

# A COMPLETE OCR DEVELOPMENT SYSTEM FOR ORIYA SCRIPT

*Sanghamitra Mohanty, Hemanta Kumar Behera*

*RC-ILTS-Oriya,*

*Dept. of Comp. Sc. And Appl.,*

*Utkal University, Bhubaneswar, Orissa, India-751004.*

*sangham1@rediffmail.com, kumar\_hemanta@yahoo.com*

## Abstract

The Optical Character Recognition is a process of automatic recognition of different characters from a document image. The process involves clear and unambiguous recognition, analysis and understanding of the document content. Document processing as it obviously means, the machine processing of data that is given in the form of a document image. Here the processing of document includes text recognition, text presentation, formatting, skew correction and omni font handling. Intelligent document processing includes the syntactic and semantic processing of text. A document image is given as input and after the intelligent processing it gives a document rather in some more complete form. The task of recognition can be broadly separated into two categories: the recognition of machine printed data and the recognition of handwritten data. Machine printed characters<sup>[4]</sup> are uniform in size, position and pitch for any given font. In contrast, handwritten characters are non-uniform; they can be written in many different styles and sizes by different writers and by the same writer.

There are lots of application areas where, OCR can help. Major areas are described below.

1. Preserve old documents in electronics format.
2. Save document images within limited space.
3. Help visually impaired persons to read the content on the document.

We developed an OCR system for Oriya script. Oriya alphabets are complex in nature and consists of 115 alphabets among which around 60 characters are very difficult to recognize because they are composite characters (conjuncts). Another major problem is due to similar shaped character. For example in Oriya the letter "L" is similar with "S". A set of algorithms have been that attempts to work with a subset of features in a character that human would typically see for the identification of machine printed characters. The feature point of human interest in an image, a place where something happens i.e. it is a point which is distinguish from all other points in that image matrix. It could be an intersection between lines, or a corner, or a dot surrounded by space.

In a broad sense Document Image Processing consists of 4 steps namely

1. Preprocessing
2. Feature Extraction and Classification
3. Recognition
4. Post Processing

Preprocessing is the most important step of character recognition, which confirms the input image into a most valuable and correct format that can be fed to the Recognition Engine. This process involves several activities which is given below

1. Digitization and Noise Cleaning
2. Skew detection and correction
3. Line Extraction
4. Word, Character Extraction and *matra* extraction.
5. Thinning
6. Zooming

An optically scanned document image is usually an integer (gray-valued) array; a process of spatial sampling and simultaneous conversion of light photons to electric signals obtain it. A high-resolution *bit map* is sufficient to capture shades of gray in the eye of the human perceiver (such images are called *half-tone*). An optically scanned document image can be converted into a bit map by a global thresholding operation: pixel values below the threshold are deemed to be black (value 1) and those above deemed white (value 0). The threshold itself can either be predetermined or calculated from the image histogram. e.g., the valley of a bimodal histogram.

The process of removing unwanted pixels from the input image is known as noise cleaning<sup>[3]</sup>. Technically, we can say noise is one of the pixel value i.e. intensity among intensity values but it has unique characteristic with respect to its neighboring pixel values i.e. intensity values. Noise arises due to so many reasons namely old document, low paper quality, and dust particles on the surface of scanner, low quality ink and low quality printing machines. Here, we developed an algorithm, which removes isolated points using global thresholding applied on the entire range of pixels and using adaptive technique.

During scanning the document may be slanted leading to problem at the time of line extraction. The first and foremost step after noise cleaning is skew detection and correction. After noise cleaning the image matrix consists of two gray values one value denotes background and other denotes foreground. In this paper we assume 0 for foreground and 1 for background. Angular projection profiles are prepared for -8 degree to +8 degree with an interval of 0.05 degree and a strip-wise histogram is constructed using these projection profiles. Then a global maxima was found out at a particular angular direction. This angle gives the detected global skew angle and then the whole document is then rotated to get the skewless image. Line extraction is one of the segmentation technique in which individual lines are extracted for the whole document image. The whole document is a matrix of 0's and 1's where 0 stands for black i.e. foreground pixel and 1 stands for white i.e. background pixel in which each row is a combination of 0's and 1's and the row which has less number of background pixels can be represented as the partition point. Here in our paper, we have developed the technique based on

