JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Special Issue-1 E-ISSN: 2347-2693

Pattern Recognition using Modified Compression Algorithm with Mexican Hat (MCAMH)

Rishav Upadhyay[1*], Prasenjit Biswas[2]
Dept. of Computer Science and Application
University of North Bengal, Dist: Darjeeling
West Bengal, Pin-734013, INDIA
[1]upadhyay.rishav@gmail.com, [2]prasenjit.lead@gmail.com

Partha Protim Poddar[3], Rakesh Kumar Mandal[4]
Dept. of Computer Science and Application
University of North Bengal, Dist: Darjeeling
West Bengal, Pin-734013, INDIA
[3]parthaprotimpoddar@gmail.com, [4]rakesh_it2002@yahoo.com

Abstract - **Making a machine understand and recognize a character is a challenge. This is also factored as many researchers have indulged in building worthy hardware and necessary software for character recognition. A well known tool to solve this problem is Artificial Neural Network (ANN). In this paper, Mexican Hat, a fixed-weight net, is used to improve the precision than the previous attempts which already have been introduced. Here 14 X 14 image matrices have been taken as input and then compressed into 7 X 7 matrices, reducing the elements which are of no or little significance. Mexican Hat is used to recognize the alphabet.**

Keywords: ANN, MCAMH, Compression, Mexican Hat

I. INTRODUCTION

Many proposals have been made on developing various Expert Systems using Artificial Neural Network (ANN) to recognize characters. During character recognition, we often encounter a situation in which we have more information about the possible correct response of the net than we are able to incorporate, specifically, when we apply a net which is trained to classify the input signal into one of the output categories. In these circumstances, a net based on competition is forced to make a decision as to which only one unit will respond. Thus, the accuracy of the output can certainly be increased [2].

Several attempts have already been made for recognizing the characters. Row-wise Segmentation Technique is an approach where the matrix is segmented row-wise and then the elements are presented to the net for recognition, [3].

In Perceptron learning method, common features among the characters are distinguished which is nothing but an ANN [1, 2, 4].

Here, a novel technique has been introduced using modified compressed algorithm and Mexican Hat to recognize characters.

The whole process comprises compression of the input matrix, linear array representation of the matrix, application of Mexican Hat algorithm and at last plotting the graph, based on that the character is recognized.

I. METHODOLOGY

Here each character is enclosed in a box comprising 14 X 14 pixel in jpeg format. The matrix is a binary matrix notifying all the pixels as 0s except the character. The character pattern is presented as sequence of 1s. As it is a large matrix containing a large number of non significant elements which impart no information about the actual

pattern, the matrix is compressed into a matrix of size 7 X 7. The matrix cannot be compressed any further as there is a risk of data loss. Mexican Hat is used to train the net. Now the trained net takes the test character as input. Based on the accuracy of the output, the efficiency is then determined.

A. Compression of 14 X 14 matrix into 7 X 7 matrix

To reduce the unnecessary elements the matrix is hereby compressed into a lower dimension, but any further compression cannot be done. Matrix M is taken as input and the output after compression is matrix M_c using the function Θ. The function Θ contains a modified compression methodology which will be illustrated in the Algorithm.
The matrix M is split into *n* uniform blocks. Each block: M_blck_i comprises *m* x *m* dimension [1].

- **Algorithm of Compression**

Step1: Input Matrix M.
Step2: Considering M_blck_i as the i^{th} block of M which is split M into n uniform blocks.
Step3: Consider iteration (i = 1 to n)
$M\_c_i = \Theta (M\_blck_i)$
End
{M_c_i is the i^{th} element of the compressed matrix M_c and
$\Theta (M\_blck_i) = 1$, if the number of 1s >= number of 0s in M_blck_i
$\Theta (M\_blck_i) = 0$, if the number of 1s < number of 0s in M_blck_i}
Step4: Stop.

Example: Given matrix M of size 4 X 4

$$M = \begin{matrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{matrix}$$

Here M can be split into 4 uniform blocks comprising 2 X 2 dimensions each.

$M\_blck_1 = \begin{matrix} 1 & 1 \\ 1 & 0 \end{matrix}$ $M\_blck_2 = \begin{matrix} 0 & 1 \\ 0 & 0 \end{matrix}$ $M\_blck_3 = \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix}$ $M\_blck_4 = \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix}$

Using function Θ:

$$M\_c = \begin{matrix} 1 & 0 \\ 1 & 0 \end{matrix}$$

In this procedure the 14 X 14 matrix is compressed into 7 X 7 matrix reducing non significant elements.

