OFFLINE HANDWRITTEN DEVNAGARI DIGIT RECOGNITION

Rajiv Kumar and Kiran Kumar Ravulakollu
Department of Computer Science and Engineering, Sharda University, India
E-Mail: rajivbec@rediffmail.com,

ABSTRACT

This paper presents a study on the performance of transformed domain features in Devnagari digit recognition. In this research the recognition performance is measured from features obtained in direct pixel value, Fourier Transform, Discrete Cosine Transform, Gaussian Pyramid, Laplacian Pyramid, Wavelet Transform and Curvelet Transform using classification schemes: Feed Forward, Function Fitting, Pattern Recognition, Cascade Neural Networks and K-Nearest Neighbor (KNN). The Gaussian Pyramid based feature with KNN classifier yielded the best accuracy of 96.93% on the test set. The recognition accuracy was increased to 98.02% by using a majority voting classification scheme at expense of 0.26 % rejection rate. The majority voting classifiers are based on features: Gaussian pyramid, Laplacian pyramid, wavelet pyramid and direct pixel value using KNN classifiers.

Keywords: digit recognition, transform domain, CPAR-2012 dataset, neural network classifier and majority voting classification scheme.

INTRODUCTION

Development of Optical Character Recognition (OCR) system for unconstrained handwritten numeral recognition is an active, yet challenging area of research [1-7]. The variations in handwriting pose major challenge in developing accurate recognition system. Due to the reason the handwriting variations induce virtually unmanageable variations that make feature definition extremely difficult. Thus, discovering a precise and efficient feature extraction method became a formidable task. However, to deal with it, there have been several efforts made to define and extract features that may have reduced effects of handwriting variations. In this paper, we present our experiences with transformed domain based feature definition and extraction techniques in developing a recognition system for unconstrained handwritten *Devnagari* digits. Some advancement in handwritten *Devnagari* digit recognition using features from spatial and transformed domains are summarized in Table-1 and Table-2 given below.

Table-1 shows that the performance of structural features [3-6, 8, 9] ranges from 40 to 97%. Only in one case [2] it is reported 99.5% while the reported accuracy of statistical features [5, 8] is in the range of 40 to 92% which is slightly less. The accuracy of spatial feature type, such as binary image [9, 10], and box [6] also reported accuracy range between 95 to 96 %.Table-2 shows some results from transformed domain using wavelets [11] and gradient features [12, 13]. The result shows that recognition result in this case is 94.25 to 99.56 which is slightly higher than spatial domain approach. As mentioned before, we thoroughly explored the performance of the features of this domain on a substantially larger, realistic and uniform benchmark dataset.

This dataset is used to for experimentation to estimate: (a) recognition accuracy: (b) recognition time and (c) training size effect. The estimation of (a) to (c) is much closer to reality. The estimated recognition time includes time taken in recognizing a digit from preprocessing to classification on Intel® core™2Duo CPU, 2.00 GHZ, 64 bit operating system, x-64 based processor with 4.00 GB RAM and MTALAB R2013a. To the best of our information no work in Devnagari digit has reported statistics about the throughput of the system. These Tables also shows that almost all the reported Hindi OCR techniques were tested and experimented with synthetic (hand created or simulated) datasets [2-6, 8-10, 13].

Table-1. Spatial domain.

Feature	Classifier	Dataset size	Result	Ref.
Structural	Fuzzy neural network	2, 000	99.5	[2]
Structural approach	Matching syntactic representation	1, 500	96	[3]
QTLR (structural)	SVM	3, 000	97	[4]
Density and moment (statistical)	MLP	2, 460	40-89.68	[5]
Box approach	Fuzzy set	3, 500	95	[6]
Moment invariant	Gaussian distribution	2, 000	92	[8]
Binary image	Clonal selection	12, 000	96	[9]
Binary image	KNN and neural network	35, 000	95.11	[10]

Table-2. Transform domain.

