Sign Language Recognition Using Hand Mark Analysis for Vision-Based System (HMASL)

Akansha Tyagi and Sandhya Bansal

Abstract Sign language recognition (SLR) is an essential study area that allows us to provide a better communicating environment between humans and computers. Some prevailing and standard features extracted from sign language gestures include scale-invariant feature transform (SIFT), speeded-up robust feature (SURF), features from the accelerated segment (FAST), and oriented FAST and rotated Brief (ORB) are used. However, these element vectors contain a few highlights that are insignificant or excess, subsequently expanding the generally computational time just as acknowledgment error of a classification framework. To counter this issue, we have proposed another object detection calculation dependent on profound hand math. A novel approach called Hand mark analysis of sign language (HMASL) has been used in this concern. It combines the concept of feature extraction and hand geometry to reduce the computation and computes only and region in complex background. HMASL is compared to other classical feature extraction method and tested on several classifiers. The experimental results show that the HMASL eases the feature aspect to a meaningful amount as well as surges the recognition accuracy.

Keywords Indian sign language · Computer vision · Feature extraction · Hand geometry · Deep learning · SVM

1 Introduction

Sign language recognition (SLR) applies to many domains featured for the deaf-mute community. Even though various device-based recognition systems like sensors, gloves have been recently used, but vision-based recognition is more approachable. Vision-based recognition is becoming widespread due to the significant scope of application areas as found in the literature [1]. However, Indian Sign Language (ISL) comprises of 6000 words which are commonly used in Indian country.

A. Tyagi (✉) · S. Bansal
Department of Computer Science and Engineering, Maharishi Markandeshwar(Deemed to be) University, Ambala, Haryana, India

N. Marriwala et al. (eds.), *Emergent Converging Technologies and Biomedical Systems*,
Lecture Notes in Electrical Engineering 841,
https://doi.org/10.1007/978-981-16-8774-7_35

SLR using machine learning and soft computing has been a ground of interest for a long time. Scientists have utilized a few methodologies and have made a ton of progress in preparing distinctive machine and profound learning models that can perceive signs comparing to various words. Most of the study that has been done is for American Sign Language (ASL), and the systems require the utilization of some sort of movement sensors or hand gloves to distinguish the places of various fingers precisely. The way that these methodologies are no uncertainty successful and can represent pretty much every sign, except these require the utilization of some exceptionally delicate equipment that cannot be utilized by everybody and commonly require explicit climate. Some different ways to deal with perceive communications through signing incorporate the utilization of deep learning models that work on skin enclosed images. Skin veiled pictures are framed by portioning out the part from the picture which coordinates with the shade of the skin. That area is given a particular tone (white), and all the rest pixels in the picture are doled out in another tone (dark). In such methodologies after skin veiling, significant highlights are extricated from the pictures utilizing a few strategies like SIFT, SURF, FAST, ORB, and profound learning models are prepared for arranging various signs. These methodologies have demonstrated to be quick progressively, yet the utilization of profound learning models requires the utilization of more assets, and they probably will not have the option to perform so well on basic devices having restricted resources [2–4].

The recent success of deep learning approaches in a task like an image classification [5] has been extended to the problem of sign language recognition [6]. Unlike other traditional soft computing methods such as neural network, KNN, or genetic algorithm (GA) where features were extracted manually, while neural network models learn features from the training database [7]. These networks save the spatial design of the issue and were created for object recognition roles, for example, manually written digit acknowledgment. They are famous because individuals are accomplishing cutting edge results on troublesome computer vision and normal language training tasks.

Another approach is that researchers have popularly started using this hand mark analysis methods; that is, hand geometry parameters are combined with graphical properties such as open pose and hand pose. The analysis of the shape and geometry of the hand provides the essential features of the hand. These methods have shown an impeccable result and giving an elevated recognition accuracy without using any sensor devices. These methods follow the state-of-the-art techniques, that is, to locate a set of essential key points representing the position of coordinates with the help of some neural network models. The sole issue with this technique is that even though it can work progressively, it requires a decent number of the dataset, and it gives a speed of 0.1 to 0.3 frames each second for the video input which is not acceptable in any way. It cannot handle outlines easily continuously.

Our methodology integrated the distances between the 0th central key point (the central key point at the extremely base in the palm) and the remainder of the 20 central key point as highlights. A hand geometry model is utilized to return the standardized directions for these central points; i.e., it returns the central key point by partitioning the x arranged by the width of the frame and y by the height of the frame, however, for a superior standardization, the new coordinates are determined by moving the root to the 0th point itself. Presently, we have the situation of central key points concerning the 0th central key points. Accordingly, the area of the hand will not have a lot of impact on the directions of these central features, and the model wants to deal with a broader scale.

