International Journal of Ocean System Engineering 2(1) (2012) 57-61
http://dx.doi.org/10.5574/IJOSE.2012.2.1.057

Recognize Handwritten Urdu Script Using Kohenen Som Algorithm

Yunus Khan [1*] and Chetan Nagar [2]

[1]*M.E Researcher JIT Borawan Khargone M.P. India*

[2] *Professor & Head Department of CSE JIT Borawan M.P. India*

(Manuscript Received December 20, 2011; Revised January 14, 2012; Accepted February 27, 2012)

Abstract

In this paper we use the Kohonen neural network based Self Organizing Map (SOM) algorithm for Urdu Character Recognition. Kohenen NN have more efficient in terms of performance as compare to other approaches. Classification is used to recognize hand written Urdu character. The number of possible unknown character is reducing by pre-classification with respect to subset of the total character set. So the proposed algorithm is attempt to group similar character .Members of pre-classified group are further analyzed using a statistical classifier for final recognition. A recognition rate of around 79.9% was achieved for the first choice and more than 98.5% for the top three choices. The result of this paper shows that the proposed Kohonen SOM algorithm yields promising output and feasible with other existing techniques.

Keywords: Pre-Classification, Neural Network, Handwritten character, SOM, Baseline, Statistical, Structural, Crux, Meticulous and Sobel edge detection.

1. Introduction

The area of Urdu character recognition have attract many researches to work in disciplines for translation of hand written of printed documents to an computer editable format such as soft copy ,automated postal address recognition system, word processing, data acquisition. Most of the work done in the field of character recognition is confined to Roman [1], English [2,3], Urdu [4,5], Chinese / Japanese languages [6,7,8]. Now a day some efforts have been reported in literature for Devanagari [9,10], Bangla [11,18], Telugu [12,13,14], Tamil [15,16,17] scripts. Most of the character recognition techniques are problem-oriented. Techniques are devised for the recognition of a particular script depending upon the nature and complexity of the character. Broadly speaking, the features can be physical, topological, mathematical or statistical in nature. These strategy used for recognition can be broadly classified into structural, statistical and hybrid. Structural techniques use some qualitative measurements as features. Statistical techniques use some quantitative measurement. In hybrid approach, these two techniques are combined at appropriate stage first representation of characters and utilizing them for recognition. In this paper we use hybrid techniques, in which structural properties of the text line are used for the first stage of preliminary classifications. A statistical classifier recognizes the unknown character as one of the members of the pre-classified group.

2. Properties of Urdu Script

In India there are twelve scripts and Urdu is one of the popular Indian scripts. Here we describe some properties of the Urdu script that are useful for building the OCR system. The modern Urdu alphabet consists of 39 basic characters. These characters are shown in Fig.1(a). Urdu has

[*] Corresponding author. Tel.: +91-7285-277862, Fax.: +91-7285-277710.
E-mail address: callyunuskhan@gmail.com

10 numerals and the numerals are shown in Fig.1(b). Like other Indian scripts in Urdu also two or more characters may combine and create a complex shape called compound characters. Examples of some compound characters are shown in Fig.2. Also depending on the positions (first, middle or last) in a word the basic shape of a character may be changed. For example see Fig.3. Here an Urdu basic character in its isolated form and its shapes in first, middle and last positions of a word are shown. As a result, the total number of characters to be recognized is very large. Thus, OCR development for Urdu is more difficult than any European language script having a smaller number of characters.

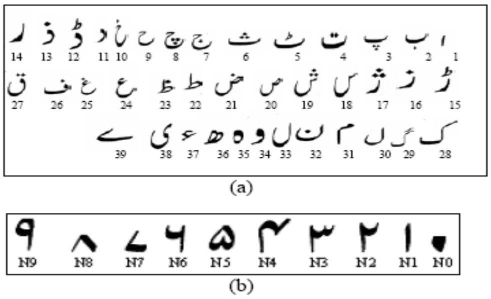


Fig.1. Examples of Urdu alphabet and numerals (a) Basic characters of Urdu alphabet (b) Urdu numerals.

