An Automated Bengali Sign Language Recognition System Based on Fingertip Finder Algorithm

[1]Angur M. Jarman, [2]Samiul Arshad, [3]Nashid Alam and [4]Mohammed J. Islam

ISSN: 2186-0114
http://cennser.org/IJEI

ARTICLE HISTORY

Received: 27 April 2015
Revised: 7 July, 2015
Accepted: 21 July, 2015

Published online: 22 July 2015

Vol.4, No.1

Abstract

This paper presents a new algorithm to identify Bengali Sign Language (BdSL) for recognizing 46 hand gestures, including 9 gestures for 11 vowels, 28 gestures for 39 consonants and 9 gestures for 9 numerals according to the similarity of pronunciation. The image was first re-sized and then converted to binary format to crop the region of interest by using only top-most, left-most and right-most white pixels. The positions of the finger-tips were found by applying a fingertip finder algorithm. Eleven features were extracted from each image to train a multilayered feed-forward neural network with a back-propagation training algorithm. Distance between the centroid of the hand region and each finger tip was calculated along with the angles between each fingertip and horizontal x axis that crossed the centroid. A database of 2300 images of Bengali signs was constructed to evaluate the effectiveness of the proposed system, where 70%, 15% and 15% images were used for training, testing, and validating, respectively. Experimental result showed an average of 88.69% accuracy in recognizing BdSL which is very much promising compare to other existing methods.

Keywords: *Bengali Sign Language, Feed-forward Neural Network, Fingertip Finder Algorithm, Linguistic Sign Language, Image Processing.*

I. Introduction

Sign language is a communication medium by which people express their opinions and feelings. Communication between hearing impaired and mute community is not easy. Mute people need to learn sign language for effective communication. For this reason, it is necessary to develop an interactive, automatic and robust system that can be used as an interpreter among those communities. Bengali Sign Language is made from several core one-handed static sign alphabet and number signs. Many researchers have been working on sign language recognition systems for various sign languages. A system developed by Pavel et al. [1] analyzes video clips of different gestures of sign languages taken as input and gives audio output. Angles of different parts of the hand with body were calculated manually by analyzing captured images from input video frames and stored manually in a database with corresponding audio meanings. The system then matches the calculated angles with the database and the corresponding audio file is played in the speaker of the pocket PC. One limitation of this system was slower frame rate,

AUTHORS INFO

Author 1
e-mail: jarman.cse@gmail.com
Address: Department of Computer Science and Engineering, Sylhet Engineering College, Sylhet, Bangladesh.

Author 2
e-mail: samiularshad@gmail.com
Address: Department of Computer Science and Engineering, Sylhet Engineering College, Sylhet, Bangladesh.

*Author 3**
e-mail: annanya_cse@yahoo.co.uk
Address: Department of Computer Science and Engineering, Sylhet Agricultural University, Sylhet, Bangladesh.

Author 4
e-mail: jahir-cse@sust.edu
Address: Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet, Bangladesh.

*Corresponding author
e-mail: annanya_cse@yahoo.co.uk

because the actual frame rate of the animation is too quick to interpret the sign language, on the other hand database was created manually which is time consuming and subjective.

An intelligent system developed by Rahman et al. [2] is more complicated. Real-time photos of hand wearing a glove containing different dots in each finger were taken as inputs. Analyzing the dots of the graphics in the image is needed to understand what sign has been shown and then pre-recorded audio and written form were shown as output. The technique involved clustering of the dots and mapping the results of this clustering to pre-defined tables. This system was limited to Bengali numerals 1-10. Begum and Hasanuzzaman [3] found a way to develop a system for recognition of Bengali vowels and number signs using PCA (Principal Component Analysis) which was tested for 6 Bengali vowels and 10 numerals only.

Human machine interface [4] was developed for a system named "*Intelligent Assistant*" to understand ten Bengali expressions using Microsoft's Voice Command and Control Engine. It was able to instantly transform the expression into text and then text was matched with previously stored word in its knowledge base. Once the word matches the database, it displays 3D graphical hand images, which are mapped according to the text. They claimed 82% accuracy with only ten words. Research by Karmokar et al. [5] proposed a system in which Negative Correlation Learning (NCL) algorithm was used. They captured coloured image of signs by webcam and then after thresholding reduced the size into 30x33 pixels matrix representation of black and white image which was later used as feature matrix for training the system with NCL algorithm. Artificial Neural Network (ANN) was used to recognize 36 Bengali sign alphabets in another research titled "Recognition of Static Hand Gestures of Alphabet in Bangla Sign Language" by Rahman et al. [6], where the ANN was trained with features of sign alphabet using feed-forward back-propagation learning algorithm, and Logarithmic sigmoid (logsig) function was chosen as transfer function. They used average height of a sign, total number of black pixels in the image and Euclidian Distance of the Centroid from the Origin as feature set. They claimed average accuracy of 80.9%, and used same method in recognition of English sign language [7].

Manar et al. [8] introduced the use of different types of neural networks in human hand gesture recognition for static images and for dynamic gestures. A coloured glove was used for fingertip detection. Thirty features were based on the segmented colour regions, which were taken from the fingertips, and their relative positions and orientations with respect to the wrist and to each other [9].They claimed 79.3 % accuracy in recognizing 30 Arabic Signs (ArSL) using multilayer feed-forward ANN and showed 95.1% accuracy using recurrent neural network.

