DOCUMENT IMAGE ANALYSIS: WITH SPECIFIC APPLICATION TO TAMIL NEWSPRINT

K.H.Aparna, Sumanth Jaganathan, P.Krishnan, V.S.Chakravarthy
Department of Electrical engineering,
IIT Madras,
Chennai-600036.
mailto: ee01m03@ee.iitm.ernet.in

ABSTRACT:

We present an early version of a complete Optical Character Recognition (OCR) system for Tamil newsprint. All the standard elements of OCR process like deskewing, preprocessing, segmentation, character recognition and reconstruction are implemented. Experience with OCR problems teaches that for most subtasks involved in OCR, there is no single technique that gives perfect results for every type of document image. We have used the strength of artificial neural networks in empirical model building for solving the key problems of segmentation and character recognition. Text segmentation of Tamil newsprint poses a new challenge owing to its italic-like font type; problems that arise in segmenting such text are discussed. The final document is reconstructed in HTML document format

1. INTRODUCTION

The document image processing includes preprocessing steps of skew correction, binarization and noise removal, segmentation of the image into blocks and classification of the blocks into text, tables, graphics, and line diagrams etc. and finally reconstruction of the original document.

A few degrees of skew become inevitable to the document image whether the document is fed manually or mechanically. An algorithm based on Hough transform, involving transformation of the coordinates to Hough space, of which the aligned pixels will give rise to peak, which gives the skew angle is described in [1]. Other methods proposed for detecting skew include projection profile analysis [2], image gradient analysis [3], morphological transforms [4] and correlation between lines at a fixed distance [5]. But most of the above methods have a drawback of computational complexity, which is proportional to the desired accuracy. Our algorithm assumes the presence of text part in the document image and uses the simple Gabor filter response in different orientations for determining the skew angle.

Binarization is the process of converting the gray scale images to binary images by comparing each pixel value with a threshold. Ostu proposed a method for threshold selection using gray scale histogram in [6]. Another binarization method based on texture features is described in [7].

The next important step of document image analysis is segmenting the page into blocks and classification of blocks into text and non-text. Wang and Srihari in their analysis of newspaper image documents [8] employed the following methods. For page segmentation homogeneous rectangular blocks are first segmented out of the image using methods such as run length smearing algorithm (RLSA) and recursive X-Y cuts (RXYC) which perform well only on documents with rectangular layouts. And their classification approach is based on statistical textural features and feature space decision techniques. Pavlidis and Zhou [9] described a class of techniques based on smeared run length codes that divides a page into gray and nearly white parts. Segmentation is then performed by finding connected components either by the gray elements or of the white, the latter forming white streams that partition a page into blocks of printed material. Their classification method is based on across scan line correlation method. Page segmentation and classification based on texture analysis using neural networks is described in [10]. Our approach combines both the neural network approach and boundary detection for improving the accuracy of segmentation and classification.

The rapid spread of computer literacy and usage in the 90's in India had resulted in a growing interest in OCR in Indian languages. Some of the works in OCR include an approach to recognition of Tamil isolated characters based on condensed run method and symbolic run method [11], a syntactic pattern analysis system with an embedded picture language has been designed for the purpose of recognition of Devanagari script [12], a rule based contextual post-processor for the recognition of isolated characters of Devanagari script [13]. All the above works on OCR deal with recognition of characters only and do not address the larger problem of document image segmentation.

Chaudhuri and Pal ([15] and [16]) work on a complete OCR system for printed Bengali documents, which uses structural feature-based tree classifier for character recognition. This is the first OCR system developed in Indian languages.

An overview of segmenting machine printed characters when touching and broken characters are encountered is discussed by Yi Lu [14], but the cases of touching italic characters has not been dealt with.

The present paper deals with development of a complete OCR system for printed Tamil documents. The block diagram shown in figure (1) gives the various steps involved in our approach.

The paper is organized as follows. In Section 2 preprocessing of the document image is described. Section 3 explains the segmentation of the page into blocks. Section 4 deals with Tamil character recognition and reconstruction of the document image. Finally the paper concludes with a discussion in Section 5.

