2020 IEEE Pune Section International Conference (PuneCon)
Vishwakarma Institute of Technology, Pune India. Dec 16-18, 2020

Indian Sign Language Interpretation and Sentence Formation

Disha Gangadia
Computer Engineering
DJ Sanghvi College of Engineering
Mumbai, India
dishagangadia@gmail.com

Varsha Chamaria
Computer Engineering
DJ Sanghvi College of Engineering
Mumbai, India
varshachamaria1999@gmail.com

Vidhi Doshi
Computer Engineering
DJ Sanghvi College of Engineering
Mumbai, India
doshividhi021@gmail.com

Jigyasa Gandhi
Computer Engineering
DJ Sanghvi College of Engineering
Mumbai, India
jigsgandhi97@gmail.com

2020 IEEE Pune Section International Conference (PuneCon) | 978-1-7281-9600-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/PuneCon50868.2020.9362383

***Abstract*—People with speech and hearing disabilities approximately constitute 1 percentage of the total Indian population. A person who is hearing and speech impaired is not able to compete or work with a normal person in a normal environment because of the lack of a proper communication medium.**

Sign Language is used for communication amongst them. Sign Language is the most natural and expressive way for the hearing and speech impaired. This paper proposes a method that recognizes Sign Language and converts it to normal text and speech for fast and improved communication amongst them and also with others. The focus is on the Indian Sign Language (ISL) specifically as there is no substantial work on ISL rendering the above requirements for these people.

The paper focuses on developing a real-time hands-on system that takes video inputs of gestures in the specified ROI and performs gesture recognition using various feature extraction techniques and Hybrid-CNN model trained using the ISL database created. The correctly identified gesture tokens are sent to a Rule-Based Grammar and for Web Search query to generate various sentences and a Multi-Headed BERT grammar corrector provides grammatically precise and correct sentences as the final output.

***Index Terms*—Indian Sign language (ISL), Hand gesture, Feature extraction, Scale-Invariant Feature Transform (SIFT), Gesture recognition, Dumb and Deaf, Convolutional Neural Network (CNN), Natural Language Processing (NLP).**

I. Introduction

Sign Language is used as a medium for communication by visually impaired, deaf and dumb people which constitute a significant portion of our population. This work aims to bridge this gap of communication by developing a system that takes Indian sign language as input and generates meaningful sentences as output using various machine learning and data mining techniques.

Communication plays an important role in the day-to-day lives of human beings. Thus, we have developed this project for the speech-impaired population. The system will receive a video from the user through the webcam available on the device. After a series of phases involved, for example, Image Preprocessing, Classification and Sentence Formation, the output will be generated which would consist of meaningful sentences, along with the voice.

Thus, to develop a real-time interactive system that can help the hearing and speech impaired to communicate with normal people using Indian Sign Language and to develop a scalable project which can be extended to capture the whole vocabulary of Indian Sign Language through manual and non-manual signs is the main aim of this work.

The social benefits of such a system are enormous.

- A Helping hand for the hearing-impaired and speech-impaired students in their early stages of development and a clearer, compact and precise way of communication.
- Improved teaching-learning process.
- Language flexibility that ensures that the impaired do not have to learn a new language. A uniform system that can be used unanimously.
- Revolutionize the communication process with the help of non-governmental organizations (NGO).
- Improved lifestyle and a great aid to the elderly.

The system has two main components, namely Gesture Recognition and Sentence Generation, which in turn have many sub-components, which includes image preprocessing, removing light variations, reducing noise, motion detection, edge detection, segmentation into frames, extracting gesture labels through Convolutional Neural Network (CNN) based learning, converting the labels to tokens, fitting the tokens inside a grammar, using web results and corpus results that include similarity checks and grammar correction to create top 10 most meaningful and relevant sentences, converting text format to speech and showing the text and speech output results to the user.

II. Literature Survey

A vision-based hand gesture recognition model is proposed by Kanchan Dabre and Surekha Dholay. The proposed architecture in [1] has a Preprocessing phase which undergoes background subtraction, blob analysis, brightness normalization and scaling; and a Classification phase that uses a Haar Cascade Classifier to classify the word and give a textual output.

Reference [2] uses a similar preprocessing phase following which a Condensation Algorithm is used for hand tracking and localization and a Hidden Markov Model (HMM) forward-backward algorithm in conjunction with a Viterbi path is used for recognition gaining 96.25 % accuracy over a batch of 8 gestures.

