Computer Vision Based Bengali Sign Words Recognition Using Contour Analysis

Muhammad Aminur Rahaman, Mahmood Jasim, Md. Haider Ali and Md. Hasanuzzaman
Department of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh
Email: aminur.wg@gmail.com, jasim@csedu.ac.bd, haider@du.ac.bd, hzaman@cse.univdhaka.edu

***Abstract*—This paper presents a computer vision based Bengali sign words recognition system using contour analysis. Haar-like feature based cascaded classifier is used to locate the predefined hand posture (Opened Hand and followed by Closed Hand postures) from the captured image, and bounded by a rectangular box that is initialized as region of interest (ROI). The system follows this ROI, crops it and normalizes into predefined size. The system segments skin-like area based on Hue and Saturation value from the normalized image. Then the system employs morphological operations and Gaussian smoothing to remove noises, and then converts it into gray image. The system extracts contours using Canny edge detector and encodes extracted contours into Vector Contours (VC). After scaling VC into predefined size, the system generates feature space based on equalized VC, value of normalized Auto-Correlation Function (ACF) and ACF descriptors for each sign word that will be used for training and/or testing process. The system recognizes sign words based on maximum similarity between tests and predefined training contour templates using Inter-Correlation Function (ICF). The system is trained and tested using 1800 (18×10×10) contour templates separately for 18 Bengali sign words from 10 signers achieving recognition accuracy of 90.11% with computational cost of 26.063 milliseconds per frame.**

Keywords*—Bengali Sign Language (BdSL); Skin Color Based Segmentation; Contour Analysis; Vector Contour (VC); Auto-Correlation Function (ACF); Inter-Correlation Function (ICF).

I. Introduction

Sign language is a separate language with its own grammar which is used by speech or hearing impaired people to communicate with other people and themselves. There are approximately 5000 sets of signs for alphabet, numbers and common words in Bengali sign Language (BdSL) [1]. Most of the words in BdSL have to be performed by using a sequence of the static signs where finger and hand positions are needed to be identified and analyzed in order to recognize. The use of sign language as an interface for human–computer interaction (HCI) is increasing rapidly [2]. But Recognition of sign words considering use of hand shapes, directions, motions, orientations, locations respective to human body is very challenging. This paper presents a Computer vision based Bengali sign word recognition system using contour analysis. The contour analysis allows describing, storing, comparing and finding the objects presented in the form of the outer contours. It performs in a simple manner using mathematical analysis for image processing with lower computational cost and algorithmic complexity [3]. The system recognizes 18 common BdSL words. The proposed system can be applied as an interpreter for sign and non-sign people communication through sign languages. It can also be used for manipulating robots or other devices without any physical contact between the human and machine. The test results of the proposed system is compared with our previous reputed BdSL recognition systems using skin-color based segmentation and KNN classifier by M. A. Rahman et al. [4] for the same dataset. This paper is organized as follows. Section II describes the proposed system. Section III presents the experimental results with discussions. Finally, the paper is concluded in Section IV.

II. Proposed System Description

Figure 1 presents the block diagram of the proposed computer vision based Bengali sign word recognition sys-

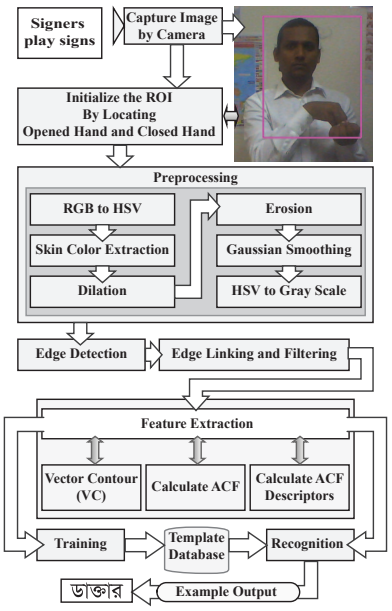


Figure 1. Block Diagram of the Proposed System.

