# PIXEL DOWNSAMPLING FOR OPTIMIZATION OF ARTIFICIAL NEURAL NETWORK FOR HANDWRITING CHARACTER RECOGNITION

Kani[1, 2], Irman Hermadi[2, 1] and Agus Buono[1]
[1]Department of Computer Science, Faculty of Mathematics and Natural Sciences, IPB University, Bogor, Indonesia
[2]Department of Information Technology, Open University, Jakarta, Indonesia
E-Mail: kani_launggu@apps.ipb.ac.id

## ABSTRACT

The aim of this study was to develop an image preprocessing model that utilize downsampling technique to reduce the pixel matrix to optimize artificial neural network in order to facilitate the handwriting recognition for letter A,B,C,D and E. In the proposed model, the handwriting images was first subjected to binarization process, the followed by the pixel matrix downsampling first using the column approach (C-DS), then combine raw and column approach (RC-DS). The compressed pixel (downsampled pixel matrix) then acted as an input vector for Artificial Neural Network (ANN). The functionality of the proposed method was demonstrated by its application to handwritten characters consisting of A, B, C, D and E examination choices. The results of the simulation indicated the proposed downsampling using combine column and row presented the higher accuracy (98.80%) and low pattern range (3.30%) with a minimum RMSE (0.1). The model further presented low execution time (560 Second) when compared to normal backpropagation. Thus base on the simulation results the proposed method outperformed the normal backpropagation and provide a reliable and efficient image preprocessing approach for the input of Artificial Neural Network.

**Keywords:** ANN, back propagation, down sampling.

## INTRODUCTION

Handwriting often tends to be indistinguishable even to the human eyes, and that they can only be distinguished by context [1]. To distinguish between such similar characters the tiny difference must be identified and one of the major problems in identification for handwritten characters is that they appear at the same relative location of the letter written by different writers even the same person may not always have the letter with the same proportion [2]. Thus identification of handwriting recognition often poses a great challenge especially to many expert systems developed by the artificial neural networks. Recognizing handwriting alphabet in English has been deemed difficult especially when handling handwriting of infinitely different character [3], [4].

Therefore, several feature extraction techniques have been applied to extract handwriting images such as multiscale training technique (MST) [5]. For instance, recently, Devanagari script which implements intersections, showdown features, chain histogram and straight-line fitting have been used [6]. Moreover, handwriting in English recognition using row-wise segmentation technique (RST) have been used to find a common feature of some character written in the different style by segmentation of the input matrix into separate row and trying to find out common rows among different styles. Therefore to identify the common features among the characters written by the different individual often image proposing techniques are usually applied to preprocess the pixel matrix for the artificial neural networks like perceptrons learning method [7], [8].

Other methods like optical character recognition (OCR) has been proposed to translate the image of handwritten, typed or printed text by means of the scanner into the machine, this approach recognizes neural network using nearest neighbor OCR algorithm [9]. Therefore, in order to identify the common feature among the characters written by individual appropriate image preprocessing techniques are always applied [12]. Recently downsampling have been adopted for compression of fingerprint images in fingerprint authentication system, however the method fail to combination both column and row of the fingerprint pixel matrix thus rendering the method less efficient and accurate for fingerprint pattern recognition [13].

Therefore, in this paper we introduce a novel downsampling image preprocessing technique of image matrix for ANN to recognize the handwritten character using the neural network. The aim of this study was to develop a feature extraction model utilizing the downsampling approach to compress pixel matrix of different unique handwriting images of the five alphabetical letters (A, B, C, D & E) written on examination answer sheets.

The rest of this paper is organized in three parts; formulation and compression of image matrix using column downsampling (C-DS) model and combine row-column downsampling RC-DS model, training the neural network followed by testing the neural network by providing the handwritten character taken from different induvial finally results and discussion then conclusion.

