

# Review of Text Recognition Works

Mohit Agarwal  
Student:M.Tech  
Abescollege  
haziabad, India

Bajinath Kaushik  
Associate Professor  
Abes college  
Ghaziabad, India

## ABSTRACT

Character recognition is an important problem in Pattern Classification. It is difficult to recognize the character in images or pdf or scanned text or handwritten characters scan. There are various ways to recognize text which comprises of Neural Network, Genetic Algorithm, Soft computing and fuzzy logic. In this paper, we compare the various ways already explored by researchers. We have tried to compare the Artificial Neural Network and Genetic Algorithm for the character recognition. The character image is initially segmented into pixel array and then normalized and skew removed with feature extraction to form an input to the neural network or the genetic algorithm. We work with two set of data, training data and test data and the aim is to recognize character correctly in test data.

## General Terms

Pattern Recognition, Review, Image Processing, Artificial Neural Network, Genetic Algorithm et. al.

## Keywords

text classification, text selection, digits recognition, neural network, genetic algorithm, ANN, GA, OCR.

## 1. INTRODUCTION

The various ways in the neural network and genetic algorithm are explored by authors in references works. We are giving a brief of the problem explored by them and ways to solve it with difficulties encountered in their approaches below.

## 2. REVIEW OF EXISTING WORK

M.Seetha et al [1] compare various neural network ways to classify images using pixel array of image. Authors compare four ways of image: Neural Network, Support Vector Machine, Fuzzy Logic and Genetic Algorithm. Its weakness is a substitute method is to apply the genetic algorithm for neural network structure optimization; it is left for future work.

Fakultamatematiky [2] helps to understand how pixel map of hand written digits or pictures of traffic lights can be mapped to neural network input nodes and get a desired correct analysis based on activation function. He compares different Neural Network ways to train the system for handwritten characters. Its weakness is: does not give examples for text characters (only digits) and its input matrix formation.

Vivek Shrivastava and Navdeep Sharma [3] uses back propagation for character recognition by taking pixel sum along some horizontal and vertical lines. Calculates 11 parameters like sum of the pixel of character along horizontal lines or vertical lines and trains neural network to feed test images with 11 parameters calculated and recognize the character. Its weakness is tested data is not shown to estimate the working of the algorithm.

Md Fazlul Kader and Kaushik Deb [4] shows the boundary detection and normalization of pixels and step wise weight

learning for an input matrix with the final test of actual data. Shows pixel map forming with normalization and feeding to NN to get an output based on calculated weights. Its weakness is proposed training algorithm calculates weight in a supervised manner and established the network by adjusting weights on which basis is not explained or given.

Ankit Sharma and Dipti R Chaudhary [5] shows the feature extraction and matching them one by one with trained patterns to get the input image in text form. It discussed feature extraction to separate characters in the continuous text and to recognize them from trained NN. Its weakness is: does not explain much how to train NN or get the recognized output from it.

Pulipati Annapurna et al [6] uses freeman chain code to recognize digits which are different from our neural network based approach. It uses a new technique of freeman chain coding to recognize a character. Its weakness is briefly telling about freeman chain code but now extensively how to put for all English characters.

Nisha Vasudeva et al [7] briefly discuss the back propagation with feature extraction for character recognition. Authors compare different font's accuracy for different hidden nodes with a single hidden layer. Tells to segment the image and extract features to recognize the image using NN. Its weakness is Assumes to input a single character and not for continuous text.

SameekshaBarve [8] tells how to use ANN to give the values of pixels in 0,1 form and then train for desired outputs with hidden layer and back propagation. Describes how OCR technique in existing software uses NN with input as pixel array and output we desire for trained characters from a trained NN. Its weakness is: does not explain how to make the pixel array or train the NN and what outputs we can pre-determine for all characters.

Sandeep Saha et al [9] tried to divide the character into 16 zones and then use ANN to train for handwritten patterns and try this ANN on test patterns. Authors discuss 40 point feature extraction of characters using the pixel value in a matrix of 16 zones matrix and picking 40 sets from these zones to be fed as the input to NN. Its weakness is The NN way is assumed to be Back Propagation without much elaboration and accuracy is not very high.

