Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 189 (2021) 76–83

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on AI in Computational Linguistics

3D-CNN based Dynamic Gesture Recognition for Indian Sign Language Modeling

Dushyant Kumar Singh

CSED, MNNIT Allahabad, Prayagraj-211003, India

Abstract

Hand gestures used in Indian Sign Language (ISL) are static and dynamic in the time domain. The Indian Sign Language is available as a standard but is still not very common among peoples. In this paper, we have used a 3-dimensional convolutional based Convolution Neural Network to model the most utilized gestures of the Indian community. The trained model can provide a natural language output corresponding to the signs of the ISL. This in turn will help in reducing the problems faced while communicating with deaf and dumb peoples. Moreover, these dynamic gestures can be used in medical, industrial and various other fields. We took 20 gestures from standard Indian Sign Language (ISL) and trained our model on the dataset made by replicating the actions of those gestures. Ten subjects volunteered to make the dataset in distinct backgrounds, light conditions and orientations. Network model used produced good results in terms of accuracy, precision, recall and f1-scores.

Keywords: Indian sign language, hand gestures recognition, 3D-CNN, dynamic gesture, video classification

1. Introduction

Hand gestures [1, 2, 4] are the largely used traits of computer vision applications. These are getting used from a very long time for sign based communications. Involvement of computers and concept of digital image processing have boosted the activities in the hand gesture modelling and recognition approaches. Sign language is an extended application of hand gestures. In spite being introduced ages ago, standard sign languages are not very common among peoples in the society. However, from a personal to societal level, hand gestures have made their existence in every domain of life. Gesture controlled wheel chair is one examples of personal level application, where sign languages eradicate the communication gap among people that contributes to societal benefits that hand gestures provide. Partially paralyzed people can express their needs via hand gestures. People with autistic challenges face difficulties in

* Dushyant Kumar Singh. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: dushyant@mnnit.ac.in

1877-0509
10.1016/j.procs.2021.05.071

associating themselves within community. They uses or could use the hand gestures to communicate. Hand gestures not only emerge as the remedy of personal and societal needs, but also, as an element of luxury. Gesture controlled lights, home automation, 3-D gesture controlled video games like PS4 being perfect examples of the use of hand gestures as an element of luxury. Recently, a famous car manufacturing company BMW has also launched hand gesture control features in their 5^{th} series cars, which is going to start the trend of touchless screens. Medical streams use hand gesture controlled 3D-MRI image viewer that would enable the doctors to navigate it during the surgery.

Hand gestures can be modelled using two approaches : Physical sensor based gesture modelling [5,6, 7] and Computer vision based gesture modelling [5, 22]. Physical sensor based hand gesture modelling uses physical sensors such as flex sensors, accelerometers and gyroscope for accumulating the data. The devices used in conjunction with physical sensors are single board computers such as raspberry pi and Arduinos for processing the data. These devices are not easy to configure and calibrate. Also, devices being fragile needs intense handling care. Computer vision based technique, on the other hand uses camera and algorithm to model the hand gestures. It mitigates human vision system to accumulate and interpret the gestures. Proposals of various fast algorithms, introduction of multiple smaller parallel computational units such as GPUs and availability of large datasets such as Kaggle have paved the path for computer vision approach to be used efficiently for modelling and recognizing hand gestures. Now computer vision based applications used for processing the hand gestures can be run on personal computers too, thereby attracting more researches to be done in this field.

Various artificial and machine learning algorithms [2, 3, 11] accompanied with image processing techniques are used to classify the hand gestures. These techniques are majorly divided into static and dynamic paradigm. Static hand gesture modelling just requires processing of a single frame of a video at one time. Static hand gesture modelling do not contain the temporal information. Dynamic hand gesture modelling on the other side poses the temporal features. Orientation and shape of hand are accompanied with the movement in time.

There are many approaches that have been proposed for the classification of static and dynamic hand gestures. Among which, Convolutional Neural Network (CNN) has emerged as one of the promising techniques in the field of pattern recognition and image classification. CNN are analogous to the traditional artificial neural networks. CNNs allow us to use image-specific features embedded in the network, making itself suitable for video and digital image. It provides focused operations, along with the reduction in the trainable parameters as compared to ANNs. Image data needs large of computational resources if processed by ANNs. Operations such as pooling and convolution introduces generalization and reduced dimensionality in the network. Recent advances and introduction to different dense CNN architectures result in the increase in efficiency and accuracy of dynamic hand gesture recognition systems. However, dynamic hand gestures systems in the real-world struggle with various open challenges including, different lighting conditions, processing time, detection of hand segment, and many more.

