International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10539-10550
 https://dx.doi.org/10.37622/IJAER/13.12.2018.10539-10550

Effective Thinning Algorithm for Recognition of Hand Written Devnagri Compound Characters Using Neural Network

Prashant Yawalkar [1)], Dr. M. U. Kharat [2)]

[1]Research Scholar, Department of Computer Engineering, MET's Institute of Engineering, BKC Nasik, India.

[2]Professor and Head, Department of Computer Engineering, MET's Institute of Engineering, BKC Nasik, India.

Abstract

Handwriting always works as a primary tool in the world of communication. Various techniques have been and are developed in order to document the handwritten text. There are certain exceptional techniques that describe the environment of handwritten scripts and further convert it into electronic data by implementing various algorithms. Devnagri is widely used for most popular and commonly used languages like Marathi and Hindi We have proposed a neural network based hand written Devnagri compound character recognition system using a novel rule-based thinning algorithm that improves the overall performance of the system. The distinctive feature of our thinning algorithm is that it is a rotation invariant method which thins characters to their central lines, helping in preserving shape of the character. The system applies different rules concurrently to each pixel in the image which results into symmetrical thinning and improvement in the overall speed of the system. Results obtained for the system prove that the system is efficient enough to preserve the topology of the compound characters written in Devnagari that further helps in improving the accuracy of the recognition system.

Keywords: Character recognition, Neural Network, Normalization, Rotation invariant, Compound character, Thinning, Feed forward, Boundary detection, Image acquisition, Segmentation, Feature extraction.

INTRODUCTION

If handwritten characters are precise and uncontaminated, they can be recognized by humans accurately. But if the same task is given to machines, then it becomes difficult for them to do it. Different languages use different scripts to write, one of them is Devnagri which is widely used for most commonly used languages like Marathi and Hindi. Presence of complicated curves and various shapes increases the complexity of recognition for these languages. All these features make Optical Character recognition (OCR) for Devnagri script specifically hard [1], [4].

Optical character recognition [5],[32][35] can be termed as the motorized or else electronic conversion of scan imagery of handwritten, typewritten or printed wording interested in system-encoded content. This process can be used for converting books and documents into electronics form, managing a record-keeping system in an office or publishing the text on a website. Using OCR we can also modify the text, search for a phrase, store it efficiently, display a replica free of scanning artifacts, as well as use processes like machine translation, text-to-speech and text mining to it.

DEVANAGARI SCRIPT

Devanagari script is unique in a number of ways as compared to other scripts. It has two-dimensional compositions of symbols: foundation characters in the central strip as well as optional modifiers. Two characters possibly will be in shadow of every one. Whereas line segments [31] are the main features used for English, the majority of the characters inside Devnagri script are formed through curves, holes as well as strokes. In Devnagri language script, the formation of uppercase and lower-case characters is not available, but it has more number of symbols than that of English. Marathi is an Indo-Aryan language mostly spoken by millions of peoples in the western and central India [8]. Marathi is one of the Prakrit languages developed from Sanskrit and is also the official language in Maharashtra state. Typically, the handwriting manner changes from person to person as well as signature also vary from person to person [30].Having a large character set with cursive also [16], [22], curves as well as lines are in the particular shape formation, which may be over lapping in a word. As per individual writing style a touching characters can touch each other at different position [19-21, 24 29].. The set of vowels and consonants in Devanagari scripts are as follows [9], [10]. A considerable measure of exploration is still required for compound character, word, sentence and document recognition, its semantics and lexicon. Different techniques for treating the matter of Devnagari character recognition have been created [7], [11], [15], [44].

Vowels	अ आ इ ई उ ऊ ऋ ए ऐ ओ औ अं अः
Consonents	क ख ग घ ङ ष च छ ज झ ञ स ट ठ ड ढ ण ह त थ द ध न क्ष प फ ब भ म त्र य र ल व श ज्ञ

Figure 1. Vowels and Consonants

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10539-10550
© Research India Publications. https://dx.doi.org/10.37622/IJAER/13.12.2018.10539-10550

FEATURES OF THE DEVANAGARI CHARACTERS:

1. Characters having End bar: Characters ending with a | are end bar characters. Eg.
अ ख घ च ज झ ञ त थ ध न प ब भ म य ल व श स
2. Characters having No bar: Characters that do not have | are non bar characters. Eg.
इ उ ऊ ए छ ट ठ द ड ढ र ह
3. Characters having Middle bar: Characters having | in middle are middle bar characters. Eg. ऋ क फ

A Devanagari character can be either an End bar character, Non bar character or Middle bar character as specified above. Further these characters can be combined to get various compound characters termed as Jod-Akshars. Recognition of such compound characters requires rigorous training and extraction of multiple features so as to get accurate results.

BASIC STAGES IN CHARACTER RECOGNITION

In some hand-writing, the characters are indistinguishable even to the human eye and that they can only be distinguished by context. In order to distinguish between such similar characters, the tiny differences that they have must be identified. One of the major problem of doing this for hand written characters is that they do not appear at the same relative location of the letter due to the different proportions in which characters are written by different writers of the language. Even the same person may not always write the same letter with the same proportions. Fig 2 shows the basic stages in hand written character recognition system. Conversion of paper document in to digital form through process of scanning is called **image acquisition**.

