Expert Systems With Applications 182 (2021) 115657

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Vision-based hand gesture recognition using deep learning for the interpretation of sign language

Sakshi Sharma [*,1], Sukhwinder Singh

ECE Department, Punjab Engineering College (Deemed to be University), Chandigarh, India

ARTICLE INFO

Keywords:
Hand gesture recognition
Human-computer interaction
Feature extraction
Classification
Deep learning
Sign language recognition

ABSTRACT

Hand gestures have been the key component of communication since the beginning of an era. The hand gestures are the foundation of sign language, which is a visual form of communication. In this paper, a deep learning based convolutional neural network (CNN) model is specifically designed for the recognition of gesture-based sign language. This model has a compact representation that achieves better classification accuracy with a fewer number of model parameters over the other existing architectures of CNN. In order to evaluate the efficacy of this model, VGG-11 and VGG-16 have also been trained and tested in this work. To evaluate the performance, 2 datasets have been considered. First, in this work, a large collection of Indian sign language (ISL) gestures consisting of 2150 images is collected using RGB camera, and second, a publicly available American sign language (ASL) dataset is used. The highest accuracy of 99.96% and 100% is obtained by the proposed model for ISL and ASL datasets respectively. The performance of the proposed system, VGG-11, and VGG-16 are experimentally evaluated and compared with the existing state-of-art approaches. In addition to accuracy, other efficiency indices have been also used to ascertain the robustness of the proposed work. The findings indicate that the proposed model outperforms the existing techniques as it has the potential to classify maximum gestures with a minimal rate of error. The model is also tested with the augmented data and is found as invariant to rotation and scaling transformation.

1. Introduction

With the advancement in computer technology and with improvement in hardware equipment, human–computer interaction (HCI) has become a regular part of our lives. In this HCI, the use of hand gestures has gained people's interest as it is a friendly way of interacting with the machine (Just, 2006; Wu et al., 2016). Hand gesture is a way of expressing your idea and thoughts with the use of palm and finger movement (Wu & Huang, 1999). In HCI, one can interact with machines directly through hand movement, without the need for any other input device. For this, a gesture performed by a person needs to be recognized by a machine for meaningful communication between a person and a machine. Hence, hand gesture recognition (HGR) is a hot research topic nowadays and is a need of today's era. It has many different applications such as virtual Reality (Sagayam & Hemanth, 2017), robot control (Tao et al., 2013), virtual games (Kulshreshth et al., 2017), and natural user interfaces (Ng et al., 2011). A major application of hand gesture recognition is in the human communication system i.e. in sign language recognition (Lichtenauer et al., 2008). Sign language is the visual form of language which uses the structured form of communicative hand gestures to express ideas (Sharma & Singh, 2020). For the speech and hearing impaired people this is the only way of communication. According to the World health organization (WHO), 5% (approx. 360 million) of the total population of the world is suffering from either medium or severe hearing loss and they can communicate only in their respective zonal sign language (World health organization (WHO), 2015). This language is not easily understandable by the hearing community and hence there remains a communication gap between hearing and speech-hearing impaired communities. Thus, gesture recognition using computer technology can be used as an interpreter for sign language translation. This would be advantageous and serve as a link bridge between these communities.

Hand gestures used in sign language can be of two types i.e. static and dynamic gestures. Static gestures are defined as the position of hands and fingers in the space without any movement w.r.t time, whereas in dynamic gestures there is a continuous movement of hands

* Corresponding author.
E-mail addresses: sak.sharma92@gmail.com (S. Sharma), sukhwindersingh@pec.ac.in (S. Singh).
[1] ORCID: 0000-0002-0291-3271.

https://doi.org/10.1016/j.eswa.2021.115657
Received 18 February 2021; Received in revised form 3 June 2021; Accepted 22 July 2021
Available online 27 July 2021

w.r.t time. The process of recognition of hand gestures for the sign language translation can be done in two different ways namely: vision-based (Moeslund et al., 2006) and contact based (Kumar et al., 2012) recognition. In the contact-based process, signer needs to wear electronic circuitry, for e.g.: data gloves, accelerometer, band, and many such devices. These components measure the change in the movement and transfer the details to the computer for the further processing. This approach has given good recognition result in the literature (Cheok et al., 2019) but it is expensive and inconvenient for the users for daily human–computer interaction interface. The vision-based method is more user-friendly as the signer's data is acquired using a camera. This method reduces the dependency of users on sensory devices and is based on image processing algorithms to process the acquired data. In this paper, a vision-based method has been proposed for the recognition of static hand gestures for sign language translation system. For this, a novel and robust technique based on convolutional neural network (CNN) have been proposed. Along with this proposed method, VGG-11 and VGG-16 have been also modified and implemented for gesture recognition. A broad dataset of static gestures of Indian sign language (ISL) has been collected indigenously in this work for the performance assessment of these approaches, as there is no publicly available dataset of it. This proposed model is found to be advantageous over the present state-of-art approaches since it has high accuracy and consumes less training time. The major contribution and novelty of the paper are as follows:

- A novel and robust model named as G-CNN has been proposed for the hand gesture recognition.
- A dataset consisting of 43 classes of ISL has been collected in this work
- All the hyper-parameters like kernel width, epochs, batch-size, learning-rate are empirically finely-tuned for efficient training of model.
- A thorough experimental analysis of the proposed work has been done by using various evaluation metrics like: accuracy, loss, recognition accuracy of each class, and training time consumed by the model. The results have been also evaluated and compared with VGG-11 and VGG-16 using same dataset.
- Generalization ability of the model has been also proved in this work by evaluating the performance on the augmented dataset. The model shows the competent results and is found to be robust as it is invariant to rotation and scaling transformation.
- The performance of the proposed work has been also evaluated using 10-fold cross validation.
- The work proposed in this model has achieved the remarkable results on each evaluation metrics. Hence it is evident that, this model can easily handle the complexity and hand occlusion of ISL signs.

The remainder of this paper is organized as follows: in Section 2 brief literature review of gesture recognition is presented. In Section 3, the proposed methodology and detail of the collected dataset are presented. Section 4 presents the experimental data and analysis. Finally, the conclusion of this work is given in Section 5.

2. Related work

In the human–computer interface, different type of methodologies are available in the literature for the gesture recognition. The primary aim of these methods is to ease the communication by deducing the correct meaning of gestures/signs performed by the user. This process includes the following step namely: acquisition and pre-processing, gesture representation, feature extraction, and classification. This section investigates the various gesture recognition technique in the context of different sign languages. A brief review of these techniques is discussed below:

ASL Akhter (2018), has presented a method for the recognition of ASL alphabets. In this PCA based features along with gabor filter and orientation based hash code is used to represent the different alphabets of ASL. Then these extracted features are classified using artificial neural network (ANN). In this paper, the performance has been evaluated on their self-created dataset of 24 static gestures. Kang et al. (2015) developed a CNN-based model for the recognition of human gestures. For this, the model is trained and tested for 31 different classes consisting of alphabets and numbers of ASL. In another method of ASL alphabet recognition by Ameen and Vadera (2017), both color and depth images of the gestures are fed into CNN model. Two convolutional layers are used in this model to extract features from each input and then the features from these layers are concatenated together and fed for classification into a fully connected layer. Apart from RGB images, many researchers have worked with depth sensor like Microsoft Kinect. Tao et al. (2018) presented a method for the recognition of sign language using CNN with multiview augmentation and inference fusion. Microsoft Kinect camera has been used to collect the depth images of the gestures. In this paper, the authors suggested the use of augmented data for the training of CNN model. This method can achieve good recognition accuracy but at the cost of high computational requirements. Another Kinect sensor based hand gesture recognition has been observed in literature for ISL recognition (Ansari & Harit, 2016). A study has been carried out in this work by using a different combination of feature extraction and machine learning algorithms for the accurate recognition of hand gestures. The performance has been evaluated on the different set of classes from a total of 140 static words. Another method using a depth sensor is proposed by Aly et al. (2019) for the recognition of ASL fingerspelling. In this paper, the authors incorporated principle component analysis network (PCANet) to extract and learn the features from the depth images. Finally, a linear SVM (Support vector machine) is employed for the classification of 24 static gestures of ASL. Gangrade & Bharti (2020) proposed an approach for the recognition of ISL by using 5-layer model of CNN. The dataset has been acquired in this paper by using Microsoft Kinect sensor from 12 different signers. The performance of this method has been tested for the numbers and alphabets of ISL only.

