On-line Handwritten Kannada Character Recognition using Feed Forward Network with Back-propagation

P. RamakanthKumar[1], Manjaiah[2], P.Subashini[3], Dr.S.C.Sharma[4]

ABSTRACT

It is a well-known fact that building character recognition systems is one of the hottest areas of research as it is shown over the Internet. Due to its wide range of prospects, Character recognition has been an important part of the civilization for many centuries. We depend on digitally assembled and recognizable characters for common communications. This paper describes a method to recognize on-line handwritten Kannada characters, the language used by the state of Karnataka. A subset of the Kannada alphabet was chosen for the study. The Back propagation neural network is used for recognition.

Keywords: Preprocessing, segmentation, feature extraction, scaling, moments, neural network.

1. INTRODUCTION

The process of handwriting recognition involves extraction of some defined characteristics called features to classify an unknown handwritten character into one of the known classes. A typical handwriting recognition system consists of several steps, namely: preprocessing, segmentation, feature extraction and classification. Several types of decision methods, including statistical methods, neural networks, structural matching (on trees, chains, etc) and stochastic processing (Markov chains, etc) have been used along with different types of features. Many recent approaches mix several of these techniques together in order to obtain improved reliability[1], despite wide variation in handwriting[2]. Kannada script is the official language of Karnataka state. Kannada script is alphabetic in nature and is written from left to right.

ಅ ಆ ಇ ಈ ಉ

ಊ ಋ ೠ ಎ ಏ

ಐ ಒ ಓ ಔ ಅಂ ಅಃ

Figure 1.1: A sample of Kannada characters

This paper is organized as follows. Section 2 explains the process of preprocessing and feature extraction. Recognition of the characters by using back propagation neural network is then described in section 3. Experimental results are provided in section 4 followed by concluding remarks in section 5.

2. PREPROCESSING AND FEATURE EXTRACTION

The input is through a mouse driven over a grid of cells. Inputted character is scaled down to uniform size (scaling)[4]. The stray pixels are removed that may be formed during mouse driven input.

[1]Research Scholar, CMRTU, R&D ,R.V.College of Engineering, Bangalore, INDIA,
Email: pramakanth_2000@yahoo.com

[2]Reader & Chairman, Department of Computer Science, Mangalore University, Mangalore, INDIA
Email: ylm123@yahoo.co.in

[3]Lecturer, Department of Computer Science, Avinashilingam University, Coimbatore, INDIA
Email: mail_suba@yahoo.com

4Principal, R.V.College of Engineering, Bangalore, INDIA
Email: rvrdscs@blr.vsnl.net.in

2.1 Feature extraction:

It involves the extraction of the features that helps in distinguishing each character. The following features are extracted from the input. 48 features are extracted from the input image[3].

- Hole Recognition
- Stroke length
- Segmentation
- Pixel Distribution
- Asymmetric feature
- Moments Feature

2.1.1 Hole Recognition

This feature recognizes the holes and returns the number of holes present in the character.

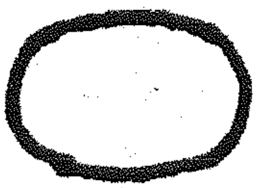

Figure 3.1 Image of a hole segment

Algorithm:

- Each white pixel that can reach the boundary is numbered as 2
- Each pixel numbered 2 is checked if it can reach a white pixel without coming across a pixel that is black. If so, it is numbered as 3, and it is part of a hole
- Holes are counted by labeling them incrementally starting with 3.
- Total number of holes = Last labeled number – 3

2.1.2 Stroke Length:

Features extracted: Length of the character in terms of pixels[5].

<100	-3 Characters
>125 & <150	-3 Characters
>100 & <125	-7 Characters

Algorithm:

- Initialize Stroke Length
- Raster Scan The Saved Character
- Increment Stroke Length For Every White Pixel

2.1.3 Pixel Distribution:

It gives the spatial distribution of pixels of the character

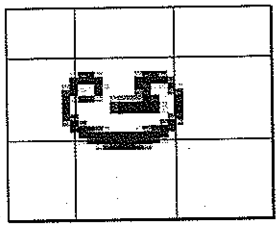

Figure 3.2 Pixel distribution of a character

Algorithm:

- Divide the character vertically into three segments such that each segment has equal distribution of pixels[6][7].
- On similar lines divide the character horizontally.
- Result is a character divided into 9 segments.

