Implementation of Real-time Handwriting Recognition System Using Touch Panel Based on Neural Network

Yi-Sung Yang [1], Cheng-Fang Huang [1], Bo-Jhih Hu [1], Teh-Lu Liao [1], Jun-Juh Yan [2,*]

[1.] Department of Engineering Science, National Cheng Kung University, Tainan 701 Taiwan, R.O.C.
[2.] Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan, R.O.C.
* Corresponding author: Tel: +886-7-6158000 #4806; Fax: +886-7-6158000 #4899
E-mail: jjyan@stu.edu.tw (corresponding author)

Abstract: Based on neural network, this study contributes to propose a real-time handwriting recognition system with Arabic numbers and lowercase letters. It includes two parts which are hardware design and software algorithm. In hardware design, after pressing the touch panel surface, analog signals are obtained and transformed into digital ones by A/D converter. In software algorithm, recognition architecture is constructed by three level back-propagation neural network and learning samples of Arabic numbers and lowercase letters are collected from nine schoolmates. Based on the illustration, the proposed handwriting recognition system of this study can achieve about 90% correction rates and satisfy the market standard.

[Yi-Sung Yang, Cheng-Fang Huang, Bo-Jhih Hu, Teh-Lu Liao, Jun-Juh Yan. **Implementation of Real-time Handwriting Recognition System Using Touch Panel Based on Neural Network.** *Life Sci J* 2012;9(3):148-154] (ISSN:1097-8135). http://www.lifesciencesite.com. 20

Keywords: Touch panel, Handwriting recognition, Back-propagation neural network

1. Introduction

Handwriting is a natural part of the daily lives of human beings. In the past, people have only been able to interact with a computer by typing on a keyboard or using a mouse. When people are unfamiliar with a method by which to input language, they must spend time studying and practicing. As technology has developed, the innovativeness of touch products has gradually improved communication between humans and machines (Rehman and Saba, 2012a). It would be convenient if we could control machines by only using a touch panel instead of a keyboard or mouse. The advantages of handwriting recognition include the direct inputting of characters and without having to learn an input method. Touch panel can form an important bridge between people whose learning ability is poor and the machines they are trying to operate.

Neural networks have been used in most of research applications such as classifications, automatic control, estimation, signal processing, recognition, etc. (Ramadan et al., 2012; Rehman and Saba, 2012b; Yousefi et al., 2008; Gao et al., 2010). Compared with developing conventional expert systems, the main advantage is that neural networks often can be quickly constructed using available data at a very low cost. In recent years, these artificial intelligence techniques have also been successfully applied in the area of handwriting recognition (Zhang and Wu, 2010; Saba and Rehman,2012). Generally speaking, handwriting recognition can be divided into two approaches: on-line and off-line. In on-line technology (Sulong et al., 2010; Rehman et al., 2011), information is acquired during the writing process through the use of a tablet and touch pen. In off-line technology (Chen and Wei, 2010; Rehman and Saba, 2011a,b; Kurniawan et al., 2009; Dilruba et al., 2006; Tay et al., 2001; Saba et al., 2011). The text data obtained by a scanner after the writing process is over. Therefore, the recognition technique for on-line handwriting is more complex than is the case with off-line situation. Because everyone has different writing habits and because noise usually exists in the acquisition process, these factors cause the recognition process to occur after acquisition (Rehman and Saba, 2012c).

In this study, an on-line recognition technique is adopted, and a method by which to develop the recognition system is viewed as a very challenging issue. We use a touch panel to communicate with a machine. After determining the coordinate handwriting location, we utilize in the recognition process to obtain more accurate coordinates and extract feature by obtain stroke characteristic. Finally, the handwriting recognition system uses an artificial neural network to obtain a recognition result. Furthermore, determining how to implement an on-line handwriting recognition system utilizing is the primary goal. The target of the proposed system is to identify handwriting in various font sizes and to create a system that can be utilized by anyone.

