World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:12, 2011

Study of Features for Hand-printed Recognition

Satish Kumar

***Abstract*—**The feature extraction method(s) used to recognize hand-printed characters play an important role in ICR applications. In order to achieve high recognition rate for a recognition system, the choice of a feature that suits for the given script is certainly an important task. Even if a new feature required to be designed for a given script, it is essential to know the recognition ability of the existing features for that script. Devanagari script is being used in various Indian languages besides Hindi the mother tongue of majority of Indians. This research examines a variety of feature extraction approaches, which have been used in various ICR/OCR applications, in context to Devanagari hand-printed script. The study is conducted theoretically and experimentally on more that 10 feature extraction methods. The various feature extraction methods have been evaluated on Devanagari hand-printed database comprising more than 25000 characters belonging to 43 alphabets. The recognition ability of the features have been evaluated using three classifiers i.e. *k*-NN, MLP and SVM.

***Keywords*—**Features, Hand-printed, Devanagari, Classifier, Database

I. INTRODUCTION

AN OCR/ICR works in various phases such as: scanning, preprocessing, feature extraction, classification and post processing. The feature extraction phase is quite important since a set of useful properties of a character pattern available as an image is defined and extracted during this phase. On the basis of these properties, the given character pattern is assigned a label. These properties are quite significant for taking classification decision and are known as features in pattern recognition terminology. The feature extraction method(s) used must be robust for expressing the properties of a character/script under consideration. If there is slight variation in character image either due to printing, writing or due to instrument used, it should be able to absorb the same. There are many feature extraction methods available in literature. The features may be local or global. The features may be extracted from an original character or from its skeleton or contour form. Govindan et al [1] categories the various feature extraction methods in three categories i.e. statistical, structural and Global transforms and series expansion. A character image is composed of black and white pixels having two levels either 0 or 255 in binary format or gray colors having a range of 256 colors between 0 and 255 in gray format. The images may be colored but in case of document images only binary or gray formats are mostly considered. The properties may be extracted from binary images, gray images, pseudo-binary or pseudo-gray depending upon the feature extraction method used. A survey on the handwritten recognition has been carried by Plamondon et al [2], Koerich et al [3] and Arica et al [4]. The work done for recognititon of Indian language script is reported by Pal et al[6]. Devanagari handwritten isolated word recognition is done by Parui et al[49] and Shaw et al[50] and isolated character recognition is carried by Deshpande et al[48].

The hand-printed patterns come from different writers and possess great variations. The recognition of hand-printed patters is difficult as compared to machine-printed. Some factors that complicate the recognition process in hand-printed character recognition in noise-less situations are: 1). Number of classes under consideration. 2). Presence of ambiguous classes. 3). Complexity of the script under consideration. 4). Variability caused in writing the given patterns.

First three factors are script dependent whereas last factor depends upon various factors such as writer, writing situation, instruments used etc. and there is no limit to it. The complicacy in recognition is further aggravated by slant writing. A robust recognition system must be able to recognize slanted characters or words that must be straightened before applying recognition process. The main reasons for slant removal prior to perform recognition are [20], [21], [22]: 1) There is a lot of difficulty in segmenting a slanted word into characters. The slant is corrected to minimize the problem of overlapping adjacent characters, due to slant, interfering in the columns of the pixels extracted from them. 2) Excessive time is required to train slanted characters in a recognition system. 3) The accuracy of a recognition system, with slanted patterns, is affected adversely.

As such there are two ways to cover the variability caused due to slant writing in a recognition system. These ways are:1) Normalize a slanted word or character before recognition. Some methods used for this purpose are based on histogram [23], [24], chain code contour [25],[26] and dynamic programming. 2) Some researchers do not perform slant normalization. In this strategy the slant is compensated during training process [25]. The amount of training data required for covering as many slant angles as possible is large. This increase in data complicates the recognition process a lot. One way to remove this problem is to use slant invariant feature extraction methods [20]. So it is essential to find out a feature that possesses better slant invariance over other for a given script. If we use a good slant invariant feature, a majority of above mentioned problems will be solve up to a lot extant[53]. A pattern recognition model when applied to hand-printed recognition requires training the machines (systems) with handwritten samples of different persons. In order to train a machine a large number of training samples are required. If the number of samples is small, the machine may not learn

Satish Kumar is with Department of Computer Science and Applications , Panjab University Regional Centre, Hoshiarpur (Punjab). His email address is satishnotra[at]yahoo.co.in.

adequately. There is a lot of impact of size of training data set on the recognition performance of a classifier. Training a classifier with adequate number of handwritten samples enrich it with generalization ability. It is essential to know the effect of size of training dataset on the recognition performance of a feature extraction method.

A feature extraction method may be used as a standalone or may be used in combination with each other. It is essential to check the performance of various features in combination with each other. Actually, this is required to know that which features combination is better for a given script. In this paper, we tried to give the answers of the following questions: 1) Recognition performance of which feature is better? 2) Recognition performance of which feature combination is better? 3) Which feature is least training data prone?

II. FEATURES IN BRIEF

The feature extraction method(s) used to recognize hand-printed characters play an important role in ICR applications. There exist many features in literature which have been used for the recognition of various hand-printed scripts. It is essential to check the performance of a feature on a given script. Given a large number of features, it is difficult to decide which feature should be used for given application. Feature which is suited for one script may not suit other. One more important factor about a set of features used to recognize a script is its size. So, one may have to apply the various features on a script by taking large dataset and find out which feature is better and what should be its size. The variability in handwritten patterns is so large that a single feature may not do well. So, it is essential to check the performance of various features in combination and sort out which combination is suited for a given application.

A pattern recognition model when applied to hand-printed recognition requires training the machines (systems) with handwritten samples of different persons. In order to train a machine a large number of training samples are required. Some features require large training samples whereas others give good generalization even with small training sample set. So, it is essential to check which feature is large data prone and which is less data prone. Some While testing the suitability of a feature to a script, one may have to redesign the existing features so that large recognition performance may be attained with small size of feature set. Our paper responds all such question and sorts out which feature is better for our application. Some features considered here are promising and have been used primary features in many applications. The various feature extraction methods considered are zoning, projection profiles, projection histograms, crossings, distance transform, chain code histograms, directional distance distribution, gradient (Sobel operators), Kirsch directional edges and neighborhood pixels weights and Total distances in four directions. The performance of various features is tested using three classifiers i.e. *k*-NN, MLP and SVM on Devanagari hand-printed database. The comparisons of some results with the results already published [51] are also made here.

A. Profiles

The profiles based features are motivated for recognition of hand-printed numerals by Shridhar et al [16], where authors used left and right profiles only. The use of profiles as complementary feature for recognition of hand-printed pattern is carried by many authors and some authors are: Heutte et al [17], Liu et al [18] and Koerich [19]. Profiles extract the structural information of outer contour and do not provide any information about the interior structure of a character image such as loops, number of strokes, etc.

B. Histograms

Projection histogram based features are motivated by Glauberman [14], where author used these in an OCR which was hardware oriented. Moreover, this technique is also used to detect orientation in a document page or segmenting a page into lines, words and characters. In order to find the projection histograms, an image is tracked along a path from a side and the number of black pixels in that path is counted. A histogram gives the width of character strokes along a particular path (either row or column). The vertical histograms are slant invariant whereas horizontal histograms are not [15]. The histograms only give the stroke information along the given path and do not cover any other properties such as number of strokes along a path, width of each stroke, location of each stroke, etc.

C. Crossings

Crossings based methods have been used in [12], [13] for hand-printed character recognition and are generally used to detect the number of strokes presented in a character along a particular path. If this path is along the rows, then it is called horizontal crossings and if this path is along the columns, then it is called vertical 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. Crossings can be extracted from an original character as well as from its skeleton. Kim et al [12] used crossings in raw form but Arica et al [13] used median of the black pixel runs in each scan line.

