Sign Gesture Recognition from Raw Skeleton Information in 3D Using Deep Learning

Sumit Rakesh, Saleha Javed, Rajkumar Saini(✉), and Marcus Liwicki

Luleå Tekniska Universitet, Luleå, Sweden
sumrak-0@student.ltu.se,
{saleha.javed,rajkumar.saini,marcus.liwicki}@ltu.se

Abstract. Sign Language Recognition (SLR) minimizes the communication gap when interacting with hearing impaired people, i.e. connects hearing impaired persons and those who require to communicate and don't understand SLR. This paper focuses on an end-to-end deep learning approach for the recognition of sign gestures recorded with a 3D sensor (e.g., Microsoft Kinect). Typical machine learning based SLR systems require feature extractions before applying machine learning models. These features need to be chosen carefully as the recognition performance heavily relies on them. Our proposed end-to-end approach eradicates this problem by eliminating the need to extract handmade features. Deep learning models can directly work on raw data and learn higher level representations (features) by themselves. To test our hypothesis, we have used two latest and promising deep learning models, Gated Recurrent Unit (GRU) and Bidirectional Long Short Term Memory (BiLSTM) and trained them using only raw data. We have performed comparative analysis among both models and also with the base paper results. Conducted experiments reflected that proposed method outperforms the existing work, where GRU successfully concluded with 70.78% average accuracy with front view training.

Keywords: Sign gesture · SLR · Recognition · Deep learning · BiLSTM (BLSTM) · GRU · Microsoft Kinect · HMM

1 Introduction

Sign language is expressed through visual gestures and physical signs performed by a person. Usually categorized into static and dynamic gestures [2]. A static gesture is spatial and it neither has significant motion nor time-dependency [23]. Whereas, a dynamic gesture is a spatio-temporal sequence that involves body motion often by hands and seldom with help of legs as well [24]. But there are many more varients of gestures around the globe and across nations. Some of the gestures are expressed with same hand motion therefore different facial expression/motion are used to distinguish between them. For instance in the

© Springer Nature Singapore Pte Ltd. 2021
S. K. Singh et al. (Eds.): CVIP 2020, CCIS 1377, pp. 184–195, 2021.
https://doi.org/10.1007/978-981-16-1092-9_16

regional sign language in the state of Uttarakhand of India, the gestures for *what* and *who* [14] have similar hand motion, hence they are distinguished by face. Furthermore, sign gestures are both single-handed and double-handed. Some gestures are performed with finger spelling [13] e.g. the word *cool* is expressed by performing gesture for each letter in the order $c \rightarrow o \rightarrow o \rightarrow l$. Another baseline observation is that the structure of sign language not only varies nation to nation, it varieties even region to region. Indian Sign Language (ISL) [12,26] typically follows the grammar order as *subject*→*object*→*verb* (SOV). Whereas, American sign language [1] typically uses *subject*→*verb*→*object* (SVO). In addition, the same gesture [25] may be used to express different words or meanings.

Researchers have been thriving to design the SLR system at character, word, and sentence level. In order to model a sign language recognition system it entails to record the actual physical sign gestures with the help of a sensor such as video cameras. The challenge here was to analyze videos to extract 3D information from it. The advanced technologies like Microsoft Kinect are capable of capturing 3D gestures with high precision and accuracy. It captures the skeleton of the human body and typically returns 20 joints in 3D. Figure 1 shows a typical look of the Kinect sensor, its axis of recording skeleton, and the 20 joints with their names.

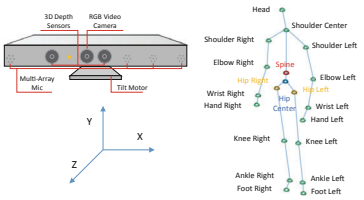


Fig. 1. Kinect, axis, and skeleton joints.

