Hybrid SIFT Feature Extraction Approach for Indian Sign Language Recognition System Based on CNN

Abhishek Dudhal, Heramb Mathkar, Abhishek Jain, Omkar Kadam and Mahesh Shirole

Abstract Indian sign language (ISL) is one of the most used sign languages in the Indian subcontinent. This research aims at developing a simple Indian sign language recognition system based on convolutional neural network (CNN). The proposed system needs webcam and laptop and hence can be used anywhere. CNN is used for image classification. Scale invariant feature transformation (SIFT) is hybridized with adaptive thresholding and Gaussian blur image smoothing for feature extraction. Due to unavailability of ISL dataset, a dataset of 5000 images, 100 images each for 50 gestures, has been created. The system is implemented and tested using python-based library Keras. The proposed CNN with hybrid SIFT implementation achieves 92.78% accuracy, whereas the accuracy of 91.84% was achieved for CNN with adaptive thresholding.

1 Introduction

Sign language is an efficient and natural way of communication for the hearing-impaired and verbally impaired community. Around 2.7 million of India's total population is hearing-impaired and 98% of this population uses Indian sign language as the primary language for communication. However, hearing-impaired community

A. Dudhal (✉) · H. Mathkar · A. Jain · O. Kadam · M. Shirole
Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India
e-mail: addudhal_b14@it.vjti.ac.in

H. Mathkar
e-mail: hkmathkar_b14@it.vjti.ac.in

A. Jain
e-mail: amjain_b14@it.vjti.ac.in

O. Kadam
e-mail: oskadam_b14@it.vjti.ac.in

M. Shirole
e-mail: mrshirole@it.vjti.ac.in

D. Pandian et al. (eds.), *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, Lecture Notes in Computational Vision and Biomechanics 30,
https://doi.org/10.1007/978-3-030-00665-5_72

experiences difficulties while communicating with people who lack knowledge of sign language. A human translator is required to translate sign language into speech. This solution is translator dependent and it fails in absence of a translator. A differently abled person can be empowered with a computer-based system for translation. A computer-based system can be trained to recognize ISL efficiently, thereby providing high availability, ease of use, and efficient navigation and trade to differently abled persons.

For Indian sign language recognition, the research work undertaken ranges from introducing a smart glove to monitor movements of fingers and hand to image processing that analyzes hand gestures captured. Heera et al. [1] introduced sensors incorporated glove-based approach to convert ISL into speech with the help of Bluetooth module and an Android smartphone. Ekbote et al. [2] proposed a method for ISL recognition using artificial neural network (ANN) [3] and support vector machine (SVM) [4] classifiers. Authors have used a self-created dataset for ISL, 0–9 numbers, which is very limited. Histogram of oriented gradients (HOG) [5] and ANN-based approach proposed by them was able to achieve 99% accuracy. Beena et al. [6] proposed a CNN [7] based ASL recognition system. They used ASL dataset with 33,000 images for 24 alphabets and 0–9 numbers and accuracy of 94.6774% was achieved. Pigou et al. [8] were able to classify 20 Italian gestures using CNN with validation accuracy of 91.7%. They used Microsoft Kinect to capture gestures. Microsoft Kinect is able to capture depth feature. Depth feature aids significantly in image classification.

The contemporary research focused on the numbers, alphabets, limited words, and single-handed gestures. In contrast, this paper aims to help the hearing-impaired community by developing a simple computer vision-based system, which works on 50 ISL words including numbers and double handed gestures.

This paper is composed of six sections; Sect. 2 discusses the basic concepts used in the paper. Section 3 discusses the proposed system. Results of the experiment are discussed in Sect. 4. A complete conclusion is drawn in Sect. 5. Section 6 highlights the future aspects of the paper.

2 Basic Concepts

2.1 Adaptive Thresholding

Image binarization can be achieved with the help of adaptive thresholding. Image binarization is a method of separation of pixel intensity in two groups. Setting black as foreground and white as background or vice versa. Image binarization and thresholding is an effective way to separate an object from the background. In adaptive thresholding, a threshold value is set such that pixel intensity below that threshold will be treated as zero while pixel intensity greater than the threshold will be treated as one. Equation 1 shows the formula for adaptive thresholding.

$$b(x, y) = \begin{cases} 0, & I(x, y) \le T(x, y) \\ 1, & I(x, y) > T(x, y) \end{cases} \tag{1}$$

where $T(x, y)$ is the threshold, $b(x, y)$ is the binarized image, and $I(x, y)$ is the intensity of pixel at (x, y).