2. Material and Methods

Overview of the system architecture:

There are four main structures in the system architecture, a four-wire resistive touch panel, an A/D converter, a Micro-Control Unit (MCU) development board and a PC. The system architecture diagram is shown in Figure 1. Descriptions of these four main structures are as follows:

i. Four-wire resistive touch panel: This is the handwriting input device with a two dimensional plane. By touching the surface with the finger or a touch pen, the touch panel controller can calculate the x-axis and y-axis coordinates using the KVL principle.

ii. A/D converter: Because the data for the touch coordinates is an analog signal, an A/D converter is used to transform the position signal into a digital signal. In this study, an ADS7846 chip is used for the data sampling.

iii. Micro-Control Unit (MCU): FS7805 GPIO ports are utilized to deliver and read measurement data from an ADS7846 chip. After analyses of the operation, the MCU sends the final data to the PC through the RS232.

iv. PC software: Visual C++ 2008 software is used to implement the handwriting recognition and to design the graphic user interface (GUI). The firmware of this system is coded by keilC software.

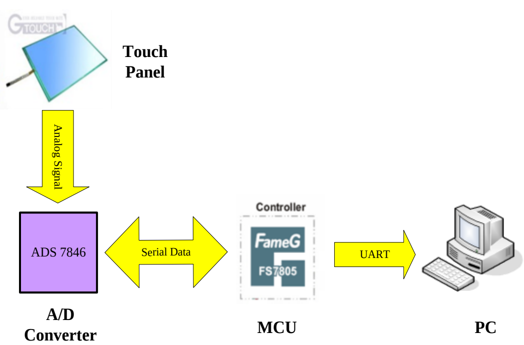

Figure 1. The system architecture diagram.

Hardware circuit design:

In the design of the circuit interface, the touch panel's X+, X-, Y+, Y- pins are respectively connected to the ADS7846, and the DCLK, DOUT, DIN, CS, BUSY pins are respectively combined with the FS7805 GPIO port. The schematic diagram of the related control pins is shown in Figure 2. The first transmission between the FS7805 and the ADS7846 is a send control byte on the DIN including the start bit, channel selection, 8/12bit mode, differential/single-ended and power mode. The FS7805 sends the corresponding control byte through the GPIO interface, and the BUSY pin is used to detect the ADS7846 converter status. Then, the FS7805 reads the conversion results from the ADS7846 and obtains coordinates data and the ADS7846 setting configurations. After the above processes, the FS7805 and ADS7846 communication is finished.

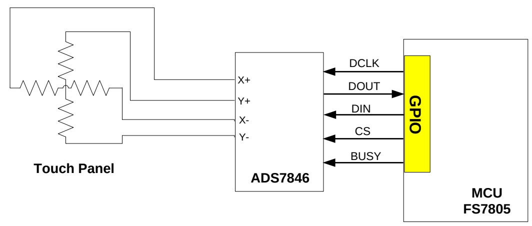

Figure 2. Schematic of connecting interface.

ADC timing diagram:

In this study, the timing diagram of the ADS7846 controller is considered (Brown 2005). The CS# is the chip-selected signal. Because this study uses a 12 bit resolution, the sampling data is 0~4095. When the CS# converts from a high voltage level to low voltage level, this means that the controller begins to operate. On the other hand, it also expresses the end of the operation. This system uses 24 clock cycles to represent an operational process. The first eight clock cycles are used to receive commands from the FS7805. At this moment, the DIN (called the "control byte") is set to capture the address coordinates. The control byte and the configuration of differential mode input are shown in Table 1 and 2. At that point, when BUSY signals are moving from high voltage level to low voltage level, the other sixteen clock cycles are started to send commands to the FS7805. Then, the DOUT of the ADS7846 begins to return the coordinate value (called the serial data output) to the FS7805. Finally, the process of the system measurement is completed. The timing diagram is shown in Figure 3.

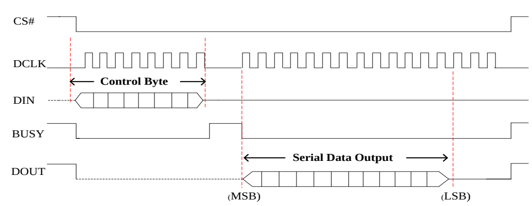

Figure 3. ADS7846 conversion timing diagram.

