Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 58 (2013) 14992-14996

Elixir ISSN: 2229-712X

A Novel Approach to Distorted English Character Recognition Using Back Propagation Neural Network

Nachamai M*, Joy Paulose, Ancy Kuruvilla, Nisha G and Sutha Devi S

Department of Computer Science, Christ University, Bangalore, India.

ARTICLE INFO

Article history:

Received: 13 March 2013;

Received in revised form:

15 May 2013;

Accepted: 17 May 2013;

Keywords

Distorted character recognition,

Back Propagation Neural Network,

OpenGL.

ABSTRACT

A person's learning of a new language starts with learning alphabets of the language. The validation of a learning process is its recognition under any circumstances. The application developed in this paper, is aimed at making the "recognition process of alphabets" after different distortions on the original structure and frame of the alphabet. The system creates the distorted alphabet with graphical transformations and recognition is done by a back propagation neural network. The graphical transformations include scaling, rotation and translation functions written in Open GL. The distorted character is recognized by the neural network coded in C#. The system can be deployed for any distorted image recognition application. The system has shown satisfactory performance for distorted character recognition.

Introduction

Character recognition is a trifling task for humans, but to make a computer program do that is awfully intricate [1]. Recognizing patterns is one of the tasks that humans do well and computers don't. The main reason would possibly be the prevalent sources of enormous variability. Noise for example, consists of random changes to a pattern, particularly near the edges, and a character with much noise may be interpreted as a completely different character, by a computer program [2]. Another source of confusion is the high level of abstraction; there are thousands styles of font type in common use and a character recognition program must recognize any of them.

There exist several different techniques for recognizing characters. One of the practical techniques distinguishes characters by the number of loops in a character and the direction of their concavities. The technique uses back propagation in a neural network and this paper will investigate how good neural networks [3] solve the character recognition problem.

Among many applications that have been proposed for neural networks in the literature [4], character recognition has been one of the most successful. Compared to other methods used in pattern recognition, the advantage of neural networks is that they offer a lot of flexibility to the designer, i.e. expert knowledge can be introduced into the architecture to reduce the number of parameters determined by the training examples.

OpenGL is a software interface that allows a programmer to communicate with graphics hardware. OpenGL is supported across many platforms and devices. Perhaps the greatest advantage that OpenGL provides to implementers is its support for extensions. If the OpenGL specification does not provide support for specific functionality, the hardware or software designer may decide to add this functionality themselves through the use of extensions. The extension facilities were adopted for connecting it to the neural network, as the system architecture demands integration for the recognition.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. It is stated that ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. This adaptable feature seems to be the best pick/ fit into this integrated recognition system.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience. The alphabets in its original frame are trained for minimum epochs.
2. Fault Tolerance: Partial information provided for the network, that is, when transformations are made, the image provided may have missing data, which can be handled.

Back propagation neural network [5] is a kind of artificial neural network that has feed forward connections and back connections from the output layer back to the hidden layer that uses a supervised learning method. BPNN is one of the most frequently utilized neural network techniques for classification and prediction, due to the fact that it considers advanced multiple regression analysis that can accommodate complex and non-linear data relationships. The learning algorithm in a BPNN differs from traditional feed-forward neural networks: first, a BPNN uses an activation function for the hidden unit and not the

Tele:

E-mail addresses: nachamai.m@christuniversity.in

input value and second, the gradient of the activation function. The output of a BPNN is compared with the target output and an error is calculated for a training iteration. This error is then back propagated to the neural network and utilized to adjust the weights, thereby minimizing the mean squared error between the network's prediction output and the target output. Consequently, the BPNN model yields predictive output that is similar to the target output.

2.. Methods And Materials

Character recognition systems aim at attaining a system that can recognize a character when an exact match is found. Distorted character recognition happens to be a great challenge, as the character might completely be of a different structure which may not relate to the original form. The work in the paper is come with a new method for identifying distorted characters, the general flow is depicted in figure 2.1. Treating the characters as graphical elements on screen, the distortions are introduced through various graphical transformations. Then the neural network is deployed to identify the distorted character.