Contact based approaches for gesture recognition has been also adopted by many researchers in the literature. Chong and Kim (2020) presented an approach for the identification of ASL using a wearable unit. In this work, 6 inertial measurable units (IMU) were used to obtain 28 ASL words, and the classification was carried out using the LSTM algorithm. A gesture recognition technique based on recurrent neural network (RNN) for Chinese sign language translation system is described in Xiao et al. (2020). In this work skeleton sequence of the signer is utilized for bidirectional communication. The performance of this method is tested on standard RGB-depth images of different static gestures. Abraham et al. (2019) introduced a sensor-based real-time hand gesture recognition system for the translation of ISL. In this paper, hand orientation and finger movements have been extracted from the sensor's data and then transmitted wirelessly to the processing device. Finally, an LSTM network is employed for the classification. This model is tested on 26 commonly used gestures of ISL. Gupta and Kumar (2020) proposed a new sensor-based technique for the recognition of ISL. Electromyograms and IMU were placed on both forearms of signers to collect the detail of signs. This system was classified using a multi-label classification based on the lexical attributes of signs and it achieved an error of 2.73%. Kakoty and Sharma (2018) have presented a contact-based method for the recognition of alphabets and numbers of ISL and ASL. After gathering sign information using data gloves; finger and wrist joint angles are extracted and pre-processed using a moving average filter. Classification is carried out using SVM with 10-fold cross-validation and an accuracy of 96.7% has been obtained.

A vision-based approach for hand gesture recognition has been proposed by Shrenika and Bala (2020). For feature extraction, grey-scale transformation and edge detection are used in this method. The

template-matching algorithm is then used to recognize the gestures and, their corresponding text is displayed on the screen. The performance of this approach is tested on the finger-spelled gestures of numeric and alphabets. In (Joshi & Gaur, 2018), a method for the recognition of ISL has been presented. This method proposes the use of the HOG descriptor to represent the feature vector of gestures and the classification is done using SVM. The overall performance of the method is tested on ISL and ASL alphabets and achieved an overall accuracy of 95%. Another recognition of ISL was proposed by Mariappan and Gomathi (2019) using a fuzzy c-means clustering machine learning algorithm. Another approach for the ISL translation system was proposed by Kaur et al. (2017). In this, authors studied the effect of orthogonal moment based local features on the classification of ISL. From the experimental analysis it is concluded that these features are rotation, scale, translation and, user-independent and also gave good recognition accuracy for the ISL dataset. Another approach for ISL recognition has been presented by Kumar and Kumar (2021). In this paper, the ISL database of 26 alphabets was created with 12 different signers by taking one sample from each. The recognition of signs has been done with traditional machine learning method by using HOG features and extreme learning machine training method. Athira et al. (2019) proposes a method for recognizing ISL gestures using traditional machine learning method. In this paper, gestures information has been extracted from the video using YCbCr skin segmentation color model. Then the key feature extraction and classification has been done with Zernike moments and SVM.

Xie et al. (2018) presented an approach for static gesture recognition using Inception V3. The static image dataset by Pugeault and Bowden (2011) of 24 English letter ASL has been used in this paper for the performance evaluation. For classification by Inception v3, a 2-stage training strategy has been used for the fine-tuning of the model and it achieved an accuracy of 91.35%. In (He, 2019) the authors presented an algorithm for SLR based on neural network. The complete recognition system consists of a hand locating network based on faster R-CNN and 3D CNN for the feature extraction, and LSTM for encoding and decoding. This method achieves good recognition accuracy but the data set is limited in scope. Wadhawan and Kumar (2020) proposed a framework for SLR system based on the deep learning model. For this CNN based architecture was used for the classification of the signs. A web-camera based dataset consisting of alphabets, digits, and words of ISL was used in this paper for the performance evaluation. A comparison of the efficacy of this system was done by using different optimizers in the CNN architecture. A brief comparison among the contact-based, Kinect-based and RB-images based hand gesture recognition is given in Table 1. *From the literature review presented in this section, the following interpretations are observed:*

- The format of gestures of ISL is more complicated as compared to other popularly used sign languages. Thus the adoption of an existing method of gesture recognition will not give the same results for ISL.
- Because of the complicated structure of ISL, the gesture recognition technique for the recognition of ISL has received much less attention.
- No publicly dataset of ISL is available. The ISL dataset used in gesture recognition methods in the literature was collected by their respective authors only. These are limited in size and acquired in limited background conditions.
- Feature extraction is a process of transforming the important information of the input data into compact feature vector. Traditional feature extraction techniques (like Shift invariant feature transform (SIFT), Principle component analysis (PCA), histogram of oriented gradient (HOG), Local binary pattern (LBP), etc.) which are used with machine learning models requires mathematical operators and manual observation for the key-feature extraction. These mathematical operations are complex in nature. Brief comparison of existing feature extraction methods is given in Table 2. It is clear from the table that the recognition accuracy obtained in the literature for limited classes of ISL are not so adequate On the contrary, the

Table 1
Brief comparison among different type of hand gesture recognition techniques.

S. No.	Authors	Year	Method of acquiring data	Features and classifiers	Dataset used
1	Tao et al.	2018	Kinect sensor	CNN	24 static gesture of ASL
2	Aly et al.	2019	Kinect Sensor	Principle component analysis network (PCANet) and SVM	24 static gesture of ASL
3	Gangrade &bharti	2020	Kinect Sensor	CNN	36 static gestures of ISL
4	Chong & Kim	2020	Contact based	Long short term memory (LSTM)	28 static gestures of ASL
5	Gupta &Kumar	2020	Contact based	Multi-label classification (MLC)	100 isolated signs of ISL
6	Abharam	2019	Contact based	LSTM	26 gestures of ISL
7	Kakoty & Sharma	2018	Contact based	SVM	8 signs of ISL 36 signs of ASL
8	Kaur et al.	2017	Vision based using RGB images	Dual-Hahn and Krawtchouk moments with 4 different types of classifiers	26 signs of ISL Jochen-Triesch's dataset of ASL
9	Joshi et al.	2018	Vision based using RGB images	Histogram of oriented Gradients, SVM	26 signs of ISL Jochen-Triesch's dataset of ASL
10	Xiao	2020	Vision based using RGB images	Recurrent Neural Network (RNN)	Chinese Sign Language
11	Shrenika & Bala	2020	Vision based using RGB images	Edge detection, Template-matching algorithm	36 signs of ASL
12	Kumar & Kumar	2021	Vision based using RGB images	(HOG), Extreme learning machine	26 signs of ISL

Table 2
Comparison of traditional way of feature extraction techniques used for ISL recognition.

S. No.	Authors	Year	Feature extraction and classification method	No. of signs of ISL	Accuracy (%)
1	Rekha et al.	2011	Principle curvature based Region detector , wavelet packet decomposition, SVM	23	91.3
2	Raheja et al., 2016	2016	Hu-moments, motion trajectory, SVM	4	97
3	Kaur and Joshi	2016	Hu moment, SVM	10	70
			Zernike Moment, SVM		76
			Krawtchouk moment, SVM		83.5
5	(Pathak and Jalal, 2019)	2019	Motion direction code, SVM 22	22	90.4
5	Athira	2019	YCbCr skin segmentation, Zernike moments, SVM	24	90.1
6	Kumar and Kumar	2020	HOG, extreme learning machine	26	80.76

feature extraction with deep learning is automatic. With each successive layer of neural networks, model automatically learn and extract the key features from the input data. This, automatic extraction of the features using deep learning is advantageous over the feature extraction algorithms (Mittal et al., 2018).

Based on these observations, a new approach has been proposed in this paper to address all the above-mentioned issues of hand gesture recognition for sign language translation.

