

Effective Character Recognition using ANN & Convolution Techniques.

Rohini Meher¹, Bidyutprabha Bhoi¹, Purnendu Mishra^{2*}, Pankaj Kumar²

¹Dept. of Physics, Vikash Degree college, Bargarh, Odisha, India.

²Dept. of ECE, Vikash Institute of Technology, Bargarh, Odisha, India.

*Corresponding Author's Email: erpurnendumishra@gmail.com

ARTICLE INFO

Article history:

Received : 03 Dec. 2016

Accepted : 18 Dec. 2016

Available online : 28 Dec. 2016

Keywords:

Character Recognition,
Artificial Neural Networks,
Self-organizing map,
Kohonen learning algorithm,
Two dimensional Convolution,
Segmentation.

ABSTRACT

Character recognition is performed off-line after the writing or printing has been completed, as opposed to on-line recognition where the computer recognizes the characters as they are drawn. Both hand printed and printed characters may be recognized, but the performance is directly dependent upon the quality of the input documents. Artificial Neural Networks (ANNs) serve for the emulation of human thinking in computation to a meager, yet appreciable extent. Of the several fields wherein they have been applied, humanoid computing in general and pattern recognition in particular of increasing activity. ANNs have enjoyed considerable success in this area due to their humanoid qualities such as adapting to changes and learning from prior experience. Another approach of character recognition by the process of convolution individual character is convolved with dataset and character in the dataset with which maximum matching comes is assign to that character. In this paper both the methods are discussed with the simulation result.

© 2016 International Journal of Advanced Research in Science and Technology (IJARST).

All rights reserved.

PAPER-QR CODE



Citation: Purnendu Mishra. et. al., Effective Character Recognition using ANN & Convolution Techniques, Int. J. Adv. Res. Sci. Technol. Volume 5, Issue 3, 2016, pp.635-639.

Method A: By Using Self Organizing Map

Introduction:

In a self-organizing map, the neurons are placed at the nodes of a lattice that is usually one or two-dimensional. Higher dimensional maps are also possible but not as common. The neurons become selectively tuned to various input patterns (stimuli) or classes of input patterns in the course of learning process. The locations of neurons so tuned (i.e., the winning neurons) become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice. A self-organizing map is therefore characterized by the formation of a topographic map of the input patterns in which the spatial locations (i.e., coordinates) of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns, hence the name “self-organizing map”. [1,2]. The principal goal of Self organizing Map (SOM) is to transform an

incoming signal pattern of arbitrary dimension into one or two dimensional discrete map and to perform this transformation adaptively in a topological ordered fashion. Figure 1 shows the schematic diagram of a two dimensional lattice neuron commonly used as the discrete map. Once the network has been properly initialized, there are three essential processes involved in the information of the self-organizing map i.e. Competition, Cooperative Process, Synaptic Adaptive Process [3,4].

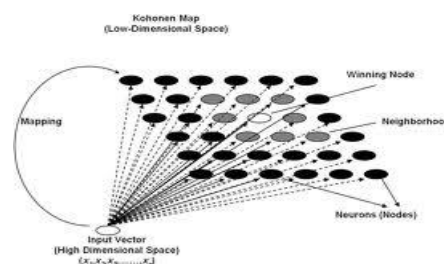


Fig. 1. Self-Organized feature maps

Image Digitization:

When a document is put to a visual reorganization, it is expected to be consisting of printed (or hand written) characters pertaining to one or more scripts or fonts. This document however, many contain information besides optical characters alone. In addition, characters which need to be singly analyzed may exist as word clusters or may be located at various points in the document. Such an image is usually processed for noise reduction and separation of individual characters from the document. . The process of digitization is important for the neural network used in the system. In this process, the input image is sampled in to a binary window which forms the input to the reorganization system. In the above figure, the alphabet T has been digitized into let, $A \times B = C$ digital cells, each having a single color, either black or white. It becomes important for us to encode this information in a form meaning full to a computer. For this, we assign a value +1 to each black pixel and 0 to each white pixel and create the binary image matrix I which is shown in the figure. So much conversion is enough for neural networking which is described next. Digitization of an image in to a binary matrix of specified dimensions makes the input image invariant of its actual dimensions. Hence an image of whatever size gets transformed into a binary matrix of fixed pre-determined dimensions. This establishes uniformity in the dimensions of the input and stored patterns as they move through the reorganization system.

