Bangla Sign Language Recognition using Convolutional Neural Network

Farhad Yasir[1], P.W.C. Prasad[1], Abeer Alsadoon[1], A. Elchouemi[2], Sasikumaran Sreedharan[3]
[1]Charles Sturt University Study Centre, Sydney, Australia
[2]Walden University, USA
[3]Department of Computer Engineering, College of Computer Science, King Khalid University, KSA

***Abstract*— This paper presents a learning based approach to Bangla Sign Language(BdSL) recognition using the convolutional neural network. In our proposed method, a virtual reality-based hand tracking controller known as Leap motion controller (LMC) has introduced to track the continuous motion of the hands. LMC provides a skeletal model of the hand with appropriate data of hand position, orientation, rotation, fingertips, grabbing and more non-linear features. This controller preprocessed all the motion features and provides error free data. This machine calibrates with the environment and builds a virtual hand in a space. LMC also calculates the rotation, orientation, and textures from hands to determine and to extract hand gesture. In the next process, an efficient method is established to proceed a sequence of frames for positional hand gestures and summarize them to a shorter and more generalized sequence of lines and curves which are added to a Hidden Markov Model. For each sign of expression, we considered a start and an end point of state and segmented the state transitions into segmented HMM. In the segmentation, we assumed the state scope of the hidden variables is discrete. The transition probabilities controlled the way of hidden state at a distinct time. If there is a histogram difference in any state, the transition state moved to new frame to achieve a new sign expression. If there is no hand gesture in the frame, the state has ended by moving to the end point of the model. In the end point, we evaluated the desired hand gesture for recognition. After evaluation, hand gesture data set are proceeded over the convolutional neural network (CNN) and built a decision network. Each neuron is built up by calculating the dot product of extracted features in the dataset. In CNN, a single vector of hand gesture data is received and connected through a series of hidden layers and in the end point computed as a single vector loss function. Each feature is considered as a hidden layer. Determining the least loss function, the network recognizes the expected sign expression. In our experiment, we considered training data first to create the neurons in our network as a supervised way. We achieved significant results from our basic sign expressions in a 3% rate of error where without distortion the rate reduced to 2%. This is an enormous achievement in the Bangla sign language recognition method.**

***Keywords*— *Bangla Sign Language Recognition, Hidden Markov model, Convolution neural network, leap motion controller*.**

I. INTRODUCTION

In recent years, the interest in sign language recognition has been raised to facilitate the communication between deaf and hearing people. Deaf communities around the world have developed sign language as a medium of communication to communicate with other people. Sign language is a natural language that conveys expressions by simultaneously conjoining hand shapes, palm orientation, movement and location of the hands, arms or body. Sometimes the language combines those factors with facial expressions to express the fluidity of a person's thoughts. Moreover, this language is designed to develop visually transfused sign patterns instead of acoustically dispatched sound patterns [4]. As deaf people can't use verbal communications, which makes this visual spatial language more sophisticated and complex than any spoken language.

However, the appearance and movement of the basic sign language are well explained in the sign language dictionaries, but when it appears in the practical world, people face varieties of differences in phonological, morphological, grammatical and lexical level. Different languages have different sign languages depending on their alphabets and regional expressions. There are multiple sign languages such as American, Arabic, French, Spanish, Chinese, and Indian etc. In Bangladesh, hearing people has to learn the sign language to communicate with the deaf people as there is no proper device or method that works as an interpreter. It is thus difficult for the deaf people to teach their sign to the hearing people. As a result, a distance has been created between the deaf and the society. In addition, Bangla Sign language uses both hands and body gestures simultaneously that makes it more complex to understand. Thus it is necessary to build an interpreter that interprets the sign languages to text or speech that helps the deaf people to communicate with the society. Considering the necessity of Bangladeshi sign language recognition (BdSL), it becomes one of the challenging topics in the field of computer vision and machine learning.

Previously, there are few approaches those tried to resolve this issue and received a respective acceptance. But most of the research works have constraints and the least accuracy in the performance which arise the necessity to introduce a new method that simultaneously received continuous data from deaf people and produce a successful result with a significant performance. In previous works, researchers captured data from static images which is a major constraint in the recognition. So considering all the constraints from the previous works, this project aims to implement leap motion controller to capture motion hand gestures in the virtual reality (VR) that helps to receive data of different features. To capture each sign from the continuous frames we introduced a time series method which used segmented Hidden Markov Model (sHMM). HMM considered the continuous frames into predefined states with proper constraints. Whenever it changes the states the frame is considered as a sign of expression. After receiving the features, we applied Convolutional Neural

Network (CNN) with multiple hidden layers and built neurons in a network. We execute our training and testing data set over this network and achieved the least error rate which enhances the recognition of Bangla Sign Language. This method improves the performance and accuracy rate in the field of Bangla Sign Language Recognition.

