Classification Performance Analysis of MNIST Dataset utilizing a Multi-resolution Technique

Ramesh Kumar Mohapatra, Banshidhar Majhi, and Sanjay Kumar Jena
Department of Computer Science and Engineering
National Institute of Technology Rourkela
Odisha, 769008, India
{mohapatrark, bmajhi, skjena}@nitrkl.ac.in

***Abstract*—Here, we propose a method for recognition of handwritten English digit utilizing discrete cosine space-frequency transform known as the Discrete Cosine S-Transform (DCST). Experiments have been conducted on the publicly availabe standard MNIST handwritten digit database. The DCST features along with an Artificial Neural Network (ANN) classifier is utilized for solving the classification issues of written by hand digit. The Discrete Cosine S-Transform coefficients are extracted from the standard images of MNIST handwritten isolated digit database. The database consists of a total of 70000 including 60000 training samples and 10000 test samples. To overcome the computational overhead, we have normalized all the images of the MNIST dataset from 28 $\times$ 28 to 20 $\times$ 20 image size by eliminating the unsought boundary pixels up to width four. Further, the classification of digits has been made by using a back propagation neural network (BPNN). This work has achieved precisely 98.8% of success rate for MNIST database.**

Keyword: Classification; MNIST dataset; Discrete Cosine S-Transform; BPNN.

I. Introduction

The problem of optical scanned character recognition [1] is to construe as well as translate apprehensible handwritten input automatically, which is of great interest as well as a gainsay in the field of object identification. It has numerous down to earth applications in real time e-processing of data, such as automatic address reading, postal mail sorting through ZIP/PIN code, bank check processing, etc. As a benchmark, the Modified National Institute of Standards and Technology (MNIST) dataset has been used often times to design afresh digit recognition system [2], [3]. In the literature study, we found a great amount of works based on MNIST dataset, suggesting many different methods [4]–[6]. Lauer et al. [7] have proposed trainable feature extractors for handwritten digit recognition. A comprehensive examination by Liu et al. [8] compares the performance of previously proposed classifiers including linear and polynomial classifier, Nearest Neighbor classifier, and different neural networks. Normally individuals don't generally compose the same digit in the very same way at a given purpose of time. Due to this within class variance of the shape of a character, it is a major challenge for the researchers to classify the character. Many feature extraction approaches such as biologically inspired features [9], higher order singular value decomposition [10], GA-based feature selection approach [11], Fuzzy model based recognition [12] have been proposed to specify the shape invariance within a class to improve the discrimination ability. It is observed that the accuracy and efficiency of many classifiers could be improved substantially by extracting direction features, local structural or curvature feature [13]. Kumar et al. [22] used mathematical morphology techniques to divide the dataset into two groups and further classify those considering the structural features of the digits. The overall recognition rate recorded as 92.5%. The system fails to identify the digits due to the broken lines with large gap, incomplete digits and the digits with uneven strokes. Also, many multi resolution techniques are used for the classification of English digit. S-transform is one of the multi resolution techniques that extracts multi-scale resolution like discrete wavelet transform [17]. It has been widely used in image restoration and texture analysis [16]. One of the major limitations of S-transform is its exponential computational overhead. As of late a speedier adaptation of the S-transform,in particular, discrete orthonormal Stockwell transform (DOST) has been intended for feature extraction. We have utilized the discrete cosine transform (DCT) rather than fast Fourier transformation in DOST which prompts Discrete Cosine S-Transform (DCST).

The remaining piece of the paper is composed as takes after. Section II gives a full depiction of the database utilized. The discrete cosine S-transform examined briefly in Section III-A. The proposed method is depicted in Section IV. Section V depicts the outcomes took after by finishing up concluding remark in Section VI.

Fig. 1: 100 sample digit images of MNIST dataset

TABLE II: Total number of samples for each digit in the test set.

Test set	0	1	2	3	4	5	6	7	8	9
Frequency	1001	1127	991	1033	980	862	1014	1070	944	978

II. Database Used

The MNIST database was gotten from NIST's Special Database 3 (SD-3) and Special Database 1 (SD-1) [2] that comprises of binary images of written by hand digits ranging from 0 to 9. These are collected by high school students and employees of the United States Census Bureau. A total of 30,000 patterns from SD-3 and 30,000 from SD-1 are gathered aimlessly to have 60000 samples in the training set. Test set comprises of 10,000 samples out of which 5,000 are chosen from SD-3 and the rest are from SD-1. The MNIST dataset contains gray-scale images of size 28×28. Thus, the dimensionality of each image sample vector is 784. Hundred sample images from the MNIST database are appeared in Fig. 1. Since the samples are as of now standardized along these lines, a little measure of preprocessing is required. For our experiment we have normalized further all the images of the MNIST dataset to 20×20 image size by killing the unsought boundary pixels up to width four. Accordingly, the dimensionality of every image test vector gets to be 400 lengths. The MNIST database is publicly available at the homepage of LeCun [3] and these are shown in Table I along with their sizes in bytes. The distribution of each digit of the test set is shown in Table II.