This paper is composed of six sections: Sect. 2 discusses the literature review. Section 3 discusses the dataset used in this paper. Proposed work is discussed in Sect. 4. Experimental work and results are discussed in Sect. 5. Lastly, Sects. 6 and 7 highlight the conclusion and future scope of this work, respectively.

2 Literature Review

The communication between human beings is carried out in spoken form by speech and non-verbal through gestures. Generally, people make gestures either consciously or unconsciously while communicating with others. Non-verbal communication among the deaf and mute community is known as sign language. They use their two hands for making gestures to communicate among themselves. The sign language among the community who live in India is known as Indian Sign Language (ISL). ISL is composed of static and dynamic gestures precisely. Indian Sign Language recognition (ISLR) is a better approach for devolving a vision-based gesture recognition system that can help the above community to bridge the communication gap. The concept of computer vision has facilitated the ISLR area for research [8–10]. Various feature extraction and soft computing algorithms have been developed to train a model by using the above steps. Most of these techniques are deployed in content-based image retrieval (CBIR) features followed by classifier such as support vector machine (SVM) [11, 12], linear discriminant analysis (LDA) [13], neural network [14–16], and convolution neural network (CNN) [17–20].

In ISLR, the research work has undertaken from pre-processing of gestures to recognize gestures directly through CNN, while SIFT has evolved as the most promising technique in terms of feature extraction [21]. Here, [22] has used SIFT algorithm for feature detection and object matching on real-time images and achieved 60% more accurate results without performing pre-processing of images. To overcome the challenges in ISLR such as the requirement of constant illumination and wearing long attire sleeves for natural background constraint, an ISLR based on pixel-based segmentation and advanced SIFT is proposed [23]. Further, due to the invariant characteristic of SIFT, over-illumination, rotation, translation, scaling, and slightly to viewpoint [24] have implemented various phases of SIFT to extract features from ISL gestures. Each image has more than 400 features with the highest peak of 80% in the bag of visual words (BOG) providing a reliable matching between disrupted images. Instead of using conventional methods which take more computation time, an improved SIFT with a fuzzy closed-loop control method has been used for object recognition in the cluttered environment [3]. Another study [25] has elaborated on SIFT and CNN-based image retrieval processes and how they enhance the system's performance. CNN works on a large dataset and extracts features from images as by layers increase, but applying SIFT for refining of features reduces the model layers and improves accuracy in few epochs.

Bedregal et al. [26] used fuzzy for recognition of LIBRAS (Brazilian Sign Language) gestures. Hand gestures are classified using a set of angles of finger joints and their segmentation. A set of finite automata is created for the segmented gestures which are classified using fuzzy rule which enhances the classification accuracy of the system. Christian Zimmermann and Thomas Brox [27] uses a deep network for the classification of 3D hand pose using RGB pose estimation using low-cost customer depth cameras for 35 static German Sign Language (GSL) symbols. Albanie et al. [28] followed the co-articulation method to classify the British Sign Language (BSL) signs. A dataset of 1000 keywords in 1000 h of video is also created to automatically localize the sign-instances keywords. Kang et al. [29] proposed an efficient method using a depth map to recognize the fingerspelling gestures. Images were captured using the depth sensors following by some image pre-processing techniques that are then classified using the convolution neural network (CNN). The proposed system achieved an accuracy of 99.99%. Li et al. [30] proposed a vision-based sign language recognition system for 2000 words/glosses. Two deep learning models were approached, one is based on visual appearance, and another is based on a 2D human pose. The proposed model has achieved an accuracy of 62.63% at top-ten words.

From the literature survey, it can be concluded that hand mark analysis or hand geometry is an important part of the ISLR. Feature extraction and selection of essential key points considering redundancy and relevancy of features can do better performance. This has motivated us to develop hand mark analysis which can be hybridized with feature extraction technique FAST-SIFT to form HMASL. The evaluation of our HMASL model is done on several classification models.

3 Dataset and Pre-processing

Sign Language Dataset: The two-hand gesture ISL words (“afraid,” “agree,” “bad,” “become,” “chat,” “college,” “from,” “today,” “which,” “you”) images are captured in uniform background as no standard dataset is available. This dataset is extended by superimposing on complex backgrounds. The samples of the dataset are shown in Fig. 1. It contains a total number of 3000 images of 300 for each class. The images are in RGB mode. This dataset is also made publicly available for further usage of ISLR.

