

Gesture Based Implementation on Neural Network: Bengali Sign Language to Text Conversion



Thesis submitted in partial fulfilment of the requirement for the degree of

Bachelor of Computer Science and Engineering

Under the Supervision of

Dr. Jia Uddin

By

S.M. Farzana Hasan (15101145)

TanjinaMehnaz Rahman (11201011)

Anika Raisa Chowdhury (13201038)

Akash Biswas (14101206)

Department of Computer Science and Engineering

August 2017

BRAC University, Dhaka, Bangladesh

Abstract

This paper presents a novel system that converts Bengali Sign language to text using an optimum system comprising of artificial neural networks and support vector model. Bengali Sign Language, has very few research papers based upon it. Therefore, for this paper we have created our own dataset. Microsoft Kinect is utilized as its input device and a complex method of bone and joint determination using contour shape, fingertip detection. Contour feature is extracted and is run through a Support Vector Machine for classification of the sign. The program has been developed in Visual Studio 2015 with a SVM wrapper.

CHAPTER 01

Introduction

Sign language is a form of communication that people with speech and/or hearing impairment utilize. This form of language combines symbols made from hand gestures to form letters, phrases, or sentences. This paper works on forming a gesture recognition system for the Bengali Sign Language. M. Jarman et al have previously created a similar system however, this paper will be using 5 layers of neural networks to train our algorithm [1]. Dataset for BSL is elusive and as ResearchGate.com done a new dataset was created for training [2]. Features that are being utilized for recognition are contour, fingertip and joint location for the hand gesture. The contour finding algorithm utilizes the convex hull method, and the algorithm for fingertip and joint location is provided in later chapters. The features extracted after detection is passed through the support vector model for recognition. Support vector models are very reliable in gesture recognition as used by [2,3]. For input the XBOX360 Kinect device has been used.

1.1 Motivation

Sign language is a term which might sound enticing to many of us as an interesting language involving various gestures but to the hearing-impaired community, this is an obligatory form of communication. This is the only medium through which they can convey and receive their messages. The prevalence of hearing impairment has grown rapidly in the past few years and the impact of hearing loss by birth is more dreadful than who loses it by noise pollution or by any other means. Innumerable researches have been done on sign language converter to accelerate the facility for the hearing impaired in different languages to ease the complexity of their communication but there are inadequate researches made for Bengali sign language. [10] Bengali language is one of the widely spoken languages in the world, so we cannot disdain any adversity related to this language. This factor stimulated us to work with Bengali sign language converter to build a correspondence between hearing impaired and normal hearing community. Our motive is to diversify our system by building an interactive communication between two different sign languages through a translator.

1.2 Contribution Summary

There are several papers that have worked on creating a gesture recognition system using neural networks and support vector models. However, the uniqueness of this paper lies in the tracking method that we have developed. Our system aims to location vital points of the hand such as the fingertips, wrist and joints to determine the input hand sign. This obliterates uncertainties that may arise from utilizing skin detection or blob detection. The vital features stated previously, is used to classify the hand sign.

1.3 Thesis Outline

This paper combines various mechanisms that promise great accuracy to create an accurate and fast system that can be utilized in the service sector. Chapter 2 describes relevant papers that have aided us in building this algorithm. It has been divided into subsectors that will explain the individual components of the system. Chapter 3 and Chapter 4 consists of the experimental setup and outcome. The paper is wrapped up in Chapter 5 where our final findings are summarized.

- Chapter 2 discusses about the background study. The Background study describes relevant papers that have aided us in building this algorithm. It has been divided into subsectors that will explain the individual components of the system.
- Chapter 3 describes our experimental setup, methodology, and dataset creation
- Chapter 4 presents the results of the experiment along with performance analysis and comparisons
- Chapter 5 concludes the paper specifying the limitations and challenges while planning future development of the project

CHAPTER 02

Background Information

2.1 Neural Networks

A neural network is a part of the machine learning field, it is basically a parallel computational model. The model mimics the human neuron system to a certain extent. This is an imitation of the neural networks that exist in the brain. The aim of the artificial neural networks is to work with the speed of an actual neuron that is relaying signals from synapse to synapse. Figure 1 illustrates the overall layout of the network.

In our system we used a five layer model. The first layer is basically the input layer, which takes in the input. In our case, the input is the scenario that is seen by the camera. The second layer figures out hand in the scenario and converts the image to grayscale. The third layer finds the convexity defect and draws a polygon, after figuring out the largest difference between the nodes. The fourth layer converts the image within the convex hull into a threshold image, which is used for mapping and matching with the stored datasets for finding similarity. If similarity has been found then the appropriate text output is displayed else an error message notifying failure is displayed as output. The output displaying layer is the fifth and final layer of our model.