TABLE I: MNIST dataset files and their sizes

Files	Size in Bytes
Training set images	9912422
Training set labels	28881
Test set images	1648877
Test set labels	4542

III. Feature Extraction

The proposed scheme uses a multi-resolution technique i.e., 2D discrete cosine S-transform (DCST) for feature extraction, which is a variation of Stockwell transform (ST) [18]. Each image from MNIST database is resized to 20×20, further the DCST coefficients are calculated at each pixel to get a feature length of 400.

A. Discrete Cosine S-Transform

The ST gives a full time decomposition of a signal and retains absolutely referenced phase information. Likewise, the ST is evinced in the Cosine domain as,

$$s(\tau, f) = \int_{-\infty}^{\infty} C(\psi + f) e^{-\frac{(2\pi\psi)^2}{2f^2}} e^{i2\pi\psi\tau} d\psi \tag{1}$$

where $C(f)$ is the Cosine spectrum of $c(t)$ and other parameters have their usual meaning. The discretization of (1) leads to the Discrete Stockwell Transform (DST) and is given by,

$$S[k, n] = \sum_{m=0}^{N-1} e^{-\frac{2\pi^2 m^2}{n^2}} H[m+n] e^{\frac{i2\pi mk}{N}} \tag{2}$$

where k is the time translation and n is the index of frequency shift with $n \neq 0$. Here $H(.)$ is the DCT of $h(.)$. The 2D-DST of an image of size $N \times N$ has the growth function $O[N^4 + N^4 log(N)]$. Due to high complexity and redundant information of S-transform, it is not frequently used in many applications. Notwithstanding, DCST can be utilized to speak to information effectively, which is the orthonormal variant of the DST, creating N point time recurrence representation for a signal of length N. In this way, DCST give features with zero data redundancy. The voice frequencies (v_x, v_y) are obtained within the bandwidth of $2^{p_x - 1} \times 2^{p_y - 1}$. If we represent the voice frequencies as a complex image where, v_x corresponds to real and v_y corresponds to imaginary part then the magnitude and phase angle is obtained by $M_v = \sqrt{v_x^2 + v_y^2}$ and $\theta_v = tan^{-1}(\frac{v_y}{v_x})$ respectively. Hence, for each sample, we obtained 400 DCST coefficients where the size of the image is 20×20. So, our feature matrix size for m samples is $m \times 400$.

B. Classification Phase

We have considered the back propagation neural network (BPNN) for the classification of the test set. The Artificial Neural Network (ANN) used has three layers as shown in the Fig. 2. The DCST features are extracted from each sample in the training set. The network is trained considering the samples from the training set. We fixed a total of 400 (i.e., n = 400) neurons in the input layer and since we have ten classes to classify, the ANN has 10 neurons in the output layer (i.e., m = 10). For better convergence performance, the neurons in the hidden layer are appositely fixed during the experiment.

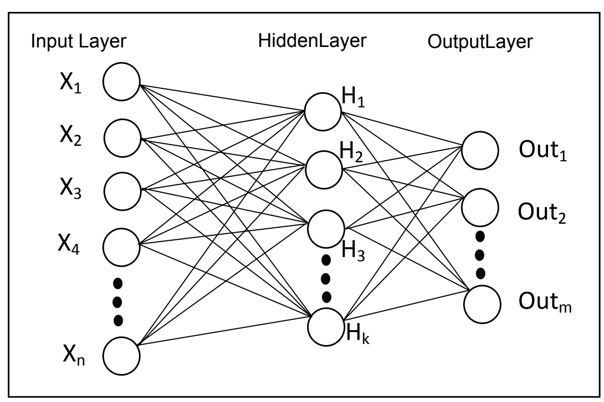

Fig. 2: General structure of a three layer ANN of size (n-k-m)