K.Venkata Reddy et al [10] tries to show results for handwritten single character recognition with MLP back propagation. Developed an NN with input nodes and hidden nodes as input parameters and trained for characters to get 2 possible values of test data. Its weakness is assuming test data to be a single character and details of NN are not given.

Vipin et al [11] present handwritten English characters recognized using shape based zoning features with the help of neural network (NN) as a classifier. Explains feature extraction by taking projections along vertical and horizontal

lines and feeding them to NN with different epochs to compare the accuracy. Its weakness is NN part is very brief and explanation of input nodes and learning is missing.

Chirag I Patel et al [12] applied some basic algorithms for segmentation of characters, normalizing of characters and De-skewing. They have used different Models of Neural Network and applied the test set on each to find the accuracy of the respective Neural Network. Help to understand edge detection, normalization and skew analysis of character images. Also shows accuracy comparison with more hidden layers and hidden nodes. Its weakness is: does not show the accuracy calculation mechanism and some insight on logic with different hidden layers.

Rahul Kala et al [13] describe the sum of square method for edge matching behavior using graph techniques and genetic algorithm (GA). Uses graph techniques to compare images to recognize a character from training data. The results improve with GA also performed between training and test data. Its weakness is it assumes input in the form of a single character and not in continuous text.

VedguptSaraf et al [14] help us feel Hindi characters deciphering using GA, but basic algorithm is not given clearly. Describes the crossover using peaks of sub words to get the complimentary Hindi character. Its weakness is: does not explain optimization function to recognize the image and how to extend to English characters.

Majida Ali Abed et al [15] Unknown character is read from a file and many operations will perform on it to manipulate it and extract its features, to compare these features with saved template's features. It gets a high accuracy in recognizing a character of different sizes from a stored character file. Its weakness is: does not describe a way of continuous text and also how crossover is done with whom to decide fitness offspring.

ChomtipPornpanomchai et al [16] Shows there are 3 main features of Thai characters which are the stroke, loop and location of loop and stroke connection. Fourth, the extracted character information is kept in the form of bits string chromosome in a genetic algorithm. Finally, the system displays the best fitness chromosome for the recognition result. Makes a 66-bit chromosome from the thai character features and uses a function for calculating fitness to obtain 88 % accuracy in hand written thai characters. Its weakness is: does not explain the way to create chromosome or fitness function and if want to extend for English Characters.

### 3. RELATED WORK

#### 3.1 Text Recognition Techniques

The text to be deciphered needs to be converted into an input form to Neural Network or Genetic algorithm. Thus, NN needs a 0-1 input to some number of input nodes and Genetic algorithm needs a chromosome formation from the image to be crossover with the initial population. We list below the various ways discussed by referenced authors to convert the image:

##### 3.1.1 Segmentation

The image is read in the form of pixel array and using white spaces it is segmented into lines, words and characters. 0 is used for the white space and 1 for a filled pixel with black or another color. The 3 types of segmentation are:

1. Line Segmentation: It helps to divide the data into lines so it needs scanning an upper full white line of pixels and a bottom full white line of pixels.
2. Word Segmentation: It segments the words with a bigger white space in between them.
3. Character Segmentation: It uses a full vertical set of white pixels on left and a full vertical set of white pixels on the right.

##### 3.1.2 Normalization

The characters of different sizes so we need to make them of the same size. For example, we have a pixel matrix from segmentation step of size 5\*6 we want to convert into a matrix of size 10\*10. We duplicate the rows to make the matrix of size 50\*60 by duplication rows and columns. We then sub divide into matrix of 5\*6 and take a 1 if the number of 1's is more than half of pixels in this sub matrix or 0 else. Thus in the end we will get a normalized matrix of 10\*10.

##### 3.1.3 Skew Detection

The characters are often rotated by an angle so we should make them vertical before processing them or trying to recognize them. We can use the line equation of the form to determine the skew:  $Y = M * X + C$ . The Angle M is calculated using the formula:  $M = (n \sum x_i y_i - \sum x_i \sum y_i) / (n \sum x_i^2 - (\sum x_i)^2)$ . Thus, skew angle is calculated and image rotated appropriately to form a vertical character image.

*amit*

The skew angle of above text: 40 degrees.