Img11	Img12	Img13
Img21	Img22	Img23
Img31	Img32	Img33

Figure 3.3 Character Image Segmentation

P11 = pixel count of img11

P12 = pixel count of img12

P13 = pixel count of img13

P21 = pixel count of img21

P22 = pixel count of img22

P23 = pixel count of img23

P31 = pixel count of img31

P32 = pixel count of img32

P33 = pixel count of img33

S1=max(P11,P12,P13,P21,P22,P23,P31,P32,P33)

Feature 1 = P11/S1

Feature 2 = P12/S1

Feature 3 = P13/S1

Feature 4 = P21/S1

Feature 5 = P22/S1

Feature 6 = P23/S1

Feature 7 = P31/S1

Feature 8 = P32/S1

Feature 9 = P33/S1

P1112 = P11 + P12

P1213 = P12 + P13

P2122 = P21 + P22

P2223 = P22 + P23

P3132 = P31 + P32

P3233 = P32 + P33

S2 = max(P1112 + P1213 + P2122 + P2223 + P3132 + P3233)

Feature 10 = P1112/S2

Feature 11 = P1213/S2

Feature 12 = P2122/S2

Feature 13 = P2223/S2

Feature 14 = P3122/S2

Feature 15 = P3223/S2

P1121 = P11 + P21

P2131 = P21 + P31

P1222 = P12 + P22

P2232 = P22 + P32

P1323 = P13 + P23

P2333 = P23 + P33

S3 = max(P1121 + P2131 + P1222 + P2232 + P1323 + P2333)

Feature 16 = P1121/S3

Feature 17 = P2131/S3

Feature 18 = P1222/S3

Feature 19 = P2232/S3

Feature 20 = P1323/S3

Feature 21 = P2333/S3

2.1.4 Asymmetric Feature

It gives the horizontal and vertical symmetry features of the character[8].

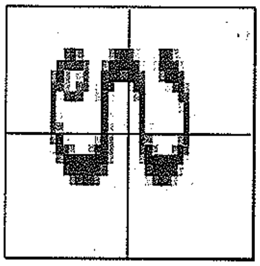

Figure 3.4 Character having Symmetry

Algorithm

- Horizontal axis is selected.
- Number of pixels that differ on either side of it is counted. the value divided by 200 gives the horizontal symmetry.
- Vertical axis is selected.
- On similar lines as in the case of horizontal symmetry, vertical symmetry of the character is calculated

Features Extracted:

2.1.5 Moments Features

Moment features are very common in character recognition systems because they capture the global shape attributes of the character. This is possible because the moment features are calculated by looking at three factors[9].

- The total number of black pixels in the image
- The number of black pixels along the horizontal
- The number of black pixels along the vertical

These three data are used to calculate moments of third order, which are independent of rotation, translation and scaling[10].

These moments are calculated by employing the raster scan on the image to calculate certain values.

Let the co-ordinates of a generic black pixel be given by (u,v), for the 2-D binary image of a segment. The central moments are given by

$$\mu_{pq} = 1 / N \sum (u_i - vbar)^p (u_j - ubar)^q$$

$$\text{Where} \quad ubar = 1/N \sum u_i$$

$$vbar = 1/N \sum v_j$$

where N is the total number of black pixels in the image.

The intermediate moment values that are needed to find the final moments are

$$\mu_{pq} = 1 / N \sum \quad ({}_i uvbar)^p (u_i \ ubar)^q$$

$$\text{Where} \quad ubar = 1/N \sum \ {}_i u$$

$$vbar = 1/N \sum \ {}_i v$$

where N is the total number of black pixels in the image.

