IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 2, March 2022

Indian Sign Language Recognition

Manasi Malge[1], Vidhi Deshmukh[2], Prof. Harshwardhan Kharpate[3]
Students, Department of Computer Engineering[1,2]
Assistant Professor, Department of Computer Engineering[3]
Cummins College of Engineering for Women, Nagpur, Maharashtra, India
manasi.malge@gmail.com[1], vidhideshmukh1705@gmail.com[2], harshwardhan.kharpate@cumminscollege.edu.in[3]

Abstract: *Sign language is one of the oldest and most natural forms of language for communication, but since most people do not know sign language and interpreters are very difficult to come by, we have come up with a real-time method using neural networks for fingerspelling-based Indian Sign Language. We collected a dataset of depth based segmented RGB image for classifying 36 different gestures (alphabets and numerals). The system takes in a hand gesture as input and returns the corresponding recognized character as output in real time on the monitor screen. For classification we used Convolutional Neural Network. Our method provides 95.7 % accuracy for the 36-hand gesture.*

Keywords: Sign Language, RGB, Gestures, Convolutional Neural Network.

I. Introduction

Sign language (SL) is a visual-gestural language used by deaf and hard-hearing people for communication purposes. Three dimensional spaces and the hand movements are used (and other parts of the body) to convey meanings. It has its own vocabulary and syntax which is purely different from spoken languages/written language. Spoken languages use the oratory faculties to produce sounds mapped against specific words and grammatical combinations to convey meaningful information. Then the oratory elements are received by the auditory faculties and processed accordingly. Sign language uses the visual faculties which is different from spoken language. Spoken language makes use of rules to produce comprehensive messages; similarly sign language is also governed by a complex grammar. A sign language recognition system consists of an easy, efficient and accurate mechanism to transform sign language into text or speech. The computerized digital image processing and a wide variety of classification methods used to recognize the alphabet flow and interpret sign language words and phrases. Sign language information can be conveyed using gestures of hands, position of head and body parts. Four essential components in a gesture recognition system are: gesture modelling, gesture analysis, gesture recognition and gesture-based application systems.

1.1 Background

Sign languages are developed primarily to aid deaf and dumb people. They use a concurrent and specific combination of hand movements, hand shapes, and orientation to convey particular information.

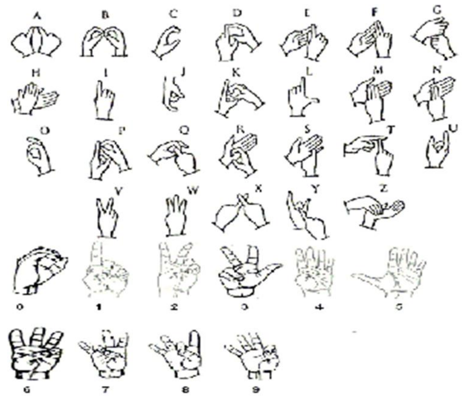


Fig 1: Indian Sign Language (A-Z and 0-9)

1.2 Motivation

The 2011 Indian census cites roughly 1.3 million people with "hearing impairment". In contrast to that numbers from India's National Association of the Deaf estimates that 18 million people –roughly 1 percent of the Indian population are deaf. These statistics formed the motivation for our project. As these speech impairments and deaf people need a proper channel to communicate with normal people there is a need for a system. Not all normal people can understand the sign language of impaired people. Our project hence is aimed at converting the sign language gestures into text that is readable for normal people.

1.3 Problem Statement

Speech impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need for a system that recognizes the different signs, gestures and conveys the information to normal people. It bridges the gap between physically challenged people and normal people. This project aims to predict the 'alphanumeric' gesture of the ISL system.