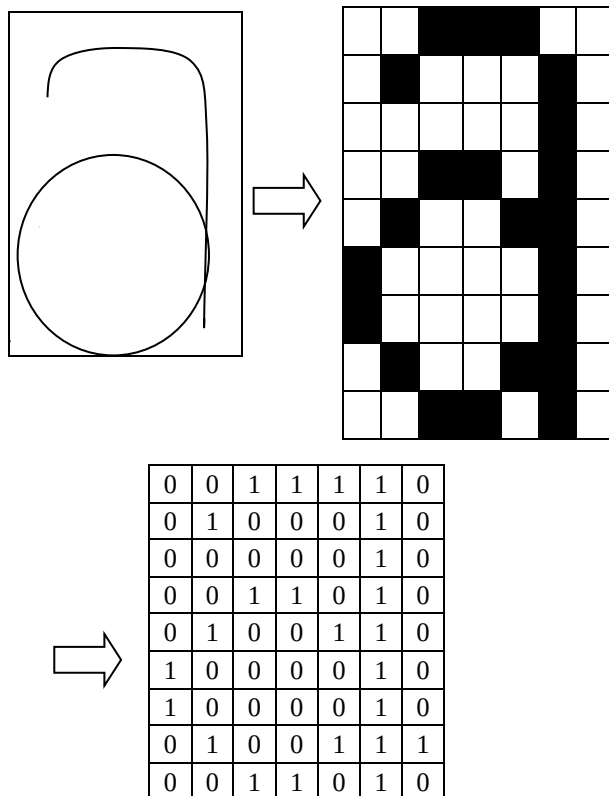


Fig: 2. Image digitization

The essence of Kohonen's SOM algorithm is that it substitutes a simple geometric computation for the more detailed properties of the Hebb-like rule and lateral interactions [6, 7].

Kohonen Learning Algorithm:

There are three basic steps involved in the application of the algorithm after initialization: sampling, similarity matching, and updating. These three steps are repeated until formation of the feature map has completed. The algorithm is summarized as follows [10, 12, 13]:

1. Initialization:

Choose the random values for initial weight vectors $w_j(0)$. The only restriction here is that the $W_j(0)$ be different for $j=1,2,3,...,l$, where l is the number of neurons in the lattice. It may be described to keep the magnitude of the weights small.

Another way of initializing the algorithm is to select the weight vector

$\{w_j(0)\}_{j=1}^l$ from the available sets of input vectors $\{x_j\}_{j=1}^N$ in a random manner.

2. Sampling:

Draw a sample x from the input space with a certain probability; the vector x represents the activation pattern that is applied to the lattice. The dimensions of the vector x is equal to m .

3. Similarity Matching:

Find the best matching (winning) neuron $i(x)$ at a time step n by using minimum distance Euclidian criterion:

$$i(x) = \arg \min_j \|x(n) - w_j\|, j=1,2,...,l$$

4. Updating:

Adjust the synaptic weight vectors of all neurons by using the update formula

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n) (x(n) - w_j(n))$$

Where $\eta(n)$ is the Learning rate parameter, and $h_{j,i(x)}(n)$ is the Neighborhood function centered around the winning neuron $i(x)$; both $\eta(n)$ and $h_{j,i(x)}(n)$ varied dynamically during learning for best results.

5. Continuation:

Continue with step 2 until no noticeable changes in the feature map are observed.

Initially the binary codes for all the alphabets (both cases), decimal numbers and some special characters are submitted. The input pattern is to be collected by X (1×63). The weight matrix W (63×63) is to be initialized. The initial learning rate is defined and the algorithm is applied. The different steps required in coding are:-

- The binary codes for all the alphabets are to be submitted.
- Chose the input pattern.
- Printing the input pattern.
- Start of Learning Process.
- Wight updating of the winning neuron
- Rounding of the weight vectors to the nearest integer value after the training.
- Printing the weight matrix of wining neuron.

The step by step procedure is represented in the form a flow chart as shown in Fig.3

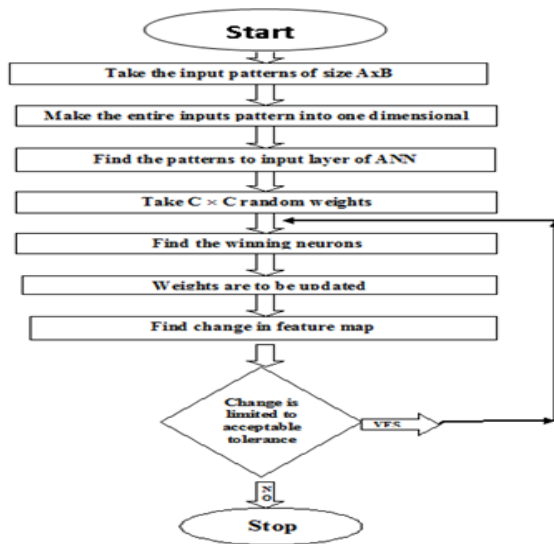


Fig: 3. flow chart of character recognition by SOM

Simulation Results:

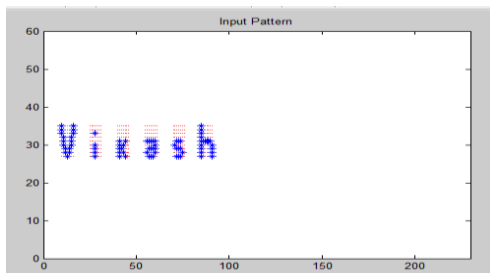


Fig: 4. Input Pattern

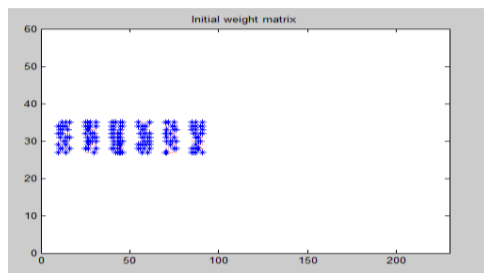


Fig: 5. Initial Weight Matrix

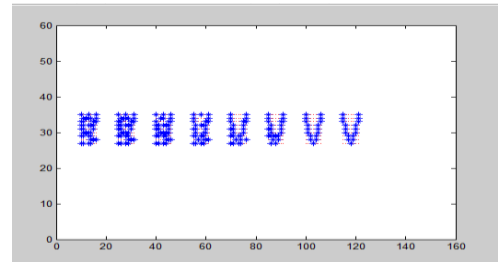


Fig: 6. Iteration of a Character

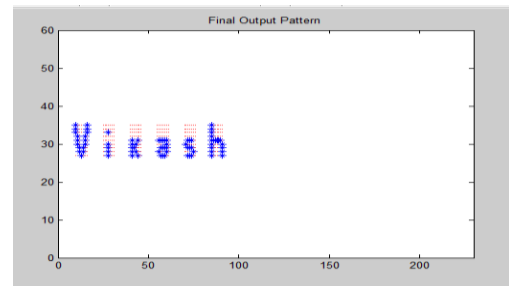


Fig: 7. Final Output Pattern

Method B: By Using Two dimensional Convolution Technique

Introduction:

Optical Character Recognition deals with the problem of recognizing optically processed characters. Optical recognition is performed off-line after the writing or printing has been completed, as opposed to on-line recognition where the computer recognizes the characters as they are drawn. Both hand printed and printed characters may be recognized, but the performance is directly dependent upon the quality of the input documents [16].

Data flow diagram:

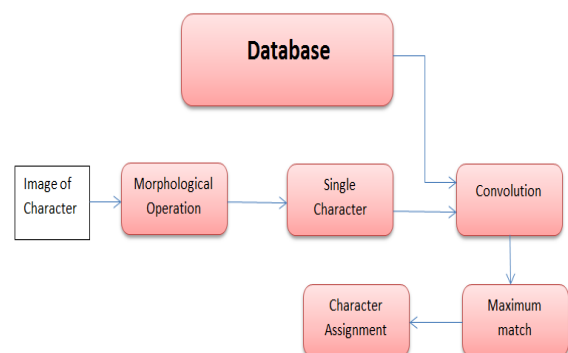


Fig: 8. Data flow diagram

Dataset:

It is the set of data of standard character image and number image used for matching with the given character to recognize. This all data is combined in single matrix. Here standard image character of image number is taken in .bmp format and is converting from .bmp image into two dimensional matrix.

Pre-processing:

At First image is converted into gray scale, then it is broken into different lines by scanning each row of image if sum of total while pixel is less than a given threshold value it sense for new line. In one line of image segmentation operation is used to label different character with different number and then each character is separated based its label number[16]. Finally we have an image of one character, and then it is resized to standard format of image in the database.

Convolution:

Given character image is scanned and broken into number of lines containing character. These line are again broken into different character by morphological operation. Then individual character is convolved with dataset and character in the dataset with which maximum matching comes is assign to that character [17,18]. Since character data is two dimensional we used here two dimensional convolution as

$$y[m, n] = x[m, n] * h[m, n] = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x[i, j] \cdot h[m-i, n-j]$$

Character Assignment:

Different characters are given different index the index for which match is maximum value. That index character is assigned to that optical character image.

