# COMPUTERIZED AID SYSTEM FOR DEAF AND DUMB USING AI

**Akershi[1], Dr. Ashish Baiswar[2]**

*[1]UG student of Department of Information Technology, Shri Ramswaroop Memorial College of Engineering and Management Lucknow, Uttar Pradesh, India*

*[2]Assistant Professor, Department of Information Technology, Shri Ramswaroop Memorial College of Engineering and Management Lucknow, Uttar Pradesh, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract -** Hand gesture is the conspicuous mechanism of correspondence for speech and hearing-disabled community to communicate their sentiments to typical individuals at public spots and the ordinary community thinks that it's hard to decipher the imparted content. Convolution Neural network is utilized for the real-time preparing of hand sign gestures as feature extraction and order is naturally handled by CNN. Feature map is acquired by convolving channel over the image. The ReLU activation function work eliminates the nonlinearity from the picture and gives the extricated feature map. The dimensionality of the feature map is diminished by max pooling layer and named. The 88% of self-gathered dataset is utilized for training also, 12% is utilized for testing. During testing stage, the trained CNN straightforwardly group the hand gesture and show the comment relating to the hand gesture. The proposed framework engineering accomplishes most extreme approval exactness of 98.96% also, 100% by fluctuating number of channels

***Key Words***: Dataset, Convolutional Neural Network, Hand Gestures, Feature extraction, Classification

## 1. INTRODUCTION

Sign Languages are the visual language that can be used to convey a meaning in visual-manual modality to impaired persons. But communication between normal people and deaf-dumb people can be a tough task. To reduce this, the Hand Gesture recognition help impaired people to communicate with normal community. These gestures are divided into Static and Dynamic gestures. Normal people find understanding the Hand Sign Gestures tough. Technology for better communication between the normal and the deaf-mute individual is restricted in real time.

Fortunately, hand Gestures plays a vital role in Human Computer Interaction (HCI) and in many other areas like sign language translation, robot remote control or musical creation etc. Research is ongoing for vision based Hand Gesture recognition since 2002 for several Hand linguistic communications like Indian Sign Language (ISL), American linguistic communication (ASL), Bangla linguistic communication (BSL) and many more. Technologies like Internet of Things, Artificial Intelligence, Machine Learning and Deep Learning are used for Vision and Sensor-based hand gesture recognition. The prediction accuracy is based on the algorithms used and changes for each technology. Current technology used by the researchers is Deep Learning algorithm which produces accuracy up to 99.9%. In this paper, we apply Convolution Neural Network (CNN) which is a Deep learning algorithm for Hand Gesture classification and prediction with maximum accuracy.

## 2. WORKFLOW

The dataset is formed for training and testing to get edges within the image 3x3 filter is convolved over the image and therefore the receptive field gives the feature map for the region because the filter slides over, the feature map continuous to grow. When the amount of filters used is more, than the combination of feature map is taken. To removes the nonlinearity from the image Rectifier linear unit (ReLU) activation function is employed which provides the extracted feature map. To remove over fitting and decrease training time the dimensionality of the feature map is reduced by pooling layer using max pooling. The dimensionality diminished feature map is then labeled.

The labeling provides help in classification of gestures. During testing phase, the trained CNN simply classifies the hand gesture and shows output the annotation like the hand gesture. 45 thousand sign images categorized to 27 classes with 1,750 images for every class, covering 27 English alphabets. The input images are resized to 128x128 pixels during preprocessing. CNN extracts and classify the extracted features. CNN classifier is trained with 25 epochs each and achieved classification accuracy of 98.96 percentage with colored and grey scale images.

## 3. PROPOSED SYSTEM

The proposed vision based system avoids the necessity for external hardware and code constraint. The Deep Learning algorithm CNN is employed for the conversion of Hand Sign Gesture to Text and further to speech. The system is assessed into three parts as depicted in Fig 1. First, Hand gesture images are sampled using webcam. Secondly, of the collected input images majority are taken to train the Convolutional Neural Network to extract feature map and classify the image to specific category. Finally the remaining gestures images are used to test the trained CNN model to evaluate the system performance based on the corresponding displayed label.