Features	Classifiers	Dataset	Result	Ref.
Wavelet	MLP	22, 556	99.04	[11]
Gradient	MQDF	22, 556	99.56	[12]
Gradient	PCA	9, 800	94.25	[13]

The standard test data set scarcity makes performance comparison of Hindi OCR techniques unrealistic. To realize these objectives we used Center for Pattern Analysis and Recognition (CPAR-2012) dataset which is briefly described in next section and its detail description is given in [10].

In this paper, we performed a comparative performance analysis of Fourier Transform, Discrete Cosine Transform, Gaussian Pyramid, Laplacian Pyramid, Wavelet Pyramid and Curvelet Transform based features using several Neural Network, KNN, and classifier combination classification schemes.

Further section describes the feature extraction and classification techniques respectively and section 5 describes experimental details and section 6 concludes the paper.

CPAR-2012 DATASET

The CPAR-2012 dataset contains images of constrained, semi-constrained and unconstrained handwritten numerals, isolated characters, unconstrained and constrained pangram text, 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.

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), Andhra Pradesh, Madhya Pradesh, Karnataka, Kerala, Rajasthan, and countries: Nigeria, China and Nepal). Dataset was collected by specially designed forms: one form was used to collect the isolated digits, characters, and writer's information and the other form was designed to collect the constrained and unconstrained handwritten words from the pangram. Writers were asked 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. These samples were collected from 2, 000 writers where each writer filled both the forms: Form-1 and 2, these forms were digitized 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 [10].

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

Figure-1. CPAR-2012 digit samples.

FEATURE EXTRACTION

Features were extracted by using the direct pixel value and transformed pixel value features. To extract these features we divided the input image in zones and measured feature values in each zone. The feature extraction process is defined in the following sections.

Direct pixel feature

The direct pixel value feature is the average pixel intensity value in a specified region or zone as depicted in Figure-2. In order to determine the region size we studied the distribution of these features in zone of varying size. We defined these zones by partitioning the size normalized image into equal number of rows and columns. In order to estimate the optimal zone size we experimented with zone size: 2x2, 3x3, 4x4, 5x5, 6x6 and 8x8 and discovered the best zone size of 5x5. In this manner we extracted 36 features from 36 zones from the size normalized 30 x 30 image.

VOL. 9, NO. 2, FEBRUARY 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

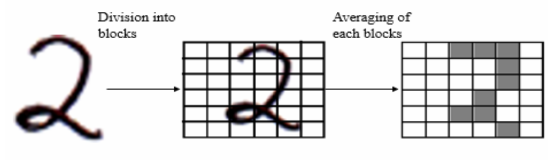


Figure-2. Direct pixel value zoning.

Fourier transform features

Fourier transform transforms the image in frequency domain. This transform shows translation invariant property. We applied Fourier transform on size normalized original image of 30 x30 (see Figure-3(a)). The transformed image is shown in Figure-3(b). Like direct pixel we divided the transformed image into 36 zones each of size 5x5. We formed 36 features by taking the average frequency value from each zone.

Discrete cosine transform

The discrete cosine transform [14] for M x N image, is computed by using Equation (1) as follows:

$$F(u,v) = \frac{1}{\sqrt{MN}} f(x,y)\alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad \text{-Eq.(1)}$$

Where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{M}} & u = 0 \\ \sqrt{\frac{2}{M}} & u = 1,2,\ldots,M-1 \end{cases}$$

F(x, y) shows original image and F(u, v) represents their corresponding DCT transforms. Like previous section the size normalized image (30 x 30) is divided into 36 zones where each zone is of size 5 x 5. In this case also 36 features vectors were formed by taking the average DCT coefficient values in each zone. Figure-3(a) and Figure-3(c) show the original and their DCT transformed image.

Gaussian pyramid

The Gaussian pyramid [15] is a multi-level and multi-resolution representation of images. In this approach we first smooth the image and then subsample the smoothed image successively. The expression given in Equation (2) below is suggested to be used to compute Gaussian pyramid [15] coefficients $G_l(i,j)$ at subsampling level l.

$$G_l(i,j) = \sum_m \sum_n W(m,n) G_{l-1}(2i+m, 2j+n) - Eq.(2)$$

Where l indexes the level of the pyramid and w (m, n) is the Gaussian weighted function.