D. Zoning

Zoning feature extraction method was implemented in Calera OCR system used to recognize machine-printed non-decorative fonts. Bosker [10] gives some details about zoning method used in Calera. Cao et al [11] used zoning on numeral contours where the images are divided into 4×4 zones. Zoning can be implemented on various forms of a character image such as original (solid character), character contour and character skeleton. In addition, this method can be used to recognize both gray as well as binary images. It is used to capture local properties of a character. In this feature extraction method, a character image (character bitmap / character bounding box) is mostly divided into zones. Mostly, non-overlapping zones are taken but Cao et al [11] divided the character bitmap into overlapping zones.

E. Kirsch Directional edges

The edge strength at a pixel (*x, y*) along horizontal (H), vertical (V), left-diagonal (L) and right-diagonal(R) is calculated from an image as follows[7],[9]:

$$G_H(x,y) = max(|5S_0 - 3T_0|, |5S_4 - 3T_4|) \tag{1}$$

$$G_V(x,y) = max(|5S_2 - 3T_2|, |5S_6 - 3T_6|) \tag{2}$$

$$G_R(x,y) = max(|5S_1 - 3T_1|, |5S_5 - 3T_5|) \tag{3}$$

$$G_L(x,y) = max(|5S_3 - 3T_3|, |5S_7 - 3T_7|) \tag{4}$$

Where S_i and T_i are defined as: $S_i = A_i + A_{i+1} + A_{i+2}$

$T_i = A_{i+3} + A_{i+4} + A_{i+5} + A_{i+6} + A_{i+7}$

The subscript *i* of pixels is taken as modulo 8 and ($i = 0, 1, 2, \ldots, 7$) is eight neighbors of pixel (*x, y*) starting from top-left pixel and moving clock-wise. Some authors who used/studied this feature for handwritten recognition are: Wen[27], Kim[12], Knerr et al[28] and Chao[29].

F. Gradient

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 [9],[30],[31]:

$$\Theta(x,y) = tan^{-1}\frac{G_y(x,y)}{G_x(x,y)} \tag{5}$$

$$G_x(x,y) = \frac{\partial I(x,y)}{\partial x}, \; G_y(x,y) = \frac{\partial I(x,y)}{\partial y} \tag{6}$$

The angle 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. 1(*a-b*)". The gradient based features have been successfully used for off-line handwritten numeral or character recognition by Srikantan et al[31], Liu et al [32], Liu et al [33] and Fujisawa et al [34] and for on-line character recognition by Kawamura et al [35].

G.Distance Transform

A distance transform assigns to each white pixel (background) of a binary image a value equal to its distance to the nearest black pixels (foreground) according to a defined

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

a)

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

b)

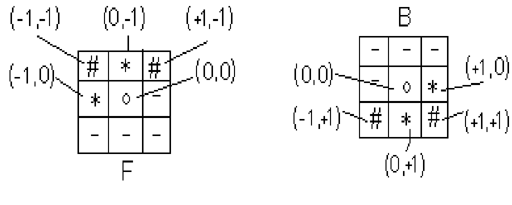
Fig. 1 Sobel operator to compute gradient component along: a) *x*-axis, b) *y*-axis.

metric. A new image, which has same size as that of an original image, is created using distance transform and this image is called as distance map (DM). In DM each background pixel has some value whereas each foreground pixel has 0 value. The DT algorithm proposed by Rosenfeld et al[44] is earliest. Borgefors[36] presented the Chamfer distance algorithm(CDA) that efficiently and accurately calculates the DT of 2 dimensional images.

It works in two passes. Initial value assigned to each location of DM as:

$$DM(x,y) = \begin{cases} 0 & if \quad I(x,y) = 1 \quad i.e. \quad black \\ val & Otherwise \end{cases} \tag{7}$$

We have used three kinds distance metrics i.e. Chamfer, Euclidean, and Chessboard in experiments. The value of symbols '#' and '*' is 4 and 3 , 1 and 1 , and $\sqrt{2}$ and 1 in case of Chamfers, Chesboard and Euclidean distance respectively The DT based features have been used/studied by Smith et al [38], Koύacs et al [39] and Oh et al[40] for handwritten recognition and Negi et al [37] for machine-printed Telugu character recognition.



(a) (b)

Fig. 2 a) Forward mask F and b) Backward mask B

H. Chain Code

Chain codes are a kind of directional codes. The codes may be 4-directional or 8-directional depending upon 4-connectivity or 8-connectivity of a pixel to its neighboring contour pixel. The direct comparison between two or more objects merely on the basis of their raw chain codes is not suitable due to some problems [42] and in this study chain code histograms [43] have been studied in contrast to raw chain codes. Chain code histogram based feature have been used by Kimura et al [43] for hand-printed numeral recognition. All possible existing contours are traced and their chain codes are produced using Freeman chain code algorithm. As already mentioned, the codes may be 4-directional or 8-directional. The 4-directional code may be obtained from 8-directional codes where the contour pixels having code 4 , 5, 6 and 7 are assigned 0, 1, 2 and 3 codes, respectively.

I. Directional Distance Distribution

In directional distance distribution (DDD) [40], the distance of a white pixel to a black or the distance of a black pixel to a white pixel in all 8 directions is taken as feature extraction criteria and all the 8 distances contribute to the feature vector. The authors [40] have used DDD in three forms no-tilling, circular-tilling and mirror-tilling.

In directional distance distribution, a ray is shot from a pixel in all eight directions and the distance traveled by a ray from a pixel to its opposite color pixels in all the 8 directions is computed. The distance traveled by a ray along all 8 directions at a given pixel is encoded using 16-cell array. The upper 8 cells of an array (at each pixel) are used if a ray is originated from a white pixel and the values of lower 8 cells are zero in this case. We have chosen circular-tilling for comparison as it is predicted best by authors. After computing direction distance distribution of all the pixels presented in an image we get 16 sub-images corresponding to one image per direction.

J. Neighborhood Pixels Weight (A New Feature)

In this feature, the weights on a pixel due to the black pixels corresponding to first, second, and third level neighborhood pixels are computed [52]. The weights on pixel (5,5), "Fig. 3(b)", due to left-top, right-top, left-bottom and right-bottom corner due to all the three level neighborhood black pixels are 0.44 , 0.56 ,0.56 and 0.11 respectively. A weight map (WM) corresponding to all the pixels for a given image is prepared. The weight map consists of four planes, each having size same as the size of normalized image.

K. Total distances in four directions(TDIST)

The size of DDD based feature vector is too large to be combined with other features whereas recognition power is quite good. To combine it with other features and avoid large dimension of final feature size, the feature is restructured. It is encoded using 4-cell per pixel where each cell contains total distances traveled by a ray for a pixel along a particular direction. The total distance for a pixel (5, 5), "Fig. 3(*a*)", along horizontal, vertical, left-diagonal and right-diagonal directions are 2, 2, 3 and 4 respectively. The total distances for background pixels are computed only. All the four cells corresponding to a black pixel are zero. This gives four TDIST planes with one plane per direction. Each TDIST plane is divided into 4×4 regions and average distance along each direction in each region is computed.

