Sādhanā (2020) 45:34
https://doi.org/10.1007/s12046-019-1250-6

A depth-based Indian Sign Language recognition using Microsoft Kinect

T RAGHUVEERA*, R DEEPTHI, R MANGALASHRI and R AKSHAYA

Department of Computer Science and Engineering, College of Engineering Guindy Campus, Anna University, Chennai 600025, India
e-mail: raghuveera@annauniv.edu; deepthiraghu96@gmail.com; rmangalashri@gmail.com; achuraju28@gmail.com

MS received 5 July 2017; revised 3 July 2019; accepted 27 September 2019; published online 30 January 2020

Abstract. Recognition of sign language by a system has become important to bridge the communication gap between the abled and the Hearing and Speech Impaired people. This paper introduces an efficient algorithm for translating the input hand gesture in Indian Sign Language (ISL) into meaningful English text and speech. The system captures hand gestures through Microsoft Kinect (preferred as the system performance is unaffected by the surrounding light conditions and object colour). The dataset used consists of depth and RGB images (taken using Kinect Xbox 360) with 140 unique gestures of the ISL taken from 21 subjects, which includes single-handed signs, double-handed signs and fingerspelling (signs for alphabets and numbers), totaling to 4600 images. To recognize the hand posture, the hand region is accurately segmented and hand features are extracted using Speeded Up Robust Features, Histogram of Oriented Gradients and Local Binary Patterns. The system ensembles the three feature classifiers trained using Support Vector Machine to improve the average recognition accuracy up to 71.85%. The system then translates the sequence of hand gestures recognized into the best approximate meaningful English sentences. We achieved 100% accuracy for the signs representing 9, A, F, G, H, N and P.

Keywords. ISL gesture recognition; multi-class SVM; SURF; HOG; LBP; depth based; gesture translation.

1. Introduction

Sign language is a system of communication using visual gestures and signs, as used by the Hearing and Speech Impaired (HSI) people. For effective communication to happen between the abled and the HSI people, knowledge and familiarity of sign language is essential on either side. This is a rarity as most of the abled are not familiar with sign language, thus leading to isolation of HSI community from the mainstream society and denial of equal opportunities, despite their potential and abilities. The goals are to develop assistive technology to enable and empower the HSI community for effective communication. Given a hand gesture/sign in Indian Sign Language (ISL), the proposed system would translate it into text (meaningful sentence(s)) and speech.

Hand gestures are an important mode for communicating with the HSI people. Various methods are being used to track hand gestures, one of which is Microsoft Kinect [1]. The stream of input data to the Kinect will be the live action of human's gestures. Microsoft Kinect captures images through different streams such as RGB, depth and skeleton.

A wide variety of sign languages exist across the world, because of differences in regional, cultural and spoken native languages. Some of them are [2] ISL (Indian Sign Language), ASL (American Sign Language), BSL (British Sign Language) and so on. The official ISL dictionary [3] is constantly getting updated with newer vocabulary and as on date it has 6000 words, which fall under various categories. ISL is much more complex compared with ASL because

- It consists of a combination of single- and double-hand gestures.
- Double-hand gestures involve overlapping of hands.
- Differences in hand locations with respect to body imply a different sign.

The evidence of less research work in the area of ISL recognition can be attributed to delayed standardization [4]. Previous research in ISL recognition involved usage of RGB and skeletal data to track hand movements [5, 6]. However, RGB-based recognition has been applied only to the set of numbers. Also, it involves usage of wearables such as gloves/HMDs, which makes the process

*For correspondence

cumbersome and is prone to inaccuracies due to varying lighting and colour conditions. Depth stream deals with images in terms of matrices of depth values between the user and the sensor. This reduces the need for maintaining the ideal physical environment while capturing images. In order to recognize the signs, there is a need for a learning methodology such as KNN, Artificial Neural Networks, HMM, etc. [6]. From the literature review, it was found that Support Vector Machine (SVM) provides high accuracy recognition rates [7]. However, SVM has not been incorporated into ISL recognition so far. Thus, a system that can use SVM classifier for ISL recognition is proposed.

The following are the key contributions of this work:

- Devising a procedure to improve accuracy of ISL gesture recognition (not just limited to numbers alone)
- Experimenting with a wider and more standardized ISL dataset including numbers, alphabets and words; achieving better recognition accuracy, including 100% accuracy for the signs representing 9, A, F, G, H, N and P.
- Our method has been successful in recognition of complex signs unique to ISL that include overlapping signs and double-hand signs.

The following are the noticeable characteristics of this work:

- Applying SVM classifier for recognition of ISL.
- Using feature ensembling for more accurate label prediction.
- Translating a sequence of hand gestures into the best approximate meaningful English sentence.

2. Literature review

Dardas and Georganas [8] proposed a system that includes detecting and tracking bare hand in cluttered background using skin detection and hand posture contour comparison algorithm after face subtraction, recognizing hand gestures via bag-of-features and multi-class SVM and building a grammar that generates gesture commands to control an application. It was able to achieve satisfactory real-time performance regardless of the frame resolution size as well as high classification accuracy of 96.23% under variable scale, orientation and illumination conditions, and cluttered background. However, this system was limited to a certain handful number of video game gestures.

Yang et al [9] proposed a gesture recognition system using depth information provided by Kinect, and implemented in a media player application. It was able to recognize eight gestures to control the media player, with a maximum confusion rate of 8.0. This system demonstrated the applicability of using Kinect for gesture recognition in a contact-less user interface.

Padmavathi et al [10] presented an ISL recognition technique using neural networks. The features of the English characters have been obtained for only single-hand alphabets and were classified using neural networks. Due to segmentation problem, the features of letters C and U were mixed up. The effect of illumination also added up to the improper segmentation.

Madhuri et al [11] proposed an ISL recognition system that used KNN and ANN classifiers. The result of these experiments is achieving up to 97.10% accuracy. The system can be useful for static ISL numeral signs only. The ISL recognizer system cannot be considered as a complete system, as for complete recognition of sign language, we have to include ISL alphabets, words and sentences. *Kim et al* [12] proposed a Korean sign language recognition system using SVM and a depth camera. As a result of the test, although the proposed system was slightly affected by angles, its recognition rates were found to be excellent.

Vijay et al [6] analysed sign language recognition methods and effectively compared them. After the survey on the approaches used in various vocabulary-based sign language recognition systems, it was found that most of the times, a combination of different methods and algorithms has to be used to achieve a moderate to acceptable rate of recognition.

Viswanathan and Idicula [4] screened gesture recognition methods for their accuracy in handling gestures in the context of complex ISL. The overall analysis of selected review clearly indicated the advancement of sign language recognition research globally and on an Indian context. Apart from a few promising works, most of the research works use static gestures for validation.

Nagashree et al [7] designed a system using Canny's edge detector for edge detection, histogram of gradients for feature extraction and the SVM classifier, which is widely used for classification and regression testing. Their system was able to identify a selected set of 20 hand gestures.

Wang and Yang [13] presented a multi-class hand posture recognition system based exclusively on computer vision. The developed system uses an ensemble of real-time deformable detectors to handle the detection of multiple classes of hand postures. This method worked only for recognizing alphabets.

Ansari and Harit [14] implemented and tested a functional unobtrusive ISL recognition system real-world data using Microsoft Kinect. They achieved above 90% recognition rates for 13 signs and 100% recognition for three signs with overall 16 distinct alphabets. However, their system tackled only static signs within a limited vocabulary.