Principal Component Analysis (PCA) was used to extract features from images of gestures in the work of Henrik and Moeslund [10] and Lamar et al. [11]. Before applying PCA normalizing of images was needed [12]. Normalizing each frame before applying PCA is computationally inefficient when it is used in dynamic gesture recognition [12].

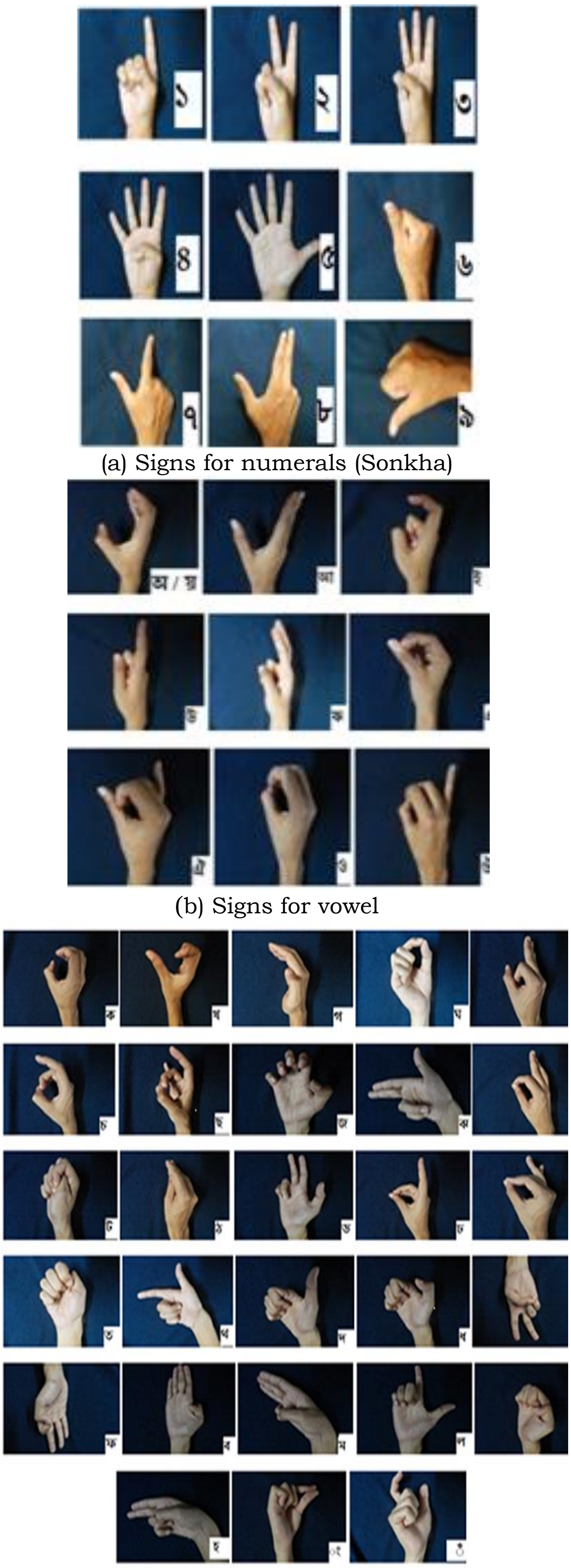
Saengsri et al. [13] used '5DT Data Glove 14 Ultra' which contained 10 sensors on fingers and 4 sensors between the fingers for measuring flexures and abductions respectively in recognizing Thai Sign Language. Kim *et al.* [14] used 'KHU-1' data glove which contained 3 accelerometer sensors, a bluetooth and a controller which extracted features like joints of hand. He performed the experiment for only 3 gestures. Weissmann and Salomon [15] used Cyberglove for measuring thumb rotation, angle between the neighbouring fingers and wrist pitch. These methods were not user friendly and were not very popular, as using gloves restrict the naturalness of the recognition system.

Finite state machine (FSM) was used in a work by Kshirsagar and Dharmpal [16] for the recognition of American Sign Language. They claimed accuracy of 61.33% for dynamic and 62.49% for static sign recognition.

This paper presents a system for recognition of BdSL alphabets using feed-forward ANN with back-propagation training algorithm. In this system, there is no requirement for using gloves or visual marking system. Bare-handed images of signs by individual people were used. Visible fingertips were found by applying the proposed fingertip finder algorithm. MATLAB was used for Image processing and feature extraction. Moreover, neural network toolbox was also used to design, train and test the ANN with the new set of extracted features.

II. Related Works and Methods

This research aims to build a system that will recognize alphabets and numerals of BdSL. One-handed static sign alphabets for Bangla Sonkha (Numbers), swaroborno (Vowels) and Benzonborno (Consonants) are shown in Fig. 1 (a, b, c) respectively.



(a) Signs for numerals (Sonkha)

(b) Signs for vowel

(c) Signs for Bengali consonant

Fig. 1.BdSL signs for Bengali Sign Gestures

The methodology is divided into three stages which can be summarized as: 1) Image acquisition and pre-processing 2) Feature extraction and 3) Sign recognition. In first stage coloured image acquisition is done by using digital camera with static background. The overall working mechanism of the proposed system is depicted in Fig. 2.

