An Intelligent Sign Communication Machine for People Impaired with Hearing and Speaking Abilities

Ashish Sharma[1](✉), Tapas Badal[2], Akshat Gupta[1], Arpit Gupta[1], and Aman Anand[3]

[1] Computer Science Engineering Department,
Indian Institute of Information Technology, Kota, Kota, India
ashishsharma.fitt@gmail.com
[2] Department of CSE, Bennett University, Noida, India
[3] Electronics and Communication Department,
Indian Institute of Information Technology, Kota, Kota, India

Abstract. People who are impaired with speaking and hearing abilities use sign language for communication between them, but it is a tough task for them to communicate with the outside world. Through this paper, we are proposing a system to convert Indian Sigh Language (ISL), American Sign Language (ISL) and British Sign Language (BSL) hand gestures to a textual format of the respective language as well as convert text in to their preferable Sign language. In this paper, we are capturing ISL, ASL, BSL gestures through a web camera. The streaming video of hand gestures is then sliced to distinct images to match the finger orientation to the corresponding alphabets. Finger orientations as features of the hand gestures in terms of angles made by fingers, numbers of fingers completely open, semi-open, fully closed, finger axis verticals or horizontal and recognition of each finger are prepossessed and required for gesture recognition. Implementation is done for alphabets uses single hand and results are explained. After prepossessing the hand part of the sliced frame in the form of masked image is projected to the extraction of features from the image frame. To classify different gestures we used SVM (Support Vector Machine), CNN (Convolutional Neural Network) for further testing the probable gesture and recording the accuracies of each algorithm. Implementation is done over our own regular ISL, BSL, ASL data-set made by us only, using the web camera of our laptops. Our Experimental results depict that our proposed work and methodology can work on different backgrounds like a background consist of different objects or may have some sort of color background etc. For text to sign conversion we create a video which tells respective text into sign language.

Keywords: Indian Sign Language Recognition (ISL) · Text to sign conversion · Hand gesture recognition · Hand segmentation · Support Vector Machine (SVM) · Convolutional Neural Network (CNN)

D. Garg et al. (Eds.): IACC 2020, CCIS 1367, pp. 75–86, 2021.
https://doi.org/10.1007/978-981-16-0401-0_6

1 Introduction

All non-vocal communication requires a particular action for a particular context like the movement of the face, flipping of hands or folding fingers or actions by any other body part is a form of gesture. Gesture recognition is a method to make a machine or a computer get to recognize these actions. Algorithms used by these methods act as a mediator between human and machine. This enables a computer to interact with humans naturally by their own without any physical contact, actually just by using cameras as their eyes. Deaf and dumb people use hand gestures in their community for communication under the name sign language. This leads to a kind of isolation between their community and ours due to language differentiation as a normal person do not want to learn such language. So if we can program our computers in such a way that they take input in sign language and process them to convert in their respective language or maybe other languages also either in speech or in the textual format then they can act as a noble inter mediator and can remove the language barrier, the difference between communities can be minimized and the most important, knowing a language will meet to its worthy result in this high-tech world as sign language can interact to English and vice versa. All these discussions lead to a need for a system which can act as a translator and converts sign language to the desired language in the desired format, so people with a different language background can have a possible conversation with the people who know only sign language due to some disabilities but literate.

Sign Language shares grammar syntax like the use of pauses, full stop, and simultaneity, hand postures, hand placement, orientation, motion of the head, face gestures with different sign languages. As a country like India is completely diverse in terms of culture, religion, beliefs, and majorly in languages, so there is not a standard sign language is adopted in India. Various social groups of Indian Sign Language with their native and historical variation are there in India in various parts of the country. But still, language skeleton is similar for the maximum gestures. Work relating to the system of contrast relationships among the speech sounds that constitute the fundamental components of ISL started in the 1970s. With the help from Woodward, National Science Foundation USA Vasishta and Wilson visit of India and collection of signs from different points in the country for language analytic.

