Genetic Algorithm Optimized Neural Network for Handwritten Character Recognition

Tarandeep Kaur
Department of Computer Science & Engineering,
Guru Nanak Dev University, Amritsar, India

Amit Chabbra
Department of Computer Science & Engineering,
Guru Nanak Dev University, Amritsar, India

ABSTRACT
Handwritten Character Recognition is well known problem which has many real world applications. Many solutions have already been proposed using various techniques (neural networks, fuzzy rules etc.) over a period of time, but no one is able to achieve 100 percent accuracy rate. Involvement of various organizations for research on handwriting recognition has been significantly exaggerated over last few decades. Solution is required which can provide higher accuracy rate in lesser amount of computation time. This paper covers introduction to problem and various terms used, proposed solution based upon Neural Networks whose weights have been optimized using Genetic Algorithm (GA) with newly designed fitness function and performance comparison of proposed design with existing techniques various constraints.

General terms
Handwritten Character Recognition, Genetic Algorithm

Keywords
Optimization, Neural Network

1. INTRODUCTION
Character recognition applications diversity makes this field of much importance in everyday life. Reliable automation procedures can reduce human effort to great extent for solving various complex problems. Combination of many elements result in advance of handwriting processing, for example: character recognition rate improvement, the use of various complex systems to integrate different kinds of useful data, and new technologies such as high speed scanners with high quality and more powerful CPUs [8]. The optimal goal of designing a handwritten recognition system with hundred percent recognition accuracy rate is near to impossible. Even people most often are unable to recognize every handwritten character with surety. For example, sometimes people find it difficult to read their self written text accurately.

Handwritten Character Recognition is well known problem in computer science. Efficient methods are important for increasing recognition accuracy in accepted computation time. Number of algorithms has been developed to address this task. Neera Saxena and Qasima Abbas Kazmi proposed a solution employing Neocognitron Neural Network Base Ensemble Classifiers. The method offers lesser computational preprocessing in comparison to other ensemble techniques as it ex-preempts feature extraction process before feeding the data into base classifiers [6]. Velappa Ganapathy, and Kok Leong Liew prposed a technique based upon the Multiscale Neural Network Training which has the ability to capture and recognize the various characters in different language that lead to the development of various "smart" devices [5]. Character recognition using neural network was proposed by Fakhraddin Mamedov and Jamal Fathi Abu Hasna which was capable of creating the Character Recognition System, in which Character Matrix and a corresponding Suitable Network Structure was created that also gave knowledge of Deriving the Input from a Character Matrix [7] .

1.1 Handwritten Character Recognition Problem
Character recognition was considered to be easily solvable problem at first. But later it turned out that this problem is much more challenging than it was expected out to be by most of the researchers involved in this field. The challenge is still there and an unconstrained recognition system for this problem which can match human performance is still unavailable. The system performance deteriorates very rapidly with the input of poor quality or with the new handwritten font introduction. Recognition of character also consumes lot of time and still results are not accurate as expected. So, adaptation to environment changes by single systems is not that easy. Systems are exposed to a large number of fonts during training phase along with their variants, so some sort of self learning systems are required. The neural networks are most suitable for this purpose as they are capable of learning from the known inputs. Neural network based handwritten character recognizer is to be designed and implemented whose weights during training phase are to be optimized using genetic algorithm. Neural Network based existing handwritten character recognizer s proposed by Fukushima is known as neocognitron. The objective of study is to improve the design of neocognitron neural network. Also, improvement in the training of proposed neural network is to be achieved by use of genetic algorithm. This will improve overall performance in recognizing handwritten character.

1.2 Neural Network
Artificial Neural Network is a collection of neurons. Every single neuron or cell is a processing element which can be connected to other neurons to form a network. Each neuron derives its input from one or more other neurons. There exist multiple weighted connections between various neurons. In this type of system design, input is distributed throughout the network and an output is obtained at the output layer from one or more activated cells.

