Journal of Computer Science 4 (7): 509-516, 2008
ISSN 1549-3636

An Improved Handwritten Tamil Character Recognition System using Octal Graph

[1]R. Jagadeesh Kannan and [2]R. Prabhakar
[1]Department of Computer Science and Engineering,
RMK Engineering College, Kavaraipettai, India
[2]Department of Computer Science and Engineering,
Coimbatore Institute of Technology, Coimbatore, India

Abstract: Problem Statement: Handwriting recognition has attracted voluminous research in recent times. The segmentation and recognition of the characters from handwritten scripts incorporates considerable overhead. Almost all the existing handwritten character recognition techniques use neural network approach, which requires lot of preprocessing and hence accomplishing these problems using neural network is a tedious task. **Approach:** In this study we propose a novel solution for performing character recognition in Tamil, the official language of the south Indian province of Tamil Nadu. Pursued by the preprocessing techniques, Segmentation, Normalization and Feature Extraction the approach utilizes octal graph conversion for recognizing off-line handwritten Tamil characters which improves the slant correction. The graph tries to represent the basic form of a letter independent of the style of writing. Using the weights of the graphs and by the appropriate feature matching with the predefined characters, the written characters are recognized. **Results:** The performance evaluation of off line handwritten Tamil character using octal graph conversion and the metrics based on ranks of the letters proves good Recognition Efficiency **Conclusion:** We show that, in practise, the proposed approach produces near optimal results besides outperforming the other methodologies in existence. Results indicate that the approach can be used for character recognition in other Indic scripts as well.

Key words: Optical character recognition, hand written recognition, segmentation, normalization, slant correction, feature extraction, octal graph conversion

INTRODUCTION

Machine simulation of human functions has been a very challenging research field since the advent of digital computers. In some areas, which entail certain amount of intelligence, such as number crunching or chess playing, tremendous improvements are achieved. On the other hand, humans still outperform even the most powerful computers in the relatively routine functions such as vision [1].

In this Overview, Character Recognition (CR) is an umbrella term, which has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CR methodologies and also creates an increasing demand on many emerging application domains, which require more advanced methodologies. Optical Character Recognition (OCR) deals with the recognition of optically processed characters rather than magnetically processed ones. OCR is a process of automatic recognition of characters by computers in optically scanned and digitized pages of text [2]. OCR is one of the most fascinating and challenging areas of pattern recognition with various practical applications. It can contribute immensely to the advancement of an automation process and can improve the interface between man and machine in many applications [3, 4].

Character and handwriting recognition has a great potential in data and word processing for instance, automated postal address and ZIP code reading, data acquisition in bank checks, processing of archived institutional records and more. Combined with a speech synthesizer, it can be used as an aid for people who are visually handicapped. As a result of intensive research and development efforts, systems are available for English language [5,6] Chinese/Japanese languages [7-9]

Corresponding Author: R. Jagadeesh Kannan, Department of Computer Science and Engineering, RMK Engineering College, Kavaraipettai, Tamil Nadu, India

and handwritten numerals[10,11]. However, less attention had been given to Indian language recognition. Some efforts have been reported in the literature for Devanagari [12], Tamil [13] and Bangla [16] scripts. The need for OCR arises in the context of digitizing Tamil documents from the ancient and old era to the latest, which helps in sharing the data through the Internet.

Tamil Scripts: Tamil, the native language of a southern state in India, is one of the old languages in the world has several million speakers and is an official language in countries such as Sri Lanka, Malaysia and Singapore. It has been influenced by Sanskrit to a certain degree. But Tamil is unrelated to the descendents of Sanskrit such as Hindi, Bengali and Gujarati. Most Tamil letters have circular shapes partially due to the fact that they were originally carved with needles on palm leaves, a technology that favored round shapes [14].

However, the case of Tamil is relatively simpler compared to other Indian scripts: the rules for character composition are far fewer than in other Indian scripts.

