

HANDWRITTEN CHARACTER RECOGNITION USING BRAINNET LIBRARY

Babatunde Akinbowale Nathaniel¹, Abikoye Oluwakemi Christiana²,
Babatunde Ronke Seyi¹, Kawu R. O.²

¹ Department of Computer Science, Kwara State University, Malete, Nigeria

² Department of Computer Science, University of Ilorin, Ilorin, Nigeria

Corresponding author: Babatunde Akinbowale Nathaniel, akinbowale.babatunde@kwasu.edu.ng

ABSTRACT: Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Given its ubiquity in human transactions, machine recognition of handwriting has practical significance, as in reading handwritten notes in a PDA, in postal addresses on envelopes, in amounts in bank checks, in handwritten fields in forms, etc. An off-line handwritten alphabetical character recognition system using multilayer feed forward neural network is described, and a method, called, diagonal based feature extraction is used for extracting the features of the handwritten alphabets. This project implements this methodology using BrainNet Library. Ten data sets, each containing 26 alphabets written by various people, are used for training the neural network and 130 different handwritten alphabetical characters are used for testing. The proposed recognition system performs quite well yielding higher levels of recognition accuracy compared to the systems employing the conventional horizontal and vertical methods of feature extraction. This system, if modified will be suitable for converting handwritten documents into structural text form and recognizing handwritten names.

KEYWORDS: Handwritten Character Recognition, Image Processing, Feature Extraction, Feed Forward Neural Networks, BrainNet Library.

1. INTRODUCTION

Handwriting is a skill that is personal to individuals. Fundamental characteristics of handwriting are in threefold. It consists of artificial graphical marks on a surface. Its purpose is communication and this is achieved by virtue of the mark's conventional relation to language. There are varieties of writing style of handwritten character that vary from one writer to another. Due to this, it becomes a very hard task for the computer to perform handwritten character recognition. The main reason for the absence of accurate system for the handwritten character recognition is the difference in writing styles of the writers.

Moreover, the accuracy of the recognition of the handwritten character depends upon the number of samples used for training and the number of epochs

(iteration) the system is trained. It will become very difficult to recognize a handwritten character due to the unwanted slants, skews and curves. Looking at the past it can be seen that a lot of methods have been developed for the handwritten character recognition. The neural network is used in most of the handwriting recognition methods due to its classification efficiency and ease of use. ([M+15]).

The concept of neural network is inspired by the human's central nervous system. In an Artificial Neural Network (ANN), simple artificial nodes, known as "neurons", "neurodes", "processing elements" or "units", are connected together to form a network which mimics a biological neural network. There is no single formal definition of what an Artificial Neural Network (ANN) is. Any class of mathematical or statistical model may commonly be called "Neural" if they possess the following characteristics ([KS15; Jef16a; Jef16b; SM16]):

- i. consist of sets of adaptive weights, i.e. numerical parameters that are tuned by a learning algorithm, and
- ii. are capable of approximating non-linear functions of their inputs.

The adaptive weights are conceptually connection strengths between neurons, which are activated during training and prediction.

Neural networks are similar to biological neural networks in performing functions collectively and in parallel by the units, rather than there being a clear delineation of subtasks to which various units are assigned. The term "neural network" usually refers to models employed in statistics, cognitive psychology and artificial intelligence. Neural network models which emulate the central nervous system are part of theoretical neuroscience and computational neuroscience.

A neural network for handwriting recognition is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (determined by the network's designer), the

activations of these neurons are then passed on to other neurons ([Jef16a; VS11]). This process is repeated until finally, an output neuron is activated. This determines which character was read. Handwriting recognition has been one of the most fascinating and challenging research area in the field of image processing and pattern recognition in the recent years ([SPK13]). It contributes immensely to the advancement of an automation process and can improve the interface between man and machine in numerous applications. Several research works have been focusing on new techniques and methods that would reduce the processing time while providing higher recognition accuracy ([Agn12; GS90; GG15]).