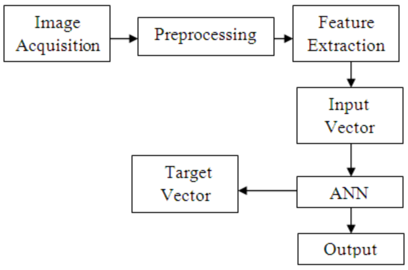


Fig. 2.Stages of proposed methodology

A. *Image acquisition and pre-processing*

Images were resized into 260x260 pixels and converted to binary by using MATLAB function im2bw. Otsu's thresholding method was used for this conversion. Median filtering with a 9x9 neighborhood was used to remove unnecessary noises and morphological operations like clean [17] was used to eliminate any isolated pixel and bridge and diag [10] to remove any discontinuity between white pixels so that our visible finger tip finder algorithm could work properly. The picture was cropped to show only the hand region by invoking MATLAB function: regionProps(binaryImage, 'BoundingBox') [18] to find object properties in the binary image. This function contains parameter named BoundingBox which was structured as [x y w h]where x denotes the column/ vertical co-ordinate, y denotes the row/ horizontal co-ordinate of the top-left corner of the bounding box and w,h are the width and height of the rectangle respectively. An extra row on top side and extra column on both left and right side were added for the proposed VisibleFingertipFinder algorithm to avoid incorrect identification. The final rectangle that was used in cropping the hand region with MATLAB imcrop(binaryImage, rectangle) function is-

```
rectangle=[x-1 y-1 w+2 h]
```

The result of cropping is shown in Fig. 3.

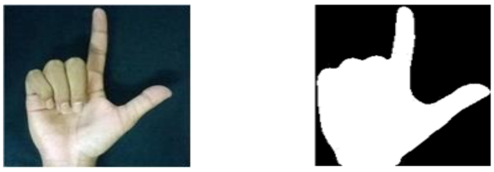


Fig. 3.Original image and cropped image

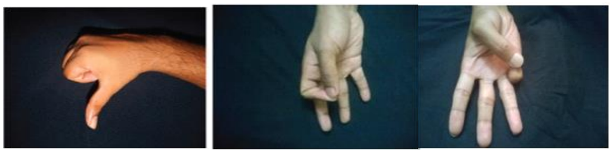
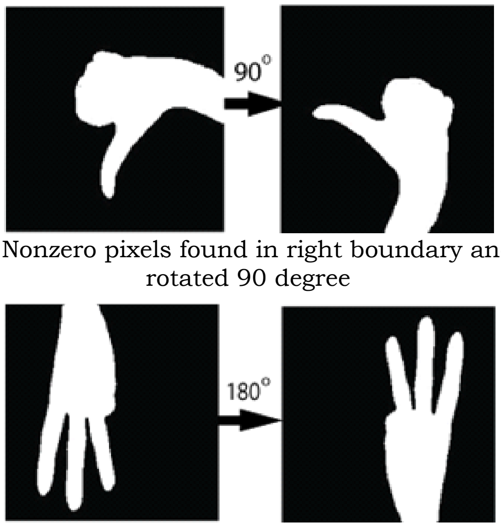


Fig. 4.Some BdSL signs which needs rotation



a)Nonzero pixels found in right boundary and rotated 90 degree

b) Nonzero pixels found in top boundary and rotated 180 degree

Fig. 5.Image rotation

Images were also rotated before cropping to make the fingers aligned vertically. Some BdSL signs which were needed to be rotated are shown in Fig. 4.The angle of rotation was determined by finding the existence of hand-wrist by searching 15 consecutive nonzero pixels at the bottom of the image, right or top border row. In case of getting 15 consecutive nonzero pixels in the bottom border of the image, no rotation was performed; otherwise images were rotated at 90 or 180 degree clockwise, as shown in Fig. 5.

B. Feature Extraction

Feature extraction stage was implemented by finding the points of visible fingertips and the centroid of hand region. Based on these points, ten features were extracted from each image. These features include the distances from centroid to each visible fingertips as well as angles between each of the fingertips and horizontal X axis crossing the centroid, as shown in Fig. 6. Area covered by the sign was calculated by the total number of nonzero elements in the binary image. These features were grouped into one vector:

V= [angle1, distance1, angle2, distance2, angle3, distance3, angle4, distance4, angle5, distance5, area]

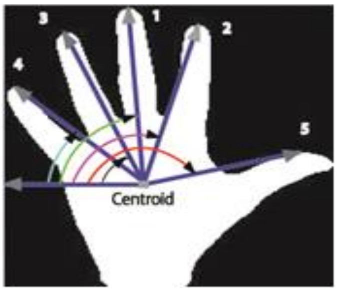


Fig. 6.Features (Fingers are numbered according to height order)

C. Visible Fingertip Finder

The proposed fingertip finder algorithm's flow chart is shown in Fig. 7.

Every pixel from the top left corner of the binary image was scanned to find the fingertips, upper portion of the image. Such a case is depicted in Fig. 8(a), where there are horizontally continuous white pixels (binary 1) with black pixel (binary 0) in both the left and right indexes. Afterwards, both upper and lower indexes were checked. If pixels of upper indexes of continuous pixels were black but pixels of lower indexes were nonzero elements, then one finger was counted and this process was continued till the end of the last column of the binary image. Al least 3 continuous white pixels are checked because of the rounding shape of fingertips.