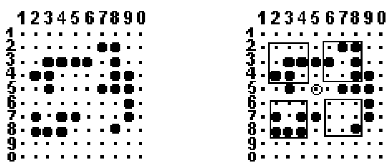


Fig. 3 a) Binary image 10×10 pixels size; b) The three layer neighborhood pixels of pixel (5, 5) on four corners

III. EXPERIMENTAL RESULTS

All the experiments have been conducted under same conditions. The slant normalization is not performed at all. The size normalization of a character image plays an important role and extracting features from a character without normalizing it to some standard size not only poses some problems in feature representation but also degrades the performance of a recognition system. The size normalization method used here is, which preserves the aspect ratio, studied by Liu et al [45]. In order to extract features, the character bitmaps or direction planes or distance maps, etc, are divided into number of regions of equal size. The size of a character image has been chosen keeping in mind the number of regions to be created. Since, in case of Devanagari characters there exists a head line at top. We suggest removing this line prior to perform size normalization, extract features and perform recognition. The algorithm used to do so has been studied in[5]. The presence of head line on the top of characters adversely affects the performance of a recognition system a lot. The size of a character image chosen for conducting experiments is 30×30, but in some cases, we have changed the size a little depending upon the requirements.

A given feature may be used in different ways. We have tested the internal as well as external performance of each feature extraction method. The internal performance is considered by conducting the experiments on various possible variations for a given feature type. A feature extraction method may be used with various variations. It is essential to know the performance of several variations of a feature type. We have conducted experiments to predict the performance of various variations of each feature type to identify which feature vector from these variations is better. A single stage classification scheme has been used in all the experiments conducted here. The experimental results using single feature and single stage classifiers with MLP, *k*-NN and SVM are given Table I.

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 in 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.

A. Feature Extraction Strategy

In Section II, we have discussed various features to be considered for examining against proposed feature. In this subsection, the strategy used to extract features to make comparison is discussed. Since the size of normalized image used is 30×30 pixels. The profiles are computed on each side, for all rows and columns, *i.e.* left, right, top and bottom and the size of feature vector used is 30×4=120. The feature is named as Pro-120. In one more instance, the character image is rotated at 45° and again all the four profiles are computed.

All the four profiles before and after rotating an image are considered. The feature vector size is 30×8=240 and feature is mentioned as Pro-240. The experiments are also conducted by taking : 1) the average of two adjoining profiles, and 2) the alternative profiles independently. In both these cases, the sizes of feature vectors are reduced by 50% but results are low as compared to Pro-120 and Pro-240 and are not reported here.

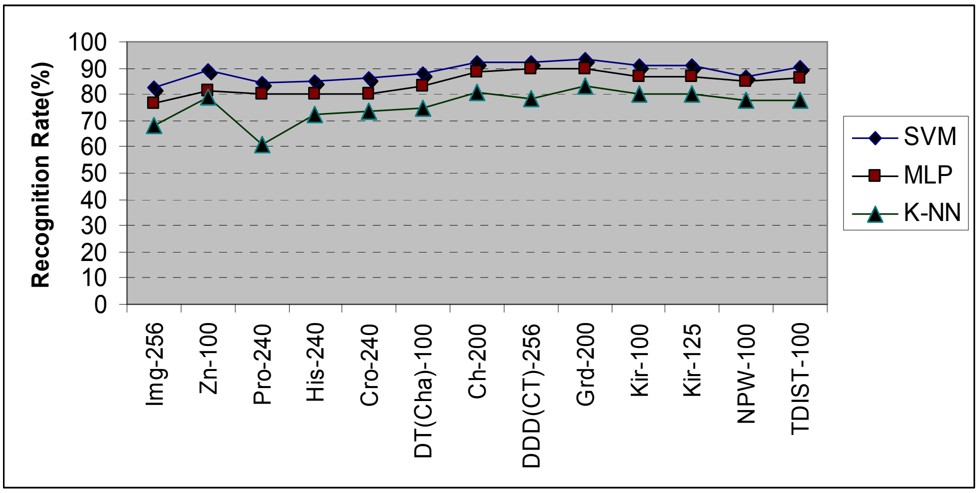


Fig. 4 Analysis of relative recognition performance of various features for three classifiers

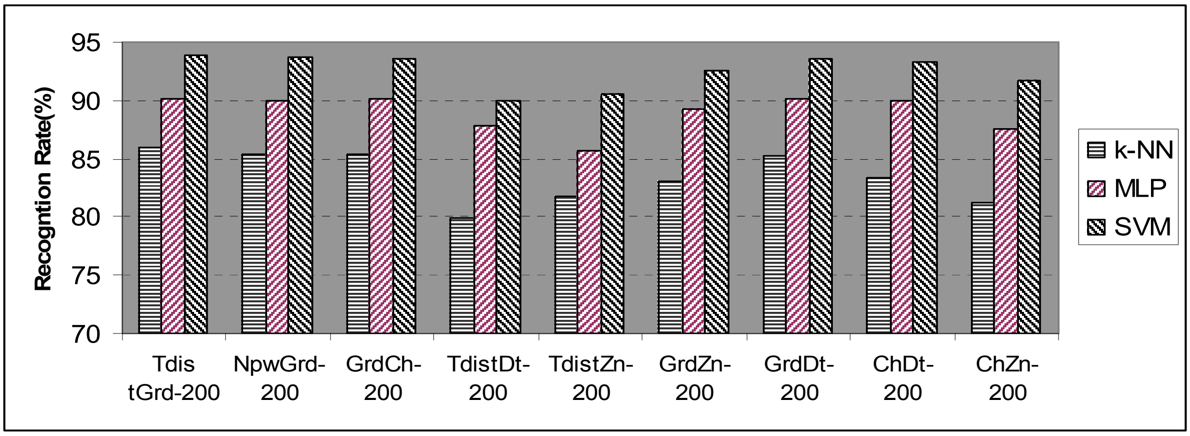


Fig. 5 Relative performance of various features in combination using three classifiers

In first case of histogram based feature, only horizontal and vertical histograms are considered. The character bitmap is size normalized to 30×30. The size of feature vector is 30+30 = 60 and feature is named as His-60. In second case, two more histograms, left diagonal and right diagonal are considered in addition to left and right histograms. Since the size of character bitmap is 30×30 and the number of possible left diagonal bins is 59. But we have considered only 49 diagonal bins and the histograms due to 10 bins (5 on one corner and 5 on another corner) are not considered. The same situation is for right diagonal histograms. The size of feature vector is 60+98=158 and feature is named as His-158. The normalized feature vector is obtained by dividing number of black pixel in each bin with the total number of pixels in each bin. The experiments are also conducted by taking the average of two adjoining bins which reduces the size of feature vector by 50%. The results are recorded low as compared to His-60 and His-158 and are not shown here.

In case of crossings, the character bitmap size chosen and number of bins taken are same as in case of projection histograms. The number of crossings in each bin is computed and the feature vector is normalized, between 0.0 and 0.1, by dividing number of crossings in each bin with the maximum value of crossings achieved out of all the bins since we have no prior knowledge about the occurrence of maximum value of crossings. The features are named as Cro-158 and Cro-60. The experiments are also conducted similar to the strategies used in [12],[13], but the results are low as compared to the results reported for Cro-158 and Cro-60.

In case of zoning, the experiments are conducted by considering different number of zones on solid normalized character image. The numbers of zones considered are 8×8, 10×10 and 12×12 and their corresponding features are named as Zn-64, Zn-100 and Zn-144 respectively. The experiments are also conducted by considering 16×16 and 14×14 zones but the recognition results are not affected. So the optimal size of number of zones is between 8×8 - 12×12. The experiments are also conducted by considering the zones on contour form of the character images. The number of zones considered are 10×10 and 12×12 and their corresponding features are named as ZnC-100 and ZnC-144. The various authors, like Oh et al [40] and Le Cun et al [47], have used or studied the performance of original binary images normalized to 16×16 size giving 256 binary feature vectors. We have also conducted experiments this way and feature is mentioned as Img-256.