2. RELATED WORK

Sanyal ([San15]) used the Hidden markov Model (HMM) in segmenting characters into their basic geometric primitives. He also trained the system with the same model for recognition. In his work, he divided the character image into a $n \times m$ zones from where he extracted features to form the feature vector. The essence of the zoning was to enable him obtain local characteristics of the the character instead of the global characteristics. He obtained different segments after zoning while the transition and emission probability of each segment of the characters were calculated in the algorithm. He matched each segment with the basic geometric primitives by taking the sum of the absolute value of the difference of the bitmap image of the segment and primitive. Lesser value means the segment is similar to the primitive. His algorithm was implemtned using Matlab 8.0 and tested on both upper and lower case. The result of the tests shows that it has 74% accuracy. However, different alphabets has different type of complexity.

Rajbala et al., ([RA12]) described some methods to retrieve feature space. Some feature extraction methods which were implemented. Such as creating a character matrix of binary values of image, Region and pixel frequency based feature extraction methods-13-region. After getting feature space from the binary character image, an efficient classifier is used to classify the class of a character. Neural Network was used for handwritten character Recognition classification. Two types of data sets are used: Dataset 1 was taken from MATLAB dataset of Dillep Gaurav for handwritten character recognition in which first 15 sets are less noisy than remaining 10 characters. Dataset 2 was built by writing characters by different writers on computer screen which have randomly noisy sets.

Majid et al., ([M+12]) introduced the base of an algorithm for pre-processing stage, so that the

desired data will be collected by this algorithm from on-line hand-drawn letters (called the raw data). Interpolating or smoothing, resembling, normalizing and pre-segmenting of the raw data inputs are the sub stage of pre-processing stage. Some of these stages are utilized for implementing a research by decision making techniques. The main idea of this algorithm is to find the local maximum and minimum points of this sequential list as strokes' tokens. The following common features were obtained by geometrical functions and all acquired critical points were used. The first feature is the direction of the straight line of each token from both the start point and end point. The second feature is the orientation of each token which is either clockwise (CW) or counter clockwise (CCW).The classification stage is based on the features extracted in binary input and using a Back Propagation with a Multilayer Perceptron (BP/MLP) neural network for recognizing the letters.

Rakesh et al., ([RM12]) used a scanner to scan the handwritten English alphabets written on a piece of A4 size paper. Each character is enclosed in a box of size 80×80 in jpeg format. The image is converted into a binary matrix of size 80×80 . Each matrix contains large number of elements which conveys no information about the pattern. In order to eliminate such elements 80×80 matrix is compressed into a matrix of size 10×10 . The 10×10 matrix is segmented column-wise into 10 segments of size 10 each. All the columns of a particular character are mapped into identical patterns used to recognize that particular character. In this way standard weight matrix is obtained for each character.

According to the design of Chirag et al., ([CRP11]) there are basically two main phases in handwritten recognition: the preprocessing and character recognition. The first phase includes preprocessing the given scanned document for separating the Characters from it and normalizing each character. To do this, it converts the input image to grayscale format, and then uses threshold to convert this grayscale image to binary. To achieve the second phase a method is used to extract a given character from the document. For this purpose the graphics 8-way connected algorithm was modified (which they called Edge Detection).

3. METHODOLOGY

A typical handwriting recognition system consists of pre-processing, segmentation, feature extraction, classification and recognition, and post processing stages. The schematic diagram of the proposed recognition system is shown in Figure 1.