horizontal projection profile that is able to extract individual lines from any document. The histograms for each line are determined and from these histograms line borders are made out as the extreme maximas. This technique can be applied to any script but for Oriya script, we have given some constraints because Oriya alphabets are generally complex in nature. The complexity arises due to so many reasons namely the occurrence of vowels after consonants etc. and is termed as modifiers (*matra*) and has different shapes. This is not only a problem of Oriya script but also for some other Indian Scripts. Hence this type of constraint can be applied to Indian script. Modifiers in Oriya script can be appeared in four places namely left, right, upper and lower. At the time of line extraction only lower and upper *matra* or lower *phala can pose problems* and a region analysis method developed basing on thresholding to solve this type of problem. A line is a combination of words and each word extracted from the respective line depending upon the threshold on the spacing between words and each word is combination of characters and *matras* and characters are extracted by forward and backward chaining method. A major problem lies when the individual characters are connected and to solve this type of problem, the baseline characters are extracted from the line by region analysis and labeling and then a average mask is calculated based upon the individual characters. The mask is allowed to move from both forward and backward directions. Thinning is the process of extracting border pixels from the image matrix preserving its connectedness. Here the character extracted from the line gets thinned by applying the algorithm developed by Zhang and Suen (1984)<sup>[1]</sup>. A feature is a point in a pixel matrix, which is distinguished from all other pixels by its unique characteristics. The most common type of features is Structural features and Topological features. Here we used structural features for character recognition. There are ten structural features extracted from the 16x16 pixel matrix which are given below

1. Upper Part Circular
2. A vertical line on the right most part
3. Holes
4. Horizontal Run code
5. Vertical Run code
6. Number of holes
7. Position of hole

After features are successfully extracted from the respective 16x16 pixel matrix, the process goes for classifying the extracted features. We developed a tree-based classification method in which each node denotes a particular feature and all leaf nodes contain individual characters, modifiers (*matras*), digits and composite characters (*phalas*). A character in the form of 16x16 pixel matrix after preprocessing is allowed to sink into the above tree and finally reach in one of the leaf node denotes a particular character. The recognition phase has two parts. In the first phase individual characters, left and right modifiers (*matra*) are recognized based on structural features whereas in the second stage upper and lower modifiers (*matra*) are recognized based on run length code. Most composite characters<sup>[1]</sup> (*Yuktas*) are recognized by applying run length code, loop and position of hole. The above classification tree, which has been developed, may be binary or tertiary. Each level containing feature is assigned to a string of 0 and 1 i.e. if a character contains a line then 1 is assigned otherwise 0, in this way 0 and 1 values are assigned to the extracted feature and finally the matrix of pixel is matched in all levels at run time and sinks to a particular leaf node which contains a ASCII and ISCII value. In each level a condition has been checked and the recognized character sinks according to the condition and

then an ISSCII or ASSCII value is dumped into the editor.

Due to nonavailability of 100% noise freed digitized image and presence of similar shaped characters the accuracy rate is affected. To tackle this type of problem the system has been integrated to spell checker with the help of dictionary and a huge corpus.

## References

- [1] Grain U and Choudhuri B B 1998 compound character recognition by run number based metric distance. *Proc. SPIE Annual Symposium on Electronic Imaging, San Jose, USA, pp 90-97.*
- [2] Digital Image Processing, by Rafael C. Gonzalez, Richard E. Woods.
- [3] S.Mohanty, K.Sahoo and H. K. Behera," A New Algorithm for the restoration of characters in old noisy document with varying level of intensities", ISC Conference, India, Jan'2003.
- [4] S. Kahan, T. Pavlidis and H. S. Bairb, "On the Recognition of Printed characters of any font and size", IEEE Trans. Pattern Analysis Machine Intelligence, Vol-9, 1987, pp. 274 -287