

NEURAL NETWORK-BASED ENGLISH ALPHANUMERIC CHARACTER RECOGNITION

Md Fazlul Kader¹ and Kaushik Deb²

¹Dept. of Applied Physics, Electronics & Communication Engineering, University of
Chittagong, Bangladesh.

f.kader@cu.ac.bd

²Dept. of Computer Science and Engineering, CUET, Bangladesh

debkaushik99@cuet.ac.bd

ABSTRACT

Propose a neural-network based size and color invariant character recognition system using feed-forward neural network. Our feed-forward network has two layers. One is input layer and another is output layer. The whole recognition process is divided into four basic steps such as pre-processing, normalization, network establishment and recognition. Pre-processing involves digitization, noise removal and boundary detection. After boundary detection, the input character matrix is normalized into 12×8 matrix for size invariant recognition and fed into the proposed network which consists of 96 input and 36 output neurons. Then we trained our network by proposed training algorithm in a supervised manner and established the network by adjusting weights. Finally, we have tested our network by more than 20 samples per character on average and give 99.99% accuracy only for numeric digits (0~9), 98% accuracy only for letters (A~Z) and more than 94% accuracy for alphanumeric characters by considering inter-class similarity measurement.

KEYWORDS

English Alphanumeric Character, Feed-forward neural network, Supervised Learning, weight-matrix, Character Recognition.

1. INTRODUCTION

Pattern recognition is the assignment of a physical object or event to one of several pre-specified categories [1]. It is an active field of research which has enormous scientific and practical interest. As [2] notes, it includes applications in “feature extraction, radar signal classification and analysis, speech recognition and understanding, fingerprint identification, character (letter or number) recognition, and handwriting analysis (‘notepad’ computers)”. Other applications include point of sale systems, bank checks, tablet computers, personal digital assistants (PDAs), handwritten characters in printed forms, face recognition, cloud formations and satellite imagery.

Character is the basic building block of any language that is used to build different structure of a language. Characters are the alphabets and the structures are the words, strings and sentences etc. [3].As in [4] character recognition techniques as a subset of pattern recognition give a specific symbolic identity to an offline printed or written image of a character. Character recognition is better known as optical character recognition because it deals with the recognition of optically processed characters rather than magnetically processed ones. The main objective of character recognition is to interpret input as a sequence of characters from an already existing set of characters. The advantages of the character recognition process are that it can save both time and effort when developing a digital replica of the document. It provides a fast and reliable alternative to typing manually.

An Artificial Neural Network (ANN) introduced by McCulloch and Pitts in 1943 is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of the ANN 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 [5]. ANNs are trainable algorithms that can learn to solve complex problems from training data that consists of a set of pairs of inputs and desired outputs. They can be trained to perform a specific task such as prediction, and classification. ANNs have been applied successfully in many fields such as pattern recognition, speech recognition, image processing and adaptive control.

A lot of scientific efforts have been dedicated to pattern recognition problems and much attention has been paid to develop recognition system that must be able to recognize an object regardless of its position, orientation and size [6]. Recently neural networks have been applied to character recognition as well as speech recognition with performance, in many cases, better than the conventional method [7]. Several neural network-based invariant character recognition system have been proposed. In [8], a pattern recognition system using layered neural networks called ADALINE was proposed. In [9], a character recognition system based on back-propagation algorithm has been proposed. In [10] a neural network based handwritten character recognition system without feature extraction was proposed.

In this paper we propose an artificial neural network based color and size invariant character recognition system which is able to recognize English characters (A~Z) and numbers (0~9) successfully. Our feed-forward network has two layers: one is input layer and another is output layer. No hidden layer is used. We have used supervise manner to train our neural network. Although we have trained and tested our system only with Times New Roman Font but the system is able to recognize character of any other font if properly trained with that font.

The rest of this paper organized is as follows. In section 2, the basic methodology of our character recognition process is described. Section 3 gives the results & discussions and finally we conclude this paper in section 4.

2. METHODOLOGY

The whole recognition process consists of four basic steps: preprocessing, normalized character matrix creation, network establishment and recognition. Preprocessing consists of digitization, noise removal and boundary detection of the digitized character matrix. In figure 1 we show the flowchart of our proposed character recognition scheme.

2.1. Input Character Image

Our system is able to recognize any colored printed character image with white background and font size is between 18 and 96. Figure 2 shows the sample character (e.g., 2 and Z) images of different sizes and colors.

