

# Indian Sign Language Detection

**Yash Bobde**

20BCI0176

School of Computer Science  
and Engineering (SCOPE)

Vellore Institute of  
Technology

yashsantosh.bobde2020@vit  
student.ac.in

**Payal Maheshwari**

20BCE2759

School of Computer Science  
and Engineering (SCOPE)

Vellore Institute of  
Technology

payal.maheshwari2020@vit  
student.ac.in

**Devavrat Dubale**

20BCE0660

School of Computer Science  
and Engineering (SCOPE)

Vellore Institute of  
Technology

devavrat.kaustubh2020@vit  
student.ac.in

UNDER THE GUIDANCE OF

**Prof. Santhi K**

**Abstract:** The goal of this project was to build a neural network able to classify which number of the Indian Sign Language (ISL) dataset is being signed, given an image of a signing hand. This project is a first step towards building a possible Indian sign language translator, which can take communication in sign language and translate them into written language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day-to-day interactions. This project uses the Convolution Neural Network and Artificial Neural Network to extract useful features of the hand, while the sign language is being presented in front of the camera. This research includes a dataset created on our own by recreating the Indian Sign Language. The technique of Gaussian Blur is implemented and this paper ends up with a total accuracy of 96%. We plan to incorporate all alphabets and numbers into this project and that it should be able to detect the entire sentence. Similarly, we hope to work on generating more training data and images for improvement in the accuracy percentage under various conditions.

**Keywords:** artificial intelligence, sign language, neural network, CNN, ANN, communication, Indian Sign Language

## I. INTRODUCTION

Sign language is a way of communicating using hand gestures and movement, instead of spoken words. It is mainly used by people who have hearing impairments. The majority of the population in the world uses spoken words to communicate, so this might seclude the deaf people. In order to overcome this, many people try to learn sign language. Making a system to convert hand gestures to letters/words is a small contribution to the same. Here, people who are not able to learn sign language, due to any reason, can still mingle with the people with hearing impairments. They can indulge in a basic conversation with the help of this system where their hand gestures will be converted to letters and then collectively words. The objective of this project is:

- The major objective of our project is to facilitate people with hearing disabilities to communicate better.
- It will enable them to exercise their basic rights and minimize the hurdles they face in day-to-day life.

- Simple hand gestures will be used to decode and interpret the message being communicated.
- The message will be clearly displayed at the user-end.

## **II. LITERATURE SURVEY**

The method described in this work uses grid-based characteristics to recognise hand positions and motions from the Indian Sign Language (ISL) in real time. With the help of this technology, the communication gap between the hearing- and speech-impaired and the general public is meant to be closed. The current solutions are either not real-time or provide just a modest level of accuracy. The output from this system is satisfactory for both parameters. It can recognise 33 hand positions and a few ISL movements. A smartphone camera records sign language, and the frames are sent to a distant server for processing. It is user-friendly since no other gear is required, such as gloves or the Microsoft Kinect sensor. [1]

Recognizing sign language is crucial for easy and natural communication between the hearing majority and the deaf population. Convolutional neural networks (CNNs) from depth maps are used in the first, extremely effective phase of an automated fingerspelling recognition system. Compared to the prior literature, we take into consideration a comparatively higher number of classes in our study. In order to classify 31 alphabets and integers, CNNs are trained using a subset of depth data that has been gathered from various people. [2]

The most effective illustration of a gradient-based learning method is a multilayer neural network taught using the back-propagation algorithm. Gradient-based learning techniques may create a complex decision surface that can categorise high-dimensional patterns, such as handwritten characters, with little to no pre-processing when given the right network design. This study evaluates several handwritten digit recognition techniques on a benchmark assignment for handwritten character recognition. Convolutional neural networks are shown to perform better than any other methods since they are particularly created to handle the diversity of

2D forms. Field extraction, segmentation identification, and language modelling are only a few of the components that make up actual document recognition systems.[3]

Long-term study has led to a breakthrough in aiding deaf-mute people: sign language recognition. Every study, however, has its own constraints that prevent it from being used economically. Some studies have been proven to be effective in understanding sign language, but they cost a lot of money to market. Researchers are receiving increasing attention these days for creating commercially viable Sign Language Recognition. Research is conducted in a variety of methods. It begins with the data collection techniques. The cost of a suitable device necessitates a variety of data collecting techniques; nonetheless, a low-cost technique is required for the commercialization of the Sign Language Recognition System. [4]