There are some cases where not all fingers are visible, and images are taken from the side. In that case, invisible fingertips will not be found (Fig. 8(b)). There are very rare cases where a single nonzero pixel appears on the upper indexes of continuous white pixels. In that case fingertips will also be found (Fig. 8(c)).

D. Finding Centroid

The centroid of a polyhedron was simply the average of the respective coordinates of all the vertices of the polyhedron [7]. For example, if the coordinates are (x_1, y_1), (x_2, y_2), (x_3, y_3) . . . (x_n, y_n) then the centroid would be-

$$X_C = \frac{x_1 + x_2 + x_3 + ... + x_n}{N} \quad \dots\dots\dots\dots \quad (1)$$

$$Y_C = \frac{y_1 + y_2 + y_3 + ... + y_n}{N} \quad \dots\dots\dots\dots \quad (2)$$

The centroid of an image was calculated using the following relations:

If x_i represents the X coordinate of each boundary pixel of the hand region and Y_i represents the Y coordinate of each boundary pixel of the hand region and N represents the total number of boundary points then centroid of the hand region is (X_c, Y_c). The centroids were marked as 'o' in Fig. 9.

E. Angle Calculation

The angle between vector from centroid to visible fingertip point and horizontal X axis that crosses centroid was calculated by MATLAB function atan2 [19]. The value returned by this function represents the angle in radians, which was later converted into degree.

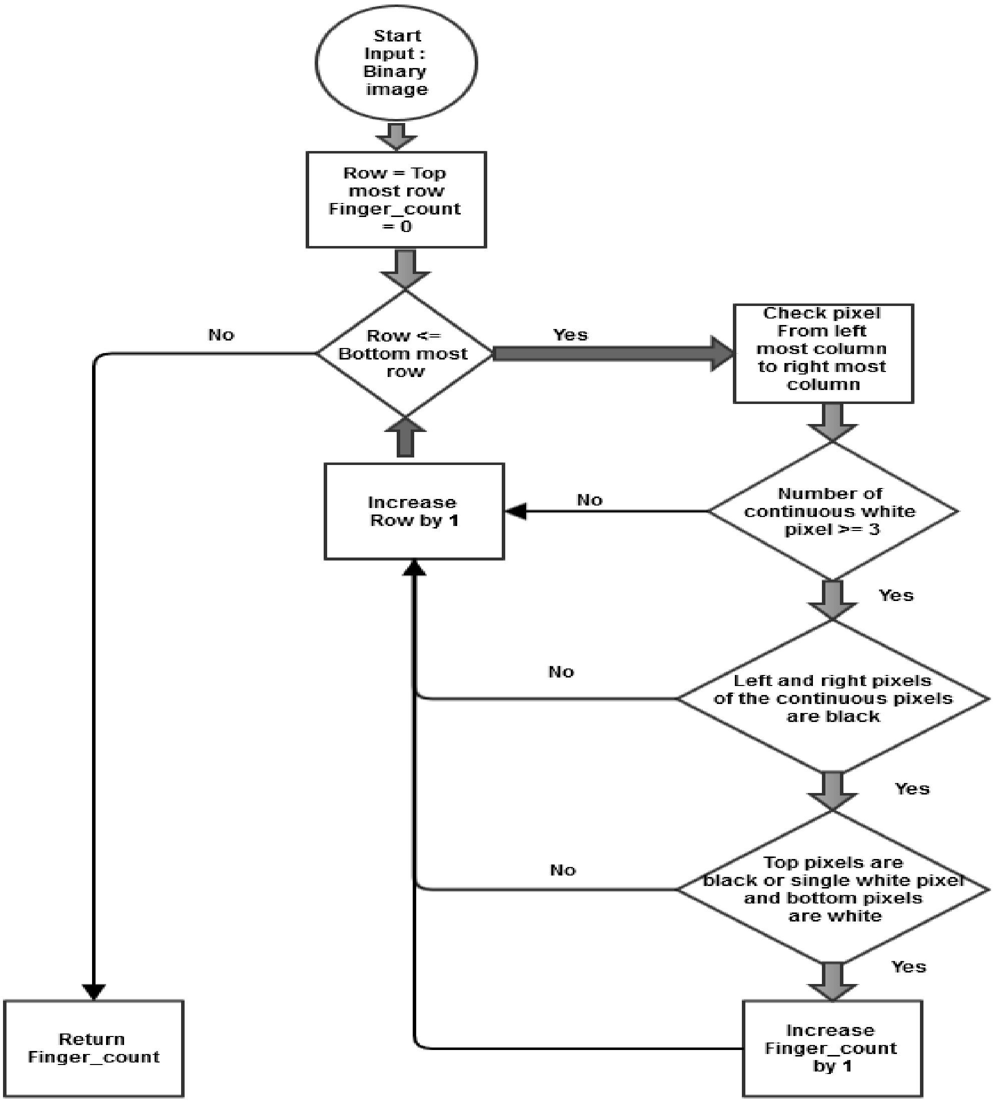
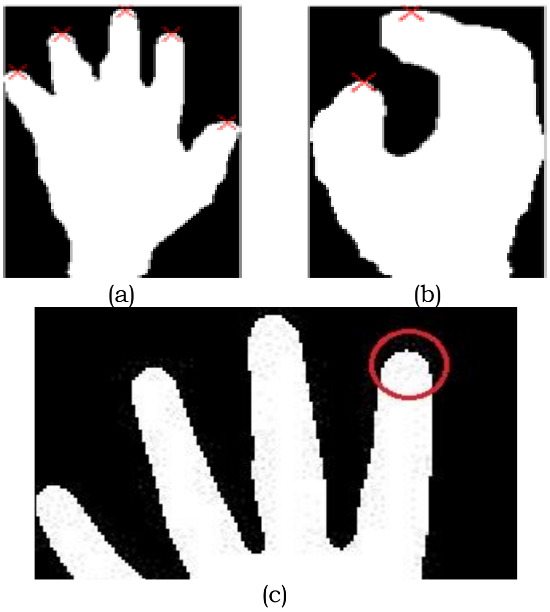