The organization of the paper is as follows: 'Sect. 2' the methods related to different technologies available in the language. 'Section 3' explains the given Sign language recognition system the method which uses algorithms for skin cropping and SVM(Support Vector Machine). 'Section 4' concerns on the implementation results and 'Sect. 5' is description and conclusion.

2 Literature Survey

This paper [14] proposes HSI color model for segmentation of images instead of RGB model. HSI model works better for skin color recognition. The optimal H and S values for hand as specified in [14] is $H < 25$ or $H > 230$ and $S < 25$ or $S > 230$. After this they use euclidean distance formula to evaluate the distance between centroid of palm and fingers. Distance transform method is used to identify the centroid of the hand. The pixel with the maximum intensity becomes the centroid. To extract each finger tip they select farthest point from centroid. Every finger is identified by predefined sign gestures. To recognize semi opened finger they divide every finger into 3 parts. and angle between the centroid and the major axis of finger is calculated (Figs. 1, 2 and 3).

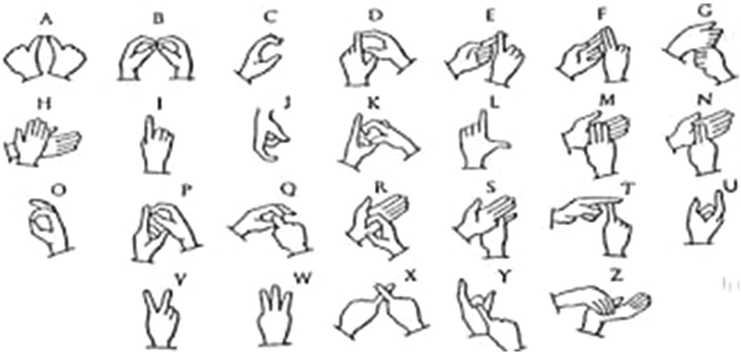


Fig. 1. Indian sign language alphabets [6]

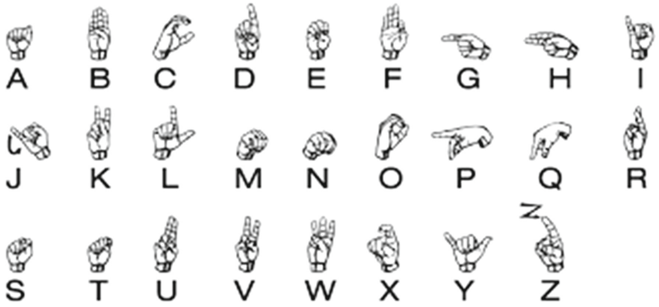


Fig. 2. American sign language alphabets [13]

In this paper [4] they used YCbCr color space, where Y channel represents brightness and (Cb, Cr) channels refer to chrominance. They use Cb, Cr channels to represent color and avoid Y since it is related to brightness only. There are some small regions near skin but not in skin so they use morphological operation. After that they select skin region and extract features to recognize hand gesture. They use three features velocity, orientation and location. They use orientation

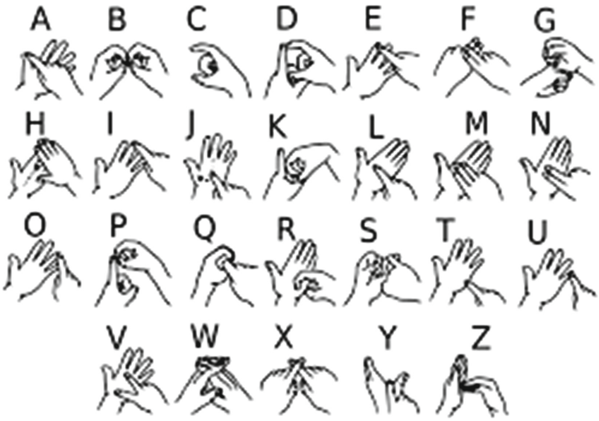


Fig. 3. British sign language alphabets [9]

feature as a main feature for their system. Then they classify features using Baum-Welch algorithm (BW). The gesture of hand motion is recognized using Left-Right Banded model with 9 stage.

