# A Complete Tamil Optical Character Recognition System

Aparna K G and A G Ramakrishnan
Biomedical Laboratory, Department of Electrical Engineering
Indian Institute of Science, Bangalore – 560 012
e-mails: {prjocr, ramkiag}@ee.iisc.ernet.in

## Abstract

The aim of the present work is to recognise printed Tamil text. Though commercial Optical Character Recognition (OCR) packages are available in the market for Roman Script, not much work has been done in the field of OCR for Indian languages. Indian scripts usually have a large number of symbols and hence, recognition is a challenging task. In the current context, a complete OCR in printed Tamil text has been developed. Attempt has been made to make it font and size independent. The methods involved can be extended to other Indian scripts as well.

## 1. 1 Introduction

Document Image processing and Optical Character Recognition (OCR) have been frontline research area in the field of human-machine interface for the last few decades. Recognition of machine printed or hand printed document is an essential part in applications like intelligent scanning machines, text to speech converters and automatic language to language translators. The objective of document image analysis is to recognise the text and graphics components in the paper document and to extract the intended information, as human beings do. Two components of document image analysis are Textual processing and Graphical processing. Textual processing deals with the text component of the document image. The graphical processing deals with non-textual line and symbol components that make up line diagrams, delimiting straight lines between text sections and company logos etc. In the current context, we limit ourselves to the textual processing part.

## 1.2 Block diagram of an OCR system

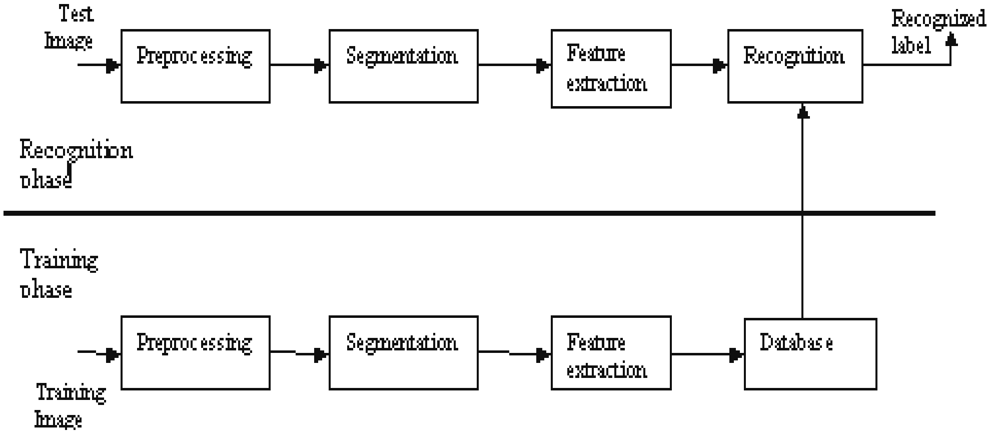


*Figure 1: Block diagram of an OCR System*

As shown in the block diagram in figure 1, an OCR system has two phases, namely the training phase and the recognition phase. Both the phases have some common steps.

The *skew correction* is the first operation carried out on the binary document image. Tilt is unavoidable when a paper is put on the scanner bed. This tilt angle is usually called the skew angle of the document. The image is to be rotated back after detecting the skew angle to make the text lines horizontal. Next, *document segmentation* is performed on the skew corrected image. In document segmentation, the text lines are segmented out first. For each text line separated out, the words in it are segmented. For each word, the individual symbols are separated and given as input to the recogniser. Putting back all the recognised symbols in position, the recognised word is encoded. In the same way, the words are put together to encode the text line. *Symbol normalisation* is the first step in symbol recognition. Each segmented symbol is brought to a normalised size and the skeleton of the symbol is found. It is the key step to make the recognition process independent of font and size. Symbol normalisation is also necessary to bring the individual symbols to a normalised size, so that they can be compared with the known symbols in the reference database. From each normalised symbol, relevant features are extracted. Features are a set of vectors carrying the shape information of the symbol. There are numerous *feature extraction* procedures reported in the literature. Based on the features extracted, the characters are classified into different classes. The *classifier* compares the feature vector of the unknown symbol, with that of the known symbols in the database, based on some distance measure, and classifies that symbol according to a predefined classification rule.