Fig. 7.Flow chart of Visible fingertip finder algorithm.

Fig. 8.Finger tips detection



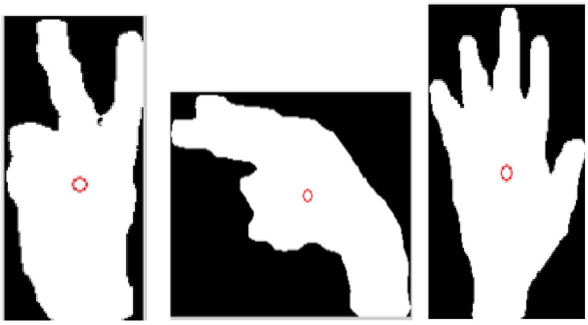


Fig. 9.Centroids finding in the region covered by hand

F. Distance Calculation

Distance between two points in the plane with coordinate (x, y) and (a, b) were calculated using equation (3) by applying Euclidean formula [20].

$$dist((x, y),(a, b)) = \sqrt{(x-a)^2 + (y-b)^2} \ \dots \dots \ (3)$$

If coordinate of visible finger tip is (X, Y) and coordinate of centroid is (X_c, Y_c) then distance between the two points will be

$$distance = \sqrt{(X - X_C)^2 + (Y - Y_C)^2} \ \dots \dots \ \dots \dots \ (4)$$

G. Area Calculation

The hand-area in the image was calculated by counting total number of white pixels. To do this, we traversed the top row to the bottom row of the image and counted the total number of nonzero pixels in each row.

III. Analysis of Experimental Result

Multilayered feed-forward neural networks were used with back-propagation training algorithm to identify the Bengali sign gesture, Fig. 10.

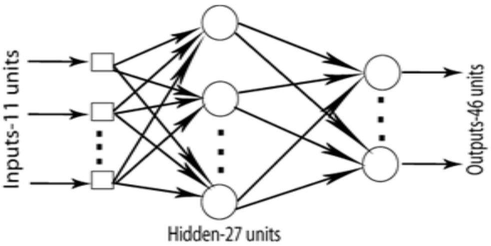


Fig. 10.ANN architecture

Each sign of the alphabet was represented by a vector containing 11 features. Input layer had 11 unit inputs and output layer was made of 46 units, as BdSL sign language have 46 static signs. There is no ideal rule for choosing the number of hidden nodes in hidden layer. Using trial and error after doing an exhaustive search, we have found 27 nodes in hidden layer gives the best performance showed by TABLE I.

TABLE I. Performance evaluation using different number of nodes in hidden layer.

Number of hidden layer used	Recognition rate
10	76.81%
17	85.50%
20	84.05%
25	86.95%
27	88.69%
29	83.78%
33	87.03%
35	78.92%

A database containing 2300 images of Bengali sign was constructed to evaluate the effectiveness of the proposed system that contains 50 sample images of each sign from different individuals, taken by a 16 megapixels Nikon coolpix L820 digital camera with frame size 4608x3456.

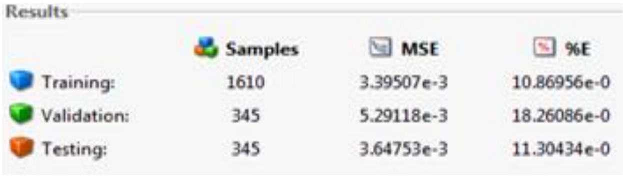
Features of 2300 images were used in training, testing, and validation of ANN. 70% percent of total images were used in training phase, 15% were used to test and 15% were used for validation purpose. Numbers of correct responses of each class were obtained by generating test confusion matrix using MATLAB nntool. The accuracy rates of the proposed recognition technique are provided in TABLE II.

Total 345 sample images were used for testing purpose. Average recognition rate was calculated as following equation-

$$Average\ recognition\ rate = \frac{\sum No.\ of\ Correct\ responses}{No.\ of\ Total\ samples} * 100\% \dots (5)$$

Average recognition rate of the proposed system was 88.69%.

Mean Squared Errors (MSE) and Percent Errors (%E) in training, validation and test phases of the proposed system are shown in Fig. 11.



Results

	Samples	MSE	%E
Training:	1610	3.39507e-3	10.86956e-0
Validation:	345	5.29118e-3	18.26086e-0
Testing:	345	3.64753e-3	11.30434e-0

Fig. 11.MSE and %Error in training, validation and testing.

Percentage of correct responses and wrong responses of each 46 classes are represented by the following graph of Fig. 12. MSE indicates the average squared difference between outputs and targets. %E indicates the fraction of samples which were misclassified.