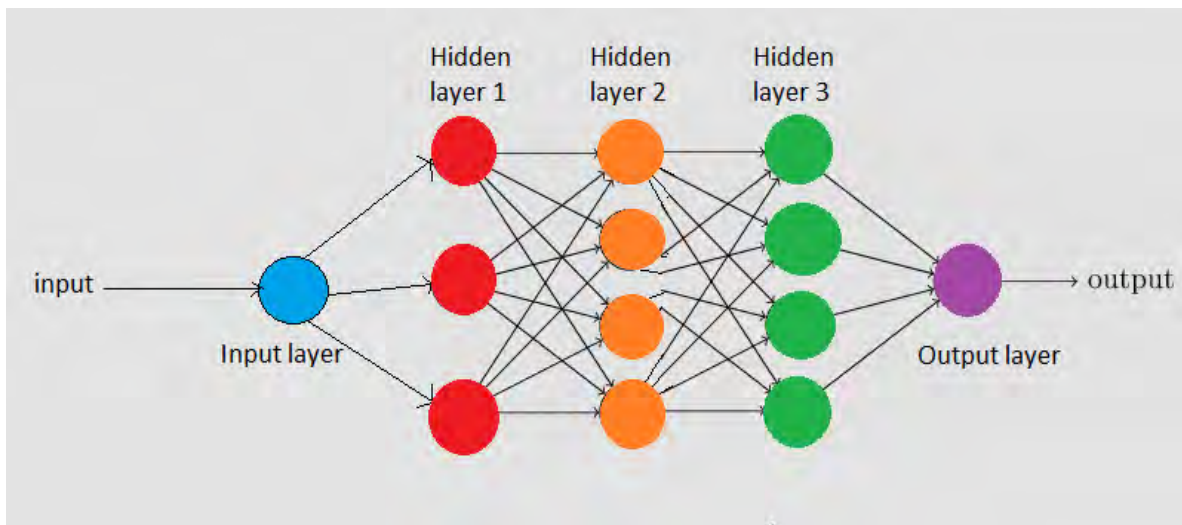


Figure 1: Neural Network description

2.1.1 Algorithm

Training the neural network involved taking in images and storing in the database for matching. In order to train, the back-propagation algorithm has been used. This is a supervised algorithm, which works as follows:

1. Initialization of weight and biases
2. Forward propagate to calculate output
3. Calculate error
4. Adjust outputs
5. Continue from 2 until error minimized

2.2 Support Vector Machine

Support Vector Machine or SVM is an algorithm used primarily for classification. This method has been very popular when running classification of gestures. Support vector machines divides data into two different classes and then hyperplane is drawn between the two classes that achieves maximum marginal distance [3,4]. Training and testing sets are required in order to classify using this method. Using the data from the testing sets the algorithm has to be able to recognize the targeted value, or in this case, hand sign. The following equations and values have been used following the steps shown in [5], in creating the SVM for this paper is utilized to aid in proper building of the recognition algorithm.

Training set of instance-label pairs are denoted by $(x_i, y_i), x_i \in R^n, y_i \in \{1, -1\}, i = 1 \dots, l$. The following equation is then solved by SVM.

$$\min_{w, b} \frac{1}{2} w^T w + \sum_{i=1}^l \xi(w, b; x_i, y_i) \quad (1)$$

Contingent to

$$y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

The previously discussed hyperplane is drawn using the following formula.

$$f(x) = \omega \cdot \varphi(x) + b = \sum_i a_i^0 y_i K(x_i, x) + b \quad (2)$$

The algorithm is constructed using reference from [5]. The data being passed from the extracted features is converted into vectors, which is then fed into the SVM. Firstly, the features will undergo a feature selection algorithm, to determine priority of features [6,7]. Figure 2 illustrates the methodology used in utilizing the SVM.

2.2.1 Algorithm

1. Equation (1) is minimized.
2. Generalization error is minimized.
3. Data acquired is converted into SVM format
4. Scaling is done on the data
5. RBF kernel is considered.
6. To find the best parameter of C and γ via cross-validation
7. The determined parameters are then used for training the whole set
8. Conduct test.

2.2.2 Flowchart

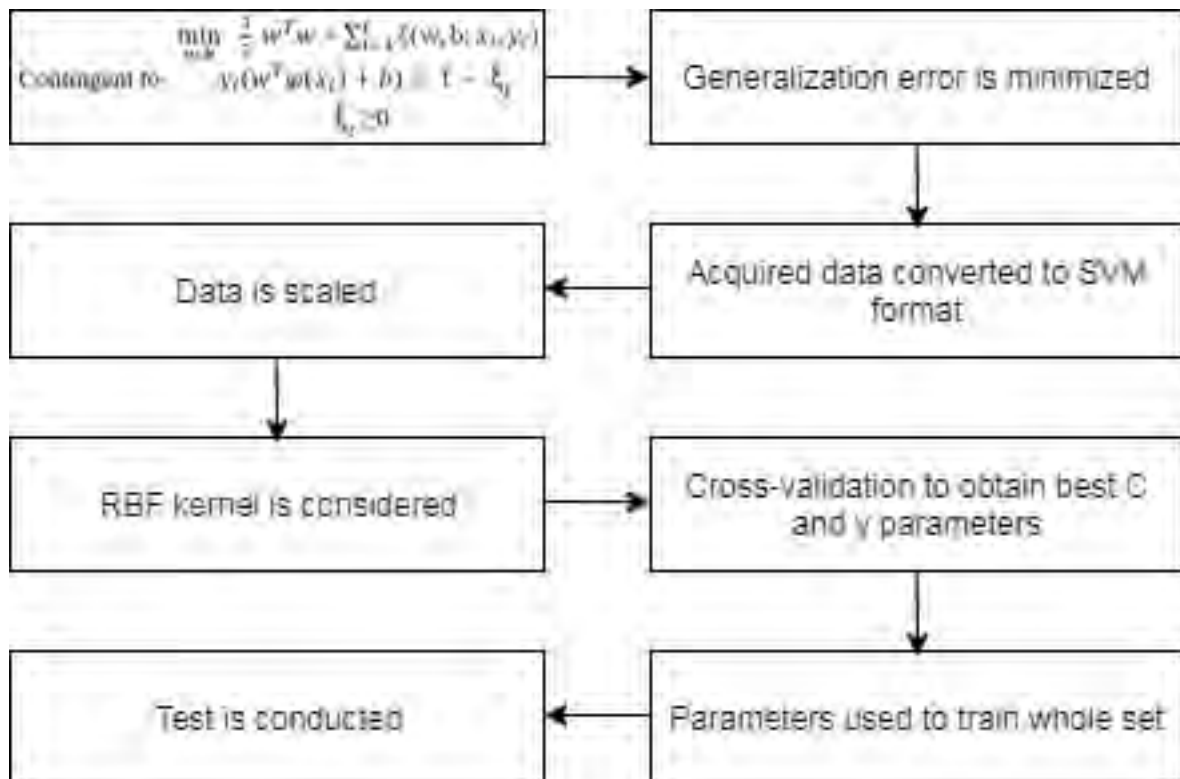


Figure 2: Block Diagram of Support Vector Machine

2.3 Wrist Detection

This algorithm is to determine the position of the wrist. This will provide a starting point at which the template should be applied. An algorithm is provided as well as its corresponding flowchart is illustrated in Figure 3.

