Journal of Scientific & Industrial Research
Vol. 81, November 2022, pp. 1186-1194
DOI: 10.56042/jsir.v81i11.52657

Real Time Static and Dynamic Sign Language Recognition using Deep Learning

P Jayanthi[1], Ponsy R K Sathia Bhama[1]*, K Swetha[2] & S A Subash[2]

[1]Department of Computer Technology, MIT, Anna University, Chennai 600 044, Tamil Nadu, India
[2]Department of Information Technology, MIT, Anna University, Chennai 600 044, Tamil Nadu, India

Received 20 July 2021; revised 29 October 2022; accepted 29 October 2022

Sign language recognition systems are used for enabling communication between deaf-mute people and normal user. Spatial localization of the hands could be a challenging task when hands-only occupies 10% of the entire image. This is overcome by designing a real-time efficient system that is capable of performing the task of extraction, recognition, and classification within a single network with the use of a deep convolution network. The recognition is performed for static image dataset with a simple and complex background, dynamic video dataset. Static image dataset is trained and tested using a 2D deep-convolution neural network whereas dynamic video dataset is trained and tested using a 3D deep-convolution neural network. Spatial augmentation is done to increase the number of images of static dataset and key-frame extraction to extract the key-frames from the videos for dynamic dataset. To improve the system performance and accuracy Batch-Normalization layer is added to the convolution network. The accuracy is nearly 99% for dataset with a simple background, 92% for dataset with complex background, and 84% for the video dataset. By obtaining a good accuracy, the system is proved to be real-time efficient in recognizing and interpreting the sign language gestures.

Keywords: Deaf-mute people, Human-machine interaction, Inception deep-convolution network, Key frame extraction, Video analytics

Introduction

Hand gesture recognition systems have become the heart of technology and are used significantly in Human-Computer Interaction[1,2] (HCI), sign language recognition systems[3–7], commanding electronic devices[8], gaming, android applications, etc. The vision-based hand gesture recognition systems used in human-computer interaction, in which hand tracking is followed by gesture recognition based on extracted hand features using background subtraction methods. Hidden Markov Model (HMM) detects movement and skin colour information solely through visual cues. Many real-time applications were developed using colour glove[9], pyro[10] Kinect sensor[11] (depth sensor), (electric sensor), and to gain input, followed by feature extraction and classification. Other real-time and vision based recognition systems (HCI)[12,13] were used to extract the hand regions from the entire image using object marking approaches,[14] and the recognition was done only on the extracted regions using bounding box collection, sliding window strategy, and region proposal algorithms,[15] followed by classification. Hence, neural networks were found to be more efficient for recognition when they were trained with sufficient amount of data. Hence, recognition systems were built using Convolutional Neural Network (CNN)[16] where the extraction was done using Region Proposal Networks (RPN) and classification was carried out using faster Region CNN.[17] Videos need to be processed in three dimensions to achieve a better learning and recognition rate, hence the 3D CNNs were developed,[18,19] that combines CNN to train spatial features and RNN along with LSTM to train temporal features for hand gesture recognition from videos. The CNN can act directly on raw inputs; inputs are split into channels, and feature representation combines these channels and regularization[20] is also done to improve the performance.

Deep neural networks typically requires enormous data to learn, however, problems like overfitting or underfitting can occur, with overfitting is being more common, which could be prevented using dropout in which randomly dropping of the neurons occurs. Furthermore, to improve the performance of the CNN and to reduce time complexity, batch-normalization is applied. In the existing literature using gloves[21], is wearisome as users have to wear motion sensor gloves whenever needed to translate signs.[22] With this

*Author for Correspondence
E-mail: ponsy@mitindia.edu

knowledge, researchers try to develop a system without the need for gloves, but it has problems with tracking hand movements.

Static Image Recognition Systems

Sign language gestures are stereo typed into two kinds: static and dynamic signs. Static signs are the one which state the gesture in an image. Examples of the static signs of Sign Language (SL) are

1. Except J and Z, the English alphabet.
2. OK.
3. Pray.
4. House.
5. Know, etc…

The static image can be interpreted based on the complex and simple backgrounds, an approach[23] which does use the localization mechanism developed for improving the accuracy of the developed system. Finger spelling signs are recognised using statistical methods[24] of machine learning and deep learning after the signs is processed using Gabor filters, Feature mixtures, etc. Most of the literature states that static sign recognition is done on only finger spelling.[25–29]

Dynamic Video Recognition Systems

Videos contain two different dimensions; one is the spatial and the other is the temporal dimension and they have to be processed in three dimensions. The compact representation of dynamic signs was extracted using a block-based histogram of optical flow.[30] Neural networks with Deep CNN along with key frame extraction could be used for recognition of spatial features of the video.[31,32] Masood *et al.*,[6] proposed a real-time sign language recognition where the spatial features are trained using the inception model deep-CNN while the temporal features are trained using the RNN. At the outset, the input is sent as a video and the processing is done on the images after the video is split up into frames. 3D CNN is being implemented to process the spatial and temporal features of dynamic gestures.