2.2 Image Smoothing Using Gaussian Blur

Image smoothing is a technique of removing noise from digital images. Smoothing can remove noise without losing important features from the image. Gaussian blur filter [9] uses a Gaussian function [10] to calculate transformation. Equation 2 represents Gaussian blur operator.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-\left(x^2+y^2\right)}{2\sigma^2}} \tag{2}$$

2.3 Key-Point Generation Using Scale Invariant Feature Transform

SIFT [11] is scale, rotation, viewpoint, illumination invariant algorithm. The key-point generation involves three steps. The first step is the generation of scale space. In the second step, Laplacian of Gaussians (LoG) [12] is generated while in the final step, key points are calculated.

Scale-Space Generation In the scale-space generation step, the original image is taken, and progressively blurred out images are generated. Then, the original image is resized to half and blurred out images are generated again. Images of the same size form an octave. The number of scales and octaves depend on the user. Blurring can be thought of as a convolution of the Gaussian operator and the image.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3}$$

where L is the obtained Blurred image, G is the Gaussian blur operator, I is the image, (x, y) are the coordinates in the image, σ is scale parameter, and * is convolution operator in (x, y). This operator applies Gaussian blur G onto the image I. Gaussian blur function is represented by Eq. 2. Let the amount of the blurring in one image be σ and then, the amount of blurring in the next image will be $k\sigma$, where k is a constant chosen by the user.

Laplacian of Gaussian (LoG) Approximation In the generation of LoG [12] step, an image is taken and blurred a little using Gaussian blur. Then, second-order derivatives are calculated on it (Laplacian). This process is computationally expensive. So to calculate LoG quickly, scale space obtained in the previous step is used. Difference between two consecutive scales is calculated. These differences of Gaussian images are approximately equal to LoG.

Finding Key Points In this step, each and every pixel is iterated, and all of its neighbors are checked. The check is done within the current image, one above and one below it. The pixel is marked as an approximate key point if it is greatest or least of all 26 neighbors. The minima or maxima lies somewhere between the pixels. So, subpixel value is calculated mathematically using Taylor expansion [13] of the image around the approximate key point.

2.4 CNN

CNN is a deep learning neural network. CNN is specialized for images, audios, videos, and speech processing. It is designed to learn features with the help of filters and hence requires very little data preprocessing and feature extraction. CNN is composed of one input layer, one or multiple convolutional layers, one or multiple max-pooling layers, a fully connected layer, and an output layer. Input layer accepts input which is passed to next layer. The convolutional layer is responsible to apply convolution operation to the input data. This layer works as eyes of CNN and looks for specific features useful for classification. The Filters are also known as the kernels. Max-pooling layer is useful for reducing parameters size and hence processing time. Fully connected layer acts as a classifier. Output layer gives an output vector consisting of probability for different classes. Each neuron uses activation function for mathematical processing of data. CNN has mechanism of dropout which is used to avoid overfitting.

2.5 Confusion Matrix

The confusion matrix is a matrix which is used to summarize the performance of the classifier. For a good classifier, it is a sparse matrix and can be represented in the form of a graph. Actual class is represented on X-axis while predicted class is represented on the Y-axis. Label to point (X, Y) represents a number of the example for which actual class is X and predicted is Y. When $X = Y$, then it is treated as accurate classification. Hence, (X, X) are treated as correctly classified examples. The confusion matrix is used to analyze the results of CNN with hybrid SIFT implementation, later in this research.

Precision Precision is a fraction of correctly identified examples to the number of examples for which that particular class is predicted as positive. Equation 4 specifies the formula for precision in the multiclass classifier.

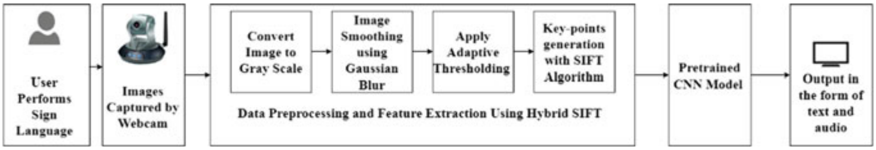

User Performs Sign Language
Images Captured by Webcam
Convert Image to Gray Scale
Image Smoothing using Gaussian Blur
Apply Adaptive Thresholding
Key-points generation with SIFT Algorithm
Data Preprocessing and Feature Extraction Using Hybrid SIFT
Pretrained CNN Model
Output in the form of text and audio

Fig. 1 Proposed system for ISL classification

$$\text{Precision}_i = \frac{\text{CF}_{ii}}{\sum_1^n \text{CF}_{ij}} \tag{4}$$

where CF_{ii} is (i, i)th entry in confusion matrix, CF_{ij} is (i, j)th entry in confusion matrix, and n is the total number of classes.

