IJNRD.ORG ISSN : 2456-4184
International Research Journal
IJNRD
Research Through Innovation
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

REAL-TIME HAND GESTURE RECOGNITION FOR SIGN LANGUAGE INTERPRETATION AND TEXT GENERATION

[1]Prince Sindhu, [2]Anshika Sahai, [3]Kavya Tewari, [4]Pragya Verma, [5]Prof (Dr) Shailendra Narayan Singh

[1]Student, [2]Student, [3]Student, [4]Student, [5]Professor
[1]Department of Computer Science and Engineering,
[1]Amity School of Engineering and Technology, Amity University, Noida

Abstract: Effective communication with individuals who have hearing loss can be challenging, but sign language has emerged as a successful solution for those with hearing and speech disabilities. Sign language facilitates integration between individuals with disabilities and those without. However, it can be difficult for those who do not know sign language or understand it in a foreign language to interpret the hand gestures accurately. Fortunately, recent advancements in technology have provided new solutions to bridge this communication gap. A new study focuses on using a web camera to capture hand gestures, which are then identified by the system and displayed as an image name. The system uses the HSV color algorithm to detect hand motions and various computer vision methods to process the images. The area of interest, such as the hand motion, is then separated and analyzed using binary pixels. The system employs Convolutional Neural Networks (CNN) to classify the images, resulting in a high level of accuracy of over 95 percent.

Keywords: Computer Vision, Conventional Neural Network (CNN), Artificial Neural Network (ANN), Machine Learning, Gesture Recognition, Sign Language Recognition, Hue Saturation Value Algorithm (HSV)

1. INTRODUCTION

Sign language is a form of visual communication that conveys meaning through visible gestures made by the speakers. It is often used as a substitute for speech by those who are deaf or have difficulty speaking. As a result, sign language has been the focus of extensive research over the years. Various sign languages, including American Sign Language, British Sign Language, Taiwanese Sign Language, and more, have been studied extensively. However, Indian Sign Language has not received the same level of attention in this field, with only a limited amount of research conducted on this particular sign language.

The lack of interest in learning sign language and the limited interaction with the hearing-impaired community have resulted in their marginalization in society. This feeling of isolation and being cursed leads to a sense of ignorance and loneliness among the hearing-impaired. To address this issue, sign language recognition technology has been developed, which has significant implications both technically and socially.

Our research aims to bridge the communication gap between hearing-impaired individuals and the general population by providing a cost-effective method of sign language recognition. Our system enables users to understand signs without the need for a professional interpreter. This is achieved by using computers in the communication process to capture, process, and recognize signs. Our technology provides a means for hearing-impaired individuals to communicate more effectively and become more integrated into society.

Several scholars have used different methods to identify sign language and hand motions. Some studies utilized static hand gestures[1], while others focused on real-time footage. To identify various hand motions in videos, researchers have used neural network-based features and hidden Markov models[2]. Skin filtering[3], moment invariants-based features[4], and Artificial Neural Networks (ANN)[5] have also been used to recognize various motions with high success rates. However, some studies faced limitations, such as using data gloves[6], which resulted in inaccurate identification of single hand motions.

To overcome these challenges, we propose a specialized image processing approach based on Convolutional Neural Networks (CNN) and ANN to accurately identify different hand sign signals in live video sequences. By conducting the experiment with bare hands, we eliminate the difficulties associated with using gloves. Our strategy has achieved a hand gesture recognition rate of over 95%. In this paper, we extend our work to video frames, providing an improved method for sign language recognition.

2. SIGN LANGUAGE AND SYSTEM OVERVIEW

2.1 HAND SIGNS

Sign language has facilitated communication between deaf and hearing individuals, allowing them to interact seamlessly. In this paper, we focused on American Sign Language (ASL) alphabets, which can be signed using one hand or both hands.

Figure 1 illustrates the 26 alphabets and 10 numeric hand gestures that we considered in our study.

