A CNN-LSTM Model for Arabic Sign Language Recognition

Basel Dabwan[1(✉)] and Mukti Jadhav[2]

[1] Department of Computer Science, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India
baseldbwan@yahoo.com
[2] Shri Shivaji Science and Art College, Chikhali District, Buldhana, India

Abstract. Gesture-based communication is a correspondence of nonverbal sort that involves the utilization of additional body parts. Demeanours of the face alongside Hand, eye, and lip movement is used for passing on data in correspondence with sign language. Individuals with hearing impairment or discourse are significantly dependent on gesture-based communication as a kind of association in their day-to-day existence. Few of the studies dealt with Arabic sign language, so in this work, we developed applied research with its video-based Arabic sign language recognition system that helps deaf and dumb people in the Arabic community. We developed our sign language model by a combination of Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNN), with two distinct neural network architectures. The first architecture is ConvLSTM and the second one LRCN. We used these two algorithms for extracting spatial and temporal features. The first model achieved a training accuracy of 99.66% and validation accuracy of 95%, and the second model achieved 99.5% training accuracy and 93.33% validation accuracy. We tested the performance of these two models in recognition between 28 classes of Arabic sign language.

Keywords: CNN · LSTM · Sign Language · Deep Learning · Machine Learning

1 Introduction

Individual existence without contact is exceptionally difficult to stay. Various conducts are utilized to impart and share their thoughts among sender and recipient. Discourse and sign are the most normal approaches to convey. Contact in the audible method is called discourse and is perceived through hearing. Then again correspondence utilizing body movement parts like hands and expressions of facial is called Gesture. Communication through signing is the language of Gesture that is gotten and perceived via the force of vision. Ordinary individuals have the alternative to utilizing gesture-based communication yet hard-of-hearing individuals utilize communication through signing as the essential language. There are "7099" communicated in dialects on the planet and "142" sign dialects utilized by handicapped individuals [1]. Table 1 show research on different gesture-based communication translation.

Communication via gestures is not a global language. It is unique from one country to another. Signs of the same letter can be performed distinctively in different sign

R. Manza et al. (Eds.): ACVAIT 2022, AISR 176, pp. 459–470, 2023.
https://doi.org/10.2991/978-94-6463-196-8_35

Table 1. The countries of different sign languages

Sign Language	Country
British Sign Language	United Kingdom (Elliott,2000)
Spanish Sign Language	Spain (. San-Segundo et al., 2008)
American Sign Language	United State of America (Vijayalakshmi and Aarthi, 2016)
Mexican Sign Language	Mexico (Caballero-Morales and Trujillo-Romero, 2012)
Arabic Sign Language	Arab Middle East (Halawani et al., 2013)
Greek Sign Language	Greece (Karpouzis etal., 2007)
Indian Sign Language	India (Vij and Kumar, 2016)

languages. For example letter, 'A' it's present in American sign language with one hand while Hindi sign language used both hands to present the same letter. Gesture-based communication is a significant tool to overcome any barrier between individuals who do not hear and the people who are listening. Gesture-based communication isn't just utilized by hearing the disabled individual, be that as it may, it is additionally utilized by the parent(s) of a hard of hearing youngster, offspring of the hard of hearing individual, instructor of the hard of hearing understudy thus numerous another space of correspondence with hard of hearing [9]. Communication through signing is a cooperative exploration region that incorporates PC vision, normal language handling, design coordinating, and phonetics. Its goal is to foster different logarithms and procedures to recognize the signs and recover the significance. In sign language recognition systems there are two main approaches: (a) sensor-based and (b) image-based. The main benefit of picture-based systems the client doesn't have to wear any devices, but this approach needs many computations in the pre-processing of the images, and also needs some set of constraints such as backdrop color, bright, nearby environment, and skin color [10]. There are many methods and techniques used to detect sign language. Machine learning has been used and it has given good results, such as the Support Vector Machine (SVM) and Key Nearest neighbor (KNN) algorithm, and then the deep learning method, which has very excellent results as it is characterized by many layers for feature extraction. Particularly when the data set size is very large, examples of deep learning algorithms are Convolution Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM). In this paper, we developed a model by the more powerful algorithms in deep learning to recognize the Arabic Sign Language (ASL). The results will be analyzed in the coming sections. The figure below shows the Arabic Sign Language alphabet.