3. Gesture recognition method

The mcethodology for the recognition of static hand gesture is summarized in Fig. 1. In this paper, firstly dataset of static gestures has been collected from multiple signers, and then images are pre-processed in order to accurately extract the gesture detail. Next, images have been fed into the proposed model of CNN for feature learning and the classification has been performed using Softmax classifier. The formation of an appropriate dataset is discussed in Section 3.1. In Section 3.2, the detail of the proposed CNN architecture and its training detail is given.

3.1. Dataset preparation

3.1.1. Dataset-I

The major problem in the field of hand gesture recognition for Indian sign language is a lack of publicly available datasets. It is also evident from the literature that authors have developed their dataset for ISL, but the number of images and classes used has been much smaller. Thus to overcome this problem, a large dataset from multiple signers under different light and background conditions (uniform and complex) has been collected in this work.

A) Capturing sign gestures images

Data collection is part of this work and is an important step to maintain the integrity of the research. Before capturing this dataset, a thorough study of Indian sign language has been carried out and then the dataset has been collected for this research work. This dataset consists of RGB-images of 43 classes of ISL performed by 50 different people, resulting in a total of 2150 gesture images. This dataset is categorized into two subsets as shown in Fig. 2. Each gesture has been acquired from multiple signers in different environments and background conditions; ensuring the natural intra-class variation for a better generalization of proposed work.

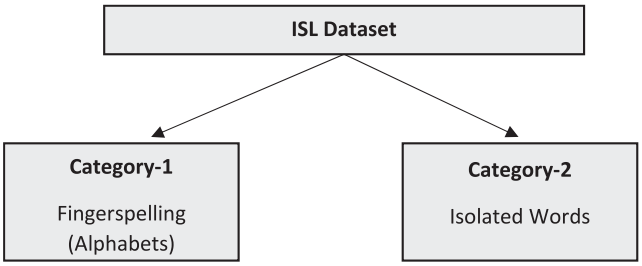

Fig. 2. ISL Dataset categorization.

B) Data pre-processing

Data pre-processing is an important task in the gesture recognition process and it has been used in this paper to improve the quality of the dataset. The collected sign gesture in the dataset was of different sizes and very high resolution, and this different nature of the dataset can affect the speed and efficiency of the classifier. So after collecting the dataset, only the signing gestures have been cropped from each image. Then in order to make the dataset usable for the machine learning model, each image is spatially down-sampled to the size of 256 × 256. This reduced image's size and resolution, minimize the computation complexity, and helps in the faster convergence of the classifier.

C) Data labelling

Data labelling is a crucial part of dataset preparation, particularly for supervised learning algorithms. It is the procedure of tagging data samples with meaningful tags, which provides a learning basis to the classifier. In this paper, collected sign images have been categorized into 43 different classes, and images of these classes have been placed in their respective different folders. Hence, the labelling of data has been done according to their class name. The sample of this dataset is shown in Fig. 3.

3.1.2. Dataset-II

To demonstrate the feasibility and validation of the proposed model

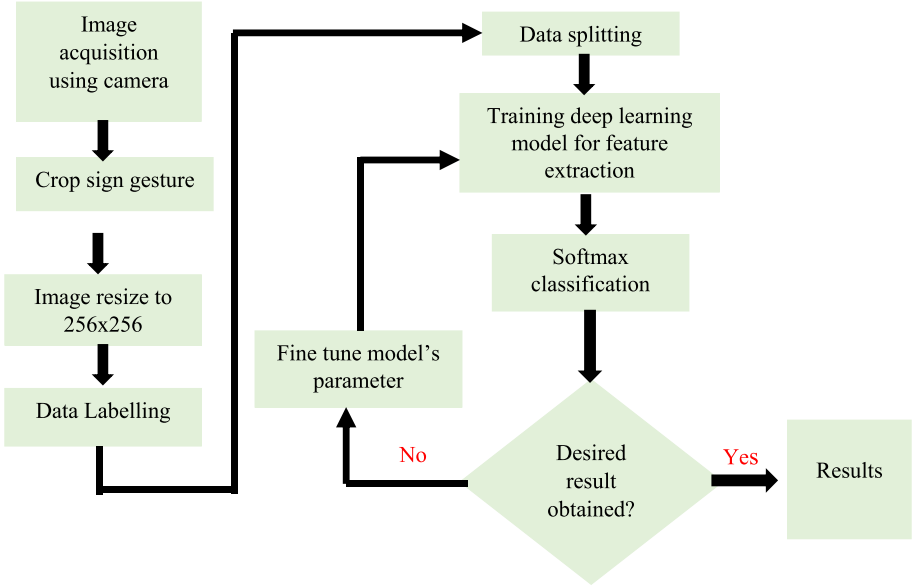

Fig. 1. Framework of the proposed hand gesture recognition (HGR) system.

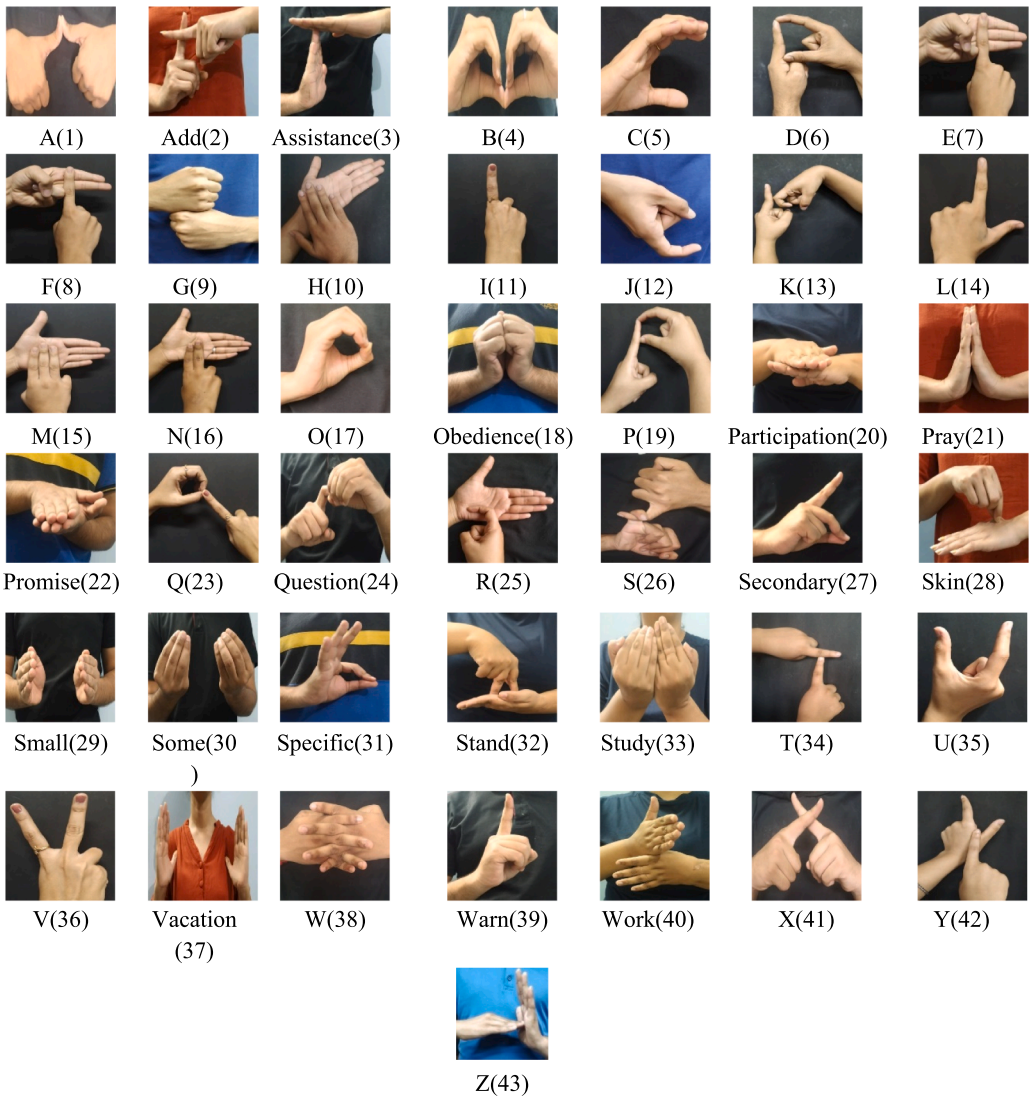

Fig. 3. Sample images of Dataset-I.

