Journal of Theoretical and Applied Information Technology
28th February 2014. Vol. 60 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HANDWRITTEN DEVNAGARI DIGIT RECOGNITION: BENCHMARKING ON NEW DATASET

[1]RAJIV KUMAR, [2]KIRAN KUMAR RAVULAKOLLU

[1,2]Sharda University, Department of Computer science & Engineering

E-mail: [1]rajivbec@gmail.com, [2]kiran.ravulakollu@sharda.ac.in

ABSTRACT

The paper presents handwritten Devnagari digit recognition results for benchmark studies. To obtain these results, we conducted several experiments on CPAR-2012 dataset. In these experiments, we used features ranging from the simple most features (direct pixel values), slightly computationally expensive, profile based features, to more complex gradient features extracted using Kirsch and wavelet transforms. Using these features we have measured recognition accuracies of several classification schemes. Among them the combined gradient and direct pixel feature using KNN classifier yielded the highest recognition accuracy of 95.2 %. The recognition result was improved to 97.87% by using multi stage classifier ensemble scheme. The paper also reports on the development of CPAR-2012 dataset that is being developed for *Devnagari* optical document recognition research. Presently, it contains 35,000 (15,000 constrained, 5,000 semi-constrained and 15,000 unconstrained) handwritten numerals, 82,609 handwritten isolated characters, 2,000 unconstrained and 2,000 constrained pangram text, and 4,000 digitized data collection forms.

Keywords: *CPAR-2012 dataset, Devnagari digit recognition, neural network classifier, majority voting, shape similar digits.*

1. INTRODUCTION

This paper reports on isolated handwritten *Devnagari* digit recognition. Hindi language, the National Language of India, uses *Devnagari* script. Besides *Hindi*, a number of languages: *Sanskrit*, *Konkani*, *Marathi*, *Nepali*, *Bhojpuri*, *Guajarati*, *Pahari* (*Garhwali* and *Kumaoni*), *Magahi*, *Maithili*, *Marwari*, *Bhili*, *Newar*, *Santhali*, *Tharu* and sometimes *Sindhi*, *Dogra*, *Sherpa*, *Kashmiri* and *Punjabi* uses *Devnagari* script. This research is an attempt towards the development of computerized system for digital processing of *Devnagari* script based documents. The focus of our research is, particularly, on the development of a reliable recognition system that can recognize reliably handwritten isolated digits that are captured in real-life environment. Such a recognition system can solve data entry problem, a bottleneck for data processing applications, by capturing data at their sources. Such systems for applications, involving Latin characters, like computer processing of: postal address [1], bank cheques [2] and historical records [1,3] are being built & deployed, and such systems are very much desired for *Devnagari* script based applications such as *Aadhar card* [4] and commercial forms [1].

The objective of this paper is two-fold. One is to study the effectiveness of a set of handwritten *Devnagari* digit recognition schemes another is to publish findings of this research for benchmarking. To meet these objectives we have created a standard dataset that we referred to as CPAR-2012 [30] dataset. To the best of our information, no standard dataset is available−at least in public domain−for benchmark studies for *Devnagari* script based document recognition. Unfortunately, research groups in *Devnagari* script recognition have paid very little attention to the importance of dataset. In most of the cases [2-6, 9, 26], they tested their algorithms on artificially created dataset or on unprofessionally collected samples of size less than 3,000 characters. This might be a reason for very limited progress in *Devnagari* script recognition as compared to Latin languages. To overcome the lacunae, we are developing CPAR-2012 dataset of *Devnagari* script for benchmark studies. Unlike the existing datasets, this dataset is much larger in size, type and other attributes than the largest dataset of 23,392 digit samples reported in [1]. Moreover, it contains colour images of handwritten numerals, characters and texts along with writers' information that provide information for handwriting analysis based application development.

We describe our test data attributes, along with its collection process, in section 2. We have tested recognition accuracies of different recognition schemes on 35,000 digit samples. These recognition schemes were developed by combining five feature types and five classifiers. We have created feature-wise and classifier-wise schemes. For feature-wise comparison, we created schemes to study the performance of single classifier on all feature types while for classifier-wise comparison, we used different classifiers to classify the same feature vector.

In addition, we have also tested the accuracies of majority voting based classifier ensemble schemes. We present feature definition, extraction process, individual classifier and classifier ensemble details in section 3, experimentation description in Section 4 and conclusion in Section 5.

2. DATASET FOR DEVNAGARI DIGIT

Devnagari script has more than 300 character-shapes. Among them, there are 13 vowels, 33 consonants, 3 composite consonant and 10 digits. Figure 1 shows *Devnagari* digit shapes. Although, these shapes are unique but *Devnagari* digit one resembles *English* digit nine (9) and *Devnagari* digit 9 has two different shapes: thus, there are11 distinct shapes.

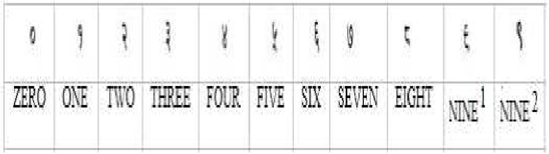
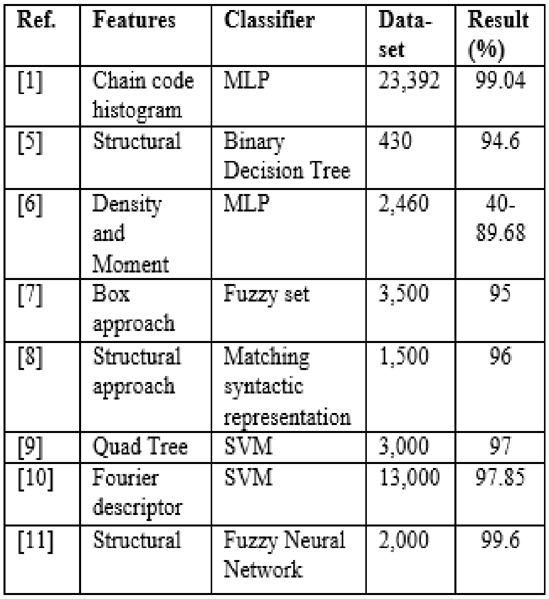


Figure 1: Devnagari Digits

The research works for developing a handwritten *Devnagari* digits (also referred to as numerals) recognition are going on for the past four decades. Since then, several recognition techniques have been developed and tested [1, 5-11]. Table-1 below shows among the best schemes that have been reported in handwritten *Devnagari* digit recognition literature. We find these schemes incomparable. For reasons, researchers tested their techniques on different test datasets−each dataset is of different and distinct quality, e.g., having shape and size variations.