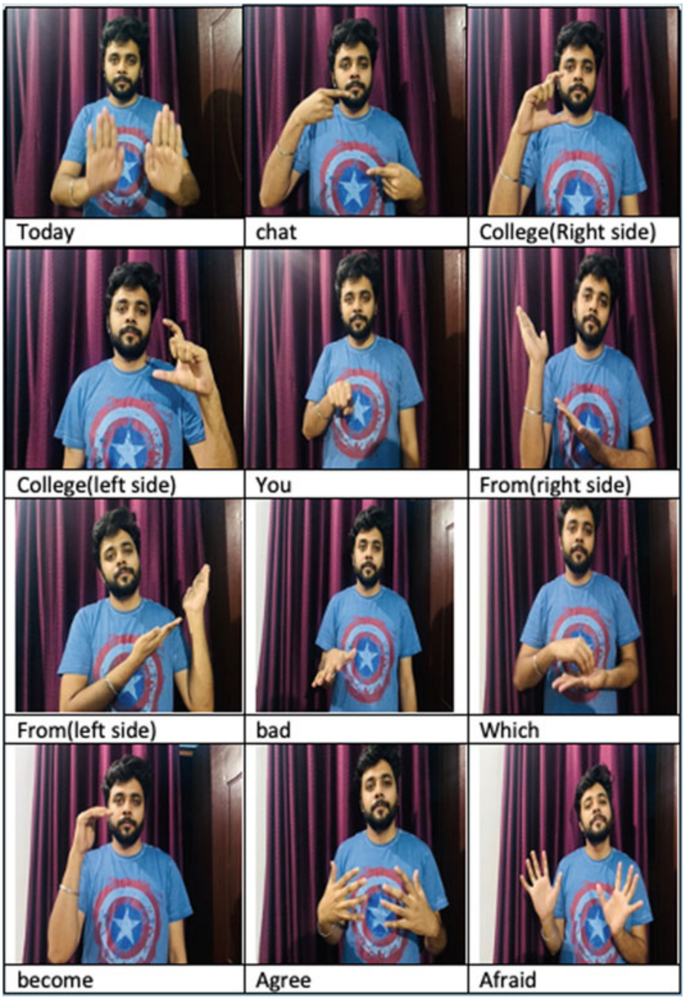

Fig. 1 Sample images from the ISL word dataset

4 Methodology

The necessary task is to extract features that are pertinent to any model and to eliminate or dispose of the ineffective pixels inside each picture test preventing the hand region. Thus, HMASL is utilized to distinguish the area of interest, that is, the area containing fundamental features called key points.

The model used for performing sign language recognition is to first process the images from the dataset. Then, we extract the features using the FAST-SIFT (FiST) algorithm from the training set. Other tools of hand geometry such as the bounding box are also used to perform the background segmentation. These images are then

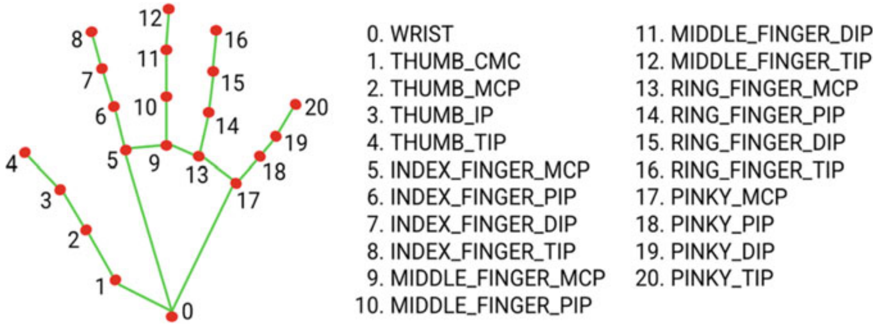

Fig. 2 Key points detected by HMASL

passed to the hand mark analysis model to detect the 21 3D hand knuckle coordinates inside the region of interest detected by the FiST. The detected hand landmarks are then passed to two separate functions. Twenty-one landmarks detected by HMASL are shown in Fig. 2.

The first function computes coordinated after the hand is moved to the 0th central points, and the subsequent capacity determines the Euclidean distance between the 0th central points and the remainder of the fundamental points. The model learns a reliable interior hand position representation and is powerful to halfway visible hands and self-impediment. To all the more likely that cover the feasible hand motions and give extra oversight on the idea of hand math, we likewise utilized complex foundations and guide it to the comparing 3D directions. At that point, this cropped area is given as aid to a second model that perceives the situation of hand landmarks. Presently, the new coordinates, distances, and the handedness (left or right) are given as a contribution to the classifier model which predicts and returns the class relating to the sign. The model can likewise have the option to effectively choose whether it is the right or the left hand.

The working of the HMASL is discussed in the above steps:

1. Capture image using the web camera of laptop.
2. Semantic segmentation is done to detect the different regions in an image and locate their respective labels. Constrained our focus here is to segment hand from the image. There are two stages to perform it:

a. Detect the hand region from the image and segment it.
b. Compute the number of fingers in the detected hand region.

