

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Soft Computing Approach for Printed Hindi Words Recognition

M. Shalini, Dr. B Indira Reddy,

Head, Department of Computer Science, Kasturba Gandhi Degree & PG College for Women, West Marredpally,
Secunderabad, Telangana, India

Associate Professor, Department of Computer Science, Kasturba Gandhi Degree & PG College for Women,
West Marredpally, Secunderabad, Telangana, India

ABSTRACT: Hindi is the national language and most widely spoken language in India. Statistics show there are millions of people who communicate in Hindi. The Optical /Character Recognition (OCR) systems developed for the Hindi language have a poor recognition rate compared to OCR system for English language, the reason being that there is no separation between the characters of texts written in Hindi, as there is in English. Here we plan to propose an OCR for Printed Hindi words in Devnagari script, using Artificial Neural Networks(ANN). It uses the Two-pass algorithm, which is relatively simple to implement and understand, the two-pass algorithm iterates through 2-dimensional, binary data. ANN is used for finding final exact result. A rugged Feature Extraction algorithm is applied for classification of characters. The tool that we have built will scan the image and monitor it by pixel by pixel. To separate individual words we first pass the entire image through a canny edge detector to make a logical image for further processing. To improve scalability, we have developed a parallel SVM algorithm (PSVM), which reduces memory use through performing a row-based, approximate matrix factorization and which loads only essential data to each machine to perform parallel computation. Kohonen's Self-Organizing Maps algorithm is used to create a correspondence between the input space of stimuli and the output space constituted of the codebook elements, the code words, or neurons. A precise skew detection approach has been employed which enables us to find out the skew angle within the range of $\pm 0.06^\circ$ about the actual value. Skew rotation is performed on a grey 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 neighbourhood. The overall recognition accuracy is around 98%.

KEYWORDS: OCR, Pre-processing, Two-pass algorithm, parallel SVM (PSVM), Kohonen's Self-Organizing Maps(KSOM)

I. INTRODUCTION

Hindi is written in Devanagari alphabet and draws vocabulary from Sanskrit. Devanagari is a form of alphabet called an abugida, as each consonant has an inherent vowel (a) that can be changed with the different vowel signs. Most consonants can be joined to one or two other consonants so that the inherent vowel is suppressed. The resulting form is called a ligature. Devanagari is written from left to right. Devanagari has no case distinction, i.e. no majuscule and minuscule letters. In this research taken main role image recognition and separating Hindi characters from image to separate document.

The OCR (Optical Character Recognition) algorithm relies on a set of learned characters. It compares the characters in the scanned image file to the characters in this learned set. Generating the learned set is quite simple. Learned set requires an image file with the desired characters in the desired font be created, and a text file representing the characters in this image file.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