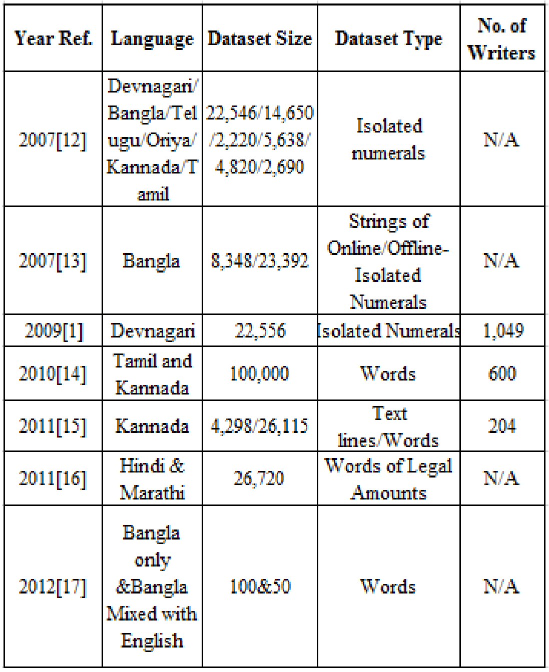
As mentioned before, for comparison or benchmark studies, a standard dataset is required. However, there are some progresses in this direction. Table-2 gives a summary of such contributions. It indicates that after 2000 Pal *et al*. [12] reported the existence of the first test dataset of *Devnagari* digits. Later, Bhattacharya & Chaudhary [1] and Jayadevan *et al*. [16] reported about their text datasets. In the meantime, to deal with this important issue, the Centre of Excellence for Document Analysis and Recognition (CEDAR) has launched an ambitious project of a test dataset collection for *Devnagari* script recognition [23]. These datasets are not available in public domain. Therefore, we started developing CPAR-2012 dataset.

Table 1: Devnagari Handwritten Digit Recognition Schemes.



Ref.	Features	Classifier	Data-set	Result (%)
[1]	Chain code histogram	MLP	23,392	99.04
[5]	Structural	Binary Decision Tree	430	94.6
[6]	Density and Moment	MLP	2,460	40-89.68
[7]	Box approach	Fuzzy set	3,500	95
[8]	Structural approach	Matching syntactic representation	1,500	96
[9]	Quad Tree	SVM	3,000	97
[10]	Fourier descriptor	SVM	13,000	97.85
[11]	Structural	Fuzzy Neural Network	2,000	99.6

Table 2: Datasets for digit recognition research for Indian languages.



Year Ref.	Language	Dataset Size	Dataset Type	No. of Writers
2007[12]	Devnagari/Bangla/Telugu/Oriya/Kannada/Tamil	22,546/14,650/2,220/5,638/4,820/2,690	Isolated numerals	N/A
2007[13]	Bangla	8,348/23,392	Strings of Online/Offline-Isolated Numerals	N/A
2009[1]	Devnagari	22,556	Isolated Numerals	1,049
2010[14]	Tamil and Kannada	100,000	Words	600
2011[15]	Kannada	4,298/26,115	Text lines/Words	204
2011[16]	Hindi & Marathi	26,720	Words of Legal Amounts	N/A
2012[17]	Bangla only &Bangla Mixed with English	100&50	Words	N/A

2.1. CPAR-2012 Dataset

The CPAR-2012 dataset contains images of constrained, semi-constrained and unconstrained handwritten numerals; isolated characters; unconstrained and constrained pangram text; digitized data collection forms. The pangram text has 13 most frequently used vowels, 14 modifiers and 36 consonants. In addition to these, it contains writer information needed for writer identification and handwriting analysis research.

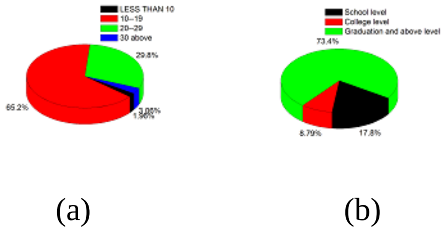


Figure 2: CPAR-2012 dataset division (a) Age group (b) Education wise

The novelty of the dataset is that it is the largest test dataset for *Devnagari* script based document recognition research. The data reflects the maximum handwriting variations as it is sampled from writers belonging to diverse population strata. They belonged to different age groups (from 6 to 77 years), gender, educational backgrounds (from 3rd grade to post graduate levels), professions (software engineers, professors, students, accountants, housewives and retired persons), regions (Indian states: Bihar, Uttar Pradesh, Haryana, Punjab, National Capital Region (NCR), Madhya Pradesh, Karnataka, Kerala, Rajasthan, and countries: Nigeria, China and Nepal). Two thousand writers participated in this experiment. Figure 2 (a-b) show age groups and education levels writer's distribution.

To collect the data, we designed two forms: Form-1 (see Figure 3 (a) to collect the isolated digits, characters, and writer's information and Form-2 (see Figure 3 (b)) to collect the constrained and unconstrained handwritten words from the pangram.

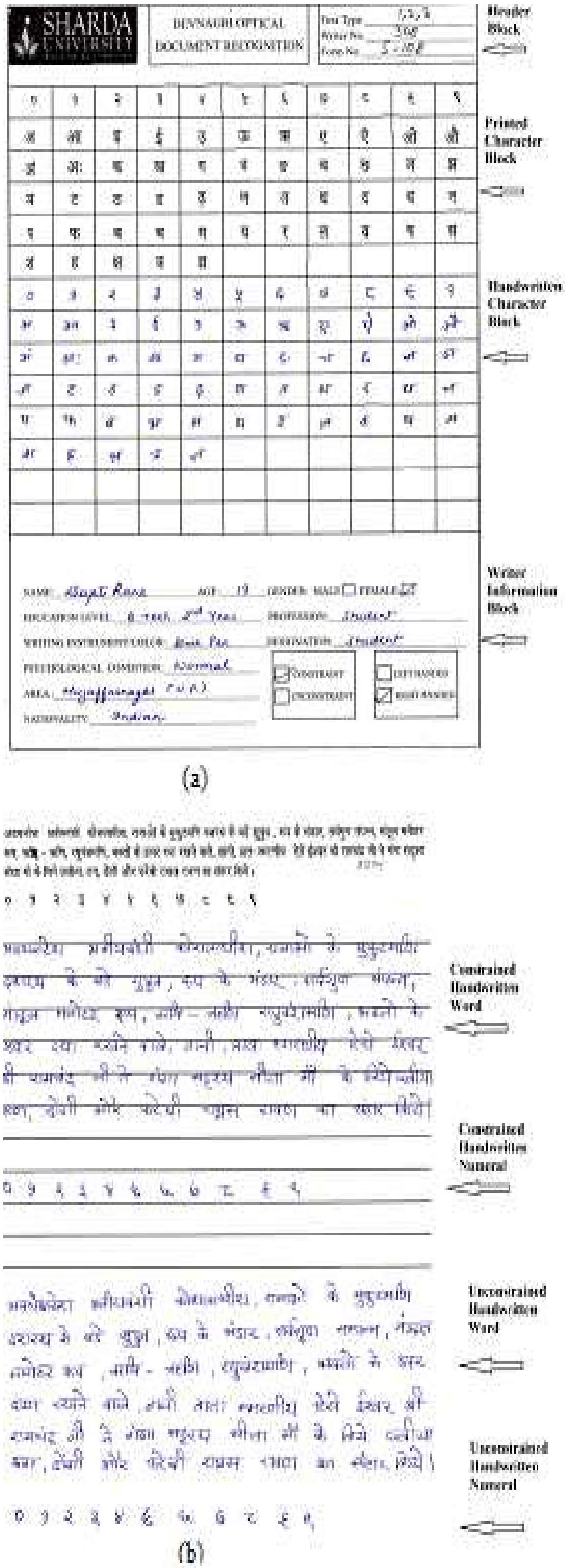

SHARDA UNIVERSITY
DEVNAGRI OPTICAL DOCUMENT RECOGNITION
Test Type
Writer No
Form No
Header Block
Printed Character Block
Handwritten Character Block
Writer Information Block
NAME
AGE
GENDER: MALE □ FEMALE ☑
EDUCATION LEVEL
PROFESSION
WRITING INSTRUMENT/COLOR
DESIGNATION
PHYSIOLOGICAL CONDITION
AREA
NATIONALITY
☑ CONSTRAINT
□ UNCONSTRAINT
□ LEFTHANDED
☑ RIGHTHANDED
(a)
Constrained Handwritten Word
Constrained Handwritten Numeral
Unconstrained Handwritten Word
Unconstrained Handwritten Numeral
(b)

Figure 3: Design of (a) Form-1 as used in isolated characters extraction (b) Form-2 as used in text extraction.

