International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

Sign Language to Text Conversion for DUMB and DEAF

Dr. Rajiv Choudhary [1], Abhishek Goswami [2], Chirag Gupta [3], Pooja Jangir [4]
Professor [1], UG student[2,3,4]
Mechanical Department, Delhi Technological University, Main Bawana Road Rohini , Delhi, India

ARTICLEINFO

Article History:

Accepted: 01 May 2023
Published: 23 May 2023

Publication Issue
Volume 9, Issue 3
May-June-2023

Page Number
223-230

ABSTRACT

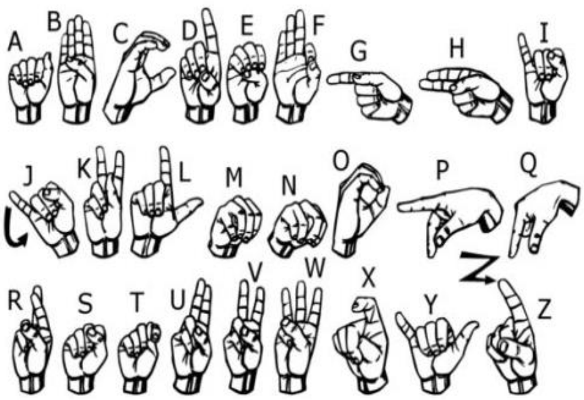
One of the oldest and most prevalent types of language for correspondence is signing, but since most people are not familiar with gesture communication and interpreters are extremely hard to come by, we have come up with a consistent method using brain networks for fingerspelling-based American Gesture-based communication. In our approach, the hand is first processed by a filter, and once the filter has been applied, the hand is processed by a classifier that determines the class of hand movements. The 26 letters in the letter set are 95.7% precisely placed using our method.
Keywords : CNN, text conversion, sign identification.

I. INTRODUCTION

American Given that the major communication-related disability that D&M people experience and their inability to engage in dialect conversation, gesture-based communication is by far the most way for them to communicate effectively. The act of conveying ideas and information through a variety of channels, including words, actions, gestures, and images, is known as communication. Dumb and deaf (D&M) individuals communicate with one another by using a variety of hand gestures. The nonverbal communications that are exchanged are motions, and eyesight is used to interpret these signals. The deaf and dumb use sign language as their primary form of nonverbal communication. A major portion of gesture-based communication involves visual language:

Fingerspelling	Word level sign vocabulary	Non-manual features
Used to spell words letter by letter .	Used for the majority of communication.	Facial expressions and tongue, mouth and body position.

Our project's main objective is to create a model that can identify hand motions based on fingerspelling and combine them to create whole words. The motions we wish to teach are shown in the figure below.



II. RELATED WORKS

In recent years, there has been a lot of interest in the identification of hand movements. We discovered through a literature review that the fundamental stages of hand gesture recognition are:-

- Data acquisition
- Data preprocessing
- Feature extraction
- Gesture classification

Data collection: To learn more about the hand gesture, you can use the methods below:

1. **Use of sensory devices**

   Electromechanical devices are used to carefully configure and position the hands using sensory aids. Information can be extracted using a variety of glove-based methods, but they are expensive and difficult to use.

2. **Vision-based approach**

   A computer camera is the information tool for noticing the data of hands or fingers in vision-based techniques. The Vision Based approaches allow for a seamless interface between people and computers using just a camera and no other hardware. By depicting fake vision frameworks these frameworks, which can be used in hardware or software, will frequently improve natural vision. The major issue with vision-based hand acknowledgment is that it should adapt to the enormous assortment of hand developments, complexions, camera points, scales, and paces, as well as the assortment of hand looks.

3. **Information preprocessing and Component extraction for vision based approach:**

- Removal of the background and threshold-based color detection are combined in [1]'s approach to hand detection. Due to their comparable skin tones, the AdaBoost face detector allows us to discriminate between hands and faces.
- We may also extract the required picture for training using a filter known as "Gaussian blur." The channel is shown in [3] and may be easily deployed using open PC vision, often known as OpenCV.
- As stated in [4], instrumented gloves can be utilized to obtain the required picture for training. When compared to applying filters to video extraction data, this can provide us information that is more precise and succinct while also speeding up pretreatment computations.
- We attempted to divide a photo by hand using a number of division techniques, but since the test paper said that skin tone and tone are highly dependent on lighting, the results of the division we attempted to carry out weren't very good. For our project, we also need to train on a ton of symbols, many of which resemble one another, such as the gesture for the number "2" and the sign "V" In order to avoid having to segment the hand depending on its backdrop eight times, we chose to maintain it a constant single hue instead of segmenting it eight times from a random background.