The distance transform (DT) can be computed using various distance metrics. We have computed DT with three distance metrics *i.e.,* Euclidean, Chessboard and Chamfer and compared their recognition performance with each other. The character bitmap size taken is 30×30. In each case the image is convolved with 3×3 windows given in "Fig. 2(*a-b*)" and distance map (DM) is computed. The distance map is divided into 10×10 regions. The average minimum distance in each region is computed. The feature vector is normalized by dividing each feature component with maximum value of average minimum distance obtained out of all the features for a given image. The features with Chamfer, Euclidean and Chessboard are mentioned as DT(Cha)-100, DT(E)-100 and DT(Che)-100 respectively.

We have conducted some experiments to know the performance of chain code histograms based features with 8-directional codes as well as 4-directional codes. To obtain 4-directional codes from 8-directional codes, the contour pixels having directional codes 4, 5, 6 and 7 are assigned codes 0, 1, 2, and 3, respectively. The size of character image considered is 30×30. For our experiments, we have partitioned chain coded image into 5×5 regions and frequency of occurrence of each code in each region is computed. The experiments are also conducted by considering 4×4 regions, but the recognition rates are low as compared to the recognition rates given for Ch-100 and Ch-200. The features with 4-directional and 8-directional codes are mentioned as Ch-100 and Ch-200 respectively.

In case of DDD, we have conducted experiments with no-tilling and circular-tilling only. The size of character bitmap considered is 32×32. The distance of a pixel in an image in all 8 directions is encoded using 16 cell array and this gives 16 DDD planes of size 32×32 for a given image. Each DDD plane is divided into 4×4 regions. The size of feature vector is 16×4×4=256 in both experiments, *i.e.*, using circular-tilling and no-tilling. Furthermore, if we divide each DDD plane into 5×5 regions then the size of feature vector is quite large, *i.e.*, 16×5×5=400 features and it becomes very difficult to combine this feature with other features. Moreover, the large size of feature vector also increases the classification time. So, we have not considered 400 features of DDD for conducting our experiments. The features for circular-tilling and no-tilling are mentioned as DDD (CT)-256 and DDD (NT)-256 respectively. We have also conducted experiments with 3×3 regions but the results recorded are quite low as compared to the results obtained with DDD (CT)-256 and DDD(NT)-256.

In case of gradient based features, we have taken original binary images only. 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 "Fig. 1(*a*)" and 1(*b*). The gradient direction of each pixel in an image is obtained by using (5) and stored in an array which has same size as that of an image. This is also called Gradient Direction Map (GDM). The gradient direction varies from 0 to 360 degree. The gradient direction is quantized into $d3$ directional levels which produce $d3$ directional sub-images for a given image with one sub-image for each quantized directional level.

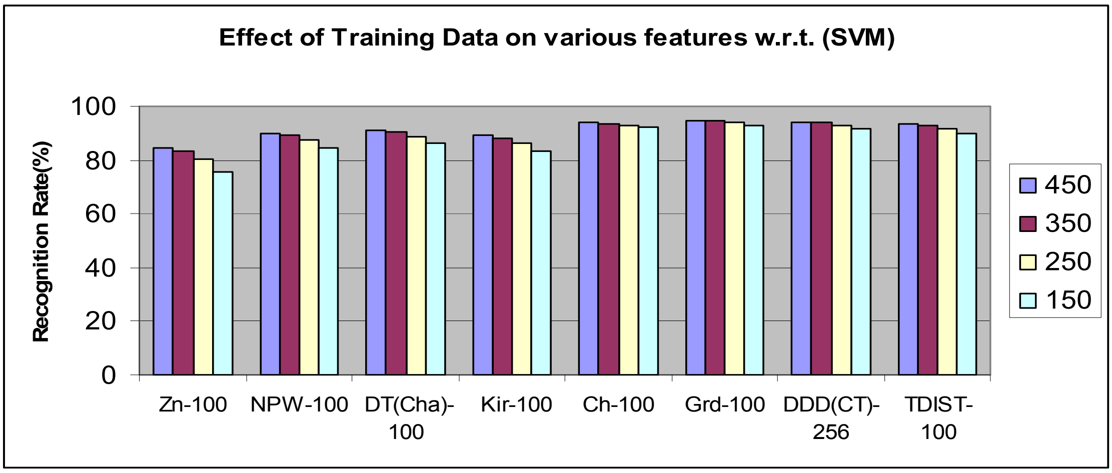


Fig. 6 Effect of training data size on recognition performance of various features and SVM classifier

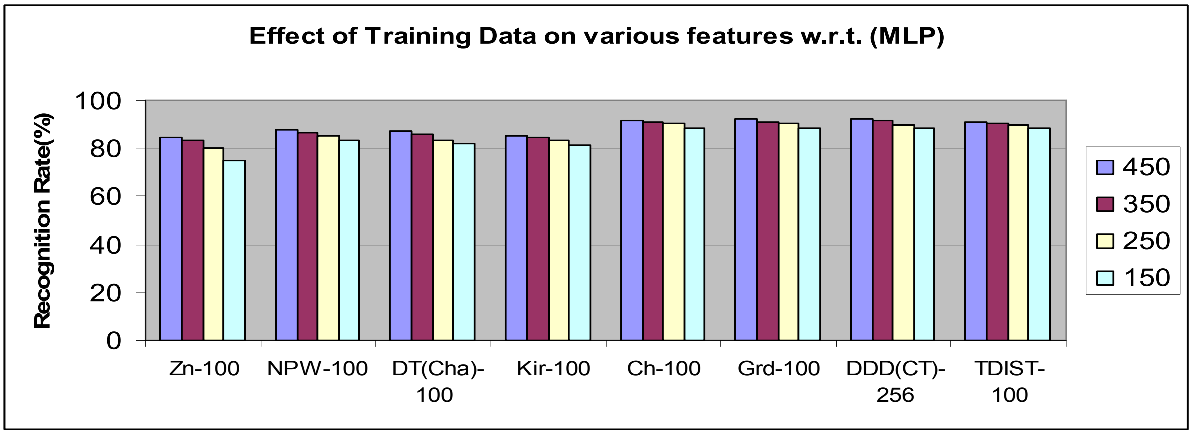


Fig. 7 Effect of training data size on recognition performance of various features and MLP classifier

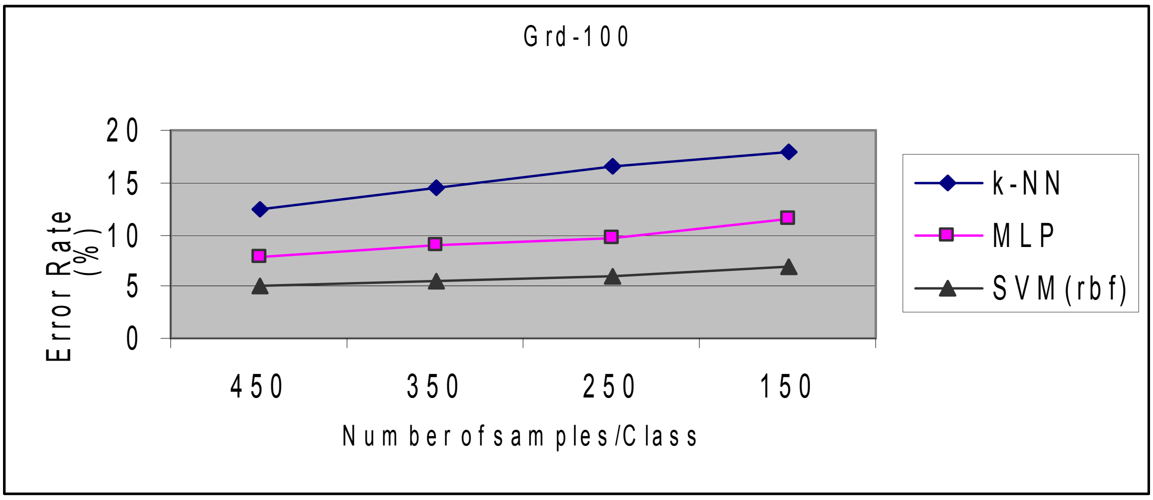


Fig. 8 Effect of training data size on error rate with *k*-NN, MLP and SVM classifier studied using gradient feature