Urdu script has some different characteristics compare to other Indian scripts. Writing style in Urdu is from right to left whereas it is left to right in other Indian scripts. It can be noted that an Urdu basic character may have four components (see character number 6,8,17,19 etc. of Fig.1(a)) while in other Indian scripts this property is rare. There is a structural similarity between Urdu and Arabic script. There are different types in Urdu script like Naskh, Nastaliq, Aswad, Batool, Jaben etc. We consider here Naskh and Nastaliq types.

خ+د+ا—خدا	ک+ا+م — کام	ا+م+ن = امن
ا+د+ب=ادب	ق+ب+ر=قبر	ب+د+ل=بدل
ر+ا+ج—راج	ع+ج+ب=عجب	چ+ا+ٹ—چاٹ
ا+ح+د—احد	ق+د+ر—قدر	د+ا+م = دام
ش+م+ر—شمر	ن+ر+م — نرم	ر+ا+گ = راگ
پ+ا+س—پاس	ج س+م—جسم	س+ا+ز = ساز

Fig.2 Some examples of Urdu compound characters.

Isolated	First	Middle	Last
ب	بـ	ـبـ	ـب
ح	حـ	ـخـ	ـح
ف	فـ	ـفـ	ـف

Fig. 3. An isolated basic character and its shapes in first, middle, and last positions in a word are shown.

3. Proposed System

Most of the recognition systems are composed of two basic subparts: Feature extraction and classification. Feature extraction deals with the basic operations like acquisition, noise reduction, scaling, segmentations etc., On the other hand, classification can be said as recognition. The aim of preliminary classification is to reduce the number of possible unknown character, to a subset of the total character set.

4. Information Collection

Data samples were collected from different writers on any sized documents. First of all, the input data are resized to 250 X 250 pixels to satisfy procedure, regardless of whether it’s an image of a single character or a word. The system was trained with both computer-generated images and scanned images of text; may it be a single character or a word. In preprocessing, noise is removed from the image by a spatial filter. It should be noted that no skew correction was done, so the scanning process is expected to be a high quality. Quality of the image is a great factor for the performance of the system.

5. Segmentation

Text area from the document, which may consist of multi lines, is extracted and the segmentation step is followed. Further, each line is segmented into individual words, and finally ach word is segmented into individual characters. The method is based on horizontal projection profile corresponds to the horizontal gaps between text lines. Each text line is identified using two-reference line known as upper line and lower line. They correspond to the minimum and maximum zero value positions adjusting a text line respectively. (See Fig. 1) First derivative of the horizontal projection profile is calculated for each segmented text line. The lines drawn across the two peaks in Fig. 1 indicate the two baselines.

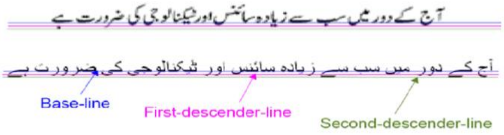

آج کے دور میں سب سے زیادہ سائنس اور ٹیکنالوجی کی ضرورت ہے
آج کے دور میں سب سے زیادہ سائنس اور ٹیکنالوجی کی ضرورت ہے
Base-line
First-descender-line
Second-descender-line

Fig.1 Reference Line Identifications

A pre-formatted paper for the collection of handwriting was used to guide the writer and simplify the process of reference line extraction. Each document has the four references line printed on it. However, these lines are completely eliminated during the binarization of the image and have no effect on the segmentations. After the references lines have been found, words and characters are extracted using the vertical projection profile of each text

line. Word boundaries and character boundaries are distinguishable since the former are much wider than the latter. One all the characters have been segmented, the minimum-bounding box of each character is identified eliminating the while space around it. Upper and lower boundary values of the minimum boundary box, along with the four reference lines, are sent to the next stage for preliminary classification.