978-1-4673-9930-2/15/$31.00 ©2015 IEEE

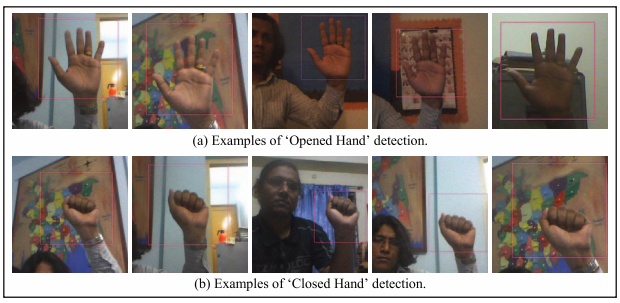


Figure 2. Example of Hand locating using Haar Like Feature Based Cascaded Classifier (a) After 'Opened Hand' locating ROI is initialized and (b) After Closed Hand locating the sytem starts to process the ROI.

tem. Following subsections briefly describe each process of the proposed system.

A. Hand Locating and ROI Initialization

To segment hand postures and face area from the captured image skin color based segmentation method is used by the system which is described in preprocessing subsection. But skin color based segmentation method does not work so well under various lighting conditions and backgrounds. This can be overcome by using Haar-like feature-based cascaded classifier [5], which is mostly illumination and background invariant. But, to segment every new sign from the captured image using this Haar classifier needs to train every time for the specific sign, which restricts the system to make universal. For this situation, we have used these two methods in combined in our proposed system. Haar-like feature-based cascaded classifier is used only for locating 'Open Hand' posture and a 'Closed Hand' posture from the captured image. After locating the 'Opened Hand' posture bounded by a rectangular box that is initialized as region of interest (ROI). After locating the 'Closed Hand' posture then the system starts to process the ROI considering all sign words performed within that ROI and skin color based segmentation method starts to segments skin-like area from the ROI instead of whole image. The confusion related to background and illumination is minimized and our test results show good performance under various illumination and backgrounds. Figure 2 presents the examples of ROI initialization by locating 'Opened Hand' postures and 'Closed Hand' postures using the trained Haar-like feature-based cascaded classifiers.

B. Preprocessing

After initialization, the system crops the ROI and normalizes the arbitrary sized cropped image into predefined size. Then the system converts the normalized RGB images to HSV color coordinate system. The system extracts hands and face areas of each BdSL word based on Hue (H) and Saturation (S) value of human skin color using the threshold (0<=H<=29 and 49<=S<=68) [6] from

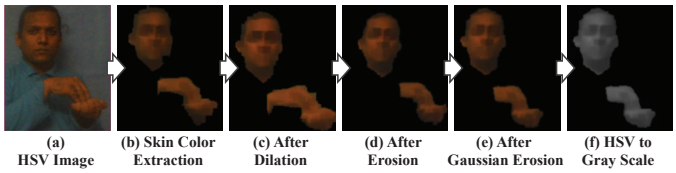


Figure 3. Example results of the preprocessing (a) HSV Image, (b) After Skin Color Extraction, (c) After Dilation, (d) After Erosion, (e) After Smoothing by Gaussian filter and (f) After converting to Grayscale Image.

the normalized cropped images. Example output of skin color based hand posture segmentation is shown in Figure 3(b). After completing the skin color based hands and face area of each BdSL word extraction, the system uses morphological filtering dilation using Eq. (1) and erosion using Eq. (2) respectively to reduce noise [7].

$$I_d = A \oplus B = \{z|(B'_z) \cap A \neq \phi\} \tag{1}$$

$$I_e = I_d \ominus B = \{z|(B)_z \subseteq I_d\} \tag{2}$$

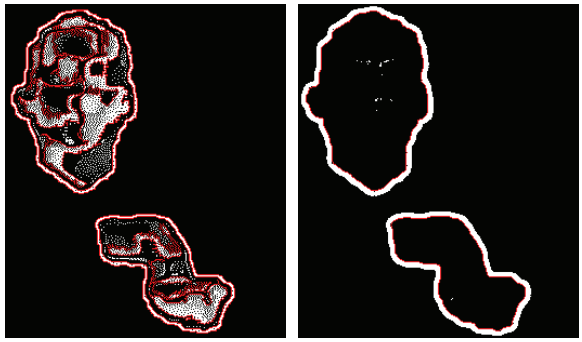
Where, I_d is the output image after dilation and I_e is the output image after erosion of the extracted image A by the structuring element B; z is the set of all points of the extracted image objects A. Using dilation operation, the system can bridge small gap of the extracted hand and face area and using erosion the system can remove unwanted noises from the extracted hand and face area. Figure 3(c) and Figure 3(d) show the example output of the dilation and erosion respectively. After completion of morphological operation dilation and erosion, the system uses Gaussian smoothing on the eroded image, I_e represented by the convolution, $J = I_e * G$ with a filter of 5×5 Gaussian kernel G by using the Eq. (3) [8].