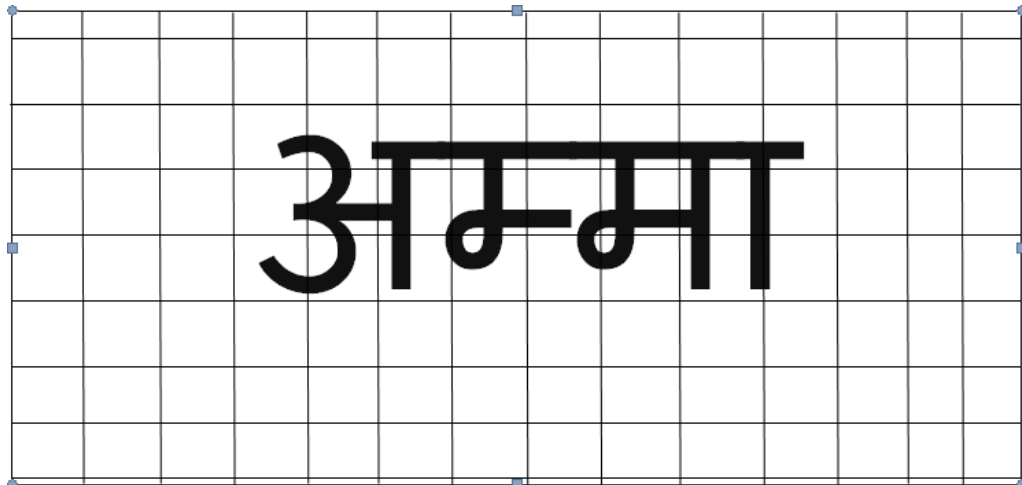
Objectives

- To study the existing methods in OCR.
- To propose an OCR for printed Hindi text in Devanagari script, using Artificial Neural Network (ANN), which improves its efficiency.
- To develop a system which accepts document image as input and produce a text file as output.
- To find an efficient method to solve the Hindi optical character recognition problem.

Image labelling algorithm

It uses the Two-pass algorithm, which relatively simple to implement and understand, the two-pass algorithm iterates through 2-dimensional, binary data. The algorithm makes two passes over the image: one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of its equivalence class.

Connectivity checks are carried out by checking the labels of pixels that are North-East, North, North-West and West of the current pixel (assuming 8-connectivity). 4-connectivity uses only North and West neighbours of the current pixel. The following conditions are checked to determine the value of the label to be assigned to the current pixel.



The above enlarged image is a pixel representation which serves purpose for our discussion. Every pixel in the bitmap image is represented by its **X and Y coordinates**. The letter "अ" in the above example shows how all the pixels are connected. The image labelling algorithm will label the entire connected pixel with the same label. The UML diagram below illustrates the flow of the algorithm.

ANN Using for finding final exact result

The ANN used in this system gets its inputs in the form of Feature Vectors. This is to say that every feature or property is separated and assigned a numerical value. The set of these numerical values that can be used to uniquely identify each character is called its Vector. Thus, a Vector Database is utilized to train the network, so as to enable it to effectively recognize each character, based on its topological properties. A character can be written in a number of ways differing in shape and properties, such as Tilt, stroke, Cursively and Overall shape. A plethora of Fonts are available for use in any commonly used Word Processing Application Software. Yet, while perceiving any text written in a variety of ways, humans can easily recognize and read each character. This is because the human perception processes the information by the features that define a character's shape in an overall fashion. Thus, while modelling the human perception model in machines, a rugged Feature Extraction algorithm is needed before an ANN (fig.1) can

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

be applied for classification of characters. Following proposed methods are inserting into ANN for getting the final result.