Recall A recall is a fraction of correctly identified examples to the number of examples available for that class. Equation 5 specifies the formula for recall in a multiclass classifier.

$$\text{Recall}_i = \frac{\text{CF}_{ii}}{\sum_1^n \text{CF}_{ji}} \tag{5}$$

where CF_{ii} is (i, i)th entry in confusion matrix, CF_{ji} is (j, i)th entry in confusion matrix, and n is total number of classes.

3 Proposed System

Figure 1 shows a flow diagram of a proposed system for ISL classification. Gestures performed by the user are captured by a webcam. The captured gesture is preprocessed and features are extracted using hybrid SIFT. Hybrid SIFT is discussed in Sect. 3.1. The preprocessed gesture is fed to a pretrained CNN model. The CNN model used for this research is explained in Sect. 3.2.

3.1 Data Preprocessing and Feature Extraction

Though CNN has the ability to work without any feature extraction and data preprocessing, data preprocessing is used for reducing computational power required and for better performance of the model. This paper presents a hybrid SIFT approach for data preprocessing and feature extraction.

Hybrid SIFT As discussed in Sect. 2.3, SIFT calculates key points. Data preprocessing before applying SIFT on the image can reduce noise and can help in better key-points generation. The key-point generation step of the SIFT is hybridized with

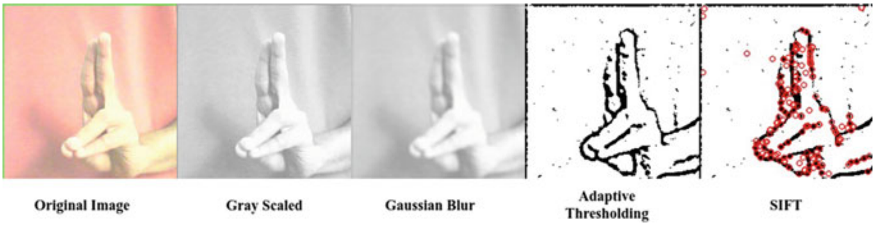


Fig. 2 Application of SIFT on preprocessed image (hybrid SIFT)

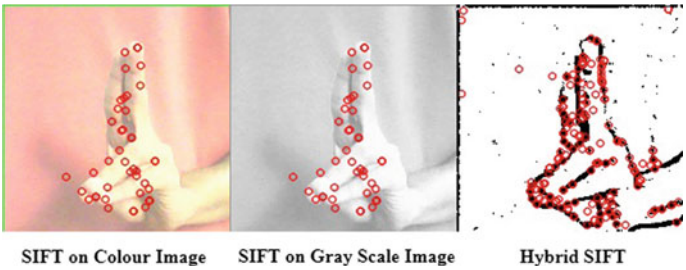
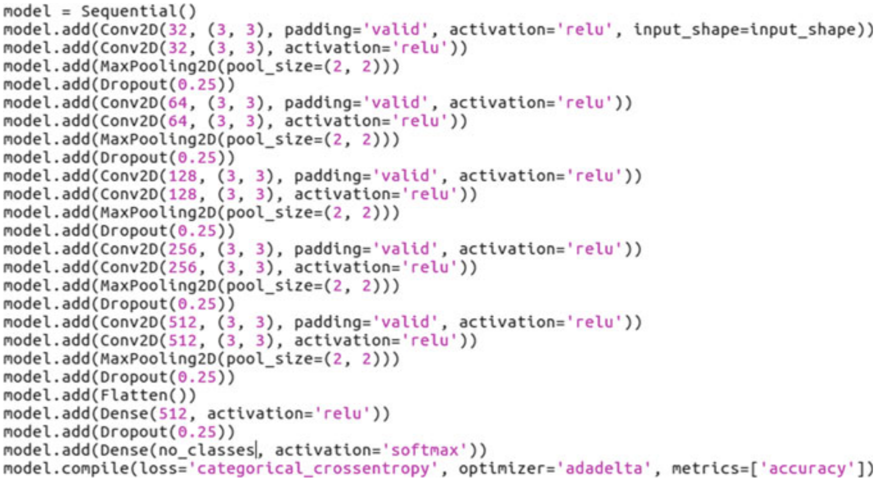


