

# AN EFFICIENT FEATURE EXTRACTION AND CLASSIFICATION OF HANDWRITTEN DIGITS USING NEURAL NETWORKS

<sup>1</sup>N. VenkateswaraRao, <sup>2</sup>Dr. A. Srikrishna, <sup>3</sup>Dr. B. RaveendraBabu,  
<sup>4</sup>G. Rama Mohan Babu

<sup>1</sup>Dept. of Computer Science & Engineering, <sup>2,4</sup>Dept. of Information Technology,  
R.V.R. & J.C. College of Engineering, Guntur, INDIA

<sup>3</sup>Director (Operations), M/s. Delta Technologies (P) Ltd., Hyderabad, INDIA

<sup>1</sup>vraonaramala@gmail.com, <sup>2</sup>atulurisrikrishna@yahoo.com,

<sup>3</sup>rbhogapathi@yahoo.com, <sup>4</sup>rmbgatram@gmail.com

## ABSTRACT

*The wide range of shape variations for handwritten digits requires an adequate representation of the discriminating features for classification. For the recognition of characters or numerals requires pixel values of a normalized raster image and proper features to reach very good classification rate. This paper primarily concerns the problem of isolated handwritten numeral recognition of English scripts. Multilayer Perceptron (MLP) classifier is used for classification. The principal contributions presented here are preprocessing, feature extraction and multilayer perceptron (MLP) classifiers. The strength of our approach is efficient feature extraction and the comprehensive classification scheme due to which, we have been able to achieve a recognition rate of 95.6, better than the previous approaches.*

## Keywords:

*Binarization, Filtering, Thinning, Feature extraction, Neural Networks.*

## 1. INTRODUCTION

OFFLINE handwritten text recognition is one of the most active areas of research in computer science and it is inherently difficult because of the high variability of writing styles. High recognition rates are achieved in character recognition and isolated word recognition, but we are still far from achieving high-performance recognition systems for unconstrained offline handwritten texts [1-7].

Automatic handwriting recognition systems normally include several preprocessing steps to reduce variation in the handwritten texts as much as possible and, at the same time, to preserve information that is relevant for recognition. There is no general solution to preprocessing of offline handwritten text lines, but it typically relies on slope and slant correction and normalization of the size of the characters. With the slope correction, the handwritten word is horizontally rotated such that the lower baseline is aligned to the horizontal axis of the image. Slant is the clockwise angle between the vertical direction and the direction of the vertical text strokes. Slant correction transforms the word into an upright position. Ideally, the removal of

slope and slant results in a word image that is independent of these factors. Finally, size normalization tries to make the system invariant to the character size and to reduce the empty background areas caused by the ascenders and descenders of some letters.

A major obstacle to research on handwritten character recognition of Indian scripts is the nonexistence of standard/benchmark databases. Previous studies were reported on the basis of small databases collected in laboratory environments. However, any fruitful work in this area primarily needs a benchmark database. Several standard databases, such as NIST, MNIST [9], CEDAR [10], and CENPARMI, are available for Latin numerals. Similar databases like [11, 12], and [13] exist for a few other scripts also.

This paper presents new techniques to normalize the size of the images, feature extraction and classification of handwritten digits using Artificial Neural Networks (ANNs). A MNIST database has been used in all the experiments; thus skew correction and line detection are skipped in this work.

## 2. PREPROCESSING AND FEATURE EXTRACTION

Handwritten image normalization from a scanned image includes several steps, which usually begin with image cleaning, page skew correction, and line detection [8]. A MNIST database has been used in all the experiments; thus page skew correction and line detection are skipped in this work. With the handwritten digit images, several preprocessing steps to reduce variations in writing style are usually performed i.e. noise removal, character size normalization and skeletonization. The algorithm for preprocessing is given in Algorithm 1.

