Segmentation Free Bangla OCR using HMM: Training and Recognition

Md. Abul Hasnat, S.M. Murtoza Habib, Mumit Khan
BRAC University, Bangladesh
mhasnat@gmail.com, murtoza@gmail.com, mumit@bracuniversity.ac.bd

Abstract

The wide area of the application of HMM is in Speech Recognition where each spoken word is considered as a single unit to be recognized from the trained word network. Using this concept some research has been done for character recognition. In this paper, we present the training and recognition mechanism of a Hidden Markov Model (HMM) based multi font supported Optical Character Recognition (OCR) system for Bangla character. In our approach the central idea is separate HMM model for each segmented character or word. We emphasize on word level segmentation and like to consider the single character as a word when the character appears alone after segmentation process is done. The system uses HTK toolkit for data preparation, model training from multiple samples and recognition. Features of each trained character are calculated by applying Discrete Cosine Transform (DCT) to each pixel value of the character image where the image is divided into several frames according to its size. The extracted features of each frame are used as discrete probability distributions that will be given as input parameter to each HMM model. In case of recognition a model for each separated character or word is build up using the same approach. This model is given to the HTK toolkit to perform the recognition using Viterbi Decoding. The experimental result shows significant performance.

1. Introduction

Hidden Markov Models (HMM) have been used in a wide range of NLP applications, ranging from continuous speech recognition to handwriting recognition. [1,2]. We use the concept of applying HMM Technique for speech recognition into our training and recognition methodology of OCR where the similar approach is used in some research of building language independent OCR system [1] and High accuracy off-line cursive handwriting recognition [2]. We follow the concept of Speech Recognition where each word is considered as a single unit to be recognized from a sentence uttered. In our approach, we create a model for each individual character or word where the character is considered as a word only at the time when it segmented separately from a segmented line at the time of recognition. Here we give emphasize on word level segmentation technique because in case of Bangla Optical Character Recognition segmentation is a significant issue that is still a vast challenge to resolve accurately. In Bangla script, the position of the dependent vowels and diacritics marks and the existence of touching characters in the scanned document make the task of segmenting the words into character level very complex. Many works has been done for character level segmentation [3 - 7]. Some approach can separate the character successfully but in application level due to the presence of noise, accuracy of performance and considering the cost of computation we choose word level segmentation approach. We should note that this is the first reported attempt at creating a HMM based segmentation free OCR for Bangla script.

We use HTK Toolkit for implementing our application after completing the preprocessing and feature extraction steps. Preprocessing includes binary image conversion, noise removing from image, skew correction, segmentation that includes line separation, word separation and character separation. The segmented portion is then converted into image array that contains binary values for each pixel. Note that in our approach the segmented character is obtained by considering the image boundary not the connected component approach. The next step is to divide the image array into certain number of frames that will be considered as the number of states for the HMM model of the segmented image. Then Feature extraction is performed using Discrete Cosine Transform (DCT) calculation on each individual pixel of each frame that modifies the pixel value in each frame accordingly. These newly calculated pixels value will be considered as discrete probability distributions which will be given as input parameter to each HMM model.

HTK Toolkit does the procedure of creating a particular model from the number of samples that is provided as training input and then save this model description into appropriate files. Based on the number of samples the model parameters (means and variance) will be estimated by internal command of HTK Toolkit. However, for recognition a temporary HMM model is build from the word or character image using the same procedure described above. Then the recognition is performed using Viterbi decoding by HTK Toolkit.

The post processing includes taking the appropriate Unicode characters from the model database against each model name that is determined by the recognition tools of HTK. Then these characters are stored and written into file sequentially.

The flow chart for training and recognition is shown in Figure 1 that clearly visualizes the actual procedure of training and recognition.