Tamil has a large alphabet size, which has 12 vowels, 23 consonants and one special character (AK). Vowels and consonants are combined to form composite letters; making a total of 247 different characters [15]. The only category of composition allowed is of Consonant-Vowel type, where a structure corresponding to a consonant and another corresponding to vowel are combined to form a C-V type character with a unique shape. Even this composition does not occur for all C-V combinations. In many cases the vowel modifier appears as a horizontally isolatable structure and hence text entry through QWERTY keyboard is cumbersome. The penetration of Information Technology (IT) becomes harder in a country such as India where the majority read and writes in their native language. Therefore, enabling interaction with computers in the native language and in a natural way such as handwriting is absolutely necessary.

Handwritten Character Recognition: Handwriting recognition inherited a number of technologies from Optical Character Recognition (OCR). The main difference between handwritten and typewritten characters is in the variations that come with handwriting. It is also worth noticing that OCR deals with offline recognition while handwriting recognition may be required for both on-line and off-line signals. (On-line means that data is captured as it is written. For offline, all the data is collected before processing starts).

On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements X(t),Y(t) as well as pen-up/pen-down switching. That kind of data is known as digital ink and can be regarded as a dynamic representation of handwriting. While the input of OCR is usually entire documents, the input of handwriting recognizers can be entire handwritten documents or small snippets of digital ink [17-19].

Off-line processing is able to use only snapshots of the handwriting without time information. From off-line data, we do not have information on the order of strokes the user has used to write the character. It involves the automatic conversion of text in an image into letter codes, which are usable within computer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting. The method we chosen are comparatively difficult for Offline handwriting Recognition. Because Recognition of any handwritten characters with respect to any language is difficult, since, the handwritten characters differ not only from person to person but also according to the state of mood of the same person. Among different branches of character recognition it is easier to recognize English alphabets [20] and Numerals [21,22] than Tamil characters[23,24].

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 [25], structural matching (on trees, chains and more) and stochastic processing (Markov chains and more) have been used along with different types of features. Many recent approaches mix several of these techniques together in order to obtain improved reliability, despite wide variation in handwriting. So we proposed a new approach, which solves the problems faced by existing approaches. The main advantage of the proposed system is which overcomes the problem of slant correction [26]. Slant is one of the characteristics that make handwriting harder to process automatically than printed text. The slant correction is so perfect that it recognizes characters written upside down too. Also the recognition is highly scalable.

MATERIALS AND METHODS

The proposed approach ventures a solution for offline handwritten recognition, which converts the letter written into an octal graph, by representing each pixel of the given character as a node of a graph. Each node has eight fields so termed as octal graph. The graph tries to represent the basic form of a letter independent of the style of writing. Using the weights of the graphs and by the appropriate feature matching with the predefined characters, the written characters are recognized. Figure 1 shows the octal graph representation of a Tamil character. An octal graph unlike a normal graph has a node with eight pointers and a data field. Based on the neighbouring pixels the pointer values are assigned to the various fields of the octal node. These octal nodes are connected to the other nodes based on the threshold value. Figure 2 shows the Octal Node Representation of a Tamil Character.

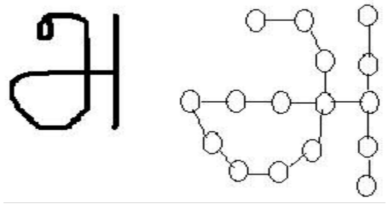

Fig. 1: Octal graph representation of a Tamil character

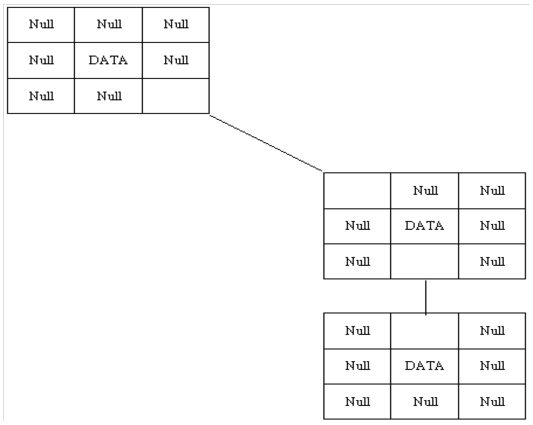

Fig. 2: Octal Node Representation of a Tamil Character

Recognition Algorithm: The main objective of the architectural design of the recognition algorithm is to segment the characters accurately and efficiently and then converted the handwritten Tamil character in to octal graph. Figure 3 show the architectural design. Figure 4 shows the Recognition sequence diagram.