6. Pre Classification

Aspiration of this classification is to reduce the number of possible characters for an unknown character, form the known one. So the characters are categorized into two groups where the characters of the first group lie in the two baselines are categorized into crux characters group. On the other hand, the character that cross the base line as Exhaustive group. Again this exhaustive group is further divided into two sub groups for easy recognition. "*Ascending exhaustive characters"* which cross the upper base line and "*Descending exhaustive characters*" are the one that cross the lower base line .Characters under this consideration, classified into the above pre-classification groups. Characters belonging to other groups like numbers and Sanskrit based characters are assumed to be invalid matches and are not considered for the recognition.

7. Feature Extraction

This feature extraction is a most important part of the Urdu character recognition procedure. Here creation of vectors from the image (binary images) is carried out. All the segmented characters images are then scaled into a common height and width (32 X 32 pixels) using a bilinear interpolation technique. Usually some unwanted portions are included in the image. This can be corrected by Sobel edge detection algorithm, using Sobel mask. The process makes the feature detection process easier. Moreover Median filtering made the sample that increases the efficiency of the process.

8. Recognition Process

Lots of activities in pre-processing stages help to process this stage very easy. Self-organizing feature maps (SOFM or SOM) are unsupervised machine learning that learns by self-organizing and competition [20]. The main idea for this is to make it simple and acceptable for Kohonen SOM. It reduces a remarkable amount of time. SOM is clustering the input vector by calculating neuron weight vector according to some measure (e.g. Euclidean distance), thus weight vector that closet to input vector comes out as winning neuron. However, instead of updating only the winning neuron, all neurons within a certain neighborhood of the winning neuron are updated using the Kohonen rule [20].

The algorithm is described as follows, suppose the training set has sample vectors X, trains the SOM network has following steps:

i) Firstly, all neuron nodes weights, defined as
W j (1), j = 1…L,
are initialized randomly.

ii) K=Maximum (K(k)), for interation step k=1…K, get an input vectorX(k) randomly or in order.

iii) Caculate Distance=X(k),. K=1…n
1…n refers to neuron nodes.

iv) Select the winner output neuron j*with minimum distance.

v) Update weights Wj(k+1)to neurons j*and its neighborhood:
Wj(k=1)=Wj(k)+α(k+1)∩(j,j*
(k+1),(k+1))[X(k+1)-Wj(k)],
j=1…L
vi) If k=K go to step (ii)

In this algorithm, α(k) is a step function that decreases monotonically with k ∩ (j ... j*(k), k as neighborhood function. It is formulated as follows:

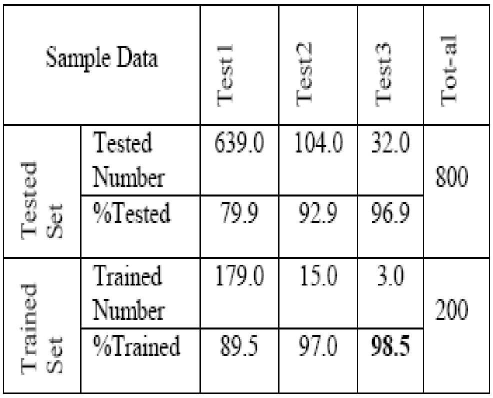
$$\cap\ A(j, j^*(k),k)=-\exp\left(\frac{d^2 j^*\cdot k(k)}{2\sigma^2(k)}\right)$$

Where σ(k) defines the width of the neighborhood which decreases in time monotonically, and $d^2j^* \cdot k(k)$ is Euclidean metric distance between the neuron to be adjusted to the winner neuron j*.

9. Tentative Result

Experimental data is divided into two distinct sets: a training set of 200 samples and a testing set of 800 samples. In experiment, total 100 text lines were subjected to segmentation and reference line identification. We conducted several test by various portion of the training data, to see how well the system represents the data it has been trained on. In all the cases, every character in each text line was correctly segmented. The reference line identification was almost 98.5% accurate resulting only 1% pre-classification error. Results of the recognition process are given in Table 2.
Kohonen SOM shows very good promise indeed, especially as compared to Neural network based ones. Not only is the accuracy rate consistently higher, the time performance to train and recognize are better as Kohonen networks do not have hidden layers.