Mitra and Acharya [5] provide an excellent review of the work on gesture recognition, including facial gestures. The detection involves primarily three steps – pre-processing, feature extraction and classification.

Tiwari et al [15] presented a sign language recognition technique using a Kinect depth camera and neural network.

Based on a particular contour position of the hand as a signal on which Discrete Cosine Transform (DCT) is applied, the first 200 DCT coefficients of the signal are fed to the neural network for training and classification and finally the network classifies and recognizes the sign. This system works only for numbers 0–9.

Wang et al [16] proposed a novel human detection approach capable of handling partial occlusion using two kinds of detectors – global detector for whole scanning windows and part detectors for local regions. It combined the trilinear interpolated Histogram of Oriented Gradients (HOG) with Local Binary Patterns (LBP) in the framework of integral image. It has been shown that the HOG–LBP feature outperforms other state-of-the-art detectors on the INRIA dataset. However, their detector cannot handle the articulated deformation of people.

Halim and Abbas [17] developed a system for detecting and understanding Pakistani sign language based on Dynamic Time Warping (DTW) algorithm. The proposed method was capable of successfully detecting gestures stored in the dictionary with an accuracy of 91%. The primary disadvantage of using DTW is the heavy computational burden required to find the optimal solution.

Nagarajan and Subashini [18] developed a static hand gesture recognition system for ASL using Edge-Oriented Histogram (EOH) features and multi-class SVM. The edge histogram count of input sign language alphabets is extracted as the feature and applied to a multi-class SVM for classification. Yet, the system is not designed to work in a background-independent environment.

Ghotkar and Kharate [19] designed and developed a new algorithm for ISL sentence creation considering limitation of continuous sign language recognition. Rule-based and DTW-based methods for ISL word recognition are developed. The DTW-based method gave better accuracy for continuous word recognition than the rule-based method. Sometimes exact sentence might not have been interpreted, but thoughts having same meaning were conveyed.

Van den Bergh and Van Gool [20] proposed a hand gesture interaction system based on an RGB camera and a ToF camera. An improved hand detection algorithm is introduced based on adaptive skin colour detection and depth. However, the method also shows the possibility to classify based only on depth information, which means systems with only a ToF camera are feasible.

Mohandes et al [21] have presented a critical review of the existing systems and methods for automatic recognition of Arabic sign language. They have classified recognition methods and systems into image-based and sensor-based.

Agarwal and Thakur [22] proposed a sign language recognition system, which makes use of depth images to recognize digits in Chinese Sign Language. However, this system uses Gaussian blur filter for noise filtering. This method does not retain edges, which is essential for accurate interpretation of ISL. Our proposed work is different from the existing works in the following ways:

- The system extracts features from three different feature classes – HOG, Speeded Up Robust Features (SURF) and LBP – and ensembles the output of these three to produce a more accurate recognition.
- Recognition of complex signs unique to ISL, involving two hands and overlapping hands, has been successful.
- This work primarily focuses on producing the best approximate interpretation of a sequence of ISL gestures in the form of meaningful English language sentences in real time.
- Applying SVM classifier for recognition of ISL.

3. Proposed work

Our ISL recognition system (figure 1) consists of five stages: pre-processing, hand(s) segmentation, extraction of three (SURF, HOG and LBP) features and sign recognition using an SVM, prediction of output sign using ensemble technique and sentence interpretation.

3.1 *Pre-processing*

This stage involves image acquisition from the depth stream of Kinect sensor in real time, followed by de-noising the depth image.

3.1.1 *Image acquisition* There have been no refined image datasets for ISL as it consists of thousands of documented and undocumented signs. Therefore, a carefully selected subset of gestures from the ISL dictionary is used for the task (figure 2) [14]. The chosen subset of ISL dataset consists of 140 static signs handpicked from ISL, performed by 21 volunteers with different possible variations, thereby accounting to up to 40 images under each sign (table 1). Apart from complete words, the dataset also has signs for alphabets and numbers (table 2). Both single-handed and double-handed variants are included. Totally, there are 4600 images in the dataset. The dataset is divided into training and validation sets with an 80:20 ratio. The ideal distance between the subject and the device must range from 1 to 3 m in order to capture the image. As we are dealing with depth image, there is no need to maintain specific lighting conditions and colour codes and wearables such as gloves. It is important that the user must pose by leaving ample distance between his hands and the rest of his body in order to achieve proper segmentation.

3.1.2 *Image denoising using median filtering* Denoising involves removing salt and pepper noise from the captured depth image with a resolution of