The intermediate moment values that are needed to find the final moments are

$$r = (\mu_{20} + \mu_{02})^{1/2}$$

$$m2 = (\mu_{20} + \mu_{02})^2 \ + 4 \ \mu_{11}^4$$

$$m3 = (\mu_{30} + 3 \mu_{12})^2 \ + \ (3\mu_{21} + \mu_{03})^2$$

$$m4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

The following are the important moments that are size, rotation and translation independent.

$$m2' = m2 / r^4$$

$$m3' = m3 / r^6$$

$$m4' = m4 / r^4$$

Normalized central moments[11] are

$$m5 = (\mu_{30} - 3\mu_{12}) \times (\mu_{30} + \mu_{12}) \times ((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2) + (\mu_{03} - 3\mu_{21}) \times (\mu_{03} + \mu_{21}) \times ((\mu_{30} + \mu_{21})^2 - 3(\mu_{12} + \mu_{30})^2)$$

$$m6 = (\mu_{20} + \mu_{02}) \times ((\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2) + 4\mu_{11} \times (\mu_{30} + \mu_{12}) \times (\mu_{30} + \mu_{21})$$

$$m7 = (3 \ \mu_{21} - \mu_{03}) \times (\mu_{30} + \mu_{12}) \times ((\mu_{30} + \mu_{12})^2 - 3 \ (\mu_{21} + \mu_{03})^2) + (\mu_{30} - 3\mu_{21}) \times (\mu_{30} + \mu_{21}) \times ((\mu_{21} + \mu_{03})^2 - 3 \ (\mu_{30} + \mu_{12})^2)$$

Affine moments are

$$I1 = (\mu_{20} + \mu_{02}) - \mu_{11}^{\ 2} / \mu_{00}^{\ 2}$$

$$I2 = (\mu_{30}^{\ 2} \times \mu_{03}^{\ 2} - 6 \ \mu_{30} \times \mu_{21} \times \mu_{12} \times \mu_{03} + 4\mu_{30} \ \times \mu_{12}^{\ 3} \ + 4 \ \mu_{21}^{\ 2} \times \ \mu_{03} - 3\mu_{21}^{\ 2} \times \mu_{12}^{\ 2}) / \ \mu_{00}^{10}$$

$$I3 = (\mu_{20} \ \times (\mu_{21} \times \mu_{30} \ - \ \mu_{12}^{\ 2}) - \mu_{11} \times (\mu_{30} \times \mu_{03} - \mu_{21} \times \mu_{12}) \ 2 \ \mu_{02} \times (\ \mu_{30} \times \mu_{12} - \mu_{21}^{\ 2})) \ / \ \mu_{00}^{\ 7}$$

$$I4 = (\mu_{20}^{\ 3} \times \mu_{03}^{\ 3}) - 6 \ \mu_{20}^{\ 2} \ \times \mu_{11} \times \mu_{12} \times \mu_{03} + 6 \ \mu_{20}^{\ 2} \times \mu_{02} \times \mu_{21} \times \mu_{03} + 9 \ \mu_{20}^{\ 2} \ \ \mu_{02} \times \mu_{12}^{\ 2} + 12\mu_{20} \times \mu_{11}^{\ 2} \times \mu_{21} \times \mu_{03} + (6 \ \mu_{20} \ \times \mu_{02} \times \mu_{30} \times \mu_{03}) - (18 \ \mu_{20} \ \times \mu_{11} \times \mu_{02} \times \mu_{21} \times \mu_{12}) + 8 \ \mu_{11}^{\ 2} \ \times \mu_{30} \times \mu_{03} - (6 \ \mu_{20} \ \times \mu_{02}^{\ 2} \times \mu_{30} \times \mu_{12}) + (9 \ \mu_{20} \ \times \mu_{02}^{\ 2} \times \mu_{21}^{\ 2}) + (12 \ \mu_{20}^{\ 2} \ \times \mu_{02} \times \mu_{30} \times \mu_{12}) - (8 \ \mu_{11} \ \times \mu_{02}^{\ 2} \times \mu_{30} \times \mu_{21}) + (\mu_{20}^{\ 3} \ \times \mu_{03}^{\ 2} \ )) \ / \ \mu_{00}^{\ 11}$$

