Journal of Theoretical and Applied Information Technology
29th February 2024. Vol.102. No 4
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SIGN LANGUAGE RECOGNITION OF WORDS AND SENTENCE PREDICTION USING LSTM AND NLP

RASHMI GAIKWAD[1], LALITA ADMUTHE[2]

[1]Research Scholar, Shivaji University, Kolhapur, Maharashtra, India.

[2]DKTE Society's Textile & Engineering Institute, Ichalkaranji, Maharashtra, India.

E-mail: [1]rsgaikwad2020@gmail.com, [2]ladmuthe@gmail.com

ABSTRACT

Sign language enables the speech and hearing impaired people with a way of communication. These people understand sign language very well and can communicate with each other easily. Problem arises when they want to communicate with other people who do not understand sign language. To bridge this gap of communication a real-time sign language recognition system is proposed in this paper where meaningful sentences are generated from a few recognized signs of words. In the first part of this research, Long and Short Term Memory (LSTM) network is used for the recognition of signs of words. The network gives a maximum accuracy of 97.53%. In the second part the recognized words are fed as input to the natural language processing module for prediction of meaningful sentences. The system can detect the signs performed by different signers even if it is trained on dataset recorded on a single signer.

Keywords: *Sign Language Recognition (SLR), American Sign Language (ASL), MediaPipe Holistic, Long Short Term Memory (LSTM), Natural Language Processing (NLP).*

1. INTRODUCTION

Sign language recognition is a vast area of research. It is a way of communication of the speech and hearing impaired people. It is a means through which education can be imparted to these people. The deaf and dumb people represent 5% of the world population according to world health organization (WHO). In India, 63 million that is 6.3 % people are deaf and dumb approximately. So, measures should be taken more aggressively to empower such people with a much simpler way of communication with each other and also with other people who do not know sign language. American Sign Language is mostly taught in Indian schools. So the children in school are able to communicate with each other. But after their primary and secondary education, most of these children do not opt for higher education as they are not able to communicate with other people with confidence. This may be because other people do not know the sign language in detail. In this paper, focus is given on designing a system which will detect a few signs of words and predict a sentence from the detected words. The system proposed in this research will reduce the gap of communication between the deaf and dumb people and the normal people. The dataset is generated using OpenCV and MediaPipe Holistic. The recognition network used is the Long and Short Term Memory (LSTM) network. The natural language processing module proposed in this research takes the recognized words as input and generates meaningful sentences from it. The paper is organized as follows. After the introduction, second section of this paper gives a brief review of the work related to the proposed research. The details of the methodology implemented in this research are explained in the third section. The results and conclusions are discussed in the fourth and fifth sections respectively.

2. LITERATURE REVIEW

Over the years a lot of research has been carried out on sign language recognition. Before the evolution of artificial intelligence (AI) traditional methods like image processing as proposed in [1] were used for the implementation of sign language recognition systems. The use of stochastic linear formal grammar (SLFG) for generating meaningful sentences from the signs detected is proposed in [2]. This system is implemented for smart home interaction using dynamic sign language recognition. The accuracy of this system is 98.65% but it is not applied to sign language recognition. With the evolution of AI the convolution neural networks (CNN) became popular among researchers due to their advantages like self-learning, increased accuracy, faster detections etc. Faster recurrent combinational neural networks (RCNN) and MobileNet network is proposed in [3] for the detection of traffic sign recognition in real time. Detection of contours and centers of traffic signals by using the RGBN color space is implemented in this research. Using this method an accuracy of 84.5% and recall of 94% is achieved for automatic traffic sign detection. RCNN can also be used for sign language recognition (SLR). A continuous SLR network using bi-directional RNN is proposed in [4] for converting signs from videos into sentences as a sequence of labels. For feature extraction a deep convolutional neural network layer is used and for creating a sequence of labels an iterative optimization process is used in this method. A temporal convolutional network with fusion of 1-D and 2-D CNN is implemented in [5] for dynamic (signs involving movements) hand gesture recognition.