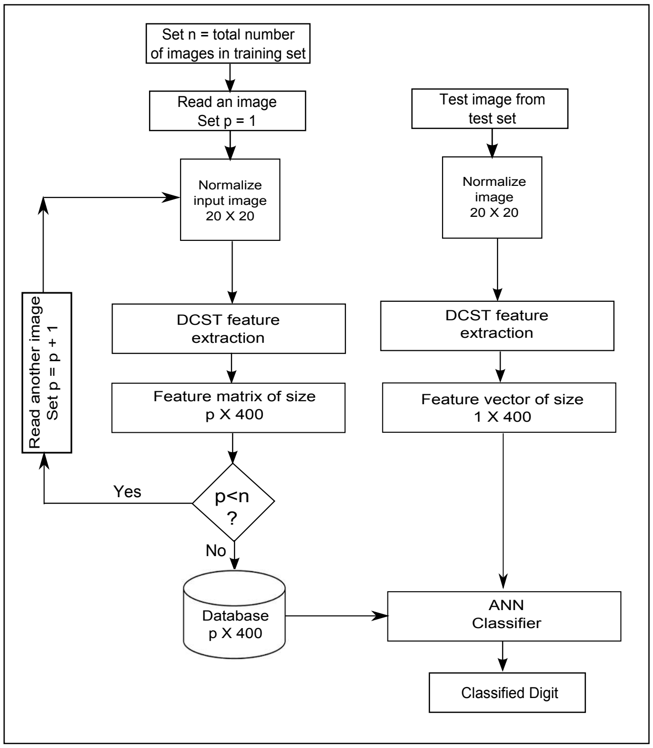

Fig. 3: Proposed scheme

IV. Proposed Method

Our proposed method is illustrated in the Fig. 3. The 400 DCST feature vector of all normalized images in the training set is extracted and stored in a feature matrix, leading to a size of 60000×400. The feature matrix along the associated labels is trained using the neural network tool in MATLAB to produce a network *'nn'*. Now, the process is repeated for each sample from test set and a class is assigned to each sample by performing a pattern matching over the network i.e., *nn*. To perform simulation, we have assumed the following:

1) The *dividerandom*() method is used to split the data set into three sets.
2) For faster training, scaled conjugate gradient method *trainscg* is used.
3) The train of the neural network stops when either the gradient reaches a value of 10^{-6} or the number of consecutive validation checks reaches 6, i.e., there is no better improvement of the gradient for six consecutive iterations.
4) In testing dataset, 70% samples are used to train the network, 15% are used for validation and the rest are used for testing the network.

The performance of the classification is evaluated with the assistance of confusion matrix that use to assess the quality of the output of a classifier on a given data set. The performance measures like true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are ascertained which thus assesses the $sensitivity$, $specificity$, $fall\ out$, and $miss\ rate$ by the following equations.

$$Sensitivity\ (or\ TPR) = \frac{TP}{TP+FN} \tag{3}$$

$$Specificity\ (or\ TNR) = \frac{TN}{TN+FP} \tag{4}$$

$$Fall\ out\ (or\ FPR) = \frac{FP}{TN+FP} \tag{5}$$

$$Miss\ rate\ (or\ FNR) = \frac{FN}{TP+FN} \tag{6}$$

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{7}$$

The accuracy of the network can be calculated from the Equation 7.

TABLE III: Confusion values for MNIST database

Digit	False Negative Rate	False Positive Rate	True Positive Rate	True Negative Rate
0	0.0009	0.0090	0.9910	0.9991
1	0.0015	0.0107	0.9893	0.9985
2	0.0024	0.0232	0.9768	0.9976
3	0.0028	0.0156	0.9844	0.9972
4	0.0017	0.0193	0.9807	0.9983
5	0.0016	0.0208	0.9792	0.9984
6	0.0009	0.0118	0.9882	0.9991
7	0.0015	0.0149	0.9851	0.9985
8	0.0024	0.0202	0.9798	0.9976
9	0.0020	0.0154	0.9846	0.9980

V. Results and Discussion

In the supervised learning of the MNIST dataset [19] we utilize 10,000 specimens out of 70000 accessible examples to decrease the computational expense. Out of these, 7000 examples are decided for training, 1500 are for validation and the rest are utilized for testing. Experimentally, the number of neurons in the hidden layer of the network is observed to be eighty for better convergence characteristics (Fig. 6). Subsequently the ANN classifier has the structure of 400-80-10. The performance parameters are recorded in the Table III demonstrating the sensitivity, specificity, fall-in and fall out separately and the curve between the cross-entropy and number of epochs is plotted and appeared in Fig. 4. The

