



ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks

Shikhar Sharma¹ · Krishan Kumar¹

Received: 12 June 2020 / Revised: 8 February 2021 / Accepted: 25 February 2021 /
Published online: 1 May 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The communication between a person from the impaired community with a person who does not understand sign language could be a tedious task. Sign language is the art of conveying messages using hand gestures. Recognition of dynamic hand gestures in American Sign Language (ASL) became a very important challenge that is still unresolved. In order to resolve the challenges of dynamic ASL recognition, a more advanced successor of the Convolutional Neural Networks (CNNs) called 3-D CNNs is employed, which can recognize the patterns in volumetric data like videos. The CNN is trained for classification of 100 words on Boston ASL (Lexicon Video Dataset) LVD dataset with more than 3300 English words signed by 6 different signers. 70% of the dataset is used for Training while the remaining 30% dataset is used for testing the model. The proposed work outperforms the existing state-of-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%). The computing time (0.19 seconds per frame) of the proposed work shows that the proposal may be used in real-time applications.

Keywords 3D CNN · Speech · ASL · Deep Learning

1 Introduction

It is known that roughly 2,500,000 people from all over the world uses sign language to communicate. ASL for word recognition is an example of two-handed sign language comprising of dynamic hand gestures for spelling American English. It incorporates hand movements in the air for communicating with others by hearing impaired. The demand for human translators for smooth communication between deaf and hearing communities is elemental. However, this is ineffectual, upscale, and inconvenient as human competence is mandatory.

Krishan Kumar is a Senior Member, IEEE

✉ Krishan Kumar
kkberwal@nituk.ac.in

Shikhar Sharma
shikhar01.cse14@nituk.ac.in

¹ Department of Computer Science and Engineering, National Institute of Technology Uttarakhand, Srinagar, Garhwal, India

Ong et al. [30] presented the use of tree-structured boosted cascades to classify different hand gestures. This achieved a good average accuracy on their dataset. However, the classifier is trained on assumed hand gestures instead of standard hand gestures. Also, the time complexity of the presented method is high. The work of Issacs et al. [7] focuses on the recognition of static ASL fingerspelling letters. They used a wavelet feature detector on intensity maps classified using a Multi-Layer Perceptron (MLP) having 24 classes. In the ASL LVD [2], each sign is signed by native ASL signers. The video sequences are captured from four different viewpoints simultaneously. Two of them are frontal views, one side view, and one is a slightly zoomed view on the face of the signer. Moreover, the annotations are included on the top of the video sequences for each signer. The dataset presents challenges that are relevant to areas such as machine learning, computer vision, and data mining. It includes the discrimination of visual motion of hand gestures into thousands of classes. Figure 1 depicts the frames from a sample video sequence from three different views.

Some approaches [5, 36, 38] used sensor gloves and magnetic trackers, so can't be considered as vision-based systems. While some vision-based methods are also presented for smaller vocabularies (20-100 signs) and mainly relied on color markers. The above-mentioned previous studies have attempted to tackle the problem of using the combination of image processing and machine learning techniques. These studies focus on tracking the hand motion, hand shape recognition using machine learning as well as pattern recognition techniques to classify the ASL LVD words. These words are widely used in ASL recognition. Talking about the application of ASL, it can vary from conveying the names, addresses, and most importantly communication between an impaired person and other community. The recognition of dynamic hand gestures poses many challenges in the case of ASL LVD. The motion of hands, trajectory of motion, hand shape, and multi-view information may vary drastically for different signers. Also, the variation acquired due to different camera viewpoint complicates the situation.

In this paper, the authors intent to overcome the challenges posed by ASL LVD. The concept of 3-D CNN cascaded for different viewpoints is implemented to analyze the multi-modal information more competently with more accurately against the existing models.

Also, with the advancement in hardware technologies of GPU, the use of CNN's in computer vision challenges has increased in past few years. Therefore, we processed the video information using 3-D CNN cascaded for learning from different viewpoints. The performance of the proposal has been comprised of the existing state-of-the-art models on dynamic hand sign recognition. The various salient features of our work are:

- 3-D CNN's cascaded is employed as a deep learning framework for recognizing the dynamic hand sign with better accuracy.
- The proposed work can work for Signer-independent as well as surroundings independent.
- To consider the different viewpoints, the cascaded CNNs have been deployed for better training.