Recognition of signs of alphabets in American Sign Language (ASL) is proposed in [6] using CNN. This method is implemented using Tensorflow and openCV. The accuracy of recognition achieved using this method is 99.838%. Bhutanese sign language recognition of digits using CNN is implemented in [7] where the results obtained from CNN are compared with other models like K-Nearest Neighbor (KNN), Logistic regression (LR), Support Vector Machine (SVM), and LeNet5. The performance was assessed based on the parameters like precision, accuracy, recall and F1-score. The highest accuracy of 97.62% was achieved for CNN for the recognition of signs of digits from 1 to 9. A real-time sign language recognition and speech generation system from input signs is proposed in [8]. The ChaLearn249 dataset consisting signs in the form of images and videos is used here. Inflated 3D convolutional network i.e. I3D ConvNet is implemented in [9]. Extracting spatial-temporal information of signs in Chinese sign language is done in [10] using a 3D CNN. This method is employed only for the detection of static signs (signs not involving movement). The information about the gestures is in the part of an image which is called as region of interest i.e. ROI. Using this technique an accuracy of 94.3% is achieved.

Another deep learning CNN architecture called as ResNet50 is implemented in [11] for classification of finger spelled words. Data augmentation is used in this method and the dataset is the readily available ASL hand gesture dataset. An accuracy of 99.03% is achieved when the system is tested on 12,048 images. In [12] a dataset is generated called as GMUASL51 using skeletal data obtained from 12 users performing 51 signs for human activity recognition. [13] proposed the use of CNN and the Dynamic Bayesian Network (DBN) for SLR. The signs are collected using Microsoft Kinect sensor and the system is implemented for human computer interaction. The accuracy of recognition achieved is 99.40%.

A sensor based American SLR system is proposed in [14]. Surface electromyography sensors (sEMG) sensors are used to collect the signs. Four common classification algorithms are evaluated for 80 ASL signs. This experiment was performed on four signers. The accuracies were calculated for intra-subject and inter-subject cross session evaluation which came as 96.16% and 85.24% respectively. A deep learning CNN is proposed in [15] for ASL alphabet recognition. Data augmentation technique is implemented in this research to increase the training data. This research proves that deep learning CNN provide a better accuracy compared to other approaches. In [16] architecture is implemented for recognition of signs and prediction of sentences from the recognized signs using a common-sense context module. Arabic alphabets sign language recognition system is developed in [17]. The dataset consists of 54049 images of alphabets in Arabic sign language with 1500 images per class. The system is implemented using various pre-trained models and emphasis is given more on the EfficientNetB4 model. The training accuracy of this network is 98% and testing accuracy is 95%.

CNN in combination with attention-based encoder-decoder model is proposed in [18]. This system recognizes signs in Chinese Sign Language and translates it into voice output. The Word Error Recognition (WER) rate for this system is 10.8%. [19] proposes the use of Transformer Encoder for recognition of static Indian signs. The system is called as a vision transformer. The signs are divided into a series of positional embedding patches. These signs are fed as input to a transformer block which has four self-attention layers and a multilayer perceptron network. An accuracy of 99.29 % is obtained using this network. [20] implemented an SLR system called as SignBERT. The BERT transformer in combination with the ResNet model is used here to extract spatial features for continuous SLR. The results achieved in this research are compared with some other methods. It is observed that this system has better accuracy with lower WER on three sign language datasets.

One of the most recent developments in sign language recognition is long short-term memory (LSTM) in combination with Convolutional neural network as proposed in [21]. The data of signs is collected using OpenCV and computer vision. An accuracy of 95.5% is achieved using this method. Deep learning using CNN and LSTM is implemented in [22]. Here, the temporal structure information is obtained using LSTM for encoding and decoding of input frames of variable length. An accuracy of 99% is achieved in this research. [23] proposed the combination of CNN, LSTM and DNN for analyzing their performance on large vocabulary tasks like English-spoken utterances SLR system for the detection of signs of alphabets in ASL using the combination of CNN and LSTM network is implemented in [24]. MediaPipe framework is used to capture the signs and extract features. An accuracy of 99% has been achieved here. [25] proposes a POS tagging Twitter data concept for the Malayalam language using a combination of bidirectional LSTM and gate recurrent units (GRU) based deep learning model. Training of the network is done on the tagged tweets. The f1-measure at word level is obtained as 0.9254 and at character level it is obtained as 0.8739.