TABLE IV: Confusion matrix for MNIST dataset

Digit	0	1	2	3	4	5	6	7	8	9
0	993	0	0	1	1	1	1	1	1	2
1	0	1114	5	0	2	2	0	0	2	2
2	1	5	969	4	1	3	0	3	5	0
3	2	0	8	1008	0	6	0	4	4	1
4	0	0	4	0	965	1	2	0	2	6
5	1	0	0	5	0	847	5	0	4	0
6	3	1	1	0	3	0	1006	0	0	0
7	0	3	3	3	1	0	0	1057	0	3
8	2	2	2	2	0	4	4	5	922	1
9	0	1	0	1	11	1	0	3	1	960

receiver operating characteristic (ROC) curves for all ten classes are shown in one place (see Fig. 5). The region under the ROC curve gives the precision too. The more the curve towards the upper left corner the more is system's exactness. The confusion matrix for testing patterns is appeared in the Table IV. It is observed that proposed scheme gives an overall accuracy of 98.8%. The portion of samples misclassified is 0.012 i.e., 1.2% error. We have also compared our scheme with some of the existing schemes that is portrayed in Table V.

TABLE V: Other results on MNIST.

Scheme	Over all Error rate(%)
Structural features [22]	7.5
Nearest neighbor classifier (Euclidean)	3.09
Higher order singular value decomposition [10]	2.0
Convolutional Net LeNet-1 [20]	1.7
Our scheme	**1.2**
Polynomial SVM [20]	1.1
Direction gradient [21]	0.42
Human [19]	0.2

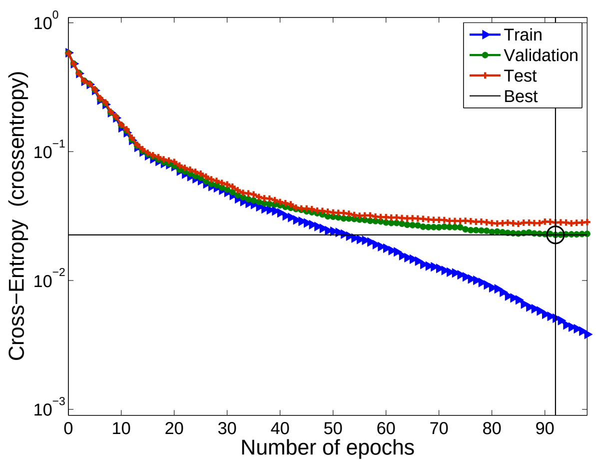

Fig. 4: Performance curve of MNIST dataset

VI. Conclusion

Our preliminary result shows that the discrete cosine S-transform features alongside back propagation neural network is adequately descriptive for the purpose of classification of handwritten English digits of the MNIST dataset. This demonstrates viability of the proposed method over some of the existing schemes. Still, further improvement is conceivable to accomplish more exactness than recommended in this paper.

Acknowledgment

Special thanks to LeCun for being the pioneer of making the MNIST dataset and made it available for research purpose.

References

[1] Richard O Duda, Peter E Hart, et al. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[2] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *Signal Processing Magazine, IEEE*, 29(6):141–142, Nov 2012.

