Array 14 (2022) 100141

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array

Indian Sign Language recognition system using SURF with SVM and CNN

Shagun Katoch [a], Varsha Singh [b,*], Uma Shanker Tiwary [b]

[a] Computer Science, National Institute of Technology, Hamirpur, India
[b] Department of Information Technology, Indian Institute of Information Technology, Allahabad, India

ARTICLE INFO

Keywords:
Hand sign recognition
Indian sign language (ISL)
Bag of visual words (BOVW)
SURF features
SVM
CNN
Pyttsx3
Google speech API

ABSTRACT

Hand signs are an effective form of human-to-human communication that has a number of possible applications. Being a natural means of interaction, they are commonly used for communication purposes by speech impaired people worldwide. In fact, about one percent of the Indian population belongs to this category. This is the key reason why it would have a huge beneficial effect on these individuals to incorporate a framework that would understand Indian Sign Language. In this paper, we present a technique that uses the Bag of Visual Words model (BOVW) to recognize Indian sign language alphabets (A-Z) and digits (0–9) in a live video stream and output the predicted labels in the form of text as well as speech. Segmentation is done based on skin colour as well as background subtraction. SURF (Speeded Up Robust Features) features have been extracted from the images and histograms are generated to map the signs with corresponding labels. The Support Vector Machine (SVM) and Convolutional Neural Networks (CNN) are used for classification. An interactive Graphical User Interface (GUI) is also developed for easy access.

1. Introduction

Communication has always played a vital role in human life. The calibre to interact with others and express ourselves is a basic human necessity. However, based on our upbringing, education, society, and so on, our perspective and the way we communicate with others can differ to a great extent from those around us. In addition to this, ensuring that we are understood in the way we intend, plays a very important role.

Despite this fact, normal human beings do not have much difficulty interacting with each other and can express themselves easily through speech, gestures, body language, reading, writing, speech being widely used among them. However, people affected by speech impairment rely only on sign language, which makes it more difficult for them to communicate with the remainder of the majority. This implies a requirement for sign language recognizers which can recognize and convert sign language into spoken or written language and vice versa. Such identifiers, however, are limited, costly, and cumbersome to use. Now, researchers from different countries are working on these sign language recognizers, which is the main reason behind the development of automatic sign language recognition systems.

Notwithstanding that India is a diverse country having nearly 17.7% of the world's population residing here, but very limited work has been done in this research area, which is quite contradictory as compared to the other countries [1–3]. Delayed standardisation can be attributed to the evidence for this. Indian Sign Language studies began in India in 1978. But since no standard type of ISL existed, its use was restricted to short-term courses only. In addition, the gestures used in most of the deaf schools varied significantly from each other and nearly 5% of the total deaf people attended these schools. It was in 2003 when ISL got standardized and grabbed the attention of researchers [4].

Indian Sign Language (ISL) involves both static and dynamic signs, single as well as double-handed signs, and in different regions of India, there are many signs for the same alphabet. It makes it very difficult to introduce such a scheme. In addition, no standard dataset is available. All these things manifest the complexity of Indian sign language.

Recently, researchers have started exploring this area. There are mainly two different approaches widely used in sign language recognition: Sensor-based approach and Vision-based approach [5]. The sensor-based approach uses gloves or other instruments that recognize finger gestures and translate them into equivalent electrical signals for sign determination, whereas web cameras are used to capture video or images in a vision-based approach. Due to its no specialised hardware requirement, the vision-based gesture recognition offers the advantage of being spontaneous, and is favoured by the signers [6]. However, hand segmentation in a complex setting, which plays an important role in identification. A framework that can overcome this problem is therefore

* Corresponding author.
E-mail address: varshagaur@gmail.com (V. Singh).

https://doi.org/10.1016/j.array.2022.100141
Received 11 September 2021; Received in revised form 10 January 2022; Accepted 29 March 2022
Available online 14 April 2022

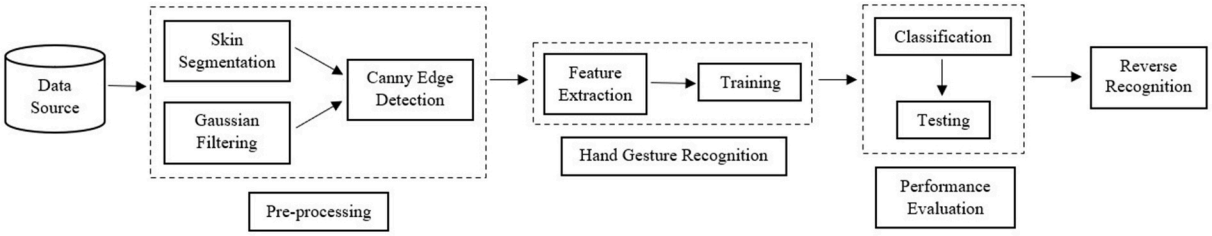

Fig. 1. Flow Diagram of the proposed approach.

suggested.

The advances in machine learning and deep learning technology are providing new methods and algorithms for recognizing Indian sign language alphabets efficiently, accurately and inexpensively. The end-to-end auto run of these models overcomes the highly subjective and inconsistent limitations of traditional methods, improving accuracy and efficiency of the results.

In this work, the authors present a methodology to build a large, assorted and robust real-time alphabets (A-Z) and digits (0–9) recognition system for Indian Sign Language. Instead of using high-end technologies like gloves or the Kinect, here authors have recognised signs from images (which are accessed from a webcam). The accuracy obtained in the result is also discussed in this paper. The real-time, accurate and efficient judgement on ISL sign recognition is required to bridge the communication gap between the abled and the hearing or speech impaired people.