Mehrotra et al. [18] developed an ISL system suing 11 out of 20 joints corresponds to upper body motion. They used 34 low-level features per frame and used Support Vector Machine (SVM) for the recognition. Similarly, Gangrade and Bharti [6] also used 11 joints with SVM for classification. Ghotkar and Kharate [7] compared a rule-based and Dynamic Time Warping (DTW) based approaches for the recognition of 10 sign gestures. These systems are based on the frontal view of the skeleton. Kumar et al. [15] proposed a SLR considering different rotation views. The signers were requested to rotate themselves (not just facing Kinect sensor frontally). They developed an Indian SLR system with low-level features on 30 sign gestures. They used Hidden Markov Model (HMM) [19] for gesture recognition. However, we aim at developing a system from raw data itself without extracting handmade features. Deep learning classifiers such as GRU (Gated recurrent unit) [8] and BiLSTM (Bidirectional Long Short Term Memory) [10] are good at modeling sequential data and have been applied in variety of applications such as handwriting [16], Natural Language Processing

(NLP) [11,22], time-series [5], activity recognition [20]. Thus, we investigated them on the SLR data set developed by Kumar et al. [15].

In this paper, we have proposed a deep learning model for sign gesture recognition. The contributions to the paper are (i) User independent sign gesture recognition system using raw sign gestures. No preprocessing (except normalization *zerocenter*, i.e. subtracting mean of each gesture from all it's frames) is done. No handmade features are used. (ii) Extensive experiments with a comparison between BiLSTM and GRU. (iii) The system outperforms the base model [15] roughly by 10% on raw sign gestures. The rest of the paper is organized as follows. Section 2 shows the proposed methodology along with used deep network architecture in detail. Experimental details and the results are discussed in Sects. 3, and 4 respectively. Finally, we conclude in the Sect. 5.

2 Proposed Methodology

2.1 Architecture

Here, we describe the network architecture used in this work. The architecture of the proposed system is depicted in Fig. 2. There are five layers in the proposed architecture, namely, *Sequence Input* (SI), *Gated Recurrent Unit* (GRU), *Fully Connected* (FC), Softmax, and *Classification*. The raw sign gestures are 60-dimensional temporal sequences, therefore, the input size in the SI layer is 60. The SI layer takes raw gestures as input and normalizes (*zerocenter*, i.e. subtracting mean of each gesture from all frames of that gesture) them. The normalized gestures are fed into the GRU layer. GRU layer processes the normalized 60D gestures and learns dependencies between time steps in the gesture sequences. The *tanh*, and *sigmoid* have been used as state and gate activation functions in the GRU layer. The output size of the layer is set to 120, therefore, it learns 120 features. The output of GRU is fed into the FC layer. FC layer learns the weights and biases of the network and outputs 30 real values that are processed by the Softmax layer. The softmax layer transforms the real values into probability like scores. The network attempts to minimize the multi-class cross-entropy loss in the classification layer. The hyperparameters of the network are discussed in Sect. 3.

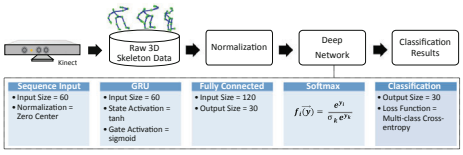


Fig. 2. Architecture of the proposed system with GRU.

2.2 BiLSTM

Bidirectional Long Short Term Memory (BiLSTM) is an extension of standard LSTM that improves model performance on sequence classification problems. As it gives access to input features from both past and future for all time frames of the sequence classification task, thus bidirectional LSTM networks performs efficiently in such models [10]. In this model forward states are used to capture past features and backward states are used to make use of future features. In this work, we have implemented BiLSTM over 60 and 120 hidden units. There are three gates in a LSTM cell, including an input gate, a forget gate, and an output gate [8]. The gated structure, especially the forget gate, helps LSTM to be an effective and scalable model for several learning problems related to sequential data. At time t, the input gate, the forget gate, and the output gate, denoted as i_t, f_t, o_t respectively. The input gate, the forget gate, the output gate and the input cell state, which are represented by colorful boxes in the LSTM cell in Fig. 4, can be calculated using the following Eqs. (1, 2, 3, 4) [4]:

$$f_t = \sigma_g(W_f x_t + U_f.h_{t-1} + b_f) \tag{1}$$

$$i_t = \sigma_g(W_i x_t + U_i.h_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma_g(W_o x_t + U_o.h_{t-1} + b_o) \tag{3}$$

$$\tilde{C}_t = tanh(W_C x_t + U_C.h_{t-1} + b_C) \tag{4}$$

where W_f, W_i, W_o, and W_c are the weight matrices mapping the hidden layer input to the three gates and the input cell state, while the U_f, U_i, U_o, and U_c are the weight matrices connecting the previous cell output state to the three gates and the input cell state. The b_f, b_i, b_o, and b_c are four bias vectors. σ_g is the gate activation function, which normally is the sigmoid function, and $tanh$ is the hyperbolic tangent function. Based on the results of four above equations, at each time iteration t the cell output state C_t, and the layer output, h_t, can be calculated using Eqs. (5, 6) [4]:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

2.3 GRU

A gated recurrent unit (GRU) was proposed by Cho et al. [3] as an improved variant of RNN that offers advantages like higher performance efficiency in computation but with lower complexity [8]. The model structure in Fig. 4 shows that GRU includes two gates; r_t and z_t; r (reset gate) that controls the contribution of new input with the previous memory value kept in z (update gate). In our experiments we have used 60 and 120 hidden units configuration to extract maximum variation with the tests. Up-date gate (z_t) ensures which part of the current hidden state h_t should be updated through the candidate hidden state c_t.

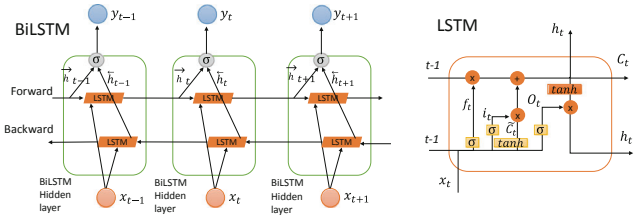


Fig. 3. BLTSM structure and LSTM cell.

Reset gate (r_t) tells that which part of the previous hidden state h_{t-1} can be ignored. For a time series sample set $x = x_1, x_2, ..., x_t, ..., x_T$, the GRU computations are shown in following Eqs. (7, 8, 9, 10, 11, 12) [9] (Fig. 3)

$$z_t = \sigma(V_{xz}.X_t + W_{hz}h_{t-1} + p_z) \tag{7}$$

$$r_t = \sigma(V_{xr}.x_t + W_{hr}.h_{t-1} + p_r) \tag{8}$$

$$c_t = tanh(V_{xc}.x_t + W_{hc}(r_t \odot h_{t-1}) + p_c) \tag{9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot c_t \tag{10}$$

$$\sigma(t) = 1/(1 + e^{-t}) \tag{11}$$

$$tanh(t) = (e^t - e^{-t})/(e^t + e^{-t}) \tag{12}$$

where x_t is the t^{th} sample in the time series sample set, c_t is the candidate state, h_t is the hidden output, σ is the activation function of update and reset gates, tanh is the activation function of candidate state, $\odot$ refers to dot product operation, V_{xz}, V_{xr} and V_{xc} represent the weight vectors between input layer and update gate, reset gate and candidate state, respectively, W_{hz}, W_{hr} and W_{hc} represent the weight vectors of the cycle connections, p_z, p_r, and p_c are corresponding biases.

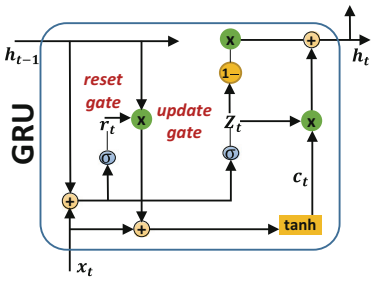


Fig. 4. A GRU cell.

We trained the sign gesture recognizer using the network depicted in Fig. 2. The dataset, experimental protocols, and results are discussed in the next section.

