RESEARCH REVIEW JOURNALS
Volume-04
Issue-06
June-2019

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com[UGC Listed Journal]

A Study of Character Recognition Using Neural Networks

[1]Koasbatwar Shamsundar Prabhakar & [2]Dr. Anil Kumar

[1]*Research Scholar, Department of Computer Science & Engineering, Sri Satya Sai University of Technology and Medical Sciences, Sehore, Madhya Pradesh (India)*
[2]*Research Guide, Department of Computer Science & Engineering, Sri Satya Sai University of Technology and Medical Sciences, Sehore, Madhya Pradesh (India)*

ARTICLE DETAILS

Article History
Published Online: 12 June 2019

Keywords
Neural Networks ,improve, techniques, zonal approach, recognition accuracy.

ABSTRACT

To improve the recognition accuracy of the system with the identified the feed forward NN as classifier, the best possible feature set is to be obtained. Among the different feature extraction techniques, the statistics based zonal feature extraction technique is chosen for investigation. In zonal approach, horizontal and vertical feature extraction techniques are studied; both these methods have symmetry about x-axis and y-axis respectively. A new feature extraction technique, called diagonal feature extraction method is proposed in this thesis, which is symmetric about its diagonal. To further enhance the recognition accuracy of the hybrid feature based CRS without increasing the complexity of the neural architecture, a new training strategy, called novel training is proposed. To implement this novel training strategy, the alphabets with low recognition accuracy are identified. Low accuracy is defined using an adaptive threshold parameter. The characters having low recognition accuracy and their corresponding confused characters are identified. Generally, equal number of samples of all characters is taken for training. In this proposed novel method, more numbers of samples of the characters with low recognition accuracy and their confused characters are included for training. The optimal value for additional data sets, Δx of low accuracy characters is determined through an analysis. With the novel training strategy, the proposed CRSs – built one for uppercase and one for lowercase, are shown to exhibit improved recognition accuracy.

1. Introduction

It is imperative that a system that goes in for recognition is able to classify an object into an appropriate set or class. *Classification* depends upon the ability of the system to discover common patterns among objects, which help in decision making, learning and many other cognitive acts. The ability to classify, in turn depends upon the lea-ability of the system.

The methods of classification and recognition are guided by the end product of the preprocessing phase. As far as character recognition is concerned, the end product of the preprocessing phase may be a binary raster image, a gray level image, a contour plot or a thinned binary image. Various authors have worked with these different type of images for recognizing handwritten characters. Ours is a thinned binary image. Our inscription characters are fairly different and unique in themselves, what comes closest to them are the technologies for character recognition in OCRs and handwritten text. We discuss some techniques for character recognition.

Neural Networks have been used by many scientists for character recognition. Many other areas of pattern recognition apart from character recognition have used the concepts of Neural Networks. The properties of these nets that have gained them popularity and success especially in different fields of pattern recognition have been discussed in the following section.

❖ Neural Networks in the Field of Pattern Recognition

Neural networks have been applied to a wide variety of different areas including speech synthesis, pattern recognition of characters, diagnostic problems, medical illnesses, robotic control and computer vision.

Neural networks have been shown to be particularly useful in solving problems where traditional artificial intelligence techniques involving symbolic methods have failed or proved inefficient. Such networks have shown promise in problems involving low-level tasks that are computationally intensive, including vision, speech recognition, and many other problems that fall under the category of pattern recognition. Neural networks, with their massive parallelism, can provide the computing power needed for these problems. A shortcoming of neural networks lies in the long training times that they require, particularly when many layers are used. Hardware advances have diminished these limitations, and neural-network-based systems have become great complements to conventional computing systems.

Neural network architecture, Fuzzy Artmap, was used to classify finger prints. A fingerprint database of the University of Balamand was created. The algorithm uses the filter bank representation which captures the local and global details in a fingerprint as a 320-byte fixed length code which is suitable for storage. In the first stage, the fingerprint is classified using FAM to one of the five categories (whorl, left loop, right loop, arch, and tented arch). Then, the algorithm matches the fingerprint by searching in the specific category in the database. Matching is done using the two corresponding codes and thus, is very fast.