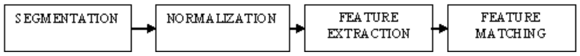

Fig. 3: Architectural designs

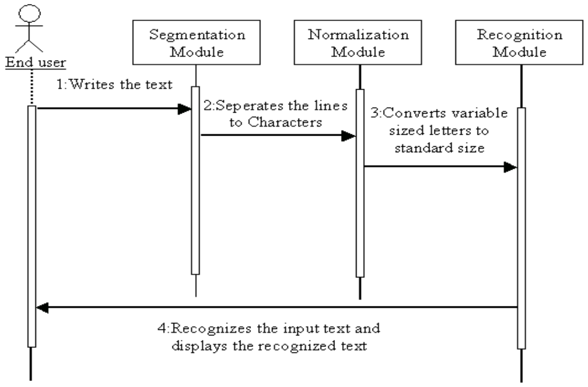
End user
Segmentation Module
Normalization Module
Recognition Module
1:Writes the text
2:Seperates the lines to Characters
3:Converts variable sized letters to standard size
4:Recognizes the input text and displays the recognized text

Fig. 4: Recognition sequence diagram

Segmentation: Text line segmentation is an essential pre-processing stage for off-line handwriting recognition in many Optical Character Recognition (OCR) systems. It is an important step because inaccurately segmented text lines will cause errors in the recognition stage. Text line segmentation of the handwritten documents is still one of the most complicated problems in developing a reliable OCR [27].

Handwriting text line segmentation approaches can be categorized according to the different strategies used [28]-31]. These strategies are projection based, smearing, grouping, Hough-based [32], graph-based and Cut Text Minimization (CTM) approach [33-35]. Fig: 5 show the Segmentation representation of a character.

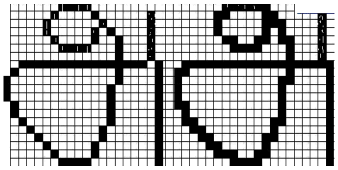

Fig. 5: Segmentation

The segmentation process separates the individual characters from the given input and the consecutive steps are as follows:

- Assign a number for each pixel with the top left and assign a number 1 for that pixel and move row by row
- If the pixel is unvisited and it has a visited pixel as its neighbour, assign its number to the current pixel. In case of more than one visited neighbouring pixel break the tie by choosing the least number. Also for an unvisited pixel with no visited neighbour, assign the next number to the pixel and continue till all pixels are assigned numbers

- In the end, assign the same number for all the continuous patterns by traversing each pattern by DFS or BFS. Now each letter is assigned a unique numbers and can be separated with those numbers

Conversion of Input into Octal Graph: The input character is converted into an octal graph by following steps: Normalization, Conversion, Identification of weighing factors and Feature matching.

***Normalization*:** Before feature extraction, a number of normalization operations are applied to a text line. The normalization operations are slant correction, width normalization and vertical scaling. It was pointed out in[36] that these operations are important to achieve high text recognition rates. Because a high text recognition rate implies a high writer identification rate [37,38], normalization is expected to be beneficial not only for text recognition, but also for writer identification. On the other hand, it can be argued that normalization is potentially harmful to writer identification, because the slant, the character width and the proportions between ascender, middle and descended zones of a text line contain writer specific information.

- **Slant correction:** First the normalization operation applied to a text line and the goal is to bring the handwriting into an upright position [39]. To correct the slant, the angle between the actual, quasi-vertical strokes and the y-axis has to be known. To calculate this angle, the angle distribution of the writing's contour points is accumulated in an angle histogram. The maximum value of the histogram is the slant angle. This angle is then used to normalize the slant of a text line.
- **Width normalization:** The goal of this operation is to scale the width of a text line so that the average character length is normalized.
- **Vertical scaling:** In that a text line can be divided into three zones: a zone containing the ascenders, the middle zone and a zone containing the descended. Our goal is to normalize the height of each of these three zones to a fixed size. This normalization is important in order to reduce the variability of the features

While normalizing images of various sizes into a single standard size, there may be unwanted pixels that are set in a single stroke. This, if passed to the next stage as such, would result in complications in octal graph construction. Normalization is the conversion of images of various dimensions into fixed dimensions. Thus the normalized image map is subject to cleaning to remove any pixels such that no pixels that form a single line have more than two neighbouring set pixels. This is done by looking for pre defined pattern of pixels and removing them for cleaning. Normalisation of a character is shown in Fig: 6.