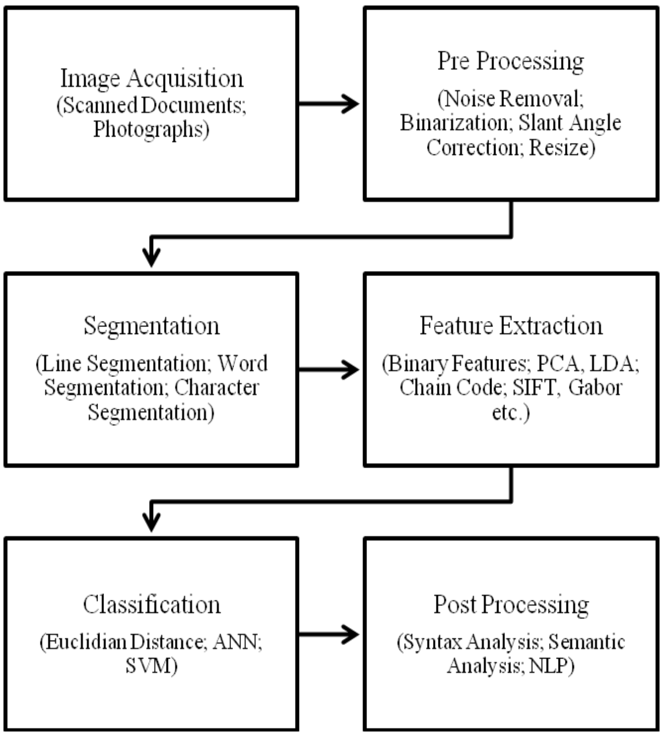

Figure 2. Basic stages of character recognition

The problems associated with word spotting and word recognition on images were addressed by Jon Almaz et al. [14]. N Vasudeva et al. suggested a method with more appropriate results in comparision of matching with normalized word-image [40]. P**reprocessing** focuses on deduction of the noise resulting due to the scanner quality used for capturing the image. In order to make the further recognition process smooth apart from noise reduction, other processing activities like thinning, normalization and segmentation [19], [27] of image are also carried out. Normalization involves resizing of characters for stroke width, slant, slope, height of the characters. Further trimming down each character image to vertical letters of uniform height made up of one pixel-wide stroke is carried out [2], [26]. The most critical process is segmentation that decides the achievement of character recognition technique. In **segmentation** process decomposing an image of a sequence of characters into sub images of individual symbols by segmenting lines and words is performed. Extracting features from the input image in the sense of information that must be similar for similar images and distinct for other images [45], [46]. Selection of **feature extraction** technique becomes a key aspect in achieving high recognition performance since these features play major role in pattern recognition. Some of the feature extraction methods like projection, bordered transition, zoning etc are carried out. In projection method, data is compressed by means of projection where in marginal distributions are generated through counts of black pixels taken along parallel lines of the image. Halftoning technique gives best compression factor 12.5% at the source [41]. All the characters are assumed to be oriented vertically by partitioning each character into four equal quadrants in Border transition technique. Zoning is a process that involves the division of the character into smaller fragment of areas. In the process of **Classification,** features of the image are used as the basis for assigning data to their corresponding class with respect to groups having homogenous characteristics. Thus decision rules are used for dividing the feature space into several classes. Classification techniques like neural network, support vector machine, Genetic Algorithms, Fuzzy Logic etc are used for recognition of handwritten character [39],[43],[47]. The symbols are grouped in to strings as a part of **Post processing** activity**.** In order to increase the accuracy of optical character recognition, the output may be constrained by a list of words that are allowed to occur in a document. The output stream may be plain text or file of characters [3].

THINNING

To identify the geometrical feature of objects thinning is commonly used method. For example, using the thinned outcome the tree structure of the bronchus can be determined. For getting superior causes of cancer or brain tumor an image skeletonization can also be used on medical images. Gradient and watershed are used for skeletonization of a binary image. High intensity values can be avoided by performing skeletonization of multiple images and implementing the morphological dilation operator with thin parameter to obtain the output image. Maher Ahmed and Rabab Ward elaborated

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10539-10550
 https://dx.doi.org/10.37622/IJAER/13.12.2018.10539-10550

various thinning algorithms used for character recognition [18].

Thinning operation is used to reduce binary valued image regions in image processing and to reduce lines that approximate the center skeletons of the regions. In order to infer shape and topology in the resultant image it is necessary to have thinned outcome connected with every individual image. Thinning is often used in the preprocessing stage in order to facilitate higher stage study and recognition in various applications like: OCR, diagram understanding, fingerprint analysis and feature detection for computer dream [13].The outline of a binary image is a key demonstration for the shape examination and is of use for a lot of pattern recognition applications. The outline of an entity is a line connecting point's midway among the boundaries [12]. Thinning has been applied in various fields like: pattern recognition, image coding, automated industrial inspection, and biological form explanation etc. Thinning performs the key task of enhancing efficiency and reducing transmission time. Various thinning algorithms have been presented broadly classifying them into two categories as sequential thinning algorithms and parallel thinning algorithms.

Pertinent features and objects of curiosity are determined in order to complete recognition of image. Conversion of binary shapes obtained from edge detection or thresholding to 1-pixel wide lines takes place for better representation and effective processing of hand written or printed Devanagari characters [6] as shown in Fig 3 below.

TYPES OF THINNING

1. Iterative thinning: - **A** pixel by pixel based approach that examines the pixels until the result is obtained. It mainly divides into two parts Parallel and sequential. Sequential thinning takes place in predetermined order in which processing takes place in fixed sequence. In parallel thinning processing is in dynamic order. The main difference between these two is: sequential depends upon previous iteration result and also all the iterations done till now. Where as in parallel thinning only the result that remains after the previous iteration is taken into consideration.

2.

Figure 3. Preprocessing of Images (a) Original Image (b) Segmented Image (c) Elimination of Shirorekha (d) Thinned Image (e) Image Edging

3. Directional approach: - **T**hinning is done with respect to the directions that are north, south, east west. In these points belonging in the same side are removed parallelly.

4. Sub-field approach: - **I**mage is sub divided into parts according to some criteria.There is no fixed criteria for dividing the image, but some parity criteria may be applied. All the pixels in one direction are removed at one time.

5. Fully parallel approach: - In these neighbourhood criteria is used. The image can be divided in any format either in k*k or n*k Example: 3*3 or 4*3 or 3*4.

6. Non-Iterative Algorithm: - It is just opposite to iterative as it does not works on the pixel to pixel examine. There are some popular methods like medical transforms, distance transforms and other on which implementation is based.

7. Medical Transforms: - **R**esults of this method contains noise in data as it works on the technique of grey level images where intensity is represented in the terms of distance to the boundary.

8. Central line: - It takes the minimum time to process as it takes a center line or median for processing and thinning is done in one process or at one time. The distance to the boundary is considered to know whether the point should be considered in the skeleton or not.

9. Rotation Invariant 4-step thinning algorithm: - A 4-step thinning algorithm is iterative in nature that focuses on removing every point lying on the exterior boundaries of the symbol, for the symbols having width greater than one pixel. Region points are assumed as 1 and background points to have value 0. This method makes improvisation in the 2-step method by making use of 8-neighborhood method (**Fig 4)** in 4-step instead of 2-step for thinning the character in the given image of the pixel

P1	P2	P3
P8	P0	P4
P7	P6	P5

Figure 4. Eight Neighborhood of the Pixel

The focus is to calculate the non-zero neighbors of the central pixel P0 which can be calculated as

$$Nz\ (P0) = P1 + P2 + P3 + P4 + P5 + P6 + P7 + P8.$$

Transition say, X(P0) is calculated as the number of 0-1 transition in the order sequence P1-P8 respectively. Find the count of nonzero neighbors and transition from 0-1 using 8-neighborhood pixel method and complete the iteration by performing 4 steps. In every iteration removal in all 4-direction i.e., P2, P4, P6, P8 is done repeatedly until no further changes occur. The 4-steps of the algorithm are as shown:

Step 1: If the non-zero neighbors for pixel P0 (Nz (P0)) is greater than 1 and less than 9, If transition for pixel P0 i.e. X(P0) = 1, If P2 = 0 and P3 = 0 then make P(0) =0.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10539-10550
© Research India Publications. https://dx.doi.org/10.37622/IJAER/13.12.2018.10539-10550

Step 2: If the non-zero neighbors for pixel P0 (Nz (P0)) is greater than 1 and less than 9, If transition for pixel P0 i.e. X(P0) = 1, If P4 = 0 and P5 = 0 then make P(0) =0.