## 1.3 Characteristics of Tamil script

Tamil Script has 12 vowels and 23 consonants. Like all other Indian languages the basic structure of Tamil script is different from Roman Script. Phonetically, vowels can occur alone or can be accompanied by a consonant, or another vowel. Vowel and some frequently used compositions of vowels, which are not accompanied by a consonant, have separate symbolic representations. When accompanied by a consonant, symbolization is in the form of an addition of ‘matra’ to the accompanying consonant. The consonants cannot appear alone. A consonant has to be either preceded or followed or both preceded and followed by a vowel. The vowel that precedes a consonant is considered as a part the previous character in the word. When a vowel follows a consonant in a word, the ‘matra’ corresponding to the concerned vowel modifies the symbolization of the consonant. There is no matra for /*a*/, which is the first vowel in the alphabet set. If /*a*/ follows a consonant, the consonant appears in its original shape. If no vowel follows a consonant, it is symbolized by adding a dot (“Pulli”) on the top of the original consonant.

In many other Indian scripts like Devanagari or Bengali, a consonant following a stop consonant is sometimes written as a compound consonant, which is a single symbol. There is no such symbol in Tamil. However in Tamil a consonant when added with a matra may get modified completely, unlike other Indian languages. There is no headline or ‘*shirorekha’* in the case of Tamil. So the characters are separated from each other. The same vowel while modifying different consonants may modify it in different ways; i.e. the shape of the ‘matra’ for a given vowel may not be unique. This is the common case for the ‘matras’ /*u*/ and /*uu*/. This is a feature not common to most other Indian languages. The figure 2 shows the basic character set of Tamil language.

அ ஆ இ ஈ உ ஊ
எ ஏ ஐ ஒ ஓ ஔ

**Vowels in Tamil Script.**

க ச ட த ப ற ய
ர ல வ ந ங ஞ ண
ன ம ள ழ ஸ ஜ
ஷ ஹ க்ஷ

**Consonants in Tamil Script.**

*Figure 2: Basic Character set of Tamil Language*

## 1.4 Assumptions

Some of the assumptions made for efficient working of OCR is

- The document should contain only text with no images.
- Columnar division in a page is not considered.
- Only printed text document is considered.
- The OCR gives good recognition on documents, which are only in Tamil.

## 2. Preprocessing

## 2.1 Skew Detection

When a document is fed to an optical scanner manually, a few degrees of skew (tilt) is unavoidable. Skew angle is the angle that the text lines make with the horizontal direction. Skew estimation and correction [5] is one of the most important steps in text processing. Even in the case of combined text and graphics scenario, skew estimation is done from the text portion, because the text portion has a favourable structure for skew estimation.

We propose a precise skew detection method, which detects the skew of a document in two steps.

### 2.11 Coarse skew estimation

The steps involved in coarse skew estimation are as follows.

- Run-length Smoothing
- “Interim Line detection” and making the “Interim Line Image”.
- Application of Hough Transform on “Interim Line Image" to estimate the coarse skew angle.

Run-length Smoothing is essentially filling up the gaps within characters and inside the characters, so that each word becomes a connected component. All background run-lengths having length less than a threshold are converted to foreground run-lengths to perform run-length smoothing.

The next step is to find the Interim Lines. Interim Line is the line mediating the background gap between two text lines. The orientation of the interim lines with respect to the positive direction of the x-axis is equal to the skew angle of the image.

Next, the rough estimate of the skew angle is obtained as the orientation angle of the interim lines with respect to the positive direction of the x-axis. Hough Transform can be applied here.

The form of the equation of a straight line used is:

$$y \cos \theta - x \sin \theta = P,$$

where P is the perpendicular distance of the line from the origin and θ is the angle between the line and the positive direction of the x axis.

From the results obtained this method enables us to find out the skew angle within the range of ± 0.25° about the actual value.

### 2.12 Fine skew detection

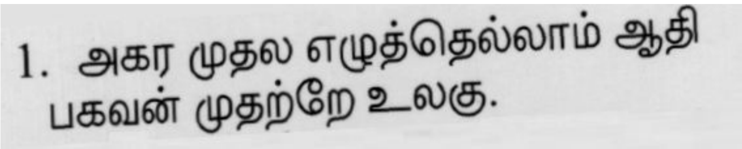
Fine Skew detection involves the following steps:

- Superposition of the segmented line images to create a scatter image.
- Finding out the principal axes of the scatter image.