Every part of the society is trying to introduce the concept of inclusion within themselves. Sign languages play a major role in this. Each country has a different sign language that contains gestures that majorly do not map with the gestures used in sign language of other countries. India also has a proper sign language dictionary that includes many useful and frequently used sign language gestures. In spite having a dictionary of sign languages for a long time, it is uncommon among people. Need is to make a robust system that can translate Indian sign language to a meaningful information that won't require presence of extra person as a translator.

In this paper we have tried to make a model that would recognize the meaning of dynamic hand gestures in Indian sign language from a real-time video stream. Building on the concepts of CNN classifiers used in hand gesture classification, we have employed the 3 Dimensional-Convolution in the network that would convolve over the sequence of image frames (3D-frames). 3-D Convolution Neural Network (3D-CNN) is able to process the time-variant frames in a sequence. We have also presented the dataset with 20 Indian sign language hand gestures.

2. Literature Survey

Many researchers have applied pre-curated features that can hold spatial and temporal information of hand gestures, which typically includes the shape, orientation, appearance, and direction of motion. Image gradient, motion divergence fields, and optical flow are the popular techniques to identify the moving object in the video clips [1, 8]. A state-based spotting algorithm is used to split continuous gestures. Features such as hand position, velocity, size, and shape are aligned for the set of frames, which are used to train the HMM model [9]. Huong [10] proposed

a model for modeling static hand gestures for Vietnamese sign language (VSL). The gaussian mixture-based background segmentation algorithm and binary thresholding with Otsu's binarization are used to remove the background. Then motion history images (MHI) and Feedforward Neural Networks (FNN) are used [11]. Dynamic hand gestures are majorly divided into three motion phases: preparation, stroke, and retraction, which help separate hand gestures from non-gesture movements [2]. Quek proposed the set of rules that distinguish gestures from non-gestures [12]. Skeletons have been extracted from the images after repetitive hole filling and pruning. The resultant skeleton image is compared with reference images in the train set, and Baddeley's distance is used as it is less sensitive to object translation and rotation than the Euclidean distance [13]. Geometric shapes including the point for the center of the palm, point for the wrist, four sets of points, including tip, articulation points and base of each finger. To fulfil this pupose, data is collected from Intel RealSense camera. A total of 22 joints are collected in the descriptor. Each descriptor is then encoded in the Fisher Vector representation. Final vector generated is fed in SVM to get the result [14]. Akmeliawati [15] proposed a model that classified thirteen New Zealand Sign Language hand gestures by using PCA for feature extraction and ANN to classify those features. Here, CAMShift was used to track the hand in the whole frame. Sawant [16] used Otsu's thresholding to segment the image from the whole video. PCA was used as a feature extraction technique.

In contrast to the methods using the pre-curated features, the current trend moves towards the deep learning paradigm where the architecture learns on the input and makes a self-curated feature map. Neverova proposed a parallel connection of single scale paths where each path independently learns from the video and classifies the video in its own temporal scale. Then all paths are combined using additive late fusion to avoid discrepancies due to pre-block errors [17]. Considering action recognition, Tran used deep 3-dimensional convolutional networks (3D ConvNet) trained on a huge dataset of hand gestures, which concluded that 3D ConvNet is more suitable with smaller convolutional kernels of shape 3 x 3 x 3, which outperformance four state-of-the-art methods on 4 benchmarks and are comparable with current Best Known Methods [18].

3. Proposed Approach

This section contains the workflow and network architecture of the proposed model for modeling dynamic hand gestures. The same is detailed in upcoming subsections.

3.1. Workflow of proposed model

The model takes video samples of gestures as input and provides a gesture label as output. It first extracts the frames from the video samples and saved them sequentially on the disk. The number of frames extracted from each video is recorded for the reference. Next, the sequence of contiguous images was fed to the 3D-CNN network [21]. Finally, the output of the network is converted into the respective class label. Fig. 1 demonstrates the work-flow of the proposed model.