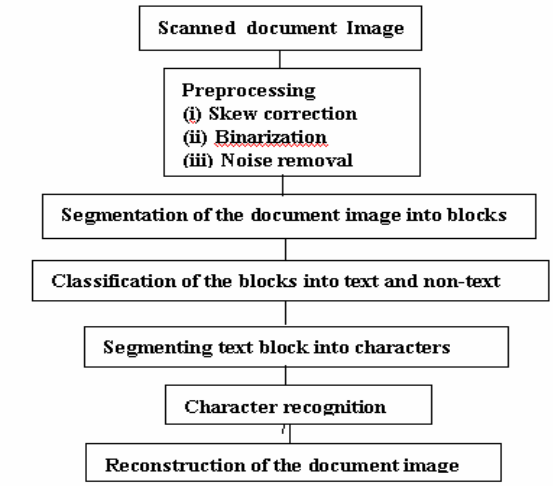


Figure 1. Steps involved in complete OCR for Tamil documents

2. PREPROCESSING

The document image obtained by the scanning of a hard copy magazine document as a black & white photograph at 300 dpi using a flat-bed scanner is represented as a two dimensional array. A document of size 8.27 X 11.69 inches scanned at 300 dpi would yield an image of 3324 X 2466 pixels.

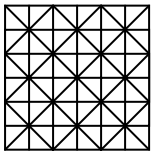
Preprocessing stage consists of four steps- compression, skew correction, binarization and noise removal.

2.1. Image size reduction:

Some of the image analysis techniques of text recognition, skew detection, page segmentation and classification are applied on scaled down images. Such reduction not only increases speed of processing, but also gives more accurate results for the specified tasks. For scaling down an image by half, a window of 2 X 2 pixels in the parent image is replaced by a single pixel whose value equals the median of the 2 X 2 window. The image that is scaled down by ¼ is referred to as *doc1by4.*

2.2 Text and non-text recognition:

Finding text regions in the document image is essential for skew estimation. For finding the text part we use a Radial Basis Function neural network (see Appendix 1(b)). The network is trained to distinguish between text and non-text (non-text includes graphics, titles, line drawings). The input patterns for training the RBF neural networks are the 20 Gabor filter as shown in figure 2 (see Appendix 1(a) for Gabor filters) responses with five each in horizontal, vertical and on both diagonal directions



F*igure 2:20 Gabor filters (40X40 window)*

The neural network has two outputs, one for text and the other for non-text. The network is presented with Gabor responses calculated from 40 X 40 windows of doc1by4 images

The *doc1by4* image and the region marked as text by the neural network are shown in Figure 3.

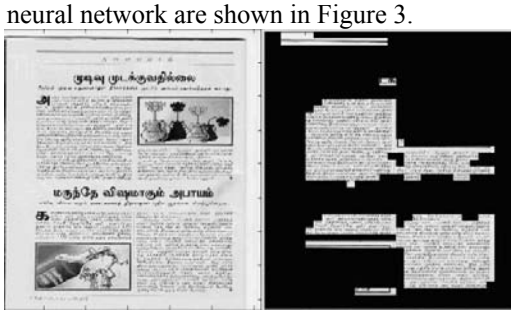


Figure 3: Reduced version (doc1by4) of the original image and neural network output after text recognition

From Figure 3 it is evident that although most of the text part is recognized correctly there are a few spaces where text is recognized as non-text and vice versa. So for a perfect text, non-text block recognition we will use this output in later stages

2.3. Skew Correction:

For skew angle detection Cumulative Scalar Products (CSP) of windows of text blocks with the Gabor filter at different orientations are calculated. Orientation with maximum CSP gives the skew angle. Alignment of the text line is used as an important feature in estimating the skew angle. We calculate CSP for all possible 50X50 windows on the text recognized image (from *doc1by4* image) and the median of all the angles obtained gives the skew angle. The skew angle for the document in figure 3 (left) is found to be 0.5 degrees

2.4. Binarization:

Binarization is the process of converting a gray scale image (0 to 255 pixel values) into binary image (0 and 1 pixel values) by selecting a global threshold that separates the foreground from background. Each pixel is compared with the threshold and if it is greater than the threshold it is made 1 or else 0. Here a threshold of 158 is chosen.

2.5. Noise removal:

The noise introduced during scanning or due to page quality has to be cleared before further processing. For this the document is scanned for noise using a moving 5 X 5 window. If all non-zero pixels are confined to the central 3 X 3 section, all those pixels are set to 0.

3. SEGMENTATION

Segmentation of the document image involves two steps: determination of page layout (segmenting the page into blocks) and classification of the blocks.

3.1. Page Segmentation:

When the binarized image of the document is observed we find that if all the wide and long white spaces are removed without touching the white spaces between text lines the page can be segmented into blocks. The result is shown in figure 4.