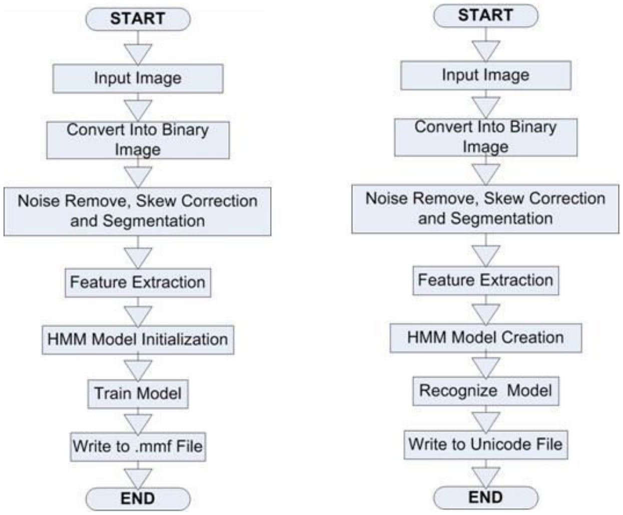


Figure 1: Flow chart of training and recognition procedure

2. Related Works

Varieties of different approach have been applied for the recognition process of OCR. The method described in [3] uses a combination of template and feature matching technique to recognize the pre-segmented character shapes. The method in [4] uses a structural-feature-based tree classifier to recognize the basic character shapes; in the case of compound characters, feature based tree classifier is initially used to separate these into small groups and then an template matching approach is employed to recognize each individual character.

Neural Network approach has been used for classification and recognition where training and Recognition phase of the neural network has been performed using conventional back propagation algorithm at [7] and using multi-layered feed-forward back propagation neural network at [8].

The continuous speech recognition provided the initial impetus for using HMMs in language processing, and the Hidden Markov Model Toolkit (HTK) was developed as a result for the speech community. [13] Since then however, HMMs have been applied to many of the NLP problems, including character recognition. [11][12] The system extracts a set of simple statistical features from each frame and then injects the sequence of the feature vectors to the Hidden Markov Model Toolkit (HTK). The process has significant similarity with Continuous Speech Recognition (CSR) system.

Hidden Markov Models (HMM) are now widely used for handwriting recognition for a number of scripts. [2,9,10] One approach is to use one HMM per word in the vocabulary, and the recognition process then matches the test word against all the models. Another is to use one model for the entire language. There has been recent efforts in building language independent OCRs by modeling each character as an HMM. [1] This implies that the fundamental models are not tied to a particular script or language, and can thus be applied to many by adapting the pre- and post-processing stages.

2. Overview

The entire procedure of recognition and training can be break down into two parts based on HTK Toolkit point of view. The first part consists of preprocessing up to Feature calculation and the second part consists of HTK Toolkit operation that is different for training and recognition process.

2.1. Segmented image to feature calculation

Here I assume that I already have the segmented image representing a subsequence of a word and the image is already converted to binary image. Let take a segmented character and a segmented word that is shown in Figure 2.

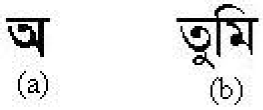


Figure 2: (a) Segmented character ‘soreo’; (b) Segmented word ‘tumi’

2.1.1. Frame calculation. Now from these images number of frame will be calculated. In our approach, we choose the frame width to be 8 and the frame height to be 90. The frame width and height is chosen according to our statistical analysis. Based on the frame width and height we divide the segmented image into several frames. The size of mean and variance vector is also determined from the frame width and height. For example the number of frame of the segmented character soreo is 3 and segmented word tumi has 6 frames. Number of frame is most important because it determines the number of states of the HMM model. So we can say that the number of states for hmm model soreo is 3 and tumi has 6 states. The above discussion is illustrated in Figure 3 and Figure 4.

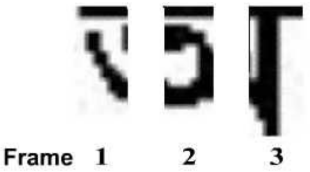


Figure 3: segmented character "soreo" with 3 frames

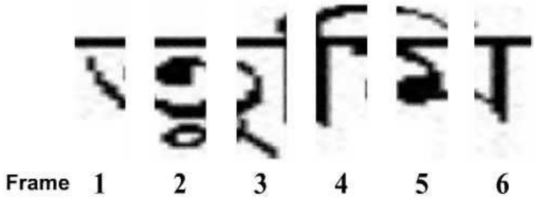


Figure 4: segmented word "tumi" with 3 frames

2.1.2. Feature Calculation. In this phase each frame is taken separately and then the feature calculation is performed on each individual pixels of the frame by applying Discrete Cosine Transform (DCT).