2.3.1 Algorithm

1. Reads pixel color from left to right, starting from bottom line to top line.
2. Change from black to white pixel is saved in an array as ThresholdBegin.
3. The location of the change from white to black pixel is also saved into an array as well. This is saved as the ThresholdEnd.
4. The distance between the two thresholds is noted.
5. The lowest Threshold distance found in the lower pixels will be considered the wrist.

2.3.2 Flowchart

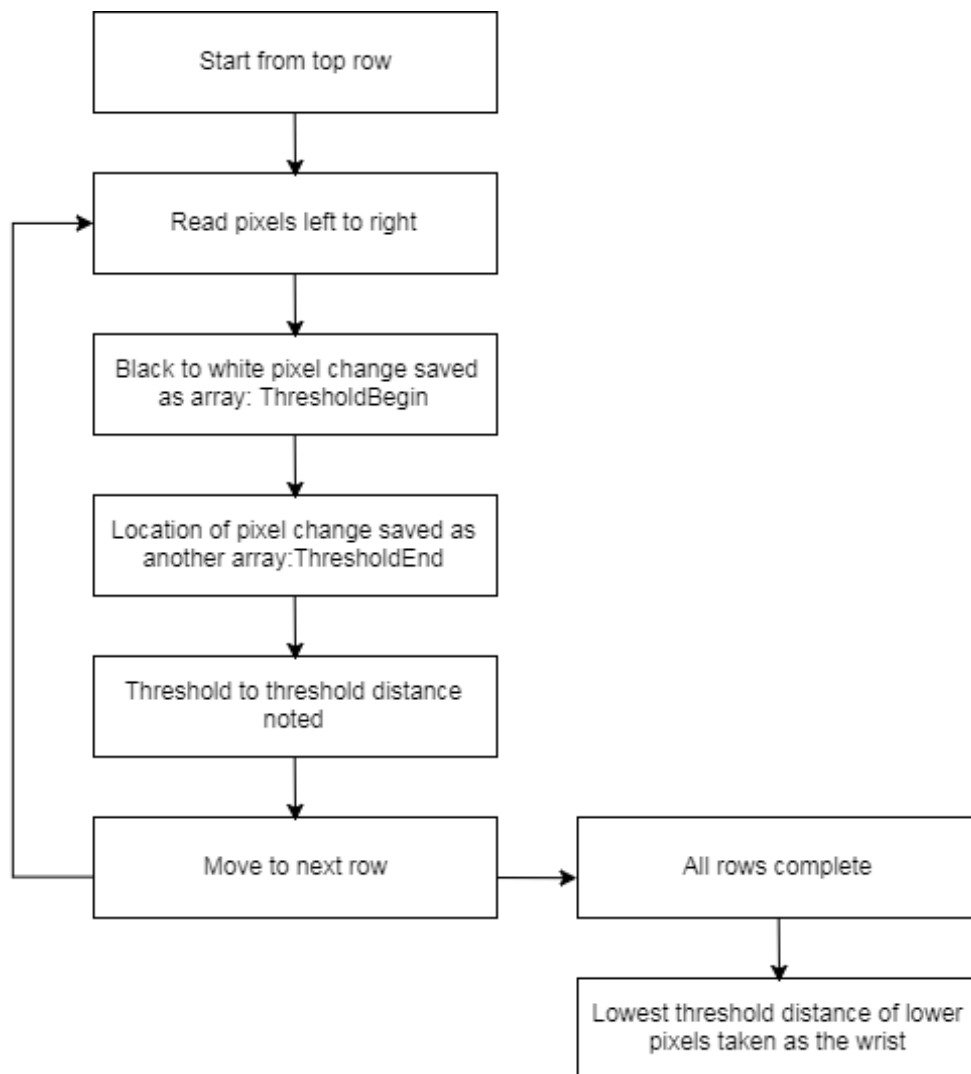


Figure 3: Block Diagram of Wrist Detection Algorithm

2.4 Finger Identification Algorithm

After the hand skeleton is detected by the Kinect, skeleton image needs to be tracked for recognition. The finger joints are figured and contours are traced to allow them to be matched with the existing data set.

The skeleton frame captures up to six persons but can track only two at a time and the person near the Kinect is taken as the primary skeleton. Although up to 20 joints can be tracked only the wrist, hand joints and fingers are obtained in this case. For tracking analyzing the feature points of hand includes locating fingertips and joints, and labeling them. In this test, we design two test cases. The first test is for locating, and the second one is for labeling fingertips and joints. The process of tracking involves feature extraction [10]. This stage is implemented by finding the points of visible fingertips and the centroid of hand region. Figure 4 is a flowchart of the algorithm described below.

2.4.1 Algorithm

1. Every pixel of the detected image is scanned from the top left. This is to find the fingertips, upper portion of the image.
2. Both upper and lower indexes are checked.
3. If pixels of upper indexes are black but lower indexes are non-zero elements then one finger is counted.
4. Points 1 to 3 are repeated till the end of the last column of the binary image. Therefore, all the fingers have to be tracked. In some cases, not all five fingers are visible. In such cases, the invisible fingertips are not considered. Among all, we have ensured better tracking by focusing on figuring out the joints within a finger using the human body ratio and the height of the obtained finger image.