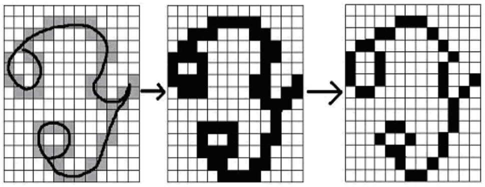

Fig. 6: Normalization

Octal Graph Conversion: The algorithm used for converting the image into an octal graph is critical because the octal graph should represent the features of the letter to a good extent. There are two factors that represent the extent to which the graph represents a letter. They are:

- The distance between two nodes should be high enough to represent the features of the letter correctly and also low enough not to take up much memory
- The number of directions in which linkages are possible is chosen as eight. This is because the number of directions in which the linkages are possible must be high enough to express the curvature of the letters correctly. Also it must be low enough to avoid a highly sparse direction pointer array

The normalized image is then converted into an octal graph by considering the following steps:

- Count the number of set neighbouring cell of each set cell and Push all the cell locations that have more than two neighbours or only one neighbour into a queue
- If no elements are inserted into the queue, insert the top left cell into the queue and make all these cells in the queue as nodes of a graph
- Repeat the process continuously for all set cells. If a cell has no unvisited set cell and none of its neighbouring cells are nodes, make the cell as a new node
- Repeat the process till no new node is created in iteration and connect all the nodes created so far with proper direction linkages. Below Fig: 7 represent the octal graph formation.

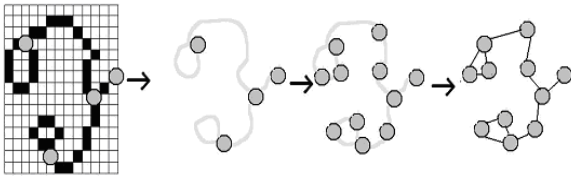

Fig. 7: Octal graph Formation

Identification of Weighing Factors: The weighing factors of the input graph are identified. This includes the loops, horizontal lines, vertical lines and so on. This weighing factor helps in finding the desired match between the input graph and the character that is in the repository. Identification of weighing factors of a character is shown in Fig: 8.

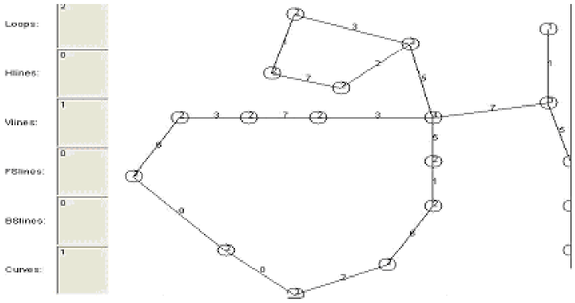

Fig. 8: Identification of weighing factors

Feature Extraction: A feature that varies from one individual writer to another is the slant of the handwriting. Training a classifier on writing that hasn't been slant corrected may require significantly more effort and training data. Consequently, slant removal is an operation that is found in almost any handwriting recognition system. A large number of features for isolated character recognition and corresponding extraction methods have been proposed. They include moments and quantities derived from series expansion, features based on projection profiles and on the contour[40], as well as structural features such as endpoints, fork points, holes, length, shape, or curvature of individual strokes that occur as part of the character under consideration[41,42].

Feature Matching: Match the input graph with that of the character in the repository by considering various factors such as loops, horizontal lines, vertical lines and more. Compare the weighing factors with that of the characters in the repository. Exchange the values of similar weighing factors and compare again these weighing factors. Compare the level of confidence of each level. The result with the highest level of confidence is displayed as the result. The major decisions that are to be taken when considering feature matching are:

- The features that are to be considered for matching
- The weight that is to be given to each feature

The activity diagram of the proposed approach is shown in Fig. 9.

