International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
ISSN: 2150-7988 Vol.2 (2010), pp.039-055
http://www.mirlabs.org/ijcisim

Devanagari Hand-printed Character Recognition using Multiple Features and Multi-stage Classifier

Satish Kumar
Panjab University Regional Centre, Muktsar, Panjab, India
satishnotra@yahoo.co.in

Abstract

Devanagari is an important script mostly used in South Asia and it is also the script of Hindi the mother-tongue of majority of Indians. Though work done for the recognition of Devanagari hand-printed characters have been reported by some authors, but the availability of good ICR for this script is still a dream. In this paper, a three tier strategy is suggested to recognize the hand-printed characters of this script. In primary and secondary stage classification, the structural properties of the script are exploited to avoid classification error. The results of all the three stages are reported on two classifiers i.e. MLP and SVM and the results achieved with the later are very good. The performance of the proposed scheme is reported in respect of recognition accuracy and time. The recognition rate achieved with the proposed scheme is 94.2% on our database consisting of more than 25000 characters belonging to 43 alphabets. The recognition rate has further improved to 95.3% when a conflict resolution strategy between some pair of characters is used.

1. Introduction

The interaction with a computer is mostly made through a keyboard, which is a slow mode of communication. In order to make human-machine interaction more effective, it is desirable that a machine automatically reads a document. Persons having a good knowledge of the script of a particular language can easily read machine-printed or hand- printed documents of that language. The machine simulation of this human behavior of reading is synonymous to recognition. If this ability of human being is created in computer system then the interaction with computer will be more significant. Character recognition, a branch of document image processing, deals with the ability of a machine to recognize a graphically expressed symbol (in the form of an image) and assign a particular label to it. The main goal of the researchers is to develop a system which can read a machine-printed or a hand-printed material accurately and fast. In off-line hand-print recognition, the language expressions are written on a document page and then scanned using a scanner and recognized in various phases such as: scanning, pre-processing, feature extraction, classification and post-processing. Among the various phases, the feature extraction and classification phases are important. In feature extraction phase a set of useful properties of a character or a word available as an image is defined and extracted, and these properties are also known as features. On the basis of these properties, a given character or word is assigned a label in classification phase. In order to properly depict an image expression (character or word), a single feature may be used or multiple features may be used. Similarly, a single stage classifier may be used to ascertain the class a given character from the number of classes or multi-stage classifiers may be used as per requirement. Some authors have also suggested combining a number of classifiers which may be homogeneous or heterogeneous in nature.

Handwritten recognition is an important task as it has number of applications. A survey on handwritten recognition is carried by Plamondon et al[3], Koerich et al [4], and Arica et al [5]. A review of the state of the art in off-line cursive Roman recogntiton is made by Bunke[6]. A overview of some methodologies used for character recognition before 1990 is reported by Govindan et al[1]. A survey on some feature extraction methods used for character recognition has been carried out by Trier et al[7]. Pal et al[2] 's work give a survey on work done for the recognition of Indian language scripts.

Devanagari script is being used in various languages, in South Asian subcontinent, such as Sanskrit, Rajasthani, Marathi and Nepali and it is also the script of Hindi, the mother tongue of majority of Indians. Recognition of handwritten characters of Devanagari alphabet set is an important area of research. There are a few papers available in literature on Devanagari handwritten character or word recognition, even though it has been used by millions of people in India and abroad and it has numerous applications. In other words, a lot of work needs to be done for good ICR of this script. A few papers are available on Devanagari hand-printed character or word recognition in the literature. Prominent among these are: U. Pal[33], Sharma et al[29] and Deshpande et al [12]'s works for isolated character recognition and Parui et al[11] and Shaw et al[10]'s works on Devanagari hand-printed isolated word recognition. As far as other Indian scripts are concerned, Rehman et al [13] and Bhowmik et al[14] woks on Bengali/ Bangla character recognition; Roy et al[15], Bhattacharya et al[16]; Wen et al[17] work for Bangla numeral recognition; Roy et al [18] work for Oriya numeral recognition; Hanmandlu et al[19] and Elnagar et al[20] work for Devanagari numeral recognition are available in the contemporary literature. Oh et al [21], Dong et al [22], Koerich et al [23] and Britto et al [24]'s works for hand-printed Roman isolated character recognition are sound contributions in literature.

The various feature extraction methods are available in literature which have been used / suggested for the recognition of Roman /Arabic/ Chinese character or numeral either as standalone or in combination with other features. We have also conducted our study on more than 10 feature extraction methods and also evaluated their comparative discrimination ability and sorted some combinations which are performing very well as compared to others for Devanagari hand-printed character recognition [8]. Two top performing combinations which emerged are GrdTdist-200 and GrdNpw-200. GrdTdist-200 is a combination of two features i.e. Gradient (Sobel) and Total distances in Four Directions (TDIST), each of which contributes 100 features. Similarly, GrdNpw200 , a combination of two features i.e. Gradient (Sobel) and Neighborhood Pixels Weights (NPW) each of which contributes 100 features. The method of computing NPW-100, Grd-100 and TDIST-100 is mentioned in Section 3.3 and their recognition performance along with feature extraction and classification time with SVM and MLP are given in Table 10. The study further concluded that among the various classifiers used for conducting comparisons, SVM classifier is performing far better, but its classification time in single stage recognition scheme is about 19 times more as compared to MLP which is second better performing classifier in respect of recognition. This study has been conducted on same Devanagari hand-printed database which is being used in this paper and consists of more than 25,000 characters belonging to 43 basic Devanagari alphabets. In order to reduce classification time without compromising recognition accuracy it has been decided to perform recognition in multiple stages.

In this paper, we have suggested a three stage scheme for Devanagari hand-printed character recognition. The suggested recognition scheme is better in respect of time and accuracy as compared to various single stage recognition schemes. Two classifiers, i.e. MLP and SVM have been used in this study and results with both these classifiers have been given and compared in respect of time and accuracy. In Section 2, Devanagari character set is presented at a glance. In Section 3, the features used in various stages of the proposed three stage recognition scheme have been elaborately explained. The classifiers used in this study for conducting experiments are discussed in brief in Section 4. The proposed three stage recognition scheme along with experimental results on each stage are given in Section 5. The comparison of results of the proposed three stage recognition scheme for Devanagari and various recognition schemes suggested by some authors for Roman and Indian language scripts have been made in Section 6. Section 7 deals with the conclusion of this study.

2. Devanagari Alphabet Set: At a Glance

Devanagari is an important script. It is not only oldest but is also the widely used script in South Asia. 'Akshar' or 'Varn' is a core linguistic unit of this script equivalent to a character in English. One or more 'Akshar' forms a 'Shavad' equivalent to a word in English. Devanagari 'Varnmala' (alphabet set) constitutes 'Swar' (vowels) and 'Vyanjan' (consonants). Some constituents of Devanagari alphabet set are as:

a). Vowels and their vowel symbols : In Devanagari, vowels are called as 'Swar' and vowel symbols are called as 'Matras'. The vowels are used as stand-alone. To modify the sound of a consonant, a vowel symbol is used instead of a vowel itself. All vowel symbols are not used in the same way. A vowel symbol is either used on above or below or to left or to right to a consonant or a vowel. The various vowels are given in Row 1 and their corresponding vowel symbols are given in Row 2 of Table 1. Out of these vowels, ऋ is not a pure vowel.

b). Consonants: In Devanagari, consonants are called as 'Vyanjan'. The various consonants are given in Table 2(A). Consonants are either used as a single character or used as multiple characters known as 'Sanyukt Varn' also called as compound characters.

c). Pure consonants: These are obtained from consonants either by removing vowel symbol 'ा' (pronounced as 'a') from a consonant or by putting a 'Halant' (a vowel omission sign) below a consonant or by removing a part of stroke from a consonant. Various pure consonants in Devanagari are given in Table 2(B).

d). Commonly used compound characters: Compound characters are formed by combination of two consonants. Four commonly used compound characters are given in Table 3(A). The compound characters exhibit complex shape as compared to basic characters, i.e., 'Swars' and 'Vyanjan'. There are some more compound characters but their frequency of use is very small.

The vowel, consonants and compound characters mentioned in Tables 1, 2(A-B), 3(A) are formed from the following 43 basic symbols given in Table 3(B).

Table 1. Various Devanagari vowels and their corresponding vowel symbols.

अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ	ऋ
	ा	ि	ी	ु	ू	े	ै	ो	ौ	ृ

Table 2. **(A)** Various consonants in Devanagari alphabet set, and **(B)** Various pure consonants in Devanagari alphabet set.

A)

क	ख	ग	घ	ङ
च	छ	ज	झ	ञ
ट	ठ	ड	ढ	ण
त	थ	द	ध	न
प	फ	ब	भ	म
य	र	ल	व	
स	ष	श	ह	

B)

क्	ख्	ग्	घ्	ङ्
च्	छ्	ज्	झ्	ञ्
ट्	ठ्	ड्	ढ्	ण्
त्	थ्	द्	ध्	न्
प्	फ्	ब्	भ्	म्
य्	र्	ल्	व्	स्
ष्	श्	ह्		

Table 3. **(A)** Commonly used compound characters in Devanagari alphabet set and **(B)** Most frequently used basic graphic symbols to represent Devanagari alphabet set..

A)

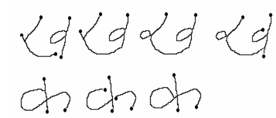
अ	इ	उ	ए	ऊ
क	ख	ग	घ	ा
च	छ	ज	झ	ञ
ट	ठ	ड	ढ	ण
त	थ	द	ध	न
प	फ	ब	भ	म
य	र	ल	व	स
ष	श	ह	क्ष	ज्ञ
त्र	श्र	ऋ		

B)

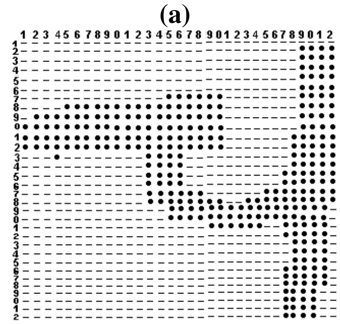
क्ष	ज्ञ	त्र	श्र

3. Features used in various Stages

Various authors have suggested multistage recognition schemes to solve hand-printed or machine-printed recognition problems. Different approaches have been suggested to divide an alphabet set of various scripts into number of subsets. Some authors have used structural features, particularly the topological features, but in our case it is difficult to divide Devanagari alphabet set into subsets merely based on topological features. Suppose we want to use the number of end points as a feature for primary classification. The number of end points may be different even for a single Devanagari alphabet character written by different individuals or an individual under different circumstances. The situation is depicted in Figure 1(a), where the numbers of end points are different for Devanagari handwritten characters क and ख written in different ways. In this case, there are four variations of ख and three variations of क. There may be more variations for these cases due to the presence of some spurious branches which are difficult to eliminate. The similar situation may arise with T-points, number of touches or cross points as some times some unwanted clusters are created during thinning process. There are a lot of variations if all the 43 classes under consideration are considered and it is very difficult to find out a set of topological features which may be used as primary or secondary stage classification criteria. All Devanagari characters in Figure 1(a) are taken after removing head line.



(a)



(b)

Figure 1(a) Some Devanagari handwritten characters with varying number of end points, and **b).** Devanagari character च normalized to 32×32 pixels size.

A three stage recognition strategy for Devanagari handwritten character recognition is suggested here. The structural properties of Devanagari is exploited here. One most important characteristic of Devanagari is the presence of side bar in some alphabet characters which can be used to divide Devanagari alphabet set into two subsets. The estimation of presence of side bar is difficult but not impossible. In first stage, Devanagari character set is divided into two subsets based on presence or absence of side bar. In this stage all the 43 classes mentioned in Table 3(B), under consideration, are grouped into two categories, says, N_Bar and Bar as:

1) Category N_Bar: It consists of Devanagari alphabet letters having no side bar. It contains 15 classes, which are follows as: फ ऋ क ट ठ ड र ए ह इ उ ऊ ढ द छ.

2) Category Bar: It consists of Devanagari alphabet letters having side bar. It contains 28 classes, which are follows as: थ ध भ म अ ख ग घ झ प य स श ष ण ल व ब त्र ज्ञ ा ज ञ च त न क्ष श्र.

3.1 Features for Primary Stage

As already mentioned that one of the most important characteristic of Devanagari is the presence of side bar in some alphabet characters which can be used to divide Devanagari alphabet set into two subsets. This property of Devanagari is exploited here to partition its alphabet set into subsets. The variations in hand-printed characters are so large that merely estimating the side bar is not enough for primary classification. A feature that can depict the overall characteristic of a character image also need to be considered so that if features based on side bar failed to provide enough information for classification, the global feature may assist and ultimately classify a character to its exact class.

3.3.1 Primary Stage Feature Set (PSFS): In primary stage classification, Primary Stage Feature Set (PSFS) is used, which is a mixture of various structural based features, in combination with gradient based features discussed in Subsection 3.3. The PSFS is a mixture of profiles and chain code based features having feature vector size 78. Out of these 78 features, 20 features are due to right first difference profiles (RFDP), 30 features are due to chain codes of right boundary of a character image and 28 features are due to raw profiles (left and right 14 each).

a) Right First Difference Profiles (RFDP) The right first difference profiles are extracted from right profiles of a normalized character image. The right first difference profiles are the difference of the right profiles corresponding to two adjoining rows. The right profiles of characters स (Figure 2(a)) and इ (Figure 2(b)) corresponding to their top 21 rows are given in Table 4 in Col. 2 and 4, respectively.

The RFDP computed from right profiles of characters स and इ are given in Table 4 in Col. 3 and 5, respectively. The variations in RFDP of इ on Row No. 7 and 10, Table 4 , is quite large whereas the variation in RFDP of स for all rows is very small or negligible. There are small variations in RFDP of character images having side bar on right side and large variation in RFDP of character images having no side bar on right side. So RFDP can be used as a distinguishing feature between bar and non-bar character images. We have taken 20 RFDP as a part of feature vector. To remove negative effect present in some features of RFDP, the value of width of the character is added to each feature component.

b) Raw Profiles : The profiles in original form are called as raw profiles. The left and right profiles of a character image from the alternative rows have been used as feature. The total features due to raw profiles are 28 with 14 features each from left and right profiles. Each feature component due to RFDP and raw profiles are normalized with the maximum value attained out of all the features due to RFDP and raw profiles.

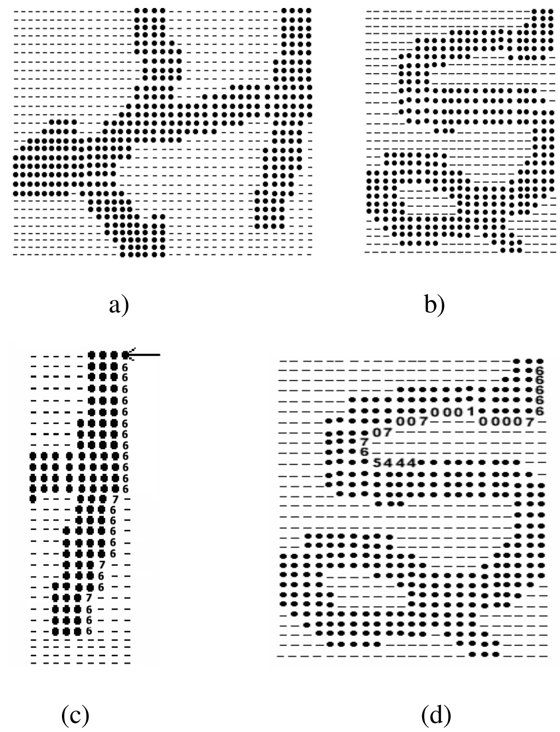


Figure 2: a) Binary image of character स with height normalized to 30 pixels size and **b)** Binary image of character इ with height normalized to 30 pixels size **c).** Codes of right boundary of स, **d)** Some codes for right boundary of इ.

Table 4. Right profiles and RFDP of binary character images स and इ corresponding to top 21 rows.