Step 3: If the non-zero neighbors for pixel P0 (Nz (P0)) is greater than 1 and less than 9, If transition for pixel P0 i.e. X(P0) = 1, If P6 = 0 and P7 = 0 then make P(0) =0.

Step 4: If the non-zero neighbors for pixel P0 (Nz (P0)) is greater than 1 and less than 9, If transition for pixel P0 i.e. X(P0) = 1, If P8 = 0 and P1 = 0 then make P(0) =0.

Step 5: Repeat the above mentioned steps for all the points in the region and stop the processing.

This algorithm shows improvement in accuracy due to shape preservance, also it is rotation invariant. However noise in input characters affects the overall performance of the system. Also increase in thickness of the input characters increases the time complexity. Further we also observed that for non-continuous characters accuracy decreases.

ROTATION INVARIANT ENHANCED THINNING ALGORITHM

Extending the 4-step thinning algorithm for construction of rules (given in Table 1) that are concurrently applied to each pixel iteratively in order to remove every point lying on the exterior boundaries of the symbol. This process is continued for all the regions of the symbol having width greater than one pixel until there are no further changes. If the Pixel has width equal to one pixel measured in any direction then it is not removed since it belongs to symbols central line in the resultant graph.

Table 1. Rules for Thinning

IF										Then									
Rule	**P1**	**P2**	**P3**	**P8**	**P0**	**P4**	**P7**	**P6**	**P5**	**Rule**	**P1**	**P2**	**P3**	**P8**	**P0**	**P4**	**P7**	**P6**	**P5**
1	1	X	0	1	1	0	1	1	X	**1**	1	X	0	1	0	0	1	1	X
2	1	1	X	1	1	0	1	X	0	**2**	1	1	X	1	0	0	1	X	0
3	1	1	1	X	1	1	0	0	X	**3**	1	1	1	X	0	1	0	0	X
4	1	1	1	1	1	X	X	0	0	**4**	1	1	1	1	0	X	X	0	0
5	1	X	0	1	1	0	X	0	0	**5**	1	X	0	1	0	0	X	0	0
6	1	1	X	X	1	0	0	0	0	**6**	1	1	X	X	0	0	0	0	0
7	1	1	1	1	1	0	1	1	1	**7**	1	1	1	1	0	0	1	1	1
8	1	1	1	1	1	1	1	0	1	**8**	1	1	1	1	0	1	1	0	1
9	X	0	0	1	1	0	1	X	0	**9**	X	0	0	1	0	0	1	X	0
10	0	0	0	X	1	0	1	1	X	**10**	0	0	0	X	0	0	1	1	X
11	X	1	1	0	1	X	0	0	0	**11**	X	1	1	0	0	X	0	0	0
12	0	X	1	0	1	1	0	0	X	**12**	0	X	1	0	0	1	0	0	X
13	0	0	0	0	1	X	X	1	1	**13**	0	0	0	0	0	X	X	1	1
14	0	0	X	0	1	1	0	X	1	**14**	0	0	X	0	0	1	0	X	1
15	1	1	1	0	1	1	1	1	1	**15**	1	1	1	0	0	1	1	1	1
16	1	0	1	1	1	1	1	1	1	**16**	1	0	1	1	0	1	1	1	1
17	0	X	1	0	1	1	X	1	1	**17**	0	X	1	0	0	1	X	1	1
18	X	1	1	0	1	1	0	X	1	**18**	X	1	1	0	0	1	0	X	1
19	0	0	X	X	1	1	1	1	1	**19**	0	0	X	X	0	1	1	1	1
20	X	0	0	1	1	X	1	1	1	**20**	X	0	0	1	0	X	1	1	1

Which ever region of the symbol has width 2 pixels wide; it indicates that the central line passes through those 2 pixels. Consider a graph of white and black pixels, where white pixels denoted by 0s specify the background, while the black pixels denoted by 1s specify the symbol consisting of an isolated symbol which is considered to be a connected graph. Further it assumes that the symbol's central lines are also joined. Some symbols like the character क have only one central line; while characters like ज or झ have multiple central lines which are connected at multiple points. In the neighborhood of 3 X 3 pixels if a black pixel with a single neighbor having black color; then the boundary of central line includes the middle pixel. During each iteration the focus is on removing all the pixels that are on the exterior boundaries of the resultant graph. However, the pixel should not be removed if its removal results in to a disconnected graph. Different classes are obtained for eight neighbor pixels pattern with central pixel as a black pixel.

CLASS A: As all the neighboring pixels are white, we should not remove the middle black pixel which is a dot symbol and its removal may lead to removal of the whole symbol.

CLASS B: As all the neighboring pixels are black, we should not remove the middle black pixel as it does not belong to the boundaries of the symbol.

CLASS C: Considering all the neighboring pixels in a clockwise direction we get one to seven continuos white pixels, followed by seven to one continuos black pixels. The central pixel should be removed as it belongs to the boundary of the symbol with an exception for the four cases mentioned in **Table 2.**

Table 2. Four possible cases

Case	P1	P2	P3	P8	P0	P4	P7	P6	P5
1	0	1	1	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	0
4	1	1	1	1	1	1	0	1	1

This class has three subclasses:

i. We should not remove the middle pixel which is on the edge of a straight line having a length of minimum 2 pixels, since the neighboring pixels consist of seven white pixels and only one black pixel.

ii. We should remove the middle pixel as it is on the boundary of the symbol with the neighboring pixels consisting of exactly two consecutive black, and six consecutive white pixels. How ever we should not remove the middle pixel as it belongs to an edge of a crisscross slanting line.

iii. We should remove the middle pixel as it is on the boundary of the symbol with the neighboring pixels consisting of minimum three consecutive black pixels followed by a white pixel.

CLASS D: When we consider the neighboring pixels in a clockwise direction the white pixels do not occur in a continuos manner. It further consists of three subclasses:

i. Among the surrounding circular pixels, separation of two white pixels with one black pixel on one side and 5 pixels (at least one of them black) on the other side; gives rise to two subcases:

a. One white pixel below or above the central pixel, with the other on its left or right (as in fig 5 (a)), with at least one x (do not care) black; in such a case removal of the pixel will result into a disconnected graph since the central pixel must belong to a central line (as in fig 5 (b)).

