I.J. Information Technology and Computer Science, **2017, 9, 11-17**
Published Online September 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2017.09.02

Offline Handwriting Recognition Using Feedforward Neural Network

Rosalina
President University, Faculty of Computing, Bekasi, Indonesia
E-mail: rosalina@president.ac.id

R.B. Wahyu
President University, Faculty of Computing, Bekasi, Indonesia
E-mail: rbw0101@gmail.com

Received: 30 March 2017; Accepted: 08 June 2017; Published: 08 September 2017

***Abstract*—**Many business especially Banks's services are expanding to include services directed not only to corporate customers but also to individual customer. Furthermore, by the increment of those services, many individual applications to be processed also increases as well. Facing an immense moment, in which requiring more improvements in how it should manage or maintain its applications, some systems or procedures must be improved to match currently increasing customers' applications. Prior to the improvements, many application forms are filled, input to machine and even to be processed and approved manually. Until recently, application fulfillment processes consists of manual information filling by applicants in an application request paper and later to be re-input by electronic data processing staff which is actually redundant. Aware of such situation, this paper proposes and idea to reduce input processes in an integrated business system by utilizing character recognition system.

***Index Terms*—**Handwriting recognition, feedforward neural netwok, form.

I. Introduction

Technology's fast and extensive growth has affected many businesses in which forcing most to cope worth it and thus hatch needs to condense their business process to be more efficient. The importance of technology growth has already been an aspect for banking business in which must always survive the challenge. Faster, reliable and efficient performances by banks are expected by each and every customer. Furthermore, software enhancement and hardware upgrades are necessary in order to keep the fast or even real-time business activities alive. Following the degree of the processes' dependencies, processing time becomes a little bit more sensitive issue in the process since number of forms to be processed may increase over time. Driven by this information high quality business process efficiency must be met at all cost.

To achieve this ideal, utilizing machine may become better solution since human is error-prone. In addition to the solution, information filled in forms by customers can be digitized, configured, validated and to be approved electronically in a better timing and a better accuracy.

Digitizing manual filled forms lead to handwriting recognition, a process of translating handwriting into machine editable text. Challenges in handwritten signatures recognition lie in the variation and distortion of handwritten signatures, since different people may use different style of handwriting and direction to draw the same shape of any character (Hindi Characters:[2,11,9]), English Characters: [6, 7, 14]).

The handwriting recognition itself in general classified into two types as offline and online handwriting recognition methods. An offline recognition involves automatic conversion of text obtained in image or scanned materials, in contrast, online recognition involves real-time conversion as the data being input through digitizers like the one embedded in PDA's (personal desktop assistance) touch screen. By utilizing handwriting recognition, digitized information can be configured more flexibly and at the same time, saves the time used to process the application.

Neural network for offline recognition of handwritten been implemented in the past research [1,3,5,13,16] and result of the experimental in isolated digits shows that conventional features with back propagation network yields good classification accuracy of 100% and recognition accuracy of 91.2%[13].

A. Objective

The objective of this work is to design a system for recognition of handwritten capital letters on a given image of a fixed form. Afterward, success rate measurements will be calculated based on the recognition results and the accuracy in recognizing letters.

B. Scope and Limitation

This work focuses only on handwritten capital letters recognition in a fixed form which is provided beforehand. The fixed form will be subject to coordinate measurements so that it can be correctly used in down-sampling process. In addition, the algorithm applied in

originally from feed forward neural network algorithm.

II. Methodology

Stages involved in offline handwritten character recognition (Fig. 1) which are divided into Image acquisition, Preprocessing, Feature Extraction, Classification and Recognition.

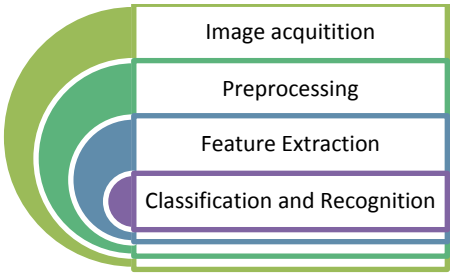

Fig.1. Offline handwritten character recognition stages

A. Image Acquisition

In this work, the input images is obtained through a scanner. The image should have specific format such as JPEG; BMT etc. The input captured may be in gray, color or binary form scanner.