### Algorithm 1: Preprocessing

*Input: Isolated handwritten grayscale image*

*Output: Preprocessed image*

#### Methodology

*Step 1: Read a gray scale image*

*Step 2: Remove the noise using median filter*

*Step 3: Convert the gray scale image to binary image using threshold value*

*Step 4: Normalize the image to a predefined size of 40×40*

*Step 5: Convert the normalized image to a single pixel thickness using thinning.*

### 2.1.Noise Removal Algorithm

Images taken with both digital cameras and conventional film cameras will pick up noise from a variety of sources. Many further uses of these images require that the noise will be (partially) removed - for aesthetic purposes as in artistic work or marketing, or for practical purposes such as computer vision.

A median filter is an example of a non-linear filter and, if properly designed, is very good at preserving image detail.

1. Consider each pixel in the image.
2. Sort the neighboring pixels in order based upon their intensities.
3. Replace the original value of the pixel with the median value from the list.

## 2.2.Binarization using Global Thresholding

Thresholding is a process of converting a grayscale input image to a bi-level image by using an optimal threshold.

1. Select an initial estimate for T. (A suggested initial estimate is the midpoint between the minimum and maximum intensity values in the image.)
2. Segment the image using T. This will produce two groups of pixels: G<sub>1</sub>, consisting of all pixels with intensity values ≥ T, and G<sub>2</sub>, consisting of pixels with value < T.
3. Compute the average intensity values μ<sub>1</sub> and μ<sub>2</sub> for the pixels in regions G<sub>1</sub> and G<sub>2</sub>.
4. Compute the new threshold value:  $T = \frac{1}{2}(\mu_1 + \mu_2)$ .
5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T<sub>0</sub>.

## 2.3.Character size normalization

Handwriting produces variability in size of written digits. This leads to the need of scaling the digit size within the image to a standard size, as this may leads to better recognition accuracy. Here we are normalizing the size of digit to a 40×40.

## 2.4.Thinning Algorithm

The thinning operation is related to the Hit-or-Miss transform and can be expressed quite simply in terms of it [15]. The thinning of an image A by a structuring element B is:

$$A \otimes B = A - (A \circledast B) \quad (1)$$

Where B is defined as  $B = \{B^1, B^2, \dots, B^8\}$ , the subtraction is a *logical subtraction* defined by.

$$X - Y = X \cap Y^c \quad (2)$$

and  $A \circledast B$  is the Hit-or-Miss transform of image A by structuring element B.

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad (3)$$

Where B<sub>1</sub> is a structuring element and B<sub>2</sub> is complement of B<sub>1</sub>.

The algorithm for thinning is as follows:

1. Input a binary image A.
2. Copy the image A into X.
3. Initialize the i=1.
4. Copy X into temp.
5. Apply the thinning process using equation. (1)with B<sup>i</sup>as structuring element and store in X.
6. Increment i by 1.
7. For getting the structuring element B<sup>i</sup>Rotate the structuring element B<sup>i-1</sup> clockwise with 45° angles shown infigure 1.
8. Repeat the processfrom step 4 until temp=X
9. Output the thinned image from X.

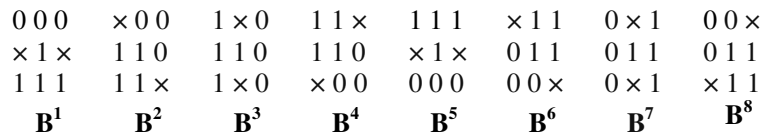


Figure 1: Sequence of rotated structuring elements used for Thinning, ×indicates don't care condition

## 2.5.Feature Extraction

After preprocessing, a feature extraction method is applied to capture the most relevant characteristics of the character to be recognized [16, 17]. This algorithm identifies 16 features by applying a grid to the image and computing values by moving diagonally as shown in figure 2(c). The complete process is shown in Algorithm 2.

### Algorithm 2: Feature Extraction

**Input:** Preprocessed image

**Output:** Feature vector containing 16 features.

#### Methodology

*Step 1: Read a preprocessed image.*

*Step 2: Every character image of size 40×40 pixels is divided into 16 equal zones, each zone of size 10×10 pixels sub image.*

*Step 3: The features are extracted from each zone pixels by moving along the diagonals of its respective 10×10 pixels.*

*Step 4: Each zone has 19 diagonal lines and the foreground pixels present along each diagonal line is summed to get a single sub-feature and thus 19 sub-features are obtained from the each zone*

*Step 5: These 19 sub-features values are averaged to form a single feature value and placed in the corresponding zone. This procedure is sequentially repeated for all the zones of image.*

*Step 6: Finally, 16 features are extracted for each image.*

## 3. ARTIFICIAL NEURAL NETWORK MODEL

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons [18].