1	0	X	1	0	X	z	a	0	z	0	0
0	1	X	0	0	X	w	1	1	w	0	1
X	X	X	X	X	X	1	y	0	1	0	0
(a)			(b)			(c)			(d)		
z	1	0	z	1	0	z	1	0	z	1	0
w	0	1	w		b	w	0	1	w	1	1
1	1	0	1	0	0	1	0	0	1	0	0
(e)			(f)			(g)			(h)		
r	1	1	w	0	1	r	1	1	w	0	1
z	1	0	t	1	1	z	1	a	t	1	1
1	w	X	z	b	0	t	w	X	z	b	0
(i)			(j)			(k)			(l)		
0	1	1	0	1		0	1		0		1
d	1	1	1	0		1	1		1	1	
X	c	0	0	0	0			0	1	1	0
(m)			(n)			(o)			(p)		

Figure 5. Different conditions for sub classes in class D

b. One white pixel lying on one corner and the other on an adjoining corner of the neighborhood as in fig 5 (c), here if a = y = 0 or a = y = 1, then the central pixel should not be removed since its removal may result in to a disconnected graph as in fig 5(d) or as in fig 5 (e).

z	a	1
w	1	b
0	y	0

If a or y = 1 then without losing the generality, assuming a as 1 and y as 0, as in fig 5 (f) where b with value 1 must belong to a central line, since it has vertical width as 1. Two paths can be obtained as an acceptable solution for the central line as it is continuous, given by fig 5 (g) and fig 5 (h) respectively. We prefer the latter solution and so do not remove the middle pixel.

ii. If we have two white pixels separated by two black pixels on one side and four pixels on the other side with atleast one of them black as in fig 5 (i) or (j) then, among the white pixels and through one of the black pixels a central line must be passing. For the different possible values of r, z, t, and w, either we should not remove the middle pixel since it results in a disconnected graph; or we can obtain multiple acceptable paths for the central line. In the latter case, minimum one of the satisfactory solution crosses through the central pixel; hence, in all of these cases we should not remove the middle pixel.

iii. If we have two white pixels separated by three black pixels on one side and three pixels on the other side with atleast one of them black as in fig 5 (k) or (l) then, the two white pixels lie:

a) Directly below and above.
b) To the left and to the right of the central pixel.
c) Recline on two corners.

In case a and case b the middle pixel must belong to a central line and should not be removed. In case c as in fig 5 (m) there must be a central line passing among these two white pixels. Possibly we can get seven different types as shown in **Table 3.** The acceptable solutions for the first type are as in fig 5 (n) or (o) or (p). We can further show that without removing the middle pixel a central line still exist, for all the other types and so we do not remove the middle pixel.

Table 3. Seven possible types for the central line passing through the white pixels

Case	P1	P2	P3	P8	P0	P4	P7	P6	P5
1	0	1	1	1	1	1	0	0	0
2	0	1	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0	0
4	0	1	1	1	1	1	0	1	0
5	0	1	1	0	1	1	1	1	0
6	0	1	1	0	1	1	1	0	0
7	0	1	1	0	1	1	0	1	0

If we remove only those pixels that come under case ii and iii as specified above then we can show that only the pixels belonging to the boundary of the symbol are removed while pixels belonging to the central lines are excluded. Repeatation of this process until there are no further changes results into generating the central lines. The above process fails where the width of the symbol is 2 or even, since in such case the above process removes both the pixels as one of them belongs to the boundary. In order to avoid such situations it is necessary to check whether the symbol has width 2 pixel wide.

DISCONNECTED CENTRAL LINES

When the rules specified inTtable1 are concurrently applied to each pixel the input image results into as described by the example shown in **Fig. 6a.** Applying all the rules concurrently to each pixel in the image peels off the inner and outer boundaries of the symbols. Few lines of even width are unfortunately deleted in the above process. However, the rotation invariant nature of the process rotates the thinned pattern by same angle by which the symbol is rotated. The above procedure results into sporadic central lines which are considered as the major drawback. If the part of a symbol having width of two pixels in the vertical or horizontal direction is not deleted then a resultant image as shown in **Fig 6b** is obtained. This results into continuous central lines having

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10539-10550
© Research India Publications. https://dx.doi.org/10.37622/IJAER/13.12.2018.10539-10550

many extraneous pixels which are caused due to non removal of the two pixels width cases. In order to overcome this problem we suggest a solution mentioned below whose results are as shown in **Fig**.6b.

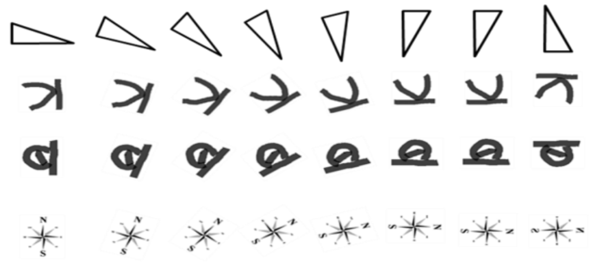

Figure 6a. Input Characters

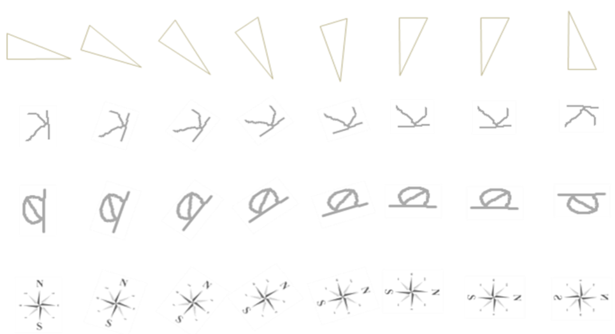

Figure 6b. Results after Thinning

ONE PIXEL WIDE CENTRAL CONNECTED LINES

Initially we need to determine under what conditions the deletion of two pixel wide symbol takes place, and then later find a solution to ovecome this problem.

Consider a segment having its central pixel as 2 pixels wide in the vertical direction as in fig 7a where z = y = 1, along with its neighbors as in fig 7b. We need to determine that z and y will be removed for what values of a, b, c, d, k, l, m, and n.

Attributes k, l, m, and n in the last column may take one of 16 possible values. Further we calculate the values of a, b, c and d that result in the removal of z or y, or both z and y, after applying the thinning rules to each of these values. Both z and y are removed for the configurations shown in fig 7c (for rows 6, 7, 14, and 15 of Table 4) and as shown in fig 7d (for row 0). The former case is undesirable, while the later is desirable where in z and y should be deleted since both of them lie on the boundary.

The configurations shown in fig 7e or 7f leads to exceptional case of z forming an extremity of a crisscross diagonal line and hence should not be removed. Likewise, if y has the configuration as shown in fig 7g or 7g' then it should also not be removed. A solution to this problem of removing two pixels-wide lines and removing the extremities of crisscross diagonal lines, a extended procedure that performs the steps given below on each pixel in the document during every iteration. Verify whether the pixel has two pixels width in which direction; vertical or horizontal. Apply the thinning rules if it does not have the two pixels width, else check the following:

Situation 1: Check whether the pixel belongs to one of the circumstances as shown in fig 7h or 7i. Remove y if it belongs, since the central line must be passing through z or y by assuming that it passes through z. If it does not; then check for Situation 2.

