outcomes of CNN which are proceeded for system's outcome. Detailed training operation of CNN along with description of its structure is available in [21].

C. Start-End Writing Measure from Numeral Image

Integrating the Start-End Writing Measure (SEWM) with CNN for improving accuracy of HNR system is the main contribution of this study. Hand written numerals are the outcomes of sequences of strokes on a paper using pen or similar writing device. Start and end points can make distinguishable features of numerals while they look similar in shape. For an example, images of Bengali numerals ১ and ৯ look similar to each other but the start and end positions are just opposite to each other, that is, the start position of the numeral ১ is at the top while the start position of the numeral ৯ is at the bottom. Since the close similarity between handwritten numerals is a well-established source of low recognition accuracy, inclusion of the start-end positions of writing to feature set would be an effective measure to resolve the ambiguity posed by primary classifier such as CNN. It is worth noting that numeral comprises a sequence of single strokes. However, finding the start and end positions from an image is a challenging task. The following sections describe the extraction method of start-end writing positions in individual numeral images and determining numeral wise start-end reference points.

Extraction of Start-End Positions in a Numeral Image: The steps to find start-end points of writing in a numeral image are extraction of skeleton through thinning and then identification of the terminal points. Thinning is actually the transformation of a numeral image so that the width of the strokes sequence is only one pixel. Zhang-Suen [31] thinning algorithm is applied for this purpose. The algorithm works on binary image and iteratively turns a pixel into black if it is a white boundary pixel until reaching a terminating condition. Zhang-Suen algorithm is a popular thinning algorithm and detailed description of the algorithm is available in [27]-[28].

A traveling salesman problem (TSP) algorithm [33] is applied to the skeleton of the image for stroke sequence extraction. The main task of TSP algorithm is to establish the connectivity among the pixels owing to find writing start-end points. In traditional TSP case, any node may consider as starting point as it is back to starting point. On the other hand, the significance action of sequence generation in HNR case is seeking the sequences start from the top left endpoint to ensure the transformation is consistent. However, endpoints

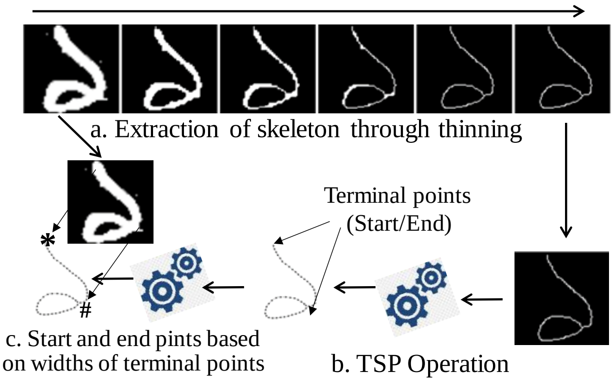


Fig. 3: Demonstration of writing start (* marked) and end (# marked) points identification on a sample image.

might be in different positions for different numerals as well as samples. In TSP operation for stroke sequence extraction, the white pixels are considered as nodes of the TSP and the distance is measured with the Euclidean distance function. As neighbouring pixels are the closest nodes, they will be visited sequentially by the writing device (e.g., pen). The outcome of the TSP algorithm is a sequence from the start point to the end point or vice versa. Actual start and end positions are not found from the TSP operation. Therefore, additional action is introduced to identify the actual start and end points of numeral image.

An innovative idea based on human writing behaviour is utilized to determine actual start and end points of sequence generated from skeleton of numeral image. Pen stroke characteristics to the sequence is considered as a realistic process of finding start and end of writing. While writing, people generally put higher pressure at the starting point and releases off the pressure at the ending point. So that width of pen strokes at the starting point is larger than at the end point and is verified in this study for considered benchmark numeral images. Therefore, widths of two end points (given by TSP sequencing) are measured from the original image and wider end is considered as writing start point and another end is considered as end of writing. Fig. 3 demonstrates the whole process of writing start (* marked) and end (# marked) points identification on a sample image with individual stepwise outcomes.

Determining Numeral wise Reference Start-End Points: Individual numeral wise reference start-end points are essential to distinguish the numerals on the basis of start and end points. The start and end points of a numeral varies among the different handwritten numeral images due to writing variation of individuals. Average position of individual numerals' start (and end) points is an easy way to get the reference start (and end) points of a particular numeral. But the technique is identified as less meaningful because miss identification of start as end (vice versa) of few hand writing cases lead to move the reference point toward inappropriate position.

A clustering-based technique is chosen for finding the numeral wise reference of start and end points. Different individuals follow different writing styles; thus, the number of clusters for start (and end) for a particular numeral may be more than one, vary in number and even unknown. Therefore, Density-based Spatial Clustering of Applications with Noise (DBSCAN)[34], [35] algorithm is employed in this study where spatial position of each start or end points and proximity of the points is the key to the clustering. Reference start and end points are marked on printed numeral image which are two points in a 2D pixel space.

Reference start and end points measurement is demonstrated for ISI Bengali and Devanagari datasets in Table II. Left three columns of the table are for Bengali numerals: start-end points marking are shown on two sample handwritten images in columns 1-2 and finally, DBSCAN cluster-based reference points for start-end points on printed image is shown in column 3. For marking, star (*) in red is used for start point and hash (#) in blue is used for end point. It is notable that start and end reference points are almost same for Bengali numerals ০, ৪ and ৫ as those are closed loop writing numerals and hence start and end points marked are overleaped. For other numerals such as ১, ২, ৩, ৬, ৭, ৮ and ৯, the start and end reference points are different and more interestingly reference points of start and end point for ১ and ৯ are on opposite locations though they look very similar in shape. Similar cases are presented for Devanagari in columns 4-6.

