A complete OCR for printed Tamil text

A.G. Ramakrishnan and Kaushik Mahata
Dept. of Electrical Engg, Indian Institute of Science, Bangalore 560 012, India

---

Abstract:

A multi-font, multi-size Optical Character Recognizer (OCR) of Tamil Script is developed. The input image to the system is binary and is assumed to contain only text. The skew angle of the document is estimated using a combination of Hough transform and Principal Component Analysis. A multi-rate-signal-processing based algorithm is devised to achieve distortion-free rotation of the binary image during skew correction. Text segmentation is noise-tolerant. The statistics of the line height and the character gap are used to segment the text lines and the words. The images of the words are subjected to morphological closing followed by connected component-based segmentation to separate out the individual symbols. Each segmented symbol is resized to a pre-fixed size and thinned before it is fed to the classifier. A three-level, tree-structured classifier for Tamil script is designed. The net classification accuracy is 99.1%.

METHODOLOGY

OCR involves skew detection and correction followed by character segmentation and recognition of segmented symbols. Operations involved in each step are elaborated below.

Skew Correction

The input binary image is first corrected for skew. We have developed a precise skew detection algorithm [1], which estimates the skew angle in two steps. A coarse estimate of the skew is obtained through interim line detection using Hough Transform [2]. The interim lines are the lines that bisect the backgrounds in between the text lines. The coarse estimate is used to segment the text lines, which are superposed on each other and the direction of the principal axis [3] of the resulting image with the larger variance is taken as the fine skew direction. The accuracy of the final estimate is +/- 0.06o. A multi-rate-signal-processing based algorithm is devised to achieve distortion-free rotation of the binary image during skew correction [4].

Text Segmentation

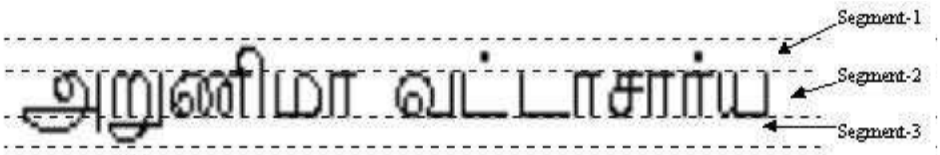
The text lines are segmented using the horizontal projection profile of the document image [5]. Subsequently, the words are segmented using the vertical projection profile. The statistics of line-height and symbol-gap are extracted first. During text line segmentation, the average line height is used to split those pairs of text lines, which cannot be segmented separately due to noise. Since some of the Tamil characters are made up of 2 or 3 disconnected symbols, we use the term symbol to denote each connected component, as different from a character. The symbol-gap statistics is used to distinguish a word boundary from a symbol boundary. From the segmented words, individual symbols are separated by successive application of the morphological closing and connected component-based segmentation algorithm [2].

Morphological closing helps in filling the gaps in the broken characters. Connected Component Analysis is useful when the symbols cannot be segmented using vertical projection profile only.

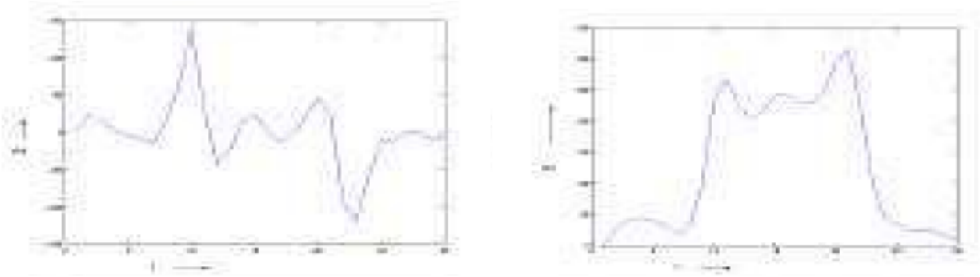
The case for a tree structured classifier for Tamil Characters

The segmented symbols are fed to the classifier for recognition. We use a classificationstrategy, which first identifies the individual symbols, and in a subsequent stage, combines the appropriate number of successive symbols to detect the character. It is desirable to divide the set of 154 different symbols into a few smaller clusters, so that the search space while recognition is smaller, resulting in lesser recognition time and smaller probability of confusion. The above objective is accomplished by designing a three level, tree structured classifier to classify Tamil script symbols.

First Level Classification Based on Height



The text lines of any Tamil text will have three different segments. We name them Segment-1, Segment-2, and Segment-3, as shown in Fig.1. Since the segments occupied by a particular symbol are fixed and remain invariant from font to font, a symbol can be associated with one of the four different classes depending upon its occupancy of these segments. Symbols occupying segment-2 only are labeled as Class-0 symbols. Those occupying segment-2 and segment-1 are termed as Class-1 symbols. Those occupying segment-2 and segment-3 are named as Class-2 symbols. Symbols occupying all of them are called as Class-3 symbols. Almost all the symbols



in Tamil occupy the segment-2 and about 60% of the symbols belong to Class-0. Thus, the horizontal projection value of any row in the segment-2 is large compared to that of a row of the segments 1 or 3. The sharp rise and the fall in the horizontal projection profile p[n] indicate the transition from segment-1 to segment-2 and the transition from segment-2 to segment-3 respectively (Refer Fig.2.). These correspond to the sharp maximum and the minimum in its first difference q[n], which is given by

$$q[n] = p[n] - p[n-1] \text{ , } n>0 \qquad (1)$$
$$p[0] = q[0].$$

The line-boundary between the segments 1 & 2 denoted by Line_1 is given by the value of n for which q[n] is maximum in the upper half of the text line. Similarly, the boundary between the segments 2 & 3 denoted by Line_2 is given by the value of n for which q[n] is minimum in the lower half of the text line. An unknown symbol segmented from the text line under consideration can now be classified accordingly.

Second Level Clustering based on matra/extensions

Symbols of class-1 and class-3 have their extensions in segment-1. The set of symbols in class-1 is divided into three groups (Groups 1, 2, and 3) based on their extensions in segment-1 (Refer Fig. 3.). Similarly, Class-2 symbols are clustered into five groups (Groups 4, 5, 6, 7, and 8) based on their extension in the segment-3 (Refer Fig.4.). No further script dependent clustering is performed among the Class-0 and Class- 3 symbols.

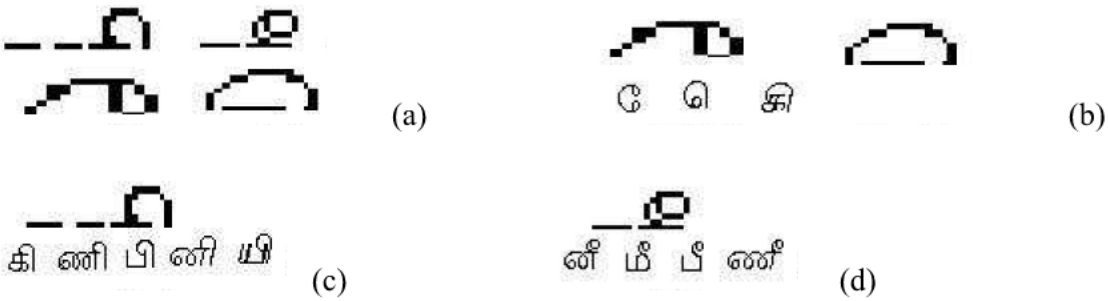


Figure 3: Illustration of second level classification in Class-1. (a) Different types of extensions of class-1 symbols captured in segment-1; (b) Group-1 symbols used and the corresponding extensions; (c) Group-2 symbols and corresponding extensions; (d) Group-3 symbols and extensions.

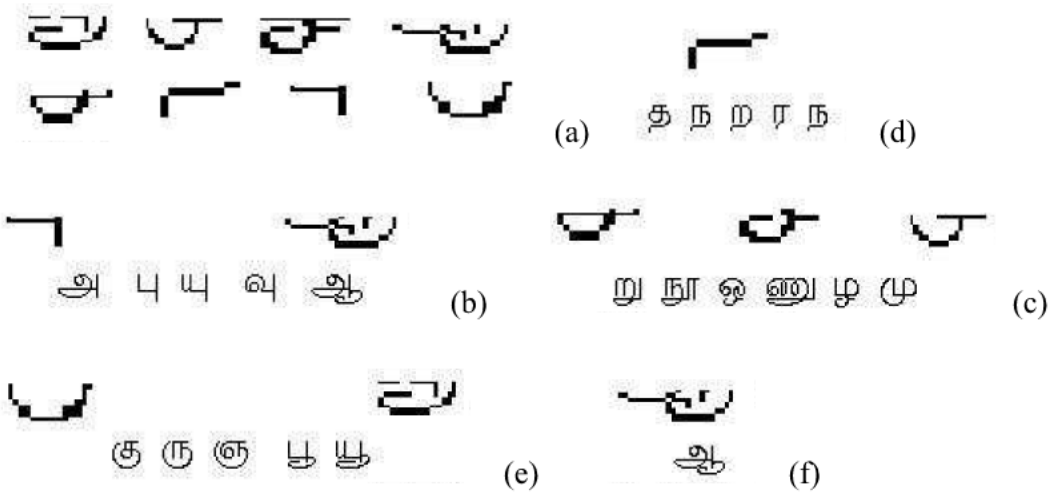


Figure 4: Illustration of second level classification in Class-2. (a) Different types of extensions of Class-2 symbols captured in segment; (b) Group-4 symbols and the corresponding extensions, (c) Group-5 symbols and corresponding extensions; (d) Group-6 symbols and extensions, (e) Group-7 symbols and corresponding extensions; (f) Group-8 symbols and the corresponding extensions.

The rectangle containing the thinned symbol is found out. The portion of the rectangle captured in the segment-1 or 3 (as applicable) is resized to a 30x30 image. This image is thinned and divided into four 15x15 blocks. Second moments [2] are calculated from each block to obtain the 12-dimensional feature vector. Nearest neighbor classifier [6] using Euclidean distance is used for classification. Thinning algorithm proposed by Zhung and Suen [7] is employed.

The tree structure of the classifier is shown in Fig.5.