3. Background subtraction: Compute the running average time over the current frame and previous frame using Eq. (1).

$$R_t = \frac{C_F}{P_F} \tag{1}$$

where R_t is the average running time, C_F is the current frame, and P_F is the previous frame.

4. The object in the background will be transformed into black, and only the hand gesture will appear in the foreground by applying the mask on the background objects. After figuring out the background B_k model using running averages R_t, now, we will use C_F which holds the foreground object F_O in addition to the background.
5. An absolute difference is calculated between the background model and the current frame to find the foreground object using Eq. (2).

$$F_O(I) = B_k - C_F \tag{2}$$

6. Thresholding: Thresholding is an assigning process of pixel intensities to 0's and 1's based on a certain threshold value, so that an individual object can be detected from an image using Eq. (3).

$$D_T(I) = T[F_O(I)] \tag{3}$$

where $D_T(I)$ is the threshold image, T is the threshold value applied to the image, and $[F_O(I)]$ is the image that contains the object. The threshold will convert unwanted regions into black.

7. Contour extraction: Result from Step 4 $D_T(I)$ is used to find the contour (C), which is an enclosed boundary of the gesture with the pixel structure that has the highest intensity. Let $D_T(I) = (x_i, y_i)$ be the edge coordinate in the edge list, and k is the angle between the direction vector and k edges. Suppose that there are n edge points (x_i, y_i), …, $(x_n\text{'}y_n)$ in the edge list. The length of a digital curve can be approximated by adding the lengths of the individual segments between pixels using Eq. (4):

$$\mathrm{C} = \sum_{i=2}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{4}$$

8. Find the approximation contour (C_{DT}), the total distance between the endpoints using Eq. (5):

$$C_{DT} = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2} \tag{5}$$

9. Find the moments, that is, pixel intensity and their corresponding location using Eqs. (6) and (7).

$$M_{ij} = \sum_x \sum_y x^i y^i I(x, y) \tag{6}$$

$$M_{ij} = m_{00}, m_{01}, m_{02}, m_{03}, m_{04}, m_{05} \ldots\ldots\ldots.m_{30} \tag{7}$$

10. Find the area of the contour and its perimeter using the moments, it will help in the case of oriented gesture and the gestures which have different dimensions, and recognition can be done using the area. The area will be a composite analysis of contour moments as shown in Eqs. (8) and (9):

$$\text{Area(A)} = C_{DTM_{ij}}. \tag{8}$$

$$\text{Perimeter(P)} = AR_{LT}.C_{DT} \tag{9}$$

where AR_{LT} is the arc length of the contour curve.

11. A convex hull is now created over the detected object to check the curve for convexity defects and correct them. It will help us to find out the bulged-out or the flat hand regions by using Eq. (10):

$$H_C = Co_H.[C_{DT}] \tag{10}$$

where H_C is the hull, Co_H is the convex hull, and C_{DT} is the contour moments.

12. A bounding box is then created over the H_C region, and further feature extraction methods are applied.
13. Non-max suppression or FiST is used to locate and compute the key points (K_p) using Eqs. (11) and (12):

$$DoG = DoG + \frac{\partial DoG^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 DoG^T}{\partial x^2}x \tag{11}$$

$$K_p = \sum_{i,j=0}^{I=t} DoG \tag{12}$$

where DoG is the difference of Gaussian used to compute values of K_p, while K_p is the key points calculated from each image.

14. The resultant K_p is then located on the $F_O(I)$ image, the result from H_C is combined, and a graph is formed to link all the essential features and store them according to their coordinates values determined for all training images.
15. The resultant data are then provided to the classifier in array, and classification will be performed.
16. The models are saved for prediction.
17. Results are analyzed based on confusion matrix, recall, precision, and F1-score calculated from experiments.

These steps are repeated for all the training images, and further results are generated over testing images. Our HMASL model has acquired a remarkable accuracy over ISL words gestures. The flowchart of the HMASL is shown in Fig. 3. The sample image of the word "college" is taken in the flowchart.