2D Deep CNN

The 2D convolutions use a 2D filter moving in (x, y) directions.[15] The directions specify the spatial dimensions. This can only be used for the image detection mechanisms but this cannot be used in the temporal dimensions as in videos. Deep CNN typically requires a larger dataset[19] to increase learning rate, which has been accomplished using offline and online dataset augmentation techniques.

3D Deep CNN

The 3D convolutions apply a 3D filter moving in (x, y, z) directions. The directions specify the spatial and temporal dimensions. 3D-CNN[5,6,18] is mainly used for event detection in videos. This can also be used for image detection. Before processing the video dataset using the network, the video dataset is split up into key frames.[27] These frames are used in training the 3D-CNN model, and the gestures are efficiently determined in the testing phase.

Regularization Techniques

Regularization is the mechanism that improves the CNN performance by reducing the time complexity and refining accuracy. This can be done by the dropout mechanism[20] to suppress the problem of overfitting that usually occurs in the deep CNN and reducing the internal co-variant shift by adding a batch-normalization layer in between the deep CNN layers. The fundamental notion is to arbitrarily drop units with links in the course of training. During training, dropout samples for an exponential quantity of diverse “thinned” nets. During testing, approximating the effect by calculating the prediction average of all thinned nets, which eases the over fitting effects. Batch Normalization (BN)[15] technique is capable of improving the performance efficiency of the convolution and the non-conv layers of the deep CNN architecture. Batch normalization requires multiple inputs from the previous layer to the normalization task in a mini-batch. First, it calculates the mean and the variance values, later performs normalization using scaling and shifting factors.

The Batch normalization algorithm consists of forward pass and backward pass as depicted in Fig. 1. Batch normalization layers are split up into sub-layers using fission mechanism and later combined with the preceding conv layer, Relu and following conv. This batch normalization process is commonly known as the Batch Normalization Fission and Fusion (BNFF). By doing this process the memory sweeps can be reduced to 1 from 3. Batch normalisation performs various operations such as data reuse optimization, pruning and approximate computing, fusion and blending layers, and training acceleration.

From all the observations it is clear that the existing recognition systems had to implement different algorithms for each phase, subsequently proposed work focuses on recognition mechanism without localization produced better result and the major idea is to combine the detection, extraction,

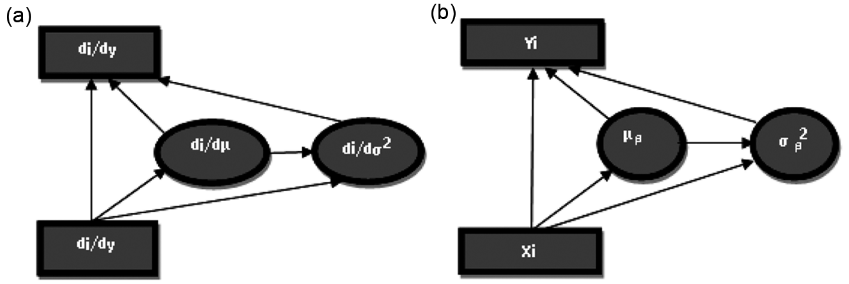

Fig. 1 — Batch normalization algorithm: (a) Forward pass, (b) Backward pass

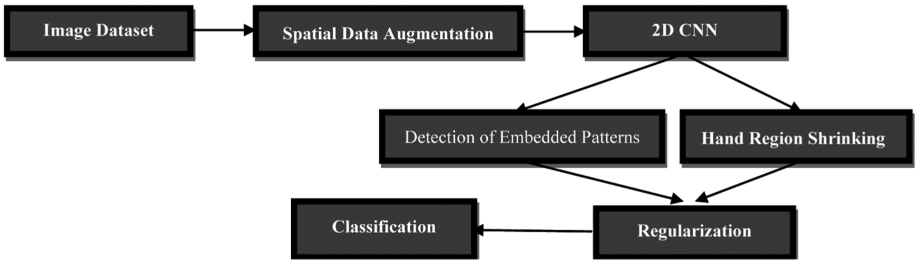

Fig. 2 — Image dataset processing model with 2DD-CNN regularization and interpretation of gestures for simple and complex background image dataset

recognition, and classification task within a single neural network without localization for both static and dynamic hand gestures. This makes the network diverse from all the other CNN networks implements the extraction and classification tasks separately.