Another work describes a neural network based approach for automated fingerprint recognition. Minutiae are extracted from the fingerprint image via a multilayer perceptron (MLP) classifier with one hidden layer. The back propagation learning technique is used for its training. Selected features are represented in a special way such that they are simultaneously invariant under shift, rotation and scaling. Simulation results are obtained with good detection ratio and low failure rate. The proposed method is found to be reliable for a system with a small set of fingerprint data.

A Multilayer Perceptron Neural Network is considered for access control, based on face image recognition by Dmitry Bryliuk and Valery Starovoitov. They have showed the robustness of NN classifiers with respect to the False Acceptance and False Rejection errors. A new thresholding approach for rejection of unauthorized persons is proposed by them.

❖ Other Prevalent Techniques for Character Recognition

To be able to recognize characters, particular algorithm to perform matching needs to be decided. Matching the input character to some stored character needs some basis on which the matching is performed. Some matching or recognitions are based on shape. Thomson has experimented that related, though not identical shapes can be deformed into alignment using simple *coordinatetransformations.* Many scientists have used this idea to match shapes after a numberof finite deformations For silhouettes, trademarks and handwritten digits. For each sample point in one shape, a sample point in the other shape is found that has the most similar shape context. An aligning transformation is estimated that maps one shape onto another.

K-Nearest Neighbor method is also used in recognizing various texts. Animproved K-Nearest algorithm, for text classification is given by Baoli et al, for Chinese text recognition. Template matching is a standard image processing technique has recognized handwritten digits using template and model matching Moment invariants are used for pattern recognition. Zemike moments have been used by several authors for character recognition of binary solid symbols or the binary raster images. Belkasim et al have compared various moment invariants including moment invariants applied to solid binary structures using a K nearest neighbor classifier. Pavlidis has extracted approximate strokes from character skeletons. Kahan has augmented them with additional features for better recognition performance.

Some matching criterions for character recognition are based on unique features of the characters, which have been taken up extensively. Having explored various techniques that are used for character recognition, we decide upon ours which is reliant on the uniqueness of our characters.

❖ Neural Network for Character Recognition

We have explored and studied various techniques as discussed above for character recognition. These techniques are for printed documents or handwritten documents. Some researchers have used neural nets while others have used techniques like K-Nearest Neighbor, template matching, moment invariants to name a few. We are dealing with inscriptions that have unique properties of their own. Taking into consideration the characters that we have at our hands, their inherent properties and. the limitations that we face with them are enumerated below:

- highly distorted shapes
- skewed characters
- broken characters
- no uniformity in style
- one stone inscription font size is entirely different from the other due to the stone quality and the tool of digging
- characters are not clustered in words
- uncontrolled shapes

We endeavor to have a system that classifies as we humans do. No statistical means and measures should constraint the recognition. Our system needs to learn to recognize patterns, patterns that are not exact but somehow look like some stored pattern. A classification of characters is imperative to improve performance. We need the system to learn from large sample data. Neural network techniques facilitate such an environment.

The philosophy behind a feed forward net with back propagation can be explained as: An application of a multi-layer feed-forward network with a back-propagation training algorithm is to learn an unknown function between input and output signals from the presentation of examples. It is hoped that the network is able to generalize correctly, so that input values which are not presented as learning patterns will result in correct output values. The confidence level of recognition required an added aid of the fuzzy set membership which is useful in complete classification. A neural based recognition processes is based on the steps presented.

