2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Exploring Discriminative HMM States for Improved Recognition of Online Handwriting

Subhasis Mandal, Himakshi Choudhury, S. R. Mahadeva Prasanna and Suresh Sundaram
Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati
Email: {subhasis.mandal, himakshi, prasanna, sureshsundaram}@iitg.ernet.in

***Abstract*—In this paper, we propose a novel approach for online handwriting recognition (HR) based on hidden Markov model (HMM). In a conventional HMM-based HR system, the input test sample is recognized by first measuring the *log*-likelihood score from each class-specific HMM, and then the class with the highest score is assigned as the recognized class. It is observed that, for a given test sample, the difference in *log*-likelihood scores of top-2 outputs (classes) is often less for faithful classification. The problem intensifies for those scripts that have a large set of similar shape characters such as the Indic script. To address this problem, first, we analyze the HMM states corresponding to the top-2 classes and identify a subset of states that most discriminate the two classes. Afterwards, the final recognition among the two classes is carried out by comparing the *log*-likelihood scores of these chosen states. Since the proposed methodology focuses only on the most discriminative states of the two classes, therefore it enhances the classification confidence as well as overall recognition accuracy with least added complexity. The proposal is demonstrated for character and limited vocabulary word recognition tasks and evaluated on the locally collected Assamese character and word databases. The experimental results are promising over the conventional HMM-based HR system.**

I. Introduction

Online handwriting recognition (HR) is one of the most active areas of research since it facilitates an easiest and efficient way to input data on the computer through a person's handwriting, and has been actively pursued on different scripts (e.g. English [1], [2], Chinese [3], [4], Arabic [5], [6], and Indic [1], [7]). These handwriting-based input systems can pretty much revolutionize the computing of those scripts having large number of characters (e.g. Indic or Chinese) since entering data using a keyboard like interface for these scripts is difficult. Therefore, there have been many efforts in literature, focusing on improvement in recognition of these scripts.

An online HR system typically comprises of two stages: feature extraction and recognition. The first one computes various attributes from the handwriting while the latter one analyses those attributes to recognize it. Among the various techniques used to model online handwriting [1], [2], [7]–[9], the most popular and widely used one is based on hidden Markov model (HMM) [5], [7], [10]–[12]. It has been employed to model stroke [7], [10], character [7], [12] and word [11] on various scripts. Bharath *et al.*, in [7] used HMM to model the strokes of Tamil script and characters of Devanagari script and eventually employed to develop a large vocabulary word recognition system. The work in [10] developed stroke recognition system considering 72 Gurmukhi strokes. Liwicki *et al.*, in [12] employed HMM to model 58 English characters and developed a word recognition system. Samanta *et al.*, in [11] employed HMM to develop holistic Bangla word recognition system. Abdelaziz *et al.*, in [5] developed an HMM-based large vocabulary Arabic word recognition system.

Despite the success of HMM-based approaches, it is worth mentioning that, top-2 outputs of HMM often forms a confusing character pair with a minor difference in their *log*-likelihood scores (particularly for scripts having large similar shape character/word class). Thus, it limits the possibility of faithful classification and thereby prone to be misclassified with the most similar one. As a result, in majority of cases, the correct class appears among top-2 outputs, even when the recognized class is wrong [13]. For instance, consider a confusing pair অ—আ (Fig. 1) from Assamese database (Section II-A). Since the two characters have a similar shape in the beginning, the first few states of the respective HMMs eventually learn same information/characteristics from the data. Now, as the characters are most separable in the end region of the patterns, therefore, the most discriminative HMM states are the last few states that capture the respective class specific characteristic. Accordingly, the consideration of the *log*-likelihood scores only from the discriminative HMM states may lead to the development of an efficient HMM-based online HR system.

In this paper, we propose an HMM-based online HR system where the final classification among top-2 most relevant classes is carried out by considering the scores from the discriminative states. Now, to find most discriminative states in a character pair, we calculate the distance between corresponding states of the two HMMs. The states of GMM-HMM system can be considered as a GMM distribution as the data in each state is modelled by GMM. Therefore, given two sets of the GMM distributions or states (of respective two HMMs), we compute earth mover's distance [14] between the states of the two models. The subset of the states surrounding the highest dissimilarity state is considered as discriminative states and can be utilized to best separate the pair.