Experimental Details

Sign Language Recognition - Static Image Dataset

The proposed system prevents the localization task and minimizes the workload for training the large dataset by using a GPU machine and an inception deep-CNN network.

Model for Sign Language Interpretation

The image dataset needs processing only in 2-dimensions known as the spatial dimensions as shown in Fig. 2 for the proposed system model for processing the image dataset, the 2D convolution network. The static images are augmented using various spatial-augmentation techniques and send to the convolution layer where the detection of embedded patterns and hand region shrinking occurs. Later regularization techniques like dropout[20] and batch normalization layers are added. Finally, classification of the gestures is carried out.

Spatial Augmentation

Spatial augmentation is usually done when the dataset size is small. Usually, Deep-CNN is designed to study and determine very huge datasets, providing good accuracy. Also, overfitting can be avoided by using a larger dataset in the training model, thereby increasing the system performance in the testing phase. Spatial dataset augmentation can be done in two ways.

- Offline dataset augmentation.
- Online dataset augmentation.

Offline Dataset Augmentation

Offline dataset augmentation can be done by performing operations such as scaling, translation rotation, flipping, adding noise, random crops, lighting condition, perspective transformation, reversal mechanism. Data augmentation is done to increase the size of the dataset by flipping, lighting conditions, random crops, and reverse ordering operations. In turn, the system can be trained more efficiently, thus leading to increased learning and good accuracy.

2DD-CNN

A 2D convolution network is capable of learning the image datasets. The designed network depicted in Fig. 3 is capable of learning only the spatial dimensions of the image and hence it cannot be applied for the dynamic video dataset. The network is an inception model which is capable of recognition tasks within single network architecture. The First

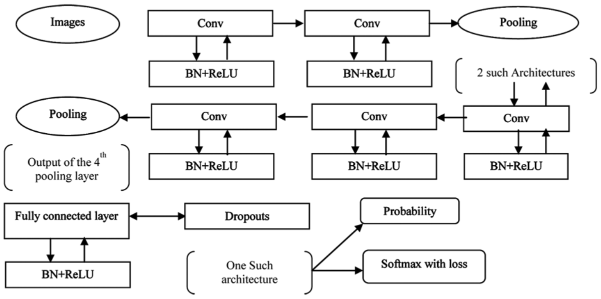

Fig. 3 — 2D-CNN flow takes image input, performs recognition and later classification done by fully-connected layers and output based on probability

layer has a kernel size of 5 × 5 and strides equal to 3 × 3 and the remaining conv layers have a kernel size equal to 3 × 3 and strides equal to 1 × 1. The key idea is to design a CNN network capable of detecting the hands in the image without performing any localization task. Thus, the implemented system consists of an inception deep CNN network that can detect only the hand region from the entire image through the designed convolution layers capable of detecting the embedded patterns in the image, which is the hand region performing the gesture excluding the other hand areas. It is always known that the hand region performing the gesture occupies only 10% of the image, and this part is embedded into the image along with the human body and human hands. The designed CNN architecture (Fig. 3) is adept at learning the spatial features so it can detect the hand gestures without extraction from the image. For the hand region shrinking mechanism to occur, system used 9 conv layers and 4 pooling layers. While performing CNN system might face the problem of overfitting are prevented by adding the dropout mechanism and the flow of the deep CNN for processing the static image dataset is shown in Fig. 3.

Regularization Using Batch-normalization

This is one of the regularization techniques applied to improve the performance of the network. The input is processed as a mini-batch; later the mini-batch variance is taken. Normalization is performed, finally scaling and shifting operations are performed. The batch normalization is used to mitigate the internal covariant shift, regularizes the model, and reduces the need for dropout. In the designed network the output from the conv layer is given as input to the batch normalization layer. The input is a mini-batch that is processed by the BN layer thus it is helpful in the reduction of the processing time. It is also used to increase the accuracy and efficiency of the system. The BN also plays a major role in the multi-class classification in the network. It is applied after the dense and before the activation. The only importance is to reduce the time complexity by processing the output as mini-batch and reducing the internal covariant shift. The mini-batch mean is first calculated where the input is taken as a batch. The input is from the convolution layer and m is limit. The mean μ is calculated using the formulae in Eq. 1.