Table 2. Recognition Process Result



Sample Data		Test1	Test2	Test3	Tot-al
Tested Set	Tested Number	639.0	104.0	32.0	800
	%Tested	79.9	92.9	96.9	
Trained Set	Trained Number	179.0	15.0	3.0	200
	%Trained	89.5	97.0	**98.5**	

10. Conclusion

We investigated a new representation of Urdu Character Recognition, and used Kohonen SOM techniques efficiently classifies handwritten and also for Printed Urdu characters. More effective and efficient feature detection techniques will make the system more powerful. There are still some more problems in recognition. They are, during letter segmentations and abnormally written characters (which misguide the system during recognition). Misrecognition could be avoided by using a word dictionary to look-up for possible character composition. The presence of contextual knowledge will help to eliminate the ambiguity. We show that, in practice, the proposed approach produces near optimal results besides outperforming the other methodologies in existence. Our future work in this regard will be analyzing the features of joined letters and incorporating better segmentation accuracy. Results indicate that the approach can be used for character recognition in other scripts as well.

Contribution

This paper introduces a new technique for Urdu character recognition such as SOM. SOM model also captures the invariant features of Urdu script. Unlike other neural network it does not hold any hidden layer. Only two layers are needed. One is for input and the other for output. This is useful for visualizing from higher dimensional input space to lower-dimensional map space.

References

[1] C. E. Dunn and P. S. P. Wang, *"Character segmentation techniques for handwritten text - a survey"*, in the Proceedings of 11th ICPR, Vol. 2, pp. 577-580, 1992.

[2] R. M. Bozinovic and S. N. Srihari, *"Off-line cursive script word recognition", IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 11, no. 1, pp. 68-83, Jan. 1989.

[3] Hu, M. K. Brown and W. Turin, *"HMM based on-line handwriting recognition", IEEE Trans. on pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 1039-1045, Oct. 1996.

[4] U. Pal and B. B. Chaudhuri, *"Indian script character recognition: a survey", Pattern Recognition*, Vol. 37(9), pp. 1887-1899.

[5] *http://en.wikipedia.org/wiki/Official_languages _of _ India Tentative System*

[6]] D. Deng, K. P. Chan, and Y. Yu, *"Handwritten Chinese character recognition using spatial Gabor filters and selforganizing feature maps", Proc. IEEE Inter. Confer. On Image Processing, vol. 3, pp. 940-944, Austin TX, June 1994.*

[7] C-H. Chang, *"Simulated annealing clustering of Chinese words for contextual text recognition", Pattern Recognition Letters,* vol. 17, no. 1, pp. 57-66, 1996.

[8] H. Yamada, K. Yamamoto, and T. Saito, *"A non-linear normalization method for handprinted Kanji character recognition–line density equalization", Pattern Recognition,* vol. 23, no. 9, pp. 1023-1029, 1990.

[9] S. D. Connell, R. M. K. Sinha and A. K. Jain, *"Recognition of unconstrained On-line Devanagari characters", in the Proceedings of 15 International Conference on Pattern Recognition (ICPR),* Vol. 2, Spain, pp. 368-371, 2000.

[10] S. D. Connell and A. K. Jain, *"Template-based online character recognition", Pattern Recognition* , Vol. 34(1), pp. 1-14, 2001.

[11] Bangla A. K. Ray and B. Chatterjee, *"Design of a nearest neighbor classifier system for Bengali character recognition", J. Inst. Elec. Telecom. Engg.*, Vol. 30, pp. 226-229, 1984.

[12] S. N. S Rajasekaran and B. L. Deekshatulu, *"Recognition of printed Telugu characters", Computer Graphics and Image Processing (CGIP),* Vol. 6, pp. 335- 360, 1977.

[13] C. V. Lakshmi and C. Patvardhan, *"A high accuracy OCR system for printed Telugu text", in the Proceedings of Conference on Convergent Technologies for Asia-Pacific Region (TENCON 2003),* Vol. 2, pp. 725-729, 2003

[14] A. Negi, C. Bhagvati and B. Krishna, *"An OCR system for Telugu", in the Proceedings of the Sixth International Conference on Document Processing,* pp. 1110-1114, 2001.