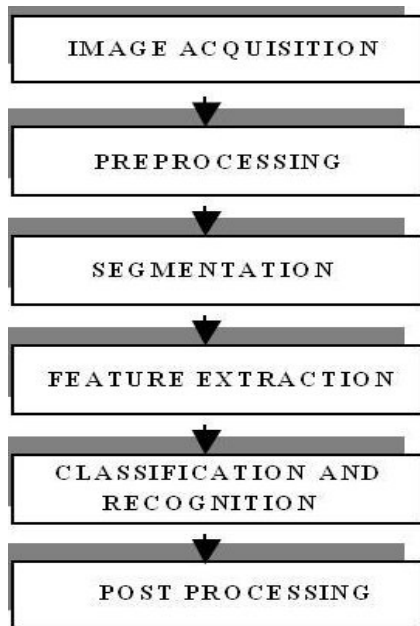


Figure 1: Schematic Diagram of the Proposed Recognition System

Two phase processes are involved in the overall processing of our proposed scheme: the Pre-processing and Neural network based Recognizing tasks.

3.1 PREPROCESSING

The pre-processing step handles the manipulations necessary for the preparation of the characters to be fed in as input to the neural network. First, the required character or part of characters needs to be extracted from the pictorial representation. The following pre-processing steps may also be required to boost the recognition process:

- i. The alphabets can be thinned and their skeletons obtained using well-known image processing techniques, before extracting their binary forms.
- ii. The scanned documents can be “cleaned” and “smoothed” with the help of image processing techniques for better performance.

Further, the preprocessing step involves the digitization of the segment grids because neural networks need their inputs to be in the form of binary (0s & 1s). The next phase is to recognize segments of the character. This is carried out by neural networks having different network parameters. Each of the digitized segments is then provided as input to the each node of neural network designed especially for the training of that segment. Once the networks trained for these segments has been able to recognize them. Characters which are similar looking but distinct are also distinguished at this time.

The Preprocessing Algorithm

Input: Isolated handwritten grayscale image

Output: Preprocessed image

```
{  
Step 1: Read a gray scale image  
Step 2: Remove the noise using median filter  
Step 3: Convert the gray scale image to binary image using threshold value  
Step 4: Normalize the image to a predefined size of 20×20  
Step 5: Convert the normalized image to a single pixel thickness using thinning.  
}
```

3.1.1 Noise Removal Algorithm

Images taken with both digital cameras and conventional film cameras will pick up noise from a variety of sources. Many further uses of these images require that the noise will be (partially) removed - for aesthetic purposes as in artistic work or marketing, or for practical purposes such as computer vision. Median filter is an example of a non-linear filter and, if properly designed, is very good at preserving image detail. The Steps employed by Median filter are:

1. Consider each pixel in the image.
2. Sort the neighboring pixels into order based upon their intensities.
3. Replace the original value of the pixel with the median value from the list.

3.1.2 Character size normalization

Handwriting produces variability in size of written digits. This leads to the need of scaling the character size within the image to a standard size, as this may leads to better recognition accuracy. Here the sizes of alphabets are normalized to a 20×20 pixel.

3.2 FEATURE EXTRACTION

Input: Preprocessed image

Output: Feature vector containing 4 features.

```
{  
Step 1: Read a preprocessed image.  
Step 2: Every character image of size 20×20 pixels is divided into 4 equal zones, each zone of size 10×10 pixels sub image.  
Step 3: The features are extracted from each zone pixels by moving along the diagonals of its respective 10×10 pixels.  
Step 4: Each zone has 19 diagonal lines and the foreground pixels present along each diagonal line is summed to get a single sub-feature and thus 19 sub-features are obtained from the each zone
```


Step 5: These 19 sub-features values are averaged to form a single feature value and placed in the corresponding zone. This procedure is sequentially repeated for all the zones of image.

Step 6: Finally, 4 features are extracted for each image.

}

3.3 CLASSIFICATION USING NEURAL NETWORKS

There are two distinct stages when using ANNs to process any kind of data, a training stage and a classification stage. These two stages can be framed in terms of handwriting recognition. In the classification stage, samples are passed as inputs into the ANN, resulting in an output representing what the ANN believes to be the most correct output.

3.3.1 Training the ANN Model

A well-known Multi-Layer Perceptron (MLP) is used for Training and Testing. The complete process is given in Algorithm 3.