Xiujuan Chai, Hanjie Wang, Fang Yin, Xilin Chen propose a hierarchical Grassmann Co-variance Matrix (GCM) model in [4] that encodes static sequences as well as continuous sequences (frame by frame) followed by a discriminator kernel Support Vector Machine (SVM) which is used for sign classification. In case of continuous sequences, probability inference technique is used for pointing the labels.

Reference [5] proposed a system that comprises of three stages: Preprocessing stage, Feature Extraction and Classification. Hand gestures are converted to meaningful sentences using some grammar rules, part-of-speech(POS) tagging and a Look-Ahead LR (LALR) parser generates tags on the framed rules.

Reference [3] proposed a grammar error correction (GEC) model which consists of a sequence labelling phase wherein the tokens are given {remain, insert, delete, substitute} labels and a grammar correction phase which uses a pre-trained BERT model to provide candidate outputs for the masked token inputs, thus performing simple grammar correction of unit span.

Eigenvalues and Eigenvectors are considered by Singha J, Das K for the feature extraction stage and finally, Eigen value-weighted Euclidean distance is employed to acknowledge the sign in paper [6]. It deals with bare hands. Skin Filtering is performed to the input video frames for detection of hand gestures, thus allowing the user to interact with the system in a natural way.

Wazalwar, S., Shrawankar, Shrawankar Urmila suggested a method in [8] which the input video of sign language is framed and segmented and the CamShift algorithm and Pseudo two-dimension hidden Markov model (P2DHMM) are used for tracking. Identification of signs is done by Haar Cascade classifier. POS tagging is done with WordNet and using the WordNet dictionary. Finally, an LALR parser is used to generate the sentence.

III. Proposed Architecture

The System Architecture is shown in Figure. 1 and consists of many stages. The preprocessing phase consists of methods that extract important discernible features from the image like

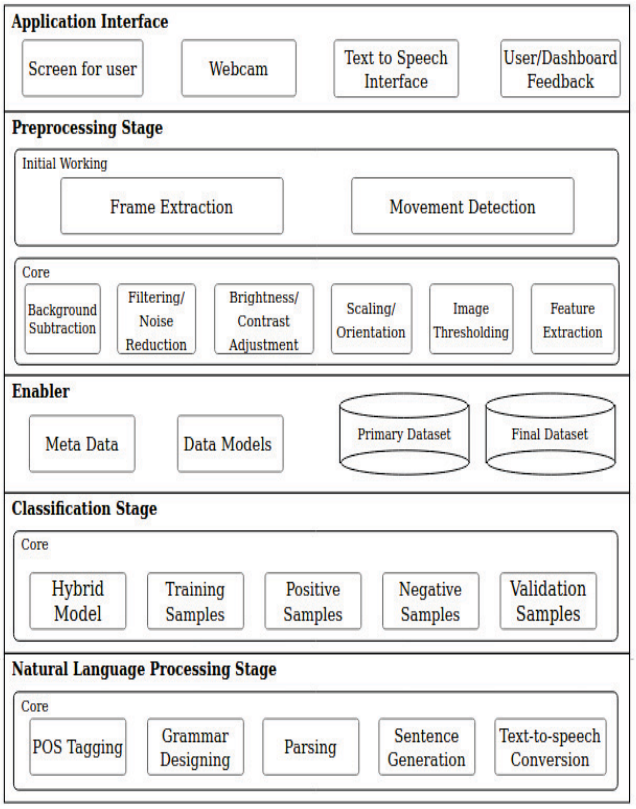


Fig. 1: System architecture.

Grayscaling, illumination normalization, noise removal, edge detection, corner detection, thresholding, etc.

The system contains a data set of preprocessed hand gestures. These are used to evaluate the performance of the proposed method. Data set is a collection of around 10000 hand gesture images. For each gesture class, 100 instances are captured. Since Indian Sign language data set (ISL) is not available, we have created a data set of 100 Indian signs. While creating the data set, image for a particular gesture is stored in 4 different formats: 1) NoFilter 2) Features from accelerated segment test (FAST) 3) Canny Edge 4) Scale-Invariant Feature Transform (SIFT)

The classification uses a CNN-hybrid model which is trained and validated using the database. Also, Data Augmentation is done on the training database images to include various transformations(geometric variance and illumination variance).

