Unconstrained Handwriting Recognition into Text using Parser

Pinagadi, Venkateswararao  
Research Scholar, Sathyabama University, Chennai, India.

S. Murugavalli  
Professor & Head., Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India.

Abstract
Reclaiming of exact keyword from the handwriting document can be possessed by keyword spotting. Repossessing of handwritten text, letter and manuscript from the historical written books in the field of neural networks based on unconstrained handwriting recognition is made through novel keyword spotting. This method will claim the exact word from any spot of manuscripts or any form of text in different writing style of various writers and convert them into a formatted image in the existing system. So follow up to this, we encountered some difficulties in keyword spotting. Owing to the analysis of reclaiming task followed by keyword spotting in underlying neural network on the produced transcription as we plan to enhance the method for converting the handwritten formatted image into text using our technique called handwritten recognition into text using parser method performs with more reliable and fast keyword spotting can be performed.

Keywords: Multilayer cluster neural network (MCNN); DTW (Dynamic Time Warping); Artificial Neural Network (ANN); K-Nearest Neighbor (KNN); Support Vector Machine (SVM).

Introduction
Recognition of unconstrained handwritten characters is one of the most appealing and challenging topics in automatic document analysis and processing. In recent years, Optical Character Recognition (OCR) has been greatly developed because of the prevalence of Internet and multimedia techniques. In OCR applications, handwritten character recognition, especially digit recognition is dealt with in postal mail sorting, bank check processing, data entry, etc. In recent decades many researchers worked on this topic. Most of the works related to handwritten character recognition are done in English, Chinese, Japanese and Arabic. However, some preface work has also been done on Indian scripts. A multistage scheme for the recognition of handwritten Bengali characters is reported in [13]. An analysis of the Bengali character set has been carried out to separate specific high-level features that can help in forming smaller sub-groups within the characters. A two-stage classification system for recognition of handwritten Devnagari numerals is described in [3]. The paper extracts horizontal and vertical strokes from hand written numeral images and work out a shape feature vector for each such stroke. Hidden Markov Models (HMM) and Multi-Layer Perception (MLP) classifiers were used for classification purpose. Character recognition has become an acute research area in recent years for the ease of access of computer applications. Numerous approaches have been projected For character recognition and extensive successes have been reported. Traditional handwritten character recognition techniques enable a computer to take delivery of and interpret intelligible handwritten input from sources such as papers, documents, touch-screens or pictures [4]. Some works have been done to recognize handwritten

The outline of this paper is as follows: Section II describes existing character recognition techniques. Section III presents the proposed Handwritten Recognize system. Comparison of Various Techniques in Handwritten Recognition System where discussed in Section IV. Finally, concluding remarks are explained in Section V.