Using the knowledge of the skew angle from the coarse estimate, text lines are segmented. Text line segmentation is performed by finding the projection profile of the original image at an angle equal to the coarse estimate of the skew angle. All the text lines are segmented this way and superposed on one another in such a way that their centres are concurrent. The image so formed is called the scatter image. The direction of the principal axis of the resulting scatter image is taken as the fine skew direction. The results obtained shows that the accuracy of the final estimate is +/- $0.06^0$.

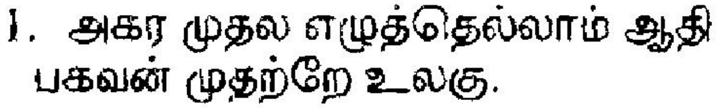
## 2.2 Skew Correction

Once the skew error is detected, the image needs to be deskewed. This process of skew correction is achieved by rotating the image by an angle equal to that of skew, but in the opposite direction. The process of skew correction is usually performed on the binarised image. The rotation algorithm used employs bilinear interpolation. But due to quantisation effects, the resultant skew corrected image has a lot of distortions. Hence an attempt has been made here to reduce the quantisation effects by performing skew correction on the gray scale image rather than the binary image. The outputs are shown in figure 3 below.



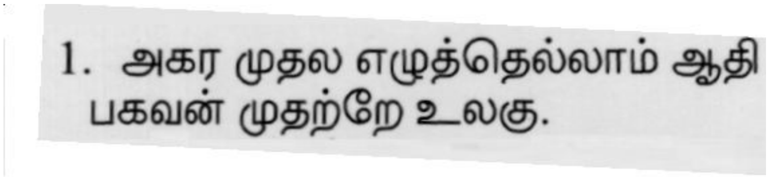
1. அகர முதல எழுத்தெல்லாம் ஆதி
பகவன் முதற்றே உலகு.

**Figure 3a: Original skewed image**



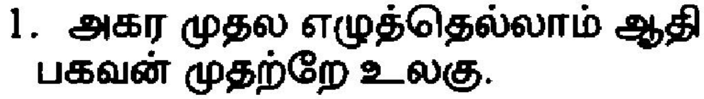
1. அகர முதல எழுத்தெல்லாம் ஆதி
பகவன் முதற்றே உலகு.

**Figure 3b: Skew correction performed on binary image**



1. அகர முதல எழுத்தெல்லாம் ஆதி
பகவன் முதற்றே உலகு.

**Figure 3c: Skew corrected gray scale image**



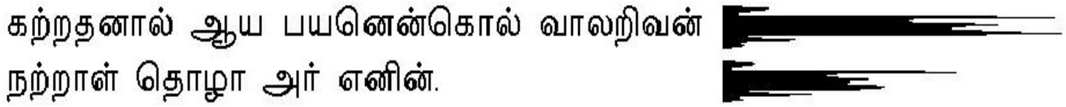
1. அகர முதல எழுத்தெல்லாம் ஆதி
பகவன் முதற்றே உலகு.

**Figure 3d: Skew corrected binarised image**

*Figure 3: Outputs of Skew correction*

Segmentation is the process of extracting objects of interest from an image. In document analysis, the first task towards text recognition is to segment the textual regions from graphics, maps and other figures. The first step in segmentation is detecting lines. Then detect the word in the line and finally detect the individual character in that word.

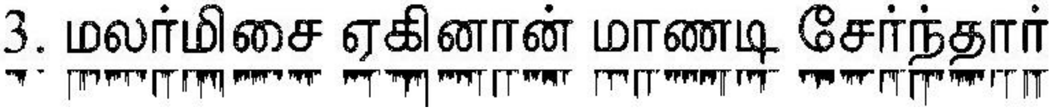
Text lines can be identified with the help of the horizontal projection profile. Projection profile of a document in a particular direction is the running sum of the pixels along that direction. The profile exhibits valley points at line boundaries and the location of these minima points mark the line boundaries. For binary images, these are the points where the profile goes to zero. Figure 4 below shows the horizontal projection profile of a binarised image.



கற்றதனால் ஆய பயனென்கொல் வாலறிவன்
நற்றாள் தொழா அர் எனின்.

*Figure 4: Text lines with corresponding horizontal projection profiles*

Word segmentation can be performed with the aid of vertical projection profile. The vertical projection profile shows valleys at points corresponding to word gaps. These word boundaries can be identified with the help of these minima points. Figure 5 below shows the vertical projection profile of a binarised image.