2.4.2 Flowchart

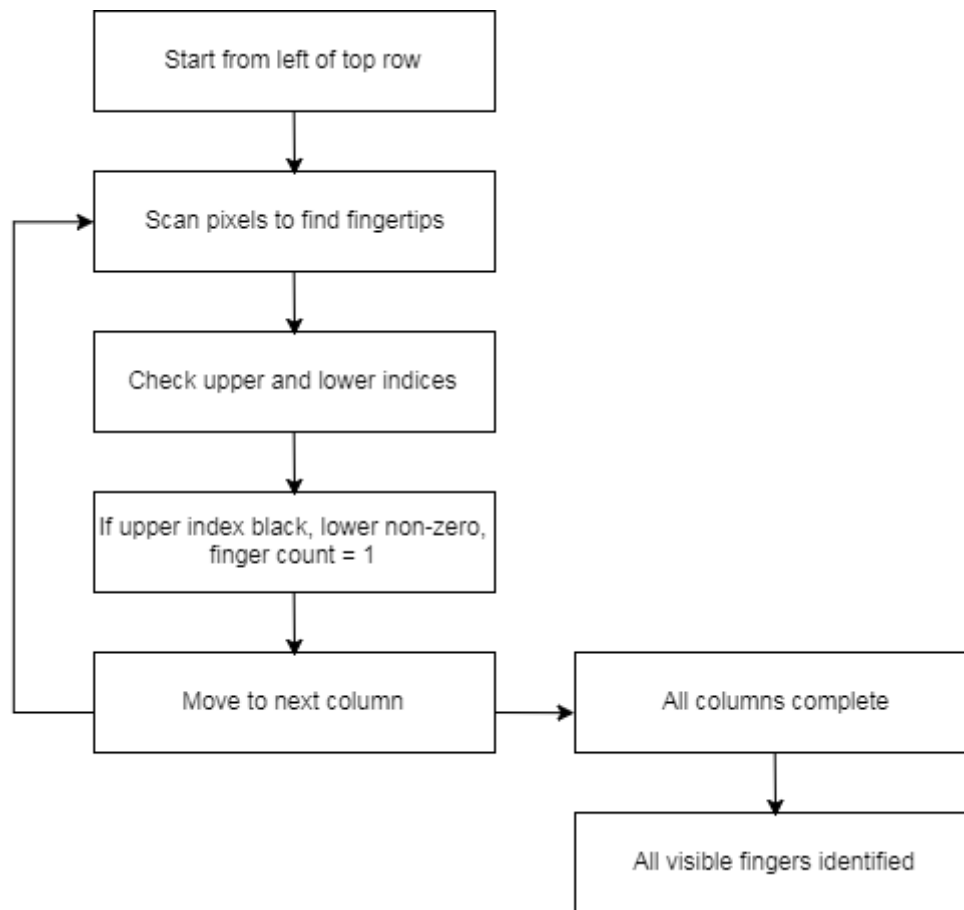


Figure 4: Block Diagram of Finger Identification Algorithm

2.5 Previous Works and Technical Overview

M. Jarman et al have presented an automated Bengali sign recognition system that utilizes the fingertip finder algorithm and a feed-forward neural network [1]. They take into consideration of the protruding fingertips, location of wrist on the binary threshold image to determine the sign that is being made. However, in the Bengali Sign Language there are many similar hand signs namely the hand signs for ঠ and ঞ. Our paper tackles this issue by taking into account the structure of the hand along with the wrist location.

Also, [8] compares various methods that have been utilized for the recognition of signs. This paper aided in finalizing the utilization of Kinect technology due to the several features present in the Kinect. [9,10] describes alternate methods utilized in association with the Kinect sensor.

Neural networks have been utilized in many sign recognition systems [1,11]. Garcia & Viesca have created such a system for ASL. They had utilized a three-layer neural network with one input layer, one hidden layer and one output layer, which we have utilized by incorporating two more hidden layer.

In [12], several recognitions methods have been described and results compared. This provided a straightforward assumption into which methods we deemed most suitable for our task. Whereas, the previous paper provided insight into the algorithms suitable for our task, [13] provided combinations of hardware and software that went along with our desired system.

Grzejszczak et al have described a very unique way to detect and recognize different hand-signs [14]. They focused on finding landmarks in the hand and localization. Theirs is a similar method to this paper utilized as both concentrate on pointing out marking features of the hand structure.

CHAPTER 03

Proposed Model

3.1 System Design

- Block 1:The program is trained using test and training images
- Block 2:An Red-Green-Blue(RGB) image is taken by Kinect
 - The image is converted to grayscale and then to a binary threshold image
 - The image is standardized to a size of 300x300 pixels
- Block 3:Contour Drawing
- Block 4:Unique Joint Detection
- Block 5: Vectors are created from acquired data and sent into SVM
- Block 6:SVM matches most probable hand-sign and display corresponding letter

Figure 5 presents a block diagram corresponding to our proposed model.