2 Deep Learning

Lately, essential AI approaches have been for the most part replaced with more significant models that use many layers and pass information in vector design between layers, ceaselessly refining the appraisal until a correct affirmation is cultivated. Such calculations are typically depicted as "deep learning" frameworks or deep neural networks,

Fig. 1. Arabic Sign Language alphabet

furthermore, they work on norms like Machine Learning frameworks portrayed above, even though with undeniably more conspicuous complexity. in light of the development of the organization, two calculations are for the most part used for different endeavors: Convolutional Neural Networks (CNNs) that fuse somewhere at least around one layer of convolutional, and Recurrent Neural Networks (RNNs) that incorporate something one at least like intermittent layer. Contingent upon the number, what's more, sort of layers, these logarithms can show different properties and are generally sensible for different sorts of tasks, while the preparation stage impacts the effectiveness of the calculation The overall standard is that bigger and more explicit datasets consider all the more impressive organization preparing, and thusly, the idea of the preparation set is a huge impact element. Extra tweaking of a model can commonly be cultivated by changing a part of the important hyper-boundaries that describe the training method [11].

2.1 Convolution Neural Network

CNN designs for classification and properties extricated in the CNN models, the main arrangement of layers incorporates low-level features which incorporate a large portion of the fundamental data about edges in the first layers. Also, the second one is more profound than the initial one, etc. A fully connected layer neuron is additional to the convolutional layers to collect the extricated features from the layers of convolutional. Different fully connected layer properties for best discrimination results. After withdrawals, all properties for each picture by CNN profound layers Classification stage is acknowledged with dense/fully connected layers after that activation functions. Finally, in the final phase of the model, the SoftMax function is working to categorize each class. It deals with multiple variety labeling and is a generalization of carrying regression so far because it will be applied to continuous data (rather than a paired category) which may reflect various choice boundaries.

2.2 Long Short Term Memory

Sepp Hochreiter and Juergen Schmidhuber created LSTM in 1997 to address the vanishing gradient issue. LSTM, which was subsequently coordinated and advocated with the commitment of many individuals, is presently generally utilized. LSTM is utilized to keep up with the mistaken esteem from various times and layers in the backdrop. By giving steadier mistake esteem, it permits the learning steps of repetitive organizations to proceed. It does this by opening another channel between causes also impact.

3 Related Work

Few research and studies dealt with the Arabic sign language, we have gathered some studies to show the methods and technology used in this area:

In (Aly et al., 2020) [12] proposed a system to distinguish Arabic gestures using three different architectures for deep learning where the system was trained using an adaptive instantaneous scaling algorithm to classify all gestures. This architecture was created using a mix of a semantic segmentation network, a convolutional SOM, and a two-way deep directional LSTMNetwork network. The hand partitions were done utilizing DeepLab3 +, and as a result of utilizing this approach, the effectiveness was expanded by 70%, and the normal precision was 89.5%.

In (El-Bendary et al., 2011) [13] develop an automated translation model for indications of the manual letter set in the ArSL. The present ArSL Letter sets Interpreter (ArSLAT) structure doesn't base on using visual markings or gloves to achieve the recognition work. As a decision, it oversaw the image of hands that empowered the client to interact with the model characteristically. The present ArSLAT architecture included five fundamental stages; pre-processing, best-frame distinguishing proof, kind identification, properties extraction, and a classification phase. The used extricated properties were the interpretation, transformation invariant, and scale to make the model dynamically adaptable. The proposed ArSLAT model was shown to have the decision to see the Arabic letters generally along with 91.3 percent and 83.7 percent accuracy using MLP classifiers and Minimum Distance Classifier (MDC), autonomously.