Related Works
B. V. Dhandra et al. [1] have performed series of operations like binarization, filtering and morphological operations as part of preprocessing for recognizing Kannada handwritten vowels. Here gray scale image is binarised using Otsu’s global thresholding and median filter is functional for removing noise due to erratic hand movements and digitization inaccuracies. After this, morphological operations are performed for removing isolated locations and spikes around the end of the vowels. Even if median filter overcomes the boundaries of the linear filters it may cause removal of corners and threads, blurring of texts in the documents. In order to overcome these limitations KanikaBansaletal. [2] Proposed an algorithm named K-algorithm, which is the combination of filtering and binarization. For reducing within-class variation of shape of unconstrained handwritten numerals, Suzete E. N. Correia et al. [3] have used slant and size normalization as preprocessing methods. NtugasNikolaos et al. [4] proposed four Binarization methods such as Otsu’s, Niblack, Sauvola’s and Bernsen’s for discriminating degraded and very poor quality gray scale Byzantine manuscript from the background based on pure thresholding and filtering. As per his work Otsu’s provide better result as compared to other three methods. In [5] Xiang Zhao et. al used morphological operations (thinning, skeletonization etc.) for recognizing characters from map.
In [8] Joohun Lim et al. have presented a comparative analysis of scale invariant feature extraction using different wavelet bases. This paper shows that Gabor wavelet basis function extracts image features more efficiently than Haar, Daubechies basis function in wavelet bases. In [9] Lee et al. have extracted features using Haar orthogonal wavelet at one resolution level. When Suzete E. N. Correia et al. [3] used Cohen-Daubechies-Feauveau (CDF) family of bi-orthogonal spline wavelets as feature extractor he finds that the recognition rate obtained with CDF 3/7 is superior to that of Haarwavelet. I K Pathan et al. have proposed an off-line approach for handwritten isolated Urdu characters in their work mentioned in [10]. Authors have used moment invariants (MI) feature to recognize the characters. MI features are well known to be invariant under rotation, translation, scaling and reflection. VKarthikeyan [11] proposed a system for recognizing Tamil characters. In his paper, the character image skeletonized using Hilditch’s algorithm and features are extracted based on the concept of image moment which is the weighted average of entire pixel intensities. Here four features are extracted from each of the character, the equation of which is derived from Hu’s moment invariants [12]. Gradient features based method is discussed in [13] by Ashutosh et al. where gradient vector is calculated at each pixel by means of sobel operator and then image is divided into different zones. Then strength of gradient is accumulated in eight standard directions in each zone. Zahedia et al used SIFT algorithm in his paper [14] for recognizing the Arabic characters. Here the preprocessed image is passed through SIFT algorithm in order to extract the features. As the first step of this algorithm, the candidate key points are found out. Sreeraj M et al [15] presented an approach for on-line grantha character recognition. Here features such as time domain, writing direction and curvature are extracted. It is a method for finding areas of an image that match to a template image, the image patch to be compared with the input image. According to Oivind Due Trier et al [16], Jasbir Singh et al. [17] have used Artificial Neural Network as classifier in his work for Devanagari character recognition. ANN consist of number of processing units called neurons distributed in three layers namely input, hidden and output that communicate with one another over a large number of weighted connections. Seong-Whan Lee et al [9] have used Multilayer cluster neural network (MCNN) as classifier for recognizing handwritten numerals. In MCNN the units in each layer are clustered and each cluster is fully connected to a corresponding cluster in following layer independently. Based on the concept of MCNN, Suzete E. N. Correia et al. [3] used three layer cluster neural network for training and classification. In [18] D K Patel used Euclidean Distance Metric in combination with artificial neural network for classification. For each unknown input pattern vector, distances to the mean vectors which characterize each pattern class are computed by EDM. For Kannada character recognition B V Dhandra et al[1] have used KNN as classifier. In [15] Sreeraj M et al. have used KNN classifier where DTW (Dynamic Time Warping) is used as a distance metric in order to enhance recognition rate. As per Jasbir Singh et al [17] KNN is simplest of all classifier for predicting the class of the test sample. For recognizing Devanagari characters Rajneesh Rani et al [13] have used SVM classifier by taking gradient based feature as its input. In [10] I K Pathan have also used SVM for training purpose based on moment invariant features of Urdu characters. As per the comparative analysis by Jasbir Singh et al [17] on Devanagari characters, SVM is a very useful technique for data classification as compared to ANN and KNN.

**Proposed Handwritten Recognition Technique**

Renovation of handwritten characters is significant for building manuscripts into a device identifiable form as a result that it can be effortlessly accessed and preserved the content of data’s. Existing conversion takes places with unconstrained handwritten CTC token passing that may convert the handwritten manuscripts into readable images. In our proposed work we tend to convert the handwritten manuscripts formatted image into text format. These may ready to lend a hand in such a way that easy understanding and immediate recognition. Primarily the hand written books formatted images are taken as inputs and then our proposed mechanism “Unconstrained handwriting recognition into text using parser” will analyze the formatted image, again sent it to recognizer via parser. The parser determines the total formatted image characters and devises them into recognizer after bit by bit analysis. The recognizer can convert the image into text, provides the text output with image display result. The expected output is in a recognized text in a file.

![Figure 1: Proposed Handwritten Recognition Technique](Image & Display)
The proposed CTC parser architecture has been chosen in order to take advantage of both generative and discriminative frameworks. A CTC network processes the input sequence to complete posterior character probabilities at every step. Then the prospects of each characters are fed to the HRT stage (using class posteriors in place of the character likelihood computed by Gaussian Mixtures Models in the traditional HRT framework) to perform the alignment of the spotting model. We now describe the HRT line spotting models that enable us to spot either word.

While converting the handwritten characters identification of character is essential. As we declared each character recognition with specification’s like ([a]-[z]) for lower cases, ([A]-[Z]) for upper cases letter. Recognition of number’s and special character also taken into part with ([0]-[9]) for digits conversion and (*) operators with special characters. These must spot a character between 0 and ∞ times or from 1 to ∞ times processed its recognition. The spacing between the character differs by pixels, so identification of pixel using character lines made with ([(#[^a]-[^z]*)#]) and equalizing the pixel format using “(Le[a]-[z]*), (Le[^a]-[^z]*#)”, or both ([(#[^A]-[^Z])o[^a]-[^z]*#]). Those are the initial level of process for identifying and translating the characters from the handwritten pages. This is a step by step conversion, each character can be processed under above method and stored, later conversions continued with the subsequent steps. Every character can be identified for whether it is a character or letter, recognized wisely with the use of this algorithm. Finally all the recognized characters stored and relocated specifically.

