"DactyloCom: Advancing Accessibility for the Stone-deaf and Hearing-Impaired Community through Dactylology"

Raghav Mehrotra[1][0009-0009-5855-9687] Piyush Motwani[2][0009-0000-1107-1160] Om Chhayala[3][0009-0001-9438-816X] Dishaa Patni[4][0009-0003-5018-1057] Kanan Palya[5][0009-0008-1594-6399] and Jyoti Kukade[6][0009-0009-4633-0965]

[1,2,3,4,5,6] Medi-Caps University, AB Rd, Pigdamber, Rau, Indore, Madhya Pradesh

Abstract: Truly disabled people are those who are unable to speak or hear. The proposed model aims to bridge the communication gap between those who are mute or deaf by developing a gesture language-to-text converter that leverages Convolutional neural networks (CNN) and socket technology. The CNN model will examine the gestures made by a finger language interpreter and will transcribe those into text simultaneously using a substantial dataset of sign language gestures to ensure high accuracy. The converted text will be sent to clients via socket technology, enabling effective communication and better understanding. With the ability to join in conversations, comprehend lectures, and interact with others more readily, this device has the potential to greatly raise the quality of life for those who suffer from hearing loss. The transformed text is transmitted smoothly because of socket technology, which makes the strategy both economical and efficient. With CNN and deliberate application of two-layer theory, the accuracy of the first layer was 95.8%, and the total accuracy of the two layers was 98.0%. Therefore, by removing obstacles to communication and strengthening social ties for people with hearing loss, this program has the potential to advance inclusion.

Keywords: Dactylology, Gesture recognition, Hearing impairments, Convolutional neural networks (CNN), Socket technology, Sign language-to-text converter, Communication challenges.

1. Introduction

One of the most important aspects of life is communication, which can be difficult for those who have hearing loss. Finger spelling, also known as dactylology, is a kind of nonverbal communication in which words and letters are represented by the fingers. For persons who are hard of hearing, this practice is highly helpful as it enables them to interact with people in their immediate environment. Spelling words and sentences correctly is made possible by dactylology, which enhances communication clarity and precision. It is a crucial part of the language used by the deaf community and a useful tool for learning new terms and phrases. However, there remains a communication barrier because not everyone is able to interpret sign language. A few years ago, notable advancements occurred with the invention of text-to-sign language conversion technologies that help people who use and do not use sign language communicate with one another. Our goal is to create a text-to-sign language converter that employs socket technology and convolutional neural networks (CNN) to provide the translated message to users.

To obtain the best possible recognition of sign language gestures, we used our dataset and advanced preprocessing techniques including thresholding and Gaussian blur. After preprocessing, we trained an extremely effective CNN model that could accurately recognize and interpret signs. To further maximize communication for those with hearing loss, we integrated socket technology, which allows instantaneous transfer of translated text in real time. Easy and effective communication is fostered by this seamless integration, which results in more fruitful interpersonal interactions.

The earlier study details the creation of a sign language to text converter system using advanced preprocessing methods including Gaussian blur and thresholding on a sizable dataset. This preprocessing step significantly improved the system's capacity to recognize sign language motions more accurately and efficiently. Following preprocessing, we constructed a highly effective Convolutional Neural Network (CNN) model that possesses exceptional accuracy in signal recognition and interpretation. The model gained resilience and the capacity to recognize unusual or complex messages from being trained on a wide range of gesture-based actions.

Our sign language-to-text converter system's smooth socket technology integration encourages inclusivity and fair communication access. It is a flexible communication tool that is very accurate and effective in a range of settings.

2. Related Work

Many more tactics that use a variety of approaches and techniques have been developed and are being researched. Text-to-sign language converter systems have been the focus of extensive research in recent years. The majority of modern systems employ machine learning methods like computer vision and deep learning to detect and decipher sign language motions. In sign language, convolutional neural networks (CNNs) are a frequently employed method for hand motion recognition.