Percentage of correct responses and wrong responses of each 46 classes are represented by the following graph of Fig. 12.

Comparative analysis of Bengali sign recognition techniques is given by following TABLE III.

A. Sensitivity of recognition method: Accuracy of fingertip detection

For the performance evaluation of the fingertip finder algorithm 50 images for each 12 Bengali signs of 50 individuals were tested. Images used to test the algorithm were taken from the database. Test result showed that fingertip finder algorithm could find fingertips accurately of 91% in average which is shown in TABLE IV.

Sensitivity of the recognition system with respect to the accuracy of fingertip finder algorithm is shown by Fig. 13. Accuracy of fingertip detection and recognition method is

TABLE II. System Performance of recognizing the BdSL using the proposed system

Class	Numbers of correct responses	Recognition Rate %	Class	Numbers of correct responses	Recognition Rate %
১	5	100	ছ	11	78.6
২	13	100	জ/য	6	100
৩	6	86.7	ঝ	5	100
৪	4	80	ঞ	7	100
৫	5	100	ট	6	100
৬	7	74.5	ঠ	8	80.49
৭	5	100	ড	9	69.2
৮	4	67.7	ঢ	6	100
৯	6	100	ণ/ন	8	100
অ/য়	3	80	ত/□	5	100
আ	10	85.5	থ	2	86.7
ই/ঈ	6	85.7	দ	11	91.7
উ/ঊ	12	63.2	ধ	11	100
ঋ/র/ড়/ঢ়	5	62.5	প	6	100
এ	5	100	ফ	5	71.4
ঐ	4	66.7	ব/ভ		100
ও	3	100	ম	5	100
ঔ	5	62.5	ল	5	100
ক/ক্ষ	10	90.9	শ/স/ষ	2	100
খ	6	66.7	হ	14	82.4
গ	7	87.5	ং	6	73.45
ঘ	7	100	ঁ	6	100
ঙ	6	85.7	Total	306	
চ	7	100		Average recognition rate : (306/345)*100% = 88.69%	

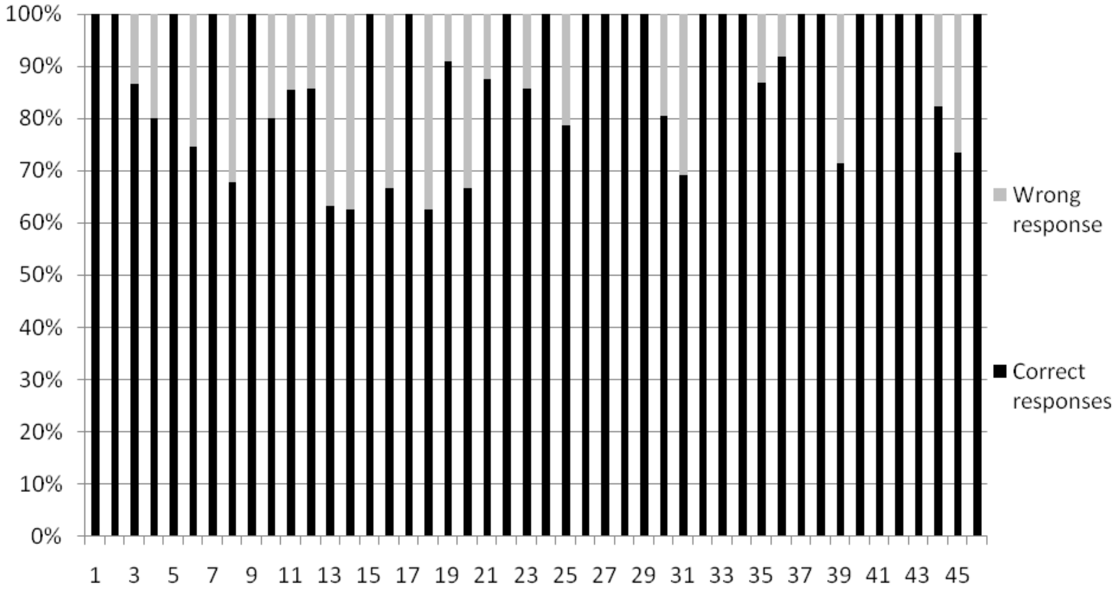


Fig. 12.Correct and wrong responses

plotted on a graph for same Bengali signs where black diagonal points represented the accuracy of fingertip detection and gray square points represented the accuracy of recognition method for same signs.

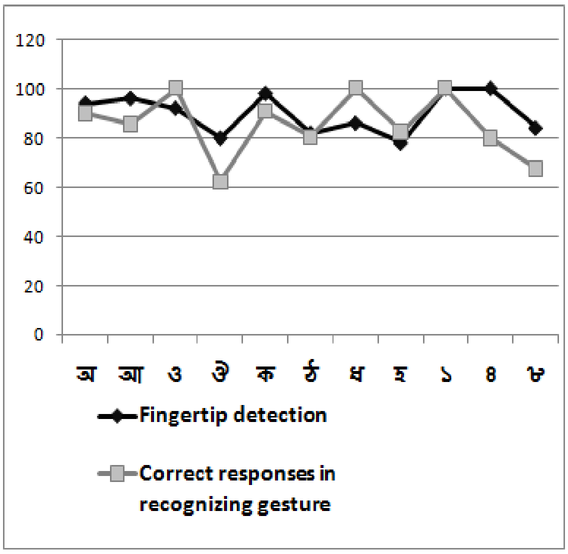


Fig. 13.Sensitivity of recognition method to find the accuracy of fingertip detection

B. *Experiment with dynamic signs*

For research purpose four dynamic signs were tested by the system. The idea of dynamic sign testing was simple. 3 seconds videos of each signs were taken and frames were extracted at a rate of 15 frames per second.

