SN Computer Science (2022) 3: 380
https://doi.org/10.1007/s42979-022-01254-2

ORIGINAL RESEARCH

Deep Leaning Based Static Indian-Gujarati Sign Language Gesture Recognition

Dhaval U. Patel[1,2] · **Jay M. Joshi**[3]

Received: 16 December 2021 / Accepted: 17 June 2022 / Published online: 16 July 2022

Abstract

In this research paper, deep learning based static sign language recognition system is developed for Gujarati (an Indian) language. Total population of Gujarat State is more than 6 crores which is more than the population of 193 out of 216 countries of the world (90%). 14% of the people of Gujarat are facing deaf–dumb disability. The aim of the research is to create a vision-based application for speech impaired people to communicate easily and meaningfully. Dataset for training and validation is created from scratch due to unavailability of standard dataset. To improve robustness of dataset, augmentation, and various environment conditions were considered. Dataset was included with and without background images in different lighting conditions. Skin color segmentation is used for improved feature extraction. Proposed CNN architecture is designed such that it has minimum possible parameters. All the parameters of CNN were selected in organized structural manner to reduce computational complexity. The robustness of CNN model was verified using tenfold cross validation. Proposed deep learning based model is compared with other state-of-the-art machine learning algorithm. Feature extraction capabilities of proposed model are compared with other well-known techniques like principal component analysis and auto-encoder. The results of the proposed network validate the robustness of the network. This is the first research work on Gujarati (an Indian) static sign language. We contributed digital dataset for Gujarati language for recognition and our system achieved overall 92.69% accuracy.

Keywords Gujarati sign language (GSL) · Skin-color-based segmentation · Deep learning · Convolution neural network (CNN) · Hand gesture recognition · Sign language recognition (SLR)

Introduction

Information sharing is one of the essential necessities for civilization to thrive. Sign language is a typical mode of communication for deaf or dumb people. Deaf and dumb people use sign gestures to communicate amongst each other, but ordinary people struggle to understand their language. Very few people understand sign language/gesture. Furthermore, in face-to-face conversations, this form of interaction is impersonal and slow. [1, 2]. Scenarios, like when an accident occurs, where written correspondence is not always possible [3], it is always crucial to interact with the emergency doctor [4, 5]. This research is aimed at advancing the area of automatic gesture recognition.

American sign language (ASL) is more explored in comparison to Indian sign language. This project aims to classify Guajarati (Indian Regional) sign language alphabets; till now, no research work is done in Gujarati (Indian Regional) sign language. One might think Gujarat is just

✉ Dhaval U. Patel
dhvl1992@gmail.com

Jay M. Joshi
jaymjoshi@yahoo.com

1 Gujarat Technological University, Nr. Vishwakarma Government Engineering College, Nr. Visat Three Roads, Visat-Gandhinagar Highway, Chandkheda, Ahmedabad, Gujarat 382424, India

2 Electronics and Communication Engineering Department, L. E. College (Diploma), Near ITI-Morbi, Ghuntu Road, Morbi, Gujarat 363642, India

3 Electronics and Communication Engineering Department, Bhagwan Mahavir Polytechnic, BMEF Campus VIP Road, Bharthana Road, Vesu, Surat, Gujarat 395007, India

a state in India but still this research is fairly important, as the total population of Gujarat is more than 6 Crores. As per world bank country-wise population 2020 data, 189 countries out of 216 have a population of less than 6 crores [6] which is more than 89% of other countries, and 14% of that are deaf and dumb.

In comparison to ASL, GSL has its own challenges like the unavailability of standardized databases hampers study in this area. Due to shortage of datasets and heterogeneity in sign language with locality causes limitation in GSL gesture recognition efforts.

Recognition of gesture methodologies is generally split into two types: static or dynamic [2, 7]. Here, rather than using expensive technology, such as gloves, leap controller or Kinect [8, 9]. We intend to tackle the problems of static gesture recognition using deep learning. We're classifying 37 static gestures of Gujarati sign language. We created the dataset by ourselves. For better feature extraction, the dataset includes images with and without background and to overcome large dataset requirement, augmentation is used.

In this paper, we discuss the literature review in the next section. The following section consists of brief overview of suggested method and experimentations that we have conducted. The experimental results, the analysis and comparison against other methods is in the next section. In the subsequent section, result and discussion are given. Ultimately conclusion and future scope are discussed in the last section.

Literature Survey

Researchers approach hand gesture recognition challenges in many different ways. It is mainly classified into two approaches: the first approach is based on gloves or sensors and second is vision based.

Glove or sensor-based methods are efficient as dataset are typically extracted from signers. Also, sensors assist in successful gesture recognizing. On the other hand, vision-based techniques recognize based on the features extraction with various image processing algorithms [10]. Yet much of the literature on gesture recognition relies on vision-based approaches rather than glove-based methods. The reason is minimizing user's extra expense of various sensors. The sensor that is now accessible on most smartphones and laptops operates with a vision-based approach. When paired with appropriate processing of the image or video methods, the system worked fairly well.

All the researcher has used different techniques and different approaches. The following are the various methods used in sign language recognition in chronological order in recent years, based on gloves and vision.

Glove Based Approach

For this approach, sensors widely used are flexion, proximity, accelerometers, abduction sensors, and IMU (inertial measurement unit) sensors.

For Indian sign language, a system was proposed in this literature [11]. It used several flex and touch sensors. This method was proposed for 120 different static signs. Here the author proposed system for Indian sign language, using 3-axis accelerometer and Flex sensors for only 8 common words and 99% accuracy was achieved. For Israeli sign language recognition system [8] was suggested in 2012. 3-axis accelerometer and flex sensors sensor were used. It was applied on alphabets of finger spelling and accuracy achieved was around 94%. Literature [12], focuses on accelerometer Bend and Hall effect sensors. The technique was used for 0–9 symbols and accuracy achieved 96%. 5-flex sensor on fingertip and the accelerometer for American sign language are used here for in [13]. The digits from 0 to 9 and the alphabet from A to Z were recognized. The accuracy achieved was about 92% in this scenario. Technique for American sign language was suggested in [14] using Flex sensors and gyroscope. But it was only for three simple gestures and accuracy was about 86.67%. In literature [15], Chinese sign language recognition system was proposed based on random forest, and the proposed method achieved 98.25% accuracy for 121 words. Sensor used in this methods are accelerometer and surface electromyography. In literature [16], system was proposed for Chinese language for that they used surface electromyography, accelerometer and gyroscope sensors. For classification, decision tree is used. Accuracy achieved in this method was 94.31%. Using the Microsoft Kinect sensor interface [17, 18] has introduced an Indian sign language gesture recognition system. By using Kinect, they were exploring RGB and depth images. The study demonstrates that the performance is enhanced with RGB-D images. The system captures HU-Moments and feeds those features to the SVM classifier, which are invariant moments of angle, position and shape. In this literature [19], the proposed algorithm was developed to detect American sign language (ASL) from the Kinect sensor's depth images. The network is trained with 1000 images for each category of gestures. By training with those images, the Artificial Neural Network (ANN) extracted features. It obtained a 99.46% accuracy for the depth images. On the GPU, the network was trained to perform better. As an extension to this work, 33 static Kinect depth image signs is used with CNN using softmax function for classification. Technique was developed for 5-static and 8-dynamic gestures and 99.4% accuracy was achieved using multiple IMU sensors.

The approaches described above for sensor-based sign language recognition system produced positive results.

However, the sensors used in these methods are quite expensive. Also users must be in a specific setup to obtain proper results. These systems are not practicable for average users.

Vision Based Approach (Traditional Methods)

Various classifiers can be used to classify static and dynamic gestures. Till now, distributed hidden Markov models (HMM) [20], principal component analysis [21], fuzzy systems [22], artificial neural networks (ANN) [23], multiclass support vector machines (SVM) [24], convolution neural network (CNN) [10, 25] etc. are used for gesture recognition and classification.