4. **Gesture classification :**

- In [1], Hidden Markov Models (HMM) are employed to categorize the gestures. The dynamic features of gestures are the main emphasis of this model. Gestures are extracted by tracking the skin-color blobs that correspond to the hand in a series of video shots into a body-facial space centered on the user's face. Perception of the deictic and representational motion classes is the goal. Using a short look-into-ordering table, the image is sorted. After filtering, pixels with skin tones are collected into blobs. To determine homogenous areas, masses are quantifiable objects in the light of the skin-color variety pixels' x, y, and Y, U, and V colorimetry.

- • To rapidly and accurately identify static hand motions, [2] use the Naive Bayes Classifier. Its foundation is the classification of different motions using geometric invariants derived from visually segmented data. Consequently, unlike many other methods of recognition, this one does not consider skin tone. Each video casing's signals are divided by a static foundation. Divide, labeling, and extracting geometric invariants from the intended objects are the first steps. After that, a K nearest neighbor computation aided by distance weighting calculation (KNNDW) is used to characterize the signals, providing a private-weighted Gullible Bayes classifier with useful data.
- In their paper named "Human Hand Motion Acknowledgment Utilizing a Convolution Brain Organization," Hsien-I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen, alumni of the Foundation of Robotization Innovation Public Taipei College of Innovation Taipei, Taiwan, guarantee that they construct a skin model to separate the hand from a picture and afterward apply a twofold limit to the whole picture. They position the edge image so that it is focused on the center core after obtaining it. They feed this image into a convolutional brain network model to prepare and foresee the outcomes. They have developed their model using more than seven hand gestures, and when they use it, they produce signals with approximately 95% accuracy

III. METHODOLOGY

The system is a vision based approach. All the signs are represented with bare hands and so it eliminates the problem of using any artificial devices for interaction.

Data Set Generation

Due to the fact that they were only rough images, we were unable to locate suitable pre-made datasets for the project. The just datasets that we could find were RGB values. We decided to create our own information collection as a result. The following are the procedures we used to create our informative index. Using the OpenCV (Open Computer Vision) package, we produced our dataset. Initially, we utilized ASL to capture 200 photos for testing and about 800 images for training for each sign. We start by capturing each edge displayed by our machine's camera. In each frame, an area of interest (ROI) is designated by a blue enclosing square, as seen in the.As can be seen below, we convert the entire image into a grayscale image by extracting our ROI, which is RGB.

We next use the gaussian blur filter on our image to extract different visual attributes. Below is what the image looks like after applying gaussian haze.

GESTURE CLASSIFICATION

The approach which we used for this project is:

Our method employs two levels of algorithm to forecast the user's final symbol.

Algorithm Layer 1:

1. To obtain the handled image after incorporating extraction, apply the Gaussian haze channel and edge to the casing captured using OpenCV.
2. The CNN model receives this processed picture for prediction. A letter is printed and utilized to create the word if it appears in more than 50 frames.
3. The gap between the words is taken into account using the blank symbol.

Algorithm Layer 2:

1. We identify several sets of symbols that, when detected, provide outcomes that are comparable.
2. To distinguish between such sets, we next utilize classifiers created specifically for those sets.

Layer 1:

CNN Model:

1. First Convolution Layer: The first convolution layer has a resolution of 128 by 128 pixels for the input image. It is first managed utilizing 32 channel loads (3x3 pixels per) in the first convolutional layer. Because of this, each Channel load will produce an image with 126 x 126 pixels.

2. First Layer of Pooling: Using maximum pooling of 2x2, the photographs are down-examined. For instance, we maintain the photo with the most noteworthy worth in the 2x2 square of the presentation. As a result, our image now measures 63 by 63 pixels.

3. Second Convolution Layer: The 63 x 63 data of interest from the result of the first pooling layer are currently sent into the contribution of the second convolutional layer. The second convolutional layer (3x3 pixels per weight) processes it utilizing 32-channel loads. As a result, a picture with 60 x 60 pixels will be created.

4. Second Layer of Pooling: After being down sampled once more to a maximum pool size of 2 by 2, the photos' resolution is reduced to 30 by 30.

5. Thickly Associated Layer: At present, the result from the second convolutional layer is reshaped to a scope of 30x30x32 =28800 esteems and utilized as a commitment to a completely connected layer with 128 neurons. The layer uses an array of 28800 values as its input.

6. The second Associated Thick Layer: deals with this layer's results. To prevent overfitting, we are employing a dropout layer with a value of 0.5.

Second Thickly Associated Layer, or Layer 6 at the moment, the results of the first thickly associated layer are used to build a totally related layer with 96 neurons.