Fig. 1 A sample video sequence from ASL LVD [2]

- The proposed work outperforms the existing state-of-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%).
- The computing time (4 milliseconds per frame) of the proposed work shows that the proposal may be used in real-time applications.

2 Related work

In the past few decades, many interesting works [28, 32, 37] have been presented to the recognition of dynamic hand gestures for sign language. The ASL dynamic hand gesture recognition is still a challenging problem despite efforts made in the field during the last decade. The requirement for the understanding of multi-modal information like hand gesture and movement in case of ASL where it makes the problem more ambiguous is very acute. Moreover, a large number of words in sign language having similar gestures with a lesser number of samples for each word makes the problem more difficult. Sometimes, the same signs from different signers or different viewpoints have different appearances. While the converse that is different signs from different viewpoints or different signers look the same.

Koller et al. [12] marked the beginning of ASL recognition using desk and wearable computer based videos. They used two different cameras and achieved different accuracies corresponding to them. One of the cameras is mounted on the user desk while another one is mounted on the cap worn by the user. The model presented used 40 words in the lexicon. The model used the Hidden Markov model (HMM) consisting four states for word recognition. Cui et al. [4] used intensity-based image sequences for hand sign recognition. They focused on hand shape recognition for hand sign recognition. They deployed the use of multidimensional discriminant analysis for selecting discriminating linear features. The features obtained are classified using a recursive partition tree approximator.

Liwicki et al. [26] used a Histogram of Oriented Gradients method to recognize British Sign Language (BSL) fingerspelling and achieved good results. Uebersax et al. [35] presented the use of Average Neighborhood Margin Maximization (ANMM) for 26 dynamic hand poses in ASL using depth maps. They achieved good accuracy using a dataset containing at least 50 images per alphabet of seven different signers. The use of deep learning techniques in sign language is deployed for recognizing Italian Sign Language [31]. The use of CNNs for ASL fingerspelling recognition was first presented by Kang et al. [10]. They used depth maps for hand segmentation, those segmented images are then fed to CNN (AlexNet) for classification. Ameen et al. [1] also tried to solve ASL fingerspelling recognition using CNN on both depth and intensity maps.

Kinect sensor [3] is used to obtain the color and depth gesture samples, and the gesture samples are processed. On this basis, a joint network of CNN and RBM is deployed for gesture recognition. In another work [23], medical images of the liver and chest X-ray of different human organs have been segment using fuzzy theory and region growing algorithm. By considering the disadvantages of greedy algorithms in sparse solution, a modified adaptive orthogonal matching pursuit algorithm (MAOMP) [24] is proposed to estimate the initial value of sparsity by matching test and will decrease the number of subsequent iterations. The step size is adjusted to select atoms and approximate the true sparsity at different stages. Aiming at the problem of the high computational complexity of the l_1 norm-based solving algorithm, a l_2 norm local sparse representation classification model

is presented [6]. SignFi [27] is proposed to recognize sign language gestures using WiFi. SignFi employed Channel State Information determined by WiFi packets as the input and a CNN as the classification algorithm. SCANet [22] utilizes two-stream CNN to learn and extract representative features and then performs the principal component analysis to select the top 25 features with high discriminability.

To analyze the video contents, a CNN based model has been proposed to summarize the important events in a video [14]. In other work, a deep CNN model has been presented to recognize the ASL [33]. A data-driven system with 3D-CNNs is employed to extract spatial and temporal features from video streams, and the motion information is captured by noting the variation in depth between each pair of consecutive frames [25]. An approach exploits sequence constraints within each independent stream and combines them by explicitly imposing synchronization points to make use of parallelism that all sub-problems share [13]. This with multi-stream HMMs while adding intermediate synchronization constraints among the streams. A computer-assisted cognitive assessment method based on the Praxis test is proposed [29]. Four methods are developed to evaluate dynamic and static gestures in the Praxis test. A challenging RGB-D dataset is collected consisting of 60 subjects and 29 gestures.