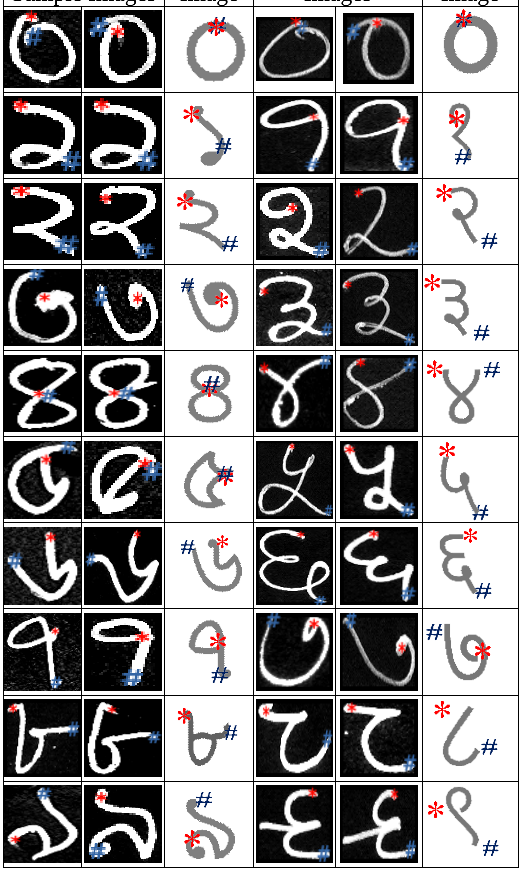
Individual numeral wise reference start-end points are used to identify for which numeral a given numeral image is closed to. At first, start and end points are calculated from the given image following the steps described in Section II(C). Then Euclidian distances are measured with reference start and end points of each of the numeral. Numeral with smallest distance is assigned to the test numeral. Suppose (S_i, E_i) is the calculated start and end positions for the given image; and reference start and end position pairs of 10 numerals are $(S_0, E_0), (S_1, E_1)$, ----, and (S_9, E_9). The distance with 0 numeral is:

$$D_0 = (DS_{i,0} + DE_{i,0})/2, \qquad (1)$$

where $DS_{i,0}$ (and $DE_{i,0}$) are Euclidian distance of start (and end) points of the given image and reference points of numeral 0. Similarly, distances with other numeral references are D_1, D_2,...,and D_9. The outcome of the SEWM (i.e., CL_{SEWM}) is the numeral for which distance is minimum, i.e., $min\{D_1, D_2, \ldots D_9\}$.

TABLE II: NUMERAL WISE START AND END POINTS MARKING ON SAMPLE IMAGES AND START AND END REFERENCE POINTS MARKING ON PRINTED NUMERALS.

Bengali Numerals		Devanagari Numerals	
Start and End Points on Sample Images	Ref. on Printed Image	Start and End Points on Sample Images	Ref. on Printed Image



D. System Outcome Considering CNN's Confidence and Start-End Writing Measure

Final decision on recognition of the proposed SEWM-CNN depends on the CNN's confidence level (σ_{CNN}) in classification of the input image. Suppose σ_{CNN} => 0.7, classifying with such high confidence value assures very low confidence levels of other categories and sum of those are 0.3 or less. On the other hand, classification with σ_{CNN} <= 0.5 indicates that confidence level in any other category might be competitive. Competitive (as well as low confidence) in two different numeral categories for a numeral image indicates that CNN confuses in making decision. Such scenario is common for similar shaped numerals. In such case, the decision from SEWM is considered as the final recognition category of the proposed SEWM-CNN system.

In short, getting the SEWM-CNN output is a selection process between decisions of CNN and SEWM as shown in Fig. 1. The selection is performed based on a defined threshold value (σ_0). The final system outcome will be CNN's recognition category (i.e., $CL_{Sys} = CL_{CNN}$) if it classifies the image with confidence level equal or above σ_0 (i.e., σ_{CNN} => σ_0). In contrary, outcome of SEWM is exposed as system outcome (i.e., $CL_{Sys} = CL_{SEWM}$) for $\sigma_{CNN} < \sigma_0$. It is here notable that such selection-based integration does not incur computational cost in system operation with respect to CNN or SEWM.

Suppose the probability values of a sample image in 10 numeral classes by CNN are [0, 0.54, 0, 0, 0, 0, 0, 0, 0, 0.46] and distances between start-end points of the numeral image with reference start-end points of individual numerals from SEWM are [13.3, 19.4, 22.0, 11.4, 13.8, 17.5, 14.0, 21.7, 7.7]. Therefore, $CL_{CNN} = 1$ with $\sigma_{CNN} = 0.54$ and $CL_{SEWM} = 9$ for the lowest distance of 7.7. If $\sigma_0 = 0.6$ (i.e., $\sigma_{CNN} < \sigma_0$), the system relies on SEWM and $CL_{Sys} = CL_{SEWM} = 9$. It seems the value of σ_0 between 0.5 and 0.8 is suitable in SEWM-CNN.