The input videos from the user are converted into frames at real-time and passed to the trained model and the gesture with the highest probability is selected and passed on to the Natural Language Processing Stage. The Natural Language Processing phase forms sentences by adding relevant words and correcting grammar and returns a text output which is converted to audio as well for the end-user. NLP phase is also called the speech synthesis phase.

(a) No-filter (b) Adaptive Thresholding (c) SIFT

Fig. 2: Gesture for "house" in different modes

IV. Implementation

A. Data set Generation

The data set consists of 10000 hand gesture instances of 100 unique gestures for the Indian Sign Language. These include the alphabets A-Z (for Proper Nouns) and digits 0-9. The rest 64 gestures are selected after doing a survey about the most frequently used words that need to be addressed in day-to-day life. The data set includes gestures from four different people. Variations for light, orientation, motion, image placement, etc. have been considered and included so as to avoid high bias. While training the model, data augmentation is done so as to create transformations of the original data for the CNN model to learn better. The image resolution is 200 * 200 pixels. This particular size of the region of interest is selected as most of the gestures (single-hand and dual-hand) can be well fit inside the square.

B. Data Preprocessing and Feature Extraction

The images of the data set need to be processed before storing and training. The following thresholding and transformation techniques are applied that help make the classification more accurate.

1) No-filter: The images are kept the way they are. This preserves the RGB values for the images in the order that they are extracted. The dimensions thus remain (200, 200, 3). It can be claimed that the Z dimension is not required for this particular gesture recognition problem because the shape is a primary concern rather than colour and also, the image size is small. The output is shown in Figure. 2(a).

2) Adaptive Thresholding: The No-Filter will fail to give accurate results and hence the image needs to be converted to Grayscale and an Adaptive Threshold is applied for the same. Simple thresholding uses a global value as a threshold parameter and this can create an imbalanced output if the lighting conditions are different in different areas of the image. Adaptive Thresholding algorithm determines the threshold for each pixel based on a small region surrounding it. The output is shown in Figure. 2(b).

$$d(x,y) = \begin{cases} maxvalue & if s(x,y) > T(x,y) \\ 0 & otherwise \end{cases} \quad (1)$$

$$T(x,y) = \sum_{i=1}^{b}\sum_{j=1}^{b} G_{ij} * S_{ij} - c \quad (2)$$

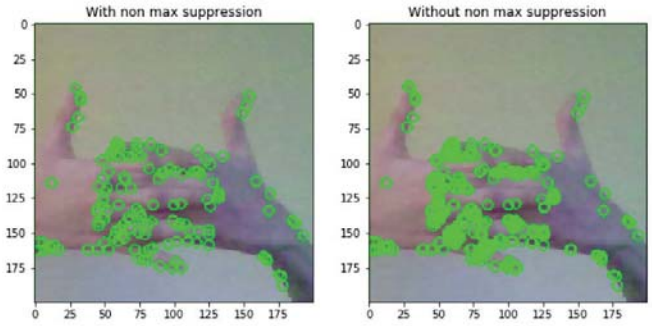


Fig. 3: FAST Feature Extraction

Here,
G_{ij} is a Gaussian weight window of size bxb
c is a constant
s is source pixel
d is destination pixel

3) Feature Extraction: Here, we use FAST key-point extraction, Canny Edge Detection and SIFT feature extraction and use a hybrid in the final model.

a) FAST Feature Extraction: Features from accelerated segment test uses a Bresenham circle of radius 3 which takes 16 pixels into consideration. A candidate pixel p is a corner if it satisfies one of the following conditions.

$$x \in S \; I_x > I_p + t \quad (3)$$

$$x \in S \; I_x < I_p - t \quad (4)$$

Here S is the set of continuous N pixels.

A high-speed test method is applied to improve performance. Situations where N<12, this method cannot be generalized. Also, the sequence in which the adjacent pixels are queried determines the speed. To improve this, a machine learning optimization is used. Detecting multiple interest points in nearby locations is solved by using Non-maximum Suppression. The output is shown in Figure. 3.

b) Canny Edge Detector: Canny Edge Detector is a process used to extract structural information from different vision objects and reduces the amount of data to be processed. It is a popular edge detection algorithm.

