Sign language identification using deep learning

Vasudeva G[1*], Medha R[2], Nadeem Gulam[2], Neha Pandey[2], Rakshith Vk[2]
[1]*Assistant Professor, Department of Electronics and Communication Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India*
[2]*UG Student, Department of Electronics and Communication Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India*

Abstract - One of the hardest problems that people with disabilities face is speech and hearing impairment. The suggested system is a platform that enables communication between those who have speech impairments and the rest of the world. For those who are unable to talk, sign language serves as a way of communication because their language is difficult for others to understand. As a result, the community's minority of people with disabilities are unable to do even the most basic tasks. As a consequence, the proposed system converts spoken and written output from Sign Language (SL). Convolutional neural networks (CNN) are used in addition to this to extract effective hand features for recognising hand motions consistent with Sign Language. So that they may successfully communicate, persons with impairments can use this model to recognise their hand movements and translate them into text and voice.

Keywords- Convolutional Neural Networks, Signlanguage, speech disability, speech impairment.

1. INTRODUCTION

Recent statistics of the (WHO), published in October 2017, estimate that more than 5 million people in India suffer from communication disorders especially people with hearing impairment. Most people with speech disabilities use hand gestures which are not understood by the common man. Hence such a system which can convert hand gestures to text is most essential. The most natural and expressive form of communication for those who are deaf is sign language. Non-deaf individuals never attempt to learn sign language in order to communicate with those who are hearing impaired. Individuals who are deaf are divided as a result. The divide between hearing people and the deaf population can be closed, though, if we can train the computer to convert sign language to text format. American denote Language (ASL) uses a single hand gesture to denote each letter. The fact that ASL already has a fully functional database is its most important feature. American Sign Language is currently the focus of many academics and intellectuals. The American Sign Language's numerous alphabets can be recognised by our proposed programme. Two more space and back space hand gestures have also been included. It will let deaf people express themselves in phrases correctly. Using 26 hand movements from American Sign Language and two extra hand motions for the keys space and backspace during text generation, a real-time sign language recognition system is developed. A camera is used to capture the indicators in real time. To contrast the extracted traits, a classification algorithm is employed. To determine the sign recognition, the features are compared to the testing database. Eventually, the recognised gesture is converted into text and displayed on the monitor. With the use of our suggested method, deaf and dumb persons can converse with non-signers without the aid of an interpretation.

2. LITERATURE REVIEW

Bikash K Yadav et al., [1] It has been proposed that the system make an effort to interpret basic sign language signs into speech and text. ASL is a visual form of communication that analyses input in addition to signing using vision. Information transfer is greatly aided by the shape, posture, and motion of the hands as well as the expressions on the face and movements of the body. Everyone in the United States uses sign language, which is not like other languages. As there are many languages spoken throughout the rest of the world, it has its own signal six languages, and each location has its own unique linguistic quirks. When comparing ASL recognition to grammatical correctness, 90% of it is recognised properly. A 96-neuron fully connected layer is now fed the output of the first densely associated layer. The output of the feature-rich layer is passed into the model's final layer, which has as many neurons as classes the model is supposed to be categorising. Each layer, both convolutional and fully linked, was activated using the Rectified Linear Unit (ReLu) method.

K. Amrutha et al., [2] The various steps of an automated sign language recognition (SLR) system are examined in this study. It takes a lot of data collecting and the best technique to train a system to interpret and comprehend indicators. A basic SLR system is used to construct an IRM. The method is based on the detection and recognition of individual hand gestures using vision. With the help of four applicants, the ML-based SLR model was assessed in a controlled environment. For feature extraction, a convex hull was utilised, and for classification, a KNN. The model's accuracy rate is 65%.

Chandra Gandhi et al., [3] creating a desktop software that uses a computer's camera to instantly translate a person's Indian Sign Language (ISL) signing gestures into relevant text and audio. The purpose of this project is to create text and audio versions of sign language motions. A convolutional neural network for gesture detection will be used to achieve this. In order to translate sign language to text and audio effectively, the convolution neural network will be trained to accurately identify the required traits of the sign language motions. The purpose of this study is to convert hand motions used in sign language to voice or text and vice versa using machine learning techniques. provides voice to user and transformed to text before displaying the appropriate sign as output, and vice versa.