Neural Network Training Algorithm

The training algorithm of back propagation involves four stages:

1. Initialization of weights
2. Feed forward
3. Back Propagation of errors
4. Updating the weights and biases.

Method

Initialization of weights

Step 1: Initialize weight to small random values.

Step 2: While stopping condition is false, do Steps 3-10

Step 3: For each training pair do Steps 4-9

Feed Forward

Step 4: Each input unit receives the input signal X_i and transmits the signals to all units in the layer above i.e. hidden units

Step 5: Each hidden unit ($Z_j, j=1, \dots, p$) sums its weighted input signals $Z_{inj} = V_{oj} + \sum X_i V_{ij}$ applying activation function $Z_j = f(Z_{inj})$ and sends this signal to all units in the layer above, i.e. output units.

Step 6: Each output unit ($Y_k, k=1, \dots, m$) sums its weighted input signals $Y_{ink} = W_{ok} + \sum Z_j W_{jk}$ and applies its activation function to calculate the output signals. $Y_k = f(Y_{ink})$

Back Propagation of Errors

Step 7: Each output unit ($Y_k, k=1, \dots, m$) receives a target pattern corresponding to an input pattern, error information term is calculated as $\delta_k = (t_k - Y_k) f'(Y_{ink})$

Step 8: Each hidden unit ($Z_j, j=1, \dots, n$) sums its delta inputs from units in the layer above

$$\delta_{inj} = \sum \delta_j W_{jk}$$

The error information term is calculated as $\delta_j = \delta_{inj} f'(Z_{inj})$

Updating of weight and bias

Step 9: Each output unit ($Y_k, k=1, \dots, m$) updates its bias and weights ($j=0, \dots, p$) the weight correction term is given by $\Delta W_{jk} = \alpha \delta_k Z_j$ and the bias correction term is given by $\Delta W_{ok} = \alpha \delta_k$

Test the stopping condition.

The stopping condition may be the minimization of the errors, number of epochs

4. BRAINNET LIBRARY

BrainNet is an artificial neural network library mainly developed using .NET framework. It uses the concept of neural network to process and recognize images. Some of the major classes of BrainNet are below:

Table 1. Classes of BrainNet

BackProp	A backward propagation neuron strategy. This is a concrete implementation of
NeuronStrategy	INeuronStrategy
Network Helper	The class is to help the user to initialize and train the network. It maintains a list of training data elements.
Neural Network	A generic neural network. This is a concrete implementation of
NeuralNetworkCollection	INeuralNetwork
Neuron	A collection of neural networks
NeuronCollection	A concrete implementation of
NeuronConnections	INeuron
NeuronConnections	A collection of INeurons
NeuronConnections	This is a hash table to keep track of all neurons connected to/from a neuron, along with the related weights

5. NETWORK FORMATION

The MLP Network implemented for the purpose of this project is composed of 3 layers, one input, one hidden and one output. The input layer constitutes of 400 neurons which receive pixel binary data from a 20 by 20 symbol pixel matrix. The size of this matrix was decided taking into consideration the average height and width of character image that can be mapped without introducing any significant pixel noise. The hidden layer constitute of 400 neurons

whose number is decided on the basis of optimal results on trial and error basis. The output layer is composed of 8 neurons corresponding to the 8-bits of Unicode ASCII character encoding. To initialize the weights a random function was used to assign an initial random number which lies between two preset integers. The weight bias is selected from trial an error observation to correspond to average weights for quick convergence. This is done after initial trials with unity weights are done.