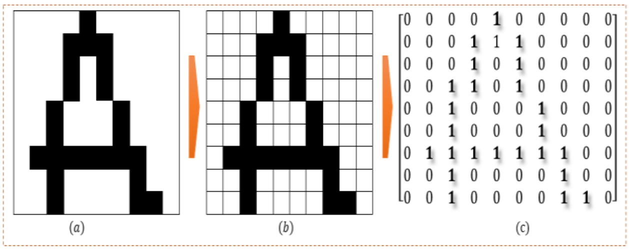
## MATERIAL AND METHODS

To implement the proposed method, 610 images capitals consisting of the image of a capital letter 'a', 'b', 'c', 'd' and 'e' size (dimension) 50x50 taken from samples of handwriting data students of the open university (UT) and has been through the process of *cropping*. The scanned image is then processed by the image processing so that a binary data; then do extraction characteristics of the models *downsampling* (reduced) pixel rows and columns and then subjected to training on ANN-BP.

### Binarization

Binarazation also referred to character coding is the conversion of the grayscale image to binary numbers of 0's and 1's form binary image. here the coloured image we initially converted into grayscale, and followed by floating (thresholding) to convert the image to binary [14]. In this case, we consider the grayscale distinguish the background and the object on the image thus we set our grayscale level 0 to 255. Furthermore, we used threshold approach to using Otsu discriminant analysis approach for variable to distinguish between two or more groups that arise naturally [15].

This approach maximized variables in order to divide the object and the background. thus we converted background image into binary exactly as object image by digitizing grid segments into binary form 0's and 1's where the background image is white while the object image is black [16] as illustrated below(Figure-1).



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

**Figure-1.** Original image; (b): Image grayscale pixel; (c): Character coding with binary digits of 0's and 1's.

### Image compression via downsampling

After characters binarization step, we subject the characters pixel matrix to downsampling to generate the input for the Artificial Neural Network (ANN). In this case, downsampling is based on the summation of pixel values for the column and combined row-column the arithmetic sum colon or combined row-column pixel matrix are the divided with the number of columns or rows. In that note to implement the proposed technique, we consider the column and combine row-column as input vector for ANN thus we represent the formulations our models based on the following steps.

### Modelling column-downsampling (C-DS)

After representing the pixel values in matrix form, we compute the summation of pixel value in each row (M) in this case; we sum all the pixel value with 1s on each row we express the matrix as:

$$\begin{bmatrix} x_{11} & + & x_{12} & + & x_{13} & + & x_{14} \\ x_{21} & + & x_{22} & + & x_{23} & + & x_{24} \\ x_{31} & + & x_{32} & + & x_{33} & + & x_{34} \\ x_{41} & + & x_{42} & + & x_{43} & + & x_{44} \end{bmatrix} = \begin{bmatrix} xC_1 \\ xC_2 \\ xC_3 \\ xC_4 \end{bmatrix}$$

thus we denotes the compressed pixel matrix for row as follows:

$$xC_n = \sum_{l=1}^{m} x_{mn}$$

where $xC_n$is the sum of the total pixel values in single row where $n$ = 1, 2, 3,... $i$ and $m$ = 1, 2, 3,... $j$. thus, we represent the input matrix for downsampled pixel as:

$$\text{input matrix } (xC_n) = \begin{bmatrix} xC_1 \\ xC_2 \\ \vdots \\ xC_n \end{bmatrix}$$

thus, we represent the input vector $n$ columns for input matrix as follows:

$$vxC_n = [xC_1 \ldots xC_n]$$

Computing input vector for ANN: to generate the input vector for the ANN we compute the arithmetic mean for each $vxc_n$ thus; we divide the summation (1) with the total number of the column (n) in the pixel matrix. Thus, we express this with equation:

$$vxC_n = \left[\frac{xC_1}{n} \ldots \frac{xC_n}{n}\right]$$

where $vxC_n$is a unit input vector for ann and $vxC_n < 1$

To illustrate the down sampling steps described above consider the core type fingerprint image in Figure-1, with dimension 10 x 10 in this case we represent the pixel matrix and input matrix as:

$$\text{pixel matrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{thus downsampled input vector}(xC) = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 2 \\ 2 \\ 6 \\ 2 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$

where the first matrix represents the original pixel matrix then row summation matrix of the original pixel matrix. in this case, the m = 10 and n = 10 therefore, using (1) and (2) we can compute a unit input vector for ANN by calculating the mean of each summation of the raw pixel by dividing the total number of column (N= 10).