##### 3.1.4 Feature Extraction

The features are extracted using graph techniques to know the equation of lines in the image. We can also calculate the projection of images on various lines at 0, 1/2, 1/4, 3/4 vertical or horizontal lines and take their absolute sum to use the values as GA chromosome or NN input nodes vector.

### 3.2 Neural Network Mechanisms

The input vector formed as in above section is used to feed to Neural Network and initially a training set is used to train for all digits or English characters. The various points to be remembered in Neural Network are:

#### 3.2.1 Back Propagation Network or Support Vector Machine

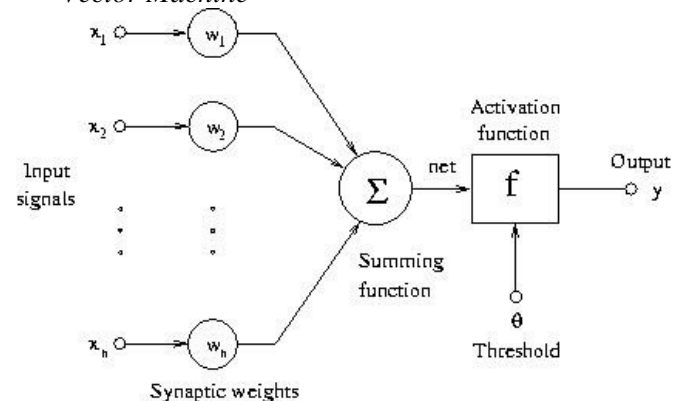


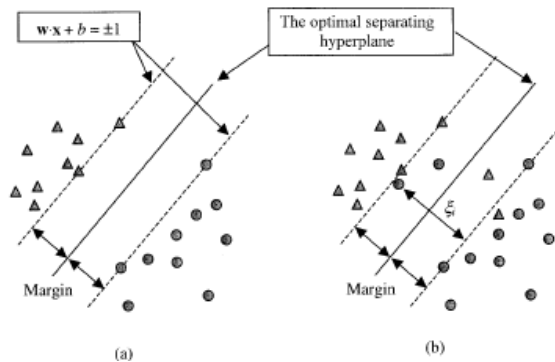
Fig 1: Neural Network diagram for training for given characters

The back propagation algorithm uses following logic:

1. Choose input  $x$  vector, compute output  $y$ , set  $E = 0$  ( $E$  – error)
2. Compute error function  $e(t) = \frac{1}{2} (d - y)^2$ ,  $E \leftarrow E + e(t)$  ( $d$  – desired output)
3. Adjust weights with delta rule (if  $e(t) > 0$ )
4. If all patterns used, then goto 5, else go to 1
5. If  $E = 0$  (all patterns in the set classified correctly), then end

else reorder inputs,  $E < 0$ , go to 1

The Support Vector Machine uses the following logic:



**Fig 2: Support Vector Machine for separating character patterns**

It tries to divide the data into hyper planes of the form

$$i \quad w \cdot x + b \geq 1$$

$$i \quad w \cdot x + b < -1$$

Where  $w = (w_1, \dots, w_n)$  is a vector of  $n$  elements.

Inequalities above can be combined into a

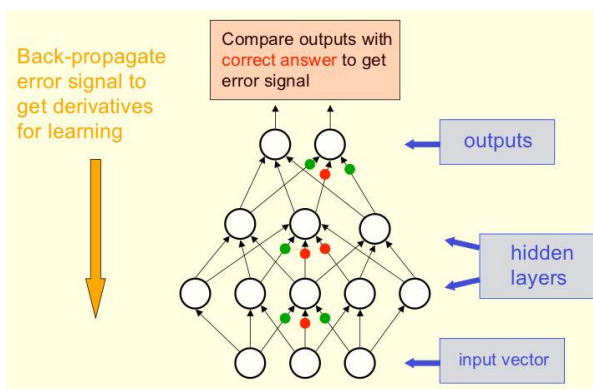
single inequality:

$$y [w \cdot x + b] \geq 1$$

This hyper plane can be found by minimizing the norm of  $w$ , or the following function:

$$F(w) = \frac{1}{2} (w \cdot w)$$

### 3.2.2 No of hidden layers and nodes



**Fig 3: Neural Network with hidden layers**

The back propagation of algorithm is tried with a various number of hidden layers and hidden nodes and accuracy or output is compared to know the best possible solution.