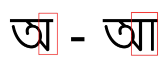

Fig. 1. Represents two similar shape Assamese characters অ and আ. The patterns are well separable in the end region marked by the rectangle.

TABLE I
DESCRIPTION OF CLASSES AND CORRESPONDING NUMBER OF SAMPLES IN ONLINE HANDWRITTEN ASSAMESE CHARACTER AND WORD DATABASES.

Task	# Class	# Samples
Digit	10	3100
Vowel-Consonant	52	13357
CV Unit	95	29891
Word	118	21877

In literature, the reevaluation of top-2 outputs is carried out in a two-stage framework, wherein the top-2 outputs of first classifier are fed to the second stage classifier for refining the outputs [13], [15]–[17]. In this approach, the second stage classifier is designed in pairwise fashion and trained for all confusing pairs. For instance, Sundaram *et al.* in [15] designed a two-stage classifier by employing SVM in both the stages for Tamil word recognition task. The second stage pairwise SVM is designed considering most discriminative region of the confusing pair. Prevost *et al.* in [13] explored a combination of model-based classifier with a probabilistic neural network for English character recognition task. However, in a two-stage system, a separate classifier is trained for each confusing pair. Also, the test example needs to be tested several times for final classification. This adds extra complexity both in training and testing in a two-stage system. On the contrary, in the proposed approach, there is no additional training/testing, this makes the overall framework simple yet improves the recognition performance. Based on the preceding discussions, the following are the major contributions of this paper.

1) Enhancement of classification ability of HMM-based HR system by exploring the discriminative HMM states.
2) Refining the top-2 most relevant outputs in a single stage classification framework.
3) Development of Assamese character and word recognition systems with the proposed approach.

The rest of the work is organized as follows: Section II provides a brief discussion of Assamese script, database and existing works on Assamese script for online HR task. Section III presents the designing of baseline HMM-based HR system. The identification of discriminative states and the overall framework of proposed HR system are elucidated in Section IV. The experimental evaluations of the proposed HR system are carried out in Section V. Finally, we conclude the paper in Section VI by highlighting the future research direction.

II. BACKGROUND

A. Assamese script and database description

Assamese is one of the Indic script and mainly used in north-eastern state of Assam. It is similar to Bangla [18] with two additional characters, both originating from the same Siddham script. The basic alphabets in Assamese script consist of 10 digits, 11 vowels, 41 consonants, 10 vowel modifiers and 2 consonant modifiers. The vowel modifier can combine with consonant and generate new symbols called "consonant with vowel modifier" (CV) unit. Further, the consonant can combine with each other and form new symbol called conjunct. However, all of these generated symbols are not frequently used to write Assamese. In this study, we have considered 10 digits, 11 vowels, 41 consonants and 95 CV units for the experiment. The data is collected from 200 writers using a Lenovo Tablet PC-X230. Each participant is asked to write the characters in isolated fashion in the respective box given in the screen of tablet PC. The data consists of (x, y) coordinate of pen trajectory, pen-down and pen-up status, and writer information. For each character, we have collected two samples from each writer in different sessions. All databases are pruned to remove unintelligible and mislabeled data. Finally, the database is categorized into three group based on shape complexity as (i) the digits (10 characters), (ii) vowels and consonants (52 characters) and (iii) consonants with vowel modifier (CV) units (95 characters).

For online handwritten Assamese word recognition task, we have also created a word database. To create the database, first, we analyze Assamese OCR corpus [18] to find the statistic of mostly used characters (that includes vowels, consonants, conjuncts and CV units). Afterwards, we identify a minimal set of words that cover all most frequent characters. This contains 118 unique words. The database is created by collecting two samples of each word from 100 users in two different sessions. This database is also pruned to remove the unintelligible and mislabeled data. The total number of samples present in each of these databases are given in Table I.