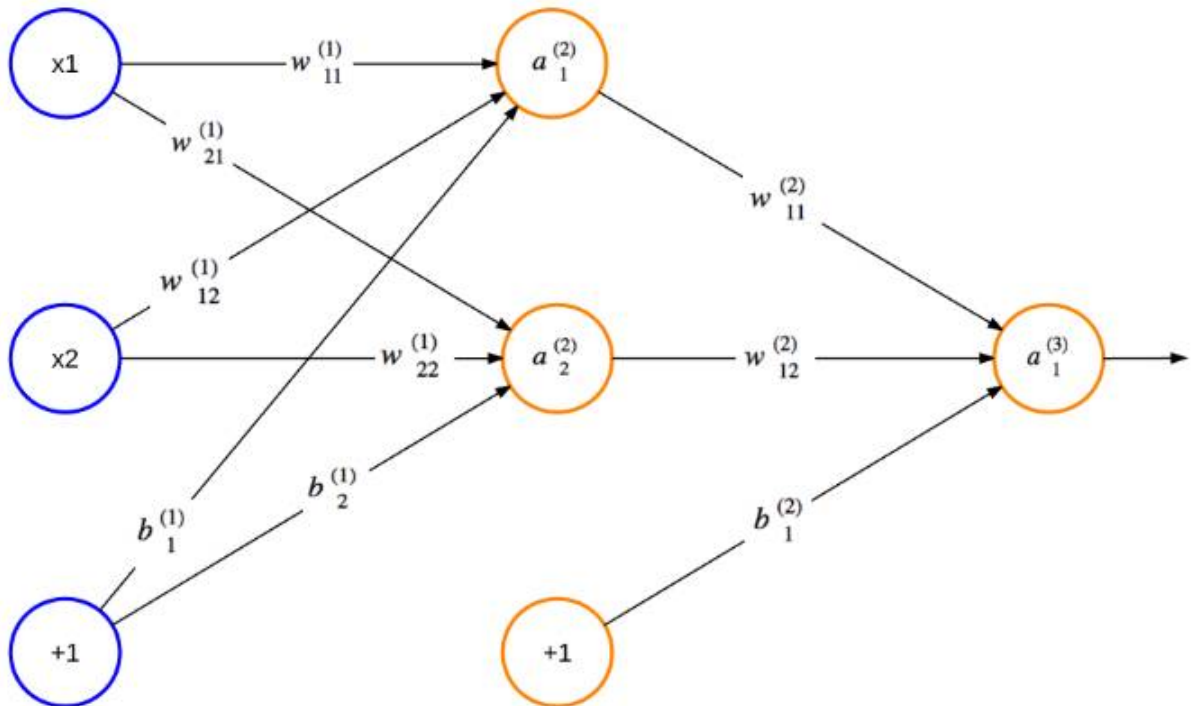


Figure 1: ANN structure

Result and Discussion(Forming words / Trimming)

Based on line and similar pixel values, the Hindi character are recognizing in this research. Scanned document / image / system's x and y axis defined below (fig.2):

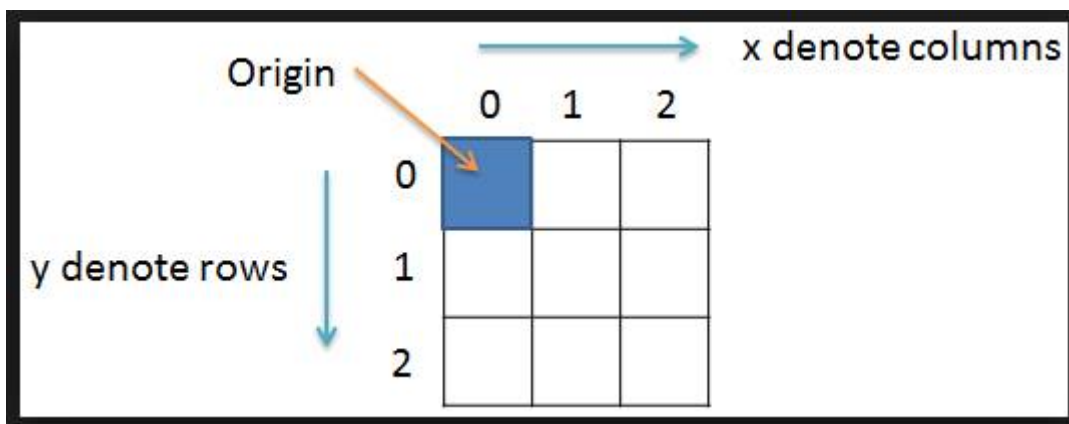


Figure 2: Pixel position in the screen

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

In the image there are x and y axis,

x-denotes column
y-denotes rows

There are two colors in the image

Sample:

Background –yellow

Foreground (text)-black

The tool that we have built will scan the image and monitor it by pixel by pixel. It neglects the image if it has yellow. It observes the black color and takes it as screen shot, when monitoring the image it avoids the yellow color between the letter and within the letters too. Hence the all the foreground images captured properly and print the letters into the database (fig.3). The tool what we created to extract the text can be called as TEXT EXTRACTION TOOL (TET). This tool developed by MATLAB.