Boundaries of the segmented blocks are found by "contour following" (see figure 5)

Figure 6 shows all the block coordinates that are stored.

This process works well for different kinds of layouts and is fairly consistent and accurate.

The problem of text and graphic together being stored as a single block is encountered in segmentation when a text part exists between a text region and non-text region

Figure 4: Image with borders and all vertical and horizontal straight lines removed used for finding the block coordinates.

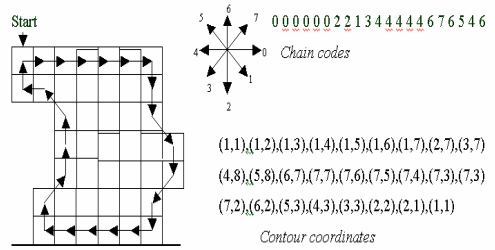


Figure 5: Contour following

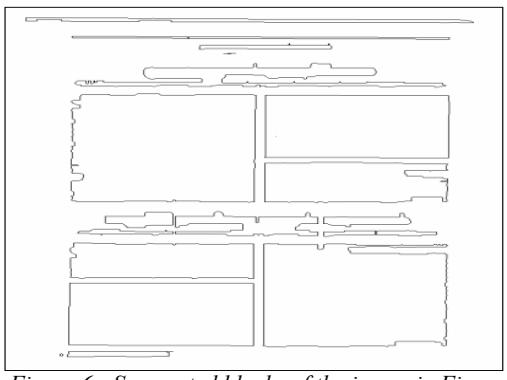


Figure 6: Segmented blocks of the image in Figure 4

3.2 Classification of the blocks:

The blocks obtained from the segmentation stage of Section ***3.1***, and the blocks obtained from text recognition stage of section ***2.2*** are combined to give an accurate delineation of text blocks. In the case of figure 3, all the four text blocks are extracted and passed on for character recognition; the four image blocks are stored as image files.

4. TAMIL CHARACTER RECOGNITION

The text blocks have to be initially segmented into lines, words and characters.

For the text block segmentation and character recognition the inverted binarized document (i.e. 0 for back ground and 1 for foreground) is being taken. For character recognition the original document (without any scaling) is being taken.

4.1 Line, word and character segmentation:

For optical character recognition, the text blocks are segmented into lines, lines into words and then into individual characters.

(i) Line Segmentation:

For segmentation of text blocks into lines the horizontal projection on the y-axis is made use of. The threshold value is chosen by manual intervention.

(ii) Word and character segmentation

Since the font used in Tamil newsprint is typically italic like, with the characters oriented at 79.21^0 with the horizontal, for segmenting the line into words and characters inclined projection is taken on the text line.

The segmentation is accurate if we have enough space between characters. If the characters are too close to each other or touching then segmenting becomes difficult, (a few such text parts are shown in figure 7).

Figure 7: Characters, which are touching or too close

For extracting characters that are too close but non-touching, connected- component extraction method is employed, in which components are segmented not by separation in one dimension but based on their connectedness.

4.2 Recognition of characters

A Radial basis function neural network (Appendix 1b) is trained for the recognition of characters. All the Tamil isolated characters including the vowels, consonants, the special or the grantha letters and the English numerals 0 to 9 and the punctuation marks making a total of 157 characters are taken for training the neural network.

The characters are placed at the center of a 52 X 52 window and the input patterns to the RBF neural network are obtained by taking the dot product of the character with each of the 40 Gabor filters with 10 along each of four directions. The RBF neural network has 157 outputs each output corresponding to an alphabet.

This trained neural network is used for the recognition of the segmented characters.

Figure 8 shows the text part given for recognition and the output of the neural network for character recognition in HTML format.

When the text block having a few touching characters (figure 8) is sent for character recognition, 94% recognition rate is obtained. In general, for other documents it varied from 85 to 90 percent depending on the touching characters present in the text part.