2.2. Digitization and Matrix Creation from Character Image

In order to able to recognize characters by computer the character image is first digitized into a matrix i.e. transformed into a binary form for the ease of handling by the computer as shown in figure 3. Color image is first converted to the gray scale image as follows [11]:

$$Y = (\text{int}) (0.33 * R + 0.56 * G + 0.11 * B) \quad (1)$$

Where, R is the red component of a color pixel, G is the green component of a color pixel, B is the blue component of a color pixel and Y is the value of a pixel in a gray scale level. The image is then converted to binary. Just replace all the gray levels of the image into binary value 0 (for gray level value 129 to 255) treated as absence of writing or 1 (for gray level value 0 to 128) treated as presence of writing.

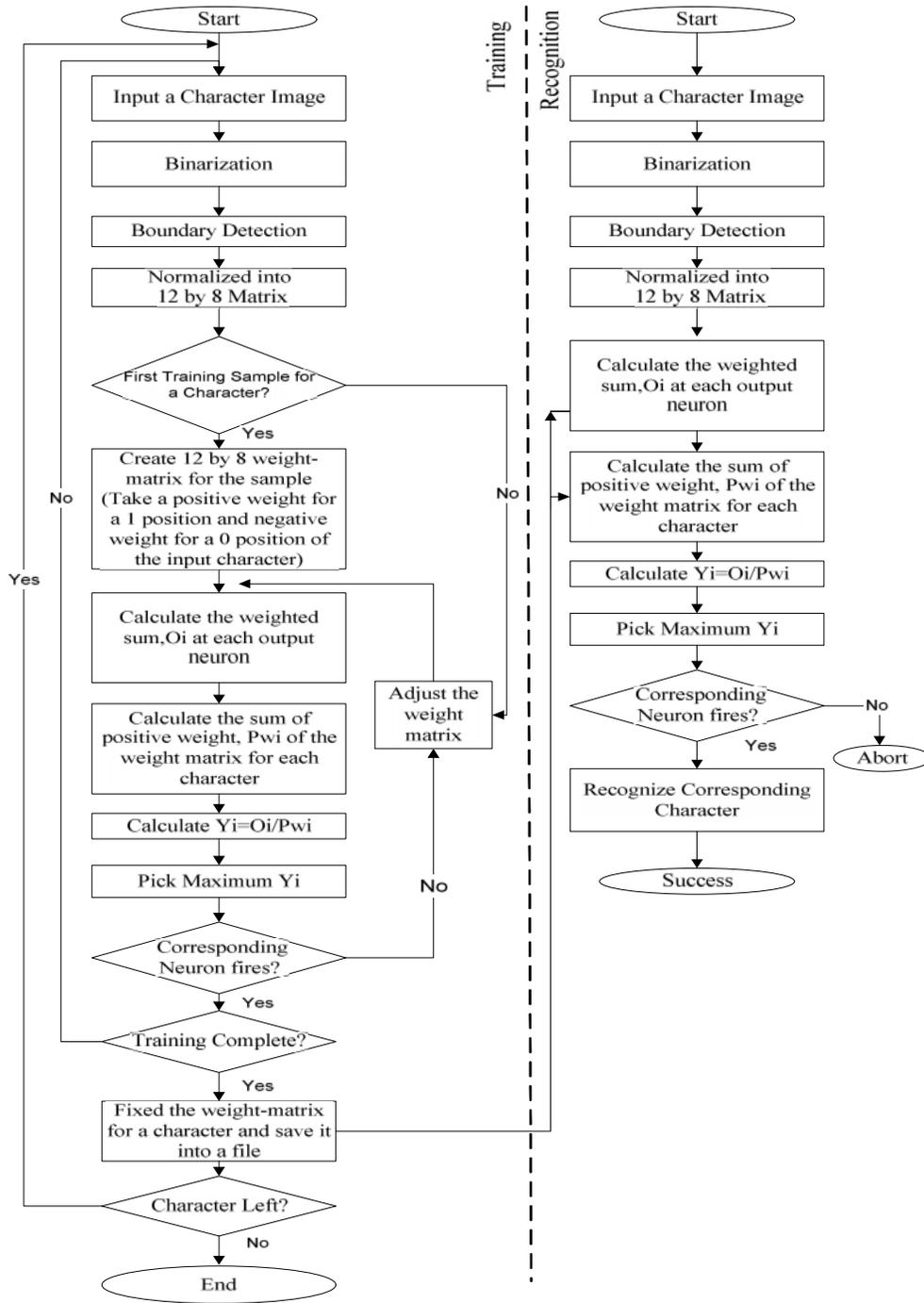


Figure 1. Proposed Character Recognition Scheme



Figure 2. Sample character images of different sizes and colors.