The handwritten character recognition is referred as one of the most complex problem by experts. There is not even single recognizer system available which can match the human performance. The success rate of a system deteriorates very sharply with degradation in the input quality or variation in the input due to the introduction of new handwritten fonts. System adaptation to such environment changes is very difficult. For such a complex problem, a self learning system is required. A back propagation neural network can be used for this purpose. This network consists of many layers of interconnected elements. Each processing element computes its output as a function based upon weighted sum of its inputs. These weights are repeatedly optimized until a desired output is produced. Various existing solutions to the problem of handwritten character recognition make use of neural networks with acceptable rate of success. These networks can be implemented for processing and results integration of the classifiers by optimizing weights to produce required outcome. The major drawback of the neural network based systems is their incapability for generality. The possibility of under or over training makes this system design vulnerable. Also, structural description which is important for artificial intelligence is not provided by neural. The approach of using neural networks for the problem of character classification has not made much progress in this field. The recent developments in this field suggests for the use of various techniques together with efficient ways of combining them. The potentially conflicting decisions can be analyzed together in combination by use of multiple classifiers together to improve the procedure accuracy by using strengths of single classifier and ignoring their weakness.. The union and intersection of decision regions are the most suitable methods for various classification techniques combination.

1.3 Genetic Algorithm

Genetic algorithms are search and optimization algorithms based on the mechanics of natural selection and natural genetics [David E. Goldberg]. Here search refers to search for optimal solution in given solution space. These algorithms are useful in solving optimization problems by emulating biological theories [1]. Based on Darwin's theory of evolution, they work for survival of fittest [2]. Genetic algorithms can automatically access the search space and adaptively adjust the search direction to improve solution because they work on probabilistic rules rather than deterministic approach [3]. Most of genetic operators also work in accordance with random approach. Any random feasible solution is referred as chromosome. A population is composed of set of chromosomes. New population is generated by applying genetic operators like crossover and mutation on selected members of existing population. The value of fitness which is measured using fitness function is responsible for selection process. Whole process is terminated after some stability is achieved.

2. DESIGN METHODOLOGY

Handwritten digit recognition system is designed based upon neural network that is used to recognize handwritten digits. The training data set is created which contains 25 images of each digit. The handwritten digit images are processed to transform them into histograms and these resulted histograms are fed as input into a proposed neural network. This neural network outputs its judgment based upon training data set for matching the input digit image against the all possible digits available (0-9). The neural network is trained and then tested over a period and its accuracy rate is analyzed. Genetic algorithm is used for optimizing edge weights of neural network during the process of training. The results can show for which digit system needs more training to achieve high accuracy rate and for which digit is difficult to be identified by the system. Proposed algorithm is as follows:

Step 1: Get path to an image training data set.

Step 2: Read images and resize images to 64*64 pixel size if required.

Step 3: Determine all histograms (horizontal, vertical, left diagonal, right diagonal) and concatenate all histogram.

Step 4: Train the network with six input units,four hidden layer,one output layer.

Step 5: Test the trained network from training data and optimize neural network weights using GA.

Step 6: Analyze the accuracy rate for processed image.

Step 7: Generate output.

The working of proposed algorithm is shown in Figure 1.

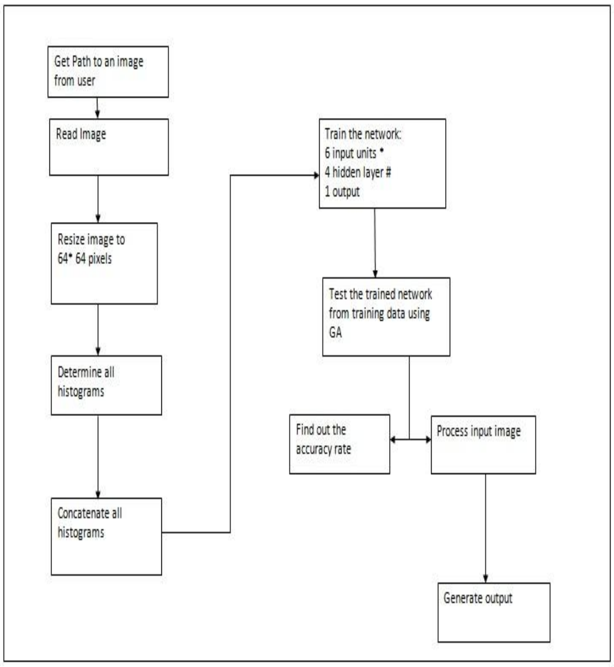

Figure 1: Working of Proposed algorithm

In order to have reasonably workable learning task, pre-processing of the various digit images is carried out and then training is done using proposed artificial neural network design structure. The pre-processing involves image resizing and histogram calculations. This is done before the digit images are fed to the neural network. The challenging task is there are some handwritten digits that often supplied together to the network or are not fully connected. Digit 5 is an example which can be misread as digit 6. The digit images are supposed to be available as individual items, though can still be of different dimensions. A normalization step is required to overcome this issue so that all digit images from data set can be equally resized. Once normalization step is done, the normalized images are then fed to the proposed artificial neural network design.