The authors outline an approach that can translate Indian Sign Language (ISL) into regular text while still recognising it. Three stages make up the methodology: a training phase, a testing phase, and a recognition phase. To develop a new feature vector for sign recognition, combinational parameters of the Hu invariant moment and structural shape descriptors are created. To train and detect ISL, a multi-class Support Vector Machine (MSVM) is used. 720 photos are used in the dataset to validate the efficiency of the suggested strategy. The suggested system can successfully recognise hand gestures with a 96% recognition rate, according to experimental results. [5]

In this paper, the authors explore how hand gestures can be used as a means of controlling robots in domestic and commercial settings. Different machine learning techniques, such as neural networks, support vector machines, and Adaptive Boosting, form the basis of various hand gesture detection algorithms (AdaBoost). AdaBoost-based hand-pose detectors are among these methods; to increase their robustness, they are trained with a condensed version of the Haar-like feature set. In addition to being accurate and reliable for more than four hand gestures, the proposed method's fast real-time performance is due to its reliance on context-free grammar. As a

workaround, we've included the convex hull algorithm for detecting fingertip gestures in addition to the standard representation for gestures involving rectangles. [6]

In this article, we provide the partial contour matching algorithm for classifying gestures. Finding the deepest gesture in a tree so that none of its children are included in a sample silhouette is the goal of our classification. In communication, hand gestures are frequently employed. In sign languages, utilising is a key example. Many silhouettes of hands in motion are included with other silhouettes of hands in motion. For instance, the V sign gesture is a component of the high five gesture because the other three fingers can be extended to form high five gesture silhouettes from the V sign gesture silhouettes. [7]

The primary goal of this study is to fine-tune the cloud-based point LSTM algorithm for hand gesture classification. With minimal computational expense, categorization utilising cloud-based point LSTM may make use of both the previous and present point, and the technique can also be applied to other action recognition. [8]

The authors analyze electromyography (EMG), which analyses the electrical activity of the skeletal muscles in the upper limb, can be used to identify hand motions. However, due to between-subject variability in EMG readings, generalising muscle activity for a specific hand gesture is difficult. The time-domain EMG features are normalised to the area under the averaged root mean square curve in order to increase the accuracy of gesture detection without explicitly training the machine learning algorithm on the subject (AUC-RMS). [9]

Based on information from a specialised glove with 10 sensors, the authors of this research introduces a method for quickly and effectively recognising hand motions as body language. 22 hand gestures for body language were made by the experiment's 10 participants. 10 repetitions were used to do each of the 22 gestures. Three machine learning methods were developed based on classifiers (probabilistic neural network, support vector machine, and k-nearest neighbours

algorithm) that were trained and validated using a tenfold cross-validation technique. [10]

This paper proposes applying deep learning to the problem of hand gesture recognition for the entire 24 hand gestures obtained from the Thomas Moeslund's gesture recognition database. The authors of this paper consider the recognition of significantly differentiable hand gestures, and as a result frequently select a few gestures from the American Sign Language (ASL) for recognition. [11]

According to the specificities of skin colour for hand gestures, the authors of this study achieve hand gesture segmentation by developing a skin colour model and an AdaBoost classifier based on haar. They also denaturize hand gestures by removing one frame of video from the video stream for analysis. In this sense, the human hand is separated from the complex background, and the CamShift algorithm also makes it possible to track hand gestures in real-time. [12]

The authors of this study present the primary goal which is to fine-tune the cloud-based point LSTM algorithm for hand gesture classification. With minimal computational expense, categorization utilising cloud-based point LSTM may make use of both the previous and present point, and the technique can also be applied to other action recognition. [13]

The author suggest a method for recognising hand gestures in three dimensions that uses stereo camera input to identify appropriately executed gestures. Skin detection is the foundation of hand tracking. 95 percent of the test signs were successfully identified in a cross validation with 120 signs carried out by 70 different people at a false positive rate of 5 percent. [14]

The author provides an ideal approach, whose principal purpose is to perform the translation of 24 static sign language alphabets and numerals of American Sign Language into humanoid or machine decipherable English writing. The first phase involves the signed input gesture's pre-processing processes. The various region attributes of the pre-processed gesture images are computed in the following stage. The transcription of signed gesture into text has been

completed in the last phase, depending on the attributes calculated in the prior phase. [15]

This paper provides an integrated framework for sign language detection, animation, and language modelling as part of the Dicta-Sign project, which aims to build the technology necessary to make sign language-based contributions to the Web feasible. Greek, British, German, and French are the four European sign languages it is intended to support. Three prototype sign language applications—a sign language to sign language translator, a sign language to sign language dictionary, and a sign language-based Wiki—are anticipated as results. It is beyond the scope of a three-year project to fully catch up with mainstream language technology and to deliver end-user products because the field of sign language technology is still relatively immature. [16]