We asked writers to write the pangram text on guided line given below the pangram for constrained handwriting sample collection and repeat the same in the blank space (without guidelines) provided for unconstrained handwriting sample collection. We collected data from 2,000 writers where each writer filled both the forms: Form-1 & 2. We digitized the duly filled forms using HP Cano LiDE 110 scanner at resolution 300 DPI in color mode, and from these forms extracted the desired data using specially made software [30].

2.1.1. Form 1 image processing

The extraction of isolated characters (digits and alphabet) and writer information from Form-1 begins with skew correction operation, if required. To speed-up the process we applied an automatic image skew correction operation before applying the images segmentation operation to extract the images of individual characters. We used Radon transform [33] for skew correction. Afterwards, in the skew free Form-1 images, the process locates automatically the machine printed character block, handwritten character block followed by writer's information block.

To extract the isolated characters the process performs the following steps.

1. Binarize Form-1 image using Otsu Method [15].
2. Remove noises (impression of other forms, salt and pepper noise, corner folding, physically damaged paper, extraneous lines, stapler pins marks) that might have occurred during the digitization process.
3. Perform hole filling morphological operations to obtain the uniform connected component.
4. Perform the labeling operation on the connected components obtained in step-3 to find the bounding box (top-left point, width and height) for each labeled region.
5. Locate and filter out all labeled components in the handwritten character block whose areas were less than a specified threshold.

The process accepted 1,700 out of 2,000 forms. From each accepted from 154 bounding boxes were detected, cropped, stored and displayed for verification. Through this process, we accepted constrained handwritten samples of 15,000 numerals and 83, 300 characters. The process rejected poor quality samples (1,400 samples). These samples have also been stored in the database for further investigation.

2.1.2. Form-2 Processing

Like before, we converted Form-2 images into binary images and removed the extraneous noises. The last line of each handwritten pangram (constrained and unconstrained) contains handwritten digits. We used three structuring elements that we referred to as SE1, SE2 and SE3 for image erosion operation. These elements are depicted in Figure 4.

$$\underset{\text{SE1}}{\begin{bmatrix}1 & \cdots & 1\\ \vdots & \ddots & \vdots\\ 1 & \cdots & 1\end{bmatrix}_{15\times 25}} \qquad \underset{\text{SE2}}{\begin{bmatrix}1 & \cdots & 1\end{bmatrix}_{1\times 100}} \quad \underset{\text{SE3}}{\begin{bmatrix}1 & \cdots & 1\\ \vdots & \ddots & \vdots\\ 1 & \cdots & 1\end{bmatrix}_{100\times 25}}$$

Figure 4: Structuring Elements Used In Form-2 Processing.

Like before, we converted Form-2 images into binary images and removed the extraneous noises. The last line of each handwritten pangram (constrained and unconstrained) contains handwritten digits. We used three structuring elements that we referred to as SE1, SE2 and SE3 for image erosion operation.

In order to extract individual digits from these lines following steps were performed.

1. Erode the binary image with structuring element SE1 so that isolated characters were merged within the word.

2. Erode the resultant image with a line structuring element SE2 that resulted image in a connected component.

3. Label the connected components obtained in step-2 to find region properties (top-left, width and height).

4. Select the last connected component and perform the following steps as a selection criteria:

    a) If found acceptable (size is more than 15 pixels) once then save it, else go-to step (b)

    b) Erode the same input image with another Structuring element SE3 and check for acceptability condition as in (a) and go-to (c).

    c) If acceptable then save the component and perform step 5, otherwise discard the component.

5. Invert the resulted component image from step- 4 and erode it to produce a list of white regions bounded by black regions.

6. Find the top left point, width and height of each white region, and that allowed to crop the

individual digit image, store and display it for manual inspections and labeling.

Through this process, we collected 15,000 unconstrained and 5,000 constrained handwritten numerals. However, in case of constrained numerals, we lost some samples because writers wrote them over the upper, lower or both guide lines.

The final dataset consists of: 83,300 isolated characters; 35,000 numerals; 2,000 constrained pangrams and 2,000 unconstrained pangrams; Writer's Information; 2,000 Form-1 images and 2,000 Form-2 images. For processing these colour images were preprocessed to remove the noise, binarized, and size normalize into 32 x 32 pixels as shown in Figure 5.

We further divided the digit dataset into two sets (dataset A and B). The distribution of digits in each set is shown in Figure 6. These sets were used as training and test set interchangeably.

We observed similarities in shape six and nine[1] see Figure 7 (a), violation in writing guidelines like (Figure 7 (b) and (c)),

Overwriting (see Figure 7 (d)) and discontinuity in shapes see Figure 7 (e). Consequently, the shape similarity created confusion. The violation of guideline converted a character image into another character. The overwritten character produces distorted image. The shape discontinuity made it impossible to extract contour [20] like features.

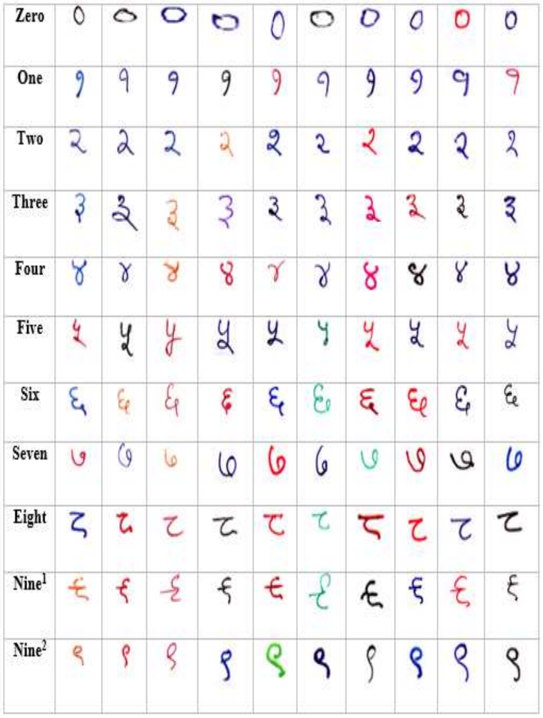


Figure 5: Samples of CPAR-2012 numeral datasets

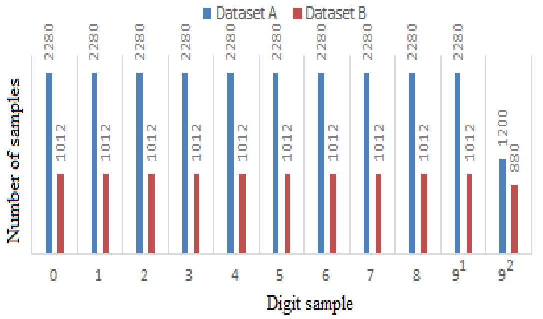


Figure 6: Distribution of digit

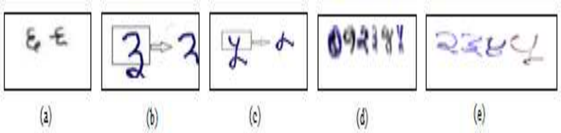


Figure 7: Peculiarities in dataset (a) six becomes nine[1] (b) three becomes two (c) five becomes four (d) overwritten and (e) broken digits