In (Omar Al-Jarrah and Alaa Halawani, 2001) [14] built an Alphabet distinguishes system in ArSL. The study was accomplished via teaching the group of ANFIS systems, Each of these ideas was offered for identifying a current sign. In the absence of the need for gloves, the image of the gesture was acquired via a cam connected to the computer. Since pre-processing release from the properties plan is dependent on the calculation of 30 vectors under the gestural foci the region near the helpful gesture segment's edge As a result, the vectors were preserved within the ANFIS framework. To place it in the appropriate category (gesture). The presented model was solid on the changes of the place of the sign, Size, also as a result of the image's orientation. it had been that the extricated properties were positive as translation, invariant movement, and scaling. The results of the simulation showed that their framework with about nine rules for each ANFIS system has had the chance to be recognized Accuracy of 93%.

(Dabwan Basel & Jadhav Mukti, 2021) [15] developed an automated recognition system for Yemeni Alphabets sign language; they used CNN layers to extract the features, A fully connected layer neuron is added to the convolutional layers to gather the extracted properties from the convolutional layers, After extractions, all features for each image by CNN deep layers Classification stage is realized with dense/fully connected layers followed by functions of activation. Lastly, the regression SoftMax function is utilized to category each alphabetical from 32 classes. The system precision achieved 93%.

4 Proposed Model

We used a CNN to obtain spatial properties at a particular time step in the sequence of the input (video) after that used LSTM to discover temporal relationships between frames in our model. Figure 1 shows the proposed model architecture (Fig. 2).

We have taken the Arabic Sign Language (ArSL), and we have used KArSL: Arabic Sign Language Database (Sidig et al., 2021), a dataset which we selected consists of 28 classes of dynamic sign Alphabets. Dataset was performed by three professional signers; the dataset was captured with a state-of-the-art multi-modal Microsoft Kinect V2 device. There are 24 videos (MP4) for each class, As a result, this dataset has a total of $24 \times 28 = 672$ videos of sign characters with Different variations. The dataset was recorded at 30 frames per second with a resolution of 1920×1080 pixels. We did all pre-processing and normalization operations, e.g., resizing the frame to a fixed size (64, 64), reducing the computations, data normalization to rang [0–1] by dividing the values of pixels by 255, also shuffle the dataset and separated it into testing training, training set = 75%, test set = 25%. The two architectures of the model that were utilized for ConvLSTM and LRCN are two algorithms that combine CNN and LSTM, as shown below.

5 ConvLSTM

In this architecture, we have developed the initial approach utilizing ConvLSTM cells in combination, ConvLSTM cell are a type of LSTM network that includes convolutions activities in the neural network. it is an LSTM with convolution implanted in the structure, This makes it appropriate for distinguishing spatial properties of the data while keeping the temporal relationship in mind. Because of this convolution architecture, this approach

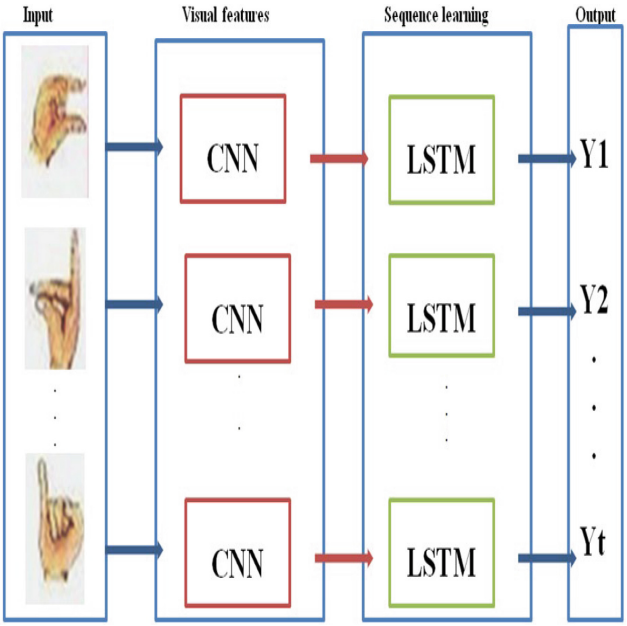

Fig. 2. Model architecture.

is equipped for taking in three-dimensional data (height, width, no of channels) though a basic LSTM just used one-dimensional input subsequently, LSTM is inconsistent for displaying Spatio-transient information all alone.