1.4 Image Processing

Image processing is a method to perform some operations on an image, to get an enhanced image, or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be an image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms a core research area within engineering and computer science disciplines too. Image processing includes the following three steps:

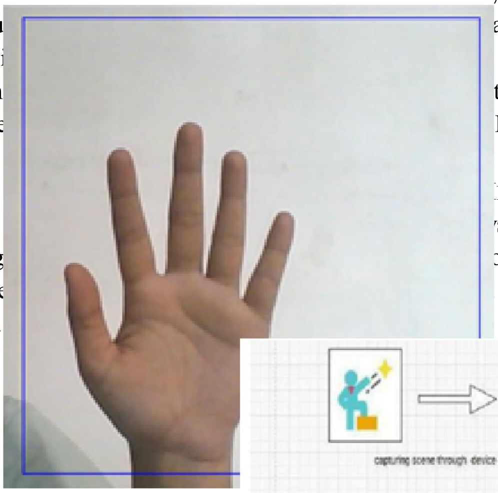
- Importing the image via image acquisition tools.
- Analysing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

1.5 Sign Language

It is a language that includes gestures made with the hands and other body parts, including facial expressions and postures of the body. It is used primarily by people who are deaf and dumb. There are many different sign languages as, British, Indian and American sign languages. British sign language (BSL) is not easily intelligible to users of American Sign Language (ASL) and vice versa. A functioning signing recognition system could provide a chance for the inattentive to communicate with non-signing people without the necessity for an interpreter. It might be wont to generate speech or text making the deaf more independent. Unfortunately, there has not been any system with these capabilities thus far. during this project, we aim to develop a system that may classify signing accurately. American Sign Language (ASL) is a complete, natural language that has the same linguistic properties as spoken languages, with grammar that differs from English. ASL is expressed by movements of the hands and face. It is the primary language of many North Americans who are deaf and hard of hearing and is used by many hearing people as well.

II. Literature Survey

Lionel Pigou and et al. [2] used two networks, consisting of three layers of convolution, each followed by max-pooling. One of the CNNs was trained to capture features from hand and the other from the upper body. The outputs of the two CNNs were concatenated and fed into a fully-connected layer. They utilized the dataset from CLAP14 [3] consisting of 20 Italian gestures by 27 subjects. They used both the depth and colour images. They achieved an accuracy of 91.7%on cross-validation containing different users with different backgrounds from the training set and testing accuracy of 95.68% but it contained users and backgrounds from the training set. Alina K. and et al. [4] used a multi-layered Random Forest model, for the dataset collected using the Open NI + NITE framework on Microsoft Kinect. They achieved an accuracy of 87% for subjects on whom the system was trained whereas 57% for a new subject. Lementec and et.al [5] used glove-based motion tracking sensor to identify gestures. However, given the limitations of having such delicate sensors and gloves makes it unfeasible to use. Hussain and et.al [6] used VGG-16 architecture (CNN) to train and classify hand-gestures. But, in this case the hand gesture must be placed in front of a constant background and any deviation in background will result in incorrect classification. Yamashita and et.al [7] proposed a deep convolutional neural network model for classifying hand-gestures. But they were only able to classify 6 hand-gestures with around 88.78% accuracy. Pei Xu ar camera, and used background subtraction techniques whereby achievi 16 gestures. B. Liao and et.al [9], used depth sensing camera and Hough training. They achieved an accuracy of 99.4% in classifying 24 gesture L).



ETHODOLOGY

Our stem using convolution neural networks which recognize various hand g frames. Then the hand pixels are segmented and the image is obtaine Thus, our system is more robust in getting exact text labels of letters.

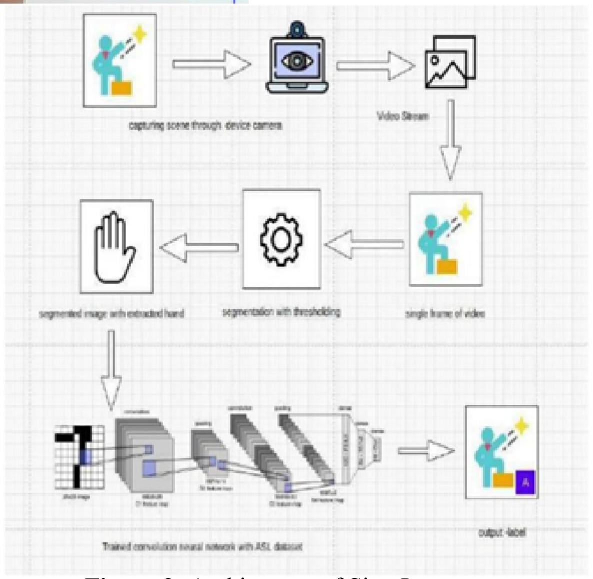

capturing scene through device camera
Video Stream
segmented image with extracted hand
segmentation with thresholding
single frame of video
A
output label
Trained convolution neural network with ASL dataset