3. மலர்மிசை ஏகினான் மாணடி சேர்ந்தார்

*Figure 5: Text lines with corresponding vertical projection profiles*

Character segmentation is the process of segmenting individual symbols from the words. These individual components in general have a vertical gap between them. Tamil, on account of its basic structure, has modifiers that are either attached at the top and bottom or distinct from the basic alphabet. In such a scenario, a simple segmentation scheme can be used. However, in the absence of vertical run of zeros in between the characters for italicised fonts, a profile-based approach cannot be undertaken. Hence, for segmenting individual symbols, a connected component approach is used. In this technique, all the foreground pixels are checked for connectivity with the neighbours and appropriately labelled. Pixels having at least one already labelled non-zero neighbour are given the label of the neighbour and those that haven’t are assigned a new label. The final segmentation is performed by isolating those pixels that have the same label values. This method works perfectly with any font.

## 4. Symbol Pre-processing

Character preprocessing is an essential part of any OCR design. No other document processing step is as vital as character preprocessing. The characters segmented from the original text image are of different sizes. The characters belonging to the same cluster need to be brought to a normalised size, in order to compare them in a particular feature space. Thus, the scaling of the characters to a normalised size is an essential step. Thinning is an optional step, though generally it is unavoidable if the recogniser is designed to be independent of the thickness of the character.

Scaling an image may be thought of as the combination of two operations. First, each row can be scaled to the desired width and then, each of the columns of the row-normalised image can be scaled to the desired height to get the required image. Thus, in spite of the fact that the image is a two-dimensional signal, the normalisation process is a composition of a number of one-dimensional resampling processes. If the image is stretched in a particular direction, it is upsampling and downsampling refers to the shrinking

One popular method of image re-sampling is bilinear interpolation. The drawback of bilinear interpolation is the loss of information in the thinned parts of the image. To overcome this drawback, we approach the problem from the viewpoint of multirate signal processing. To re-sample a signal of length Mk to a length Lk (k being the greatest common factor of Mk and Lk), we pass the signal through an expander followed by a low pass filter followed by a decimator. The expansion ratio of the expander is L, the decimation factor of the decimator is M and the passband of the low-pass filter is from $-\pi/l$ to $+\pi/l$, where $l = \max(L, M)$. The block diagram of the system is shown in figure 6. The combination of the LPF and the decimator can be implemented as a polyphase realisation to reduce computation. Non-causal low-pass filters can be realised, since there is no constraint from causality point of view.

In the frequency domain, the resampling process is equivalent to the stretching (shrinking) of the spectrum by a factor of M/L in case M > L (M<L). Thus resampling can be achieved by padding (L-M)k zeros symmetrically around the Mk/2th coefficient of the DFT vector of the signal, in case L>M. If L<M, then (M-L)k number of coefficients can be truncated symmetrically around the Mk/2-th coefficient of the DFT vector of the signal. The IDFT of the resulting truncated or zero padded DFT is the re-sampled signal. In the case of under-sampling, truncating nonzero coefficients of DFT corresponds to low-pass filtering to avoid aliasing of the signal and causes the associated loss of information.

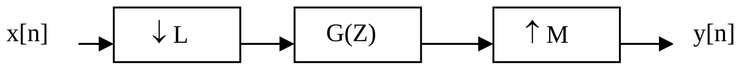


*Figure 6: Block diagram of re-sampling process. x[n] is the signal to be re-sampled. y[n] is the resampled signal.*

Thinning is a process of converting the object of interest to a contour of one pixel width. Here the thinning algorithm given in [2] has been employed. Figure 7 below show a segmented character and the corresponding normalised and thinned images.