We developed the model by utilizing ConvLSTM2D Keras, layers with recurrent. The ConvLSTM2D layer additionally takes kernel size and no of filters demand executing the convolutional activities. The result of the layers is flattened in the ending and made it input to the softmax actuation in the Dense layer this gives the probability of each sign category. Also used layers of MaxPooling3D with (2,2) filter size to reduction in feature, decrease the frame's dimensions and avert not important calculations and Dropout layers to avoid the model overfitting. Also to avoid vanishing gradient we used softmax with small batch size for training the model trained with the flowing parameters (x = features_train, y = labels_train, epochs = 20, batch_size = 4,shuffle = True, validation_split = 0.2, callbacks = [early_stopping_callback]). The Fig. 3 shows the ConvLSTM model construct.

5.1 LRCN

In this architecture, we have implemented the LRCN method by joining LSTM and Convolution layers in one model. Another comparable situation methodology utilizes the LSTM model and CNN model trained independently. The CNN model can be utilized to get spatial properties from video frames, and for this reason, it utilized a pre-trained model, that can be fine-tuned for the issue. And the CNN-extracted features can then be used by the LSTM model., to recognize the signing class being acted in the input.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv_lst_m2d_8 (ConvLSTM2D)  (None, 20, 62, 62, 4)     1024
_________________________________________________________________
max_pooling3d_8 (MaxPooling3 (None, 20, 31, 31, 4)     0
_________________________________________________________________
time_distributed_6 (TimeDist (None, 20, 31, 31, 4)     0
_________________________________________________________________
conv_lst_m2d_9 (ConvLSTM2D)  (None, 20, 29, 29, 8)     3488
_________________________________________________________________
max_pooling3d_9 (MaxPooling3 (None, 20, 15, 15, 8)     0
_________________________________________________________________
time_distributed_7 (TimeDist (None, 20, 15, 15, 8)     0
_________________________________________________________________
conv_lst_m2d_10 (ConvLSTM2D) (None, 20, 13, 13, 14)    11144
_________________________________________________________________
max_pooling3d_10 (MaxPooling (None, 20, 7, 7, 14)      0
_________________________________________________________________
time_distributed_8 (TimeDist (None, 20, 7, 7, 14)      0
_________________________________________________________________
conv_lst_m2d_11 (ConvLSTM2D) (None, 20, 5, 5, 16)      17344
_________________________________________________________________
max_pooling3d_11 (MaxPooling (None, 20, 3, 3, 16)      0
_________________________________________________________________
flatten_2 (Flatten)          (None, 2880)              0
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 11524
=================================================================
Total params: 44,524
Trainable params: 44,524
Non-trainable params: 0
_________________________________________________________________
Model Created Successfully!
```

Fig. 3. ConvLSTM model construct

But here, we have implemented another methodology called the Long-term Recurrent Convolutional Network (LRCN), which joins CNN and LSTM layers in one model. The Convolutional layers are utilized for spatial properties extrication from the video frames, then, the extricated spatial properties are directly input to LSTM layer(s) at every time-step for temporal sequence modeling. In This approach the network learns spatiotemporal properties straightforwardly in end-to-end training, resulting in a vigorous model. We were also utilizing the TimeDistributed wrapper layer, which permits applying the same layer to each video frame independently. So it prepares a layer (around which it is wrapped) equipped with taking input of form(no of frames, width, height, no of channels) if originally the layer's input form was (width, height, num_of_channels) which is exceptionally helpful as it permits to include the entire video into the model in a solitary shot. We implemented LRCN approach utilizing Conv2D time-distributed layers which are going to be followed by the MaxPooling2D and layers of Dropout. Flattened the properties extricated from the Conv2D layers using the Flatten layer and input it to an LSTM layer. Then, the softmax activation in the Dense layer will use the result from the LSTM layer to This gives the probability of each sign category. Figure 4 shows the LRCN model construct.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
time_distributed_22 (TimeDis (None, 20, 64, 64, 16)    448
_________________________________________________________________
time_distributed_23 (TimeDis (None, 20, 16, 16, 16)    0
_________________________________________________________________
time_distributed_24 (TimeDis (None, 20, 16, 16, 16)    0
_________________________________________________________________
time_distributed_25 (TimeDis (None, 20, 16, 16, 32)    4640
_________________________________________________________________
time_distributed_26 (TimeDis (None, 20, 4, 4, 32)      0
_________________________________________________________________
time_distributed_27 (TimeDis (None, 20, 4, 4, 32)      0
_________________________________________________________________
time_distributed_28 (TimeDis (None, 20, 4, 4, 64)      18496
_________________________________________________________________
time_distributed_29 (TimeDis (None, 20, 2, 2, 64)      0
_________________________________________________________________
time_distributed_30 (TimeDis (None, 20, 2, 2, 64)      0
_________________________________________________________________
time_distributed_31 (TimeDis (None, 20, 2, 2, 64)      36928
_________________________________________________________________
time_distributed_32 (TimeDis (None, 20, 1, 1, 64)      0
_________________________________________________________________
time_distributed_33 (TimeDis (None, 20, 64)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 32)                12416
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 132
=================================================================
Total params: 73,060
Trainable params: 73,060
Non-trainable params: 0
```