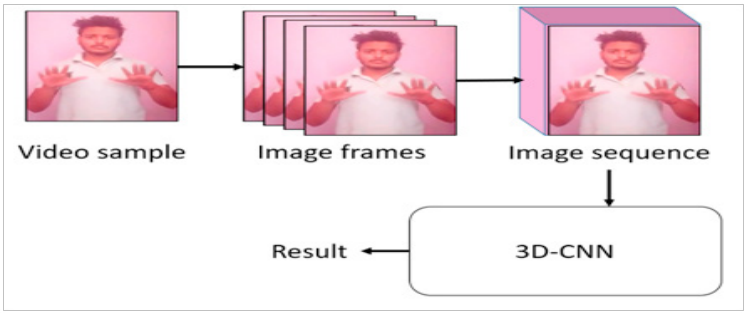

Fig. 1. Flow of image data in the system

3.2. Network Architecture

The proposed model is designed with a series of Convolution_3D operations for convoluting multiple frame block in a single process, followed by Maxpooling_3D. The result obtained is then flattened and fed to Multi-Layer Perceptron (MLP) having softmax layer applied at the end for getting probability for each class as the result that shows corresponding gesture representation. Table 1 represents the proposed model's architecture containing 156,544,980 trainable parameters, which need a good amount of computing and computational resources for the learning process. The network proposed contains several essential elements, which are described as follows:

Input blocks: Input images are appended one after another to make an image sequence. Images blocks made are in the form of $B^{h,w,c,s}$, where h and w are the height and width of the image, c is the channel and s is the sequence. We have set the values of image blocks in our cases as $B^{128,128,3,20}$.

Convolution: 3D convolution-based network is well-suited for holding and processing of Spatio-temporal features. The image block is convolved with the kernels of size (3 x 3 x 3). Tran mentioned the difference of 2D and 3Dconvolution in a simple pictorial form, and it was found that the selected kernel outperforms the previously used kernels [18].

Maxpooling: Maxpooling is applied after the convolution layer to reduce representation's spatial size, thereby reducing the computation needed. We have applied Max-pooling of (2x2x2) that helps in generalization. It effectively combines neighbor pixel values into a single pixel.

Dense: Multi-Layer Perceptron (MLP) neural network is used in the second half of the network, after the convolution layer, to process the reduced feature maps. Feature maps are flattened and fed to the MLP. Nodes in the MLP adjust their weights according to the input image sequence and the expected label.

Dropout & Regularization: Dropout of 30% is applied to reduce the overfitting condition. The decay of $1 \times \exp^{-6}$ is applied to the model. L1 regularization is also applied for better results.

Softmax: The last layer of the used model is Softmax to normalize the output vector to the prediction distribution for the 20 classes. The output of Softmax for ith element in the list is calculated by equation 3, where z is the output

Table 1. Structure of network

Layer (type)	Output Shape	Parameters
conv3d_1 (Conv3D)	(None, 18, 118, 118, 32)	2624
max_pooling3d_1 (MaxPooling3)	(None, 18, 59, 59, 32)	0
conv3d_2 (Conv3D)	(None, 16, 57, 57, 64)	55360
max_pooling3d_2 (MaxPooling3)	(None, 16, 28, 28, 64)	0
conv3d_3 (Conv3D)	(None, 14, 26, 26, 64)	110656
max_pooling3d_3 (MaxPooling3)	(None, 14, 13, 13, 64)	0
flatten_1 (Flatten)	(None, 151424)	0
dense_1 (Dense)	(None, 1024)	155059200
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 256)	262400
dense_4 (Softmax)	(None, 20)	5140

Total params: 156,544,980
Trainable params: 156,544,980
Non-trainable params: 0

vector from the neurons, K is the size of the vector (Here its value is 20).

$$Softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \qquad for \quad i = 1....K \tag{1}$$

4. Experiments, Results & Aanlysis

4.1. Experimental Setup

The platform used for computation uses an Intel core i7 processor accompanied by 8 GB of RAM. It also uses Nvidia GeForce GTX 1050-Ti GPU, comprising 768 CUDA cores and 4 GB GDDR5 frame buffer for intensive computation. The GPU is configured using CUDA 9.0.176 and CuDNN 7.6.0. Anaconda 2019.03 with Python 3.7 is used for the execution platform. As execution environment, Keras 2.2.4 and Tensorflow 1.9.0 are used to implement the network. The network is trained on the sign language hand gesture dataset collected in various scenarios. It is trained over 100 epochs for 32 batch sizes and network weights are updated using backpropagation.

4.2. Dataset

Dataset consists of 20 Indian Sign Language Hand Gestures in complex backgrounds. The entire dataset is inspired by the Indian Sign Language Research and Training Centre (ISLRTC) [19]. Twenty sign language gestures were selected from the ISL Dictionary. Selected signs were for the words- “Ache”, “Admire”, “Adult”, “Answer”, “Ask”, “Calm”, “Close”, “Compare”, “Complain”, “Confuse”, “Decide”, “Develop”, “Discuss”, “Doubt”, “Edit”, “Finish”, “Group”, “Important”, “Like”, and “Near”.