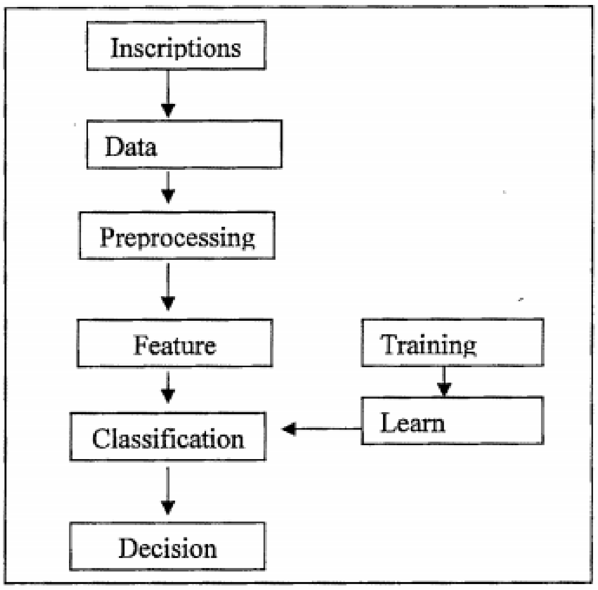

Figure 1 Outlines for Neural Based Recognition Process

2. Neural Networks as a Basis for Recognition:

The human visual system behaves as an adaptive system. The human brain, effortlessly, can perform innumerable tasks with the real-time interaction of the millions of competing and cooperating neurons. They are our high speed, distributed, non-linear, massively parallel processing units. They are explained in detail in the following sections. In conformance to the working of the human brain, we observe that *attention precedes recognition* We may see a very old, tom and damaged photograph of an old lady, put together all our efforts to place the person in the photograph. Faces of all the people we know have ever seen come together and our brain is able to sieve through them and point out the identity of the face.Quite easy for us, but not so for the conventional computer. Maintaining a large database of images and then matching the photograph in question is one solution. But in all possibility there may not be a perfect match or the image may be of youth and the database has an aged version, leading to a complete mismatch, which is not so. This is typically the problem that we have at hand. The human brain will still recognize the face. We look, pay attention and then recognize. Even in the case of not-so-crisp data, our brain accomplishes the astonishing fete and recognizes correctly.

This task is very special in it and is specific to the resultant a character to which matching is intended. Though we have revived the data on stone slabs, we are yet not able to make much sense to it. Despite the novel techniques used, the characters are broken, incomplete, and fuzzy. To be able to see a broken 'O' and recognize it as an 'O', is a task which is simple only to the brain. Again, to see a slanting, small sized 'o' and recognize it as a 'O', is not an easy task even for an intelligent machine. Calculation of the Euclidean distance or the Hamming distance will only confirm a mismatch between the two characters, taking us further away from our resolve. Only the millions of neurons in the brain, busy processing in parallel, will categorize the correct character. The brain has learnt a variety of characters over time, and knows that a 'similar looking shape' is probably the character in question. Infact, the brain may not have learnt anything about the image, and is seeing an image for the first time, but still may be able to categorize an image into a 'type'. Such learning is coined as 'supervised' or 'unsupervised' learning as per the path that led to recognition.

1.Working of the Brain and its Learn-ability

The brain is an Information Processing System, containing about 100 billion nerve cells or neurons. The brains' network of neurons is capable of massive parallel processing, as against a single processor executing a single series of instructions. Statistics show that a human neuron can operate at a maximum rate of 100 Hz, while, several hundred million machine level operations per second are executed by a conventional CPU. The hardware of the brain despite being slow, has remarkable capabilities.

- Under partial damage, its performance tends to degrade gracefully. On the other hand, a machine stops to work if any part is damaged or removed.
- It can learn from experience. In other words it can reorganize itself.
- Healthy units can learn from damaged areas which facilitates recovery. Machines are unable to exhibit such a property. Of the 100 billion neurons that we are bom with, many of these are not replaced when they die. Despite of our continuous loss of neurons we continue to learn.
- The brain can perform massively parallel computations extremely efficiently. A complex visual perception occurs in time comparable to only 10 processing steps.

Many complex short distant connections exist between a neuron and its immediate neighbors. A neuron has three main parts, dendrites which are the inputs, cell body and axon or output. A neuron receives input from other neighboring neurons, these inputs are summed together and once input exceeds a critical level, the neuron discharges a spike, an electrical pulse that travels from the body, down the axon, to the next group of neurons. This link is called a 'synapse'. Each neuron is connected to other neurons using thousands of synapse. The extent to which the signal from one neuron is passed on to the next depends on many factors, like the amount of chemical available, the number and arrangement of receptors or the receiving neurons, the frequency with which a signal is transmitted etc. A nerve cell may fail to fire at an instance of time.