3 Experimental Setup

We designed variety of experiments in order to evaluate performance of both selected techniques i.e. BiLSTM and GRU over extensive tests. Network used for experiments is the one we discussed in Sect. 2 that we tested on the dataset developed by Kumar et al. [15]. As our focus was to evaluate our method against the one presented in base paper thus we also used cross-validation method. Each user U^i is tested while the system is trained with all other users $\{U^j | \, j \neq i\}$. This method is called leave-one-user-out strategy and is explained further in Sect. 3.2. Purpose of this strategy is to conduct extensive experiments over given dataset in manner that dataset itself is used for bringing much more variety of different combination of tests.

3.1 Dataset Description

The dataset [15][1] consists of 30 isolated sign gestures of Indian Sign Language (ISL) collected with the help of *Anushruti Academy for the Deaf*, and *IIT Roorkee*, Roorkee campus, Haridwar, Uttarakhand, India. The sign gestures were been performed by 10 different signers. Each gesture has 9 instances. Hence, the dataset size is 2700 (i.e. $30 \times 9 \times 10$). The dataset has gestures from three different directions, namely, *Front view, Side view 1 (mid),* and *Side view 2 (side).*

3.2 Experimental Protocol and Network Hyperparameters

All the experiments were done in user-independent mode. For this, we trained the system in *leave-one-user-out* strategy i.e. the network was trained using the gestures from 9 signers and tested the gestures of the last signer at a time. This was done for each signer. Thus, 10 different models were trained. Experiments were also conducted view-wise. All the experiments were conducted on normalized gestures without any handmade features. The hyperparameters used

Table 1. Hyperparameter settings

Hyperparameter	Value
Number of epocs	50
Mini batch size	30
Optimizer	Adam
Initial learning rate	0.01
Learning rate drop period	2
Learning rate drop factor	0.27
Gradient threshold	0.6
Shuffle	Every epoc

[1] http://parimal.iitr.ac.in/dataset.

in this work are shown in Table 1. The hyperparameters were chosen based on the validation on the gestures of signer *apurve*.

4 Results and Discussion

Experiments were carried out with training from different views and hidden units with BiLSTM and GRU. Table 2a shows the gesture recognition performance (average) of BiLSTM and GRU (hidden units = 120) when the training was done on front view (user-independent). Rows correspond to test results on the gestures from different views (front, fid, side). GRU (70.78%) outperforms BiLSTM (58.89%) by 14.89%. Similarly, GRU outperforms BiLSTM by 12%, and 7.45% when the system was trained on *mid* (Table 2b) and *side* (Table 2c) views respectively. Apart from the training from individual views, we also trained the system with gestures from all three views. Table 2 the gesture recognition performance where gestures from all views (say *mixed view*) were used in training. The average performances of GRU and BiLSTM are 69.59% and 58.22% respectively.

Table 2. Recognition performance when trained on different views with 120 hidden units

(a) Training = Front view

%	**BiLSTM**	**GRU**
Front	55.89	70.78
Mid	24.89	32.78
Side	9.89	15.78

(b) Training = Mid view

%	**BiLSTM**	**GRU**
Front	23.00	24.22
Mid	48.89	60.89
Side	34.00	45.00

(c) Training = Side view

%	BiLSTM	GRU
Front	13.56	15.78
Mid	33.00	38.33
Side	42.33	49.78

(d) Training = Mixed view

%	BiLSTM	GRU
Front	62.89	74.78
Mid	60.67	71.00
Side	51.11	63.00
Mixed	58.22	69.59

Though the GRU (hidden units = 120) has an average performance 69.59% on *mixed view*, it improves the performance. Figure 5 shows the recognition rate on individual view when the system was trained on *mixed view*. The horizontal axis title shows the rate when training was done on individual views. The *mixed view* improves the performance for each view. Figure 6 shows the extended results for all views per user. Figure 7 shows the validation and test accuracies of each signer with GRU (120). The validation accuracy is network's best performance for each signer during network training, whereas, the test accuracy is when the network has finished training. Thus, the test accuracy is the correct measure for system performance.