II. PROPOSED METHOD

We decided to introduce Leap Motion Controller (LMC) in our project for detecting hand gesture. LMC helps us to preprocess the continuous frames, capture hand and finger data and extract features from the frames. But it is a challenge to segment each sign from the continuous frames. So we introduced Hidden Markov Model (HMM) to segment the sign from the time series. This segmented HMM divided the continuous frames into the specific state. Whenever there are any changes in the state it considered as a new sign. After the time series recognition, we proposed Convolutional Neural Network (CNN) on training features and build a multilayer network which produces a single vector result for each sign. This resultant vector is the output of the recognition of each sign. Later, we computed the result and plot a graph of the error rate of the recognition.

A. Pre- processing with Leap motion controller

Two handed gestures are used for conveying expressions in BdSL. We have chosen LMC which is a small USB device that tracks the position of a hand in a space. It provides a skeletal model of the hand with appropriate data of hand position, orientation, rotation, fingertips, grabbing and more non-linear features. This controller preprocessed all the raw data and provides an error free features. It removes the garbage collection and creates virtual hands in the virtual reality. The machine calibrates with the environment and builds a virtual hand in a 3D world. LMC also calculates the rotation, orientation, and textures from hands which we considered as our vital features that we extracted later to form a decision tree for each sign expression.

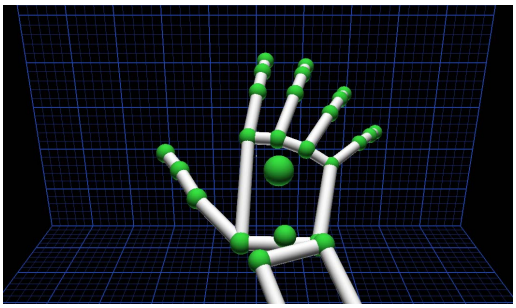

Figure 1 Leap motion controller with hand visualizer

B. Feature Extraction

LMC provides non-linear features of the hand gestures it captured from the continuous frame. We considered hand position, orientation, rotation, and hand grabbing and palm orientation as our features. These are all non-linear functions that we need to proceed with a multilayer network for recognition. We divided our collected features into hand, finger, and gesture.

1) Hand data

- Type – This feature describes the either the hand is left hand or right. It considers the hand orientation and the thumbs finger to detect the hand type.
- Type – This feature describes the either the hand is left hand or right. It considers the hand orientation and the thumbs finger to detect the hand type.
- Palm position- As LMC considers hands as a 3D object, it takes the texture and the palm position by computing the depth of the frames and the colors.
- Grab strength- Finger position and stress detection are used to calculate the grab strength of a hand
- Pinch strength- The holding strength of a pinch hand pose. The strength is zero for an open hand and blends to 1.0 when a pinching hand pose is recognized. Pinching can be done between the thumb and any other finger of the same hand.
- Confidence – It measures how strict the internal hand model that fits the observed data. A low value indicates that there are significant discrepancies of finger positions, even hand identification could be incorrect.
- Arm – The basic vectors for position, orientation, and size of forearms are measured by LMC as a hand data.
- Translation – This feature provides the hand translation in the virtual reality to understand the actual position of the hand
- Rotation - Hand rotation is measured as vectors in Leap motion controller. We consider it as one of our main features to process the hand in the standard position
- Scale factor – This feature is optional as we can have scaled the hand to get a proper translation and rotation vectors from the continuous frames.

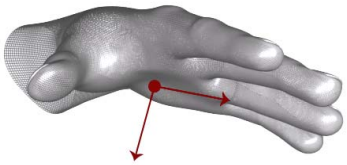

Figure 2 Palm orientation and direction in LMC

2) Finger data

- Type – This feature is applied on the fingers of a hand and detected the finger types. There are five different types of fingers: Thumb, Index finger, Middle finger, Ring finger and Pinky finger.