E. Significance of the Proposed System

There is significant difference between the proposed SEWM-CNN and the traditional techniques for HNR. The main significance of the present study is the development of a novel HNR system based on the hypothesis of human writing style. Diverse writing styles produce very closely similar shaped numerals; thus, image-based method CNN confuse in recognition such numeral images in appropriate category. To encounter the situation, another numeral writing hypothesis is integrated in this study. The hypothesis of start-end writing positions of individual numerals is considered as an additional technique to distinguish such similar shaped numerals. Start-end points measure from a numeral image is a challenging task and different innovative steps are taken into account for the task.

Integration of SEWM with CNN is also significant in terms of computational cost although computational cost is not so important due to easy availability high computing machines now-a-days. In the proposed system, CNN training and numeral wise reference start-end points identification are the main computational tasks and these two tasks are independent of one another. Reference start-end points identification took less computational operation than CNN training on certain iterations. More importantly, integration of SEWM does not incur computational cost in system operation. System with SEWM obviously computationally effective with respect to other exiting approaches to enhance CNN's performance. The well performed CNN-based method of [21] followed a kind of data augmentation which seems three times computationally heavy with respect to a single CNN.

III. Experimental Studies

These section presents the verification of the proposed SEWM-CNN method on the ISI datasets of Bengali and Devanagari scripts. The descriptions of the datasets are already given in previous section. The performance of the proposed method is investigated for different important issues such as varying threshold values (σ_0) of system in decision making in between CNN and SEWM. Finally, the proficiency of the proposed SEWM-CNN is validated comparing with other prominent methods.

The proposed system was implemented in python using Keras and Tensorflow. The experiments were carried out onto Windows 10 OS and Python 3.6 Anaconda environment on a Desktop PC with the following configuration: Intel(R) Core i7-7770 CPU @ 4.20 GHz, 16 GB RAM and Nvidia GTX 1050ti 8GB GPU.

A. Experimental Results and Analysis

The training and test samples are available separately in ISI datasets. Training samples are used to train the CNN, and measuring the start-end reference points of individual numerals; samples of the test set are used to measure final efficiency (i.e., generalization ability) of the system. Training conducted with different batch sizes (BS) as the number of samples in a training batch has an effect on the performance of a system. On the other hand, σ_0 value is an important parameter of SEWM-CNN, therefore the performance is measured against different it values. Experimental outcomes of standard CNN (i.e., CNN alone) are also considered to realize the effect of SEWM integration with CNN in the proposed SEWM-CNN.

Table III shows the test set recognition accuracy for Bengali and Devanagari with CNN alone and proposed SEWM-CNN varying BS from 8 to 128. Due to batch size variation, recognition accuracy is shown to vary for both CNN and SEWM-CNN for Bengali and Devanagari. And, smaller batch size values showed relatively better recognition accuracy for both the scripts. For a small batch size, CNN is updated considering relatively smaller number of training samples at a time. How the performance improves with CNN training is shown in Fig. 4 for Bengali for a batch size of 16. It is observed from the figure that at the beginning (iteration up to 20) the recognition accuracy is low and improves with iteration for both CNN and SEWM-CNN with any σ_0 value.

TABLE III: Test Set Recognition Accuracy of CNN and Proposed SEWM-CNN with Different Threshold Values for Different Batch Sizes.

(a) Bengali

Batch Size	Recognition accuracy (%) of CNN	Recognition accuracy (%) of proposed SEWM-CNN with different σ_0 values			
		0.5	0.6	0.7	0.8
8	98.98	99.13	**99.20**	99.10	99.03
16	98.98	99.03	**99.20**	99.03	99.0
32	98.60	98.70	98.85	98.83	98.65
64	98.58	98.68	98.80	98.73	98.60
128	98.43	98.58	98.65	98.60	98.48

(b) Devanagari

Batch Size	Recognition accuracy (%) of CNN	Recognition accuracy (%) of proposed SEWM-CNN with different σ_0 values			
		0.5	0.6	0.7	0.8
8	99.08	99.13	**99.23**	99.10	99.08
16	99.10	99.15	**99.23**	99.15	99.13
32	98.78	98.88	98.90	98.85	98.83
64	98.78	98.85	98.83	98.78	98.75
128	98.50	98.58	98.70	98.55	98.53

It is notable that SEWM-CNN always outperformed CNN; and SEWM-CNN with σ_0 = 0.6 is shown to be the best. But accuracy is shown to decline after 80 iterations for any case indicating the overfitting. The best recognition accuracy of a method is considered to compare in Table III.

Remarkable observation from Table III is that SEWM-CNN outperformed CNN for any BS value which indicates the proficiency of the proposed approach. As an example, the best recognition accuracy of CNN for Bengali (in Table III(a)) is 98.98% for BS values of 8 and 16. On the other hand, for the same BS values, SEWM-CNN achieved recognition accuracy 99.20% for σ_0 = 0.6. Accuracy of SEWM-CNN for any other σ_0 value (i.e., 0.5, 0.7, or 0.8) is also better than CNN. Similar observation is also available for Devanagari in Table III(b); CNN is shown to be the best recognition accuracy 99.10% for BS 16; SEWM-CNN with any σ_0 value