Simulation Results:

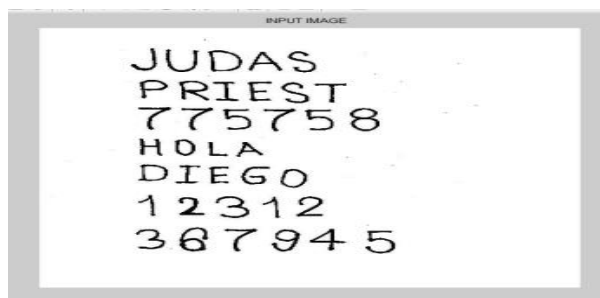


Fig. 9. Input Image

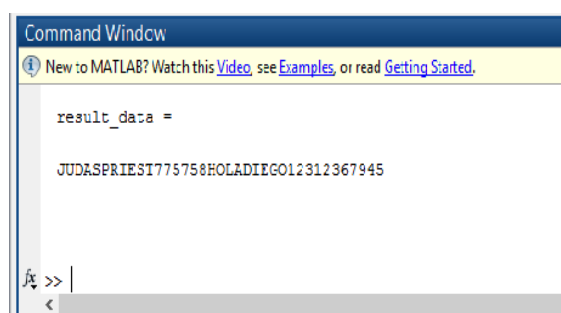


Fig. 10. Final output

Conclusion:

This paper tells about Despite of computational complexity involved, ANN offers several advantages in pattern recognition and classification in the scene of

emulating adaptive human intelligence to a small extent. Hence ANN for character recognition method is more effective as compared to the classical methods and the OCR system of recognizing the both hand printed & printed characters may be recognized, but the performance is directly dependent upon the quality of the input documents.

Acknowledgement:

The author would like to thank the Management members, Director and Principal for providing the required infrastructure and other supports to carry out the research work.

References:

1. Anil K. Jain, Jianchang Mao, K. M. Mohiuddin, Artificial Neural Networks: A Tutorial, Computer, March 1996.
2. Simon Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall
3. Laurence Fausett : Fundamentals of Neural networks : 3rd Indian Reprint.
4. Alexander J. Faaborg, using Neural Networks to create an adaptive character recognition system, March 2002, available at http://web.media.mit.edu/~faaborg/research/cornell/hci_neuralnetwork_finalpaper.pdf
5. A kohonen Neural Network Controlled All-optical router system. E.E.Frietman, M.T.Hill, G.D.Khoe <http://www.ph.tn.tudelft.nl/~ed/pdfs/IJCR.pdf>
6. Image compression by Self-Organized kohonen Map Christophe Amerijckx, Philippe Thissen. IEE Transition on Neural Networks 1998. <http://www.dice.ucl.ac.be/~verleyse/papers/ieeetnn98ca.pdf>
7. Kohonen, T. and Honkela, T. (2007). "Kohonen network". Scholarpedia. http://www.scholarpedia.org/article/Kohonen_network.
8. Ultsch, Alfred (2007). Emergence in Self-Organizing Feature Maps, In Proceedings Workshop on Self-Organizing Maps (WSOM '07). Bielefeld, Germany. ISBN 978-3-00-022473-7.
9. Haykin, Simon (1999). "9. Self-organizing maps". Neural networks - A comprehensive foundation (2nd edition ed.). Prentice-Hall. ISBN 0-13-908385-5.
10. "Intro to SOM by Teuvo Kohonen". SOM Toolbox. <http://www.cis.hut.fi/projects/somtoolbox/theory/somalgorithms.shtml>. Retrieved 2006-06-18.
11. Ultsch A (2003). U*-Matrix: a tool to visualize clusters in high dimensional data. University of Marburg, Department of Computer Science, Technical Report Nr. 36:1-12.
12. Yin H. Learning Nonlinear Principal Manifolds by Self-Organising Maps, In: Gorban A. N. et al. (Eds.), LNCSE 58, Springer, 2007 ISBN 978-3-540-73749-0
13. Liu, Y., R.H. Weisberg, and C.N.K. Mooers (2006), Performance evaluation of the Self-Organizing Map for feature extraction. Journal of Geophysical Research, 111, C05018, doi:10.1029/2005jc003117.
14. Kaski, S.. Data exploration using self-organizing maps, Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82, Espoo 1997, 57 pp..

15. Shalin A. Chopra, Amit A. Ghadge, Onkar A. Padwal, Karan S. Punjabi, Prof. Gandhali S. Gurjar, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 1, January 2014
16. Er. Neetu Bhatia ,Recognizing Cursive Numeral Handwritten Character Paper, International Journal of Advanced Research in Computer Science and Software Engineering, Volume-4, Issue-6, June-2014
17. https://en.wikipedia.org/wiki/Multidimensional_discrete_convolution
18. <https://graphics.stanford.edu/courses/cs178/applets/convolution.html>.