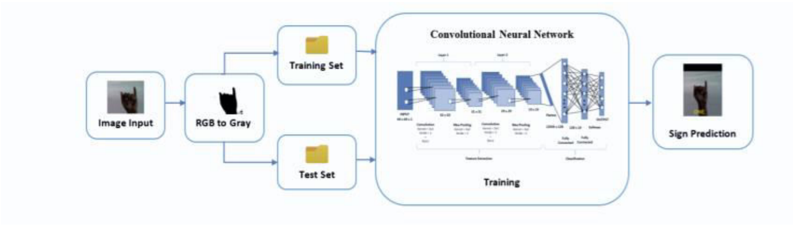


**Fig -1**: Overall Proposed System Architecture

## 4. EXPERIMENTAL SETUP

### 4.1 Image Processing

The images are collected for American Sign Language. Figure 2, Red-Blue-Green (RGB) images are collected and converted into gray-scale images for processing. Images are collected under plain and simple background condition. The images are captured from video sequence as frame by frame, stopped by keyboard interrupt and resized into 64 x 64 image resolutions. These images are saved as testing and training images. The below image Fig. 3 shows the front-end window that displays the threshold version of the user's gesture input.

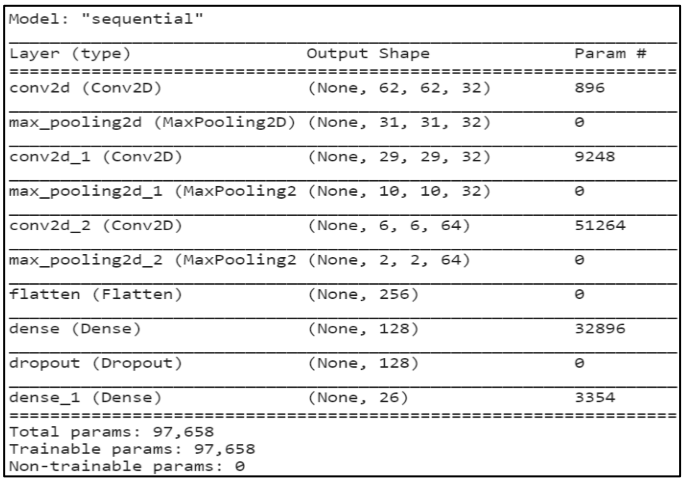
**Fig -2** User's Gesture Input

**Fig -3** Thresholded Input

### 4.2 Training

Sampled images within the dataset are trained using Convolutional Neural Network with three hidden layers to increase recognition rate. The system is trained for plain and simple backgrounds of pastel colors and augmentations like scale, shear and horizontal flip. Here two most vital parts of CNN training are feature extraction and therefore the classification. In deep learning concept, CNN could also be a category of deep neural network generally applied to vision related applications. The Convolutional Neural Network is used to detect complex features of data and also classifies the unstructured input data like images, text.

The multi-layered architecture of CNN is employed for feature extraction and classification. The building block includes Convolutional layer, Pooling layer and Fully connected layer alongside activation functions. The rectified linear measure is employed at hidden layers to avoid vanishing of feature descriptor and softmax activation function is employed at output layer for multiclass classification. Feature extraction is completed with series of convolutional layers and pooling layers. Fully Connected layer classifies the extracted features. The layers of CNN architecture are shown in Fig.4