Table 1. Control byte of ADS7846.

Bit map	Name
Bit7(MSB)	S
Bit6	A2
Bit5	A1
Bit4	A0
Bit3	MODE
Bit2	$SER/\overline{DFR}$
Bit1	PD1

Bit0(LSB)	PD0

Table 2. Configuration of differential mode input.

A2	1	0
A1	0	0
A0	1	1
Y-		
X+		+IN
Y+	+IN	
X-Position	Measure	
Y-osition		Measure
X-Drivers	On	Off
Y-rivers	Off	On

Firmware Coding on the MCU:

All types of touch panels must contain pressure detection on their input surfaces, X coordinate measuring, Y coordinate measuring, and repeated detection. At the beginning of the programming design, this system must do the MCU initialization. After touching the touch panel, the ADS7846 pen-interrupt signal starts. The MCU generates a timing sequence to the ADS7846 and sends a control-byte to measure the X and Y coordinates. Therefore, the touch panel firmware design is focused on the control-byte settings. Figure 4 shows the FS7805 firmware flowchart.

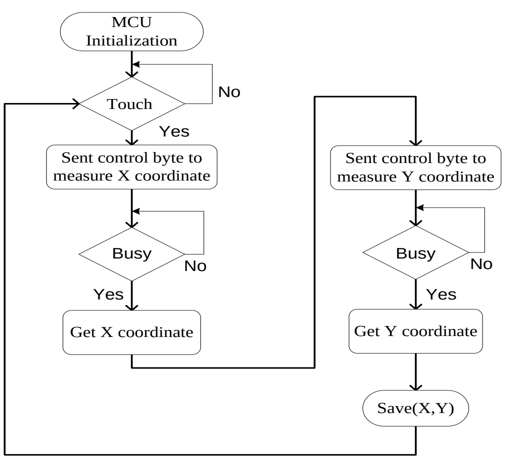

Figure 4. Flowchart of firmware design.

Algorithm of handwriting recognition:

Before handwriting recognition occurs, a great deal of data and information about the coordinates obtained from the touch panel must be set up and transmitted to the PC processer through the RS232. The handwriting recognition algorithm can be divided into several steps that include the coordinate acquisition, pre-processing, feature extraction and the neural network module. Figure 5 shows the software schematic design.

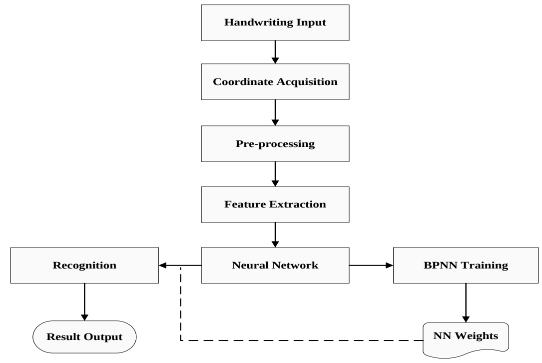

Figure 5. Flowchart of software design.

Coordinate Acquisition:

In the previous parts, the MCU has been processed to control the method by which coordinate formation from the touch panel is received. The formula is shown as follows:

$$X = 2^{11} \cdot x_{11} + 2^{10} \cdot x_{10} + \cdots + 2^{0} \cdot x_{0}$$
$$Y = 2^{11} \cdot y_{11} + 2^{10} \cdot y_{10} + \cdots + 2^{0} \cdot y_{0} \qquad (1)$$

where X、Y is the coordinate value after quantification. $x_0 \sim x_{11}$ and $y_0 \sim y_{11}$ are defined as the returned coordinate data from touch panel.