3. Components of an Artificial Neural System:

Artificial neural network consists of numerous, simple processing units or 'neurons' that we can program for computation. We can program or train neural networks to store, recognize and associatively retrieve patterns. Artificial neural systems may contain many nonlinear neurons and interconnecting synapses. Neural networks can learn new patterns and recall old patterns. Neural networks are model-free estimators. This means that those neural networks do not use any mathematical model of the system's outputs' dependence on the input. The same network architecture can be applied to many problems; Even if a pattern cannot be defined by us in terms of mathematical formulations, it can be recognized by the neural nets. This property is known as the property of *recognition without definition.* It characterizes much intelligent behavior, by means of generalization. The neural nets store patterns with distributed encoding on the many synaptic connections between the neurons. Distributed encoding enables neural networks to complete 'partial patterns' and clean up noisy patterns. This property motivated us in using this network for pattern retrieval and recognition. The encoding also enables graceful degradation while, fault tolerance.

An artificial neural net is defined completely with the aid of the following main components:

- The constituent neurons
- Arrangement of neurons in layers
- Activation rules
- Learning rules
- Supervised Vs, Unsupervised learning
- Connection weights
- Learning parameter

4. Learning Rules: Pereeptron And Delta

How a network learns, and how well it memorizes, depends upon the connectionist weights. These connectionist weights undergo iterative changes under some learning rule such that the output is as desired by the training environment. The Pereeptron Learning Rule and the Delta rule are the basic rules.

The Pereeptron and its Learning Rules: The inputs and outputs are binary. The response of the pereeptron is a function of the stimulus applied at the input and the weights at the connections. Though the one given by Rosenblatt was more complex, this figure correctly represents the concept.

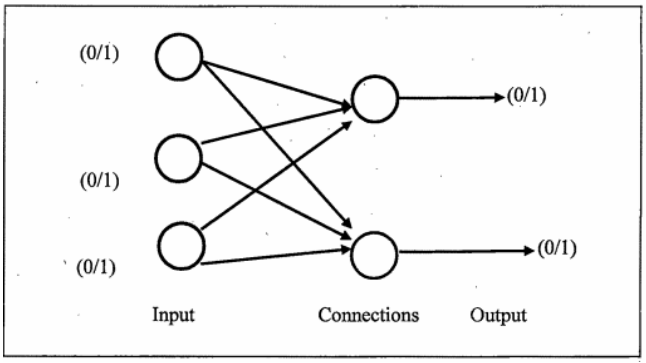

Figure 2 a Simplified Perceptron

The Perceptron Learning Rule(PLR) uses the iterative weight adjustment that is more powerful than the Hebb's rule given in Equation 4.9. Rosenblatt first proved the perceptron learning theorem. Perceptron learning can be proved to converge to correct weights if such a set of weight exists. The system learns only in case of error, i.e. the output is 1 when it should be 0 or it is 0 when it should be 1. When the output is 1 when it should be 0, the activation of the output neuron is higher than required, that means that too large a weight is attached to a connection with an active neuron. So a decrease in strength is proposed. If the output is 0 when it should be 1 implies that too low a weight is attached to a connecting active neuron, an increase in strength is required to correct the error.

Delta Rule: The Perceptron Learning Rule adjusts the connection weights to the output neuron whenever the response is incorrect. The Delta Rule adjusts the weights to reduce the difference between the input to the output neurons and the desired output. This rule is used in the Adaptive Linear Neuron (ADALINE) where the activation function is identity function. The learning rule strives to minimize the mean squared error between the activation and the desired target value. This leads to the improved ability of generalization. By generalization we mean that the system is able to respond to input which, if not identical, is similar to the one with which training was done. This Windrow Hoff training rule for the single layer network, ADALINE is a ground rule for the Back propagation network of multiple layers. An ADALINE is a single neuron that receives input from several input neurons.

5. Architecture of Multi-Layer Feedforward Neural Network:

Addition of hidden neurons to a single layer network increases the class of problems that are solvable by feed forward networks. A learning rule to define the optimum weights is needed. The activation functions tend to be non-linear, and the original delta rule discussed in the above section needs to be extended. The training method used under this architecture is the *generalized delta rule* or the *Back propagation rule.* It is a nonlinear extension of the LMS rule or Delta rule. Thisalgorithm overcame the limitations of Perceptrons. It is a gradient descent method for minimizing the total squared error of the output computed by the network.

As the name suggests the network will have at least two layers of interconnected weights which can go up to many layers, depending upon the number of hidden layers. Back propagation algorithm does not always converge, so an appropriate learning algorithm needs to update the connectionist weights. Some choices of initial conditions lead to oscillations, which we have handled in our experiments with Script. Optimum choices of the learning parameter, momentum, number of hidden layers and the neurons in the hidden layers help to dissipate these problems.