3. RECOGNITION METHOD

In this present study, we are in the process of conducting a set of experiments on the entire dataset. In these experiments, we are implementing the best performing recognition techniques as reported in the *Devnagari* numeral recognition literature [1, 5-11]. The objective of these experiments is to provide recognition results for benchmark studies.

3.1. Feature Extraction

A discriminative feature vector is preferred for high recognition results at comparable cost. From literature it is clear that structural feature and gradient based feature perform well in similar shape recognition like *Devnagari* digits. Since in present study we provide benchmark results for future study of newly created CPAR-2012 dataset, we are not proposing a new feature but will use state-of-the-art feature vector used in digit handwriting recognition. We measured the performance with features ranging from the simple most features in which each feature vector element was the direct pixel value [19] to more computationally expensive features obtained from simple profiles [6] and gradient & wavelet transform [1]. On further study we combined the direct pixel, profile feature with the gradient and wavelet transform features. A brief description of feature extraction and classification scheme is given in this section.

3.1.1. Direct pixel value features

We started our experiments with a simple feature definition - the pixel value. We formed feature vectors by storing the size normalized two dimensional digit images into one dimensional (in column major) feature vectors where each feature element is the pixel value. For this we resized the images into 16, 64,256 and 1024 pixels in column major. We used the recognition results of these experiments as a baseline for comparison purposes, assuming that it represents the worst recognition scenario.

3.1.2. Profile based features

For comparative study we used simple profile [32] features, which are easy to extract. Due to their simplicity and usefulness, several variations, like features from left, right, top and bottom profiles are being used. We performed experiments considering all four profiles forming 128 pixels (32 x 4) to define feature vectors. Where each feature element depicts the profile value, is formed by combining the above mentioned profiles respectively.

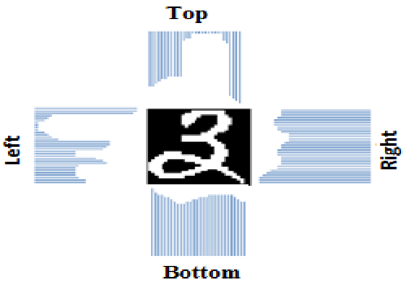


Figure 8: Left, Right, Top and Bottom profile of digit 3.

The profile feature values range from 1 to 32 pixels which is length and width of each image. Figure 8 shows profiles of handwritten a numeral three from CPAR-2012 dataset.

3.1.3. Gradient and wavelet features

Gradient are based on local derivatives of image which is bigger at locations of the image where the image function undergoes rapid changes. The gradient based operator is used to indicate such locations in the image. Keeping this in view we used Kirsch Operator because it detects four directional edges more accurately than other operator [21]. The gradient feature vectors for horizontal (H), vertical (V), right-diagonal (R), and left-diagonal (L) directions are calculated according to:

$G\ (i,\ j)_H = \max(|5S_0 - 3T_0|,|5S_4 - 3T_4|)$
$G(i,j)_V = \max(|5S_2 - 3T_2\ |,\ |5S_6 - 3T_6\ |)$
$G(i,j)_R = \max(|5S_1 - 3T_1\ |,\ |5S_5 - 3T_5\ |)$
$G\ (i,\ j)_L = \max(|5S_3 - 3T_3|,\ |5S_7 - 3T_7|)$

Where $S_k= A_k + A_{k+1} + A_{k+2}$
$T_k= A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7}$
Further to reduce the size of feature vector and capture the local and global feature we used db1 wavelet transformation. The feature formation process is depicted in Figure 9.
In these experiments images were resized to 32 x 32 pixels. In order to capture the local features from a size normalized image we formed four images by applying H, V, L and R Kirsch operators. On each of these images, db1' wavelet transformation [29, 31] was applied and four sets of features (16 features per image) were generated by taking the LL components of each image. Likewise, to capture the global information, the db1' wavelet transformation was applied on the original size normalized image and 16 features were generated by taking its LL components. Finally we combine both the features and formed a feature vector of 80 elements representing 64 local and 16 global features.

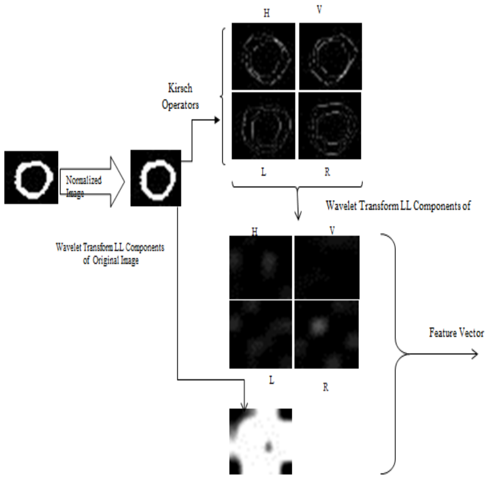


Figure 9: Gradient features extraction.

3.2. Classifier

In this section we give a brief description of classification techniques used in this study. For this study, we chose neural network classifiers- Pattern Recognition (PR), Feed forward (FFN), Fitness Function (FFT), Cascade Neural Network (CCN), statistical classifier- KNN (k-nearest neighbor) classification methods. These classifiers are available with MATLAB [22].

3.2.1. Neural network classifier

An N-layered feed-forward multilayer neural network contains one input (the first) layer, one output (the last) layer, and N- hidden (intermediate) layers. Starting from the first layer, neurons of every pairs of layers say layers k-1 and k, are connected with each other via a weight matrix $W^{k}_{m_k,m_{k-1}}$ where m_k and m_{k-1} are the total number of neurons in the kth and (k-1)th layers respectively. The element $W^{k}_{m_k,m_{k-1}}(i,j)$, where $1 \leq i \leq m_k$ and $1 \leq j \leq m_{k-1}$, denotes the weight between the ith neuron of the kth layer and the jth of neuron of the (k-1)th layer.

The output of ith neuron of the kth layer is a function of the ith row of $W^{k}_{m_k,m_{k-1}}$ and the output O^{k-1} $1 \leq j \leq m_{k-1}$ of the (k-1)th layer neurons, the output of the ith neuron of the kth layer is $o_i^k = f(net_i^k)$

where $net_i^k = \sum_{j=1}^{m_{k-1}} w_{i,j}^k \times O_j^{k-1} + b_i^k$, O^{k-1} is a column vector of size m_{k-1} where each element is an output of the (k-1)th layer neurons, $\mathbf{b}^k$ is a column vector of size m_k where each element is a bias for kth layer neurons.