While Kukade and Panse et al. [1] create a vocabulary of typical occurrences from movie scenes, they have trouble identifying outliers. In order to discover anomalies, they investigate poorly supervised multi-instance learning (MIL), segment movies for effective training, and present a deep learning model that combines CNNs and LSTM networks. Their model promises advancements in the video processing profession by accurately locating anomalies in surveillance footage.

CNN algorithms and vision-based methods were used by Reddy et al. [2] to develop an interactive ASL recognition system. Their method is able to accurately identify ASL motions (94.8%), especially finger spelling. It fills in the communication gaps for the deaf and mute by providing speech or text outputs.

Using a deep learning model that has already been trained, Kaur et al. [3] suggest a real-time method for converting sign language to voice. Their technique greatly improves the accessibility of communication for the deaf, with accuracy rates of 98.23% for training and 97.07% for validation. A significant advancement in inclusive communication technology has been made with the integration of deep learning with Jetson Nanotechnology.

The limitations of conventional techniques are emphasized in Kukade and Panse et al.'s [4] analysis of anomaly detection in surveillance footage. Promising substantial improvements in security and public safety, their deep learning approach combines LSTM networks with CNNs. Testing with the UCSD Pedestrian dataset attempts to demonstrate its capability for identifying different kinds of irregularities.

SignSpeak is a mobile application designed by Rane et al. [5] that uses CNN technology to translate sign language fluently. Their research demonstrates the efficacy of SignSpeak in overcoming communication gaps by carefully analyzing current technologies. Its development process and operational functioning are explained in detail, and future research directions are suggested.

Kukade et al. [6] are in favor of using autonomous vehicles for crime detection and warning generation in an effort to combat the rising incidence of violence in society. Their real-time video processing method achieves object detection accuracy of over 90%, which enhances surveillance and reduces crime rates.

A method that translates American Sign Language (ASL) into text and speech is presented by Bharathi et al. [7] in order to help those with speech and hearing impairments communicate more easily. Their technique effectively bridges communication gaps by achieving an astonishing 88% accuracy in ASL gesture detection using CNNs.

S Rajaganapathy et al.'s system [8] converts voice from human sign language by utilizing motion capture technology. Their creative method, which makes use of Microsoft Kinect, provides smooth interaction without the need for conventional controls, marking a substantial breakthrough in accessible technology.

In order to characterize Indian Sign Language (ISL), machine learning approaches are investigated by Radha S. Shirbhate et al. [9]. Their technique, which offers a promising method for ISL recognition, combines skin segmentation, feature extraction, and supervised learning models.

Using a variety of sensors, Nicholas Born et al. [10] create a glove-based system for hand sign language detection. Their system efficiently translates hand movements using classification techniques such as Parallelepiped and Euclidean distance classifiers.

A framework for weakly supervised continuous sign language recognition using deep neural networks is proposed by Runpeng Cui, Hu Liu, Changshui Zhang et al. [11]. Their method extracts sequence labels from video performance image sequences with accuracy by fusing RNNs and CNNs.Grobel and Assan et al.

Utilizing HMMs, [12] analyzes isolated sign recognition, attaining up to 94% identification rates. Their system offers promising improvements in sign language recognition by concentrating on the manual components of sign language.

Deep convolutional networks are used by Bheda and Radpour et al. [13] to study ASL gesture detection with the goal of improving multimodal communication. In order to facilitate efficient interpretation, their research focuses on precisely categorizing ASL pictures of letters and numbers.

Using multi-dimensional HMMs, Yang and Xu et al. [14] offer a gesture-based system that achieves excellent accuracy in gesture detection tasks. Applications involving human-computer interfaces and telerobotics could greatly benefit from their approach.

Pigou et al. [15] used CNNs and Kinect to create an autonomous ASL recognition system that recognized gestures with a high degree of accuracy. In gesture-spotting competitions, their system does well and has high generalization skills.