CTC parsing algorithm:
Begin
CTC Parser()
Lower Cases (#[^a]-[^z]*#) //identify the lowercase letters
Upper Cases (#[^A]-[^Z]*#) or Digits (#[^0]-[^9]*#) // identify the upper case letters
using * or + operators
(Spotting between 0 and ∞ times a character, or spotting between 1 and ∞ times a character) #[^0][^9]*#
Lower cases ([(#[^a]-[^z]*)#])
Beginning of a line ([(#[^a]-[^z]*)#])
Identify the pixel ([(#[^a]-[^z]*)#])
Equal pixel format (Le[^a]-[^z]*#), or both ([(#[^A]-[^Z])o[^a]-[^z]*#]).
Convert->text format
S> store
Process->Continue
(#[^0][^9]*#) or word beginning by one upper case
(^[A-Z][^a-z]*#) word beginning by one lower case
Relocate Data from Pm to Cloud
Convert > number format
S> store
Process->Continue
Relocate number from (#[^0]-[^9]*)
Relocate word from (#[^A-Z]*[^a-z])
End

The Table 1 comprises the various evaluation techniques used in handwritten recognition techniques. These comparisons are based on the related works as mentioned above and all about the conversion of manuscript into computerized image into languages. The character image mentioned in the above table contains various methodologies used for those respective techniques. The phases of character recognition include preprocessing, future extraction and classification. In preprocessing the format used for the conversion s can be mentioned. In feature extraction the algorithm is specified to character reorganization. Classification contains different classification models which are the basic classification in character recognition. Finally this table comprises with the exact accuracy claimed in each techniques. The accuracy is related to the conversion of handwritten character recognition.

Table 1.1: Comparisons of Various Techniques in Handwritten Recognition Technique

<table>
<thead>
<tr>
<th>Paper</th>
<th>Character Image</th>
<th>Pre-Processing</th>
<th>Feature Extraction</th>
<th>Classification</th>
<th>Accuracy Claimed</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Kannada</td>
<td>Handwritten Numbers</td>
<td>Orthogonal Spline Uniset</td>
<td>Multilayer Cluster Neural Network</td>
<td>94.70</td>
</tr>
<tr>
<td>[2]</td>
<td>Vowel</td>
<td>Handwritten Numbers</td>
<td>Wavelet Transform</td>
<td>NN</td>
<td>98.32</td>
</tr>
<tr>
<td>[3]</td>
<td>Optical</td>
<td>Handwritten Numbers</td>
<td>Wavelet Transform</td>
<td>Multilayer Cluster Neural Network</td>
<td>93.56</td>
</tr>
<tr>
<td>[4]</td>
<td>Urdu</td>
<td>Movement Invariant Feature</td>
<td>SVM</td>
<td>SVM</td>
<td>93.50</td>
</tr>
<tr>
<td>[5]</td>
<td>Tamil</td>
<td>Biometric</td>
<td>Image Moment</td>
<td>SVM</td>
<td>88</td>
</tr>
<tr>
<td>[6]</td>
<td>Devanagari</td>
<td>Median Filter</td>
<td>Gradient</td>
<td>SVM</td>
<td>92.11</td>
</tr>
<tr>
<td>[7]</td>
<td>Amharic</td>
<td>Noise Removal</td>
<td>SIFT</td>
<td>SVM</td>
<td>98.46</td>
</tr>
<tr>
<td>[8]</td>
<td>Grapha</td>
<td>Time Domain Feature</td>
<td>DTW</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>[9]</td>
<td>Handwritten</td>
<td>Biometric</td>
<td>Wavelet</td>
<td>NN With Subdivided distance Metric</td>
<td>SVM</td>
</tr>
</tbody>
</table>

Performance Evolution:
The graph shows the performance appraisal for recognition of text from the formatted image in handwritten recognition methods related to our parser method. Back propagation technique and SIFT algorithm are the existing technique which has the lower training time and accuracy when compared to our proposed technique parser. This method is much enable for the user to highly efficient and actually keen in the vicinity of obtains the exact result. For the existing techniques the accuracy and time seconds can be quite varies. But for our parser technique it is much differs by its accuracy maintaining and time. That is shown by our result analysis graph, the exact character can be recognized and the status determines the relative classes for character matching. These declares the rate of character to distinguish the exact character.
Experimental Result Analysis
This field holds sample inputs of different elevated hand written character. In the field of classes that will matches with the list of prevalence. Several classifications for the single character recognition are also available in the sample subject. Exactly the hand written character matches with the one. This kind of character recognition is quite easy to identify and may the technique involved is also practicable.

Conclusion
There is a need for converting the handwritten manuscript for now a day users. The existing conversion model varies from different phases of conversion. Our proposed technique parser conversion is quite similar but it absolutely varied by its ease of utilization. The conversion of handwritten books into formatted image faces unreliable difficulties but the conversion of formatted handwritten image into text has highly efficient usage and very reliable. The working output produced by our proposed method is appreciable for its ease of understanding and quite gracious for the users. Our method enhances with recognition accuracy able to perform better consequence and distinguishes with fine ability provides the way for emerging future techniques in character and handwritten recognition.
References