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

Where, x and y are the horizontal and vertical distances from the origin of the eroded image, I_e and σ is the standard deviation of the Gaussian distribution. Figure 3(e) shows the example output of the smoothed image. After removing the noise, the HSV color image of hand and face area is converted to grayscale image. Figure 3 shows the effects of preprocessing steps on a captured image.

C. Edge Detection and Filtering

After preprocessing, the system uses canny edge detection algorithm which performs better than other edge detectors [9]. From the smoothed grayscale image, the system computes the gradient components, estimates the edge strength and orientation of the edge normal respectively. Non-maxima suppression is applied to the strength image and then the system detects edges by applying two thresholds, lower threshold and higher threshold dynamically on all pixels of the images. All visited points



(a) Edge Detection (b) After Edge Linking and Filterring

Figure 4. Example output imges of the (a) Canny edge detection, (b) After edge linking and unwanted edge filtering.

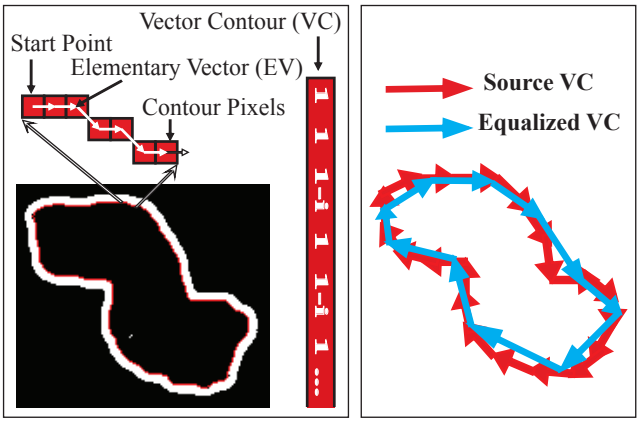


Figure 5. Example process of (a) Encoding of a contour into VC and (b) Contour equalization from source VC

in the connected contour found are stored for contour analysis. Figure 4(a) shows example output image from the Canny edge detection algorithm. After Canny edge detection the contour is generated with unwanted edges and various noises. The system further uses morphological image filtering to link broken edges. Then the system calculates the contour length and area where, a contour length is the total number of the contour pixels and contour area is the maximum (heightXwidth) of the closed contour shape. Then the system filters out the unwanted edge by intersecting with the minimum contour length ($k < 200$) and minimum contour area ((height×width) < 400) from the Canny edge detected image. Figure 4(b) shows the example output of unwanted edge filtration.

D. Feature Extraction

In this step, the system encodes the selected contour into Vector Contour (VC). The encoded Vector Contours (VC) of BdSL words are used as main feature for this proposed system. For encoding the contour, at first the system sets a starting point on the contour and then scans the selected contour in a clockwise direction. Each vector of offset is represented by a complex number 'a+ib'. Where 'a' is the point offset on x axis, and 'b' is the point offset on y axis. Offset is represented concerning the previous point [3]. Figure 5(a) shows the example process of encoding of a contour into VC.

For the three-dimensional objects, the last vector of a contour always leads to the starting point. Each pixel (vector) of a contour is represented by 'Elementary Vector' (EV) and sequence of complex-valued numbers is represented by Vector Contour (VC). The Vector Contour (VC) of length k can be expressed by Eq. (4).

$$\Gamma = (\Upsilon^0, \Upsilon^1, ..., \Upsilon^{k-1}) \tag{4}$$

Where, Γ represents a VC and Υ represents each EV of the VC.