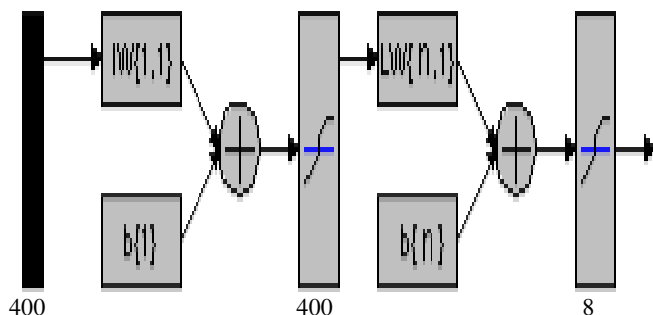


Figure 2: Network Formation of MLP Network

6. TRAINING

The training set involves the binary codes of alphabets. This was automated so that the entire letter is input to the system, and then the shape of the segment needed is extracted from this full letter instead of drawing the shape of the segment itself. The examples of training set patterns can be seen below:

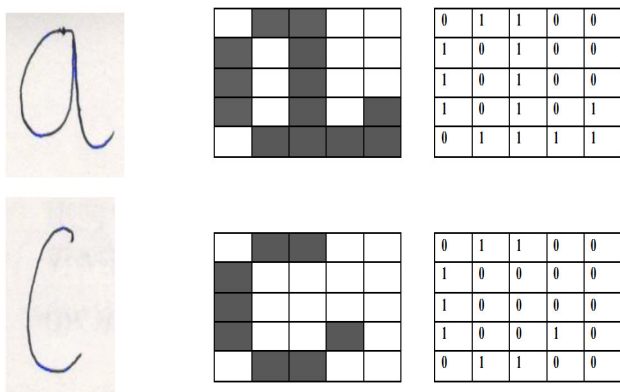


Figure 3: Training Sets

7. IMPLEMENTATION OF THE SYSTEM

The BrainNet library was used to implement this methodology by developing a neural network, which can be trained to perform handwritten character recognition functions. Several objects and methods of the library were used to process handwritten images, train and recognize them accurately, some of which include:

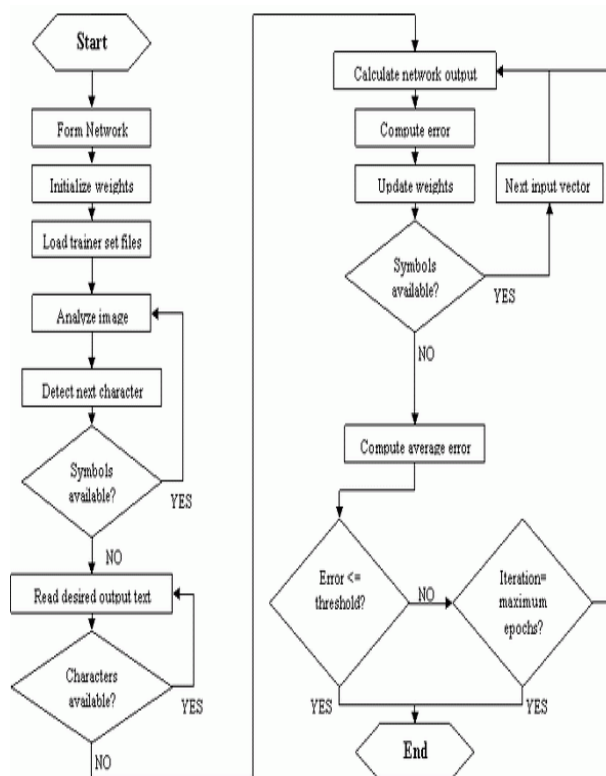


Figure 4: Flowchart for Training the Network

The Train Pattern function: Instead of training the Network directly, Network Helper object was used to train the network. Using a Network Helper object, you can add images and the corresponding ASCII codes directly. Add Training Data method of Network Helper class is gracefully overloaded, so that it can accept various parameters.

The Detect Pattern function obtains the output, after the input must have provided to the network. To convert an image to an array of 1s and 0s, we use the Array List from Image function inside the Image Processing Helper class. After obtaining the output from the network, it should be converted to the equivalent ASCII code to display the character in the textbox. For this, the Number from Array List function in the Pattern Processing Helper class was used. Similarly, the Pattern from Array List function converts an array list to a string (normally, a string of 1s and 0s).