2. Related works

Depending on the nature of sign language and the signs, different authors have employed different methodologies.

J. Singha et al. [7] proposed a method for real time recognition where Eigen value-weighted Euclidean distance was used to classify signs. *P. Kishore* et al. [8] proposed a system by finding active contours from boundary edge map using Artificial Neural Network (ANN) to classify the signs. Another approach used the Viola Jones algorithm with LBP functions for hand gesture recognition in a real-time environment [9]. It had the advantage of requiring less processing capacity to detect the movements. Segmentation is the primary and one of the most important steps in hand processing, in general Otsu's algorithm gave a fairly high rate of accuracy [10]. In an attempt [11], moving block distance parameterization method was used to skip the initialization and segmentation steps. High precision static symbols and 33 basic word units were used.

Most of these works were based on pattern recognition, feature extraction, and so on [12]. However, in most of the cases, a system with single feature is not enough. Therefore, hybrid approaches were introduced to solve this problem. For eg, *A. Nandy* et al. [13] used hybrid approaches with K-Nearest Neighbor (KNN) and Euclidean distance to classify gestures from orientated histogram features. The limitation of this approach was the poor performance in case of similar gestures. *K Manjushree* et al. [14] used single handed sign classification with histogram of oriented gradients and feature matching. *S. Kanade* et al. [15] designed a system with custom dataset using PCA features and SVM, and obtained good accuracy. *A. Sahoo* [16] proposed ISL recognition for both single and double handed character signs. *Geetha. M* et al. [17] used B-Spline approximation for the shape matching of static gestures of ISL alphabets & numerals. In Ref. [18], a method was proposed to classify word symbols using the Neuro-Fuzzy approach and natural language processing (NLP) technology to display the final word. *Q. Chen* et al. [19] proposed a method to recognize hand gestures using haar-like features and AdaBoost algorithm. They also described random context-free grammars to fully recognize gestures. The combination of PCA and the local coordinate system produced high calculation accuracy and was found to be superior to the method based on condensation algorithm [20].

However, for real-time systems, researchers needed a faster way to solve this problem. The advancements in Deep Learning technologies have enabled automation of image recognition using various image recognition models. For e.g., Convolutional neural networks have made great strides in the field of deep learning in recent years [21,22]. *G. Jayadeep* et al. [23] used a CNN (Convolutional Neural Network) to extract image features, LSTM (Long Short Term Memory) to classify these gestures and translate them into text. *Bin* et al. [24] proposed the InceptionV3 model to use depth sensors to identify static signs. It eliminated the steps of gesture segmentation and feature extraction. In Ref. [25], *Vivek Bheda* et al. proposed a methodology for using a mini-batch supervised learning method of stochastic gradient descent to classify images for each digit (0–9) and American Sign Language letter using deep convolutional neural networks.

Looking into these works, authors were motivated towards creating a custom dataset and an algorithm that would work fully on that dataset without affecting the accuracy of the video detection. We decided to use SURF features because it would decrease the time of measurement and make the system invariant to rotation. The authors of the paper have also addressed the problem of background dependency so that the system can be used anywhere and not only in controlled environments.

3. Proposed work

Sign language recognition requires efficient and robust data to design a highly accurate system that would be helpful for real-time users. Here authors have used the custom-built dataset to solve the sign detection and classification problem. The data flows at different stages for sign language recognition viz. Dataset, Image Acquisition, Data Pre-processing, Feature Extraction, Sign Classification is shown in Fig. 1.

3.1. Dataset collection

It is a very crucial part of the research works in all the arenas as it is fundamental to foster the development of any machine or deep learning model. However, it is full of challenges. During data collection, the biggest challenge we faced was that there were no standard datasets for Indian sign language available. Therefore, as part of this project, we attempted to manually construct a dataset that could help us overcome this problem.

First of all, we captured the videos using a webcam where various signs were taken into account. 26 different alphabets (A-Z) and 10 numeric signs (0–9) were considered from 3 persons. For the quality of the pictures and elimination of the background noises, the position of the camera is very critical. To add variations in the dataset, two options were used for capturing the images. The first one is the default method, which performs the skin segmentation on the image and can be used with a plain colour background.

In the second method, we have used the concept of running averages, in which some of the initial frames are considered as background and any new object after the initial frames is considered as foreground,

Fig. 2. Isl signs.

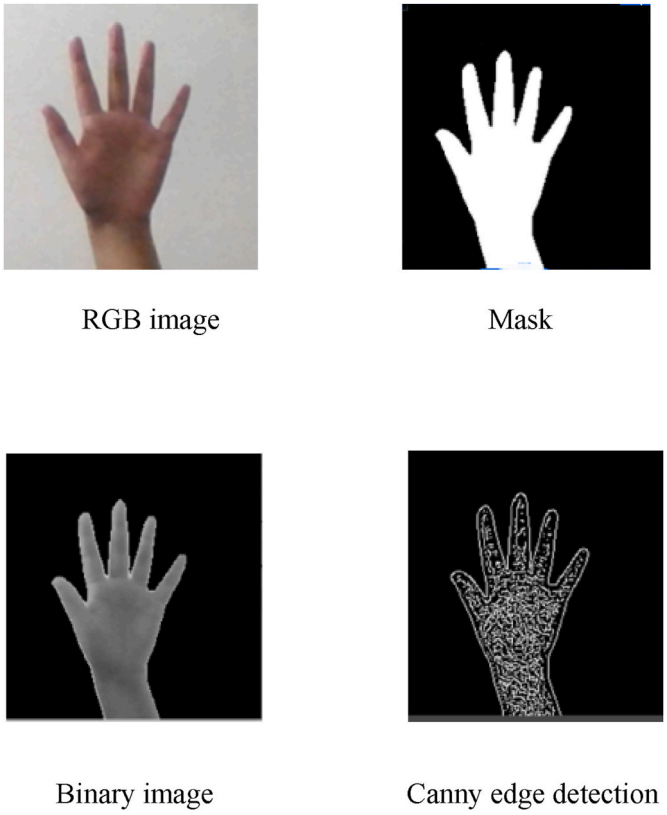

Fig. 3. Preprocessing steps.

thereby making the extraction process easier. The dataset was created by taking into account both of these approaches in order for the model to perform well in diverse scenarios.