Deep learning architectures RNN, LSTM and Gated Recurrent Unit (GRU) are used in [26] for the formation of sentence by combining words. The system is implemented for Malayalam and Tamil nouns and verbs. An accuracy of 99% is attained using this method. [27] propose a Natural Language Processing system for analyzing the similarity between two sentences or texts. The RNN-LSTM network with word embedding features is used for the implementation of this purpose. The dataset for this research are the sentences gathered from Telugu newspapers.

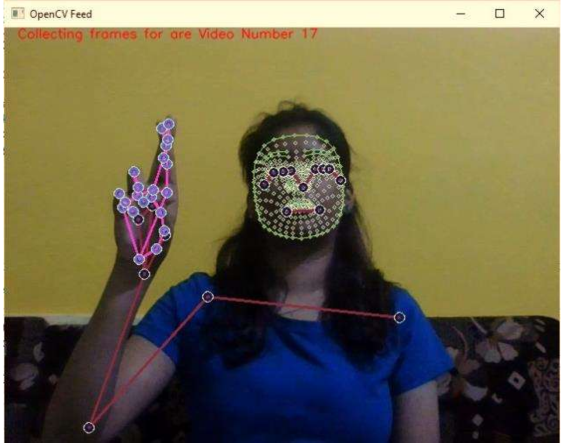
It is observed from the literature review that the recent advances in the SLR implementation is using OpenCV to access web camera for capturing signs performed by a signer. A lot of work is concentrating on the recognition of signs of alphabets, numerical digits, traffic signs and human postures. Very little work is done on recognition of signs of words. Many new modules based on deep learning neural networks have evolved which can be combined with some other networks to bring flexibility and accuracy. Real-time sign language recognition systems are more beneficial for communication using sign language so this should be focused more. LSTM networks are gaining popularity due to its sequence generation advantage. Sentence generation or sentence prediction from the recognized signs of words is an area which is not yet explored much. Based on these findings a solution for sign language recognition of words using LSTM network is proposed in the following part of this paper.

3. METHODOLOGY

This research is implemented for real-time American Sign Language recognition of words and generation of meaningful sentences from the recognized words. All the implementation in this research is carried out using Python version 3.7.3. Jupyter notebook IDE is used for python programing of all the modules.

3.1 Data Acquisition

The dataset for this research was generated using OpenCV and MediaPipe Holistic. The web camera on the computer was accessed using OpenCV. MediaPipe Holistic [28] is an open source framework for capturing real-time videos and images. It creates a complete human body landmarker by combining the landmarks from the pose, face and hands. It outputs a total of 543 landmarks divided as 33 pose landmarks, 468 face landmarks, and 21 hand landmarks per hand in real-time. The OpenCV feed for capturing signs using MediaPipe Holistic is shown in fig. 1.



Collecting frames for are Video Number 17

Figure 1. OpenCV feed for capturing signs

3.2 LSTM Network

LSTM network is specially intended for models with spatial inputs, like images or videos and the applications of generating textual descriptions from sequences of images. They have feedback connections which help them remember the previous inputs. Using LSTM motion can be detected from hand gestures as it allows entire sequences of data instead of only single data points.

Since LSTM are predominantly better at processing sequences of data such as text and speech it is used in this research for recognition of signs of words in a sequence one after another.

The decision about information flow in LSTMs i.e. about how the information comes inside is stored and how it leaves the network is taken by a series of gates. There are three gates in a typical LSTM network. They are the forget gate, input gate and output gate. These gates are nothing but filters with each having their own neural network. One such architecture of an LSTM network is shown in fig. 2. The first gate is the forget gate which decides which bits of the cell state are useful and which are not given that both the previous hidden state and new input data. It uses sigmoid activation function where each element is in the interval [0, 1]. The forget gate gives an output close to 0 when a component of the input is considered irrelevant and close to 1 when relevant.