Features = (m2,m3,m4,m5,m6,m7,I1,I2,I3,I4)

3. Back-Propagation Algorithm

The type of neural network being used is feed forward network. Feed forward network is composed of a hierarchy of processing units, organized in a series of two mutually exclusive sets of neurons or layers[12]. The first or input layer serves as a holding site for the inputs applied to the network. The last or output layer is the point at which the overall mapping of the network input is available. Between these two extremes lie zero or more layers of hidden units, it is in these internal layers that additional remapping or computing takes place.

Error back propagation algorithm is used for the training of the neural network[13].

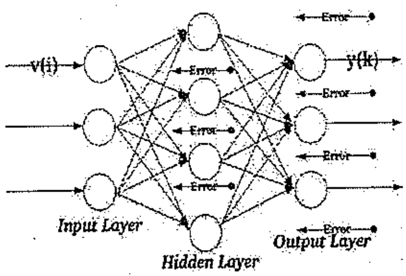

Figure 3.5 Feed forward Neural network architecture

Totally 48 features are extracted from the input image.

The input layer consists of 48 neurons.

The network is experimented with different number of hidden layers. For example 10, 15, 20, 25, 30 etc.

The output layer consists of 1 neuron.

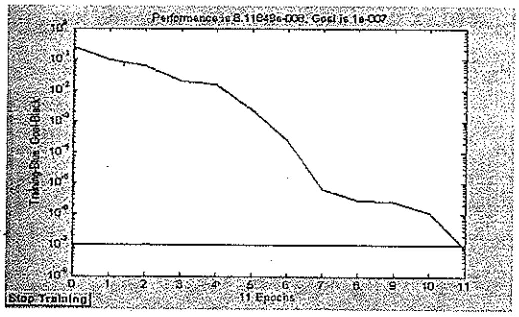

Figure 3.6 Learning graph

4. Experimental Results

Testing Cases

a. Discontinuous characters: For looping structure employed to fill empty cells.

b. Recognition of Hole incomplete: Image was raster scanned to reassign tables, to recognize hole.

c. Change in the features due to the unexpected pixels in the input: Image is raster scanned to remove the stray pixels.

d. We were able to achieve 97% recognition accuracy.