B. Preprocessing

The aim of the preprocessing stage is to enhance the quality of recognition. In this stage the output of the image acquisition is fed as input to the processing stage then will be applied number of procedures for thresholding, smoothing, filtering, resizing, and normalizing so that successive algorithm to final classification can be made simple and more accurate[4].

C. Feature Extraction

In this work, the feature extraction applied boundary detection of a handwritten character and eight neighbor adjacent methods has been adopted as shown in Fig.2.

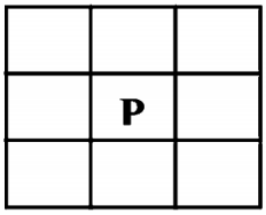

Fig.2. Eight neighbor adjacent of pixel

The binary image is scan until the boundary is founded in clockwise direction, once a white pixel is detected, it checks another new white pixel and so on. If the first pixel is found, the system will assigned the coordinates of that position to indicate that this is an origin of the boundary. During the boundary tracing process, the system will always check the condition whether the first coordinates of the boundary are equal to the last coordinates. Once it is obtained; means the whole boundary has been traced and boundary tracing process completes [13].

Note that the way to handle image file and text file is different because image file to be processed is not to be taken by simple file stream which is generally used in handling text file but rather considering image pixels and the information inside those pixels. To crop an image from the scanned form, there is one thing that should be set beforehand which is the coordinate numbers. These numbers represent the exact positioning of the important information in the image in Cartesian coordinate. Since the setting is different between forms because not all forms have the same structure and positions, the coordinates setting must be done beforehand. For a clearer look in the coordinate numbers, Fig.3. shows static setting while for testing form shown in Fig.4.

```
public void setCoordinate(){
        coordinateList.add("8,11"); coordinateList.add("219,12");
        coordinateList.add("431,13"); coordinateList.add("646,14");
        coordinateList.add("863,14"); coordinateList.add("10,157");
        coordinateList.add("221,156"); coordinateList.add("435,157");
        coordinateList.add("649,157"); coordinateList.add("864,158");
        coordinateList.add("10,303"); coordinateList.add("221,303");
        coordinateList.add("435,303"); coordinateList.add("650,302");
        coordinateList.add("864,302"); coordinateList.add("10,449");
        coordinateList.add("222,450"); coordinateList.add("436,451");
        coordinateList.add("651,450"); coordinateList.add("865,450");
    }
```

Fig.3. Coordinate Number

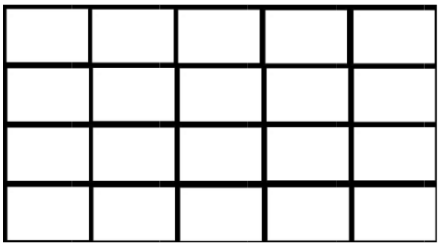

Fig.4. Sample Testing Form

After images containing important information is successfully cropped from the scanned image, all images preparations have been completed. After the image reading is fixed, the next thing to do is to set data based on the pixels read from the corresponding image. The first thing to do in searching pixels in the image is bounding the image so that minimum whitespaces are read (for faster pixels reading performance). If the pixels contain information, in this case has districtive value other than whitespaces, a Boolean value: true will be assigned to it, otherwise, false will be assigned to it. The Boolean value assigment will be very precise since it also uses Cartesian coordinae location to determine the exact position to set the value. After Boolean value setting has been completed then more conversion must be done based on following rule:

a. If the Boolean value assgined in the position [x,y] is false, then assign the value of 0.0 to the exact given position.
b. If the Boolean value assigned in the position [x,y] is true, then assign the value of 1.0 to the exact given position

c. If the value to be assigned is not stated in the given position, assume them to be whitespaces (ignoring possible noise) and assign the value of 0.0 to them.