A technique was proposed for the 2-D Indian sign language in [26]. In this literature, the edge frequency histogram was used to extract the feature and the SVM was used for classification. 26 alphabets were recognized and the accuracy was approximately 98.1%. In literature [17], the ISL recognition method was designed. A color-based gloves for segmentation and for identification of PCA was used. Real-time data frames are taken as inputs to be recognized in any 20th frame. For the 2-D American sign language, a technique was implemented in [27]. In this literature, for feature extraction Zernike Moment is used and for classification and recognition purpose SVM used. 24 alphabets were recognized and the accuracy was approximately 96%. In [28], B-spline approximation technique for Indian sign language recognition was used for the extraction of features and SVM classification was used. 29 signs were recognized and classified (A–Z alphabets and 0–5 number) and the accuracy was approximately 90%. Here Eigen value weighted Euclidean distance technique used in [29]. for ISL. 24 static gestures have been recognized and classified. The accuracy of this technique was 95%. In [23], the segmentation was based on skin color, the distance transform and the Fourier descriptor were used for ISL recognition, for classification purposes here in this literature, the artificial neural network was used. Accuracy for 26 alphabets and 0–9 digits was 91.11%. In this technique [30] an American sign language was proposed based on the fingertip and the palm position method. For feature extraction and classifcation purpose, PCA, optical flow and CRF used. An accuracy of 97% for 9 alphabets was achieved. In [31], for Indian sign language direct pixel value, hierarchical centroid, K-nearest neighbor, Neural network pattern and recognition tool were used. The recognized digits were 0–9 and an accuracy of 97.10% is achieved by this technique. Edge detection technique for hand gesture recognition was introduced in [4]. Using the sorting and edge detection functions, features were extracted. The template model was applied to the generated database to predict the gesture. Both dynamic signs and static signs have been identified by this technique. For recognition of Indian sign language, in [32] sequential minimal optimization is used for feature extraction and classification of Zernike moments. 5 alphabets were recognized and 94.4% accuracy was achieved. For Indian sign language, discrete wavelet transform and HMM are used in [20]. 91% accuracy was achieved for 10 types of sentences.

In this literature [33], an American sign language feature is extracted using skin-color segmentation, Zernike Moment, and fingertip detection. The SVM classifier used. 24 static alphabets and four dynamics have been classified. 93% accuracy achieved for static and 100% accuracy achieved for dynamics. A fuzzy-based approach is implemented in [22], using a fuzzy membership function, the system extracts the spatial characteristics of the signs. The nearest neighbor classifier calculates and matches with an appropriate measure of symbolic similarity.

Here in this literature [34], SIFT and key-point-extraction-based technique for 26 alphabets were proposed for ISL recognition. In this literature, more emphasis was given to time and not to accuracy. The implementation stated that for classification purposes, the handcrafted function is insufficient as the classes for classification increases. To address this issue CNN is used and it will surpass the accuracy of other conventional methods [17]. For ISL, an android app was developed in [35]. The android device capture images and sends them to the Matlab server. In the server, the features are extracted using the Sobel operator and for classification or recognition, CNN was used. The system identifies and classifies gestures and produces a text output. Two techniques for 3D vision based technique for Indian sign language are mentioned here. For Indian sign language, 3-D position trajectories and adaptive matching techniques have been used in [36]. 20 acts have been classified for 10 different subjects and the accuracy obtained is 98%. The [37] technique for Indian sign language was introduced in this literature. Using 3D-motionlets and adaptive kernel matching for recognition of 500 signals 98.9%.

In literature [38], Thai finger spelling sign language recognition system was proposed based on KNN classification algorithm, proposed method is achieved 97.25%. For feature extraction pyramid histogram of oriented gradients and local feature were used. Same author has work on same dataset with different approach in literature [39]. Here in this approach for classification purpose they have used SVM technique. To train SVM they used 4 different techniques: RBF, Linear, polynomial and sigmoid. They get best results when SVM is combined with RBF it is around 91.20%. In past few years, the deep learning has been highly efficient not only for gesture recognition but for various computer vision application[5].

Here, traditional vision based techniques are discussed. Primary challenge with these methods is that the researcher has to develop handcrafted feature extraction algorithm and

classifier algorithm. Because these methods do not automatically extract features, they have a low level of accuracy. These methods can classify limited number of gestures with great computational complexity. This is the motivation behind the growing interest of researchers in deep learning-based methods for SLR.

Vision Based Approach Using Deep Learning

The primary goal of deep learning approaches is automatic feature engineering. The objective is to automatically learn a feature set from raw data that can be used in sign language recognition. In this way, it avoids the difficult process of hand-crafted feature engineering by learning as a feature set automatically. Many researches published deep-learning-based methods for sign language recognition in recent past.

For vision-based gesture recognition, Bheda and Radpour [40] proposed an American Sign Language-based letter and digit recognition system. The suggested CNN-based architecture comprised of three convolution layers with a max-pool layer, a dropout layer, and two groups of fully connected layers. Using a stochastic gradient descent optimizer, the acquired datasets were preprocessed using the background subtraction method and achieved an accuracy of 82.5% on alphabets and 97% on digits.

Ankita et al. [41] proposed an algorithm for Indian sign language for 35,000 signs and 100 static signs using CNN and achieved 99.90% accuracy.

Rao et al. [42] used Deep CNN to create a selfie-based sign language recognition method. They developed a database that executes 200 signs at various angles and with varied backgrounds. On CNN, they used mean pooling, max-pooling, and stochastic pooling techniques, and it was discovered that stochastic pooling surpassed other pooling techniques with a recognition rate of 92.88%.

Huang et al. [43] used 3D CNN to create a Kinect-based sign language recognition method. They employed a 3D CNN to extract spatial–temporal features from raw datasets, which aids in the extraction of authentic features to cater to the wide range of hand gestures. This model was evaluated on a dataset containing 25 signs, achieving a recognition rate of 94.2%.

Huang et al. [44] presented a sign language recognition method based on real-sense experience. They gathered a total of 65,000 image frames containing 26 alphabet signs, 52,000 of which were used for training and 13,000 for testing. Deep belief network was used to train and classify neural network model, which attained an accuracy of 98.9% with real sense and 97.8% with Kinect.

Nagi et al. [45] suggested a max-pooling CNN. Color segmentation was used to obtain hand contour, and morphological image processing was used to eliminate noisy edges. Only 6000 sign images from six gesture classes were used in the studies, which produced a 96% accuracy.

Pigou et al. [3] worked on Microsoft Kinect and a CNN-based recognition system. For preprocessing, they utilized threshold method, background removal, and median filtering in this system. They used NAG optimizer to identify Italian gestures and achieved a validation accuracy of 91.7%.

Rioux-Maldague and Giguere [46] described a feature extraction algorithm for hand posture detection utilizing depth and intensity images collected with Kinect. They used a threshold on the highest hand depth for segmentation, resized the image, and preprocessed it with image centralization. Using a deep belief network, the results were analyzed on known and unknown participants. With known ones, recall and precision were 99%; with unknown ones, recall and precision were 77% and 79%, respectively.

Molchanov et al. [47] demonstrated a multi-sensor system for recognizing driver hand gestures. They adjusted data from depth, radar, and optical sensors before using neural network to classify 10 different motions. The test findings revealed that the system attained the highest accuracy of 94.1% when all three sensors were used together.

Tushar et al. [48] suggested a deep CNN-based numeric hand sign recognition method. They provided a layer by layer optimized architecture in which batch normalization speeds up training convergence and the dropout technique reduces data overfitting. The acquired American Sign Language (ASL) data were optimized using CNN's Adadelta optimizer, yielding an accuracy of 98.50%.