Figure 2: Diagonal feature extraction scheme

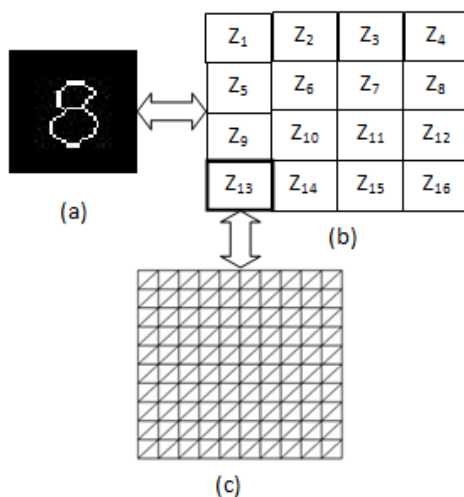
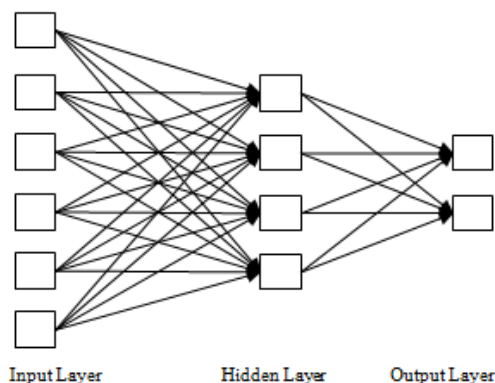


Figure 3: An example of a simple feed forward network



### 3.1. Classification using Neural Networks

There are two distinct stages when using ANNs to process any kind of data, a training stage and a classification stage. These two stages can be framed in terms of handwriting recognition. In the classification stage, samples are passed as inputs into the ANN, resulting in an output representing what the ANN believes to be the most correct output. However, to be successful, classification must be preceded by a training stage in which the ANN is given a set of inputs and the corresponding set of correct outputs. In this way, the ANN is able to adapt by adjusting its weights. In other words, the ANN “learns” to associate the correct output with each corresponding input [14].

### 3.2. Training the ANN Model

A well known Multi Layer Perceptron (MLP) is used for Training and Testing. The complete process is given in Algorithm 3.

#### Algorithm 3: Neural Network Training

The training algorithm of back propagation involves four stages:

1. Initialization of weights
2. Feed forward
3. Back Propagation of errors
4. Updating the weights and biases.

#### Method

##### Initialization of weights

- Step 1: Initialize weight to small random values.  
 Step 2: While stopping condition is false, do Steps 3-10  
 Step 3: For each training pair do Steps 4-9

##### Feed Forward

- Step 4: Each input unit receives the input signal  $X_i$  and transmits this signal to all units in the layer above i.e. hidden units  
 Step 5: Each hidden unit ( $Z_j, j=1, \dots, p$ ) sums its weighted input signals  $Z_{inj} = V_{oj} + \sum X_i V_{ij}$  applying activation function  $Z_j = f(Z_{inj})$  and sends this signal to all units in the layer above, i.e. output units.  
 Step 6: Each output unit ( $Y_k, k=1, \dots, m$ ) sums its weighted input signals  $Y_{ink} = W_{ok} + \sum Z_j W_{jk}$  and applies its activation function to calculate the output signals.  $Y_k = f(Y_{ink})$

##### Back Propagation Of Errors

- Step 7: Each output unit ( $Y_k, k=1, \dots, m$ ) receives a target pattern corresponding to an input pattern, error information term is calculated as  $\delta_k = (t_k - Y_k) f'(Y_{ink})$   
 Step 8: Each hidden unit ( $Z_j, j=1, \dots, n$ ) sums its delta inputs from units in the layer above

$$\delta_{inj} = \sum \delta_j W_{jk}$$

The error information term is calculated as  $\delta_j = \delta_{inj} f'(Z_{inj})$