A method for recognizing sign language using vision-based characteristics and HMMs is proposed by Zaki and Shaheen et al. [16]. The RWTH-BOSTON-50 database yields a recognition error rate of 10.90% for their system.

Using pattern recognition techniques, Mukai et al. [17] present a method for identifying fingerspelling in Japanese sign language. With an average accuracy rate of 86%, their method can differentiate between fingerspellings that are easily recognized and those that are more challenging.

With good accuracy for both alphabets and numbers, Kang et al. [18] demonstrate an automatic fingerspelling recognition system that uses CNNs from depth maps. With a 3 ms processing time per image prediction, their technology exhibits the highest accuracy and speed.

Comparing the stress levels of HI and normal students, Notash and Elhamkia et al. [19] find that HI students have greater stress levels. They promote therapy and preventative actions.

In order to support the cognitive, psychological, and cultural development of deaf people, Ghari et al. [20] emphasize the importance of sign language. In order to overcome these obstacles, sign language is crucial, since their study demonstrates how communication limitations affect education, emotions, and culture.

3. Proposed Model

3.1 CNN Architecture

The proposed system by Vivek Bheda and Dianna Radpour[6] inspired the model's design. This paper completed a task for ASL gesture classification. The model comprises fully connected layers, convolutional blocks, and a final classification layer. The CNN architecture extract features from skin-segmented images. It includes three convolutional blocks, each with two 2D convolutional layers followed by max pooling, ReLU activations, and dropout layres, concluded by two convolutional layers with identical activation functions. The kernel sizes are kept constant throughout the model at 3x3.

The 128x128 pixel input image is subjected to 32 filter weights, each measuring three by three pixels, in the first convolutional layer. Utilizing the maximum 2x2 pooling for downsampling, the method produces an image resolution of 63x63 pixels. Subsequently, the second convolutional layer is applied, yielding an image resolution of 60x60 pixels, with 32 filter weights and 3x3 pixel sizes.The downsampling process is then repeated using the maximum pooling of 2x2 to create images with a resolution of 30x30 pixels. The 128 neurons in the neural network's first layer convert the input pictures into a 30x30x32 array, yielding 28,800 numerical values. To prevent overfitting, a dropout layer with a dropout rate of 0.5 is employed. The output of the first layer is received by the 96 neurons that comprise the second fully connected layer. After receiving the output from this layer, the last layer—which houses the alphabet and the blank symbol—accommodates as many neurons as there are classes.

Rectified Linear Unit (ReLU) is an activation function observed in all layers of convolutional and fully connected neurons. By applying nonlinearity to every input pixel, the Max (x, 0) function enhances the model's ability to describe complex characteristics. Additionally, ReLU quickly solves the vanishing gradient issue and speeds up training by cutting down on computation time. ReLU needs to be applied at every layer in order to increase the model's effectiveness and efficiency in learning complex features. ReLU is also utilized by the pooling layer of a neural network, which sets a pool size of (2, 2) for maximum pooling on the input image.This is a useful technique for minimizing the number of parameters, which prevents overfitting issues and reduces computing costs.