Fig. 4. LRCN model construct

6 Result and evaluation

We used a combination of CNN and LSTM logarithms to implement our models, we used two architectures: ConvLSTM and LRCN to classify the Arabic Sign language (ArSL), the dataset includes 28 classes of alphabetic signs video, 24 videos for each class, with Different variations and dimensions, Total of videos = 672 videos. The dataset was divided into 75% for the training and 25% for the testing, the first model ConvLSTM achieved a training accuracy of 99.66% and validation accuracy of 95%, and the second model LRCN achieved 99.5% training accuracy and 93.33% validation accuracy. Figure 5, 6, 7, and 8 show the accuracy and loss of the two models.

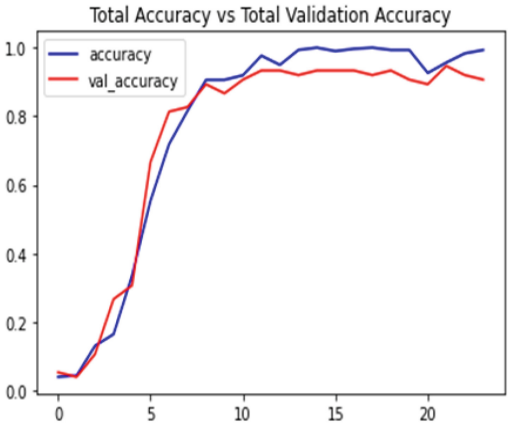

Fig. 5. ConvLSTM model accuracy

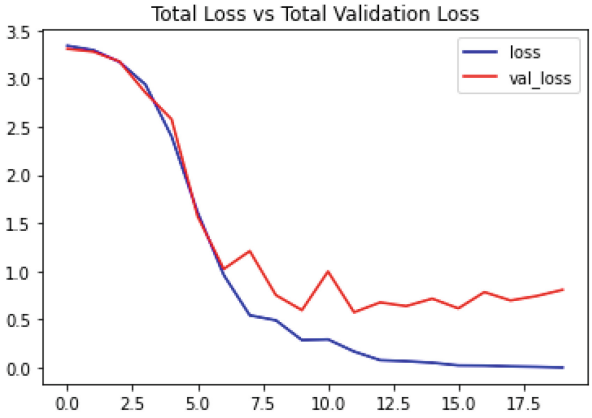

Fig. 6. ConvLSTM model loss

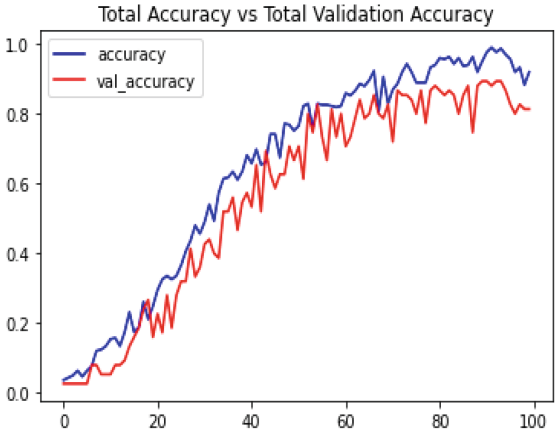

Fig. 7. LRCN model accuracy

In fact, to date, there are no researches that used this architecture to recognize the Arabic sign language, but we will compare it with studies that used the same algorithm in different sign languages, Table 2 displays The Comparing of our models with existing models.

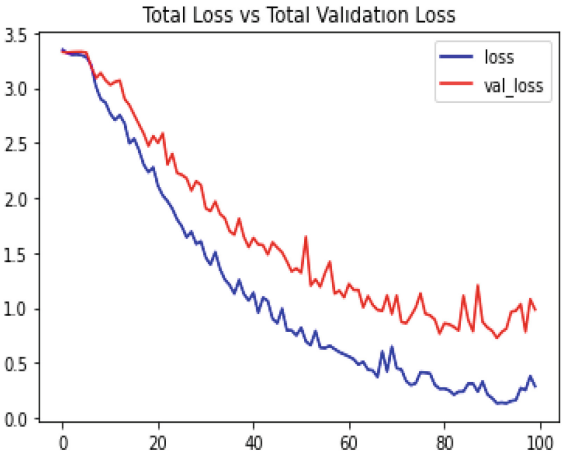

Fig. 8. LRCN model loss

Table 2. Comparing of proposed models with the existing models.

Method	Feature type and Classifier	Accuracy
baseline method (Shipman et al.,2015)	Hand-crafted features + SVM	69.23%
Baseline + RNN (Shipman et al.,2015)	Hand-crafted features + RNN	78.02%
CNN + SVM (Shipman et al.,2017)	2- stream CNN features + SVM	79.15%
CNN + RNN(Borg & Camilleri,2019)	Stream CNN features + RNN	87.67%
Our ConvLSTM model	CNN features + RNN	95%
Our LRCN model	CNN features + RNN	93.33%

7 Conclusion