Training and Validation samples of gestures were collected from 10 persons in 2 different light conditions, three different orientations, and doubled by vertical flipping of whole data as video data augmentation process, which makes 120 video samples per gesture. The overall video sample count is 2400, which is further divided 80:20 ratio for training and testing sets. It means that 1920 samples are used to train the network while the remaining 480 samples are used for testing purposes. Data used for testing the trained network model was isolated at the time of training. Frames extracted from the video samples were found to be 162,840. The screenshots of some of the respective gestures are illustrated in fig. 2. The synthesized hand gesture dataset includes males and females from different age groups (mostly 22-35 year olds) to capture video clips. All the video clips are captured in different environmental conditions, such as lightning variation and background scenes (e.g., simple and complex). This makes the dataset more suitable for the purpose of the study.

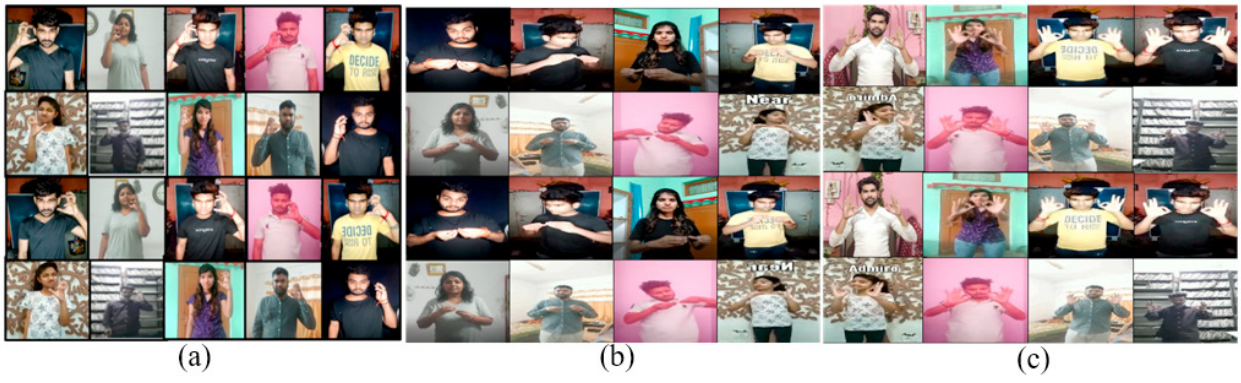

Fig. 2. Sample screenshots for gesture : (a) ”Ache” (b) ”Near”, (c) ”Admire”

4.3. Performance Measures

Performance measurs are used to evaluate the network performance of the proposed model. This work uses accuracy, precision, recall, and f1-score as performance measure, which are formulated as shown in equation 2:

$$Accuracy = \frac{TP + TN}{N + P}; \quad Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN}; \quad F1\ score = 2 * \frac{precision}{recall} \tag{2}$$

4.4. Experiments & Analysis

Our trained network model's performance has been analyzed on the test data that was kept separate from validation data. Test data contains 1176 image sequences of different sign language gestures. Test data was collectively fed into the propose model and provides experimental outcomes. Table 2 presents the above-mentioned performance metric's values for each gesture class that is generated by processing test data on the proposed model.

The experiments were executed over 100 epochs for 20 gesture classes and evaluated precision and recall measures for measuring the proposed work's performance. If precision is much higher than recall value as shown in classes like "Adult", "Compare", "Confuse", "Discuss", and "Group", the number of false-positive instances decreases while false negative instances increases. On the other hand, if precision is lower than recall value as shown in classes like "Ache", "Calm", "Complain", "Develop", and "Near", the number of false-positive instances increases while false negative decreases. The evaluated metrics in table 2 shows high precision and recall value for four classes that are "Answer", "Ask", "Decide", and "Doubt" and produces 100% results in term of f1 measure, which shows all the instances have been correctly classified. The class named "group" has a complicated symbol representation that shows a fall in f1-measure, which means most instances have been misclassified.

The training and validation accuracy for 32 batch sizes are graphed in fig. 3. It shows that training and testing accuracy are almost the same up to 7 epochs, while it achieves 99.67% and 88% accuracy for training and validation, respectively, over 100 epochs.

