Advances in Computational Research
ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 4, Issue 1, 2012, pp.-38-41
Available online at http://www.bioinfo.in/contents.php?id=33

OFF-LINE MIXED DEVNAGRI NUMERALS RECOGNITION USING ARTIFICIAL NEURAL NETWORK

PATIL S.B.[1*] AND SINHA G.R.[2]

[1]Department of Electronics & Telecommunication, Shri Shankaracharya Technical Campus, Bhilai, CG, India.
[2]Department of Engg. & Tech., Shri Shankaracharya Technical Campus, Bhilai, CG, India.
*Corresponding Author: Email- patilsandeepb1212@gmail.com, drgrsinha@ieee.org

Received: February 21, 2012; Accepted: March 06, 2012

Abstract- In this paper we are collecting 100 Devnagri numerals from 10 different persons belonging to three different states of India. The data collected in a plane paper is scanned in the form of a bit map image. The collected data has to undergo the preprocessing steps first and then some morphological operations like opening, edge detection, dilation, hole filling and numerals detection are performed. After numerals detection the boundary of the numerals in row, are calculated and stored in the database. Calculating the total number of numerals in a row, the individual width and height of each numeral are measured. The features can be extracted by three steps; first the extreme coordinates of the numerals can be measured, then grabbing the numerals into grids and finally numerals digitization. Digitized numerals are then further trained with multilayer neural network. The proposed research work gives us 100 % accuracy and hence recognized all 100 numerals correctly.
Keywords- Devnagri, numerals detection, edge detection, dilation, grids, neural network

Citation: Patil S.B. and Sinha G.R. (2012) Off-Line Mixed Devnagri Numerals Recognition Using Artificial Neural Network. Advances in Computational Research, ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 4, Issue 1, pp.-38-41.

Introduction

The handwriting character recognition system has been inspired from the human capability to recognize any pattern, which is difficult task for normal computer system having computing power of more than billions instruction per second. In this system the input is given in the form of a digital image by using writing pad, optical scanner or digital camera. This input image is processed to extract the information by using various algorithm and technique of digital image processing. This extracted information is then sent to artificial neural network for character recognition process, which has been inspired from the structure of human brain made up of networks of cells known as neuron.

Two classes of recognition systems are usually distinguished: online system for which handwriting data are captured during the writing process, which makes available the information on the ordering of the strokes, and offline system for which recognition takes place on a static image captured once the writing process is over. Neural Nets (NN) and Hidden Markov Model (HMM) [1, 7] are the popular, amongst the techniques which have been investigated for handwriting recognition. Neural network are recently being used in various kind of pattern recognition. In this paper, efforts have been made to develop automatic handwritten Devnagri numeral recognition system with high recognition accuracy. Here we have used neural network is the multi-layer perceptron (MLP) and trained with back-propagation which is the most popular and versatile forms of neural network classifier and is among the most frequently used traditional classifiers for handwriting recognition.

Devnagri script

The Devnagri is a form of alphabet called an abugida, as each constant has an inherent vowel that can be changed with different vowel signs [2-5]. Vowel can be written as independent characters, or by using a variety of diacritical marks which are written above, below, before or after the consonants they belong to. This Devnagri script and Unicode is shown in "Fig. (1)"

Fig. 1- Devnagri Script

Data Acquisition

The handwritten numerals data are collected from 10 different persons belong to three different state of India. The collected data is taken on the plane paper and later on scan it to form a standard bit map image. No restriction was imposed on the content or style of writing, the only exception was the stipulation on the isolation of characters [2-4]. The data acquisition of 100 mixed Devnagri numerals from zero to nine is shown in "Fig. (2)".

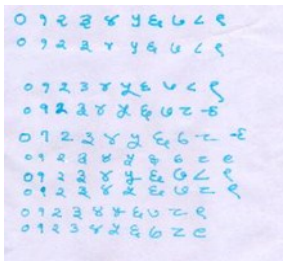

Fig. 2 - Scanned Image

Image Preprocessing

The first step of this processing is to converts the scan text RGB image into gray image and then binary. The binary image is then inverted because when we are applying morphological operations such as edge detection which is applicable to black background i.e. with pixel '0' and white foreground with pixel '1'. The main aim is to do that for simpler and faster calculations for further processing.