Situation 2: Check whether the pixel belong to an extremity of a crisscross diagonal line. Stop further calculations for this pixel if it belongs, else apply the thinning rules.

(a)

0
z
y
0

(b)

a	0	k
b	z	l
c	y	m
d	0	n

(c)

x	0	x
1	z	1
1	y	1
x	0	x

(d)

x	0	0
1	z	0
1	y	0
x	0	0

(e)

0	0	0
0	z	0
1	1	0
1	0	0

(f)

0	0	0
0	z	0
0	1	1
0	0	1

(g)

1	0	0
1	1	0
0	y	0
0	0	0

(g')

1	1	1
1	0	0
0	0	0

(h)

x	0	x
1	z	1
1	y	1
x	0	x

(i)

x	1	1	x
0	z	y	0
x	1	1	x

(j)

x	0	x
1	w	1
1	1	1
x	0	x

(k)

x	0	x
1	1	1
1	w	1
x	0	x

(l)

x	1	1	x
0	w	1	0
x	1	1	x

(m)

x	1	1	x
0	1	w	0
x	1	1	x

(n)

0	0	0
0	w	0
1	1	0
1	0	0

(o)

0	0	0
0	w	0
0	1	1
0	0	1

(p)

1	0	0
1	1	0
0	w	0
0	0	0

(q)

0	0	1
0	1	1
0	w	0
0	0	0

(r)

0	0	0	0
0	w	1	0
0	0	1	1

(s)

0	0	1	1
0	1	w	0
0	0	0	0

(t)

0	0	0	0
0	1	w	0
1	1	0	0

(u)

1	1	0	0
0	1	w	0
0	0	0	0

Figure 7. Conditions for different stages in the algorithm.

Algorithmic steps:

Let Pat_{11} to Pat_{22} be the different patterns represented using Fig 7(a) to 7(u) respectively

Repeat the following until no further changes occur;

For every iteration perform the following:

For every pixel P in the document D perform the following:

1. If (P belongs to two pixels wide in the vertical direction) then

    If ($P \in Pat_{11}$)

    then

    Stop further computations for P;

Else
If (P Є Pat_{12})
then
Remove P and stop further computations for P;
Else
If (P Є Pat_{13})
then
Stop further computations for P;
Else
If (P Є Pat_{14})
then
Remove P and stop further computations for P;
If (P Є (Pat_{15} ‖ Pat_{16} ‖ Pat_{17} ‖ Pat_{18}))
then
Stop further computations for P;
Else
Apply all the Thinning rules and stop further computations for P;

2. Else
If (P belongs to two pixels wide in the horizontal direction) then
If (P Є (Pat_{19} ‖ Pat_{20}))
then
Stop computations for P;
Else
If (P Є (Pat_{21} ‖ Pat_{22}))
then
Stop computations for P;
Else
Apply all the thinning rules and stop further computations for P.

Applying this modified procedure, the symbols get thinned to their central lines which are of one pixel width and without extraneous branches as shown in **Fig.6b.**

RECOGNITION SYSTEM

During the last 2 decades variety of character recognition systems using neural network that can recognize offline characters in English have been implemented. Further lots of improvement interms of recogintion efficiency can also be seen[17],[36],[37]. Later systems that can recognize online characters were also developed[25],[33],[34]. We have designed and implemented a artificial neural network based color and size invariant character recognition system which can recognize Devnagari characters अ to ज्ञ as well as Devnagari Compound characters like क्ष and jodakshars. A two layer feed-forward network trained in supervise manner with an input-output layer and no hidden layer is used. Recognition system works in four basic steps: Preprocessing, Normalized character matrix creation, Network establishment and Recognition. Preprocessing includes: digitization, noise removal, thinning and boundary detection of the digitized character matrix. For recognition of characters by computer the character image is initially digitized into a matrix i.e. transformed from color image to the gray scale image and then into a binary form for the ease of handling by the computer. In order to convert gray scale image into binary form we replace all the gray levels in the range 129-255 to binary value 0 and all the gray levels in the range 0-128 to binary value 1 treating it as absence or presence of writing respectively.

Table 4. Conditions for deletion of y and z

No	k	l	m	n	Condition for Deletion of z	Condition for Deletion of y	Condition for Deletion of z and y
0	0	0	0	0	(a=x, b=c=1, d=x) or (a=b=0, c=1, d=x)	(d=x, b=c=1, a=x) or (b=1, c=d=0, a=x)	(a=x, d=x) and (b=c=1)
1	0	0	0	1		Never	Never
2	0	0	1	0			
3	0	0	1	1			
4	0	1	0	0	Never	(d=x, b=c=1, a=x) Or (b=1, c=d=0, a=x)	Never
6	0	1	1	0	(a=x, b=c=1, d=x) Or (a=b=0, c=1, d=x)		(a=x, d=x) And (b=c=1)
7	0	1	1	1			
8	1	0	0	0	Never		Never
12	1	1	0	0	Never		Never
14	1	1	1	0	(a=x, b=c=1, d=x) or (a=b=0, c=1, d=x)		(a=x, d=x) And (b=c=1)
15	1	1	1	1			

BOUNDARY DETECTION

Once the digitized binary matrix is created from the input character image, it becomes essential to detect the boundary in order to correctly recognize the input character. The boundary detection process involves the following steps:

i. In order to detect the top boundary the character matrix is scanned from top-left corner by removing all rows having only 0's. It is necessary to have at least two consecutive 1's in two consecutive rows for detecting top boundary and further selecting the top boundary as the first row from the two consecutive rows.

ii. In order to detect the bottom boundary the character matrix is scanned from bottom-left corner by removing all rows having only 0's. It is necessary to have at least two consecutive 1's in two consecutive rows for detecting bottom boundary and further selecting the bottom boundary as the first row from the two consecutive rows.

iii. In order to detect the left boundary the character matrix is scanned from top-left corner by removing all columns having only 0's. It is necessary to have at least two consecutive 1's in two consecutive columns for detecting left boundary and further selecting the left boundary as the first column from the two consecutive columns.

iv. In order to detect the right boundary the character matrix is scanned from top-right corner by removing all columns having only 0's. It is necessary to have at least two consecutive 1's in two consecutive columns for detecting right boundary and further selecting the right boundary as the first column from the two consecutive columns.

NORMALIZATION

In the process of Normalization we equate the size of all binary arrays. To deal with input character of variant size the input character is converted into 12×8 normalized matrix. The normalization process includes the following steps:

i. Assuming that the top row and left column consist of salient features, they are considered for processing.
ii. Keep on considering alternate row and column and deleting few until desired matrix of size 12×8 is obtained.