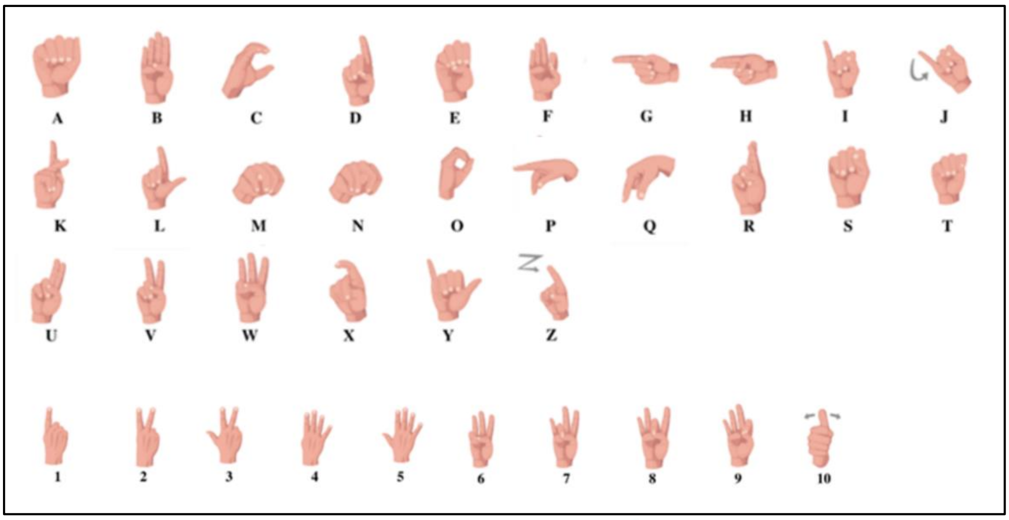


fig 1: various alphabets & number's hand gestures[7]

2.2 SYSTEM OVERVIEW

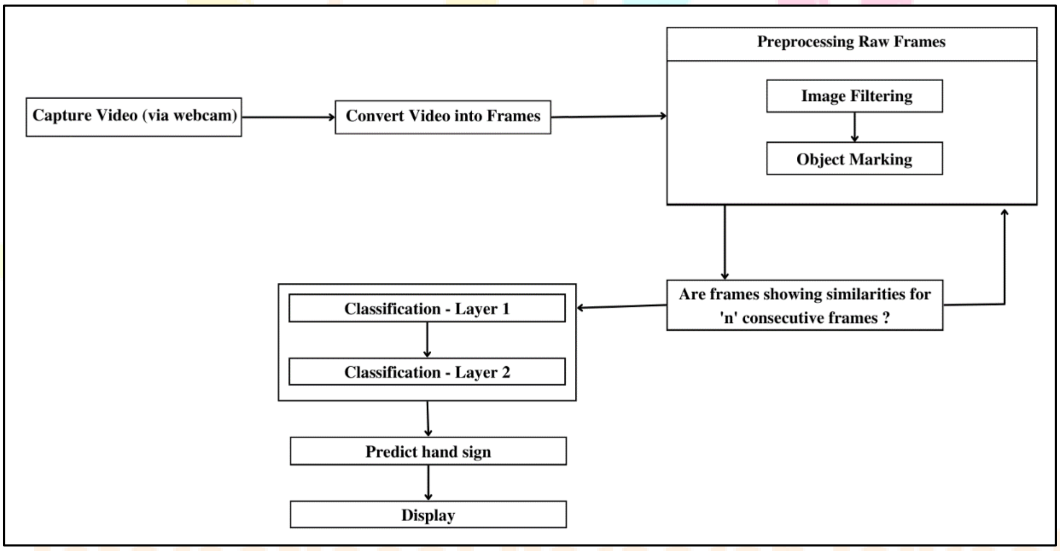


fig 2: proposed system (overview)[8]

Figure 2 illustrates the three primary stages of our proposed system. The first stage involves converting the input video into frames for pre-processing, where techniques such as video frame filtering and object marking are utilized to detect hand gestures in the frames. The first classification layer then identifies objects among the 37 hand gestures. The second classification layer further aids in accurately identifying hand gestures within groups of similar-looking hand gestures. In the subsequent sections, we will delve into the details of each stage.

3. PREPROCESSING OF SIGN LANGUAGE RECOGNITION

3.1 DATA ACQUISITION

In our proposed approach, the first stage is the capture of webcam video, where multiple hand gestures were considered. A total of 10 numeric hand gestures and 27 unique alphabets were included in the testing phase. Figure 3 showcases a selection of the collected continuous video frames.