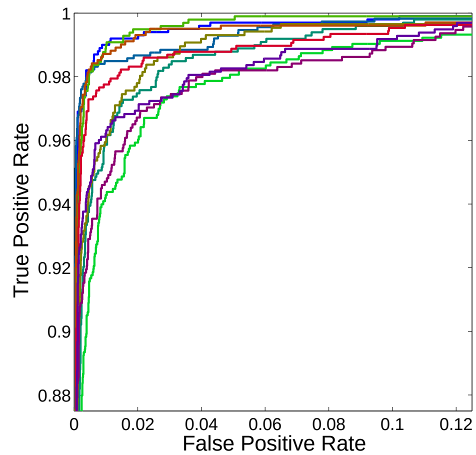

Fig. 5: ROC curves for 10 classes of MNIST dataset

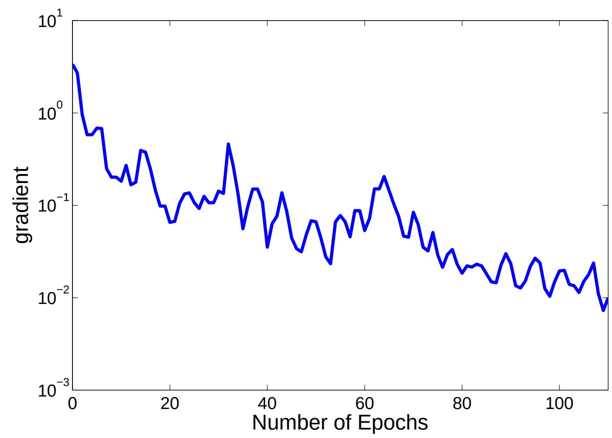

Fig. 6: Training convergence characteristics of MNIST dataset

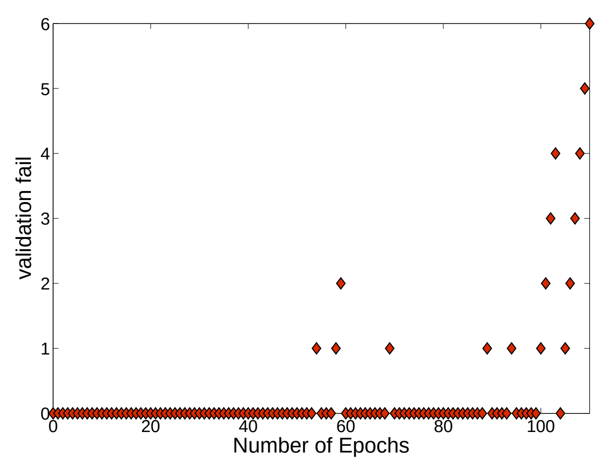

Fig. 7: Validation checks during the training of MNIST dataset

[3] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits.

[4] Ernst Kussul and Tatiana Baidyk. Improved method of handwritten digit recognition tested on {MNIST} database. Proceedings from the 15th International Conference on Vision Interface. *Image and Vision Computing*, 22(12):971 – 981, 2004.

[5] Zhihong Man, Kevin Lee, Dianhui Wang, Zhenwei Cao, and Suiyang Khoo. An optimal weight learning machine for handwritten digit image recognition. Special issue on Machine Learning in Intelligent Image Processing. *Signal Processing*, 93(6):1624 – 1638, 2013.

[6] Ping Zhang, Tien D. Bui, and Ching Y. Suen. A novel cascade ensemble classifier system with a high recognition performance on handwritten digits. *Pattern Recognition*, 40(12):3415 – 3429, 2007.

[7] Fabien Lauer, Ching Y. Suen, and Grard Bloch. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6):1816 – 1824, 2007.

[8] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*, 36(10):2271 – 2285, 2003.

[9] Angelo Cardoso and Andreas Wichert. Handwritten digit recognition using biologically inspired features. *Neurocomputing*, 99(0):575 – 580, 2013.

[10] Berkant Savas and Lars Eldn. Handwritten digit classification using higher order singular value decomposition. *Pattern Recognition*, 40(3):993 – 1003, 2007.

[11] C. De Stefano, F. Fontanella, C. Marrocco, and A. Scotto di Freca. A ga-based feature selection approach with an application to handwritten character recognition. Frontiers in Handwriting Processing. *Pattern Recognition Letters*, 35(0):130 – 141, 2014.

[12] M. Hanmandlu and O.V. Ramana Murthy. Fuzzy model based recognition of handwritten numerals. *Pattern Recognition*, 40(6):1840 – 1854, 2007.

[13] Jie Zhou, Adam Krzyzak, and Ching Y. Suen. Verificationa method of enhancing the recognizers of isolated and touching handwritten numerals. Handwriting Processing and Applications. *Pattern Recognition*, 35(5):1179 – 1189, 2002.

[14] Ying Wen and Lianghua He. A classifier for bangla handwritten numeral recognition. *Expert Systems with Applications*, 39(1):948 – 953, 2012.

[15] Hossein Khosravi and Ehsanollah Kabir. Introducing a very large dataset of handwritten farsi digits and a study on their varieties. *Pattern Recognition Letters*, 28(10):1133 – 1141, 2007.

[16] Robert Glenn Stockwell. A basis for efficient representation of the s-transform. *Digital Signal Processing*, 17(1):371–393, 2007.

[17] Rafael C Gonzalez. *Digital image processing*. Pearson Education India, 2009.

[18] Yanwei Wang and Jeff Orchard. The discrete orthonormal stockwell transform for image restoration. In *16th IEEE International Conference on Image Processing (ICIP)*, pages 2761–2764. IEEE, 2009.

[19] Yann LeCun, LD Jackel, L Bottou, A Brunot, C Cortes, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on Artificial neural networks*. volume 60, pages 53–60, 1995.

[20] Lecun, Y. and Bottou, L. and Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*. volume 86(11), pages 2278–2324, 1998.

[21] Liu, Cheng-Lin and Nakashima, Kazuki and Sako, Hiroshi and Fujisawa, Hiromichi Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition*. volume 36(10), pages 2271–2285, 2003.

[22] Kumar, V Vijaya and Srikrishna, A and Babu, B Raveendra and Mani, M Radhika. Classification and recognition of handwritten digits by using mathematical morphology. *Sadhana*. volume 35(4), pages 419–426, 2010.