We have conducted experiments by taking 3 different values of *d3, i.e.*, 4, 8 and 12. The size of each sub-image is same as the size of binary image, *i.e.*, 30×30. To reduce size of feature vector, each directional sub-image is divided into 5×5 regions. The features for 4, 8 and 12 directional level are mentioned as Grd-100, Grd-200 and Grd-300 respectively. The experiments are also conducted considering 3×3 and 4×4 regions but results are recoded low as compared to Grd-100, Grd-200 and Grd-300. The experiments are also conducted on pseudo-gray images with 4-directional level and feature is mentioned as Grd-100G. The strength along horizontal (H), vertical (V), left-diagonal (L) and right-diagonal(R) direction is calculated from binary image using (*1-4*). The size of character image taken is 30×30. The experiments are conducted with this feature by extracting the features from 4 sub-images available due to 4 directional edges. Each sub-image is divided into 5×5 regions giving feature vector size $4\times5\times5=100$ and feature is mentioned as Kir-100. Experiments are also conducted by taking original image in addition to 4 directional sub-images, as it is done in [12], [29], giving feature vector size $5\times5\times5=125$ and feature is mentioned as Kir-125.In case of neighborhood pixels weights(NPW), we have performed our experiments by considering the pixels of all the three levels. Here we have considered the weights on a pixel due to black pixels on all the four corners. A weight map (WM) corresponding to all the pixels in an image is prepared. The weight map consists of four planes, each having *30*×30 pixels size. 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 *5×5* regions and average weight in each region is computed and the feature is mentioned as NPW-100.

We have also investigated one simple form of DDD feature to get small size of feature vector so that it may be combined with other features. Rather than taking distance in all 8-directions and encoding the feature using 16-cell array at each pixel, we take total distance in 4-directions as explained in Subsection II(*K*) and there are four planes corresponding to an image. The distances of background pixels are considered only. Each plane is divided into 5×5 regions giving 4×5×5=100 features. This feature is mentioned as TDIST-100 feature in our study. This reduces the size of feature vector a lot making it convenient to combine this feature with other features.

B. Result Analysis

From results given in Table I, it is clear that in case of zoning, profiles, histograms, crossings and Kirsch directional edges; Zn-100, Prof-240, His-158, Cro-158 and Kir-125 performs respectively. In case of gradient (Sobel), DDD, chain code histograms and distance transform; Grd-200, DDD-(CT)-256, Ch-200 and DT (Cha)-100 performs respectively. The analysis of some performing features given in Table *I* along with Img-256, NPW-100 and TDIST-100 is given in "Fig. *(4)"*.

Among the various features studied here the gradient based feature Grd-200 is performing better for SVM and *k*-NN and Pro-240 with *k*-NN and Img-256 with all three classifiers are least performing. However, the performance of DDD (CT)-256 and Grd-200 with MLP is neck-to-neck. Also the performance of chain code with SVM and *k*-NN is better as compared to DDD(Cha)-256. It means DDD(Cha)-256 performs well with MLP as compared to *k*-NN and SVM.

The performance of MLP classifier is lying between the performance of SVM and *k*-NN classifiers. This is true for all the feature extraction methods. As far as feature extraction methods for recognition are concerned, the four methods, *i.e.* profiles, histograms, crossings and Kirsch directional edges are not performing much better. Their recognition rate is between 84-86% on Devanagari database using SVM classifier in single stage recognition scheme. Among the various feature extraction methods analyzed in our study, the features extracted from binary images normalized to 16×16 size having 256 binary feature components is least performing and its recognition rate with SVM classifier is 82.3%. The zoning, distance transform, TDIST and NPW based features are better as compared to above mentioned features. The chain code histograms, directional distance distribution and gradient (Sobel operators) based features are performing features. Among the various feature extraction methods a gradient based feature extraction method, Grd-200, is outperforming individually. Its recognition rate is 93.5% using SVM classifier.

IV. Study Of Some Significant Factors

In this section, the effects of multiple features, training set size and slant writing is studied on the recognition performance of various features. The effect of these factors has been studied to achieve a feature vector which is slant invariant, less training data prone and gives better recognition accuracy for our application.

A. Multiple Features (Single Stage Classifier)

We have also conducted experiments to know the recognition performance of various features in combination. It is not possible to conduct experiments for all combinations. We have conducted experiments for various possible feature combinations but the results obtained after combining six top features are reported here. We have chosen small sized feature vector due to each feature type to combine with other feature. The purpose is to get a feature vector of low dimension.
Although, the performance of DDD (CT)-256 feature is better as compared to TDIST-100 feature (see Table I) but we have not used this feature for combination, rather TDIST-100 feature is used, as it gives very high dimensional feature vector after combining with other features. The performance of gradient, chain code histograms, TDIST, distance transform, NPW and zoning based features has not improved much, rather degraded a lot, when they are combined with crossings, histograms, profiles and Kirsch directional edge based features. We have not listed those results here. Some results with combination of two features for recognition are given in Table II. The performance comparison of results given in Table II is made in "Fig. *(5)"*.

Consider the case of SVM classifier. The recognition rates of the recognition schemes, where zoning method combined with gradient, chain code histograms and TDIST based methods, are low. The zoning has even degraded the

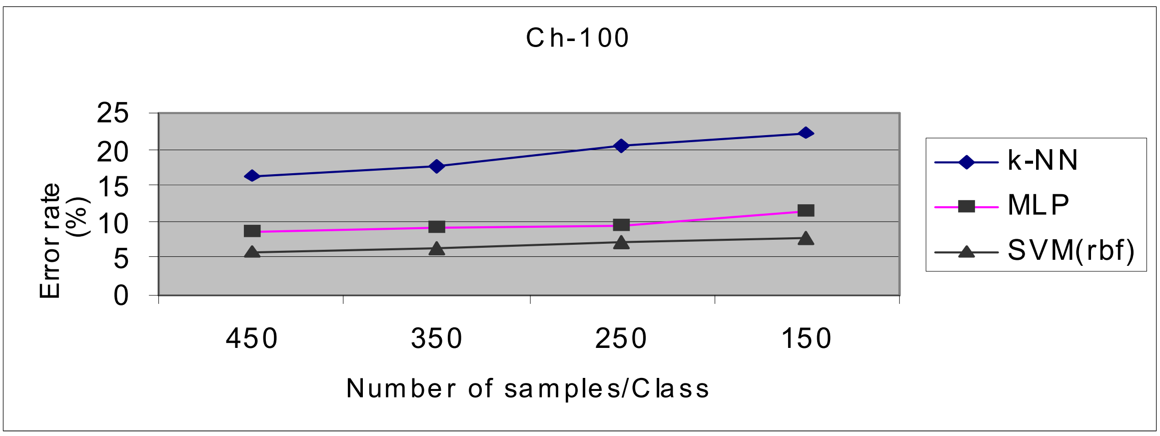


Fig. 9 Effect of training data size on error rate with *k*-NN, MLP and SVM classifier studied using chain code histograms feature