In this case the original image was resized to 64x64 pixels, and applied the Gaussian pyramid level -1 that produced the transformed image of 32x32 containing Gaussian pyramid coefficients. As before the resized transformed image is divided it into 36 zones. In this case also 36 feature vector was formed by taking the average Gaussian pyramid coefficient values in each zone. Figure-3(a) and Figure-3(d) show the original image and their Gaussian transformed image at level -1.

Laplacian pyramid

The Laplacian pyramid was introduced by Burt and Adelson (1983) for image processing. This transform was applied as explained in [15] where the laplacian pyramid is computed by using Equation (3).

$$L_i = G_i - EXPAND(G_i) = G_l - G_{l+1} - Eq.(3)$$

Where G_l is the Gaussian transformed value at level-l and G_{l+1} is the expansion.

In this case the original image was resized to 64x64 pixels, and applied the Laplacian pyramid level -1 that produced the transformed image of 32x32 containing Laplacian coefficients. As before the resized, the transformed image was divided it into 36 zones. In this case also we formed 36 features by taking average Laplacian coefficient values in each zone. Figure-3(a) and Figure-3(e) shows the original image and their Laplacian transformed image.

Wavelet transform

Wavelet transform [16] provides multi resolution analysis of an image. The transform leads to decomposition of an image into four components: the approximation (LL) and the details in three orientations (horizontal: LH, vertical: HL, and diagonal: HH). Wavelet transforms are available in many varieties. However Daubechies (db-1) wavelets are compactly supported in the image processing and provide good spatial-frequency localization. For this reason in this work Daubechies wavelet pyramid was applied.

In this case the original images are resized to 64x64 pixels, and applied the wavelet transform (db-1) level -1 that produced the transformed image of 32x32 containing four wavelet coefficients. We resized the transformed image (approximation coefficient- LL) into 30 x30 pixels and divided it into 36 zones. In this case also we formed the 36 features by taking the average wavelet transformed coefficient values in each zone. Figure-3(a) and Figure-3(f) shows the original image and their wavelet transformed image.

Curvelet transform

Candes and Donoho [17] introduced Curvelet transform. This transform has very high directional sensitivity and highly anisotropic. Wavelet provides good features for representing point discontinuities but it is not good for representing edge discontinuity.

In this case Curvelet transform was applied on the size normalized image (30 x 30). The transformed images were divided into 36 zones where each zone is of size 5 x 5. In this case also the 36 features vectors were formed by taking the average Curvelet transform coefficient values in

ARPN Journal of Engineering and Applied Sciences

each zone. Figure-3(a) and Figure-3(g) show the original image and their Curvelet transformed image.

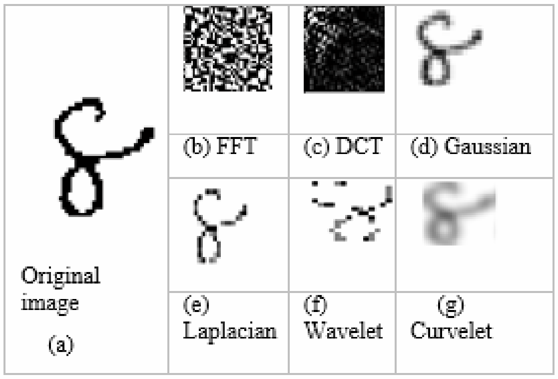


Figure-3. (a) Original image (b) Fourier transformed (c) DCT (d) Laplacian (e) Gaussian (f) Wavelet and (g) Curvelet transformed image.

CLASSIFIERS

The performance of above feature extraction techniques was compared using neural network classifiers [18] - Pattern Recognition (PR), Feed forward (FFN), Fitness Function (FFT), Cascade Neural Network (CCN) and statistical classifier: KNN (k-nearest neighbor) classification methods.

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 k^{th} and $(k-1)^{th}$ layers respectively. The element $W^k_{m_k m_{k-1}}(i,j)$, where $1 \le i \le m_k$ and $1 \le j \le m_{k-1}$, denotes the weight between the i^{th} neuron of the k^{th} layer and the j^{th} of neuron of the $(k-1)^{th}$ layer.