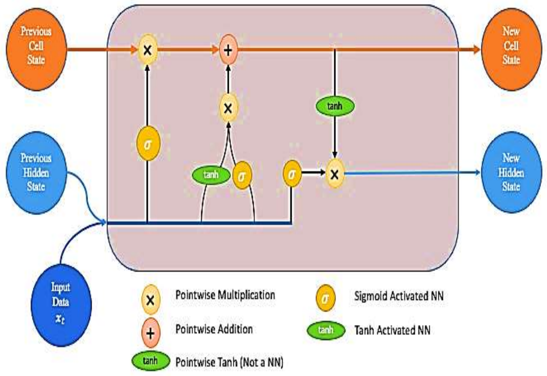


Figure 2. LSTM Architecture [29]

The second gate decides what new information should be added to the networks cell state given the previous hidden state and new input data. The new memory network is a tan h activated neural network which learns how to combine the previous hidden state and new input data to generate a new memory update vector. The third gate is the output gate which takes information on newly updated cell state, the previous hidden state and the new input data. This filter is applied to the newly updated cell state. This ensures that only necessary information is output and saved to the new hidden state.

Table 1 gives the details of the network implemented. It consists of the sequential model with three LSTM layers followed by three dense layers. The summary of the model shows the layer type, output shape and parameters. Here, ReLU activation function is used with Adam Optimization technique.

Table 1: Model Summary

Layer Type	Output shape	Parameters
lstm (LSTM)	(None, 30, 64)	442112
lstm_1 (LSTM)	(None, 30, 128)	98816
lstm_2 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 64)	4160
dense_1(Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 4)	142

Total parameters: 596,708
Trainable parameters: 596,708
Non-trainable parameters: 0

3.3 Natural Language Processing

Natural language processing (NLP) is the field of AI which deals with understanding the text and spoken words by a machine similar to that perceived by human beings. Some of the major NLP tasks are text and speech processing, text classification, language generation and interaction etc. The signs of words are detected by the LSTM network in real-time after training and testing of the network. Now, these recognized words will be fed as input to NLP module called as 'keytotext' which will predict a sentence from the input words.

3.3.1 The module Keytotext

The "keytotext" module [30] can predict sentences from input keywords. This module is built using the T5 model which is an NLP framework. T5 stands for Text-To-Text Transfer Transformer. All NLP problems can be converted into a text-to-text format using the T5 model. It is introduced by Google to achieve NLP tasks where the input and output are text strings. The data from WebNLG and DART (Open-Domain Structured Data Record to Text Generation) is used for training of the "keytotext". Variations from all domains are included in the training dataset.

3.3.2 T5 model

Fig. 3 shows the basic T5 model. There are many NLP modules introduced by Google and OpenAI to carry out different NLP tasks. The T5 transformer model [31] has encoder-decoder structure. It comprises of 12-pair blocks of encoder-decoder. Each block comprises of a self-attention layer, a feed-forward network, and an optional encoder-decoder attention layer. The dataset used for the training of the T5 model is the C4 dataset called as Colossal Clean Crawled Corpus which is collected from Common Crawl, a publicly available web archive. It consists of 750 GB clean English text scraped from the Internet. After extracting from Common Crawl, offensive words, filler sentences, code brackets, duplicates and sentences that don't end with a punctuation mark were removed from the dataset. Thus, it is a clean and huge dataset which means that the model can be trained on the dataset without ever repeating the same data.

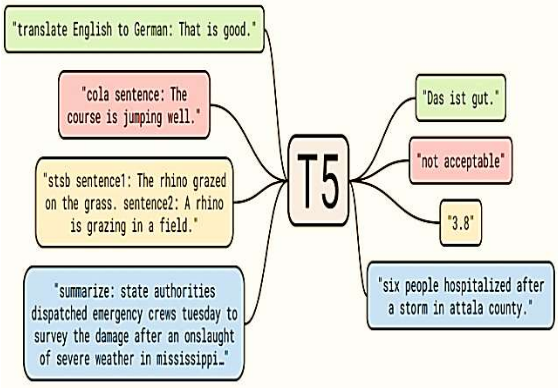

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 3. Basic T5 Model [32]

4. RESULTS

4.1 Dataset Generated

Overall, 30 videos for each of the 11 signs with 30 frames in length per video are captured. For each sign 1662 key point values are collected. Thus, the dataset consists of 11signs x 30 videos = 330 images. Table 2 shows the dataset generated for 11 signs.