The system is trained using the procedure as follows:

i. A random sample of any character for training is considered and then a 12×8 matrix is generated.
ii. Further the corresponding initial weight_matrix is generated by replacing 3 for 1 and -3 for 0 in the input matrix.
iii. Compute the weighted sum Oi(net activation) as follows:

$$Oi = \sum_{j=1}^{96} Wji\ Xj$$

Where, i=0, 1, 2……..45.

iv. Calculate the sum of positive weight, Pwi of the weight_matrix.
v. Calculate Yi=f (Oi) =Oi/Pwi.

Where, Oi →Net activation for each character i (e.g.अ, आ,…, ज्ञ).

Pwi→ the sum of positive weights of the weight_matrix for each character.

vi. Pick the maximum Yi.
vii. Check if the corresponding neuron fires. If neuron fires then save the weight matrix into a file.
viii. Check with other training samples of the same character and update the weight described as above.
ix. Fixed the weight matrix and save into a file after the final training samples for that character.
x. Repeat steps 1 to 9 for any other character.
xi. Completion of training establishes the network

The testing procedure of our proposed system is as follows:

i. For a random sample of any character generate a 12×8 matrix.
ii. Compute the weighted sum O*i*(net activation) using

$$Oi = \sum_{i=1}^{96} Wji\ Xj$$

Where, i=0, 1, 2……..45.

iii. Calculate the sum of positive weight, Pwi of the weight_matrix.
iv. Calculate Yi=f (Oi) =Oi/Pwi.

Where, Oi →Net activation for each character i (e.g. अ, आ,…, ज्ञ)

Pwi→ the sum of positive weights of the weight_matrix for each character.

v. Pick the maximum Yi.
vi. Check if the corresponding neuron fires.
vii. If neuron fires, recognize corresponding character.
viii. Repeat steps 1 to 7 for any other character to be tested.

4. RESULTS AND DISCUSSION

We have generated the dataset by collecting samples of handwritten devnagri characters from different peoples. Further we have trained the neural network for 17-22 samples of each character (total 1050 samples) in a supervised manner by considering inter-class similarity. A testset consisting of 15 samples per character total 690 test characters is used to test the system. The recognition system has been implemented using combined approach of neural network and thinning as a major pre-processing activity. The implemented system when tested on the given testset showed considerable improvement of 5-12% in recognition of the various hand written non compound characters in Devnagri when preprocessed with and without rotation invariant thinning algorithm (**Table 5**). The system has also been trained and tested for hand written compound characters in Devnagri which gave an accuracy of around 35-55% (Table5) during the recognition.

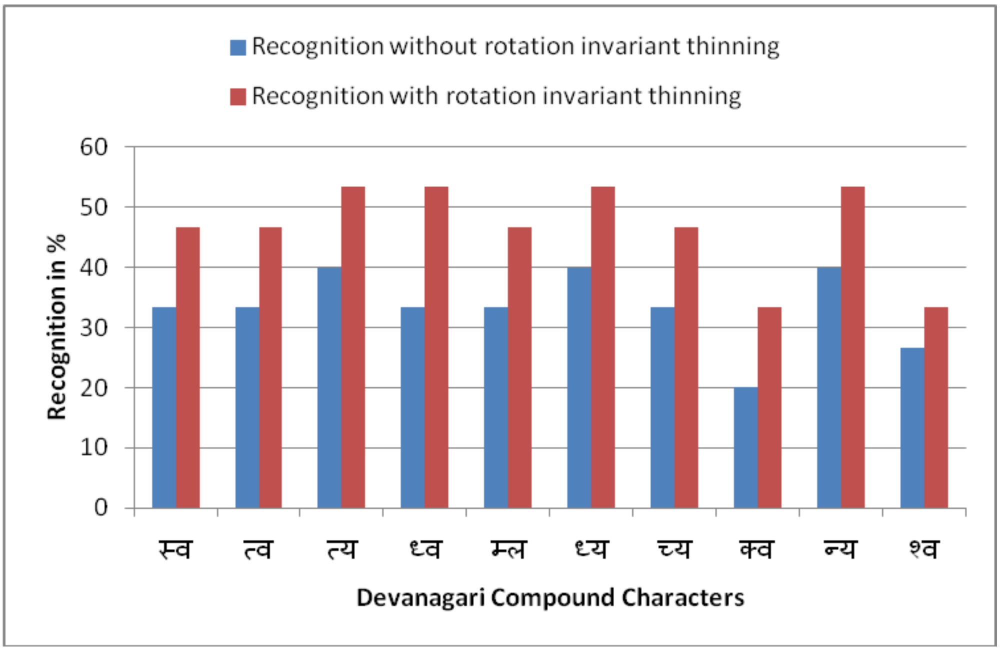

Figure 8. Results plot for Compound Characters

CONCLUSION

The proposed thinning algorithm preserves the shapes of input characters by representing the characters through their central line by parallelly applying the various rules to each pixel. Applying rotation invariant enhanced Thinning technique has helped in improvisation of the recognition results for hand written Devnagri non compound characters approximately by 5-12 % for many of the characters given as input. It has also helped in recognizing the hand written Devnagri compound characters to a better extent (approximately 35-55% accuracy) as compared to other thinning technique giving accuracy of 34-40%.

Table 5. Results obtained for the test set

Input Character	**Correctly Recognized characters out of 15**		**Accuracy Without Rotation Invariant**	**Accuracy With Rotation Invariant**	**Input Character**	**Correctly Recognized characters out of 15**		**Accuracy Without Rotation Invariant**	**Accuracy With Rotation Invariant**
	Without Rotation invariant	With Rotation invariant				Without Rotation invariant	With Rotation invariant		
अ	12	13	80.00	86.66	ढ	10	11	66.66	73.33
आ	13	14	86.66	93.33	ण	13	13	86.66	86.66
इ	13	14	86.66	93.33	त	12	13	80.00	86.66
ई	13	14	86.66	93.33	थ	11	12	73.33	80.00
उ	12	13	80.00	86.66	द	12	14	80.00	93.33
ऊ	12	13	80.00	86.66	ध	12	12	80.00	80.00
ए	13	14	86.66	93.33	न	13	14	86.66	93.33
ऐ	13	14	86.66	93.33	प	12	13	80.00	86.66
ओ	11	12	73.33	80.00	फ	11	12	73.33	80.00
औ	11	12	73.33	80.00	ब	10	12	66.66	80.00
अं	10	11	66.66	73.33	भ	10	11	66.66	73.33
अः	11	11	73.33	73.33	म	10	11	66.66	73.33
क	14	14	93.33	93.33	य	10	11	66.66	73.33