Oyedotun and Khashman [25] proposed a static hand gesture recognition system based on vision that recognizes 24 American Sign Language alphabets. The entire hand gestures were acquired from Thomas Moeslund's gesture recognition database, which is freely available to the public. They used the CNN network and the stacked denoising autoencoders (SDAE) network, and their accuracy on testing data were 91.33% and 92.83%, respectively.

Tang et al. [49] proposed a method using the Kinect sensor to create a hand posture recognition system. For preprocessing on the acquired data, they used hand detection and tracking algorithms. The suggested network is trained on 36 different hand postures using a CNN model based on LeNet-5. DBN and CNN were used in the testing, and it was discovered that DBN outperformed CNN with an overall average accuracy of 98.12%.

Yang and Zhu [50] demonstrated CNN for video-based Chinese sign language (CSL) identification. They collected data using 40 everyday vocabularies and demonstrated that the developed method simplifies the hand segmentation method while avoiding information loss during feature extraction. They utilized the Adagrad and Adadelta optimizers to learn CNN and discovered that Adadelta surpassed Adagrad.

Koller et al. [51] presented a hybrid approach for continuous sign recognition that combines the strong classification properties of CNN with the sequence modelling characteristic of the hidden Markov model (HMM).

A dynamic programming-based method was used to preprocess the acquired image. The hybrid CNN-HMM method outclasses the other state-of-the-art approaches, according to the results.

Kumar et al. [52] presented a two-stream CNN architecture that uses two color-coded images as input: the joint distance topographic descriptor (JDTD) and the joint angle topographic descriptor (JATD). They acquired and constructed a dataset of 50,000 Indian sign language videos and attained an accuracy of 92.14%.

This paper aims to develop a complete system based on deep learning models to recognize static signs of Indian sign language. The major challenge regarding our problem is that in India there was no standard dataset for sign language recognition. Unavailability of dataset is because no one has worked on Guajarati sign language till now. So, we borrowed idea from this literature [7], decided to create our own dataset by collecting from different users. Gujarati sign gestures dataset was developed after consulting few teachers of Government Gujarat State School for the deaf and dumb. They provide signs for all the alphabets. The final dataset is of 37 gestures and a total of 27,708 images of 9 different people. A web camera-based dataset of a variety of shapes and skin tones with different background and lightning condition is collected. As we discussed in literature review for each signs 700–1000 images are adequate to training and 150–200 images are adequate for validation and testing. This dataset is sufficient for primary investigation as no one explore this field yet.

Methodology

Flow chart of the proposed technique is shown below in Fig. 1. Pre-processing for dataset creation, Skin Segmentation and Process to design deep learning based CNN are mentioned here.

Gujarati sign gestures were developed based on data acquired from the Government School of Gujarat for deaf and dumb. The database consists of two kinds of images, the first kind has gestures with background and second kind ha skin-color-based segmented gestures. The database was created from 9 people, except for the letters ઉ (Au), ઊ (Aau), ઐ (ai) and ઔ (oai), as they are dynamic gestures.

For database creation 9 people helped. All from different age group in different background and lighting conditions. We used a Bison easy camera, with a 720p HD resolution to acquire dataset. Dataset dimension are 128×128 pixels and color-map is grayscale. To increase the dataset variety, we applied several augmentation filters 10°degree rotations clockwise and counter clockwise, width and height wise shift of 10%, and a shear angle of 15°degrees, Thus, the final dataset comprised of 37 gestures and a total 27,708 images. A variety of forms and skin colors were given by such a large sample size, as shown in Fig. 2 different examples of signs K ((ક)). In Fig. 3, image of some other gestures can be seen.

Pre-processing

For better training performance of CNN, it is important to extract relevant features from input. For that Segmentation techniques and pre-processing were imposed.

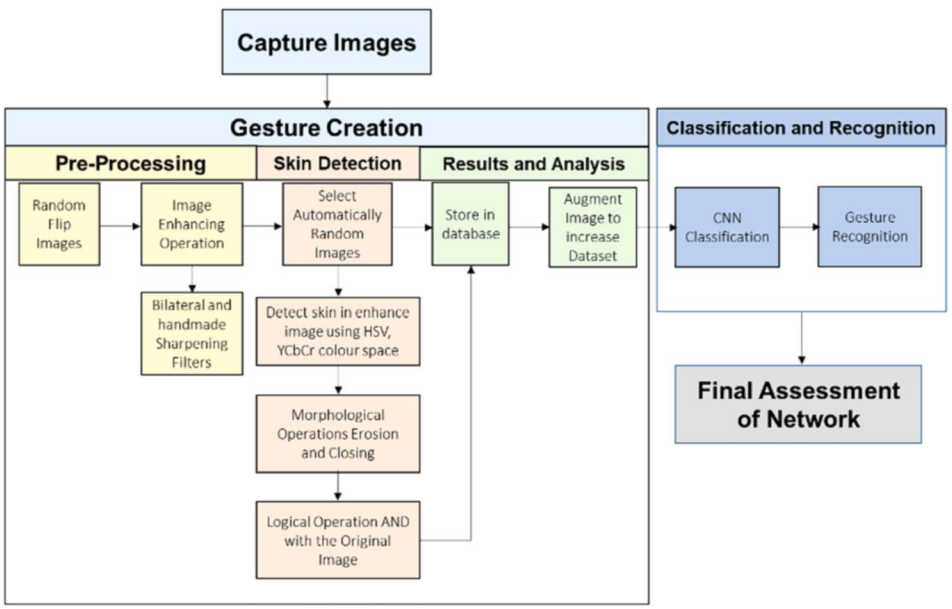


Fig. 1 Flow of proposed approach

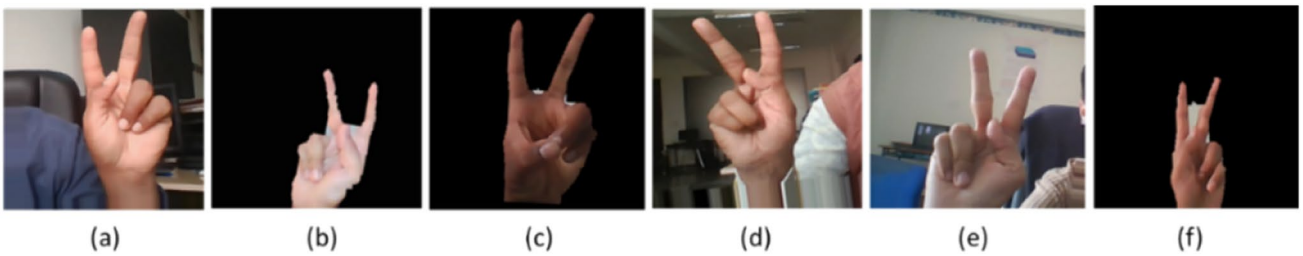


Fig. 2 Image for gesture K(ক)

Fig. 3 Some Images from the dataset

For pre-processing, a bilateral filter is used for smoothening images and reducing noise. Other filters like averaging filter or median filter can also be used, but it results in a loss of important edge information. To counter this problem, the bilateral filter is used here. The morphological filters such as erosion is used for closing gaps, dilation is helpful in eliminating or emphasizing features from a segmentation.

Skin detection is the process of finding skin-color in an image for detecting hands in an image for hand gesture recognition [53]. For detection, the formulating mathematical model to describe the distribution of skin color is an important step. A blend of blood (red) and melanin (yellow, brown) produces the color of human skin. In these two extreme hues, skin tones exist and are quite saturated.