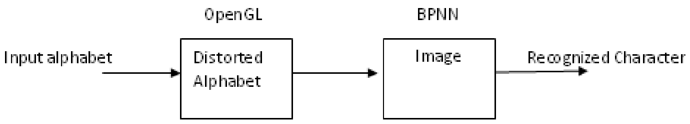

Figure 2.1 General Flow of the System

2.1 Graphical Transformations

The first phase of the work includes selecting a particular character on screen and applying various graphics primitives. The implementation of phase 1 is done using OpenGL [6]. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives.

OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels. This is done by a graphics pipeline known as the OpenGL state machine. Most OpenGL commands either issue primitives to the graphics pipeline or configure how the pipeline processes these primitives.

The graphical transformations defined in the system are:

- Translation
- Scaling
- Rotation on x- axis
- Rotation on y-axis
- Shearing
- Reflection
- Orthogonal projection

The GUI of the system is depicted in figure 2.1.1.

The Left panel has object properties- which include the color change option for the selected alphabet. In continuation left panel has the alphabets and rotation menu for X axis and Y-axis with directions – clock wise and anticlockwise. The Center region is the character display region. The right panel has the remaining alphabets.

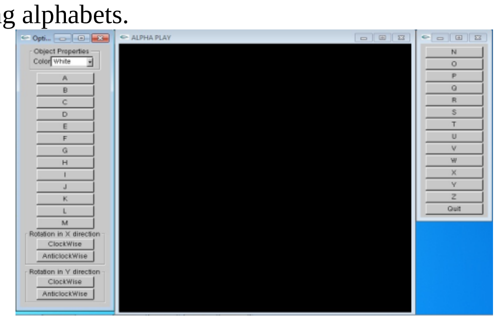

Figure 2.1.1 GUI of the system

The alphabet "A" is clicked on the left panel and it appears on the character display region as shown in the figure 2.1.2.

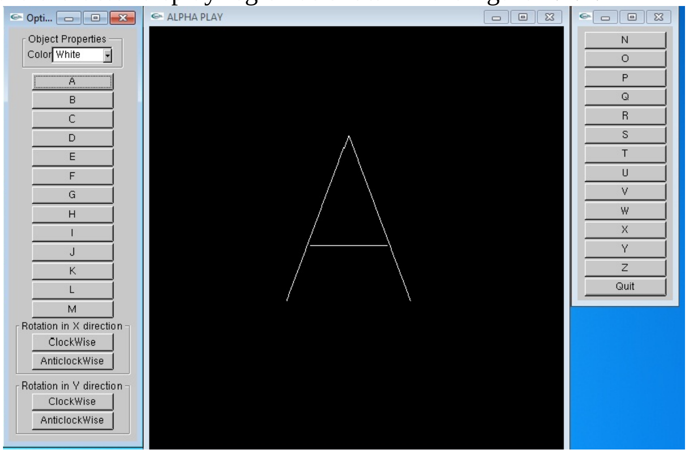

Figure 2.1.2 Alphabet Display

The figure 2.1.3 shows the color menu that can be applied on the character selected. The system is not restricted to balck and white or gray sclae, it is vulnerable to color characters too.

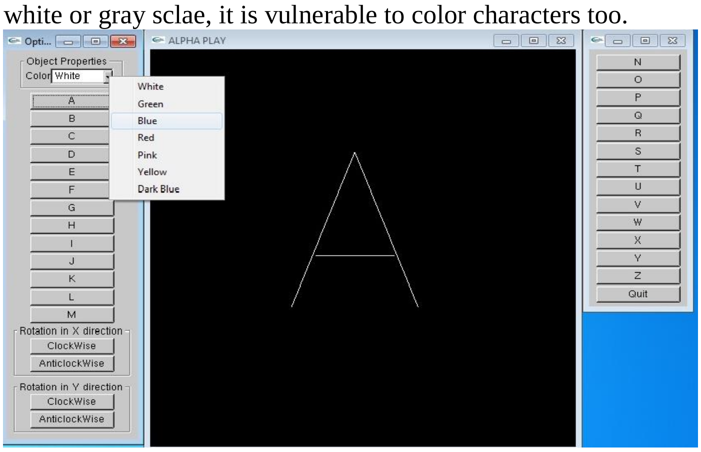

Figure 2.1.3 Color Menu.