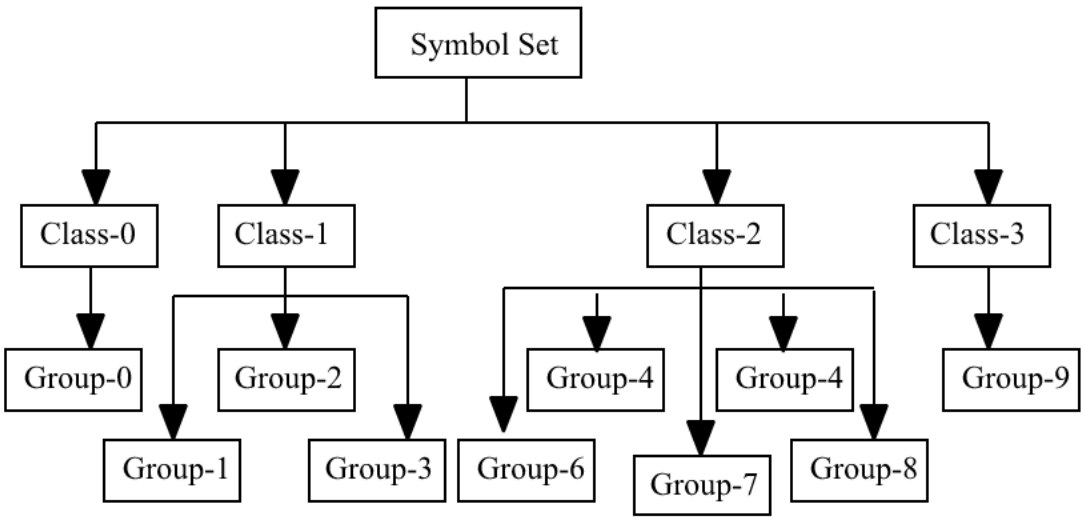


Fig.5 Tree structure of the classifier

Fig. 6. Example of Class-1 normalisation (a) Class-1 symbol, (b) Normalized symbol, (c) segment-1 extension separated

Fig. 7. Example of Class-2 normalisation. (a) Class-2 symbol, (b) Normalized symbol; (c) segment-2 extension separated

Recognition at the third level

In the third level, feature-based recognition is performed. The symbols are to be normalized first to a predefined size to make it possible to compare them with those in the training set. The normalization strategy varies from group to group. First, the rectangle containing the symbol is

cropped. The cropped rectangle is interpolated to a size of 45x60 and thinned if the symbol belongs to Class-0. For a symbol belonging to class-1, 2 or 3, the portion of the cropped rectangle captured in the segment-1 or 3 is normalized to a rectangle of height 10. The portion of the rectangle captured in the segment-2 is normalized to a rectangle of height 50, keeping the same normalized width. These individual images are concatenated back and thinned to get the normalized symbol (Refer Figs. 6 & 7). The normalized width is 45 for group-1. It is 60 for the groups 3, 4, 6, 7, 8, 9. The width for groups 2 and 5 is 75. This normalization strategy helps to bring in the font independence in the OCR. Geometric moment features are extracted from the normalized symbols. The normalized symbols are split into 15x15 non-overlapping blocks and from each block, second order geometric moments are calculated. Nearest neighbour classifier using Euclidean distance is employed to recognize the symbols. A symbol is rejected if the distance to its nearest neighbour is larger than a predefined threshold. The value of the threshold is taken as 30.

Classification Results

Training set is generated form the symbols extracted from regular Tamil texts appearing in books. The algorithm is tested on some other pages of the same texts. Some of the symbols are very rare in regular Tamil texts. These symbols belong to Group-3, Group-5 and Group-9. Computer generated font is used for both the training and the test set for these symbols. The summery of the results is given in the following table. The classification accuracy is calculated based on the number of symbols correctly recognized.

	No.of test patterns	No of training patterns	Percentage Recognition Accuracy	Percentage Rejection
Class-0	1832	69	99.4	0.3
Class-1	423	45	98.3	0.3
Class-2	983	69	99.3	0.4
Class-3	122	21	95.2	0.2

Net Classification accuracy is 99.01%.

References

[1] Kaushik Mahata and A. G. Ramakrishnan, Precision Skew Detection through Principal Axis. Submitted to International Conference on Multimedia Processing and Systems, Chennai, Aug. 13-15, 2000.

[2] R.C.Gonzalez & R.E.Woods, Digital Image Processing. Addison-Wesley.

[3] G.Strang, Linear Algebra and its Applications. Academic press.

[4] Kaushik Mahata and A. G. Ramakrishnan, A Signal Processing Approach to Rotation of Document Images, submitted to Intern. Conf. on Commn., Control and Signal Processing in the next millenium, Bangalore, July 25-28, 2000.

[5] T.Akijama & N.Hagita, Automatic entry system for printed documents. Pattern Recognition, vol 23, pp 1141 - 1154, 1990

[6] R.O.Duda & P.E.Hart, Pattern Classification and Scene Analysis. John Wieley & Sons.

[7] T.Y.Zhung & C.Y.Suen, A fast parallel Algorithm for thinning digital patterns. Comm ACM, vol. 27, no. 3, pp. 337-343.

Handwritten Tamil Character Recognition Using Neural Network

N. Dhamayanthi
Department of Computer Science, Engineering & Application
Crescent Engineering College, Vandalur, Chennai - 600 048.
E-mail : dhamay@hotmail.com

P. Thangavel
Department of Computer Science
University of Madras, Chepauk, Chennai - 600 005.

Abstract

A Neural Network approach is proposed to build an automatic off-line handwritten Tamil character recognition system. We have used a Back Propagation Network (BPN) as a character recognizer. Once trained, the network has a very fast response time. However, the learning phase of this recognizer is a relatively difficult task in this application. The input image of the handwritten character is given as input to the BPN and the character most closely resembling the block of pixels is given as output. This system uses a three layer backpropagation neural network .

Keywords : Pattern Recognition; Neural Networks; Backpropagation; Optical Character Recognition; Handwritten Character; Handwritten stroke; Segmentation

1. Introduction

As the developments in the computer field are tremendous, there is a need to improve the man machine interface. If computers can be made intelligent enough to understand human handwritings, it will be possible to make man-computer interfaces more ergonomic and attractive. That is an alternative method of entering data should be devised which should be very user friendly and it should not require a prior knowledge of typing. Many researches are going on in Handwritten Character Recognition and Voice Recognition. Users who need to type scores of page everyday should have prior knowledge of typing to use the traditional keyboard. So if we could develop a system which can recognize the characters out of users hand strokes, it would be a boon to those who find it very easy to write instructions rather than type it. Thus this work is carried out to realize the dream of replacing the traditional keyboard with an electronic paper.

Recently Tamil is being extensively used in computers by international Tamil community. As Tamil is official and spoken language in several foreign countries, the use of Tamil in Information Technology will be more in future. In order to promote this further, a system is

developed to recognize the handwritten Tamil Characters, which may be useful for recognizing Tamil texts.

The origin of character recognition can be found in 1870 when Carey invented the retina scanner, an image-transmission system using a mosaic of photo-cells. Recognition of isolated units of writing, such as a character, numeral or a word has been extensively studied in literature [1-10]. In this paper, we have designed a three-layer neural network model using backpropagation algorithm for recognition of off-line handwritten Tamil character. This paper is organized as follows. Section 2 briefs about the character recognition problem. In section 3, we introduce the concept of Artificial Neural Networks. Section 4 shows the architecture of our system and explains implementation of BPN to recognize handwritten character. Experimental results and discussions are presented in section 5 and conclusion is given in section 6.

2. The Character Recognition Problem

The field of Character Recognition can be divided into two classes, off-line recognition and on-line recognition. On-line recognition refers to the recognition mode in which the machine recognizes the handwriting while the user writes on the surface of a digitizing tablet with an electronic pen. The digitizing tablet captures the dynamic information about handwriting such as number of strokes, stroke order, writing speed etc. all in real time. Off-line recognition, by contrast, is performed after the handwriting has been completed and its image has been scanned in. Thus, dynamic information is no longer available. Because of the more tightly constrained feature space, the reduced need for segmentation and the ability to train the system, on-line recognition has produced much more encouraging results than off-line recognition for both hand generated print and script.

Machine recognition of handwritten characters continue to be a topic of intense interest among many researchers, primarily due to the potential commercial applications in such diverse fields as document recognition, check processing, forms processing, address recognition etc. The need for new techniques arises from the fact that even a marginal increase in recognition accuracy of individual characters can have a significant impact on the overall recognition of character strings such as words, postal codes, zip codes, courtesy amounts in checks, street number recognition etc.

3. Artificial Neural Networks

The usage of Neural Networks made the process of recognition more efficient and reliable. The properties of the artificial Neural Networks of abstracting essential characteristics from inputs containing irrelevant data, learning from experience and generalizing from previous examples to new ones came in very handy for pattern Recognition and therefore for OCR. Lippmann [4] has reported a comprehensive survey of prominent ANNs. Of the various models, the feed forward model of Multi Layered Perceptron (MLP) has been reported to yield encouraging results by many many researchers. The backprogation algorithm is used in MLP.

4. Implementation of ANN

An Artificial Neural Network (ANN) technique is used for recognizing the correct character from the given input. We have used a completely connected feedforward Neural Network with the classical backpropagation learning algorithm[11-14] more simply known as the Backpropagation Network (BPN). The advantage of using BPN is that, it can be trained to identify various forms of the same character. The following steps are followed while implementing the ANN.

1. An Artificial Neural Network (ANN) using Backpropagation method is first designed.
2. The training data is prepared and is used to train the ANN.
3. After the training is completed, the character to be recognized is given as input.
4. The ANN gives as output, the closest resembling character for each block.