Comparison of hidden layers and accuracy can be seen in (from [12]):

#### i) Single hidden layer

Epochs	Number of Hidden Layer	Configuration (No. of nodes in HL)	Accuracy (%)
300	1	26	20
600	1	26	65
1000	1	26	82
300	1	52	25
600	1	52	69
1000	1	52	88
300	1	78	27
600	1	78	71
1000	1	78	91

**Fig 4: Accuracy for 1 hidden layer with different epochs**

#### ii) 3 Hidden layers

Epochs	Number of Hidden Layer	Configuration (No. of nodes in HL)	Accuracy (%)
300	3	26-52-26	31
600	3	26-52-26	65
1000	3	26-52-26	82
300	3	26-52-78	29
600	3	26-52-78	74
1000	3	26-52-78	92
300	3	78-26-78	27
600	3	78-26-78	71
1000	3	78-26-78	91

**Fig 5: Accuracy of character recognition with 3 hidden layers**

### 3.2.3 No of epochs used to calculate the weight matrix

The number of iterations in back propagation is called epochs used in Back Propagation to solve the weight matrix feasible for all input matrix. The comparison of epochs and accuracy is given in below figure (from [7]):

No of epochs	300	600	900
Font Style	Error		
Arial	3.44%	2.33%	1.11%
Tahoma	2.11%	1.11%	0
Times new Roman	0	0	1.11%
Bookman old style	2.11%	1.11%	0

Fig 6: Accuracy with different epochs and fonts

### 3.3 Genetic Algorithm Approaches

A genetic algorithm (or GA) is a soft computing technique used to find true or approximate solutions to optimization and search problems.

Genetic algorithms use techniques similar to evolutionary biologies such as inheritance, mutation, selection, and crossover .

The new population created is used in the next iteration of the algorithm.

Mostly, the algorithm terminates when either a satisfactory fitness level has been reached for the population, or a maximum number of generations has been produced

If the algorithm terminates due to a maximum number of generations, a satisfactory solution may or may not have been reached.

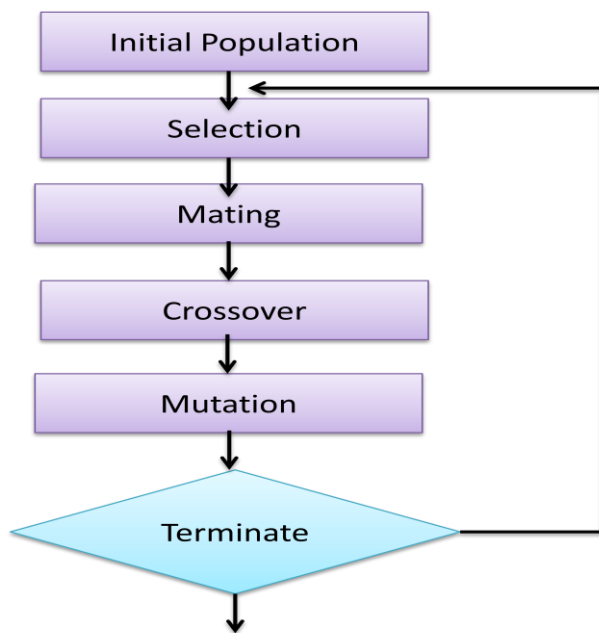


Fig 6: Flow Chart for Genetic Algorithm

The evolution usually starts from a population of randomly generated individuals or recorded set of individuals. The solution is found through repetitive application of the mutation, crossover, inversion and selection operators. Individual solutions are selected through a fitness-based process/function. This generational process is repeated until a termination condition as described above has been reached.

#### 3.3.1 Cross over

The chromosome created in Section II is crossed with trained or existing set of patterns chromosomes and next generation is created and it is compared with best fitting output using either

graph edge detection that is deviation of edges to know the input image pattern.

The graphical detection uses the following algorithm:

Step1: For every character c in language

Step2: For every input i for the character c in test data

Step3: Generate Graph grh of i

Step4: Generate graph inp of input image

Step5: For every character c in language

Step6: Use Genetic Algorithm to generate hybrid graphs

Step7: Return character corresponding to graph with the minimum most fitness function (out of the graphs generated in any genetic operation)

#### 3.3.2 Mutation

The mutation in output offspring is calculated using flipping 0s and 1s in the output chromosome and comparing with the existing chromosomes to see with the minimum mutation is the existing character in the test image.