$$\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \dots (1)$$

After calculating the mean, the variance is determined using the formulae as illustrated in Eq. 2.

$$\sigma_\mu^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \qquad \dots (2)$$

Further normalization is performed using the formulae given in Eq. 3 and the value of $\hat{x}_i$ is found. After that, scale and shift operations are performed as illustrated in Eq. 4, to obtain the batch normalization output.

$$\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} \qquad \dots (3)$$

$$y_i = \gamma\, \hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}\,(x_i) \qquad \dots (4)$$

where, γ and β are the scale and shift values.

The output obtained in y_i as illustrated in Eq. 4, is sent to the next layers of the CNN for further processing.

Sign Language Recognition for Dynamic Video Dataset

The proposed system meant for classifying dynamic gesture is depicted in Fig. 4. For processing, the video dataset system has a 3DD-CNN architecture. The variation with respect to the 2DD-CNN is that the conv layers are removed and instead the Softmax activation introduced. The number of pooling layers remains the same. However, the keyframes are taken from the input video dataset and used in the training phase. The dynamic videos are given as input and later split up into keyframes. The keyframes are processed through the 3D Deep-CNN network followed by detecting embedded patterns & hand region shrinking. Regularization is done using dropout and batch normalization. Finally, classification is based on the probability and interpretations of gestures are carried out.

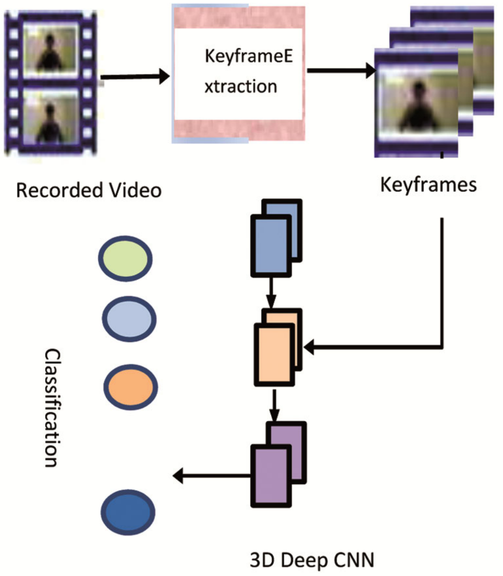

Fig. 4 — Architecture of SLR for dynamic video dataset

Keyframe Extraction

The keyframe extraction is a powerful technique for summary video content, it is categorized into 4 types based on short boundary, visual information, movement analysis, and clustered method. The main idea behind keyframe extraction is to convert the video into one dimensional signal that eases the training and testing phases when processed using the D-CNN network. Pre-processing phase is meant to process the video in such a manner that reduces the computational complexity of the classification models. Keyframe extraction is the key aspect in feature extraction since it extracts those frames responsible for the key movement of hand signs and it eliminates repletion of frames. The keyframe extraction algorithms illustrate the mechanism followed to extract keyframes.

Keyframe Extraction Algorithm

1. Calculate interframe differences (inter-diff) of a video.
2. Calculate the sum of pixels in each frame.
3. Calculate the mean for each inter-diff frame.
4. Convolute the mean array with the ‘Humming Window’ array obtained by using the formula: $w(n) = 0.5-0.5 \times \cos(2\prod n/M-1)\ 0 \leq n \leq M-1$.
5. Calculate the relative local extrema from the obtained array convolution.
6. The respective frame indexes are obtained using local extrema.
7. The keyframes from the video are extracted using the frame indices.

3D Deep CNN

In the 3D CNN network, the first layer is the conv layer which has 3 dimensions (x, y, and z) for spatial and temporal dimensions and is capable of learning the video datasets. To perform the conv operation a kernel size equal to 7 × 7 × 7 with the strides of 3 × 3 × 3 has been used, second conv layer has a kernel size 5 × 5 × 5 with the strides equal to 3 × 3 × 3 and the remaining conv layer with filter size 3 × 3 × 3 with strides equal to 3 × 3 × 3. A CNN has a convolution configuration, limits the neural association amid layers, and is made to allocate the unchanged weights in a layer. BN layer is added after the conv layer 7 conv layers in the 3D CNN network which has the same functioning used in the 2D CNN network (Fig. 3). The Softmax activation function is used to incorporate the two important properties of the same, as calculated values should be in the range of 0 to 1, and the sum of all the probabilities should be equal to 1. ReLU activation takes the values from 0 to ∞ whereas the Softmax activation restricts the required inputs based on the greater probability, thus reducing the workload of the system during processing thus favouring learning. The hand region shrinking and the detection of embedded pattern is done similarly as in the 2DD-CNN model. Gesture recognition system is designed to interpret the sign language signs and the system designed to improve the real-time efficiency trained using a GPU machine, the online available workspace “collaborator” is an open-source GPU-based machine, with a good storage capacity.