Edge Detection

In practice, edge detection is performed in the spatial domain, because it is computationally less expensive and often yields better results. Since edges correspond to strong illumination gradients, we can highlight them by calculating the derivatives of the image. In this numeral recognition process we are using the method of gradient edge detection with Sobel operator [6-9]. Gradient edge detection is the more widely used technique. Here, the image is convolved with only two masks, one estimating the gradient in the x-direction Gx, the other the gradient in the y-direction Gy. The two masks are shown in "Fig. (3)" below. These masks are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one mask for each of the two perpendicular orientations. The mask can be applied separately to the input image, to produce separate measurements of the gradient components in each orientation. These are then can be

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Fig. 3- Sobel convolution mask

Combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by

$$|G| = \sqrt{Gx^2 + Gy^2} \quad \text{..... (1)}$$

The angle of orientation of the edge gives rise to the spatial gradient is given by

$$\emptyset = arctan(Gy|Gx) - 3\pi/4 \quad \text{..... (2)}$$

In this case, orientation '0' is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anti-clockwise from this. This absolute magnitude is the only output the user sees. The two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator as shown in "Fig. (4)".

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

Fig. 4- Pseudo-convolution masks used to compute approximate gradient magnitude

Using this mask the approximate magnitude is given by

$$|G| = |(P1 + 2*P2 + P3) - (P7 + 2*P8 + P9)| + |(P3 + 2*P6 + P9) - (P1 + 2*P4 + P7)| \quad \text{..... (3)}$$

The image after edge detection is shown in "Fig. (5)".

Fig. 5- Edge Detection

Dilation

Dilation is typically applied over hear to gradually enlarge the boundaries of regions of the foreground pixels thus the area of foreground pixels grow in size while holes within this regions be-

comes smaller. The dilation operator takes two pieces of data as input. The first is the image which is to be dilated and the second is a set of coordinate point known as structuring element. The structuring element determines the precise effect of the dilation on the input image.

Mathematically, dilation of A by B is denoted by A $\oplus$ B and is defined as [6-8, 10].

$$A \oplus B = \{Z|(B)z \cap A \neq \emptyset\} \quad \text{..... (4)}$$

Dilation operation for the image under analysis is shown in "Fig. (6)".

Fig. 6- Image after Dilation

Hole filling

Hole filling is determined by selection of marker and mask images. Here, we choose the marker image is f_m to be '0' everywhere except on the image border, where it is set to '1' [6, 10-11].

$$\begin{cases} 1 - f(x,y) & if(x,y) is\ on\ the\ border\ of\ 'f' \\ 0 & otherwise \end{cases} \quad ..(5)$$

The effect of hole filling is shown in "Fig. (7)".

Fig. 7- Image after hole filling.

Character Detection

From the above mentioned steps the character detection is quite simple. The algorithm search from left to right for white pixels starting from left top corner of the area specified for writing. A trace of the pixels is the indication of presence of a character.

Calculating the number of rows

The algorithm searches for the presence and absence of white pixels going from top to bottom. The continuous absence of white pixels could be a gap between two rows. To make sure whether it is a gap, algorithm searches from left to right against every black pixel, if there is no trace of white pixel for the entire row, the gap is confirmed. In this way, all the horizontal gaps in the image are traced out and from this number of rows are calculated. After calculating the number of rows, the individual width and height of each numeral is measured.

Feature Extraction

Feature extraction consist of three steps: extreme coordinates measurement, grabbing character into grid, and character digitization. The handwritten numerals are captured and are subdivided into a rectangular grid of specific rows and columns. The algorithm automatically adjusts the size of grid and its constituents according to the dimensions of the numeral. Then it searches the presence of character pixel in every box of the grid. The boxes found with character pixels are considered 'on' and the rest are marked 'off'. A binary string of each character is formed locating the 'on' and 'off' boxes and presented to the neural network input for training and recognition purposes. The total number of grid boxes represented the number of binary inputs. A 7x5 grid thus resulted in 35 inputs to the recognition model [7-8, 17]. The "Fig. (8)" below shows the feature extraction.

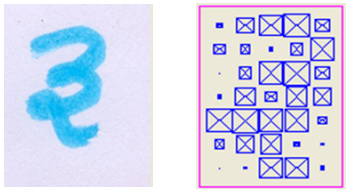

Fig. 8- Feature Extraction for Numeral 3.

Neural Network in numerals recognition

Neural network classifier exhibit powerful properties and they have been used in handwriting recognition particularly with digits, isolated characters, and words in small vocabularies. In this work, multilayer feed-forward back propagation neural network has been implemented. The neural network had three layers: an input layer consisting of 35 nodes, a hidden layer consisting of 20 nodes, and an output layer 10 nodes one for each numerals [11-15]. The "Fig. (9)" shows the three layer artificial neural network for numeral recognition. The network uses back-propagation in addition to bias weight and momentum.

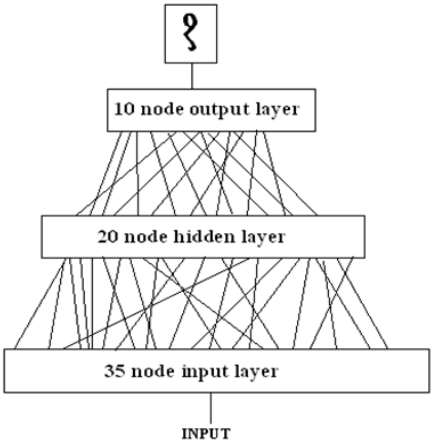

Fig. 9- Three layer artificial neural network for numeral recognition.