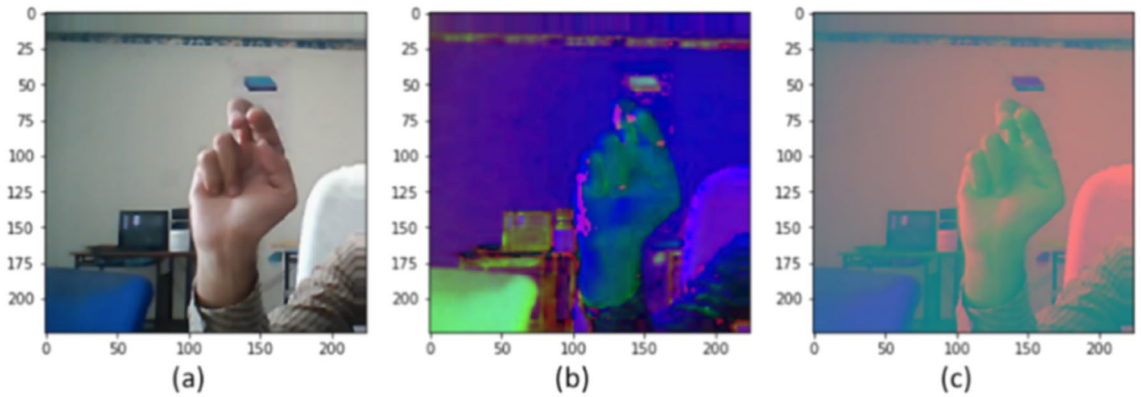


Fig. 4 Hand image in RGB, HSV and YCbCr color space

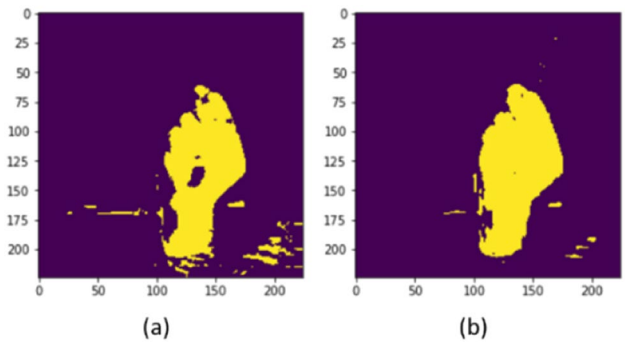


Fig. 5 Holes and noise observed in the hand region

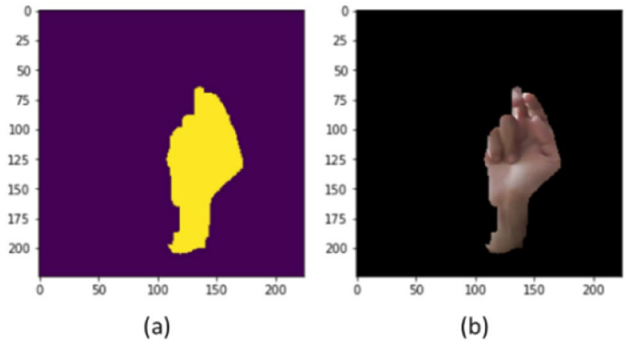


Fig. 6 Holes and noise observed in the hand region

For detection filter design using two color space HSV and Ycbcr and not RGB because RGB is more related to primary colors and not much useful for skin detection. HSV and YCbCr color space is similar to how humans perceive color [53–56]. RGB, HSV and YCbCr color space of image is shown in Fig. 4. HSV, YCbCr range for segmentation is as follows

$$0 < H < 25, 40 < S < 255, 0 < V < 255$$

$$0 < Y < 255, 135 < Cb < 173, 67 < Cr < 133$$

Segmentations display gaps and noise in the hand area, in Fig. 5, Morphological filter dilation, filling and erosion for noise reduction are applied to resolve these issues as shown in Fig. 6. Lastly between the segmentations and the actual images, AND operation is applied. Here we can see that many gestures having similarity in its shape in Fig. 7. By keeping the orientation, position of the hand and features details we can have resolved the similarity challenge, In Fig. 7 similar gestures are shown but with preserved palm feature and finger positions similarity reduced.

Convolutional Neural Networks and Tuning and Optimizing Hyper-parameters

CNN consist of three types of layer. input layer, hidden layer and output layer. Input layer consists of dataset images. Hidden layer consist of several other layers and parameter that we have to select for optimal results. First type of layer is convolution, second is pooling and third is fully connected dense layer. Simple CNN structure is illustrated in Fig. 8. Convolution neural network depends on many parameters as per specification. Here is some example of some key parameters that we need to adjust to optimize the performance of CNN e.g. number of convolution layer, pooling layers, dense layers, neuron number in each layer, activation function, learning rate and dropout value, Kernel window size, stride value and padding etc.

Convolution Layer

The convolution layer is the first layer of CNN. Its purpose is to extract features from the input images. Convolution is a mathematical operation between an input picture and a kernel filter of a certain size XxY. By striding (sliding the

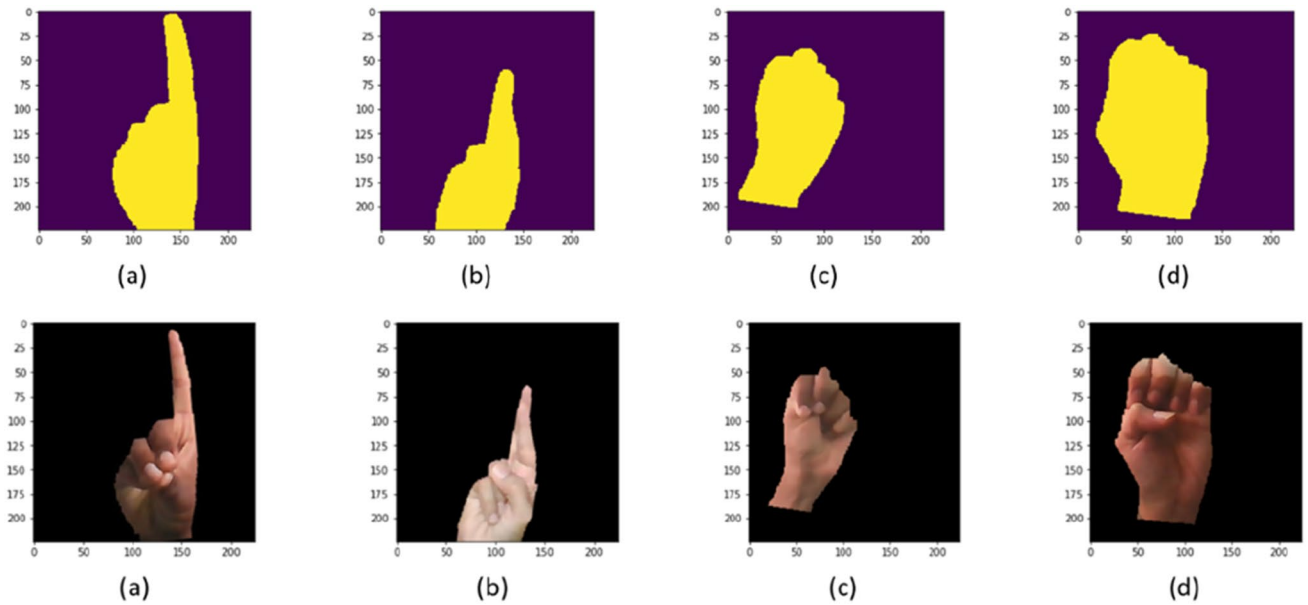
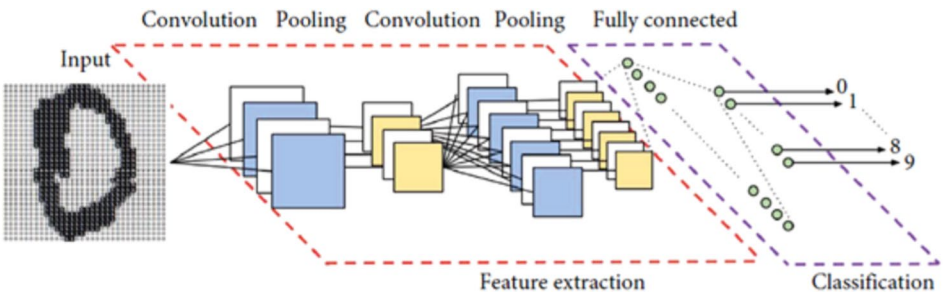


Fig. 7 Gestures with similarities in without preserving features and with preserving features

Fig. 8 CNN and its layer [2]



filter) over the input image and the dot product is calculated between the filter and the regions of the input picture that are proportional to the filter's size (XxY).