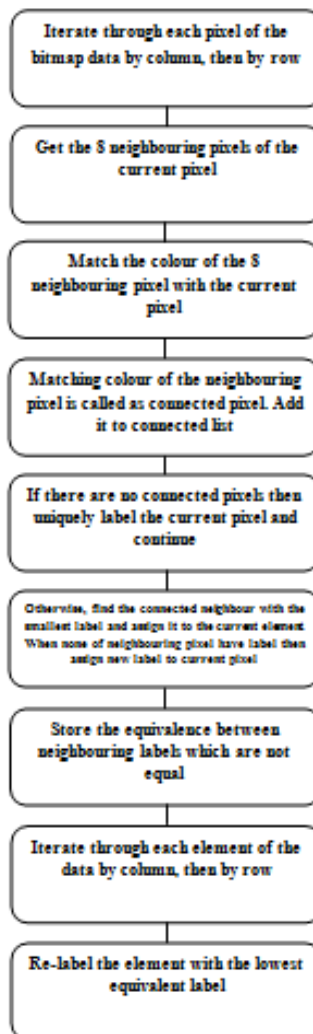


Figure 3: Steps for analysing exact pixel values

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Pre-Processing

The pre-processing stage yields a clean document in the sense that maximal shape information with minimal noise and maximal compression on normalized image is obtained. These details need to provide for the Tool before start the recognition process (fig.4). The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. Here used gray scale values for find the text occurred area.

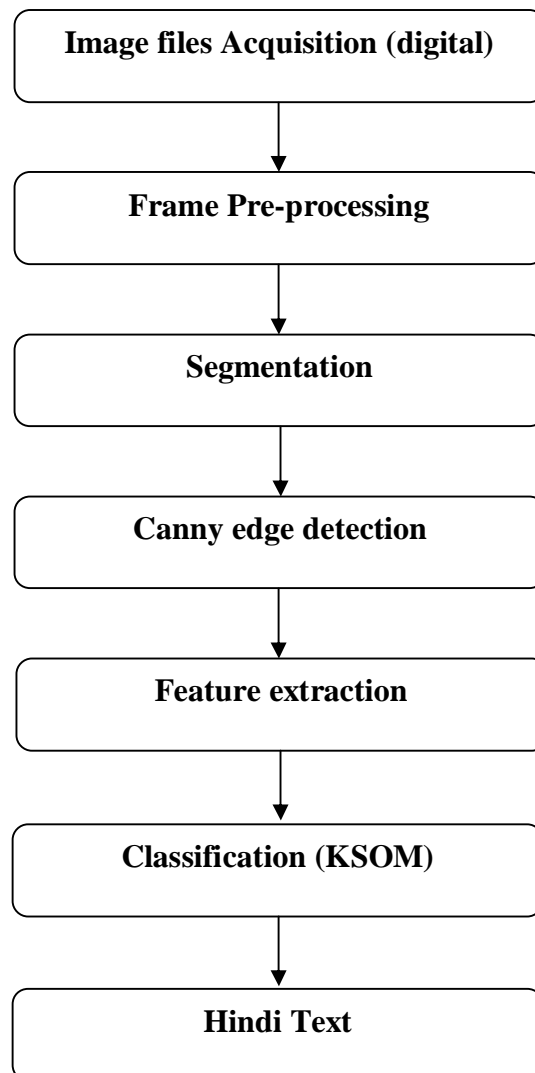


Figure4: Steps of Research Overview

Segmentation

First step of segmentation process is segmenting the text region into lines, also called as line segmentation. First we need to calculate header lines and base lines for the Line segmentation as shown in the figure 5a, 5b. Header lines are rows with maximum number of black pixels and base lines are rows with minimum number of black pixels.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Finding header line is a challenge because of skew in headline. Till now most of the researchers are detecting the header line by finding the row with maximum pixel density, but it cannot work for skew variable text.

O	O	O
O	O	O
O	O	O

(a)

O	O	O
O	3	O
O	O	O

(b)

Figure 5: Character with in that Matrix

This method gives good results for uniform and non-uniform skewed lines. But average line height is assumed to be 30 pixels. As different users handwritings are different, so average line height of users may be greater than 30 pixel or less than 30 pixels. Average line height of high resolution images is greater than low resolution images. If 450×540 image has average line height as 30 pixels, 900×1080 image average line height will be 60 pixels, 225×270 image average line height will be 15 pixels. In the above two situations this algorithm fails to detect the lines accurately.

The new algorithm can work for the images having different average line height. In this algorithm we need to find average line height using horizontal projection based method. Divide the Image into three equal halves (stripes), because the skew in entire line may be high, as compared to skew in the first half. Perform the following steps in first half.