The output of i^{th} neuron of the k^{th} layer is a function of the i^{th} row of $W^k_{m_k m_{k-1}}$ and the output O^{k-1}, $1 \le j \le m_{k-1}$ of the $(k-1)^{th}$ layer neurons, the output of the i^{th} neuron of the k^{th} layer is computed by Equation (4)

$$O_i^k = f\left(net_i^k\right) - Eq.(4)$$

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, b^k is a column vector of size m_k where each element is a bias for k^{th} layer neurons.

In this experiment neural network were created with 10 hidden layers in all the neural network classifiers. This classifier uses logsig transfer function. This functions calculating the layers output from its input. The output layer of feed forward neural network is given by Equation (5).

$$O_i^k = f\left(net_i^k\right) = \frac{1}{1+\, e^{-net_i^k}} - Eq.(5)$$

The second classifier used was pattern recognition classifier. This function is similar to feedforwardnet except, it uses tansig transfer function in the last layer (as shown in Equation (6).

$$O_i^k = tansig\left(\, net_i^k\right) = \frac{2}{\left(1+\, e^{-2*net_i^k}\right)-1} - Eq.(6)$$

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 was 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 was function fitting neural network. This classifier uses feed forward neural network function to fit input-output relationship and returns a fitting neural network.

Statistical classifier

The classifier [19] predicts the class label of the test pattern 'x' from predefined class. The classifier finds the number of neighbor '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 this experiment used Euclidean distance metric (as shown in Equation (7)).

$$dist(X_i.Y_j) = \sqrt{\left(\sum_{i=a}^{k} [\![(\, x_{ja} - x_{ia}]\!])\right)^2} - Eq.(7)$$

EXPERIMENTS WITH CPAR-2012 DIGIT DATASET

The objective of these experiments is to compile the recognition performances of handwritten Devenagri digit recognition techniques in various transform domains for benchmark studies. All experiments were conducted on binarized, resized (32 x 32/30x30 pixels) and noise removed digit samples. In these experiments, the recognition performances measured with features starting from the direct pixel values (DPV) and their various transform domains. The transform domain starts from

VOL. 9, NO. 2, FEBRUARY 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

simple most Fourier Transform (FT), Discrete Cosine Transform (DCT), Gaussian Pyramid (GP), Laplacian pyramid (LP), Wavelet transforms (WT), and Curvelet transform (CT). For classification we choose: Pattern recognition network (PR), Feed-forward network (FFN), Fitness function network (FFT), Cascade neural network (CCN), and k-nearest neighbor (KNN) classification methods from MATLAB (R2013a).

All neural network (NN) classifier models were trained using scale conjugate back-propagation (SCG) [14] 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. We used these sets as training and test sets, as indicated in the Table-3, interchangeably, we conducted four experiments with combination of eight feature extraction techniques and five classifiers (40 experiments). In experiment I (Exp. I) the classifiers were trained on dataset set B (11, 000 samples) and tested on dataset A (24, 000 samples). In order to assess the effect of training set size on recognition, in experiment II (Exp. II), the classifiers were trained on dataset A (the smaller dataset) and tested them on dataset B (the larger dataset) while in experiment III (Exp. III) they were trained and tested on dataset A and B.

Table-3. Experimental training and test dataset.

Exp.	Training dataset	Test dataset
Exp. I	B 11, 000	A 24000
Exp. II	A 24000	B 11000
Exp. III	A and B 35, 000	A and B 35, 000

The five classifiers were tested on direct image zoning with various parameters so as to choose an appropriate parameter for each feature extraction technique. For this, image is divided into various blocks starting from 2x2, 3x3, 4x4, 5x5, 6x6 and 8x8 to 8x8. The feature vectors were formed by storing the average value of pixels intensity in each zones forming a feature vector of length 256, 100, 64, 36, 25 and 16 (See Figure-4).