B. Existing studies on Assamese handwriting

Assamese script contains large character set and is written in both discrete and cursive form. The characters are more complex considered to English and the number of pen-ups is also more. These factors increase the challenges in recognition of Assamese handwriting. In the literature, there is a limited number of works on Assamese script. An HMM-based Assamese online handwritten numeral recognition system is developed in [19]. In [20], the authors compared the performance of two HMM-based character recognition systems, which are built at both stroke level and character level. Combination of different recognition system has also explored in various works [21]–[23]. An online and offline recognition system trained using coordinate based and image-based feature, respectively are combined for Assamese numeral recognition in [21]. In [22] and [23] support vector machine (SVM) and HMM are combined for recognition Assamese numerals and characters. The work in [9] developed a convolution neural network based Assamese character recognition system.

III. HANDWRITTEN CHARACTER AND WORD MODELING USING HMM

This section describes a baseline Assamese character and word recognition systems using HMM. It consists of three modules: (i) the preprocessing, where the variations among the samples of same class is reduced, (ii) the feature extraction, where the sequence of (x, y) points are transformed to a sequence of feature vector, and (iii) modeling, where character and word HMMs are built for classification. In the following, we describe the details of each of these steps.

A. Preprocessing

The preprocessing stage is applied to online handwriting to compensates for variations in scale and time among the samples of same class. The procedures employed for preprocessing are smoothing, resampling, and size normalization. To smooth the data, a moving average filter with a window size of 3 points is applied to each stroke. This reduces the noise or jitters in handwriting. Next, we resample each stroke separately to obtain uniform spacing between successive points in the data. Finally, the variation in the size of character sample is eliminated by normalizing the y coordinates of the pattern to [0,1] range by maintaining the aspect ratio. Now, size normalization of word sample is slightly different than the isolated character and described as follows:

In general, Assamese word can be divided into three zones, viz., upper, middle and lower zones by computing two horizontal lines called "headline" and "baseline" (refer to [11]). To calculate these line [12], first, the maxima and minima of y coordinates of each stroke are computed. Afterwards, two linear regressions are performed along the minima and maxima with constraints that the two lines are parallel to each other. The regression lines are then two times performed by eliminating least fitting points. This results in the estimation of a headline (the line fit through maxima) and a baseline (line fit through minima). The height of the middle region (i.e. the region between headline and baseline) is normalized to [0,1] range. The heights of the upper region and lower region are normalized proportionally preserving the original aspect ratio.

B. Feature extraction

Extraction of relevant features is a crucial step for developing any online HR system. For character recognition, a set of fourteen features is computed at each of the preprocessed (x, y) coordinates [1]: preprocessed x and y coordinates (2); first derivative of x and y coordinates (2); second derivative of x and y coordinates (2); writing direction (2); curvature (2); aspect ratio (1); linearity (1); slope (1); context map (1). Let, the final feature vector sequence corresponding to a character sample having T-(x, y) points be denoted as:

$$\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T] \tag{1}$$

Each of the feature vector in $\mathbf{O}$ can be written as $\mathbf{o}_t = [o_{t1}\ o_{t2}\ ...\ o_{td}]^{\mathbf{T}} \in \mathbb{R}^d$, with d equals to fourteen.

For word recognition task, a 16-dimensional feature vector is extracted at each (x, y) point of the word sample which consists of the above mentioned 14 features along with two additional ascender and descender features [2].

C. Modeling

HMM is one of the widely used generative model for characterizing online handwriting. It is characterized by $\lambda = [\Pi, A, B]$, where $\Pi = \{\pi_i\}$ is initial state probabilities, $A = \{a_{ij}\}$ is transition matrix and $B = \{b_i(\mathbf{o}_t) = \sum_{k=1}^{M} w_{ik}\mathcal{N}(\mathbf{o}_t, \mu_{ik}, \Sigma_{ik})\}$ is the observation probability matrix and $i, j = 1\ to\ N$. Here, N is the total number of states and $\mathbf{o}_t$ is the feature vector at point t. $\mathcal{N}$ is Gaussian distribution with mixture weight w_{ik}, mean vector μ_{ik} and covariance matrix Σ_{ik} for the k^{th} component in GMM in state i. To develop a HMM-based character recognition system having C character class, we build C character HMMs denoted by λ_p, $p = [1, 2, ..., C]$. The HMMs are trained with left-to-right topology. The number of states and mixtures in GMM are optimized during training. The class label is decided by maximizing the following: $\hat{c} = \arg \max_{1 \leq p \leq C} p(\mathbf{O}|\lambda_p)$.