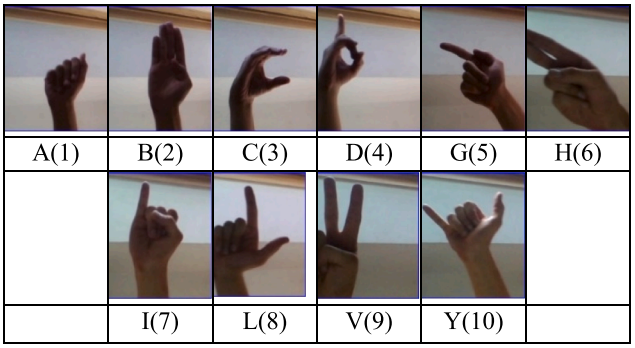

Fig. 4. Sample images of Dataset-II.

with the existing state-of-art approaches, the standard ASL benchmark dataset has also been used in this paper. This dataset known as Jochen-Triesch's dataset consists of hand gestures of ASL and was developed by Frankfurt University (Triesch & Von Der Malsburg, 2001). The uniform background dataset consists 3000 samples for each 10 different classes in light and dark background. The sample of this dataset is shown in Fig. 4.

3.2. Feature learning based on CNN models

Convolutional neural network (CNN) is a feed-forward artificial neural network (ANN) which consist of multiple neural network layer with multiple neurons in each layer (Lecun et al., 1998). A CNN model consists of different arrangements of a convolutional layer, pooling layer, fully connected layer, and softmax layer (Arefnezhad et al., 2020). In this, a convolutional layer is used to extract features directly from

images by sliding a filter along with the input image. Convolutional layer is generally followed by a pooling layer, which reduces and summarizes the obtained features. A fully connected layer is used to classify the features extracted by convolutional/pooling layer into a label. Finally, a softmax layer is used for classification that predicts the class based on the probability distribution. The architecture of the proposed CNN model is discussed in the next section.

3.2.1. Proposed system architecture

In this study, a CNN based model has been specifically designed for gesture-based sign language recognition. This model is proven to be useful (from experimental analysis discussed in Section 4), as it is simple in structure, energy and resource efficient, and requires lesser computational time. In this work, the proposed model is termed as Gesture-CNN (G-CNN) and it consists of 4 convolutional layers, 3 pooling layers, 2 dropout layers, 2 fully-connected layers, and 1 softmax layer, with 12 layers in total, as shown in Fig. 5. In the weighted layer, a small filter size of 3, 2, and 1 has been employed instead of other architecture of CNN which is based on large filter size.

The processing of input gesture images of size [256 × 256] begins with the convolutional layer which extracts features by sliding a filter window over an input image. The weights of these filters are learned and updated automatically in parallel to the extraction of features from an input image. In this layer, 32 convolutional filters with the dimension of [3 × 3 × 32] have been used. As a result, 32 high-level features represented by [256 × 256 × 32] dimension are extracted. To learn the non-linear decision boundaries, this convolutional layer is followed by a non-linear activation function *hyperbolic tangent*. As the proposed architecture of CNN is not so deeper and hence the calculation load of *tanh* has not affected the system's efficiency. The use of this function has also leads to the faster training process of the model (training time of the model is given in result section). Hence, the use of *tanh* function is found as advantageous in this work. The role of activation function is also explained in Algorithm 1. The activation function *tanh* can be expressed using Eq. (1).

$$f(x) = \frac{1 - exp^{-2x}}{1 + exp^{-2x}} \tag{1}$$

The size of the resulting features map is further scaled downed by the factor of 2 using max-pooling operation. Similar to this, other sets of convolutional and max-pooling layers are stacked over this to generate the spatiotemporal representation of the gestures. A total of 4 convolutional layers with a stride of 1 and *tanh* activation function are applied. Kernel sizes used in each convolutional layer are 3, 3, 1, and 3 with the convolutional depth of 32, 64, 64, and 128 respectively, as they are placed in the model. The small kernel size has been used in this model to learn the small texture of the signs. For pooling operation, max-pooling has been used to reduce the size of feature maps by using a filter size of 2 with a stride of 2. After this, a set of the fully connected layer is used to link all previously extracted features for the classification. The number of hidden units used in 2 fully connected layers are 512 and 84. During the training of this model, two dropout layer with 0.3 and 0.2

Table 3
Configuration of proposed CNN (G-CNN).

Layer Type	No. of Filter	Feature Map Size	Kernel size	Stride used
Input Image layer	–	256 × 256	–	
Convolution 1	32	256 × 256 × 32	3 × 3	1 × 1
Max-pooling 1	1	128 × 128 × 32	2 × 2	2 × 2
Convolution 2	64	128 × 128 × 64	3 × 3	1 × 1
Convolution 3	64	128 × 128 × 64	1 × 1	1 × 1
Max-pooling 2	1	64 × 64 × 64	2 × 2	2 × 2
Convolution 4	128	64 × 64 × 128	3 × 3	1 × 1
Max-pooling 3	1	32 × 32 × 128	2 × 2	2 × 2
Dropout 1				
Fully connected 1		512 × 1		
Fully connected 2		84 × 1		
Dropout 2				
Output Layer		43 × 1		

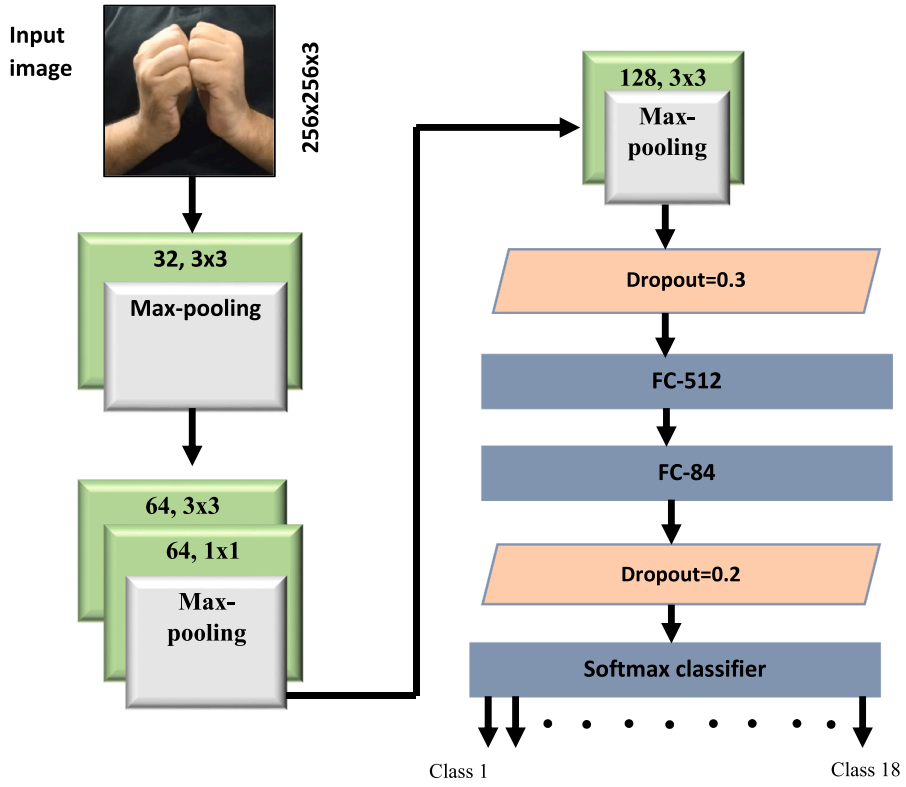

Fig. 5. Architecture of G-CNN model.

probability of discarding the inactive neurons has been used to avoid the problem of overfitting. Finally, the output from the last fully connected layer is fed into the soft-max layer to predict the classes by computing their respective probability distribution using Eq. (2).

$$P(y=i|x)=\frac{e^{x^T w_i}}{\sum_{k=1}^{K} e^{x^T w_k}} \tag{2}$$

here x^T denotes the T-th element of the array and K is the count of the total number of elements in the array x. The complete configuration of this model is given in Table 3. Pseudo code for the G-CNN is given in algorithm 2

As discussed in this section, this model has been specifically designed for the gesture recognition to deal with the sign gestures of sign language. In order to evaluate the effectiveness of this model, two state-of-art CNN architecture, namely VGG-11 and VGG-16 has been also studied and tested for this work. The detail of these models is discussed in the next section.