Ankit Ojha et al., [4] Developing a desktop programme that captures a individual's present motions for (ASL) and translate them into voice and pictorial form. The resulted input gesture will be captured in a written text before being converted to audio. We are using a finger spelling sign language convertor in this method. We are using a CNN to recognise gestures. After appropriate training, a CNN is very reliable in addressing CV issues and has capability of recognising the necessary characteristics with accuracy which is very high.

Carl Jose et al., [5], this study is about sign language translation into text; this technology is for persons who have issues communicating, i.e., those who have speech impairments, hearing impairment, and are deaf. Sign language translation into text is performed via the use of sign language that can be translated into writing. DL is an ML and AI strategy that duplicates the individuals obtain information; deep learning algorithms' complexity and abstraction are heaped in a hierarchy. The researchers collected and acquired the data of their system utilising different approaches of translating sign language into text using (CNN), (CTC), and (DBN). A thorough literature review is performed in this work to analyse and analyse existing findings on sign language translation using deep learning techniques. A recommended model for translating sign language into text will be produced based on this data.

Tiago Oliveira et al., [6] Hearing impairment affects 15-26% of the world's population. The I-Assisted Communication for Education programme attempts to promote communication efficiency in an educational setting, the use of sign language enables for successful communication between deaf and hearing persons, making it a vital tool to assist communication. their respective country. Oliveira and colleagues expect that an automatic translator between text and sign language will assist deaf and hearing students and teachers interact. They say that the technique aims to improve on this by including an entire alphabet based only on hand postures and a wide vocabulary. The lexicon ranges from little under 500 words for the less complete languages to over 2000 words for the original language, Portuguese.

Mahesh Kumar N B et al.,[7] Sign Language is primarily used to communicate between deaf and dumb persons. However, if we can programme the computer to recover text from sign language, we can lessen a distance among regular people and the deaf population. ASL and BSL are some other sign languages. That is, there must be a relevant and succinct description of the photos. The major distinction is that the training accuracy is based on photographs from which the network has trained, allowing the network to overfit to noise in the training data. The validation accuracy is a real assessment of the network's performance because it is evaluated on a data set that is not present in the training datax. In the vision-based technique, the camera is applied to acquire picture motions.

Rajesh singh et al., [8] Sign Language is used to communicate between deaf and stupid persons. This article displays the identification of 28 hand gestures in American sign language, including backspace and whitespace motions. The proposed system contains five modules: setting up the model, training the model, and writing about graphs and labels. American Sign Language (ASL) includes 26 hand gestures to represent distinct alphabets. The most significant component is that ASL already has a completely operational database. They've also incorporated two new hand gestures for producing space and backspace. It will aid deaf folks in expressing themselves in acceptable language form. Sign language in every nation differs from one another. ISL communicates using hand, arm, face, and head/body signs. A simple one-handed gesture indicating "hello" or "goodbye" communicates the same message all over the world.

Akshatha Rani K et al.,[9] Hand gestures are employed in sign language recognition systems due of the method's emphasis on the hand. The dependable hand tracking technique offered by the cross-platform Media Pipe is implemented for this purpose and reliably recognises the hand before training and categorising the pictures using the ANN architecture.

As a result, in order to overcome this challenge, a sign language recognition system is a strong tool, and various research are being undertaken in this area that are very valuable to society. In this research, static sign language is applied, which means that the data is in the form of images, and a hand tracking technique is used to efficiently follow the hand. This approach recovers hand outlines from video frames by darkening the photos and retrieving the hand's white border. This border is used to differentiate the curves of the hands. Deep learning was proposed for static sign language recognition. They gathered characteristics from photographs, categorised them, and compared the outcomes of all CNN designs using different CNN architectures for training and testing. CNN and CV based hand gesture recognition has been prescribed. These shots are pre-processed, with thresholding and intensity rescaling operations conducted, and the system will analyse the images in which the hand is detected following the pre-processing. Dataset for Sign Language This is the first and most crucial stage in machine learning. Following these pre-processes, the system attempts to distinguish the hand in the generated shots and evaluates the images in which it can track the hand to construct a final dataset that is exploited for further processing. 3. The system that observes the hand and compares it to training photographs to anticipate the sign is letter 'A' with a probability % of prediction. The recommended methodology in this essay relied on hand tracking, which is a highly successful strategy.