2.1.2.1. Discrete Cosine Transform (DCT). The Discrete Cosine Transform (DCT) converts a continuous signal into its elementary frequency components, and as such, closely related to the Discrete Fourier Transform (DFT). The DCT of an image can represent it as a sum of sinusoids of varying magnitude and frequencies. Because the DCT of an image captures the most visually significant information in just a few coefficients, it is widely used in image algorithms such as compression. DCT of an image is given by the following

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \le p \le M-1 \\ 0 \le q \le N-1 \end{matrix}$$

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases} \qquad \alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

where A and B are the input and output images of M rows and N columns respectively. The DCT is a linearly separable transformation, so computing a two-dimensional DCT can be done with one-dimensional DCTs along each dimension.

After applying DCT on the pixels values of each frame the updated information is stored into a file which will be given as hmm model input for the particular character or word that is considered to be trained.

Up to this stage the image processing methodology is exactly same for both the operation of training and recognition. The only difference is that, training mechanism uses a certain number of samples for modeling a particular character or word that is used for estimating model parameters in HTK Toolkit but recognition mechanism create model only for the particular image character or word to be recognized.

So at this stage we can create HMM model for the character "soreo" and word "tumi" which is shown in Figure 5 and Figure 6.

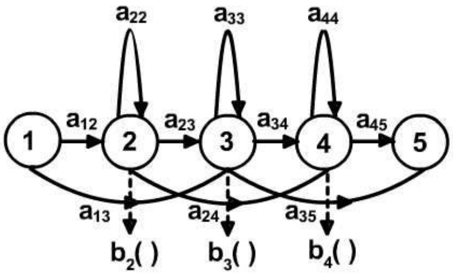


Figure 5: HMM model for character "soreo"

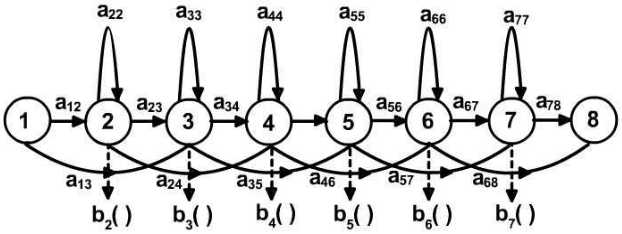


Figure 6: HMM model for word "tumi"

These models clearly prove the description of the framing and feature extraction methodology described above. We can see from these models that the segmented character "soreo" has 3 states without the start and end state that is common to each model. Similarly the segmented word has 6 states without the start and end states. Now the HMM model components (number of states and calculated feature values) for the segmented character and word constructed in this stage is ready to be given as HTK Toolkit input for initializing the HMM model or performing the recognition task.

2.2. HTK Toolkit Operations for Training and Recognition

2.2.1. The HTK Toolkit. The Hidden Markov Model Toolkit (HTK) was initially designed for continuous speech recognition technology development; in recent years, it has been used for everything from speech synthesis and character recognition to DNA sequencing. See [13] for more details on HTK.

Parts of HMM modeling using HTK Toolkit are divided into four phases:

1. Data Preparation
2. Model Training
3. Pattern Recognition
4. Model Analysis

2.2.1.1. Data Preparation. In the Data Preparation stage, we first define a word network of word-to-word transition using HTK's Standard Lattice Format (SLF). HTK also provides a grammar definition format, and an associated HParse tool, that can be used to build this word network automatically. We now create a sorted list of the character or sequence of characters to be trained. Then we use the tool HSGen to generate the prompts for test sentences. Note that the data preparation stage is required only for recognition purpose. It has absolutely no usage for the training purpose.

2.2.1.2. Model Training. The first task is to define a prototype for the HMM model to be trained, which includes the model topology, and transition and output distribution parameters. This task will of course depend on the number of states and the extracted feature of each character or word. In our application, the observation state is finite for each frame (8 x 90 = 720 states), so we use discrete distributions. The HMM model parameters contain the probability distribution or estimation of each model, which we now have to initialize. By far the most prevalent probability density function is the Gaussian probability function that consists of means and variances and this density function is used to define model parameters by the HTK. For this we have to invoke the tool HInit. After the initialization process is completed the HMM model is written into .mmf file that contains all the trained models and is used in recognition.