Pre-processing:

The system uses a 4-wire resistive touch panel with a voltage divider to obtain the voltage value and to convert it to a digital value using an A/D converter. When a finger or stylus touches the panel, this might cause the voltage divider to be incorrect for the location and the calculation may obtain a neighborhood location. Therefore, the noise filter gives a threshold if the coordinate is too close to eliminated. If the difference between the current coordinate information and the next one is greater than 5, the information is recorded, or the filter automatically considers it to be meaningless jitter. Through this approach, the noise jitter can be removed.

Feature Extraction:

This study uses two main steps: First, users write a character, and the system begins to normalize. This step can conform the range of the original character to the optimally judged process range. Regardless of whether the sizes of the characters are big and small, the optimum area can be judged effectively. After obtaining the judged range, the maximum X coordinate, maximum Y coordinate, minimum X coordinate and minimum Y coordinate can be picked out. Then, the system can extract the features of the determinant handwriting region. Secondly, because the writing sequences from everybody are different, this study is aimed at identifying the character correctly while neglecting the writing sequences. By holding the maximum and minimum coordinates, the regions of Arabic numbers and lowercase alphabet letters are separately divided into grids measuring 8×6 and 8×8. When the segment of a character is through the grid, the system records "1" from the present grid; otherwise, it records "0". Figure 6 shows that the feature extraction and the features of the Arabic number "3" are recorded as 000000; 011110; 000010; 011110; 000010; 000010; 011110, and 000000. The features of the Lower-case alphabet letter "b" are recorded as 01000000; 01000000; 01100000; 00111110; 00100010; 01100010; 01101110, and 00110000.

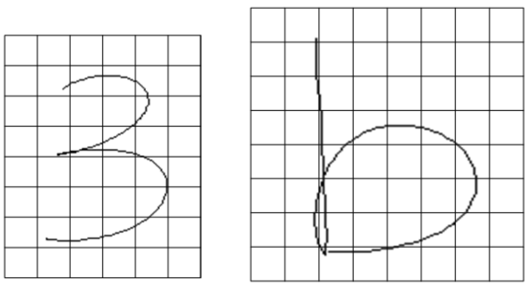

Figure 6. Two examples of the feature extraction.

BPNN training algorithm

The back-propagation algorithm uses a method of gradient descent that can optimize the weights of the neural network. It is utilized iteratively to adjust weights, which can achieve the minimum error between the input and target values, and the ANN will finally be convergent. The complete flow of learning is shown as follows, and Figure 7 shows the flow chart for the learning model.

Step 1: The numbers for the ANN, which contains the input, hidden, and output layer numbers of the neuron are chosen first. Because the BPNN belongs to a supervised learning network, it is needed to provide the sample numbers and to target into the ANN.

Step 2: The initialized weights that adopt the random hypothesis are set up.

Step 3: The patterns of learning, which include the input and target dimensions are inputted.

Step 4: The ANN output values, which include the neurons from both the hidden and output layers, are calculated.

Step 5: The magnitudes of error, which include the neurons from both the hidden and output layers, are calculated. This key point returns the output values from the output layers into the neuron of the hidden layers, which can adjust the magnitudes of error and obtain new weights for the ANN.

Step 6: All weights, including the hidden and output layers of the ANN, are adjusted.

Step 7: If the error achieves the standard of convergence, the learning process is stopped and the weights are saved, or it repeats again from step 3.

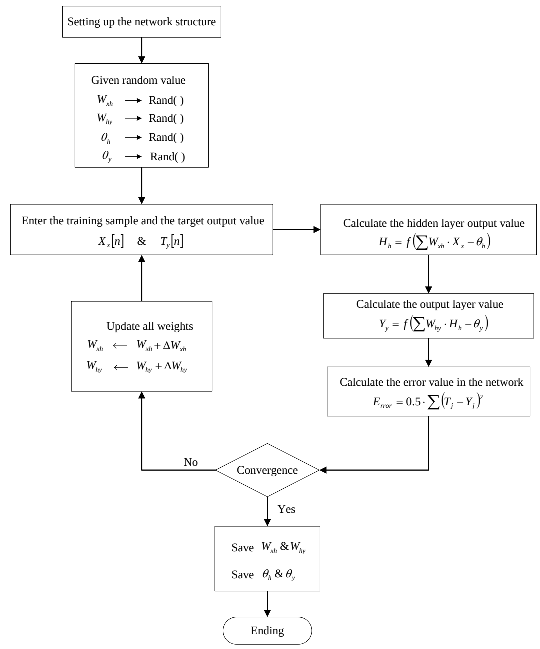

Figure 7. Flowchart of BPNN learning process.