Figure 2: Architecture of Sign Language

3.1 Training Model

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. To create such model, it is necessary to go through the following phases:

1. Model Construction
2. Model Training
3. Model Testing
4. Model Evaluation

3.2 Dataset Generation

For the project we tried to find already made datasets but we couldn't find dataset in the form of raw images that matched our requirements. All we could find were the datasets in the form of RGB values. Hence, we decided to create our own data set. Steps we followed to create our data set are as follows.

We used Open computer vision (OpenCV) library in order to produce our dataset. Firstly, we captured around 400 images of each of the symbol in ISL for training purposes and around 300 images per symbol for testing purpose. First, we capture each frame shown by the webcam of our machine. In each frame we define a region of interest (ROI) which is denoted by a blue bounded square as shown in the image below.

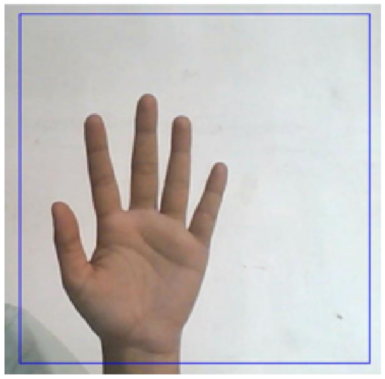


Fig 3: Sign Capture and Prediction Panel

From this whole image we extract our ROI which is RGB and convert it into grey scale Image as shown below.

Fig 4: After Grey Scale

Finally, we apply our gaussian blur filter to our image which helps us extracting various features of our image. The image after applying gaussian blur looks like below.

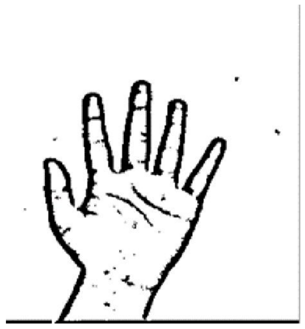


Fig 5: After Gaussian Blur Filter

3.3 Gesture Classification

Our approach uses two layers of algorithm to predict the final symbol of the user.

Algorithm Layer 1:

- Apply gaussian blur filter and threshold to the frame taken with OpenCV to get the processed image after feature extraction.
- This processed image is passed to the CNN model for prediction and if a letter is detected for more than 50 frames then the letter is printed and taken into consideration for forming the word.
- Space between the words are considered using the blank symbol.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 2, March 2022
Impact Factor: 6.252

Algorithm Layer 2:

- We detect various sets of symbols which show similar results on getting detected.
- We then classify between those sets using classifiers made for those sets only.

Layer 1:

CNN Model:

1. **1st Convolution Layer:** The input picture has resolution of 128x128 pixels. It is first processed in the first convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.
2. **1st Pooling Layer:** The pictures are down sampled using max pooling of 2x2 i.e. we keep the highest value in the 2x2 square of array. Therefore, our picture is down sampled to 63x63 pixels.
3. **2nd Convolution Layer:** Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer. It is processed in the second convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 60 x 60-pixel image.
4. **2nd Pooling Layer:** The resulting images are down sampled again using max pool of 2x2 and is reduced to 30 x 30 resolution of images.
5. **Densely Connected Layer:** Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to an array of 30x30x32 =28800 values. The input to this layer is an array of 28800 values. The output of these layer is fed to the 2nd Densely Connected Layer. We are using a dropout layer of value 0.5 to avoid overfitting.
6. **2nd Densely Connected Layer:** Now the output from the 1st Densely Connected Layer are used as an input to a fully connected layer with 96 neurons.
7. **Final layer:** The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

Activation Function

We have used ReLu (Rectified Linear Unit) in each of the layers (convolutional as well as fully connected neurons). ReLu calculates max(x,0) for each input pixel. This adds nonlinearity to the formula and helps to learn more complicated features. It helps in removing the vanishing gradient problem and speeding up the training by reducing the computation time.