TABLE I
RESULTS OF VARIOUS FEATURES WITH THREE CLASSIFIERS

Sr.No.		Size		Recognition Results (%)		
	Feature Name	Image	Feature vector	SVM (RBF)	MLP	*k*-NN
1	Normalized Binary Image(16×16) (Img-256)	16×16	256	82.3	76.6	68.2
2	Zoning (Zn-100)	30×30	100	88.3	80.3	77.6
3	Zoning (Zn-64)	32×32	64	87.9	79.5	**78.0**
4	Zoning (Zn-144)	36×36	144	88.2	79.1	78.0
5	Zoning (ZnC-100)	30×30	100	85.0	76.0	73.0
6	Zoning (ZnC-144)	30×30	144	85.0	76.7	70.8
7	Profile (Pro-240)	30×30	240	84.4	80.2	60.6
8	Profile (Pro-120)	30×30	120	81.7	76.5	60.2
9	Histogram (His-60)	30×30	60	72.6	67.4	55.6
10	Histogram (His-158)	30×30	158	84.7	80.3	72.3
11	Crossings (Cro-158)	30×30	158	85.9	80.3	73.2
12	Crossings (Cro-60)	30×30	60	65.6	58.7	50.7
13	Chamfer (DT(Cha)-100)	30×30	100	88.1	83.3	74.9
14	Euclidean (DT(E)-100)	30×30	100	88.1	82.6	74.8
15	Chessboard (DT(Che)-100)	30×30	100	87.5	82.8	74.8
16	Chain Code Histograms(Ch-100)	30×30	100	92.2	88.7	81.1
17	Chain Code Histograms(Ch-200)	30×30	200	92.4	88.7	80.9
18	Directional Distance Distribution (DDD(CT)-256)	32×32	256	92.0	**89.7**	78.6
19	Directional Distance Distribution No-Tilling (DDD(NT)-256)	32×32	256	92.0	88.8	78.0
20	Gradient(Sobel) (Grd-200)	30×30	200	**93.5**	**89.6**	83.3
21	Gradient(Sobel) (Grd-100G)	30×30	100	92.5	88.6	85.2
22	Gradient(Sobel) (Grd-100)	30×30	100	93.2	88.9	84.6
23	Gradient(Sobel) (Grd-300)	30×30	300	**93.5**	89.5	81.9
24	Kirsch Edge+ Original(Kir-125)	30×30	125	90.8	86.9	80.3
25	Kirsch(Kir-100)	30×30	100	86.0	82.2	64.8
26	Neighborhood Pixels Weights(NPW-100)	30×30	100	86.9	85.0	77.5
27	Total Distance in 4-directions(TDIST-100)	30×30	100	90.5	86.4	77.9

performance of these three features. On the other hand, the performance of recognition system improved when a DT based feature combined with gradient or chain code histograms based features but when a DT based feature combined with TDIST, the performance is not improved. The recognition performance is also not improved much as gradient based feature combined with chain code histograms based feature.

The performance is improved when gradient based feature, Grd-100, combined with neighborhood pixels weights based feature NPW-100 or TDIST based feature TDIST-100. The top three combinations using SVM classifier are: TdistGrd-200, NpwGrd-200 and GrdCh-200 which have achieved recognition rates 93.9%, 93.7% and 93.6 %, respectively using SVM classifier. With MLP classifier, the top three combinations are: TdistGrd-200, GrdDt-200, NpwGrd-200 and their recognition rates are 90.1%, 90.1% and 90.0%, respectively.

The increase in performance in these three cases is due the fact that the gradient and chain code histograms based features extract the properties of a character image based on direction of pixels on outer contour or near to contour and these features are also called as directional features whereas distance transform and directional distance distribution based features extract the properties of the character image on the basis of the distance of a pixel from its opposite color pixel in all directions and these features are also called as distance based features. Hence, directional and distance based features contribute different information for discrimination. This behavior can be observed in case of TdistGrd-200 and GrdDt-200 combinations, where recognition rates have improved a lot. But in case of TdistDt-200 combination the performance has not improved much as TDIST and DT are distance based features. This is not only true for SVM but also for MLP and *k*-NN classifiers. The performance is also not improved much in case of chain code histograms and gradient based combination as both these features are direction based and contribute same kind of information to resultant feature vector. Also, projection profiles, histograms, crossings and Kirsch directional edge based features are not good to be used as supporting features for final recognition. The good choice in combination is TdistGrd-200 and NpwGrd-200 features for all the three classifiers. Though, the NPW-100 feature is not performing much when used alone but in combination it is performing much better as compared to some other features studied here.

B. Effect of Training Set Size

The training set size affects the recognition performance of a recognition system a lot. But a question arises, whether the

TABLE II
EXPERIMENTAL RESULTS WITH MULTIPLE FEATURES

Name of Feature Extraction Methods	Size of Feature Vector	Classifiers	Recognition Rate (%)				
			A	B	C	D	Average
TDIST-100+Grd-100 (TdistGrd-200)	200	*k*-NN,	89.0	84.2	83.5	87.1	**85.9**
		MLP	92.5	87.9	88.1	91.8	**90.1**
		SVM(RBF)	95.3	92.2	93.0	95.3	**93.9**
NPW-100+Grd-100 (NpwGrd-200)	200	*k*-NN,	88.0	83.9	83.3	86.5	85.4
		MLP	93.0	87.5	87.7	91.7	90.0
		SVM(RBF)	95.3	92.3	92.6	94.8	93.7
GRD-100+Ch-100 (GrdCh-200)	200	*k*-NN,	87.9	84.0	83.3	86.6	85.4
		MLP	92.9	87.9	87.8	91.0	89.9
		SVM(RBF)	95.0	92.2	92.5	94.7	93.6
TDIST-100+DT(Cha)-100 (TdistDt-200)	200	*k*-NN,	83.5	78.0	77.1	81.2	79.9
		MLP	90.6	85.1	86.1	89.6	87.8
		SVM(RBF)	93.6	88.8	89.1	92.8	91.1
TDIST-100+Zn-100 (TdistZn-200)	200	*k*-NN,	84.2	80.5	78.7	83.8	81.8
		MLP	88.7	83.8	82.2	87.6	85.6
		SVM(RBF)	92.9	88.7	88.8	91.7	90.5
Grd-100+Zn-100 (GrdZn-200)	200	*k*-NN,	85.2	82.2	80.6	84.3	83.1
		MLP	92.1	87.1	87.2	90.5	89.2
		SVM(RBF)	94.4	90.9	90.8	94.3	92.6
Grd-100+DT(Cha)-100 (GrdDt-200)	200	*k*-NN,	88.0	83.6	83.0	86.4	85.2
		MLP	92.6	88.3	88.0	91.6	**90.1**
		SVM(RBF)	95.1	92.3	92.2	95.0	93.6
Ch-100+DT(Cha)-100 (ChDt-200)	200	*k*-NN,	86.3	80.6	81.2	85.1	83.3
		MLP	92.7	87.9	87.5	91.6	89.9
		SVM(RBF)	95.1	91.7	92.1	94.2	93.3
Ch-100+Zn-100 (ChZn-200)	200	*k*-NN,	83.5	80.2	78.8	82.4	81.2
		MLP	90.5	85.1	85.3	89.6	87.6
		SVM(RBF)	93.6	90.4	90.1	92.6	91.7

discrimination ability of all the feature extraction methods is affected equally or there is a little effect on the performance of some features, with small training sample set size, as compared to others? The performance of which feature type is least or mostly effected? To know the effects of training sample size on discrimination ability of various features and classifiers, we have conducted experiments by taking different sizes of training sample set. For testing purpose all 150 samples/class from test samples of set A are taken. The classifiers are trained on different sizes of training data taken from training subsets B, C and D. The experimental results are given Table III. The analysis of effect of size of training data set on the recognition rate of various features using SVM and MLP classifiers are given in “Fig. (6)” and “Fig. (7)”, respectively.

From “Fig. (6-7)”, it is clear that the size of training data set affects the recognition rate of various features a lot but it is not uniform in case of all feature extraction methods. The performance of some features has a lot of effect whereas some has very small effect. This effect is large on Zn-100, Kir-100, DT(Cha)-100 and NPW-100 and small on Ch-100, DDD(CT)-256(except in case of MLP), TDIST-100 and least on Grd-100 feature. It is very high in case of Zn-100. It means that zoning, Kirsch direction edge, distance transform and neighborhood pixels weights based features are more training data prone whereas other features are less.