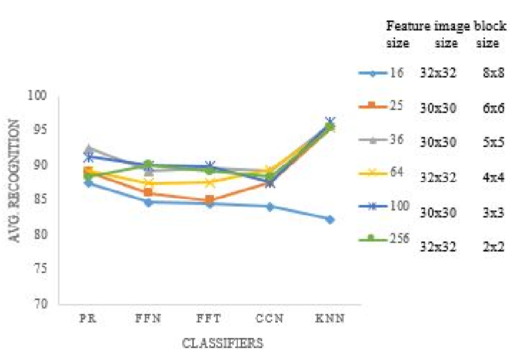


Figure-4. Recognition accuracy on various feature and block sizes.

From result it is clear that if the image is divided into 5x5 zones, it gives better result. Further Exp. (I-III) were conducted with this size. From Exp. I (see Figure-5) it is clear that KNN classifier yielded best results in all features. Fourier transform and DCT transform yielded poor results. Gaussian pyramid yielded highest recognition results. All other feature gives result in between.

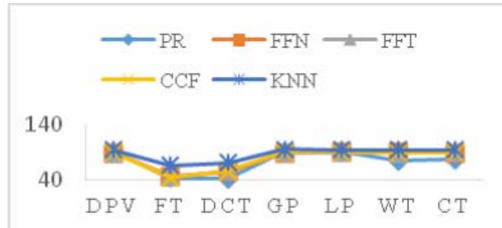


Figure-5. Recognition accuracy of Exp. I.

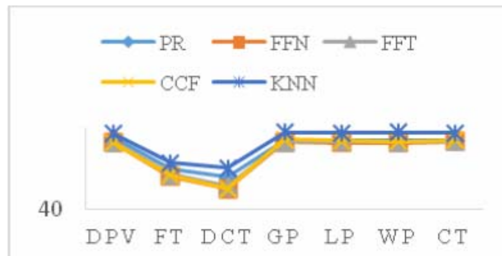


Figure-6. Recognition accuracy of Exp. II.

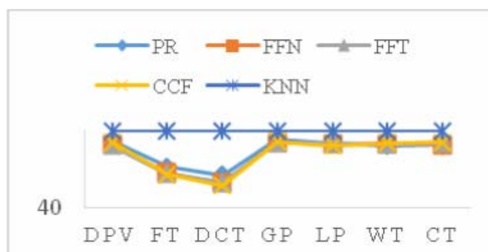


Figure-7. Recognition accuracy of Exp. III.

Exp. II (see Figure-6) shows that KNN classifier performed best in all. Gaussian pyramid yielded better results than all other features. Like before FFT and DCT yielded poor results. All other feature performed in between.

From the Exp. III (see Figure-7) it is clear that KNN classifier yielded 100 % result on training set samples. Also recognition results in this experiment are higher than previous experiments this clearly indicates that training set size affects the recognition results.

Table-4 shows the execution time in seconds. The first column specifies the technique used for feature extraction, second column shows time taken for computing features of all samples in seconds and last column specifies time taken for classifying these samples in Exp. I, II and III using KNN classifier. This shows that Curvelet transform took more time for extracting features whereas Gaussian pyramid took less time. All other

VOL. 9, NO. 2, FEBRUARY 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

techniques lie in between. It also clear that as we increase the test size (see Exp. II and III) it takes more classification time. So we can also say that test size also affects the recognition time.

In an attempt to improve the digit recognition accuracy of Exp. I and II. We applied the majority voting [20] techniques. These techniques were applied using the results obtained by the four best performing features - Curvelet transform (CT), Wavelet transforms (WT), direct pixel values (DPV), Laplacian pyramid (LP) and Gaussian pyramid (GP) (see Table-5).

The majority voting classifier rejected the conflicting samples (having no majority). Thereby in this case the recognition accuracy was improved. In Exp. I and II the Gaussian pyramid feature yielded highest recognition accuracy 95.2% and 96.93% respectively whereas the accuracy of majority voting increased to 98.02 % at the expense of 0.26% rejection rate. Some of the rejected samples are shown in Figure-8. In this case only 1.98% samples were misclassified. Before majority voting we tried to use all the features but the accuracy was only increased to 97.2% which is slightly higher than Gaussian pyramid feature but lower than majority voting.