3 Proposed work

The proposal has been divided into two sections as the ASL LVD video sequences are pre-processed and then recognized using cascaded 3-D CNNs. Sections 3.1 and 3.2 describes about pre-processing done for better training of cascaded CNNs and cascaded CNNs respectively. The major components of the proposed model are shown in Fig. 2.

3.1 Pre-processing

To effectively train the CNNs, some pre-processing has been done. This reduces the chances of CNNs being trained on noising elements resulting in degraded performance. Since pre-processing is only done while training the network so it is a prior expense of time. Below we outline the various pre-processing stages.

- Each video sequence is first converted into several frames, then each frame is processed individually.
- Original color frame is first converted to a gray-scale image. Then unwanted noise and spots in the frame are removed using median filtering.
- The illumination variations in the frame are canceled out using Histogram equalization. To reduce the computation, each frame is resized to 512×384 and normalized to $[0, 1]$.
- Each video sequence is then reduced to the size of 25 distinct frames.
- The processed frames are then combined again to form the video sequence for training 3-D CNNs.

The above process generates the processed video sequences having gray-scale frames. The video sequences that are processed were manually trimmed. This ensures only hand gestures and motions to be present in video sequences for training CNNs. As outlined earlier, the works presented to recognize dynamic ASL are less in number as compared to static ASL recognition. Several authors have tried various feature extraction methods followed by the use of different learning techniques like HMMs, Recursive partition tree, and ANMM. But the use of deep learning techniques has not yet been presented. So, we tried to explore the

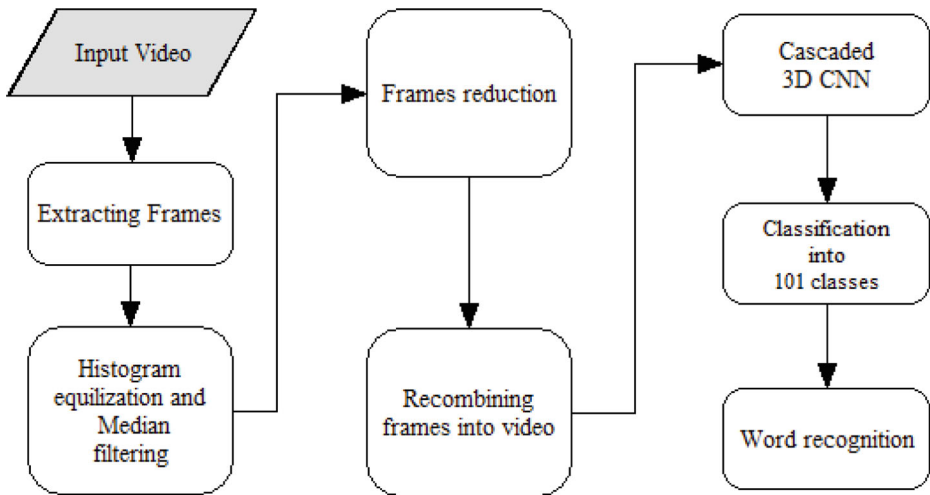


Fig. 2 Various components of the proposed model

deployment of CNNs to resolve the problem of dynamic ASL recognition. Algorithm 1 elaborated the flow of the proposed model.

Algorithm 1 ASL-3DCNN System.

```

1: procedure ASL3DCNN(V)
2:    $V = f_1, f_2, \dots, f_N$ 
3:   for  $i = 1$  to  $N$ 
4:      $g_i = \text{Gray\_scale}(f_i)$ 
5:      $h_i = \text{Noise\_filter}(g_i)$ 
6:      $r_i = \text{Histogram}(h_i)$ 
7:      $s_i = \text{resize}(r_i)$ 
8:      $s_1, s_2, \dots, s_N$  reduced to  $p_1, p_2, \dots, p_{25}$ 
9:     for  $i = 1$  to 25
10:       $o_i = \text{cascaded\_3CNN}(p_i)$ 
11:       $w_i = \text{class}_{o_i \in \{1, 2, \dots, 101\}}(o_i)$ 
12:   return  $w_i$ 
13: end procedure
  
```

3.2 Deep sign recognition architecture

The concept of neural networks came into existence by the works [15, 17–21], while the concept of deep learning is coined fairly recently in the mid-2000s by Hinton and his collaborators [8, 9]. Figure 3 shows architecture of CNN in proposed method.