### Updation of Weight and Biases

Each output unit ( $Y_k$ ,  $k=1, \dots, m$ ) updates its bias and weights ( $j=0, \dots, p$ ) The weight correction term is given by  $\Delta W_{jk} = \alpha \delta_k Z_j$  and the bias correction term is given by  $\Delta W_{0k} = \alpha \delta_k$

*Step 9: Test the stopping condition.*

The stopping condition may be the minimization of the errors, number of epochs

## 4. RESULTS AND DISCUSSIONS

All experiments reported in this paper are conducted on handwritten digits of MNIST database. There are 60,000 and 10,000 training and test samples in the MNIST database for handwritten English numerals. Five hundred samples (50 from each class) are randomly taken to classify. From these samples 70% are used for training and 15% for testing and remaining 15% are validation purposes. The present recognition scheme misclassifies an average of 4.4% of the training and test samples of handwritten English numerals. A Sample from MNIST database is given in Figure 4(a).

The proposed scheme provided an average classification of 95.6% for both 20 and 30 Hidden nodes. It is clear that the classification rate for each character varies with different hidden nodes, but the average classification is same in all the cases. For any case the classification rate for each character is above 95% which varies on different runs. The detailed recognition results are provided in Table I and II. To the best of our knowledge, the performance of above algorithm is efficient than previous algorithms.

To analyze the network response, a display of the confusion matrix which shows various types of errors that occurred for the final trained network is shown in figure 6. The diagonal cells in each table show the number of cases that were correctly classified, and the other cells show the misclassified cases. The blue cell in the bottom right corner shows the total percent of correctly classified cases (in green) and the total percent of misclassified cases (in red).

A plot of the training errors, validation errors, and test errors are shown in the figure 8. It is observed that the best validation performance occurred at iteration 89.

The Receiver Operating Characteristic (ROC) curve for each of the four categories of the simple test problem is shown in figure 7. Each axis in this curve represents the four different categories. It is observed that in all cases the true positive rate is high.

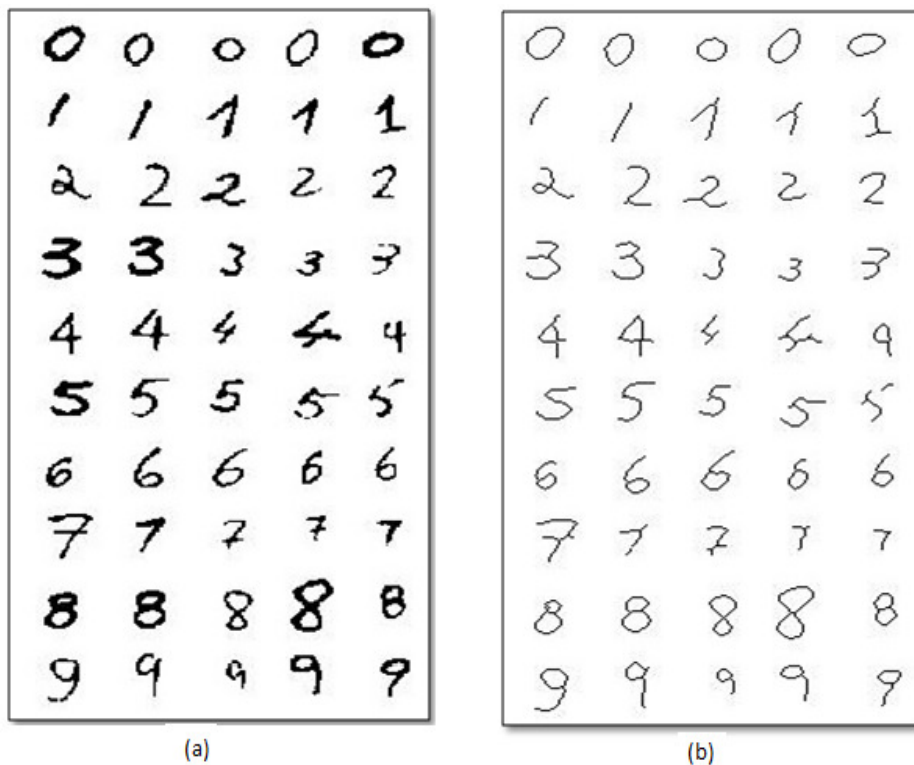


Figure 4: A Sample Isolated Digit Images of MNIST.  
 (a) Before preprocessing (b) After preprocessing