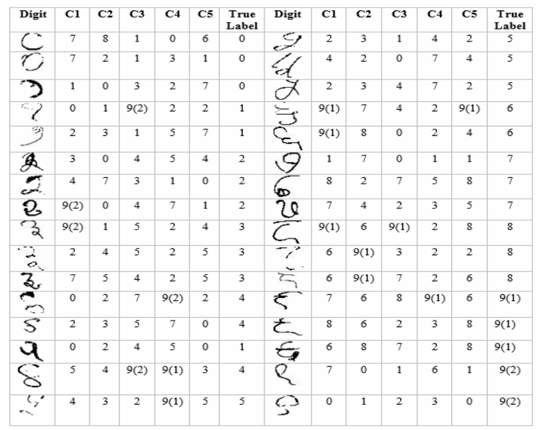
Table-4. Feature extraction and classification execution time.

Technique	Feature extraction time (seconds)	Classification time (seconds)		
		Exp. I	Exp. II	Exp. III
DPV	78.05	26.44	27.19	137.96
FT	107.21	27.35	28.82	136.55
DCT	74.17	26.77	28.65	130.45
GP	20.93	11.43	12.98	62.94
LP	104.98	25.27	27.22	123.22
WT	486.33	26.69	30.61	135.71
CT	3000	27.16	28.59	115.16

Table-5. Recognition accuracy of features using KNN classifier for majority voting.

Feature	Exp. I	Exp. II
CT	94.2	96.41
WT	94.3	96.55
DPV	94.67	95.88
LP	94.8	96.31
GP	95.2	96.93

From this confusion matrix (see Figure-9) it is clear that in Devnagari script there is shape similarity among the digits. The Figure-10 shows this shape similarity because maximum times digit zero is confused with seven, one is confused with seven and nine (2), two is confused with three, three is confused with two, four is confused with five, five is confused with four, six is confused with nine (1), seven is confused with zero and one, eight is confused with two and nine (1), nine (1) is confused with six and nine (2) is confused with one.



Digit	C1	C2	C3	C4	C5	True Label	Digit	C1	C2	C3	C4	C5	True Label
	7	8	1	0	6	0		2	3	1	4	2	5
	7	2	1	3	1	0		4	2	0	7	4	5
	1	0	3	2	7	0		2	3	4	7	2	5
	0	1	9(2)	2	2	1		9(1)	7	4	2	9(1)	6
	2	3	1	5	7	1		9(1)	8	0	2	4	6
	3	0	4	5	4	2		1	7	0	1	1	7
	4	7	3	1	0	2		8	2	7	5	8	7
	9(2)	0	4	7	1	2		7	4	2	3	5	7
	9(2)	1	5	2	4	3		9(1)	6	9(1)	2	8	8
	2	4	5	2	5	3		6	9(1)	3	2	2	8
	7	5	4	2	5	3		6	9(1)	7	2	6	8
	0	2	7	9(2)	2	4		7	6	8	9(1)	6	9(1)
	2	3	5	7	0	4		8	6	2	3	8	9(1)
	0	2	4	5	0	1		6	8	7	2	8	9(1)
	5	4	9(2)	9(1)	3	4		7	0	1	6	1	9(2)
	4	3	2	9(1)	5	5		0	1	2	3	0	9(2)

Figure-8. Some of the rejected samples.

Digit	0	1	2	3	4	5	6	7	8	9(1)	9(2)	%age
0	1001	3	0	0	0	0	0	5	0	0	1	99.11
1	0	991	2	1	0	0	0	6	1	1	7	98.22
2	4	7	974	14	0	3	1	0	1	0	1	96.92
3	0	1	7	998	0	1	0	1	0	0	2	98.81
4	0	1	0	1	981	10	2	0	5	3	5	97.32
5	0	2	5	6	8	985	0	1	0	3	1	97.43
6	0	0	1	1	1	4	982	1	4	10	4	97.42
7	5	1	3	1	1	1	1	995	1	1	1	98.42
8	0	1	4	2	0	0	0	0	997	4	2	98.71
9(1)	0	0	2	2	0	0	9	1	0	992	4	98.22
9(2)	4	7	3	1	0	0	2	1	1	2	858	97.61
Average recognition accuracy												98.02

Figure-9. Confusion matrix after majority vote classification.