Table 2: Dataset

Signs of Words	Videos per sign	Keypoint values
India	30	1662
New Delhi	30	1662
UK	30	1662
capital	30	1662
London	30	1662
USA	30	1662
New York	30	1662
I	30	1662
stay	30	1662
Goa	30	1662
football	30	1662

4.2 Training the Network

The network is trained for 1000 epochs. The accuracy of the network during training reaches 100% after 400 epochs as shown in fig. 4 and the total loss of the network reduces to 0 as shown in fig. 5.

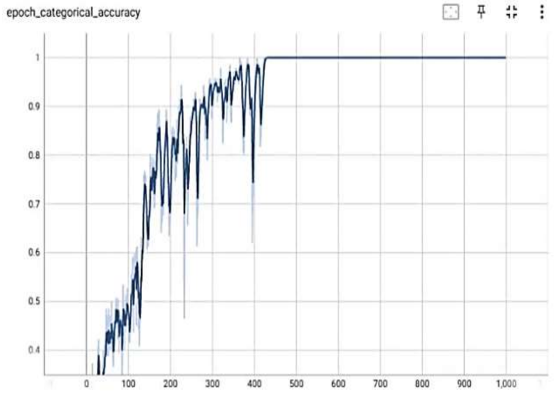


Figure 4. Accuracy vs. Epoch

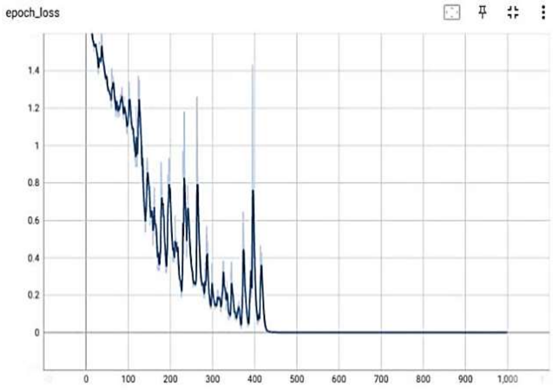


Figure 5. Loss vs. Epoch

4.3 Testing the Network

10% of the total dataset is used for the testing of the network. The accuracy precision and recall for every sign are calculated using equation (1) (2) and (3) respectively.

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FP+FN} \qquad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (3)$$

Here, TP and FP denote the numbers of true positive and false positive values and TN and FN show the number of true negative and false negative values. Table 3 shows the testing accuracy, precision and recall values of the signs of words detected in percentage. The highest accuracy is 97.53% for the word “New York”.

Table 3: Output after Testing the Network

Signs of words	Accuracy	Precision	Recall
India	93.82%	93.33%	100%
New Delhi	93.82%	93.5%	100%
UK	96.29%	94.87%	100%
capital	92.59%	96%	100%
London	93.82%	93.5%	100%
USA	93.82%	93.33%	100%
New York	97.53%	94.93%	100%
I	92.59%	96%	100%
stay	96.29%	94.87%	100%
Goa	93.82%	93.33%	100%
football	96.29%	94.87%	100%

4.4 Output of LSTM Network

After training and testing of the LSTM network the signs performed by the signer are detected in real-time as shown in fig. 6. The network can also detect the signs performed by different signers as shown in fig.7. The words detected on the output screen appear one after another. These words are given as input to the NLP module for prediction of sentences.