BPNN structure design:

This study uses a three-level layer back-propagation neural network, which includes input layers, hidden layers, and output layers as the framework for the neural network.

i. Design of the input and output layers: Due to treating the samples differently, the numbers of input neurons are changed from the size of the input data. In this study, the Arabic numbers mode and lowercase alphabetic mode are set at 48 and 64, respectively. The output layer is used to describe the results of the network output. And then, the numbers of the output layer neurons in the Arabic number mode and the lowercase alphabetic mode are set at 10 and 26, respectively.

ii. Design of the hidden layer: This layer is used to describe the neuron interaction between the input and output layers. Number of layer numerous is very important. If the numbers of layer numerous are set suitably, the optimal weights can be found easily, and the error function can converge better. The optimized formula is shown as the following equation:

$$h = \sqrt{x \times y} + c \qquad (2)$$

where x is the number of input layer neurons, y is the number of output layers neurons; h is the number of hidden layers neurons, and c is chosen to be a number between 1 to 5. In this study, the numbers of hidden layer neurons in the Arabic number mode and the lowercase alphabetic mode are set at 18 and 230, respectively.

iii. The determination of activation functions:

Sigmoid function $f(x) = 1/(1+\exp(-ax))$, $a > 0$ can be used as the activation function, where a is the slope parameter of the sigmoid function, and by changing the value a, the activation function with a different slope can be obtained. Sigmoid function is shown in Figure 8.

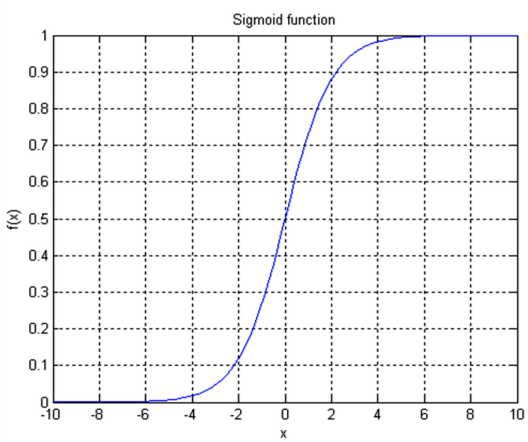

Figure 8. Sigmoid function.

3. Results

Illustration and Testing:

Before testing the system must collect training samples that include Arabic numbers and lowercase letters. The handwriting samples are written randomly on the touch panel. Based on the training intended to adjust the BPNN weights, the convergence of the mean square error (MSE) can be reached (Choudhary et al. 2010).

Illustration in Arabic Numbers:

In part of the Arabic numbers training, complete stroke of the feature extracting contains 48 parameters, which set 48 neurons in the input layer and 9 neurons in the output layer representing Arabic number (0~9). We adjust the neuron numbers in the hidden layer and discuss the numbers in the hidden layer. All parameters for the BPNN are shown in Table 3, which follows. Figure 9 compares the numbers in the hidden layer neuron of each MSE convergence situation. After trial and error, the BPNN have convergence below 0.01 via the neuron amount 18 from the hidden layer.

Table 3. Neural network structure of Arabic part.

Parameters	values
Input neurons	48
Hidden neurons	18
Output neurons	10
Learning rate	0.5
Momentum Term	0.5
Activation function	Sigmoid function
Initial weights and biased values	Randomly set values between 0 and 1

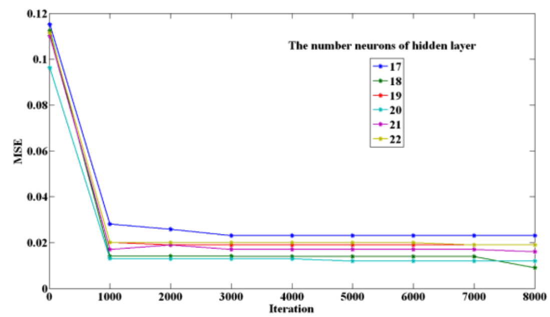

Figure 9. Schematic of MSE.