Therefore, the downsampled pixel matrix for input vector of ANN is expressed:

$$\mathrm{vxC_n} = \begin{bmatrix} \frac{0}{10} & \frac{1}{10} & \frac{2}{10} & \frac{2}{10} & \frac{2}{10} & \frac{6}{10} & \frac{2}{10} & \frac{2}{10} & \frac{2}{10} & \frac{0}{10} \end{bmatrix}$$

**Modelling Row and Column-Downsampling (RC-DS)**

For proposed RC-DS model we formulated the input matrix of ANN by combine summation of row (m) and column (n) for the image pixel matrix thus we represented the matrix as:

$$\begin{bmatrix} x_{11} & + & x_{12} & + & x_{13} & + & x_{14} \\ x_{21} & + & x_{22} & + & x_{23} & + & x_{24} \\ x_{31} & + & x_{32} & + & x_{33} & + & x_{34} \\ x_{41} & + & x_{42} & + & x_{43} & + & x_{44} \end{bmatrix} = \begin{bmatrix} xC_1 \\ xC_2 \\ xC_3 \\ xC_4 \end{bmatrix}$$
$$[\, xR_1 \quad xR_2 \quad xR_3 \quad xR_4 \,]$$

Thus we express the summation of the row by the equation:

$$\mathrm{xR_n} = \sum_{l=1}^{n} \mathrm{x_{nm}}$$

thus the summation of matrix row for pixel matrix as:

$$\mathrm{vxR_n} = [\mathrm{xR_1} \ldots \mathrm{xR_n}]$$

To obtain the input vector for the ANN we divide with the total number of row in the matrix therefore we express the input vector as:

$$\mathrm{vxR_n} = \left[\frac{\mathrm{xR_1}}{\mathrm{M}} \cdots \frac{\mathrm{xR_n}}{\mathrm{M}}\right]$$

To generate the proposed RC-DS input vector for ANN we combine equation 1 and 2 thus we express the input vector as:

$$\mathrm{vxRC_n} = \left[\frac{\mathrm{xC_1}}{\mathrm{N+M}} \cdots \frac{\mathrm{xC_n}}{\mathrm{N+M}} \frac{\mathrm{xR_1}}{\mathrm{N+M}} \cdots \frac{\mathrm{xR_n}}{\mathrm{N+M}}\right]$$

The illustration of the proposed RC-DS model as shown below, we consider the same pixel matrix used in the C-DS model with pixel size of 10 x 10 thus we represent pixel matrix and compressed matrix as:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{Thus Iput Vector } (xR) = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 2 \\ 2 \\ 6 \\ 2 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$
$$\mathrm{xR} = [0 \quad 1 \quad 5 \quad 2 \quad 2 \quad 2 \quad 6 \quad 1 \quad 0 \quad 0]$$

Where the first matrix represents the original pixel matrix and second matrix compressed pixel of the original pixel matrix. herein the values for M = 10 and N = 10 therefore, combing equation (1) and (2) we compute the input vector for ann by dividing the summation of raw and column pixel by dividing the total number of row and column (10+10 =20). Hence, we expressed the input vector of ANN as:

$$vxRC = \begin{bmatrix} \frac{0}{20} & \frac{1}{20} & \frac{2}{20} & \frac{2}{20} & \frac{2}{20} & \frac{6}{20} & \frac{2}{20} & \frac{2}{20} & \frac{2}{20} & \frac{0}{20} & \frac{0}{20} & \frac{1}{20} & \frac{5}{20} & \frac{2}{20} & \frac{2}{20} & \frac{2}{20} & \frac{6}{20} & \frac{1}{20} & \frac{0}{20} & \frac{0}{20} \end{bmatrix}$$

**Backpropagation algorithm**

To confirm handwriting authenticity, we subject the downsampled pixel to neural network training. Therefore, back-propagation is utilized as the standard way of training the neural network. In this case, data for input layer of the neural net is downsampled pixel vector. The neural network is then run normally to check if the output is actually the same as the input. The actual number of identified handwriting in the output is then compared to the desired number of the handwriting in the input.