B. Training by Mexican Hat

The 7 X 7 matrix is segmented column-wise comprising 7 columns. The matrix is converted into a linear-array (L) consisting 49 elements where the elements of the matrix is put into the array column-wise. Hence the 25^{th} element is considered as the middle element (L_i). The elements on the either side of the X_i are symmetric and the left hand-side elements are considered as L_{i-1}, L_{i-2}, L_{i-3} .....L_{i-24} whereas the right hand-side elements are considered as L_{i+1}, L_{i+2}, L_{i+3} .....L_{i+24}.

Illustration of the segmentation:

$$M\_c = \begin{matrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix}$$

Figure 1: 7X7 Matrix Showing the "I" Character after Compression

The linear array formed from the Matrix:

L={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}

Here the L_i is the 25th element of L. Therefore the Mexican Hat orientation is as follows:

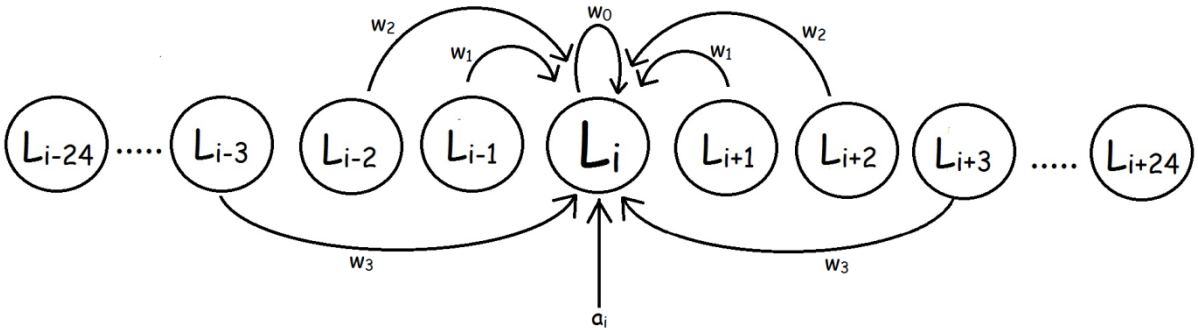

Figure 2: Mexican Hat interconnections for Li [2]

Based on the linear-array L, we get three positive interconnections on the either-side of L_i. And the remaining are 0s.
Based on the external signal L, the activation function for the net is:

$$f(m) = \begin{cases} 0 & if\ m < 0 \\ m & if\ 0 \le m \le 49 \\ 49 & if\ m > 49 \end{cases}$$

II. ALGORITHM MCAMH

Different parameters used in the Algorithm:

Ra_1 = Radius of region with positive reinforcement
Ra_2 = Radius of region of interconnection; L_i is connected to units L_{i+j} and L_{i-j} for j=1,...,Ra_2
m = vector of activations
m_old = vector of activations at previous time step.
k_max = number of iteration
L = external signal
w_j = weight on inter connections between L_i and units L_{i+j} and L_{i-j}
w_j is positive for $0<=j<=Ra_1$
w_j is negative for $Ra_1<j<=Ra_2$

As the algorithm corresponds to external signal being given only for first iteration (Step5.2) of the contrast enhancing iterations we have:

Step1: Take the matrix M which is of dimension
m x m as input.
Step2: Convert it into the compressed matrix M_c
which is of dimension n x n (m>n)
by applying Algorithm of Compression.
Step3: Convert the array from the matrix into linear
vector.
Step4: Set the whole array as external signal
(m=L).
Step5: Apply the Mexican Hat algorithm [2].
Step5.1: Initialize parameters k_max, Ra_1,
Ra2, w_j
k_max = 3 [setting the iteration as desired]
$Ra_1 = 3$
$Ra_2 = 24$
$w_j = Ca_1$ for j = 0,...., Ra_1 (Ca_1>0)
$w_j = Ca_2$ for j = Ra_1 + 1,....,Ra_2 (Ca_2<0)
Initialize m_old as 0.

Step5.2: Taking the external signal L
m = L
Save the above in m_old (for x = 1, ..., 49)
$m\_old_x = m_x$
Set the iteration counter k = 1

Step5.3: While k is less then k_max,
repeat Steps 5.4 – 5.7.