The general architecture of a multilayered Neural net is shown .

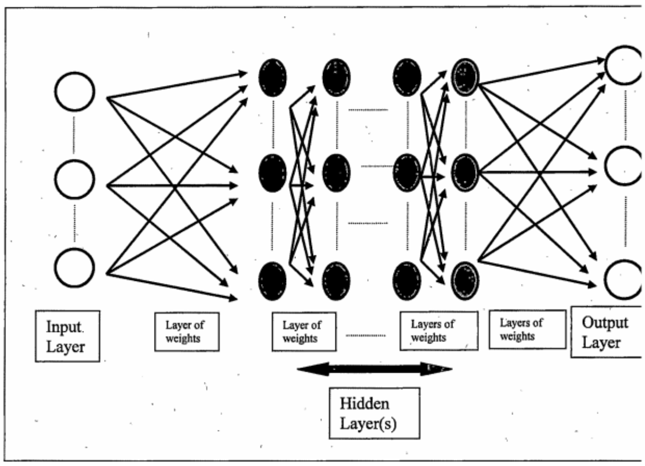

Figure 3 Multi-layer Net

Back-propagation networks use supervised learning. For a given input, they are trained for the expected response or target. The neurons of the output layer compute the error as the function of the response generated by the network and the requisite response. The weights on the connections are re-evaluated using the learning rule. The focus is to decrease the error in the output response. The error is propagated backwards from the output layer to the hidden layer just beneath it, the error contribution due to this hidden layer is calculated, adjustment made, and propagated back to one hidden layer below it, and so on, till the input layer is reached. With the updated set of weights, the network feeds the input in the forward direction again and checks the error of the output layer. The error in the hidden layer is estimated as the weighted average of the error of the output neurons. If an output cell has a large positive error, and if the connection weight between a hidden layer neuron and that output neuron is large, it implies that hidden neuron has a substantial effect in the output neuron error. The adjustment procedure is same for all the neurons; the error calculation procedure differs in the hidden layers because there is no expected value from the hidden layers.

The training of the feed forward network by Back propagation involves the following stages:

- Feed forward of the input training pattern
- Back propagation of the calculated error
- Adjustment of all weights, and repeat the first stage till all training patterns give no error or the error is in a tolerated limit

After the training is achieved, the application requires only the feed forward phase. A trained net produces outputs in much less a time as compared to the slow training process.

We categorize these architectural efforts as successful when the network *memorizes* all training patterns that it is trained on, i.e. when presented with the samepattern again, it gives the right output. On the other hand, it should also be able to give a reasonably good response, or *generalize,* when the test pattern is different from the ones on which training is done. In our work, the characters that we are wanting to recognize are distorted, and many times bear only partial resemblance to the actual character on which the training is done. We would want the system to recognize all such and more characters correctly. This prompts us to use it for recognition of inscriptions. One hidden layer is sufficient in many learning environments, and the number of hidden layers may be increased for other applications. Our experiments with the characters have used the neural network, which we refer to as *Net.* We have formulated the net with a single hidden layer, and also those with two hidden layers.

6. Net: Character Recognition Feedforward Neural Architecture

We start our recognition with training a single-hidden-layer feed forward back propagating network with the training sets of characters, each of dimension 8x8 pixels. Table 2-6 shows some exemplars that the network is trained on. A total of 198 exemplars is used. The initial condition of the number of hidden layers and number of neurons in each of them is varied and experimental result noted. All the results are shown. The number of hidden layers is increased to two and the effect noted. The error per cycle is noted and those values chosen that gave good tolerance levels.

The learning rate is varied between 0.01 and 1.0. The activation rule is formulated as a sigmoid function. Its derivative is used in the learning rule, which is the generalized delta rule. This rule incrementally decreases the output error. As the recognition is for 33 characters, the output layer consists of 6 neurons, to accommodate upto 36 - 1 unique character. As the input characters for training as well as testing is 8 x 8 pixels wide, the input layer has 64 neurons. The general architecture for the single layer Net is given in Figure.