In this paper [10] they used YCbCr color space. This color model is implemented by defining skin range in RGB model then convert these values into YCbCr model using conversion formula. They used support vector machine (SVM) algorithm. This algorithm use hyper plane to differentiate between two classes. Hyper plane is defined by the Support vectors which are nothing but the subset of training data. This algorithm also used to solved multi-class problem by demising it into two-class problem.

They [7] create data-set using an external camera having some specifications like 29 fps, 18 MP ans Canon EOS with 18–55mm lens. They eliminate background and extract hand region from left-out upper body part. They used RGB configuration of frame having dimensions of 640 * 480 then they extract key frames from video. They use orientation histogram to extract key frames. They used different distance metrics (Chess Board Distance, Euclidean distance etc) to recognise a gesture. After successful recognition of gesture they classified them for text formation.

They [8] use Fully convolution network algorithm. In particular they used 8 layers FCN model which achieves good performance and used for solving dense prediction problems. The output segmentation of this network is robust under various face conditions because it consider a large range of context information. After that they use CRF algorithm for image matting.

They [9] used Convolution neural network to generate their trained model. In this network they used 4 layers, in first stage they used five rectified linear units (ReLu), in second stage two stochastic pooling layers then one dense and one SoftMax output layer. They took frames of 640 * 480 dimensions then resize these frames into 128 * 128 * 3. They took 200 frames by 5 different people and at 5 different viewing angles. Their data-set size is of 5000 frames.

In this paper [13] they used CNN to recognize static sign gestures. They use American Sign Language (ASL) data-set to train their model which is provided by Pugeault and Bowden in 2011. There are around 60,000 RGB images they used for training and testing. They perform some operations on this data-set because not every is image has same depth according to their dimensions. They used V3 model to perform color features then for better accuracy they combined it with depth features. They use 50 epoch and 100 batch size to train their model using CNN.

Suharjito et al. [1] reviewed the different methods and techniques that researchers are using to develop better Sign Language.

Kakoty et al. [6] address the sign language number and alphabets recognition using hand kinematics with hand glove. They achieved the 97 % recognition rate of these alphabets and numbers.

In this article [11] the proposed system is translating the English text into Indian Sign Language (ISL). Authors have used human-computer interaction to implement it. The implemented system consists of the ISL parser, the Hamburg Notation System, the Signing Gesture Mark-up Language and generates the animation for ISL grammar.

Paras et al. [12] used the wordnet concept to extend and expansion of the dictionary and further construct the system to develop the Indian sign language system for dump and deaf peoples.

Matt et al. [5] address the video-based feedback information to students to learn the American Sign Language (ASL).

In this artical [3] authors address the deep learning based Gesture Images implementation for sign language. The validation accuracy obtained for this implementation using the different layers of deep learning is more than 90%.

3 Proposed Work

Flow Chart. The given flow chart explains the work flow of our project includes segmentation of video and then masking of image followed by canny edge detection which is used surf library and then features of images projected to clustering and comparisons between clusters of training and testing data is further done by svm library as described below flowchart.

Segmentation. As to recognize and classify each and every character of the input video it is required to apply image processing on it. For that purpose, the input video is converted into frames so that different image processing algorithm can be applied to them.

So for that in this step Input video is converted into frames, this step converts video into 30 frames per second (fps) which is default for the webcam used for making video but as we required less frames per second to recognize the each and every character in the video, so by giving delay to the function which is converting video to frame we are able to achieve the required output which is to get 5 frame per second of a input video.

Skin Masking. The reasoning behind a process such that to remove the extra noise in the segmented frame, after the masking there should be only the Region of Interest (ROI), which contains only useful information in the image. This is achieved via Skin Masking defining the threshold on RGB schema and then converting RGB colour space to grey scale image (Fig. 4).