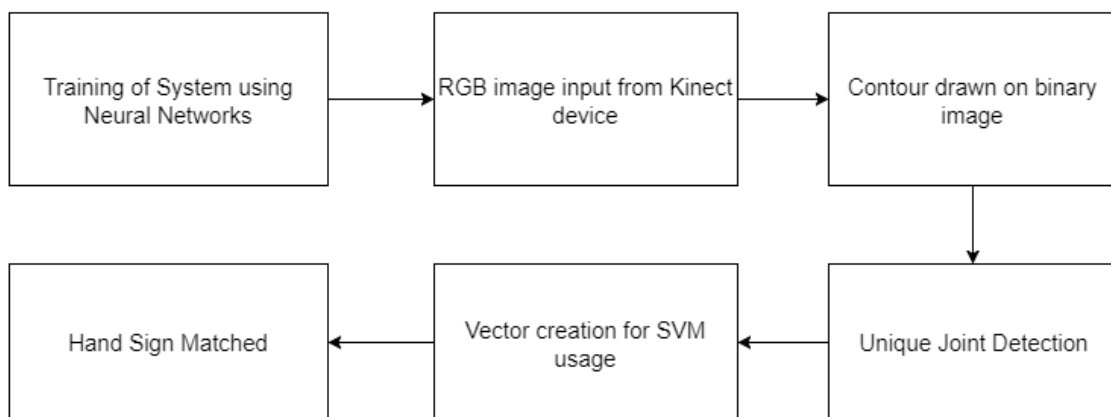


Figure 5: Block diagram of the Proposed Model

The neural network trains the system with the provided test and train images. Upon starting the system, the Kinect will take a picture of the subject in front of it. The algorithm will then continue with the input image and then convert it into grayscale, then threshold binary, after which the image is then cropped to a 300x300 pixel format. Thus, completing the processing of the images

The binary image is then used as the base image when undergoing the other algorithms. The values from the contour detection algorithm is then passed into SVM. Along with the SVM, the positions of located Joints are matched with training images with a leeway of 3 pixels of difference, allowing some variance in the layout.

3.1.1 Algorithm

1. Standardize image
 - a. Convert image to Grayscale
 - b. Convert grayscale image to Binary Threshold image
2. Contour image is using convex hull
3. Wrist is located
4. Joint Detection
5. Vector created from the equations
6. Vectors run through the SVM wrapper
7. Wrapper identifies the hand sign
8. The image matched is located
9. Name of folder (which is the hand-sign name) the image resides in is printed.

3.2 Unique Joint Detection Algorithm

This algorithm has been newly devised for this paper. It attempts to estimate the bone and joint layout in the hand. This is theoretically more accurate than blob and skin detection as those are dependent on vague structure and light. However, as we are already using a Kinect that works independent of light availability both these issues are eradicate.

3.2.1 Algorithm

1. Reads pixel color from right to left, starting from bottom line to top line.
2. Change from black to white pixel is saved in an array as an integer variable called ThresholdBegin. This is the new pointer.
3. From the point of ThresholdBegin, the 3 pixels adjacent to each other on the row of pixels above and lower it is noted, keeping the ThresholdBegin point as center of the three.
4. 3 pixels above will be named as A, B, C. The system stores the pixel name location at which the threshold is found.
5. If, in the above row, if there is another threshold in the same named pixel, then the current pointer is shifted to that position. However, if the threshold above does not match the previous named pixel, that point is considered the Joint Position.
6. Else if, there is no threshold found in the above pixels, the lower 3 pixels named similar is then noted, and a new a threshold is found. This point is considered as a Joint Position.
7. The Joint Positions are saved till fingertip positions are reached by the pointer.

CHAPTER 04

Experimental Results

4.1 Methodology

4.1.1 Dataset

We created our own database as Bengali Sign Language datasets are very elusive. The images were compressed into dimensions of (300x300) for both input and training images. The input image is converted to grayscale and then to binary threshold. Figure 6 is a sample of the dataset we have created.