7. Final layer: The last densely connected layer receives the output of the second layer, who's number of neurons corresponds to the number of classes we are categorizing (alphabets plus the blank sign).

The function of Activation: In each layer (convolutional and totally coupled neurons), we have used ReLu (Amended Direct Unit). For each information pixel, ReLu determines max(x, 0). Now that the formula is nonlinear, it is easier to comprehend more intricate characteristics. It helps resolve the vanishing angle problem and expedite preparation by requiring less math and timing.

Layer for Pooling: We perform maximum pooling to the input picture with a pool size of and a relu activation function By decreasing how much limits, this brings down the expense of the calculation and diminishes overfitting.

Dropout Protection Layers: Overfitting is a problem when the loads of the organization are so redone to the preparation models that they perform seriously when given new examples subsequent to preparing. By setting them to nil, this layer "exits" an atypical arrangement of enactments in that layer. The network need to be able to offer the correct categorization or output for a specific example even if some activations are lost.

Optimizer: Due to the effect of the unlucky capability, we used Adam as a streamlining agent to update the model. Adam combines the benefits of the root mean square proliferation (RMS Prop) expansions of two stochastic slope plummet techniques and the versatile angle calculation (ADA Graduate).

Layer 2: We are using two layers of algorithms to validate and anticipate symbols that are increasingly similar to one another in order to come as near to correctly recognizing the displayed symbol as feasible.

During our testing, we found that the following symbols were giving other symbols in addition to not displaying correctly:

1. For D: R and U
2. For U: D and R
3. For I: T, D, K and I
4. For S: M and N

So to handle above cases we made three different classifiers for classifying these sets:

1. {D, R, U}
2. {T, K, D, I}
3. {S, M, N}

Finger spelling sentence formation Implementation:

1. Whenever a letter's count exceeds a certain number and no other letter is within a predetermined distance of it (in our code, we maintained the value at 50 and the distance threshold at 20), we display the letter and add it to the current string.
2. In any event, we remove the continuing word reference that contains the coordinates of the current image to reduce the possibility of an incorrect letter being expected.
3. On the off chance that the continued support is empty and whenever the number of clear (plain foundation) identified exceeds a certain value, no spaces are recognized.
4. In the alternative scenario, the current word is appended to the text below and a space is printed to signify when a word will stop.

Autocorrect Feature:

For each (incorrect) input word, we use the Python package Hunspell_suggest to suggest appropriate replacements, and the user may choose a word from a list of terms that match the current word to add it to the phrase. This helps to reduce spelling errors and makes it easier to anticipate difficult words.

Training and Testing:

We utilize Gaussian haze to grayscale our RGB input photographs to eliminate additional commotion. To separate our hands from the background, we resize our images to 128 x 128 and employ flexible limitations. The input photos are first preprocessed for training and testing, and then each of the aforementioned procedures is carried out on them. The forecast layer figures the probability that the image has a place with one of the classes. Normalization of the output ensures that the values fall within a range of 0 to 1, and that the sum of the values in each class equals 1. The softmax function was used to achieve this. The outcome of the expectation layer will initially be somewhat far from the true worth. We have organized the businesses using market data to better it. A presentation estimation used in the grouping is the cross-entropy. It is a continuous function that is positive when the labeled value is different from other values and exactly zero when it is the same. In order to get the best performance, we thus decreased the cross-entropy as much as we could. To do this, the weights of our neural networks are modified in the network layer. The cross entropy may be calculated using Tensor Flow's built-in functionality. After determining the cross entropy capacity, we simplified it by including Angle Drop. Actually, Adam booster is the best inclination-plummet booster.

Challenges Faced:

We encountered many challenges while working on the project. The dataset was the initial problem we ran into. It was much simpler to deal with square photos alone, thus we intended to apply CNN in Keras to work with square images and raw images. Since there was no existing dataset for it, we made the decision to generate our own. The second challenge was deciding which filter to apply to our photos in order to obtain the necessary characteristics so that we could input that image into the CNN model. We experimented with a range of filters, including Gaussian blur, fuzzy edge detection, and binary threshold. But eventually, we chose the Gaussian haze channel. Additional difficulties were provided by the accuracy of the model we trained in earlier rounds, which we ultimately enhanced by enlarging the input picture and by enhancing the dataset.

Proposed system:

In the suggested approach, we provide a Deep Learning technology that uses CNN and learned transfer models to automatically identify photos. This could be very helpful in situations like these. These methods make it simple for us to identify and comprehend the meaning of a sign. The proposed strategy's block graph is shown beneath.