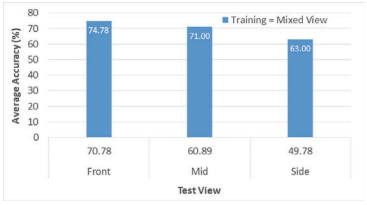


Fig. 5. Average sign gesture recognition performance view-wise when trained on Mixed view.

Fig. 6. Sign gesture recognition performance for each signer view-wise.

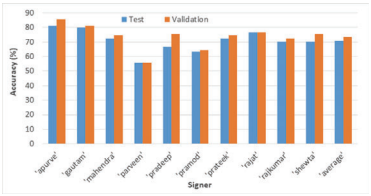
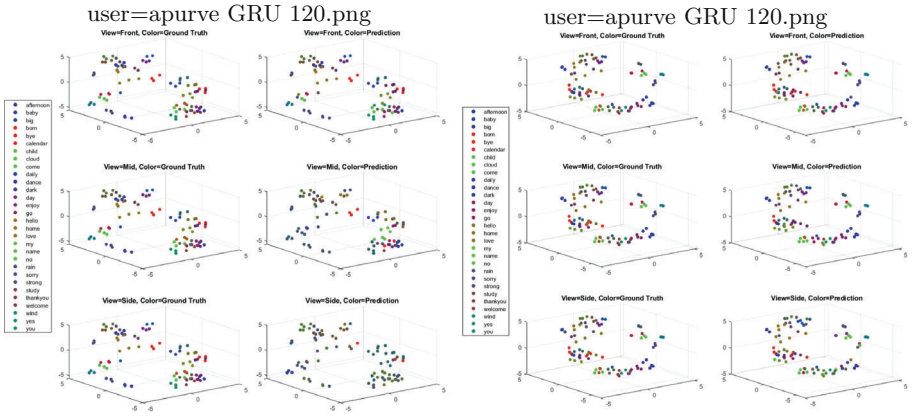


Fig. 7. Validation and Test accuracy of each signer GRU (120).

Confusion matrices are also computed. Figure 10 shows the confusion matrices of systems trained on different views. There are 30 instances of each gesture in each view (*front, mid*, and *side*). Hence, the sum of each row is 30. The diagonal shows the correct recognition, rest elements show the number of instances incorrect recognition in other gesture classes. It is to note that there are 90 samples for each gesture class in *mixed view*. Therefore, the sum of each in Fig. 10d is 90.

The network learns 120 features (activations) for each sample in GRU layer (say, GRU(120)) which is hard to visualize. Thus, we used t-sne [17] to reduce the dimension from 120 to 3. Figure 8a and Fig. 8b show the 3D t-sne visualizations of sign gestures performed by the signer *apurve* for *front*, and *mixed* views respectively. Classes are coded into colors. It can be noticed that GRU is able to learn class distinguishable features. GRU in *Mixed view* learns better and improves performance in each view as shown in Fig. 5.

We also made experiments using both the BiLSTM and GRU with 60 hidden units in the respective layer. Figure 9 shows the recognition rate on all views using GRU and BiLSTM.



(a) Training = front view. (b) Training = mixed view.

Fig. 8. t-sne visualization of features (apurve) learned by GRU (120). Color represents gesture class. (Color figure online)

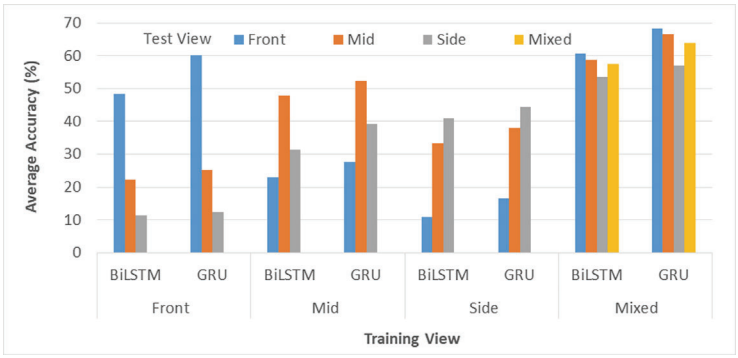
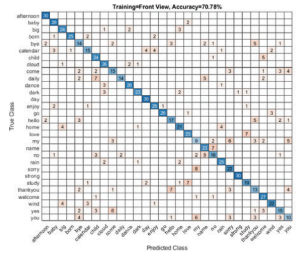
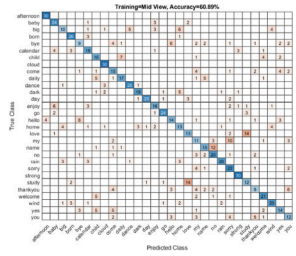