Input Character	**Correctly Recognized characters out of 15**		**Accuracy Without Rotation Invariant**	**Accuracy With Rotation Invariant**	**Input Character**	**Correctly Recognized characters out of 15**		**Accuracy Without Rotation Invariant**	**Accuracy With Rotation Invariant**
	Without Rotation invariant	With Rotation invariant				Without Rotation invariant	With Rotation invariant		
ख	12	13	80.00	86.66	र	12	13	80.00	86.66
ग	13	14	86.66	93.33	ल	12	13	80.00	86.66
घ	12	13	80.00	86.66	व	10	11	66.66	73.33
च	13	14	86.66	93.33	श	11	12	73.33	80.00
छ	10	13	66.66	86.66	ष	10	12	66.66	80.00
ज	12	13	80.00	86.66	स	11	12	73.33	80.00
झ	10	13	66.66	86.66	ह	11	12	73.33	80.00
ट	13	13	86.66	86.66	क्ष	10	11	66.66	73.33
ठ	12	12	80.00	80.00	ज्ञ	10	11	66.66	73.33
ड	12	13	80.00	86.66	त्र	10	11	66.66	73.33
स्व	5	7	33.33	46.66	ध्य	6	8	40.00	53.33
त्व	5	7	33.33	46.66	च्य	5	7	33.33	46.66
त्य	6	8	40.00	53.33	क्व	3	5	20.00	33.33
ध्व	5	8	33.33	53.33	न्य	6	8	40.00	53.33
म्ल	5	7	33.33	46.66	श्व	4	5	26.66	33.33

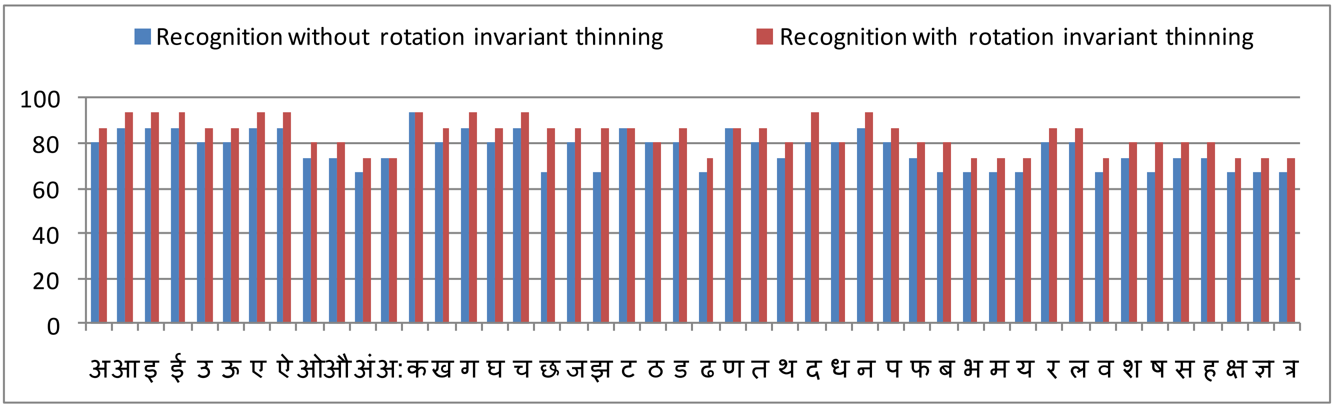

Figure 9. Results plot for Non-Compound Characters.

REFERENCES

[1] Choudhary, Rishi, "Improving The Character Recognition Efficiency of Feed Forward BP Neural Network", *International Journal of Computer Science & Information Technology* (IJCSIT), Vol 3, No 1, Feb 2011, pp. 85-96.

[2] PrachiMukherji, PritiRege, "Shape Feature and Fuzzy logic based offline Devanagari handwritten optical character Recognition", *International Journal of Pattern Recognition Research*, Vol. 4, 2009, pp. 52-68.

[3] Amit Choudhary, Nikita Walia, "Evaluation of Neural Network Accuracy for Handwritten Character Recognition", *International Journal of Information and Computation Technology* Volume 3, Number 3 (2013), pp. 119-124.

[4] M. Hanmandlu, K. R. Murali Mohan, Vivek Gupta, "Fuzzy logic based handwritten character recognition", DOI: 10.1109/ICIP.2001.958046 Conference: Image Processing, 2001, Proceedings. 2001 International Conference on, Volume: 3, pp. 714-117.

[5] Sara L. Su, Chenyu Wu, Ying-Qing Xu, "A Hybrid

Approach to Rendering Handwritten Characters", Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 04 January2004, pp. 1-8.

[6] Vikas J Dongre, Vijay H Mankar, "Devanagari Document Segmentation Using Histogram Approach", *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, Vol.1, No.3, August 2011, pp. 46-53.

[7] Mitrakshi B. Patil, VaibhavNarawade, "Recognition of Handwritten Devanagari Characters through Segmentation and Artificial neural networks", *International Journal of Engineering Research & Technology (IJERT)* Vol. 1 Issue 6, August - 2012 ISSN: 2278-0181, pp. 1-5.

[8] Ajmire P.E., Warkhede S.E., "Handwritten Marathi character (vowel) recognition", *Advances in Information Mining*, ISSN: 0975–3265, Volume 2, Issue 2, 2010, pp. 11-13.

[9] Satish Kumar, "An Analysis of Irregularities in Devanagari Script Writing- A Machine Recognition Perspective", *International Journal on Computer Science and Engineering* Vol. 2, No. 2, 2010, pp. 274-279.

[10] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, M Kundu, "Recognition of Non-Compound Handwritten Devanagari Characters using a Combination of MLP and Minimum Edit Distance", *International Journal of Computer Science and Security (IJCSS)*, Volume (4) : Issue (1), pp. 1-14.

[11] M. Egmont Petersena, D. de Ridderb, H. Handelsc, "Image processing with neural networks-a review", Copyright © 2002 *Elsevier Science*, Vol. 35, No. 10, 2002, pp. 2279-2301.

[12] M. Senthilnayaki, S. Veni, K.A.NarayananKutty "Hexagonal Pixel Grid Modeling for Edge Detection and Design of Cellular Architecture for Binary Image Skeletonization", Published in Sept. 2006 in Annual IEEE India Conference, New Delhi DOI:10.1109/INDCON.2006.302802, ISSN: 2325-940X, pp. 1-6.

[13] Mr. R.M.Noorullah, Dr. A. Damodaran, "Innovative Thinning and Gradient Algorithm For Edge Field and Categorization Skeleton Analysis Of Binary And Grey Tone Images", *1n Journal Of Theoretical And Applied Information Technology*, Vol 5. No1.2009, pp. 75-80.

[14] Jon Almazan, Albert Gordo, Alicia Forn_es, and Ernest Valveny, "Word Spotting and Recognition with Embedded Attributes", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 36, No. 12, December 2014, pp. 2552- 2566.