Fig. 3 Key-point calculation on original image with SIFT, grayscaled image with SIFT and hybrid SIFT

Gaussian blurring and adaptive thresholding. Steps involved in performing hybrid SIFT are shown in Fig. 2. In the first step, the image is captured and resized to 200 × 200 pixels. The second step involves converting the resized original image to grayscaled image. In the third step as discussed in Sect. 2.2, the grayscaled image is smoothened using a Gaussian blur filter. In next step, the smoothened grayscaled image is binarized using adaptive thresholding. Finally, key-point generation step of SIFT algorithm is applied to the binarized image. Figure 3 shows a comparison of key-point calculation using SIFT algorithm on a simple grayscaled image and hybrid SIFT algorithm. Figure 3 suggests that SIFT applied on the adaptive thresholded image gives better key points than when applied directly to the original image.

3.2 Architecture of CNN Model

This paper presents a self-designed CNN model for gesture recognition. Figure 4 shows the code snippet of the proposed CNN model. Proposed CNN model has 10 convolutional layers. Convolutional layers are divided into five groups. Each group contains two convolutional layers. For the first group, the number of filters is kept 32, for second group 64, for third group 128, for fourth group 256, and for the final group, it is kept 512. Each convolutional layer has a filter size of 3 × 3. Max-pooling



```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='valid', activation='relu', input_shape=input_shape))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), padding='valid', activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3, 3), padding='valid', activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(256, (3, 3), padding='valid', activation='relu'))
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(512, (3, 3), padding='valid', activation='relu'))
model.add(Conv2D(512, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(no_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adadelta', metrics=['accuracy'])
```

Fig. 4 Code snippet for CNN model

layer with pooling window of 2 × 2 is applied after each group of the convolutional layer. After each max-pooling layer, to avoid overfitting dropout ratio of 0.25 is used. Each convolution layer uses rectified linear unit (RELU) as the activation function [14]. Optimizer for training was kept fixed as Adadelta [15]. The model uses Softmax [16] as a classifier. To calculate the performance of model, the loss function is used as cross-entropy [17]. The CNN model is coded by using python-based library Keras [18].

4 Results

The proposed CNN model is applied to self-created dataset of ISL with 50 signs and 100 images per sign. CNN model is trained with an incremental increase of epochs approach. Batch size for the model training is kept fixed at 64. Starting from 10 epochs, epochs were increased by 5 till model converged with constant validation accuracy. With increments in the number of epoch approach, proposed model converged with constant validation accuracy at 25 epochs. The model was trained till 50 epochs. Section 4.1 describes the system requirement for the proposed system, Sect. 4.2 elaborates data acquisition process. The accuracy of the proposed CNN model is discussed in Sect. 4.3 which shows a confusion matrix generated for the proposed model and dataset. Finally, we discuss the comparison of the proposed system with related research.

4.1 System Requirements

The proposed system tried to keep user interaction with the system using desktop application. System requirements are as follows:

- 4 GB Ram
- 1 GB Free Space
- Web Cam.

4.2 Data Acquisition

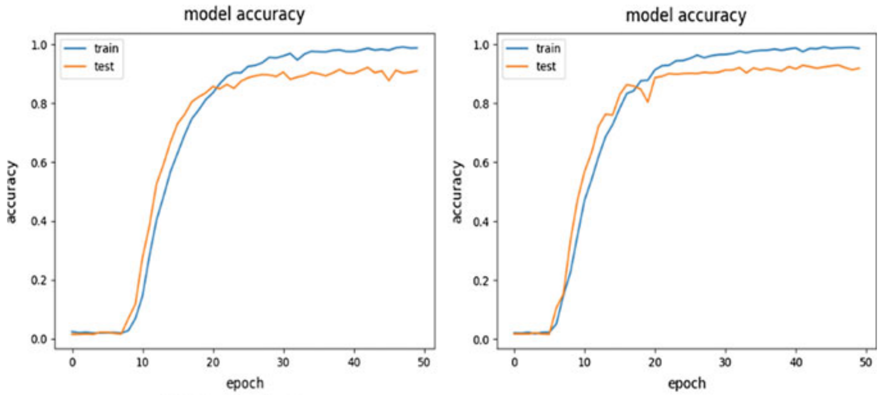
The standard dataset for Indian sign language is not available. Two sets of datasets each of 5000 images were created. One was used for training and validation purpose while other for the testing purpose. Dataset was created with the help of 5 MP webcam attached to a laptop. Dictionary of 50 most used signs is created by taking help of Deaf and Dumb School, Mumbai. Each dataset contains total 5000 images, where for each sign 100 images of 200 × 200 pixel are captured. Dataset is created with the help of 20 people. The age group varies from 16 to 50. Around 70% of the people were in the age group of 20–25. Both males and females have participated in dataset creation.