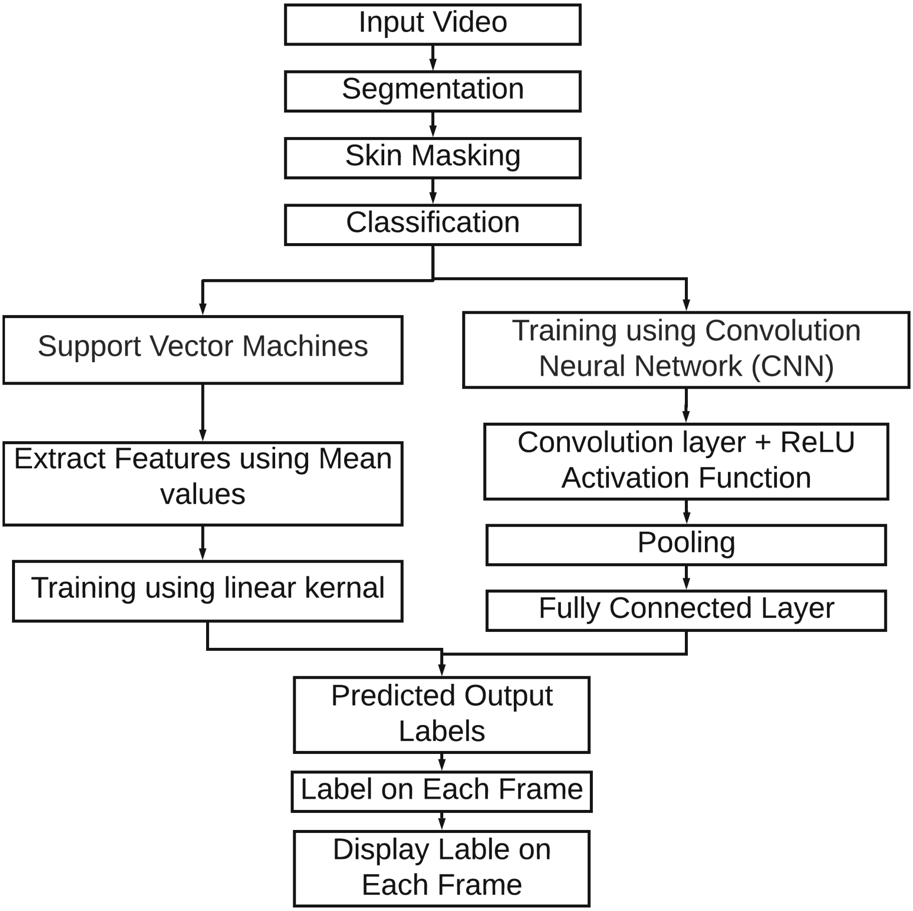

Input Video
Segmentation
Skin Masking
Classification
Support Vector Machines
Training using Convolution Neural Network (CNN)
Extract Features using Mean values
Convolution layer + ReLU Activation Function
Pooling
Training using linear kernal
Fully Connected Layer
Predicted Output Labels
Label on Each Frame
Display Lable on Each Frame

Fig. 4. Design flow for the project

So to achieve skin masking various image processing functions has been used. Firstly, the frame is convert into a gray schema. This output gray image will help us to convert it to HSV schema which will help us to detect the skin colour which is the main objective of ours so that we can identify the hand region. After identifying the hand region we have removed the noise from the image using blur function.

Classification. Sign classification is perform by using different techniques such as: a) Support Vector Machine (SVM) b) Convolution Neural Network (CNN).

Mainly we have use SVM for testing the accuracy of our system by dividing the 70% data as testing data and remaining as testing data where as we have used CNN mainly for live recognition of the sign of the alphabets.

Classification using SVM involves various steps as shown in figure. We convert saved masked image into 100 * 100 pixels then calculate its mean and variance oven 10 pixels.

$$\text{Mean } \mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\text{Variance} : \sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}$$

We use these parameters as classification features for SVM. We use linear regression as base kernel of our SVM algorithm. After creating model of these specifications (gamma = default and c = 1) we train our model with our data-set. To check accuracy of our model we use 30% data-set for testing purpose.