In MATLAB [22] there are several implementations of this model.

In this experiment we created neural network with 10 hidden layers in all the neural network classifiers. This classifier uses logsig transfer function. This functions calculate the layers output from its input. The output layer of feedforward neural network is given by

$$o_i^k = f(net_i^k)$$

$$= \text{logsig}(net_i^k) = 1/\ 1 + e^{-net_i^k}$$

The second classifier used were pattern recognition classifier. This function is similar to feedforwardnet except, it uses tansig transfer function in the last layer.

$$o_i^k = \text{tansig}(net_i^k) = 2/(1+(e^{-2*net_i^k})) -1$$

This network is more commonly used for pattern recognition purposes. This function is good where speed is important and the exact shape of the transfer function is not important.

The third classifier used were cascade forward neural network. This classifier uses function that is similar to feed forward networks but include a weight connection from input to each layer and for each layer to successive layers e.g. layer 1 to layer 2 , layer 2 to layer n and layer 1 to layer n. The three –layer network also has connection from input to all three layers. The additional connection improves the speed at which the network learns the desired relationship.

The fourth classifier used were function fitting neural network. This classifier uses a function to fit input-output relationship and returns a fitting neural network.

3.2.2. Statistical classifier

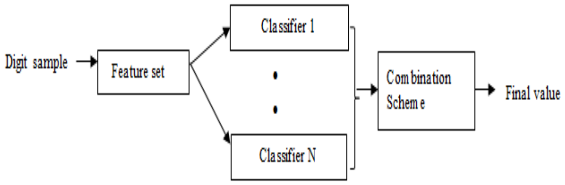
The classifier [24] predicts the class label of the test pattern x from predefined class. The classifier finds the k closest neighbor of x and finds the class label of x using majority voting. The performance of KNN classifier depends on the choice of k and distance metric used to measure the neighbor distances. In our experiment we used Euclidean distance metric.

3.2.3. Classifier ensemble

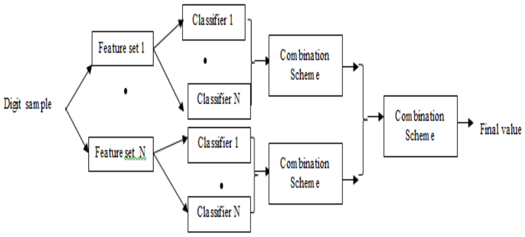
We obtained the final recognition results by combining the decisions of several classifiers [25]. Figure 10 (a) & (b) show the proposed framework for classifier ensemble schemes. Figure 10 (a) is a single stage and Figure 10 (b) is a multistage classifier ensemble scheme.

In single stage ensemble we used single feature type and multiple classifiers. In this scheme we classify an unknown digit by combining the decisions of all the classifiers. Whereas in multistage classifier ensemble, in the first stage we pooled decisions of several single stage ensemble scheme (using same set of classifier on different feature types). Afterword's, obtain the final decision by combining the pooled decision. We combined the classification decision, in all the cases, using majority voting rules.

In this scheme an unknown digit is recognized as the one that is supported by majority of classifiers, otherwise it is rejected. This scheme assumes that all classifiers have equal vote value. But in reality the recognition performance differ from classifier to classifier.



(a)



(b)

Figure 10: Classifier ensemble (a) single stage and b. multistage

4. EXPERIMENTS WITH CPAR-2012 DIGIT DATASET

We conducted all experiments on binarized, resized (32 x 32 pixels) and noise removed digit samples. In these experiments, we first evaluated the performance of recognition schemes formed by feature as explained in section 3 and classifier coupling.

For classification we chose five −Pattern recognition network (PR) [22], Feed-forward network (FFN) [22], Fitness function network (FFT) [22], Cascade neural network (CCN), and k-nearest neighbor (KNN) classification methods from MATLAB (R2012a).In addition to these schemes, two classifier ensemble: single layer and multilayer classifier ensemble also has been studied.

All neural network (NN) classifier models were trained using resilient back-propagation (RP) [27] and scale conjugate back-propagation (SCG) [28] learning algorithms.

To maintain the uniformity in all our experiments, we divided the dataset into two sets, namely set A of size 11,000 and set B of size 24,000 samples respectively. We used these sets as training and test sets, as indicated in the Table 3, interchangeably, In experiment with experimental dataset I the classifiers were trained on dataset set B (24,000 samples) and tested on dataset A (11,000 samples). In experiment with experimental dataset II they were trained and tested on dataset B while in experiment with experimental dataset III they were trained and tested on dataset A. In order to assess the effect of training set size on recognition, in experiment with experimental dataset IV, we trained the classifiers dataset A (the smaller dataset) and tested them on dataset B (the larger dataset).

Table 3: Experimental dataset with training and test

Dataset	Training Set	Test set
I	B 24,000	A 11,000
II	B 24,000	B 24,000
III	A 11,000	A 11,000
IV	A 11,000	B 24,000

4.1. Experiments with Direct Pixel Features

In this experiment, the recognition performance of the simplest feature element, i.e., the direct pixel value, is measured on neural networks and KNN classifiers. Figure 11 and Figure 12 show the results of neural network classifiers using SCG & RP learning algorithms respectively. Figure 11 also shows the result of KNN classifier that yielded the high recognition accuracy. Among the neural network classifiers, the pattern recognition classifier yielded the better recognition score. In experiments II and III, in comparison of the first set of experiments, all classifiers have yielded better recognition scores. The reason is that the classifiers were trained and tested on the same datasets. In experiments IV all the classifiers performed poorly because they were trained on the smaller dataset and tested on the larger dataset. It indicates the effect of the training sample size on the recognition performance. In all plots y axis denotes average recognition accuracy in percentage.

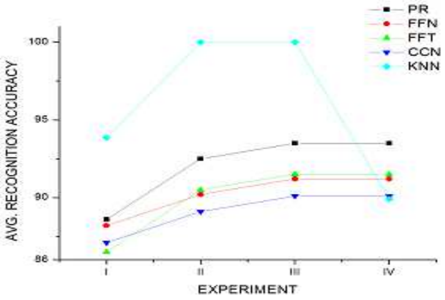


Figure 11: Results of SCG Training & KNN

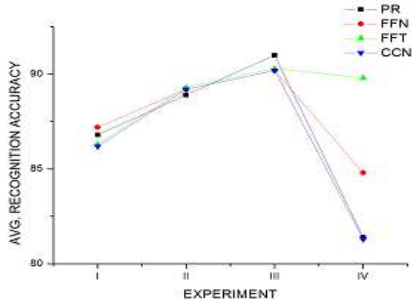


Figure 12: Results of RP Training

In an attempt to measure the effect of varying image sizes on recognition performance and recognition speed, we conducted experiments using NN classifiers with SCG learning algorithm. Figure 13 shows the results obtained on experimental dataset I. As expected, the results are stabilized for larger sized images. The Table 4 shows the average recognition speed. From this table, we noted that the NN classifier which is trained with SCG learning algorithm recognize digits with faster speed.