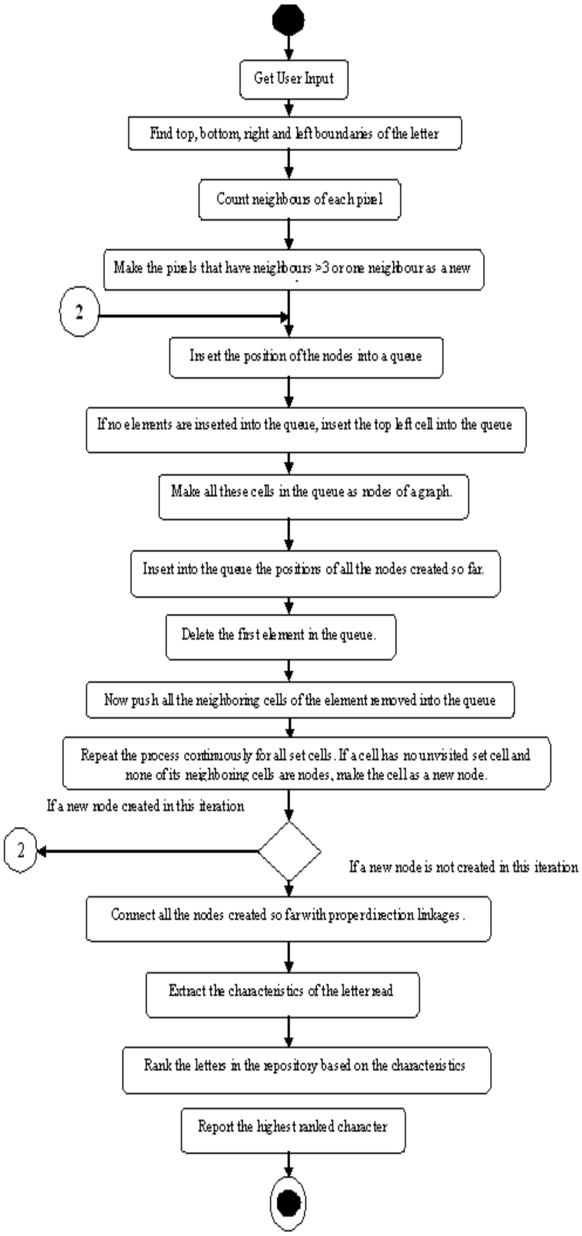
Get User Input
Find top, bottom, right and left boundaries of the letter
Count neighbours of each pixel
Make the pixels that have neighbours >3 or one neighbour as a new
2
Insert the position of the nodes into a queue
If no elements are inserted into the queue, insert the top left cell into the queue
Make all these cells in the queue as nodes of a graph.
Insert into the queue the positions of all the nodes created so far.
Delete the first element in the queue.
Now push all the neighboring cells of the element removed into the queue
Repeat the process continuously for all set cells. If a cell has no unvisited set cell and none of its neighboring cells are nodes, make the cell as a new node.
If a new node created in this iteration
2
If a new node is not created in this iteration
Connect all the nodes created so far with proper direction linkages.
Extract the characteristics of the letter read
Rank the letters in the repository based on the characteristics
Report the highest ranked character

Fig. 9: Activity diagram

RESULTS AND DISCUSSION

An extensive performance evaluation for handwritten Tamil character using octal graph conversion and the metrics we evaluated are listed shows good Recognition Efficiency. The recognition ranks the letters and displays the top 3 ranked letters. If the letter is in the first spot, then it is 100% success, if it is in the second, it is 75% success, if it is in the third, then it is 50% success or else its failure. Based on this ranking an evaluation was done for each letter in the learning set and efficiency of recognition was evaluated for each letter. In Fig. 10, the graph displays the result of the evaluation.