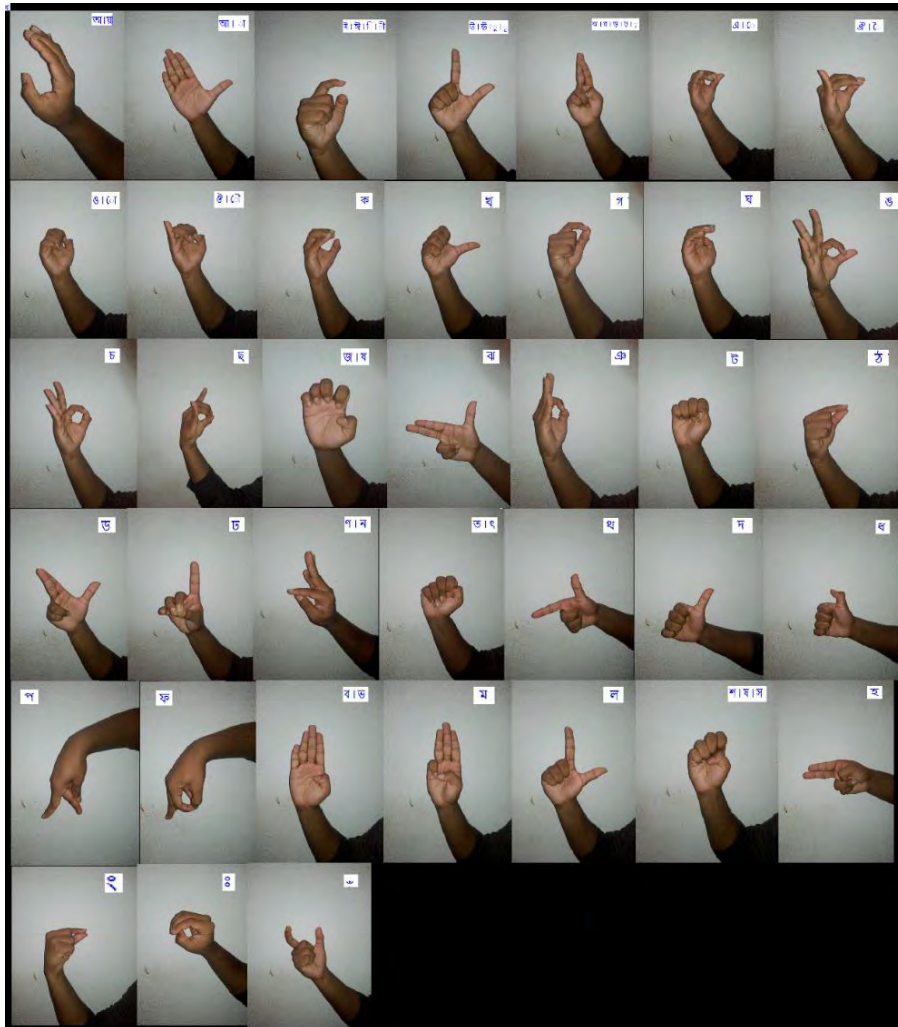


Figure 6: Sample Dataset of Bengali Sign Language

There is a total of 38 hand signs in the Bengali Sign language. The স্বরবর্ণ (Shoroborno) or vowels consists of 9 signs. The ব্যঞ্জনবর্ণ (Benjonborno) or the consonants is a set of 27 signs. The actual Bengali alphabet has a total of 51 letters. However, due to some of the letters having the same phonetics, the same hand sign is used. For example, letter খ (Kho) and ক্ষ (Khio). We created 4 variations of each letter to provide ourselves a large enough data set for the system to utilize. We have followed the image in Figure 7 as our template.



Figure 7: Image acquired online for Bengali Sign Language

4.1.2 Tools used

Kinect Xbox 360 is used as an input device. This device has been manufactured by Microsoft for their Xbox 360 console for motion-sensing purposes and also creating an interface between console and user.

Visual Studio 2015 was used for this system. As there are no inbuilt methods for the Threshold Image, SVM and Neural Networks as MATLAB does, extensions were applied via NuGet. The LIBSVMsharp is a C# wrapper as LIBSVM is not utilized in C# code naturally, as most opt for the in-built SVM of MATLAB. For the threshold technique, we had utilized the Imaging package from AForge.NET in order to apply a grayscale filter on to the input image.

4.1.3 Interface

The following images will display on the interface created for this program as well as show the different outputs and steps taken. Figure 8 is a picture of the threshold image conversion from on the hand sign of the letter Kho (Kho).

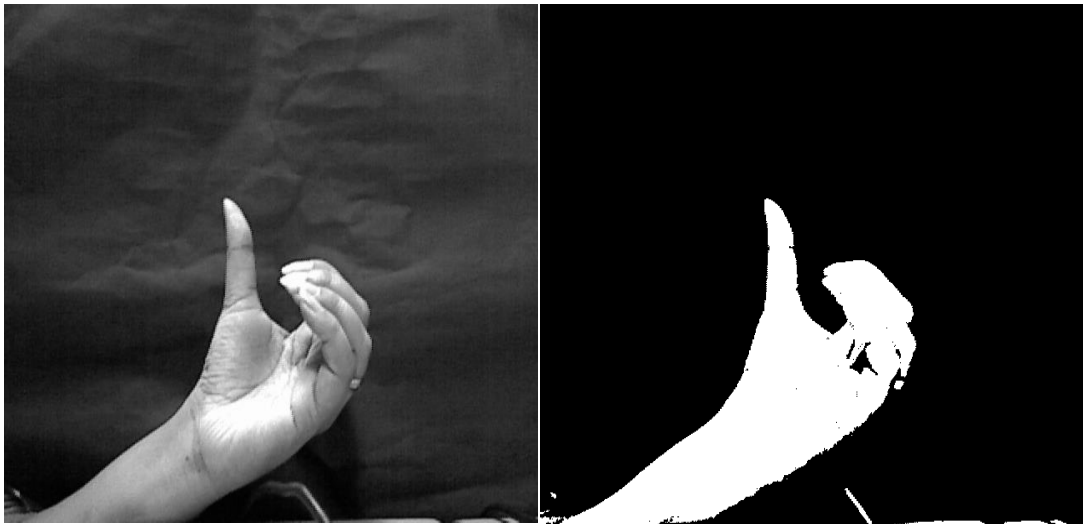


Figure 8: Getting Threshold Image of a Hand Sign

Upon testing out the different contour detection algorithms the output displayed in Figure 9 was achieved. We created an interface to display the contour initially to verify the code's accuracy.

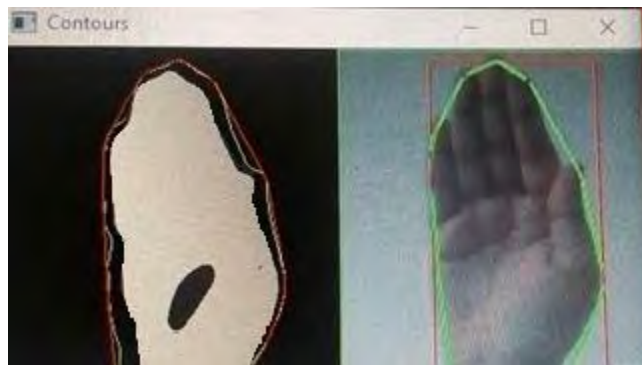


Figure 9: Contour detection of Hand

4.2 Conducting the Tests

We have created another recognition algorithm to compare with ours. The comparative algorithm, and the proposed model are both run 4 times. The time taken to get a result, as well as, the accuracy of the result has been calculated and tested. Figure 11 shows the output achieved by our proposed system, and Figure 12 describe our results for both systems regarding time.