The signs obtained from the live video were converted into frames, which were further extracted using a pixel value threshold. The produced frames had a resolution of 250*250 so that less computational power is required for pre-processing. Each sign folder contained around 1000 images of each sign. Hence the total number of images in the dataset were 36,000 for both image acquisition methods. The signs involved the use of a single hand as well as of both hands. The images were captured in different rotations and stored in grayscale format with. jpg extension. The dataset images can be seen in Fig. 2 (see Fig. 3).

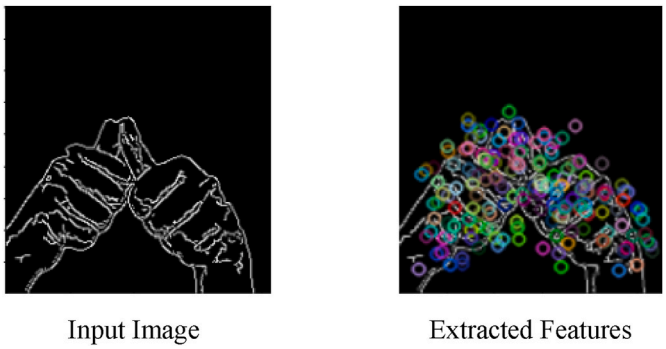

Fig. 4. SURF feature extraction.

3.2. Preprocessing

The image is made ready for feature detection and extraction in this phase. To preserve uniformity of scale, the dimensions of all the images are kept the same.

In the default option, the captured video frame is converted into HSV colour space for the images acquired with the plain background. As the hue colour of the skin is different from that of the background, it gets extracted easily. An experimental threshold is then applied to the frame that calculates hue and filters out the skin coloured pixels from the image. Further, the image is binarized, blurring is done to remove noises and maximum contour is obtained from the result assuming that the contour with the largest area represents the hands. Errors are further removed by applying the median filter and morphological operations.

In the second method for the images with the running background, the first 30 frames are considered as background and for the remaining frames the absolute difference is calculated between the additive sum of those 30 frames and the new frame, which gives us the foreground region of the current frame. The images are first converted into grayscale and then Gaussian filter is applied. For hand segmentation, a mask is created by extracting the maximum connected region in the foreground assuming it to be the hand. Noises are further removed by applying morphological operations like erosion and dilation.

After this, the canny function is used in which the gradient of each pixel calculates the edge strength and direction of the images. Compared to the original image, this results in a shift of intensity and the edge is detected easily. The pre-processed images from both the options are shuffled to add variation in the dataset.

3.3. Feature extraction

This phase involves building a Bag of Visual Words (BOVW) which includes feature extraction, clustering of features, codebook construction for the model, and generation of histograms.

The Bag of Visual Words (BOVW) is a widely used image classification model whose definition is adapted from data retrieval and NLP'S (Natural Language Processing) Bag of Words (BOW) [26]. In this, we count the number of times each word appears in a text, use each word's frequency to get the keywords and produce a histogram of frequency from it. This idea is changed in such a way that instead of words, we use the image features as words. To construct a vocabulary where each

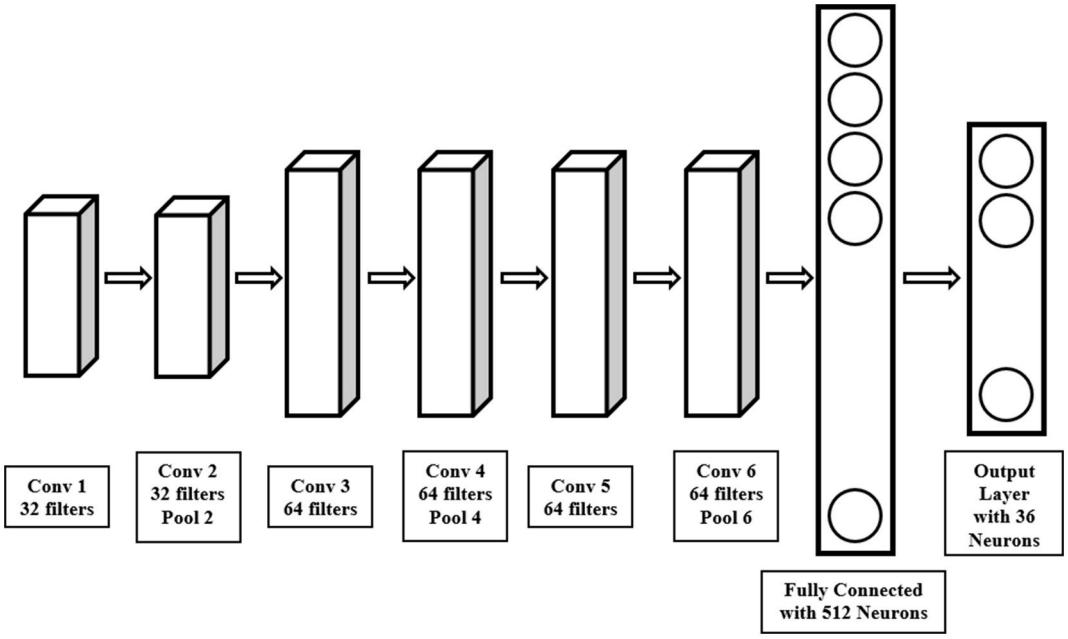

Fig. 5. CNN architecture.