The overall efficiency of the system was found to be 82%. The evaluation cases consisted of ten samples, which consist of 4 good samples, 4 misaligned samples and 2 extremely disfigured samples. This performance of our system is high considering the fact that the existing systems do not recognize disfigured inputs and misaligned inputs at all.

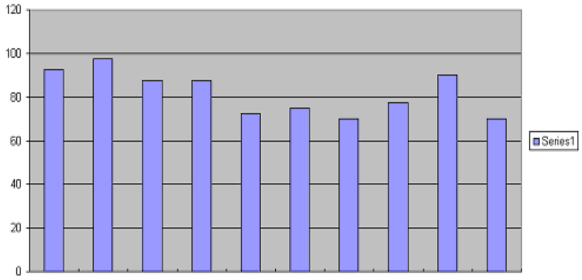

Fig. 10: Graph-recognition Efficiency

CONCLUSION

In this study we have proposed a new method of Handwritten Tamil Character Recognition where we have used octal graph conversion method to get the maximum possible efficiency of the algorithm. Classification and Normalization of off-line handwritten characters has been proven to be efficient on application of octal graph conversion models, which improves slant correction. Significant increase in accuracy levels has been found on comparison of our method with the others for character recognition. With the addition of sufficient pre processing the approach offers a simple and fast structure for fostering a full OCR system. The experimental results show that the accuracy is really improved than the previous study.

REFERENCES

1. Arica, N. and F.T. Yarman-Vural, “An overview of character recognition focused on off-line handwriting”, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol.31, Issue.2, pp. 216 - 233, May 2001, DOI: 10.1109/5326.941845.
2. Pal, U. and B.B. Chaudhuri, “Indian script character recognition: A survey”, Pattern Recognition, vol. 37, no.9, pp. 1887-1899, 2004, DOI:10.1016/j.patcog.2004.02.003.
3. Mantas, J., “An overview of character recognition methodologies”, Pattern Recognition, Volume 19, Issue 6, pp. 425-430, 1986, DOI: 10.1016/0031-3203(86)90040-3.
4. Govindan, V.K. and A.P. Shivaprasad, “Character recognition - a review”, Pattern Recognition, Volume 23 , Issue 7, pp.671 - 683, July 1990, DOI: 10.1016/0031-3203(90)90091-X.
5. R.M. Bozinovic, S.N. Srihari, "Off-Line Cursive Script Word Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 1, pp. 68-83, Jan., 1989, DOI: 10.1109/34.23114.
6. Jianying Hu, Michael K. Brown, William Turin, "HMM Based On-Line Handwriting Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 1039-1045, October 1996, DOI: 10.1109/34.541414.
7. Deng, D., K.P. Chan and Y. Yu, “Handwritten Chinese character recognition using spatial Gabor filters and self-organizing feature maps”, Proceeding of the IEEE International Conference on Image Processing, Volume 3, pp. 940 - 944, 13-16 Nov 1994, DOI: 10.1109/ICIP.1994.413707.
8. Chang, C.H., “Simulated annealing clustering of Chinese words for contextual text recognition”, Pattern Recognition, Volume 17, Issue 1, pp. 57-66, 10 January 1996, DOI:10.1016/0167-8655(95)00080-1.
9. Yamada, H., K. Yamamoto and T. Saito, “A non-linear normalization method for hand printed Kanji character recognition-line density equalization”, 9th International Conference on Pattern Recognition, vol.1, pp. 172-175,14-17 Nov 1988, DOI: 10.1109/ICPR.1988.28198.
10. S-W Lee, “Off-line recognition of totally unconstrained handwritten numerals using multiplayer cluster neural network”, IEEE Trans. on Pattern Anal. Mach. Intell., vol. 18, no. 6, pp. 648-652, June 1996, DOI: 10.1109/34.506415.