The output of an ANN in the present study is given by :

$$OUT = 1 / (1 + e^{-net})$$

where net is the activation element given by :

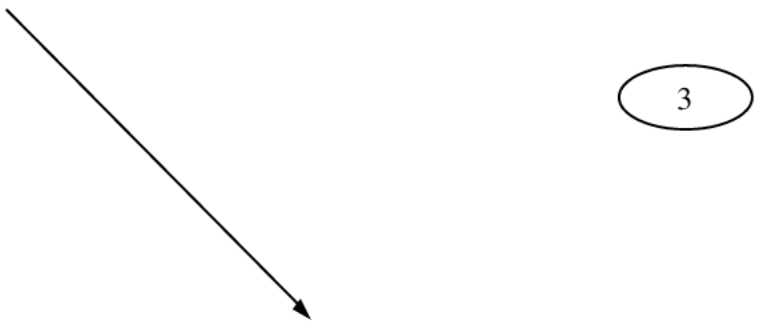
$$net = \sum_{i=1}^{n} w_i x_i$$

n being number of inputs to the neuron.

The neurons are arranged in layers. The user can specify the network topology i.e. the number and size of the hidden layers as well as the values of weights, biases, learning rates and momentum factors.

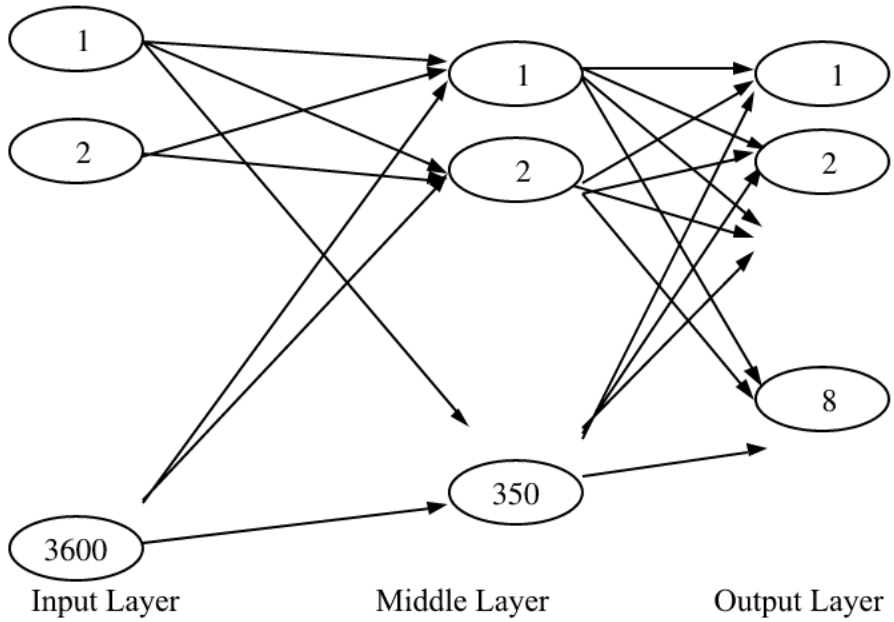
4.1. Designing the Network

To build a BPN, there are many parameters to choose from dealing with the network size or the learning law. Unfortunately, there is no way to determine them rigorously since they are strongly dependent on the application. The first is the number of hidden layers, which has been settled to one [4], since many authors consider that a single hidden layer is sufficient for most applications. The number of neurons on the input layer (Ni) is 3600, since each character is represented in a matrix of 60(60 pixels. The number of neurons of the output layer (No) is eight, since we have to recognize 247 alphabets. We have trained the network only for 30 Tamil characters (vowels & consonants). It is not so easy to find the number of neurons on the hidden layer (Nh) whose upper limit is theoretically 2Ni + 1 [12]. After many trails, we have decided to



have 350 neurons in the hidden layer. The organization of layers for the feedforward backpropagation network used to solve this problem is shown in fig. 1.

Fig.1 Organization of layers of BPN



5. Results and Discussion

The experiment was conducted for various number of cycles. It was found that maximum recognition rate was achieved at 175 cycles. Fig. 2. shows the sample test data. Fig. 3. shows the output as recognized by the network. Table 1 gives the recognition rate achieved for various number of input samples, when the number of neurons in the hidden layer is 350 & number of cycles is 175. Maximum recognition rate of 90% was achieved when 10 input samples were used.

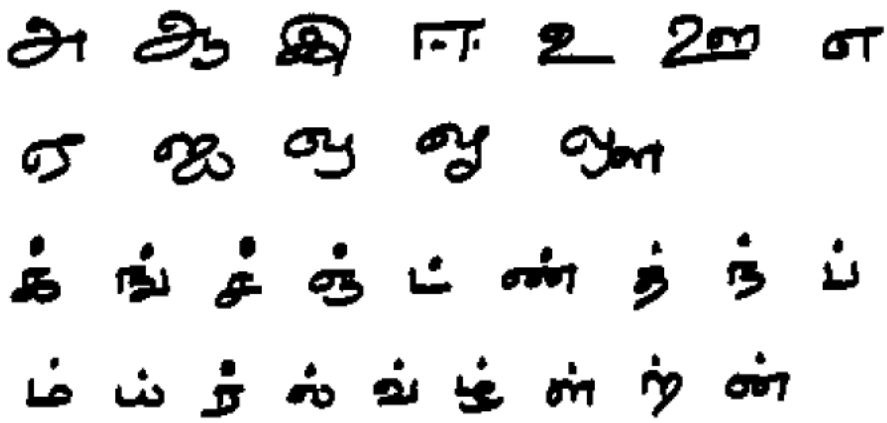


Fig.2. Sample testing input

அ ஆ இ ஈ உ ஊ எ
ஏ ஐ ஒ ஓ ஔ
க் ங் ச் ஞ் ட் ண் த் ந் ப்
ம் ய் ர் வ் வ் ழ் ன் ந் ன்

Fig.3. Output of the sample Test Sample

Fig. 4 Recognition rate for various number of input samples at 175 cycles

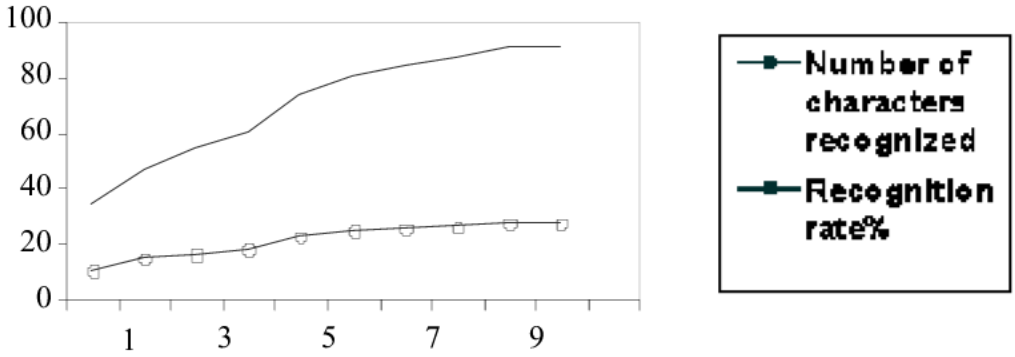


Table 1: Determination of optimum number of input samples

Number of cycles = 175 Number of neurons in the hidden layer = 350
Error tolerance = 0.001 Learning parameter = 0.01

S.No	Number of input samples	Number of characters recognized out of 30	Recognition rate %
1	1	10	33.33
2	2	14	46.7
3	3	16	53.3
4	4	18	60.0
5	5	22	73.3
6	6	24	80.0
7	7	25	83.3
8	8	26	86.7
9	9	27	90.0
10	10	27	90.0

6. Conclusion

In this paper, we have proposed a method to recognize handwritten Tamil characters using a feedforward multilayer Neural Network with backpropagation algorithm. A recognition experiment has been conducted with 10 sets of 30 Tamil Characters (vowels & consonants). The Recognition rate of this experiment is 90%. Our approach is easily extensible to different

character set and different writing styles. For eg., the system can recognize all alphanumeric characters 0-9,'+','-' & '$' if the corresponding templates are added to the reference set. Furthermore, our approach can handle large character sets.

Acknowledgement

N. Dhamayanthi would like to thank the Management, Correspondent, Director, Principal and Prof. & Head of CSE&A department of Crescent Engineering College for their encouragement and motivation.

References

[1] Cao J., Ahmadi M. and Shridhar M., 'A Hierarchical Neural Network Architecture for Handwritten Numeral recognition', Pattern Recognition, vol. 30, No. 2, 1997, pp. 289-294.

[2] Huang J.S. and Chuang K., 'Heuristic Approach to Handwritten Numeral recognition', Pattern Recognition, vol. 19, 1986, pp. 15-19.

[3] Kimura F. and Shridhar M., 'Handwritten numerical recognition based on multiple recognition algorithms', Pattern Recognition, vol. 24, No. 11, 1991, pp. 969-983.

[4] Lippman R.P., 'An introduction to computing with neural nets', IEEE ASSP, April 1987, pp. 4 -22.

[5] Lam L. and suen C.Y., 'Structural classification and relaxation matching of totally unconstrained handwritten Zip code numbers', Pattern recognition, Vol. 21, No. 1, 1998, pp. 19-31.

[6] Seun C.Y., Nadal C., Legault R., Mai T.A. and Lam L., 'Computer recognition of unconstrained handwritten numerals', Proc. IEEE, vol. 80, 1992, pp. 1162-1180.

[7] Shridhar M. and Bedreldin A., 'Recognition of isolated and simply connected handwritten numerals', Pattern Recognition vol. 19, No. 1, 1986, pp. 1-12.

[8] Tappert C.C., Suen C.Y. and Wakahara T., 'The state of art in on-line handwriting recognition', IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, No. 8, 1990, pp. 787-808.

[9] Taxt T., Olafsdottir J.B. and Daehlen M., 'Recognition of handwritten symbols', Pattern Recognition, vol. 23, No.11, 1990, pp. 1155-1166.

[10] Xiaolin L. and Yeung D Y., 'On-line Handwritten Alphanumeric Character Recognition using Dominant Points in Strokes', Pattern Recognition, vol. 30, No.1, 1997, pp. 31-44.

[11] Wasserman P. D., 'Neural computing : Theory and Practice', Van Nostrand Reinhold, New York, 1989.

[12] Freeman J. A., Skapura D.M., 'Neural Networks : Algorithms, Applications and Programming Techniques', Addison-Wesley, New York,1991.

[13] Yegnanarayana B., 'Artificial Neural Networks', PHI, New Delhi, 1999.