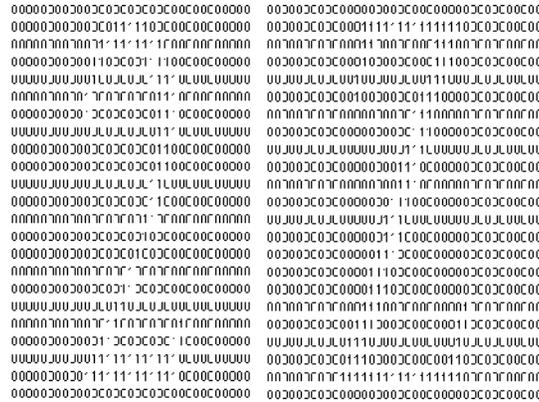


Figure 3. Digitized binary matrix of the sample character 2 and Z respectively

2.3. Boundary Detection

After creating the digitized binary matrix from the input character image, the detection of boundary is very much important to recognize character correctly. The boundary detection procedure is therefore given by [12]

- i. For top boundary detection, scan the character matrix starts at the top-left corner and remove all rows from top having only 0's. To detect top boundary there must be at least two consecutive 1's in two consecutive rows. Then the first row of the two consecutive rows from top will be selected as top boundary.
- ii. For bottom boundary detection, scan the character matrix starts at the bottom-left corner and remove all rows from bottom having only 0's. To detect bottom boundary there must be at least two consecutive 1's in two consecutive rows. Then the first row of the two consecutive rows from bottom will be selected as bottom boundary.
- iii. For left boundary detection, scan the character matrix starts at the top-left corner and remove all columns from left having only 0's. To detect left boundary there must be at least two consecutive 1's in two consecutive columns. Then the first column of the two consecutive columns from left will be selected as left boundary.
- iv. For right boundary detection, scan the character matrix starts at the top-right corner and remove all columns from right having only 0's. To detect right boundary there must be at least two consecutive 1's in two consecutive columns. Then the first column of the two consecutive columns from right will be selected as right boundary.

In figure 4, we show the binary matrix of the sample character 2 and Z shown in figure 3 after detecting boundary.

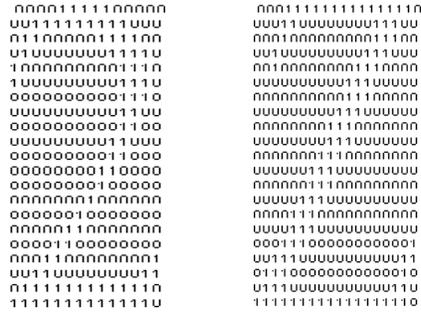


Figure 4. Boundary detection of the digitized sample character 2 and Z respectively shown in figure 3.

2.4. Normalization

Normalization is the process of equating the size of all extracted character bitmaps (binary array). For size invariant character recognition, we have converted the boundary detected input character matrix into 12x8 normalized matrix. The normalization procedure is as follows:

- i. Take top row and left column assuming they contain salient features.
- ii. Then take alternate row and column until desired 12x8 character matrix is found. For example, the size of input character (after boundary detection) is 15x11. Then during normalization,
 - a. take first row
 - b. delete row 2,4,6
 - c. Take all the remaining row
 - d. Converted the matrix to 12x11
- iii. Similar process for column as in ii (a-d). Finally the matrix is converted to 12x8.

In figure 5, we show the two different representation of normalized 12x8 matrix of the each sample character 2 and Z. The position of 1's in two normalized character matrix may be different because of different sizes of the same character are normalized into 12x8 matrix.