For modelling the word, we follow holistic word recognition approach [11] which is conventionally followed for small vocabulary task. Here, the complete word pattern is modelled by a single HMM. The HMM parameters are optimized by following the same procedures as in case of character HMM.

IV. PROPOSED HANDWRITING RECOGNITION SYSTEM

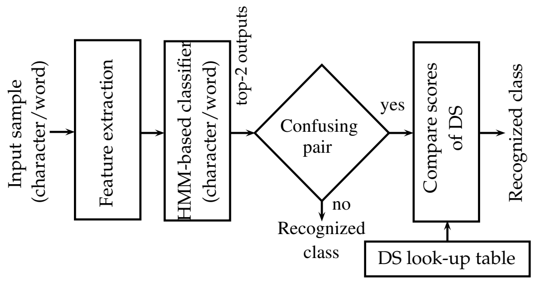

Fig. 2. A block diagram of the proposed online HR system.

Fig. 2 presents the block schematic of the proposed online HR system. The input sample (character/word) is passed through the feature extraction module. It preprocess and converts the data to a sequence of feature vector and then fed to HMM-based HR system for recognition. The HMM selects top-2 most relevant classes among C classes. Now, if the two classes are confusing pair, we analysis the discriminative states of the respective HMMs to refine the output. In the following, we describe the details of discriminative states, recognition technique and present an example illustrating the superiority of our proposal.

A. Selection of discriminative states of HMM

In the proposed system, we need to identify the discriminative states (DS) of a confusing pair. Suppose (c_1, c_2) is one such pair. To identify the DS, we consider the HMMs λ_{c_1} and λ_{c_2} trained for character c_1 and c_2 (the HMM training methodology is given in subsection III-C). Now, in the HMM, initial state distribution π_i is 1 for a left-right HMM. On the other hand, B captures the data distribution through GMMs of each state. Accordingly, the probability distribution B plays the only role for determining the DS of the two HMMs. Now, given a trained model λ_p, we can think of it as sequence of GMMs, i.e., $B^{\lambda_p} = \{b_1^{\lambda_p}, b_2^{\lambda_p}, ..., b_N^{\lambda_p}\}$ where N is total states in λ_p. Formally, for the character pair (c_1, c_2), we can

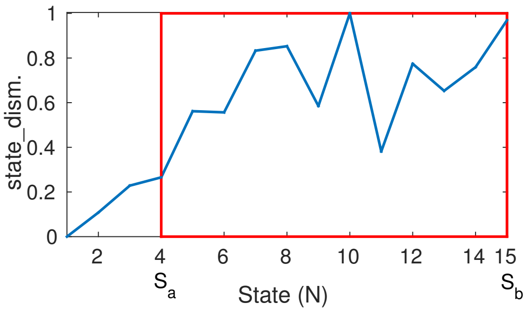

Fig. 3. Depicts the degree of dissimilarity value of character pair অ—আ (shown in Fig. 1) at each i^{th} state. The identified discriminative states (DS) surrounding the highest dissimilarity value is marked by the rectangle where S_a and S_b denotes the starting and ending indices of the DS, respectively.

write the two corresponding sequence of GMMs as : $B^{\lambda_{c1}} = \{b_1^{\lambda_{c1}}, b_2^{\lambda_{c1}}, ..., b_N^{\lambda_{c1}}\}$, and $B^{\lambda_{c2}} = \{b_1^{\lambda_{c2}}, b_2^{\lambda_{c2}}, ..., b_N^{\lambda_{c2}}\}$. Next, we need to compute the distance between the respective b^{λ_p} (i.e. the GMMs) of the two character which will eventually quantify the dissimilarity between the respective states. Now measuring dissimilarity between $b^{\lambda_{c1}}$ and $b^{\lambda_{c2}}$ is not straight forward as done in vector space since b^{λ_j} are the models. To measure the dissimilarity between the two models, we use earth mover's distance (EMD) [14]. The dissimilarity value is calculated for each state as follows:

$$state\_dism^{(\lambda_{c1},\lambda_{c2})}(i) = EMD(b_i^{\lambda_{c1}}, b_i^{\lambda_{c2}}),\ i = 1:N. \quad (2)$$