Results and Discussion

Dataset Preparation

The dataset for the sign language with the simple background was collected from the project “Sign language and Static gesture recognition using scikit-learn”. The bench mark dataset LSA64 data is used for the dynamic video processing; LSA64 dataset consists of 64 signs and includes both one-handed (R: Right hand) and two-handed signs (B: Both hands).

Static Signs Dataset

The American Sign Language (ASL) dataset with simple backgrounds having high-resolution quality images in which the images had 35% human hand and the rest is background, initially contained only 1500 images for 14 different users posing different signs from A-Z except J and Z or simple background. To increase the number of images in a dataset the spatial augmentation technique is being used; hence, 26880 images were obtained from 14 different users having 1920 images. The dataset for the sign language with complex background contains 10 different ASL signs starting from 0–9. The human hand performing the sign occupies only 10% of the image. It was collected from 14 different users having a total of 1400 images. The size of the dataset is increased by performing the spatial augmentation and obtained 11200 images. It contains 80 images per gesture and 800 images per user.

Dynamic Signs Dataset

Proposed work used the first 30 signs of the LSA64, 1500 videos of 10 signers with 5 repetitions of each gesture. The list of gesture is shown in Table 1, where Name depicts the gesture and H designates only one hand (Right hand-R) is involved in the gesture or both hands (B) are involved.

Results

The dataset is processed on an online available workspace "collaboratory", supports all the python and Open CV library packages and works on GPU. The memory is available for processing 12 GB and 3 GB RAM. Static datasets were processed by loading each dataset separately, video dataset is split into 3 parts of 2.8 GB each. Initially the first part is being used to train the system and the model is saved, second batch of data is then used to train the system with the saved model.

This is done 3 times and the accuracy obtained is 84% wherein testing produced the correct interpretation (80%) of gestures accurately. Accuracy obtained in the designed 2DD-CNN and 3DD CNN networks is illustrated in Fig. 5 and Fig. 6, respectively. How the designed network works better than other benchmark networks for the static, complex background datasets can be observed from

Table 1 — List of gesture for the dynamic gesture recognition

ID	Name	H	ID	Name	H	ID	Name	H	ID	Name	H
01	Opaque	R	09	Women	R	17	Call	R	24	Argentina	R
02	Red	R	10	Enemy	R	18	Skimmer	R	25	Uruguay	R
03	Green	R	11	Son	R	19	Bitter	R	26	Country	R
04	Yellow	R	12	Man	R	20	Sweet milk	R	27	Last name	R
05	Bright	R	13	Away	R	21	Milk	R	28	Where	R
06	Light-Blue	R	14	Drawer	R	22	Water	R	29	Mock	B
07	Colors	R	15	Born	R	23	Food	R	30	Birthday	R
08	Photo	B	16	Learn	R						