Algorithm 1: Pseudo code for activation function

// Input: Extracted feature from the convolution layer
// Output: Elimination of the negative value
int i = 1
Extract the feature vector from the first convolutional layer C_i
for 2 < i < 5 ; for 4 convolutional layer of the G-CNN
{
obtain the feature vector extracted by previous layer L_{i-1}
Apply the Activation function : *tanh*

$$y=\frac{1-exp^{-2x}}{1+exp^{-2x}}$$

Pass the feature vector refined by activation function to the next layer of the model
Extract the feature with next convolution layer C_{i+1}
}

Algorithm 2: Pseudo code for G-CNN

// it will detect and classify the signs
Input: input an image as X ={X^1,X^2,X^3……X^k} ; where k is total number of classes
G-CNN: Input image will be sent to this model to get the key features
// perform model prediction
model.fit()
perform softmax classification for 43 classes

$$P(y=i|x)=\frac{e^{x^T w_i}}{\sum_{k=1}^{K} e^{x^T w_k}}$$

// Output: Predicted Class

Advantages of the G-CNN:

1. The proposed model finds the key features from the input frame automatically. As a result, this method outperforms the feature-extraction based recognition system.
2. 4 convolutional layer, followed by 3-pooling layer, 2-dropout, 2-fully connected layer and 1 softmax layer has formed compact representation of CNN architecture. This compact representation will produce less trainable parameters which will lead to low computational load. This is the most desired quality of any model for any real-time application.
3. Even with less-deepened architecture of G-CNN, this method produces good recognition result with less training time consumption over the state-of art models of deep learning.

3.2.2. VGG-11 and VGG-16

The VGG network architecture was proposed by Simonyan and Zisserman (2014) to study the effect of a deeper convolutional network on classification accuracy. The VGG models have a wide range of applications in object detection, image captioning, texture recognition, and many more (He et al., 2019; Liu et al., 2021). To the best of author's

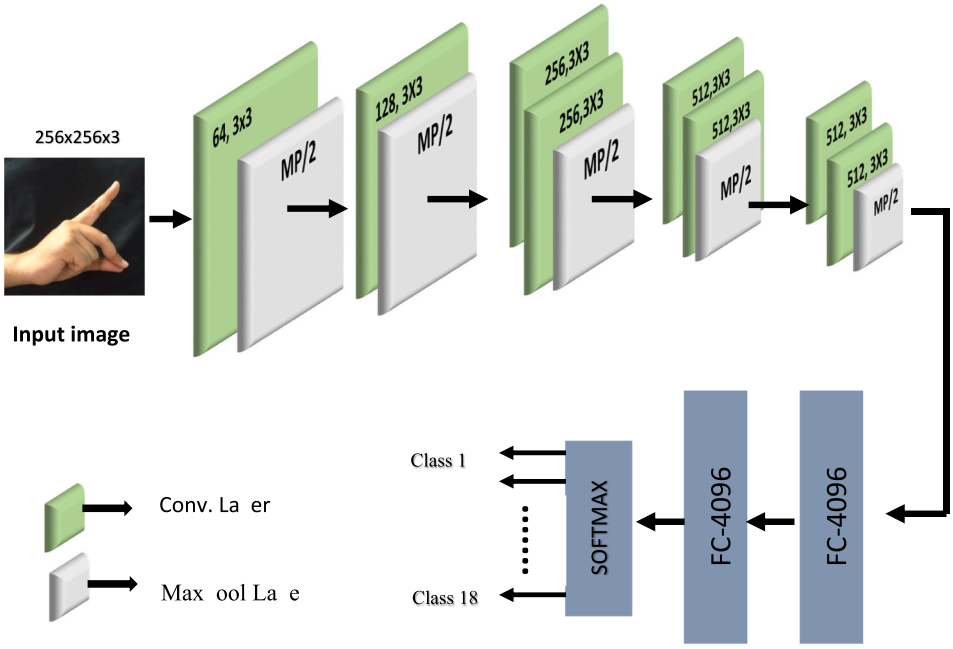

Fig. 6. Architecture of VGG-11model.

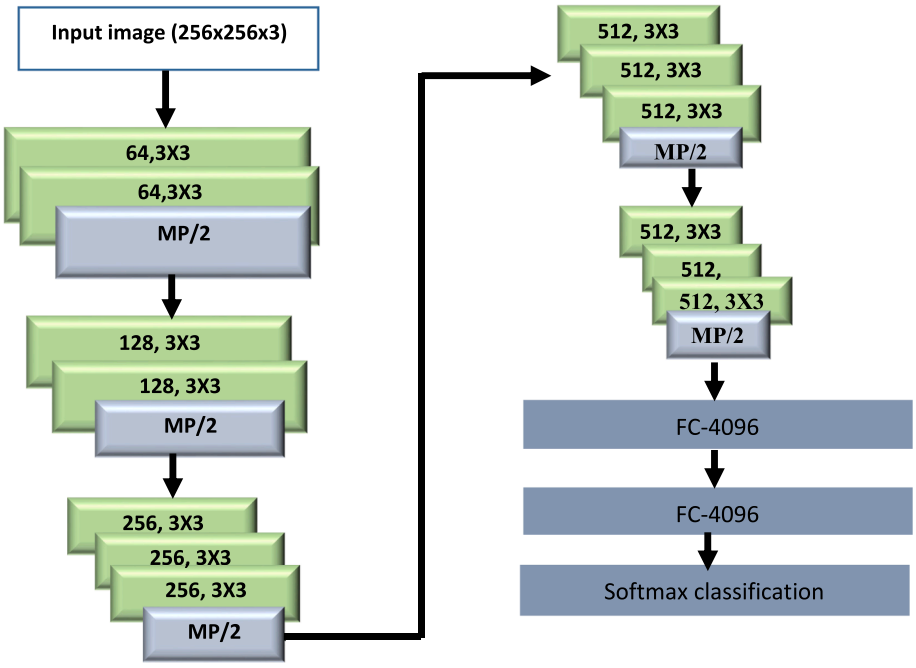

Fig. 7. Architecture of VGG-16 model.

knowledge, none of the researchers had tested the efficiency of VGG models for recognition of fingerspelling and isolated words. Hence, their potential for achieving good classification results in the sign language translation system is undetermined.

In this paper, the performance of VGG-11 and VGG-16 on hand gesture recognition in context with sign language has been examined. The original models were designed and tested with an input image size of [224 × 224 × 3]. In this paper, the acquired dataset has been scaled downed to the size of [256 × 256]. Thus the VGG-11 and VGG-16 models have been modified accordingly to these prepared datasets. The VGG-11 consists of a total of 11 weighted layers comprising of 8 convolutional layers and 3 fully connected layers followed by a single Softmax layer. In VGG-11 all convolutional layer uses a filter of size [3 × 3] with a stride of 1 and pooling layer uses a filter of size [2 × 2] with a stride of 2. Similar to this, the VGG-16 model consists of a total of 16 weighted layers i.e. 13 convolutional layers and 3 fully connected layers followed by a Softmax layer. The detail of these models is given in Figs. 6 and 7.