The variable $state\_dism$ is a vector that contains N dissimilarity value corresponding to N states of the two λ_{c_1} and λ_{c_2}. A smaller value of $state\_dism(i)$ represents that the respective states have similar characteristic whereas a higher value of $state\_dism(i)$ represents that the corresponding states are most discriminative. Accordingly, a set of state (with state dissimilarity value above a threshold) surrounding the i^{th} state that has highest dissimilarity value of $state\_dism$ is considered as the DS. Formally, for a character pair (c_1, c_2), $DS^{(c_1,c_2)} = [S_a\ S_b]$, $1 \leq S_a < S_b \leq N$, where S_a and S_b are the starting and ending indices of DS, respectively. Fig. 3 illustrates the identified DS for character pair অ—আ. The above procedure is repeated for all confusing character pair, and the corresponding DS information is stored in a table called "DS state lookup table" and used at recognition time. It is to be noted that, the list of confusing pair is identified from confusion matrix of the baseline system.

B. Recognition methodology

As shown in Fig 2, for a confusing pair, the DS for the two characters are fetched from "DS lookup table". Next, the *log*-likelihood scores for both classes are recalculated using only the DS. The character that gets highest *log*-likelihood score is then declared as final recognized class. Formally, given a character pair (c_1, c_2), consider that the starting and ending indices of DS are S_a and S_b. The *log*-likelihood score $\mathcal{L}_p$ from the DS of p^{th} character can be calculated as:

$$\mathcal{L}_p = \frac{\phi_{t_v}(S_b) - \phi_{t_u}(S_a)}{t_v - t_u}, \quad p = c_1, c_2 \quad (3)$$

Here, t_u and t_v are the time instances in the optimal state sequence where state S_a begins and state S_b ends, respectively, and,

$$\phi_t(j) \quad = \quad \max_{1\leq i\leq N}[\phi_{t-1}(i) + log\ a_{ij}] + log[b_j(O_t)].$$

The term $\phi_t(j)$ stores the highest possible probability along a single path from observation point o_1 to o_t with state at o_t being j. Let $\mathcal{L}_{c_1}$ and $\mathcal{L}_{c_2}$ correspond to *log*-likelihood scores of character c_1 and c_2, respectively obtained from the DS using (3). The assignment to the class is based on the following criterion:

$$\hat{c'} = arg \max_{p=c_1,c_2} L_p. \quad (4)$$

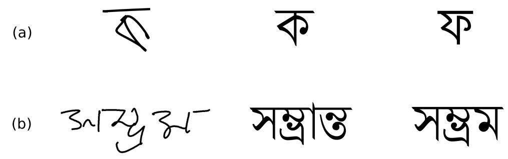

Fig. 4. (a) (From left) First subplot depicts a test sample of character class ক. The second and third panel present the reference patterns of character class ক and ফ, respectively where ক—ফ forms a confusing pair. (b) Represents the same as in (a) for word sample. The first subplot depicts the pattern of a test sample of word সম্ভ্রম, where the next two subplots depict the reference patterns of words সম্ভ্রান্ত and সম্ভ্রম, respectively, noting that সম্ভ্রান্ত—সম্ভ্রম is a confusing word pair.

C. Illustrative experiment

To demonstrate the effectiveness of discriminative states in online HR task, we consider one confusing character pair ক—ফ and one confusing word pair সম্ভ্রান্ত—সম্ভ্রম from Assamese database (Section II-A). The first subplot in Fig. 4(a) presents a test sample of character ক. The next two subplots in Fig. 4(a) presents the reference patterns of ক and ফ, respectively. To conduct the experiment, we build an HMM-based system considering there are only two classes present in the system. Now, to recognize the test character of Fig. 4(a), the baseline system assigns recognition score of -675.40 and -578.88 to ক and ফ, respectively. As a result, the test sample gets misclassified. When we consider the scores from discriminative states of the two models, character ক get -249.15 and ফ get -295.76 *log*-likelihood scores. Since ক got highest score in this case, the test sample is correctly recognized as ক in the proposed approach. Table II summarizes the respective *log*-likelihood scores of baseline and proposed systems. A similar experiment is also conducted for word recognition case. The three subplots in Fig. 4(b) presents a test sample of সম্ভ্রম, reference patterns of সম্ভ্রান্ত and সম্ভ্রম, respectively. The *log*-likelihood scores for recognizing the word of Fig. 4(b) by the proposed and baseline systems are also given in Table II.