Figure 3 Fingers basic vectors for direction

- Metacarpal bone: This bone is figured out in Figure 4. LMC computed the center vector, direction vector and up vector for metacarpal bone
- Proximal phalanx bone: This bone is figured out in Figure 4. LMC computed the center vector, direction vector and up vector for proximal phalanx bone
- Intermediate phalanx bone: This bone is figured out in Figure 4. LMC computed the center vector, direction vector and up vector for intermediate phalanx bone
- Distal phalanx bone: This bone is figured out in Figure 4. LMC computed the center vector, direction vector and up vector for distal phalanx bone.

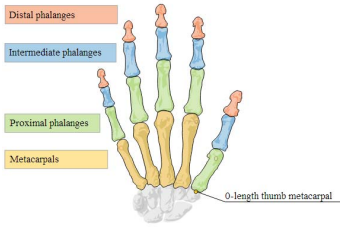

Figure 4 Finger bones identification

C. Time series detection for sign expression using sHMM

An efficient method is proposed to process a sequence of frames for positional hand gestures and summarize them to a shorter and more generalized sequence of lines and curves which are added to a Hidden Markov Model as a discrete state. So for each sign of expression, we will consider a start and end points of states using discrete data of HMM. We will also consider the state transitions for each of the sign and segmented HMM with respect to those transitions.

In this purpose, we considered the fingers directional vector as our state transitions. For each of the finger, we separated our states and calculate a probability distribution function for each state. We started our transition state from frame state. From any of the state, method can travel to frame state. We considered 5 states for 5 different types of fingers. If any of the finger transition state changes, it travels back to the frame state to pursue a new sign from the continuous frame.

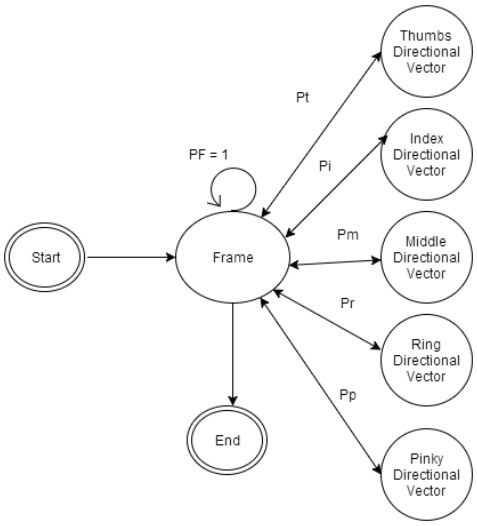

Figure 5 sHMM state transition for time series recognition

In figure 5, a time series state transition model is introduced to separate the sign of expressions from the continuous frame. Starting point redirects to the frame state. The probability of frame state is PF = 1. As the frame state is the continuous visiting state in the model. There is 5 independent state for 5 fingers consists the conditional probability distribution which is Pt, Pi, Pm, Pr, and Pp respectively. We assumed that each frame contains hand sign in the model and for each hand, we separated the 5 fingers from the hand with directional vectors. Let's assume that the directional vector function $x = y(t)$ where t is the time of that moment. So for each of the frame changes the probability function for each finger state is below.

$$P_t = \frac{\mathrm{y(t)\text{-}}y(t-1)}{\sum_t y(t)} \quad (1)$$

In this equation, the probability distribution for each state is considered as the difference of directional vector. If there is a distinct difference in the time series, it is considered as a sign expression changes in the frame, so the state goes back to the frame state to achieve new sign expression.

We assumed the state scope of the hidden variables is discrete in the hidden Markov model. The transition probabilities controlled the way of hidden state at time t. If there is a histogram difference in any state, the transition state moved to frame state to achieve a new sign expression. If there is no hand gesture in the frame, the state has ended by moving to the end point of the model.

D. Convolutional neural network

We considered a set of layers to process our BdSL features which are captured in the leap motion. Our CNN architecture is designed on basis of the features from the leap motion controller. So we started with the input layer consists of all features that received by the LMC. After that moved the

features as a parameter to the next layer which is the convolutional layer. The finger features are considered as a hyper parameter. After Convolutional layer, we passed the output to the pooling layer which measured the comparison and the loss function and forwarded to the fully connected output layer.

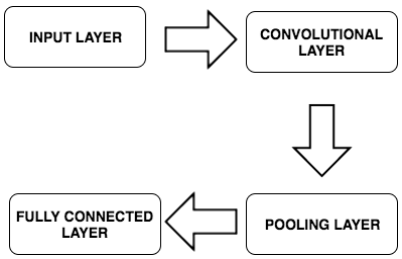

Figure 6 Layers on processing BdSL features

In CNN, each layer shares the parameter and pass the hyper parameter if there is any. In our case, we dealt with high-dimensional inputs from the input layer and scaled those in fixed 32x32 images. Instead of connecting each neuron to all the previous neurons, we connected the neuron to a local region of the input volume. The spatial extent of the connectivity is a hyper parameter which is called the receptive field of the neuron. The respective field also consists of the depth of the neuron as a requirement of the convolutional network.