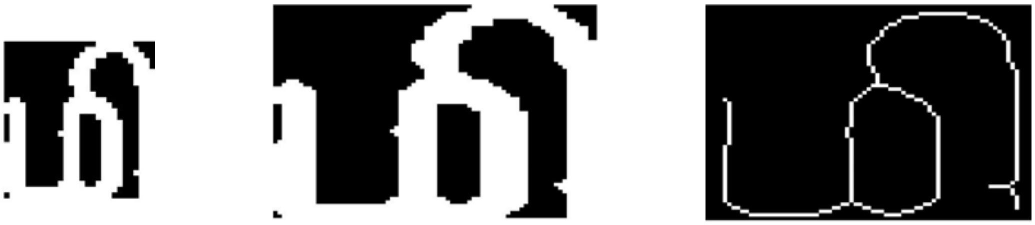


*Figure 7a: Segmented symbol of original size 28 x 28*

*Figure 7b: Symbol after size normalisation to 60 x 75*

*Figure 7c: Thinned symbol 60 x 75*

*Figure 7: Normalised and Thinned images*

## 5. Symbol Recognition

The biggest challenge in developing OCR in Indian languages is the vastness of the character set and the complex geometry of the symbols to be recognised. Some of the Tamil characters are made up of 2 or 3 disconnected symbols, and this calls for a classification strategy which first identifies the individual symbols, and in a subsequent stage, combines the appropriate number of successive symbols to detect the character. The alphabet contains 154 different symbols. This increases the recognition time and the complexity of the classifier. It is desirable to divide the characters into some clusters, so that the search space while recognition is smaller which in turn results in lesser recognition time. This also reduces the probability of confusion or wrong clustering. A tree structured classification scheme is introduced, which is meant for Tamil script. It is possible to design similar type of structure for other Indian languages like Hindi, Bengali, Kannada and Marathi.

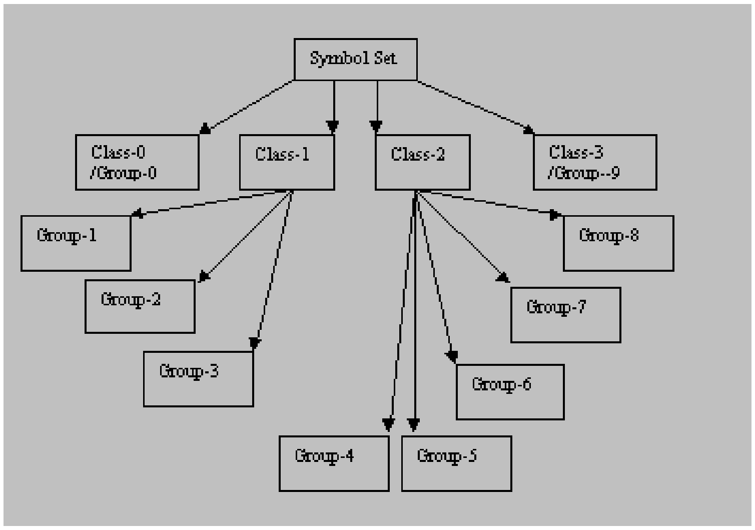


*Figure 8: Tree structure of the classifier.*

There are three different levels of classification in Tamil

- Classification based on height
- Classification based on matras/extensions
- Recognition at the third level.

## 5.1 Classification based on height

All the characters in Tamil occupy the portion 2 of the text line. Moreover, 60% of the characters appearing in a regular Tamil text belong to Class-0 from our analysis, where the characters are confined to segment-2 only. So, if the horizontal projection profile of a text line is taken, the projection of any row of this segment has a very high value compared to that of a row belonging to the segments 1 or 3. The sharp rise of the profile indicates the transition from segment-1 to segment-2. Similarly, the sharp drop in the profile value is associated with the transition from segment-2 to segment-3. The line boundaries can be detected by detecting the above mentioned sharp changes in the value of the projection profile. The text line of any Tamil text having 3 different segments is shown in the figure 9 below. Since the segments occupied by a particular symbol are fixed and generally invariant to font differences, a symbol can be associated with one of the four different classes depending upon its occupancy of these segments:

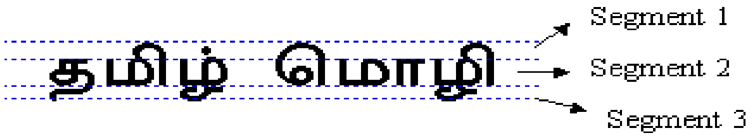


*Figure 9: Output of 4-line segmentation*

Symbols occupying segment 2 alone - Class 0.
Symbols occupying segments 1 and 2 - Class 1.
Symbols occupying segment 2 and 3 - Class 2.
Symbols occupying all 3 segments - Class 3.

## 5.2 Classification based on matras/extensions

This level of classification is applied only to symbols of classes 1 and 2, which have upward (matras) and downward extensions. These are further classified into Groups, depending on the type of ascenders and descenders present in the character. This level of classification is feature based i.e. the feature vectors of the test symbol is compared with the feature vectors of the normalised training set. The features used in this level are second order geometric moments and the classifier employed is the nearest neighbour classifier.