The color application of "blue" is shown in figure 2.1.4.

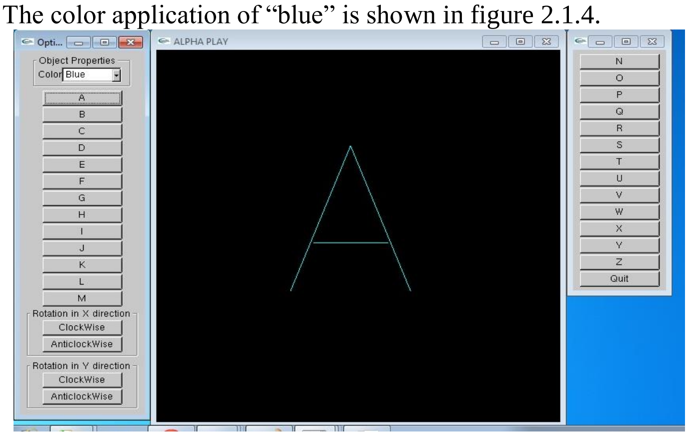

Figure 2.1.4 Color Change on the Alphabet

A sample demonstration of the transformations such as the anti clockwise rotation of the character in Y-axis, translation of the alphabet and scaling of the alphabet are depicted in figure 2.1.5, 2.1.6 and 2.1.7 respectively.

2.2 Character Recognition

The recognition of character is done using the BPNN. The BPNN architecture comprises one input layer, many hidden layers and one output layer. The BPNN parameters include a number of hidden layers, a number of hidden neurons, an activation function, learning rate, momentum, etc. All of these parameters have significant impact on neural network performance. In most cases, one hidden layer is sufficient for computing arbitrary decision boundaries that can approximate any continuous function. Therefore, the number of hidden layers is usually set at 1. After performing some trials the final number of neurons in the hidden layer can be determined.