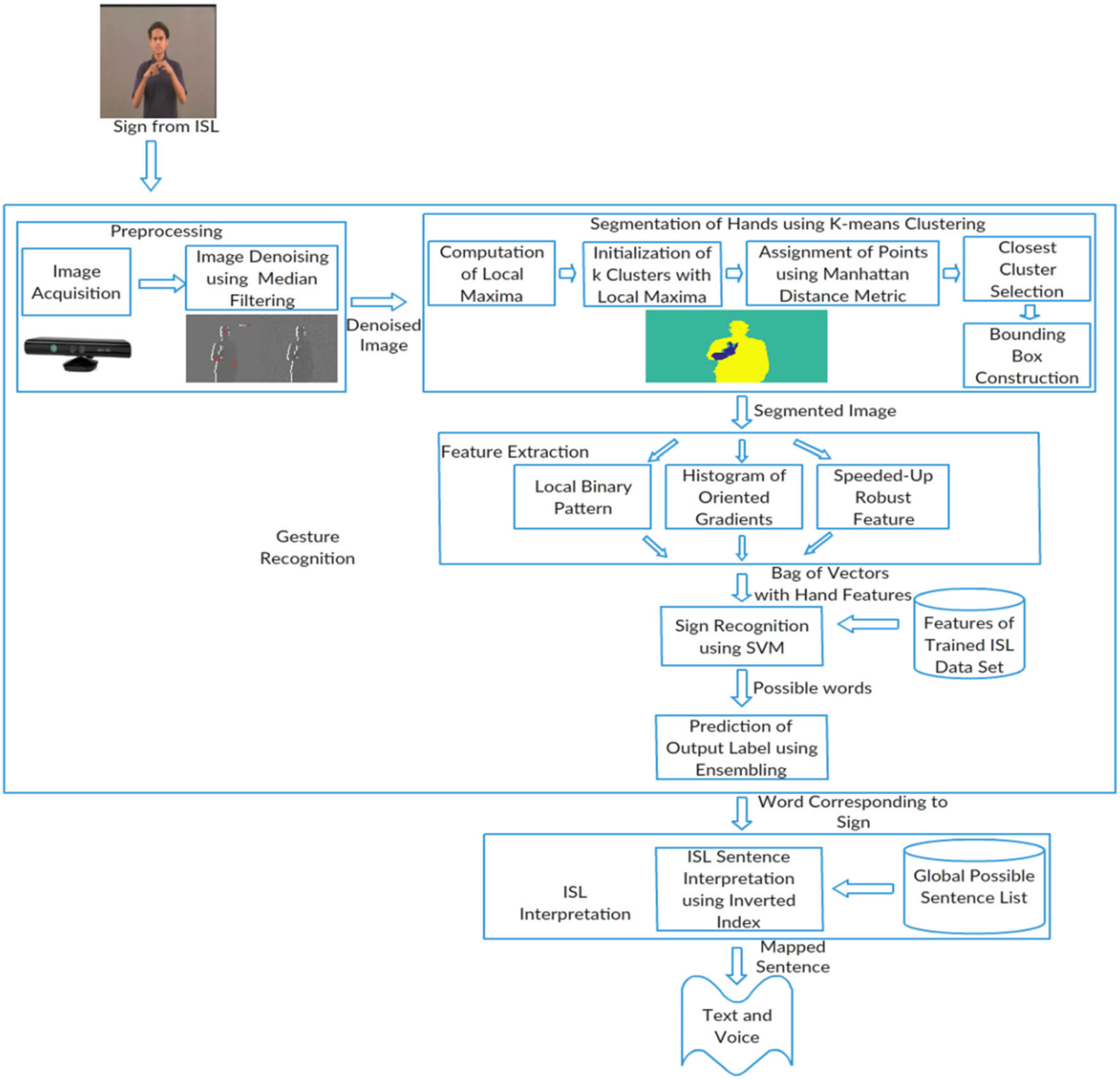

Sign from ISL
Preprocessing
Image Acquisition
Image Denoising using Median Filtering
Denoised Image
Segmentation of Hands using K-means Clustering
Computation of Local Maxima
Initialization of k Clusters with Local Maxima
Assignment of Points using Manhattan Distance Metric
Closest Cluster Selection
Bounding Box Construction
Segmented Image
Feature Extraction
Local Binary Pattern
Histogram of Oriented Gradients
Speeded-Up Robust Feature
Gesture Recognition
Bag of Vectors with Hand Features
Features of Trained ISL Data Set
Sign Recognition using SVM
Possible words
Prediction of Output Label using Ensembling
Word Corresponding to Sign
ISL Interpretation
ISL Sentence Interpretation using Inverted Index
Global Possible Sentence List
Mapped Sentence
Text and Voice

Figure 1. Proposed system.

640 × 480. This depth image consists of pixels with values in the range 0–2047 that is proportional to the distance between the user and the depth sensor. The noise is represented in the form of 2047 pixel values. This is because they are the points that are closer than 0.8 m and farther than 3.5 m (figure 3a). For denoising we need a fast, efficient filter that can remove most of the 2047 values without affecting edge information, as this might affect feature extraction. Median filtering is used to get rid of the spikes and to preserve the edge information (figure 3b).

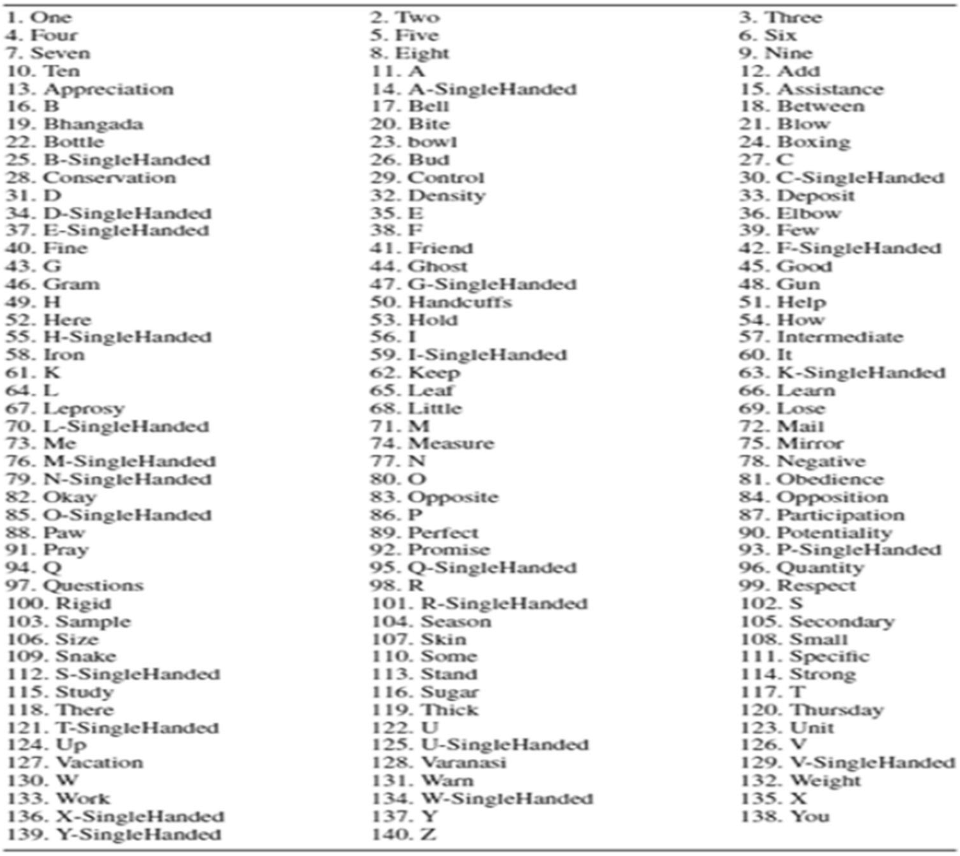
3.2 *Segmentation of hands using K-mean clustering*

Segmentation involves separating the useful regions of the input image, hands up to the elbows, in this case [23]. The hands lie closer to the camera than the rest of the body.

Once the depth image has been segmented, we can choose the segment that is closest to the camera to identify the hands [23]. In order to work efficiently in real time, segmentation must be fast. Thus, we develop a process to speed up *K*-mean clustering to segment the image into closely connected clusters.

3.2.1 *Computation of local maxima and initialization of K clusters with local maxima*

There are a number of local maxima in the histogram of depth values (figure 4). These local maxima show the approximate depths of major objects in the space in front of the Kinect. In our case, there are three major regions at three different depths that are to be segmented – the hands, the body and the background. The first three high-frequency and distinct local maxima from the histogram of depth values account for the seed values. This is obtained by setting thresholds:



1. One	2. Two	3. Three
4. Four	5. Five	6. Six
7. Seven	8. Eight	9. Nine
10. Ten	11. A	12. Add
13. Appreciation	14. A-SingleHanded	15. Assistance
16. B	17. Bell	18. Between
19. Bhangada	20. Bite	21. Blow
22. Bottle	23. bowl	24. Boxing
25. B-SingleHanded	26. Bud	27. C
28. Conservation	29. Control	30. C-SingleHanded
31. D	32. Density	33. Deposit
34. D-SingleHanded	35. E	36. Elbow
37. E-SingleHanded	38. F	39. Few
40. Fine	41. Friend	42. F-SingleHanded
43. G	44. Ghost	45. Good
46. Gram	47. G-SingleHanded	48. Gun
49. H	50. Handcuffs	51. Help
52. Here	53. Hold	54. How
55. H-SingleHanded	56. I	57. Intermediate
58. Iron	59. I-SingleHanded	60. It
61. K	62. Keep	63. K-SingleHanded
64. L	65. Leaf	66. Learn
67. Leprosy	68. Little	69. Lose
70. L-SingleHanded	71. M	72. Mail
73. Me	74. Measure	75. Mirror
76. M-SingleHanded	77. N	78. Negative
79. N-SingleHanded	80. O	81. Obedience
82. Okay	83. Opposite	84. Opposition
85. O-SingleHanded	86. P	87. Participation
88. Paw	89. Perfect	90. Potentiality
91. Pray	92. Promise	93. P-SingleHanded
94. Q	95. Q-SingleHanded	96. Quantity
97. Questions	98. R	99. Respect
100. Rigid	101. R-SingleHanded	102. S
103. Sample	104. Season	105. Secondary
106. Size	107. Skin	108. Small
109. Snake	110. Some	111. Specific
112. S-SingleHanded	113. Stand	114. Strong
115. Study	116. Sugar	117. T
118. There	119. Thick	120. Thursday
121. T-SingleHanded	122. U	123. Unit
124. Up	125. U-SingleHanded	126. V
127. Vacation	128. Varanasi	129. V-SingleHanded
130. W	131. Warn	132. Weight
133. Work	134. W-SingleHanded	135. X
136. X-SingleHanded	137. Y	138. You
139. Y-SingleHanded	140. Z	

Figure 2. Dataset words.