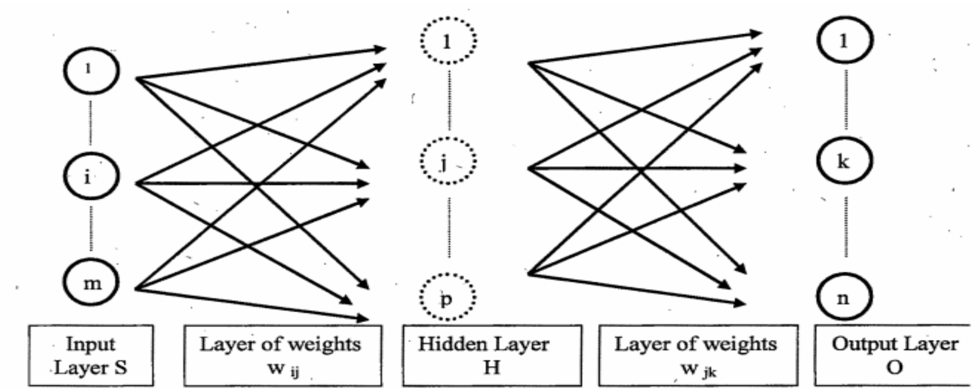

Figure 4 Single Hidden Layer NN Architecture used in our Results, m=64, n=6

Table 1 gives a list of terminology that we shall be using in explaining the architecture and learning rule of Net.

Table 1 Symbolic Notation for Net with One Hidden Layer

Terminology used:	
S_i	The neurons of the input layer, i=1,2. .. m
s_i	Activations of the input layer neurons, i=1,2. .. m
H_j	The neurons of the hidden layer, j=1,2. ..p
	The input to H_j, $input(H_j) = \Sigma\, s_i\, w_{ij} + w_{0j}$, w_{0j} is bias
h_j	Activations of the hidden layer neurons, j=1,2. .. p
	$h_j = f(input(H_j))$
O_k	The neurons of the output layer, k=1,2. ..n

	The input to O_k, $input(O_k) = \Sigma h_j w_{jk} + w_{0k}$
o_k	Activations of the output layer neurons, k=1,2. .. n $o_k = f(input(O_k))$
w_{ij}	Weights on connections between input and hidden layer neurons
w_{jk}	Weights on connections between hidden and output layer neurons
l	Learning parameter
input	sigma function to calculate the net input into a neuron
f	activation function, sigmoid in this case
δ_k	Error correction weight adjustment for the weight w_{jk} due to error in the output neuron O_k
δ_j	Error correction weight adjustment for the weight w_{ij} due to backpropagation of error from the output neuron to the hidden neuron H_j
δw_{ij}	Weight adjustment on the connection between i input neuron and j middle neuron as calculate by the steepest descent method
δw_{jk}	Weight adjustment on the connection between j middle neuron and k output neuron as calculate by the steepest descent method

The choice of connection weights will influence whether a net reaches a global or only a local minimum. The update of the connection weights between neurons is a function of the upper neurons' activation function and the activation of the lower neuron. The weights should be so chosen such that the activations and derivatives of activations are not zero. If the initial weights are very large, then the input to the hidden layers or output neurons will fall in a region where the saturation has a very small value. If the initial weights are too small, the net input to the hidden or output neurons will be very low which will lead to very slow learning. The values may be positive or negative and can be put in a range like,we have started with randomly chosen weights on connections. During training we also give choice to save the weights into a file, and then with changed parameters resume training with the saved connection weights.

7. Conclusion

This research has successfully included various parameters related to quality, quantity, and category along with miscellaneous during the process of development thereby making it benchmark level. A platform independent methodology is used during the process. Thus making the methodology applicable for any script of a given language. Though the dataset is at a benchmarking level, it is verified against a series of feature extraction and classification methods for its stability. Verification of accuracy factor against recognition has proven the consistency and stability of the data set during document analysis. Also, an applicative level testing is carried out to determine the efficiency of dataset during writer-identification process. Thus the aim of this research is successfully completed.

Optical character recognition (OCR) architectures use set of handcrafted features for recognition of isolated characters. We employed an unsupervised feature learning using DBN. DBN has stack of RBM layers. RBM extracts efficient feature representation of character images. Weighted parameters of unsupervised model were fine-tuned using supervised back propagation method. Deep learning uses raw pixel of character image to learn about the structure and form the feature vector.