We use Convolution Neural Network as our second algorithm. To create CNN model we use Sequential Model in which three layers have been used:

1) Rectified linear unit (ReLU) Layer
2) pooling Layer
3) Fully connected layer for this we used Sigmoid activation function

After successful creation of this model we use our 70% data-set to train this model and remaining to test and validate our model.

We used CNN model for real time gesture recognition instead of SVM model because of its better performance (in terms of time).

Output Labels and Sentence Formation. Proposed System predict output labels using CNN model and show that label in real time on each frame. To form sentence system need to store that character into their words and provide space in between sentence. In this stage proposed system generate character from user input and arranged it into a word from dictionary, user can choose to provide space between words by pressing a key manually.

Text to Video. To convert text into sign video generation function is applied. We use sign of alphabets to convert text into sign language.

4 Experiment Setup

Data-Set. As we have searched on the internet and we found no resources from where we can get Indian Sign Language dataset. So after a long effort in searching and finding dataset from different resources, then we only made our own ISL dataset as in our lighting conditions and in other factors like own environmental setup. There we have 26 × 15O = 3900 static training images and 26 × 30 = 60 images which will use for testing. The actual resolution of the images is 640 × 480, which will be cropped and normalized into 120 × 120. The samples from the video are 320 × 260 in size and they are taken in a various lighting environment. Same process we used on two other sign languages American sign language and British sign language.

We have made one interface where we have given choice to the user in which language he/she wants to do operation i.e whether in ISL, BSL or ASL. After that another two other choices will come in which user have to tell whether he/she wants to do sig-text conversion or text-sign conversion. It makes our system user friendly and a normal people can easily use it for communication.

Algorithms

- Support Vector Machine Algorithm The support vector machine (SVM) is an algorithm which is used for two-class problems (Binary classification problems) in which the concerned data can be separated by a different plane like linear plane, parabolic plane etc depending upon the number of features of the sets. Hyper plane basically refers to a virtual plane that can be drawn in the 3D properties plot of the given data in order to separate them on the basis of some features. Different classes are separated using it which uses the training data to do the supervised learning of the system. Every feature in the training data set is send with the target value to do the learning of the system according to it. Support vector machine is mainly used to predict the targeted value of the given testing data set features according to the plane which is drawn by the algorithm for the distinguish of the different features in the training data set [2].

  Both Classification or regression function can be used for the mapping of function. When there are non-linear functions for the distinction non-linear plane is used according the features of it to convert it into n-d space distinction. Fig represents the plane which is drawn to separate the n-features in n-d plane. Then the creation of Maximum-margin hyper planes can be done. Proposed model works over only a subset of the training data set as per the class boundaries. Similarly, This model can also be produced by SVR (support vector regression).

  SVM uses different values of gamma and c to draw the hyper plane between the two clusters for distinct of them. Larger the value of gamma more it considered the points far from the hyper plane which will give the better result and c will tell how smoothly will the plane gonna be larger the value of c greater distinguish it will take in consideration.
- Convolution Neural Network The combination of neurons with biases and weights is known as Convolution Neural Network. The neurons which are there in the layer gets the input from the its parents layers. Computation of product between the weights and input is done, and posses an option to follow

the processes output with a non-linearity. Implementation of the properties in the CNN is done with the assumption that of taking all the inputs as images. The CNN architecture has been classified in to different layers, it contains many convolution layer along with the activation function layer called ReLU layer and Pooling step. Standard architecture of CNN is in the last layer. is plenary connected is a standard architecture of CNN.