III. System Analysis and Design

A. Mathematical Operations

As Fig.5 may suggest, there are at least three layers of the network with an input layer as the first layer, an output layer as the last layer and a hidden layer as the medium. In the first layer, a set of input is supplied to the model; in this case, a set of English capital letters is supplied. In purpose of mathematical operations to be applied, the input set is constructed in such a way so that it forms an array of mathematical unit, e.g.: integer, double, float, big decimal, etc. In contrast to the general form of Feedforward neural network, the design of the model has a slight change in the input layer. Instead of having one value assigned into each input neuron, the author supplies a new standalone Feedforward neural network model. In this way, each character representation will be calculated in its own neural network beforehand so that the general model can have many advanced inputs which give the model some advantages such as more exact input representation and faster time in general error calculations.

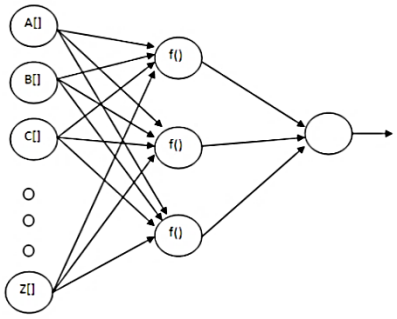

Fig.5. System's feedforward neural network model

Furthermore, array of double representing a character can be seen in Fig.6: election of maximum number in the given double array will show a significant distinguishable character as shown in Fig.7.

```
public static double[][] inputLetters_ = new double[][]{
    new double[][//A
                0.0, 0.0, 1.0, 1.0, 0.0,
                0.0, 0.0, 1.0, 1.0, 0.0,
                0.0, 1.0, 1.0, 1.0, 0.0,
                0.0, 1.0, 0.0, 1.0, 0.0,
                1.0, 1.0, 1.0, 1.0, 1.0,
                1.0, 1.0, 0.0, 0.0, 1.0,
                1.0, 0.0, 0.0, 0.0, 1.0
            },
```

Fig.6. Character in double array

```
public static double[][] inputLetters_ = new double[][]{
    new double[][//A
                0.0, 0.0, 1.0, 1.0, 0.0,
                0.0, 0.0, 1.0, 1.0, 0.0,
                0.0, 1.0, 1.0, 1.0, 0.0,
                0.0, 1.0, 0.0, 1.0, 0.0,
                1.0, 1.0, 1.0, 1.0, 1.0,
                1.0, 1.0, 0.0, 0.0, 1.0,
                1.0, 0.0, 0.0, 0.0, 1.0
            },

    new double[][//B
                1.0, 1.0, 1.0, 1.0, 1.0,
                1.0, 0.0, 0.0, 0.0, 1.0,
                1.0, 0.0, 0.0, 0.0, 1.0,
                1.0, 1.0, 1.0, 1.0, 1.0,
                1.0, 0.0, 0.0, 1.0, 1.0,
                1.0, 0.0, 0.0, 0.0, 1.0,
                1.0, 1.0, 1.0, 1.0, 1.0
            },
```

Fig.7. Selected maximum number in double array representation

The array of doubles act as input set representing English capital letters supplied through input layer. Afterward, the input will go to the next layer: hidden layer. In hidden layer, mathematical calculation will be applied to the input. Each input's dot product which formula shown in equation 1 is the result for the mathematical calculation in prior description.

$$a.b = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots a_n b_n \qquad (1)$$

Before all computations are finished, the operations in hidden layer will still be running. Furthermore, proceeding to output layer in the calculation step, the output layer will try to compare between the result of dot product and the desired output. The gap between the two objects is recognized as error signal. More concrete representation of the calculation is shown in Fig.8.