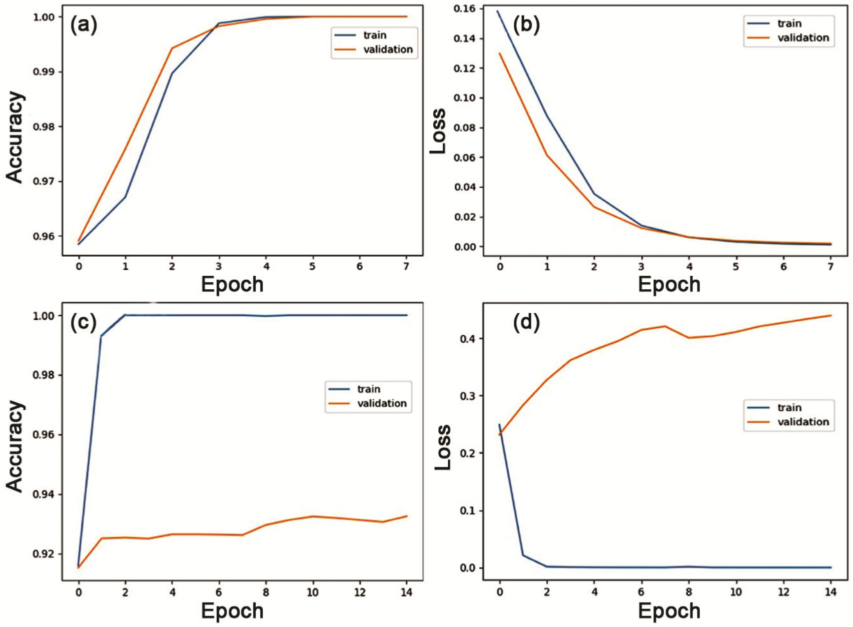

Fig. 5 — Accuracy and loss for simple and complex background of static 2DD-CNN: (a) Accuracy (simple background), (b) Loss (simple background), (c) Accuracy (complex background), (d) Loss (complex background)

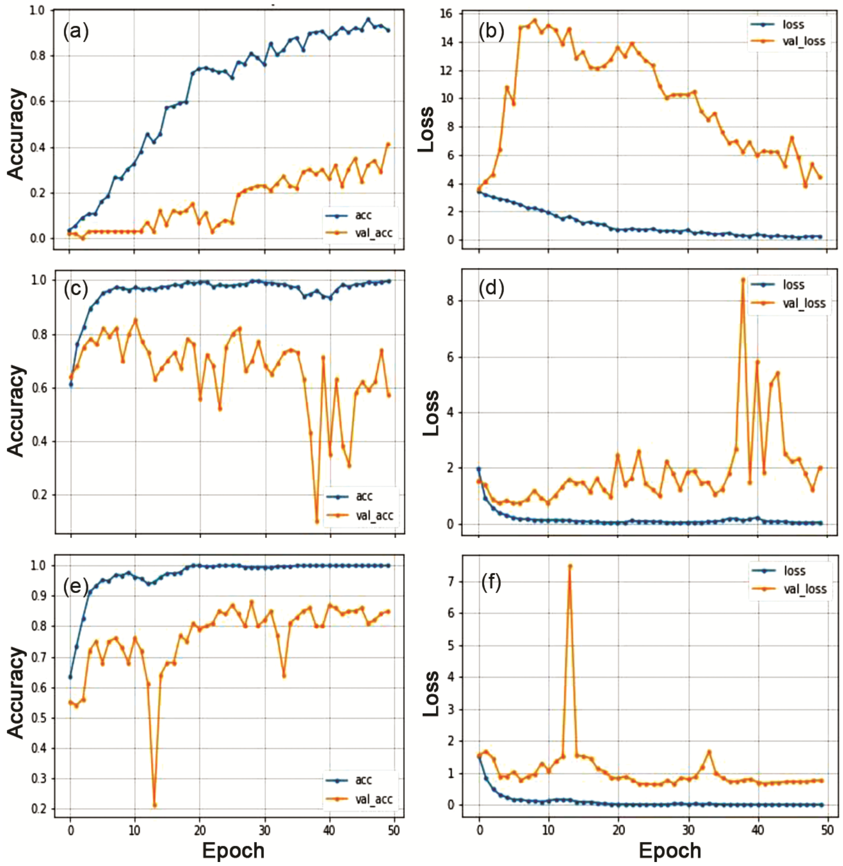

Fig. 6 — Results of dynamic video dataset processing for different number of videos: (a) Accuracy (500), (b) Loss (500), (c) Accuracy (1000), (d) Loss (1000), (e) Accuracy (1500), (f) Loss (1500)

Table 2 — Comparison study of various systems and the proposed network for static, dynamic datasets

Types	Network	Accuracy	Precision	Recall	F1 Score
Dynamic Dataset	3D-CNN	0.838	0.894	0.843	0.840
	AlexNet	0.782	0.776	0.784	0.780
	VGG19	0.738	0.740	0.754	0.747
Static Simple Dataset	2D-CNN	0.992	0.991	0.990	0.991
	AlexNet	0.959	0.961	0.957	0.957
	VGG19	0.968	0.969	0.968	0.961
Static Complex Dataset	2D-CNN	0.919	0.903	0.919	0.911
	AlexNet	0.909	0.892	0.909	0.900
	VGG19	0.920	0.903	0.920	0.912

Table 2. Key frame extraction is a pre-processing phase which reduces the computational complexity of the classification model. Local maxima key extraction methods extract the frames which are responsible for the hand movements and eliminates the repetition of frames. 3DD-CNN network initially trained for the key, frames first 500 videos and appended with the learning of 1000 videos finally considers all the 1500 videos. The system converges to a better accuracy as the batch is increased. Since the proposed work has used batch normalization, regularization technique, both are also applied in the AlexNet and VGG-19 architectures for comparison.