[14] Patterson D.W., 'Artificial Neural Networks - Theory and Applications', Prentice Hall, Singapore, 1996.

rm suresh

rm suresh

rm suresh

rm suresh

rm suresh

High precision Optical Character Recognition of Printed Tamil Characters

M K Saravanan,
Design Engineer,
The AU-KBC Centre for Internet & Telecom Technologies,
Madras Institute of Technology, Anna University,
Chromepet, Chennai 600 044 - INDIA
<Email: mksarav@mitindia.edu>

Abstract

To build a digital library reasonably fast from printed text books, we need Optical Character Recognition (OCR) software. Currently OCR packages are available for English, Chinese, and many other foreign languages. So far, no commercial OCR software are available for Indian Languages. Developing OCR package for Indian Languages especially for tamil is a challenging job. Any usable OCR package must have atleast 99% recognition rate. We can easily develop OCR package for Tamil with recognition rate of 85% to 90%. To attain higher recognition rate one has to go for advanced image processing techniques integrated with artificial intelligence, neural networks, graph theory etc.., This paper explains one such advanced approach which uses Optical Font Recognition (OFR) to attain higher recognition rate.

Introduction

Web education, Virtual University, Online electronic libraries etc.., are becoming more popular these days. In coming years we can find large volumes of book in electronic form on Internet. To build a digital library from the available huge collection of printed text books, one must need a high performance OCR package. Currently we have OCR package with reasonable accuracy for English, Chinese and many other foreign languages. Unfortunately we don't have such packages for Indian Languages. Of all the Indian Languages, Tamil is the first one to reach the Internet. Project Madurai (http://www.tamil.net/projectmadurai) is one of the best e.g. for electronic archive of tamil books. Tamilnadu Government has taken all steps to create a Tamil Virtual University. Surely such efforts will involve creation of huge electronic archive of tamil books, which inturn will need a high precision Tamil OCR. To develop such a package, Open Source Code / Free Software is the best solution. To achieve higher recognition rate expertise in the areas such as Digital Image Processing, Artificial Intelligence, Neural Networks, Graph Theory etc.., are necessary. We need lot of volunteers from the respective fields, to share their expertise with others to build a full fledged high precision OCR package for Printed Tamil Characters.

Need for High Recognition Rate

Any OCR software to be really useful it must have atleast 99% accuracy. The running text printed on a A4 size paper can easily contain an average of 2000 characters per page. That

means OCR software with 99% recognition rate will produce 20 errors per page. In manual typewriting, this is the worst case error rate. A good typist, will commit an average of 4 errors per page. If we really want to replace a typist with OCR, it must have atleast 99.9% accuracy. One way we can achieve this recognition rate is by using an OFR system as a part of OCR.

OCR Models

OCR systems can be broadly classified as mono font OCR, multi font OCR and Omni font OCR. Mono font OCR systems are easy to build. Theoretically we can achieve 99.9% recognition rate with mono font OCR. In a multi font OCR system, features will be extracted from a known set of commonly used fonts. These learned features will then be used to compare with the features of the sample text image. It is common to find plain text, italics, bold, and italics-bold with different sizes (10pt, 12pt, 14pt etc..,) in a given text page. In a multi-font OCR system it is very difficult to discriminate each of these features between different fonts. This in turn will considerably reduce the recognition rate. In an omni font OCR system, theoretically it will recognise characters printed with any fonts. But Practically it is impossible to build such a system.

Existing OCR Technologies

Current OCR technologies are largely based on one of the following approach:

(i) Template Matching

It is the most trivial method. Character templates of all the characters from most commonly used fonts are collected and stored in a database. The recognition consists of finding the closest matching template using one of the minimum distance matching algorithms. Template matching techniques assumes the a priori knowledge of the font used in the document and are highly sensitive to noise, skew etc.., in the scanned image. This method is not suitable for omni font OCR system, because character templates of all the variants of the characters in all the fonts must be stored in the database.

(ii) Structural Approach

In this approach, characters are modeled using their topological features. The main concentration will be on structural features and their relationship. The most common methods under this category are

- String matching methods where character are represented by feature string.
- Syntactic methods where character features are determined by the vocabulary and grammar of the given language.
- Graph based methods consists of graph construction where nodes contain features.

All of the above methods are superior to template matching but with respect to omni font OCR we cannot achieve desirable recognition rate using this approach.

(iii) Statistical Approach

This approach is based on the statistical decision theory where each pattern is considered as a single entity and is represented by a finite dimensional vector of pattern features. The most commonly used methods in this category are based on Bayesian classification, stochastic and nearest neighbor classification. In the recent past, classification based on Neural Networks are also used significantly to enhance the recognition rate.

OFR Approach

Optical Font Recognition approach can be used to overcome the limits of existing omnifont OCR technologies. As stated previously monofont OCR will give high recognition rate. If we are able to discriminate the text in various fonts in a document, then they can be submitted to the corresponding monofont OCR engine. This approach is called 'A Priori Optical Font Recognition' [Ref.1]. Fig.2 shows the block digram of the 'A Priori Optical Font Recognition System'. It consists of identifying the text font without any knowledge of the characters that appear in the text. The OFR can be based on features extracted from global properties of the text image, such as the text density, letters size, orientation and spacing etc.., Features may further be extracted from text entities with various lengths such as words, lines, or even paragraphs. Global features can also be tolerant of the image conditions, i.e. they can be extracted from binary image scanned at low resolution.

High Precision OCR System Architecture

Fig.1 shows the overall architecture of the high precision OCR system.

(i) Scanning

The text document is scanned using a flat bed scanner and converted into 8-bit (256 grey level) grey level image. Using appropriate binarisation algorithm this inturn will be converted into a binary (bilevel) image.

(ii) Pre Processing

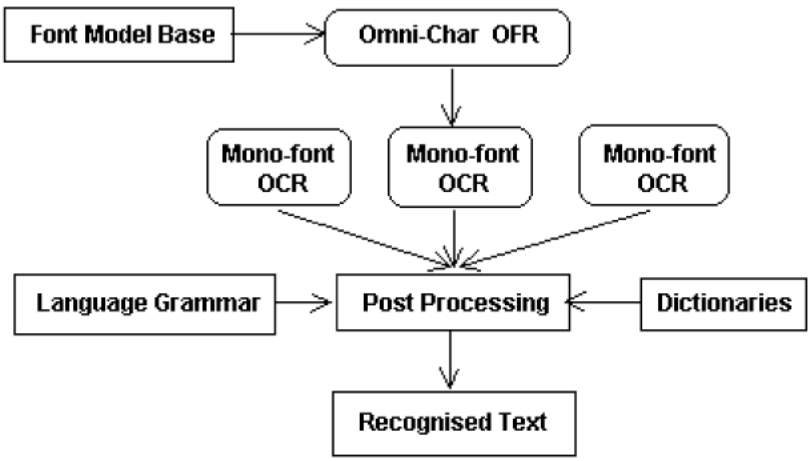


Fig.1 - High Precision OCR System Architecture

Scanned documents almost always contain noise, which results in image degradation. Preprocessing is done mainly to remove the noise and also for skew detection and correction, character contour smoothing or thinning etc.., These techniques can be applied on the whole image or on a single pattern. They may therefore be performed before and or after segmentation. Several preprocessing techniques are explained by Gonzalez & Woods [Ref.2].

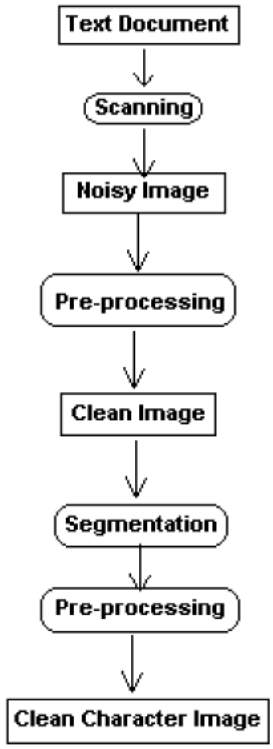


Fig.2 - A Priori Optical Font Recognition

(iii) Segmentation

Segmentation allows the extraction and location of each character in the image. Several segmentation algorithms are explained by Parker[1997] [Ref.3]. Segmentation is a difficult process. For e.g. touching and broken characters will increase the error rate significantly.

(iv) Omni-Char OFR

Using the font model base (obtained by learning process from known fonts) the omni-char OFR will discriminate text in different fonts and renders them to the corresponding mono-font OCR. Fig.3 shows the font probability estimation using Omni-Char OFR. The system returns a list of $<f_i, P(f_i)>$ where

f_i, represent a font identifier

$P(f_i)$ represent conditional probability that the text was printed with f_i.

f_i, for which $P(f_i)$ is maximum is the matching font.

(v) Mono-Font OCR

Character recognition is performed by a monofont OCR using a base of font dictionaries. Fig.4 shows the block diagram of mono-font OCR module. Each dictionary includes character models of a given font. The system returns a list of $<c_i, P(c_i)>$ where

c_i, represent a character class and

$P(c_i)$ indicates the probability that the pattern corresponds to c_i.

c_i, for which $P(c_i)$ is maximum is the matching character.