The system calculates the sum of all EVs of a VC using Eq. (5). If the sum of all EVs is equal to zero, then the system decides that the detected contour is a closed contour and considers this VC for contour analysis.

$$\Omega = \sum_{n=0}^{k-1} (\Upsilon^n) \tag{5}$$

Where, k is dimensionality of a VC and Υ^n is the n^{th} EV. The extracted closed contours have arbitrary length. For training and recognition process, the system forces to make all of contours length uniform by equalizing the contours with a predefined length k=100. Figure 5(b) shows the example of contour equalization process from source VC. After extraction of equalized Vectors Contour 'Φ' with length k=100, the system calculates autocorrelation function (ACF) by using Eq. (6) [3].

$$ACF(m) = (\Phi, \Phi^{(m)}) \tag{6}$$

Where, ACF(m) measures the similar length of contours with the value among 0 to 1 and m=0, 1, 2, ..., k-1. After generation of ACF, the system normalizes the ACF. The normalized ACF is symmetric concerning a central reference k/2. Before direct ACF comparison, its descriptors are compared to increases the performance of ACF comparison. The descriptors are calculated as Wavelet convolution [10] of ACF using four different filters such as filter1 = { 1, 1, 1, 1 }; filter2 = { -1, -1, 1, 1 }; filter3 = { -1, 1, 1, -1 }; and filter4 = { -1, 1, -1, 1 }. After feature extraction, the system generates templates based on extracted equalized VC, value of normalized ACF and ACF descriptors for each part of each BdSL word that will be used for training and/or testing process. Figure 6 shows the example output of feature extraction and template generation of BdSL word 'ডাক্তার' (Doctor).

E. Training

After template generation, the system stores maximum suitable templates of equalized VCs with their properties

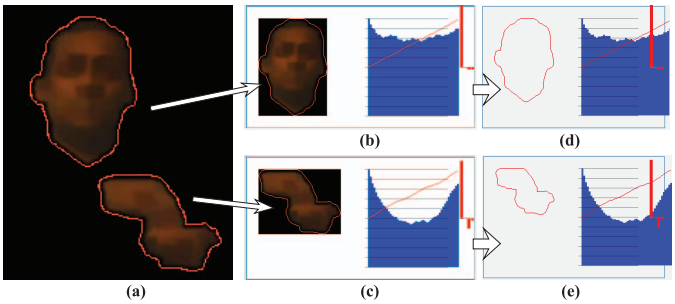


Figure 6. Example output of feature extraction and template generation (a) A BdSL word 'ডাক্তার' (Doctor) containing two contour parts with ACF and ACF descriptor: (b) Face area and (c) Hand posture area; (d) Corresponding contour template of face area and (e) Corresponding contour template of hand posture with extracted features which is stored in binary serialized files as training database.

in binary serialized file for each BdSL word. For training new word, the system displays all identified contours for all parts of a BdSL word, then user can select which ones the user would like to add to the binary serialized file of training database. In this module, the system is trained individually for each selected word. Face area contour is a common contour for each word but hand area contour is different. Some BdSL words contain single contour with single posture in single state (represented by '1State sign') and some words contain more than one contour with several postures in multiple state (represented by 'mState sign'). In our proposed system, '1State sign' and '2States sign' of BdSL word are considered. Figure 6(a), Figure 6(b) and Figure 6(c) show an example of training process of the system for a '1State' BdSL word 'ডাক্তার' (Doctor).

F. Recognition

In recognition process, the proposed system captures images and generates the template of test BdSL word same as training template generation. Then the system recognizes the test BdSL word by measuring maximum ICF among the contours of test word and training BdSL word using Eq.(7).

$$ICF_{max} = \left(\frac{ICF(m)}{|\Phi|\,|\tau|} \right) \tag{7}$$

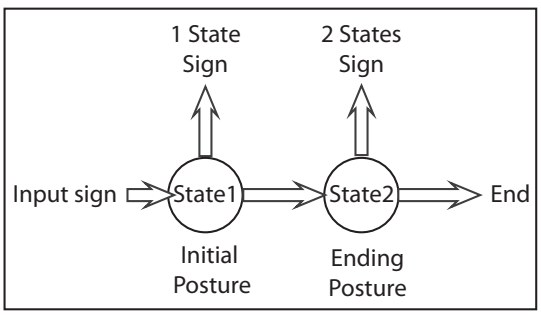


Figure 7. Proposed recognition model of BdSL word recognition.

Where, ICF_{max} measures the similarity shape of hand signs with the value among 0 to 1. $|\Phi|$ and $|\tau|$ represent the normalized length of training contours,Φ and τ respectively. ICF(m) is the Inter-Correlation Function between Φ and $\tau^{(m)}$ which is calculated using Eq.(8). Where, $\tau^{(m)}$ represents a vector of contour of test contours received from τ by cycle shift by its EV on 'm' of elements.

$$ICF(m) = \left(\Phi, \tau^{(m)} \right) \tag{8}$$

Two states recognition model is proposed in our system considering initial state (State1) and ending state (State2) to recognize the BdSL words as shown in Figure 7. Example output of this model is shown in Figure 8 and Figure 9.