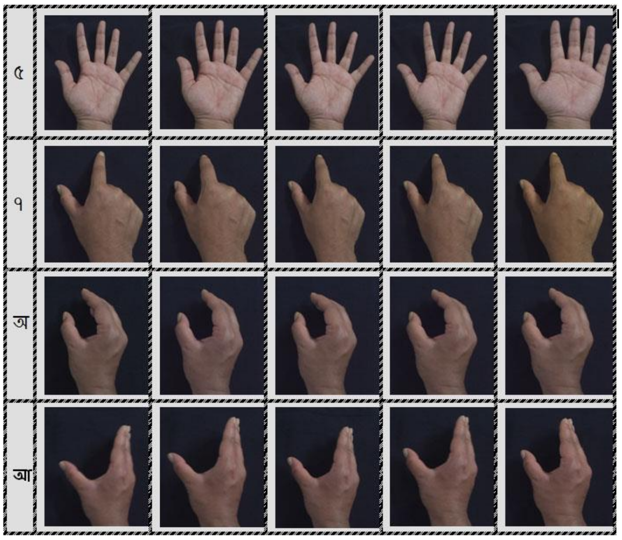


Fig. 14.Some extracted images from video

TABLE III. Comparative analysis of Bengali Sign Language Detection Techniques

Author reference	Method used	Sign Language	Number of Recognized classes	Number of signs recognized / Total number of tested signs	Average accuracy
Adnan Eshaque, Tarek Hamid, Shamima Rahman, and M. Rokonuzzaman[4]	Matching signs with previously stored knowledgebase	Bengali	10	208/256	82%
M. Atiqur Rahman, Ahsan-Ul-Ambia[6]	Multillayer Feedforward ANN	Bengali	36	233/288	80.9%
Manar Maraqa, Farid Al-Zboun, Mufleh Dhyabat and Raed Abu Zitar[8]	Multillayer Feedforward ANN	Arabian	30	714/900	79.33%
M. Atiqur Rahman, Ahsan-Ul-Ambia and Md. Aktaruzzaman[7]	Multillayer Feedforward ANN	American	26	169/210	80.28%
Proposed method	fingertip finder algorithm with multilayered feed-forward neural network with back propagation training	Bengali	46	306/345	88.69%

Features were extracted from the images and were tested by previous system that has been trained with static images. Some extracted images are shown in Fig. 14.

Accuracy of recognizing those extracted images of signs was 95.55% which shown in TABLE V.

TABLE IV. Accuracy of Fingertip Finder Algorithm

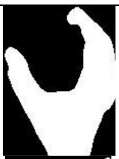
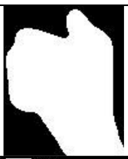
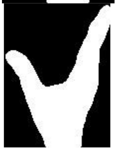
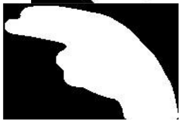
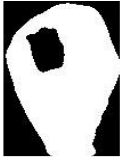
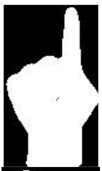
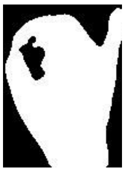
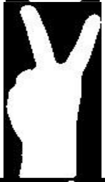
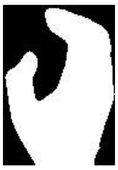
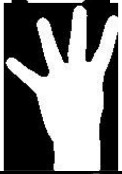
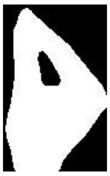
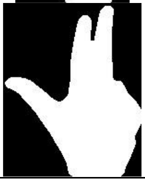
BdSL Sign	Image of Sign	No of Correct recognitions/ No of test conducted	Fingertip Detection Rate	BdSL Sign	Image of Sign	No of Correct recognitions/ No of test conducted	Fingertip Detection Rate
অ		47/50	94%	ধ		43/50	86%
আ		48/50	96%	হ		39/50	78%
ও		46/50	92%	১		50/50	100%
ঔ		40/50	80%	২		50/50	100%
ক		49/50	98%	৪		50/50	100%
ঠ		41/50	82%	৮		42/50	84%

TABLE IV. Accuracy in dynamic sign recognition

Alphabet for sign	Number of images extracted/Number of images correctly recognized	Accuracy	Average Accuracy
৫	45/45	100%	95.55%
৭	42/45	93.33%	
অ	43/45	95.55%	
আ	42/45	93.33%	

IV. Conclusions

In this paper a feed-forward ANN with back-propagation learning algorithm was used, where Feature extraction was started after converting the RGB images to binary, and cropping the binary image to hand region. The system was made to recognize 46 Bangali signs, including alphabet and numerals. MATLAB 2013a was used in an Intel core i3 2.30 GHz machine with 64 bit Windows7 Operating system. Fingertips of a sign were found by the proposed fingertip finder algorithm, without the need for glove, sensor or any marking system. Accuracy of this system in recognizing BdSL is very much promising compare to other methods. Eleven features were extracted, based on distances and angles between finger tips and horizontal X axis that crossed the centroid of the area of hand, and were used in training a multilayer feed-forward neural network. Average accuracy of this system was 88.69%. Accuracy could be increased by increasing the number of sample images in training.