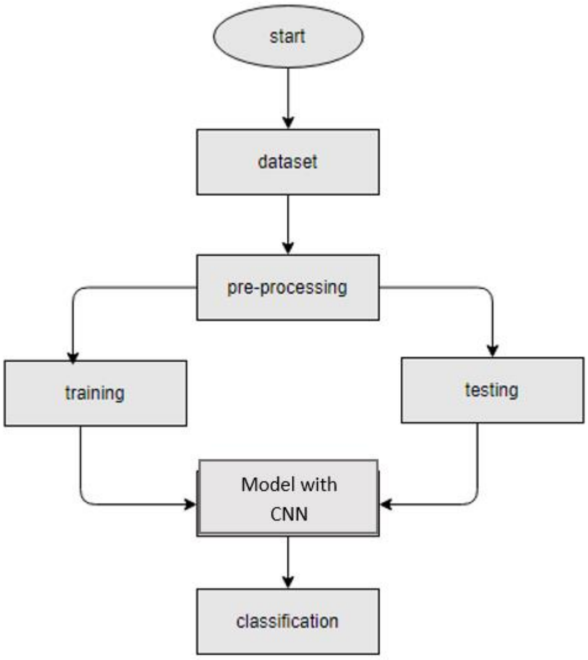
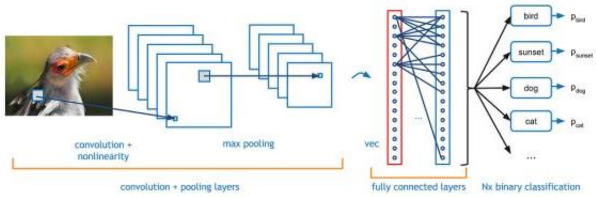


Figure 1: Block diagram of proposed method

IV. IMPLEMENTATION

The project has implemented by using below listed algorithm.

Neural Convolutional Network: The layers of CNN are made up of neurons arranged in three dimensions: in contrast to conventional neural networks' depth, height, and breadth. Instead of being fully connected to all of the neurons, the neurons in a layer will only be connected to a small portion of the layer that came before it, which is the size of a window. Since we would condense the entire image into a single vector of class scores at the conclusion of the CNN process, the final output layer would also contain aspects or the number of classes.



1. Convolution Layer: In the convolution layer, a modest window size, generally 5*5, is used to cover the depth of the input matrix. The layer contains window-sized learnable filters. Each iteration involves sliding the window by a certain amount, usually one, and computing the dot product of the input values and filter entries at a specific location. We will create a 2-D activation matrix with a response for each spatial point as we carry out this process. To put it another way, the organization will discover how to react when it sees a visual aspect like a direction or shade edge or smear.

2. Pooling Layer: The activation matrix is reduced using the pooling layer, which also reduces the total number of learnable boundaries. Pooling can be done in two ways:

a) The Largest Pool: In max pooling, we use a window size of [for a model window of size 2*2] and a maximum of four characteristics. Close this window and happen with the methodology until the enactment lattice is sliced down the middle.

b) Conventional Pooling: In normal pooling, we normal all qualities inside the window.

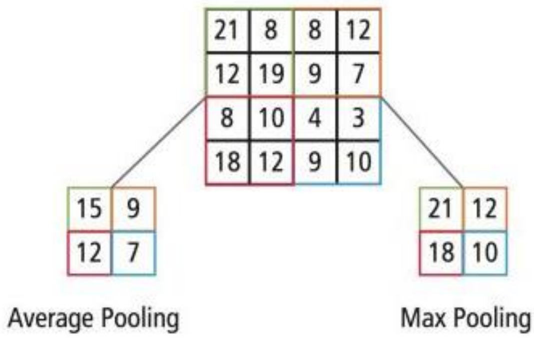


Fig. types of pooling

3. Layer with all connections: While neurons in a convolution layer are only linked to a limited area, they are connected to all inputs in a fully connected region.

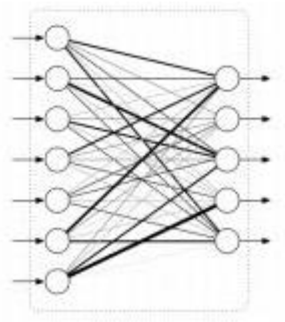


Fig. fully connected layer

4.Final Layer of Output: The last layer of neurons will forecast the likelihood that each picture will belong to a different class and have a count equal to the a total number of classes following receipt of the fully connected layer's results.

TensorFlow: The numerical computing software package TensorFlow is open-source. The actual computation takes place within a session after the computation graph's nodes have been specified. TensorFlow is frequently used in AI.