As the name suggests, it focuses on the development of a sequence for feature recognition maps, stacking one layer on top of the previous layer, and where each layer recognizes the extended features provided the previous layer, with the final layer performing classification [39]. For example, to recognize objects in images, the first layer learns to understand patterns in edges, the second layer combines that pattern of edges to form motifs, the next

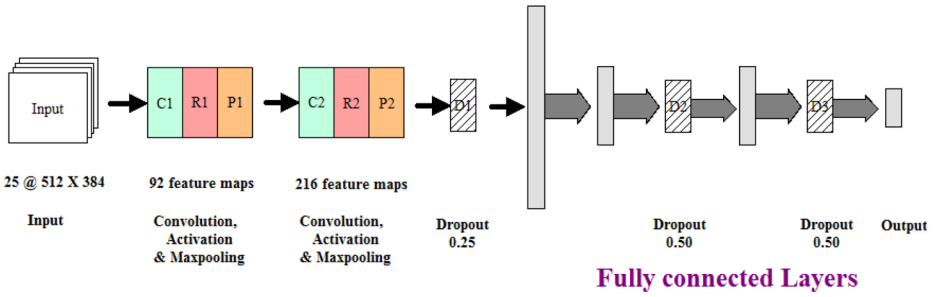


Fig. 3 Work flow of the proposed model

layer learns to combine motifs to attain patterns in parts, and the final layer learns to recognize objects from the parts identified in the previous layer [16]. The summary of CNN is given in Table 1.

The convolution operation is widely used in the field of image processing. The convolution layers on CNN also work on the same principle. The convolution of an input x with kernel k is computed by (1), where x is an image in the input layer or a feature map in the subsequent layers. The convolution kernel, k is a square matrix having dimension specified by the user. The number of feature maps is a hyper-parameter that is determined experimentally. For a 3D kernel, the convolution is defined as in (1).

$$(x * k)_{ijm} = \sum_{p,q,r=0}^{s-1} (x_{i+p,j+q,m+r}) \times (k_{s-p,s-q,s-m}) \tag{1}$$

In our work, x is the pixel, k is kernel, i, j and m are 3 Dimensional values, 92 feature maps or filters are used after the input layer. The kernel size defines the receptive field of the hidden neurons in feature maps.

Table 1 Configuration of 3-D CNN model for Sign recognition

Fully connected	#neurons: 101
Dropout	Ratio : 0.5
ReLU	
Fully connected	#neurons: 1024
Dropout	Ratio : 0.5
ReLU	
Fully connected	#neurons: 4096
Dropout	Ratio : 0.25
Maxpooling	kernel : $2 \times 2 \times 2$, stride : 2
ReLU	
Convolution	#filters: 216, kernel : $15 \times 15 \times 3$, stride : 1
Maxpooling	kernel : $2 \times 2 \times 2$, stride : 2
ReLU	
Convolution	#filters: 92, kernel : $25 \times 25 \times 6$, stride : 1
Input	$512 \times 384 \times 25$ gray-scale video

In the case of the 3D kernel, the last dimension specifies the number of frames falls into the receptive field. It acts as the filter for searching a specific pattern in the input image. The stride defines the movement of the kernel across the input image. Lesser the stride more accurate the feature maps regarding the patterns. In our model, we took the stride of 1 in all the kernel dimensions. We used ReLU (Rectified Linear Unit) as activation function [40], which enhances the learning process of the network. For the input x the output of ReLU is defined as in (2).

$$f(x) = \max(0, x) \quad (2)$$

A smooth approximation to ReLU is the analytic function also called soft plus function is defined as in (3).