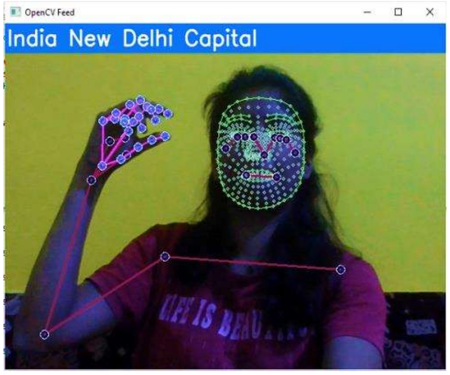


Figure 6. OpenCV feed of signer 1

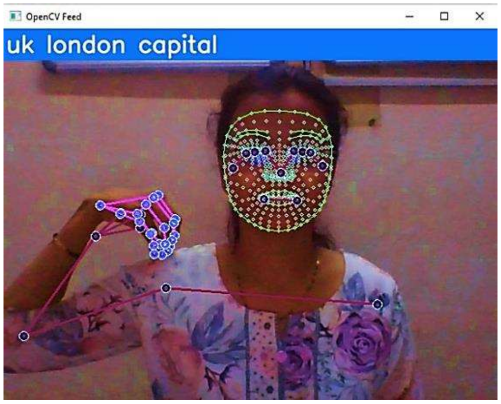


Figure 7. OpenCV feed of signer 2

4.5 Output of NLP module

The first sequence of words i.e. “India”, “New Delhi”, “Capital” is given as input to the NLP module. The output of this module is the predicted sentence “India’s New Delhi is the capital of India.” Similarly, another set of words is recognized as “UK”, “London”, “Capital”. The output is the second predicted sentence given as “London is the capital of the UK”. In this way some more words are recognized and fed to the NLP module to predict meaningful sentences given in table 4.

Table 4: Output of NLP Module

Input Keywords	Output Sentences Predicted	Expected Output Sentence
“India”, “New Delhi”, “Capital”	India’s New Delhi is the capital of India.	New Delhi is the capital of India
“UK”, “London”, “Capital”	London is the capital of UK	London is the capital of UK

'I', 'stay', 'Goa'	I stayed in Goa.	I stayed in Goa.
'USA', 'capital', 'New York'	New York is the capital of USA.	New York is the capital of USA.
'I', 'football'	I am a football player.	I play football
'I', 'stay', 'Goa'	I stayed in Goa.	

5. CONCLUSION

A real-time American Sign Language recognition system is implemented in this paper using LSTM network. The dataset is generated using OpenCV and MediaPipe Holistic. The network is trained for 1000 epochs and gives highest accuracy of 97.53%. The system can detect the signs performed by different signers even if it is trained on dataset recorded on a single signer. Faster detections and less dataset requirement are the advantages of using LSTM network over other networks. The recognized signs of words are fed as input to the NLP module to generate meaningful sentences from the input words. In this way the normal people will be able to communicate with the speech and hearing impaired people by performing minimum signs. This research can be expanded to inclusion of more signs of words in the dataset. Also, the proposed system can be used for the implementation of recognition of other sign languages.

REFERENCES:

[1] P.S. Rajam and G. Balakrishnan, "Real-time Indian sign language recognition system to aid deaf-dumb people," 13th International Conference on Communication Technology 978-1-61284-307-0/11/ ©2011 IEEE, pp 737-742, 2011.

[2] M. R. Abid, E. M. Petriu and E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 3, pp. 596–605, 2015.

[3] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," IEEE Transactions on Intelligent Transport System, vol. 20, no. 3, pp. 975-984, March 2019.

[4] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," IEEE Trans. Multimedia, vol. 21, no. 7, pp. 1880–1891, 2019.

[5] J. Liu, W. Yan and Y. Dong, "Dynamic hand gesture recognition based on signals from specialized data glove and deep learning algorithms," IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021.

[6] A. Kasapbasi, A. Eltayeb, A. Elbushra, O. Al-Hardanee and A. Yilmaz, "DeepASLR: A CNN based human computer interface for American sign language recognition for hearing impaired individuals," Elsevier, ISSN: 2666-9900, 2022.

[7] K. Wangchuk, P. Riyamongkol and R. Waranusast, "Real-time Bhutanese sign language digits recognition system using convolutional neural network," The Korean Institute of Communications and Information Sciences(KCIS), Elsevier B.V, ISSN:2405-9595, 2020.

[8] A. Thakur, P. Budhathoki, S. Upreti, S. Shrestha and S. Shakya, "Real-time sign language recognition and speech generation," Journal of Innovative Image Processing Vol.02/ no. 02, pp. 65-76, ISSN: 2582-4252, 2020.

[9] N. Sarhan and S. Frintrop, "Transfer learning for videos: From action recognition to sign language recognition," IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, pp. 1811-1815, 2020.

[10] J. Huang, W. Zhou, H. Li and W. Li, "Attention-based 3D-CNNs for large-vocabulary sign language recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2822-2832, Sept. 2019.

[11] P. Rathi, R. Gupta, S. Agarwal, A. Shukla and R. Tiwari, "Sign language recognition using ResNet50 deep neural network architecture," 5th International Conference on Next Generation Computing Technologies, February 2020.