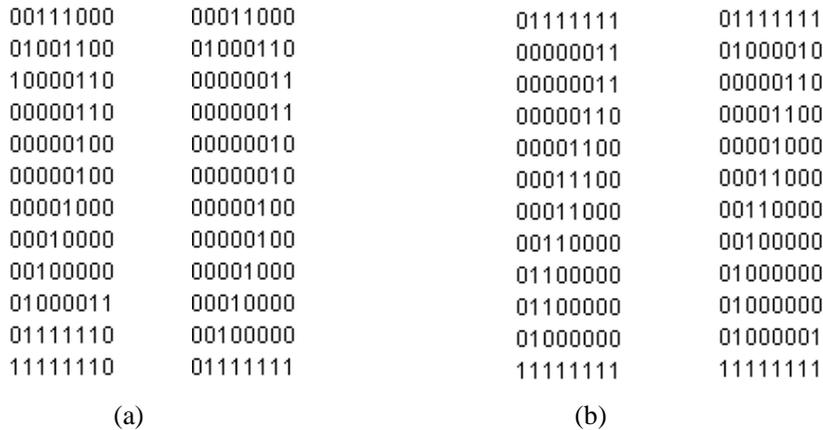


Figure 5. Different representation of normalized matrix (a) 12x8 matrix of numeric digit 2 (b) 12x8 matrix of character Z.

2.5. Network Architecture

Figure 6 shows the proposed feed-forward neural network architecture to recognize English alphanumeric characters. The network consists of 96 input and 36 neurons in its output layer to

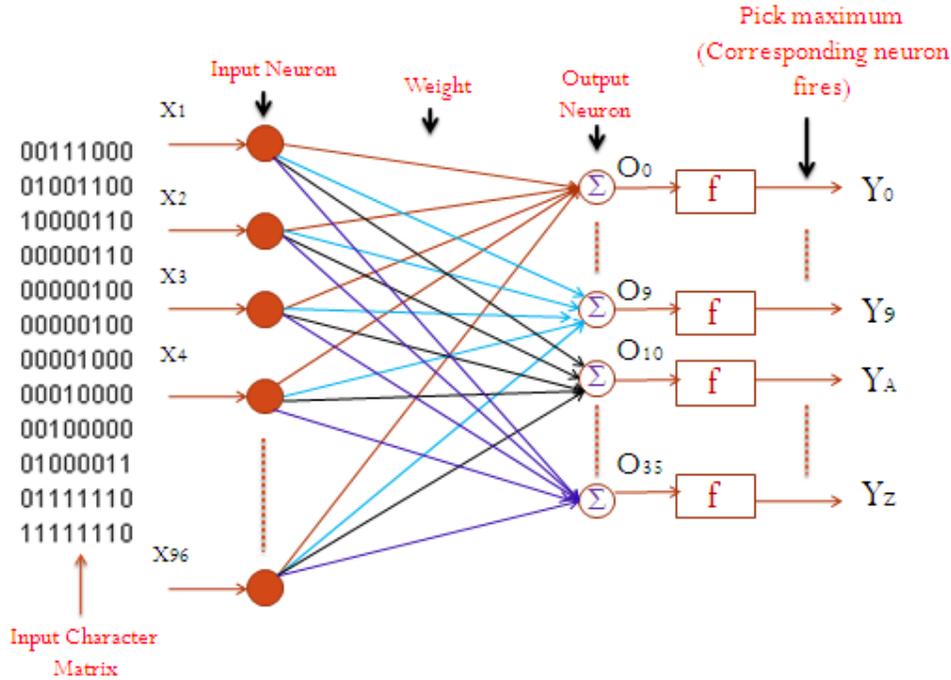


Figure 6. Proposed feed forward network architecture

identify the character. Each input neuron corresponds to one binary value of the normalized character matrix and each output neuron corresponds to one of the 36 characters (0~9 and A~Z). For example, X₁ for top-left corner value of the matrix and X₉₆ for the bottom-right and so on. No hidden layer is used. So we have to set only one type of weight, input – output layer weights. These weights represent the memory of the network, where final training weights can be used when running the network. Note that in the next subsequent section we use matrix instead of normalized input character matrix for simplicity.

From figure 5 it can be seen that binary value 1 is not always at the same location in corresponding matrix (e.g., for digit 2 in different matrix.). Some value is always same at the same location (e.g., 1 at location row0 and col4 in figure 5(a)). So take a positive weight for these locations i.e., take a positive weight (e.g., 3) for common 1 in same location and negative weight (e.g., -3) for common 0 locations as initial weight. The advantage of negative weight is that it will produce a lower weighted sum for other characters. For uncommon position, if more than equals two matrix has 1 at the same location take a positive 3 for that location. Otherwise, take a positive value such as 3 or 1 or 1.5 could be taken by considering interclass similarity measurement. Similar process should be taken for 0. In this way, fixed a weight matrix for each character such that corresponding neuron fires as the process shown in figure 6. The training procedure of our network is as follows:

- 1) Take a random training sample for any character (it is better to go ahead in sequence i.e., 0-9 and A-Z) and then generate 12×8 matrix.