2.1 Neural Network Processing

The proposed design structure is a three layered feed forward network. The input is a digit image in question which is resized to the dimension 64 x 64 that equally corresponds to the dimension of a normalized training data set images. The structure of proposed neural network which is fixed in nature can be shown in Figure 2. The first layer contains 6 groups of units. Group 1 and group 2 contain 64 units (for horizontal and vertical histogram) per group. Group 3 and group 4 contain 127 units (for right end diagonal and left end diagonal) per group. Group 5 and group 6 contain 1 unit per group. Figure 3 shows the histogram values that are fed as input to neural networks first layer. The second layer contains 4 groups of units. Group 1 contains 64 units (for sum of horizontal and vertical histogram. Group 2 contains 127 units (for sum of right end diagonal and left end diagonal). Group 3 contains 382 units (sum of all histogram). Group 4 contains 1 unit. The third layer contains 1 group with single unit. This unit gives identified character as final output.

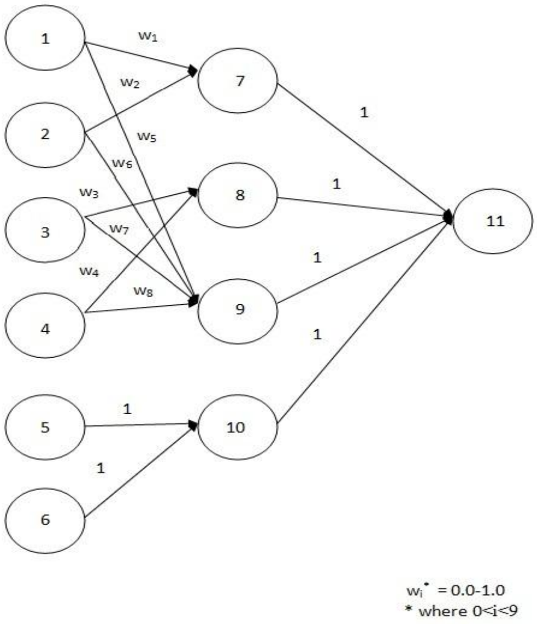

Figure 2: Structure of Proposed Neural Network

Set of handwritten character images is provided as training set to neural network. Each image is resized to 64 X 64 for normalization. Every image upon normalization is fed into the network to train one by one. The weights on neural network edges are randomly generated and then optimized in accordance with training data using genetic algorithm. The information is saved into the network. Figure 4 shows the histogram values for whole training data set during training phase which are calculated as output at second layer of the network. Once training phase is complete, network becomes ready for processing. Any handwritten image can be supplied as input and system based upon its training can evaluate the digit present in input image.

After training, the network testing is performed and the accuracy rate is analyzed to be 99% on average. This is a much acceptable recognition accuracy rate. The proposed network was stable for particular data set but as more variation is done in training data it becomes unstable because the training results changes with change in number of sample images available for a particular digit in training set. If we take numeral "2" as an example, In order to reach 99% accuracy today 20 times training may be required. But to reach same accuracy level tomorrow, we may have to train 25 times. So, revised training of the network is required with change in input data sets.

```
Processing image of 64 * 64
neural input: 0000000000000000125543333222122133334161000000000000000000000000000011111111111111113665444433323324444517111111111
917.jpg
Processing image of 64 * 64
neural input: 0000000000000000344322222222223222355610103000000000000000000000000011111111111111114554333333333343334667111141111
918.jpg
Processing image of 64 * 64
neural input: 0000000000000763222222222222232411416163500000000000000000000000000011111111111118743333333333334352151717461111111
919.jpg
Processing image of 64 * 64
neural input: 0000000043433234333233233323434449270000000000000000000000000000000111111111545443454443443444345455102811111111111111
920.jpg
Processing image of 64 * 64
neural input: 0000000000000000643211111112222223312394000000000000000000000000001111111111111111754322222222333333442241051111111
921.jpg
Processing image of 64 * 64
neural input: 0000000000000000185444233333333553000000000000000000000000000000001111111111111111296555344444444663111111111111111
922.jpg
Processing image of 64 * 64
neural input: 0000000000000000000943322221212222234104789400000000000000000000001111111111111111111105443333232333333451158910511
```