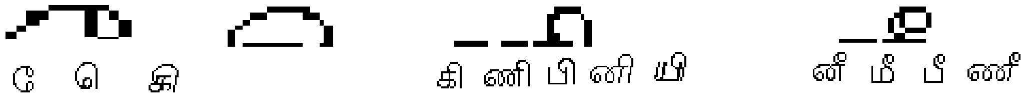


*Figure 10a: Group 6*

*Figure 10b: Group 7*

*Figure 10c: Group 8*

*Figure 10: Subgroups of Class 1*

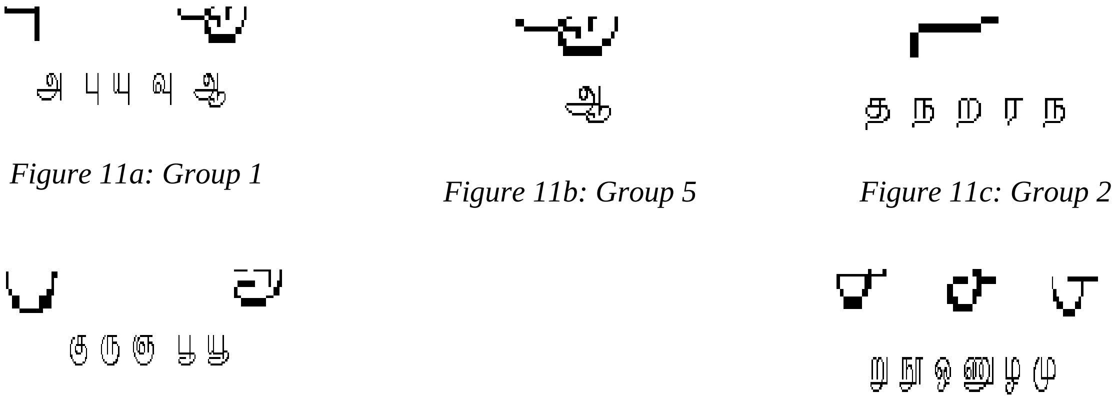


*Figure 11a: Group 1*

*Figure 11b: Group 5*

*Figure 11c: Group 2*

*Figure 11d: Group*

*Figure 11e: Group 3*

*Figure 11: Subgroups of Class 2*

## 5.3 Recognition at the third level.

In the third level, feature-based recognition is performed. For each of the groups, the symbol normalisation scheme is different. The dimensions of the feature vector are different for different groups, as their normalisation sizes are different. Truncated DCT coefficients are used as the second set of features. Nearest neighbour classifier is used for the classification of the symbols. The symbol is labelled with the label of symbol in the training set, which is nearest to it in the feature space. Euclidean distance is used in the case of DCT features. A symbol is declared unknown if its nearest neighbour is beyond a certain threshold.

## 5.4 Post –Processing

Some of the confused characters as shown below in figure 12 go for one more level wherein certain heuristics are applied to solve the confusion.

| | |
|---|---|
| ஒ | ஓ |
| ல | வ |
| ற | ந |
| ஐ | ஜ |
| ம | ய |
| ன | ள |

*Figure 12: Confused characters*

## 6. Training set

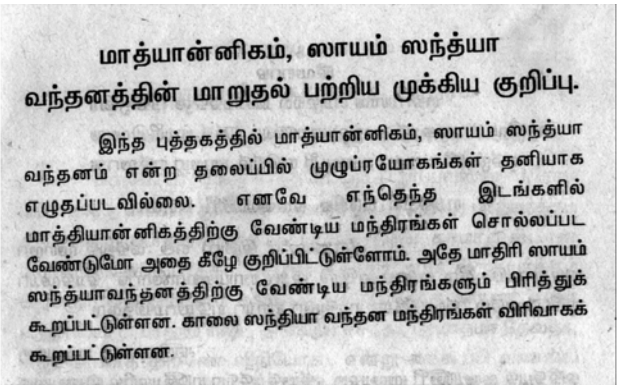
In order to obtain good recognition accuracy, we created a vast database of training set is created of size exceeding 4000 samples. Each character has 25 to 50 samples collected from various magazines, novels, and technical papers and from various Tamil books. The database also covers bold and italic characters, as also special symbols like comma, semicolon, colon and numerals. Fonts like TM-TT

Valluvar, TAB_Arulmathi, Inaimathi, TM-TT Bharathi and TAM-Aniezhai provided by Tamil text editors like Kamban, Murasu Anjal and iLEAP are also included. We have handled font sizes from 14 to 20 in testing the system.