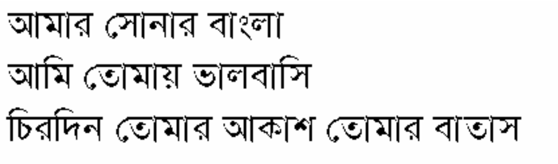
2.2.1.3. Pattern Recognition. Comparative to the training, recognition is much simpler. To complete this task we have to create a HMM model of the character or word image. Then this model will match with all the HMM models and the most likely model will be given as output. To perform this task using HTK we have to invoke the recognition tool HVite that uses the word network describing the allowable word sequence build up from task grammar, the dictionary that define each character or word, the entire list of HMMs and the description of each HMM model. HVite is a general-purpose Viterbi word recognizer. It will match a HMM model against a network of HMMs and output a transcription for the recognized model into a Master Label File .mlf.

After the recognition process is completed, the model name is read from the Master Label File (.mmf) and the associated Unicode character for the recognized model is written to the output file.

3. Performance Analysis

In our approach the recognizer performance is a function of the number of trained characters and words. Usually the recognizer does not give any transcription as output if the hmm model for the character or word to be recognized not likely to the trained models of the system. In some cases the recognizer give wrong output when the HMM model to be recognized not trained previously and there exists a similar type model in the system. In such case HTK output a transcription to which the model is most likely that means when the score of the model exceeds the threshold value. So we can say that the recognizer produce maximum performance when the system is trained with a large training corpus.

Here we start with an example that shows the performance measurement of the recognizer. The test image to be recognized is shown in Figure: 8.



আমার সোনার বাংলা
আমি তোমায় ভালবাসি
চিরদিন তোমার আকাশ তোমার বাতাস

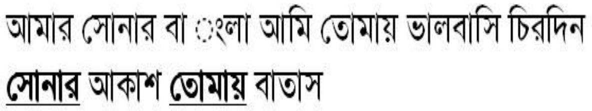
Figure 8: Test Image for measuring the performance of the recognizer

Assume that we have a small training corpus is with the models listed in Table 1.

Table 1: List of existing training models

Word	Model Name	Unicode Sequence
আমার	h0800	0986, 09AE, 09BE, 09B0
সোনার	h0801	09B8, 09CB, 09A8, 09BE, 09B0
বা	h0802	09AC, 09BE
ংলা	h0803	0982, 09B2, 09BE
আমি	h0804	0986, 09AE, 09BF
তোমায়	h0805	09A4, 09CB, 09AE, 09BE, 09DF
ভালবাসি	h0806	09AD, 09BE, 09B2, 09AC, 09BE, 09B8, 09BF
চিরদিন	h0807	099A, 09BF, 09B0, 09A6, 09BF, 09A8
আকাশ	h0808	0986, 0995, 09BE, 09B6
তাস	h0809	09A4, 09BE, 09B8

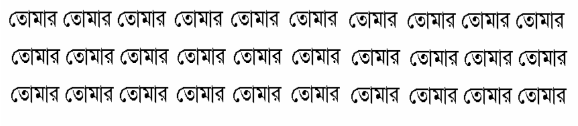
We can clearly observe from Figure 8 and Table 1 that the word তোমার is not trained in the existing corpus. So now the recognizer is not supposed to recognize it. Instead, it will do not provide either any transcription or it can output the transcription of that model which is most likely or the model that is score meet the required threshold. The output from the recognizer with this existing corpus is given Figure 9.



আমার সোনার বা ংলা আমি তোমায় ভালবাসি চিরদিন
সোনার আকাশ **তোমায়** বাতাস

Figure 9: Performance of the recognizer without properly trained

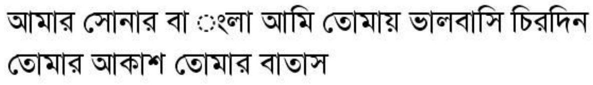
Here we see that the recognizer's output is improper without the training of the word তোমার. In the output we observe that the recognizer output two different words in place of তোমার (In figure: 9 the bold and underline words). So it is proved that without the training the performance is not accurate. To improve performance, the word তোমার have to be trained. Let the model name for the word তোমার is h0810, the Unicode sequence will be 09A4, 09CB, 09AE, 09BE, 09B0. The trained image with thirty samples is shown in Figure 10.



তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার
তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার
তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার তোমার

Figure 10: Trained image of তোমার with thirty samples

Now our training corpus contains all required models to recognize the test image of Figure: 8 with maximum performance. Let us run the recognizer again and observe the performance. The observation result is shown in Figure 11.



আমার সোনার বা ংলা আমি তোমায় ভালবাসি চিরদিন
তোমার আকাশ তোমার বাতাস

Figure 11: Performance of the recognizer with proper training

The result shows that the performance is improved to maximum level after proper training.

4. Conclusion

Here we present the training and recognition mechanism of the Hidden Markov Model (HMM) based Optical Character Recognizer (OCR) for Bangla character. The system is mainly divided into two parts; one is preprocessing up to feature calculation and the other part consists of HTK Toolkit operations that are used to initialize the HMM model from a certain number of samples and to recognize a particular model. Multi font character recognition depends on training corpus with different font. It shows significant performance for trained character. Performance of the recognizer depends on the number of trained models.

5. References

[1] Z. Lu, I. Bazzi, A. Kornai, J. Makhoul, P. Natarajan and R. Schwartz, “A Robust, Language-Independent OCR System”, Proc. *SPIE Vol. 3584, 27th AIPR Workshop: Advances in Computer-Assisted Recognition*, 1999, pp. 96-104.

[2] W. Wang, A. Brakensiek, A. Kosmala and G. Rigoll, “HMM Based High accuracy off-line cursive handwriting recognition by a baseline detection error tolerant feature extraction approach”, *7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2000.

[3] U. Pal, B.B. Chaudhuri, “OCR in Bangla: an Indo-Bangladeshi Language”, *Pattern Recognition, Vol. 2 - Conference B: Computer Vision & Image Processing, Proceedings of the 12th IAPR International*, 1994.

[4] B.B. Chaudhuri, U. Pal, “An OCR System to Read Two Indian Language Scripts: Bangla and Devnagari (Hindi)”, IEEE Computer Society, 1997.

[5] A. Bishnu, B.B. Chaudhuari, “Segmentation of Bangla Handwritten Text into Characters by Recursive Contour Following”, *IEEE Computer Society*, 1999.

[6] U. Garain, B.B. Chaudhuri, “Segmentation of touching characters in Printed Devnagari and Bangla Scripts using Fuzzy Multifactorial Analysis”, *IEEE Computer Society*, 2001.

[7] J.U. Mahmud, M.F. Raihan and C.M. Rahman, “A Complete OCR System for Continuous Bangla Characters”, *IEEE TENCON-2003: Proceedings of the Conference on Convergent Technologies for the Asia Pacific*, 2003.

[8] A.A. Chowdhury, E. Ahmed, S. Ahmed, S. Hossain and C.M. Rahman, “Optical Character Recognition of Bangla Characters using neural network: A better approach”, *2nd International Conference on Electrical Engineering (ICEE)*, Khulna, Bangladesh, 2002.

[9] A. Kundu, P. Bahl and Y. Yang, “Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach”, *Journal of the Pattern Recognition Society*, Pergamon Press, Vol. 22, No. 3, 1989, pp. 283-297

[9] W. Wang, A. Brakensiek, A. Kosmala, and G. Rigoll. “HMM based high accuracy off-line cursive handwriting recognition by a baseline detection error tolerant feature extraction approach”, In *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, Amsterdam, 2000, pp. 209-218.

[10] J. Doménech, A.H. Toselli, A. Juan, E. Vidal, and F. Casacuberta, “An off-line HTK-based OCR system for isolated handwritten lowercase letters”, In *Proc. of the IX Spanish Symposium on Pattern Recognition and Image Analysis, volume II*, Benicàssim, Spain, May 2001, pp. 49-54.

[11] S.A. Al-Qahtani and M.S. Khorsheed, “A HTK-Based System to Recognise Arabic Script”, in *Proc. 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, Marbella, Spain, ACTA Press, 2004.

[12] S.A. Al-Qahtani and M.S. Khorsheed, “An Omni-font HTK-Based Arabic Recognition System”, in *Proc. 8th IASTED International Conference on Artificial Intelligence and Soft Computing*, Marbella Spain, ACTA Press, 2004.

[13] The HTK Book (for HTK Version 3.3) available at http://htk.eng.cam.ac.uk/docs/docs.shtml