3.3. Training

In this work the proposed model of CNN, and other architecture of CNN (VGG-11, VGG-16) has been trained and tested for different datasets of sign language (given in Section 3.1). The images of each class are shuffled before feeding them to the feature learning model and are divided into three sets: 70% for training, 10% for validation, and the remaining 20% for testing. The data is fed into the network with a batch size of 32 samples in each training step and a total of 60 iterative training rounds have been conducted. This model for hand gesture recognition is trained using Adadelta optimizer, considering its ability to adapt to learning rates based on moving window of gradient updates. The initial learning rate and decay factor of Adadelta optimizer are set to 1 and 0.95 respectively.

Table 4
Classification results for Dataset-I.

Model		G-CNN	VGG-11	VGG-16
Accuracy (%)	Category-1	94.83	93.60	93
	Category-2	99.96	97.87	97
Loss	Category-1	0.3561	0.463	0.491
	Category-2	0.013	0.0613	0.176

4. Experiment and analysis

4.1. Experiment set-up

The proposed work has been implemented on the system with Intel Xeon W-2155 (10 Core, 3.3 GHz), nVIDIA Quadro P4000 8 GB graphics card using Python 3.6 with the Tensorflow and Keras framework.

4.2. Experimental result and analysis

In this paper, two different architecture of CNN (VGG-11 and VGG-16) and proposed model of CNN (G-CNN) has been tested for sign language recognition. The experimental findings for these models are discussed in this section, whereas the comparative study with other state-

Table 5
Classification accuracy for the augmented dataset-I.

Model	Category-1		Category-2	
	Original	Augmented	Original	Augmented
G-CNN	94.83	96.43	99.96	99.97
VGG-11	93.60	94.20	97.87	98.54
VGG-16	93.00	94.68	97	97.12

of-art approaches is presented in Section 4.3. For their performance evaluation, different matrices such as accuracy, loss, processing time and classification prediction results are used.

4.2.1. Accuracy

Classification accuracy is the most used quality measure index to measure the effectiveness of the classifier. It is defined as the ratio of correctly predicted samples to the total number of the samples of the dataset, as described in Eq. (3).

$$Accuracy = \frac{TN + TP}{FN + FP + TN + TP} \tag{3}$$

where TN, TP, FN, and FP are true negative, true positive, false negative, and false positive, respectively. The classification accuracy for dataset-I using G-CNN, VGG-11, and VGG-16 is shown in Table 4. The final accuracy achieved by the G-CNN model for ISL alphabets and static words is 94.83% and 99.96% respectively. The classification accuracy achieved by VGG-11 model for dataset-I is 93.60% and 97.87% respectively for category-1 and category-2. The accuracy for VGG-16 for the same dataset is 93% and 97%. The results reveal that superior performance is achieved with G-CNN over the VGG-16 and VGG-19.

The performance of the VGG-11, VGG-16 and G-CNN has been also computed on the augmented dataset. This has been done to generalize the trained models. Data augmentation is the process of generating new samples by transforming the original collected dataset. In this paper, four additional samples per each sample of the signer has been generated by using rotation and scaling operation. For this, a random rotation between the $[-20^{\circ}\ to + 20^{\circ}]$, and random inward and outward scaling of [0.8–1.5] has been used. The classification results for the augmented dataset are shown in Table 5. The outstanding results on the augmented dataset are convincing enough to prove the generalization ability of the trained models.

4.2.2. Loss

The categorically cross-entropy loss function is used in this paper to calculate the loss that occurred in the classification of multiple gestures of sign language. Mathematically, it can be represented using Eq. (4).

$$Loss = \sum_{i=1}^{n} O_i.log\widehat{O_i} \tag{4}$$

where $\widehat{O_i}$ is the i-th value in model output, O_i is the corresponding target value, and n is the number of the scalar values in the model's output. The value of loss observed for 3 different models for dataset-I is listed in Table 4. This calculated loss for all the three different models of CNN drops continuously with the increase in the iteration for some time, and later on, achieves a fix value. For the category-1 of dataset-I, the average loss for G-CNN, VGG-11, and VGG-16 drops to 0.3561, 0.463, and 0.491 respectively. For category-2, the loss for G-CNN drops to 0.013, for VGG-11 it drops to 0.0613, and for VGG-16 drops to 0.176. From Fig. 8, it is clear that G-CNN converges more rapidly than the VGG-11 and VGG-16.

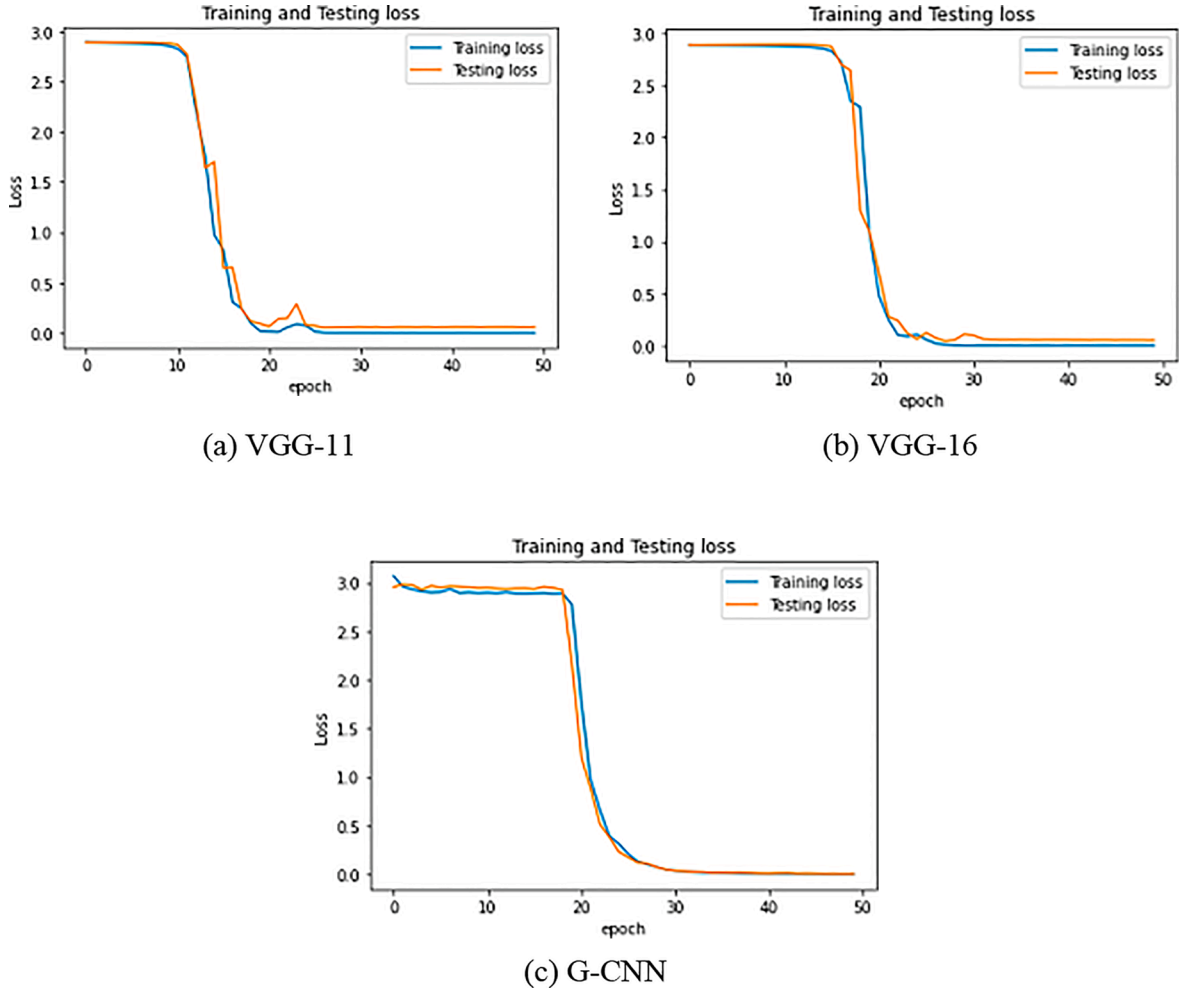

Fig. 8. Loss plot for ISL dataset.