Nonetheless, in this study, the pattern error is the variation between the actual total identified handwriting patterns and the desired number of identified handwriting patterns. Below is the proposed back-propagation algorithm for handwritten character identification system.

**BACK-PROPAGATION ALGORITHM**

```
1.  Initialize the weights $\omega(k)$ to small random values, and choose a positive CONSTANT $c$.
2.  Repeatedly Set $\xi_1^{(0)}, \ldots, \xi_{M0}^{(0)}$ equal to the features of samples 1 to v cycling back to sample 1 after sample n is reached.
3.  Feed forward step. for $\kappa = 0, .., \kappa - 1$, Compute
        $\xi_j^{(k+1)} = \varphi\left(\sum_{i=0}^{M_k} \omega_{\iota\phi}^{(\kappa+1)} \xi_\iota(\kappa)\right)$
    For nodes $\varphi = 1, \ldots, M_{\kappa+1}$. we use the sigmoid threshold function $\phi(\sigma) = 1/(1+\varepsilon^{-\sigma})$.
4.  Back-Propagation step. for the nodes in the output layer, $\varphi = 1, \ldots, M_\kappa$ , compute
        $\delta_\varphi^{(\kappa)} = \xi_\varphi(\kappa)(1 - \xi_\varphi^{(\kappa)})(\xi_\varphi^{(\kappa)} - \delta_\varphi)$
    For Layers $k = K - 1, \ldots, 1$ Compute
        $\delta_\varphi^{(\kappa)} = \xi_\iota^{(\kappa)}(1 - \xi_\varphi^{(\kappa)})\left(\sum_{j=0}^{M_{k+1}} \Sigma\delta_\varphi^{(\kappa+1)} \omega_{\iota\varphi}^{(\kappa+1)}\right)$
    FOR $i = 1, \ldots, M_\kappa$
5.  $P\varepsilon\pi\lambda\alpha\chi\varepsilon\ \omega_{\iota\varphi}^{(\kappa)}$ BY $\omega_{\iota\varphi}^{(\kappa)} - \chi\delta_\varphi^{(\kappa)} \xi_\iota^{(\kappa-1)}$ FOR $\iota, \varphi, \kappa$.
6.  Repeat Steps 2 to 5 until weights $\omega_{\iota\varphi}^{(\kappa)}$cease to change significantly.
```

**Artificial neural network architecture**

Accuracy and speed of pattern recognition handwritten letters are also dependent on artificial neural network model are used. Thus, it is very important to choose a model of the artificial neural network for efficiency performance. so in our case, we apply ANN with three layers, namely; input layer which receives input vectors downsampling result, hidden layer, we use only one with a variation of the number of neurons and the output layer that identifies the image or pattern of handwritten letters.

**Input layer:** In the proposed method the input layer is considered as the first layer of the neural network and is used to input downsampled pixel from input fingerprint file thus it contains input vector for downsample pixel. Note that the input layer of the neuron only takes downsampled pixel values as input data and transmit to the hidden layer.

**Hidden layer:** This is the middle layers between input layer and output layer. In the proposed method by downsampling pixel we reduce the number of hidden layers. In this case the hidden layer compute the weight of neurons from input layer and generate a signal with the help of activation function $\varphi(\sigma) = 1/(1 + \varepsilon - \sigma)$, and transmit the signal to the output layer.