Sample output

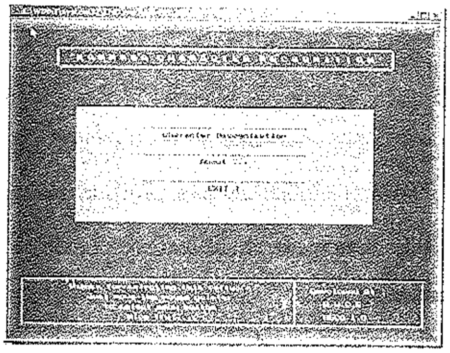

Figure 4.1 Sample GUI

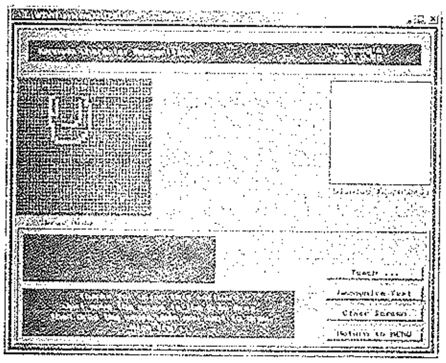

Figure 4.2 Sample input handwritten character

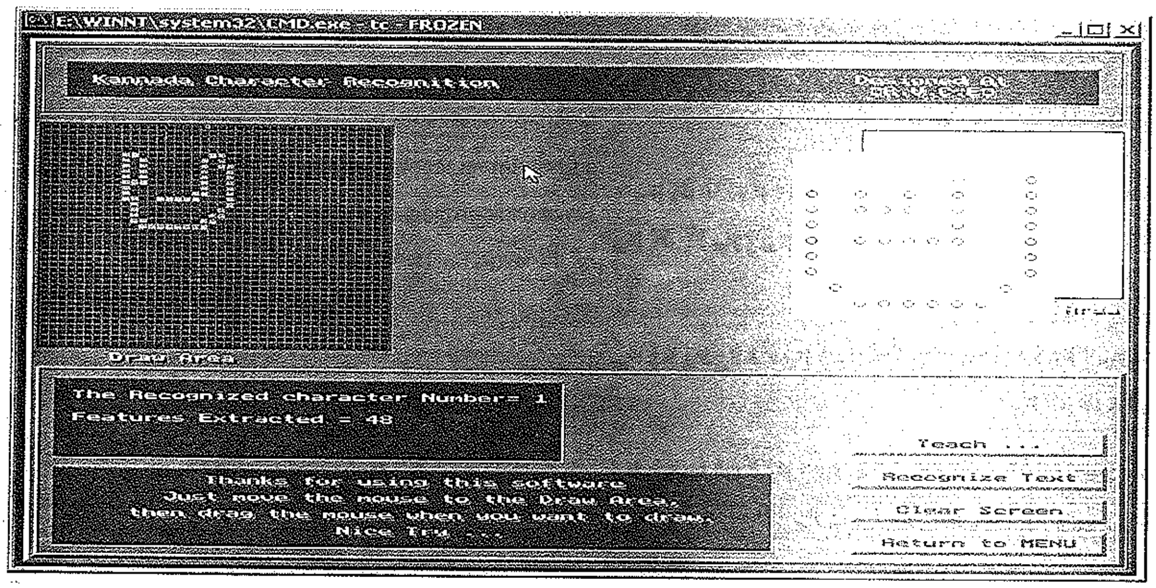

Figure 4.3 Recognized input character

5. Conclusion and future work

In this paper we have presented a system for recognizing handwritten Kannada characters (only vowels). A neural Network based classifier was used for the recognition. Since handwritten characters can be considered as a construction of line segments at different orientations and lengths, an orientation selective method such as Gabor filtering should produce effective features [14][15]. Future versions of the system are planned to cater for the full Kannada Alphabet. The use of a language model, which introduces linguistic knowledge into the system and thereby improves recognition accuracy, is also consideration.

REFERENCES

1. LTG (Language Technologies Group), 2003. *"Optical Character Recognition for Printed Kannada Text Documents"* SERC, IISc Bangalore.
2. VijayaKumar, B., 2001. *"Machine Recognition of Printed Kannada Text. IISc Bangalore."* The Unicode Standard Version 3.0, Addison Wesley.
3. Gonzalez, R.C., Woods, R.E., Eddins, S.L., 2004. *"Digital Image Processing Using MATLAB."* Pearson.
4. 2D grey-level skeleton computation: a discrete 3D approach, di Baja, G.S.; Nystrom, I, *"Proceedings of the 17th International Conference Pattern Recognition,"* 2004. ICPR 2004 on Volume 2, 23-26 Aug. 2004 Page(s):455 - 458 Vol.2
5. Rapid recognition of object outlines in reduced resolution images, Richard W. Stimets, Wu Li Ying and K. Rajasekharan, Pattern Recognition, Volume 19, Issue 1, 1986, Pages 21-33.
6. Blurred character recognition by complementing features of blurred regions Osamu Shiku, Akira Nakamura, Sueharu Miyahara, Terumitsu Ohyama Systems and Computers in Japan,Volume 36, Issue 6, 2005. Pages 59-68

[7] L.R.Rabiner, *"A tutorial on Hidden Markov Models and Selected applications in Speech Recognition,"* Proc.IEEE, vol.77,no.2,1989,pp.257-285

[8] H.Bunke, M.Roth and E.Talamazzini, *"Off-line cursive Handwritten Recognition using Hidden Markov Models,"* Pattern Recognition, vol.28, no.9,1995,pp1399-1413

[9] H.J.Kim, K.M.Kim, S.K.Kim and J.K.Lee, *"On-line Recognition of Handwritten Chinese Characters based on Hidden Markov Models,Pattern Recognition,"* vol.30, no.9,1997,pp-1489-1500

[10] G.Loudon, C.Hong, Y.Wu and R.Zitserman, The Recognition of handwritten Chinese characters from paper *"records, IEEE TECCON, Digital Signal Processing Applications,"* 1996,pp.923-926

[11] R.M.Bozinovic and S.N.Srihari, *"Off-line Cursive Script Word Recognition,"* IEEE Trans. On PAMI, vol.11, no.1,1989, pp. 68-83.