It is a multi-stage algorithm and goes through following stages:

1) Noise Reduction: It first smooths the image using a Gaussian Filter. Here, we use a 5*5 Gaussian Filter.

$$H = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 16 & 26 & 41 & 26 & 16 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} * A \quad (5)$$

Fig. 4: Canny Edge Detection

2) Intensity Gradient: The smoothed image is then filtered with a Sobel kernel to get first derivative in both directions in the 2D image, (G_x) and (G_y).

$$Edge_Gradient(G) = \sqrt{G_x^2 + G_y^2} \tag{6}$$

$$Angle(\theta) = \tan^{-1}(\frac{a}{b}) \tag{7}$$

3) Non-maximum Suppression: After intensity gradient calculation, a full scan is performed to remove pixels that do no contribute to an edge. It is checked for each pixel whether it is the local maximum as compared to it's adjacent pixels in the direction of gradient. The result here is a binary image with "thin edges".
4) Hysteresis Thresholding: This stage of the algorithm checks which edges are really edges and which are not.

The output is shown in Figure. 4

c) SIFT Mode: The image obtained from Adaptive Thresholding is now applied to Scale-Invariant Feature Transformation. SIFT is used as it can identify objects and extract local features even in cluttered and partially obstructive images. It is invariant to uniform scaling, orientation, lighting changes. This algorithm finds all the key-points which is the maxima/minima of Difference of Gaussians at different scales σ.

$$D(x, y, \sigma) = L(x, y, \sigma_1) + L(x, y, \sigma_2) \tag{8}$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{9}$$

Here, G is the Gaussian Blur with scale, σ, and * is the convolution sign. This is followed by key-point localization to remove the unnecessary low-contrast key-points and to eliminate edge responses. This requires the principal curvature which is given by the eigenvalues of 2*2 Hessian matrix.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{10}$$

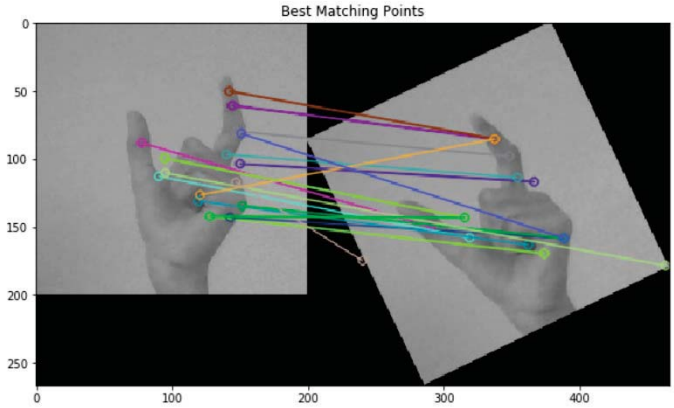


Fig. 5: SIFT algorithm extracting features from the gesture "moon" .

$$R = \frac{(D_{xx} + D_{yy})^2}{D_{xx}D_{yy} - D_{xy}^2} \tag{11}$$

$$if\ R > \frac{(r_{th} + 1)^2}{r_{th}} \qquad Reject\,the\,key - point \tag{12}$$

The descriptor is relative to the orientation calculated and thus results in rotation invariance.

$$\theta = arctan(L(x, y+1) - L(x, y-1), L(x+1, y) - L(x-1, y)) \tag{13}$$

Finally a key-point descriptor is generated. (4*4) neighbourhood pixels are considered with 8 bins for 8 directions. This gives a descriptor of dimension (4*4*8) = 128.

The outputs for SIFT are shown in Figure. 2(c) and Figure. 5. Thus, a FAST, Canny and SIFT feature database is created for all the training images data set. These algorithms alone can be used for object recognition, but for this work, we save the features from their descriptors and make a hybrid system with CNN.

C. Hybrid Model

We calculate CNN for No-filter using various convolution layers, max pooling, and dropouts. Also, the features extracted from these above descriptors are passed through Fully Connected Neural Network Layer consisting of 4096, 2048 and 4096 neurons respectively.

The results from all these layers are merged in the final result and the classification is done using a 100 neuron final layer using the Softmax activation function. The model is shown in Figure 6.

D. Natural Language Processing

The tokens are available from the hybrid model. Now, the tokens are to be used to make a sentence by adding proper and relevant prepositions, articles, conjunctions, tenses, etc.