The training feature set contains the features obtained from the normalised and thinned symbols and a label to identify the character. The features of the unknown symbol are compared with the features of the known symbols in the training set and a label of the one that closely matches in the training set is assigned to the test character.

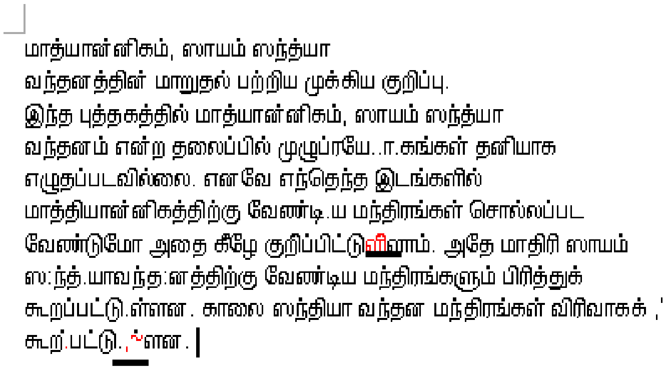
## 7.1 Classification Results

The overall recognition accuracy on a fairly good document (with considerably less amount of noise) is around 98%. Some of the results are shown in figures 13 and 14 below.



மாத்யான்னிகம், ஸாயம் ஸந்த்யா
வந்தனத்தின் மாறுதல் பற்றிய முக்கிய குறிப்பு.
இந்த புத்தகத்தில் மாத்யான்னிகம், ஸாயம் ஸந்த்யா
வந்தனம் என்ற தலைப்பில் முழுப்ரயோகங்கள் தனியாக
எழுதப்படவில்லை. எனவே எந்தெந்த இடங்களில்
மாத்தியான்னிகத்திற்கு வேண்டிய மந்திரங்கள் சொல்லப்பட
வேண்டுமோ அதை கீழே குறிப்பிட்டுள்ளோம். அதே மாதிரி ஸாயம்
ஸந்த்யாவந்தனத்திற்கு வேண்டிய மந்திரங்களும் பிரித்துக்
கூறப்பட்டுள்ளன. காலை ஸந்தியா வந்தன மந்திரங்கள் விரிவாகக்
கூறப்பட்டுள்ளன.

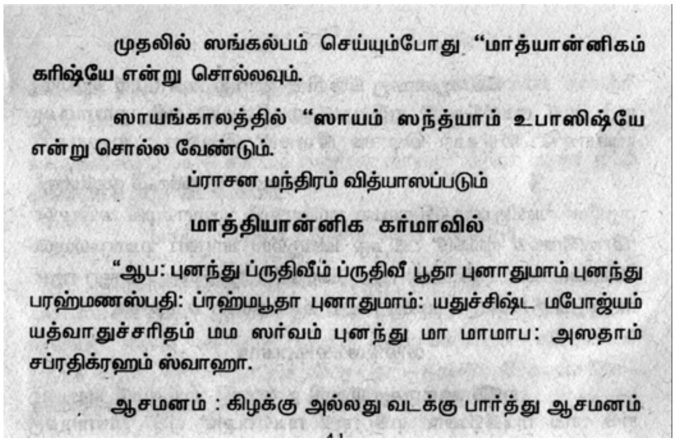
*Figure 13a: Original document*



மாத்யான்னிகம், ஸாயம் ஸந்த்யா
வந்தனத்தின் மாறுதல் பற்றிய முக்கிய குறிப்பு.
இந்த புத்தகத்தில் மாத்யான்னிகம், ஸாயம் ஸந்த்யா
வந்தனம் என்ற தலைப்பில் முழுப்ரயே..ா.கங்கள் தனியாக
எழுதப்படவில்லை. எனவே எந்தெந்த இடங்களில்
மாத்தியான்னிகத்திற்கு வேண்டி.ய மந்திரங்கள் சொல்லப்பட
வேண்டுமோ அதை கீழே குறிப்பிட்டுளிளாம். அதே மாதிரி ஸாயம்
ஸ:ந்த்.யாவந்த:னத்திற்கு வேண்டிய மந்திரங்களும் பிரித்துக்
கூறப்பட்டு.ள்ளன. காலை ஸந்தியா வந்தன மந்திரங்கள் விரிவாகக் ,'
கூறப்.பட்டு.,~ளன.