Fig. 9. Average recognition performance when trained on different views with 60 hidden units.

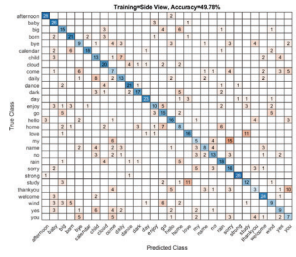
Pradeep et al. [15] showed the performance of their system using the raw data on the front view where they recorded the accuracy of 60.67%. Our proposed model, GRU(120), outperforms their work by 10.11%. All the experiments were done in *Matlab R2020a* (execution mode = CPU) installed on *Lenovo ThinkPad* with 16 GB of RAM. The 50 epoch of GRU and BiLSTM roughly took 3 and 7 min respectively (Table 3).



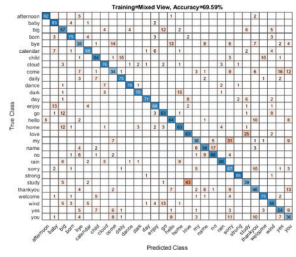
(a) Training = Front View.



(b) Training = Mid View.



(c) Training = Side View.



(d) Training = Mixed View.

Fig. 10. Confusion matrices

Table 3. Average performance comparison

%	Accuracy
Pradeep et al. [15]	60.67
BiLSTM(60)	48.33
GRU(60)	60.11
BiLSTM(120)	55.89
GRU(120)	70.78

5 Conclusion

Sign gesture systems are important as unlike other subjects sign language is not taught in schools or colleges for normal kids or adults in many countries. Usually, people start learning sign language when they have hearing/speech impaired kids in the family. However, this is not enough as hearing/speech impaired people interact with others in the society who don't know sign language. SLR systems can bridge that gap. In this paper, we developed a sign gesture recognition system from raw data in the user-independent scheme. We also made experiments on gestures from different views. We compared the performance between BiLSTM, and GRU with 60, and 120 hidden units where GRU performs better than BiLSTM in our experiments. However, there are other hyperparameters which may also be tuned. This could be done with dedicated libraries such as *SigOpt*

[21]. In addition, Affine transformations can be used to preprocess the raw gestures before feeding into the deep network. Also, more training from other similar datasets might improve the system. Ensembling of classifiers may help as well. In the future, we shall extend our work considering these points and will develop better systems.

