INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

IMPLEMENTATION OF NEURAL NETWORK FOR CHARACTER RECOGNITION

CHETAN S. SURADKAR[1], PROF. L. S. KHEDEKAR[2]

Department of Computer Science and Engineering, RSCE, Maharashtra, India.

Accepted Date: 05/03/2015; Published Date: 01/05/2015

Abstract: This paper describes a NEURAL NETWORK based technique for feature extraction applicable to segmentation-based word recognition systems. The proposed system extracts the geometric features of the character contour. The system gives a feature vector as its output. The feature vectors so generated from a training set, were then used to train a pattern recognition engine based on Neural Networks so that the system can be benchmarked. Since, an attempt was made to develop a system that used the methods that humans use to perceive handwritten characters. Pattern recognition can be used to model human perception. The mathematics that Pattern recognition requires is extremely fundamental. Thus, any algorithm developed using Pattern recognition would require relatively simple and short calculations. Due to simplicity of calculations, they can be implemented on any hardware or software platform without too much concern for computing power. In this paper first part is about introduction to character Recognition. Then next part giving short introduction to Neural network implementation for image processing using MATLAB.

Keywords: MATLAB, Neural Network

Corresponding Author: MR. CHETAN S. SURADKAR

Access Online On:

www.ijpret.com

How to Cite This Article:

Chetan S. Suradkar, IJPRET, 2015; Volume 3 (9): 125-130

PAPER-QR CODE

INTRODUCTION

There are two distinct areas of research concerning Handwritten Character Recognition

(1) Off-line Character recognition, and

(2) On-line Character Recognition.

Here we use Off-line method. The Off-line character description data provides much more information for recognition. This is due to the reason that these images are usually of characters that were written earlier and later converted to digital format using a digital scanner.

The challenge with Off-line character recognition is the development of a system that can recognize these characters. This requires a system that requires very simple and short calculations. If not, the time taken to recognize the characters will render the system useless. Thus, the method selected for this was Pattern recognition. In this, we present the handwritten character as part from common application Character and of recognizing the alphanumeric characters.

The feature extraction algorithm applied to the characters is novel and leads to a very fast recognition. Although a number of good recognition algorithms have been proposed for handwritten character recognition, the achieved performance is still far from those of human beings in context free handwritten character recognition. The major obstacles have been the different handwriting styles and changeable writing conditions.

These two aspects make handwritten characters extremely variable. Observing that the different writing styles or individual writing style of each writer is very important problem in handwritten recognition. The handwritten identification requires the use of optical mouse as a scanner application for offline conversion. It is the examination of the design, shape and structure of handwritten character. In this handwriting identification, writer are required to write the same fixed text or also called text dependent.

In this technique to determine whether the character wrote is match with the character of the database was presented. This technique is based on computing the main Euclidean distances between these two characters. Then the distance transformation is used to classify whether the handwriting belongs to the same or different character based on the main distances.

MODES IN CHARACTER RECOGNITION

The basic mode of the system is given below. This system is divided in two parts.

One is training mode and other is recognition mode. During training mode database is prepared and store the results. And during recognition mode sample character is compared with stored patterns in database and computes the result. The input data is given to the system by mouse by drawing character in Paint window or by other form, but must be in digital form.

The overall analysis can be divided into four stages:

1. Image acquisition for database preparation

2. Preprocessing

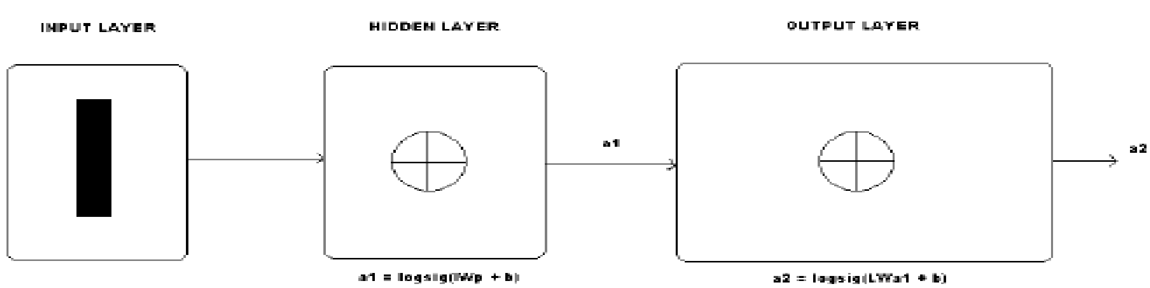
3. Feature extraction, and

4. Identifying character that is recognition.

NEURAL NETWORK

The network receives the 35 Boolean values as a 35-element input vector. It is then required to identify the letter by responding with a 26-element output vector. The 26 elements of the output vector each represent a letter. To operate correctly, the network should respond with a 1 in the position of the letter being presented to the network. All other values in the output vector should be 0.