[12] A. A. Hosain, P. S. Santhalingam, P. Pathak, J. Košecká, and H. Rangwala, ''Sign language recognition analysis using multimodal data,'' in Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA), pp. 203–210, Oct. 2019.

[13] Q. Xiao, Y. Zhao, and W. Huan, ''Multi-sensor data fusion for sign language recognition based on dynamic Bayesian network and convolutional neural network,'' Multimedia

Tools Appl., vol. 78, no. 11, pp. 15335–15352, Jun. 2019.

[14] J. Wu, L. Sun and R. Jafari, “A wearable system for recognizing american sign language in real-time using IMU and Surface EMG sensors,” IEEE Biomed Health Inform, pp-1281-1290, 2016.

[15] A. Mannan, A. Abbasi, A. R. Javed, A. Ahsan, T. R. Gadekallu and Q. Xin, “Hyper tuned deep convolutional neural network for sign language recognition,” Computational Intelligence and Neuroscience, Hindawi, Vol. 2022.

[16] I. Infantino, R. Rizzo and S. Gagli, “A framework for sign language sentence recognition by common-sense context,” IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 37, no. 5, September 2007.

[17] M. Zakariah, Y. A. Alotaibi, D. Koundal, Y. Guo and M. M. Elahi, “Sign language recognition for arabic alphabets using transfer learning technique,” Computational Intelligence and Neuroscience, Hindawi, Volume 2022.

[18] Z. Wang, T. Zhao, J. Ma, H. Chen, K. Liu, H. Shao, Q. Wang and J. Ren, “Hear sign language: a real-time end-to-end sign language recognition system,” IEEE Transactions on Mobile Computing, 2020.

[19] D. R. Kothadiya, SC. M. Bhatt, S. Tanzila, A. Rehman and S. A. Bahaj, “SIGNFORMER: Deep vision transformer for sign language recognition,” IEEE Access, pp. 1-1. 10.1109/ACCESS.2022.3231130, 2022.

[20] Z. Zhou, W. L. T. Vincent and Y. L. Edmund, “SignBERT: A BERT-based deep learning framework for continuous sign language recognition,” IEEE Access, Volume 9,. DOI-10.1109/ACCESS.2021.3132668, (2021).

[21] W. Li, H. Pu. and R. Wang, “Sign language recognition based on computer vision,” IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp- 919-922, 2021.

[22] S. He, “Research of a sign language translation system based on deep learning,” International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), Dublin, Ireland, pp. 392-396, 2019.

[23] T. N. Sainath, O. Vinyals, A. Senior and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, pp. 4580-4584, 2015.

[24] B. Sundar, T. Bagyammal, “American sign language recognition for alphabets using MediaPipe and LSTM,” Procedia Computer Science,Volume 215, pp. 642-651, 2022.

[25] S. Kumar, M. Anand Kumar and K. P. Soman, "Deep learning based part-of-speech tagging for Malayalam twitter data," Journal of Intelligent Systems, vol. 28, no. 3, pp. 423-435, 2019.

[26] B. Premjith, K. P. Soman and M. Anand Kumar, “A deep learning approach for Malayalam morphological analysis at character level,” Procedia computer science 132 pp. 47-54, 2019.

[27] A. D. Reddy, M. Anand Kumar, and K. P Soman, “LSTM based paraphrase identification using combined word embedding features,” Soft computing and signal processing, Springer, Singapore, pp. 385-394. 2019.

[28] MediaPipeHolistic, url: https://github.com/google/mediapipe/blob/master/docs/solutions/ho listic.md, last accessed 2023/11/15.

[29] Rian Dolphin, LSTM Networks, Towards Data Science url: https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9, last accessed 2023/11/15.

[30] Gagan Bhatia, “keytotext”, url: https://github.com/gagan3012/keytotext, Github, last accessed 2023/11/15.

[31] Hugging Face, url: https://huggingface.co/docs/transformers/model_doc/t5, last accessed 2023/11/15.

[32] Prakhar Mishra, Understanding T5 Model : Text to Text Transfer Transformer Model, Towards Data Science url: https://towardsdatascience.com/understanding-t5-model-text-to-text-transfer-transformer-model-69ce4c165023, last accessed 2023/11/15.