Row No.	Right Profile स	RFDP स	Right Profiles इ	RFDP इ
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	1
7	0	0	1	12
8	0	0	13	2
9	0	0	15	0
10	0	0	15	-13
11	0	0	2	-1
12	0	0	1	-1
13	0	1	0	0
14	1	0	0	0
15	1	0	0	0
16	1	0	0	0
17	1	0	0	0
18	1	0	0	0
19	1	1	0	0
20	2	0	0	0

c) Chain Codes of Right Boundary: Chain codes are directional codes and represent a contour of an image using sequence of codes where each code stands for the direction of a particular pixel w.r.t. its adjoining contour pixels. The chain codes of right side of character images having side bar and no side bar also varies a lot. The right boundary of a character image is tracked to extract chain codes. The top right black pixel is taken as starting pixel. The codes for right boundary of स and इ are given in Figures 2(c) and 2(d) respectively. Here 30 codes have been used as a part of feature vector. The codes are normalized by dividing each vector by 7 as 7 is maximum value of the code used. The first 24 codes, leaving two codes due black pixels present on extreme right on two top rows, of right side for स and इ are given below:

Codes of स : 6 6 6 6 6 6 6 6 6 6 6 7 6 6 6 6 6 7 6 6 7 6 6 6.
Codes of इ : 6 6 6 6 7 0 0 0 0 1 0 0 0 7 0 0 7 0 7 6 5 4 4 4.

The first four codes for both the characters are same but other codes vary a lot. In case of स, the value of codes lie between 6 and 7 whereas in case of इ the codes are different than 6 and 7.

3.2 Features for Secondary Stage

In secondary stage, the members of category N_Bar can be sub-categorized based on the middle bar and no middle bar. But the sub-categorization error is quite large so we do not further divide N_Bar subset into subsets further. The members of category Bar can be sub-categorized into two subsets depending upon whether there exists one touch or two touches with head line. Let us take single touch alphabet subset as ST_Bar (single touch with side bar) and two touch alphabet subset as TT_Bar (two touch with side bar). On the basis of topological features, there are two ways to find the number of touches with head line. In the first method, a character image is crossed horizontally from top, just below head line, either from the left or from the right side and a number of white to black transitions is counted. But it is very difficult to find the row which can be used as a cross path so that the number of touches with head line in a character can be detected exactly. This problem arises as some times(in writing) the character stroke do not touch the head line either from left or right as given in Figure 3. In such a situation, if we cross the character image at point 'A', the classification is correct and if we cross the image at point 'B' the classification is incorrect.

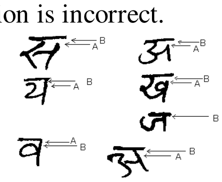


Figure 3. Some samples, where it is difficult to estimate number of touches with head line based on horizontal crossings from top.

Here some experiments are conducted, on our whole database, to detect number of touches based on a horizontal cross at various locations. The classification accuracy on the characters of both sets, ST_Bar and TT_Bar, is given in Table 5 (Sr. No. 1-3). The majority of the errors observed are due to क्ष and श्र. This is due to the fact that some persons write these characters as touching with head line on left side and some do not. The second way of finding the number of touches is based on end points and/or T-points associated with a character on upper 20% portion of a character image. In this case the character image is converted into skeleton before removing head line. If upper 20% portion of the skeletonised character image satisfies some conditions mentioned in Sr. No. 4, Column 2 of Table 5, the character image is considered to have single touch otherwise the character image is considered to have two or more touches. Here all the possibilities that may arise in case of single touch characters are considered. The examples of such characters are given in Figure 4.

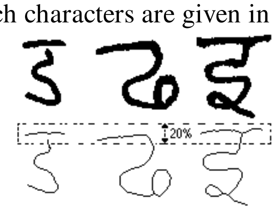


Figure 4. Devanagari characters with two end points and one touch (right), three end points only(left) and one end point only (middle).

The classification accuracy based on this is also given in Sr. No. 4 (Table 5). Hence, both the cases do not yield the exact solution. Similar results are yield with the combination of these two features. Some features that exploit the structural properties of these alphabets by statistical means are used here and reached to a solution which segregates the members of the two subsets, i.e., TT_Bar and ST_Bar with least accuracy. The features that exploit structural properties of Devanagari character images in secondary stage are called as secondary stage feature set (SSFS).

3.2.1 Secondary Stage Feature Set (SSFS):

In secondary stage, the members of alphabet subset having no side bar can be sub- categorized based on the presence of middle bar and no middle bar but this separates only three alphabet characters, i.e., क, ऋ and फ from a subset of 15 characters. Also the sub-categorization error is quite large that dividing the alphabet subset N_Bar, based on presence or absence of middle bar, into smaller subsets is not much lucrative.

Table 5. Secondary stage classification accuracy based on some topological features.

Features	Classification accuracy (%) On all four sets(A,B,C,D)	
	ST_Bar	TT_Bar
Crossing horizontally from right top tip	98.2	80.3
Crossing horizontally after leaving some portion from top tip (at-least 5-6) pixels	85.3	98.5
Crossing horizontally after leaving some portion from top tip (at-least 3-4) pixels	95.4	95.9
Convert a character image to skeleton 1) two end points and one touch. 2) three end points only. 3) one end point only	98.0	86.4

So N_Bar subset is not further divided into smaller subsets. The members of alphabet subset having side bar can be sub-categorized into two subsets depending upon whether there exists one touch or two touches with head line. On the basis of topological features, it is difficult to sub-categorize this subset further as discussed earlier. In secondary stage classification, a Secondary Stage Feature Set (SSFS) is used, which is a mixture of various structural based features, in combination with gradient based features discussed in Subsection 3.3 for this purpose. The SSFS is a mixture of profiles, histograms and crossings based features having 75 feature vectors. Out of these 75 features, 24 features are due to left, top and bottom profiles with 8 features from each side, 32 features are due to horizontal and vertical cumulative histograms with 16 features from each side, 8 features are due to width of a character at 8 locations and 11 features are due to horizontal crossings.

a) Cumulative Histograms: To extract horizontal cumulative histograms, consider an image of Devanagari character च , of size 32×32, given in Figure 1(b). Horizontal histograms of its top 16 rows are given in Col. 2 of Table 6. The cumulative sum of black pixels at each row is given in Col. 3, the cumulative sum of all pixels (black and white) at each row is given in Col. 4 and the feature component due to cumulative histogram corresponding to each row is given in Col. 5 of Table 6. The values of various feature vectors due to cumulative histograms are already between 0.0 and 1.0 so there is no need to normalize the feature vector further.

b) Raw Profiles: In order to find profiles, an image is projected from outside and the distance of its outer contour from the boundary of the bounding box of a character image, which is either square or rectangle, is obtained. The left and right profiles of a character image च , as shown in Figure 1(b) at row no. 15 are 12 and 0 respectively.

Table 6. The horizontal histograms and horizontal cumulative histograms of a binary image च.

Row No.	Hori. Hist.	Cummu. Sum (Black Pixels)	Cummu. Sum (All Pixels)	Cummu. Hist.
1	0	0	32	0.0
2	4	4	64	0.06
3	4	8	96	0.08
4	4	12	128	0.09
5	4	16	160	0.1
6	4	20	192	0.1
7	10	30	224	0.13
8	20	50	256	0.2
9	23	73	288	0.25
10	24	97	320	0.3
11	25	122	352	0.35
12	25	147	384	0.38
13	10	157	416	0.38
14	9	166	448	0.37
15	10	176	480	0.37
16	10	186	512	0.36

c) Crossings: The crossings can be considered as the number of transitions either from black to white or from white to black pixels along a particular path. If this path is along the rows, then it is called as horizontal crossings and if this path is along the columns, then it is called as vertical crossings. The value of horizontal crossings for character च ,Figure 1(b), at row no. 15 is 2.

d) Character Widths: The width of a character at a given row can be computed from its horizontal profile at that row. The normalized left and right profiles on y^{th} row are $P_L[y]$ and $P_R[y]$ and these are computed using (1).

$$P_L[y] = \frac{l_1}{N} \quad \text{and} \quad P_R[y] = \frac{l_2}{N} \tag{1}$$

Where l_1 is distance of left most foreground pixel from left boundary of bounding box of character image and l_2 is distance of right most foreground pixel from right boundary of bounding box of character image on y^{th} row and N is the width of box bounding a character image.

If at least one foreground pixel is present on y^{th} row in a character then its normalized width $W[y]$ at this row is:

$$W[y] = 1.0 - (P_L[y] + P_R[y]) \tag{2}$$

If there is no foreground pixel in a row then the left and right profiles of character image on that row are absent. Consequently, the normalized width of image at that row is zero. The normalized width of character Pk at row no. 15 is 0.6.

3.3 Feature Set for Final Recognition

In order to perform final recognition, the characters of each category are recognized using combination of three features. Only those features are considered for this purpose, which have been proved as best features in combination for Devanagari character recognition in single stage recognition scheme. As already mentioned in Section 1 that the top two combinations that perform on our Devanagari database are TdistGrd-200 and NpwGrd-200. Hence, we combine gradient, TDIST and NPW features for final recognition. Rather than taking 100 features of each feature type, we take 64 features, to avoid large size of feature vector.