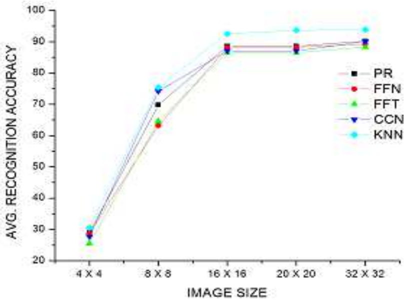


Figure 13: Average recognition result with various image sizes.

Table 4: Execution time of NN classifier in seconds with (a) SCG and (b) RP learning

SIZE	PR	FFN	FFT	CCN	KNN
4x 4	162.19	535.35	158.62	345.61	15.29
8x 8	207.7	448.9	504.1	615.27	22.3
16 x 16	199.9	470.13	535.5	1021.03	145.53
20 x 20	197.6	475.21	320.36	1031.27	267.87
32 x 32	180.3	406.007	318.29	1407	526.94

(a)

SIZE	PR	FFN	FFT	CCN
4 x 4	160.8	560.3	210.6	361.1
8 x 8	305.2	624.2	714.2	605.1
16 x 16	190.76	580.24	580.61	1107.31
20 x 20	200.42	570.23	425.81	1205.21
32 x 32	230.51	450.87	430.35	1501.3

(b)

4.2. Experiments with Profile Features

Figure 14-15 below show the profile feature based recognition results. In all these experiments, the classifiers were trained and tested on datasets A & B as described in the last section. In this case, the pattern recognition classifier yielded the highest and cascade forward neural network the worst recognition scores in all the experiments. In experiments II and III all the classifiers yielded better scores and in experiments IV they yielded poor scores for the reasons described in the last section. The performance of the KNN classifier remains almost the same like before.

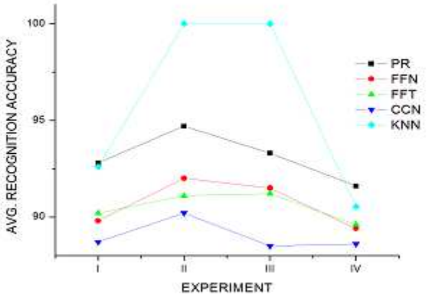


Figure 14: Results of SCG Training & KNN

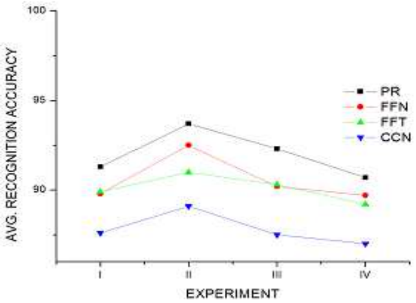


Figure 15: Results of RP Training

4.3. Experiments with Gradient and Wavelet Features

Like all other experiments, in these experiments, KNN classifier yielded the highest recognition score. Among the neural network classifiers, the pattern recognition neural network classifier yielded the best recognition score in all the experiments and the feed forward neural network yielded the poor recognition score in almost all the experiments (see Figure16 -17).

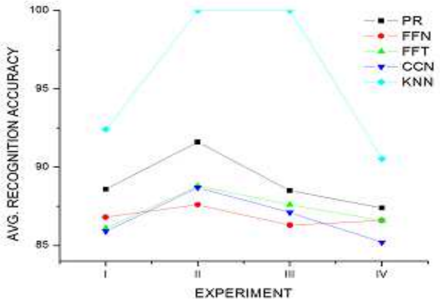


Figure 16: Results of SCG Training & KNN

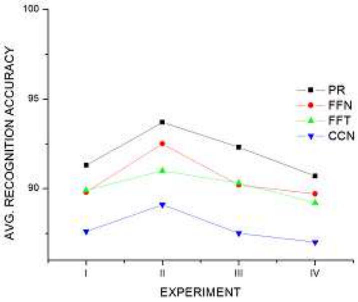


Figure 17: Results of RP Training

4.4. Experiments With Combined Profile And Gradient Features

As mentioned before, the gradient feature captures the local properties while the profile feature the global properties of a pattern. In an attempt to measure the effects of combining the local and global properties on recognition performance, we conducted experiments with the feature vector that we formed by combining the gradient and profile features, and conducted the similar experiments on experimental datasets A and B as described in Sections 4.1. Figure 18-19 show the results of these experiments.

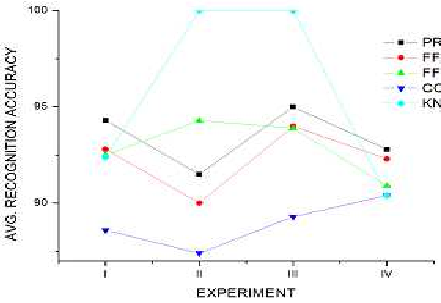


Figure 18: Results of SCG Training & KNN

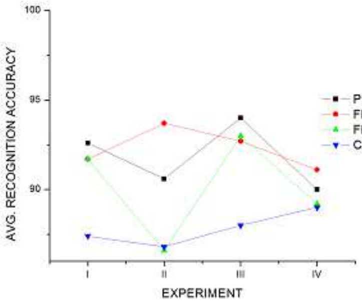


Figure 19: Results of RP Training

4.5. Combined gradient and direct pixel

In search of improved recognition accuracy, we conducted one more set of similar experiments, on the dataset A & B, using feature vector that we formed by combining the gradient and direct pixel features. Figure 20-21 show results obtained in these experiments. Again, the KNN classifier yielded the highest and cascade neural network poor recognition score respectively in almost all the experiments. No significant difference was observed in other experiments.

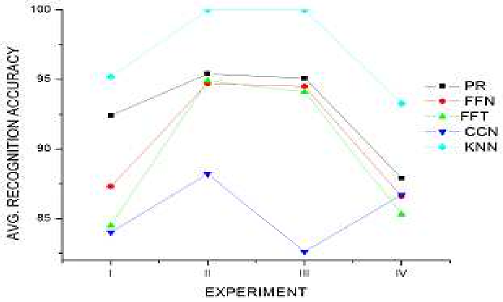


Figure 20: Results of SCG Training & KNN

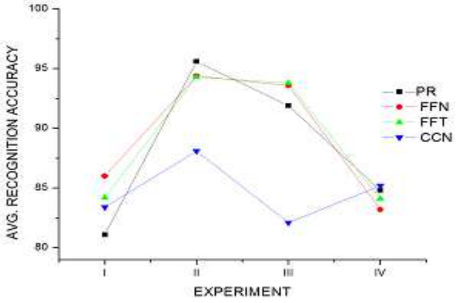


Figure 21: Results of RP Training

4.6. Experiments with Classifier ensemble

In an attempt to measure the recognition performance of schemes that combine classifiers, we conducted two experiments: One with the single stage and another with multi-layer classifier ensemble scheme. Both scheme uses majority voting scheme for final decision as explained in section III. Figure 22 show the results of majority voting ensemble scheme. This figures depict the recognition score obtained by combing the classifiers for different features sets e.g., KW (kirsch & wavelet), SP (profile Feature), SPKW(profile and kirsch & wavelet), DPKW(direct pixel and kirsch & wavelet) and DP (direct pixel).