TABLE II

RECOGNITION SCORE (*log*-LIKELIHOOD VALUE) OF BASELINE AND PROPOSED SYSTEMS IN CLASSIFYING THE TEST SAMPLES OF FIG. 4.

Test sample	Class	Baseline system	Proposed system
Character	ক	-675.40	**-249.15**
	ফ	-578.88	-295.76
Word	সম্ভ্রান্ত	-2.46×10^{-3}	-2.25×10^{-3}
	সম্ভ্রম	-2.79×10^{3}	$\mathbf{-1.85 \times 10^{-3}}$

V. EXPERIMENT AND DISCUSSION

The performance of the proposed HR system is evaluated for Assamese character and word recognition tasks. The implementation has been done using MATLAB and HTK toolkit.

A. Experimental protocol and system parameter tuning

To conduct character and holistic word recognition experiments, we randomly divided the databases into disjoint training and testing sets with 2:1 ratio in writer independent scenario. This process of random selection of training and testing set is repeated over 5 repetitions, and the performance is reported by averaging the 5 repetitions results. Moreover, the tunable parameters such as HMM states, GMM mixtures etc., are optimized for first round of validation process, and same values are used for remaining 4 rounds. Table III presents the recognition accuracy for character recognition task with different HMM states (with 20 GMMs in each state) for the first round of validation process. Again, the number of (x, y) points in a word are generally more than a single character. Therefore, a higher number of states are required to model a word. The recognition accuracy for word recognition task with different HMM states is also given in Table III. It can be seen that best recognition accuracy is obtained with 15 states for character recognition task and it is 45 states for word recognition task.

B. Character/Word recognition experiment

Table IV presents the character recognition accuracy of the proposed online HR system for different tasks with varying state dissimilarity threshold values (Section IV-A). For comparison, the top-1 and top-2 performances of baseline HMM-based HR system are also reported. It is to be noted that, a lower state dissimilarity threshold value (Section IV-A) select more states whereas a higher threshold value select few states for the reevaluation. Accordingly, with increasing threshold values from 0.1 to 0.3, we get an improvement in recognition performance. The further increment in threshold value degrades the recognition accuracy since it selects only a few states which may not be sufficient for classification. From the table it can see that, a state dissimilarity threshold value of 0.3 results in best recognition accuracy. Further, we can see that there is a notable improvement in performance of the proposed system compared to the baseline system (top-1 performance). For the digit recognition task, since few digit classes form confusing pairs, therefore, the respective improvement is also less. In case of Vowel-Consonant and CV unit recognition tasks, the number of confusing pairs is high which results in higher improvement. Table V shows some of the test samples that are misclassified by the baseline system but has been correctly classified by the proposed system. The recognition result obtained for holistic Assamese word recognition task with baseline and proposed systems are also given in Table IV. It can be observed that the proposed approach has improved the performance significantly. Thus, we can say that the consideration of DS of the HMMs can enhance the recognition performance.

Again, it is to be noted that, since the proposed method refines the top-2 outputs of the baseline system, therefore, the maximum recognition accuracy that can be achieved by the proposed system is equivalent to the top-2 performance of baseline system (Table IV). However, though the performance has been improved, it is less compared to the top-2 performance of baseline system. To analyze the same, we plot some of the test samples in Fig. 5 for which the correct class lies on top-2 outputs, but final recognition is wrong. It can be seen that some of the misclassifications are due to poor handwriting.

TABLE III

RECOGNITION ACCURACY (IN %) WITH VARYING HMM STATES.