$$f(x) = \ln(1 + e^x) \quad (3)$$

To avoid the exploding gradient problem, we employed dropout layers having the ratio to be 0.50 & 0.25 [34]. Dropout layers neglect the input from some neurons in previous layers. This avoids the exploding as well as the vanishing of the gradient. Moreover, this also avoids the over-fitting of the network while training, promising higher accuracy of test data. A pooling operation is applied to reduce the impact of translations and reduces the number of trainable parameters that would be needed. All the layers discussed above collectively act as a single convolution layer. In our model, we deployed two convolution layer. One having 92 feature maps and the other having 216 feature maps. Moreover, we also changed the kernel size in the next layer for better training and testing.

A fully connected layer could be understood as the feed-forward neural network. The feature maps obtained after both convolution layers act as input to fully connected layers. The last convolution layer contains 216 feature maps with a matrix having $115 \times 78 \times 4$ as the dimension is reshaped into a single 7,750,080 dimensional vector. This vector act as input to a three-layer neural net with 4096 nodes in the first hidden layer, 1024 in the second hidden layer, and 101 class nodes, one for each word in the lexicon and last one for *NULL* denoting that the word doesn't belong to above 100 classes. The softmax layer transforms the output of various neurons into probability.

With time, many robust and fast training algorithms have been presented by many authors. More recently, Kingma et al. [11] proposed a new training algorithm called Adam optimization basically in neural networks for speeding up the learning process. They used the concept of second-order moments and their correction in training. We used an Adam optimization technique for backpropagating the error. It has many benefits over the traditional Stochastic gradient descent method (SGD). It removes the major drawback of SGD viz. slow training. The various parameters in Adam optimization are stepsize (α), exponential decay rates for moment estimation (β_1, β_2), objective function $f(\theta)$ and moment vectors (m_0, v_0).

The gradient concerning $f(\theta)$ at any time instance is defined as in (4).

$$g_t = \nabla_{\theta} f_t(\theta_{t-1}) \quad (4)$$

The moment estimates are updated as in (5).

$$\begin{aligned} m_t &= \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t, \\ v_t &= \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \end{aligned} \quad (5)$$

These moments are then used to obtain corrected moments as defined in (6).

$$m_t = \frac{m_t}{1 - \beta_1^t}, v_t = \frac{v_t}{1 - \beta_2^t} \quad (6)$$

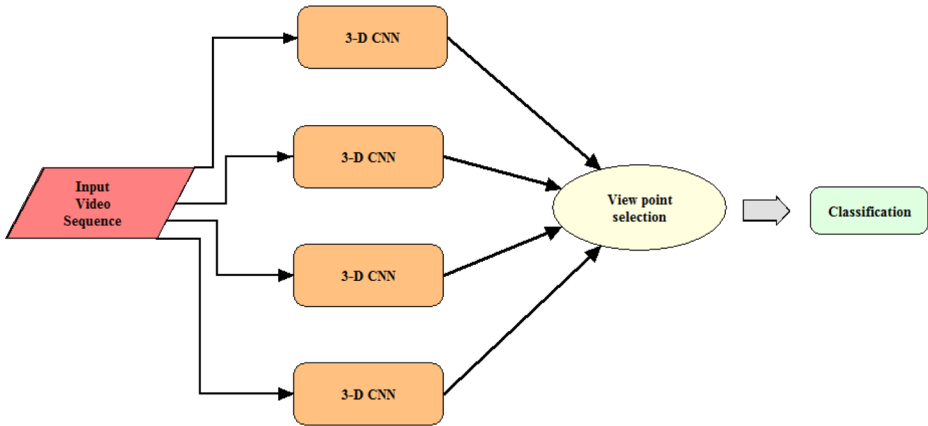


Fig. 4 Cascaded CNN architecture used in proposed work

These moments are then used to update the parameters used in the network as defined in (7),

$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot m_t}{\sqrt{v_t} + \epsilon} \quad (7)$$

Where the ϵ is set to 10^{-8} by default. Here, the objective function is categorical cross-entropy as defined in (8).

$$f(\theta) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(p_{ij}) \quad (8)$$

Where n is the number of sampled and m is a number of categories.