Vatsal Patel et al., [10] More and more new ways are important for the production and enhancement of this sort of activity to analyse the accurate findings. The image is initially passed through filters in this manner, followed by running it through a classifier to which detects the hand motions. To produce the processed picture during feature extraction, apply the gaussian blur filter and threshold to the image captured using OpenCV in the first layer. The new picture is transmitted to the prediction model, which is subsequently exhibited and employed to generate the word. The creators of this book experimented with and examined the consequences of many approaches for converting sign language to spoken format. The researchers uncovered a number of devices that are supposed to replace sign language interpreters. Block The proposed method's diagram The gathered data is now ready to be employed to develop the model. Picture processing is used to simplify and characterise a picture by blurring the background, altering the image's colour, and establishing the adaptive threshold. After performing all of the aforementioned activities, we transfer the pre-processed input pictures to our model for training and testing. We have to make sure that the model does not employ a background subtraction approach.

Pigou et al., [11] presented a convolutional neural network-based method for identifying sign language. They made use of an American Sign Language (ASL) dataset made up of more than 1,400 videos, each of which featured a single sign. Multiple layers of convolution and pooling were utilised in their suggested CNN model, which was followed by fully linked layers for classification. On their test set, they were 87.4% accurate, outperforming the prior state-of-the-art procedures at the time. Convolutional layers and data augmentation were the most significant aspects impacting the performance of the authors' model, according to a sensitivity study they did to analyse the importance of various model components.

Zaki et al., [12] put forth a vision-based feature-based technique for sign language recognition. Their approach was based on manual detection, segmentation, and feature extraction employing edge detection, histograms of oriented gradients (HOG), and local binary patterns (LBP). Their accuracy was 98% utilising a dataset of 600 pictures from five separate signs. The authors acknowledged that the little dataset was a constraint of their research and that additional testing on bigger datasets would be advantageous.

Mukai et al., [13] suggested a classification tree and machine learning based sign language recognition system solely for Japanese fingerspelling. Their accuracy was 90.8% utilising a dataset of 60 gestures done by 10 participants. The authors discovered that their technique was efficient in detecting gestures made by diverse participants in varied lighting settings. Additionally, they proved that their machine learning technique outperformed a conventional rule-based approach by compared their system with it.

Kang et al., [14] proposed a CNN-based real-time sign language fingerspelling recognition system. They attained a 98.1% accuracy rate utilising a dataset of 20,000 depth maps of the fingerspelling alphabet. Multiple layers of convolution and pooling were utilised in their suggested CNN model, which was followed by fully linked layers for classification. The authors also illustrated how their CNN technique outperformed a typical handwritten feature-based approach by compared it with their system. They asserted that their approach was computationally effective and could be utilised for real-time applications on low-power devices.

3. METHODOLOGY

The system employs a vision-based approach, relying solely on hand gestures to represent all indications. This eliminates the need for artificial devices or gadgets to facilitate interaction. By utilizing hand gestures, the system can recognize and interpret natural human movements, enabling more intuitive and seamless interaction with the technology.

3.1 DATA SET GENERATION:

We looked for available datasets that matched the requirements of our research, but we were unable to find any that included unprocessed images. We only found datasets that were represented by RGB values. As a result, we made the decision to create our own dataset using the OpenCV programme. We collected around 200 photos of each ASL sign for testing and about 800 for training in order to create the dataset. A blue defined square was then used to create a Region of Interest inside each frame by utilising the camera on our computer, as seen in the picture below