3.3.1 Gradient: The importance of gradient direction was illustrated by Birk et al [25] for object recognition. Srikantan et al [26] encouraged its use for first time in OCRs where they applied it for hand-printed character recognition. The gradient of an image is a measure of the magnitude and direction of greatest change in image intensity $I(x, y)$ at each pixel (x, y). The gradient direction at any pixel (x, y) gives the direction of greatest change in image intensity and it is given as [25,26,27]:

$$\Theta(x,y) = tan^{-1}\frac{G_y(x,y)}{G_x(x,y)} \quad (3)$$

$$G_x(x,y) = \frac{\partial I(x,y)}{\partial x}, \; G_y(x,y) = \frac{\partial I(x,y)}{\partial y} \quad (4)$$

The angle Θ is measured with x-axis (horizontal axis). The direction of the edge at a pixel (x, y) is perpendicular to the gradient vector at that point. Where G_x and G_y are gradient components along x-axis and y-axis and in case of Sobel gradient these are obtained at any pixel (x, y) by convolving the given image with 3×3 windows given in Fig. 5(a-b).

-1	0	+1
-2	0	+2
-1	0	+1

a)

-1	-2	-1
0	0	0
+1	+2	+1

b)

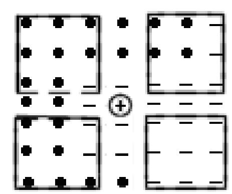
Figure 5. Sobel operator to compute gradient component along: a) x-axis b) y-axis.

The gradient components in x-direction and y-direction of a binary image at a pixel are computed by convolving the image with Sobel masks given in Figures 5(a) and 5(b) respectively. The gradient direction of each pixel in an image is obtained by using (3) and stored in an array which has the same size as that of an image. This is also called Gradient Direction Map (GDM). The gradient direction varies from 0 to 360 degree which is quantized into 4 directional levels that produce 4 directional sub-images for a given image. To reduce the size of a feature vector, each directional sub-image is divided into 4×4 regions giving the size of feature as 64 and the feature is names as Grd-64. If each directional sub-image is divided into 5×5 regions, the size of Grd feature grows to 4×5×5=100 and the feature is mentioned as Grd-100.

3.3.2 Neighborhood Pixels Weight (A New Feature): In this feature, the weights of neighboring pixels of a pixel are considered for feature computation. The neighboring pixels which are presented on any one of the four corners to a pixel have been considered in this study. A pixel has 8, 16 and 24 neighboring pixels if we consider the first level, second level and third level neighboring layers respectively. The first, second and third level layer pixels have been marked as 1,2,3 respectively in Figure 6(a).

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	P	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

(a)



(b)

Figure 6 (a) Neighborhood pixels of all the three level neighborhood layers, and **(b)** Neighboring pixels of a pixel (14, 8), given in Figure 1(b), on four corners, due to first, second and third level neighborhood layers.

Consider the case of entire three level neighborhood layer based feature extraction method. In our experiments, we have considered the weights on a pixel due to black pixels on all the four corners. At a given pixel the feature is encoded using 4-cell array. A cell contains the weight on a pixel corresponding to its neighborhood pixels on any one of the four corners and entire three level neighborhood pixels have been used. Suppose we want to encode the feature of a pixel (14, 8) for image given in Figure 1(b). The weights at this pixel corresponding to all the four corners are given in Table 7. Initially, all the four cells corresponding to pixel (14,8) are zero. Consider the case of top-left corner neighborhood pixels where eight pixels out of nine pixels are black. The weight of this cell is obtained by summing weights of (weight of each black pixel is 255 and of white pixel is 0) all the eight foreground pixels i.e. 8×255=2040 divided by maximum possible weight due to all the nine pixels i.e. 9×255=2295 which is 0.88. Similarly the weights of bottom-left, bottom-right and top-right cells are 0.77, 0.00 and 0.44 respectively. The performance of this feature is compared with some features in[36].

Table 7. Weights due to neighboring pixels on four corners.

Corner	Weights
TopRight	0.44
TopLeft	0.88
Bott.Left	0.77
Bott.Right	0.00

In case of neighborhood pixels weights (NPW), experiments are conducted by considering the pixels of all the three levels. A weight map (WM) corresponding to all the pixels in an image is prepared. The weight map consists of four planes, each having same size as of normalized image. Each plane is due to neighborhood pixels weights along a particular corner (one out of four corners) for all the pixels in an image. To extract feature vector, each WM plane is divided into 4×4 regions and average weight in each region is computed. The size of feature is 4×4×4=64 and the feature is mentioned as NPW-64. If each WM plane is divided into 5×5 regions, then the size of NPW feature grows to 4×5×5 and the feature is mentioned as NPW-100.

3.3.3 Total distances in four directions: This feature is obtained by refining directional distance distribution (DDD) feature[21]. In DDD based feature, if we take the distance traveled by a ray in all 8-directions and encode feature vector using 16-cell per pixel and subsequently divide each DDD plane into 4×4 regions, the size of feature vector grows to 16×4×4=256. The size of feature vector is too large to be combined with other features. In order to combine this feature with others, we restructure the DDD feature to smaller size. Rather than taking the distance traveled by a ray in all 8-directions, we take the total distances in four directions and the feature is encoded using 4-cell per pixel. The feature is named as TDIST in this work. The total distances of a pixel (14,8), as shown in Figure 1(b), along horizontal, vertical, left-diagonal, and a right-diagonal directions are 10,3, 5 and 2 respectively.

According to all pixels in an image, there are 4 TDIST planes. The size of each TDIST plane is same as the size of original image. The distances of the background pixels to the foreground pixels are only considered. Each TDIST plane is divided into 4×4 regions and the average of total distances in each region is computed giving 4×4×4=64 features and the feature is mentioned as TDIST-64. If each TDIST plan is divided into 5×5 regions, the size of TDIST feature grows to 4×5×5=100 and the feature is mentioned as TDIST-100.

In final stage recognition, a combination of three features i.e Gradient, NPW and TDIST is used where each feature type contributes 64 features giving the size of features as 192.

4 Classification

The various experiments are conducted on Devanagari handwritten characters using MLP and SVM classifier. These classifiers are briefed as:

4.1. Multilayer Perceptron(MLP)

Multilayer feed forward neural network with error back propagation is the widely used classifier for hand-printed problems and it performs better as compared to many other classifiers [30]. In error back-propagation algorithm, the gradient-descent method is generally used to minimize the squared error cost function. The resilient propagation algorithm has been contributed by Riedmiller et al[31] to overcome the shortcomings of gradient descent method in which the size of change in weights, say Δw_{jk} , depends upon the learning rate η as well as on partial derivatives $\partial E/\partial w_{jk}$ of the error surface. The unforeseeable behavior of partial derivative blurs the adapted learning rate η in gradient descent. Resilient propagation changes the size of weight update Δw_{jk} directly without considering the size of partial derivative. The resilient propagation has been used for conducting experiments here. In our experiments we have used implementation of resilient propagation algorithm available in [35].

4.2 Support Vector Machine (SVM)

The foundation of support vector machine is due to Vapnik [32] and its formulation is based on structural risk minimization (SRM) rather than empirical risk minimization (ERM). It is based on the concept of decision planes that defines decision boundaries. The decision plane is generally a hyper plane, which constitutes a line like function. SVM classifier will classify new training example say, u, as

$$g(u) = sign(\sum_{i=1}^{U_S} \lambda_i d_i K(u, u_i) + c) \quad (5)$$

The value of parameters λ_i and c are computed by maximizing (4.53) and d_i is label of u_i.

$$L_D = \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j d_i d_j K(u_i, u_j) \quad (6)$$

$$\text{s.t.} \quad 0 \leq \lambda_i \leq C$$

A comprehensive tutorials on SVM are given by Burges[9] . In our experiments, we have used SVM software due to Chang et al[34].