MinPeakHeight as 70 and MinPeakDist as 70 initially. If the number of peaks obtained is less than 3, MinPeakDist is decremented by 5 until it finds 3 peaks corresponding to hands, torso and background.

3.2.2 *Assignment of points using Manhattan distance metric and closest cluster selection* The peaks obtained are initialized as seed values for the *K*-mean algorithm in order to speed up its performance instead of using random seeds by default. We used Manhattan distance as the distance metric. We then select the closest cluster on the basis of the clusters' mean depth.

3.2.3 *Bounding box construction* In real-world situations, it is impossible for all human beings to stand perfectly straight and display their hands separately from the rest of their body (figure 5). In several cases, the closest depth region consists of not only hands but also some portions, like the belly, of the user's torso. In order to separate the hands from the rest of the nearest depth cluster (figure 6), we construct a bounding box (figure 7) as per the algorithm presented in figure 8.

3.3 *Feature extraction and recognition*

Feature extraction involves extracting the distinct features from the segmented image in order to generate feature vectors that uniquely identify a given sign. The feature vectors are then clustered using *K*-mean algorithm and categorized into a bag of vectors Later, feature matching is done between the training set images and the test set image in real time. There are several feature extraction techniques available. After careful considerations based on our application, we use three different feature extraction methods in order to account for shape detection, rotation invariance and scale invariance.

3.3.1 *Feature extraction*

- LBP
  The LBP feature vector, in its simplest form, is created by dividing the examined window into cells and for each pixel in a cell, compares the pixel to each of its 8 neighbours. Where the value of the centre pixel is greater than the neighbour's value, write "0"; otherwise, write "1". This gives an 8-digit binary number. Compute the histogram, over the cell, of the frequency

Table 1. Dataset statistics.

	Number			
Type	One	Two	No. of sign variants	No. of sign performers
Words	27	54	40	21
Alphabets	27	22		
Numerals	9	1		

Table 2. Digits and alphabets.

Sign number in dataset	Alphabet
1–10	Digits
11,14	A
16,25	B
27,30	C
31,34	D
35,37	E
38,42	F
43,47	G
49,55	H
56,59	I
61,63	K
64,70	L
71,76	M
77,79	N
80,85	O
86,93	P
94,95	Q
98,101	R
102,112	S
117,121	T
122,125	U
126,129	V
130,134	W
135,136	X
137,139	Y
140	Z

of each “number”. This histogram can be seen as a 256-dimensional feature vector. Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window. It has been determined that when LBP is combined with the HOG descriptor, it improves the detection performance considerably on some datasets [16]. LBP is rotation invariant.

- HOG
  The HOG features are used for object shape detection. HOG decomposes an image into small squared cells, computes a HOG in each cell, normalizes the result using a block-wise pattern and returns a descriptor for each cell [16]. The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation.

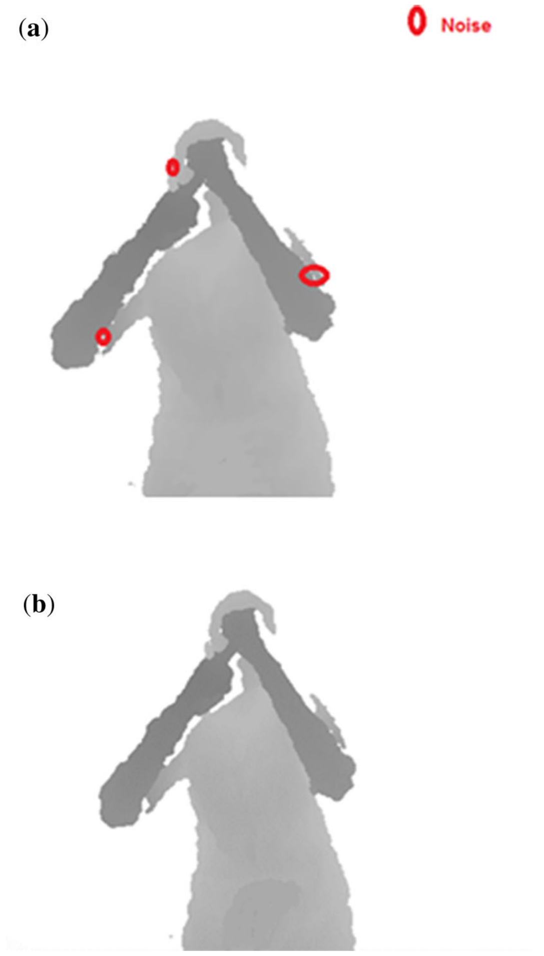


Figure 3. **a** Image with noise, **b** image without noise.

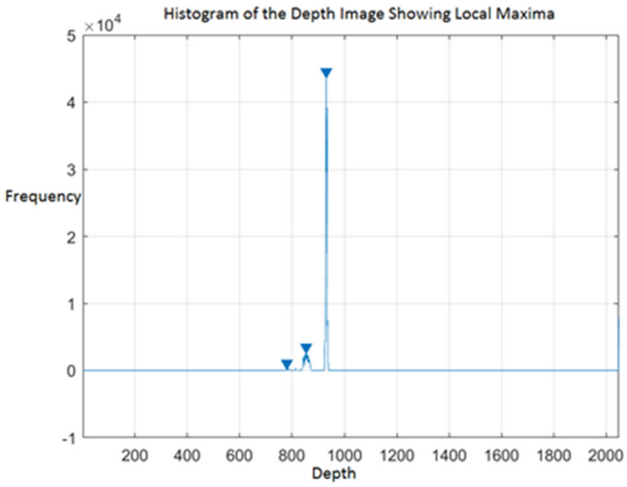


Figure 4. Local maxima.

- SURF To detect points of interest, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with three

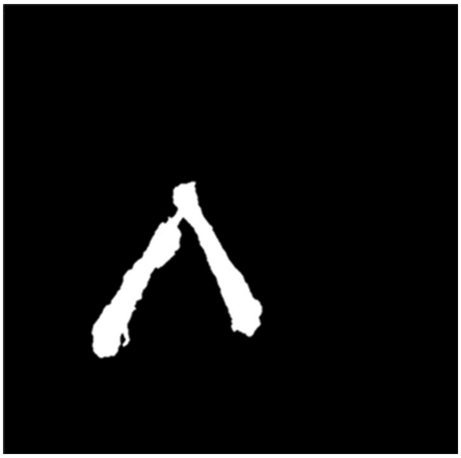


Figure 5. Segmented hands.

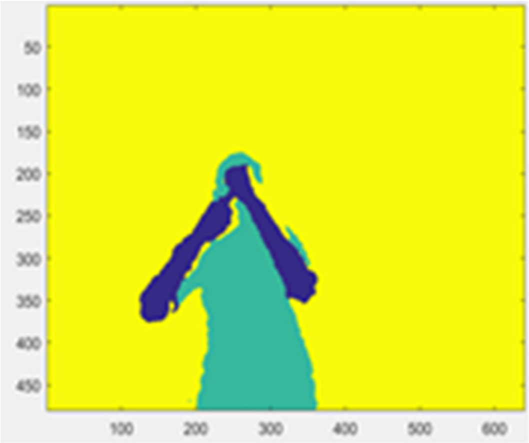


Figure 6. Three clusters.