However, it is important to note that the network should be trained with sufficient samples (and sufficient number of times), to obtain desired results. Anyway, it is almost impossible to say that the output of a neural network will be 100% accurate for any input.

Another important point to note is that, the same network can be trained to learn various handwritten images. The network will learn which output to produce for a given input, after a number of training rounds.

8. RESULTS AND DISCUSSION

The pattern detection program can 'learn' the ASCII characters, corresponding to a bitmap (20 x 20 pixel

size). First of all, you need to train the network. To train the network, give some images and corresponding ASCII character value from the 'Train This Network' section. This Network' section.

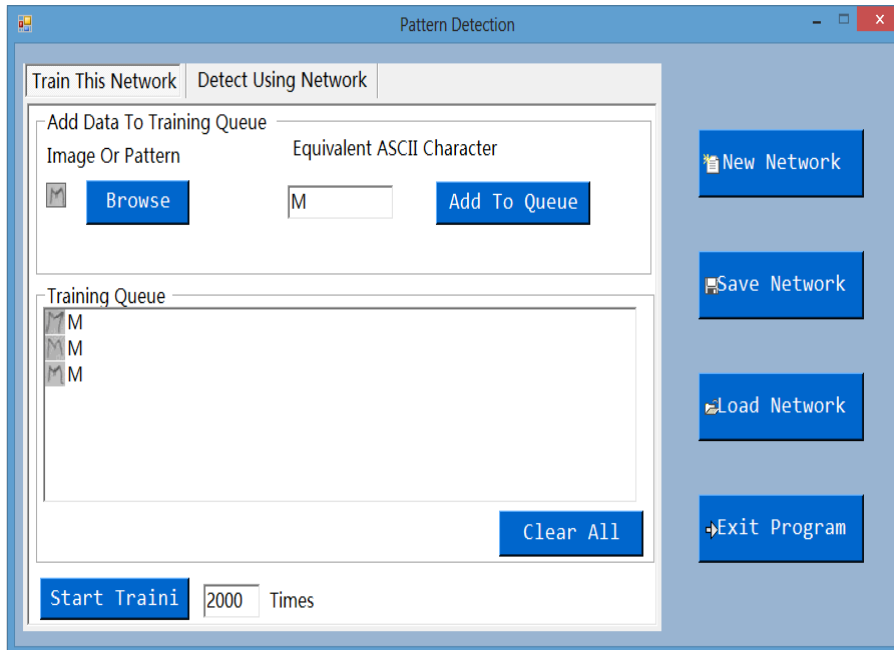


Figure 5: Training - Adding images and corresponding character

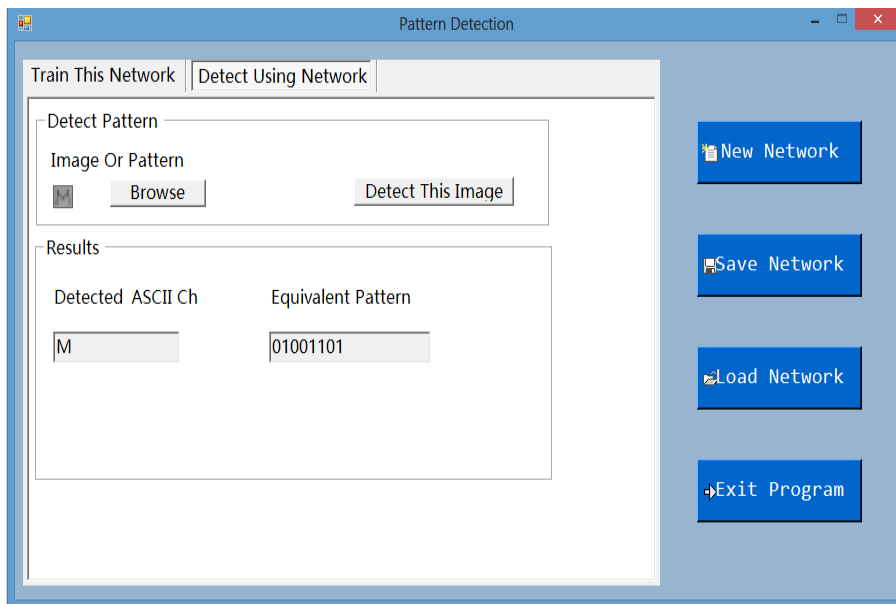


Figure 6: Detecting the Image

For training and simulating purposes certain documents was scanned. There are 2 types of documents train documents and test documents. The train documents are the images of the documents which are to be used for training. Similarly test documents are the images of documents which are to be used for test. According to the characters in the documents, the neural network was trained and the test documents were applied.

A Models of Neural Networks was used. Hence, record of certain parameters like training time,

accuracy etc. was taken to find the effectiveness of the Neural Network.

The result is tabulated in Table 2.

It can be observed from the table that recognition accuracy is lower for the similar input patterns like: C & E; I, J, L & R; and U & V. for these similar patterns in different handwriting, even human eye cannot be able to easily distinguish, hence the machine needs lots of training epochs to recognize them but some with misclassifications.

Table 2. The results

Alphabet	No. of samples for training	No. of samples for testing	No. of epochs (iterations)	Accuracy
A	10	5	5000	94.0
B	10	5	5000	83.0
C	10	5	5000	71.0
D	10	5	5000	88.0
E	10	5	5000	64.0
F	10	5	5000	85.0
G	10	5	5000	89.0