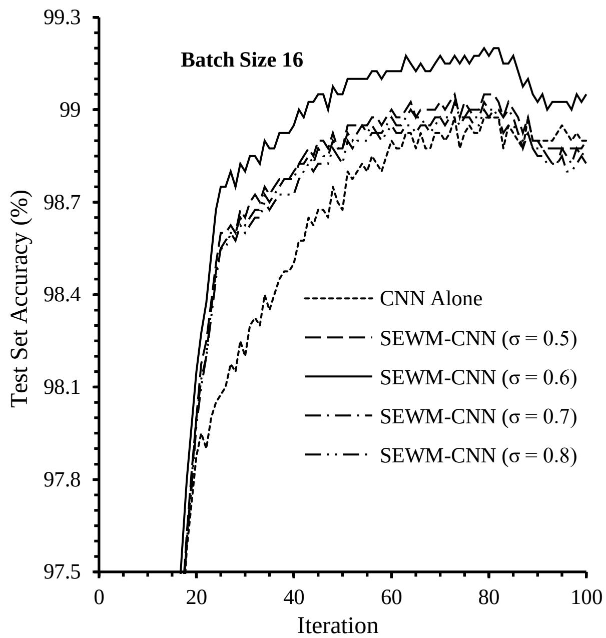


Fig. 4: Performance of CNN and proposed SEWM-CNN varying iteration on ISI Bengali dataset.

achieved better accuracy than CNN. Similar to Bengali, SEWM-CNN achieved the best accuracy with $\sigma_0 = 0.6$ and the value is 99.23%. It is notable that SEWM-CNN considers decision's from SEWM technique in relatively large number for higher σ_0 values (e.g., 0.8) and similar start-end reference points among several numerals might incur wrong decisions in several cases. On the other hand, for lower σ_0 values (e.g., 0.5), SEWM-CNN mostly considers CNN's outcome as system decision ignoring SEWM and hinders the uses of start-end reference measure. According to the results presented in Table III the $\sigma_0 = 0.6$ is found to be the best suitable value for both Bengali and Devanagari numerals although all other values is shown to improve recognition accuracy of the system rectifying CNN's decision in a range.

Individual numeral wise recognition analysis for better understanding is presented in Table IV for both Bengali and Devanagari. For a script, (Bengali / Devanagari) outcomes of CNN and SEWM-CNN with σ_0 value 0.6 are presented for same BS value 16. According to Table IV(a), there are total of 41 test samples (out of 4000) Bengali dataset misclassified by CNN and the number reduced to 32 for SEWM-CNN; hence recognition accuracy improved from 98.98% to 99.20% as presented in the Table III(a). CNN misclassified Bengali numeral ১ as ৯ in 12 alone out of total 18 misclassified cases as seen in Table IV(a). In the contarry, SEWM-CNN misclassified ১ as ৯ in seven cases and total misclassified number reduced to 16. Promising result has been found by SEWM-CNN in case of ৯; all eight missclasified cases as ১ by CNN are rectified by SEWM-CNN and total misclassified number is now reduced from 10 to 2. It is already mentioned that numerals ১ and ৯ are similar in shape even in printed form and it is sometines difficult to distinguish between them. Table IV(b) also shows similar observation for Devanagari due to SEWM total number of missclassified test samples reduced from 36 (out of 3763) to 29; hence recognition accuracy improved from 99.10% to 99.23% as presented in the Table III(b). In individual numeral case, interchangeably misclassification between ४ and ५, and between ६ and ९ are reduced. Finally, reduction of interchangibly missclassication of the numerals clearly indicates the effectiveness of SEWM to improve performance in the proposed SEWM-CNN method.

Table V presents observation on several test samples from Bengali and Devanagari datasets to realize the proficiency of SEWM-CNN as well as the cause of the misclassified samples. According to Table IV, although SEWM-CNN outperformed CNN but all the samples misclassified by CNN were not truly classified by SEWM-CNN. Table V(a) shows several handwritten numeral images from test set of Bengali

TABLE IV: INDIVIDUAL NUMERAL WISE PERFORMANCE OF CNN AND PROPOSED SEWM-CNN ON TEST SAMPLES.

(a) Bengali

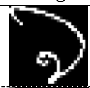
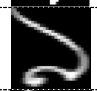
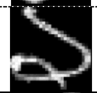
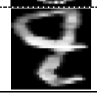
Bengali Numeral	Total Samples	CNN		SEWM-CNN	
		Truly Classified	Misclassification to Other Numerals and Count	Truly Classified	Misclassification to Other Numerals and Count
0	400	399	৫(1)	399	৯(1)
১	400	382	২(1)-৪(1)-৫(2)-৬(1)-৭(1)-৯(12)	384	২(1)-৪(3)-৫(2)-৭(2)-৮(1)-৯(7)
২	400	398	৫(1)-৯(1)	399	৮(1)
৩	400	400	-	399	0(1)
৪	400	400	-	399	৫(1)
৫	400	394	২(2)- ৪(2)-৬(1)-৭(1)	395	১(1)-২(1)-৭(1)-৮(1)-৯(1)
৬	400	397	৩(1)-৫(2)	398	0(1)-১(1)
৭	400	399	৬(1)	399	৪(1)
৮	400	400	-	398	২(1)-৩(1)
৯	400	390	১(8)-৬(1)-৭(1)	398	৬(1)-৭(1)
Total	**4000**	**3959**	**41**	**3968**	**32**