Table 1
Class wise accuracy table.

Label	SVM (%)	CNN (%)	Label	SVM (%)	CNN (%)	Label	SVM (%)	CNN (%)
0	100	100	**C**	100	100	**O**	99	99
1	99	100	**D**	100	100	**P**	100	100
2	98	100	**E**	96	97	**Q**	100	100
3	96	98	**F**	95	100	**R**	98	98
4	100	100	**G**	98	100	**S**	100	100
5	100	99	**H**	100	100	**T**	99	100
6	100	100	**I**	98	100	**U**	99	100
7	100	100	**J**	100	100	**V**	100	100
8	98	100	**K**	100	100	**W**	99	100
9	98	100	**L**	99	100	**X**	100	99
A	100	100	**M**	100	99	**Y**	99	100
B	100	100	**N**	99	100	**Z**	100	100

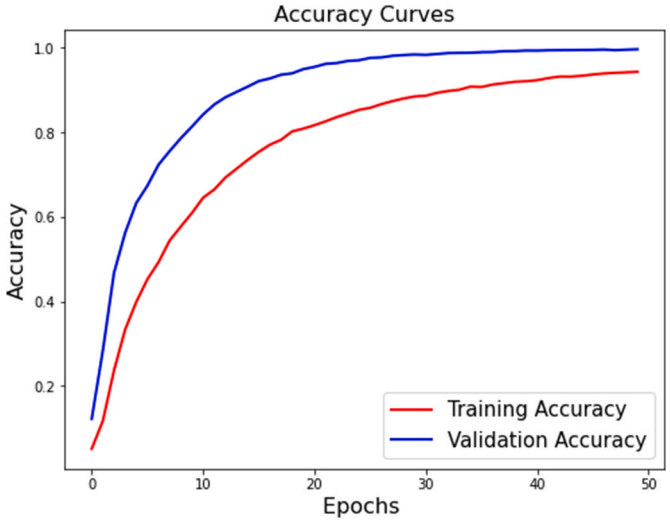

Fig. 6. Accuracy graph of CNN.

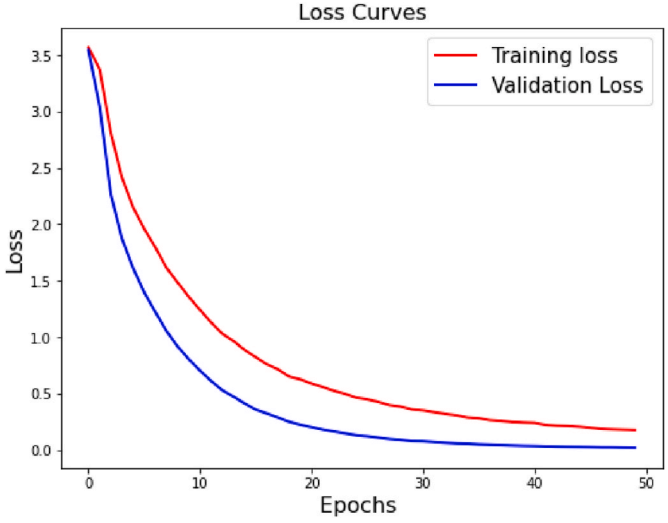

Fig. 7. Loss graph of CNN.

Table 2
Accuracy table.

SVM	CNN
99.17%	99.64%

Table 3
Performance metrics table.

Measure	SVM	CNN
Precision	99.09	99.57
Recall	99.02	99.57
F1 Score	99.09	99.57

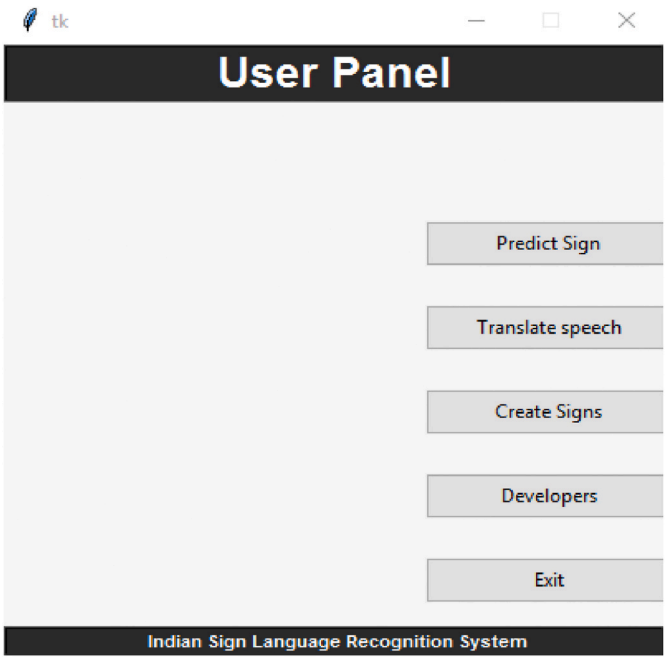

Fig. 8. System GUI.

image is represented as a frequency histogram of characteristics obtained, the image descriptors and key points are used. Later on, the category of another comparable image can be predicted from this frequency histogram.