Overfitting is a prevalent issue in neural networks. This is addressed by using the Dropout method. In order to prevent the network from become overly reliant on any one feature, it accomplishes this by randomly deactivating certain neurons throughout training. The Dropout layer appears before the output layer and after the fully linked layers in a neural network architecture.This phenomenon often leads to a deterioration in the network's performance in real-world applications. Thus, it's critical to create methods that prevent overfitting and enhance neural networks' capacity for generalization. To address this, the network has dropout layers, which randomly remove certain units during training, avoiding overfitting and prevents any one unit from becoming too dependent on another.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
conv2d (Conv2D)             (None, 128, 128, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 64, 32)       0
_________________________________________________________________
conv2d_1 (Conv2D)           (None, 64, 64, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 32)       0
_________________________________________________________________
flatten (Flatten)           (None, 32768)             0
_________________________________________________________________
dense (Dense)               (None, 128)               4194432
_________________________________________________________________
dense_1 (Dense)             (None, 128)               16512
_________________________________________________________________
dropout (Dropout)           (None, 128)               0
_________________________________________________________________
dense_2 (Dense)             (None, 96)                12384
_________________________________________________________________
dropout_1 (Dropout)         (None, 96)                0
_________________________________________________________________
dense_3 (Dense)             (None, 64)                6208
_________________________________________________________________
dense_4 (Dense)             (None, 27)                1755
=================================================================
Total params: 4,240,859
Trainable params: 4,240,859
Non-trainable params: 0
```

Fig. 1. Model Summary

3.2 Architecture Context Diagram:

The ensuing context diagram shows the general organization and connections of the architecture:

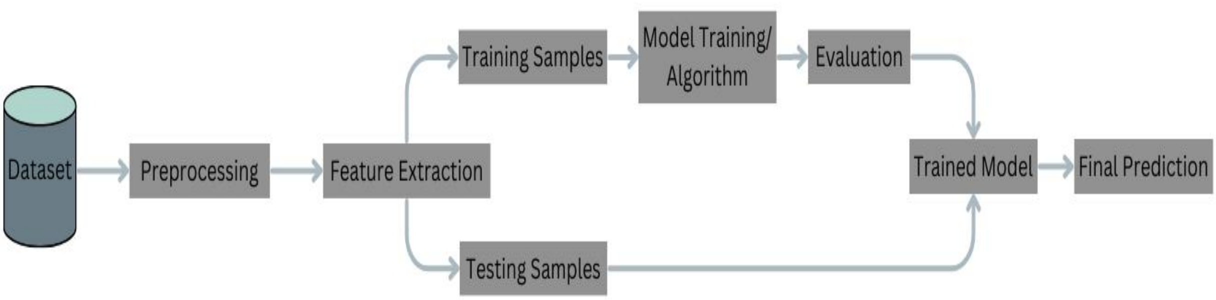


Fig. 2. Architecture Diagram

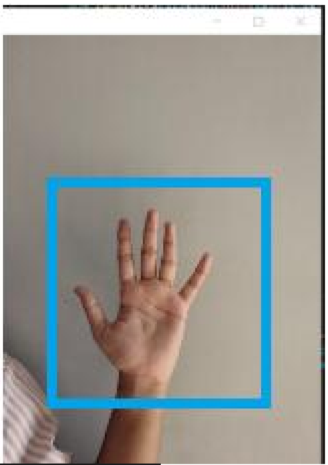
4. Experiment Details

4.1 Data Collection:

Use the following methods to learn more about hand movements:

Using electromechanical devices to achieve precise hand configuration and placement is normal technique. Glove-based methods, however costly and complex, are commonly used to extract data. A vision-based method is the use of the webcam on a computer as an input device to record hand and/or finger movements. With the use of simply a camera, this approach not only saves money but also permits smooth and simple user-computer interaction.