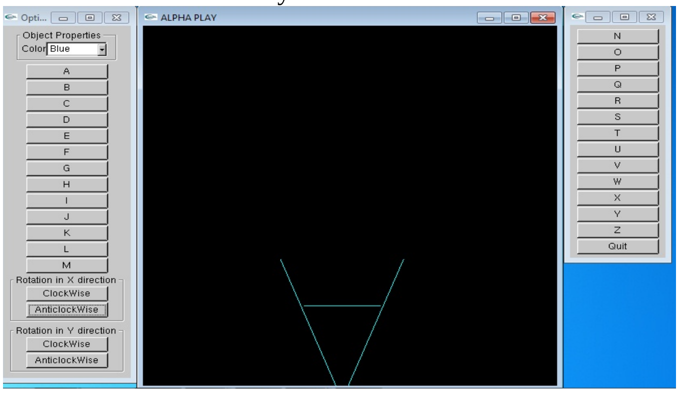

Figure 2.1.5 Anti clock Wise Rotation of the Alphabet

Figure 2.1.6 Scaling of the Alphabet

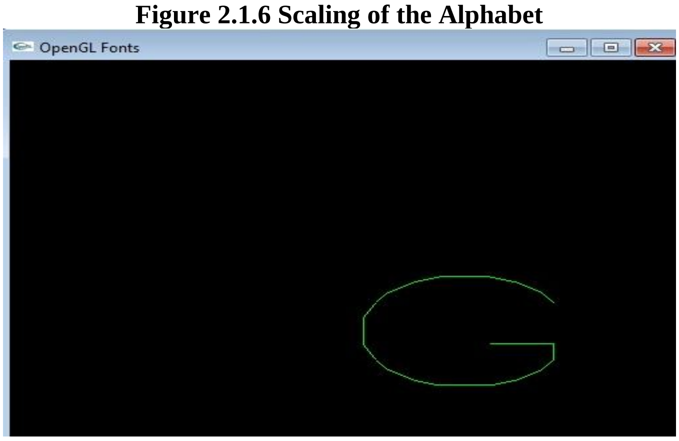

Figure 2.1.7 Translation of the Alphabet

Back propagation algorithms are generally termed to be slow in learning as there are many back-propagated values into the hidden layer. The back propagation algorithm used in this system uses a Resilient Propagation (RPROP) algorithm [7]. The RPROP adaptation of the weight-step is not "blurred" by gradient behavior; instead, each weight has an individual evolving update-value. [8] The weight-step is only determined by its update-value and the sign of the gradient. [9] RPROP requires no parameter tuning and learning. Adaptation is only affected by the sign of the partial derivative, learning is equally spread over the network and so RPROP is a fast learner. [10] The number of learning steps is significantly reduced due to a local learning scheme.

The System GUI for the neural network is shown in figure 2.2.1. The learning step and recognition process are illustrated in the figure 2.2.2.

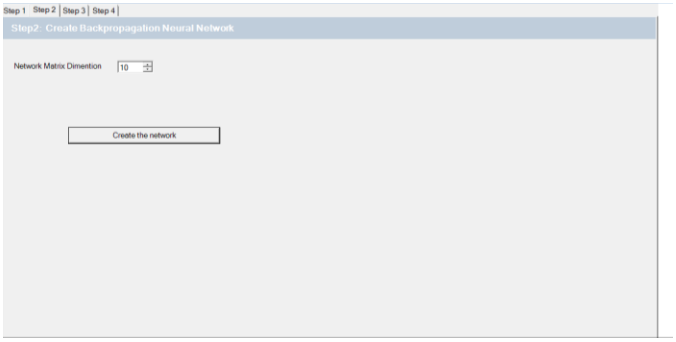

Figure 2.2.1 BPNN GUI

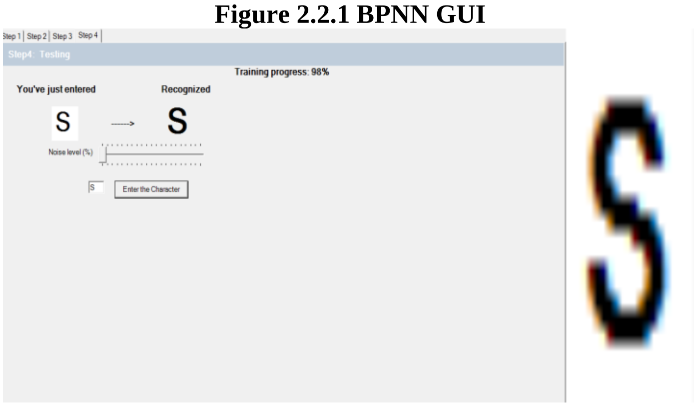

Figure 2.2.2 Recognized Alphabet

3. Experimental Result

The system was tested on English alphabets grouped into two categories; the test set 1 was framed with the entire set of alphabets – all the 26 alphabets, the test set 2 was a random pick of any 10 alphabets. Each test set was tested in three different variations applied to. The levels of transformations were taken as standard transformation, medium transformation and high transformation. The standard transformation would be up to 30% transformations applied on the chosen character, all the transformations were applied one by one and only once to the chosen alphabet. The medium transformation was 60% and high transformation was up to 90%.

The table 3.1 shows the results of recognition after application of each transformation on the alphabet and the total recognition score. The tables 3.2 and 3.3 show the recognition scores for the medium and high transformations applied.

The tabulation result allows making a clear conclusion that the transformations such as scaling and translation, does not affect the recognition process. The column data for translation and scaling proves evidential for the above conclusion.

The table 3.3 above shows the invariance for transformation lesser than 90%. The result analysis has identified that the shearing and projection of the alphabet has not proven to be a good take for the recognition. When an alphabet is put forth for a transformation of shearing or projection the structural frame of the alphabet changes, it might be the possible reason for the lesser recognition accuracy when compared to the other transformations applied.

The graphical results for the different levels are shown in figure 3.1.