The zoning is highly training data prone feature. The error rate on DDD(CT)-256 and Grd-100 using SVM classifier with 450 sample/class are 5.9% and 5.1% , respectively and grows to 8.3% and 7.0% as the number of samples/class are reduced to 150. It means gradient based feature is less data prone as compared to DDD based. Similarly, the effect of training sample size on the recognition performance of various classifiers using two features, gradient and chain code histograms, has been analyzed in “Fig. (8)” and “Fig. (9)”, respectively.

From “Fig. (8-9)”, it is clear that the error rate of *k*-NN with small sample size is larger as compared to other two classifies, *i.e.*, MLP and SVM. The error rate with 450 samples/class using chain code histograms and *k*-NN combination “Fig. (9)” is 17.8% and grown to 22.2% when samples are reduced to 150 per class. The difference in error rates is 5.8%. This difference is less for MLP and SVM classifiers, which is 2.9% and 1.9%, respectively. The same is true for gradient based feature “Fig. (8)” for all the three classifiers. It means that *k*-NN based classifier is more training data prone as compared to MLP and SVM. The SVM classifier is least training data dependent as its performance is effecting not much as compared to other two classifiers on reducing the size of training data.

TABLE III
EFFECT OF TRAINING SAMPLE SET SIZE ON RECOGNITION PERFORMANCE OF VARIOUS FEATURES

Name of Feature Extraction method	Size of Feature vector	Recognition Rate (%) (Tested on 150 samples /class, test set A)				
		Classifiers	Number of Training Samples/class 450	350	250	150
Profile (Pro-240)	240	*k*-NN,	64.9	62.6	60.3	58.2
		MLP	84.9	83.8	82.5	78.8
		SVM(RBF)	87.9	87.4	85.6	82.4
Histogram (His-158)	158	*k*-NN,	75.7	73.6	70.1	67.3
		MLP	84.1	83.5	81.6	78.9
		SVM(RBF)	87.8	86.5	84.4	81.6
Crossings (Cro-158)	158	*k*-NN,	77.4	75.8	72.3	70.0
		MLP	84.5	78.1	76.2	74.1
		SVM(RBF)	88.5	87.9	87.0	85.4
Zoning (Zn-100)	100	*k*-NN,	79.9	78.4	76.1	73.6
		MLP	84.8	83.2	80.1	75.3
		SVM(RBF)	90.7	90.1	88.3	85.6
Distance Transform (DT(Cha)-100)	100	*k*-NN,	79.3	77.7	75.5	73.1
		MLP	87.1	85.7	83.3	82.1
		SVM(RBF)	90.8	90.2	88.9	86.2
Kirsch Directional Edges (Kir-100)	100	*k*-NN,	67.9	65.9	62.7	61.4
		MLP	85.3	84.5	83.4	81.2
		SVM(RBF)	89.0	87.8	86.2	83.6
Chain Code Histograms (Ch-100)	100	*k*-NN,	83.6	82.2	79.5	77.8
		MLP	91.4	90.8	90.5	88.5
		SVM(RBF)	94.0	93.6	92.8	92.1
Gradient (Sobel) Grd-100	100	*k*-NN,	87.7	85.6	83.5	82.1
		MLP	92.2	91.1	90.4	88.5
		SVM(RBF)	94.9	94.4	94.1	93.0
Directional Distance Distribution (DDD(CT)-256)	256	*k*-NN,	82.0	80.6	77.9	75.5
		MLP	92.0	91.5	89.7	88.4
		SVM(RBF)	94.1	93.8	92.6	91.7
Neighborhood Pixels Weights (NPW-100)	100	*k*-NN,	80.4	78.9	75.9	74.1
		MLP	87.7	86.4	85.1	83.5
		SVM(RBF)	90.1	89.1	87.3	84.4
Total Distance in 4-directions (TDIST-100)	100	*k*-NN,	81.5	80.3	77.7	75.6
		MLP	91.1	90.4	89.6	88.4
		SVM(RBF)	93.3	92.9	91.6	89.6

C. *Time Factor*

From Table IV, it is clear that the classification time using SVM classifier is more than 19 times as compared to MLP classifier in single stage for our application containing 43 classes. However the increase in feature extraction time is about 1.0-1.5 ms if we combine two features. The recognition performance of SVM classifier is far better as compared to MLP classifier. It means we should use SVM classifier but we must have to sort out a way to reduce the classification time. A strategy to do so has been made on Devanagari [54].

V. DISCUSSION AND CONCLUSION

In this paper the recognition performance of more than ten feature extraction methods is presented in context to Devanagari hand-printed character recognition using three classifiers *i.e.* *k*-NN, MLP and SVM. The performance of MLP classifier is lying between the performance of SVM and *k*-NN classifiers. This is true for all the feature extraction methods. However, the classification time of SVM is very high as compared to MLP. As far as feature extraction methods for recognition are concerned, the four methods, *i.e.*, profiles, histograms, crossings and Kirsch directional edges are not performing much better. Their recognition rate is between 84-86% on Devanagari database using SVM classifier in single stage recognition scheme. Among the various feature extraction methods analyzed in our study, the features extracted from binary images normalized to 16×16 size having 256 binary feature components is least performing and its recognition rate with SVM classifier is 82.3%. The zoning, distance transform, TDIST and NPW based features are better as compared to above mentioned features. Chain code histograms, directional distance distribution and gradient (Sobel operators) based features are performing features. Among the various feature extraction methods a gradient based features, Grd-200, is outperforming individually. Its recognition rate is 93.5% using SVM classifier. In overall, the gradient based feature is outperforming the other features, studied here, in majority of aspects. In order to enhance the recognition performance of a recognition system, various authors have suggested using multiple features. We have also combined various features to know which feature combination is better for our application. The profiles, histograms, crossings, zoning, Kirsch directional edges based features degrade the performance of chain code histograms, gradient, TDIST, NPW and DT based features when combined with them. When zoning, chain code histograms, gradient, TDIST, NPW and distance transform based features are combined with each other, the performance in case of some combination is not affected much. The top three combinations, using SVM classifier in single stage, are TdistGrd-200, NpwGrd-200 and GrdCh-200 which have achieved recognition rates 93.9%, 93.7% and 93.6 %, respectively. With MLP classifier, the top three combinations are TdistGrd-200, GrdDt-200, NpwGrd-200 and their recognition rates are 90.1%, 90.1% and 90.0%, respectively. The recognition performance of a DDD based feature, DDD (CT)-256, has been least affected as test samples are slanted either to left or to right as compared to gradient and chain code histograms based features. The effect of taking less training samples is large on zoning, Kirsch directional edge, distance transform and less on chain code histograms, directional distance distribution (except in case of MLP), TDIST and NPW and least on gradient based features. The effect of small training data size on recognition rate of SVM classifier is least. This effect is large in case of MLP and very large on *k*-NN classifiers. There is small decrease in recognition rate in case of SVM based classifier for all feature types. This effect in decrease in recognition rate is large in case of MLP and larger in case of *k*-NN based classifiers for all feature extraction methods. The recognition performance of SVM classifier is far better as compared to MLP. Since, the classification time using SVM classifier is more than 19 times as compared to MLP classifier in single stage for our application containing 43 classes. It means we should use SVM classifier but we must have to sort out a way to reduce the classification time without compromising recognition rate.