In this part, the training samples are written with unlimited strokes on the touch panel, which writes the Arabic number (0~9) 1000 times. After training, we ask nine people to write an Arabic number 100 times. Table 4 shows the recognition results with candidate situation of Arabic number can reach 92%. The recognition index was defined as follows:

$$\text{Recognition Rate} = \frac{\text{The number of the correct recognition}}{\text{Amount of the samples}} \times 100\% \qquad (3)$$

Table 4. Testing results of Arabic number.

Arabic number	Recognition rate	Arabic number	Recognition rate
0	97.5	6	95
1	97.5	7	92.5
2	92.5	8	85
3	92.5	9	98.75
4	93.75	Average	92.6
5	81.25		

Illustration in Lowercase Letters:

In the part consisting of training in the lowercase letters, a complete stroke of the feature extraction contains 64 parameters, which set 64 neurons in the input layer and 26 neurons in the output layer representing lowercase letters (a~z). In this section, we adjust the neuron numbers in the hidden layer and discuss the numbers in the hidden layer. All parameters of the BPNN are shown in Table 5, which follows. Because training in lowercase letters have a high level of complexity, we use the decrease in the learning rate to substitute for the increase in the hidden layer in order to obtain good convergence. Improper selection of a learning rate may cause a local minimum problem or large training runs, and as a result, can decrease ANN performance (Zhang and Wu, 2010). Figure 10 compares the numbers of the hidden layer neurons with each MSE convergence situation.

Table 5. Neural network structure of letters part.

Parameters	Values
Input neurons	64
Hidden neurons	300
Output neurons	26
Learning rate	0.2
Momentum Term	0.2
Activation function	Sigmoid function
Initial weights and biased values	Randomly set values between 0 and 1

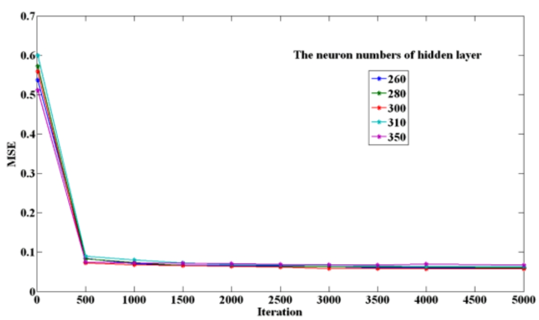

Figure 10. Schematic of MSE.

In this section, the training samples are written with unlimited strokes on the touch panel, which writes the lowercase alphabet (a~z) 2600 times. After training, we ask nine people to write 130 times. Table 6 shows the recognition result with candidate situation can reach 93%.

Table 6. Testing results of lowercase alphabet.

Alphabet	Recognition rate	Alphabet	Recognition rate
a	94.3	o	97.1
b	94.3	p	94.3
c	94.3	q	88.6
d	97.1	r	97.1
e	86.4	s	94.3
f	94.3	t	94.3
g	94.3	u	94.3
h	88.6	v	97.1
i	83	w	97.1
j	97.1	x	97.1
k	97.1	y	83.5
l	94.3	z	94.3
m	85.7	Average	93.2
n	94.3		

4. Discussions

In this experiment, we consider handwriting strokes that are simple or one directional to complete to have a high recognition rate. Poor handwriting trajectory, it may cause incorrect results. Therefore, the system compares the situation with join candidate word condition. The recognition results can be more than 90%. Figure 11 shows the comparison of the experimental results.