**Output layer:** This is the last layer of ANN and this layer is used to show the results of the data that was trained, the number of outputs to be achieved only 5 outputs, each output produced is taken is the highest value of 5 outputs exist, for example the letter 'a' with an output [1 0 0 0 0] , if the output produced is [0.5 0.3 0.1 -1 0:01], then this recognized as 'a' and since the highest value is 0.5 however, if the character is not a, then output is [0.1 0.4 0.5 0.2 0.7 ] it is the recognized letter character 'e', this is done literately by ANN for all the letters characters to obtain the final output.

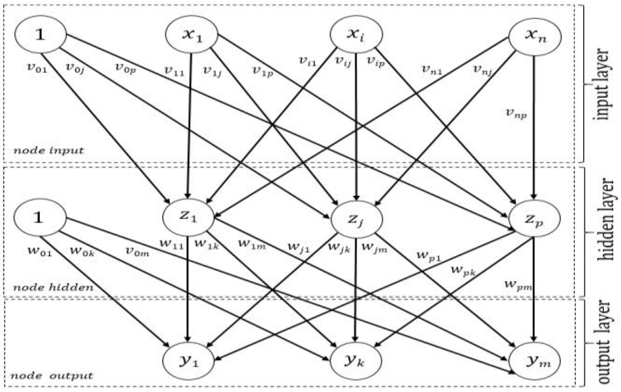


**Figure-2.** Artificial neural network with single hidden layer and single output layer for back-propagation algorithm.

## RESULTS AND DISCUSSIONS

In this study, 3820 handwriting datasets were obtained from examination answers sheet images from Open University, Indonesia. The data was divided into two sets, that is the training data (3210) and testing data (610). out of 3820 handwritten sample 764 consisted of letter 'A', 764 letter 'B', 764 'C', 764 the letter 'D', and 764 letter 'E'. The data were subjected to pixel matrix column downsampling (C-DS) model and pixel combine raw-column pixel matrix downsampling (RC-DS) model for evaluation of performance. From the simulation, the training and testing results for both models were obtained as follows.

## C-DS MODEL

From c-ds analysis (Table-2) it was observed to detect letter e when the number of the hidden layers set at 40 higher accuracies (99.4%) was recorded on the other hand when the number of neurons set to 15 for detection of letter 'b' the algorithm performed with low accuracy 60.1%). from this analysis, we the mode when set the NHL = 40 the proposed model recorded the highest accuracy (90.8%) with low RMSE (0.09) and execution time (452 seconds). furthermore, when the NHL=30 the remarkably low precision (90.5%) was recorded. this might have been attributed to the difference in character for the four letters thus precision range for handwriting pattern recognition was recorded to be 26.9% for letter b when compared with other four letters (A,C, D and E). For our simulation 620, handwriting data sample was used as testing data for ANN backpropagation c-ds and the testing results (Table-3).

From this analysis, we observed that when NHL= 30 and NHL= 40 for detection of letter c, the algorithm recorded higher precision (96.7%), this was in contrast with detection of letter b which was recognized with low precision (60.7%). although remarkable high accuracy (84.6%) was recorded when the NHL =30 adjustment of the NHL resulted in the reduction of algorithm performance due to overfitting.

Surprisingly character “B” recognition presented the low accuracy (63.1%) with a high range (29.5%) when the number of the neuron is set 35. Therefore, based on this analysis we realised that recognition of letter b was recognized with low accuracy for both training and testing data since majority of letters in testing data were recognized either as e or d thus though c-ds model performed thus some letters like b was not accurately recognized from both training and testing data (Figure-4).

Table-1 illustrates the results of c-ds comparison analysis for testing dataset when NHL is adjusted from 15 to 40 neuron nodes in the hidden layer. Again during this analysis, we used similar parameters setting in this case we set the total iterations (epoch=4000), learning rate (0.009), total hidden layer (single hl), vector =50, and output (class).