The figure 3.1 depicts the transformations applied at the standard level, i.e. When translation is applied its maximum value is 30%, it's evident from the graph the shearing recognition score is the lowest of all. The figure 3.2 shows the graph for medium level of transformations.

Table 3.1 Level - standard -transformations applied lesser than 30%

	Translation	Scaling	Rotation X –axis	Rotation Y-axis	Shearing	Projection	Total Recognition score
All 26 alphabets	100	100	99	99	87	86	95.17
Radom data pick(10 at a time)	100	100	100	100	88	89	96.17

Table 3.2 Level - Medium transformations applied lesser than 60%

	Translation	Scaling	Rotation X -axis	Rotation Y-axis	Shearing	Projection	Total Recognition score
All 26 alphabets	100	100	97	95	80	82	92.33
Radom data pick(10 at a time)	100	100	95	90	77	80	90.33

Table 3.3 Level - High transformations applied lesser than 90%

	Translation	Scaling	Rotation X –axis	Rotation Y-axis	Shearing	Projection	Total Recognition score
All 26 alphabets	100	100	92	89	75	70	87.67
Radom data pick(10 at a time)	100	100	90	92	79	72	88.83

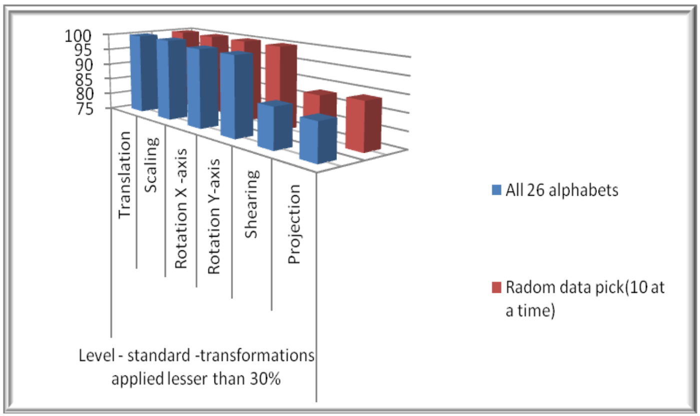

Figure 3.1 Standard Transformations

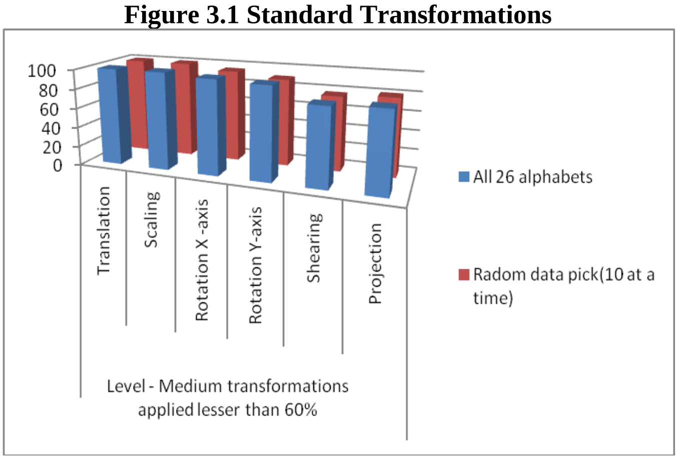

Figure 3.2 Medium Transformation

The graph elucidates lucidly the recognition scores for standard and high is better for the Random dataset (RDS). The All alphabet dataset happens to be greater than RDS for the medium dataset. The better recognition in RDS might be possibly due to the reason that the dataset comprises of only lesser data in process, and it is random in nature. The advantage would also be the same; randomness in data pick is the way a real time system would operate.