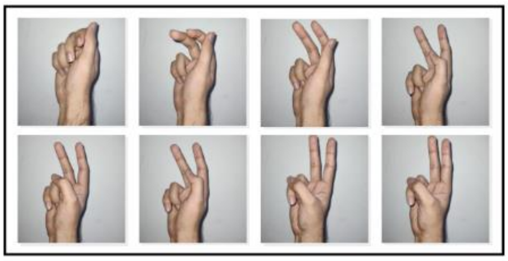


fig 3: sequence of continuous video frames[9]

3.2 HAND GESTURES DETECTION

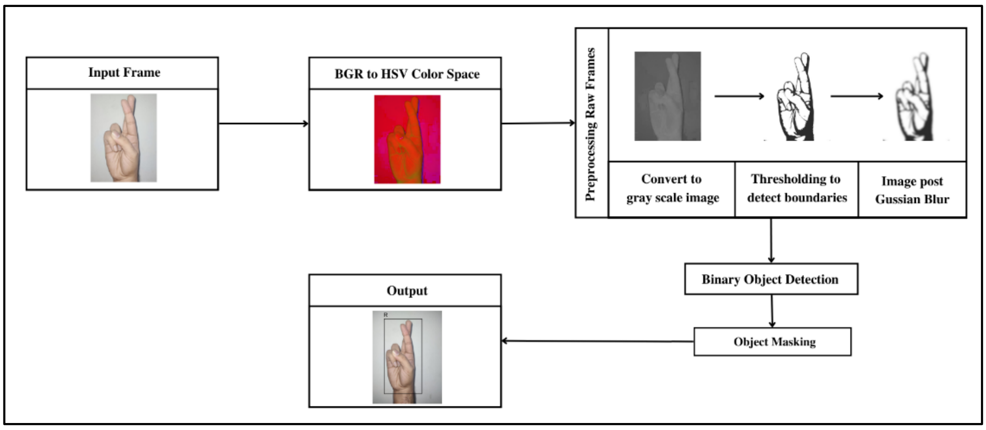


fig 4: pre-processing process diagram[10]

In order to identify hand gestures, the input video frames underwent image filtering to extract the relevant hand from the background. The thresholding technique was then used to outline the hand boundary and identify gestures. The steps performed during this stage are shown in Figure 4.

The first step in gesture detection involved converting the input frame into the HSV color system, which is less affected by changes in lighting compared to RGB. The frame was then filtered, blended, and converted to grayscale. The boundary of the hand was identified through thresholding, resulting in a final image with a white background and black hand boundaries, as shown in Figure 5. This processed image was used for further classification.

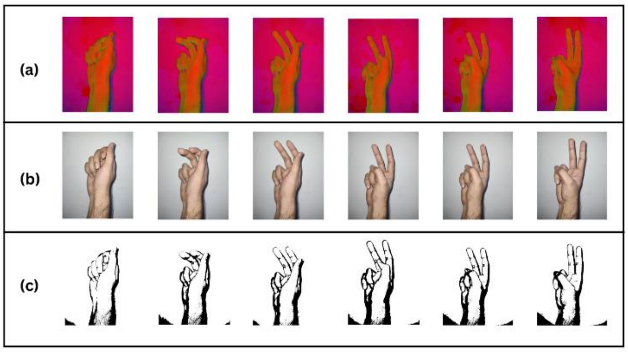


fig 5: (a) hsv color space of frames, (b) grayscale result, (c) final output after thresholding[11]

4. DATA PRE-PROCESSING AND FEATURE EXTRACTION

In [12], hand detection is achieved using background subtraction and threshold-based colour detection. Adaboost face detector is used to differentiate between faces and hands due to their similar skin tones. Gaussian blur filter is used to extract the necessary image for training, as described in [13]. Instead of applying filters to video-extracted data, instrumented gloves, as explained in [14], can be used to save pre-processing computation time and provide clearer and more accurate data.

Manually segmenting an image using colour segmentation techniques was attempted but proved unsuccessful due to the effect of lighting on skin tone and colour, as stated in the research paper[15]. Therefore, it was decided to keep the background of the hand a stable single colour to avoid segmenting it based on skin colour. With a large number of symbols to be trained for the project, including similar-looking gestures such as the symbol "V" and digit "2," it was decided to focus on improving the accuracy of the system.

5. CLASSIFICATION

Our approach for this project involves two layers of algorithm for predicting the user's final symbol. The first layer of the algorithm includes the following steps:

1. Applying the Gaussian blur filter and threshold to the frame captured using OpenCV to obtain the processed image after feature extraction.
2. Providing this processed image to the CNN model for prediction, and if a letter appears in more than 50 frames, it is printed and taken into account when forming the word.
3. The blank symbol is used to indicate spaces between words.

The second layer of the algorithm involves the following steps:

1. Identifying various symbol sets that produce similar outcomes when detected.
2. Classifying between those sets using classifiers that are designed specifically for those sets.