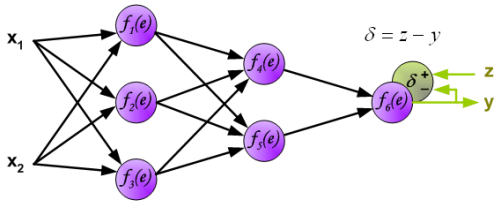

Fig.8. Calculations Involved [8]

Prior to the gap calculation result by subtracting those objects, a root mean square (RMS) error is also calculated. The purpose of calculating an RMS error is to identify how far a value possibly differs from the true value (ideal value). There is no exact value acting as the gap between true value and calculation value but only an estimation approach.

The RMS error rate will then be remembered, and updated if there is a smaller number result from the next calculation/iteration. The purpose of getting the error is to know which calculation results in minimum error. Later, the minimum error will be referred as local minimum since it represents the smallest error number in one calculation covered in a model's iteration. In general, the formula to get the RMS error rate is shown in equation 2.

$$x_{rms} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots x_n^2}{n}} \quad (2)$$

Preceding the calculation, the next step after the gap is calculated, as the learning technique's name may suggest, the result is then propagated back to the previous layer as a new coefficient for the next calculation. The backpropagation flow is iterated until the calculation reaches the minimum error level suggested prior to any calculatios started, e.g.: terminate after error rate is below 7%. The backpropagation flow is described briefly in Fig.9.

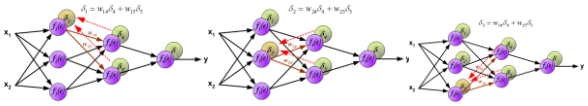

Fig.9. Backpropagation Flow [8]

The propagated value sent tot the previous neuron/layer will be used as a new weight value, thus weighting some percentage for the calculation in order to converge. The calculation will terminate itself after the certain level of error rate or threshold is met. Upon finishing the calculations, the network will know which calculations having the least local error to be mapped to the output.

In short, the system enables the users to recognize handwritten text on a scanned application form. Prior to any recognizing operation, the system must be trained so that it can gather the initial classification of probable input set and the desired output based on the calculations applied to the input.

B. System Overview

The development of handwriting recognition system is fused with neural network implementation which enables the system to recognize the input more accurately since the network allows alternate version of input. In addition, the combination may hint that the system will not work in Boolean manner in which only true or false response given in every recognizing attempt. Instead, the system can work in somewhat fuzzy environment where noised or alternate input can be recognized. Furthermore, the working system is a combination of layers of neural network, simple image processing tool, and the classification module.

C. General Development

Neural network implemented in the system must be trained prior to any possible operations so that it can gather the initial mapping/classification of possible input set and desired output. Input data can be gathered using simple image processing tool such as image down-sampling for the input image or simply by gathering all images using third party image processing software. Afterward, the system will run mathematic calculations in order to determine which input belongs to what output based on the calculation result. The calculation itself is a form of local minimum convergence computation or in simpler word, a form of computation to find minimum error in each attempt (assuming many input is feed into the training set). As a result, the one having minimum error will be recognized as the probable desired output for the given input (the winning neuron at the iteration).

IV. System Architecture

In general, the system has three different tasks to perform in order to finish one cycle of recognizing process. In addition, as Fig.10 may suggest, the tasks are handled in different methods but each methods still stand in dependency to other method in order to finish the recognizing process mention earlier. For example, the system must be trained first before any other operations can run. Afterward, if the training result is satisfactory, then it can be reused in other operation in order to recognize the given input. Furthermore, the reusability of the training result is relative to its learning process which mean that there's a possibility to retrain the system in the next batch of recognizing process.

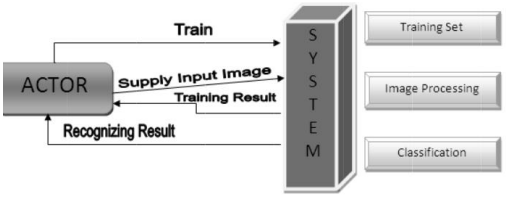

Fig.10. System Architecture

V. Testing and Results

There are three major modules to be developed in order to engineer the complete working system: Training module, Image module, and Files module. Some other complements such as application launcher and interfaces will be provided by the integrated development environment (IDE) code auto-generation.