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 10, 10, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 6, 6, 64)          51264
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 128)               32896
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 26)                3354
=================================================================
Total params: 97,658
Trainable params: 97,658
Non-trainable params: 0
```

**Fig -4**: CNN Architecture Parameter

Rectified Linear unit is a non-linear activation function which involves easier mathematical operations of various functions. The equation of ReLU is given by,

f(x) = max(0,x) …….(1)

where x is positive or 0 otherwise. Let I be the dimensions of the input image with Kernel size K, padding P and Stride S. Mathematically, the output of the convolutional layer is given by,

Output = $I - K + 2P\ S + 1$ …….(2)

For example, input image is 64x64 with kernel size 3, stride 1 and 0 padding, the output is given by 62. The parameter of the convolutional layer is given by,

Parameter = ((K)*S+1)*F)…….(3)

where, K is the kernel size, S is Stride and F is Filter For example, the parameter for kernel size 3, stride 1 and 32 filter is given by 320. Pooling layer is that the hidden layer and is employed to scale back the spatial size of the convolved feature. Here, we use maxpooling layer of size 2x2 to urge maximum values. The matrix obtained after pooling layer is named pooled feature map. The output of the pooling layer is given by half the convolved feature. Parameter passed for pooling layer is none. Output of flattening is obtained by multiplying previous pooling layer's output filter and current input filter. Parameter of fully connected layer is given by,

Parameter = (previous_layer+1)*current_layer ......(4)

For instance, current layer is output 128 and former layer output is 12754 then, parameter of fully connected layer is given by 1805770.

### 4.3 Testing

During CNN testing phase the system is tested with various plain pastel colored background images and with image augmentations like scale, shear and flip horizontal.

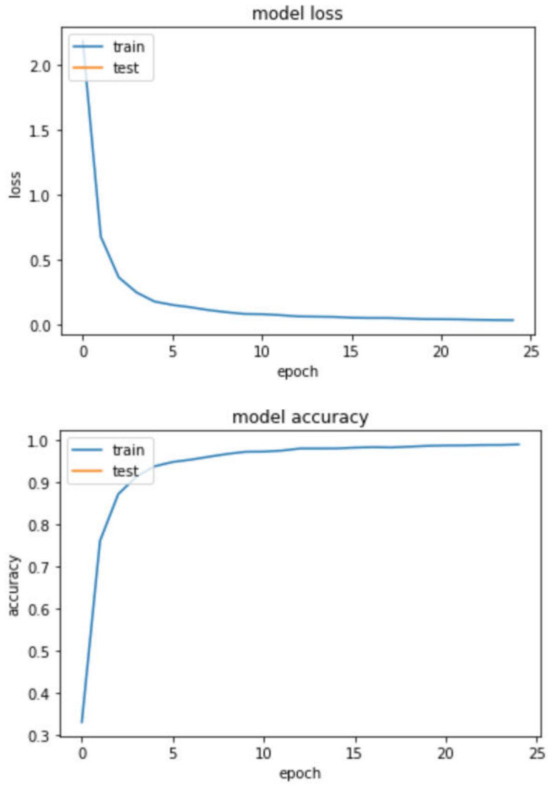
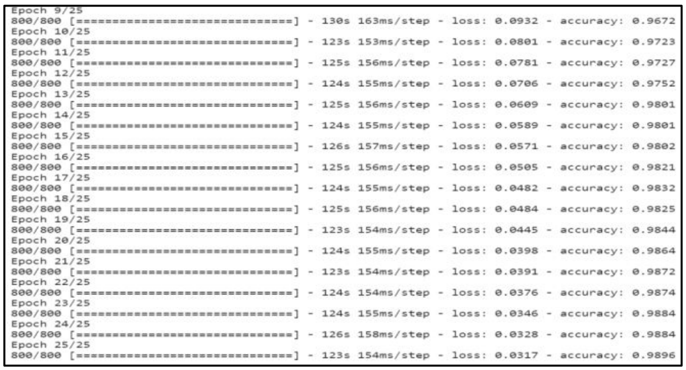


**Fig -5** Graphical representation of Accuracy and Loss obtained in CNN performance for 25 epochs.

Testing phase predict and display the text description corresponding to the given input hand gesture real time. Figure 5 shows the text displayed corresponding to the hand gestures in the Region of Interest (ROI).

The filter used here is 32 which obtained higher model accuracy from around 20 epochs in optimizer. Figure 5 shows representation of the accuracy and the loss occurrence in our CNN model for 25 epochs. The experiment also conducted by varying the number of filters like 32, 64 and 128 in five layers.



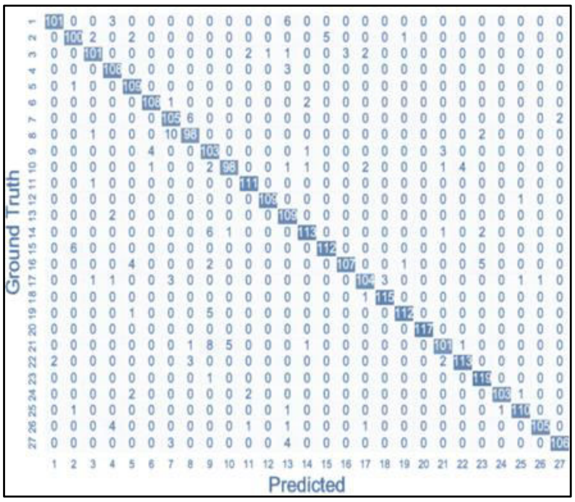
```
Epoch 9/25
800/800 [==============================] - 130s 163ms/step - loss: 0.0932 - accuracy: 0.9672
Epoch 10/25
800/800 [==============================] - 123s 153ms/step - loss: 0.0801 - accuracy: 0.9723
Epoch 11/25
800/800 [==============================] - 125s 156ms/step - loss: 0.0781 - accuracy: 0.9727
Epoch 12/25
800/800 [==============================] - 124s 155ms/step - loss: 0.0706 - accuracy: 0.9752
Epoch 13/25
800/800 [==============================] - 125s 156ms/step - loss: 0.0609 - accuracy: 0.9801
Epoch 14/25
800/800 [==============================] - 124s 155ms/step - loss: 0.0589 - accuracy: 0.9801
Epoch 15/25
800/800 [==============================] - 126s 157ms/step - loss: 0.0571 - accuracy: 0.9802
Epoch 16/25
800/800 [==============================] - 125s 156ms/step - loss: 0.0505 - accuracy: 0.9821
Epoch 17/25
800/800 [==============================] - 124s 155ms/step - loss: 0.0482 - accuracy: 0.9832
Epoch 18/25
800/800 [==============================] - 125s 156ms/step - loss: 0.0484 - accuracy: 0.9825
Epoch 19/25
800/800 [==============================] - 123s 154ms/step - loss: 0.0445 - accuracy: 0.9844
Epoch 20/25
800/800 [==============================] - 124s 155ms/step - loss: 0.0398 - accuracy: 0.9864
Epoch 21/25
800/800 [==============================] - 123s 154ms/step - loss: 0.0391 - accuracy: 0.9872
Epoch 22/25
800/800 [==============================] - 124s 154ms/step - loss: 0.0376 - accuracy: 0.9874