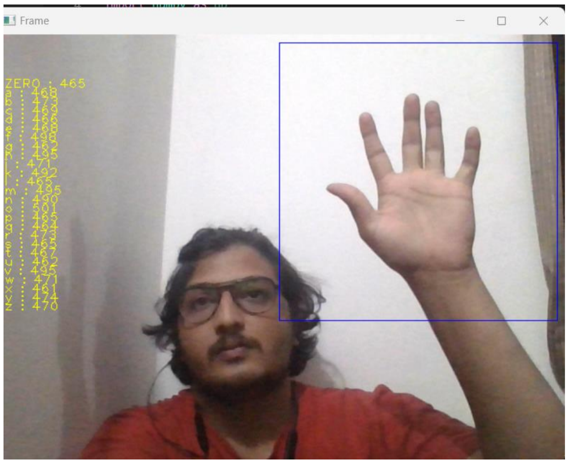

Figure 1: Defining Region of Interest (ROI)

After that, we use the Gaussian Blur Filter to extract different characteristics from our image. After using Gaussian Blur, the image appears as follows:

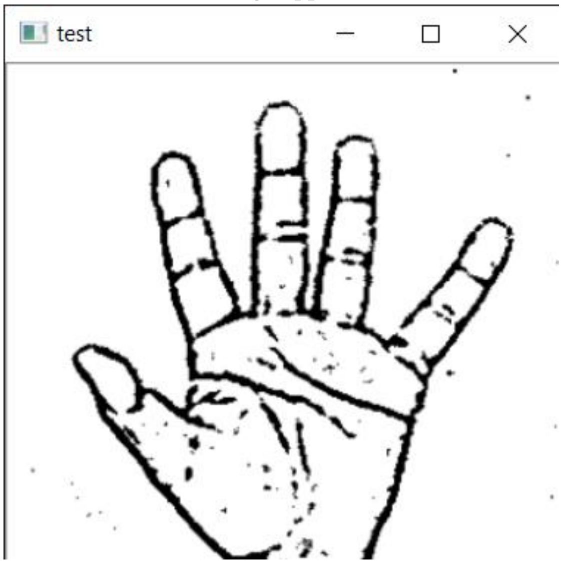

Figure 2: Image after applying gaussian blur.

3.2 GESTURE CLASSIFICATION:

Our solution involves using a dual-layered algorithm to anticipate the user's ultimate symbol

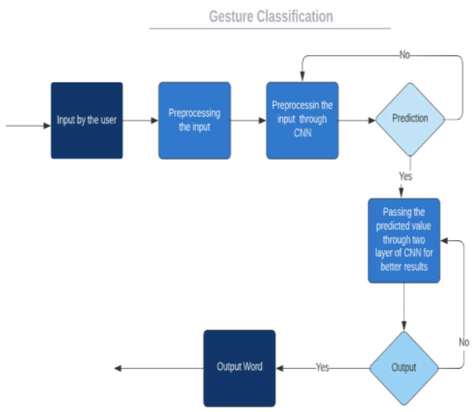

Figure 3: Gesture classification flow diagram

We use two CNN layers to carry out the classification.