The suggested system uses a vision-based methodology, meaning that all signs can be read with just hands, negating the need for any artificial equipment. Sadly, we were unable to find any pre-existing raw image datasets that satisfied our exact requirements. As a result, we discovered datasets with RGB values and used the OpenCV library to create our own dataset. We painstakingly collected about 200 photographs for each American Sign Language (ASL) symbol for testing, and an astounding 800 images for training each symbol. We marked the region of interest (ROI) with a blue square surrounded by a blue border. We collected the data by taking a picture of each frame that came from the webcam.



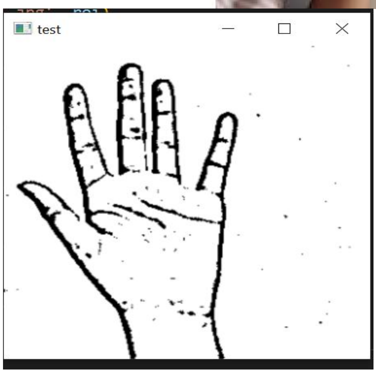


Fig. 3. Display Window

Fig. 4. ROI (Region of Interest)

4.2 Data Pre-processing:

The suggested method for hand detection made use of a number of methods, such as threshold-based color detection and background subtraction. Given their comparable skin tones, it can be difficult to discern between hands and faces; this is where the AdaBoost face detector comes in handy. With help from OpenCV, the group applied the Gaussian Blur filter to the image in order to smooth it out before training. Because lighting circumstances affect skin tone and color, it was found that manually segmenting a picture using color segmentation approaches was inefficient. Because many of the necessary training symbols were identical, the team decided against segmenting the hand based on skin tone and instead to keep the hand's background color constant. This strategy is anticipated to yield more accurate results.

4.3 Visual Classification:

Our method involves the usage of duo layers of algorithms to forecast the ultimate symbol accurately.

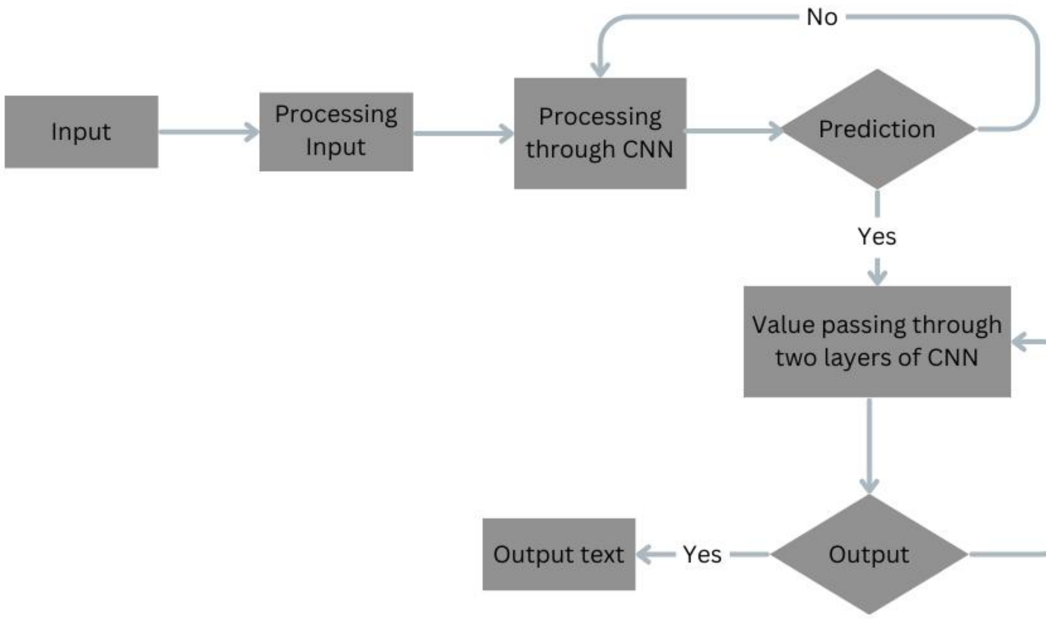


Fig. **5**. System's gesture classification process

Algorithm-

Layer-1:

After capturing a frame with the help of OpenCV, the obtained image is further processed by applying a Gaussian Blur filter and threshold to improve its quality. The CNN model takes the processed image as an input to make predictions. To accurately identify a letter as part of a word, the CNN model must recognize it across more than fifty frames. Furthermore, the vacant symbol serves to denote the gap between words.

Layer-2:

We have discovered an alternative symbol set that, when identified, produces similar results. To address this, we have employed classifiers to distinguish between these sets. Our testing revealed that certain symbols such as D, U, I, and S were often misclassified as other symbols.

We developed three diverse classifiers to handle the following sets:

{D, R, U},

{T, K, D, I},

{S, M, N}

4.4 Integration with Socket Technology:

To enable the transmission of the converted message to the client, our project utilizes socket technology. Specifically, we have developed a server-side script that listens for incoming connections on a designated port. When a client establishes a connection with the server, the server accepts the connection and stores the client's socket object. Once the server has received the converted message from the CNN model, it sends the message to the client utilizing the client's socket object. We have created a script that enables the client to connect to the server by using the server's IP address and port number. Once the connection to the server is established, the client receives a byte stream through the socket object, which is then converted into a string message. Finally, the converted message is displayed on the client-side user interface. This seamless process allows for the efficient and effective transmission of the converted message from the server to the client.

4.5 Testing and Evaluation:

Our two-layer method achieved 95.8% accuracy for the first layer and 98.0% for both layers combined. This accuracy is higher than that of most current American sign language research papers. We did not incorporate a background-subtraction algorithm, which could affect the accuracy if applied. Unlike many models that require expensive and inaccessible devices such as Kinect, we aimed to create a model that can be used with commonly available resources. Our model utilizes a simple webcam, making it accessible to a broader audience.

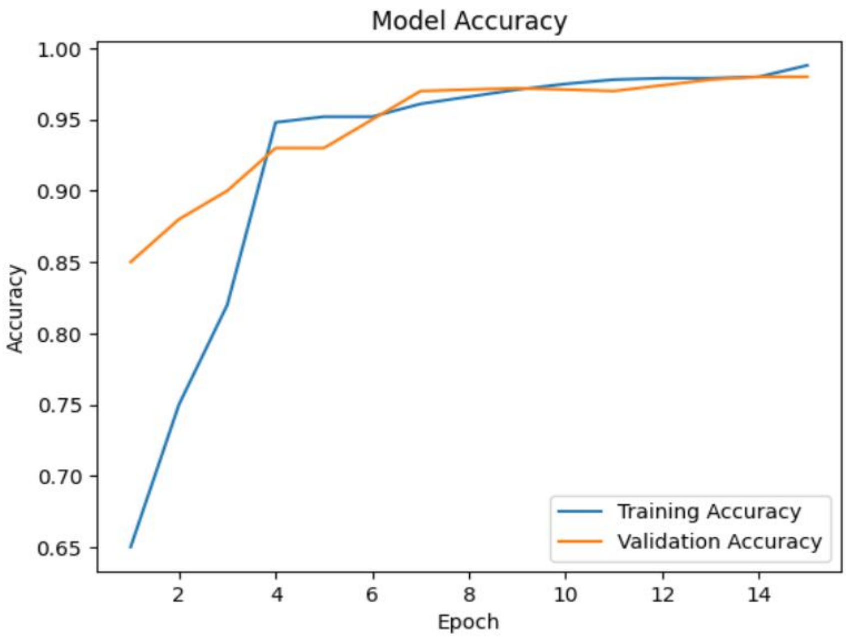


Fig. 6. Model Accuracy Graph

Below are the confusion matrices of our results:

						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

Algo 1

Fig. **7**. Resulting Confusion Matrices of Algorithm 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

Algo 1 + Algo 2

Fig. **8**. Resulting Confusion Matrices Algorithm 1 + Algorithm 2

5. Conclusion

DactyloCom, the sign language to text converter using CNN and socket technology is a highly effective and efficient way. Our CNN model was trained on sign language motions using data collection and preprocessing methods like Gaussian blur and thresholding to accurately convert motions into text.

The real-time transmission of the converted message is made possible by the integration of socket technology, which enables seamless communication between the server and the client. Overall, this strategy has the potential to significantly enhance communication for those who primarily use sign language. Even more advanced and precise sign language-to-text converters might be created with additional research and development.

References

1. Kukade, Jyoti, and Prashant Panse. "Advanced Deep Learning Model for Anomaly Detection Based Video Surveillance System." *International Journal of Intelligent Systems and Applications in Engineering* 12, no. 5s (2024): 477-485.

2. Reddy, C. Kishor Kumar, Sahithi Reddy Pullannagari, Srinath Doss, and P. R. Anisha. "Dactylology Prediction Using Convolution Neural Networks." *In Fostering Cross-Industry Sustainability With Intelligent Technologies,* pp. 61-72. IGI Global, 2024.

3. Kaur, Binwant, Aastha Chaudhary, Shahina Bano, Yashmita, S. R. N. Reddy, and Rishika Anand. "Fostering inclusivity through effective communication: Real-time sign language to speech conversion system for the deaf and hard-of-hearing community." *Multimedia Tools and Applications* (2023): 1-22.

4. Kukade, Jyoti, and Prashant Panse. "Designing a Deep Learning Model for Video Anomaly Detection-Based Surveillance." In *International Conference on ICT for Sustainable Development*, pp. 257-269. Singapore: Springer Nature Singapore, 2023.