**Table I:** Classification using 20 Hidden nodes

<i>Digit</i>	<i>Classification %</i>	<i>Mis-classification %</i>
0	96	4
1	98	2
2	100	0
3	98	2
4	92	8
5	92	8
6	100	0
7	90	10
8	98	2
9	92	8
<b>Total</b>	<b>95.6</b>	<b>4.4</b>

**Table II:** Classification using 30 Hidden nodes

<i>Digit</i>	<i>Classification %</i>	<i>Mis-classification %</i>
0	96	4
1	94	6
2	98	2
3	97	3
4	97	3
5	94	6
6	98	2
7	94	6
8	96	4
9	92	8
<b>Total</b>	<b>95.6</b>	<b>4.4</b>

**Confusion Matrix**

Output Class	1	48 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	2	1 0.2%	49 9.8%	0 0.0%	0 0.0%	2 0.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	94.2% 5.8%	
	3	0 0.0%	0 0.0%	50 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	98.0% 2.0%	
	4	0 0.0%	0 0.0%	0 0.0%	49 9.8%	0 0.0%	3 0.6%	0 0.0%	1 0.2%	0 0.0%	2 0.4%	89.1% 10.9%
	5	1 0.2%	0 0.0%	0 0.0%	0 0.0%	46 9.2%	0 0.0%	0 0.0%	2 0.4%	0 0.0%	0 0.0%	93.9% 6.1%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	46 9.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.9% 2.1%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	1 0.2%	50 10.0%	0 0.0%	0 0.0%	0 0.0%	96.2% 3.8%
	8	0 0.0%	1 0.2%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	45 9.0%	0 0.0%	0 0.0%	95.7% 4.3%
	9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	49 9.8%	1 0.2%	98.0% 2.0%
	10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.4%	1 0.2%	46 9.2%	93.9% 6.1%
			96.0% 4.0%	98.0% 2.0%	100% 0.0%	98.0% 2.0%	92.0% 8.0%	92.0% 8.0%	100% 0.0%	90.0% 10.0%	98.0% 2.0%	92.0% 8.0%
		1	2	3	4	5	6	7	8	9	10	