References

1. Cheng, Q., Mayberry, R.I.: Acquiring a first language in adolescence: the case of basic word order in American sign language. J. Child Lang. **46**(2), 214–240 (2019)
2. Cheok, M.J., Omar, Z., Jaward, M.H.: A review of hand gesture and sign language recognition techniques. Int. J. Mach. Learn. Cybern. **10**(1), 131–153 (2019)
3. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR abs/1406.1078 (2014). http://arxiv.org/abs/1406.1078
4. Cui, Z., Ke, R., Wang, Y.: Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. CoRR abs/1801.02143 (2018). http://arxiv.org/abs/1801.02143
5. Elsayed, N., Maida, A.S., Bayoumi, M.: Deep gated recurrent and convolutional network hybrid model for univariate time series classification. arXiv preprint arXiv:1812.07683 (2018)
6. Gangrade, J., Bharti, J.: Real time sign language recognition using depth sensor. Int. J. Comput. Vis. Robot. **9**(4), 329–339 (2019)
7. Ghotkar, A.S., Kharate, G.K.: Dynamic hand gesture recognition and novel sentence interpretation algorithm for Indian sign language using Microsoft kinect sensor. J. Pattern Recogn. Res. **1**, 24–38 (2015)
8. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space Odyssey. arXiv e-prints arXiv:1503.04069, March 2015
9. Haidong, S., Junsheng, C., Hongkai, J., Yu, Y., Zhantao, W.: Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing. Knowl.-Based Syst. **188**, 105022 (2020). https://doi.org/10.1016/j.knosys.2019.105022. http://www.sciencedirect.com/science/article/pii/S0950705119304289
10. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991 (2015). http://arxiv.org/abs/1508.01991
11. Kovács, G., Szekrényes, I.: Applying neural network techniques for topic change detection in the HuComTech corpus. In: Hunyadi, L., Szekrényes, I. (eds.) The Temporal Structure of Multimodal Communication. ISRL, vol. 164, pp. 147–162. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-22895-8_8
12. Kumar, P., Kaur, S.: Sign language generation system based on Indian sign language grammar. ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP) **19**(4), 1–26 (2020)
13. Kumar, P., Gauba, H., Roy, P.P., Dogra, D.P.: Coupled HMM-based multi-sensor data fusion for sign language recognition. Pattern Recogn. Lett. **86**, 1–8 (2017)
14. Kumar, P., Roy, P.P., Dogra, D.P.: Independent Bayesian classifier combination based sign language recognition using facial expression. Inf. Sci. **428**, 30–48 (2018)
15. Kumar, P., Saini, R., Roy, P.P., Dogra, D.P.: A position and rotation invariant framework for sign language recognition (SLR) using Kinect. Multimedia Tools Appl. **77**(7), 8823–8846 (2017). https://doi.org/10.1007/s11042-017-4776-9

16. Liwicki, M., Graves, A., Fernàndez, S., Bunke, H., Schmidhuber, J.: A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In: Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007 (2007)
17. Maaten, L.v.d., Hinton, G.: Visualizing data using T-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
18. Mehrotra, K., Godbole, A., Belhe, S.: Indian sign language recognition using Kinect sensor. In: Kamel, M., Campilho, A. (eds.) ICIAR 2015. LNCS, vol. 9164, pp. 528–535. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20801-5_59
19. Rabiner, L.R., Lee, C.H., Juang, B., Wilpon, J.: HMM clustering for connected word recognition. In: International Conference on Acoustics, Speech, and Signal Processing, pp. 405–408. IEEE (1989)
20. Saini, R., Kumar, P., Kaur, B., Roy, P.P., Dogra, D.P., Santosh, K.: Kinect sensor-based interaction monitoring system using the BLSTM neural network in healthcare. Int. J. Mach. Learn. Cybern. **10**(9), 2529–2540 (2019). https://doi.org/10.1007/s13042-018-0887-5
21. SigOpt: Sigopt hyperparameter optimization. https://sigopt.com/product. Accessed 03 July 2020
22. Tang, X., Chen, Y., Dai, Y., Xu, J., Peng, D.: A multi-scale convolutional attention based GRU network for text classification. In: 2019 Chinese Automation Congress (CAC), pp. 3009–3013. IEEE (2019)
23. Tolentino, L.K.S., Juan, R.O.S., Thio-ac, A.C., Pamahoy, M.A.B., Forteza, J.R.R., Garcia, X.J.O.: Static sign language recognition using deep learning. Int. J. Mach. Learn. Comput. **9**(6), 821–827 (2019)
24. Wario, R., Nyaga, C.: A survey of the constraints encountered in dynamic vision-based sign language hand gesture recognition. In: Antona, M., Stephanidis, C. (eds.) HCII 2019. LNCS, vol. 11573, pp. 373–382. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-23563-5_30
25. Wikipedia: Ok gesture. https://en.wikipedia.org/wiki/OK$_$gesture$#$cite$_$note-1. Accessed 04 July 2020
26. Zeshan, U., Vasishta, M.N., Sethna, M.: Implementation of Indian sign language in educational settings. Asia Pac. Disabil. Rehabil. J. **16**(1), 16–40 (2005)