(b) Devanagari

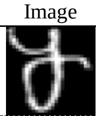
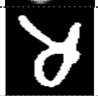
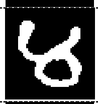
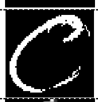
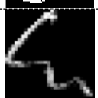
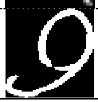
Devanagari Numeral	Total Samples	CNN		SEWM-CNN	
		Truly Classified	Misclassification to Other Numerals and Count	Truly Classified	Misclassification to Other Numerals and Count
०	369	364	४(1)-७(3)-८(1)	364	४(1)-७(3)-८(1)
१	378	376	०(1)-३(1)	376	०(1)-३(1)
२	378	376	१(1)-५(1)	376	१(1)-५(1)
३	377	375	६(1)-९(1)	374	५(1)-६(1)-९(1)
४	376	372	०(1)- ५(3)	374	०(1)- ५(1)
५	378	370	३(1)-४(7)	374	३(2)-४(2)
६	374	367	३(1)-८(1)-९(5)	370	३(1)-८(1)-९(2)
७	378	377	०(1)	377	०(1)
८	377	376	५(1)	374	५(2)- ९(1)
९	378	374	६(4)	375	६(2)- ८(1)
Total	**3763**	**3727**	**36**	**3734**	**29**

TABLE V: PROFICIENCY OF SEWM-CNN COMPARING CNN ON SELECTED TEST SAMPLES.

(a) Bengali

	Sl.	Handwritten Image	CL_{CNN} - CL_{Sys} - True Category	Remarks on the Sample. N.B.: $\sigma_0 = 0.6$
Samples for SEWM rectified CNN' s Wrong Decision	1		১ - ৯ - ৯	$\sigma_{CNN} = 0.53$ and $CL_{Sys} = CL_{SEWM} = 9$ (i.e., ৯)
	2		১ - ৯ - ৯	$\sigma_{CNN} = 0.54$ and $CL_{Sys} = CL_{SEWM} = 9$ (i.e., ৯)
	3		৯ - ১ - ১	$\sigma_{CNN} = 0.46$ and $CL_{Sys} = CL_{SEWM} = 1$ (i.e., ১)
Samples for SEWM-CNN Misclassified	4		৯ - ৯ - ১	$\sigma_{CNN} = 0.59$ and $CL_{Sys} = CL_{SEWM} = 9$ (i.e., ৯)
	5		৩ - 0 - ৬	$\sigma_{CNN} = 0.51$ but $CL_{Sys} = CL_{SEWM} = 0$ (i.e.,0)
	6		৪ - ৪ - ৭	$\sigma_{CNN} = 0.67$, so $CL_{Sys} = CL_{CNN} = 4$ (i.e., ৪) ignoring SEWM

(a) Devanagari

	Sl.	Handwritten Image	CL_{CNN} - CL_{Sys} - True Category	Remarks on the Sample. N.B.: $\sigma_0 = 0.6$
Samples for SEWM rectified CNN' s Wrong Decision	1		४ - ५ - ५	$\sigma_{CNN} = 0.52$ and $CL_{Sys} = CL_{SEWM} = 5$ (i.e., ५)
	2		४ - ५ - ५	$\sigma_{CNN} = 0.55$ and $CL_{Sys} = CL_{SEWM} = 5$ (i.e., ५)
	3		५ - ४ - ४	$\sigma_{CNN} = 0.49$ and $CL_{Sys} = CL_{SEWM} = 4$ (i.e., ४)
Samples for SEWM-CNN Misclassified	4		८ – ८ - ०	$\sigma_{CNN} = 0.48$ and $CL_{Sys} = CL_{SEWM} = 8$ (i.e., ८)
	5		९ – ९ - ६	$\sigma_{CNN} = 0.54$ and $CL_{Sys} = CL_{SEWM} = 9$ (i.e., ९)
	6		० – ० - १	$\sigma_{CNN} = 0.69$, so $CL_{Sys} = CL_{CNN} = 0$ (i.e., ०) ignoring SEWM

data set and gives individual numeral wise classification by CNN, SEWM-CNN and remarks on actions. Among six samples, first three samples were misclassified by CNN but truly classified by SEWM-CNN showing proficiency of SEWM integration with CNN and rests of the samples are in the category for those SEWM-CNN was also failed to classify truly. Remarks on three true classified cases by SEWM-CNN are common for both ১ and ৯ numeral cases; CNN's confidence in wrong classification was below the threshold value (i.e., below 0.6) and SEWM-CNN considered SEWM's decision which was in true numeral category. On the other hand, the reason of SEWM-CNN misclassification is the decision of SEWM were not considered as CNN wrongly classified with high confidence value (Sl. 6) or SEWM agreed to CNN's misclassification (Sl. 4). Observations are also similar for Devanagari samples as shown in Table V(b): SEWM-CNN provides correct result while SEWM rectified wrong decision of CNN (Sl. 1-3); SEWM-CNN failed to classify truly in cases when CNN wrongly classified with high confidence value (Sl. 6) or SEWM agreed with CNN's wrong classification (Sl. 4 and 5). Both Bengali and Devanagari misclassified samples are so confusing even by human eye judgment. It is notable that test samples were reserved to check the proficiency of the system and did not include in any phase of system development. Therefore, misclassification of several test samples of those are much harder to understand even by human being is acceptable logically.