III. Experimental Result And Discussion

The proposed system uses a built-in webcam of ASUS A42F series laptop for image capturing. The system uses an ASUS A42F series laptop with Intel 2.40 GHz Corei3 processor and 2GB RAM. The system uses EmguCV (C# and OpenCV wrapper) [11] in 32-bit operating system of MS Windwos7 as system development platform. The proposed system is trained for 18 BdSL word with the equalized length of VC, k=100 for each contour part of the BdSL word. 10 images of each predefined BdSL word are captured from each signer among ten signers for training the system. This resulted in 1800 (18×10×10) set training images for BdSL word. The system uses two performance parameters such as, Accuracy and Computational cost. Accuracy is calculated using Eq.(9).

$$Accuracy = \frac{R \times 100}{T} \tag{9}$$

Where, R=Number of correctly recognized Sign word and T=Total Number of the Sign word. The time to capture an image, preprocess, features extraction, template generation and matching it with the training contour template database is considered as computational cost in Milliseconds per frame.

Table I presents the summarized results of 18 BdSL word recognition by the proposed system (CA) and comparison analysis with another Bengali sign language recognition (BdSLR) system using KNN [4] for the same dataset. From the test results, it is evident that BdSL words are recognized by the proposed system which is comparatively better than our previous system [4].

From Table I, the mean Accuracy of the proposed system is decreased due to similarity of equalized contours of the hand postures among the words 'দুঃখিত' with 'কিছু'; 'কষ্ট' with 'খুশি' (State2); 'মা' with 'তাকাও'; and 'জ্বর' with 'আস্-সালামু আলাইকুম' as shown in Figure 8.

Particularly for sign word 'খুশি' (Happy), recognition Accuracy is decreased to 80% because of its wrong recognition of one state (among two states) that leads to wrong recognition for the whole word as shown in Figure 9(b).