# states →	5	7	9	11	13	15	17
Digit	98.46	98.74	98.84	99.03	99.03	99.03	98.96
Vow-Con	93.81	94.76	95.29	95.87	95.87	96.10	96.16
CV Unit	93.92	94.90	95.46	95.83	95.97	96.12	96.02
# states →	15	25	35	45	55	65	75
Word	89.20	92.30	93.30	93.69	93.65	93.47	93.50

TABLE IV

RECOGNITION ACCURACY (IN %) OF THE PROPOSED SYSTEM WITH VARYING STATE_DISM. THRESHOLD VALUE FOR ASSAMESE CHARACTER AND HOLISTIC WORD RECOGNITION TASKS. THE PERFORMANCE OF THE BASELINE SYSTEM IS ALSO REPORTED FOR COMPARISON.

Task	Baseline system		Proposed system (Top-1) state_dism. threshold value				
	Top-1	Top-2	0.1	0.2	0.3	0.4	0.5
Digit	98.88	99.64	99.10	99.21	99.28	99.04	98.72
Vow-Con	96.04	98.75	96.46	96.62	96.86	96.39	95.58
CV Unit	95.89	98.13	96.51	96.64	96.84	96.26	95.35
Word	93.91	96.11	94.36	94.68	95.07	94.48	93.51

TABLE V

ILLUSTRATIONS OF FEW TEST SAMPLES FROM ASSAMESE CHARACTER DATABASE THAT ARE MISCLASSIFIED BY BASELINE SYSTEM BUT PROPOSED METHOD CORRECTLY CLASSIFIES THEM.

Test sample		Baseline system	Proposed system
Character	Pattern	Recog. character	Recog. character
২		১	২
গী		ণী	গী
ম		স	ম
খৃ		খূ	খৃ

TABLE VI
PROCESSING TIME (IN MS) OF BASELINE AND PROPOSED SYSTEMS.

System↓ Task→	Digit	Vow-Con	CV Unit	Word
Baseline	13.10	138.42	244.50	2863.2
Proposed	13.16	138.50	244.59	2864.3

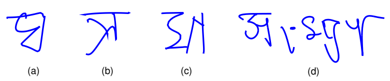

Fig. 5. Assamese (a) “ঘ” misclassified as “ধ; (b) “স” misclassified as “ম; (c) “ঘা” misclassified as “মা; (d) “সংখ্যা” as “উনপঞ্চাশ”.

However, few samples are correctly written but misclassified for which further exploration can be done in future work.

We also compare the average processing time of baseline and proposed systems for the different dataset in Table VI. It can be seen that the processing time of the proposed system is slightly higher. This is because, the proposed system reevaluates the decision of HMM classifier for the confusing pair and it adds extra computation to the system, though it is almost insignificant.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel approach to online handwriting recognition employing hidden Markov model (HMM) framework. The proposed approach is based on the fact that since mostly the correct class systematically appears among top-2 outputs of HMM, therefore, we refine the top-2 classes by analyzing the discriminative states (DS) of the two HMMs. Unlike the traditional way of building a second-stage classifier for refining top-2 outputs, the proposed approach considers the scores from few HMM states that well separate the top-2 characters. Thus, it does not include an extra testing phase for refinement yet enhance the recognition accuracy notably when evaluated for Assamese character and word recognition tasks.

We now present the different research directions that can be explored in future. In the current work, the effectiveness of analyzing DS is shown in GMM-HMM framework. The same concept can be explored and verified in the state-of-the-art deep neural network (DNN)-HMM framework. Further, DS can be explored for large vocabulary word recognition task.