Pooling Layer

We apply **Max** pooling to the input image with a pool size of (2, 2) with ReLu activation function. This reduces the number of parameters thus lessening the computation cost and reduces overfitting.

Dropout Layers

The problem of overfitting, where after training, the weights of the network are so tuned to the training examples they are given that the network doesn't perform well when given new examples. This layer "drops out" a random set of activations in that layer by setting them to zero. The network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out.

Optimizer

We have used Adam optimizer for updating the model in response to the output of the loss function. Adam combines the advantages of two extensions of two stochastic gradient descent algorithms namely adaptive gradient algorithm (ADA GRAD) and root mean square propagation (RMSProp).

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 6.252
Volume 2, Issue 2, March 2022

Layer 2:

We are using two layers of algorithms to verify and predict symbols which are more similar to each other so that we can get us close as we can get to detect the symbol shown. In our testing we found that following symbols were not showing properly and were giving other symbols also:

1. For D: P and K
2. For Q: Y
3. For M: N

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

1. {D, K, P}
2. {Q, Y}
3. {M, N}

3.4 Finger Spelling Sentence Formation

A. Implementation

1. Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold, we print the letter and add it to the current string (In our code we kept the value as 50 and difference threshold as 20).
2. Otherwise we clear the current dictionary which has the count of detections of present symbol to avoid the probability of a wrong letter getting predicted.
3. Whenever the count of a blank (plain background) detected exceeds a specific value and if the current buffer is empty no spaces are detected.
4. In other case it predicts the end of word by printing a space and the current gets appended to the sentence below.

B. Autocorrect Feature

A python library **Hunspell_suggest** is used to suggest correct alternatives for each (incorrect) input word and we display a set of words matching the current word in which the user can select a word to append it to the current sentence. This helps in reducing mistakes committed in spellings and assists in predicting complex words.

3.5 Training and Testing

We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. We apply adaptive threshold to extract our hand from the background and resize our images to 128 x 128.

We feed the input images after pre-processing to our model for training and testing after applying all the operations mentioned above. The prediction layer estimates how likely the image will fall under one of the classes. So, the output is normalized between 0 and 1 and such that the sum of each values in each class sums to 1. We have achieved this using SoftMax function.

At first the output of the prediction layer will be somewhat far from the actual value. To make it better we have trained the networks using labelled data. The cross-entropy is a performance measurement used in the classification. It is a continuous function which is positive at values which is not same as labelled value and is zero exactly when it is equal to the labelled value. Therefore, we optimized the cross-entropy by minimizing it as close to zero. To do this in our network layer we adjust the weights of our neural networks. TensorFlow has an inbuilt function to calculate the cross entropy. As we have found out the cross-entropy function, we have optimized it using Gradient Descent in fact with the best gradient descent optimizer is called Adam Optimizer.

IV. Design

4.1 Dataflow Diagram

The DFD is also known as a bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. It maps out the flow of information for any process or system, how data is processed in terms of inputs and

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 2, March 2022

Impact Factor: 6.252

outputs. It uses defined symbols like rectangles, circles, and arrows to show data inputs, outputs, storage points, and the routes between each destination. They can be used to analyse an existing system or model a new one.

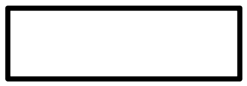
A DFD can often visually "say" things that would be hard to explain in words and they work for both technical and non-technical. There are four components in DFD:

1. External Entity
2. Process
3. Data Flow
4. Data Store

A. External Entity

It is an outside system that sends or receives data, communicating with the system. They are the sources and destinations of information entering and leaving the system. They might be an outside organization or person, a computer system, or a business system. They are known as terminators, sources, and sinks or actors. They are typically drawn on the edges of the diagram. These are sources and destinations of the system's input and output.

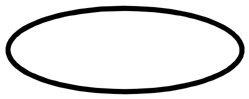
Representation:



B. Process

It is just like a function that changes the data, producing an output. It might perform computations for sort data based on logic or direct the data flow based on business rules.