The traditional Mean Squared Error (MSE) emphasis more on the incorrect outputs, so it is slightly ineffective if used along with the softmax layer. Figure 4 illustrates the cascaded 3-D CNN structure as used in the proposed work.

The use of 3-D CNN overcomes the problem of dynamic hand gestures in the ASL Lexicon Video dataset. Moreover, the video sequences in the dataset are recorded from four different viewpoints. So, to increase the efficiency of training we used four 3-D CNNs, each one for different viewpoints. These four different CNNs are trained using video sequences of a particular viewpoint. Once trained for all different viewpoints, it can be used to classify words pertaining to any one of four viewpoints. Figure 5 shows some sample predicted using cascaded 3-D CNN.

4 Experiment and discussion

As mentioned earlier, the use of deep learning techniques in ASL LVD¹ recognition has not been presented so far in the best of our knowledge. The architecture was implemented using the Python library Keras for deep learning based on CUDA as well as CuDNN on a standard dual-core computer having NVIDIA GeForce GTX 660 GPU (2GB memory and 960 CUDA cores).

¹<http://www.bu.edu/av/asllrp/dai-asllvd.html>



Fig. 5 Illustrations of signs predicted by 3-D CNN

To enable comparison, the same experimental methodology as [4, 32, 35] is adopted. That is, given n users, a model is first developed using the data from the first $n - 1$ users and tested by the n th user. Next, a model is trained in all the data except the $(n - 1)$ th and tested on the $(n - 1)$ th user, etc. This results in n values, which are averaged to produce estimates of the precision and recall measures. The dataset is split into two parts: *Training* (70%) and *Testing* (30%) datasets.

Table 2. shows the comparison of average precision, recall, and F-measure values obtained by previous works and our work for different signers given by (9), (10), and (11)

Table 2 Comparison of precision, recall and F-measure

Model	Precision (%)	Recall (%)	F-measure (%)
Pentland et al. [32]	92.3	92.8	92.5
Cui et al. [4]	78.9	86.5	82.7
Uebersax et al. [35]	77.8	85.4	81.5
Koller et al. [12]	78.6	86.2	82.4
Proposed work	96.0	97.1	96.4

Best results are shown in bold

Table 3 Computing time comparison

Model	Training epochs	Execution Time per frame(sec)
Pentland et al. [32]	030	0.56
Uebersax et al. [35]	250	0.97
Cui et al. [4]	200	1.76
Koller et al. [12]	80000	0.49
Proposed work	010	0.19

Best results are shown in bold

respectively.

$$P = \frac{TP}{TP + FP} \quad (9)$$

$$R = \frac{TP}{TP + FN} \quad (10)$$

$$F_{\beta} = \frac{(1 + \beta^2) \times P \times R}{\beta^2 \times P + R} \quad (11)$$

Where, P , R and F_{β} denotes precision, recall and F-measure respectively. For equal priorities of P and R , we used $\beta = 1$. The results are evaluated by computing recall and precision measures for each word. These values are then averaged to get final recall and precision values. The experiments were run for 10 epochs until the network converged. The use of Adam optimization in training the network instead of the traditional Stochastic Gradient Descent (SGD) approach has reduced the training time drastically. Also, the use of GPU for computation also aids in saving training time.

The Table 3 consists the time requirement of proposed work and existing models. From the Table it is obvious that the proposed model meets the real time constraints of the problem. This help us to process 5 frames having dimensions 512×384 in one second from an input video.

5 Conclusion and future work

This work highlights ASL based dynamic gesture recognition system. It could automatically recognize sign language; would be beneficial for the people that use sign language to communicate. It can classify dynamic ASL hand signs into 100 different words. We showed the efficiency of using convolution neural networks on both depths as well as intensity maps for ASL fingerspelling recognition system. The evaluation of the presented work has shown the promising performance of the method with lesser time requirements. The proposed work outperforms the existing state-of-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%). The computing time (0.19 seconds per frame) of the proposed work shows that the proposal may be used in real-time applications. Also, to the best of our knowledge, we are first to present the use of deep learning for ASL LVD recognition. Moreover, the challenge of using both hands can be solved using spatial transform layers along with CNN to recognize in multi-label environment.