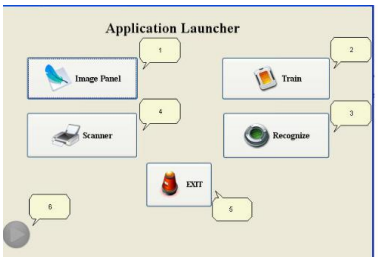

Fig.11. Image Supplied Recognizer

Training module covers the algorithm for neural network learning including mathematical calculations hidden beneath the training name. The training module is developed separately in one package inside the main project so that the reusability of the module is better. Instead of locating the mathematical calculations directly in the training module, some are extracted into one independent package. Following the calculations package,

there are some matrices operations that are directly referenced from third party code.

The initial user interface is shown in Fig.11. ImagePanel is used to invoke imaging utility required in the system such as image loader and image down-sampling.

Legend:
1.Button ImagePanel
2.Button Train
3.Button Recognize
4.Button Scanner
5.Button Exit
6.Training tatus

In case of simplicity, the image cropping based on the predefined frame coordinate can be invoked by only clicking anywhere in the internal frame with the image loaded prior to the action. On exit, the ImagePanel will be closed and return the user to the application launcher. Furthermore, the training can be started by clicking button Train, on training start, the system will ask the user to input the hidden layer value by showing an input pop-up dialog box. Having the network fully trained, the system can now start the recognition. Assuming that the input files for recognition have been residing in the predefined directory for input file, user may click the button Recognize to invoke recognition.

A. Performance Test

Performance test is unlike stress testing which require system robustness on heavy load. Performance testing covers a broad range of evaluations such as material, product, and system with specified specifications applied in a normal condition. In short, the purpose of the performance test is to test system's effectiveness in a normal condition and specified conditions. Applying five concurrent tests having seven hidden layers in the network model, the system performed as good as expected.

Based on the logs retrieved from the performance test, there are several points that can be drawn such as:

a) System performance depends on the likelihood of the mathematical calculations' convergence. If the calculations diverged, the performance of the system will take more time compared to the one which is converged.
b) With thread applied to the system, the process management becomes a lot better: no interference of heavy background processes to interface.
c) Different training and input method applied to the test will result in different system performance outcome.

The performance of the system on the test is rated as good as expected since there is a possibility of divergence in the background process's mathematical calculations. The divergence is expected since it is also the FeedforwardNetwork algorithm's disadvantage.

The case of divergence in calculations may happen if the network model is set in such a way so that previous calculations can affect the next calculations. In one hand, the affection of each calculation may lead to divergence; in the other hand, ideally this way should be fine if the calculation can affect each other and force the calculation to convergence state, provided that the momentum is correct.

B. Cross Test

In this system, the test will run on cross-image-sources such as:

- Training file set and the input set are provided using the same third party image processing tool (e.g.: Adobe Photoshop)
- Training file set is provided using third party image processing tool (e.g.: Adobe Photoshop) while input file set is gathered internally (utilizing image panel to crop and downsample the set)
- Training file set is provided using internal image panel while the input set is gathered externally (using third party image processing tool: Adobe Photoshop)
- Training file and input set file are provided using the same internal image panel

The purpose of conducting such test is to see whether the system can take the training and input file set and use them for recognition. Furthermore, different set for training and input file set may lead to negative result in recognition since there are many things to consider having the differences affecting the system.

a. Cross Test Case – 01

In this test environment, the images provided are all noise-free since the third party image processing tool has better control over noise that probably generated whenever the file is encoded in JPEG format. In addition, the noise may be generated if the tool has poor anti-aliasing handler.