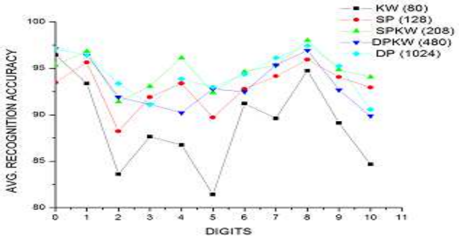


Figure 22: Results Of Simple Majority Using Single Stage Classifier Ensemble

In order to reduce the misrecognition rate, once again we combined the decision obtained from this ensemble scheme and also we introduced a rejection criterion i.e., reject a digit if there is no majority consensus .The rejection yielded 97.87 % recognition accuracy. Table-5 shows its confusion matrix. Figure 23 compares the recognition accuracy of majority voting multi-stage classifier ensemble scheme with rejection (MVR) and without rejection (MV) criterion. MVR shows better results than MV in all digits.

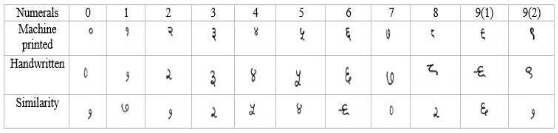
Table 5 shows the confusion matrix, we discovered that digit zero (0) is confused mostly with one, digit one with seven, digit two with one, digit three with two, digit four with five, digit five with four, digit six with nine(1), digit seven is mostly confused with zero, digit nine(1) with six, digit nine(2) is confused with digit one. These confusion arose because of shape similarity. Table-6 shows shape similarity in *Devnagari* digits where each column indicates the similarity in shape of a digit given in first row with digits given in other rows.

Table-7 shows shapes of some misclassified test digits.

Table 5: Final Confusion Matrix Using Multi-Level Classifier Ensemble After Rejecting The Disagreed Values.

	0	1	2	3	4	5	6	7	8	9^1	9^2	Total Digits	%age
0	986	9	0	0	0	0	1	5	0	0	1	1002	98.40
1	1	987	3	0	0	0	0	5	0	2	3	1001	98.60
2	3	6	975	5	0	1	0	0	3	2	1	996	97.89
3	0	2	23	967	0	0	0	1	0	0	3	996	97.09
4	0	1	0	0	967	6	3	0	4	4	3	988	97.87
5	0	3	11	1	15	938	1	0	0	0	1	970	96.70
6	0	0	3	0	0	5	953	3	2	8	4	978	97.44
7	14	3	1	0	1	0	1	986	1	0	0	1007	97.91
8	1	0	1	2	0	0	0	0	996	1	0	1001	99.50
9(1)	0	0	3	0	0	0	12	0	2	972	6	995	97.69
9(2)	2	16	0	0	0	0	1	1	2	0	835	857	97.43
Average Recognition													97.87

Table 6: Shape similarity in Devnagari digits.



Numerals	0	1	2	3	4	5	6	7	8	9(1)	9(2)
Machine printed	०	१	२	३	४	५	६	७	८	९	९
Handwritten	०	१	२	३	४	५	६	७	८	९	९
Similarity	१	७	१	२	५	४	९	०	२	६	१

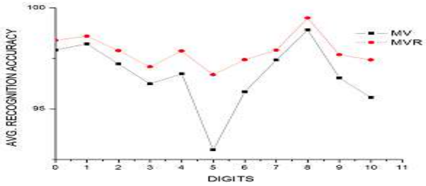
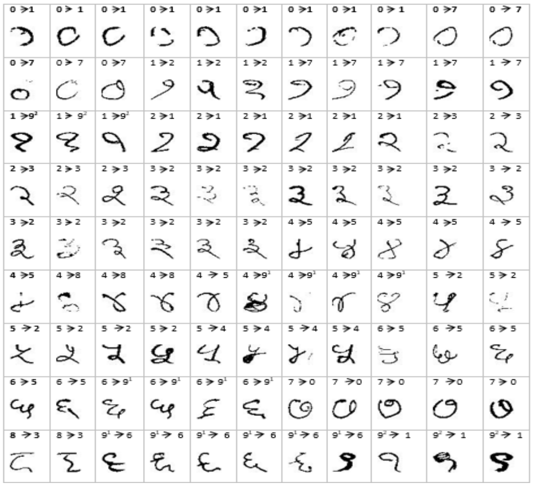


Figure 23: Comparison Of Simple Majority Voting With And Without Rejection Criterion.

Table 7: Some Of The Misclassified Test Digits By Using Multi-Level Majority Voting Classifier Ensemble. The Text Above Each Digit Shows The True Label Followed By Assigned Label.


0 → 1
0 → 1
0 → 1
0 → 1
0 → 1
0 → 1
0 → 1
0 → 1
0 → 1
0 → 7
0 → 7
0 → 7
0 → 7
0 → 7
1 → 2
1 → 2
1 → 2
1 → 7
1 → 7
1 → 7
1 → 7
1 → 7
1 → 9²
1 → 9²
1 → 9²
2 → 1
2 → 1
2 → 1
2 → 1
2 → 1
2 → 1
2 → 3
2 → 3
2 → 3
2 → 3
2 → 3
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
3 → 2
4 → 5
4 → 5
4 → 5
4 → 5
4 → 5
4 → 5
4 → 8
4 → 8
4 → 8
4 → 5
4 → 9¹
4 → 9¹
4 → 9¹
4 → 9¹
5 → 2
5 → 2
5 → 2
5 → 2
5 → 2
5 → 2
5 → 4
5 → 4
5 → 4
5 → 4
6 → 5
6 → 5
6 → 5
6 → 5
6 → 5
6 → 9¹
6 → 9¹
6 → 9¹
6 → 9¹
7 → 0
7 → 0
7 → 0
7 → 0
7 → 0
8 → 3
8 → 3
9¹ → 6
9¹ → 6
9¹ → 6
9¹ → 6
9¹ → 6
9¹ → 6
9² → 1
9² → 1
9² → 1

5. CONCLUSION

The paper reports on the development of a benchmark dataset— CPAR-2012 — for handwritten *Devnagari* character recognition and the *Devnagari* digit recognition performance. The salient features of this dataset is that it contains multi-type texts like isolated numerals, characters and words along with writers information, and a set of pangram text written by two thousand writers. This dataset can be used to test and benchmark the recognition techniques and algorithms for isolated numeral, character, word recognition, handwriting analysis, writer identification, handwriting individuality analysis, text recognition, and similar research.

We have measures the recognition performance on handwritten digits of CPAR-2012 using different features: direct pixel, simple profile, gradient feature, gradient feature combined with direct pixel and gradient feature combined with profile feature on six different classifiers: four neural networks—pattern recognition, function fitting, cascade and feed-forward neural network—and statistical classifier: KNN.

In an attempt to improve recognition accuracy we used classifier ensemble scheme: the single stage and multi-stage. In both we applied simple and weighted majority voting scheme using kappa coefficient as weight (computed from the confusion matrix of a classifier) for final decision. The recognition accuracy improved from 95.18 % to 97.87 % using multi-stage classifier with simple as well as weighted majority scheme. No significant difference was noted in the results between these two voting schemes.