Figure 6: Confusion Matrix

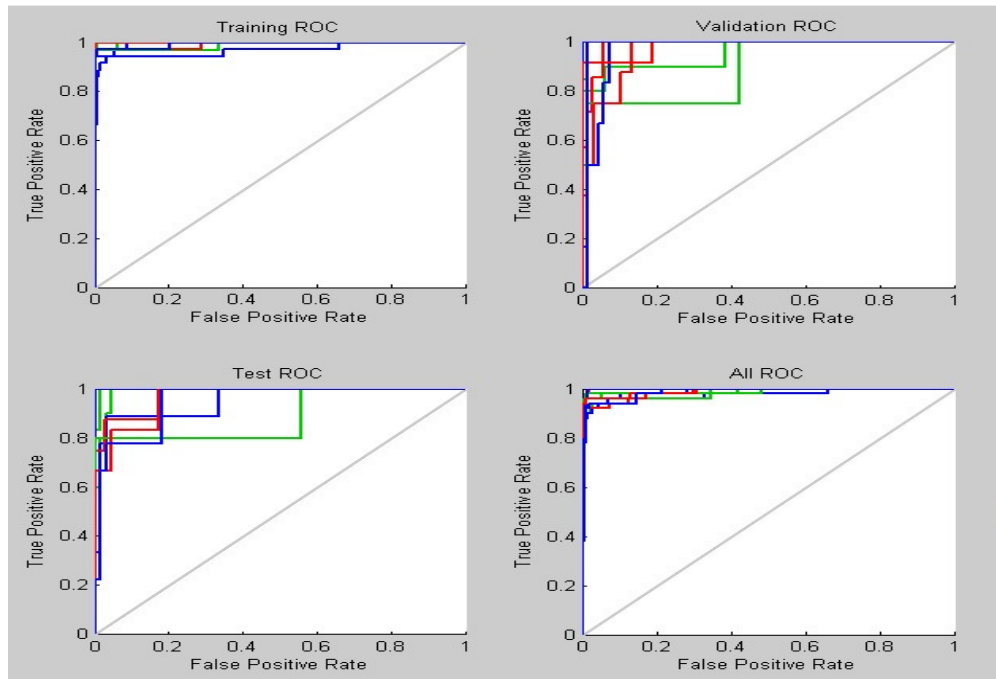


Figure 7: Receiver Operating Characteristic (ROC) Curve



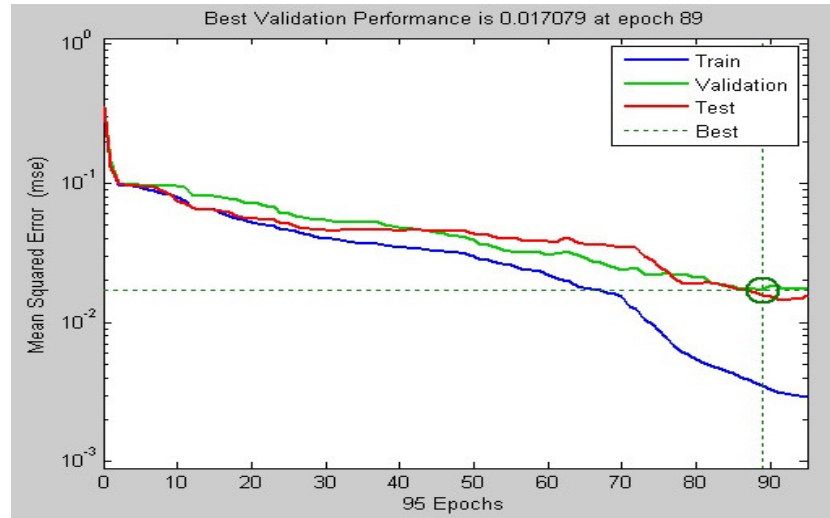


Figure 8: Validation Performance

## 5. CONCLUSION

In this paper, we have proposed a feature extraction technique to improve the recognition results of two similar shaped digits, and an ANN based system for classifying unconstrained offline handwritten digits. The key features of the classification system are the novel approach to preprocessing, feature extraction and classification.

A multilayered method for high accuracy recognition of these handwritten numerals has been introduced. The overall efficiency of the algorithm is 95.6% which is better than the previous approaches. This approach can be extended to character and Text recognition. The results for all three data sets (training, validation, and testing) show very good recognition.

## REFERENCES

- [1] T. Steinherz, E. Rivlin, and N. Intrator, "Offline Cursive Script Word Recognition—A Survey," *Int'l J. Document Analysis and Recognition*, vol. 2, no. 2, pp. 90-110, 1999.
- [2] R. Plamondon and S.N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, Jan. 2000.
- [3] N. Arica and F. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Rev.*, vol. 31, no. 2, pp. 216-233, May 2001.
- [4] AVinciarelli, "A Survey on Off-Line Cursive Word Recognition," *Pattern Recognition*, vol. 35, no. 7, pp. 1433-1446, 2002.
- [5] H. Bunke, "Recognition of Cursive Roman Handwriting—Past Present, and Future," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 448-459, Aug. 2003.
- [6] AKoerich, R. Sabourin, and C. Suen, "Large Vocabulary Off-Line Handwriting Recognition: A Survey," *Pattern Analysis and Applications*, vol. 6, no. 2, pp. 97-121, 2003.
- [7] H. Fujisawa, "Forty Years of Research in Character and Document Recognition—An Industrial Perspective," *Pattern Recognition*, vol. 41, no. 8, pp. 2435-2446, 2008.
- [8] U.-V. Marti and H. Bunke, "Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition Systems," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, pp. 65-90, 2001.
- [9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