5.1 LAYER 1

The CNN model for this project consists of several layers. The input image has a 128x128 pixel resolution and is first processed by a 3x3 filter with 32 filter weights in the first convolutional layer, producing a 126x126 pixel image. The first pooling layer down samples the images using maximum pooling of 2x2, resulting in a 63x63 pixel image. The second convolutional layer processes the

63x63 pixel image using 32 filter weights of size 3x3, producing a 60x60 pixel image. The resulting image is then down sampled to a resolution of 30x30 using a maximum pool of 2x2 in the second pooling layer.

The output of the second pooling layer is reshaped into an array of 30x30x32 values and passed through a fully connected layer with 128 neurons in the first layer with dense connectivity. The second densely connected layer receives the output of these layers, and to prevent overfitting, a dropout layer with a value of 0.5 is used. Another fully connected layer with 96 neurons uses the output from the first densely connected layer as input.

The ReLu activation function is used in each layer (convolutional and fully connected neurons) to add nonlinearity and learn more complex features. Max pooling with a pool size of (2, 2) is applied to the input image to reduce the number of parameters and overfitting. Dropout layers are used to prevent overfitting by randomly dropping out a set of activations in the layer, helping the network provide the correct classification or output for a given example.

The Adam optimizer is used to update the model in response to the loss function's output. Adam combines the benefits of adaptive gradient algorithm (ADA GRAD) and root mean square propagation, two extensions of two stochastic gradient descent algorithms (RMSProp).

5.2 LAYER 2

In order to improve the accuracy of symbol identification, we are implementing a two-layered approach that predicts and validates symbols that are closely related. Through our experimentation, we observed that certain symbols were being misidentified or were frequently associated with other symbols. These include R and U; T, A and E; S, M, and N;7 and F: and I and J; To address these issues, we developed three separate classifiers to categorize these symbol sets, namely {R, U}, {T, A, E}, {S, M, N}, {F, 7} and {I, J}.

6. EXPERIMENTAL RESULTS AND ANALYSIS

Real-Time Hand Gesture Recognition for Sign Language Interpretation and Text Generation was implemented using VSCode (Version 1.70) as software and 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz processor machine, Windows 11 Home (64 bit), 8GB RAM and a laptop integrated 0.95 MP webcam with 1280x720 resolution.

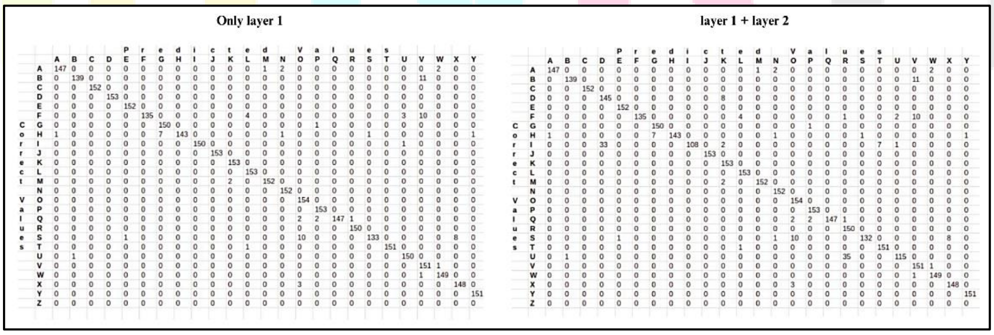
6.1 DATASET AND PARAMETERS CONSIDERED

In our project, we were unable to find any pre-existing datasets that met our requirements and were available in the form of raw images. We found datasets in the form of RGB values, but they were not suitable for our needs. Therefore, we decided to create our own dataset using the Open Computer Vision (OpenCV) library. Our dataset consists of approximately 800 images for each ASL symbol, used for training, and around 200 images for each symbol used for testing.

To capture these images, we used the computer's webcam to take a picture of each frame. A square is drawn around the Region of Interest (ROI) in each frame. We then convert the RGB image of the ROI to grayscale using OpenCV. To extract different features from the image, we apply a Gaussian blur filter to it. This step is followed by thresholding, which is used to separate the foreground from the background. Once the image is pre-processed, it is ready for use in training and testing our models.