B. *Performance Comparison*

This section compares performance of proposed SEWM-CNN with prominent existing works in recognition of Bengali and Devanagari handwritten numerals. Along with test set recognition accuracy, dataset uses and distinguished properties of individual methods are also presented for better understanding in comparison Table VI. Both CNN-based methods and feature-based methods are included in comparison. Few exiting methods are only tested on both the scripts. Several feature-based methods used self-prepared datasets and number of samples in training and test sets are different from ISI datasets used in this study. However, the proposed method outperformed any feature-based method for both Bengali and Devanagari. For Bengali, among the feature-based methods, the most recent work with Memory-Based Histogram + GA for feature selection and classification with KNN [28] is shown the best recognition accuracy; the achieved recognition accuracy 98.40% is inferior to the proposed method. On the other hand, the pioneer work with wavelet filter and classification with cascade of several MLPs on the ISI datasets [24] is still shown the best recognition accuracy for Devanagari; the achieved recognition accuracy 99.04% is also inferior to the proposed method.

The proposed SEWM-CNN is with integration of SEWM measure with CNN; therefore, its performance comparison with other CNN-based methods is more appropriate. The exiting CNN-based methods presented in Table VI are also tested on the same ISI datasets which makes the comparison

TABLE VI: COMPARISON OF PROPOSED SEWM-CNN WITH PROMINENT METHODS FOR BENGALI AND DEVANAGARI HNR IN TERMS OF RECOGNITION ACCURACY, DATASET USED AND METHOD'S SIGNIFICANCE.

Work Ref., Year	Dataset; Training and Test Samples	Recognition Accuracy		Method's Significance in Feature Selection and Classification
		Bengali	Devanagari	
Wen et al. [2], 2007	Postal system; 6000 and 10000	95.05%	-	PCA based feature selection and SVM for classification.
Bhattacharya and Chaudhuri [24], 2009	ISI[23]; 19392 and 4000	98.20%	99.04%	Wavelet filter-based feature selection and cascade of four MLPs for classification.
Wen and He [36], 2012	Postal system; 30000 and 15000	96.91%	-	Feature selection using eigenvalues and eigenvectors and classification using kernel and Bayesian discriminant.
Das et al. [3], 2012	CMATERdb 3.1.1[37]; 4000 and 2000	97.70%	-	Feature selection in different stages using GA and classification using SVM.
Nasir and Uddin [4], 2013	Self-prepared, 300	96.80%	-	Bayes' theorem, k-means clustering and Maximum Posteriori for feature selection and SVM for classification.
Kumar and Ravulakollu [38], 2014	CPAR – 2012[38]; 24000 and 11000	-	97.87%	Features are based on profile and gradient and classification using NNs (in ensemble and cascade manners) and KNN.
Singh et. al. [25], 2014	Samples from ISI[23]; 1400 and 600	-	98.53%	Feature selection using information theoretic based MRMR and classification using NNs and ensemble of NNs.
Arya et al. [26], 2015	ISI[23]; 19798 and 3763	-	98.06%	Feature selection using Gabor filter and classification using KNN and SVM.
Singh et al. [39], 2016	CMATERdb 3.2.1 [37]; 2000 and 1000	-	98.92%	Moment based six different features and classification using MLP.
Guha et al. [28], 2019	Self-prepared + Samples from ISI[23]; 10000 and 500	98.40%	97.60%	Memory-Based Histogram + GA for feature selection and KNN for classification.
		98.05%	95.05%	Memory-Based Histogram + GA for feature selection and MLPs for classification.
Akhand et al. [22], 2016	ISI[23]; 18000 and 4000	98.45%	-	Standard CNN
Shopon et al. [29], 2016	ISI[23]; 19313 and 3986	98.29%	-	Auto-encoder (AE) with CNN
Akhand et al. [21], 2019	ISI[23]; 18000 and 4000 (Bengali) / 3763 (Devanagari)	98.98%	99.31%	Ensemble of three CNNs; one is trained with available samples and other two used rotation based generated data.
		98.96%	98.96%	CNN is trained using available data plus rotation based generated data.
Proposed SEWM-CNN	ISI[23]; 18000 and 4000 (Bengali) / 3763 (Devanagari)	99.20%	99.23%	Start-End Writing Measure is integrated with CNN's decision.

more justified. Among several CNN-based methods for Bengali, the most recent work with rotation based generated patterns [21] is shown the best recognition accuracy. In [21], two additional training sets are created rotating original training samples with fixed defined angle clock wise and anti-clock wise; along with original samples two different approaches are considered to train CNN. In case of multiple CNNs, three training sets are used to train three different CNNs (with same architecture) individually and final outcome is generated combining decisions of the three CNNs. In another approach, a single CNN is trained combining the three training sets. The multiple CNN case is shown better performance than single CNN case and the achieved recognition accuracy is 98.98% for Bengali. The weakness of the method is reported for misclassification of similar shaped numerals (e.g., ১ and ৯) interchangeably. The method investigated in this study owing to tackle the issue integrating SEWM with CNN and shown to achieve acceptable result. Therefore, the proposed SEWM-CNN outperformed the multiple CNNs case with recognition accuracy 99.20% even with single CNN. The work of [21] also investigated for Devanagari and achieved recognition accuracies 99.31% and 98.96% for multiple CNNs and single CNN case, respectively. The proposed SEWM-CNN is shown a recognition accuracy of 99.23% for Devanagari and the value is better than single CNN and competitive to multiple CNNs case of [21]. However, better performance of SEWM-CNN with an ordinary trained CNN (training with available data) than a CNN training with three times larger training set in [21] is a remarkable achievement. Finally, the achieved recognition accuracy of the proposed SEWM-CNN in comparison with standard CNN based methods (e.g., [22]) and CNN with data augmentation (e.g., [21]) revealed the proposed method as an effective method for recognizing Bengali and Devanagari handwritten numerals.