H	10	5	5000	92.0
I	10	5	5000	72.0
J	10	5	5000	73.0
K	10	5	5000	91.0
L	10	5	5000	71.0
M	10	5	5000	86.0
N	10	5	5000	82.0
O	10	5	5000	94.0
P	10	5	5000	88.0
Q	10	5	5000	82.0
R	10	5	5000	70.0
S	10	5	5000	88.0
T	10	5	5000	79.0
U	10	5	5000	80.0
V	10	5	5000	77.0
W	10	5	5000	94.0
X	10	5	5000	91.0
Y	10	5	5000	71.0
Z	10	5	5000	90.0

9. CONCLUSION AND FURTHER WORK

Handwritten Character Recognition is the process of taking images of text or handwriting and converting it to digital form. This is done by a "training" and "Recognition" process in software. If there are any discrepancies, it will be corrected and the software will save that character for future matching. HCR is a very helpful tool, with lots of implementation possibilities. What you otherwise take a lot of time to do manually, is a relatively quick process. HCR allows for any user to convert a document accurately.

From the above results, it can be concluded that with the Built in support for image processing/detection

and pattern processing and the Built in support for advanced training using Training Queues in BrainNet Library the recognition/ detection of handwritten character can be achieved easily and accurately with the library. From the results, the following observations are made:

Increasing the number of epochs has positive proportionality relation to the performance of the network. However in further increasing the number of epochs has an adverse effect of introducing more wrong recognitions. This is called over-learning.

Increase in number of Input characters usually has adverse effect on accuracy which is quite obvious. As the network has to compare input symbol with more symbols, the decision will be subjected to more ambiguity.

Size of the input symbol is also found to impact the detection process. This is due to the fact that large size of symbol is scaled to only 400 pixels and this might not be sufficient to represent all symbols distinctly.

As a research for the future, we intend to increase the number of neurons, that is the size of image and also explore other nature inspired computational techniques such as Artificial Immune System and Bacteria Foraging Optimization Techniques.