The CNN architecture has been classified in to different layers: **(1) Convolution Layer:** We extract features from our frame in this convolution layer, Some parts of image is link to the upcoming layer convolution layer. Computation of the dot product is19 done in the receptive area and a kernel [3 * 3 filter] on all the image as shown in the image. The output of the dot product gives as the integer value which is known as features as shown in fig. After that feature extraction is done using filter or kernel of small matrix. **(2) Padding Process:** Padding means to do the summation of all the features which we got in the feature map and finally putting the summation in the middle of the 3 × 3 matrix. This is done to get the equal dimension of output which we have used in the input volume.

(3) Rectifier Activation Function (ReLU):
After the implementation of convolution layer on the image matrix, we will use ReLU layer to get the non-linearity to the system by applying ReLU (non-linear activation function) to the feature matrix. There are many activation function are present but here we are using ReLU as it does not which makes the network hard to train.

(4) Pooling Layer:
Controlling of over fitting and decreasing the dimension of the image is done in Pooling layer. It can be done in three ways first one is max, second one is average and third one is mean pooling, here we are using the max pooling, it is used to take maximum value from the input which we are convoling with features.

(5) Fully Connected Layer:
This one of the important layer of convolution layer as it gives the classified images according to the training data set. We have used the different sign images for the training set as discussed above.

(6) Epochs:
During the whole data set is going backward and forward propagation through networks is called epochs.

(7) Training Accuracy:
Training accuracy given by the model, when we are applying training on training data sets.

(8) Validation Accuracy:
After the successful training of the model then it is evaluated with help of test data sets then accuracy of model is predicted.

5 Experimental Result

We have performed the training on three different sign languages each having 45,500 training images and performed the testing on 20,800 images.

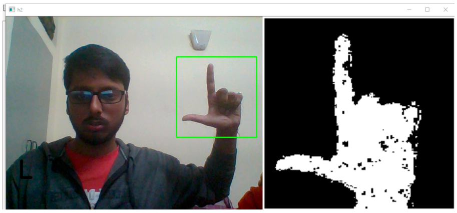
Accuracy. We have compare the performance of different languages using both the classification method below is the comparison table of that in terms of accuracy of each sign language (Tables 1 and 2).

Table 1. Performance of different languages using both the classification methods.

Language	CNN-accuracy	SVM-accuracy
Indian Sign Language (ISL)	0.9988209	0.9876898
American Sign Language (ASL)	0.98949781	0.9796472
British Sign Language (BSL)	0.98645851	0.97289481

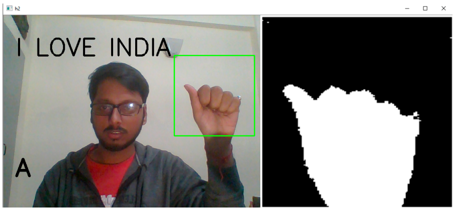
Table 2. Accuracy table for different algorithms on ISL.

Algorithm	Accuracy
K-nearest neighbour	0.6628820960698
Logistic regression	0.7554585152838
Naive bayes	0.6283842794759



L

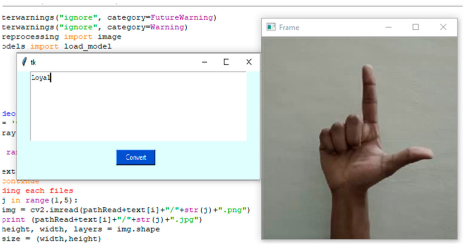
Fig. 5. Output of 'L' sign recognized by the system



I LOVE INDIA

A

Fig. 6. Output showing sentence formation using the system



```
terwarnings("ignore", category=FutureWarning)
terwarnings("ignore", category=Warning)
reprocessing import image
odels import load_model

deo
= '
ray

 ra

ext
continue
ding each files
j in range(1,5):
img = cv2.imread(pathRead+text[i]+"/"+str(j)+".png")
print (pathRead+text[i]+"/"+str(j)+".jpg")
height, width, layers = img.shape
size = (width,height)
```

Loyal

Fig. 7. Output showing sentence formation in sign language

6 Conclusion

We have successfully perform different sign languages conversion into their respective alphabets as well as form sentences using these alphabets. We has used our webcam to capture gesture instead of some specified powerful camera like RGB-D. Our systems also work on text to sign conversion. So it is works like communication medium between who can communicate only in sign language and those who don't understand it (Fig. 5, 6 and 7).