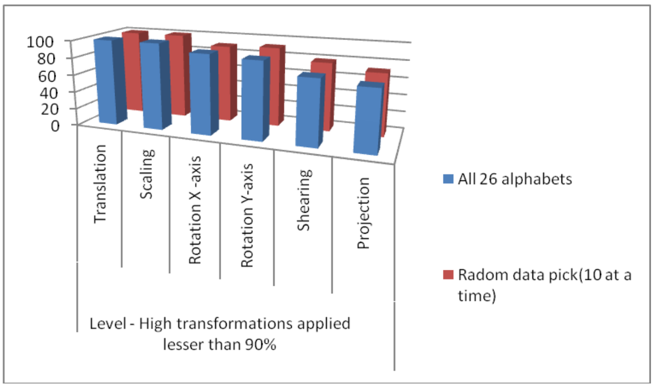

Figure 3.3 High Transformation

The figure 3.3 shows the high transformation with the maximum value being 90%.

The comparison of the recognition scores with the three levels of transformations is shown in the figure 3.4.

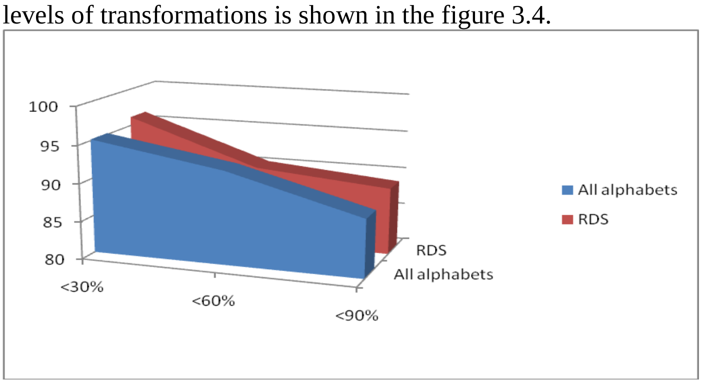

Figure 3.4 Transformation Comparison Amongst Data Sets

4. Conclusion and Future Work

The interface of the system is user friendly. The recognition process of the system is efficient with the steps involved in developing it. The system has explored the possibility of training on synthetically generated font data and the performance was good. The system can recognize distorted characters. The generation and manipulation of characters is two dimensional. Rotation of certain characters had to be blocked for visibility purposes, for ex: “Z” and “N” has a coincidental feature when rotated to a particular angle. The system can now only recognize English synthetically generated font characters. The system could accommodate Z-axis transformations. The same system can be enhanced to support recognition of alphabets of other languages too.

Bibliography

[1] S. Gheorghita, R. Munteanu, M. Enache, “Study of Neural Networks to Improve Performance for Character Recognition”, in Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference, Cluj-Napoca, 2012.

[2] Huy Phat Le, GueeSang Lee, “Noise Removal From Binarized Text Images”, in Computer and Automation Engineering (ICCAE), 2012 The Second National Conference, Gwangju, 2010.

[3] Yanhua Shao, “Research About Neural Network Approach”, in Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference, Guiyang, 2011.

[4] Bishop M Christopher, “Neural Networks: A Pattern Recognition Perspective”, Introduction to Neural Networks, Neural Computing Research Group, Aston University, Birmingham, 1996.

[5] Li Fuliang, Gao Shuangxi, “Character Recognition System Based on Back Propagation Neural Network”, in Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference, Shijiazhuang, 2010.

[6] Heam Donald, Baker M Pauline, “Computer Graphics With OpenGL”, Third Edition, Prentice Hall, 2003.

[7] Martin Riedmiller, Heinrich Braun, “A Direct Adaptive Method For Faster Back Propagation Learning: The RPROP Algorithm”, in Neural Networks, IEEE International Conference, Karlsruhe, 1993.

[8] Christian Igel, Michael Husken, “Improving the Rprop Learning Algorithm”, Proceedings of the Second International Symposium on Neural Computation, NC’2000, pp. 115–121, ICSC Academic Press, 2000

[9] G.D. Magoulas, M.N. Vrahatis, and G.S. Androulakis, “Improving the Convergence of the Backpropagation Algorithm Using Learning Rate Adaptation Methods,” Neural Computation, 11, 1999, pp. 1769-1796.

[10] M. Riedmiller, and H. Braun, “A direct adaptive method for faster backpropagation learning: The RPROP algorithm”, Proceedings International Conference on Neural Networks, San Francisco, CA, 1993, pp. 586-591.