As discussed, the first step in building a bag of visual words (BOVW) is to extract descriptors from each image in the dataset. The descriptor is a 64-member vector for each interest point in the execution used which defines the distribution of the intensity material within the neighborhood of the interest point. For this, SURF (Speeded Up Robust Features) [27] is used which is a local feature detector and descriptor. We have used SURF as they are robust against rotation, variance, point of view occlusion, and provide operators with box filters for fast computation.

An image is represented as a set of image descriptors given by SURF as Eq (1).

$$Im = \{d_1, d_2, d_3, \ldots\ldots\ldots\ldots d_n\} \tag{1}$$

where d_i is the colour, shape, etc. of the hands. and n denotes the total image descriptors. Fig. 4 shows the extracted SURF features when a binary image representing sign A is passed to the SURF.

The next step in the extraction of features is to cluster all the features that are obtained after the SURF is applied. This is done to group similar features so that it is possible to use the core and cluster them as the dictionary's visual keyword. The clustering can be performed using the K-means algorithm, but we have used mini batch K-Means as the data is very large. It is comparable to K-means, but is better in terms of processing time and memory use. It utilises small random batches of fixed-size data at a time, thus reducing the need to have all the data in the memory at the same time. A new random sample from the dataset is obtained in each iteration and used to update the clusters, which is repeated until convergence. We have a value of k as 180 for this purpose.

For codebook generation, the resulting cluster centres (i.e., centroids) are treated as our code vectors. A codebook is used for quantizing features where it takes a feature vector as input and maps it to the index of the nearest code vector. The constructed vocabulary can be represented as:

$$v = \{w_1, w_2, w_3, \ldots\ldots\ldots\ldots w_k\} \tag{2}$$

where k is the total number of clusters i.e. 180. The mapping of each descriptor to the nearest visual word is done according to the Eq (3).

$$w(d) = argmin\ Dist(w, d) \tag{3}$$

where w (d_i) depicts the visual word assigned to the ith descriptor and Dist (w,d_i) represents the distance between the visual word w and descriptor d_i.

The last step is the generation of the histograms for all the images which is done by calculating the frequency of occurrence of each visual word in an image. The count of bins in the histogram is equal to the total number of visual words in the dictionary i.e. k and is represented by Eq (4).

$$bin_i = C(D_i)$$
$$\text{Where } D_i = \{d_j, j\varepsilon 1, \ldots\ldots n | w(d_j) = w_i\} \tag{4}$$

Here D_i is the set of all the descry iptors corresponding to a particular visual word w_i in the image and C (D_i) is the cardinality representing the count of the elements in set D_i. For every visual word in the image, this is repeated to obtain final histograms that are then passed for recognition to the classifier along with their respective labels.

3.4. Classification

We proceed to the classification stage once the feature detection and extraction process is finished. It involves classification using a Support Vector Machine (SVM) and using a Convolutional Neural Network (CNN).

3.4.1. Support Vector Machine

The Support Vector Machine (SVM) is a supervised model that can solve both linear and non-linear problems for classification and regression problems. It operates on the idea of decision planes that specify boundaries for decisions.

For this classification, we have used SVM with a linear kernel. We have passed the histograms of visual words to the SVM as feature vectors for the classification and recognition of ISL signs. The training is done using a total of 28,800 images. After the training is completed, the performance of the classifier is checked on the testing set which has a total of 7236 images, and its performance is evaluated on various parameters like accuracy, precision, recall, etc.

3.4.2. Convolutional neural networks

CNNs are functional extraction models inspired by the human brain's visual cortex. CNNs compare images piece by piece where a filter map slides over the local patches of the image. Such pieces are called features, and they compare two images by finding approximately the same features at approximately the same locations. CNNs have a better ability to see images and classify them than other neural networks.

Our general architecture is a fairly common CNN architecture, consisting of multiple convolutional and dense layers. Each CNN is 3 layers deep. The architecture starts with a group of 2 convolutional layers which have 32 filters with a window size of 3 × 3 followed by a max-pool layer and a dropout layer. It is then followed by another group of 2 convolutional layers with 64 filters, a max pooling layer and dropout layer. Further, there are another 2 convolutional layers with 64 filters and a max pooling layer and at the end, there is a fully connected hidden layer with 512 neurons of the ReLU activation function and an output layer of the softmax activation function. The first convolution layer takes an input image of size (100,100) whereas the final output layer consists of 36 neurons corresponding to each category of the ISL signs. The architecture diagram for the same is shown in Fig. 5.

3.5. Output sign

Predicted class labels which are returned as numeric vectors are automatically translated by the system in the form of text and speech.

Fig. 9. Snapshots of the proposed system.

This is done to provide better communication and ease to the user. Once the label is identified by the classifier, it is passed to a dictionary as a key which returns the corresponding sign as value. This is then shown to the user. For text to speech conversion, python text to speech module, Pyttsx3 is used. As it causes a delay in the live video stream by making the frames process at a very slow rate, threading is performed. Due to this, the prediction of signs and the translation of text to speech can be achieved at the same time. This ensures that the sound is played continuously, without any disturbance.

3.6. Reverse recognition

The reverse process is essential in a sign language recognition system to provide a dual mode of communication between the speech impaired and hearing majority [28]. We have implemented this mode of communication in our system. Here text (English alphabets) is given as the input in the form of speech by the user, where it is mapped onto the labels and corresponding signs (images stored in database) are displayed to the user in a sequence. The speech recognition is done using the Google Speech API.