In addition, the network should be able to handle noise. In practice, the network does not receive a perfect Boolean vector as input. Specifically, the network should make as few mistakes as possible when classifying vectors with noise of mean 0 and standard deviation of 0.2 or less.



ARCHITECTURE

The neural network needs 35 inputs and 26 neuronsin its output layer to identify the letters. The network is a two-layer log-sigmoid/log-sigmoid network. The log-sigmoid transfer function was picked because its output range (0 to 1) is perfect for learning to output Boolean values.

The hidden (first) layer has 10 neurons. This number was picked by guesswork and experience. Ifthe network has trouble learning, then neurons can beadded to this layer.

The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0's. However, noisy input vectors can result in the network's not creating perfect 1's and 0's. After the network is trained the output is passed through the competitive transfer function. This makes sure that the output corresponding to the letter most like the noisy input vector takes on a value of 1, and all others have a value of 0. The result of this post processing is the output that is actually used.

INITIALIZATION

Create the two-layer network with newff.net= newff(alphabet,targets,10,{'logsig','logsig'});

TRAINING

To create a network that can handle noisy input vectors, it is best to train the network on both ideal and noisy vectors. To do this, the network is first trained on ideal vectors until it has a low sum squared error.

Then the network is trained on 10 sets of ideal and noisy vectors. The network is trained on two copies of the noise-free alphabet at the same time as it is trained on noisy vectors. The two copies of the noise-free alphabet are used to maintain the network's ability to classify ideal input vectors.

Unfortunately, after the training described above the network might have learned to classify some difficult noisy vectors at the expense of properly classifying a noise-free vector. Therefore, the network is again trained on just ideal vectors.

This ensures that the network responds perfectly when presented with an ideal letter.

USE OF MATLAB WORKING COMMAND

Compare

Compare model output and measured output

SYNTAX

compare(data,m);

compare(data,m,k) compare(data,m,k,'Samples',sampnr,'InitialState',init,' OutputPlots',Yplots) compare(data,m1,m2,...,mN)

compare(data,m1,'PlotStyle1',...,mN,'PlotStyleN') [yh,fit,x0]= compare(data,m1,'PlotStyle1',...,mN,'PlotStyleN',k)

DESCRIPTION

Data is the output-input data in the usual iddata object format data can also be an object with frequency-response data. Compare computes the output yh that results when the model m is simulated with the input u. m can be any idmodel or idnlmodelmodel object. The result is plotted together with thecorresponding measured output y. The percentage of the output variation that is explained by the model

fit = 100*(1 - norm(yh - y)/norm(y-mean(y))) is also computed and displayed. For multiple-outputsystems, this is done separately for each output. Forfrequency-domain data (or in general for complexvalued data) the fit is still calculated as above, butonly the absolute values of y and yh are plotted.

When the argument k is specified, the k step-ahead prediction of y according to the model m are computed instead of the simulated output. In the calculation of the model can us outputs up to time : , , … (and inputs up to the current time t).

The default value of k is inf, which gives a pure simulation from the input only. Note that for frequency-domain data, only simulation (k = inf) is allowed, and for time-series data (no input) only prediction (k not inf) is possible.

CONCLUSION

The above-discussed method has all the characteristics that are required for an off-line handwritten character system. That is simplicity and shortness of calculations. The method was found to be extremely reliable in preliminary dataset. The method can easily be applied to any application that requires handwritten character recognition, regardless of its computing power.

This is due to the low computational requirement. Thus, the proposed algorithm can be implemented on any type of software platform. The method can also be applied to an on-line system if the coordinate data sent into the system can be sent in as a time ordered sequence of data.

REFERENCES

1. N. Sharma, U. Pal, F. Kimura, S. pal, "Recognition of Offline Handwritten Devnagari Characters using Quadratic Classifier", ICCGIP 2006, LNCS 4338, 2006.

2. R. Kapoor, D. Bagai and T.S. Kamal, "Representation and Extraction of Nodal Features of DevNagri Letters", Proceedings of the 3rd Indian Conference on Computer Vision, Graphics and Image Processing.

3. Reena Bajaj, Lipika Dey, and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connection classifiers", Sadhana, Vol.27, part. 1, 2002

4. Bellili, M. Gilloux, P. Gallinari and An MLP-SVM combination architecture for o²ine handwritten digit recognition: reduction of recognition errors by support vector machines rejection mechanisms, Int. J. Document Analysis and Recognition, 5(4): 2003.