CNN Layer 1:
1.To process the OpenCV frame, one can apply the Gaussian Blur filter and threshold after performing feature extraction. This will result in obtaining a processed image that can be used for further analysis or processing.
2. If a letter is detected in the input image for over fifty frames, the CNN model will identify and use it to form the corresponding word.
3. The space between each word is represented by the blank symbol.
CNN Layer 2:
1. We identify certain collections of symbols that, when discovered, have identical consequences.
2. We then use classifiers designed specifically to distinguish between those sets.
CNN Layer 1
1. First Convolution Layer: The convolutional neural network (CNN) used for image recognition begins with a 128x128 pixel image. The first convolutional layer applies 32 filters, each consisting of 3x3 pixels, resulting in a 126x126 pixel image for each filter.
2. First Pooling Layer: It uses a 2x2 max pooling technique to downsample the image, keeping the highest value in each 2x2 square. This reduces the image to 63x63 pixels..
3. Second Convolution Layer: The first pooling layer's output is subsequently used as input for the second convolutional layer. This layer also uses 32 filters with 3x3 pixels, resulting in a 60x60 pixel image for each filter.
4. Second Pooling Layer: The second pooling layer applies 2x2 max pooling to downsample the image further, resulting in a resolution of 30x30 pixels.
5. First Densely Connected Layer: After the second pooling layer, the output is flattened and passed through the first densely connected layer. This layer is composed of 128 neurons, which apply weighted linear transformations to the input. To avoid overfitting, a dropout layer with a value of 0.5 is incorporated, which randomly drops out half of the connections between the neurons during each training iteration. This technique ensures that the model does not become too reliant on specific connections, preventing it from fitting the training data too closely and overgeneralizing. By using a dropout layer, we improve the robustness of our model and enable it to better generalize to unseen data. Overall, this layer is a crucial component of the neural network architecture, as it allows for complex non-linear transformations of the input data, enabling the model to learn more intricate representations.
6. 2nd Densely Connected Layer: The output from the first densely connected layer is then fed into the second densely connected layer, which consists of 96 neurons.
7. Final layer: The last step in our neural network architecture involves passing the output of the second densely connected layer into the final layer. This layer is specifically designed to produce predictions for each of the classes being classified, including the blank space symbol. The number of neurons in this layer is equal to the number of distinct classes in the dataset. The purpose of this layer is to produce a probability distribution over the possible output classes based on the features learned by the previous layers. Once the probability distribution is obtained, we can choose the most probable class as the final prediction of our model. This final layer is a crucial component of the architecture, as it allows us to make accurate predictions on unseen data and evaluate the performance of our model.

• Activation Function:
Our activation goal in building a convolutional neural network involves using Rectified Linear Units (ReLUs) in each layer of fully connected and convolutional neurons. ReLU computes Max(x,0) for every input pixel, adding more nonlinearity to the formula and making it easier to memorize complex properties. This approach helps address the vanishing gradient problem and shortens computation time, making training faster.

• Pooling Layer: In order to decrease the computational burden and prevent overfitting, we utilize a pooling layer with a pool size of (2,2) and the Rectified Linear Unit (ReLU) activation function. This technique, also known as "Max pooling," downsamples the input image by selecting the maximum value within a window of size (2,2). By doing so, the number of features in the image is reduced, while still retaining the most relevant information. This process helps to simplify the representation of the image, making it easier for the network to learn and generalize to new data. Overall, the pooling layer is an important component of the neural network architecture, allowing for efficient and effective processing of the input data.
• Dropout Layers:

To address overfitting, we use a technique called "Dropout Layers." This method randomly sets a subset of activations in a layer to zero, effectively "dropping out" these activations. This ensures that the network can provide accurate classification or output even if certain activations are excluded.

• Optimizer:
In order to update our model based on the loss function output, we utilize the Adam optimizer. This optimizer is a combination of two stochastic gradient descent extensions, namely ADA GRAD and RMSProp. The goal of this optimizer is to enhance the performance of our model by adapting the learning rate of each parameter based on their historical gradients. This allows the model to learn faster and more efficiently. Additionally, the Adam optimizer also utilizes the concept of root mean square propagation to keep the update steps in a normalized range, preventing the optimization process from diverging. Overall, the Adam optimizer is a powerful tool for training deep learning models and has been shown to be effective in a wide range of applications.

CNN Layer 2:
Layer 2 of the Convolutional Neural Network (CNN) uses two layers of algorithms to analyze symbols that become increasingly similar to each other and predict the supplied symbol as accurately as possible. However, during the testing phase, we observed that some symbols were not displaying correctly and, in some cases, additional symbols were being displayed.
Specifically, the following symbol sets were problematic
1. R and U for D
2. D and R for U
3. T, D, K, and I for I
4. M and N for S
To address these issues, we developed three separate classifiers to categorize each set of symbols:
1. {D, R, U}
2. {T, K, D, I}
3. {S, M, N}
By implementing these classifiers, we were able to accurately categorize the problematic symbol sets and improve the accuracy of the CNN's recognition capabilities

3.3 IMPLEMENTATION OF SENTENCE FORMATION:

1. When the count of a detected letter reaches a certain threshold and no other letter is close (in our code, the number was set to 50 and the range of distance was set to 20), we display it and add it to the current string.
2. If not, in order to reduce the possibility that the incorrect letter would be anticipated, we remove the current dictionary, which keeps track of how often the current symbol has been found.
3. If the current buffer is empty and the number of blank (plain background) detections exceeds a predefined threshold, no gaps are detected.
4. The current sentence is added below the space that is written to indicate the word's ending in the alternative case.