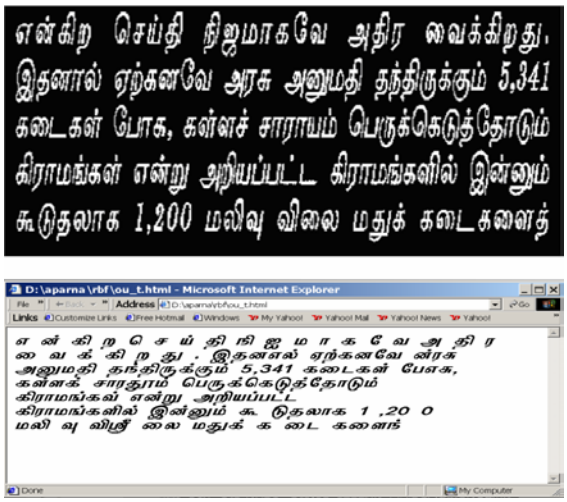


Fig 8. Reconstructed text part (in HTML format)

4.3 Reconstruction of the document image

The non-text parts are stored in files and all the recognized text and non-text files are put together in HTML format. Figure 9 shows the document image of Figure 3 reconstructed into HTML format.

Figure 9 : Document reconstructed in HTML format

5. Conclusions and Discussions:

We present a complete document image analysis system for Tamil newsprint. The system includes the full suite of processes from skew correction, binarization, segmentation, text and non-text block classification, line, word and character segmentation and character recognition to final reconstruction. Only the reconstruction step is currently done manually. Neural networks are applied to 2 subtasks: 1) text block identification, and 2) character recognition.

In the entire OCR process, the toughest challenges are faced in document segmentation and character recognition. Document segmentation is recognized as a hard problem and it may not possible to formulate a single algorithm, which works with all kinds of documents. Our approach gave reasonable segmentation results with the class of document images chosen in the present work (We placed some more reconstructed documents at the following URL-http://www.geocities.com/aparna_1308/Text_recognized_images.html. http://www.geocities.com/aparna_1308/segmented_image.html). The applicability of the technique for a larger class of Tamil newsprint is yet to be seen. Moreover, we have to overcome the errors in segmentation, which occur when the text and non-text are too close.

Currently characters of only a single font and font size are being recognized. To handle more fonts we propose to train a separate neural network for each font. We assume that font type used in a given newsprint sample is known as prior knowledge. The case of touching characters presents a serious difficulty in character segmentation. This problem will be taken up as part of our future extensions of the current system.

An important lacuna in our present system is absence of a suitable document model. A document is treated as a collection of disparate items without any logical structure connecting all of them. Future efforts will be focused on embedding various document components into a logical structure. This will help to make the reconstruction process also automatized.

REFERENCES:

(1). Bin Yu and A.K.Jain, " *A robust and fast skew detection algorithm for generic documents*", Pattern Recognition, Vol. 29,No 10,pp 1599-1629,1996.

(2). A.Bagdanov and J.Kanai, " *Projection profile based Skew estimation Algorithm for JBIG compressed images*", Intl' Conf. On Document Analysis and Recognition, August 18-20, Ulm, Germany, p 401-405, (1997).

(3). J.Sauvola and M. Pietikainen, "*Skew angle detection Using Texture direction Analysis*" In Proc, of the 9th Scandinavian Conference on Image analysis, pages 1099-1106, Uppsala, Sweden, June 1995

(4). S.Chen and R.M. Haralick. "*An automatic algorithm for Text Skew estimation in document images using recursive morphological transforms*", In proc. Of the first IEEE International conference on Image Processing, pages 139-143, Austin, Texas, 1994.

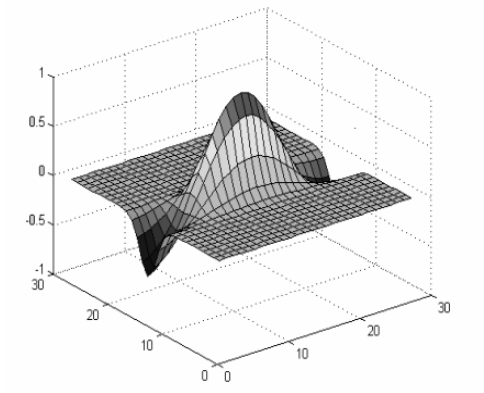
(5). Hong Yan, "*Skew correction of Document Images Using Interline Cross correlation*", CVGIP, Vol.55, No.6, November, pp. 647-656,1982.