Interpretation

The classification in the CNN is done based on the probability by the fully connected layer which does the multi-class classification and the gestures are separated into classes and after training they are interpreted. The interpretation is in text format. This makes the normal user understand the gestures performed by the deaf-mute people. The interpretation

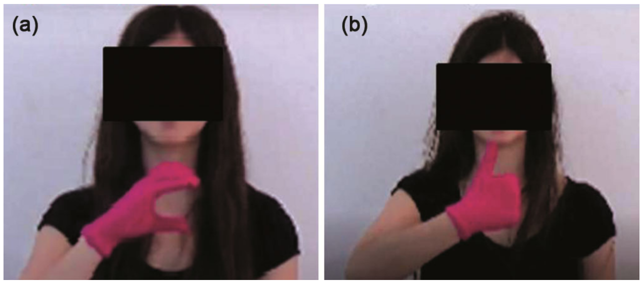

Fig. 7 — The interpretation of the gesture in the form of text after classification based on the probability during training: (a) Opaque; (b) Water.

of the gestures in the form of text output is given in Fig. 7 and Table 2 presents the performance metrics of the proposed network of 2D-CNN and 3D-CNN with AlexNet, VGG19. The designed network works better than other benchmarks networks for the static, complex, and video dataset. There is a significant increase in the accuracy while detecting static signs and dynamic gestures using proposed method. Evaluation of these performance metrics for the CNN model indicates that the proposed method outperforms with the other two networks being in concern for both Static Dataset and Dynamic Dataset.

Conclusions

The proposed methodology is a stimulating technique for gesture recognition of sign language in simple background static image dataset, complex background static image dataset, and dynamic video dataset with an accuracy of 99%, 92%, and 84% respectively. The static signs are interpreted with 2D-CNN with the augmentation to overcome overfitting and the 3D-CNN recognizes the dynamic gesture after performing the local maxima key frame extraction, which is crucial for the recognition of signs. The system performance is further improved by adding batch normalization layer in both 2D-CNN and 3D-CNN. The proposed system is capable of recognizing and interpreting the gestures in both images and videos efficiently, while its limitation is to handle only A-Z of static signs except J and Z. Additionally, it could be deployed as a web application, where the client performs the signs and requests the server to predict the actual recognition of the sign in the form of edge computing.

References

1 Bhatt R, Fernandes N & Dhage A, Vision based hand gesture recognition for human computer interaction, *Int J Innov Sci Eng Technol*, **2** (2013) 110–114.

2 Rautaray S S & Agrawal A, Vision based hand gesture recognition for human computer interaction: A survey, *Artif Intell Rev*, **43** (2015) 1–54.

3 Starner T E, *Visual Recognition of American Sign Language Using Hidden Markov Models*, MS dissertation, Massachusetts Institute of Technology, USA, 1995.

4 Anjo M D S, Pizzolato E B & Feuerstack S, A real-time system to recognize static gestures of Brazilian sign language (libras) alphabet using Kinect, Br*azilian Symp on Human Factors in Computer Systems* (Brazil) 2012, 259–268.

5 Huang J, Zhou W, Li H & Li W, Sign language recognition using 3D convolutional neural networks, *Proc IEEE Int Conf Multimedia Expo* (Torino, Italy) 2015, 1–6.

6 Masood S, Srivastava A, Thuwal H C & Ahmad M, Real-time sign language gesture (word) recognition from video sequences using CNN and RNN, *Proc Int Conf Front Intell Comput; Theory Appl* (Odisha, India) 2018, 623–632.

7 Joys J, Balakrishnan K & Sreeraj M, Sign quiz: A quiz based tool for earning finger spelled signs in Indian sign language using ASLR, *IEEE Access*, **7** (2019) 28363–28371.

8 Lee D & Park Y, Vision-based remote control system by motion detection and open finger counting, *IEEE Trans Consum Electron*, **55** (2009) 2308–2313.

9 Lamberti L & Camastra F, Real-time hand gesture recognition using a color glove, *Proc Int Conf Image Analysis & Process, (ICIAP)* (Ravenna, Italy) 2011, 365–373.

10 Erden F & Çetin A E, Hand gesture based remote control system using infrared sensors and a camera, *IEEE Trans Consum Electron*, **60** (2014) 2308–2313.