Flowchart:

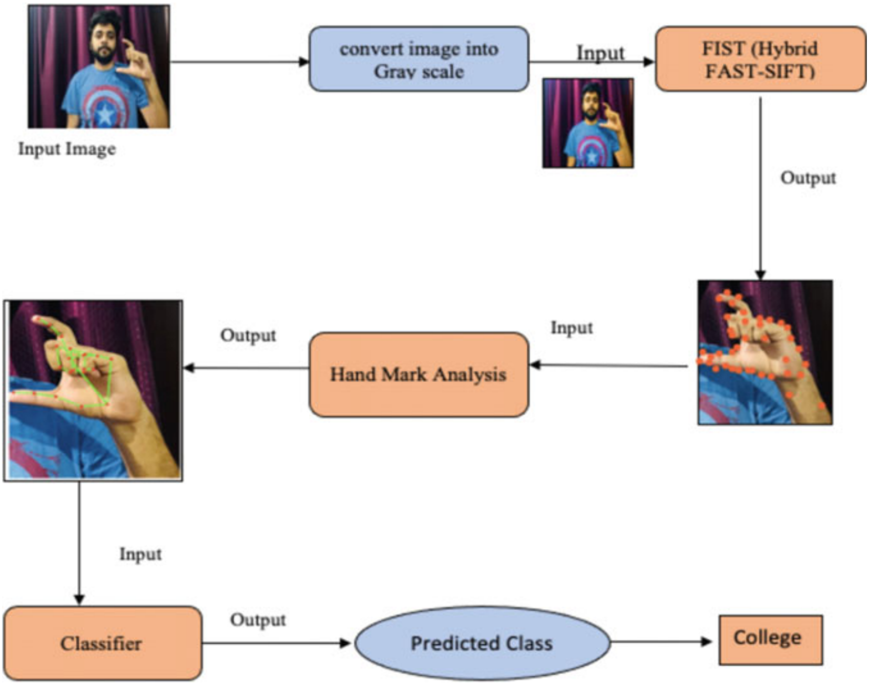

Fig. 3 Flowchart for the process

5 Experiment and Result

5.1 Experimental Setup

Python 3 Jupyter Notebook has been used for performing the experiments presented in this article. Specifications of the system are: Intel® Core™ i5-@1.8 GHz, 8 GB RAM, and 256 cache per core, 3MB cache in total. Graphics with GPU type with VRAM 1536 MB. TensorFlow is used as the backend for the CNN model. To store FAST-SIFT key points, NumPy commands have been used. VLFeat, CuPy, Scikit-learn, and CUDA can also be used on Windows or Linux platform. These experiments are performed on the macOS platform.

All the models are trained on 2600 images present in the dataset. The public dataset is used for the validation of the model. Each network is trained for 20 epochs with a batch size of 128. An accuracy of 96.74 is achieved by the proposed model on the deep learning models. The proposed methodology is tested on several other classifier such as SVM, MLP, and KNN. The results of HMASL on several classifiers are shown in Table 1.

The confusion matrix here is used to summarize the performance at classification stage. A good classifier represents a sparse matrix in the form of graph. Symbols

Table 1 Results of HMASL on different classifiers

Classifier	Accuracy	Precision	Recall
SVM	96.73	99	99
MLP	95.65	97	98
KNN	93.47	96	98
NN	96.34	98	96

are represented by X-axis, while the predicted class is represented by 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 is equal to Y, then it shows the accurate classification. The confusion matrix in Fig. 4 represents the misclassification between gestures (1–10) in terms of precision and recall per gesture, with an average classification accuracy of 96.73% on the SVM classifier.

Likewise, the confusion matrix in Fig. 5 refers to the MLP classifier with an average accuracy of 95.65%. Figures 6 and 7 represent the accuracy of KNN and NN classifier, that is, 93.47% and 96.34%, respectively.

Precision for the precisely identified gestures to the number of particular predicted gestures is specified by using the formula shown in Eq. (13).