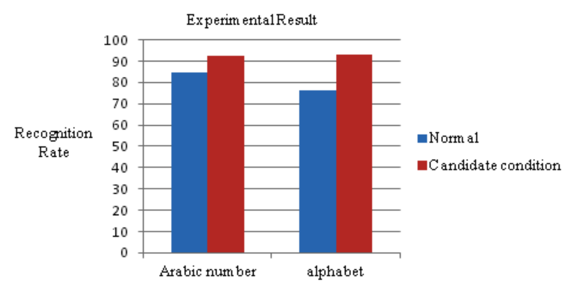

Figure 11. Comparison of illustration.

5. Conclusion

In this study, an on-line handwriting recognition system was implemented by using the artificial neural network. The proposed system includes hardware design and software algorithm. In hardware design, analog signals from the touch panel surface are transformed into digital ones by A/D converter. In software algorithm, recognition architecture is constructed by three level back-propagation neural network. Based on the illustration, the proposed handwriting recognition system of this study can achieve about 90% correction rates and satisfy the market standard.

References

1. Ramadan ABM, El-Garhy AM, Amer FZ, Hefnawi MM. Forecasting gamma radiation levels using digital image processing. Life Science Journal 2012;9(1):701-10.
2. A Rehman, T Saba (2011a) Document skew estimation and correction: analysis of techniques, common problems and possible solutions

3. Applied Artificial Intelligence vol. 25 (9), pp:769-787 .
4. Rehman, A. and Saba, T. (2012a) Neural Network for Document Image Preprocessing" Artificial Intelligence Review, DOI: 10.1007/s10462-012-9337-z.
5. Yousefi H, Hirvonen M, Handroos H, Soleymani A. Application of neural network in suppressing mechanical vibration of a permanent magnet linear motor. Control Engineering Practice 2008;16(7):787-97.
6. Gao Y, Deng C, Jiang G. Application of BP neural network in the digital recognition system. Computer Engineering and Technology (ICCET), 2010 2nd International Conference on 2010;6: 522-6.
7. Rehman, A. Kurniawan, F. and Saba, T. (2011) "An Automatic Approach for Line detection and Removal without Characters Smash-up". Imaging Science Journal, vol. 59(3), pp. 171-182.
8. Zhang X, Wu L. Handwritten Digit Recognition Based on Improved Learning Rate BP Algorithm. Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on 2010:1-4.
9. Sulong, G. Saba, T. and Rehman, A. (2010). Dynamic Programming Based Hybrid Strategy for Offline Cursive Script Recognition. 2nd IEEE International Conference on Computer and Engineering, vol. 2, pp. 580-584.
10. Chen Z, Wei Z. A Handwriting Numeral Character Recognition System. Multimedia Technology (ICMT), 2010 International Conference on 2010:1-5.
11. A. Rehman, and T. Saba (2011b). Performance Analysis of Segmentation Approach for Cursive Handwritten Word Recognition on Benchmark Database. Digital Signal Processing, vol. 21(3), pp. 486-490.
12. Kurniawan, F. Rehman, A. and Dzulkifli Mohamad (2009). "Contour Vs Non-Contour based Word Segmentation from Handwritten Text Lines. An Experimental Analysis". International Journal of Digital Content Technology and its Applications, vol. 3(2), pp. 127-131.
13. Brown B. Touch Screen Controller-ADS7846 Datasheet. Texas Instruments; 2005.
14. Choudhary A, Rishi R, Ahlawat S, Dhaka VS. Performance analysis of feed forward MLP with various activation functions for handwritten numerals recognition. Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on 2010;5: 852-6.
15. Rehman, A. and Saba, T. (2012b). Off-line Cursive Script Recognition: Current Advances, Comparisons and Remaining Problems, Artificial Intelligence Review, vol. 37(4), pp: 261-268.
16. Saba, T. Rehman,A. and Elarbi-Boudihir, M. (2011). "Methods and Strategies on off-line Cursive Touched Characters Segmentation: A Directional Review". Artificial Intelligence Review, DOI 10.1007/s10462-011-9271-5.pp:45-54.
17. T. Saba, and A. Rehman (2012), Machine Learning and Script Recognition, Lambert Academic Publisher, ISBN-10: 3659111708, pp: 98-110.

5/21/2012