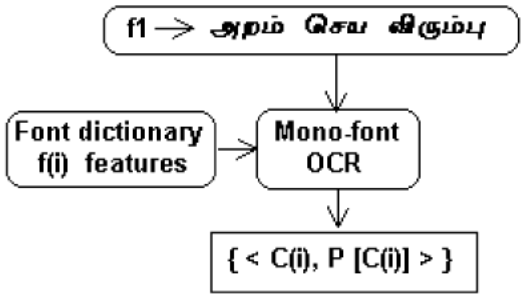


Fig.4 : Mono-font OCR System

(vi) Post Processing

It is used to improve the character recognition especially to correct spelling based on language grammar, dictionaries, n-gram techniques etc..,

(vii) Recognised text

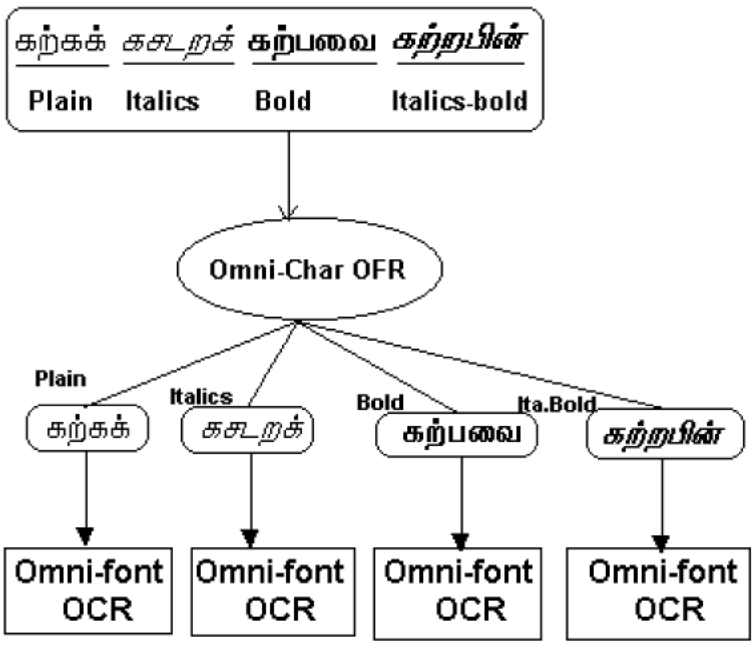


Fig.3 : Font Probability Estimation Using Omni-Char OFR

The recognised text can be stored in suitable encoding format like TAB (Tamil Bilingual Encoding Standard) or TAM (Tamil Monolingual Encoding Standard).

Conclusion

If an OCR to be used practically then it recognition rate must be high enough so that manual typing can be substituted by OCR. This can be achieved only if the recognition rate is greater than or equal to 99.9%. Using omnifont OCR, it is not possible to attain this recognition rate. At the same time monofont OCR can give the desired recognition rate if the font is known already. Omnichar OFR system is able to discriminate various fonts present in the document image. By combining Omni-Char OFR with OCR system, we can build a high precision OCR system for Printed Tamil Characters. Eventhough the recognition rate can be improved by using OFR, it still depends on various factors such as noise level, skew factor, resolution of the scanned image etc.., Discussion of these problems are beyond the scope of the current topic.

References

[1] 'Optical Font Recognition using Typographical Features' by Abdelwahab Zramdini & Rolf Ingold, IEEE transactions on Pattern Analysis & Machine Intelligence, Vol.20, No.8, Aug.1998.

[2] 'Digital Image Processing' by Rafael Gonzalez & Richard E Woods, Addison Wesley ISE Reprint, 1998.

[3] 'Algorithms for Image Processing & Computer Vision' by J R Parker, John Wiley & Sons Inc., 1997.

அச்சிட்ட தமிழ் எழுத்துக்களை அடையாளம் காணல்

சு.சீனிவாசன்,
கணிப்பொறிக் கோட்டம், இந்திராகாந்தி அணுவாராய்ச்சி மையம்,
கல்பாக்கம்-603102, காஞ்சிபுரம் மாவட்டம், தமிழ்நாடு

இராம.சுந்தரம்,
முன்னாள் துறைத் தலைவர்,
அறிவியல் தமிழ் மற்றும் தமிழ் வளர்ச்சித்துறை,
தமிழ்ப் பல்கலைக்கழகம், தஞ்சாவூர்-613 005, தமிழ்நாடு

முன்னுரை

கணிப்பொறித் துறையில் நாள்தோறும் ஏற்பட்டுவரும் வளர்ச்சி காரணமாக, இன்று கணிப்பொறிகளுக்குக் கட்புலன், செவிப்புலன், பேச்சுத் திறன் ஆகியவற்றை ஊட்டும் முயற்சிகள் உலகின் பல மூலைகளில் நடைபெற்று வருகின்றன. இவற்றில் கணிசமான முன்னேற்றமும் ஏற்பட்டுள்ளது. இப்பணிகள் அனைத்திற்கும் அடிப்படையில் தேவைப்படுவது தகவலை இலக்கமாக்கும் (digitising) வழிமுறையாகும். உரையிலிருந்து பேச்சு உருவாக்கம் (text-to-speech), கையெழுத்தை அடையாளம் காணல், பேச்சை அறிதல், பேசுவோரை இனம்காணல் ஆகியவற்றுக்கு இன்று மென்பொருள்கள் உருவாக்கப்பட்டிருக்கின்றன. இப்பணிகள் அனைத்தும் மொழி சார்ந்தவை என்பதால், மொழியின் பண்பை அறிவது அவசியமாகிறது. அவற்றுள் ஒன்று தமிழில் அச்சிட்ட எழுத்துக்களை அடையாளம் காணுதலாகும். இதற்கு எங்ஙனம் ஒரு வழிமுறையை உருவாக்கலாம் என்பது இங்கு விளக்கப்பட்டுள்ளது.

தமிழ் வரிவடிவு

தமிழ் எழுத்துக்களின் வரைவியல் குறித்த வருணனை(graphical description) தமிழ் இலக்கண நூல்களில் விரிவாகக் கூறப்படவில்லை. இரண்டாயிரம் ஆண்டுப் பழமை வாய்ந்த தமிழ்மொழியின் வரிவடிவு(script) காலந்தோறும் மாறி வந்துள்ளது. வடமொழியின் தாக்கத்தால் சில எழுத்துக்கள் தமிழ் வழக்கில் இணைத்துக் கொள்ளப்பட்டன. இக் கூடுதல் வடிவுகளும் தமிழர்களாலேயே உருவாக்கப்பட்டன என்பது கவனத்தில் கொள்ளத்தக்கது. இன்றைய தமிழ் ஏடுகளில் புழங்கும் அனைத்து எழுத்துக்களும் படம் 1-இல் காட்டப்பட்டுள்ளன. இவற்றிலுள்ள 313 எழுத்துக்களில் அடையாளம் காணவேண்டிய எழுத்துருக்கள் (characters) 147 மட்டுமே. இவை படம் 1-இல் தடித்த வடிவுகளாக (bold fonts) காட்டப்பட்டுள்ளன.

கருவிக்குக் கட்புலன் ஊட்டல்

அரசுப் பள்ளிகளில் படிக்கும் குழந்தைகள், இரண்டாம் வகுப்பிலேயே அனைத்துத் தமிழ் வரிவடிவுகளையும் எழுதக் கற்றுக்கொண்டு விடுகின்றனர். இது தொடர்ந்த பயிற்சியின் காரணமாகவே சாத்தியமாகிறது. அவர்கள், எழுத்துக்களை விரைந்து அடையாளம் கண்டு வாசிக்கவும் பயிற்சி பெற்றுவிடுகிறார்கள். இந்தக் கற்றல் நிகழ்வின்(learning phase) பின்னணியில் இன்றைய தமிழ் எழுத்து வடிவின் பல்வேறு கூறுகள், நினைவில் நிறுத்தப்படுகின்றன.

தமிழ் வரிவடிவின் பண்புகளை உணர்வு நிலையில் கருவிக்கு ஊட்டுவது எளிய பணியன்று. இச் செயலுக்குச் 'செயற்கை நரம்பணு வலையமைப்பு' (Artificial Neural Network) என்னும் வழிமுறை பின்பற்றப்படுகிறது. சிக்கல் வாய்ந்த 'கையெழுத்தை அடையாளம் காணும் பணி' முதலியவற்றுக்கு இந்த வழிமுறை உகந்தது. இவ்வியலில் அதற்கு மாற்றாக, வரைவியல் அடிப்படையில் தமிழில் அச்சிட்ட எழுத்துக்களை எவ்வாறு பகுத்து அடையாளம் காணலாம் என்பதற்குச் சில வழிமுறைகள் கொடுக்கப்பட்டுள்ளன. இப் பணிக்கு 'ஒளிசார் எழுத்துரு காணல் முறை' (Optical Character Recognition method) செயல்வடிவம் கொடுக்கவல்லது.

எழுத்துருக்களின் வரைவுப் பண்புகள் கருவிச் செயலாக்கத்திற்கு மிகவும் ஏற்றவை. இவற்றின் அடிப்படையில் அடையாளம் காண வேண்டிய அச்சு எழுத்துருக்களை நான்கு வகையாகப் பிரிக்கலாம். அவை வருமாறு:

1. கிடையாகச் செல்லும் எழுத்துக்கள்
2. மேலே நீளும் எழுத்துக்கள்
3. கீழே நீளும் எழுத்துக்கள்
4. கீழும்-மேலும் நீளும் எழுத்துக்கள்

தமிழர் எழுதும் முறை, ஏட்டில் இடமிருந்து வலமாகவும் மேலிருந்து கீழாகவும் செல்கிறது. அடையாளம் காணவேண்டிய எழுத்துருக்களை நுணுகிப் பார்க்கும்போது (zoom in) பல கூடுதல் தகவல்கள் கிடைக்கின்றன. (காண்க படம் 2.) தமிழ் வரிவடிவை ஒரு பறவை (bird) போல உருவகப்படுத்தும்போது அதன் சிறகு விரிப்பு (span) பற்றிய தகவல்கள் கிடைக்கின்றன; அதே வரிவடிவை ஒரு மண்புழு (earthworm) போல உருவகப்படுத்தும்போது, அதன் உடல் நீளத்தை-அதாவது எழுத்துருக்களின் பொருண்மையை (character mass) மதிப்பிட முடிகிறது.

எழுத்துருக்கள் கிடையாக எடுத்துக்கொள்ளும் இடஅளவு எழுத்துரு அகலம்(character width) எனப்படுகிறது. இவற்றின் மதிப்பு மற்றும் நிகழ்தகவு அட்டவணை 1-இல் காட்டப்பட்டுள்ளன. தமிழ் எழுத்துருக்களின் அகல மதிப்புகள் ஒரு கூட்டல் தொடரில் (arithmetic progression) அமைவதை இந்த அட்டவணையில் காணலாம். முதல் உறுப்பு (ா) 1.613 மி.மீ. அகலமும் மற்ற உறுப்புகள் 0.0812 மி.மீ. வேறுபாட்டில் (common difference) கூடுவதையும் அறிய முடிகிறது. ஐம்பது உறுப்புகளைக்கொண்ட இக் கூட்டல் தொடரில் பத்து உறுப்புகள் மட்டுமே வெறுமையாக(void) இருக்கின்றன. தமிழ் எழுத்துருக்களின் அகலம் ஒரு குறிப்பிட்ட இடைவெளியில் வேறுபடும் பண்பு, அவற்றை வகைப்படுத்தி அடையாளம் காண உதவுகிறது.