References

1. Ameen S, Sunil V (2017) A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images, *Expert Systems*
2. Athitsos V et al (2008) The american sign language lexicon video dataset, *Computer Vision and Pattern Recognition Workshops, IEEE Computer Society Conference on*
3. Cheng WT, Sun Y, Li GF, Jiang GZ, Liu HH (2019) Jointly network: A network based on CNN and RBM for gesture recognition. *Neural Comput Appl* 31(Suppl 1):309–323
4. Cui Y, Juyang W (2000) Appearance-based hand sign recognition from intensity image sequences. *Comput Vision Image Understand* 78.2:157–176
5. Gao W, Fang G, Zhao D, Chen Y (2004) Transition movement models for large vocabulary continuous sign language recognition. *Autom Face Gesture Recognit* 553–558
6. He Y, Li GF, Liao YJ, Sun Y, Kong JY, Jiang GZ, Jiang D, Liu HH (2019) Gesture recognition based on an improved local sparse representation classification algorithm. *Clust Comput* 22(Suppl 5):10935–10946
7. Isaacs J, Foo S (2004) Hand pose estimation for american sign language recognition, *System Theory, 2004. In: Proceedings of the thirty-sixth southeastern symposium on. IEEE*, pp 132–136
8. Hinton G, Osindero S, Teh Y (2005) A fast learning algorithm for deep belief nets. *Neural Comput* 18:1527–1554
9. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313:504–507
10. Kang B, Tripathi S, Nguyen TQ (2015) Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In: *Pattern recognition (ACPR), 3rd IAPR asian conference on. IEEE*
11. Kingma D, Ba J (2014) Adam: A method for stochastic optimization, arXiv:1412.6980
12. Koller O et al (2016) Deep sign: Hybrid CNN-HMM for continuous sign language recognition. *Proc British Machine Vision Conf* 1–6
13. Koller O et al (2019) Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Trans Pattern Anal Machine Intell*
14. Kumar K, Shrimankar D (2017) F-DES: Fast and deep event summarization. *IEEE Trans Multimed* 20(2):323–334
15. Lecun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, vol 3361
16. Lecun Y, Bengio Y, Lhinton G (2015) Deep learning. *Nature* 521:436–444
17. Lecun Y, Boser B, Denker GE, Henderson D, Howard RE, Hubbard W et al (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1:541–551
18. Lecun Y, Boser B, Denker JS, Howard RE, Hubbard W, Jackel LD, Henderson D (1990) Handwritten digit recognition with a back-propagation network. *Adv Neural Inform Process Syst* 396–404
19. Lecun Y, Bottou L, Orr G, Müller K-R (1989) Efficient BackProp. In: Orr G, Müller K-R (eds) *Neural networks: Tricks of the trade*, vol 1524. Springer, Berlin, pp 9–50
20. Lecun Y, Galland CC, Hinton GE (1988) GEMINI: Gradient Estimation through matrix inversion after noise injection. *InNIPS* 141–148
21. Lecun Y, Jackel L, Boser B, Denker J, Graf H, Guyon I et al (1990) Handwritten Digit recognition: Applications of neural net chips and automatic learning, *Neurocomputing*. Springer, Berlin, pp 303–318
22. Li Y, Hailong H, Zhangqian Z, Gang Z (2020) SCANet: Sensor-based continuous authentication with two-stream convolutional neural networks. *ACM Trans Sensor Netw (TOSN)* 16(3):1–27
23. Li GF, Jiang D, Zhou YL, Jiang GZ, Kong JY, Manogaran G (2019) Human lesion detection method based on image information and brain signal. *IEEE Access* 7:11533–11542
24. Li GF, Tang H, Sun Y, Kong JY, Jiang GZ, Jiang D, Tao B, Xu S, Liu HH (2019) Hand gesture recognition based on convolution neural network. *Clust Comput* 22(Suppl 2):2719–2729
25. Liang Z-J, Liao S-B, Hu B-Z (2018) 3D convolutional neural networks for dynamic sign language recognition. *Comput J* 61.11:1724–1736
26. Liwicki S, Everingham M (2009) Automatic recognition of fingerspelled words in british sign language. In: *Computer vision and pattern recognition workshops IEEE Computer Society Conference on*, pp 50–57
27. Ma Y, Gang Z, Shuangquan W, Hongyang Z, Woosub J (2018) SignFi: Sign language recognition using WiFi. *Proc ACM on Interact Mob Wearable Ubiquitous Technol* 2(1):1–21
28. Ma J et al (2000) A continuous chinese sign language recognition system. *Automat Face Gesture Recognit* 428–433