The network receives the 35 Boolean values as a 35-element input vector. It is then identify the numerals by responding with a 10-element output vector. The 10-element output vectors each represent a numeral. The network is a two-layer log-sigmoid/log-

sigmoid network. To operate correctly, the network should respond with a '1' in the position of the numerals being presented to the network, all other values in the output vector should be '0'. [9-11,13-18]. The log-sigmoid transfer function was picked because its output range (0 to 1) is perfect for learning Boolean values and also the network is capable to handle noise.

Training

To create a network that can handle noisy input vector it is best to train the network on both ideal and noisy vector. To do this, the network is first trained on ideal vector 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 vector. Unfortunately, after the training described above the network may have learned to classify some difficult noise vector at the expense of property 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 digit.

Experimental Result:

The devnagri 100 numerals from different persons were taken, ten for each class. The numerals were trained with and without noise for a maximum of 5000 epochs. The graphical user interface of the experimental work is shown in "Fig. (10)". The Table 1 shows the percentage of the recognition result for all the 100 devnagri numerals.

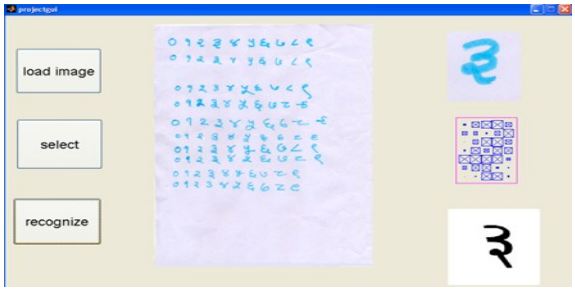

Fig. 10- Graphical User Interface for Numeral recognition

Table 1- Shows recognition accuracy

Applied Devnagri Numerals	Number of Epochs	Numerals recognize correctly	% recognition
०	5000	10	100
१	5000	10	100
२	5000	10	100
३	5000	10	100
४	5000	10	100
५	5000	10	100
६	5000	10	100
७	5000	10	100
८	5000	10	100
९	5000	10	100

Conclusion

In this work the Devnagri data acquired from the different persons were trained for several times on various input vectors. Training a network on different sets of noisy vector forced the network to learn how to deal with noise. This technique gives 100% result if it is trained for 5000 epoch.

Reference

[1] Sandeep B. Patil, Sinha G.R. and Vaishali S. Patil (2011) *Second international conference on Emerging Applications of Information technology*, 185-189.
[2] Pradeep J., Srinivasan E. and Himavathi S. (2011) *International conference on computer communication and Electrical Technology*, 59-63.
[3] Yuefeng Chen, Chunlin Liang, Lingxi Peng and Xiuyu Zhong. (2010) *International conference on Computer Applicationand System Modeling,* 134-139.
[4] Anita Pal and Dayashankar Singh (2010) *International Journal of computer science and application*, 1(2), 141-144.
[5] Rajashekararadhya S.V. and Vanaja Ranjan P. (2008) *J. of Theoretical and Applied information Technology,* 97-101.
[6] Rafel C. Gonzalez and Richard E. Woods (2006) *Digital Image Processing Using Matlab*.
[7] Muhammad Faisal Zafar, Dzulkifli Mohamad and Razib M. Othman (2005) *World academy of science, Engineering and technology*, 232-237.
[8] Castleman K.R. (1996) *Digital Image Processing 2nd Edition*.
[9] Jarek M. Zarda (1999) *Introduction to artificial neural system*.
[10] Anil K. Jain (1995) *Fundamentals of Digital Image Processing*.
[11] Rabiner L.R. (1989) *A tutorial on hidden markov models and selected application in speech recogniton, IEEE,* 257-286,
[12] Huiqin Lin, Wennuan Ou and Tonglin Zhu (2011) *Sixth International conference on Image and Graphics,* 118-122.
[13] Ajmire P.E. and Warkhede S.E. (2011) *Advances in information Mining*, 2(2), 11-13.
[14] Siddhartha Choubey, Sinha G.R. and Abha Choubey (2011) *Advances in information technology and Mobile communication,* 147, 422-426.
[15] Dhandra B.V., Benne R.G. and Mallikarjun Hangarge (2010) *IJCA Special Issue on recent trends in image processing and Pattern Recognition*, 155-159.
[16] Siddharth Choubey and Sinha G.R. (2011) *International conference on Network and Computer Science*, 5, 26-30.