The result is known as the Feature map, and it contains corners and edges information of the picture. This feature map is then applied to next layers, which learn various different features from the input image.

Mathematical equation of convolution layer is as follows [10]:

$$f_{i,j,k} = b_k + \sum_{u=1}^{fx} \sum_{v=1}^{fy} \sum_{k'=1}^{fn} x_{i',j'} \cdot w_{u,v,k},$$

$F_{i,j,k}$ is the results of convolution of row i, column j and feature map k of layer L. f_x, f_y is the dimension of the $L-1$ filter matrix. f_n is the amount of filters in $L-1$ layer. $x_{i',j'}$ is the pixel value with of i', j' in $L-1$ layer, where convolution performed. $w_{u,v,k}$ is the value of pixel in kth filter with value of index u, v. b_k: bias of kth feature map [10].

Pooling Layer

A convolutional layer is usually followed by a pooling layer. This layer's purpose is to lower the size of the convolved feature map to reduce computational expenses. Pooling operations are classified into many types based on the approach.

The largest pixel value from the feature map is used in max pooling. average pooling computes the average of values in a predefine image section. Sum pooling computes the total sum of the values in the designated Image section. Typically, the pooling layer acts as a link between the convolutional layer and the FC layer. For max pooling mathematical function is expressed as

$$y = \max(x, 0), \quad y = \max \; i(xi).$$

Here, $[xi]$ describes the number of all xi, which is a receptive field.

Fully Connected Layer

The FC layer which includes weights, biases and neurons. It is used to link neurons from other layers. These layers are often placed prior to the output layer.

Here preceding layers' input images are flattened and applied to the FC layer, where the mathematical function operations are often performed. At this point, the categorization procedure is initiated. Mathematical equation for fully connected network is as follows:

$$a = \sigma(Wx + b).$$

Here, W is weight and b is bias.

Dropout

When all of the featured are linked to the FC layer or model with more neurons and convolution layer, the model tend to be overfitting. Overfitting happens when model performs so well on training data that it has an unfavorable influence on the model's performance when applied to fresh unknown data.

To counter this issue, a dropout layer is used, in which a few neurons are removed from the neural network during the training process, resulting in a smaller model. When a dropout of 0.3 is reached, 30% of the nodes in the neural network are dropped out at random.

Activation Functions

The activation function is a critical parameter in the CNN model. They are used to learn and estimate any type of continuous and complicated connection between network variables. In other words, it determines which information model should be sent forward.

It introduces nonlinearity into the network. There are various activation functions that are mostly used, like ReLU, Softmax, tanH, and Sigmoid functions. Each of these functions have a special use. The sigmoid and softmax functions are recommended for a binary classification CNN model, while softmax is often utilized for a multi-class classification. As in the following equation, the softmax activation equations can be defined below.

$$y_j = \frac{e^{x^T wj}}{\sum_{k=1}^{K} e^{x^T wk}}.$$

Experiments

For experimental work, hardware are as follows: CPU is of Intel (R)-Core(TM) i5-7300HQ @2.50 GHz, NVIDIA GeForce GTX-1050 GPU and 8 GB RAM. For software computation Python 3.0, Tensorflow API 2.0 is used. We have designed a CNN architecture such that it has minimum possible parameter without compromising the accuracy. So for tuning the parameter, we follow a systematic procedure. We have analyzed 82 different model and from that we systematically selected all the parameters. In Fig. 9 loss and accuracy graph of all CNN model is illustrated.

As per literature, from all the parameters, first of all we need to decide how many filters would be enough for our dataset and when our model is overfitting. We can see the results in Fig. 10. It shows as filter value increased, accuracy is also increasing (loss is decreasing). But for 128 filters, model is overfitting. So, we selected 64 filters for each layers.