5 Experimental Results

The experiments are conducted with various features as suggested for each stage as mentioned in Section 3. Prior to perform recognition, the head line from each character is removed using a line removal algorithm which is beyond the scope of this paper. A character is size normalized to 32 ×32 after line removal using aspect ratio preservation method studied by Liu et al [28]. Since our database consists of more than 600 characters per class, the characters of each class are numbered. For our experiments we have used 600 characters per class from each class (alphabet character). In order to cross validate the results we have partitioned our database into four subsets: A, B, C and D. The size of each subset is equal. In each trial, 75% data is used for training and 25% data is used for testing, i.e. one subset is used to test and three subsets are used to train the classifier. Meaning thereby, four fold cross validation has been used. The experimental results for various stages with two classifiers are as:

a). The experimental results with MLP and SVM for primary stage classification are given in Table 8. Average classification accuracy means average accuracy (%) due to all the four sets / trials. Average primary classification time mentioned in Table 8 do not includes the feature extraction time. For primary classification, the GrdPsfs-142 feature set is dominating in respect of classification accuracy for both classifiers. The classification accuracy with SVM is 0.5 % large as compared to the classification accuracy with MLP. But classification time with SVM is about 6.2 times as compared to MLP. The classification accuracy of TdistPsfs-142 and NpwPsfs-142 features are 98.7% and 98.2%, respectively with SVM classifier, which are more than 0.5% less as compared to GrdPsfs-142 feature type with SVM. The classification error obtained using various features with MLP classifier is greater than 0.5% as compared to GrdPsfs-142 feature type with SVM classifier. The primary classification plays an important role. If a character is erroneously classified in first stage, this error is unrecoverable. The error must be reduced at this stage within reasonable classification time. So, it is suggested to use GrdPsfs-142 and SVM combinations for primary classification.

b). The secondary stage classification performance with various features explained in Sub-section 3.3 and SSFS, using SVM and MLP classifiers is given in Table 9.

Table 8. Primary stage classification accuracy with various features using SVM and MLP classifiers along with classification time.

Feature Type	Classifier	Avg. Classification accuracy (%)	Average Primary Classification time/ char (Milliseconds)
NPW-64+ PSFS-78 (NpwPsfs-142)	SVM	98.2	3.8
	MLP	98.5	0.63
Grd-64+ PSFS-78 (GrdPsfs-142)	SVM	**99.4**	**3.7**
	MLP	98.9	0.63
TDIST-64+ PSFS-78 (TdistPsfs-142)	SVM	98.7	3.5
	MLP	98.4	0.65

The combination of gradient(Grd-64) and secondary stage feature set (SSFS-75) is performing as compared to other combinations in presence of SVM classifier. The performance of MLP classifier is 0.8 % less as compared to SVM. From Tables (5 & 9), it is clear that it is better to use the statistical features that exploit the structural properties of Devanagari character for performing secondary stage classification as compared to using topological features.

c). Now, for final classification, there are subsets. The alphabet characters of these subsets are as follows:

1). Characters classified as characters having no side bar in primary stage classification. Characters are फ ऋ क ट ठ ड र ए ह इ उ ऊ ढ द छ. This subset is called as N_Bar.

2). Characters classified as characters having side bar in primary stage classification and the characters classified as characters with two / more touches with the head line in secondary stage classification. Characters are: थ ध भ म अ ख ग घ झ प य स श ष ण. This subset is called as TT_Bar.

3). Characters classified as characters having side bar in primary classification and the characters classified as characters with single touch with the head line in secondary stage classification. Characters are: ल व ब त्र ज्ञ ा ज ञ च त न क्ष श्र. This subset is called as ST_Bar.

In order to perform final recognition, the characters of each category are recognized using combination of three features i.e. Grd-64, NPW-64 and TDIST-64 each having 64 features giving 192 features in total. The average recognition accuracy on each set with average final stage classification time / character in milli-seconds is given in Table 10. The average classification accuracy mentioned here does not include the classification error mentioned in primary and secondary stages. This is due to the final individual stage only.

Table 9. Secondary stage classification accuracy based on training based model with various features using MLP and SVM classifiers.

Feature Type	Clas-sifier	Average Classification Accuracy	Average Secondary Classification time/ char (Milliseconds)
Grd-64+ SSFS-75 (GrdSsfs-139)	SVM	**99.1**	**2.6**
	MLP	98.3	0.6
Tdist-64+ SSFS-75 (TdistSsfs-139)	SVM	98.5	2.3
	MLP	98.1	0.6
NPW-64+ SSFS-75 (NpwSsfs-139)	SVM	98.1	2.5
	MLP	97.6	0.7

Table 10. Final stage recognition performance of some features with MLP and SVM classifiers for all three Devanagari alphabet subsets.

Devanagari Alphabet Subset	Features set	Classifier Type	Avg. Clas. Accuracy (%)	Avg. Final Stage Class. time/ char (ms)
N_Bar	Grd-64+TDIST-64+NPW-64 (GrdTdistNpw-192)	SVM	97.6	5.6
		MLP	95.6	1.8
TT_Bar	Grd-64+TDIST-64+NPW-64 (GrdTdistNpw-192)	SVM	91.1	5.8
		MLP	90.3	1.9
ST_Bar	Grd-64+TDIST-64+NPW-64 (GrdTdistNpw-192)	SVM	95.8	5.7
		MLP	93.8	1.8

The recognition performance of this combination is different on different Devanagari alphabet subsets. The recognition rate on N_Bar and ST_Bar subsets is higher as compared to TT_Bar subset. The reason is that the TT_Bar subset contains large number of conflicting pairs of characters. It is difficult to distinguish such characters, even some times visually. We have suggested a method to resolve the conflict between some pairs of characters based on their structural property and has been discussed in Subsection 5.2.

The features used in primary, secondary and final stage recognition of Devanagari handwritten characters is known as DFS (Devanagari Feature Set). It comprises 345 features, where 142 features are used in the primary stage, 139 features are used in the secondary and 192 features are used in the final stage recognition. Here 64 features due to gradient are common in all the three stages. It is clear from our previous discussion that the characters of subset N_Bar have to pass through two classifiers and the characters of subsets TT_Bar and ST_Bar have to pass through three classifiers for final recognition. We rather suggest to use SVM classifier for primary and secondary stage classification as it gives minimum classification error on each stage. The recognition rate in percentage and average recognition time /character in milliseconds using SVM classifier for both primary and secondary stages and SVM or MLP classifier in final stage are given in Table 11. The results have been recorded by testing complete Devanagari character recognition system on all four test subsets using cross validation.

Time period underlined means time period used for primary or secondary stage classification using SVM classifier. In this three stage recognition scheme, even if we use MLP classifier for final stage recognition instead of SVM, the recognition rate is 93.0% which is 1.2% less as compared to using SVM as a classifier for final stage recognition. The time period with a SVM as final stage classifier is about 1.55 times large as compared to the time period with MLP classifier. Our first choice is to use SVM classifier and the second choice is to use MLP classifier for final stage recognition. For primary and secondary stage classification, we have already suggested to use SVM classifier.

Table 11. Recognition rate in percentage and average recognition time /character in milliseconds using SVM classifier for primary, secondary stages and SVM or MLP classifier in final stage.

SVM classifiers for Primary and Secondary Stages				
Clas. for F. Stage	Alphabet Characters Subset			
	N_Bar	TT_Bar	ST_Bar	Avg.
SVM	9.3 ms (3.7+5.6)	12.1 ms (3.7+2.6 +5.8)	12.0 ms (3.7+2.6 +5.7)	11.1 ms
	96.8%	91.4%	94.5%	94.2%
MLP	5.5 ms (3.7+1.8)	8.2 ms (3.7+2.6 +1.9)	8.1 ms (3.7+2.6 +1.8)	7.2 ms
	95.2%	89.1%	94.8%	93.0%

5.1 Comparative Study (Recognition Accuracy and Time)

As already mentioned, the classification time also depends upon the number of classes (the size of alphabet set of a script) under consideration in addition to the size of feature vector and the type of classifier used. The size of alphabet set not only affects the training time but the recognition time too. The number of classes considered for recognition in a classifier contributes a lot to the classification time. In Sections 1, some results have been discussed which are obtained by designing some single stage classification schemes using single or multiple features. In this Section, we are making the comparisons of the average computational time (feature extraction and classification) required per character and the recognition accuracy between various single stage schemes and a proposed three stage recognition scheme.

In the proposed three stage recognition scheme, the primary and secondary classification is performed using SVM classifier. For final stage recognition, the results due to both MLP and SVM classifiers are presented. The feature extraction time means the time required to extract feature(s) including the time required to perform preprocessing and image reading from respective directories. The comparison performance is given in Table 12.