Step5.4: Compute the net input
(i=1,2,....,49)

$$m_i = Ca_1 \sum_{k=-Ra_1}^{Ra_1} m\_old_{i+j} + Ca_2 \sum_{k=-Ra_2}^{-Ra_1-1} m\_old_{i+j} + Ca_2 \sum_{k=Ra_1+1}^{Ra_2} m\_old_{i+j}$$

Computation algorithm:
Step5.4.1: initialize i=1 repeat
Step 5.4.2 to 5.4.12 until i<=49
Initialize a=0, b=0, c=0
Step5.4.2: initialize j=-Ra_1 repeat step 5.4.3 and 5.4.4 until j<=Ra_1
Step5.4.3: if ((i+j)>=1 and (i+j) <=49) then set a= a+Ca_1*m_old_{i+j}
Step5.4.4: update j=j+1
Step5.4.5: set j=-Ra_2 repeat step 5.4.6 and 5.4.7 until j<=-Ra_1-1
Step5.4.6: if ((i+j)>=1) then set
b=b+Ca_2*m_old_{i+j}
Step5.4.7: update j=j+1
Step5.4.8: set j=Ra_1+1 repeat step 5.4.9
and 5.4.10 until j<=Ra_2
Step5.4.9: if ((i+j)>=1 and (i+j) <=49) then set c=c+Ca_2*m_old_{i+j}

Step5.4.10: update j=j+1
Step5.4.11: set m_i=a+b+c
Step5.4.12: update i=i+1
Step5.5: Apply activation function:
m_i = min(m_max,max(0,m_i)) (i = 1,2,...,49)
Step5.6: Save m_i to m_old_i:
$m\_old_i = m_i$ (i = 1,2,....., 49)
Step5.7: Set k = k+1
Step5.8: If k > k_max then Stop;
otherwise continue.
Step6: Put the output m_i (i = 1, 2,....., 49) into graph for each iteration up to k<=k_max
Step7: Stop.

Example: Considering the input Matrix as M, and the compressed Matrix as M_c (from fig 1.1) which represents the character "I", the external signal is:
L={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
Initializing
k_max = 3
Ra_1 = 3
Ra_2 = 24
And determining Ca_1 = 0.6 and Ca_2 = -0.4 from the activation function f(m) and m=L.
Inputting the values in algorithm MCAMH we get:
m_1={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.2,1.2,2.2,3.2,4.2,3.2,2.2,1.2,0.2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} (for k = 1)
m_2={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 3.88, 7.079999, 9.279999, 10.280001, 9.279999, 7.079999, 3.88,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} (for k = 2)
m_3={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 10.216, 19.496, 26.57, 30.45, 26.57, 19.496, 10.216,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0} (for k = 3)

As k = k_max we are stopping and putting the values in the graph. The graph is like as follows:

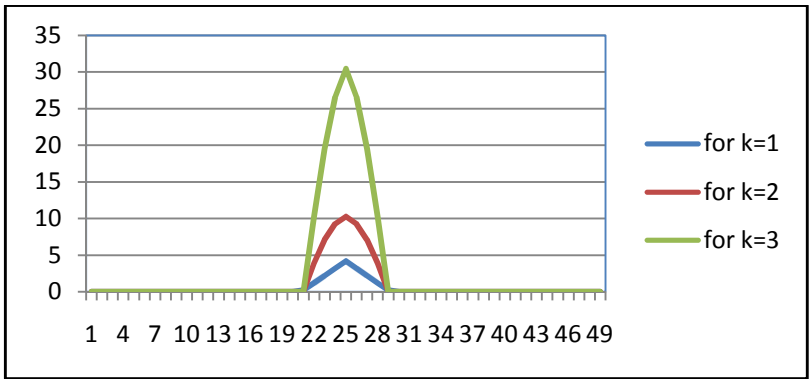

Figure 3: Result of the Mexican Hat

As the graph is forming Mexican Hat, the net has identified the character "I" completely.

III. DISCUSSION

The result shows that the MCAMH net is very efficient. As there is a liberty to determine the number of iterations, the time and effort required for building the whole net is very little. Accuracy of identifying characters is also phenomenal. The compressed matrix is very helpful for the MCAMH net to get to the results very quickly with more precision. Although the method has been tested for one input, the process represents no flaw at all.

IV. CONCLUSION

Based on the analysis it can be acclaimed with certainty that MCAMH is highly efficient in contrast to the previously made attempts. The compression algorithm has proven to be an absolute adherent to MCAMH net by eliminating all unwanted elements without losing data from the actual input and making the compressed input an accurate one.

REFERENCES

[1] Hand Written English Character Recognition using Column-wise Segmentation of Image Matrix (CSIM), Rakesh Kumar Mandal and N R Manna.

[2] Fundamentals of Neural Networks, Architectures, Algorithms and Applications, Laurene Fausett, Pearson Education

[3] Rakesh Kumar Mandal and N R Manna, Hand Written English Character Recognition using Row-wise Segmentation Technique (RST), International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC) 2011, Proceedings published by International Journal of Computer Applications (IJCA).

[4] Neural Networks, G.N. Swamy, G. Vijay Kumar, SCITECH