6.2 RESULT AND RECOGNITION RATE

We achieved a 85.8% accuracy using only the first layer of our algorithm. Combining both layers, our accuracy increased to 95.0% as shown in graphs below in figure 6, surpassing recent research papers on American Sign Language that typically employ Kinect-like devices for hand detection. For instance, in [16], a Flemish sign language recognition system is built using convolutional neural networks and Kinect, achieving a 2.5% error rate. [17] employs a vocabulary of 30 words and a hidden Markov model classifier to attain an error rate of 10.90%. [18] reports an average accuracy of 86% for 41 static gestures in Japanese sign language. In [19], the accuracy of 83.58% and 85.49% is achieved for new signers and 99.99% for observed signers using depth sensors, with CNN used for their recognition system. Our model doesn't use a background subtraction algorithm, unlike some of the models mentioned above, and this may affect its accuracy when such a process is applied. However, our goal was to create a project that could be used with widely available resources, and thus, we utilized a standard laptop webcam rather than a costly and hard-to-obtain Kinect sensor.



Only layer 1

						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	0	10	0	0	0	133	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

layer 1 + layer 2

						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

fig 6: accuracy plots

7. CONCLUSION AND FUTURE WORK

This study presents the creation of a practical Real-Time Hand Gesture Recognition system for Sign Language Interpretation and Text Generation specifically designed for individuals with disabilities and motor impairments. The system is designed to recognize ASL alphabets, and after applying two layers of algorithms, we achieved a final accuracy of 95.0% on our dataset. The algorithm improves prediction by verifying and predicting symbols that are more similar to each other. However, the accuracy may vary depending on factors such as the quality of the display, the presence of background noise, and sufficient lighting. We are currently experimenting with different background subtraction algorithms to enhance accuracy even in poorly lit and complex backgrounds. In addition, we are considering developing a model that predicts whole words and sentences to aid D&M individuals in conveying their message through this system.

8. ACKNOWLEGDGEMENT

We are sincerely thankful to Prof (Dr) Sanjeev Thakur, Head of the Department & Prof (Dr) Shailendra Narayan Singh, Department of Computer Science & Engineering, Amity University, Noida for their kind help and encouragement for the proposed work.

9. REFERENCES

[1] Islam, M.Z., Hossain, M.S., Islam, R.U. and Andersson, K. 2019. Static Hand Gesture Recognition using Convolutional Neural Network with Data Augmentation. In: 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR). IEEE.

[2] Moni, M.A. and Ali, A.B.M.S. 2009. HMM based hand gesture recognition: A review on techniques and approaches. In: 2009 2nd IEEE International Conference on Computer Science and Information Technology. IEEE.

[3] Sun, J.-H., Ji, T.-T., Zhang, S.-B., Yang, J.-K. and Ji, G.-R. 2018. Research on the Hand Gesture Recognition Based on Deep Learning. In: 2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE). IEEE.

[4] Rahagiyanto, A., Basuki, A. and Sigit, R. 2017. Moment Invariant Features Extraction for Hand Gesture Recognition of Sign Language based on SIBI. EMITTER International Journal of Engineering Technology, 5(1).

[5] Singha, J. and Laskar, R. 2015. ANN-Based Hand Gesture Recognition Using Self coarticulated Set of Features. IETE Journal of Research.

[6] Lu, D., Yu, Y. and Liu, H. 2016. Gesture recognition using data glove: An extreme learning machine method. In: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE.

[7] Hand showing sign language alphabets illustration set. Retrieved from https://www.freepik.com/

[8] Prince Sindhu (2023). Primary stages of hand sign recognition system. Unpublished figure.

[9] Screenshots of continuous input frames. Unpublished figure.

[10] Prince Sindhu (2023). Flow chart representing analysis and feature extraction from a input frame using various image filtering techniques. Unpublished figure.

[11] Prince Sindhu (2023). Feature extraction from continuous image frames. Unpublished figure.

[12] T. Yang, Y. Xu, and "A. , Hidden Markov Model for Gesture Recognition", CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ.,Pittsburgh,PA, May 1994.

[13] Information is based on this link: https://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

[14] Mohammed Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language.

[15] Abdulateef, S.K. and Salman, M.D. 2021. A Comprehensive Review of Image Segmentation Techniques. Iraqi Journal for Electrical and Electronic Engineering, 17(2), pp.166-175.

[16] Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham.

[17] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision based features. Pattern Recognition Letters 32(4), 572–577 (2011) 25.

[18] N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," 2017 Nicograph International (NicoInt), Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9.

[19] Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen" Real-time sign language fingerspelling recognition using convolutional neural networks from depth map" 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR).

[20] Accuracy graph plots based on predicted value vs actual value of tested data on different models.