Table 2. Experimental Outcomes

Class	Precision	Recall	F-1 score	Support
Ache	0.778	1.000	0.875	49
Admire	0.776	0.789	0.783	57
Adult	0.926	0.893	0.909	56
Answer	71.000	1.000	1.000	68
Ask	1.000	1.000	1.000	50
Calm	0.833	1.000	0.909	50
Close	0.895	0.879	0.887	58
Compare	1.000	0.873	0.932	55
Complain	0.907	1.000	0.951	49
Confuse	1.000	0.904	0.949	52
Decide	1.000	1.000	1.000	61
Develop	0.764	0.873	0.815	63
Discuss	1.000	0.922	0.959	64
Doubt	1.000	1.000	1.000	64
Edit	0.964	0.947	0.956	57
Finish	0.741	0.729	0.735	59
Group	0.585	0.407	0.480	59
Important	0.885	0.821	0.852	56
Like	0.911	0.923	0.917	78
Near	0.773	0.817	0.795	71
Average (%)	88.69	88.88	88.52	58.80

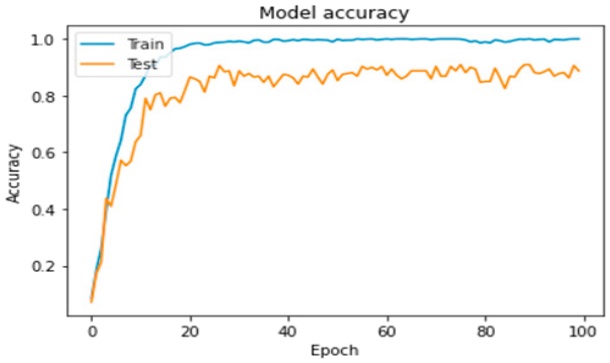

Fig. 3. Training and validation accuracy

The confusion matrix generated from the test data is shown in the fig. 4, where rows depict the true labels and columns presents the predicted labels. Each cell in the matrix gives count corresponding to the gesture of one row when predicted by the model as a gesture listed in different column's. The primary diagonal shows the number of elements which are correctly predicted by the model.

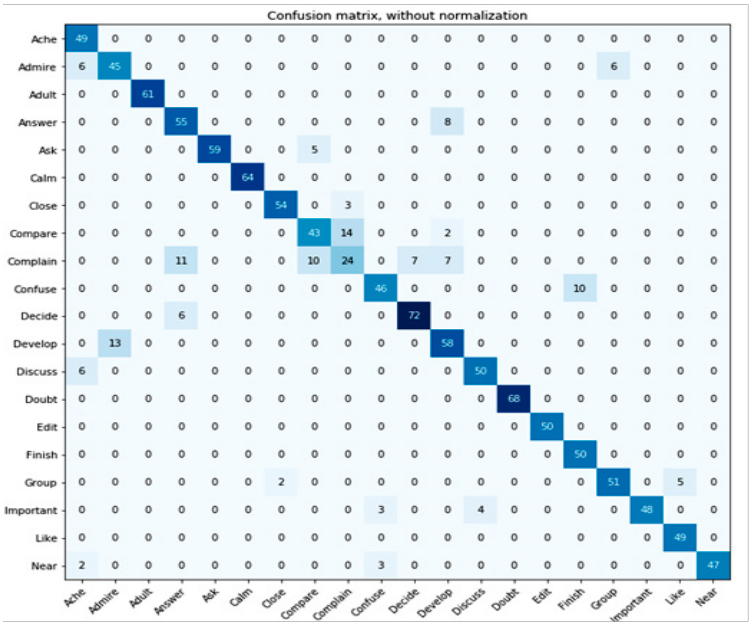

Fig. 4. Confusion matrix for test data result

Table 3 presents the comparison of proposed model with existing state-of-the-art appraoches. The existing appraoches are implemented in our self created dataset and found that our proposed model is providing efficient outcomes in terms of accuracy compared to others' existing approaches.

Table 3. Comparison with existing approaches

Model	Ionescu et al. [13]	Varun et al. [20]	Okan et al. [4]	Proposed Model
Accuracy	86.18%	81.85%	87.86%	88.24%

5. Conclusion

We aimed to make a deep neural network that would model and recognize the Hand Gestures of standard Indian Sign Language. We have collected the dataset from different groups of age and complex backgrounds. The base 3D-CNN architecture is used for analyzing the modeling exercise for these dynamic gestures. Experimentation outcomes justifies the model performance with a good accuracy values achieved. Every person does not commonly know hand gestures in society. Moreover, each country is having its own set of symbols is a challenge. Standardization at a global level is not there, which leads to intricacy in communicating outside the country. Nevertheless, we have modeled the gestures with the anticipation of having a large dictionary that would convert one country's hand gestures to another.