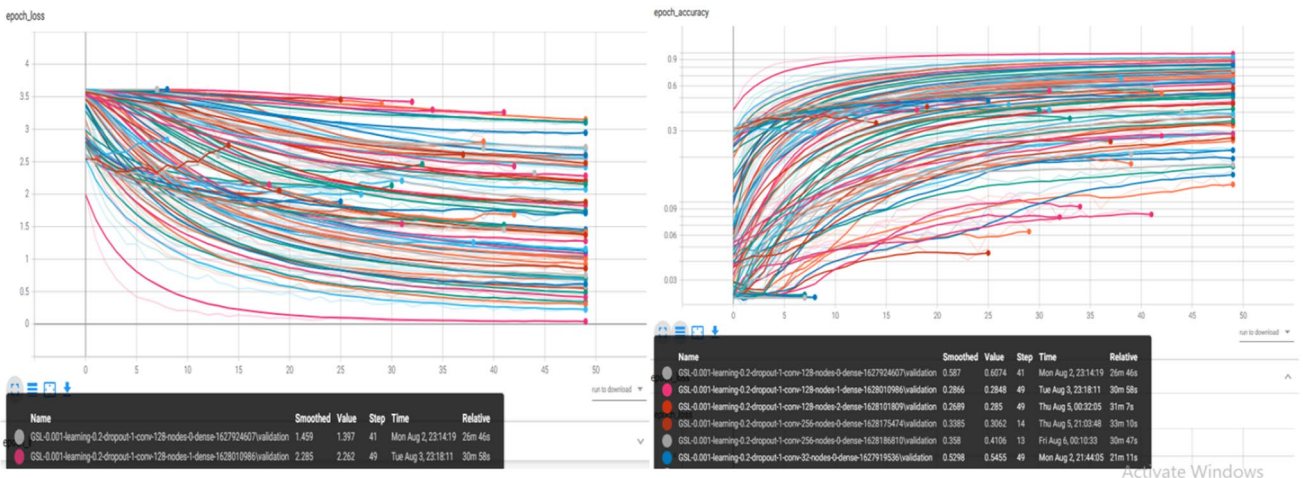


Fig. 9 Validation accuracy and loss of 82 models

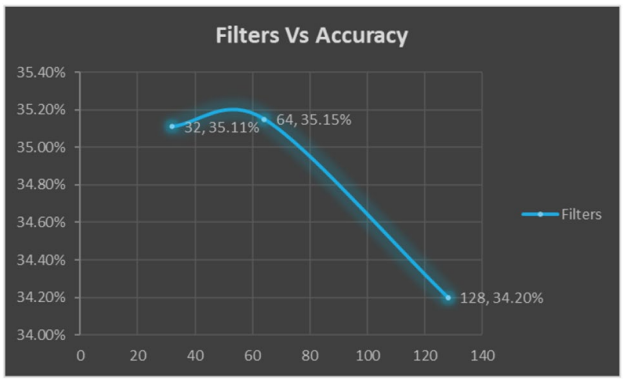


Fig. 10 Filter vs accuracy

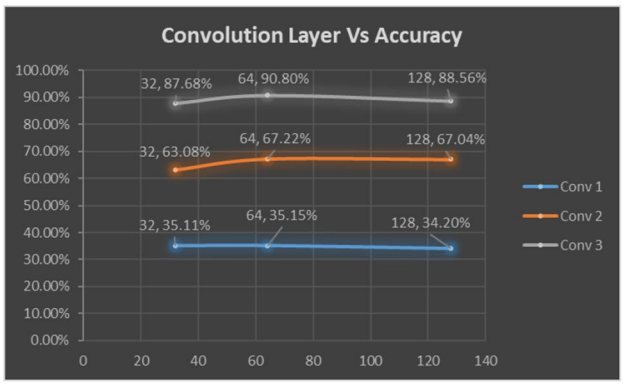


Fig. 11 Convolution vs accuracy

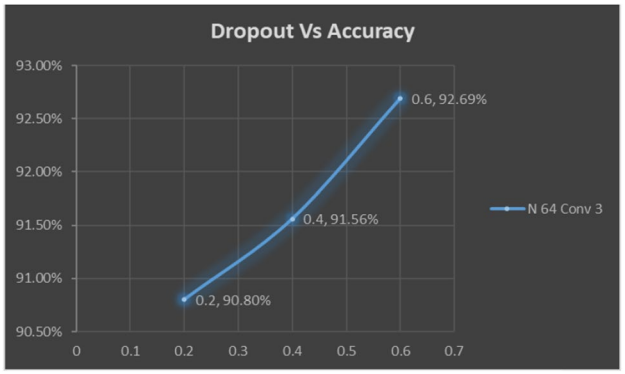


Fig. 12 Dropout vs. accuracy

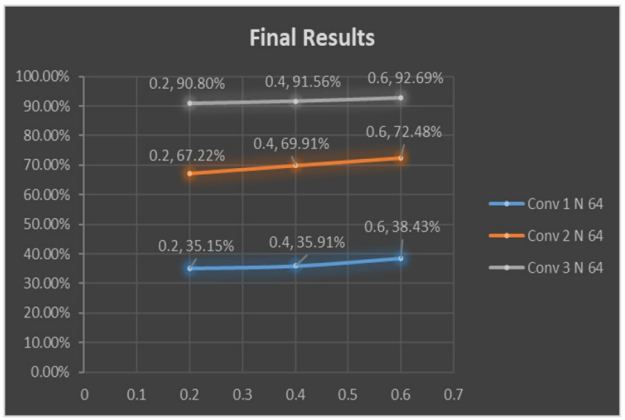


Fig. 13 Final result

After selecting neurons value, we analyzed the results of convolution layers. As can be seen in Fig. 11, for all different values of convolution layers' filter's values are mimicking the same pattern which increases our confidence with 64 filters in each convolution layer. Here as we are increasing convolution layer model accuracy is also increasing as can be seen in below figure for 3 convolution layer and 64 filters model is giving best accuracy 90.80%. So we selected 3 convolution layer.

After selecting number of convolution layers, we analyzed dropout values. Here in Fig. 12, we can clearly see that as dropout value increases, accuracy also increases. We selected a maximum value of 0.6. We can even increase dropout value to 0.8 or as close as possible to 1 but results were not improving. So we selected a dropout value of 0.6.

In Fig. 13, we can see all the results and we can clearly see that for 64 filters, 3 convolution layers and 0.6 dropout model is giving model perform best. Here we have selected dense layer value as 0. For value of Dense layer 1 and 2 model was overfitting.

To further analyze the robustness of the model, we perform tenfold cross validation. Result are shown in Table 1. It confirms the robustness of the Model. Validation accuracy and confusion matrix's results are shown in Table 2 and architecture of proposed network is illustrate in Table 3 and Fig. 14.

Table 1 Results of each folds

	Accuracy	Loss
Fold 1	89.91%	0.31
Fold 2	92.82%	0.24
Fold 3	91.42%	0.27
Fold 4	91.71%	0.26
Fold 5	91.26%	0.26
Fold 6	91.58%	0.26
Fold 7	90.27%	0.31
Fold 8	91.38%	0.28
Fold 9	92.28%	0.26
Fold 10	90.15%	0.30
Average	91.28%	0.27

Table 2 Outcomes of suggested architecture

Parameter	Architecture CNN1
Val accuracy (in %)	92.69%
Val loss (in %)	0.28%
Precision (in %)	93.65%
Recall (in %)	93.23%
F1 score (in %)	93.01%

Results

Comparison with Traditional Algorithms

We have applied our own GSL dataset on various state of the art machine learning techniques like, SVM, KNN, random forest and decision tree. We have also analyzed the results of all algorithm combined with different feature extraction methods—PCA and auto encoder. For PCA, 5000 features were considered from feature distribution variance. Feature distribution variance is as shown in Fig. 14.

Result of all 15 different techniques is shown in Fig. 15.

Comparison with Existing Method

The comparative analysis of the proposed technique with other CNN model using our own dataset is shown in Table 4. It has been observed that the proposed system outperformed all the other existing ISL systems with an accuracy of 92.69% (Fig. 16). Another thing to be observed is that the proposed neural network is one with greatly less parameters compared to existing techniques which indicates that the proposed network has considerably less complexity and requires less training time.