In these experiments, the KNN classifier consistently yielded high recognition accuracy of almost 100% on training sets. Among Neural Network classifier, the Pattern recognition (PR) classifier with scaled conjugate back-propagation learning rules yielded the best recognition results, in terms of accuracy and speed, as compared to all other neural network classifiers. It is also observed that training set size affects the recognition accuracy.

We have organized the dataset as relational database. It is being integrated into an integrated research environment. The environment is being designed to facilitate the sharing of data and results among researchers and allow them to expand the dataset by storing a large variety of handwriting samples along with writer's attributes. Such a dataset would help in discovering ground truth from the handwriting samples and in turn would help in authenticating the reliability of handwriting based systems.

REFERENCES:

[1] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 3, Mar. 2009, pp. 444–457.

[2] Yousef Al-Ohali, Mohamed Cheriet, ChingSuen, "Databases for recognition of handwritten Arabic cheques", Pattern Recognition, vol. 36, no. 1, 2003, pp. 111-121.

[3] R. Jayadevan, Satish R. Kolhe, Pradeep M. Patil and Umapada Pal, "Offline Recognition of Devanagari Script: A Survey", IEEE Transactions on Systems, Man, and Cybernetics, Part C , vol. 41 no. 6, 2011, pp. 782-796.

[4] http://uidai.gov.in/what-is-aadhaar-number.html

[5] K. Sethi and B. Chattarjee, "Machine recognition of Hand printed Devnagari Numerals". Journal of Instituions of Electronics & Telecommunication Engineers, India, vol. 22, 1976, pp. 532-535.

[6] Reena Bajaj, LipikaDey, and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, vol. 27, no. 1, 2002, pp. 59-72.

[7] M. Hanmandlu and O. V. R. Murthy, "Fuzzy model based recognition of handwritten numerals," Pattern Recognition, vol. 40, 2002, pp. 1840–1854.

[8] Elnagar, A., Harous, S. "Recognition of handwritten Hindi numerals using structural descriptors" J. Exp. Theor. Artif. Intell., 2003, pp. 299-314.

[9] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "A novel framework for automatic sorting of postal documents with multi-script address blocks," Pattern Recognition, vol. 43, 2010, pp. 3507–3521.

[10] G. G. Rajput and S. M. Mali, "Fourier descriptor based isolated Marathi handwritten numeral recognition," Int. J. Comput. Appl., vol. 3, no. 4, 2010, pp. 9–13.

[11] P. M. Patil and T. R. Sontakke, "Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network," Pattern Recognition, vol. 40, 2007, pp. 2110–2117.

[12] U. Pal, T. Wakabayashi, N. Sharma, F. Kimura, Handwritten Numeral Recognition of Six Popular Indian Scripts, in: Proceedings of 9th International Conference on Document Analysis and Recognition (ICDAR), 2007, pp. 749–753.

[13] B.B. Chaudhuri, A complete handwritten numeral database of Bangla—A major Indic script, in: Proceedings of the 10th International Workshop on Frontiers of Handwriting Recognition, La Baule, France, 2006, pp. 379–384.

[14] B Nethravathi, C P Archana, K Shashikiran, A G Ramakrishnan, V Kumar, Creation of a huge annotated database for Tamil and Kannada OHR, in: Proceedings of IWFHR, 2010, pp. 415-420.

[15] A. Alaei, P. Nagabhushan, and U. Pal, A Benchmark Kannada Handwritten Document Dataset and Its Segmentation , in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 2011, pp.

141-145.

[16] R. Jayadevan, Satish R. Kohle, Pradeep M. Patil, Database Development and Recognition of Handwritten Devanagari Legal Amount Words, in: Proceedings of International Conference on Document Analysis and Recognition, 2011, pp. 304-308.

[17] R.Sarkar, N.Das, S.Basu, M.Kundu, M.Nasipuri, D.K.Basu, CMATERdb1: A database of unconstrained handwritten Bangla and Bangla–English mixed script document image, International Journal on Document Analysis and Recognition, , Volume 15, Issue 1, March 2012, pp. 71-83.

[18] Ostu, Nobuyuki. "A threshold selection method from gray-level histogram." IEEE Transactions on Systems, Man and Cybernetics , vol. 9, no. 1, 1979, pp. 62-66.

[19] G. Mayraz, G.E. Hinton, Recognizing handwritten digits using hierarchical products of experts, IEEE Trans. Pattern Analysis Machine Intelligence, vol. 24, no. 2, 2002, 189–197

[20] P. Ahamed and Yousef Al-Ohali. "TAR based shape features in unconstrained handwritten digit recognition." WSEAS Transactions on Computers, vol. 9, no. 5, 2010, pp. 419-428.

[21] C.L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques," Pattern Recognition, vol. 36, no. 10, Oct. 2003, pp. 2271-2285.

[22] http://www.mathworks.in/help/nnet/recognizing-patterns.html

[23] S. Setlur, S. Kompalli, V. Ramana prasad, and V. Govindaraju, "Creation of data resources and design of an evaluation test bed for Devanagari script recognition," in Proc. 13th Int. Workshop Res. Issues Data Eng.:Multi-lingual Inf. Manage. (RIDE-MLIM), 2003, pp. 55-61.

[24] T.M. Cover, P.E. Hart, Nearest neighbour pattern classification, IEEE Trans. Informat. Theory IT-13, 1967, pp. 21–27.

[25] Rahman, Ashfaqur, and Brijesh Verma. "Effect of ensemble classifier composition on offline cursive character recognition." Information Processing & Management vol. 49, no. 4, 2013, pp. 852-864.

[26] Prachi Mukherji, and Priti P. Rege. "Fuzzy stroke analysis of Devnagari handwritten characters." *WSEAS Transactions on Computers* 7, no. 5, 2008, pp. 351-362.

[27] M. Riedmiller and H. Braun, "A direct daptive method for faster back-propagation learning: The RPROP algorithm," in Proceedings of IEEE Int. Conf. Neural Networks, San Francisco, CA, 1993, pp. 586–591.

[28] Møller, Martin Fodslette. "A scaled conjugate gradient algorithm for fast supervised learning." Neural networks vol. 6, no. 4, 1993, pp. 525-533.

[29] Rahbar, Kambiz, Muhammad Rahbar, and Farhad Muhammad Kazemi. "Handwritten numeral recognition using multi-wavelets and neural networks." In Proceedings of the 5th WSEAS international conference on Signal processing, 2006, pp. 56-58.

[30] Rajiv Kumar, Amresh Kumar, P Ahmed , "A Benchmark Dataset for Devnagari Document Recognition Research", 6th International Conference on Visualization, Imaging and Simulation (VIS '13), Lemesos, Cyprus, March 21-23, 2013, pp. 258-263.

[31] Stephane G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, 1989, pp. 674-693.

[32] Rajiv Kumar, Mayank Kumar Goyal, Pervez Ahmed, and Amresh Kumar. "Unconstrained handwritten numeral recognition using majority voting classifier." In *Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on*, pp. 284-289. IEEE, 2012.

[33] J. Coetzer, B.M. Herbst, J.A. duPreez, Offline Signature Verification Using the Discrete Radon Transform and a Hidden Markov Model, EURASIP Journal on Applied Signal Processing, 2004, pp. 559–571.