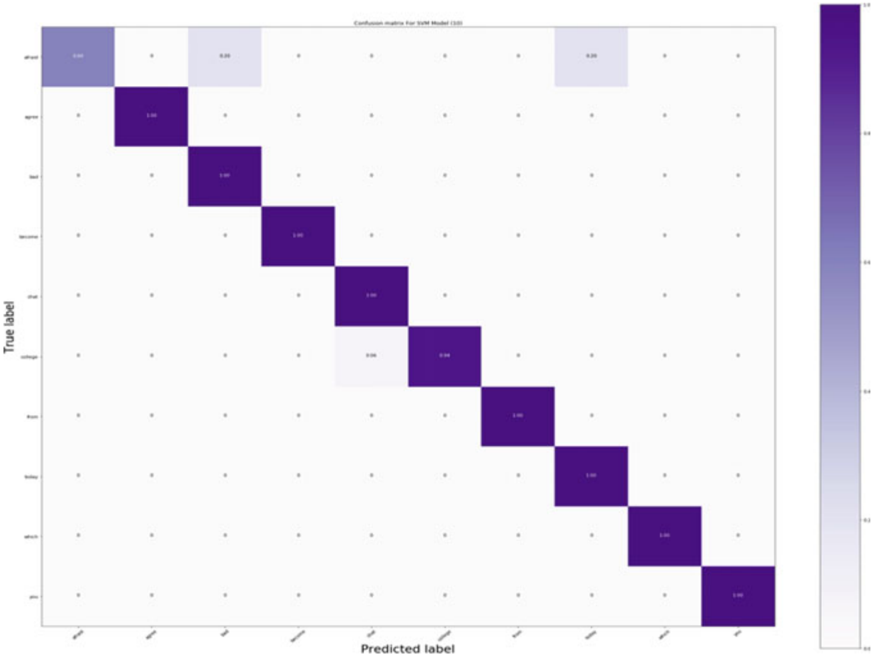

Fig. 4 Confusion matrix for SVM

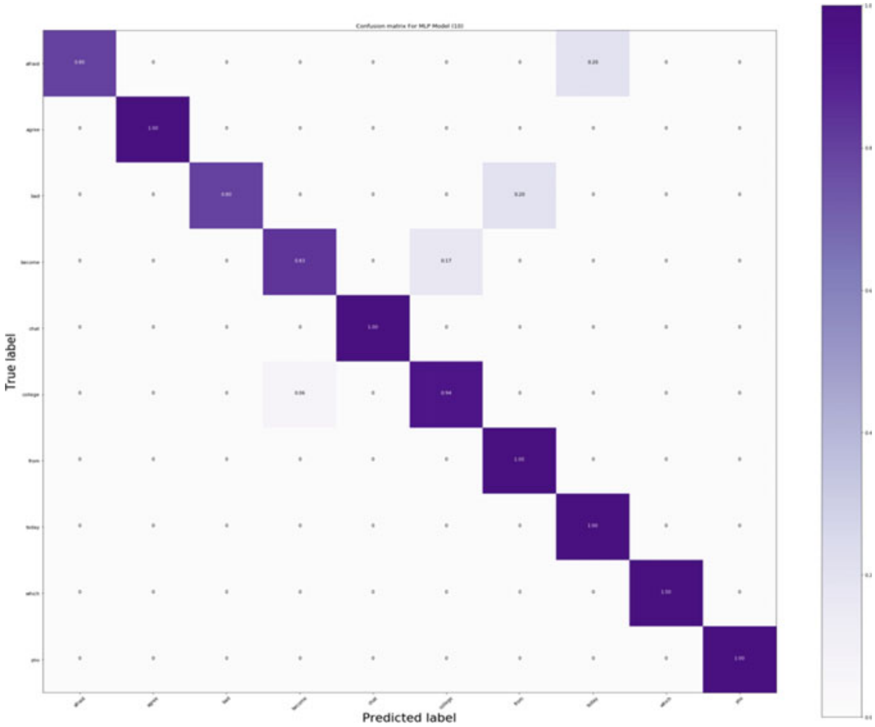

Fig. 5 Confusion matrix for MLP

$$\text{Precision} = \frac{CF_{ii}}{\sum_{1}^{n} CF_{ij}} \tag{13}$$

where CF_{ii} is (i, i)th entry in the confusion matrix, CF_{ij} is (i, j)th entry in the confusion matrix, and n is the total number of classes. Further to calculate the ratio of correctly matched gestures to the number of gestures available for that class, recall function is used as shown in Eq. (14).

$$\text{Recall} = \frac{CF_{ii}}{\sum_{1}^{n} CF_{ji}} \tag{14}$$

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

To seek a balance between recall and precision, the F1-score is also calculated using Eq. (15).

$$f1 = 2 * \frac{\text{Precsion} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