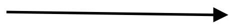
Representation:



C. Data Flow

A data flow represents a package of information flowing between two objects in the data-flow diagram, Data flows are used to model the flow of information into the system, out of the system, and between the elements within the system.

Representation:



D. Data Store

These are the files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label.

Representation:

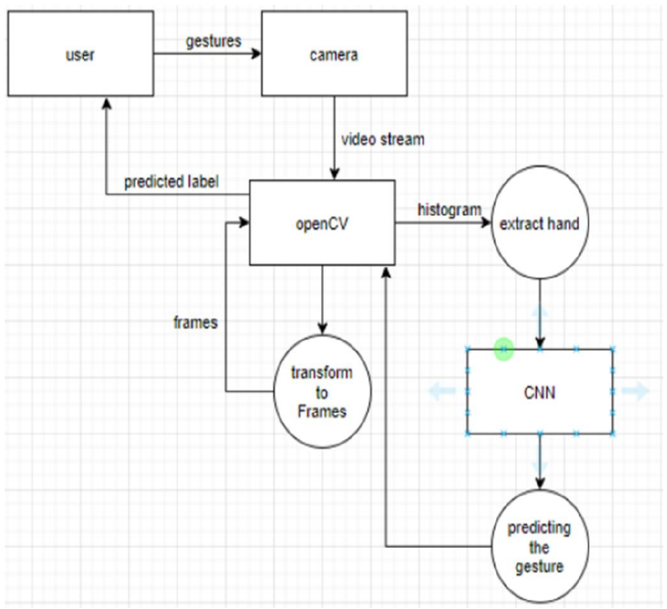
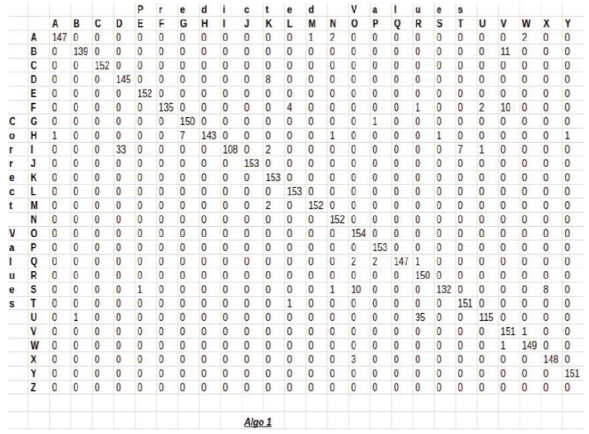


Fig 6: Dataflow Diagram for Sign Language Recognition

V. Result

We have achieved an accuracy of 95.8% in our model using only layer 1 of our algorithm, and using the combination of layer 1 and layer 2 we achieve an accuracy of 98.0%, which is a better accuracy then most of the current research papers on Indian Sign Language. Most of the research papers focus on using devices like Kinect for hand detection. In [5] they build a recognition system for Flemish sign language using convolutional neural networks and Kinect and achieve an error rate of 2.5%. In [6] a recognition model is built using hidden Markov model classifier and a vocabulary of 30 words and they achieve an error rate of 10.90%. In [7] they achieve an average accuracy of 86% for 41 static gestures in Japanese sign language. Using depth sensors map [8] achieved an accuracy of 99.99% for observed signers and 83.58% and 85.49% for new signers. They also used CNN for their recognition system. One thing should be noted that our model doesn't uses any background subtraction algorithm whiles some of the models present above do that. So, once we try to implement background subtraction in our project the accuracies may vary. On the other hand, most of the above projects use Kinect devices but our main aim was to create a project which can be used with readily available resources. A sensor like Kinect not only isn't readily available but also is expensive for most of audience to buy and our model uses a normal webcam of the laptop hence it is great plus point. Below is the confusion matrices for 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: Confusion Matrices of Algorithm 1