In this research, the Arabic sign language (ArSL) recognition models were introduced by two architectures of CNN in combination with RNN, the first model is ConvLSTM. A ConvLSTM cell is a variant of an LSTM neural network that includes convolutions activities in the network. it is an LSTM with convolution implanted in the approach, which prepares it to fit for distinguishing spatial properties of the data while keeping into account the temporal relation. Because of this convolution architecture, the ConvLSTM is equipped for interacting in three-dimensional input (height, width, no of channels) though a basic LSTM just interacts in one-dimensional input subsequently LSTM is inconsistent for displaying Spatio-transient information all alone. This model achieved validation accuracy of 95%, and the second model is LRCN, we have implemented the LRCN model by joining Convolution and LSTM layers in one model, The layers

of Convolutional are utilized for spatial properties extrication from the video frames, and the extricated spatial properties directly input to LSTM layer(s) at every time-steps for temporal sequence modeling. In This approach the network teaches spatiotemporal properties straightforwardly in end-to-end training, resulting in a vigorous model. We were also utilizing the TimeDistributed wrapper layer, which permits applying the same layer to each video frame independently. So it makes a layer (around which it is wrapped) equipped with taking input of form(no of frames, width, height, no of channels) if originally the layer's input form was (width, height, n of channels) which is exceptionally helpful as it permits to include the entire video into the model in a solitary shot, This model achieved validation accuracy 93.33%. When were compared our architectures with previous research, we found a significant difference in performance. We used CNN algorithm to obtain spatial properties, and we benefited from the RNN algorithm to preserve the frame sequence in the video, and we combined them into one model. The two algorithms also work in the same layer.