Figure 3: Reading Training Set Images

```
Histogram values for training data are::
[[74, 134, 96, 83, 387], [62, 120, 113, 107, 402], [102, 164, 113, 111, 490], [82, 144, 93, 91, 410], [66, 128, 86, 84, 364], [86
[[5, 67, 43, 41, 156], [44, 97, 73, 71, 285], [12, 72, 52, 48, 184], [6, 66, 37, 33, 142], [18, 80, 47, 45, 190], [8, 72, 8, 8, 9
[[96, 158, 98, 105, 457], [97, 159, 90, 106, 452], [98, 160, 109, 107, 474], [82, 144, 102, 100, 428], [67, 121, 68, 67, 323], [8
[[87, 139, 99, 96, 421], [81, 141, 103, 99, 424], [97, 161, 97, 97, 452], [74, 138, 65, 74, 351], [73, 137, 73, 73, 356], [75, 13
[[59, 119, 90, 86, 354], [68, 128, 99, 95, 390], [108, 166, 150, 144, 568], [40, 100, 80, 76, 296], [61, 102, 102, 97, 362], [90,
[[95, 155, 117, 113, 480], [80, 140, 102, 98, 420], [78, 138, 100, 96, 412], [65, 125, 87, 83, 360], [81, 143, 92, 90, 406], [79,
[[97, 159, 108, 106, 470], [81, 141, 112, 108, 442], [97, 159, 108, 106, 470], [93, 144, 124, 120, 481], [69, 131, 98, 96, 394],
[[62, 126, 62, 62, 312], [57, 121, 57, 57, 292], [73, 137, 73, 73, 356], [55, 117, 66, 64, 302], [57, 119, 68, 66, 310], [69, 131
[[111, 175, 111, 111, 508], [86, 146, 108, 104, 444], [85, 135, 108, 103, 431], [124, 176, 136, 133, 569], [88, 146, 121, 115, 47
[[95, 157, 106, 104, 462], [68, 126, 110, 104, 408], [85, 125, 109, 103, 422], [84, 144, 106, 102, 436], [85, 147, 96, 94, 422],
```

Figure 4: Training Data Histogram Values

2.2 Genetic Algorithm Processing

The GA involves three major steps- problem parameters encoding to form chromosomes in the form of strings, genetic operators like crossover and mutation application, fitness function based selection of individuals to create a new population. Genetic algorithm is applied to optimize edge weight in proposed neural network. Figure 5 shows the randomly selected initial population with 6 individuals and final optimized output after 10 generations. This output shows the value of edge weights taken for proposed structure of neural network.

```
initial pop
0.50.20.70.50.30.20.00.6111111
0.60.20.90.30.70.30.51.0111111
0.20.10.80.30.40.60.90.7111111
0.20.40.90.60.80.20.50.7111111
0.40.61.00.70.41.00.80.6111111
1.00.80.80.90.90.70.70.9111111
Optimized Weights for Neural Network after training by GA are: 0.80.50.20.60.10.90.50.5111111
```

Figure 5: Optimized Edge weights using GA

2.2.1 Encoding chromosome
The problem variables are edge weights for proposed neural network design structure. Each of the variables will be encoded into a string of 14 bit length. First 8 bits can have values between 0.0 and 1.0. Remaining six bits are fixed to 1. Initial population is created with such randomly generated 6 string.

2.2.2 Population
Population size is fixed as 06. So, for initial population 06 random chromosomes are generated with above specified constraints. This number was considered acceptable for used training data set. Population size may vary if there is any change in structure of neural network.

2.2.3 Fitness Function
An objective function is chosen in accordance with histogram values of testing data. Edge weight is multiplied with correspondent histogram value for each image. Then average for all images is taken as final fitness for particular bit. The sum for all bits is taken as final fitness of each string.

2.2.4 Selection
Once the population of chromosomes has been generated, the selection of parents has been done based upon a respective fitness function. The Roulette wheel selection method or the fitness proportionate selection method has been used for this problem. Under this method, chromosome with higher fitness has higher chances of being selected as parent. Five pairs of parents are selected for each iteration from one current population to form new population. The probabilities are calculated on the basis of survival chances of the individuals obtained in the terms of the fitness value from fitness function.

2.2.5 Crossover and Mutation
It is obvious that single point crossover in this case would have little effectiveness because of the large string length. The mutation can be much more effective in this case. Single point crossover is adopted for this problem. Crossover is applied with some probability on each pair of selected parents. The crossover probability for this problem is kept fixed at 0.8. Mutation alters one or more allele values in a chromosome from its initial state. Each of the bits in the chromosome is chosen individually to undergo mutation with some random probability. The mutation probability is taken as 0.7.