4. Experiment and results

The dataset is divided into 2 sets. The training set consists of 80% of the total data and the remaining 20% is used as testing means. Both the classifiers (SVM and CNN) have given high accuracy on the images, but

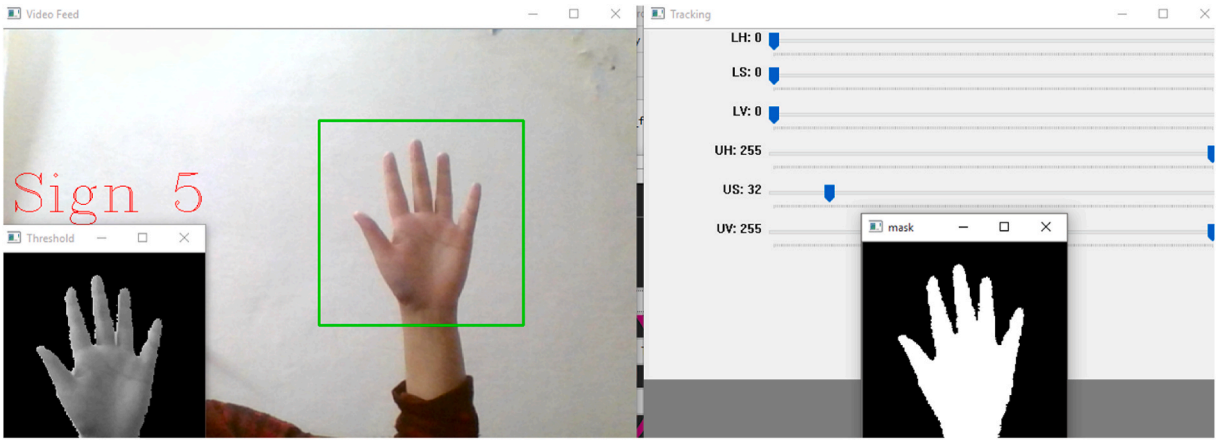

Fig. 10. Real time testing.

Table 4
Comparison table.

Dataset	Models Used	Accuracy
Single Handed Dataset	SVM + HOG features	97.1%
Single + Double Handed dataset with variations	SVM	99.17%
	CNN	99.64%

CNN has performed better with a lesser no. of features. The system is trained to recognize 36 signs (26 alphabets and 10 numerals). Current results are promising, keeping in mind that few improvements could provide better results.

4.1. SVM performance

SVM has given an accuracy of 99.14% on the test data. The calculated values of precision and recall of alphabets and digits classified using SVM show an overall accuracy of 99%. The class wise accuracy can be seen in Table 1.

4.2. CNN performance

Using CNN, we have observed an overall accuracy of 94% on the training set on the last epoch whereas a testing accuracy greater than 99%. The total epochs are 50. We have trained our model with a categorical cross entropy loss function and softmax function as the activation function, which has given a training loss of 0.1748 on the last epoch and a testing loss of 0.0184. The class wise accuracy can be seen in Table 1. The accuracy graph for our experiment is shown in Fig. 6 whereas the loss graph can be seen in Fig. 7.

4.3. Quantitative analysis

4.3.1. Accuracy

Accuracy is the most innate performance measure and is simply a ratio of correctly predicted observations to the total observations. In terms of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), the formula of the accuracy can be written as-

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

The comparison of test accuracies of both the classifiers is done in Table 2. Both the classifiers have performed well on the test data with accuracies greater than 99%.

4.3.2. Performance metrics

Precision is the ratio of correctly predicted positive observations to the total observations which are positive. The recall is the ratio of correctly predicted positive labels to the total no. of labels which are positive whereas the F1 score is the weighted average of precision and recall. The results can be seen in Table 3.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2*(Recall*Precision)}{Recall + Precision}$$

4.4. Real Time Testing

An interactive GUI is designed for the system (Fig. 8.) users with a fully functioning sign in and sign up system using Tkinter. The users can predict signs based on the model trained using our dataset by clicking on the predict sign button or can create their database using the create signs button. An option for speech to sign conversion is also provided. The screenshots for real-time video testing are shown in Fig. 9 (see Fig. 10).

Here user is given two options for capturing sign input images with plain background and without plain background.

The suggested method of recognition based on the SURF characteristics has the benefit of fast computations and is robust against rotation, orientation, etc. Being a user-independent model, it is also capable of solving the problem of background dependency with the condition of keeping the camera still. However, with plain backgrounds, it can be used freely. The previous works have either used a plain background or have used complex background under some controlled environments. The recognition accuracy is close to 0.94 for most of the models. However, most of the signs reported use single hands or simple hand movements [29]. Our model is capable of recognizing double hand signs and machine translation of visual information into text or speech with high accuracy of 99%. It is a basic step further in removing some of the drawbacks by helping researchers to use this approach with a standard dataset.

In comparison to Ref. [14], where SVM and HOG features are used to build a sign language recognition system with data of single hand signs only. Here we have used single and double handed custom dataset with 2 different data collection ways. On training with SVM and CNN our model has performed significantly better in terms of accuracy in Table 4.

5. Conclusion and future work

A novel approach to classify and recognize Indian sign language signs (A-Z) and (0–9) using the SVM and CNN is presented in the paper.

The main goal of our work is to provide a more real-time recognition utility so that the system can be used anywhere. It is achieved by constructing a custom data set, making the system invariant to rotation and solving the background dependency problem. The system is successfully trained on all 36 ISL static alphabets and digits with an accuracy of 99%. In future, the dataset can be expanded by adding more signs from different language of various countries, thereby achieving a more effective framework for real-time applications. The method can be extended for the formation of simple words and expressions for both continuous and isolated recognition tasks. The secret to true real-time applications is enhancing the response time.