IV. Conclusions

HNR is a complex task and postures challenges for the similar shaped numerals. Although convolutional neural network (CNN) is the most successful model for image classification, it suffers from low accuracy of recognising similar shaped numerals. This study investigated an innovative HNR method focusing on similar shaped numerals accompanying a significant property of human writing style. Individual numerals have significant start and end positions; and the start-end writing measure (SEWM) technique and its integration with CNN are the major contributions of this study. Proposed SEWM-CNN has been tested on Bengali and Devanagari benchmark datasets and is shown to achieve better recognition accuracy compared to standard CNN.

A number of potential future directions of research can be foreseen out of the present study in the development of better HNR system. The proposed method with necessary modification might show even better performance for other languages. The CNN of SEWM-CNN is trained with the available samples; the use of data augmentation might also improve performance of the method and remain as future study. More promising future direction of research of this study is to use the idea of start-end measure in a different form rather than simple Euclidian distance-based method of this study. Another challenging task is to develop different way to rectify the decision of CNN with the different hypothesis rather than start-end writing measure.

Acknowledgment

The authors would like to express gratitude to Dr. U. Bhattacharya of Indian Statistical Institute, Kolkata, India for providing the benchmark datasets used in this study.

References

[1] R. Plamondon and S. N. Srihari, “Online and off-line handwriting recognition: a comprehensive survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, 2000, doi: 10.1109/34.824821.

[2] Y. Wen, Y. Lu, and P. Shi, “Handwritten Bangla numeral recognition system and its application to postal automation,” *Pattern Recognit.*, vol. 40, no. 1, pp. 99–107, Jan. 2007, doi: 10.1016/j.patcog.2006.07.001.

[3] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, “A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application,” *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1592–1606, May 2012, doi: 10.1016/j.asoc.2011.11.030.

[4] M. K. Nasir, “Hand Written Bangla Numerals Recognition for Automated Postal System,” *IOSR J. Comput. Eng.*, vol. 8, no. 6, pp. 43–48, 2013, doi: 10.9790/0661-0864348.

[5] T. Hassan and H. A. Khan, “Handwritten Bangla numeral recognition using Local Binary Pattern,” in *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2015, pp. 1–4, doi: 10.1109/ICEEICT.2015.7307371.

[6] T. Choudhury, A. ; Rana, H. S. ; Bhowmik, “Handwritten Bengali Numeral Recognition using HOG Based Feature Extraction Algorithm,” in *5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida*, 2018, pp. 687–690, doi: 10.1109/SPIN.2018.84742155.

[7] H. A. Khan, “MCS HOG Features and SVM Based Handwritten Digit Recognition System,” *J. Intell. Learn. Syst. Appl.*, vol. 09, no. 02, pp. 21–33, 2017, doi: 10.4236/jilsa.2017.92003.

[8] M. Das, Nibaran; Pramanik, Sandip; Basu, Subhadip; Saha, Punam; Sarkar, Ram ; Kundu, Mahantapas; Nasipuri, “Recognition of Handwritten Bangla Basic Characters and Digits using Convex Hull based Feature Set,” in *International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09)*, 2009, doi: 10.13140/2.1.3689.4089.

[9] S. G Rajput, Ganapatsingh; Mali, “Marathi Handwritten Numeral Recognition using Fourier Descriptors and Normalized Chain Code,” *Int. J. Comput. Appl.*, pp. 141–145, 2010.

[10] D. Romero, A. Ruedin, and L. Seijas, “Wavelet-Based Feature Extraction for Handwritten Numerals,” 2009, pp. 374–383.

[11] E. M. Seijas, Leticia; C. Segura, “A Wavelet-based Descriptor for Handwritten Numeral Classification,” in *International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 649–654.

[12] A. Almuttardi, Bubakir; Ambarek, Abdulsalam; Khadija, “Handwritten Numeral Recognition Using Wavelet Transform and Neural Networks,” in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2013)*, 2013, pp. 146–158.

[13] T.-N. Do and N.-K. Pham, "Handwritten Digit Recognition Using GIST Descriptors and Random Oblique Decision Trees," 2015, pp. 1–15.

[14] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002, doi: 10.1109/34.993558.

[15] S. M. Shamim, M. ; Miah, Badrul Alam Miah, A. Sarker, and A. Al Rana,Masud; Jobair, "Handwritten Digit Recognition using Machine Learning Algorithms," *Glob. J. Comput. Sci. Technol. D Neural Artif. Intell.*, vol. 18, no. 1, pp. 17–23, 2018.

[16] S. Bernard, S. Adam, and L. Heutte, "Using Random Forests for Handwritten Digit Recognition," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, 2007, pp. 1043–1047, doi: 10.1109/ICDAR.2007.4377074.

[17] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification," *IEEE Trans. Cybern.*, pp. 1–15, 2020, doi: 10.1109/TCYB.2020.2983860.