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 6.252

Volume 2, Issue 2, March 2022

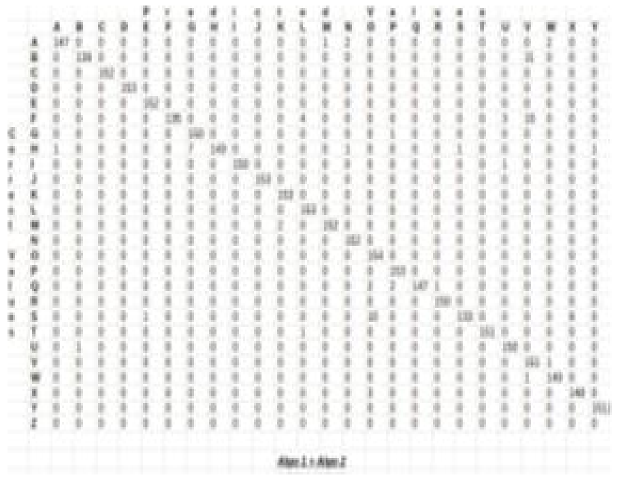


Fig 8: Confusion Matrices of Algorithm 1 + Algorithm 2

The accuracy rate of different classifiers obtained are shown below:

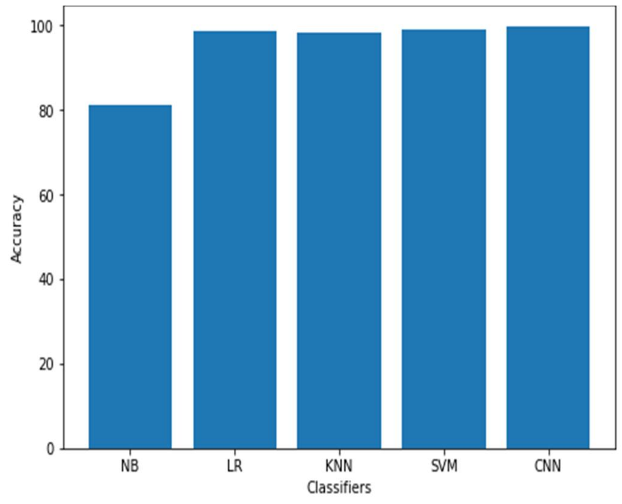


Fig 9: Accuracy Plot Graph

VI. Conclusion

Nowadays, applications need several kinds of images as sources of information for elucidation and analysis. Several features are to be extracted to perform various applications. When an image is transformed from one form to another such as digitizing, scanning, communicating, storing, etc. degradation occurs. Therefore, the output image has to undertake a process called image enhancement, which contains a group of methods that seek to develop the visual presence of an image. Image enhancement is fundamentally enlightening the interpretability or awareness of information in images for human listeners and providing better input for other automatic image processing systems. The image then undergoes feature extraction using various methods to make the image more readable by the computer. A sign language recognition system is a powerful tool to prepare expert knowledge, edge detects and the combination of inaccurate information from different sources. The intend of convolution neural network is to get the appropriate classification.

In this report, a functional real time vision based Indian Sign Language recognition for D&M people have been developed for ISL Alphabets. We achieved final accuracy of 98.0% on our dataset. We are able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other.

This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

VII. Future Scope

The proposed sign language recognition system used to recognize sign language letters can be further extended to recognize gestures facial expressions. Instead of displaying letter labels, it will be more appropriate to display sentences as a more appropriate translation of language. This also increases readability. The scope of different sign languages can be increased. More training data can be added to detect the letter with more accuracy. This project can further be extended to convert the signs to speech. We are planning to achieve higher accuracy even in case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving the pre-processing to predict gestures in low light conditions with a higher accuracy.

References

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

[2]. Pujan Ziaie, Thomas Muller, Mary Ellen Foster, and Alois Knoll "A Naive Bayes Munich, Department of Informatics VI, Robotics and Embedded Systems, Boltzmannstr. 3, DE-85748 Garching, Germany.

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

[4]. Aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural Networks-Part-2/

[5]. 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

[6]. 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).

[7]. 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

[8]. 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)

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

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

[11]. https://ieeexplore.ieee.org/document/7507939

[12]. https://ieeexplore.ieee.org/document/7916786

[13]. https://www.sciencedirect.com/science/article/pii/S1877050917320720

[14]. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neuralnetwork-cnn-deep-learning-99760835f148

[15]. https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-ToUnderstanding-Convolutional-Neural-Networks/