படம் 2-இல் அடையாளம் காண வேண்டிய அச்சு எழுத்துருக்கள் அனைத்தும் கணிப்பொறியின் துணைகொண்டு பெருக்கிக் காட்டப்பட்டுள்ளன. இப்படத்தை நுணுகி ஆராயும்போது சிறகு விரிப்பு 3:4:3 என்ற விகிதத்தில் அமைவதைக் காணலாம். அதாவது உடல்பகுதி 4 அலகுகளாகவும், மேலும் கீழும் நீண்ட பகுதிகள் ஒவ்வொன்றும் 3 அலகுகளாகவும் அமைவதைக் காணமுடிகிறது. மேலும், இங்கு எழுத்துக்கள் ஓடுகள் (tiles) பாவிய செவ்வகத் தளத்தில் வரையப்பட்டுள்ளது போலக் காட்சி அளிக்கின்றன. இத் தன்மையை அடிப்படையாகக் கொண்டு, எத்துணை ஓடுகள் வழியே எழுத்துரு செல்கிறது என மதிப்பிட முடிகிறது. இம் மதிப்பை எழுத்துரு பொருண்மை எனக் கொள்ளலாம். மெய் எழுத்தைக் குறிப்பதற்கு உதவும், 'புள்ளி'யை ஓர் அலகாகக் கொண்டு எழுத்துக்களின் பொருண்மை வரையறுக்கப்படுகிறது.

அடுத்து, வளைகோட்டுத் தன்மையுடைய தமிழ் எழுத்துருக்கள் X-அச்சிலும் Y-அச்சிலும் எத்துணை வெட்டுத்துண்டுகளை (intercepts) ஏற்படுத்துகின்றன என மதிப்பிடப்படுகிறது. இவ் வெட்டுத்துண்டுகளை கீழிலிருந்து மேல்நோக்கி X-அச்சில் வருடிச் சென்று (scanning) மதிப்பிட முடிகிறது. அப்போது ஓர் எழுத்துரு கிடையாக (horizontal) ஏற்படுத்தும் வெட்டுத்துண்டுகளின்

எண்ணிக்கையைத் தொடக்கம், உச்சஅளவு, முடிவு ஆகிய நிலைகளில் மதிப்பிட முடிகிறது. இவ்வாறே Y-அச்சிலும் வெட்டுத் துண்டுகளின் எண்ணிக்கையை மதிப்பிட எழுத்துருவின் இடமிருந்து வலம் நோக்கி வருடிச் சென்று காணமுடிகிறது. அப்போதும், எழுத்துரு நெடுக்கில் (vertical) ஏற்படுத்தும் வெட்டுத் துண்டுகளின் எண்ணிக்கையைத் தொடக்கம், உச்சஅளவு, முடிவு ஆகிய நிலைகளில் மதிப்பிட முடிகிறது. இவ்வருடல் முயற்சியின் பயனாக X,Y அச்சுகளில் வெட்டுத் துண்டுகளின் மதிப்புகள் இணைகளாகக்(pairs) கிடைக்கின்றன. (காண்க படம்-3.)

இவ்வாறு எழுத்துருவை, அதன் அகலம், ஏற்படுத்தும் வெட்டுத்துண்டுகளின் எண்ணிக்கை, பொருண்மை ஆகியவற்றின் அடிப்படையில் அடையாளம் காணச் செய்யும் SWIM (Script Width Intercept Mass) உத்தி (method) கருவிக்கு எளிதாகவும் ஏற்புடையதாகவும் தோன்றுகிறது. இவ்வடிப்படையில் கிடைத்த தகவல்கள் அட்டவணை 1 முதல் 5 வரை தொகுத்துத் தரப்பட்டுள்ளன.

படிநிலைகளில் எழுத்துருவை அடையாளம் காணல்

கருவியின் துணைகொண்டு எழுத்துருக்களை அடையாளம் காணும் பணி இரண்டு கட்டங்களில் (stages) நடைபெறுகிறது. முதல் கட்டத்தில் எழுத்துருக்கள்- கிடையாகச் செல்வன, மேல் நீள்வன, கீழ் நீள்வன, கீழும் மேலும் நீள்வன என 4 இனங்களாக (classes) வகைப்படுத்தப்படுகின்றன. உரைப்பகுதியில் இவற்றின் புழக்கம் முறையே 38.0, 31.3, 21.3, 9.4 விழுக்காடாக இருக்கின்றது. புள்ளி பெறும் எழுத்துக்கள் மேல் நீள்வன வகையாகக் கொள்ளப்பட்டுள்ளன. சிறகுகள் குவித்த எழுத்துருக்களை எழுதும் முயற்சியும் எளிதாக இருக்கிறது. இரண்டாவது கட்டத்தில் எழுத்துரு X-அச்சிலும் Y-அச்சிலும் ஏற்படுத்தும் வெட்டுத்துண்டுகளின் உச்ச எண்ணிக்கை, தொடக்க-முடிவு எண்ணிக்கை, எழுத்துரு அகலம், பொருண்மை ஆகிய நான்கு படிநிலைகளில் (steps) ஆராயப்படுகின்றன.

ஓர் எழுத்துருவின் அடையாளத்தை எல்லாப் படிநிலைகளின் வாயிலாகவும் உறுதிப்படுத்துவது அவசியமன்று. காட்டாகச் சில எழுத்துருக்கள் படிநிலை-1 அளவில்(அதாவது வெட்டுத்துண்டுகளின் உச்ச எண்ணிக்கை அடிப்படையில்) அடையாளம் கண்டு கொள்ளப்படுகிறது. இங்ஙனம் 31 எழுத்துருக்களை அடையாளம் காண முடிகிறது. அவையாவன:

ட ூ ச ய ம உ ஈ ண ஊ (காண்க அட்டவணை -2)
ட் ச் கீ ஃ வீ ஸீ ணி (காண்க அட்டவணை -3)
பு ர யு டி த மு ஷ ஞ கூ ஷ ணூ (காண்க அட்டவணை -4)
ற் ஜீ ஷ் ஞ் கூ ஷ் (காண்க அட்டவணை -5)

உரைப்பகுதியில் 20 விழுக்காடு இவ் எழுத்துருக்களைக்கொண்டு அமைகின்றன.

படிநிலை-2 அளவில்(அதாவது எழுத்துருவின் தொடக்கம் மற்றும் இறுதியிலுள்ள வெட்டுத்துண்டுகளின் எண்ணிக்கை அடிப்படையில்) மேலும் 49 எழுத்துருக்களை அடையாளம் காண முடிகிறது. அவையாவன:

ப ா டூ எ ங ல வ கூ ை ஸ (காண்க. அட்டவணை -2)
ப் ர் ய் ரி ம் ரீ கி (காண்க. அட்டவணை -3)
ந ற யூ ஏ று மு அ ஜ து ங ஐ ழு லு ஞ லூ ஞூ னு னூ ணு (காண்க. அட்டவணை -4)
ழ் நீ ழீ டு றி நி டீ ஜ் இ ஜி கூீ ஹீ (காண்க. அட்டவணை -5)

உரைப்பகுதியில் 41 விழுக்காடு இவ் எழுத்துருக்களைக்கொண்டு அமைகின்றன.

படிநிலை-3 அளவில்(அதாவது எழுத்துரு அகலத்தின் அடிப்படையில்) 57 எழுத்துருக்களை அடையாளம் காண முடிகிறது. அவையாவன:

க சு ள ன (காண்க. அட்டவணை -2)
சீ பீ ஶ மே மீ க் சி பெ பி மி ங் யீ ஙீ ல் லீ வ் ள் ளீ னீ ஙி லி ஸ் வி ளி னி ஸி (காண்க. அட்டவணை -3)
ழ பூ ஙு ரு நு கு ஆ வு ரூ றூ மூ சூ நூ தூ வூ ஹ ளு ளூ (காண்க. அட்டவணை -4)
ந் றீ தி ழி ஸ்ரீ ஷீ ஸ்ரி ஹி க்ஷி (காண்க. அட்டவணை -5)

உரைப்பகுதியில் 32 விழுக்காடு இவ் எழுத்துருக்களைக்கொண்டு அமைகின்றன.

படிநிலை-4 அளவில்(அதாவது எழுத்துருவின் பொருண்மை அடிப்படையில்) இதுவரை அடையாளம் காணப்படாத 10 எழுத்துருக்களை வேறுபடுத்திக் காண முடிகிறது. அவையாவன:

ன் யி; ண் ணீ (காண்க. அட்டவணை -3)
ஒ ஓ (காண்க. அட்டவணை -4)
த் தீ; ஹ் ஷி (காண்க. அட்டவணை -5)

இவை உரைப்பகுதியில் 7 விழுக்காடு பயன்படுகின்றன.

வருடல் சோதனை

அச்சிட்ட தாளை மேலிருந்து கீழ்நோக்கி வருடும்போது கருவிக்குப் புலப்படும் வெண்மைப் பகுதி வரிகளைப் பிரித்து(line separation) அடையாளம் காண உதவுகிறது. இதன் மூலம் தாளில் இடம்பெற்றுள்ள வரிகளின் எண்ணிக்கையை அறியலாம். இதைப் போல தாளை இடமிருந்து வலம்நோக்கி வருடும்போது புலப்படும் வெண்மைப் பகுதி எழுத்துருக்களைப் பிரித்து(character separation) அடையாளம் காண உதவுகிறது. இதன் மூலம் அச்சிட்ட பகுதியிலுள்ள ஒவ்வொரு எழுத்துருவின் அகலத்தை நேரடியாக அளவிட முடிகிறது. அட்டவணை-2,3.4,5 அகியவற்றைக் கொண்டு படிநிலை-2 அளவில் 80 எழுத்துருக்களை(அடையாளம் காண வேண்டிய மொத்த எழுத்துருக்கள்-147) ஐயத்திற்கு இடமின்றி அடையாளம் காண முடிகிறது.