In this test, the specifications of the image files are set in the following points:

- The training and input file set are all provided using third party image processing software (Adobe Photoshop)
- Twenty seven image files for training set: twenty six English capital letters with an addition of whitespace character
- Twenty image files of input file set for recognition. In case of simplicity, the recognition is limited to the given number.

b. Cross Test Case – 02

In this test environment, noises are expected in the training file or input file set since it is gathered using internal imaging panel with rather poor JPEG encoder. The specifications in this test are listed as follow:

- Training set is provided using third party image processing software (Adobe Photoshop) while input file set is gathered using internal image panel or vice versa.
- Twenty seven image files for training set: twenty six English capital letters with an addition of whitespace character.
- Twenty image files of input file set for recognition. In case of simplicity, the recognition is limited to the given number.

Fig.12. Input File Set

The training files are expected to be the basis for recognition and thus they should be seen as the ideal. It is not recommended to have noisy training files while the input file set is noise-free. If the training files provided are ideal, given that the input file set might be noisy, then there are several points listed below to be considered as possible way out for the system to be able to cope with noises in input file set:

- Applies better pixel
- Applies better JPEG encoder so that generated noise can be forced into minimal
- Applies better image processing routine so that it can be used to clear seen noises (need human interaction)
- Applies better down-sampling method so that only really distinguished pixel can be taken as input pixel, e.g.: check the color of the pixel, if it is not really black then it will be considered as possible noises
- Applies better recognition algorithm to read and parse probable image noises and exclude the noises from recognition

c. Cross Test Case – 03

In this test environment, noises are expected in the training file and input file set since it is gathered using internal imaging panel with rather poor JPEG encoder.

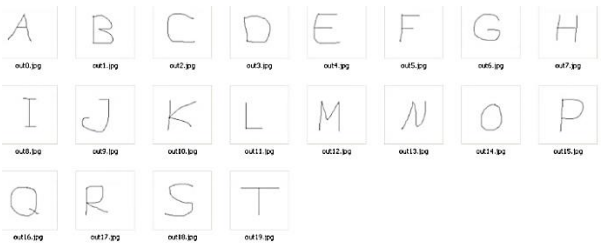

Fig.13. Case03 - Input File Set

The system is likely not being able to determine little differences between letter ‘G’ and ‘D’. A little error like this is expected since the training file and input file are noisy. The little difference between look-alike-characters should be taken as change-variable affecting recognition but in the case of noisy training and input file, the recognition result can be categorized as adequate.

d. Cross Test Case –4

In case of simplicity for this test environment, the images provided are all noise-free since they are all gathered using third part y image processing tool. The specifications of the image files are set in the following points:

- Two sets of the training and input file set are all provided using third party image processing software (Adobe Photoshop)
- Twenty seven image files for training set: twenty six English capital letters with an addition of whitespace character.
- Twenty image files of input file set for recognition. In case of simplicity, the recognition is limited to the given number.

Even though the training and input set are gathered using third party image processing tool with noise-free images, the recognition reading still in negative result. The result is expected because this test is like comparing two sets of different person’s handwriting thus, resulting in a little-matched and poor recognition result. There are a lot of points affecting the poor recognition; some of them are listed below:

- The fact that two sets of input and training files may represent two distinctive person’s handwriting.
- Even though say that the two sets are provided by the same person, the letters have only a little chance to be similar to the one provided before. Hence, the recognition using crossed-set is expected to be low.
- Even though the result yields low mark in recognition result, there are some letters that are perfectly recognized even though it is crossed-tested. Those letters are the one with little or no curving pattern such as letter ‘T’ and ‘E’

Clearer representation of the recognition accuracy is shown in Table 1.

Table 1. Recognition Accuracy

Training Set	Input Set	Accuracy	Expected Error	Accuracy Delta
Training Set 01	Input Set 01	100%	0.01	99.99%
Training Set 01	Input Set 02	25%	0.01	24.99%
Training Set 02	Input Set 01	25%	0.01	24.99%
Training Set 02	Input Set 02	100%	0.01	99.99%

REFERENCES

[1] Aiquan Yuan, Gang Bai, Lijing Jiao, Yajie Liu, "Offline Handwritten English Character Recognition based on Convolutional Neural Network" in the 10th IAPR International Workshop on Document Analysis Systems, pp. 125 – 129. 2012.