(6) N.Ostu, "*A threshold selection method from gray scale Histograms*", IEEE Trans on man Cybernet.,62-66,1979.
(7). Y.Liu and S.N.Srihari, "*Document image binarization based on texture features*", IEEE transactions on pattern analysis and machine intelligence, Vol. 19,No.5 may 1997.
(8). D.Wang and S.N.Srihari, *"Classification of newspaper image blocks using texture analysis",* Comput. Vision Image Graphics Process. 47, 1989,327-352.
(9). T.Pavlidis and J.Zhou, " *Page Segmentation and Classification*", CVGIP Vol. 54,No 6,pp 484-496,Nov 1992.
(10). A.K.Jain and Yu Zhong, "*Page segmentation Using Texture analysis*", Pattern recognition, Vol. 29,No.5,pp. 743-770,1996.
(11) G.Siromoney, R. Chandrasekaran and M. Chandrasekaran, "*Machine recognition of printed Tamil characters*", Pattern Recognition, vol. 10 (1978).
(12). R.M.K.Sinha and H.Mahabala, "*Machine recognition of Devanagiri script*", IEEE Trans. Syst. Man Cybern. Vol. 9(1979).
(13). R.M.K. Sinha, *"Rule based contextual post-processing for Devanagiri text recognition*", Pattern Recognition vol. 20,p 475-485 (1987)
(14). Yi Lu, "*Machine printed character recognition-An overview",* Pattern recognition, Vol.28, No 1, pp 67-80,1995.
(15) B.B.Chaudhuri and U.Pal, *"A complete printed Bangla OCR system",* Pattern Recognition, Vol. 31,No 5,pp 531-549,1998.
(16). B.B.Chaudhuri and U. Pal, "*An OCR system to read two Indian language scripts: Bangla and Devanagari (Hindi)*", Intl' Conf. On Document Analysis and Recognition, August 18-20, Ulm, Germany, p1011-1015, (1997).
(17). J. E. Moody, C. J. Darken, "*Fast Learning in Networks of Locally Tuned Processing Units*," Neural Computation Vol.1, pp. 281-294,1989.

Appendix 1

(a) *Gabor Filter*

The Gabor functions are Gaussians modulated with a cosine.



Gabor Function

Gabor filters: -
1.Are localized in both spatial and frequency domain.
2.Extract image information necessary for recognition.
3.Are known to emulate receptive fields of human visual system.

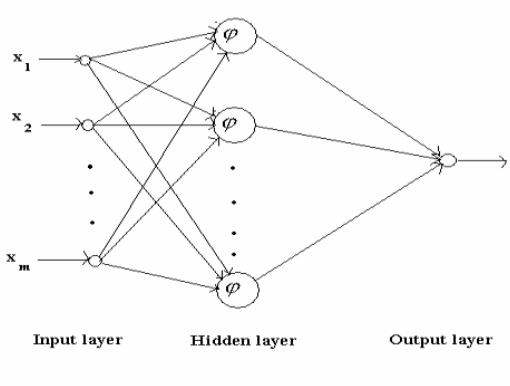
$$f\ (x,y;\sigma,\lambda,\omega,\theta)=\frac{1}{2\pi\sigma^2\lambda}\exp\left\{-\frac{\lambda^2x'^2+y'^2}{2\lambda^2\sigma^2}\right\}\cos(2\pi\omega x'),$$

$$x'=(x-x_0)Cos\theta+(y-y_0)Sin\theta$$
$$y'=-(x-x_0)Sin\theta+(y-y_0)Cos\theta$$

Where $\lambda=r_{sh}/r_{lg}$, where r_{sh} is half the width of the Gabor filter, r_{lg} is half the length of the Gabor filter,(x_0, y_0) are the coordinates of center and θ the orientation of the Gabor filter.

Since Gabor function responds optimally to periodic stripe pattern (with wave length $(1/2\pi\omega)$ and orientation (θ), they are aptly suited for recognizing text blocks, which may be regarded as oriented stripe patterns.

(b) *RBF Neural network*



RBF Neural network

1.RBF Neural network fits an unknown function using a weighted sum of Radially Symmetric Functions e.g., Gaussians.
2.It has a three-layered architecture comprising
–Input Layer
–Hidden layer/RBF Layer/Kernel Layer
–Output Layer

The output of the i^{th} output node, $f_i(\mathbf{x})$, when an input vector **x** is presented, is given by

$$f_i(x)=\sum w_{ij}\varphi_j(x)$$

Where $\varphi_j(x)=\varphi(\|x-x_j\|)$ is an RBF.

$$\varphi_j(x)=\exp\left[\frac{-\left(\|x-x_j\|^2\right)}{2\sigma^2}\right]$$

Where σ determines the width of the receptive field, w_{ij} are the second layer weights and x_j's are the mean vectors of RBF's. A method for training RBF neural network is described in [17]. The RBF network is chosen of all the feed forward networks for reasons of speed and ease of training.