#### 3.3.3 Maximum Offspring generation iterations

A simple and easy to implement stopping criterion is to stop the simple GAs if no improvement of the best solution has been made for a (large) predefined number of generations. This limits the accuracy of the results.

### 3.4 Other Mechanisms

#### 3.4.1 Freeman Chain Coding

Freeman Chain code is used to represent a boundary by a connected sequence of straight line segments of specified length and direction. This technique is based on 4 or 8 connectivity of the segments. The direction of each segment is coded by using a numbering scheme. A boundary code formed as a sequence of such directional numbers is called Freeman Chain Code. The chain code of a boundary depends on the starting point. Working with code numbers offers a unified way to analyze the shape of the boundary. Chain Code follows the graph of an image in the counter clockwise manner and keeps track of the directions as we go from one graph pixel to next. This uses the following algorithm to recognize character:

- 1) The binary code of the digit is saved i.e.; the border pixels are 1's and the remaining pixels are 0's.
- 2) From the binary code, we get the freeman code using 8-connectivity.
- 3) Similarly, we find the freeman codes, of all the digits (from 0-9) when we draw a test image its freeman code is compared with all the training data and finally the digit is recognized.
- 4) For comparison of freeman codes we find the number of 0's, the number of 1's ...to the number of 7's since it is 8- connectivity only 0 to 7 digits present. It checks for both training data and test data.
- 5) After that, we compare by finding those differences in code and recognize the digit with the minimum difference by the method.
- 6) Finally the system recognizes the digit.

## 4. PROPOSED WORK

### 4.1 Neural Network

The work done can be extended using hetero associative neural network to store the trained patterns and compare with the test data to get the test data character value with minimum of output values from 1 of 10 nodes for each digit. The input vector is formed by modifying the input matrix of pixels to vector form of 1 dimension.

Character: a						
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	0	0	0
0	1	1	1	1	1	0
1	1	0	0	0	1	0
0	0	0	0	0	1	1
0	1	1	1	1	1	1
1	1	0	0	0	1	1
1	0	0	0	0	1	1
1	1	0	0	1	1	1
0	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

Fig 7: Pixel map for character a

This matrix of 15\*7 is transformed to 125 nodes input vector of NN.

### 4.2 Genetic Algorithm

The digits are first given in a specified font size to the machine learning process. Algorithm is developed to parse the image and find out the pixels which are separated by white spaces on top and bottom and left and right. These pixels are stored in dynamic memory. We make similar array for test patterns and try to match the width of maximum 1 cross section to be able to fit with 1 cross section of training pattern at various heights e.g. 0, ¼ of max height, ½ of max height or ¾ of max height and max height. The training pattern with most matched is figured out to be the character present at the image location we are trying.

cross section width of digits at various heights

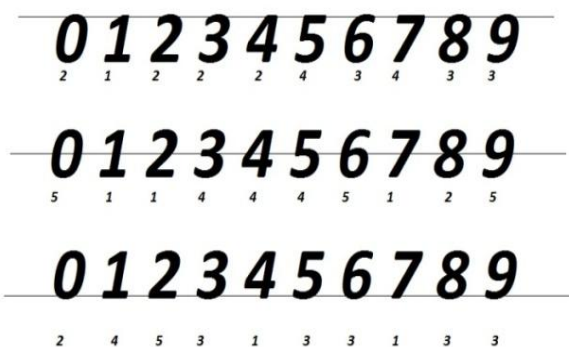


Fig 8: Projected tests for Genetic Algorithm

Table 1: GA Projected Test Case

GA test cases	Input/output
Test Image	i am mohit agarwal i live in ghaziabad i work in abes hope it passes the test
Output	i am mohit agarwal. i live in ghaziabab. i work in abes. hope it passes the test.