References

1. Ehnolgue, 2018,"Sign Language." [Online]. Available:https://www.ethnologue.com/subgroups/signlanguage. [Accessed: 20- Jun-2018]
2. R. Elliott, J.R. Glauert, J.R. Kennaway, I. Marshall, The development of language processing support for the ViSiCAST project, in: Proceedings of the fourth international ACM conference on Assistive technologies - Assets '00, 2000 , pp. 101–108.
3. R. San-Segundo, et al.,Speech to sign language translation system for Spanish, Speech Commun., vol. 50, no. 11–12, 2008, pp. 1009–1020, 2008.
4. P. Vijayalakshmi, M. Aarthi, Sign language to speech conversion, in: Fifth International Conference on Recent Trends in Information Technology, 2016, pp. 1–6
5. S.O. Caballero-Morales, F. Trujillo-Romero, 3D Modeling of the Mexican Sign Language for a Speech-to-Sign Language System, Comput. y Sist., vol. 17, no. 4, 2012, pp. 593–608.
6. S.M. Halawani, D. Daman, S. Kari, A. R. Ahmad, An Avatar Based Translation System from Arabic Speech to Arabic Sign Language for Deaf People, Int. J. Comput. Sci. Netw. Secur., vol. 13, no. 12, 2013, pp. 43–52.
7. K. Karpouzis, G. Caridakis, S. Fotinea, E. Efthimiou, Educational resources and implementation of a Greek sign language synthesis architecture, Comput. Educ., vol. 49, no. 1, 2007, pp. 54–74,
8. P. Vij, P. Kumar, Mapping Hindi Text To Indian sign language with Extension Using Wordnet, in: Proceedings of the International Conference on Advances in Information Communication Technology & Computing, 2016, pp. 1–5.
9. T. Dasgupta, A. Basu, Prototype machine translation system from text-to-Indian sign language, in the 13th International Conference on Intelligent User Interfaces, no. January, 2008, pp. 313–316.
10. Kausar S, Javed MY, A survey on sign language recognition. In: 2011 frontiers of information technology.,(2011),pp95–98. https://doi.org/10.1109/FIT.2011.25
11. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, vol. 521, no. 7553, 2015, pp. 436–444, 2015
12. S. Aly, W.A. Aly, DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition, vol. 8,2020 pp. 83199–83212.
13. N. El-Bendary, H. Zawbaa, M. Daoud, A. Hassanien, K.. Nakamatsu, ArSLAT: Arabic sign language alphabets translator. Int J Comput Inf Syst Ind Manag Appl 3,2011, pp.498–506.

14. O. Al-Jarrah, A. Halawani, Recognition of gestures in Arabic sign language using neuro-fuzzy systems. Elsevier, Amsterdam, 2001, pp. 117–138
15. B. Dabwan, M. Jadhav, A Deep Learning based Recognition System for Yemeni Sign Language, 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI),2021, pp.1-5,doi: https://doi.org/10.1109/MTICTI53925.2021.9664779.
16. A. Sidig, H. Luqman, S Mahmoud, M. Mohandes, KArSL: Arabic Sign Language Database. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20, 1, Article 14 (2021,pp.19, https://doi.org/10.1145/3423420
17. F. Shipman, R. Gutierrez-Osuna, T. Shipman, C. Monteiro, V. Karappa, Towards a distributed digital library for sign language content, in Proc. .15th ACM/IEEE-CS Joint Conference on Digital Libraries, 2015, JCDL '15, pp. 187–190.
18. F. Shipman, S. Duggina, C. Monteiro, R. GutierrezOsuna, Speed-Accuracy Tradeoffs for Detecting Sign Language Content in Video Sharing Sites, in Proc. ACM SIGACCESS. 2017, ASSETS '17, pp. 185–189, ACM.
19. M. Borg, K. Camilleri, Sign Language Detection "in the Wild" with Recurrent Neural Networks. ICASSP 2019- 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1637–1641.