In single stage classification scheme, the classification time using SVM classifier is more than 19 times greater as compared to the classification time using MLP classifier for all the feature types. In the proposed three stage recognition scheme, the classification time using SVM as primary, secondary and final stage classifier is about 1.5 times greater as compared to the classification time using SVM as primary and secondary and MLP as final stage classifier. The classification time using SVM classifier in single stage recognition scheme is about 2.0-2.5 times greater as compared to the classification time using SVM classifier in all the three stages of the proposed three stage recognition scheme. Furthermore, the classification time using MLP classifier in single stage classification scheme is about 6.0-6.5 times smaller as compared to the classification time using SVM as primary and secondary and MLP as final stage classifier in the three stage classification scheme. Hence, using SVM classifier in single stage classification scheme is most expensive and using MLP classifier in single stage classification scheme is least costly in respect of classification time. However, the classification time for three stage classification scheme with SVM as primary and secondary and MLP as final stage classifier is not much high. The time period of using SVM as classifier in all the three stages of three stage recognition scheme is small as compared to the time period of using SVM as classifier in single stage classification scheme. The characters P and S mentioned in Figure 7 represent the primary and secondary classifiers.

The recognition performance of the three stage recognition scheme having feature DFS-345 using SVM classifier in all the three stages is better as compared to the various single stage recognition schemes. The average time required to recognize a character using this scheme is 40.1 milliseconds. The recognition rates of TdistGrd-200, NpwGrd-200 and Grd-200 features with SVM in single stage classification schemes are 93.9%, 93.7% and 93.5%, respectively and are somewhat good but the computational times are quite large.

Table 12. Comparison of classification and recognition (feature extraction) time in case of some single stage and a three stage recognition schemes.

Feature Type	Number of Stages (Classifier Type) SS-Single Stage		Average Time/Character (Milliseconds)			Rec. Rate (%)
			Feat. Extra.	Clas s.	Total	
Grd-200	SS (SVM)		26.9	27.9	54.8	93.5
	SS(MLP)		26.9	1.1	28.0	89.6
TdistGrd-200	SS(SVM)		28.1	23.7	51.8	93.9
	SS(MLP)		28.1	1.2	29.3	90.1
NpwGrd-200	SS(SVM)		28.3	23.6	51.9	93.7
	SS (MLP)		28.3	1.2	29.5	90.0
DFS-345(PSFS-78,SSFS-76, Grd-64, TDIST-64, NPW-6 4)	Three Stages (SVM as Primar y, Second ary)	SVM (Final)	29.0	11.1	**40.1**	94.2
		MLP (Final)	29.0	7.2	**36.2**	93.0

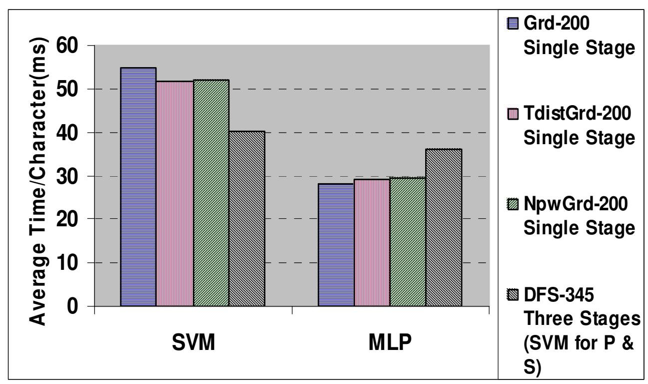


Figure 7. Comparison analysis of time required in case of some best single stage and a proposed three stage recognition schemes.

The recognition times of these schemes are more than 1.3 times large as compared to the proposed three stage recognition scheme. So, the proposed recognition scheme is better both in respect of time and accuracy. If we further want to decrease the computational time, then the second option is to use SVM classifier in primary and secondary stages and the MLP classifier in final stage recognition. Again DFS-345 feature set is used for recognition. The recognition accuracy with this scheme is 93.0% having 36.2 milli-seconds per character as average computational time.

5.2 Resolution of Conflicts

If we look at the Table 13, the most errors are between भ vs. म or घ vs. ध or य vs. थ or प vs. य or ब vs. व or त्र vs. ञ pairs of characters. We call these characters as conflicting pairs of Devanagari alphabet characters. These conflicts are obvious as there is very little difference in shapes of these pairs of characters. Out of these conflicting pairs, the errors are very large in case of भ vs. म and घ vs. ध. But there is still a scope to resolve the conflicts and reduce the recognition errors between some of these pairs. We have used a strategy to resolve the conflict between भ vs. म or घ vs. ध. There is a small gap in the head line in case of भ whereas, there exists no such gap in case of म. These characters can be distinguished on the basis of this gap. To resolve the conflict we compute the top profiles (from 15 locations without removing head line) from those characters which have been recognized as either भ or म in the final stage of recognition process. A binary classification scheme is designed, using top profile as feature and SVM as classifier, by assigning a label 0 to the characters of भ and a label 1 to the characters of म. The average recognition error of this binary classifier on all the 2×600=1200 test characters is 2.6%. The average classification time/character is 0.6 milliseconds. When, all the characters which are recognized as either भ or म are tested through this classifier, the number of conflicting errors of भ are reduced from 85 to 16 and errors of म are reduced from 40 to 2.

The same strategy is used to resolve the conflict between alphabet characters of pair घ vs. ध. Since, the same gap in the head line of character ध exists where as there is no such gap in case of character घ. The average recognition error of this binary classifier on all the 2×600=1200 test characters is 4.9%. When all the characters recognized as either ध or घ in final stage of recognition are tested through this classifier, the number of conflicting errors of घ are reduced from 59 to 12 and ध are reduced from 159 to 30. The overall recognition rate of recognition system, designed with three stage classification scheme using DFS-345 feature and SVM as classifier for all the three stages and with a conflict resolution strategy between घ versus ध and भ versus म, have improved from 94.2% to 95.3%. The details of ten alphabets having largest error before and after conflict resolution are given in Table 13. A tree representation of the three stage recognition scheme is given in Figure 8.

Table 13. Some Devanagari alphabets along with errors out of 600 test samples / alphabet before and after conflict resolution

Alphabet Name	Before Resolving conflict Errors out of 600 test pattern	Maximum Errors due to Alphabet	After Resolving conflict between ध vs घ and भ vs म
ध	174	घ -159	45
भ	109	म -85	40
य	92	थ -43, प -29	92
घ	88	ध-59	41
ञ	75	भ -18, त्र-16, अ-8	75
म	58	भ-40	20
व	53	ब-30, त -5	53
ल	44	त -10, न -10, ब -5, ड्ढ -5	44
त्र	42	ञ-21	42
स	39	ख-11	39

6 Comparisons of Results

In this Section, the results of the proposed three stage recognition scheme are compared with some results on Roman script or Indian language script available in literature.

6.1 Comparison of Results on Devanagari with some best Results on Roman Script

Many results have been reported in literature on Roman handwritten character recognition. But here only those results are considered which are latest and have been tested on some big datasets. The purpose of comparing the results is just to know the performance of some feature extraction and the classification methods on Devanagari script in comparison to the performance of some other best methodologies used for Roman script recognition. The results are given in Table 16. The recognition rate achieved on Devanagari is 94.2% with DFS-345 features and SVM as classifier for all the three stages in the three stage recognition scheme. The maximum recognition rates achieved on Roman capital and small letters are 92.3% and 84.6%, respectively by Koerich et al [23]. The recognition rate of 92.3% is achieved by Dong et al [22] on lower case letters. The recognition rate on Devanagari further improved to 95.3% when a conflict resolution strategy on some ambiguous pairs of characters is adopted (see Subsection 5.2).

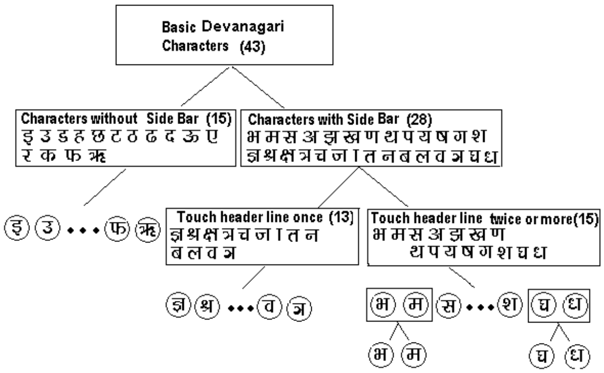


Figure 8. Tree representation of proposed three stage scheme for Devanagari handwritten characters recognition with a conflict resolution strategy.