*Figure 13b: Recognised document*

**Recognition accuracy**
Total number of symbols = 351
Rejected = 1 (~ sign)
Misclassified = 2
Recognition rate = 99.48%



முதலில் ஸங்கல்பம் செய்யும்போது "மாத்யான்னிகம் கரிஷ்யே என்று சொல்லவும்.

ஸாயங்காலத்தில் "ஸாயம் ஸந்த்யாம் உபாஸிஷ்யே என்று சொல்ல வேண்டும்.

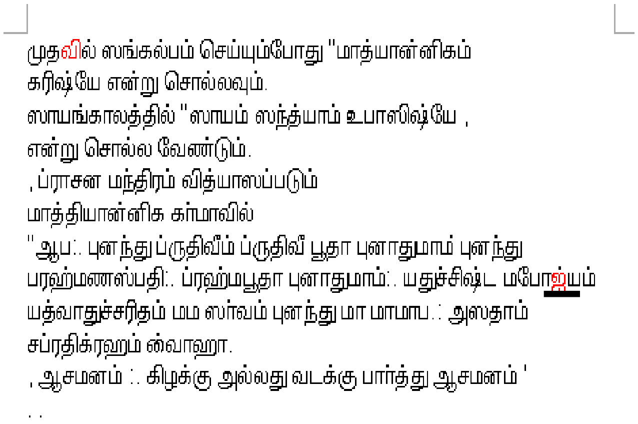
ப்ராசன மந்திரம் வித்யாஸப்படும்

மாத்தியான்னிக கர்மாவில்

"ஆப: புனந்து ப்ருதிவீம் ப்ருதிவீ பூதா புனாதுமாம் புனந்து பரஹ்மணஸ்பதி: ப்ரஹ்மபூதா புனாதுமாம்: யதுச்சிஷ்ட மபோஜ்யம் யத்வாதுச்சரிதம் மம ஸர்வம் புனந்து மா மாமாப: அஸதாம் சப்ரதிக்ரஹம் ஸ்வாஹா.

ஆசமனம் : கிழக்கு அல்லது வடக்கு பார்த்து ஆசமனம்

*Figure 14a: Original document*



முதவில் ஸங்கல்பம் செய்யும்போது "மாத்யான்னிகம்
கரிஷ்யே என்று சொல்லவும்.
ஸாயங்காலத்தில் "ஸாயம் ஸந்த்யாம் உபாஸிஷ்யே ,
என்று சொல்ல வேண்டும்.
, ப்ராசன மந்திரம் வித்யாஸப்படும்
மாத்தியான்னிக கர்மாவில்
"ஆப:. புனந்து ப்ருதிவீம் ப்ருதிவீ பூதா புனாதுமாம் புனந்து
பரஹ்மணஸ்பதி:. ப்ரஹ்மபூதா புனாதுமாம்:. யதுச்சிஷ்ட மபோஜ்யம்
யத்வாதுச்சரிதம் மம ஸர்வம் புனந்து மா மாமாப.: அஸதாம்
சப்ரதிக்ரஹம் ஸ்வாஹா.
, ஆசமனம் :. கிழக்கு அல்லது வடக்கு பார்த்து ஆசமனம் '
. .

*Figure 14b: Recognised document*

**Recognition accuracy**

Total number of symbols = 330
Rejected = 0
Misclassified = 2
Recognition rate = 99.39%

## Conclusion

In this paper an attempt has been made to develop a complete Tamil OCR system which works on a font independent and size independent scenario. A precise skew detection approach has been employed which enables us to find out the skew angle within the range of ± 0.06° about the actual value. Skew rotation is performed on a gray level image to avoid quantisation effects and to reduce the distortions present in the character. Geometric moments and discrete cosine transforms are the features employed and the classifier used is nearest neighborhood. The overall recognition accuracy is around 98%.

## References

[1] U. Pal and B. B. Choudhuri. *A Complete Printed Bangla OCR Systrem*. Pattern Recognition. Vol 31. May 1998.

[2] R C Gonzalez and R E Woods (1999). *Digital Image Processing*, Addison – Wesley Press, New York.

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

[4] R O Duda and P E Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons.

[5] Kaushik Mahata and A. G. Ramakrishnan, *Precision Skew Detection through Principal Axis*, Proc, Intern. Conf. on Multimedia Processing and Systems, Chennai, Aug. 13-15, 2000.