[2] B.Indira,M.Shalini,M.V. Ramana Murthy,Mahaboob Sharief Shaik,"Classification and Recognition of Printed Hindi Characters Using Artificial Neural Networks", IJIGSP, vol.4, no.6, pp.15-21, 2012

[3] Anita Pal, Dayashankar Singh, "Handwritten English Character Recognition using Neural Network" International Journal of Computer Science and Communication, Vol. 1, No. 2, pp. 141-144, 2010.

[4] Azizah Suliman, Mohd. Nasir Sulaiman, Mohamed Othman, Rahmita Wirza, "Chain coding and pre processing stages of handwritten character image file," Electronic Journal of Computer Science and Information Technology (eJCSIT), vol. 2, no. 1, pp. 6-13, 2010.

[5] Ashok Kumar, Pradeep Kumar Bhatia, "Offline handwritten character recognition using improved backpropagation algorithm" International Journal of Advances in Engineering Sciences, Vol. 3 (3), 2013.

[6] Yusuf Perwej, Ashish Chaturvedi, "Neural Networks for Handwritten English Alphabet Recognition", International Journal of Computer Applications (0975 –8887), Volume 20 – No.7, April 2011.

[7] Liangbin Zheng, Ruqi Chen, Xiaojin Cheng, "Research on Offline Handwritten Chinese Character Recognition Based on BP Neural Networks", 2011 International Conference on Computer Science and Information Technology(ICCSIT2011), IPCSIT vol. 51 (2012), 2012, IACSIT Press, Singapore

[8] Mariusz Bernacki, Przemysław Włodarczyk, "Principles of training multi-layer neural network using backpropagation", URL: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

[9] Performance Testing URL: http://agiletesting.blogspot.com/2005/02/performance - vs - load - vs - stress - testing.html

[10] Sandeep B. Patil, G.R. Sinha,"Real Time Handwritten Marathi Numerals Recognition Using Neural Network", IJITCS, vol.4, no.12, pp.76-81, 2012.

[11] Shailendra Kumar Dewangan,"Real Time Recognition of Handwritten Devnagari Signatures without Segmentation Using Artificial Neural Network", IJIGSP, vol.5, no.4, pp.30-37, 2013.DOI: 10.5815/ijigsp.2013.04.04

[12] Soumen Bag and Gaurav Harit, "Topographic Feature Extraction For Bengali And Hindi Character Images", International Journal of Signal & Image Processing, Vol. 2, No. 2, pp. 181-196, June 2011

[13] Sumedha B. Hallale, Geeta D. Salunke, "Offline handwritten digit recognition using neural network" International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, Vol. 2, Issue 9, pp. 4373 – 4377, 2013.

[14] Sutha.J, Ramraj.N, "Neural Network Based Offline Tamil Handwritten Character Recognition System", IEEE International Conference on Computational Intelligence and Multimedia Application, 2007,2, 13-15, Dec.2007, Page(s):446-450, 2007.

[15] Tirtharaj Dash and Tanistha Nayak, "English Character Recognition using Artificial Neural Network", Proceedings of National Conference on AIRES-2012, Andhra University, pp. 7-9, 29-30 June, 2012

[16] Vijay Laxmi Sahu, Babita Kubde "Offline handwritten character recognition techniques using neural network: a review" International Journal of Science and Research (IJSR), Vol. 2 Issue 1, pp. 87 – 94, 2013.

Authors' Profiles

Rosalina: Received B.Sc and M.Sc from President University. Currently working as a Lecturer in the Faculty of Computing, President University, Indonesia.

Dr. R.B. Wahyu: Received Bachelor degree from University of Indonesia, Master degree from Curtin University of Technology and Doctoral degree from Institute of Technology Bandung, Indonesia. Currently working as a Lecturer in the Faculty of Computing, President University, Indonesia.

How to cite this paper: Rosalina, R.B. Wahyu, "Offline Handwriting Recognition Using Feedforward Neural Network", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.9, pp.11-17, 2017. DOI: 10.5815/ijitcs.2017.09.02