11 Wang Y & Yang R, Real-time hand posture recognition based on hand dominant line using Kinect, *Proc IEEE Int Conf Multimed Expo Worksh (ICMEW)* (San Jose, CA, USA) 2013, 1–4.

12 Mishra S R, Krishna D, Sanyal G & Sarkar A, A feature weighting technique on SVM for human action recognition, *J Sci Ind Res*, **79** (2020) 626–630.

13 Chen Z H, Kim J T, Liang J, Zhang J & Yuan Y B, Real-time hand gesture recognition using finger segmentation, *Sci World J*, **2014** (2014) 2456–2459.

14 Gokgoz K, *The Nature of Object Marking in ASL*, Ph.D Thesis, Purdue University, United States, 2013.

15 Girshick R, Donahue J, Darrell T & Malik J, Region based convolutional networks for accurate object detection and segmentation, *IEEE Trans Pattern Anal MachIntell*, **38** (2015) 142–158.

16 Mahajan P, Abrol P & Lehana P K, Scene based classification of aerial images using convolution neural networks, *J Sci Ind Res*, **79** (2020) 1087–1094.

17 Kopuklu O, Gunduz A, Kose N & Rigoll G, Real-time hand gesture detection and classification using convolutional neural networks, *Proc Int Conf Automatic Face Gesture Recognit* (Lille, France) 2019, 1–8.

18 Molchanov P, Gupta S, Kim K & Kautz J, Hand gesture recognition with 3D convolutional neural networks, *IEEE Int Conf Comput Vis Pattern Recognit Worksh (ICCVPR)* 2015, 1–7.

19 Ji S, Xu W, Yang M & Yu K, 3D convolutional neural networks for human action recognition, *IEEE Trans Pattern Anal Mach Intell*, **35** (2012) 221–231.

20 Srivastava N, Hinton G, Krizhevsky A, Sutskever I & Salakhutdinov R, Dropout: A simple way to prevent neural

networks from overfitting, *J Mach Learn Res*, **15** (2014) 1929–1958.

21 Kishore P V V, Anil Kumar D, Chandra Sekhara Sastry A S & Kumar K, Motionlets matching with adaptive kernal for 3-D Indian sign language recognition, *IEEE Sens J*, **18** (2018) 3327–3337.

22 Pan J, Luo Y, Li Y, Khong C, Chun-Huat T, Aaron H & Thean V-Y, Wireless multi-channel capacitive sensor system for efficient glove-based gesture recognition with AI at the edge, *IEEE Trans Circuits Syst*, **67** (2020) 1624–1628.

23 Bao P, Maqueda A I, Del-Blanco C R & Garcia N, Tiny hand gesture recognition without localization via a deep convolutional network, *IEEE Trans Consum Electron*, **63** (2017) 251–257.

24 Kanchana P, Kosin C & Jing-Ming G, Signer independence finger alphabet recognition using discrete wavelet transform and area level run lengths, *J Vis Commun Image Represent*, **38** (2016) 658–677.

25 Nandy A, Prasad J S, Mondal S, Chakraborty P & Nandi G C, Recognition of Isolated Indian Sign Language Gesture in Real Time, *J Commun Comput Inf Sci*, **70** (2010) 102–107.

26 Li Y & Zhang P, Static hand gesture recognition based on hierarchical decision and classification of finger features, *J Sci Prog*, **105(1)** (2022) 163–170.

27 Gupta S, Jaafar J & Ahmad W F W, Static hand gesture recognition using local Gabor filter, Procedia Engineering, *Int Symp Robot Intell Sensors* (Kuching, Sarawak, Malaysia) 2012, 827–832.

28 Naveed M, Quratulain Q & Shaukat A, Comparison of GLCM based hand gesture recognition systems using multiple classifiers, *Proc IEEE Int Conf Robot Autom*2 (Xi'an, China) 2021, 1–5.

29 Ghosh D K & Ari S, "Static hand gesture recognition using mixture of features and SVM classifier, *5th Int Conf Commun Syst Netw* (Gwalior, MP, India) 2015, 1094–1099.

30 Lim K M, Tan A W C & Tan S C, Block based histogram of optical flow for isolated sign language recognition, *J Vis Commun Image Represent*, **40** (2016) 538–545.

31 Nielson M, *Neural Networks and Deep Learning* (Determination Press, San Francisco, CA, USA) 2015.

32 Pan W, Zhang X & Zhongfu Ye, Attention-based sign language recognition network utilizing key frame sampling and skeletal features, *IEEE Access*, **8** (2020) 215592–215602