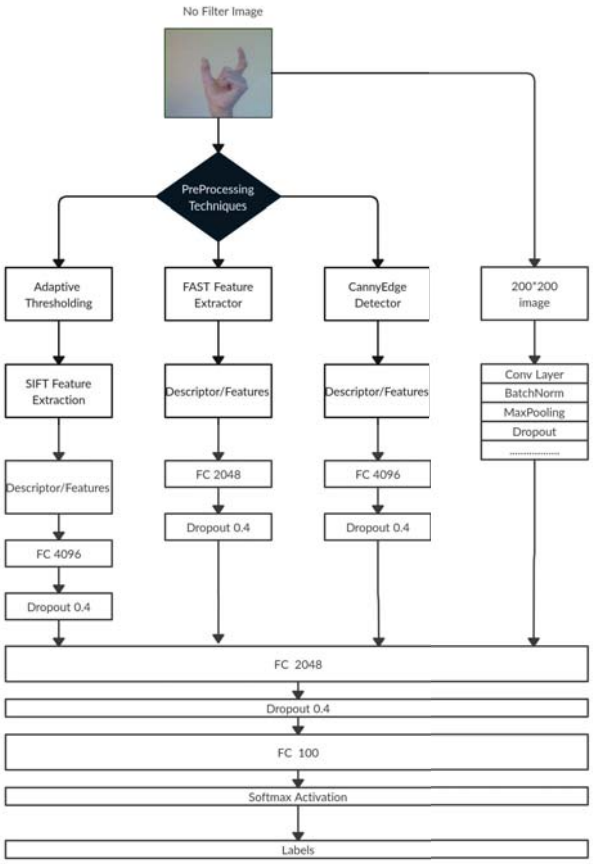


Fig. 6: Hybrid Model

1) Grammar Rules: The words from the database are to be fit inside a detailed context-free-grammar (a brief is shown in the Figure. 7) In addition to the words in the database, which mostly contains Nouns, Verbs, and Adjectives, another set of words for Prepositions, Conjunctions, Determiners is added to the grammar. The working is as follows:

- The 64 tokens are POS-tagged from the brown corpus.
- The tokens are passed to the grammar.
- The tag with the most probability is chosen as final tag.
- The grammar considers the possible sequences by putting all the permutations and combinations.
- All the sentences are returned.

2) Web Results: The grammar may not consist all the words required to make a complete sentence structure. This is where web search results are used. It is observed that web search queries are quite organized in a way that form an entire sentence. Thus, web search results for the given tokens are extracted and stored on run time.

- The results are converted to tokens.
- All the results that have cosine similarity with the array of input tokens smaller than a threshold value are rejected.

$$\cos(\mathbf{A},\mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n}\mathbf{A}_i\mathbf{B}_i}{\sqrt{\sum_{i=1}^{n}(\mathbf{A}_i)^2}\sqrt{\sum_{i=1}^{n}(\mathbf{B}_i)^2}} \tag{14}$$

- The sentences along with the sentences from the grammar are sent to the grammar correction phase.

```
Noun->moon, house, man, abroad,...
Verb->go, is, like, fly, see, eat, walk,...
Adjective->good, very, new, first, little,...
Pronoun->I, me, you, him, her, it,...
Determiner->the, a, an, this, that,...
Preposition->from, to, on, near,...
Conjunction->and, or, but,...
S → NP VP | VP | Aux NP VP | Wh NP VP | S CONJ S
NP → Pronoun | Proper-Noun | Det Nominal | NP CONJ NP
Nominal → Noun Nominal | Noun | ADJ Nom
VP → Verb | Verb NP | Verb NP PP | Verb ADJ | Verb PP | VP CONJ VP
PP → Preposition NP
```

Fig. 7: Grammar Rules

3) Grammar Correction: The grammar is checked using Bidirectional Encoder Representations from Transformers (BERT) which is used on the data set of Lang-8 Corpus of Learner English . A multi-headed language model is generated that uses BERT as encoder and decoder Transformer for grammar correction (excluding spelling correction) with Replace and Range Heads options. The Transformer is bidirectional and helps in understanding the context of the sentence. We have used a MASK of 15%. The multi-headed BERT gives a suggested correction. The correction is stored in final results along with probability which is given as:

$$P(X) = \prod_{i=1}^{n} x_i * cos(InputX, OutputX) \tag{15}$$

Here, x_i is the probability of gestures from the hybrid model and cos is the Cosine Similarity between the input and suggested correction of BERT.
Final steps:

- The top 10 most probable sentences are stored in final results and also stored in database for future referencing.
- The sentences are POS tagged and the unseen words are added to the Grammar.
- Give a Text-To-Speech audio output along with text.