[18] Y. Hu, G. Wen, H. Liao, C. Wang, D. Dai, and Z. Yu, "Automatic Construction of Chinese Herbal Prescriptions From Tongue Images Using CNNs and Auxiliary Latent Therapy Topics," *IEEE Trans. Cybern.*, pp. 1–14, 2019, doi: 10.1109/TCYB.2019.2909925.

[19] W. Wu, Y. Yin, X. Wang, and D. Xu, "Face Detection With Different Scales Based on Faster R-CNN," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 4017–4028, Nov. 2019, doi: 10.1109/TCYB.2018.2859482.

[20] H. Wang, P. Chen, and S. Kwong, "Building Correlations Between Filters in Convolutional Neural Networks," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3218–3229, Oct. 2017, doi: 10.1109/TCYB.2016.2633552.

[21] M. A. H. Akhand, M. Ahmed, M. M. H. Rahman, and M. M. Islam, "Convolutional Neural Network Training incorporating Rotation-Based Generated Patterns and Handwritten Numeral Recognition of Major Indian Scripts," *IETE J. Res.*, vol. 64, no. 02, p. 176, 2018, doi: 10.1080/03772063.2017.1351322.

[22] M. A. H. Akhand, M. Ahmed, and M. M. H. Rahman, "Convolutional Neural Network based Handwritten Bengali and Bengali-English Mixed Numeral Recognition," *Int. J. Image, Graph. Signal Process.*, vol. 8, no. 9, pp. 40–50, 2015, doi: 10.5815/ijigsp.2015.09.06.

[23] B. . Bhattacharya, U.; Chaudhuri, "Offline Handwritten Bangla and Devanagari Numeral Databases. Kolkata: Computer Vision and Pattern Recognition Unit, Indian Statistical Institute." 2009.

[24] U. Bhattacharya and B. B. Chaudhuri, "Handwritten Numeral Databases of Indian Scripts and Multistage Recognition of Mixed Numerals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 444–457, Mar. 2009, doi: 10.1109/TPAMI.2008.88.

[25] P. Singh, A. Verma, and N. S. Chaudhari, "Devanagri Handwritten Numeral Recognition using Feature Selection Approach," *Int. J. Intell. Syst. Appl.*, vol. 6, no. 12, pp. 40–47, Nov. 2014, doi: 10.5815/ijisa.2014.12.06.

[26] S. Arya, I. Chhabra, and G. S. Lehal, "Recognition of Devnagari Numerals using Gabor Filter," *Indian J. Sci. Technol.*, vol. 8, no. 27, pp. 1–6, Oct. 2015, doi: 10.17485/ijst/2015/v8i27/81856.

[27] P. S and R. Dinesh, "Handwritten Devanagari Characters and Numeral Recognition using Multi-Region Uniform Local Binary Pattern," *Int. J. Multimed. Ubiquitous Eng.*, vol. 11, no. 3, pp. 387–398, Mar. 2016, doi: 10.14257/ijmue.2016.11.3.37.

[28] R. Guha, M. Ghosh, P. K. Singh, R. Sarkar, and M. Nasipuri, "M-HMOGA: A New Multi-Objective Feature Selection Algorithm for Handwritten Numeral Classification," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1453–1467, Dec. 2019, doi: 10.1515/jisys-2019-0064.

[29] M. Shopon, N. Mohammed, and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," in *2016 International Workshop on Computational Intelligence (IWCI)*, 2016, pp. 64–68, doi: 10.1109/IWCI.2016.7860340.

[30] Yangqing Jia, Chang Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3370–3377, doi: 10.1109/CVPR.2012.6248076.

[31] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[32] W.-H. Hsu and Y.-S. Chen, "A modified fast parallel algorithm for thinning digital patterns," *Pattern Recognit. Lett.*, vol. 7, no. 2, pp. 99–106, 1988.

[33] M. A. H. Akhand, S. Akter, M. A. Rashid, and S. B. Yaakob, "Velocity tentative PSO: An optimal velocity implementation based particle swarm optimization to solve traveling salesman problem," *IAENG Int. J. Comput. Sci.*, vol. 42, no. 3, pp. 1–12, 2015.

[34] M. Daszykowski and B. Walczak, "Density-Based Clustering Methods," in *Comprehensive Chemometrics*, Elsevier, 2009, pp. 635–654.

[35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[36] Y. Wen and L. He, "A classifier for Bangla handwritten numeral recognition," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 948–953, Jan. 2012, doi: 10.1016/j.eswa.2011.07.092.

[37] S. Basu, D.K.; Nasipuri, M.; Kundu, M.; Basu, S.; Das, N.;Sarkar, R.; Mollah, A.F.; Saha, "CMATERdb 3.1.1 & CMATERdb 3.2.1: Handwritten Bangla and Devanagari Numeral Databases. Kolkata: Center for Microprocessor Application for Training Education and Research, Jadavpur University." 2011.

[38] R. Kumar and K. K. Ravulakollu, "Handwritten devnagari digit recognition: Benchmarking on new dataset," *J. Theor. Appl. Inf. Technol.*, vol. 60, no. 3, pp. 543–555, 2014.

[39] P. K. Singh, R. Sarkar, and M. Nasipuri, "A Study of Moment Based Features on Handwritten Digit Recognition," *Appl. Comput. Intell. Soft Comput.*, vol. 2016, pp. 1–17, 2016, doi: 10.1155/2016/2796863.