5. Rane, Omkar, Trisha Shishodiya, Roshan Sawant, and Anand Godbole. "SignSpeak-Sign Language Interconverter Using CNN-based Approach." In *2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1186-1191. IEEE, 2023.

6. Kukade, Jyoti, Swapnil Soner, and Sagar Pandya. "Autonomous anomaly detection system for crime monitoring and alert generation." *Journal of Automation, Mobile Robotics and Intelligent Systems* 16, no. 1 (2022): 62-71.

7. Bharathi, C. Uma, G. Ragavi, and K. Karthika. "Signtalk: Sign language to text and speech conversion." In *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, pp. 1-4. IEEE, 2021.

8. Rajaganapathy, S., B. Aravind, B. Keerthana, and M. Sivagami. "Conversation of sign language to speech with human gestures." *Procedia Computer Science* 50 (2015): 10-15.

9. Shirbhate, Radha S., Vedant D. Shinde, Sanam A. Metkari, Pooja U. Borkar, and Mayuri A. Khandge. "Sign language recognition using machine learning algorithm." International Research Journal of Engineering and Technology (IRJET) 7, no. 03 (2020): 2122-2125.

10. Born, Nicholas. "Sign Language Glove." (2010).

11. Cui, Runpeng, Hu Liu, and Changshui Zhang. "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7361-7369. 2017.

12. Grobel, Kirsti, and Marcell Assan. "Isolated sign language recognition using hidden Markov models." In *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, vol. 1, pp. 162-167. IEEE, 1997.

13. Bheda, Vivek, and Dianna Radpour. "Using deep convolutional networks for gesture recognition in american sign language." *arXiv preprint arXiv:1710.06836* (2017).

14. Yang, Jie, and Yangsheng Xu. *Hidden markov model for gesture recognition*. Carnegie Mellon University, the

Robotics Institute, 1994.

15. Pigou, Lionel, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. "Sign language recognition using convolutional neural networks." In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*, pp. 572-578. Springer International Publishing, 2015.

16. Zaki, Mahmoud M., and Samir I. Shaheen. "Sign language recognition using a combination of new vision based features." *Pattern Recognition Letters* 32, no. 4 (2011): 572-577.

17. Mukai, Nobuhiko, Naoto Harada, and Youngha Chang. "Japanese fingerspelling recognition based on classification tree and machine learning." In 2017 *Nicograph International (NicoInt)*, pp. 19-24. IEEE, 2017.

18. Kang, Byeongkeun, Subarna Tripathi, and Truong Q. Nguyen. "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map." In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 136-140. IEEE, 2015.

19. Notash, Farnaz D., and Elahe Elhamkia. "Comparing loneliness, depression and stress in students with hearing-impaired and normal students studying in secondary schools of Tabriz." *International Journal of Humanities and Cultural Studies* 4, no. 1 (2016): 711-719.

20. Ghari, Zohreh. "The cognitive, psychological and cultural impact of communication barrier on deaf adults' content of speech in Iran." *Commun Disord Stud Hearing Aids* 4, no. 3 (2016): 2375-4427.