11. Jinhai Cai, Zhi-Qiang Liu, "Integration of Structural and Statistical Information for Unconstrained Handwritten Numeral Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 3, pp. 263-270, Mar., 1999, DOI: 10.1109/34.754622.
12. Bansal, V. and R.M.K. Sinha, “On how to describe shapes of Devanagari characters and use them for recognition”, Proceedings of the Fifth International Conference on Document Analysis and Recognition, pp.410-413, 20-22 Sep 1999, DOI: 10.1109/ICDAR.1999.791811
13. Chinnuswamy, P. and S.G. Khrishnamoorthy, “Recognition of hand printed Tamil characters”, Pattern Recognition, Volume 12, Issue 3, 1980, pp. 141-152, DOI: 10.1016/0031-3203(80)90038-2.
14. N.Dhamayanthi, and P.Thangavel, "Handwritten Tamil character recognition using neural network", in: Proc. of Tamil Internet 2000, K. Kalyanasundram and S. Senthil Nathan (Eds.), Singapore, pp. 171-176, July 22-24, 2000. http://www.infitt.org/ti2000/tamilinaiyam/conference_hub.html
15. R. M. Suresh, L. Ganesan, "Recognition of Printed and Handwritten Tamil Characters Using Fuzzy Approach," iccima, pp.291-286, Sixth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'05), 2005, DOI: http://doi.ieeecomputersociety.org/10.1109/ICCIMA.2005.47
16. Chaudhuri, B.B. and U. Pal, “A complete printed bangla OCR system Pattern Recognition”, Volume 31, Issue 5, 1 March 1998, Pages 531-549, DOI:10.1016/S0031-3203(97)00078-2.
17. Deepu, V., S. Madhvanath and A. Ramakrishnan, “Principal component analysis for online handwritten character recognition”, Proceedings of the 17th International Conference on Pattern Recognition, August 23-26, 2004, pp: 327-330. DOI: 10.1109/ICPR.2004.708.
18. Tappert, C., C. Suen and T. Wakahara, “The state of the art in online handwriting recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 12, Issue 8, Aug 1990, pp. 787 - 808, DOI: 10.1109/34.57669.
19. Impedovo, Sebastiano. "Frontiers in Handwriting Recognition", Fundamentals in Handwriting Recognition. Ed. Sebastiano Impedovo. New-York: Springer-Verlag, 1994. ISBN: 0-387-57450-6.
20. Suresh, R.M. and S. Arumugam, “Fuzzy context-free grammar to handwritten numerical recognition”, Proceedings. Third International Conference on Computational Intelligence and Multimedia Applications, pp. 459 - 463, 1999, DOI: 10.1109/ICCIMA.1999.798574.
21. M. Hanmandlu, O.V. Ramana Murthy, “Fuzzy model based recognition of handwritten numerals", Pattern Recognition, Volume 40, Issue 6, pp: 1840-1854, June 2007, DOI: 10.1016/j.patcog.2006.08.014.
22. Suresh, R.M., S. Arumugam and L. Ganesan, “Fuzzy approach to recognize handwritten Tamil characters”, Proceedings. Third International Conference on Computational Intelligence and Multimedia Applications, 1999, pp. 459 - 463, DOI: 10.1109/ICCIMA.1999.798574
23. Sethi, I.K. and B. Chatterjee, “Machine recognition of constrained hand printed devanagari”, Pattern Recognition, Volume 9, Issue 2, Pages 69-75, July 1977, DOI:10.1016/0031-3203(77)90017-6.
24. G. Siromoney, R. Chandra sekaran and M. Chandra sekaran, “Computer recognition of printed Tamil character”, Pattern Recognition, Volume 10, Issue 4, Pages 243-247, 1978, DOI: 10.1016/0031-3203(78)90032-8.
25. Senior, A.W., “Off-line handwriting recognition: A review and experiments”, Technical Report 105, Cambridge University Engineering Department, England, December 1992, http://www.research.ibm.com/people/a/aws/documents/papers/senior_tr105.pdf.
26. Guillevic, D. and C.Y. Suen, “Cursive script recognition: A sentence level recognition scheme”, International Workshop on the Frontiers of Handwriting Recognition (IWFHR), pp: 216-223, December, 1994. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.6925
27. Likforman-Sulem, L., A. Zahour and B. Taconet, “Text line segmentation of historical documents: A survey”, International Journal on Document Analysis and Recognition, Volume 9, Issue 2, pp.123 - 138, 2007, DOI: 10.1007/s10032-006-0023-z.
28. Nicolas, S., T. Paquet and L. Heutte, “Text Line segmentation in handwritten document using a production system”, Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, October 26-29, pp: 245-250, 2004. DOI: 10.1109/IWFHR.2004.100.
29. Sriganesh Madhvanath, Evelyn Kleinberg, Venu Govindaraju, "Holistic Verification of Handwritten Phrases," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 12, pp. 1344-1356, December, 1999, DOI: http://doi.ieeecomputersociety.org/10.1109/34.817412.