29. Negin F et al (2018) PRAXIS: Towards automatic cognitive assessment using gesture recognition. *Expert Syst Appl* 106:21–35
30. Ong E-J et al (2004) A boosted classifier tree for hand shape detection. *IEEE Autom Face Gesture Recognit* 889–894
31. Pigou L et al (2014) Sign language recognition using convolutional neural networks, Workshop at the European Conference on Computer Vision. Springer, Cham
32. Sagawa H et al (2000) A method for recognizing a sequence of sign language words represented in a Japanese Sign Language sentence. *Autom Face Gesture Recognit* 434–439
33. Sharma S, Kumar K, Singh N (2020) Deep Eigen Space based ASL Recognition System, *IETE Journal of Research*
34. Srivastava N et al (2014) Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958
35. Uebersax D, et al. (2011) Real-time sign language letter and word recognition from depth data. *Computer Vision Workshops IEEE International Conference on*
36. Vogler C, Metaxas DN (2003) Handshapes and movements: Multiple-channel American Sign Language recognition. *Gesture Workshop* 247–258
37. Wang C, Shan S, Gao W (2002) An approach based on phonemes to large vocabulary Chinese Sign Language recognition. *Autom Face Gesture Recognit* 411–416
38. Yao G, Yao H, Liu X, Jiang F (2006) Real time large vocabulary continuous sign language recognition based on OP/viterbi algorithm. *Int Conf Pattern Recognit* 3:312–315
39. Yosinski J et al (2014) How transferable are features in deep neural networks? In: *Advances in neural information processing systems*, pp 3320–3328
40. Zeiler MD et al (2013) On rectified linear units for speech processing. In: *Proc. ICASSP*

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Shikhar Sharma is currently pursuing his B. Tech (Computer Science and Engineering) from National Institute of Technology, Uttarakand, India. His current research areas are: Computer Vision, image processing, object detection, pattern recognition, machine learning and Deep learning.



Dr. Krishan Kumar is currently shouldering the responsibility as Head of the department in Computer Science & Engineering, National Institute of Technology, Uttarakhand (NITUK), India. Currently, Dr. Kumar is serving IETE Journal of Research (SCI) as an Editor. He received Ph. D. for his thesis title “Performance Enhancement of Events Detection and Summarization Models in Videos over Cloud,” in the Computer Science & Engineering, at Visvesvaraya National Institute of Technology, Nagpur, India in 2019. He has more than 10 years of Teaching, Research and Administrative experience. Dr. Kumar is serving NITUK since July 2014. He supervised 06 M. Tech. students and supervising 03 Ph. D. scholars. He has organized and attended more than 30 Workshops/STTP/STC/FDP including one International Conference. He published more than 75 articles in reputed International Conferences, book chapters and reputed Journals including IEEE Transactions. Currently, Dr. Kumar is working on a DST sponsored project titled “Secret Sharing Scheme Based Technology for Multimedia Security over Cloud”. His research interests include Computer Vision, Artificial Intelligence, Machine Learning, Deep Learning, Video Processing, Multimedia Analysis, Real Time Systems, Virtualization, Cloud Security, Natural Language Processing, and Recommender Systems. He was selected among Top 50 Distinguished HODs (CSE (or) IT Department) in Higher Education across India for the Year 2019 for uLektz Wall of Fame. He is Senior Member IEEE and ACM. He is also Life Member of IETE, ISTE and IUPRAI.