[12] Y.zhu, T.Tan and Y.Wang , *"Font Recognition based on Global Texture Analaysis,"* IEEE Trans. On PAMI, vol.23, no.10, 2001, pp.1192-1200

[13] Cheng-Lin Liu, Masahi Koga, *"Lexicon Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading",* IEEE Transactions on Pattern Recognition and Machine Intelligence, vol.24, no.11, Nov.2002

[14] A.EL-Yacoubi, M.Gilloux, R.Sabourin, and C.Y.Suen, *"An HMM based approach for Offline unconstrained Handwritten word modeling and Recognition"*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.21,pp.752-760-379-556,1999

[15] Y.H.Tseng and H.J.Lee, *"Recognition based handwritten Chinese character Segmentation using probabilistic Viterbi Algorithm,"* Pattern Recognition letters, vol.20,no.8,pp.791-806, 1999

Author's Biography

P. Ramakanthkumar working as a Research Scholar, in CMRTU, R&D, R.V.College of Engg., Campus Bangalore. INDIA. Currently doing research in the area of off-line handwriting recognition. Completed B.E. in Computer Science and M.S., in Software Systems during 1993 and 1999 respectively from S.J.C.E.,Mysore and BITS Pilani. His area of specializations are Digital Image Processing, Pattern Recognition. He has published around 10 National and International Conference Publications and Journals. He has guided 20 - Under Graduate and 5 - Post Graduate Industry Projects. He is the member of Professional bodies like ISTE, CSI.

D.H. Manjaiah, working as a READER & CHAIRMAN in the Department of Computer Science, MANGALORE UNIVERSITY, MANGALORE, INDIA. Currently doing research on design and development of Visual Shop Floor Control Systems. Received B.E. in Computer Science and M.Tech., in Computer Science during the year 1990 and 1993 respectively from National Institute of Karnataka Surathkal., (Formerly Mangalore University), INDIA. His area of specializations are Internet Programming and Web Technologies, E-commerce, Advanced Communication Networks, Object oriented programming and Operations Research. He has been published totally 30 National and International Conference Publications. He has guided 20 - Under Graduate and 75 - Post Graduate Industry Projects. He is the member of Professional bodies like ISTE, CSI, IKSM (Information Knowledge System Management) USA and ISOC (Internet Society) USA. His personnel web site http://manju.8m.com/ and E-mail: ylm321@yahoo.co.in

P.Subashini completed her B.Sc Mathematics, and obtained MCA in Bharathiar University. She completed her M.Phil (Computer Science) in Avinashilingam University, Tamilnadu. She is working as a selection grade lecturer in Avinashilingam University, Coimbatore. She has 12 years of teaching experience. Her Areas of Specialization are Object oriented technology, Image processing and Pattern recognition. She has published many national and international level papers in those areas.

Email: mail_suba@yahoo.com

Dr. S.C.Sharma holds Doctorate in Mechanical engineering, Doctor of Science in Computer Science and engineering, Doctor of Science in Material science & Technology and Doctor of Engineering in Mechanical engineering. Recently he got Doctor of Science(Honori's Causa) from the Avinashilingam University, Coimbatore. He is working as the principal of RV college of Engineering, Bangalore. He is a senior Scientist for the college of Engineering and Applied Science, University of Wisconsin-Milwaukee, USA. And Adjunct Professor of Engineering, Insitute of Technology, West Virginia University, Montgomery, WV-25136, USA.