Acknowledgments

The authors are grateful to Dr. William R Ward, University of Liverpool, UK for his help in preparing the manuscript.

References

[1] S. H. Pavel, T. Mustafiz, A. I. Sarkar, and M. Rokonuzzaman," Geometrical Model Based Hand Gesture Recognition for Interpreting Bengali Sign Language Using Computer Vision," In: *International Conference on Computer and Information Technology.* United International University, Dhaka, Bangladesh, 2007, pp. 642-647.

[2] S. Rahman, N. Fatema, and M. Rokonuzzaman, "Intelligent Assistants for Speech Impaired People," In: *International Conference on Computer and Information Technology,* East West University, Dhaka, Bangladesh. 2002.

[3] S. Begum and M. Hasanuzzaman, "Computer Vision-based Bangladeshi Sign Language Recognition System," In: *International Conference on Computer and Information Technology,* MIST, Dhaka, Bangladesh, 2009, pp. 414 - 419.

[4] A. Eshaque, T. Hamid, S. Rahman, and M. Rokonuzzaman, "A Novel Concept of 3D Animation Based 'Intelligent Assistant' for Deaf People: for Understanding Bengali Expressions," In: *International Conference on Computer and Information Technology***,** East West University, Dhaka, Bangladesh, 2002, pp 252-256.

[5] B. C. Karmokar, K. M. R. Alam, and Md. K. Siddiquee, "Bangladeshi Sign Language Recognition employing Neural Network Ensemble," *International Journal of Computer Applications,* vol.*58*, 2012, pp. 0975 – 8887.

[6] M. A. Rahman, Ahsan-Ul-Ambia, M. I. Abdullah, and S. K. Mondal, "Recognition of Static Hand Gestures of Alphabet in Bangla Sign Language" *IOSR Journal of Computer Engineering (IOSRJCE),* vol. 8 (1), 2012, pp. 07-13.

[7] M. A. Rahman, Ahsan-Ul-Ambia, and M. Aktaruzzaman, "Recognition Static Hand Gestures of Alphabet in ASL," *International Journal of Computer and Information Technology,* vol. 2 (1), 2011, pp. 75-78.

[8] M. Manar, A. Z. Farid, M. Dhyabat, and R. A. Zitar, "Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks," *Journal of Intelligent Learning Systems and Applications*, vol. 4, 2012, pp. 41-52.

[9] K. Assaleh and M. A. Rousan, "A New Method for Arabic Sign Language Recognition," *EURASIP Journal on Applied Signal Processing,"* Hindawi Publishing Corporation, New York, 2005, pp. 2136-2145.

[10] B. Henrik and T. B. Moeslund, "Recognizing Gestures from the Hand Alphabet Using Principal Component Analysis," Master's Thesis, *Laboratory of Image Analysis,* Aalborg University, Denmark, October 1996.

[11] M. V. Lamar, S. Bhuiyan, and A. Iwata, "Hand Alphabet Recognition Using Morphological PCA and Neural Networks," In: *Proc. of IEEE Intl. Joint conf. on Neural Network*, Washington, DC, vol. 4, 1999, pp. 2839-2844.

[12] K. Sheth, G. Nitish, and P. R. Futane, "A Survey of Gesture Recognition Systems for Indian Sign Language Recognition," *Intl. Journal of Engineering and Advanced Technology*, vol. 2(6), 2013, pp. 284-289.

[13] S. Saengsri, V. Niennattrakul, and C.A. Ratanamahatana, "TFRS: Thai Finger-Spelling Sign Language Recognition System," In: *Proc. of IEEE Second Intl. Conf. on Digital Information and Communication Technology and its Application*, Bangkok, 2012, pp. 457-462.

[14] J. H. Kim, N. D. Thang, and T. S. Kim, "3-D Hand Motion Tracking and Gesture Recognition Using a Data Glove," *IEEE International Symposium on Industrial Electronics* Seoul Olympic Parktel, Seoul, Korea, 2009, pp. 1013-1018.

[15] J. Weissmann and R. Salomon, "Gesture Recognition for Virtual Reality Applications Using Data Gloves and Neural Networks", In: *Proc. of IEEE Intl. Joint conf. on Neural Network*, Washington, DC, vol. 3, 1999, pp. 2043-2046.

[16] P. K. Kshirsagar and D. Dharmpal, " Object Based key Frame Selection for Hand Gesture recognition," *International Conference on Advances in Recent Technologies in Communication and Computing*, Kottayam, Kerala, India , 2010, pp. 181-185.

[17]https://nf.nci.org.au/facilities/software/Matlab/toolbox/images/bwmorph.html. Accessed on 18 December 2014.

[18]http://www.mathworks.com/help/images/ref/regionprops.html. Accessed on 23 December 2014.

[19]http://www.mathworks.com/help/matlab/ref/atan2.html. Accessed on 18 March 2015.

[20]http://www.cut-the knot.org/pythagoras/DistanceFormula.shtml Accessed on 24 March 2015.