**Table-1.** Evaluation of character recognition C-DS.

| **NHL** | **Tested characters** | **Accuracy** (%) | **Range** | **Time** (sec) |
|---|---|---|---|---|
| 15 | A,B,C,D,E | 77.9 | 32.0 | 0.06 |
| 20 | A,B,C,D,E | 81.5 | 36.9 | 0.07 |
| 25 | A,B,C,D,E | 83.3 | 32.8 | 0.09 |
| 30 | A,B,C,D,E | 84.6 | 34.4 | 0.11 |
| 35 | A,B,C,D,E | 82.6 | 29.5 | 0.12 |
| 40 | A,B,C,D,E | 83.1 | 36.1 | 0.12 |

Figure-2 illustrates the c-ds comparison analysis for testing dataset and training data when NHL is adjusted from 15 to 40.

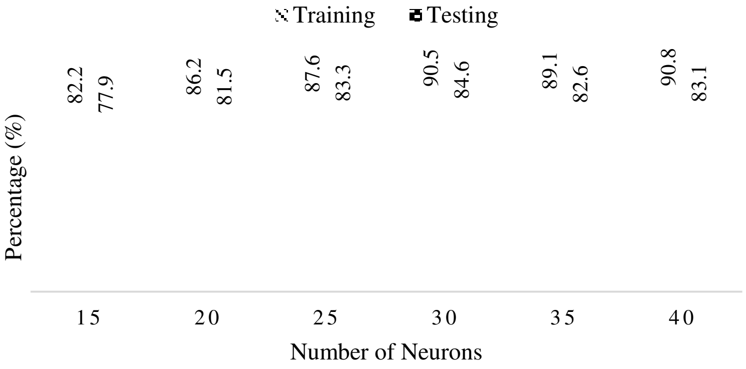


**Figure-3.** The performance comparison of training and testing of C-DS Model.

## RC-DS MODEL

One prime objective of this study was to optimize the ANN recognition of handwriting letters for the college exam; therefore, we introduced a combined model of downsampling by incorporating both the row and column for the pixel matrix (RC-DS). The simulation results (Table-3) indicated that when we set NHL=40 the algorithm recorded the higher precision (99.3%) however when we set the NHL at minimum (15 neurons) letter “A” was recognized with the accuracy of 95%. The interesting part of the proposed approach is its reliability and efficiency in the detection of all the handwriting letters with minimal difference in the handwriting recognition. Furthermore, when we increased NHL from 15 to 40 the minimal range is recorded likewise the RMSE except when we set NHL= 25 and NHL=30 which record RMSE of 0.10. On the other hand, when we set NHL= 40 the algorithm records low range 1.40% after which the NHL is increased the long execution time is experienced by the algorithm.

In addition when considered the testing data for RC-DS the proposed model presented highest accuracy (94.3%) when we set NHL= 35 with a range of 4.10 and execution time of 14 seconds, on the other hand, low accuracy (90.2%) was recorded when we set the number of neuron at 15 with range of 9.02 % and execution time of 8 seconds. surprisingly, we realized that when we set the NHL= 35 and NHL= 40 handwriting letter a was recognized at high accuracy (96.7%) with an average range of 5.33. In order to determine the optimal number neurons in the RC-DS model, we selected the best four number of neuron in both the training and testing data. in case we considers the neuron with the highest accuracy from our simulation we discovered 25,30, 35 and 40 gave the highest accuracy (98.8%, 98.8%, 98.9% and 99.3% respectively), likewise, for testing data when we considered same neurons the RC-DS model presented an accuracy of 92.0%, 90.8%, 94.3% and 93.8% respectively. Nevertheless, we further considered the lowest range; on that premise when we set the neuron 30 gave the optimal range 1.6 % and 3.3% for training and testing data respectively. We discovered that when we consider the NHL= 40 the range 6.6 % which render the algorithm less accurate. Furthermore, when we compute the interquartile range 0.2% (1.6%-1.4%) for training data on another hand for testing we also discovered the algorithm gave high interquartile range 3.3% (6.6%-3.3%) refer to Tables 2 and 3 in the result section. Base on simulation analysis the proposed RC-DS model outperformed C-DS model since the model compressed the pixel matrix of the handwriting letter by both row and column and the result in high accuracy leading to low range and RMSE.