படிநிலை-2 அளவில் எழுத்துருக்களின் உயரம், சாய்வு, தடிப்பு ஆகிய பண்புகள் குறுக்கிடுவதில்லை. அடுத்து, கருவிகொண்டு மதிப்பிட்ட எழுத்துரு அகலத்திற்கும் அட்டவணையிலுள்ள எழுத்துரு அகலத்திற்கும் நேர் விகித தொடர்பு காணப்படுகிறது. ஒரு குறிப்பிட்ட உயரமுள்ள எழுத்து வடிவுக்கு(font size) இவ் விகிதம்(ratio) ஒரு மாறிலியாகும்(constant). இதைப் போல, கருவிகொண்டு மதிப்பிட்ட எழுத்துரு பொருண்மைக்கும் அட்டவணையிலுள்ள எழுத்துரு பொருண்மைக்கும் நேர் விகித தொடர்பு காணப்படுகிறது. இவ் விகிதமும் ஒரு மாறிலியாகும்.

படிநிலை-3 அளவில் எழுத்துருக்களை அடையாளம் காண எழுத்துரு அகல ஒப்பீடு(விகித மதிப்பு) உதவுகிறது. இறுதி க் கட்டமாக, படிநிலை-4 அளவில் எழுத்துருக்களை ஐயத்திற்கு இடமின்றி அடையாளம் காண எழுத்துரு பொருண்மை ஒப்பீடு(கிடைக்கும் மற்றோர் விகித மதிப்பு) உதவுகிறது. எழுத்துருக்களுக்கு சாய்வு கொடுக்கும்போதும் தடிப்பு கொடுக்கும்போதும் இந்த ஒப்பீடு ஒரு வரம்புக்குள்(range) மாறுபடுகின்றது.

புழக்கம் மிகுந்த எழுத்துருக்களை அறிதல்

அச்சில் புழங்கும் தமிழ் எழுத்துருக்களின் நிகழ்வை(occurrence) கணிக்கப் பின்வரும் சோதனை மேற்கொள்ளப்பட்டது. இணையத்தின் வாயிலாக ஜூலை 1997 முதல் ஜூன் 1998 வரையுள்ள ஆனந்தவிகடன் வார இதழில் வெளியான சிறுகதை, சுயசரிதை, கட்டுரை, கவிதை, புதினம், தலையங்கம் ஆகிய பகுதிகள் சேமிக்கப்பட்டு எழுத்துப் புழக்க மதிப்பீடு கணிக்கப்பட்டது. இத்தொகுதியில் ஏறக்குறைய எட்டு இலட்சம் எழுத்துருக்கள்(characters) இடம் பெற்றிருந்தன. இதிலிருந்து எழுத்துருக்களின் புழக்கமும்(frequency) நிகழ்தகவும்(probability) கணிக்கப்பட்டன. இவ்வாறு கணித்த மதிப்புகள் நான்கு அட்டவணைகளிலும் எழுத்துருவை அடுத்துக் கொடுக்கப்பட்டுள்ளன. இவற்றினின்று சில சுவையான தகவல்களைப் பெற முடிகிறது.

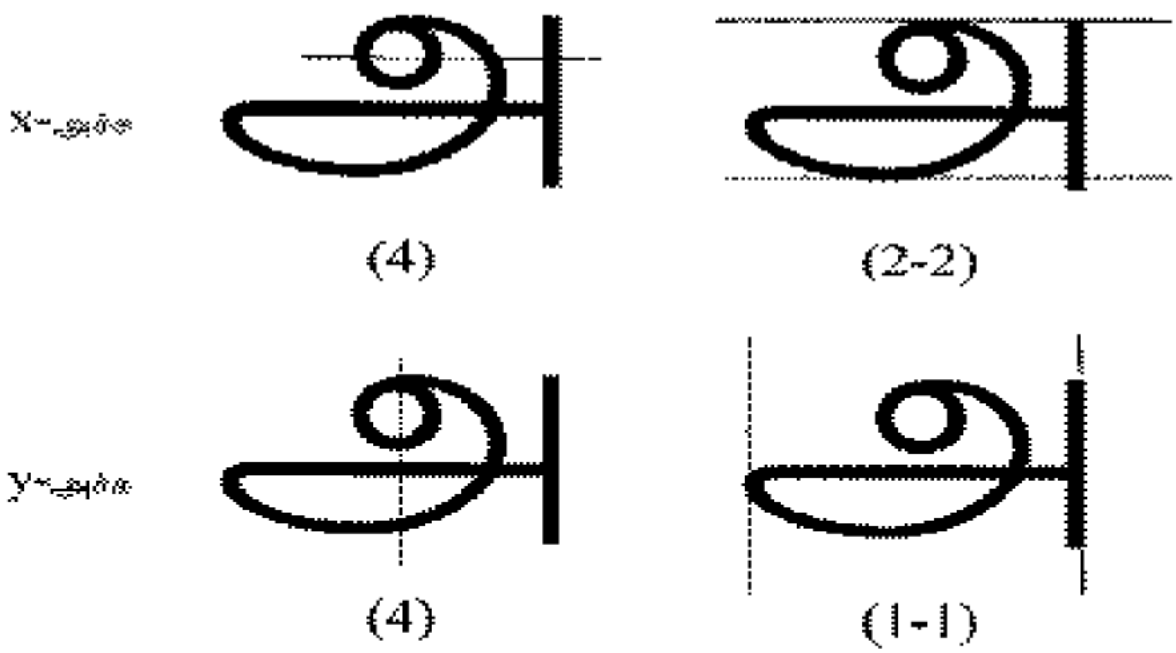
நிகழ்தகவு மதிப்பு, ஒரு விழுக்காட்டுக்கும் மிகுந்த எழுத்துருக்களின் எண்ணிக்கை 37 மட்டுமே. இவை பயன்பாட்டு அடிப்படையில் கீழே இறங்குவரிசையில் தரப்பட்டுள்ளன.

ா	(நிகழ்தகவு > 8)
க, த	(நிகழ்தகவு > 4)
ை, ப, ன், ே, வ	(நிகழ்தகவு > 3)
ம், க், து, ல், ெ, த், ம	(நிகழ்தகவு > 2)
ய, ட, ன, அ, ல, ர் , ரு, ர, ப், ச, ந், கு, ட், தி, டு, எ, இ, ற, ள், டி, வி, ண்	(நிகழ்தகவு > 1)

மேற்குறிப்பிட்ட 37 எழுத்துருக்களைக் கொண்டு அச்சுப்பகுதியின் 82 விழுக்காடு எழுத்துருக்கள் அமைகின்றன என்பது இங்குக் குறிப்பிடத்தக்கது. புதிதாக மொழி கற்போருக்கு இவ்வெழுத்துருக்களில் பயிற்சி அளிப்பது பயனளிக்கக் கூடியதாகும்.

முடிவுரை

தமிழ் வரிவடிவைப் பிள்ளைப் பருவத்தில் உணர்வு அடிப்படையில் எழுதப் பயிற்சி மேற்கொள்ளும் போக்கு ஒருபுறம் நிகழ, மறுபுறம் வரைவியல் அடிப்படையில் வரிவடிவை ஆராயும் போக்கும் தொடர்கிறது. உணர்வு அடிப்படையில் கற்றல் முறை, நினைவில் வைத்துக்கொள்ள எளிதாகிறது; வரைவியல் முறை, கருவிக்குக் கட்புலன் ஊட்டுவதற்குத் தோதாகிறது. இலக்க நூலகமாக்கும்(digital library) பணிக்குத் தமிழ் ஆவணங்களைக் கணிப்பொறிக் கோப்புகளாக மாற்ற வேண்டி இருக்கிறது. இதற்கு இங்கு விரித்துரைத்த 'தமிழில் அச்சிட்ட எழுத்துக்களை அடையாளம் காணும் முறை' பயனளிக்கக் கூடியது. இப்பணிக்கு விரைந்து கருவி அமைப்பது காலத்தின் கட்டாயமாகும்.



படம் 3. வேர்த்தொகுதிகளின் எண்ணிக்கை: x அச்சு மற்றும் தொடக்கம்-முடிவு

அட்டவணை-1. கிளைமாகச் சேரும் எழுத்துக்களை அடையாளம் காணல்

ஈறு	பகுதிவகை-1	பகுதிவகை-2		பகுதிவகை-3		அடையாள எழுத்து	எழுத்துகளில் நிகழ்தகவு
வெட்டுக்குறியீட்டு எண்ணிக்கை (உயரம்)		எழுத்து நிலை		வெட்டுக்குறியீட்டு எண்ணிக்கை இடைவெளி-முடிவு			(P_i)
மேலுள்ள	கீழுள்ள	மேலுள்ள	கீழுள்ள	மேலுள்ள	கீழுள்ள		(%)
1	1	20	1.00	1-1	1-1	a	1.975
2	1	21	1.05	2-1	1-1	ஆ	8.291
2	1	24	1.20	1-2	2-1	இ	3.292
2	2	24	1.20	1-1	1-1	ஈ	0.053
2	3	28	1.40	1-2	1-2	உ	1.975
3	1	32	1.60	2-3	1-1	ஊ	0.685
3	2	32	1.60	3-2	1-1	எ	2.078
3	3	33	1.65	2-1	1-1	ஏ	4.993
3	3	36	1.75	2-1	1-1	ஐ	0.246
3	4	36	1.80	1-1	2-1	ஒ	0.659
4	2	27	1.35	2-1	1-2	ஓ	0.013
4	3	30	1.50	2-1	1-1	ஔ*	1.253
4	3	36	1.80	1-1	1-1	ஃ*	0.385
4	3	42	2.10	2-2	1-1	க	0.092
5	3	42	2.10	2-1	1-1	ங	1.752
5	3	44	2.20	3-2	1-1	ச	6.196
5	3	47	2.35	2-2	1-1	ஞ	3.124
5	3	50	2.50	3-2	1-1	ட	1.903
5	3	52	2.60	3-1	1-1	ண	0.256
6	3	48	2.40	3-2	1-1	த*	3.332
6	3	48	2.40	3-2	1-1	ந*	0.042
7	3	58	2.90	4-3	1-1	ன	0.408
7	4	78	3.90	1-3	2-2	ப	0.054