- 2) Generate the corresponding 12×8 initial weight_matrix by taking 3 for a 1 and -3 for 0 of the input matrix.
- 3) Calculate the weighted sum O_i (net activation) as follows:
- 4)

$$O_i = \sum_{j=1}^{96} w_{ji} x_j$$

Where, $i=0, 1, 2, \dots, 35$.

- 5) Calculate the sum of positive weight, P_{wi} of the weight_matrix.
- 6) Calculate $Y_i = f(O_i) = O_i / P_{wi}$.
Where,
 O_i Net activation for each character i (e.g. 0, 1, 2, ..., 9, A, B, ..., Z)
 P_{wi} the sum of positive weight of the weight_matrix for the each character.
- 7) Pick the maximum Y_i .
- 8) Check if the corresponding neuron fires. If neuron fires then save the weight matrix into a file.
- 9) Check with other training samples of the same character and update the weight described as above.
- 10) Fixed the weight matrix and save into a file after the final training samples for that character.
- 11) Repeat steps 1 to 9 for any other character.
- 12) If the training is complete then the network is established.

Figure 7 shows the sample weight_matrix to recognize character Z. Then we have tested the

```

3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0
3.0 -3.0 -3.0 -3.0 -3.0 3.0 3.0 3.0
-3.0 -3.0 -3.0 -3.0 -3.0 3.0 3.0 -3.0
-3.0 -3.0 -3.0 -3.0 3.0 3.0 -3.0 -3.0
-3.0 -3.0 -3.0 3.0 3.0 -3.0 -3.0 -3.0
-1.0 -3.0 3.0 3.0 -3.0 -3.0 -3.0 -3.0
-1.0 -3.0 3.0 3.0 -3.0 -3.0 -3.0 -1.0
-1.0 -3.0 3.0 3.0 -3.0 -3.0 -3.0 -1.0
-1.0 -3.0 3.0 -3.0 -3.0 -3.0 -3.0 -3.0
-1.0 3.0 -3.0 -3.0 -3.0 -3.0 -3.0 -1.0
-1.0 3.0 -3.0 -3.0 -3.0 -3.0 -3.0 3.0
-3.0 3.0 3.0 3.0 3.0 3.0 3.0 3.0
    
```

Figure 7. Sample weight_matrix to recognize character Z

network with other samples that are not used in training. The testing procedure of our proposed system is as follows:

- 1) Take a random sample for any character and then generate 12×8 matrix.
- 2) Calculate the weighted sum O_i (net activation) as follows:

$$O_i = \sum_{j=1}^{96} w_{ji}x_j$$

Where, $i=0, 1, 2, \dots, 35$.

- 3) Calculate the sum of positive weight, P_{wi} of the weight_matrix.
- 4) Calculate $Y_i=f(O_i) = O_i/P_{wi}$.

Where,

O_i Net activation for each character i (e.g. 0, 1, 2, ..., 9, A, B, ..., Z)

P_{wi} the sum of positive weight of the weight_matrix for the each character.

- 5) Pick the maximum Y_i .
- 6) Check if the corresponding neuron fires.
- 7) If neuron fires, recognize corresponding character.
- 8) Repeat steps 1 to 7 for any other character.

We have trained the network by 5-10 samples per character in a supervised manner by considering inter-class similarity and more than 20 samples per character on average is used to test the system.

3. RESULTS AND DISCUSSIONS

Based on the proposed method described in the preceding sections, our system is able to recognize 10 English digits (0~9) and 26 capital letters (A~Z). We have used more than 1000 character samples to test our system. The proposed system is able to recognize only numeric digits (0~9) with 99.99%, only letters (A~Z) with 98% accuracy and alphanumeric characters (0~9, A~Z) with more than 94% accuracy on average by considering inter-class similarity measurement. Among the all characters, the recognition rate of inter-class similar characters is lower than the others. Table 1 shows the empirical result for only inter-class similar characters.