6.2 Comparison of Results on Devanagari with some Results on Indian Language Scripts

The work done for the recognition of Indian language handwritten scripts is not much reported in literature. Here the comparison of results due to some schemes suggested by some authors on Indian language script is made with the results on Devanagari with proposed three stage recognition scheme. The comparisons are given in Table 14. The recognition error for various Devanagari symbols using three stage recognition scheme with DFS-345 features and SVM classifier for all three stages and conflict resolution strategy between ध versus घ and भ versus म is given in Table 16.

7 Discussions and Conclusion

The classification time of SVM classifier is very high as compared to MLP but the recognition error is low for our application consisting of 43 classes. It is essential to divide the Devanagari dataset into subsets so that the classification time may be reduced without compromising accuracy.

From the experimental results given in Table 5, it is clear that using topological features for dividing Devanagari set into subsets is not fruitful. PSFS and SSFS feature sets, extracted by exploiting structural properties of Devanagari script, in combination of gradient based feature are quite useful for partitioning Devanagari dataset into subsets. The characters of each subset, so obtained, are further recognized using SVM classifier. The recognition rate on Devanagari is 94.2% using DFS-345 features in combination with the three stage recognition scheme having SVM as classifier for all the three stages. The recognition rate on Devanagari further improved to 95.3% as a conflict resolution strategy is used, which exploited Devanagari script characteristic. From the Tables 14 and 15, it is clear that the result with proposed three stage recognition scheme on Devanagari script is better as compared to the results on Roman / Indian language available in literature. The recognition error with MLP classifier in any stage is large as compared to SVM. So our first choice is to use SVM classifier in all the stages and the second choice is to use SVM classifier for primary and secondary and MLP classifier for final stage recognition.

Table 14. Comparison of recognition results on some Indian language handwritten scripts with Devanagari handwritten script.

Author Name	Script Number of Classes	Methodology	Data Size		R. Rate (%)
		Feature type	Training	Test	
Rehman et al[13]	Bengali (49 Char)	Low level and high level features, Multistage (Multi-expert)	-	-	88.3
Bhowmik et al [14]	Bangla (50 Char.)	Stroke Based (200) MLP	17500	4500	84.3
Deshpande et al[12]	Devanagari (50 Char)	Chain code Histograms, Regular expression with minimum edit	-	5000	82.0
Sharma et al [29]	Devanagari	Directional Chain code, Quadratic Classifier	11270	-	80.6
Pal [33]	Devanagari	Directional Gradient, Modified Quadratic Classifier	-	36172	94.2
Proposed 3-Stage Rec. Scheme	Devanagari (43 Basic Characters)	DFS-345(PSFS-78, SSFS-76, Grd-64, TDIST-64, NPW-64), SVM(3-Stage)	19350 (Each time)	6450 ×4 (Cross Valid.)	94.2
		DFS-345(PSFS-78, SSFS-76, Grd-64, TDIST-64, NPW-64) and Top Profile for conflict resolution, SVM(3-Stage)	-do-	-do-	95.3

Table 15. Recognition results with three stage recognition scheme using DFS-345 features and SVM classifier with conflict resolution strategy between म versus भ and ध versus घ

	इ	उ	ड	ह	छ	ट	ठ	ढ	द	ऊ	ए	र	क	फ	ऋ	स	अ	झ	ख	भ	म	घ	ण	थ	ध	प	य	ष	श	ग	श्र	क्ष	त्र	च	ज	।	ज्ञ	त	न	ब	ल	व	ञ	Error	Correct	Total
इ	591	1	2	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	9	591	600
उ	0	581	8	0	0	0	4	0	0	5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	581	600
ड	6	14	578	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	578	600
ह	3	0	2	578	6	3	0	0	4	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	578	600
छ	0	0	0	9	573	0	1	0	5	0	0	0	0	1	0	0	0	2	0	0	0	7	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	27	573	600
ट	0	0	0	0	1	589	1	0	0	0	1	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0	11	589	600
ठ	0	1	0	3	0	8	579	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	21	579	600
ढ	0	0	0	0	0	8	2	587	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	587	600
द	2	0	0	5	4	1	1	17	570	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	570	600
ऊ	0	3	0	0	1	0	1	0	0	583	0	0	1	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	2	0	0	0	1	0	1	17	583	600
ए	0	0	2	0	1	0	0	0	1	1	568	5	0	2	1	3	0	1	1	0	0	0	0	0	1	6	1	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	32	568	600
र	0	0	0	1	0	6	0	0	0	0	6	575	1	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	7	0	25	575	600
क	0	0	0	0	0	0	0	1	1	2	0	3	586	2	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	14	586	600
फ	0	0	0	0	3	0	0	0	0	0	0	0	5	586	1	0	0	0	1	0	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	586	600
ऋ	0	0	0	0	1	0	0	0	0	0	0	0	3	0	589	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	11	589	600
स	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	561	3	3	11	3	8	0	0	0	0	0	2	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	39	561	600
अ	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	582	0	2	4	4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	18	582	600
झ	0	0	0	0	2	0	0	0	0	3	0	0	0	0	0	5	8	568	2	2	3	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	3	0	0	0	0	0	2	32	568	600
ख	0	0	0	0	1	0	2	2	2	2	0	0	1	0	2	14	1	0	564	0	0	0	0	0	0	0	6	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	36	564	600
भ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	1	2	560	16	0	0	2	0	0	0	0	1	2	4	3	0	0	0	0	0	0	0	0	0	0	6	40	560	600
म	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	5	2	0	0	2	580	0	1	1	0	1	0	1	0	4	1	0	0	0	1	0	0	0	0	0	0	0	0	20	580	600
घ	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	559	0	2	12	4	3	3	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	41	559	600
ण	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	587	1	0	0	0	6	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	13	587	600
थ	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	2	0	0	567	8	2	15	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	33	567	600
ध	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	30	2	9	555	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	45	555	600
प	0	0	0	0	0	0	0	0	0	0	3	0	0	2	0	0	0	0	0	1	0	1	0	3	0	562	24	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	38	562	600
य	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	3	6	1	43	0	29	508	5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	92	508	600
ष	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	2	4	2	0	3	0	2	0	0	4	576	1	0	0	0	0	0	0	0	0	0	0	1	0	2	0	24	576	600
श	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	7	0	0	1	0	0	0	0	0	586	3	0	0	0	0	0	0	1	0	0	0	0	0	0	14	586	600
ग	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0	0	0	0	0	595	0	0	0	0	0	0	0	0	0	0	0	0	0	5	595	600
श्र	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	12	1	0	0	0	2	0	0	0	0	0	562	3	9	0	2	0	0	1	0	2	0	0	4	38	562	600
क्ष	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	2	0	3	0	4	1	1	0	0	1	1	0	0	0	0	1	574	0	1	0	0	4	1	1	0	1	1	0	26	574	600
त्र	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	7	0	558	0	0	0	0	2	3	2	0	3	21	42	558	600
च	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	3	589	0	0	0	0	2	0	0	2	0	11	589	600
ज	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	2	570	0	4	0	10	1	2	0	3	30	570	600
।	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	597	0	0	0	0	0	0	0	3	597	600
ज्ञ	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	4	0	0	4	0	580	0	5	0	1	0	0	20	580	600
त	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	1	0	0	586	4	0	3	2	0	14	586	600
न	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	2	3	0	0	0	0	0	0	0	0	0	0	2	2	1	4	0	1	8	568	6	0	1	0	32	568	600
ब	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3	0	1	0	0	1	4	563	2	19	1	37	563	600
ल	0	0	0	0	0	3	0	0	0	2	1	1	2	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	5	10	10	5	556	0	0	44	556	600
व	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	1	1	2	0	0	0	0	0	3	4	0	0	5	3	30	0	547	0	53	547	600
ञ	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	8	4	0	18	6	0	0	0	0	0	0	0	0	0	3	1	16	5	4	0	3	0	4	0	0	0	525	75	525	600
																																												1202	24598	25800

Recognition Rate(%):95 .3
Error Rate(%):4.7

Table 16. Comparison of some best recognition results on English alphabet set with Devanagari alphabet set.