4.3 Accuracy of Proposed CNN Model

Table 1 summarizes accuracy matrix of the proposed model. For CNN with adaptive thresholding, training accuracy (TA) achieved is 97.58% and validation accuracy (VA) is 91.84%. For CNN with hybrid SIFT, training accuracy increases to 98.83% and validation accuracy increases to 92.72%. It shows that CNN with hybrid SIFT performs better than CNN with only adaptive thresholding. Figure 5 shows the accuracy graph of the proposed model. In the accuracy graph, accuracy is represented as dependent variable on Y-axis and number of epochs as an independent variable on X-axis. Accuracy graph lists accuracy for training dataset by blue line and accuracy on validation dataset by the orange line. Accuracy graph shown in Fig. 5 suggests that early stopping of the model at around 25 epoch can avoid overfitting issue.

Table 1 Accuracy matrix for proposed CNN model

CNN with adaptive thresholding		CNN with hybrid SIFT	
TA	VA	TA	VA
97.58	91.84	98.83	92.72



CNN with Adaptive Thresholding

Hybrid SIFT

Fig. 5 Model accuracy

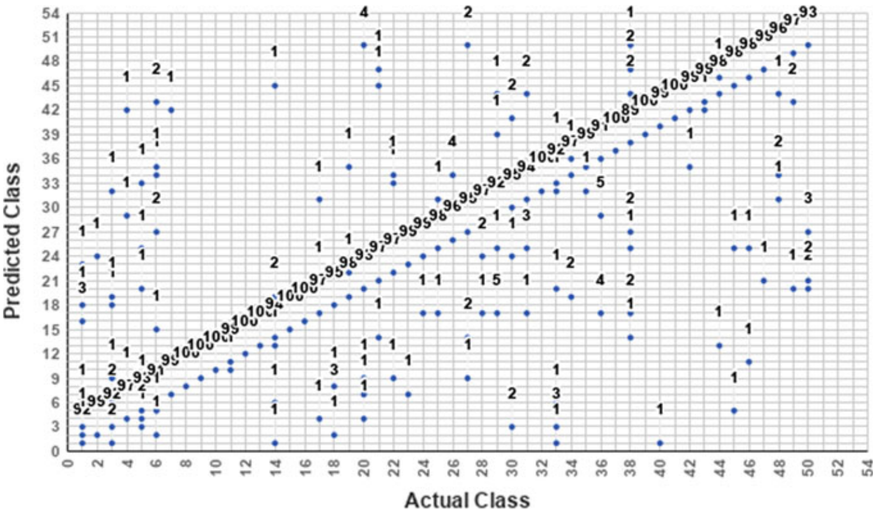


Fig. 6 Confusion matrix obtained by running trained CNN model on testing dataset

4.4 Confusion Matrix of Proposed System

The hybrid SIFT algorithm is first applied on the dataset and then CNN model is trained by using the same dataset. As discussed in Sect. 2.5, the confusion matrix is created by running trained CNN model on the testing dataset. Figure 6 shows the confusion matrix in a graphical format. The *Y*-axis of the graph represents predicted class while *X*-axis represents the actual class. Confusion matrix obtained for the proposed model is of sparse nature.