3.4 AUTOCORRECT FEATURE:
Our system incorporates an autocorrect feature that suggests suitable replacement words for incorrect input words. To achieve this, we utilize the Hunspell_suggest Python tool. The tool provides a list of possible substitute words, and the user can select a phrase from a set of phrases that match the current word. This feature enhances the accuracy of word prediction and effectively reduces spelling errors.

Figure 4: basic flow diagram

3.5 TRAINING AND TESTING:
Our image processing pipeline begins by grayscale-converting the RGB input photographs and applying Gaussian blur to decrease noise. In order to

distinguish the hand from the backdrop, we next utilise adaptive thresholding and downscale the photos to 128 128. We train and test our model after preparing the input photographs. The chance that a picture belongs to each class is determined by the prediction layer. The output probabilities between 0 and 1 are normalised using the SoftMax function such that the total of all values in each class equals 1.

We utilised tagged data to train our neural networks in order to increase the precision of our model. As a performance indicator for classification, we employed cross-entropy. When the anticipated value deviates from the actual value, the cross-entropy function, which is continuous, has a positive value; otherwise, it has a zero value. We reduced cross-entropy to improve our model. This was done by modifying the weights of our neural networks at the network layer using TensorFlow's built-in cross-entropy calculation tool. Gradient Descent, notably the Adam Optimizer, the best gradient descent optimizer, was used to optimise the cross-entropy function as it was found.

4. EXPERIMENTAL RESULTS

In our work, layer 1 and layer 2 were combined to reach a remarkably high accuracy rate of 98.0% in identifying American Sign Language (ASL) motions. This accuracy rate performs better than the majority of earlier study works in this field. But even with just layer 1, we were able to reach a high accuracy rate of 95.8%. Our method uses a distinct strategy to identify and understand ASL movements, whereas other studies mostly concentrated on employing technologies like Kinect to recognise hand motions. The development of sign language recognition systems utilising various methods has been the subject of several research investigations. For instance, a system using Kinect and convolutional neural networks was constructed in one research [11] to recognise Flemish sign language with a 2.5% mistake rate. Another study [12] built a recognition model using a hidden Markov model classifier and achieved an error rate of 10.90% for a vocabulary of 30 items. A separate research investigation found that 86% of 41 static motions in Japanese sign language could be recognised accurately [13].

In recent research [14], Map was able to recognise new signers using depth sensors with an accuracy rate of 83.58% and 85.49% and 99.99% for signers who were visible. Convolutional neural networks (CNNs), a recognition technique, were also used in some of the experiments previously stated. It's important to note that our model is different from some of the other models mentioned above in that we don't use a backdrop removal technique. As a result, there can be some differences in the way background subtraction is used in our project.

In contrast to the majority of the research previously mentioned, which rely on specialised equipment like Kinect sensors, our main goal was to develop a project that could be carried out utilising easily accessible resources. Therefore, instead of using a Kinect sensor, which might be difficult to get and would be too expensive for many of our potential customers, we used a regular laptop webcam in our solution.