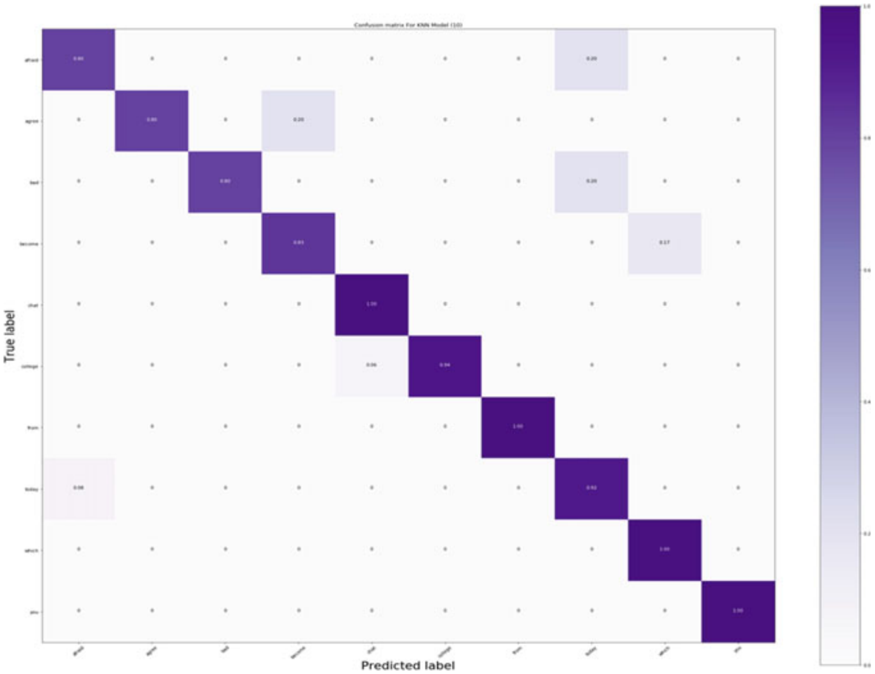

Fig. 6 Confusion matrix for KNN

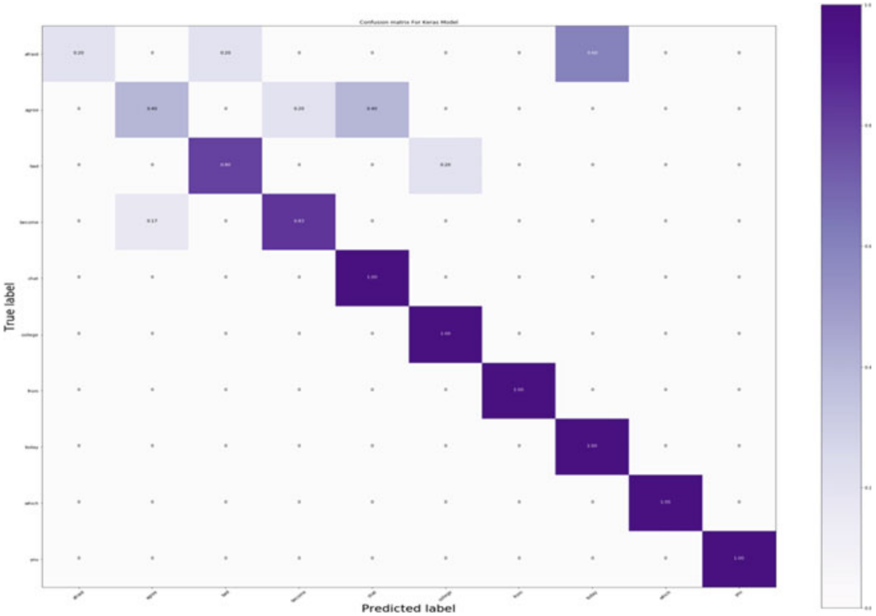

Fig. 7 Confusion matrix for NN

Table 2 Precision, recall, and F1-score obtained by HMASL for SVM

Sign	Precision	Recall	F1-score
Afraid	100	100	100
Agree	100	100	100
Bad	98	100	97
Become	100	97	98
Chat	100	100	100
College	98	100	99
From	100	100	100
Today	100	98	100
Which	100	100	100
You	97	96	98

All values were calculated for a multiclass classifier using the above equations.

Table 2 shows the precision, recall, and F1-score obtained from the SVM classifier, and all the parameters were calculated for other classifiers also.

6 Conclusion

An HMASL has been proposed for a vision-based system for complex background gestures. Hand mark analysis-based features are capable of representing the main points representing the hand, and they do not require any image pre-processing. Therefore, in multiclass, shape classification hand mark analysis has been proved effective and efficient. Hybridization of FAST-SIFT is also done to detect and compute the main features from the hand. These features along with features detected by applying hand mark analysis are stored. The stored dataset values are then used for classification. This work is important in that robust hand gesture recognition system with the complex background is recognized with an accuracy of 96.34%. The dataset in this paper contains only ten ISL words.

7 Future Scope

Further work can be done to increase the number of signs as well as images per sign. In the future, more real-world gestures can be used. HMASL can also be implemented for motion-based Indian signs. In the future, the proposed system work may include dynamic gestures based on some real-world problem using soft computing techniques that can be implemented for real-time usage.