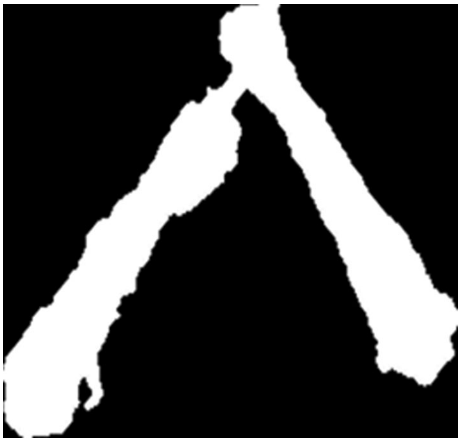


Figure 7. Bounding box.

integer operations using a pre-computed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. SURF is a fast algorithm for extracting features, and is invariant to scale orientation [8].

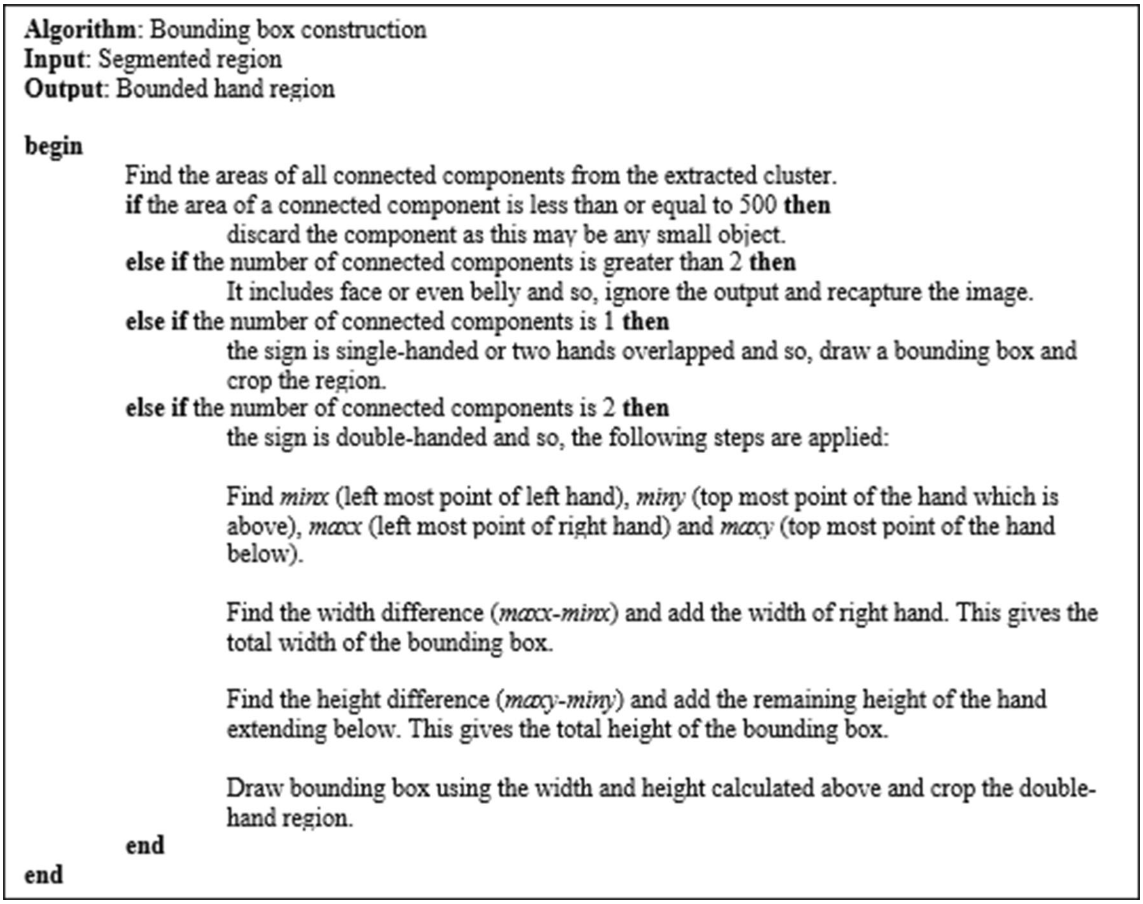
3.3.2 *Sign recognition using SVM* After mapping all the key points that represent every training image with its generated bag-of-words vector using *K*-mean clustering, we feed every bag-of-words vector with its related class or label number into a multi-class SVM classifier to build the multi-class SVM training classifier model. The SVM classifier is a multi-class classifier, which looks for an optimal hyperplane as a decision function. Once trained on images containing some particular gestures, the SVM classifier can make decisions regarding the signs. It carries out classification by creating an *N*-dimensional hyperplane that optimally divides the data into two groups. In the SVM literature, a predictor variable is known as an attribute, and a transformed attribute that is used to define the hyperplane is known as a feature. The operation of selecting the most appropriate representation is called as feature selection. A group of features that describes one case is known as a vector. Therefore, the objective of SVM modelling is to define the optimal hyperplane that divides clusters of vectors in such a way that cases with one class of the target variable are on one side of the plane and cases with the other classes are on the other side of the plane. The vectors near the hyperplane are the support vectors.

3.4 *Prediction of output label using ensembling*

Ensemble is the process of combining diverse set of features (individual models) together to improvise on the stability and predictive power of the system. We combine all the predictions from the three features together in order to achieve the most accurate results as possible. The three most popular methods for combining the predictions from different models are bagging (building multiple models (typically of the same type) from different subsamples of the training dataset), boosting (building multiple models (typically of the same type) each of which learns to fix the prediction errors of a prior model in the chain) and stacking (building multiple models (typically of differing types) and supervisor model that learns how to best combine the predictions of the primary models).

For our application, we have used boosting method. Boosting is a two-step approach, where one first uses subsets of the original data to produce a series of averagely performing models and then “boosts” their performance by combining them together using a particular cost function (Majority Vote) as described here:

- If the labels predicted by the three models are identical, then predict that label as the original output label.
- If the labels predicted by any two models are identical, then predict that label as the original output label.



```
Algorithm: Bounding box construction
Input: Segmented region
Output: Bounded hand region

begin
    Find the areas of all connected components from the extracted cluster.
    if the area of a connected component is less than or equal to 500 then
        discard the component as this may be any small object.
    else if the number of connected components is greater than 2 then
        It includes face or even belly and so, ignore the output and recapture the image.
    else if the number of connected components is 1 then
        the sign is single-handed or two hands overlapped and so, draw a bounding box and
        crop the region.
    else if the number of connected components is 2 then
        the sign is double-handed and so, the following steps are applied:

        Find minx (left most point of left hand), miny (top most point of the hand which is
        above), maxx (left most point of right hand) and maxy (top most point of the hand
        below).

        Find the width difference (maxx-minx) and add the width of right hand. This gives the
        total width of the bounding box.

        Find the height difference (maxy-miny) and add the remaining height of the hand
        extending below. This gives the total height of the bounding box.

        Draw bounding box using the width and height calculated above and crop the double-
        hand region.
    end
end
```

Figure 8. Algorithm for bounding box construction.

- If the labels predicted by the three models are all different, then predict the label given by SURF as the original output label.