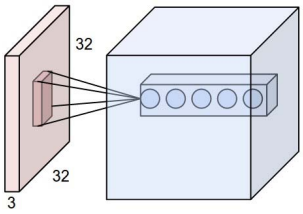

Figure 7 Depth of the Convolutional Layer

The neuron in each layer computed the dot product of the weights with the input features which produced a non-linearity. The neuron connectivity is bounded in the local spatial as shown in Figure 7.

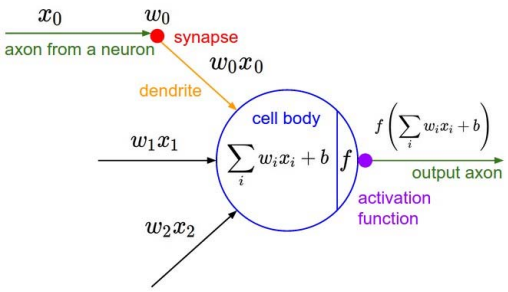

Figure 8 Neuron model for convolutional neural network

In the spatial arrangement of each neuron, depth, stride, and padding are attached to complete the parameters. We computed the size of the spatial as a function of the input features which is given below where volume is W, receptive parameters are F, stride that applied to the neuron is S and padding is considered as P.

$$Size = \frac{W\text{-}F + 2P}{S} \quad (2)$$

Backward pass for convolution is also introduced to derive the dimensional depth in the layer which is considered as an improvised version of backward propagation.

We introduced a pooling layer in between the convolutional and the fully connected layer to normalize and to reduce the amount of parameter which fastens the performance. This layer functionally progressed with reducing the spatial size and the respective parameter and compute for the next later network. This layer generated independently on each depth to reduce the spatial size for maximizing the output accuracy.

After passing through the pooling layer, all the parameters are converged with a single layer vector which is considered as an output vector. In this layer all the neurons from the previous layer's connection to a fully connected single neuron. If the resultant function meets the threshold of the desired sign expression, it considered as a successful ride through the convolutional neural network.

III. Experimental Results

We have received a limited number of data from leap motion controller, as we discussed earlier, we considered the hand and finger specific features in out input layer of the convolutional neural network. We generated the error rate from the few data we received. We considered training data first to create the neurons in our network as a supervised way. In the training dataset, we received a low range of error rate. In the testing data, the results are higher but acceptable for recognition. We used OCTAVE to generate our error function and plot it in figure 8 and 9.

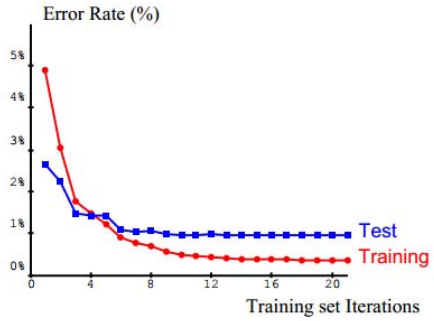

Figure 9 Error rate of CNN over BdSL recognition using both training and test data set

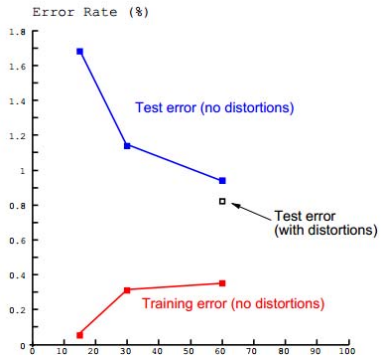

Figure 10 Distortion error rate of CNN over BdSL recognition using both training and test data set

In the experiment, we achieved the significant result from our basic sign expressions in a 3% rate of error where without distortion the rate reduced to 2%. This is an enormous achievement in the recognition method.