Having implemented clustered BPFFNN, we have seen a substantial improvement in the recognition rate. It has gone up from 60 % to 75%. Broken and noisy characters are satisfactorily recognized. Kohonen has helped form *crisp* classes, which have resolved the problem of mis-recognition to a great extent. The classes formed have been through a robust process of clustering, wherein apart from the standard characters, the training has been affected on variations and distortions of the characters. This learning strategy has resulted in forming better classes with maximum cohesion and least coupling.

Among our observations, we have seen that some characters though very crisp in revival, show prevalent confusion despite being classified. This was due to the features of certain broken and noisy characters. As per the model and algorithm of Clustered NN given by us, a character is recognized by the class members of its immediate group only. But in our unique environment of stone inscriptions, imprecise data leads to imprecise results during certain recognitions.

References

1. B.B. Chaudhuri and SumedhaBera, "Handwritten Text Line Identification In Indian Scripts", Proceedings of the tenth international conference on Document Analysis and Recognition, Vol. No 2, pp.636-640, October 2009.
2. U. Pal, S. Sinha and B. B. Chaudhuri, "Multi - Script Line identification from Indian Documents", proceedings of the seventh international conference on Document Analysis and Recognition, Vol. No 2, pp. 880- 882, August 2003.
3. U. Bhattacharya and B. B. Chaudhuri, "Handwritten Numeral database of Indian Scripts and Multistage Recognition of Mixed Numbers" IEEE Transactions on Patten Analysis and Machine Intelligence, Vol.31, No.3, pp. 444 - 457, March.2009.
4. VeenaBansal and R. M. K. Sinha, "Segmentation of Touching and Fused Devanagari Characters", Pattern Recognition , Vol. 35, No. 4, pp. 875-893, April 2006
5. A.W. Senior and J. Robinson, "An Offline Cursive Handwriting Recognition System", IEEE Transactions on Patten Analysis and Machine Intelligence, Vol.20, No.3, pp. 309-321, March 2008
6. Reena Bajaj, LipikaDey and S. Chaudhury, "Devanagari Numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, Vol.27, part I, pp. 59-72, February 2012.
7. LuizOliveria, F. Bortolozzi and R. Sabourn, "Automatic Recognition of Handwritten Numeral Strings: A Recognition and Verification Strategy", IEEE Transactions on Patten Analysis and Machine Intelligence, Vol.24, No.11, pp. 1438-1453, November 2012.
8. F. Camastra, M. Spinetti and A. Vinciarelli, "Offline Cursive Character Challenge: a New Benchmark for Machine Learning and Pattern Recognition Algorithms" proceedings of the eighteenth international conference on Pattern Recognition, pp.913- 916, August 2006
9. F. Camastra, M. Spinetti and A. Vinciarelli, "Offline Cursive Character Challenge: a New Benchmark for Machine Learning and Pattern Recognition Algorithms" proceedings of the eighteenth international conference on Pattern Recognition, pp.913- 916, August 2006
10. VeenaBansal and R. M. K. Sinha, "On How to Describe Shapes of Devanagari Characters and Use Them for Recognition", proceedings of the fifth international conference on Document Analysis and Recognition, pp.410- 413, September 2009.
11. BrijeshVerma, "A Contour Code Feature Based Segmentation For Handwriting Recognition", proceedings of the seventh international conference on Document Analysis and Recognition, Vol. No 3, pp.203 - 207, September 2005.
12. Ahmed, M., Ward, R., &Kharma, N. (2004). Solving mathematical problems using knowledge-based systems. *Mathematics and Computers in Simulation*, *67*(1), 149-161.
13. Álvaro, F., Sánchez, J. A., &Benedí, J. M. (2016). An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognition*, *51*, 135-147.
14. Alvaro, F., Sanchez, J., &Benedi, J. (2012, September). Unbiased evaluation of handwritten mathematical expression recognition. *International Conference on Frontiersin Handwriting Recognition (ICFHR),* (pp. 181-186). IEEE.
15. Aly, W., Uchida, S., & Suzuki, M. (2008). Identifying subscripts and superscripts in mathematical documents. *Mathematics in Computer Science*, *2*(2), 195-209.