Authors	Number of Classes and script	Methodology	Dataset Size		R. Rate (%)
		Feature type and Size of Feature Vector	Training	Test	
Oh et al [21]	26(Upper Case)	Directional Distance Distribution, (256), MLP	26000	11941	90.0
Dong et al [22]	26(Lower Case)	Gradient - Directinal+PCA, (160), GLVQ And MLP	23937	10688	92.3
Koerich et al [23]	26 (Upper Case)	Projection Histogram+Contour profiles+Directional Code,(108), MLP	37440	11941	92.3
	26(Lower Case)		37440	12000	84.6
	52(Upper , Lower Case)		74880	23941	85.5
Britto et al [24]	26 (Upper Case)	Foreground and Background, (47), HMM	37440	11941	90.0
	26(Lower Case)		37440	12000	84.0
	52(Upper, Lower Case)		74880	23941	87.0
Proposed 3-Stage Rec. Scheme	43(Devanagari Basic characters)	DFS-345(PSFS-78, SSFS-76, Grd-64, TDIST-64, NPW-64), SVM(3-Stage)	19350 (Each time)	6450×4 (Cross Valid.)	94.2
		DFS-345(PSFS-78, SSFS-76, Grd-64, TDIST-64, NPW-64) and Top Profile for conflict resolution, SVM(3-Stage)+ Conflict resolution	-do-	-do-	95.3

Acknowledgement

Author is very thankful to Dr. Chandan Singh, Professor, Department of Computer Science, Punjabi University, Patiala, Punjab(India), for his valuable guidance to conduct this research.

Dr. Satish Kumar is a Associate Professor in Panjab University, Chandigarh(India), currently posted at Panjab University Regional Centre, Muktsar (Punjab) and has about 12 years experience of teaching Post-graduate classes and actively engaged in research activities. His areas of interest are Image processing, Pattern recognition and artificial intelligence involved with these fields.

References

[1] V. K. Govindan and A. P. Shivaprasad, "Character Recognition - A Review," Pattern Recognition, Vol. 23, No. 7 (1990).

[2] U. Pal, B.B. Chaudhuri, "Indian script character recognition: A Survey", Pattern Recognition, 37, pp. 1887-1899 (2004).

[3] R. Plamondon and S. N. Srihari, "On-line and Off-line Handwriting Recognition: a Comprehensive Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No.1,pp. 63-84(2000).

[4] A. L. Koerich, R. Sabourin and C.Y. Suen, "Large Vocabulary Off-line Handwriting Recognition: a Survey", Pattern Analysis Applications, Vol. 6, pp. 97-121(2003).

[5] N. Arica and T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-line Handwriting", IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Vol. 31, No. 2(2001).

[6] H. Bunke, "Recognition of cursive Roman handwriting past, present and future", Proceedings of Seventh International Conference on Document Analysis and Recognition, Vol. 1, pp 448-459.

[7] O. D. Trier, A. K. Jain and T. Taxt, "Feature Extraction Method for Character Recognition - a Survey", Pattern Recognition, Vol. 29, No. 4, pp. 641-662(1996).

[8] S. Kumar, "Recognition of Pre-Segmented Devanagari Handwritten Characters using Multiple Features and

Neural Network Classifier", Ph.D. Thesis, Punjabi University, Patiala, Punjab, India(2008).

[9] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, Vol. 2, No. 2, pp. 121-167 (1998).

[10] S. K. Parui and B. Shaw, "Off-line Devanagari Handwritten Word Recognition: An HMM based approach", Proc. of PReMI-2007(Springer), LNCS-4815, pp. 528-535(Dec. 2007).

[11] B. Shaw, S. K. Parui and M. Shridhar, "Off-line Handwritten Devanagari Word Recognition: A Segmentation Based Approach", IEEE (2008).

[12] P. S. Deshpande, L. Malik and S. Arora, "Fine Classification & Recogntiton of Handwritten Devanagari Characters with Regular Expression & Minimum Edit Distance Method", Journal of Computers, Vol. 3, No. 5, pp. 11-17(May 2008).

[13] A. F. R. Rahman, R. Rahman and M. C. Fairhurst, "Recognition of Handwritten Bengali Characters: A Novel Multistage Approach", Pattern Recognition, Vol. 35, No. 5, pp. 997–1006 (2002).

[14] T. W. Bhowmik, U. Bhattacharya and S. K. Parui, "Recognition of Bangla Handwritten Characters using an MLP Classifier Based on Stroke Features", Proceedings of ICONIP, Kolkata, India, pp. 814-819(2004).

[15] K. Roy, S. Vajda, U. Pal and B.B. Chaudhuri, " A System Towards Indian Postal Automation", Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition, PP. 580 - 585 (2004).

[16] U. Bhattacharya and B. B. Chaudhuri, "A Majority Voting Scheme for Multiresolution Recognition of Handprinted Numerals", Proceedings of the Seventh International Conference on Document Analysis and Recognition", Vol. 1, pp. 16- 20(2003).

[17] Y. Wen, Y. Lu and P. Shi, "Handwritten Bangla Numeral Recognition System and its Application to Postal Automation", Pattern Recognition, Vol. 40, pp. 99-107(2007).

[18] K. Roy, T. Pal, U. Pal and F. Kimura, "Oriya Handwritten Numeral Recognition System", Proceedings of the Eight International Conference on Document Analysis and Recognition, Vol. 2, pp. 770- 774 (2005).

[19] M. Hanmandlu and O.V. Ramana Murthy, "Fuzzy Model Based Recognition of Handwritten Hindi Numerals", Pattern Recognition, Vol. 40, Issue 6, pp. 1840-1854(2006).

[20] A. Elnagar and F. Al-Kharousi, S. Harous, "Recognition of Handwritten Hindi Numerals using Structural Descriptors", Journal of Experimental and Theoretical Artificial Intelligence, Vol.15, No. 3, pp. 299-314 (2003).

[21] II-S. Oh, C.Y. Suen, "Distance Features for Neural Network-based Recognition of Handwritten Characters", International Journal on Document Analysis and Recognition, Vol.1, pp. 73-88(1998).

[22] J. X. Dong, A. Krzyzak and C. Y. Suen, "Local Learning Framework for Recognition of Lowercase Handwritten Characters", Proceedings of International Workshop on Machine Learning and Mining in Pattern Recognition, Leipizig, Germany, pp. 226-238(2001).

[23] L. Koerich, "Large Vocabulary Off-line Handwritten Word Recognition", Ph. D. Thesis, École de Technologie Supérieure, Montreal-Canada (2004).

[24] S. Britto Jr., R. Sabourin, F. Bortolozzi and C. Y. Suen, "Foreground and Background Information in an HMM-based Method for Recognition of Isolated Characters and Numeral Strings", Ninth International Workshop on Frontiers in Handwriting Recognition, Kokubunji, Tokyo, Japan, pp. 371-376(2004).

[25] J. Birk, R. Kelley, N. Chen and L. Wilson, "Image Feature Extraction using Diameter-Limited Gradient Direction Histograms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, pp. 228-235(1979).

[26] G. Srikantan, S. W. Lam and S.N. Srihari, "Gradient Based Contour Encoding for Character Recognition", Pattern Recognition, Vol. 29, No. 7, pp. 1147-1160(1996).

[27] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", 2nd Ed., Pearson Education (2002).

[28] C.-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques", Pattern Recognition, Vol. 37, pp. 265-279(2004).

[29] N. Sharma, U. Pal et al ,"Recognition of off line handwritten Devanagari characters using quadratic classifier", ICVGIP 2006,LNCS 4338,pp. 805-816(2006).

[30] G. P. Zhang, "Neural Networks for Classification: A Survey", IEEE Transactions on Systems, Man, and Cybernatics—Part C: Applications and Reviews, Vol. 30, No. 4, pp. 451-462(2000).

[31] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Back-propagation Learning: The RPROP Algorithm," Proceedings of the IEEE International Conference on Neural Networks, Vol. 1, pp. 586-591 (1993).

[32] V. Vapnik, “The Nature of Statistical Learning Theory” Springer-Verlag, New Tork (1995).

[33] U. Pal, N. Sharma, T. Wakabayashi, F. Kimura, "Off-Line Handwritten Character Recognition of Devanagari Script," icdar,pp.496-500, Ninth International Conference on Document Analysis and Recognition , Vol. 1(2007).

[34] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001, software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.

[35] Neural Network Toolbox, version 5.0.2 The Math Works, Inc.

[36] Satish Kumar, “Neighborhood Pixels Weights-A New Feature Extractor”, International Journal of Computer Theory and Engineering, Vol. 2, No. 1, pp. 1793-8201(2010).