30. Y. Pu, Z. Shi, “A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents”, Advances in Handwriting Recognition by Seong-Whan Lee, World Scientific Publishing, pages 141-152, 1999, ISBN: 9810237154, 9789810237158.
31. Shridar, M. and F. Kimura, “Segmentation-Based Cursive Handwriting Recognition”, In: Handbook of Character Recognition and Document Image Analysis, Bunke, H. and P.S.P. Wang (Eds.). World Scientific Publishing Company, pp: 123-156, 1997, ISBN 981022270X 9789810222703.
32. G. Louloudis, K. Halatsis, B. Gatos, I. Pratikakis, "A Block-Based Hough Transform Mapping for Text Line Detection in Handwritten Documents", 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), pp. 515-520, La Baule, France, October 2006, http://www.iit.demokritos.gr/~bgat/cr1062.pdf
33. Z. Shi and V. Govindaraju, "Line Separation for Complex Document Images Using Fuzzy Run length" 1st Workshop on Document Image Analysis and Libraries, pp. 306-313, Palo Alto, CA, 2004. DOI: 10.1109/DIAL.2004.1263259.
34. Shi, Z. and V. Govindaraju, “Historical document image enhancement using background light intensity normalization. Proceedings of the 17th International Conference on Pattern Recognition, Volume: 1, pp. 473- 476, 23-26 Aug. 2004, DOI: 10.1109/ICPR.2004.1334167.
35. Sesh Kumar, K.S., A.M. Namboodiri and C.V. Jawahar, “Learning segmentation of documents with complex scripts”, Lecture Notes in Computer Science, Volume 4338/2006, pp.749-760, ISBN: 978-3-540-68301-8, DOI: 10.1007/11949619_67.
36. Marti, Urs-Viktor and Horst Bunke, “Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system”, World Scientific Series in Machine Perception and Artificial Intelligence Series, pp. 65 - 90, 2001, ISBN: 981-02-4564-5
37. Schlapbach, A. and H. Bunke, “Off-line handwriting identification using HMM based recognizers”, Proceeding of the 17th International Conference on Pattern Recognition, IEEE Computer Society, USA., August 23-26, pp: 654-658, 2004. DOI: 10.1109/ICPR.2004.1334343.
38. Schlapbach Andreas and Horst Bunke, “Using HMM based recognizers for writer identification and verification”, Proceeding of the 9th International Workshop on Frontiers in Handwriting Recognition, IEEE Computer Society, USA., October 26-29, pp: 167-172, 2004. DOI: 10.1109/IWFHR.2004.107.
39. Caesar, T., J.M. Gloger and E. Mandler, “Preprocessing and feature extraction for a handwriting recognition system”, proceeding of the 2nd International Conference on Document Analysis and Recognition, October 20-20, pp: 408-411, 1993. DOI: 10.1109/ICDAR.1993.395706.
40. Thien M. Ha, Horst Bunke, "Off-Line, Handwritten Numeral Recognition by Perturbation Method," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 5, pp. 535-539, May, 1997, DOI:http://doi.ieeecomputersociety.org/10.1109/34. 589216.
41. Suen, C.Y., C. Nadal, R. Legault, T.A. Mai and L. Lam, “Computer recognition of unconstrained handwritten numerals” Proceedings of the IEEE, Volume 80, Issue 7, Jul 1992, pp.1162 - 1180, DOI: 10.1109/5.156477.
42. Franke, J., “Isolated hand printed digit recognition”, In: Bunke, H., Wang, P.S.P. (Eds.), Handbook of Character Recognition and Document Image Analysis, World Scientific Publishing Company, Singapore, pp. 103-121, 1997, ISBN 981022270X, 9789810222703.