REFERENCES

[1] S. Mandal, S. R. M. Prasanna, and S. Sundaram, “GMM posterior features for improving online handwriting recognition,” *Expert Systems with Applications*, vol. 97, pp. 421 – 433, 2018.
[2] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and 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, May 2009.
[3] X.-Y. Zhang, Y. Bengio, and C.-L. Liu, “Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark,” *Pattern Recognition*, vol. 61, pp. 348–360, 2017.
[4] X.-D. Zhou, D.-H. Wang, F. Tian, C.-L. Liu, and M. Nakagawa, “Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 10, pp. 2413–2426, Oct 2013.
[5] I. Abdelaziz, S. Abdou, and H. Al-Barhamtoshy, “A large vocabulary system for Arabic online handwriting recognition,” *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 1129–1141, Nov 2016.
[6] N. Tagougui, M. Kherallah, and A. M. Alimi, “Online Arabic handwriting recognition: a survey,” *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 209–226, 2013.
[7] A. Bharath and S. Madhvanath, “HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 670–682, April 2012.
[8] J. Sternby, J. Morwing, J. Andersson, and C. Friberg, “On-line Arabic handwriting recognition with templates,” *Pattern Recognition*, vol. 42, no. 12, pp. 3278 – 3286, 2009.
[9] B. Graham, “Spatially-sparse convolutional neural networks,” *arXiv preprint arXiv:1409.6070*, 2014.
[10] S. Singh, A. Sharma, and I. Chhabra, “Online handwritten Gurmukhi strokes dataset based on minimal set of words,” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, no. 1, p. 1, 2016.
[11] O. Samanta, U. Bhattacharya, and S. Parui, “Smoothing of HMM parameters for efficient recognition of online handwriting,” *Pattern Recognition*, vol. 47, no. 11, pp. 3614 – 3629, 2014.
[12] M. Liwicki and H. Bunke, “HMM-based online recognition of handwritten whiteboard notes,” in *Proceedings of international workshop on frontiers in handwriting recognition*, 2006, pp. 595–599.
[13] L. Prevost, L. Oudot, A. Moises, C. Michel-Sendis, and M. Milgram, “Hybrid generative/discriminative classifier for unconstrained character recognition,” *Pattern Recognition Letters*, vol. 26, no. 12, pp. 1840 – 1848, 2005, Artificial Neural Networks in Pattern Recognition.
[14] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.
[15] S. Sundaram and A. G. Ramakrishnan, “Performance enhancement of online handwritten Tamil symbol recognition with reevaluation techniques,” *Pattern Analysis and Applications*, vol. 17, no. 3, pp. 587–609, 2013.
[16] C. Zanchettin, B. Bezerra, and W. Azevedo, “A KNN-SVM hybrid model for cursive handwriting recognition,” in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, June 2012, pp. 1–8.
[17] J. Milgram, R. Sabourin, and M. Cheriet, “Two-stage classification system combining model-based and discriminative approaches,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1, Aug 2004, pp. 152–155 Vol.1.
[18] S. Ghosh, P. K. Bora, S. Das, and B. B. Chaudhuri, “Development of an Assamese OCR using Bangla OCR,” in *Proceeding of the Workshop on Document Analysis and Recognition*. ACM, 2012, pp. 68–73.
[19] G. S. Reddy, B. Sarma, R. K. Naik, S. R. M. Prasanna, and C. Mahanta, “Assamese online handwritten digit recognition system using hidden Markov models,” in *Proceeding of the Workshop on Document Analysis and Recognition*, ser. DAR ’12. New York, NY, USA: ACM, 2012, pp. 108–113.
[20] H. Choudhury, S. Mandal, S. Devnath, Prasanna, SRM., and S. Sundaram, “Comparison of Assamese character recognizer using stroke level and character level engines,” in *Proceedings of national conference on communications*, Feb 2015, pp. 1–6.
[21] G. Reddy, P. Sharma, S. Prasanna, C. Mahanta, and L. Sharma, “Combined online and offline Assamese handwritten numeral recognizer,” in *Communications, 2012 National Conference on*, Feb 2012, pp. 1–5.
[22] B. Sarma, K. Mehrotra, R. Krishna Naik, S. Prasanna, S. Belhe, and C. Mahanta, “Handwritten Assamese numeral recognizer using HMM and SVM classifiers,” in *Proceedings of National Conference on Communications*, Feb 2013, pp. 1–5.
[23] H. Choudhury, S. Mandal, S. Devnath, S. R. M. Prasanna, and S. Sundaram, “Combining HMM and SVM based stroke classifiers for online Assamese handwritten character recognition,” in *2015 Annual IEEE India Conference (INDICON)*, Dec 2015, pp. 1–6.