Table-3 illustrates the results of RC-DS comparison analysis testing data when NHL is adjusted from 15 to 40 neuron nodes in the hidden layer. The parameters used in the training data such as, total iterations (epoch=4000), learning rate (0.009), total hidden layer (single hidden layer), vector =100, and output (class).

**Table-2.** Evaluation of character recognition Downsampling.

| NHL | Tested characters | Accuracy (%) | Range | Time (sec) |
|---|---|---|---|---|
| 15 | A,B,C,D,E | 90.2 | 9.02 | 0.08 |
| 20 | A,B,C,D,E | 91.3 | 4.10 | 0.09 |
| 25 | A,B,C,D,E | 92.0 | 6.56 | 0.10 |
| 30 | A,B,C,D,E | 93.8 | 3.28 | 0.12 |
| 35 | A,B,C,D,E | 94.3 | 4.10 | 0.14 |
| 40 | A,B,C,D,E | 93.8 | 6.56 | 0.17 |

**Table-3.** Evaluation of RC-DS performance.

| Tested characters | Total number recognized |
|---|---|
| A | 637 of 642 (99.22%) |
| B | 634 of 642 (98.75%) |
| C | 637 of 642 (99.22%) |
| D | 635 of 642 (98.91%) |
| E | 634 of 642 (98.75%) |
| TOTAL | 3177 of 3210 |
| rc-ds ACCURACY | 98.97% |

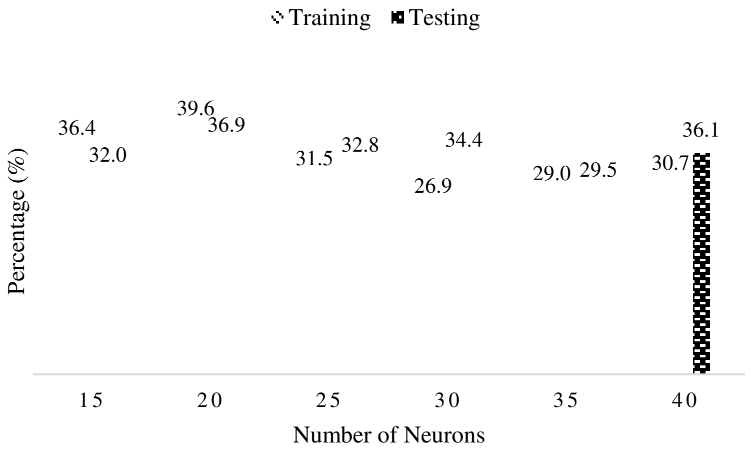


**Figure-4.** The performance comparison of training and testing RC-CD Model.

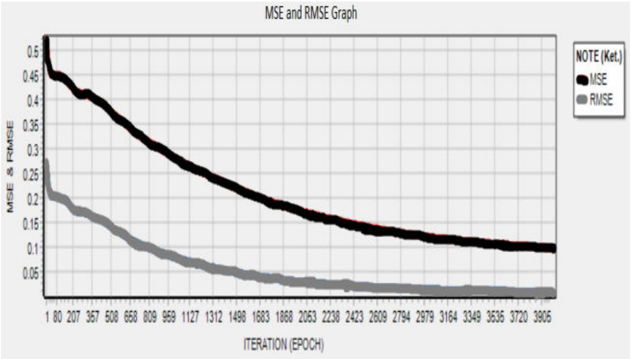


**Figure-5.** Illustrate RMSE and MSE estimation RC-DS at 1000 epoch.

## CONCLUSIONS

This work presents a novel downsampling image pre-processing model for optimization of ANN for handwriting pattern recognition. The downsampling model based on c-ds has indicated the capability of ANN recognizing the handwriting patterns with and accuracy 84.6% when a number of the neuron (NHL) 30 with an execution time of 11 seconds with RMSE of 0.21. In addition, the RC-DS model was the most efficiency from recognition of all the handwriting characters with an accuracy of 94.3% when the number of neurons set 35, with execution time14 seconds with RMSE of 0.09. although c-ds model presented the low execution time RC-DS model performed better with high accuracy and low RMSE thus making the proposed model more efficient and reliable method for image compression (reduction) for handwriting pattern recognition.