REFERENCES

- [Agn12] **V. P. Agnihotri** - *Offline Handwritten Devanagari Script Recognition*, Information Technology and Computer Science, Vol. 8, pp. 37 – 42, 2012.
- [CRP11] **I. P. Chirag, P. Ripal, P. Palak** - *Neural Network for Handwritten English Alphabet Recognition*, International Journal of Science & Engineering Research, Vol. 2, No 5, 2011.
- [GG15] **S. J. Goyal, R. Goyal** - *Feature Extraction Based on Diagonal Direction for Handwritten Recognition System Using Neural Network*. International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, No 4, pp. 1372-1374, 2015.
- [GS90] **V. K. Govindan, A. P. Shivaprasad** - *Character Recognition – A Review*, Pattern Recognition. Vol. 23, No 7, pp. 671-683, 1990.
- [Jef16a] **S. Jeffrey** - *Data Analytics Using Open Source Tool*, 2016.

- [Jef16b] **S. Jeffrey** - *Predictive Analytics Using R*, 2016.
- [KS15] **P. Kumar, P. Sharma** - *Artificial Neural Networks – A Study*, International Journal of Emerging Research and Technology, Vol. 2, No 2, pp. 143-148, 2015.
- [M+12] **H. Majid, M. Dzulkifli, S. M. R. Mohd, M. H. Sami, A. Mahboubeh** - *Handwritten Arabic Character Recognition Based on Minimal Geometric Features*, International Journal of Machine Learning and Computing, Vol. 2, No. 5, 2012.
- [M+15] **D. M. Mark, D.T. Migel, V. Tony, V. S. Andre, T. Jonathan** - *Fast, Simple and Accurate Handwritten Digit Classification by Training Shallow Neural Network Classifiers with the 'Extreme Learning Machine Algorithm*, Plos one, Vol 5, No 8, 2015.
- [PS00] **R. Plamondon, N. S. Sargur** - *On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No 1, 2000.
- [PS12] **J. S. Pramod, K. V. Suraj** - *Unicode Optical Character Recognition and Translation Using Artificial Neural Network*, International Conference on Software Technology and Computer Engineering (STACE), 2012.
- [PSH11] **J. Pradeep, E. Srinivasav, S. Himavathi** - *Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network*, International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No 1, 2011.
- [RA12] **T. Rajbala, B. Aruna** - *A Comparative Analysis of Feature Extraction Techniques for Handwritten Character Recognition*, International Journal of Advanced Technology & Engineering Research (IJATER), Vol. 2, No 4, pp. 2-15, 2012.
- [RM12] **K. M. Rakesh, N. R. Manna** - *Hand Written English Character Recognition using Column wise Segmentation of Image Matrix (CSIM)*, Department of Computer Science & Application, University of North Bengal, PO: NBU, Distt: Darjeeling, West Bengal-734013, INDIA, Vol.11, No 5, 2012.
- [San15] **G. Sanyal** - *Handwritten Character Recognition Using Hidden Markov Model Segmenting into Equivalent Basic Geometric Primitives*, Journal of Science and Management (LJSM), Vol. 1, pp. 55-59, 2015.
- [SM16] **A. R. Srinivasa, G. Malleswari** - *Application of Artificial Neural Networks in Image Processing and Segmentation*, International Journal of Modern Electronics and Communication Engineering, Vol 4, No 2, pp. 9-12, 2016.
- [SPK13] **N. Sharma, T. Patnaik, B. Kumar** - *Recognition for Handwritten English Letters: A Review*, International Journal of Engineering and Innovative Technology (IJEIT), Vol. 2, No 7, pp. 318-321, 2013.
- [VS11] **D. Vinita, D. Sunil** - *Handwritten Character Recognition using Artificial Neural Network*, Advances in Computing, Vol. 1, No 1, pp. 18-23, 2011.
- [V+11] **R. N. Venkateswara, A. Srikrishna, B. G. Raveendra, M. B. Rama** - *An Efficient Feature Extraction and Classification of Handwritten Digits Using Neural Networks*, International Journal of Computer Science, Engineering and Application (IJCSEA), Vol. 1, No 5, 2011.
- [YA11] **P. Yusuf, C. Ashish** - *Neural Networks for Handwritten English Alphabet Recognition*, International Journal of Computer Applications (0975 - 8887), Vol. 20, No 7, 2011.