- [10] J. J. Hull, "A Database for Handwritten Text Recognition Research," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, pp. 550-554, 1994.
- [11] T. Saito, H. Yamada, and K. Yamamoto, "On the Database ELT9 of Handprinted Characters in JIS Chinese Characters and Its Analysis (in Japanese)," Trans. IECEJ, vol. J.68-D, no. 4, pp. 757- 764, 1985.
- [12] Y. Al-Ohali, M. Cheriet, and C. Suen, "Databases for Recognition of Handwritten Arabic Cheques," Pattern Recognition, vol. 36, pp. 111-121, 2003.
- [13] T. Noumi, T. Matsui, I. Yamashita, T. Wakahara, and T. Tsutsumida, "Tegaki Suji Database 'IPTP CD-ROM1' no ichi bunseki (in Japanese)," Autumn Meeting of IEICE, vol. D-309, Sept. 1994.
- [14] Bharath, Ramachandran. Neural Network Computing. McGraw-Hill, Inc., New York, 1994. pp. 4-43.
- [15] R. Gonzalez and R. Woods Digital Image Processing, Addison-Wesley Publishing Company, 1992, pp 518 – 548.
- [16] Anil.K.Jain and Torfinn Taxt, "Feature extraction methods for character recognition-A Survey," Pattern Recognition, vol. 29, no. 4, pp. 641-662, 1996.
- [17] H. Takahashi, "A neural net OCR using geometrical and zonal pattern features" in proceedings of the First International Conference on Document Analysis and Recognition, (Saint-Malo, France), pp. 821-828, 1991.
- [18] Le Cun, Y.L., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., Back - Propagation Applied to Handwritten Zip Code Recognition, NeurComp, 1989, pp. 541-551.

## BIOGRAPHY

**N. VenkateswaraRao**, received his M.Sc degree in Computer Science Department from AcharyaNagarjuna University, India. He did his M.Tech in Computer Science & Technology from Andhra University, India. He is currently working as Associate Professor, in the Department of Computer Science & Engineering at RVR & JC College of Engineering, Guntur, India. He has 10 years of teaching experience. He is pursuing his Ph.D from AcharyaNagarjuna University, India, in Computer Science & Engineering under the guidance of Dr. B. RaveendraBabu. His research areas of interest include Artificial Neural Networks, Image Processing, and Pattern Recognition. He is life member of ISTE.



**Dr. A. Srikrishna** received the AMIE degree in Electronics & Communication Engineering from Institution of Engineers; Kolkatta in 1990, M.S degree in Software Systems from Birla Institute of Technology and Science, Pilani in 1994, M.Tech degree in Computer Science from Jawaharlal Nehru Technological University (JNTU) in 2003. She received her Ph.D from Jawaharlal Nehru Technological University Kakinada (JNTUK) in 2010. She has worked as lecturer from (1991-95). She is working in RVR &JC College of Engineering, Guntur from 13 years as Assistant Professor and Associate Professor. Her research interest includes Image Processing and Pattern Recognition. She is Associate member of IE (I) and member of CSI.



**Dr. B. RaveendraBabu**, obtained his Masters in Computer Science and Engineering from Anna University, Chennai. He received his Ph.D. in Applied Mathematics at S. V. University, Tirupati. He is currently leading a Team as Director (Operations), M/s. Delta Technologies (P) Ltd., Madhapur, Hyderabad. He has 26 years of teaching experience. He has more than 25 international & national publications to his credit. His research areas of interest include VLDB, Image Processing, Pattern analysis and Wavelets.



**G. Rama Mohan Babu**, received his B.Tech degree in ECE Engineering from Sri Venkateswara University (SVU), Tirupathi, India. He did his M.Tech in Computer Science & Engineering from Jawaharlal Nehru Technological University (JNTU), India. He is currently working as Associate Professor, in the Department of Information Technology at RVR & JC College of Engineering, Guntur, India. He has 13 years of teaching experience. He is pursuing his Ph.D from AcharyaNagarjuna University, India, in Computer Science & Engineering under the guidance of Dr. B. RaveendraBabu. His research areas of interest include Image Processing, Pattern Recognition, and Artificial Neural Networks. He is life member in professional bodies like ISTE and CSI.