- 1) Find out the rows with minimum number of pixels and replace that row pixels with black pixels .So that we can separate text lines with black rows.
- 2) Find out the height of the text lines using those black rows.
- 3) Store the heights of lines in array and use sorting technique to sort the elements and take median as average height of the lines.
- 4) By using average line height calculate minimum height of the consonant in a line. Minimum height of the consonant = Average line height / 4.

Word segmentation is easier than line segmentation and character segmentation. Space between two words is generally more than three pixels. Words are segmented by the projection based method. Algorithm for straitening header line of a word after performing thinning operation on binary image.


1. Identify the most visible line in first 50% area and consider it as expected header line.
2. Find out a point where expected header line and actual header line meets.
3. From that point trace the actual header line and find out the differences between actual header line and expected header line on both sides. a. $Diff_j = \text{expected header line} - \text{actual header line } j$, i.e., $j - \text{column}$
4. If $diff_j = 0$ column remains same.
5. If $diff_j > 0$ move the entire column upside according to $diff_j$ value.
6. If $diff_j < 0$ move the entire column downside according to $diff_j$ value.
7. With last five steps proposed step added for line break. Because our segmentation based on line.

International Journal of Innovative Research in Science, Engineering and Technology

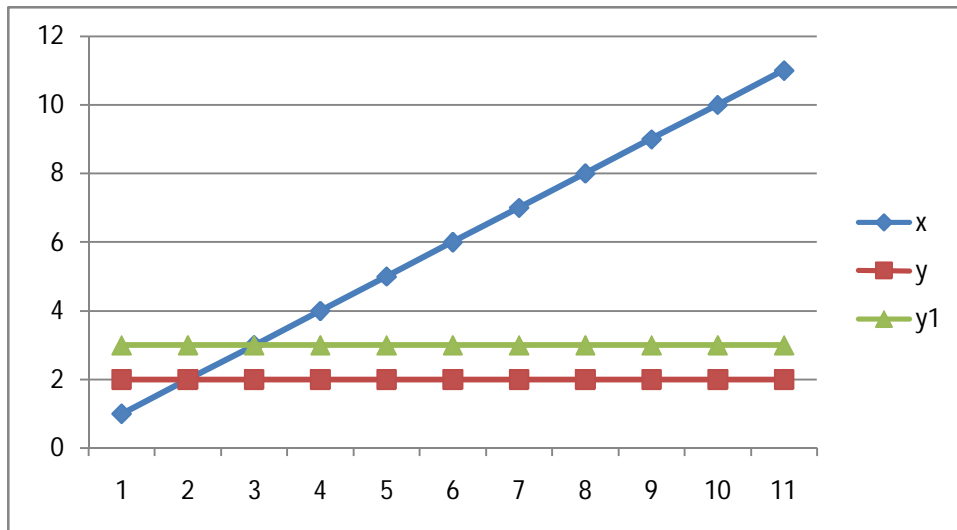
(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Following structure explain the segmentation of the image pixel area virtually,

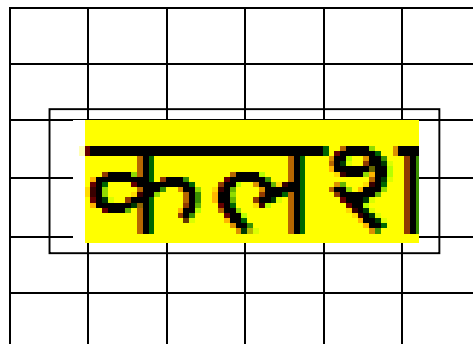
R/C	C1	C2 – C10	C11
R1	P(0,0)	P(1,0) to (10,0)	P(11,0)
R2	P(0,1)		P(11,1)
R3	P(0,2)		P(11,2)
R4	P(0,3)		P(11,3)
R5	P(0,4)	P(1,4) to (10,4)	P(11,4)

II. RESULT OF STRUCTURE



III. CANNY EDGE DETECTION

To separate individual words we first pass the entire image through a canny edge detector to make a logical image for further processing. Purpose of this filter is to reduce noise and recovery of information in extreme lighting conditions where normal threshold fails. We then remove holes and small area noise. The image is dilated to obtain clusters that could be declared as a word. Then we use these clusters as base values, to compute bounding box of each word and the words thus identified are separated for skew correction. This way we ensure that the words are separated.