3.5 *ISL sentence interpretation using inverted index*

The final stage is to connect the input sign(s) (figure 9) into a meaningful word (figure 10) or sentence. This includes several challenges as ISL differs in phonology, morphology, grammar and syntax from other country languages. Moreover, it has several conflicting signs such as "positive" and "add", which could be interpreted only based on the context. Also, some signs can be interpreted into the right words with the help of facial expressions, which are not dealt by the system.

In order to provide a clever solution to the interpretation of sentences based on a limited vocabulary defined by the dataset used, inverted indexing concept is adopted [19].

3.5.1 *Index table creation* Each keyword is stored along with sentence number list in index table. For each insertion of keyword, the index table is scanned and corresponding entry is updated.

3.5.2 *Finding the output sentence* Once the index table is created, the next step is finding the possible sentence based on input keywords. After finding sentence numbers against each input keyword, intersection operation is applied on each sentence numbers set, the final sentence number is retrieved and corresponding sentence (table 3) is displayed as text (figure 11) and speech.

4. Experimental results

At every stage of the process, there are a number of factors upon which the accuracy depends and varies accordingly.

- In feature extraction and recognition, the important parameter is the "cell size". The input image, at this stage, gets divided into a number of cell groups based on size, after which feature vectors are generated for each cell. The cell size can be 2 × 2, 4 × 4, 8 × 8, etc.

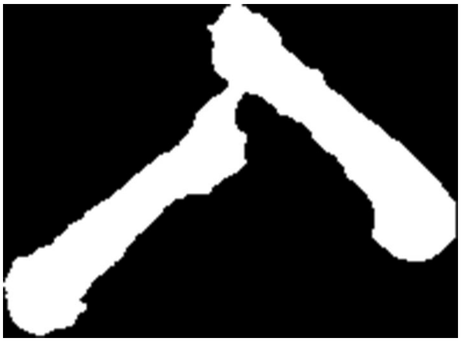
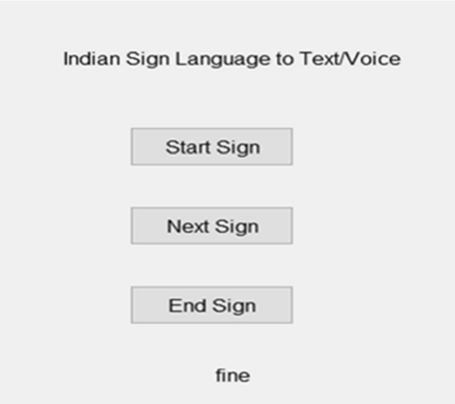


Figure 9. Input gesture for the word “Fine”.



Indian Sign Language to Text/Voice

fine

Figure 10. Resulting text.

We conducted experiments by varying the cell size for each of the 3 feature classes, and it can be observed that the accuracy achieved is greater when a cell size of 4 × 4 is used for SURF feature and 8 × 8 for HOG and LBP features (figure 12, table 4).

- When individual features were used separately, we were able to achieve overall system accuracy as shown in figure 13 and table 4.
  However, the usage of feature ensembling technique has improved the overall accuracy.

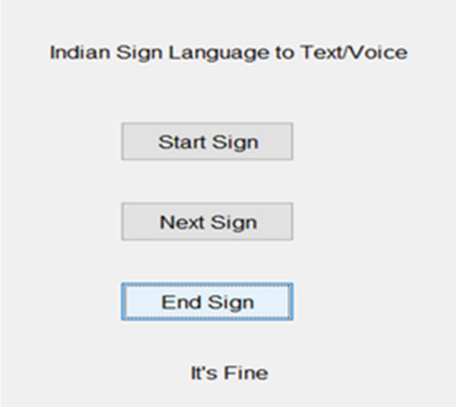
4.1 *Class-wise accuracy*

Table 5 and figure 14 show the class-wise accuracy for 115 classes. We achieved 100% accuracy for classes 9, 11, 14, 26, 27, 38, 41, 42, 47, 55, 67, 74, 77, 85, 86, 96, 100, 109 and 111. The dataset was divided in the ratio 80:20. Training set consists of 3680 images and test set consists of 920 images with 8 images per class.

- In the fingerspelling category (table 2), we achieved
  - 100% accuracy for the signs representing 9, A, F, G, H, N and P.
  - 75% accuracy for the signs representing 1, 2, 6, 7, 8, C and I-single handed.

Table 3. Possible sentences list.

Keywords	Possible ISL interpretations
“Mail, me”	“Send me a mail”
“I, fine”	“I am fine”
“Sample, leprosy”	“The sample shows signs of leprosy”
“How, you”	“How are you”
“B, bottle”	“B for Bottle”
“Blow, leaf”	“Blow the leaf away”
“You, strong”	“You are strong”
“What, weigh”	“What is your weight?”
“Vacation, fine”	“The vacation was fine”
“Unit, strong”	“Unity is strength”
“Boxing, strong	“Boxing makes us strong”
“Keep, up”	“Keep it up”
“Bud, perfect”	“The bud looks perfect”
“Appreciation, work”	“Appreciation of work”
“Deposit, thick, density”	“The deposit has thick density”
“I, mail, you”	“I will mail you”
“Measure, density”	“Measure the density”
“Negative, opposite, good”	“Negative is the opposite of good”
“Snake, bite, me”	“A snake bit me”
“I, lost, weight”	“I have lost my weight”
“Add, small, quantity, sugar”	“Add small quantity of sugar”
“Learn, respect”	“Learn to respect”
“I, good, friend”	“I am a good friend”
“Buds, specific, season”	“These buds are specific to a season”
“Opposition, control”	“Opposition is in control”
“Pray, Varanasi”	“Pray at Varanasi”



Indian Sign Language to Text/Voice

It's Fine

Figure 11. Interpreted sentence.

- – Overall, 8 distinct signs were recognized with an average rate of 87.50%.

4.2 *Performance metrics*

Several other relevant performance metrics are evaluated, and the values obtained are as shown in table 6. The following subdivisions explain the parameters used. Here, *tp*

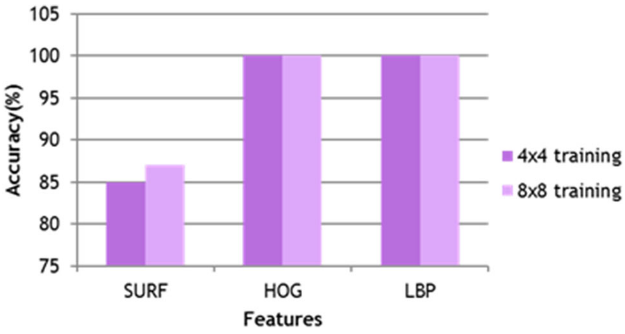


Figure 12. Accuracy for training set.

Table 4. Accuracy—training and testing.

Features	Cell size	Training	Testing
LBP	8 × 8	100	68.59
SURF	4 × 4	85	68.00
HOG	8 × 8	100	66.30

stands for true positive, *tn* stands for true negative, *fp* stands for false positive and *fn* stands for false negative.

4.2.1 *Gamma evaluation* Gamma is used to configure the sensitivity to differences in feature vectors:

$$gamma = \frac{1}{n} \tag{1}$$

where n is the number of features.

4.2.2 *Precision* Precision or positive predictive value is defined as the proportion of the true positives against all the positive results (both true positives and false positives):

$$precision = \frac{tp}{tp + fp}. \tag{2}$$

4.2.3 *Negative predictive value* Negative predictive value (NPV) is the proportion of negative results in statistics and diagnostic tests that are true negative results:

$$NPV = \frac{tn}{tn + fn}. \tag{3}$$

4.2.4 *Sensitivity* Sensitivity relates to the test's ability to identify positive results:

$$sensitivity = \frac{tp}{p} = \frac{tp}{tp + fn}. \tag{4}$$

4.2.5 *Specificity* Specificity relates to the test's ability to identify negative results:

Table 5. Class-wise accuracy.