Credit author statement

Shagun Katoch: Methodology, Software, Investigation, Data curation, Writing – original draft. **Varsha Singh**: Conceptualization, Visualization, Validation, Formal analysis, Resources, Project Management, Writing- Reviewing and Editing. **Uma Shanker Tiwary**: 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

[1] Bantupalli Kshitij, Xie Ying. American sign language recognition using machine learning and computer vision. Master of Science in Computer Science Theses 2019; 21.

[2] Shadman Shahriar, Ashraf Siddiquee, Tanveerul Islam, Abesh Ghosh, Rajat Chakraborty, Asir Intisar Khan, Celia Shahnaz and Shaikh Anowarul Fattah. Real-time American sign language recognition using skin segmentation and image category classification with convolutional neural network and deep learning. In TENCON, IEEE Region 10 International Conference.

[3] Shivashankara S, Srinath S. A comparative study of various techniques and outcomes of recognizing American sign language: a review. In: International Journal of Scientific Research Engineering & Technology (IJSRET); 2017. *ISSN 2278 – 0882. 6(9)*.

[4] Viswanathan Daleesha M, Idicula Sumam Mary. Recent developments in Indian sign language recognition: an analysis. Int J Comput Sci Inf Technol 2015;6(1): 289–93.

[5] Nair Anuja V, Bindu V. A review on Indian sign language recognition. Int J Comput Appl 2013;73(22).

[6] Athira K, Sruthi CJ, Lijiya A. A signer independent sign language recognition with Co-articulation elimination from live videos: an Indian scenario. J King Saud Univ Comput Inf Sci 2022;34(3):771–8.

[7] Singha J, Das K. Recognition of Indian sign language in live video. Int J Comput Appl 2013;70(19):17–22.

[8] Kishore PVV, Kumar DA. Optical flow hand tracking and active contour hand shape features for continuous sign language recognition with artificial neural networks. In: IEEE 6th international conference on advanced Computing; 2016.

[9] Swamy Shanmukha, Chethan MP, Gatwadi Mahantesh. Indian sign language interpreter with android implementation. Int J Comput Appl 2014:975–8887.

[10] Agrawal SC, Jalal AS, Bhatnagar C, Ieee. Recognition of Indian sign language using feature Fusion. 2012.

[11] Aviles-Arriaga HH, Sucar-Succar LE, Mendoza-Duran CE, Pineda-Cortes LA. A comparison of dynamic naive bayesian classifiers and hidden markov models for gesture recognition. J Appl Res Technol 2011;9:81–102.

[12] Rokade Yogeshwar I, , et alJadav Prashant M. Indian sign language recognition system. In: International Journal of Engineering and Technology July; 2017.

[13] Nandy Anup, Prasad Jay Shankar, Mondal Soumik, Chakraborty Pavan, Nandi Gora Chand. Recognition of isolated Indian sign language gestures in real time. In: International Conference on Business Administration and Information Processing; 2010.

[14] Manjushree K, Divyashree. Gesture recognition for Indian sign language using HOG and SVM. International Research Journal of Engineering and Technology 2019;6 (7).

[15] Kanade Sudhir S, Deshpande Padmanabh D. Recognition of Indian sign language using SVM classifier. International Journal of Scientific Research and Development 2018;2(3).

[16] Sahoo Ashok Kumar, Kumar Ravulakollu Kiran. Vision based Indian sign language character recognition. J Theor Appl Inf Technol 2014;67(3).

[17] Geetha M, Manjusha UC. A vision based Recognition of Indian sign language Alphabets and Numerals Using B-Spline Approximation *International Journal on Computer Science and Engineering (IJCSE)*. 2012.

[18] Bhavsar Hemina, Trivedi Jeegar. Indian sign language recognition using framework of skin color detection, Viola- Jones algorithm, correlation-coefficient technique and distance based neuro-fuzzy classification approach. Emerging Technology Trends in Electronics, Communication and Networking 2020;1214: 235–43.

[19] Chen Q, Georganas ND, Petriu EM. Hand gesture recognition using Haar-like features and a stochastic context-free grammar. IEEE Trans Instrum Meas 2008;57 (8):1562–71. https://doi.org/10.1109/TIM.2008.922070.

[20] Dan L, Ohya J. Study of recognizing multiple persons' complicated hand gestures from the video sequence acquired by a moving camera. In: Rogowitz BE, Pappas TN, editors. Human Vision and Electronic Imaging Xv, vol. 7527; 2010.

[21] Sahoo Ashok K, Mishra Gouri Sankar, Kumar Ravulakollu Kiran. sign language recognition: state of the art. In: ARPN Journal of Engineering and Applied Sciences; 2014.

[22] Bachani Shailesh, Dixit Shubham, Chadha Rohin, Bagul Prof Avinash. sign language recognition using neural network. International Research Journal of Engineering and Technology (IRJET) 2020;7(4).

[23] Jayadeep G, Vishnupriya NV, Venugopal V, Vishnu S, Geetha M. Mudra: convolutional neural network based Indian sign language translator for banks. In: 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020; 2020. p. 1228–32.

[24] Xie B, He Xy, Li Y. RGB-D static gesture recognition based on convolutional neural network. J Eng 2018;2018(16):1515–20.

[25] Vivek Bheda and N. Dianna Radpour. Using Deep Convolutional Net-works for Gesture Recognition in American sign language. In Department of Computer Science Department of Linguistics State University of New York at Buffalo.