Table 1. Empirical Result: Interclass Similarity Comparison

Input Character	No. of Input sample	No. of Correct Identification	No. of Wrong Identification	Accuracy (%)
1	30	25	5	84
I	30	27	3	90
B	30	25	5	84
8	30	25	5	84
2	30	28	2	94
Z	30	30	0	100
0	30	25	5	84
D	30	29	1	97
0	30	25	5	84
O	30	24	6	80

Though we have trained and tested our system with the Times New Roman font of size between 18 and 96 but the system is able to recognize character of any other font if trained properly. The recognition rate for inter-class similar character is lower than others in comparison because the characters are very much similar with each other. In addition, our proposed method is size and color invariant but not rotation invariant. Though we have trained and tested our system with only printed characters but the system may also be able to recognize handwritten characters if all the samples of same character are of same angle-oriented but may be of different sizes and colors. The system is implemented with java language in windows environment.

4. CONCLUSION

In this paper, we have proposed an artificial neural network-based simple color and size invariant character recognition system to recognize English alphanumeric characters. Our proposed system gives excellent result for numeric digits and letters when they are trained and tested separately but produce satisfactory result when they are processed together. In addition our system is computationally inexpensive and easier to implement.

REFERENCES

- [1] R.O. Duda, P.T. Hart, D.G. Stork, Pattern Classification, 2nd ed., A Wiley-Interscience Publication, 2001.
- [2] R.J. SCHALKOFF, Artificial Neural Networks, The McGraw-Hill Companies Inc., New York, 1997.
- [3] Y. LeCun, B. Boer, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel "Handwritten zip code recognition with multilayer networks" International Conference on Pattern recognition, pp. 35-44, 1990.
- [4] T. SITAMAHALAKSHMI, A. VINAY BABU, M. JAGADEESH, "Character Recognition Using Dempster-Shafer Theory combining Different Distance Measurement Methods," International Journal of Engineering Science and Technology, Vol. 2(5), 2010, 1177-1184.
- [5] J.C. Principe, Neil R. Euliano, Curt W. Lefebvre, Neural and Adaptive Systems: Fundamentals through Simulations, ISBN 0-471-35167-9.
- [6] J. Kamruzzaman and S. M. Aziz, "A Neural Network Based Character Recognition System Using Double Backpropagation," Malaysian Journal of Computer Science, Vol. 11 No. 1, pp. 58-64, June 1998.
- [7] Guyan et. al., "Design of a Neural Network Character Recognizer for a Touch Terminal", Pattern Recognition, Vol. 24, No. 2, pp. 105-119, 1991.
- [8] B. Widrow, R. Winter and R. A. Bayter., "Layered Neural Nets for Pattern Recognition", IEEE Trans. Acoustics, Speech and Signal Process. Vol. 36, No. 7, pp. 1109-1118, 1998.
- [9] F. Li, and Gao, "Character Recognition System Based on Back-Propagation Neural Network," International Conference on Machine Vision and Human-Machine Interface (MVHI), pp. 393-396, 2010.
- [10] J. Pradeep, E. Srinivasan, and S. Himavathi, "Neural network based handwritten character recognition system without feature extraction" International Conference on Computer, Communication and Electrical Technology (ICCCET), pp. 40-44, 2011.
- [11] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd Ed..
- [12] M.F. Kader, M.K. Hossen, Asaduzzaman and A.S.M Kayes, "An Offline Handwritten Signature Verification System as a Knowledge Base," Computer Science and Engineering Research Journal, CSE, CUET, Vol. 04, 2006.

Authors

Md. Fazlul Kader received the B. Sc. Engineering Degree in Computer Science and engineering (CSE) from Chittagong University of Engineering and Technology (CUET), Bangladesh, in 2005. From 2007 to onwards he is a faculty member of the Dept. of Applied Physics, Electronics and Communication Engineering, University of Chittagong, Bangladesh. He is currently working toward the M.Sc. Engineering degree as a part-time basis in the Department of CSE, CUET, Bangladesh. His major research interests include Pattern Recognition, Image processing and Cognitive radio Network etc.



Kaushik Deb received his B.Tech. and M.Tech. degrees from Department of Computer Science and Eng. of Tula State University, Tula, Russia, in 1999 and 2000, respectively. Since 2001, he has been serving as a faculty member in Department of Computer Science and Engg., Chittagong University of Engineering and Technology (CUET), Chittagong, Bangladesh. He received his Ph.D. from the Graduate School of Electrical Eng. and Information Systems, University of Ulsan, Ulsan, Korea in 2010. His research interests include computer vision, pattern recognition, intelligent transportation systems (ITSs) and human-computer interaction.