[15] P. M. Yawalkar and M. U. Kharat, " Handwritten Character Recognition Systems", *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 5, No. 6, Nov-Dec 2016, pp. 95-110.

[16] Nafiz Arica and Fatos T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 24, No. 6, June 2002, pp. 801- 813.

[17] Jaehwa Park, "An Adaptive Approach to Offline Handwritten Word Recognition", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 24, No. 7, July 2002, pp. 920-931.

[18] Maher Ahmed and Rabab Ward, "A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition" , *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 24, No. 12, December 2002,pp. 1672- 1678.

[19] P. M. Yawalkar, M. U. Kharat and S. V. Gumaste, "Segmentation of Multiple Touching Hand Written Devnagari Compound Characters: Image Segmentation for Feature Extraction", *IGI Global*, Chapter 8, May 2018, pp.140-163.

[20] Alessandro Vinciarelli, SamyBengio, and Horst Bunke, "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 26, No. 6, June 2004, pp. 709-720.

[21] Simone Marinai, Marco Goriand Giovanni Soda, "Artificial Neural Networks for Document Analysis and Recognition", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 27, No. 1, January 2005, pp.23-35.

[22] Tal Steinherz, Ehud Rivlin, Nathan Intrator, and PredragNeskovic, "An Integration of Online and Pseudo-Online Information for Cursive Word Recognition," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 27, No. 5, May 2005, pp. 669-683.

[23] TheingiHtike, YadanaThein, "Handwritten Character Recognition Using Competitive Neural Trees", *International Journal of Engineering and Technology*, Vol. 5, No. 3, June 2013 pp. 1-5.

[24] Matthias Zimmermann, Jean-Cedric Chappelier and Horst Bunke, "Offline Grammar-Based Recognition of Handwritten Sentences", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 28, No. 5, May 2006, pp. 818- 821.

[25] Alain Biem, "Minimum Classification Error Training for Online Handwriting Recognition", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 28, No. 7, July 2006, pp. 1041-1051.

[26] Yu Qiao, MikihikoNishiara, and Makoto Yasuhara, "A Framework Toward Restoration of Writing Order from Single-Stroked Handwriting Image", *IEEE Transactions On Pattern Analysis And Machine*

Intelligence, Vol. 28, No. 11, November 2006, pp. 1724- 1737.

[27] Thierry Artieres, SanparithMarukatat, and Patrick Gallinari, "Online Handwritten Shape Recognition Using Segmental Hidden Markov Models", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 29, No. 2, February 2007, pp. 205-217.

[28] Michael S. Brown, Mingxuan Sun, Ruigang Yang, Lin Yun, and W. Brent Seales, "Restoring 2D Content from Distorted Documents", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 29, No. 11, November 2007, pp. 1904- 1916.

[29] Yi Li, YefengZheng, David Doermann, and Stefan Jaeger, "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 30, No. 8, August 2008, pp. 1313-1329.

[30] D.S. Guru and H.N. Prakash, "Online Signature Verification and Recognition: An Approach Based on Symbolic Representation", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 31, No. 6, June 2009, pp. 1059- 1073.

[31] MoussaDjioua, and RejeanPlamondon "A New Algorithm and System for the Characterization of Handwriting Strokes with Delta-Lognormal Parameters", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 31, No. 11, November 2009, pp. 2060- 2072.

[32] SushamaShelke, ShailaApte, "A Multistage Handwritten Marathi Compound Character Recognition Scheme using Neural Networks and Wavelet Features", *International Journal of Signal Processing, Image Processing and Pattern Recognition*,Vol. 4, No. 1, March 2011, pp. 1-14.

[33] Dong Xiao Ni, "Application of Neural Networks to Character Recognition", Proceedings of Students/Faculty Research Day, CSIS, Pace University, May 4th, 2007, pp. C4.1-C4.6.

[34] Ankit Sharma, Dipti R Chaudhary, "Character Recognition Using Neural Network", *International Journal of Engineering Trends and Technology (IJETT)* - Volume4 Issue4- April 2013, pp. 662-667.

[35] Jonathan J. Hull, Alan Commike, Tin-Kam Ho, "Multiple Algorithms forHandwritten Character Recognition", Int. Workshop on Frontiers in Handwriting Recognition, Montreal, Canada, April 2-3, 1990, pp. 117-129.

[36] Rakesh Kumar Mandal, N R Manna, "Hand Written English Character Recognition using Column-wise Segmentation of Image Matrix (CSIM)", *Wseas Transactions On Computers*, Issue 5, Volume 11, May 2012, pp. 148-158.

[37] Anita Pal, Dayashankar Singh, "Handwritten English Character Recognition Using Neural Network", *IJCSC*, 1, No. 2, July-December 2010, pp. 141-144.

[38] M. S. Kumbhar, Y. Y. Chandrachud, "Handwritten Marathi Character Recognition Using Neural Network", *IJETAE*, Volume 2, Issue 9, September 2012, pp. 55-58.

[39] SameekshaBarve, "Optical Character Recognition Using Artificial Neural Network", *IJATER*, Volume 2, Issue 2, MAY 2012, pp. 139- 142.

[40] NishaVasudeva, H Jyotsana Parashar, Singh Vijendra, "Offline Character Recognition System Using Artificial Neural Network", *IJMLC*, Vol. 2, No. 4, August 2012, pp. 449-452.

[41] H. B. Kekre, M. U. Kharat and S. R. Sange, "Image data compression using new halftoning operators and run length encoding", *Springer 2011*, DOI 10.1007/978-81-8489-989-, pp. 208-213.

[42] Khalafkhatatneh, Ibrahiem M.M El Emary and Basem Al- Rifai , "Probabilistic Artificial Neural Network For Recognizing the Arabic Hand Written Characters", *Journal of Computer Science* 2 (12), 2006 ISSN 1549-3636 , pp. 879-884.

[43] Gaurav Kumar and Pradeep Kumar Bhatia, "Neural Network based Approach for Recognition of Text Images", *IJCA* (0975 – 8887) Volume 62– No.14, January 2013, pp.8-13.---43

[44] ReetikaVerma, Mrs. Rupinder Kaur, "Review on Offline Handwritten Character Recognition using Feed Forward Neural Network and SURF Feature", *IJARCCE* Vol. 3, Issue 5, May 2014,pp. 6665-6668.

[45] Richard G. Casey, Eric Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 18, No. 7, July 1996, pp. 690-706.

[46] Suruchi G. Dedgaonkar, Anjali A. Chandavale, Ashok M. Sapkal, "Survey of Methods for Character Recognition", *IJIET*, Volume 1, Issue 5, May 2012, pp. 180-189.

[47] MdFazlul Kader, Kaushik Deb, "Neural Network-Based EnglishAlphanumeric Character Recognition", *International Journal of Computer Science, Engineering and Applications (IJCSEA)* Vol.2, No.4, August 2012, pp. 1-10