References

1. Tyagi A, Bansal S (2021) Feature extraction technique for vision-based Indian sign language recognition system: a review. In: Computational methods and data engineering, pp 39–53
2. Pisharady PK, Saerbeck M (2015) Recent methods and databases in vision-based hand gesture recognition: a review. Comput Vis Image Underst 141(December):152–165
3. Nie H, Long K, Jun M, Yue D, Liu J (2015) Using an improved sift algorithm and fuzzy closed-loop control strategy for object recognition in cluttered scenes. PLoS ONE 10(2):1–15
4. Gangrade J, Bharti J, Mulye A (2020) Recognition of Indian sign language using ORB with bag of visual words by Kinect sensor. IETE J Res 2020:1–15
5. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
6. Huang J, Zhou W, Li H, Li W (2015) Sign language recognition using 3D convolutional neural networks. In: 2015 IEEE international conference on multimedia expo, pp 1–6
7. Zhang J, Tao C, Wang P (2017) A review of soft computing based on deep learning. In: Proceedings—2016 international conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICIICII 2016), pp 136–144
8. Adaloglou N et al (2020) A comprehensive study on sign language recognition methods. arXiv, pp 1–12
9. Bragg D et al (2019) Sign language recognition, generation, and translation: an interdisciplinary perspective. In: ASSETS 2019—21st international ACM SIGACCESS conference on computers and accessibility, pp 16–31
10. Tyagi A, Bansal S, Kashyap A (2020) Comparative analysis of feature detection and extraction techniques for vision-based ISLR system. 2020 sixth international conference parallel, distributed and grid computing, pp 515–520
11. Huang D, Hu W, Chang S (2009) Vision-based hand gesture recognition using PCA+Gabor filters and SVM. In: 2009 Fifth international conference on intelligent information hiding and multimedia signal processing, pp 1–4
12. Raheja JL, Mishra A, Chaudhary A (2016) Indian sign language recognition using SVM. Pattern Recognit. Image Anal 26(2):434–441
13. Kumar N (2017) Sign language recognition for hearing impaired people based on hands symbols classification. In: Proceeding—IEEE international conference on computing communication and automation (ICCCA 2017), pp 244–249
14. Sharma M, Pal R, Sahoo AK (2014) Indian sign language recognition using neural networks 9(8):1255–1259
15. Bhavsar H (2018) Image based sign language recognition using neuro - fuzzy approach 3(1):487–491
16. Theodorakis S et al (2016) Recognition of alphabets of Indian sign language by Sugeno type fuzzy neural network. Pattern Recognit Lett 30(December 2012):737–742
17. Dudhal A, Mathkar H, Jain A, Kadam O, Shirole M (2018) Hybrid sift feature extraction approach for indian sign language recognition system based on CNN. In: International conference on ISMAC in computational vision and bio-engineering, pp 727–738
18. Sun X, Lv M (2019) Facial expression recognition based on a hybrid model combining deep and shallow features. Cognit Comput 587–597
19. Rao GA, Syamala K, Kishore PVV, Sastry ASCS (2018) Deep convolutional neural networks for sign language recognition. In: 2018 conference on signal processing and communication engineering systems (SPACES), pp 194–197
20. Kishore PVV, Anantha Rao G, Kiran Kumar E, Teja Kiran Kumar M, Anil Kumar D (2018) Selfie sign language recognition with convolutional neural networks. Int J Intell Syst Appl 10(10):63–71
21. Elouariachi I, Benouini R, Zenkouar K, Zarghili A (2020) Robust hand gesture recognition system based on a new set of quaternion Tchebichef moment invariants. Pattern Anal Appl 23(3):1337–1353

22. Alhwarin F, Wang C, Ristić-Durrant D, Gräser A (2008) Improved SIFT-features matching for object recognition. In: Visions of computer science-BCS international academic conference, pp 179–190
23. Abraham A, Krömer P, Snášel V (2015) Afro-European conference for industrial advancement: proceedings of the first international Afro-European conference for industrial advancement AECIA 2014. Adv Intell Syst Comput 334:359–360
24. Patil SB, Sinha GR (2017) Distinctive feature extraction for Indian sign language (ISL) gesture using scale invariant feature transform (SIFT). J Inst Eng Ser B 98(1):19–26
25. Zheng L, Yang Y, Tian Q (2018) SIFT meets CNN: a decade survey of instance retrieval. IEEE Trans Pattern Anal Mach Intell 40(5):1224–1244
26. Bedregal BC, Costa ACR, Dimuro GP (2006) Fuzzy rule-based hand gesture recognition. In: IFIP international conference on artificial intelligence in theory and practice. Springer, Boston, MA
27. Zimmermann C, Brox T (2017) Zimmermann, Brox - 2017—learning to estimate 3D hand pose from single RGB images (2).pdf. ICCV, pp 4903–4911
28. Albanie S et al (2020) BSL-1K: scaling up co-articulated sign language recognition using mouthing cues. arXiv, pp 1–18
29. Kang B, Tripathi S, Nguyen TQ (2016) Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In: Proceedings—3rd IAPR Asian conference on pattern recognition, ACPR 2015, pp 136–140
30. Li D, Opazo CR, Yu X, Li H (2020) Word-level deep sign language recognition from video: a new large-scale dataset and methods comparison. In: Proceedings—2020 IEEE winter conference on applications of computer vision, WACV 2020, pp 1448–1458