## ACKNOWLEDGEMENT

The author expresses their sincere gratitude to the department of computer science at IPB University for insightful comments and contribution and search related to this study. We also give thanks to Computer Center Section at Open University for data to accession.

## REFERENCES

[1] Chen H, Huang X, Cheng P, Xu Y. 2013. Off-line handwritten Chinese character recognition based on GA optimization BP neural network. International Conference on Information, Business and Education Technology (ICIBIT). 1(1): 164-167.

[2] Pradeep J, Srinivasan E, Himavathi S. 2011. Diagonal based feature extraction for handwritten alphabets recognition system using neural network. International Journal of Computer Science and Information Technology (IJCSIT). 3(2): 27-38

[3] S. Farhad, Gharehchopogh, A. Ezzat. 2012. Artificial Neural Network Application in Letters Recognition for Farsi/Arabic Manuscripts. International Journal of Scientific and Technology. 1(8): 90-94.

[4] Basu JK, Bhattacharyya D, Kim TH. 2010. Use of artificial neural network in pattern recognition. International Journal of Software Engineering and Its Applications, 4(2): 23-34.

[5] Rakesh K M, Manna NR. 2012. Hand Written English Character Recognition using Column-wise Segmentation of Image Matrix (CSIM). WSEAS Transactions on Computers. 11(5): 148-158.

[6] Yang W, Xu L, Chen X, Zheng F, Liu Y. 2014. Chi-squared distance metric learning for histogram data. Mathematical Problems in Engineering. pp. 1-12.

[7] Pradeepta K. Sarangi and Kiran K R. 2014. Feature Extraction and Dimensionality Reduction in Pattern Recognition Using Handwritten Odia Numerals. Middle-East Journal of Scientific Research. 22(10): 1514-1519.

[8] Perwej Y, Chaturvedi A. 2011. Neural networks for handwritten English alphabet recognition. International Journal of Computer Applications. 20(7): 1-5.

[9] Mahesh Goyani, Harsh Dani, Chahna D. 2013. Handwritten Character Recognition - A Comprehensive Review. International Journal of Research in Computer and Communication Technology. 2(9): 702-707.

[10] Ochieng PJ, Kani, Harsa H, Firmansyah. 2014. Fingerprint Authentication System Using Back-Propagation with Downsampling Technique. International Conference on Science and Technology IEEE proceedings.

[11] N. Bhargava, A. Kumawat, R. Bhargava. 2014. Threshold and binarization for document image analysis using Otsu's algorithm. International Journal of Computer Trends and Technology (IJCTT). 17(5): 272-275.

[12] Sanjay Kumar, Narendra Sahu, Aakash Deep, Khushboo Gavel, Miss Rumi Ghosh. 2016. Offline Handwriting Character Recognition (for use of medical purpose) Using Neural Network. International Journal of Engineering and Computer Science. 5(10): 18612-18615.

[13] Vala MHJ, Baxi A. 2014. A review on image segmentation algorithm Otsu. International Journal of Advanced Research in Computer Engineering and Technology (IJARCET). 2(2): 387-389.

[14] Yang W, Xu L, Chen X, Zheng F, Liu Y. 2014. Chi-squared distance metric learning for histogram data. Mathematical Problems in Engineering. pp. 1-12.

[15] Yoo SS, Kim YT, Youk SJ, Kim JH. 2006. Adaptive-binning color histogram for image information retrieval. International Journal of Multimedia and Ubiquitous Engineering. 1(4): 45-53.

[16] Saravanan S, Kumar PS. 2014. Image contrast enhancement using histogram equalization techniques: a review. International Journal of Advances in Computer Science and Technology. 3(3): 1-10.