1)	75%	24)	88%	47)	100%	70)	50%	93)	75%
2)	75%	25)	88%	48)	88%	71)	88%	94)	38%
3)	50%	26)	100%	49)	75%	72)	50%	95)	38%
4)	25%	27)	100%	50)	50%	73)	63%	96)	100%
5)	50%	28)	75%	51)	75%	74)	100%	97)	63%
6)	75%	29)	63%	52)	63%	75)	38%	98)	63%
7)	75%	30)	75%	53)	75%	76)	50%	99)	25%
8)	75%	31)	75%	54)	50%	77)	100%	100)	100%
9)	100%	32)	88%	55)	100%	78)	38%	101)	88%
10)	88%	33)	50%	56)	38%	79)	63%	102)	88%
11)	100%	34)	75%	57)	75%	80)	75%	103)	88%
12)	88%	35)	88%	58)	50%	81)	88%	104)	88%
13)	50%	36)	63%	59)	75%	82)	88%	105)	75%
14)	100%	37)	88%	60)	75%	83)	63%	106)	75%
15)	25%	38)	100%	61)	25%	84)	88%	107)	38%
16)	88%	39)	75%	62)	88%	85)	100%	108)	63%
17)	63%	40)	63%	63)	63%	86)	100%	109)	100%
18)	63%	41)	100%	64)	88%	87)	88%	110)	75%
19)	75%	42)	100%	65)	88%	88)	50%	111)	100%
20)	25%	43)	88%	66)	75%	89)	63%	112)	88%
21)	63%	44)	63%	67)	100%	90)	38%	113)	75%
22)	38%	45)	63%	68)	75%	91)	50%	114)	50%
23)	88%	46)	63%	69)	75%	92)	50%	115)	63%

Table 6. Performance metrics.

Metric	Value
Gamma	1.13722E–8
Precision	73.61%
Negative predictive value	99.75%
Sensitivity	71.85%
Specificity	99.75%
Miss rate	28.15%
False discovery rate	26.39%
*F*1 score	71.45%
Response time (sample)	35176 ms

$$specificity = \frac{tn}{tn + fp}. \tag{5}$$

4.2.6 *Miss rate* Miss rate can be viewed as the percentage of misses per total tries:

$$miss\ rate = \frac{fn}{p} = \frac{fn}{fn + tp}. \tag{6}$$

4.2.7 *False Discovery Rate* False discovery rate (FDR) is the expected proportion of discoveries (results) that are false (incorrect rejections):

$$FDR = \frac{fp}{fp + tp}. \tag{7}$$

4.2.8 *F1 score* $F1$ score is the harmonic mean of precision and sensitivity:

$$F1 = \frac{2tp}{2tp + fp + fn}. \quad (8)$$

4.2.9 *Response time* Response time is the total time taken for the system to recognize a sign and display the output from the time the user starts performing a sign. The response time is largely dependent on the sensors' performance.

4.3 *Pre-test*

Considering only the input and output, with the system as a black box, we observe that for a sample input sign sequence, say: "Pray, Varanasi", the system interprets the output sentence: "Pray at Varanasi". Furthermore, other input signs and their corresponding ISL interpretations by the system are tabulated in table 3. The performance of the proposed method has been captured through various metrics as shown in table 6. Certain ISL words have geometrically exact signs and can be understood only in a specific context. These signs are handled as conflicting signs (table 7). For our experimentation, we have restricted the usage of each such conflicting sign to mean a specific exact word at all times.

4.4 *Post-test*

The experimental results have been documented and performance of the system has been measured (table 6).

5. Results analysis and evaluation

5.1 *Comparison with the state-of-the-art*

Since [14] is the only recent work on ISL after standardization, we have focused on comparison of our work with [14]. Table 8 shows a few metrics that highlight the performance of proposed work in comparison with existing work.

The existing system [14] has 5041 images in the dataset. For our experimentation purposes, we have used an extended version of the same dataset as used in [14], by appending gestures from a few more subjects. The dataset also has many conflicting signs (table 7): signs whose hand regions are similar with differences only in facial expressions and body language, based on the context of usage. For our experimentation we have restricted the usage of each such conflicting sign to mean a specific exact word at all times. The average recognition accuracy of the proposed system in comparison with the existing system [14] is presented in figure 15. Since we have added multiple variations of every sign to the dataset, our system was able

Table 7. Conflicting sets of words.

1	Up	Warn	
2	K-single handed	V-single handed	V
3	W-Single Handed		
6	Okay		
7	L	L-single handed	
9	W-single handed		
R	Respect		
Appreciation	There		
Between	Intermediate		
Boxing	Control	Strong	
B-single handed	Season		
Conservation	Keep		
Few	Little	Unit	
Friend	Promise		
F-single handed	Good		
Ghost	Paw		
It	You		
Learn	Study		
Opposite	X		
Potentiality	S-single handed		
Quantity	Sample		

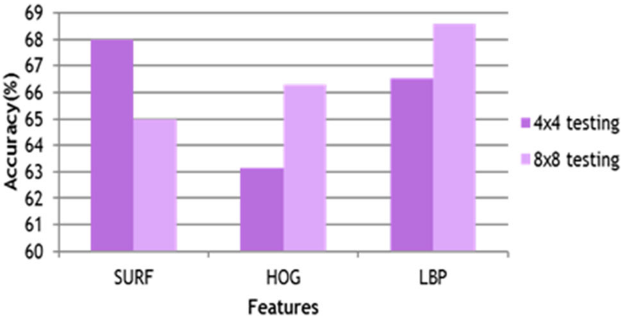


Figure 13. Accuracy for test set.

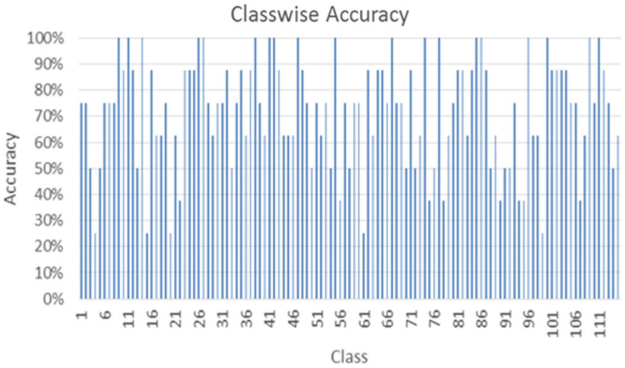


Figure 14. Class-wise accuracy graph.

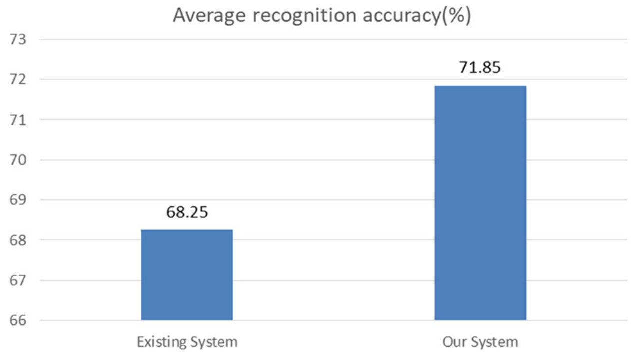


Figure 15. Average recognition accuracy of our system vs. existing system [14].

to identify the features, and hence the signs, with much distinction. Moreover the feature size of each image is 19116, which is one of the reasons for delayed responses.