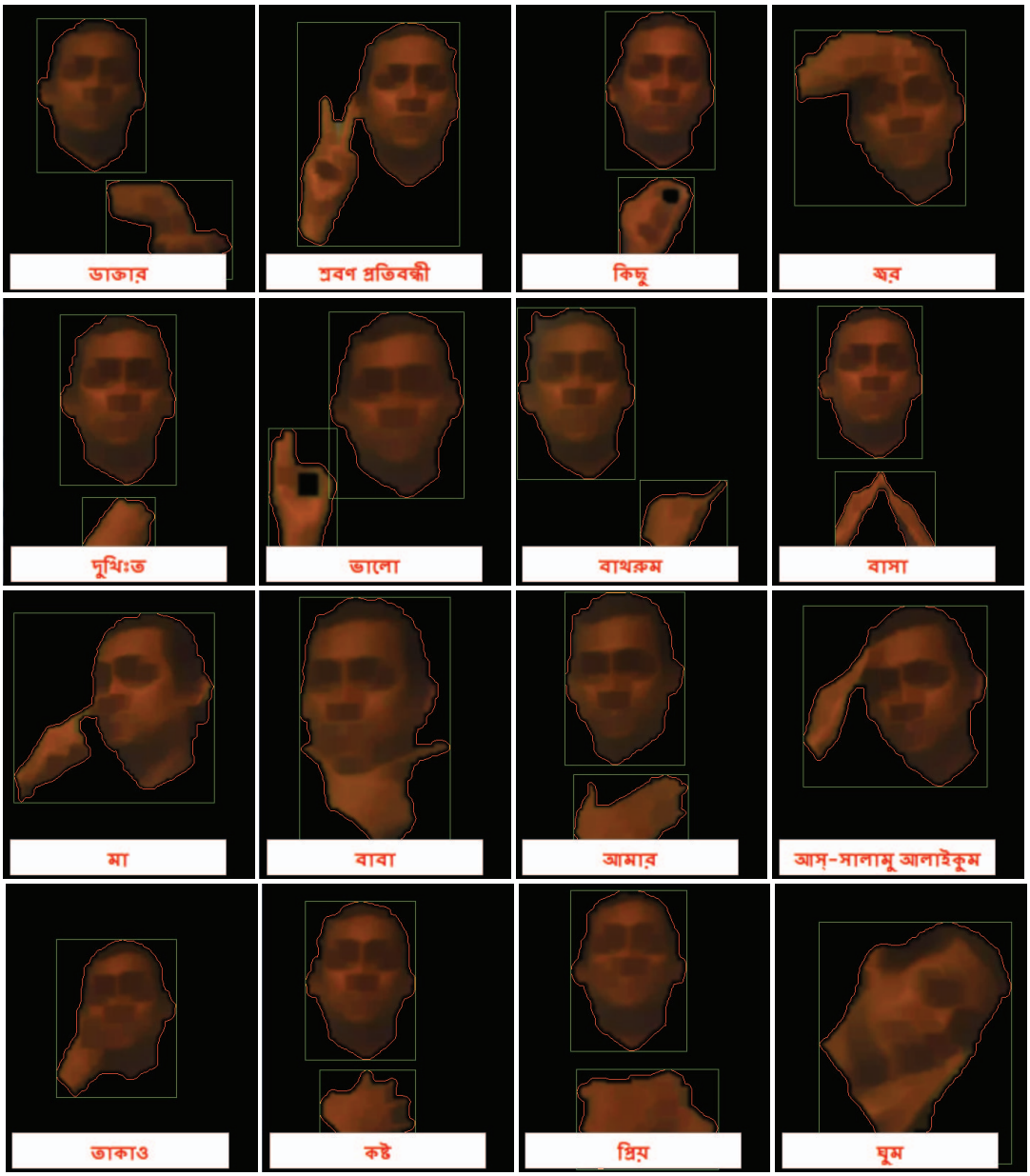


Figure 8. Example recognition of '1State sign' of BdSL words.

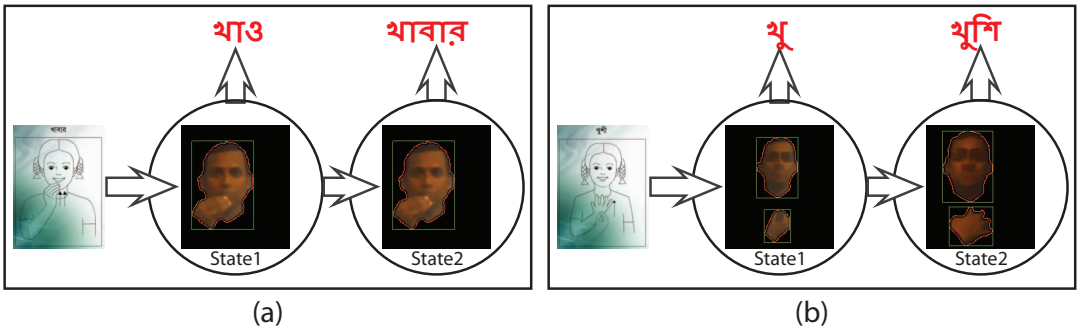


Figure 9. Example recognition of '2States sign' of BdSL words (a) 'খাবার' (Food), where output of State1 represents another BdSL word 'খাও' (Eat) and (b) 'খুশি' (Happy).

Table I
RResult Of BdSL Words Recognition For The Proposed System (CA) And BdSLR System Using KNN [4]

BdSL Word	Total Word Signs (T)	Correctly Recognized (R)		Accuracy (%)	
		CA	*KNN*	*CA*	*KNN*
ডাক্তার(Doctor)	100	93	81	93	81
শ্রবণ প্রতিবন্ধী (Hearing impaired)	100	92	85	92	85
কিছু (Some)	100	87	83	87	83
জ্বর(Fever)	100	87	77	87	77
দুঃখিত(Sorry)	100	89	83	89	83
ভালো(Good)	100	91	94	91	94
বাথরুম(Toilet)	100	97	95	97	95
বাসা(House)	100	89	77	89	77
মা(Mother)	100	87	90	87	90
বাবা(Father)	100	90	87	90	87
আমার(My)	100	92	83	92	83
আস্-সালামু আলাইকুম (Salam)	100	86	84	86	84
তাকাও(Look)	100	87	67	87	67
কষ্ট(Trouble)	100	88	78	88	78
প্রিয়(Favorite)	100	97	89	97	89
ঘুম(Sleep)	100	96	88	96	88
খাবার(Food)	100	94	78	94	78
খুশি(Happy)	100	80	72	80	72
Mean Accuracy				**90.11**	**82.83**

Table II
Comparative Analysis Of The Result Of The Proposed System (CA) With BdSLR System Using KNN [4] For Computational Cost

Method	Computational Cost (Milliseconds per frame)
Proposed System (CA)	26.063
KNN	88.093

Table II represents the comparative analysis of the computational costs of the proposed system (CA) with the mentioned BdSLR system using KNN [4].