Epoch 23/25
800/800 [==============================] - 124s 155ms/step - loss: 0.0346 - accuracy: 0.9884
Epoch 24/25
800/800 [==============================] - 126s 158ms/step - loss: 0.0328 - accuracy: 0.9884
Epoch 25/25
800/800 [==============================] - 123s 154ms/step - loss: 0.0317 - accuracy: 0.9896
```

**Fig -6** CNN Performance

Figure 6 depicts the training performance of our model after 25 epochs, gradually achieving the accuracy of 98.96%, after running 800 runs for each epoch.

## 5. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this study, for experimental analysis, we had applied the above mention technique on our database of American Sign Language which consists of forty five thousand images i.e. 1,750 images per character and we was able to recognize all 26 characters from sign language and the developed approach is for static characters only. Figure 6 shows the accuracy rate (confusion matrix) for each hand gesture. When implementing the recognition system using background. subtraction, drawbacks and accuracy issues.

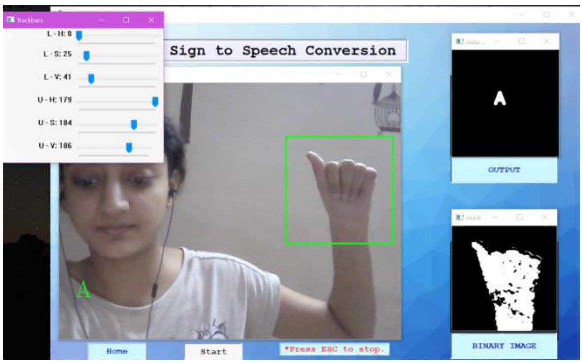


| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 101 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 100 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 108 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 109 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 108 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 105 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 8 | 0 | 0 | 1 | 0 | 0 | 0 | 10 | 98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 103 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 98 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 111 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 109 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 13 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 109 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 1 | 0 | 0 | 0 | 113 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
| 15 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 112 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 107 | 0 | 0 | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 1 | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 104 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 115 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 112 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 117 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 8 | 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 101 | 1 | 0 | 0 | 0 | 0 | 0 |
| 22 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 113 | 0 | 0 | 0 | 0 | 0 |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 119 | 0 | 0 | 0 | 0 |
| 24 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 103 | 1 | 0 | 0 |
| 25 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 110 | 0 | 0 |
| 26 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 105 | 0 |
| 27 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 108 |

**Fig –7** Confusion Matrix

Background subtraction cannot affect sudden, drastic lighting changes resulting in several inconsistencies. This method also requires relatively many parameters, which must be selected intelligently. Thanks to these complications faced, we made a choice to utilize contours, convexity defects to detect the item (hand). The mixture of those methods enabled us to realize a greater range of accuracy and overcome the challenges faced during the utilization of background subtraction.

The experiment also conducted by varying the amount of filters like 32, 64 and 128 in five layers architecture (3 convolutional layers, 2 fully connected layers). It shows a gradual increasing rate in training accuracy. From fig. 6 it's evident that 25 epoch is nice for accurate classification. In cases where the background wasn't plain, the objects within the background proved to be inconsistencies to the image capture process, leading to faulty outputs. Thus, the accuracy wasn't nearly as good, in scenarios with not plain background. After observing the results produced by the gesture recognition system in several backgrounds, it's recommended that this technique be used with a clear background to supply the simplest possible results and great accuracy. Below shown figure 8 shows the live results of the work where user gesture 'A' is shown and after image processing and CNN prediction the output of the gesture is given in style of text also as in speech.