International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Pixel wise Binary pattern 6 * 6

0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0
0	0	0	0	0	0

The steps of implementation is as following:

1. For every pixel F(j, k) calculate $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial y}$

A0	A1	A2
A7	F(j, k)	A3
A6	A5	A4

2. $\frac{\partial F}{\partial x} = \frac{1}{K+2} ((A2 + K * A3 + A4) - (A0 + K * A7 + A4));$

and

3. $\frac{\partial F}{\partial x} = \frac{1}{K+2} ((A0 + K * A1 + A2) - (A6 + K * A5 + A4));$

4. Calculate the $\sqrt{\left(\frac{\partial F}{\partial x}\right)^2 + \left(\frac{\partial F}{\partial y}\right)^2}$,

5. Exam the distribution of the gradient map, set a threshold and a percentage of pixels with greatest gradient value to be the edges.
6. The pixel with the gradient value greater than the threshold will be the edge, which means its gray value equals 0 in edge map; the pixel with the gradient value less than the threshold will not be the edge, which means its gray value equals 255 in edge map.
7. Output the binary image of the edge map.

IV. RESULT AFTER EDGE DETECTION:

कलश

Feature Extraction

After segmentation of the character, feature extraction like top and bottom, height, width, horizontal line, vertical line detection is done.

Using PSVM: Support Vector Machines (SVMs) suffer from a widely recognized scalability problem in both memory use and computational time. To improve scalability, we have developed a parallel SVM algorithm (PSVM), which reduces memory use through performing a row-based, approximate matrix factorization and which loads only essential data to each machine to perform parallel computation. Let n denote the number of training instances, p the reduced matrix dimension after factorization (p is significantly smaller than n) and m the number of machines. PSVM reduces the memory requirement from O(n²) to O(np=m), and improves computation time to O(n p²=m). Empirical studies

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

show PSVM to be effective. PSVM-Instead of a standard support vector machine (SVM) that classifies points by assigning them to one of two disjoint half- spaces, points are classified by assigning them to the closest of two parallel planes (in input or feature space) that are pushed apart as far as possible.

Kohonen’s Self-Organizing Maps

The goal of this algorithm is to create a correspondence between the input space of stimuli and the output space constituted of the codebook elements, the code words, or neurons. After learning, these last ones have to approximate the vectors in the input space in the best possible way. All neurons, or code words, are physically arranged on a square grid; it is thus possible to define k- neighbourhoods on the grid, which include all neurons whose distance (on the grid) from one (central) neuron is less or equal to k.

Each of the M code words is represented by its weight $X_j \in R^N$, where N is the dimension of the space ($= \tau * \tau$ in the compression scheme). For each presentation of an input vector $X \in R^N$ during the training phase, the index of the codeword nearest from X is determined, according to the Euclidean distance,

$$d(X, X_i) = \min(d(X, X_j)), 1 \leq j \leq M. \quad (1)$$

The selected neuron i, and all neurons in a k-neighbourhood of neuron i, are then “moved” in the direction of the input vector X, according to the relation,

$$X_m(t + 1) = X_m(t) + s(t)(X - X_m) \quad (2)$$

where represents the index of all neurons in the k-neighbourhood of neuron i and s(t) the learning factor. A(t) k and must decrease during the learning to ensure a good convergence of the algorithm. The topology-preserving property of KSOM, resulting from this learning process, means that two vectors in the input space, close with respect to the Euclidean distance, will activate two close centroids on the grid. KSOM’s have two properties used in our compression scheme. First, it quantizes the space like any other vector quantization method, what constitutes a first (lossy) compression of the image. Then, the topology preserving property of KSOM, coupled with the hypothesis that consecutive blocks in the image will often be similar, and to a differential entropic coder, constitutes a second compression of the information. Fig.6 denotes the images and its text.