IV. Conclusion

This paper presents a computer vision based Bengali sign words recognition system using contour analysis. Haar-like feature based cascaded classifier is used to locate the predefined hand posture (Opened Hand and followed by Closed Hand postures) from the captured image which is defined as region of interest (ROI). Then the system follows this ROI, crops it and normalizes it into the predefined size. After necessary preprocessing, the system extracts contours using Canny edge detector and encodes extracted contours into Vector Contours (VC). The system generates contour feature spaces based on equalized VC, the value of normalized Auto-Correlation Function (ACF) and ACF descriptors for each predefined Bengali sign word that are used for training and/or testing process. The system recognizes sign words based on maximum similarity between test contour templates and predefined training contour templates using Inter-Correlation Function (ICF). The system is trained and tested using 1800 (18×10×10) contour templates separately for 18 BdSL words from ten signers achieving recognition accuracy of 90.11%. Mean computational cost of 26.063 milliseconds per frame with equalized contour length, k=100. The limitation of this research is that the input BdSL word images should have sharp and connected edge in order to identify each part or component of the word correctly. Further research will be focus on preprocessing, segmentation method and enhancement the image that contained broken edges. Despite some limitation, the proposed system is attractive for the simplicity and high-speed performance. The proposed system can be used for Human Machine Interaction (HMI) using sign language. It can also be applied as an interpreter for sign and non-sign people communication through sign languages.

Acknowledgment

This research is supported and funded by the Information and Communication Technology (ICT) Division, Ministry of Posts, Telecommunications and IT, Government of the People's Republic of Bangladesh.

References

[1] Centre for Disability in Development (CDD), "Manual on Sign Supported BangIa", In Computer Vision and Image Understanding, (2002), p.1-50.

[2] M.K. Bhuyan, K. F. MacDorman, M. K. Kar, D.R. Neog, B. C. Lovell, P. Gadde, "Hand pose recognition from monocular images by geometrical and texture analysis." In: Journal of Visual Languages and Computing, Vol. 28, (2015), p.39–55.

[3] R. Kolar, A. Thakar, M. Shabad, "Image segmentation for text recognition based on boundary analysis", International Journal of Emerging Technology and Advanced Engineering, Vol. 4(2), (2014), p. 294-298, ISSN 2250-2459.

[4] M. A. Rahaman, M. Jasim, M. H. Ali, M. Hasanuzzaman, "Real-Time Computer Vision-Based Bengali Sign Language Recognition." In: Procedings of the 17th Int'l Conf. on Computer and Information Technology (ICCIT2014), pp. 192-197, IEEE, Dhaka, Bangladesh (22-23 December, 2014), doi: 10.1109/ICCITechn.2014.7073150.

[5] P. Viola, M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", In: IEEE CVRP, 2001 (2001), p. 511.

[6] S. L. Phung, A. Bouzerdoum, D. Chai, "Skin Segmentation Using Color Pixel Classification: Analysis and Comparison", In: IEEE transactions on pattern analysis and machine intelligence, vol. 27(1), (2005), pp. 148-154.

[7] R. C. Gonzalez, R. E. Woods, "Digital Image Processing." (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2006).

[8] M. Jasim, T. Zhang, M. Hasanuzzaman, "A real-time computer vision-based static and dynamic hand gesture recognition system" In: International Journal on Image and Graphics, vol. 14(01n02) (2014), doi: 10.1142/S0219467814500065

[9] J.Canny, "A Computational approach to edge detection." In: IEEE Transaction Pattern Analysis Machine Intelligence, vol. 8(6), (Nov. 1986), pp. 679-698.

[10] F. P. R. Antonio, R. Robles, "The convolution theorem for the continuous wavelet transform", In: Elsevier B.V., Signal processing, Vol. 84(1), (January 2004), P.55–67.

[11] http://sourceforge.net/projects/emgucv/files/latest/download Accessed at 08/04/2015.