5.2 *Error analysis*

From table 6, it is evident that the accuracy of the proposed system on the chosen dataset comes out to be 71.85%. The NPV and specificity record higher values, whereas the FDR and sensitivity are quite low. This is primarily driven by the segmentation threshold value we have estimated (as discussed in section 3.2.1) while performing *K*-mean clustering, which is then fed into the SVM classifier. In this case, we look for interpretations with minimal false positives. Hence, we are more concerned about higher specificity. ISL has nearly similar signs for different words, leading to an almost equivalent feature vector. This affects the recognition rate.

Most of the erroneous translations are attributed to WSD – word sense disambiguation, where the same word has multiple meanings. Our system does not handle WSD. Exceptions exist in the translation when the user performs signs that are not a part of the dataset. The response time is largely dependent on the sensor performance (here Kinect). There is also an observed trade-off between accuracy and response time. In addition, response time is also affected by the algorithms used. The response time we have observed does not suit the needs of true real-time gesture translation; thus, more work needs to be done in this direction.

6. Conclusion and future scope

The proposed system takes, as input, a sequence of hand gestures in ISL, and outputs a meaningful English sentence. We have developed a system to overcome most of the challenges unique to ISL, to enable smooth communication between the abled and the HSI. We have managed to improve the average recognition accuracy up to 71.85% with our method. We achieved 100% accuracy for the signs representing 9, A, F, G, H, N and P. However, our method does not take into account the context of gestures, leading to erroneous translations on a few occasions.

Future scope for this system is as follows:

- By including more signs from various domains, the dataset can be expanded, thereby achieving a more efficient system for real-time applications.
- The sentence table could be appended with more sentences with respect to different keywords and combinations of keywords.
- Improving the response time is the key to true real-time applications.
- The language dictionary used can be updated dynamically every time a user performs a sign, making the system even more user friendly.

Table 8. Comparison with state-of-the-art.

Criterion	*Ansari and Harit* [14]	*Agarwal and Thakur* [22]	Proposed
Dataset	ISL (140 signs) – 18 subjects	Chinese (CGD 2011)	ISL (140 signs) – 21 subjects
Dataset composition	Alphabets, digits and words (both single and double handed)	Digits only (0–9)	Alphabets, digits and words (both single and double handed)
Average recognition accuracy (only on numbers 0–9, %)	68.25	87.67	71.85
Features	SIFT, SURF, VFH, NN	Only HOG	Ensembles HOG, SURF and LBP
Method used	Uses depth profile and classifies based on NN	Creates depth and motion profile and trained on SVM	Uses just depth profile and ensembles features trained on SVM
Gesture translation to sentences	–	–	Translates gesture sequence to English sentence

- Conflicting words are handled using "/" in our work. For instance, if a user performs a sign corresponding to "+", it will be interpreted as "add/positive". This can be made context sensitive, in order to provide better interpretation based on the usage in the context.
- The system could be extended to work for signs that involve movements (dynamic gestures) as well.
- Explore new sensors with improved response times for true real-time gesture translation.
- Envisage sensor-independent methodology for universal applications.

References

[1] Biswas K K and Basu S K 2011 Gesture recognition using Microsoft Kinect®. In: *Proceedings of the 5th International Conference on Automation, Robotics and Applications (ICARA)*, Wellington, New Zealand, 6–8 December, pp. 100–103

[2] Parton B S 2006 Sign language recognition and translation: a multidisciplined approach from the field of artificial intelligence. *J. Deaf Stud. Deaf Educ.* 11(1): 94–101

[3] Indian Sign Language Research and Training Centre (ISLRTC) http://www.islrtc.nic.in/

[4] Viswanathan D M and Idicula S M 2015 Recent developments in Indian sign language recognition: an analysis. *Int. J. Comput. Sci. Inf. Technol.* 6(1): 289–293

[5] Mitra S and Acharya T 2007 Gesture recognition: a survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 37(3): 311–324

[6] Vijay P K, Suhas N N, Chandrashekhar C S and Dhananjay D K 2012 Recent developments in sign language recognition: a review. *Int. J. Adv. Comput. Technol.* 1(2): 2126.

[7] Nagashree R N, Stafford M, Aishwarya G N, Beebi H A, Jayalakshmi M R and Krupa R R 2015 Hand gesture recognition using support vector machine. *Int. J. Eng. Sci.* 4(6): 42–46

[8] Dardas N H and Georganas N D 2011 Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Trans. Instrum. Meas.* 60(11): 3592–3607

[9] Yang C, Jang Y, Beh J, Han D and Ko H 2012 Recent developments in Indian sign language recognition: an analysis. In: *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, pp. 297–298

[10] Padmavathi S, Saipreethy M and Valliammai V 2013 Indian sign language character recognition using neural networks. *Int. J. Comput. Appl. (Special Issue: Recent Trends in Pattern Recognition and Image Analysis)* (1): 40–45

[11] Madhuri S, Ranjna P and Soho A K 2014 Indian sign language recognition using neural networks and KNN classifiers. *ARPN J. Eng. Appl. Sci.* 9(8): 1255–1259

[12] Kim K, Kim S K and Choi H I 2015 Depth based sign language recognition system using SVM. *Int. J. Multimed. Ubiquitous Eng.* 10(2): 75–86

[13] Wang Y and Yang R 2013 Real-time hand posture recognition based on hand dominant line using Kinect. In: *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, San Jose, CA, USA, 15–19 July, pp. 1–4

[14] Ansari Z A and Harit G 2016. Nearest neighbor classification of Indian sign language gestures using Kinect camera. *Sadhana* 41(2): 161–182

[15] Tiwari V, Anand V, Keskar A G and Satpute V R 2015 Sign language recognition through Kinect based depth images and neural network. In: *Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, pp. 194–198

[16] Wang X, Han T X and Yan S 2009 *An HOG–LBP human detector with partial occlusion handling*. In: *Proceedings of the IEEE 12th International Conference on Computer Vision*, pp. 32–39

[17] Halim Z and Abbas G 2015 Kinect-based sign language hand gesture recognition system for hearing- and speech-impaired: a pilot study of Pakistani Sign Language. *Assist. Technol. J.* 27(1): 34–43

[18] Nagarajan S and Subashini T S 2013 Static hand gesture recognition for sign language alphabets using edge oriented histogram and multi class SVM. *Int. J. Comput. Appl.* 82(4): 28–35

[19] Ghotkar A S and Kharate G K 2015 *Dynamic hand gesture recognition and novel sentence interpretation algorithm for Indian sign language using Microsoft Kinect sensor*. *J. Pattern Recognit. Res.* 1: 24–38

[20] Van den Bergh M and Van Gool L 2011 Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In: *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, Kona, HI, pp. 66–72

[21] Mohandes M, Deriche M and Liu J 2014 Image-based and sensor-based approaches to Arabic Sign Language recognition. *IEEE Trans. Hum. -Mach. Syst.* 44(4): 551–557

[22] Agarwal A and Thakur M K 2013 Sign language recognition using Microsoft Kinect. In: *Proceedings of the Sixth International Conference on Contemporary Computing (IC3)*, Noida, pp. 181–185. https://doi.org/10.1109/IC3.2013.6612186

[23] Fujimura K and Liu X 2006 Sign recognition using depth image streams. In: *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR'06)*. https://doi.org/10.1109/FGR.2006.101