References

[1] X. Shen, G. Hua, L. Williams, and Y. Wu. Dynamic hand gesture recognition: An examplar based approach from motion divergence fields. Image and Vision Computing, 2012.

[2] Dwivedi N., Singh D.K. (2019) Review of Deep Learning Techniques for Gender Classification in Images. In: Yadav N., Yadav A., Bansal J., Deep K., Kim J. (eds) Harmony Search and Nature Inspired Optimization Algorithms. Advances in Intelligent Systems and Computing, vol 741. Springer, Singapore. https://doi.org/10.1007/978-981-13-0761-4_102

[3] Ojha, Utkarsh, Utsav Adhikari, and Dushyant Kumar Singh. "Image annotation using deep learning: A review." 2017 International Conference on Intelligent Computing and Control (I2C2). IEEE, 2017.

[4] Köpüklü, Okan, et al. "Real-time hand gesture detection and classification using convolutional neural networks." 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), IEEE, 2019.

[5] Ansari, Aquib, and Dushyant Kumar Singh. "An Approach for Human Machine Interaction using Dynamic Hand Gesture Recognition." 2019 IEEE Conference on Information and Communication Technology. IEEE, 2019.

[6] Karami A, Zanj B, Sarkaleh AK, "Persian sign language (PSL) recognition using wavelet transform and neural networks". Expert System Applications, volume 38: pages 2661–2667, 2011.

[7] Kim J-S, Jang W, Bien Z, "A dynamic gesture recognition system for the Korean sign language (KSL)". IEEE Transactions on System, Man Cybernetics 26:354–359

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale visual recognition. In ICLR, 2015.

[9] Z. Yang, Y. Li, W. Chen and Y. Zheng, "Dynamic hand gesture recognition using hidden Markov models," 2012 7th International Conference on Computer Science & Education (ICCSE), Melbourne, VIC, 2012, pp. 360-365, doi: 10.1109/ICCSE.2012.6295092.

[10] Huong TN, Huu TV, Le Xuan T, "Static hand gesture recognition for Vietnamese sign language (VSL) using principle components analysis". International conference on communications, management and telecommunications (ComManTel), IEEE, pp. 138–141. 2015

[11] M. I. N. P. Munasinghe, "Dynamic Hand Gesture Recognition Using Computer Vision and Neural Networks," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, 2018, pp. 1-5, doi: 10.1109/I2CT.2018.8529335.

[12] F. K. H. Quek, "Toward a vision-based hand gesture interface", Proc. Virtual Reality Software and Technology Conference (VRST '94), pp. 17–29, Singapore, Republic of Singapore.

[13] Ionescu, B., Coquin, D., Lambert, P. et al. "Dynamic Hand Gesture Recognition Using the Skeleton of the Hand". EURASIP J. Adv. Signal Process.236190, 2005

[14] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre. "Skeleton-Based Dynamic Hand Gesture Recognition" in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 1-9, 2016.

[15] Akmeliawati R, Dadgostar F, Demidenko S, Gamage N, Kuang YC, Messom C, Ooi M, Sarrafzadeh A, SenGupta G, "Towards real-time sign language analysis via markerless gesture tracking". Instrumentation and measurement technology conference, I2MTC'09, IEEE, pp 1200–1204, 2009.

[16] Shreyashi Narayan Sawant and M. S. Kumbhar, "Real Time Sign Language Recognition using PCA", IEEE International Conference on Advanced Communication Control and Computing Technologies (lCACCCT), 2014.

[17] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. "Multiscale deep learning for gesture detection and localization" In ECCVW, 2014.

[18] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning spatiotemporal features with 3d con-volutional networks". In ICCV, 2015

[19] Indian Sign Language Research and Training Centre (ISLRTC), Weblink: http://islrtc.nic.in

[20] K. S. Varun, I. Puneeth and T. P. Jacob, "Hand Gesture Recognition and Implementation for Disables using CNN'S," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0592-0595, doi: 10.1109/ICCSP.2019.8697980.

[21] Polat, Huseyin, and Homay Danaei Mehr. "Classification of pulmonary CT images by using hybrid 3D-deep convolutional neural network architecture." Applied Sciences 9.5 (2019): 940.

[22] Singh, Dushyant Kumar. "Recognizing hand gestures for human computer interaction." 2015 International Conference on Communications and Signal Processing (ICCSP). IEEE, 2015.