$\sum P_i = 99.993$

அட்டவணை-2. கோடை நீரூற்று மழைக்காலச் சூழல் மாசுபாடு மதிப்பீடு

நிலை-1 கொடுக்கப்பட்ட எண்ணிக்கை (உதாரண)		நிலை-2 எடுத்து நோக்கு		நிலை-3 கொடுக்கப்பட்ட எண்ணிக்கை தொடக்கம்முதல்		தவறான வழியை எடுத்து	எடுத்துக்காட்டின் நிகழ்தகவு : P_i :
x-முதல்	y-முதல்	பாலின்	தூய்மை	x-முதல்	y-முதல்		(%)
1	2	21	1.05	1-1	1-2	C	1/35
2	2	23	1.15	2-1	1-1	C	1/20
2	2	25	1.25	1-1	1-1	C	1/20
2	4	30	1.50	1-1	1-2	A	0.501
3	2	33	1.65	2-1	1-1	D	0.499
3	2	43	2.15	3-1	1-1	B	0.653
3	3	34	1.70	1-1	1-1	D	2.375
3	3	41	2.05	2-1	1-2	A	0.05
3	4	38	1.90	2-1	1-1	A	0.251
3	4	42	2.10	1-1	1-1	D	0.049
3	4	44	2.20	1-1	1-1	A	0.046
3	4	47	2.35	1-1	1-2	B	3.15
3	4	49	2.45	2-1	1-1	A	0.454
3	6	50	2.50	1-1	1-1	B*	0.550
3	4	51	2.55	1-1	1-2	B*	0.270
3	5	52	2.60	2-1	1-2	A	0.033
4	2	28	1.40	2-1	1-1	D	0.049
4	3	45	2.25	2-1	1-1	B	2.139
4	3	54	2.70	2-1	1-1	C	0.054
4	3	59	2.95	2-1	1-1	B	0.225
4	4	43	2.15	2-1	1-1	A	0.537
4	4	47	2.35	2-1	1-1	B	0.2234
4	4	60	3.00	3-1	1-1	B	0.910
4	4	62	3.10	3-1	1-1	A	0
5	3	44	2.20	2-1	1-1	D	2.190
5	3	46	2.30	3-1	1-1	A	1995
5	3	50	2.50	2-1	1-1	D	0.050
5	3	52	2.60	3-1	1-1	B	3.207
5	3	55	2.75	2-1	1-1	A	0.222
5	3	59	2.95	3-1	1-1	B	0.203
5	3	62	3.10	3-1	1-1	D	0.567
5	3	66	3.30	3-1	1-1	B	0.014
5	3	73	3.65	3-1	1-1	D	0
5	4	42	2.10	2-1	1-2	A	0.049
5	4	50	2.50	3-1	1-1	D	0.250
5	3	62	3.10	3-1	1-1	B	0.306
6	3	65	3.25	4-1	1-1	B	0.213
6	3	71	3.05	6-1	1-1	B	0.052
6	3	77	3.85	3-1	1-1	B	1.12
6	4	63	3.15	3-1	1-1	A	0.004
7	3	70	3.50	4-1	1-1	B	0.027
7	3	73	3.65	4-1	1-1	B	1.203
7	3	84	4.20	4-1	1-1	B	0.043
8	5	89	4.45	5-1	1-1	B	0.47
							$\sum P_i$ <0.05

அட்டவணை-3. கீழே தரும் எழுத்துக்களை அடையாளம் காண்பதற்கு...

6	5	30A	5.28	1-3	2-1	ஐ	0.001
8	6	115	5.05	1-4	2-1	ஔ	0.005
9	5	127	6.45	1-4	2-1	ஔ	0.001

$\Sigma P_i = 0.1206$

அட்டவணை-4. கீழுள்ளவை தரும் எழுத்துக்களை அடையாளம் காணல்

மாதிரி-1		மாதிரி-2		மாதிரி-3			
மேட்டிழுப்புவளைவின் எண்கோடம்		எழுத்து தரம்		மேட்டிழுப்புவளைவின் எண்கோடம்		அடையாளம் காணும் எழுத்து	எழுத்துக்கான நிகழ்தகவு
(ஆரம்)				தொடக்கம்/முடிவு			(P_i)
x-அச்சு	y-அச்சு	மதிப்பு	ஆரம்	x-அச்சு	y-அச்சு		(%)
3	3	39	1.95	1-1	1-1	அ	0.547
3	4	45	2.25	1-1	1-1	ஆ	1.005
3	4	52	2.60	1-1	1-1	இ	3.017
3	5	55	2.75	1-1	2-2	ஈ	0.745
3	5	42	2.75	1-1	2-3	உ	2.109
3	6	50	2.95	1-1	1-1	ஊ	0.152
3	6	62	3.10	1-1	2-3	எ	0.057
3	6	66	3.40	1-1	2-2	ஏ	0
4	4	56	2.80	1-1	2-3	ஐ*	0.516
4	3	59	2.95	1-1	1-1	ஒ*	1.275
4	4	56	2.80	1-1	1-2	ஓ	0.105
4	4	63	3.15	2-1	2-2	ஔ	0.287
4	4	65	3.25	1-1	1-2	க	0.230
4	4	70	3.45	1-1	2-1	ங	1.379
4	4	60	4.30	1-1	2-1	ச	0.124
4	6	78	3.90	2-1	2-2	ஞ	0.008
5	5	92	4.70	2-1	2-1	ட	0.025
5	5	85	4.45	2-1	1-1	ண	1.201
5	5	90	4.95	1-1	1-2	த	0.010
5	3	70	3.50	1-1	1-1	ந	0.054
5	4	70	3.50	1-1	1-2	ப	0.141
5	5	85	4.25	1-1	1-1	ம	0.0005
5	5	87	4.35	1-1	1-1	ய	0
7	3	90	4.50	1-1	1-1	ர	0.0001
7	4	87	4.35	1-1	1-1	ல	0.002
7	4	92	4.60	1-1	1-1	வ	0.015
7	4	87	4.65	1-1	1-1	ஷ	0
7	5	102	5.10	1-1	1-2	ஸ	0.002
7	6	106	5.30	1-1	1-1	ஹ	0.0001
8	4	110	5.50	1-1	1-1	ஜ	0.030
8	4	112	5.60	1-1	1-1	ஶ	0.0001

$\Sigma P_i = 9.403$

அட்டவணை-3. கீழே தரும் வழுத்துக்களை ஒலைபாயம் காண்க

ஈ படிநிலை-1 வெட்டுத்தளநிலைச் சார்பீட்டை		படிநிலை-2 வழுத்து நிரல்		படிநிலை-3 வெட்டுத்தளநிலைச் சார்பீட்டை தொடக்கம் முடிவு		ஒலைபாயை வழும் வழுத்து	வழுத்துவரிசை நிகழ்தகவு (P)
x-ஒன்று	y-ஒன்று	ஒன்று	ஒப்பு	x-ஒன்று	y-ஒன்று		(%)
2	2	29	7.45	1-2	1-1	பு	0.655
3	2	35	1.25	1-1	1-1	ர	1.515
3	3	37	1.85	1-3	1-1	ஒ	0.273
3	3	39	1.90	2-1	1-1	ஃ	1.515
3	3	37	1.85	1-2	1-1	ற	1.515
3	3	43	2.15	2-1	1-1	ஸ	0.842
3	3	48	2.40	1-2	2-1	ஜ	0.273
3	3	57	2.85	2-2	2-2	ஞ	0.079
3	3	55	3.25	1-3	2-1	ஹ	0.026
3	4	42	2.10	2-2	2-2	ஷ	4.161
4	2	31	3.05	2-3	1-1	ஶ	0.497
4	3	41	2.05	2-1	1-1	ஏ	0.021
4	3	46	2.30	2-2	1-1	ஈ	0
4	3	54	2.70	1-2	1-1	ஊ	1.020
4	3	51	3.05	1-3	1-1	ஃ	0.497
4	4	61	3.05	1-2	1-1	ஜ்	1.97
4	4	61	3.05	2-2	2-2	ஜ்	0.025
4	4	53	3.15	1-2	1-1	ஜ்	0.012
4	4	66	3.30	2-2	1-1	ஷ்	1.368
4	4	65	3.25	1-3	1-1	ஷ்	0.205
4	4	92	4.60	1-2	1-1	ஷ்	0.436
4	5	63	3.15	2-2	2-2	ஸ	0.044
4	5	67	3.35	2-2	1-1	ஹ	2.222
4	5	69	3.45	1-3	2-1	ஹ	0
5	3	51	2.55	2-2	1-1	ஷ	0.297
5	3	56	2.80	2-3	1-1	ஷ	0.024
5	3	60	3.00	1-3	1-1	ஷ	0.030
5	3	75	3.75	2-3	1-1	ஷ	0.0001
5	4	68	3.40	1-2	1-1	ஷ	0.049
5	4	71	3.55	1-3	1-1	ஷ	0
5	4	77	3.85	2-2	1-1	ஷ	0.022
5	5	49	2.45	1-1	1-1	ஷ	0.437
5	5	57	2.85	2-1	1-1	ஷ	0.099
5	5	74	3.70	1-3	2-1	ஷ	0.200
5	5	79	3.95	1-2	2-1	ஷ	0.003
5	5	81	4.05	1-2	2-1	ஷ	0.037
5	5	69	3.45	1-2	1-1	ஷ	0.123
5	6	71	3.55	1-2	1-2	ஷ	0.010
5	6	85	4.25	1-2	2-1	ஷ	0
6	6	88	4.40	1-3	2-1	ஷ	0.010
7	5	88	4.40	1-2	1-1	ஷ	0.001
7	6	91	4.55	1-3	1-1	ஷ	0.004
7	6	84	4.20	1-2	1-1	ஷ	0.107
7	6	92	4.60	1-2	1-1	ஷ	0.003
7	6	91	4.55	1-3	2-1	ஷ	0.311
7	6	92	4.60	1-2	2-1	ஷ	0