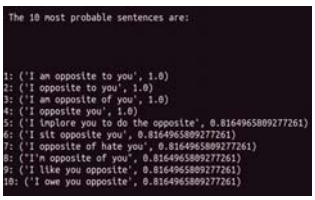
V. Results

The results for the hybrid model are as shown in the Table. I. A real time working of the entire work is shown in Figure 8. And Table. II shows the probability of sentences that are actually expected.

(a) I

(b) Opposite

(c) You



(d) Probable sentences from tokens: I, Opposite, You

Fig. 8: Example Input and Output

TABLE I: Comparison of models for Image Processing part

Model-Type	Accuracy	Precision	Recall	Fscore
SIFT	0.841	0.90	0.85	0.87
CNN on No-Filter	0.852	0.89	0.88	0.88
CNN-SIFT hybrid	0.934	0.91	0.89	0.90
Canny-SIFT hybrid	0.911	0.84	0.87	0.85
Hybrid-model	0.942	0.92	0.88	0.88

TABLE II: Examples showing the probability for the most relevant sentence

Tokens	Most Relevant Sentence	Probability
IorMe, Opposite, You	I am opposite to you.	1.0
IorMe, Abroad	I am going abroad.	0.90
IorMe, Moon	I see the moon.	0.81
IorMe, House	I live in house.	0.82
You, Man	You are a man.	1.0

VI. Conclusion

A system is developed that can recognize a set of Indian Sign Language gestures and convert them into meaningful text/speech using various Image Processing and Machine Learning/Deep Learning techniques. It makes a foundation for a scalable project that can be extended to capture the whole vocabulary of Indian Sign Language through manual and non-manual signs. Using a hybrid model gives the benefits of all the feature extraction techniques and gives substantial accuracy along with decreasing the computational time required. The sentences formed from the language model ensure the correctness and preciseness when compared to expected results.

VII. Future Scope

The system can be extended to include the knowledge of phonological and morphological information. Dependency Parsing and Text Similarity can be used if the input is a paragraph. The application can be extended to take speech as input and generate its corresponding gesture. We can also add more instances into the data set such that it covers all the spheres of communication and provides variations. A personal assistant who actively interacts with people with disabilities can be developed by extending this system. From the keywords extracted, sentences can be generated in various languages like Hindi, Marathi, Gujarati, etc. with proper sentence formation schemes to ensure grammatically correct sentences.

References

[1] Kanchan Dabre, Surekha Dholay, "Machine Learning Model for Sign Language Interpretation using Webcam Images", International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA), IEEE, 2014.

[2] Tanatcha Chaikhumpha, Phattanaphong Chomphuwiset, "Real — time two hand gesture recognition with condensation and hidden Markov models", International Workshop on Advanced Image Technology (IWAIT), 2018.

[3] Yiyuan Li, Antonios Anastasopoulos, Alan W Black, "Towards Minimal Supervision BERT-based Grammar Error Correction", arXiv:2001.03521 Cs, Jan 2020.

[4] Xiujuan Chai, Hanjie Wang, Fang Yin, Xilin Chen, "Communication tool for the hard of hearings: A large vocabulary sign language recognition system", International Conference on Affective Computing and Intelligent Interaction (ACII), IEEE, 2015.

[5] Sumeet R. Agarwal, Sagarkumar B. Agrawal, Akhtar M. Latif, "Sentence Formation in NLP Engine on the Basis of Indian Sign Language using Hand Gestures", International Journal of Computer Applications (0975–8887) Volume 116 – No. 17, April 2015.

[6] Singha J, Das K, "Recognition of Indian sign language in live video", International Journal of Computer Applications 70(19):17-22, May 2013.

[7] Mundher Al-Shabi, Wooi Ping Cheah, Tee Connie, "Facial Expression Recognition Using a Hybrid CNN–SIFT Aggregator", arXiv:1608.02833 Cs, 2016.

[8] Sampada Wazalwar, Urmila Shrawankar, "Interpretation of sign language into English using NLP techniques", Journal of Information and Optimization Sciences 38(6):895-910, August 2017.

[9] Sunita Nayak, Sudeep Sarkar, Barbara Leoding, "Automated Extraction of Signs from Continuous Sign Language Sentences using Iterated Conditional Modes", Conference on Computer Vision and Pattern Recognition, IEEE, 2009.

[10] Aradhana Kar, Pinaki Sankar Chatterjee, "An Approach for Minimizing the Time Taken by Video Processing for Translating Sign Language to Simple Sentence in English", International Conference on Computational Intelligence & Networks, IEEE, 2015.