- We have worked on the stationary hand gesture but sign language can have moving hands also. So, in future it can be done for both moving hands also.
- The major problem with the project is it is mainly depend on the lighting condition so in future the effect of lighting can be overcome.

References

1. Abraham, A., Rohini, V.: Real time conversion of sign language to speech and prediction of gestures using artificial neural network. Proc. Comput. Sci. **143**, 587–594 (2018). https://doi.org/10.1016/j.procs.2018.10.435. http://www.sciencedirect.com/science/article/pii/S1877050918321331. 8th International Conference on Advances in Computing & Communications (ICACC-2018)
2. Dai, H.: Research on svm improved algorithm for large data classification. In: 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), pp. 181–185, March 2018. https://doi.org/10.1109/ICBDA.2018.8367673
3. Das, A., Gawde, S., Suratwala, K., Kalbande, D.: Sign language recognition using deep learning on custom processed static gesture images. In: 2018 International Conference on Smart City and Emerging Technology (ICSCET), pp. 1–6 (2018)
4. Elmezain, M., Al-Hamadi, A., Michaelis, B.: Real-time capable system for hand gesture recognition using hidden Markov models in stereo color image sequence. J. WSCG **16** (2008)
5. Huenerfauth, M., Gale, E., Penly, B., Pillutla, S., Willard, M., Hariharan, D.: Evaluation of language feedback methods for student videos of American sign language. ACM Trans. Access. Comput. (TACCESS) **10**(1), 1–30 (2017). https://doi.org/10.1145/3046788

6. Kakoty, N.M., Sharma, M.D.: Recognition of sign language alphabets and numbers based on hand kinematics using a data glove. Proc. Comput. Sci. **133**, 55–62 (2018). https://doi.org/10.1016/j.procs.2018.07.008. http://www.sciencedirect.com/science/article/pii/S1877050918309529. International Conference on Robotics and Smart Manufacturing (RoSMa2018)
7. Liu, L.: Research on logistic regression algorithm of breast cancer diagnose data by machine learning. In: 2018 International Conference on Robots Intelligent System (ICRIS), pp. 157–160, May 2018. https://doi.org/10.1109/ICRIS.2018.00049
8. Qin, S., Kim, S., Manduchi, R.: Automatic skin and hair masking using fully convolutional networks. In: 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 103–108, July 2017. https://doi.org/10.1109/ICME.2017.8019339
9. Rao, G.A., Syamala, K., Kishore, P.V.V., Sastry, A.S.C.S.: Deep convolutional neural networks for sign language recognition. In: 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), pp. 194–197, January 2018. https://doi.org/10.1109/SPACES.2018.8316344
10. Reshna, S., Jayaraju, M.: Spotting and recognition of hand gesture for Indian sign language recognition system with skin segmentation and SVM. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 386–390, March 2017. https://doi.org/10.1109/WiSPNET.2017.8299784
11. Sugandhi, Kumar, P., Kaur, S.: Sign language generation system based on Indian sign language grammar. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **19**(4), 1-26 (2020). https://doi.org/10.1145/3384202
12. Vij, P., Kumar, P.: Mapping Hindi text to Indian sign language with extension using WordNet. In: Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2979779.2979817. https://doi.org/10.1145/2979779.2979817
13. Xie, B., He, X., Li, Y.: RGB-D static gesture recognition based on convolutional neural network. J. Eng. **2018**(16), 1515–1520 (2018). https://doi.org/10.1049/joe.2018.8327
14. Zhou, Q., Zhao, Z.: Substation equipment image recognition based on sift feature matching. In: 2012 5th International Congress on Image and Signal Processing, pp. 1344–1347, October 2012. https://doi.org/10.1109/CISP.2012.6469854