Figure-10. Shape similarity between handwritten digit samples.

CONCLUSIONS

In this paper the performance of feature extraction on Devnagari handwritten digit recognition techniques in transform domain is evaluated. The performance measured in terms of (a) recognition accuracy (b) recognition time and (c) training size effects on new large scale dataset CPAR-2012. For measuring the performance we used features: direct pixel value, Fourier Transform, Discrete Cosine Transform, Gaussian pyramid, Laplacian pyramid, Wavelet pyramid and Curvelet transform feature on five 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 classifier ensemble scheme using majority voting scheme were used. The recognition accuracy improved from 96.93 % to 98.02% by this scheme by rejecting 0.26% samples. This indicates that isolated handwritten digit recognition is still a challenging problem. Since there is not much work done on this language so results of our work will serve as benchmark for future research in this fields. Moreover we have introduced a rejection criteria based on majority voting scheme.

ACKNOWLEDGEMENT

The author would like to acknowledge the support of late prof. P. Ahmed and all Sharda University towards the development of the database and recognition techniques described in this paper.

REFERENCES

[1] R. Jayadevan, Satish R. Kolhe, Pradeep M. Patil and Umapada Pal. 2001. Offline Recognition of Devanagari Script: A Survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C. 41(6): 782-796.

[2] P. M. Patil and T. R. Sontakke. 2007. Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network. Pattern Recognition. 40: 2110-2117.

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

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

[5] Reena Bajaj, Lipika Dey and S. Chaudhury. 2002. Devnagari numeral recognition by combining decision of multiple connectionist classifiers. Sadhana. 27(1): 59-72.

[6] M. Hanmandlu and O. V. R. Murthy. 2007. Fuzzy model based recognition of handwritten numerals. Pattern Recognition. 40: 1840-1854.

[7] P. Ahamed and Yousef Al-Ohali. 2010. TAR based shape features in unconstrained handwritten digit recognition. WSEAS Transactions on Computers. 9(5): 419-428.

[8] R. J. Ramteke and S. C. Mehrotra. 2006. Feature extraction based on moment invariants for handwriting recognition. In Proc. IEEE Conf. Cybern. Intell. Syst. pp. 1-6.

[9] U. Garain, M. P. Chakraborty and D. Dasgupta. 2006. Recognition of handwritten Indic script digits using clonal selection algorithm. In Lecture Notes in Computer Science 4163, H. Bersini and J. Carneiro, Eds. New York: Springer-Verlag. pp. 256-266.

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

[11] U. Bhattacharya and B. B. Chaudhuri. 2009. Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. IEEE Trans. Pattern Analysis and Machine Intelligence. 31(3): 444-457.

[12] U. Pal, T. Wakabayashi, N. Sharma and F. Kimura. 2007. Handwritten numeral recognition of six popular Indian scripts. In: Proc. of 9th Conf. Document Anal. Recognit. pp. 749-753.

[13] C. V. Lakshmi, R. Jain and C. Patvardhan. 2007. Handwritten Devnagari numerals recognition with higher accuracy. In: Proc. Int. Conf. Comput. Intell. Multimedia Appl. pp. 255-259.

[14] Møller Martin Fodslette. 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural networks. 6(4): 525-533.

[15] Zhen Xiantong and Shao Ling. A local descriptor based on Laplacian pyramid coding for action recognition. Pattern Recognition Letters. 34(15): 1899-1905.

[16] Stephane G. Mallat. 1989. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 11(7): 674-693.

[17] Emmanuel Candes, Laurent Demanet, David Donoho and Lexing Ying. 2006. Fast discrete curvelet transforms. Multiscale Modeling and Simulation. 5(3): 861-899.

[18] http://www.mathworks.in/help/nnet/pattern-recognition-and-classification.html.

[19] Cover Thomas and Peter Hart. Nearest neighbor pattern classification. Information Theory, IEEE Transactions on. 13(1): 21-27.

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