[26] Sivic J, Zisserman A. Video Google: a text retrieval approach to object matching in videos. In: null. IEEE; 2003. p. 1470.

[27] Bay Herbert, et al. SURF: speeded up robust features. In: European Conference on Computer Vision (ECCV); 2006.

[28] Tripathi Kumud, Baranwal Neha, Nandi GC. Continuous Indian sign language gesture recognition and sentence formation. In: Eleventh International Multi-Conference on Information Processing (IMCIP); 2015.

[29] Patel Ravi. A review on image based Indian sign language recognition. In: International Journal of Innovative Research in Computer and Communication Engineering; 2018.

Further reading

[30] Singha J, Das K. Hand gesture recognition based on karhunen-loeve transform. In: Mobile and Embedded Technology International Conference (MECON); 2013. p. 365–71.

[31] Lal Raheja Jagdish, Mishra Abhijit, Chaudhary Ankit. Indian sign language recognition using SVM. Pattern Recogn Image Anal 2016.

[32] Dixit Karishma, Singh Jalal Anand. Automatic Indian sign language recognition system. In: Advance Computing Conference (IACC). IEEE International; 2013.

[33] Manikandan K, Patidar Ayush, Walia Pallav, Roy Aneek Barman. Hand gesture detection and conversion to speech and text. In: International Conference on Innovations and Discoveries in Science, Engineering and Technology(ICIDSET); 2018.

[34] Mali Deepali, Limkar Nitin, Mali Satish. Indian sign language recognition using SVM classifier. In: Proceedings of International Conference on Communication and Information Processing (ICCIP); 2019.

[35] Ekbote Juhi, Joshi Mahasweta. Indian sign language recognition using ANN and SVM classifiers. In: International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS); 2017.

[36] Sruthi CJ, Lijiya. A. Signet: a deep learning based Indian sign language recognition system. In: International Conference on Communication and Signal Processing (ICCSP); 2019.

[37] Sarkar Alakesh, Kumar Talukdar Anjan, Kumar Sarma Kandarpa. CNN-based real-time Indian sign language recognition system. In: International Conference on Advances in Computational Intelligence and Informatics, ICACII: Advances in Computational Intelligence and Informatics; 2019. p. 71–9.

[38] Reshna S, Sajeena A, Jayaraju M. Recognition of static hand gestures of Indian sign language using CNN. AIP Conf Proc 2020;2222(30012).

[39] Bhattacharya Abhiruchi, Zope Vidya, Kumbhar Kasturi, Borwankar Padmaja, Mendes Ariscia. Classification of sign language gestures using machine learning. International Journal of Advanced Research in Computer and Communication Engineering 2019;8(12).

[40] Tolentino Lean Karlo S, Ronnie O, Juan Serfa, Thio-ac August C, Pamahoy Maria Abigail B, Forteza Joni Rose R, Garcia Xavier Jet O. Static sign language recognition using deep learning. International Journal of Machine Learning and Computing 2019;9(6).

[41] Li Dongxu, Rodriguez Opazo Cristian, Yu Xin, Li Hongdong. Word-level deep sign language recognition from video: a new large-scale dataset and methods comparison. In: Winter *Conference on Applications of Computer Vision*; 2020.

[42] Kishore PVV, Anil Kumar D, Sastry ASCS, Kiran Kumar E. Motionlets matching with adaptive kernels for 3D Indian sign language recognition. IEEE Sensor J 2018; (8).

[43] Joshi Garima, Renu Vig, Singh Sukhwinder. DCA-based unimodal feature-level fusion of orthogonal moments for Indian sign language dataset. IET Comput Vis 2018;(5).

[44] Mittal Anshul, Kumar Pradeep, Roy Partha Pratim, Raman Balasubramanian, Chaudhuri Bidyut B. A modified LSTM model for continuous sign language recognition using leap motion. IEEE Sensor J 2019;(16).

[45] De Souza Cesar Roberto, Pizzolato Ednaldo Brigante. sign language recognition with Support vector machines and hidden conditional random fields: going from

fingerspelling to natural articulated words. In: 9th international conference, Machine learning and data mining in pattern recognition, New York, USA. Proceedings; 2013. p. 84–98.
[46] Gangrade Jayesh, Bharti Jyoti, Mulye Anchit. Recognition of Indian sign language using ORB with bag of visual words by Kinect sensor. IETE J Res 2020:1–15.
[47] Tolentino Lean Karlo S, Ronnie O, Juan Serfa, Thio-ac August C, Pamahoy Maria Abigail B, Forteza Joni Rose R, Garcia Xavier Jet O. Static sign language recognition using deep learning. International Journal of Machine Learning and Computing 2019;9(6).
[48] Uchil AP, Jha S, Sudha BG. Vision based deep learning approach for dynamic Indian sign language recognition in healthcare. In: Smys S, Tavares J, Balas V, Iliyasu A, editors. Computational Vision and Bio-Inspired Computing. ICCVBIC 2019. Advances in Intelligent systems and computing, vol. 1108. Cham: Springer; 2020.
[49] Bhagat NK, Vishnusai Y, Rathna GN. Indian sign language gesture recognition using image processing and deep learning. In: Digital Image Computing: Techniques and Applications (DICTA); 2019. p. 1–8. Perth, Australia.
[50] Dutta KK, Bellary SAS. Machine learning techniques for Indian sign language recognition. In: International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC); 2017. p. 333–6. Mysore.
[51] Das A, Gawde S, Suratwala K, Kalbande D. sign language recognition using deep learning on custom processed static gesture images. In: International Conference on Smart City and Emerging Technology (ICSCET); 2018. p. 1–6. Mumbai.