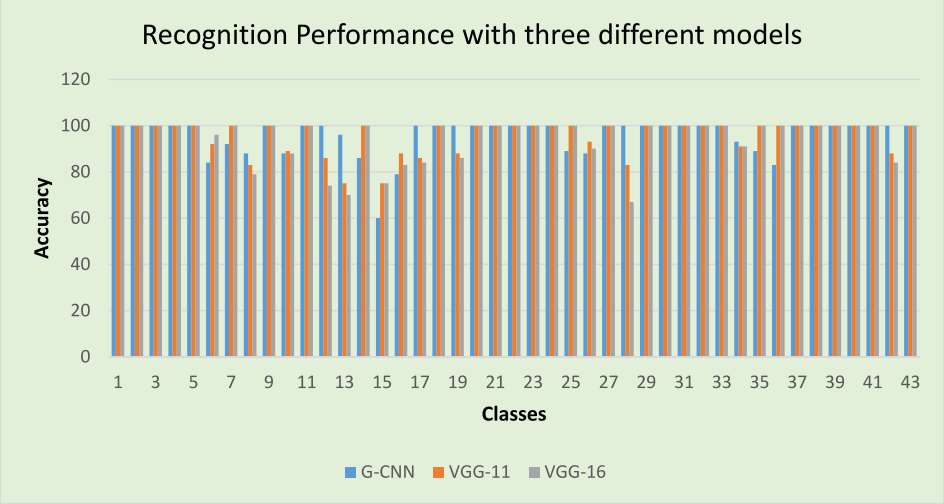

Fig. 9. Recognition performance of each class of dataset-1.

Table 6
Training time and total trainable parameters required by 3 different architectures for dataset-1.

Model	Time utilized (in mins)	Parameters used
G-CNN	9:21	67,250,830
VGG-11	40:15	160,297,362
VGG-16	44:33	165,791,570

4.2.3. Classification prediction result

In order to further evaluate the proposed work, another performance index called confusion matrix is also computed in this paper. This matrix summarizes the correctly and incorrectly predicted samples of each class and, hence, the accuracy of recognition of each class can be extracted from this. Fig. 9 depicts the detail of recognition accuracy for each class of dataset-I attained by the three different models of CNN. It can be seen that the proposed G-CNN gives promising results for most of the classes.

4.2.4. Other parameters

In the real-time application, computational time is a critical parameter for the recognition of hand gestures. Table 6 shows the training time utilized by three different architectures of CNN. The detail of parameters used by these models is also given in this table to determine the complexity of each model. The total number of trainable parameters can be computed using following expressions

Total parameters (P_{conv}) for each convolutional layer can be calculated with Eq. (5).

$$P_{conv} = \left(\left(width_{filter} * height_{filter} * no.offiltersofpreviouslayer + 1\right) * no.offilters \right. \quad (5)$$

Total parameters for each Fully connected layer (P_{fc}) can be calculated with Eq. (6).

$$P_{fc} = ((currentlayerc * previouslayerp) + 1 * c) \quad (6)$$

It is evident from the findings that the proposed architecture of G-CNN uses fewer parameters and less computational time than the advanced CNN models.

4.2.5. 10-fold cross validation

K-fold is a cross validation technique to stabilize the performance of the model. In order to evaluate the performance on the entire range of the data, 10- fold cross validation has been used for G-NN. Evaluation results for 10 folds are presented in the Table 7

4.3. Comparison of classification result

In this section, the performance of this method has been compared with the existing methods of the same classification problem of hand gesture recognition. The broad detail of this comparison is given in Tables 8 and 9. Comparison has been done based on their achieved accuracy only, as it is the only widely used performance metric in all the state-of-art approaches. In Table 4, the comparison of G-CNN has been made for ISL dataset. From this table, it has been found that Ansari & Harit (2016), Kaur and Joshi (2016), Joshi et al.(2017), Rekha et al. (2011), and Rao and Kishore (2018), have worked with limited numbers of signs and their achieved accuracy is 63.79%, 90%,93.4% 91.3%, and 90% respectively. It is evident from these findings that the G-CNN model surpasses all the other methods as it achieves the highest accuracy of 94.83%, and 99.96% for ISL fingerspelling and ISL isolated words respectively. Table 9 gives the comparison of this proposed work with the existing work for publicly available Triesch's ASL dataset. It is apparent from the findings that the G-CNN also proves to be powerful for this dataset.

Table 7
10- fold cross validation results of proposed work (G-CNN) with dataset-I.

k-fold		k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8	k = 9	k = 10
Accuracy (%)	Category-1	94.83	94.25	94.68	93.18	94.35	95.26	92.78	93.68	94.25	94.87
	Category-2	99.96	98.55	99.32	97.95	99.75	98.05	98.73	98.81	99.84	98.89

Table 8
Comparison of accuracy results of G-CNN with other methods on ISL dataset.

Author	No. of gestures	No. of signers	Repetition of dataset	Accuracy (%)
Ansari & Harit, 2016	16 signs (Category-1)	18	Not Mentioned	63.78
	16 signs (Category-2)	18		63.39
Kaur & Joshi, 2016	10 signs	72	Single	90
(Joshi et al., 2017)	26 signs	90	Single	93.4
Rekha et al., 2011	23 signs	10	Single	91.3
Rao et al., 2018	18 signs	10	Not Mentioned	90
Athira et al., 2019	24 signs	7	Multiple	90.1
Kumar et al., 2020	26	12	Single	80.76
Proposed (G-CNN)	Category-1(26)	50	Single	94.83
	Category-2(17)			99.96

Table 9
Comparison of G-CNN with published work for Treisch's ASL dataset.

Method	Accuracy (%)
EGM (Triesch & Von Der Malsburg, 2001)	92.9%
MCT (Just et al., 2006)	89.9%
WESF (Kelly et al., 2010)	85.1%
TM, Hu Moments and Geometric features (Dahmani & Larabi, 2014)	84.63% (LB)
	95.5 % (DB)
KM (Kaur & Joshi, 2017)	93.4%
DHM (Kaur & Joshi, 2017)	96.5%
DWT + F-ratio (Sahoo et al., 2018	95.42%
DHM + KM (Joshi et al., 2018)	98.8%
Proposed	100%

5. Conclusion

In this paper, a hand gesture recognition technique is presented for vision-based recognition of sign language. For this, a deep learning based G-CNN model packed with compact representation is proposed. In addition to this, two other architectures VGG11, and VGG-16 have been also examined and modified for the classification of sign language hand gestures. The proposed vision- based model eliminates the user dependency and need of external hardware equipment, thereby making it more practical to use. The significant contribution of this paper is its ability to recognize the complex signs of ISL with good recognition results over the state-of-art approaches. The performance of this work is tested for the self-collected 43 distinct ISL gestures and publicly available ASL dataset. By the mean of thorough experimental assessment, it is clear that for the 3 different categories of gestures used in this paper, the G-CNN model achieves the highest classification accuracy of 94.83%, 99.96%, and 100%. In addition to accuracy, other efficiency indices have been also used to ascertain the robustness of the proposed work. The model is also tested with the augmented data and is found as invariant to rotation and scaling transformation. It is evident from the comparative study that this model is robust enough in the classification of 43 different gestures with a low error rate. For future work, the architecture of these deep learning based models can further be optimized for hand gesture recognition and more detailed comparison can be made. These architectures can be explored more to minimize the error rate in real-time recognition of sign language.

Funding

Authors declare that no funding was received for this research work.