						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

Figure 5: CNN layer 1 confusion matrix

						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

Figure 6: CNN layer 2 confusion matrix

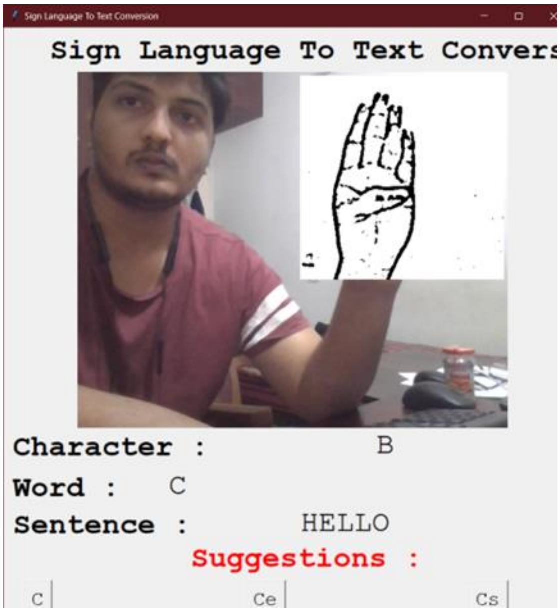

Figure 7: GUI with the results

CONCLUSION

In this paper, a real-time vision-based functional technique that would allow individuals with hearing and speech impairment to recognise and comprehend the American Sign Language (ASL) alphabet. We were able to attain an accuracy rate of 98.0% for our dataset using a number of computational procedures. Using two tiers of algorithms to enhance our predictions, our model was created to precisely anticipate and validate indicators that were similar to one another.The system can successfully recognise a variety of ASL symbols as long as the signs are presented correctly, in good illumination, and with no background noise.We aim to improve accuracy especially in the presence of complex backgrounds by employing a variety of background elimination strategies. We are also considering ways that will improve pre-processing in order to better anticipate gestures in conditions of low light. To make it easier for consumers to access this project, it might be improved by being developed as a web/mobile application. Additionally, the project as it stands only supports ASL; but, with the suitable data collection and training, it might be expanded to support additional native sign languages. However, sign languages are sometimes spoken in context, with each gesture representing an item or phrase. This project uses a finger spelling translator to translate sign language into speech. Therefore, further processing would be needed as well as natural language processing (NLP) to recognise this type of contextual signing.

REFERENCE

[1] Bikash K. Yadav, Dheeraj Jadhav, Hasan Bohra, Rahul Jain. " Sign Language to Text and Speech Conversion" Sign Language to Text and Speech Conversion. ISSN: 2454-132X Impact Factor: 6.078 (Volume 7, Issue 3 - V7I3-1560).

[2] K Amrutha P Prabu K Amrutha; P Prabu.” ML based sign recognition system” 2021 International Conference on Innovative Trends in Information Technology (ICITIIT).

[3] Mrs s Chandra Gandhi, Aakash raj r, Muhammed Shamil ml, Akhil s, Prabha Shankar. “Real time translation of sign language to speech and text” International Advanced Research Journal in Science, Engineering and Technology. Vol.8, Issue 4, April 2021.

[4] Ankit Ojha, Aayush Pandey, Shubham Maurya, Abhishek Thakur, Dr. Dayananda P “Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network.”IJERT NCAIT – 2020 (Volume 8 – Issue 15). 21-09-2020

[5] Carl Jose M. Guingab, Ron Andrew Relayo, Mark Joseph C. Sheng, John Ray D. Tamayo " Using Deep Learning in Sign Language Translation to Text." Proceedings of the International Conference on Industrial Engineering and Operations Management Istanbul, Turkey, March 7-10, 2022

[6] Tiago Oliveira; Paula Escudeiro; Nuno Escudeiro; Emanuel Rocha; Fernando Maciel Barbosa” Automatic Sign Language Translation to Improve Communication” 2019 IEEE Global Engineering Education Conference (EDUCON)

[7] Mahesh Kumar N B “conversion of sign language into Text” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7154-7161

[8] Rajesh Singh, Satyam Shekhar, Shashank Shaurya, Shivang Kumar, Dr. Rekha. K.S “Sign Language to Text” ISSN 2321 3361 © 2020 IJESC Volume 10 Issue No.6

[9] Akshatha Rani K, Dr. N Manjanaik “Sign Language to Text-Speech Translator Using Machine Learning” International Journal of Emerging Trends in Engineering Research volume 9, No 7, July 2021

[10] Vatsal Patel, Maahi Patel “Sign to Text Conversion- Helping Aid” International Journal of

Scientific Research in CS and IT, Engineering Volume 7, Issue 05 Oct 2021.
[11] 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
[12] 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).
[13] 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
[14] 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)