IMAGE	TEXT
कल	कल
कलश	कलश
वकील	वकील
पकाना	पकाना
कैसे	कैसे
कुपित	कुपित
आज	आज
आजकल	आजकल
माता	माता
नमस्कार	नमस्कार

Figure 6: Text conversion

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

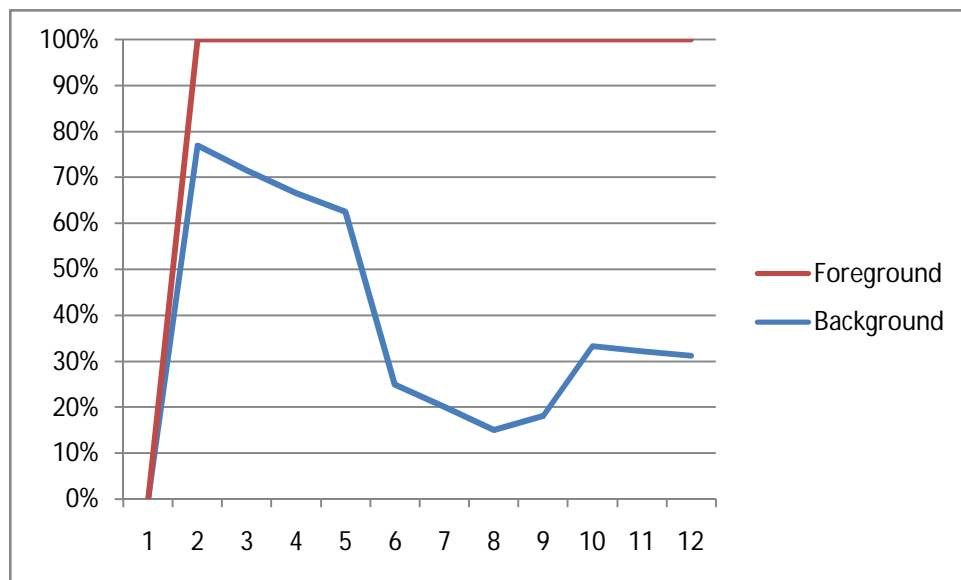


Figure 7: Graphical representation of Text conversion

V. CONCLUSION

In this paper an attempt has been made to develop a complete Hindi 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 $\pm 0.06^\circ$ about the actual value. Skew rotation is performed on a grey 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 neighbourhood. The overall recognition accuracy is around 98%.

REFERENCES

1. AbhishekSharma,VikramVerma, Handwritten Hindi Character Recognition, ISSN: 2278 – 909X International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 5, Issue 5, May 2016.
2. NishaVasudeva, Hem JyotsanaParashar and Singh Vijendra," Offline Character Recognition System Using Artificial Neural Network", International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012.
3. MehaMathur, Anil Saroliya, " HCR Using K-Means Clustering Algorithm", Volume 3.Issue 7, July 2014.
4. NeerajPratap and Dr.ShwetankArya ," A Review of Devnagari Character Recognition from Past to Future ",Volume 3, Issue 6, June 2012.
5. SonikaDogra and ChandraPrakash,"PEHCHAAN:HINDI HANDWRITTEN CHARACTER RECOGNITION SYSTEM BASED ON SVM", Vol. 4 No. 05 May 2012.
6. Sonal P Ajmire K. G. BagadeP.LRamteke," An Analytical Study of Handwritten Devanagari Character Recognition", Volume 5, Issue 10, October-2015.
7. J.Pradeep ,E.Srinivasan and S.Himavathi, " DIAGONAL BASED FEATURE EXTRACTION FOR HANDWRITTEN ALPHABETS RECOGNITION SYSTEM USING NEURAL NETWORK", Vol, 3, No 1, Feb 2011.
8. KanakUpmanyu, Mr.ShahidHussain and Dr.Rizwan Beg," Handwritten character recognition system with Devanagari script (SWARS)", Volume 2, Issue 9, September 2014.
9. Dr. N.Rajalingam and K.Ranjini,"Hierarchical Clustering Algorithm – A Comparative Study", Volume 19– No.3, April 2011.