Table 3 Suggested convolution neural network architecture

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 126, 126, 32)      320
_________________________________________________________________
activation_1 (Activation)    (None, 126, 126, 32)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 63, 63, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 63, 63, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 59, 59, 64)        51264
_________________________________________________________________
activation_2 (Activation)    (None, 59, 59, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 57, 57, 64)        36928
_________________________________________________________________
activation_3 (Activation)    (None, 57, 57, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 28, 28, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 24, 24, 64)        102464
_________________________________________________________________
activation_4 (Activation)    (None, 24, 24, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 22, 22, 64)        36928
_________________________________________________________________
activation_5 (Activation)    (None, 22, 22, 64)        0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 11, 11, 64)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 11, 11, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 7744)              0
_________________________________________________________________
dense_1 (Dense)              (None, 37)                286565
_________________________________________________________________
activation_6 (Activation)    (None, 37)                0
=================================================================
Total params: 514,469
Trainable params: 514,469
Non-trainable params: 0
```

Conclusion and Future Scope

First of all, creating a dataset for GSL was a mammoth of a task. As no standard dataset is available, we are the first

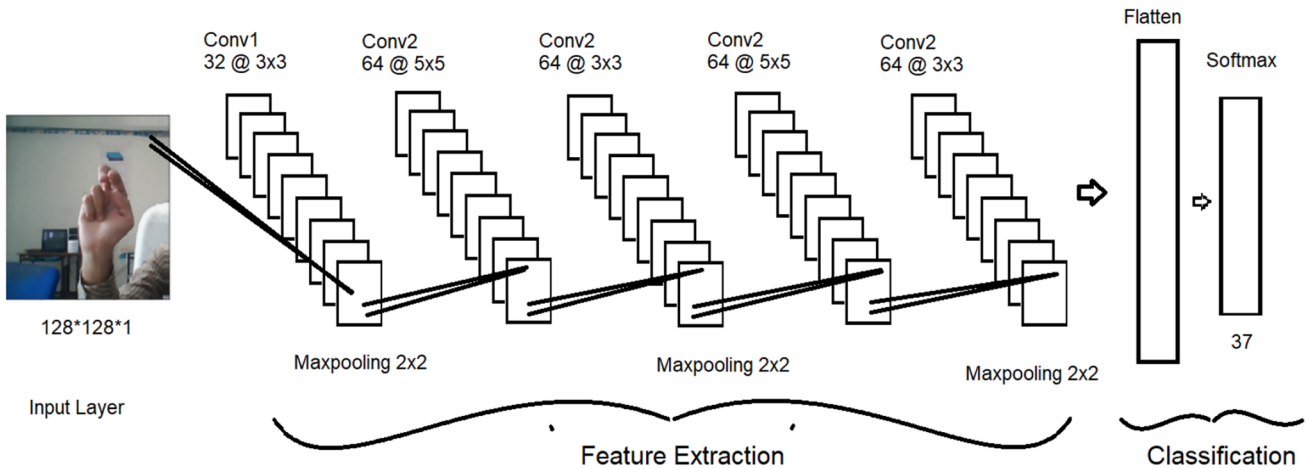


Fig. 14 High level general CNN architecture

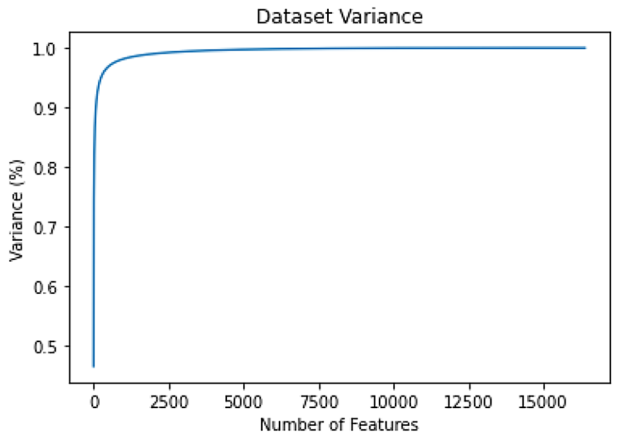


Fig. 15 Feature variance ratio

to create a digital fingerspell alphabet dataset for the GSL (Gujarati sign language). Dataset consists of 37 gestures and a total 27,708 images. After dataset creation, we have analyzed more than 80 different convolution neural networks and strategically designed convolutional neural network with minimum possible parameters. We compared our proposed technique with other state-of-the-art machine learning technique like SVM, KNN, random forest and decision tree. We even compare results of all different algorithm combined with PCA and auto-encoder feature extraction techniques. Results demonstrate that CNN with an optimal pre-processing methodology produces outstanding results from all the other machine learning techniques. CNN performance was reduced when combined with PCA and auto encoder based model. In PCA, we manually decided 5000 features for classification purpose, whereas in auto-encoder, an algorithm omits some irrelevant feature automatically to reduce computational complexity but surely at the expense of accuracy. Base CNN model considers all features and achieved maximum 92.69% accuracy. For future work, we need to collect more dataset to improve performance in real time. With more dataset, complexity will be increased; to counter it certain parameters are needed to be analyzed in the future. We have to analyze different optimizer like adam, adaboost, etc. To converge the model quickly batch normalization and to avoid overfitting dropout function has to be analyzed with different values. In addition, we can go for dynamic gesture recognition. For that we need to create video-based dataset. This work can be extending to mobile application also.

Table 4 Comparision of the proposed technique with other state of the art SLR CNN model using our own dataset

Author	Accuracy (%)	# of trainable parameters	Remarks
Md Shafiqul et al. [57]	Model overfitted	5,837,469	# of neurons are more with each layer (more than required) very complex model
Rao et al. [42]	69.07%	4,982,373	–
Ankita et al. [41]	90.32%	4,073,540	–
Proposed Method	92.69%	514,469	–

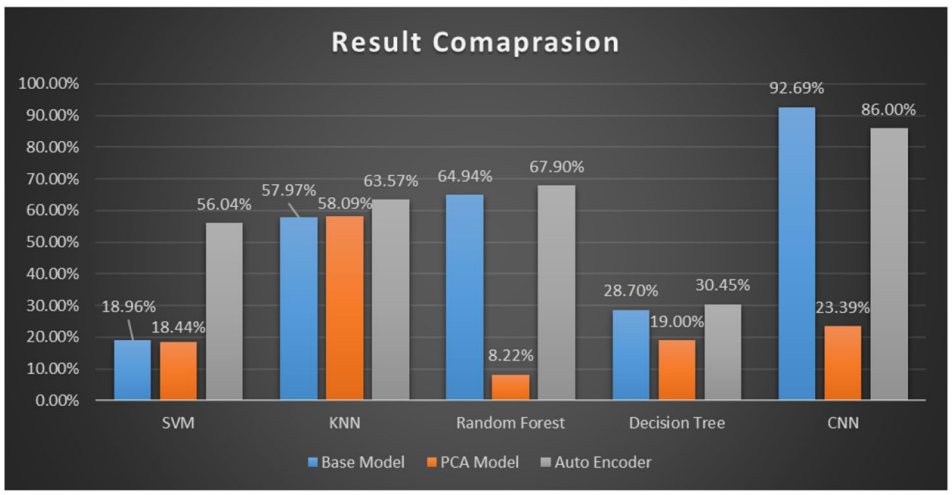


Fig. 16 Comparison of all techniques

Data Availability In this article, the database was created by the researchers and is presently unable to be made freely accessible. However, on request, author might make it accessible.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report.

References

1. N. R. C. Committee on Developments in the Science of Learning with additional material from the Committee on Learning Research and Educational Practice. How people learn: brain, mind, experience, and school: expanded edition. Washington, D.C.: National Academy Press; 2000.
2. Pinto RF, Borges CDB, Almeida AMA, Paula IC. Static hand gesture recognition based on convolutional neural networks. J Electr Comput Eng. 2019;2019:4167890.
3. Pigou L, Dieleman S, Kindermans P-J, Schrauwen B. Sign language recognition using convolutional neural networks. In: European Conference on Computer Vision. 2014. p. 572–78.
4. Nanivadekar PA, Kulkarni V. Indian sign language recognition: database creation, hand tracking and segmentation. In: 2014 International conference on circuits, systems, communication and information technology applications (CSCITA). 2014. p. 358–63.
5. Pisharady PK, Saerbeck M. Recent methods and databases in vision-based hand gesture recognition: a review. Comput Vis Image Underst. 2015;141:152–65.
6. Bank TW. World development indicators database—population 2020 [Online].
7. Jain S, Raja KS, Mukerjee M-PA. Indian sign language character recognition. Indian Institute of Technology, Kanpur Course Project-CS365A. 2016.
8. Cohen MW, Zikri NB, Velkovich A. Recognition of continuous sign language alphabet using leap motion controller. In: 2018 11th international conference on human system interaction (HSI). 2018. p. 193–99.
9. Riofrío S, Pozo D, Rosero J, Vásquez J. Gesture recognition using dynamic time warping and kinect: a practical approach. In: 2017 international conference on information systems and computer science (INCISCOS). 2017. p. 302–08.
10. Sruthi C, Lijiya A. Signet: a deep learning based Indian sign language recognition system. In: 2019 International conference on communication and signal processing (ICCSP). 2019. p. 0596–600.
11. Lokhande P, Prajapati R, Pansare S. Data gloves for sign language recognition system. Int J Comput Appl. 2015;975:8887.
12. Chouhan T, Panse A, Voona AK, Sameer S. Smart glove with gesture recognition ability for the hearing and speech impaired. In: 2014 IEEE global humanitarian technology conference-south Asia satellite (GHTC-SAS). 2014. p. 105–10.
13. Abhishek KS, Qubeley LCF, Ho D. Glove-based hand gesture recognition sign language translator using capacitive touch sensor. In: 2016 IEEE international conference on electron devices and solid-state circuits (EDSSC). 2016. p. 334–37.
14. Das A, Yadav L, Singhal M, Sachan R, Goyal H, Taparia K et al. Smart glove for sign language communications. In: 2016 international conference on accessibility to digital world (ICADW). 2016. p. 27–31.
15. Su R, Chen X, Cao S, Zhang X. Random forest-based recognition of isolated sign language subwords using data from accelerometers and surface electromyographic sensors. Sensors. 2016;16:100.
16. Yang X, Chen X, Cao X, Wei S, Zhang X. Chinese sign language recognition based on an optimized tree-structure framework. IEEE J Biomed Health Inform. 2016;21:994–1004.