2.2.6 New Population and Termination Criteria
After cross over and mutation newly obtained individual are added to found new population. Whole processer is repeated for 10 number of generation to get optimize value for edge weights.

3. PERFORMANCE ANALYSIS
The trained network is tested many times for different digits. The results are shown in Figure 6.

Numeral	Test 1	Test 2	Tests	Test 4	Tests	Tests	Test 7	Test8	Tests	Test 10	Average
	Errors	Errors	Errors	Errors	Errors	Errors	Errors	Errors	Errors	Errors	
0	2	1	1	1	1	1	1	2	1	1	1
1	1	1	1	1	2	1	1	1	1	1	1
2	1	2	2	1	1	1	3	1	1	1	1
3	1	1	1	2	1	1	1	1	2	3	1
4	2	1	1	2	1	1	1	1	1	3	1
5	4	2	3	4	3	3	5	5	3	4	4
6	1	1	1	1	3	2	2	1	1	1	1
7	2	1	1	1	1	1	4	3	1	1	2
8	1	1	1	1	2	1	1	3	2	2	2
9	3	1	2	3	1	1	1	1	1	2	2

Figure 6: Errors for different digits

The results show that digit 5 in particular shows more errors as compared to other digits. This may be due to its resemblance to digits like 6 or 8. If training data set size is increased, then this problem can be minimized to great extent but on the other side it increases the training time required. Even with these issues an average accuracy rate of 99 percent is achieved, which is believed to be very good under this problem domain. Neera Saxena, Qasima Abbas Kazmi, Chandra Pal and, Prof. O.P. Vyas have achieved the only higher accuracy rate of 99.4 percent , but they used the data set of 18000 images which was not available for comparison. Also, they used original structure of neocognitron neural network proposed by Prof. Kunihiko Fukushima [4], which recognizes after 10 layers processing of the network. In comparison 99% of accuracy is achieved with training data set of 250 images and neural network structure of three layers only. Also, neural network weights are not kept constant to one, but were optimized using genetic algorithm. The results for enhanced performance are shown in Comparison table in Figure 7.

Algorithm	Technique Used	Average Test Accuracy Achieved	Data Set Size (Number of Images)	Time (Sec)		Memory (Mb)
				Train	Test	
Single Neocognitron Classifier	Neocognitron neural network proposed by Prof. Kunihiko Fukushima	99%	18000 (Size 16x16)	1342	123	1211
Ensemble of 5 Neocognitron Classifier	Neocognitron neural network proposed by Prof. Kunihiko Fukushima	99.4%	18000 (Size 16x16)	1955	235	1531
Proposed Algorithm	GA Optimized Neural Network	99%	250 (Size 64x64)	28	04	100

Figure 7: Performance Comparison

Another test is performed with same constraints. In order to achieve 99% accuracy rate for every digit, network is trained until hits that accuracy mark. The results are shown in Figure 7. It can be seen from results that digit 5 still needs more data set images for training to achieve acceptable accuracy rate. For this particular test, data set size is kept constant, but few images are supplied as input multiple times till the network becomes able to recognize the characters with desired accuracy rate. The structure of the network is also kept constant as proposed in earlier sections. So, it can be seen that system shows some problems in identifying digit 5 because of its similarities with other digits.

Numerals	Training 1	Training 2	Training 3	Training 4	Training 5	Training 6	Training 7	Training 8	Training 9	Training 10	Average
0	29	24	24	24	26	25	23	28	26	20	24.9
1	20	26	25	25	28	23	22	21	24	26	24
2	22	27	27	25	25	23	31	26	23	24	25.3
3	21	25	23	29	24	24	25	24	28	30	25.3
4	27	23	23.	28	26	24	26	25	26	31	25.9
5	35	29	32	34	32	30	38	39	30	36	33.5
6	22	21	23	24	31	28	29	23	25	26	25.2
7	28	23	24	24	25	25	35	31	26	25	26.6
8	24	25	26	26	29	26	21	32	28	28	26.5
9	27	26	28	30	26	24	24	25	21	29	26

Figure 7: Training required to achieve 99% accuracy rate

The performance of the system in terms of optimizing edge weights is also analyzed. For same training data set, the changes in crossover and mutation probability is studied with its effect on time taken (in seconds) to produce desired output i.e. the optimized weights.

By keeping the factors like number of generations, population size and the probability of mutation fixed at a certain value, the crossover probability is analyzed. The Figure 8 shows that the probability of crossover, gives better results in the range between 0.6 and 0.8.