REFERENCES

[1] V. K. Govindan and A. P. Shivaprasad, "Character Recognition – A Review", Pattern Recognition, Vol. 23, No. 7 , 1990.
[2] 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.
[3] 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.
[4] 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.
[5] Satish Kumar, "The Headline Removal Algorithm and its Effect on Recognition of Devanagari Handwritten Characters", International Journal of Systemics, Cybernetics and Informatics, April 2009.
[6] U. Pal, B.B. Chaudhuri, "Indian script character recognition: a survey", Pattern Recognition, 37, pp. 1887-1899, 2004.
[7] R. Kirsch, "Computer Determination of the Constituent Structure of Biomedical Images," Computers and Biomedical Research, Vol 4, pp. 315–328, 1971.
[8] 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.
[9] W. K. Pratt, "Digital Image Processing", Third Edition, Wiley, New York , 2001.
[10] M. Bosker, "Omnidocument Technologies", Proceedings of the IEEE, Vol. 80, No. 7 , 1992.
[11] J. Cao, M. Ahmadi and M. Shridhar, Recognition of Handwritten Numerals with Multiple Feature and Multistage Classifier, Pattern Recognition, Vol. 28, No. 2, pp. 153-160, 1995.
[12] K. M. Kim, J.J. Park, Y.G. Song, I. C. Kim and C. Y. Suen, "Recognition of Handwritten Numerals Using a Combined Classifier with Hybrid Features", SSPR & SPR, LNCS 3138, pp. 992-1000, 2004.
[13] N. Arica and F. T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 6, 2002.
[14] M. H. Glauberman, "Character Recognition for Business Machines", Electronics, pp. 132-136, 1956.
[15] 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.
[16] M. Shridhar and A. Badreldin, "Recognition of Isolated and Connected Handwritten Numerals", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 142-146, 1984.
[17] L. Heutte, T. Paquet, J. Moreau, Y. Lecourtier and C. Olivier, "A Sturctural / Statistical Feature Based Vector for Handwritten Character Recognition", Pattern Recognition Letter, Vol. 19, pp. 629-641, 1998.
[18] C.- L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-Art", Pattern Recognition, No. 36, pp. 2271-2285 , 2003.
[19] L. Koerich, "Large Vocabulary Off-line Handwritten Word Recognition", Ph. D. Thesis, École de Technologie Supérieure, Montreal-Canada , 2004.
[20] R. M. Bozinovic and S. N. Srihari, Offline Cursive Script Word Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 1, pp. 68-83 , 1989.
[21] A. D. S. Britto Jr., R. Sabourin, E. Lethelier, F. Bortolozzi and C. Y. Suen, "Improvement in Handwritten Numeral String Recognition by Slant Normalization and Contextual Information", Proceedings of the Seventh International Workshop on Frontiers of Handwriting Recognition, Amsterdam-Netherlands, pp. 323-332 , 2000.
[22] E. Kavallieratou, N. Fakotakis and G. Kokkinakis, "Slant Estimation Algorithm for OCR Systems", Pattern Recognition, Vol. 34, pp. 2515-2522, 2001.
[23] D. Guillevic and C. Y. Suen, "Cursive Script Recognition: A Sentence Level Recognition Scheme", Proceedings of International Workshop on Frontiers of Handwriting Recognition, pp. 216-223 , 1994.
[24] G. Nicchiotti and C. Scagliola, "Generalized Projections: A Tool for Cursive Handwriting Normalization", Proceedings of the Fifth International Conference on Document Analysis and Recognition, Bangalore, India, pp. 729-732 , 1999.
[25] S. Madhvanath, G. Kim and V. Govindaraju, "Chaincode Contour Processing for Handwritten Word Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 9, pp. 928-932 , 1999.
[26] F. Kimura, Y. Miyake, and M. Shridhar, "Handwritten ZIP Code Recognition using Lexicon Free Word Recognition Algorithm", International Conference on Document Analysis and Recognition, Montreal, Que., Canada, pp. 906-910 , 1995.
[27] 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.
[28] S. Knerr, L. Personnaz and G. Dreyfus, "Handwritten Digit Recognition by Neural Networks with Single-Layer Training", IEEE Transactions on Neural Networks, Vol. 3, No. 6, pp. 962-968, 1992.
[29] S.-B. Cho, "Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals", IEEE Transactions on Neural Networks, Vol. 4, No. 1, pp. 43-53, 1997.
[30] L. S. Davis, "Survey of Edge Detection Techniques", Computer Graphics and Image Processing, Vol. 4, pp. 248-270, 1975.
[31] 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.
[32] C.- L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-Art", Pattern Recognition, No. 36, pp. 2271-2285 , 2003.
[33] H. Liu and X. Ding, "Handwritten Character Recognition using Gradient Feature and Quadratic Classifier with Multiple Discrimination Schemes", Proceedings of the Eighth International Conference on Document Analysis and Recognition, pp. 19-25, 2005.
[34] H. Fujisawa and C.–L. Liu, "Directional Pattern Matching for Character Recognition Revisited", Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp. 794-798, 2003.
[35] A. Kawamura, et al, "On-line Recognition of Freely Handwritten Japanese Characters using Directional Feature Densities", Proceedings of the Eleventh International Conference on Pattern Recognition, Vol. II, pp. 183-186, 1992.
[36] G. Borgefors, "Distance Transformations in Digital Images", Computer Vision, Graphics and Image Processing, Vol. 34, pp. 344-371, 1986.
[37] A. Negi, C. Bhagvati and B. Krishna, "An OCR System for Telugu", Proceedings of the Sixth International Conference on Document Processing, pp. 1110-1114, 2001.

[38] S. J. Smith, M. O. Bourgoin, K. Sims and H.L. Voorhees, "Handwritten Character Classification using Nearest Neighbor in Large Database", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.16, No. 9, 915-919, 1994.

[39] Zs. M. Kovics and R. Guerrieri, "Massively-Parallel Handwritten Character Recognition Based on the Distance Transform", Pattern Recognition, Vol. 28, No. 3, pp. 293-301, 1995.

[40] 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.

[41] H. Freeman, "Computer Processing of Line Drawings", Computing Surveys, Vol. 6, pp. 57-97, 1974.

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

[43] F. Kimura and M. Shridhar, Handwritten Numeral Recognition Based on Multiple Algorithms, Pattern Recognition, Vol. 24, No. 10, pp. 969-983,1991.

[44] A. Rosenfeld and J. L. Pfaltz, "Distance Functions on Digital Pictures", Pattern Recognition, Vol. 1, No. 1, pp. 33-61, 1968.

[45] 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.

[46] N. Arica and F. T. Yarman-Vural, "Optical Character Recognition for Cursive Handwriting", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 6, 2002.

[47] Y. Le Cun, O. Mattan, B. Boser, J.S. Denker et al, "Handwritten Zip Code Recognition with Multilayer Networks", Proceedings of International Conference on Pattern Recognition, Atlantic City, USA, Vol. 2, pp. 35-40, 1990.

[48] P. S. Deshpande, L. Malik and S. Arora, Journal of Computers, Vol. 3, No. 5, pp. 11-17, May 2008.

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

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

[51] Satish Kumar, "Performance Comparison of Features on Devanagari Hand-printed Dataset", International Journal of Recent Trends in Engineering, Vol. 1, No. 2, pp. 33-37, May 2009.

[52] Satish Kumar, "Neighborhood Pixels Weights-A New Feature Extractor", International Journal of Computer Theory and Engineering, Vol. 1, No. 6, pp. 69-77, Feb 2010.

[53] Satish Kumar, " A Study of Discrimination Ability of Features on Hand-printed Characters in Context to Noisy and Slanted Patterns" , Punjab Institute of Management and Technology(PIMT) Journal of Research, Vol. 2, No. 1, pp. 66-72, March 2009.

[54] Satish Kumar, "Devanagari Hand-printed Character Recognition using Multiple Features and Multi-stage Classifier", International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), Vol. 2, pp.039-055, 2010.