L - H: 0
L - S: 25
L - V: 41
U - H: 179
U - S: 184
U - V: 186

**Fig –8** Practical Results

## 6. CONCLUSION

Hand Gesture Recognition and Text conversion helps Hearing Impaired person to speak with others. CNN helps to converts hand gestures to text with good performance with maximum accuracy because the system is trained for various affine transformations and straightforward backgrounds because the input images don't have self-occlusion or inter object occlusion the popularity rate is maximum. The performance of Convolutional Neural Network algorithm is evaluated by changing the optimizers and number of filters. It's observed that the training accuracy increases with the Adam optimizer and increase in number of filters. In future the efficiency and accuracy of the training are often studied by varying CNN hyper parameters and architectures.

## 7. FUTURE WORK

By properly studying the restrictions and shortcomings of this implemented processes like feature extraction and classification, a more accurate hand gesture recognition system are often developed. a bigger database will help the classification process to realize better accuracy. Features which are best in differentiating between the gestures got to be extracted. In future we might wish to improve the accuracy further and add more gestures to implement more functions. Finally, we target to increase our domain scenarios and apply our tracking mechanism into a spread of hardware including digital TV and mobile devices. We also aim to increase this mechanism to a variety of users.

## ACKNOWLEDGEMENT

We would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. The authors would like to thank the Department of Information Technology of Shri Ramswaroop Memorial College of Engineering and Management, Lucknow, Uttar Pradesh, India for their generous support. We would also thank to all Staff Members of Information Technology department for their valuable guidance. We also are thankful to reviewer for his or her valuable suggestions and also thank the faculty authorities for providing the specified infrastructure and support.

## REFERENCES

1. Abhishek B, Kanya, Meghana M, Md. Daaniyaal, Anupama H S, "Hand Gesture Recognition using Machine Learning Algorithms", Int. J. of Recent Technology and Eng (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019.
2. Ankita W, Parteek K. "Deep learning-based sign language recognition system for static signs", Neural Comput. and Applns, _ Springer-Verlag London Ltd., part of Springer Nature 2020.
3. Asifullah , Anabia , Umme Z, and Aqsa Saeed, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks",Computer Vision and Pattern Recog, 2020.
4. Ching-Hua , Eric , Caroline Guardino, "American Sign Language Recognition Using Leap Motion Sensor" , 13th Int. Conf. on Machine learning and Applications(ICMLA) pp 541-544,2014.
5. Di Wu, Lionel , Pieter-Jan , Nam Le, Ling , Joni , and JeanMarc Odobez, "Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition", IEEE Transactions On Pattern Analys. And Machine Intell., January 2016.
6. Jing-Hao , Ting-Ting , Shu-Bin , Jia-Kui , Guang-RongJi, "Research on the Hand Gesture Recognition Based on Deep Learning", 12th Int. Sympo. on Antennas, Propagation and EM Theory (ISAPE),2018.
7. Juan C, RaúlCabido, JuanJ. Pantrigo, AntonioS. Montemayor, JoséF. Vélez, "Convolutional Neural Networks and Long Short-Term Memory for skeletonbased human activity and hand gesture recognition", Int.J.Pattern Recog. 76,pp 80–94, 2018.
8. Juhi and Mahasweta , "Indian Sign Language Recognition Using ANN And SVM Classifiers", Int. Conf. on Innovations in info. Embedded and Commn. Systems (ICIIECS)2017.
9. Kshitij and Ying , "American Sign Language Recognition using Deep Learning and Computer Vision", IEEE Int. Conf. on Big Data, pp 4896-4899, 2018.
10. KusumikaKrori , Sunny ," Machine Learning Techniques for Indian Sign Language Recognition", Int, Conf. on Current Trends in Computer, Electrical, Electronics and Commun. (ICCTCEEC) pp 333-336,2017.
11. Marouane, Abdelhak "Machine Learning for Hand Gesture recognition Using Bag-of-words", Int. Conf. on Intelli. Systems and Computer Vision (ISCV), 2018