Table 2 Precision and recall obtained by running trained CNN model on testing dataset

Sign	Precision	Recall	Sign	Precision	Recall	Sign	Precision	Recall
House	94.8454	92	High	97.9381	95	Rose	97.0588	99
A board	97.0588	99	How many	95.1456	98	See	98.913	91
All Gone	92.9293	92	I	94.898	93	Seven	99.0099	100
Baby	96.0396	97	Man	96.0396	97	Short	100	89
Beside	97.8947	93	Marry	98.9796	97	Six	99.0099	100
Book	91.9192	91	Meat	99	99	Superior	100	99
Bowl	97.0588	99	Medal	96.1165	99	Ten	98.0392	100
Bridge	98.0392	100	Mid Day	92.4528	98	Thick	97.0588	99
Camp	96.1538	100	Middle	100	96	Thin	96.1165	99
Cartridge	99.0099	100	Money	93.1373	95	Three	94.2308	98
Eight	99	99	Moon	100	97	Tobacco	98	98
Five	100	100	Mother	93.8775	92	Two	98.9899	98
Fond	98.0392	100	Nine	100	95	Up	97.0588	99
Four	95.9184	94	One	96.9072	94	Watch	100	96
Friend	99.0099	100	Opposite	97.0874	100	Write	100	97
Glove	97.0874	100	Prisoner	97.8723	92	You	93	93
Hang	86.6071	97	Ring	92.381	97			

Precision and recall were calculated for multiclass using formula shown in Sect. 2.5. Table 2 summarizes precision and recall in percent for proposed model and 50 signs available in the dataset. Highest precision obtained is 100% while least precision is 86.6071%. Highest recall obtained is 100% while least recall is 89%.

5 Conclusion

The sensor incorporated glove-based systems which are most common in sign language recognition are usually costly and difficult to use; in contrast, image classification-based system proposed in this paper is much cheaper and easier to use. CNN is an important and efficient algorithm for image classification. This paper proposed a system for Indian sign language recognition using classification by CNN and feature extraction by hybrid SIFT. CNN is robust and stable such that it requires very little image preprocessing. But image processing using hybrid SIFT improves the performance of CNN classifier. The model proposed in this paper has achieved validation accuracy of 92.78% for CNN with hybrid SIFT approach while 91.84% accuracy was achieved for CNN with adaptive thresholding approach. The system proposed in this paper can work just with a laptop and web camera and hence can be used with mobility by the hearing-impaired community. The system proposed in this paper can also be used for learning Indian sign language.

6 Future Work

Dataset used in this paper contains 50 Indian signs. Further work can be done to increase the number of signs as well as images per sign. This paper considers only static Indian signs. In the future, CNN can also be implemented for motion-based Indian signs. The proposed system can be extended to work with handheld mobile devices by optimizing memory and power requirement.

Acknowledgements We would like to thank Principal of Deaf and Dumb School, Mumbai for her help in understanding ISL gestures. We would like to thank teachers and students of Deaf and Dumb School for their help in the creation of ISL dataset.

References

1. Heera SY et al (2017) Talking hands—an Indian sign language to speech translating gloves. In: 2017 International conference on innovative mechanisms for industry applications (ICIMIA)
2. Ekbote J et al (2017) Indian sign language recognition using ANN and SVM classifiers. In: 2017 International conference on innovations in information, embedded and communication systems (ICIIECS)
3. Duch W (2005) Artificial neural network biological inspirations. In: ICANN 2005: 15th international conference, Warsaw, Poland, September 11–15, 2005, proceedings, Pt. 1. Springer, Heidelberg
4. Qi X et al (2017) Data classification with support vector machine and generalized support vector machine
5. Vo T et al (2015) Tensor decomposition and application in image classification with histogram of oriented gradients. Neurocomputing 165:38–45
6. Beena MV, Agnisarman Namboodiri MN (2017) Automatic sign language finger spelling using convolution neural network: analysis. Int J Pure Appl Math 117(20)
7. Aghdam HH, Heravi EJ (2017) Convolutional neural networks. In: Guide to convolutional neural networks, pp 85–130
8. Pigou L et al (2015) Sign language recognition using convolutional neural networks. Computer vision—ECCV 2014 workshops lecture notes in computer science, pp 572–578
9. Gaussian Smoothing (2008) Wolfram Demonstrations Project
10. Gaussian function. Springer Reference
11. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60(2):91–110
12. Laplacian of Gaussian Filtering (2008) Wolfram Demonstrations Project
13. Berz M (1997) From Taylor series to Taylor models
14. Ramachandran P et al (2017) Searching for activation functions. In: ICLR 2018 conference

15. Zeiler MD (2012) ADADELTA: an adaptive learning rate method
16. Pellegrini T (2015) Comparing SVM, Softmax, and shallow neural networks for eating condition classification. Interspeech 2015
17. Dahal P, Classification and loss evaluation—Softmax and cross entropy loss. https://deepnotes.io/softmax-crossentropy
18. Keras: The Python Deep Learning library. https://keras.io/