IV. Conclusion

In last decade, sign language recognition is a challenging topic in computer vision and machine learning. In recent approaches, the continuous image is captured and segmented on the initial segmentation stage. Researchers considered the movement of hand gestures and facial expressions as an additional feature. But all those approaches are experimental and still developing. In this report, we considered leap motion controller to capture the continuous frame and preprocessed features. We extracted the vital features from hand and fingers using LMC. We introduced segmented HMM to separate sign of expression from the continuous frame using transition states. After fetching the expression, we executed all the features as an input layer we passed all of them as the parameter to the convolutional layer. In the convolutional layer, it passed all the hidden layer and finally processed in the pooling layer to normalize and to reduce the spatial neurons which fasten the performance of recognition. As we have managed to receive the limited data set, so we achieved a significant rate of error which established our method as robust from the previous work. But in future, we have planned to execute huge amount of data set to firm our approach on this topic. We will work on this issues to make our approach as a valid one in the machine learning field. Also in future, we would like to introduce more additional feature as body movements and facial expressions to tenacious the Bangla Sign Language Recognition.

References

[1] Boulares,M, Jemni,M. (2012). 3D motion trajectory analysis approach to improve Sign Language 3D-based content recognition. *Proceedings of the International Neural Network Society Winter Conference (INNS-WC 2012)*, (pp. 133 – 143).

[2] ChandraKarmokar, B., Md. Rokibul Alam, K., & Kibria Siddiquee, M. (2012). Bangladeshi Sign Language Recognition employing Neural Network Ensemble. *International Journal of Computer Applications*, 43-46.

[3] Ildarabadi, S., Ebrahimi, M., & Pourreza, H. (2014). Improvement Tracking Dynamic Programming using Replication Function for Continuous Sign Language Recognition. *International Journal of Engineering Trends and Technology*, 97-101.

[4] Kong, W., Ranganath S. (2013). Towards subject independent continuous sign language recognition: A segment and merge approach. *Pattern Recognition*, 1294–1308

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25-33.

[6] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2278 - 2324.

[7] M. Hrúz, J. Trojanová, M. Železný. (2012). Local Binary Pattern based features for sign language recognition. *Pattern Recognition and Image Analysis*, 519-526.

[8] McCartney, R., Yuan, J., & Bischof, H.-P. (2015). Gesture Recognition with the Leap Motion Controller. *International Conference on Image Processing, Computer Vision, & Pattern Recognition.*

[9] Nair, A.,Bindu,V. (2013). A Review on Indian Sign Language Recognition. *International Journal of Computer Applications*, 33-38.

[10] Nasri, S., Behrad, A., & Razzazi, F. (2014). A novel approach for dynamic hand gesture recognition using contour-based similarity images. *International Journal of Computer*, 662–685.

[11] Phadtare, L.K. , Kushalnagar, R.S. ,Cahill, N.D. (2012). Detecting hand-palm orientation and hand shapes for sign language Gesture recognition using 3d images. *Image Processing Workshop (WNYIPW)* (pp. 29 - 32). New York, NY: IEEE.

[12] Potte,L. E. ,Araullo, J.,Carter, L. (2013). The Leap Motion controller: a view on sign language. *5th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, (pp. 175-178).

[13] Rahaman, M., Jasim, M., Ali, M., & Hasanuzzaman, M. (2014). Real-Time Computer Vision-Based Bengali Sign Language Recognition. *17th Int'l Conf. on Computer and Information Technology*, (pp. 192-197). Dhaka.

[14] Rahim F. ,Mursalin T., Sultana. N. (2010). Intelligent Sign Language Verification System – Using Image Processing, Clustering and Neural Network Concepts. *International Journal of Engineering Computer Science and Mathematics*.

[15] Sansanee,A.,Suwannee,P. Wattanapong ,S. , Phonkrit,C , Nipon,T. (2013). Thai sign language translation using Scale Invariant Feature Transform and Hidden Markov Models. *Pattern recognition Letters*, 1291–1298.

[16] Wang ,X., Liang,J., Wang,M. (2013). On-line fast palmprint identification based on adaptive lifting wavelet scheme. *Knowledge-Based Systems*, 68-73.

[17] Yang, H.,Lee, S. (2013). Robust sign language recognition by combining manual and non-manual features based on conditional random field and support vector machine. *Pattern Recognition Letters*, 2051-2056.

[18] Yasir, F., Alsadoon , A., Prasad, P., & Elchouemi, A. (2015). SIFT based approach on Bangla sign language recognition. *2015 IEEE 8th International Workshop on Computational Intelligence and Applications* (pp. 35 - 39). IEEE.

[19] Zaki,M. M., Shaheen,S. I. . (2011). Sign language recognition using a combination of new vision based features. *Pattern Recognition Letters*, 572-577.