Table 2: Accuracy chart of this GA technique is

GA Accuracy	Accuracy Chart		
S.No.	Chars in test	Correct char o/p	Accuracy
1	61	60	98%
2	51	48	94%
3	148	132	89%
4	54	52	96%
5	70	65	93%

## 5. ACKNOWLEDGMENTS

Our sincere thanks to friends and faculty for helping in achieving our goal in making this paper for benefit of image processing research and future path of further accurate work in industry.

## 6. CONCLUSION

The art of recognizing characters in images could be manipulated by biological to machine mapping mechanisms like Neural Network and Genetic Algorithms. These are understood subjects of computer science and provide a lot of help in making algorithms for pixel map of an image to recognition of a character. The data is divided into training data and testing data and from an image the written text is extracted and recognized. A comparison of accuracy on different mechanisms is obtained by existing tests by other authors and could be extended for usage in our scenario.

## 7. REFERENCES

- [1] M.Seetha, I.V.Muralikrishna, Member, IEEE B.L. Deekshatulu, Life Fellow Member, IEEE, B.L.Malleswari, Nagaratna, P.Hegde, "Artificial Neural Networks and other methods of image classification", GJournal of Theoretical and Applied Information Technology © 2005 - 2008 JATIT.
- [2] Fakulta matematiky, fyziky a informatiky, "Image Classification using Artificial Neural Networks", Univerzita Komenského v Bratislave
- [3] Vivek Shrivastava and Navdeep Sharma, "Artificial Neural Network based optical character recognition", Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.5, October 2012 DOI :

- [4] Md Fazlul Kader and Kaushik Deb, “Neural Network-Based English Alpha Numeric Character Recognition”, *International Journal of Computer Science, Engineering and Applications (IJCSEA)* Vol.2, No.4, August 2012
- [5] Ankit Sharma, Dipti R Chaudhary, “Character Recognition Using Neural Network”, *International Journal of Engineering Trends and Technology (IJETT)* - Volume 4 Issue 4 - April 2013
- [6] Pulipati Annapurna, Sriraman Kothuri, Srikanth Lukka, “Digit Recognition Using Freeman Chain Code”, *International Journal of Application or Innovation in Engineering & Management (IJAIEM)* Volume 2, Issue 8, August 2013 ISSN 2319 - 4847
- [7] Nisha Vasudeva, Hem Jyotsana Parashar and Singh Vijendra, “Offline Character Recognition System Using Artificial Neural Network”, *International Journal of Machine Learning and Computing*, Vol. 2, No. 4, August 2012
- [8] Sameeksha Barve, “Optical Character Recognition Using Artificial Neural Network”, *International Journal of Advanced Technology & Engineering Research (IJATER)*
- [9] Sandeep Saha, Nabarang Paul, Sayam Kumar Das, Sandip Kundu, “Optical Character Recognition using 40-point Feature Extraction and Artificial Neural Network”, *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 4, April 2013
- [10] K.Venkata Reddy, D.Rajeswara Rao, U.Ankaiah, K.Rajesh, “Handwritten Character and Digit Recognition Using Artificial Neural Networks”, *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 4, April 2013
- [11] Vipin, Rajeshwar Dass, Rajni, “Character Recognition using Neural Network”, *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 2, No.3, May - June 2013
- [12] Chirag I Patel, Ripal Patel, Palak Patel, “Handwritten Character Recognition using Neural Network”, *International Journal of Scientific & Engineering Research* Volume 2, Issue 5, May-2011
- [13] Rahul Kala, Harsh Vazirani, Anupam Shukla and Ritu Tiwari, “Offline Handwriting Recognition using Genetic Algorithm” *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 2, No 1, March 2010.
- [14] Vedgupt Saraf, D.S. Rao, “Devnagari Script Character Recognition Using Genetic Algorithm for Get Better Efficiency”, *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-2, Issue-4, April 2013
- [15] Majida Ali Abed, Ahmad Nasser Ismail and Zubadi Matiz Hazi, “Pattern recognition Using Genetic Algorithm”, *International Journal of Computer and Electrical Engineering*, Vol. 2, No. 3, June, 2010
- [16] Chomtip Pornpanomchai, Verachad Wongsawangtham, Satheanpong Jeungudomporn, and Nannaphat Chatumpun, “Thai Handwritten Character Recognition by Genetic Algorithm (THCRGA)”, *IACSIT International Journal of Engineering and Technology*, Vol.3, No.2, April 2011