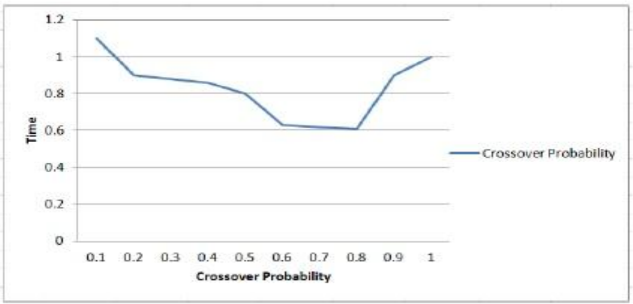

Figure 8: Crossover Probability Comparison Graph

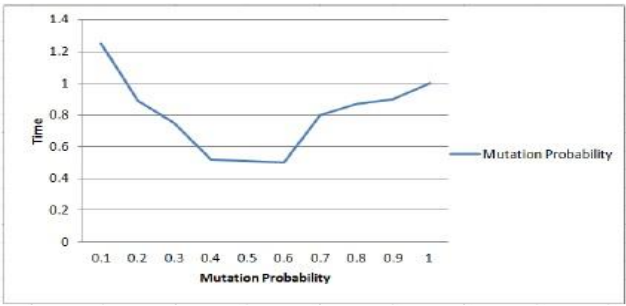

Figure 9: Mutation Probability Comparison Graph

By keeping the generation count, population size and the crossover probability fixed, the effect of change in mutation probability is analyzed. The analysis graphical representation as in Figure 9 shows that, the probability of mutation, pm, gives optimal results at a value between 0.4 and 0.7.

Thus for this particular training data set, value of 0.8 for probability of crossover and value of 0.7 for probability of mutation is chosen in order to optimize edge weights of proposed neural network structure.

4. CONCLUSIONS

Proposed system design of Neural Network in combination with genetic algorithm to recognize handwritten digits achieved high accuracy rate. The designed neural network was trained with data set contacting 25 sample images of each digit. The average accuracy rate of proposed system was analyzed to be 99% which is very acceptable. Though structure of neural network was fixed, the edge weights were taken randomly and were optimized using genetic algorithm in accordance with given training set. Use of genetic algorithm during training process improved overall results and performance of the system. It was concluded from the results analysis that the proposed system produced more errors while identifying images of digit 5. This may be due to its similarity to the digit 6 or maybe supplied images of this digit were not fully connected. Other than this scenario, the proposed system performed well with very high recognition accuracy rate on average. The proposed design comes out to be really efficient as number of layers have been minimized to great extent.

5. ACKNOWLEDGMENTS

Our thanks to Department of Computer Science & Engineering, Guru Nanak Dev University, India for providing us all required resources necessary to study and implement the solution to this problem. We also want to thank all the authors of referenced papers which guided us all the time as base for this study.

6. REFERENCES

[1] David. E. Goldberg, “Genetic algorithms in Search, Optimization & Machine learning”, Addison Wesley, Publishing Co. Inc. ,Boston, MA, pp- 1-25, 1990.

[2] Melanie Mitchell, “An Introduction to Genetic algorithms”, The MIT press, Cambridge, Massachusetts, England, 5th printing, pp 2-20, 1999.

[3] Pratibha Bajpai et al., “Genetic Algorithm- an Approach to Solve Global Optimization Problems”, Indian Journal of Computer Science and Engineering, 2010.

[4] K. Fukushima, “Neocognitron for handwritten digit recognition” Neurocomputing, Vol. 51 ,pp. 161– 180 ,2003.

[5] Velappa Ganapathy, and Kok Leong Liew, “Handwritten Character Recognition Using Multiscale Neural Network Training Technique”, World Academy of Science, Engineering and Technology 15, 2008.

[6] Neera Saxena1, Qasima Abbas Kazmi, Chandra Pal and, Prof. O.P. Vyas, “Employing Neocognitron Neural Network Base Ensemble Classifiers To Enhance Efficiency Of Classification In Handwritten Digit Datasets”, D.C. Wyld, et al. (Eds): CCSEA 2011, CS & IT 02, pp. 408–416, 2011. © CS & IT-CSCP 2011.

[7] Mamedov, Fakhraddin and Jamal Fathi Abu Hasna, “Character Recognition Using Neural Networks”, Near East University, North Cyprus, Turkey via Mersin-10, KKTC.

[8] Faisal Mohammad and Jyoti Anarase, “Optical Character Recognition Implementation Using Pattern Matching”, International Journal of Computer Science and InformationTechnologies, Vol. 5 (2) , 2014.