Keras: Keras is an undeniable level brain network library for Python that encases TensorFlow. It is used when a neural network needs to be built and evaluated quickly with as few lines of code as possible. This bundle executes layers, objectives, enactment capabilities, enhancers, and different devices to make managing text and pictures less complex.

OpenCV: OpenCV (Open Source PC Vision) is an open-source assortment of programming devices for ongoing PC vision. It is mostly used for face and object recognition research, video capture, and photo processing. Although it supports Python, MATLAB/OCTAVE, and Java as bindings, the majority of its interface is written in C++.

V. Results and Discussion

The accompanying pictures will serve as a visual representation of our project's progress.

When layer 1 and layer 2 are combined, we are able to build a model with an accuracy of 98.0%, which is greater than the bulk of the existing research publications on American Sign Language. Using only layer 1 of our algorithm, we were able to produce a model with an accuracy of 95.8%. The vast majority of research articles focus on hand detection using Kinect-style gadgets. In [7], they develop a system that can understand Flemish Using convolutional neural networks and a Kinect, we were able to teach sign language with a 2% error rate. 8] Forms an acknowledgment model with a 10.90% blunder rate utilizing a secret Markov model classifier and a dataset of 30 words. For 41 static Japanese communication through signing developments, they normal 86% exactness [9]. Map [10] was able to obtain an accuracy of 99.99 percent for signers who were observed and 83.58% and 85.49 percent for new signers by making use of depth sensors. In addition, their recognition system incorporated CNN. In contrast to a number of the models mentioned above, our model does not employ a background removal strategy. Thus, when we attempt to involve foundation deduction in our task, the precision might change. Then again, most of what preceded.

In some projects, Kinect devices are used, but our main goal was to create a solution that could be used with anything. The confusion matrices for our findings are listed below. A sensor like the Kinect is not only difficult to obtain but also prohibitively expensive for the majority of the audience. On the other hand, our model makes use of the built-in webcam on the laptop, which is a significant benefit.

						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

						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

VI. Conclusion

This paper has created a functional real-time vision-based American sign language recognition system for all alphabets for D&M people. We attained a final accuracy of 98.0% on our dataset.

We are able to increase the accuracy of our prediction by employing two layers of algorithms that check and forecast symbols that are increasingly similar to one another. We are able to recognize virtually all of them in this way, if the symbols are presented correctly, there is no background noise, and there is sufficient illumination.

VII.REFERENCES

[1]. T. Yang, Y. Xu, and "A., Hidden Markov Model for Gesture Recognition," Carnegie Mellon University, CMU-RI-TR-94 10. Pittsburgh (Pa.) in May 1994.

[2]. Mary Ellen Foster, Alois Knoll, Pujan Ziaie, Thomas Muller, and Mary Ellen of Robotics and Embedded Systems, Informatics VI, Boltzmann St. 3, Garching, Germany, DE-85748

[3]. https://docs.opencv.org/2.4/doc/instructional exercises/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

[4]. Mohammed Waleed Kalous, PowerGloves-based machine recognition of Auslan signs: toward the recognition of sign language with a large lexicon.

[5]. aeshpande3.github.io/A-Beginner%27s-Guide-To-U.

[6]. http://www-i6.informatik.rwth-aachen.de/~dreuw/database.php

[7]. Kinder Mans PJ, Dieleman S, Pigou L, Schrauwen B. (2015) Communication via gestures Acknowledgment Utilizing Convolutional Brain Organizations. In: Agapito L., Bronstein M., Rother C. (eds) PC Vision - ECCV 2014 Studios. ECCV 2014. Volume 8925 of Lecture Notes in Computer Science. Cham, Springer

[8]. Shaheen, Zaki, M.M., and S.I.: Communication through signing acknowledgment utilizing a mix of new vision based highlights. Letters on Pattern Recognition, 32(4), 572–577

[9]. N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Acknowledgment In view of Grouping Tree and AI," 2017 Nicograph Worldwide (NicoInt), Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9

[10]. Byeongkeun Kang , Subarna Tripathi , Truong Q. Nguyen "Constant gesture based communication fingerspelling acknowledgment utilizing convolutional brain networks from profundity map" 2015 third IAPR Asian Meeting on Example Acknowledgment (ACPR)

[11]. https://opencv.org/

[12]. https://en.wikipedia.org/wiki/TensorFlow

[13]. https://en.wikipedia.org/wiki/Convolutional_neural_network.

Cite this article as :

Dr. Rajiv Choudhary, Abhishek Goswami, Chirag Gupta, Pooja Jangir, "Sign Language to Text Conversion for DUMB and DEAF", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 3, pp.223-230, May-June-2023.