17. Sajanraj T, Beena M. Indian sign language numeral recognition using region of interest convolutional neural network. In: 2018 second international conference on inventive communication and computational technologies (ICICCT). 2018. p. 636–40.
18. Raheja J, Mishra A, Chaudhary A. Indian sign language recognition using SVM. Pattern Recognit Image Anal. 2016;26:434–41.
19. Beena M, Namboodiri MA. ASL numerals recognition from depth maps using artificial neural networks. Middle-East J Sci Res. 2017;25:1407–13.
20. Tripathi K, Baranwal N, Nandi GC. Continuous dynamic Indian sign language gesture recognition with invariant backgrounds. In: 2015 international conference on advances in computing, communications and informatics (ICACCI). 2015. p. 2211–16.
21. Nguyen T-N, Huynh H-H, Meunier J. Static hand gesture recognition using principal component analysis combined with artificial neural network. J Autom Control Eng. 2015;3:40–5.
22. Nagendraswamy H, Kumara BC, Chinmayi RL. Indian sign language recognition: an approach based on fuzzy-symbolic data. In: 2016 international conference on advances in computing, communications and informatics (ICACCI). 2016. p. 1006–13.
23. Adithya V, Vinod P, Gopalakrishnan U. Artificial neural network based method for Indian sign language recognition. In: 2013 IEEE conference on information & communication technologies. 2013. p. 1080–85.
24. Huang D-Y, Hu W-C, Chang S-H. Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. Expert Syst Appl. 2011;38:6031–42.
25. Oyedotun OK, Khashman A. Deep learning in vision-based static hand gesture recognition. Neural Comput Appl. 2017;28:3941–51.
26. Lilha H, Shivmurthy D. Evaluation of features for automated transcription of dual-handed sign language alphabets. In: 2011 international conference on image information processing. 2011. p. 1–5.
27. Otiniano-Rodrıguez K, Cámara-Chávez G, Menotti D. Hu and Zernike moments for sign language recognition. In: Proceedings of international conference on image processing, computer vision, and pattern recognition. 2012. p. 1–5.
28. Geetha M, Manjusha U. A vision based recognition of Indian sign language alphabets and numerals using b-spline approximation. Int J Comput Sci Eng. 2012;4:406.
29. Singha J, Das K. Indian sign language recognition using eigen value weighted Euclidean distance based classification technique. 2013. http://arxiv.org/abs/1303.0634.
30. Hussain I, Talukdar AK, Sarma KK. Hand gesture recognition system with real-time palm tracking. In: 2014 Annual IEEE India conference (INDICON). 2014. p. 1–6.
31. Sharma M, Pal R, Sahoo AK. Indian sign language recognition using neural networks and KNN classifiers. ARPN J Eng Appl Sci. 2014;9:1255–9.
32. Sharma K, Joshi G, Dutta M. Analysis of shape and orientation recognition capability of complex Zernike moments for signed gestures. In: 2015 2nd international conference on signal processing and integrated networks (SPIN). 2015. p. 730–35.
33. Kumar A, Thankachan K, Dominic MM. Sign language recognition. In: 2016 3rd international conference on recent advances in information technology (RAIT). 2016. p. 422–28.
34. Patil SB, Sinha G. Distinctive feature extraction for Indian sign language (ISL) gesture using scale invariant feature transform (SIFT). J Inst Eng (India) Ser B. 2017;98:19–26.

35. Loke P, Paranjpe J, Bhabal S, Kanere K. Indian sign language converter system using an android app. In: 2017 international conference of electronics, communication and aerospace technology (ICECA). 2017. p. 436–39.
36. Kumar DA, Sastry A, Kishore P, Kumar EK. Indian sign language recognition using graph matching on 3D motion captured signs. Multimed Tools Appl. 2018;77:32063–91.
37. Kishore P, Kumar DA, Sastry ACS, Kumar EK. Motionlets matching with adaptive kernels for 3-d Indian sign language recognition. IEEE Sens J. 2018;18:3327–37.
38. Pariwat T, Seresangtakul P. Thai finger-spelling sign language recognition employing PHOG and local features with KNN. Int J Adv Soft Comput Appl. 2019;11:94–107.
39. Pariwat T, Seresangtakul P. Thai finger-spelling sign language recognition using global and local features with SVM. In: 2017 9th international conference on knowledge and smart technology (KST). 2017. p. 116–20.
40. Bheda V, Radpour D. Using deep convolutional networks for gesture recognition in American sign language. 2017. http://arxiv.org/abs/1710.06836.
41. Wadhawan A, Kumar P. Deep learning-based sign language recognition system for static signs. Neural Comput Appl. 2020;32:7957–68.
42. Rao GA, Syamala K, Kishore P, Sastry A. Deep convolutional neural networks for sign language recognition. In: 2018 conference on signal processing and communication engineering systems (SPACES). 2018. p. 194–97.
43. Huang J, Zhou W, Li H, Li W. Sign language recognition using 3d convolutional neural networks. In: 2015 IEEE international conference on multimedia and expo (ICME). 2015. p. 1–6.
44. Huang J, Zhou W, Li H, Li W. Sign language recognition using real-sense. In: 2015 IEEE China summit and international conference on signal and information processing (ChinaSIP). 2015. p. 166–70.
45. Nagi J, Ducatelle F, Di Caro GA, Cireşan D, Meier U, Giusti A et al. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: 2011 IEEE international conference on signal and image processing applications (ICSIPA). 2011. p. 342–47.
46. Rioux-Maldague L, Giguere P. Sign language fingerspelling classification from depth and color images using a deep belief network. In: 2014 Canadian conference on computer and robot vision. 2014. p. 92–7.
47. Molchanov P, Gupta S, Kim K, Pulli K. Multi-sensor system for driver's hand-gesture recognition. In: 2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG). 2015. p. 1–8.
48. Tushar AK, Ashiquzzaman A, Islam MR. Faster convergence and reduction of overfitting in numerical hand sign recognition using DCNN. In: 2017 IEEE region 10 humanitarian technology conference (R10-HTC). 2017. p. 638–41.
49. Tang A, Lu K, Wang Y, Huang J, Li H. A real-time hand posture recognition system using deep neural networks. ACM Trans Intell Syst Technol (TIST). 2015;6:1–23.
50. Yang S, Zhu Q. Video-based Chinese sign language recognition using convolutional neural network. In: 2017 IEEE 9th international conference on communication software and networks (ICCSN). 2017. p. 929–34.
51. Koller O, Zargaran S, Ney H, Bowden R. Deep sign: enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. Int J Comput Vis. 2018;126:1311–25.
52. Kumar EK, Kishore P, Kumar MTK, Kumar DA. 3D sign language recognition with joint distance and angular coded color topographical descriptor on a 2–stream CNN. Neurocomputing. 2020;372:40–54.
53. Kolkur S, Kalbande D, Shimpi P, Bapat C, Jatakia J. Human skin detection using RGB, HSV and YCbCr color models. 2017. http://arxiv.org/abs/1708.02694.
54. Xu G, Xiao Y, Xie S, Zhu S. Face detection based on skin color segmentation and AdaBoost algorithm. In: 2017 IEEE 2nd advanced information technology, electronic and automation control conference (IAEAC). 2017. p. 1756–60.
55. Khaled SM, Islam MS, Rabbani MG, Tabassum MR, Gias AU, Kamal MM et al. Combinatorial color space models for skin detection in sub-continental human images. In: International visual informatics conference. 2009. p. 532–42.
56. Gonzalez RC, Woods RE, Masters BR. Digital image processing third edition. J Biomed Opt. 2008;14: 029901.
57. Islalm MS, Rahman MM, Rahman MH, Arifuzzaman M, Sassi R, Aktaruzzaman M. Recognition Bangla sign language using convolutional neural network. In: 2019 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT). 2019. p. 1–6.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com