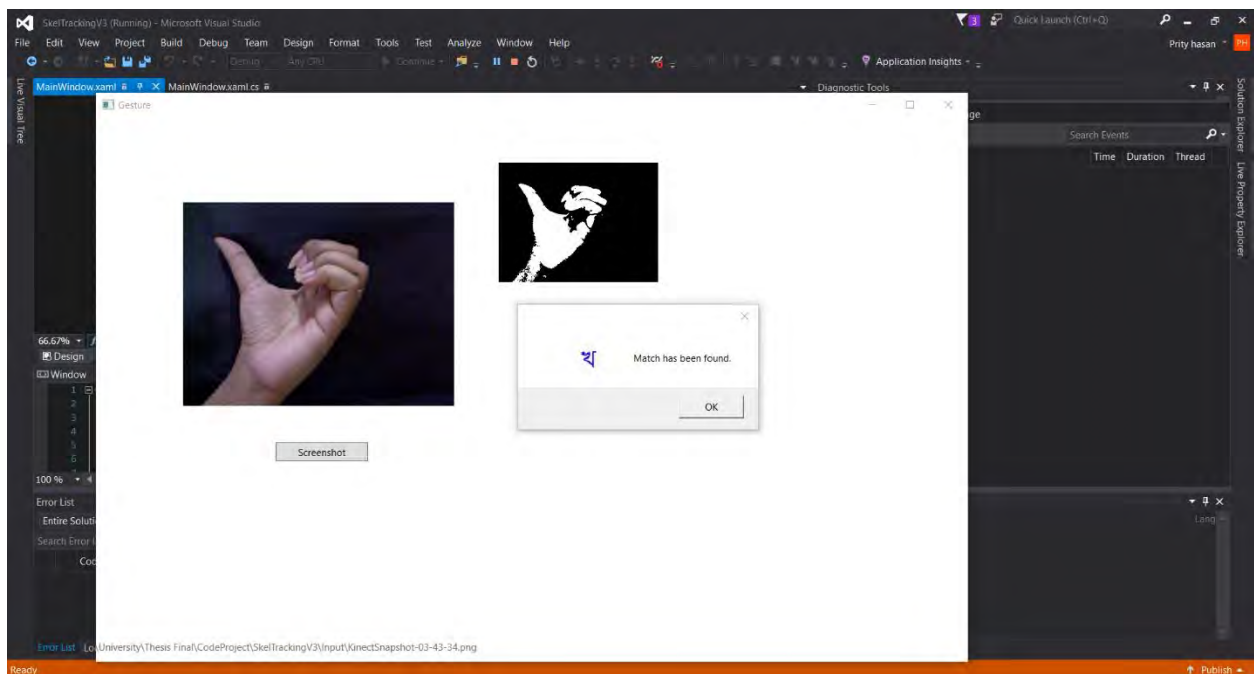


Figure 10: Getting the Result

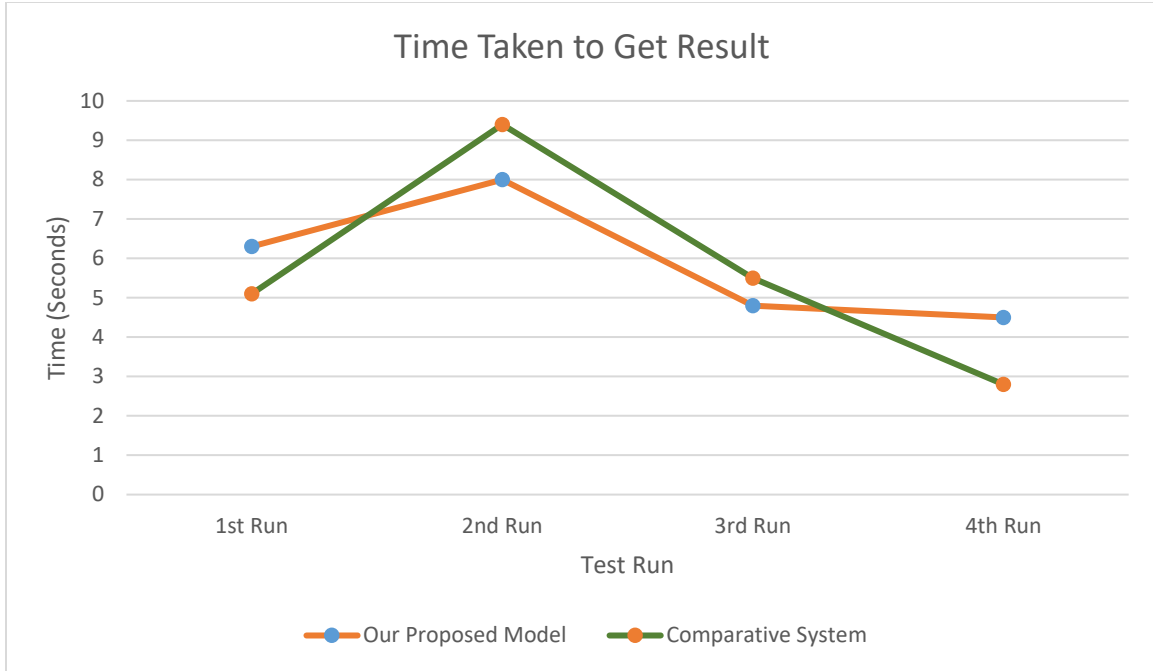


Figure 11: Time Taken Comparison.

For the first run, a simple gesture was tested to be identified, namely **ক** (KO). As can be seen from Figure 8, the hand sign is very prominent and thus, our proposed model was able to easily form its contour. Also, as the finger has several discrete angles the unique joint algorithm is able to detect the angle changes and thus locate the joints. Due to the hand sign being a favorable composition, it took less time for the system to run it

The second run, the letter **খ**(Kho) was used. Upon consulting Figure 8 again, we can see that this letter bares resemblance to **অ** (Shorey-O).