CRediT authorship contribution statement

Sakshi Sharma: Conceptualization, Methodology, Software, Writing original draft, Writing review & editing, Visualization, Validation. **Sukhwinder Singh:** Resources, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abraham, E., Nayak, A., & Iqbal, A. (2019). In *Real-Time Translation of Indian Sign Language using LSTM* (pp. 1–5). IEEE.
Akhter, S. (2018). In *Orientation hashcode and articial neural network based combined approach to recognize sign language* (pp. 1–5). IEEE.
Aly, W., Aly, S., & Almotairi, S. (2019). User-independent American sign language alphabet recognition based on depth image and PCANet features. *IEEE Access, 7*, 123138–123150.
Ameen, S., & Vadera, S. (2017). A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. *Expert Systems, 34*(3), e12197. https://doi.org/10.1111/exsy.v34.310.1111/exsy.12197
Ansari, Zafar. Ahmed., & Harit, GAURAV (2016). Nearest neighbour classification of Indian sign language gestures using kinect camera. *Sadhana, 41*(2), 161–182.
Arefnezhad, S., Samiee, S., Eichberger, A., Frühwirth, M., Kaufmann, C., & Klotz, E. (2020). Applying deep neural networks for multi-level classification of driver drowsiness using vehicle-based measures. *Expert Systems with Applications, 162*, 113778. https://doi.org/10.1016/j.eswa.2020.113778
Athira, P. K., Sruthi, C. J., & Lijiya, A., (2019). A signer independent sign language recognition with co-articulation elimination from live videos: an indian scenario. Journal of King Saud University-Computer and Information Sciences.
Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics, 10*(1), 131–153.
Chong, T. W., & Kim, B. J. (2020). American sign language recognition system using wearable sensors with deep learning approach. *The Journal of the Korea institute of electronic communication sciences, 15*(2), 291–298.
Dahmani, D., & Larabi, S. (2014). User-independent system for sign language finger spelling recognition. *Journal of Visual Communication and Image Representation, 25*(5), 1240–1250.
Gangrade, J., & Bharti, J. (2020). Vision-based hand gesture recognition for indian sign language using convolution neural network. *IETE Journal of Research*, 1–10.
Gupta, R., & Kumar, A. (2020). Indian sign language recognition using wearable sensors and multi-label classification. *Computers & Electrical Engineering*, Article 106898.
He, S. (2019). In *Research of a Sign Language Translation System Based on Deep Learning* (pp. 392–396). IEEE.
He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding box regression with uncertainty for accurate object detection. In *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2888–2897).
Joshi, G., Vig, R. & Singh, S., (2017). CFS-InfoGain based Combined Shape-based Feature Vector for Signer Independent ISL Database. In ICPRAM (pp. 541-548).
Joshi, G., & Gaur, A. (2018). In *Interpretation of indian sign language using optimal hog feature vector* (pp. 65–73). Singapore: Springer.
Joshi, G., Vig, R., & Singh, S. (2018). DCA-based unimodal feature-level fusion of orthogonal moments for Indian sign language dataset. *IET Computer Vision, 12*(5), 570–577.
Just, A., Rodriguez, Y., & Marcel, S. (2006). In *Hand posture classification and recognition using the modified census transform* (pp. 351–356). IEEE.
Just, A., (2006). Two-handed gestures for human-computer interaction (No. REP_WORK). IDIAP.
Kakoty, N. M., & Sharma, M. D. (2018). Recognition of sign language alphabets and numbers based on hand kinematics using a data glove. *Procedia Computer Science, 133*, 55–62.
Kang, B., Tripathi, S., & Nguyen, T. Q. (2015). In *Real-time sign language fingerspelling recognition using convolutional neural networks from depth map* (pp. 136–140). IEEE.
Kaur, B., & Joshi, G. (2016). Lower order Krawtchouk moment-based feature-set for hand gesture recognition. *Advances in Human-Computer Interaction, 2016*, 1–10.
Kaur, B., Joshi, G., & Vig, R. (2017). Identification of ISL alphabets using discrete orthogonal moments. *Wireless Personal Communications, 95*(4), 4823–4845.

Kelly, D., McDonald, J., & Markham, C. (2010). A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters, 31*(11), 1359–1368.
Kulshreshth, A., Pfeil, K., & LaViola, J. J. (2017). Enhancing the gaming experience using 3D spatial user interface technologies. *IEEE computer graphics and applications, 37*(3), 16–23.
Kumar, A., & Kumar, R. (2021). A novel approach for ISL alphabet recognition using Extreme Learning Machine. *International Journal of Information Technology, 13*(1), 349–357.
Kumar, P., Rautaray, S. S., & Agrawal, A. (2012). In *Hand data glove: A new generation real-time mouse for Human-Computer Interaction* (pp. 750–755). IEEE.
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.
Lichtenauer, J. F., Hendriks, E. A., & Reinders, M. J. (2008). Sign language recognition by combining statistical DTW and independent classification. *IEEE transactions on pattern analysis and machine intelligence, 30*(11), 2040–2046.
Liu, Y., Sun, P., Wergeles, N., & Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object Detection. *Expert Systems with Applications, 114602*.
Mariappan, H. M., & Gomathi, V. (2019). In *Real-Time Recognition of Indian Sign Language* (pp. 1–6). IEEE.
Mittal, A., Hooda, R., & Sofat, S. (2018). LF-SegNet: A fully convolutional encoder–decoder network for segmenting lung fields from chest radiographs. *Wireless Personal Communications, 101*(1), 511–529.
Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding, 104*(2-3), 90–126.
Ng, W. L., Ng, C. K., Noordin, N. K., & Ali, B. M. (2011). In *Gesture based automating household appliances* (pp. 285–293). Berlin, Heidelberg: Springer.
Pathak, B., & Jalal, A. S. (2019). Motion Direction Code—A Novel Feature for Hand Gesture Recognition. In *Computational Intelligence: Theories, Applications and Future Directions-Volume I* (pp. 487–493). Singapore: Springer.
Pugeault, N., & Bowden, R. (2011). In *Spelling it out: Real-time ASL fingerspelling recognition* (pp. 1114–1119). IEEE.
Raheja, J. L., Mishra, A., & Chaudhary, A. (2016). Indian sign language recognition using SVM. *Pattern Recognition and Image Analysis, 26*(2), 434–441.
Rao, G. A., & Kishore, P. V. V. (2018). Selfie video based continuous Indian sign language recognition system. *Ain Shams Engineering Journal, 9*(4), 1929–1939.
Rekha, J., Bhattacharya, J., & Majumder, S. (2011). In *Shape, texture and local movement hand gesture features for indian sign language recognition* (pp. 30–35). IEEE.
Sagayam, K. M., & Hemanth, D. J. (2017). Hand posture and gesture recognition techniques for virtual reality applications: A survey. *Virtual Reality, 21*(2), 91–107.
Sahoo, J. P., Ari, S., & Ghosh, D. K. (2018). Hand gesture recognition using DWT and F-ratio based feature descriptor. *IET Image Processing, 12*(10), 1780–1787.
Sharma, S., & Singh, S. (2020). In *Vision-based sign language recognition system: A Comprehensive Review* (pp. 140–144). IEEE.
Shrenika, S., & Bala, M. M. (2020). In *Sign Language Recognition Using Template Matching Technique* (pp. 1–5). IEEE.
Simonyan, K. & Zisserman, A., (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Tao, W., Leu, M. C., & Yin, Z. (2018). American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence, 76*, 202–213.
Tao, L., Zappella, L., Hager, G. D., & Vidal, R. (2013). In *Surgical gesture segmentation and recognition* (pp. 339–346). Berlin, Heidelberg: Springer.
Triesch, J., & Von Der Malsburg, C. (2001). A system for person-independent hand posture recognition against complex backgrounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(12), 1449–1453.
Wadhawan, A., & Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. *Neural Computing and Applications, 32*(12), 7957–7968.
World health organization (WHO). (2015). Deafness and hearing loss. Key Facts. Available online: http://www.who.int/mediacentre/ factsheets/fs300/en/. (Accessed 10 January 2021).
Wu, Y. & Huang, T.S., (1999). Human hand modeling, analysis and animation in the context of HCI. In Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348) (Vol. 3, pp. 6-10). IEEE.
Wu, C.-H., Chen, W.-L., & Lin, C. H. (2016). Depth-based hand gesture recognition. *Multimedia Tools and Applications, 75*(12), 7065–7086.
Xiao, Q., Qin, M., & Yin, Y. (2020). Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. *Neural Networks, 125*, 41–55.
Xie, B., He, X., & Li, Y. I. (2018). RGB-D static gesture recognition based on convolutional neural network. *The Journal of Engineering, 2018*(16), 1515–1520.