গ(Ga), was the sample utilized for the third run. This hand sign is completely unique in its form and thus, it takes less time for the primary hand locations, i.e. the fingertips, wrist and joints are easily located.

In the fourth run, the **স্বরবর্ণ** (Shoroborno) -**অ**(Shore-O), is used. During this test, the fingertip detection algorithm had completed its task the fastest, allowing the other tasks a head start in comparison to the other hand signs.

Table 1 below illustrates the result we had found for 4 hand signs when conducting 22 tests. The total accuracy achieved from this system is 84.11%.

	Number of run																						
ବର୍ମାଲା (Bornmala)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Accuracy %
ଅ (Shorey-O)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	90.901
କ (Ko)	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	86.374
ଗ (Ga)	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	81.861
ଞ (Kho)	✗	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	77.274

✓	Matched
✗	Not Matched

Accuracy = (No. of times matched / total times tested)*100%

Table 1: Accuracy Result of the Proposed Model

CHAPTER 05

Conclusion and Future Work

5.1 Conclusion

In our proposed model, we have trained and tested datasets using neural network which made our work more reliable, [18] since neural network provides an adaptive learning technique with problem solving capability to complex real world problem like ours. We have used XBOX360 Kinect device to take depth image as input and converted the image to threshold binary image to reduce the noise which was a great help to get better features for extraction. However, a minor trouble we have encountered is the background which needs to be a single colored, since we are not considering depth image. The three basic features we have extracted from the image were contour using convex hull method, fingertip and our self-developed joint location for fair and better result. Choosing these three as features was the best decision which made our work comparatively better because of the accuracy rate of the result analysis. SVM algorithm was run to process the end result based on these the supporting features. We had a little difficulty setting up the XBOX360 Kinect with some other technical set up which can be considered as a minor issue. Experiment result of our model depicts that SVM algorithm has made a strong remark on our work.

5.2 Future Work

In our paper, we have worked on isolated sign which refers to static gesture but we have a plane to work with continuous gesture for real time processing and practical use in formal and official settings. Now, we have only one function which is converting BSL to textual format and we are looking forward to incorporate more features like translation from Bengali language to any other language, creating audio output, full sentence in textual format to make the system more convenient and acceptable. A broader spectrum system for the disabled in collaboration with other fields of machine learning can be developed in the future.

REFERENCE

- [1] Angur M. Jarman, Samiul Arshad, Nashid Alam, J. Islam , “An Automated Bengali Sign Language Recognition System Based on Fingertip Finder Algorithm.” (2015)
- [2] Stergiou, C. (n.d.). What is a Neural Network? Retrieved from https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/cs11/article1.html
- [3] Patil, A. R., & Subbaraman, S. S.(n.d.). A Review on Vision Based Hand Gesture Recognition Approach Using Support Vector Machines. Second International Conference on Emerging Trends in engineering (SICETE), 07-12. Jaysingpur: Dr.J.J. Magdum College of Engineering
- [4] deSouza, C. R., & Pizzolato, E. B. (2013). Sign Language Recognition with Support Vector Machines and Hidden Conditional Random Fields: Going from Fingerspelling to Natural Articulated Words. *MLDM'13 Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition*, (pp. 84-98). Berlin.
- [5]Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A Practical Guide to Support Vector Classification. Taipei: National Taiwan University.
- [6] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (n.d.). Feature Selection for SVMs.
- [7] Chang, Y.-W., & Lin, C.-J (2008). Feature Ranking Using Linear SVM. *JMLR: Workshop and Conference Proceedings* 3, (pp.63-64). Taiwan
- [8] Vijay, P. K., Suhas, N. N., Chandrashekhar, C. S., & Dhananjay, D. K. (2013). Recent Development in Sign Language: A Review (2nd ed., Vol. 1, Rep.).
- [9] Lang, S. (2011). Sign language recognition with kinect. Bachelor, Institut für Informatik, Freie Universität Berlin

- [10] Z. Ren, J. Yuan and J. Meng, "Robust part-based hand gesture recognition using kinect sensor", IEEE Transactions on Multimedia, vol. 15, no. 5, (2013).
- [11] Garcia, B., & Viesca, S. A. (n.d.). Real-time American Sign Language Recognition with Convolutional Neural Networks (pp. 225-232, Rep.).
- [12] Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review, 43(1), 1-54.
- [13] Corera, S., Krishnarajah, N. (2011, June 23). Capturing Hand Gesture Movement: A Survey on Tools, Techniques, and Logical Considerations
- [14] Grzejszczak, T., Kawulok, M., & Galuszka, A. (2015, February 18). Hand landmarks detection and localization in color images. Retrieved from Springerlink: 2015.
- [15] She Yinying, Jia Yunzhe, Gu Ting, An Integrated Approach of Real-time Hand Gesture Recognition Based on Feature Points, International Journal of Multimedia and Ubiquitous Engineering Vol.10, No.4 (2015)
- [16] Nahid Sultana Juthy, Broj Gopal Shaha, Md. Sharafat Ali Shojol. Ishara Bhashay Jogajog (Communicating through Sign Language). Dhaka: Center for Disability in Development, 2005.
- [17] Bor Wang, Adriana Quattoni, Louis-Philippe Morency, David Demirdjian, Trevor Darrell. Hidden Conditional Random Field for Gesture Recognition. Computer Science and Artificial Intelligence Laboratory, MIT
- [18] Alauddin M., Joarder A.H. (2004) Deafness in Bangladesh. In: Suzuki JI., Kobayashi T., Koga K. (eds) Hearing Impairment. Springer, Tokyo