The authors introduce a new Web tool for translating text to VRML animation sequences for H-anim compatible avatars and associated Sign Language notation. The device can be used as a text-to-sign language translator as well as a dictionary for sign language. It is built on a SignWriting dictionary, which is a widely used writing method for Sign Language. More specifically, the dictionary entries are in SWML (SignWriting Markup Language), a recently created XML-based format for storing, indexing, and interpreting SignWriting notation. Each sign box (basic sign) is first transformed into a series of Body Animation Parameters (BAPs) of the MPEG-4 standard that match to the displayed motion for use in sign synthesis. These sequences are then used to animate H-anim compliant VRML avatars, duplicating the precise actions described in sign language notation. [17]

In this study, the authors describe the steps for sign language recognition are detailed. Examined are the data collection, pre-processing, transformation, feature extraction, classification, and results. Final thoughts on the paper: The primary emphasis of current systems is on static signs, manual signs, alphabets, and numerals. For all nations, regions, and languages, the standard dataset is not available. The necessity for a vast vocabulary bank stems from the present situation. The use of continuous or dynamic signs and

nonverbal forms of communication should be prioritised. Systems should be able to concurrently distinguish between the face, the right and left hands, and other body parts. Systems should carry out recognition tasks more quickly and conveniently. [18]

This study uses machine learning algorithms to classify Indian sign languages. This work provides a comprehensive dataset implementation for the recognition of Indian sign language. The steps of implementation—picture collection, image pre-processing, feature extraction using K-means clustering, (visual words collection), and classification—have been covered step by step. Real-time gesture detection has also advanced beyond the realm of static images. This study can be expanded to include basic ISL words and phrases, such as alphabets and numbers. [19]

This study's goal is to examine various ways to sign language identification and identify the most effective one. [20]

The authors discuss the Sign Language Recognition System that is employed by scientists. Data collection and categorization are the first two stages of the development of sign language recognition application systems. Data Gathering Cameras are the primary input method for sign language recognition (SLR). The SLR input data is an easily captureable gesture image by the camera. Some researchers still take their images with basic cameras. Neural networks are another form of processing. 10 employs a 3D Convolutional Neural Network, a type of neural network (CNN). The 2D CNN approach, which is utilised in voice and picture recognition, forms the foundation for the 3D CNN technique. The local receptive field kernel window is used to extract a layer from a feature map and implement the 2D CNN. [21]

The authors contend that sign perception research's lessons must be incorporated into ASLR if the goal is for an ASLR system to handle natural sign language. Nowadays, automated sign language recognition (ASLR) seldom ever makes use of perceptual expertise in sign language recognition. By indicating which components or phases of a sign are crucial to

understanding its meaning, such knowledge can help ASLR. The current generation of data-driven ASLR techniques includes drawbacks that can be intractable without the aid of expertise in the analysis of human sign language. Nearly all current approaches are data-driven, which makes it challenging for them to detect indicators that significantly diverge from the examples used to train the algorithm. The conclusions include the following: not all phases and components of a sign are equally informative; identifying the "proper" form for a sign is not an easy task; statistical ASLR methods do not always result in sign representations that match those of humans. [22]

The author of this study take into account a vision-based system that can instantly understand user motions and manage windows and objects in a graphical user interface. From each frame of the collected image sequence, a binary hand blob is initially extracted using a hand segmentation algorithm. The hand blobs are represented by Fourier descriptors, which are input to radial-basis function (RBF) networks for pose categorization. The gesture recognizer receives motion data as well as the pose likelihood vector from the output of the RBF network. Hidden Markov models (HMM) and recurrent neural networks (RNN) were used to study the performance of gesture recognition. [23]

The authors of this work provide a method for converting hand motions used in Indian Sign Language (ISL) into suitable text messages. The webcam is used in this work to record the hand motions that correspond to the ISL English alphabets. The alphabet is recognised by the neural network once the hand is segmented in the frames that were taken. The neural network receives information like the angle between fingers, the number of fingers that are fully open, totally closed, or partially closed, and the identification of each finger as input. The findings of an experiment on single-handed alphabets are compiled. [24]

The authors of this study take a hand gesture analysis approach with the goal of identifying a digit. An induction graph is used to combine a series of features that were extracted from a hand image as the basis for the analysis. The

placements of the fingers, the heights of the fingers, and the separation between each pair of fingers are the most significant features we extract from each image. Our method entails the following three steps: (i) Localizing the hand, (ii) separating the fingers, and (iii) identifying the traits and combining them to recognise the digits. Assuming that each input image only contains one hand against a black backdrop, we apply a classifier based on one skin tone to determine the skin pixels. Based on the features of the hand anatomy, we aim to remove every component of the hand save the fingers during the finger extraction step. The digits are recognised in the final stage, which is based on the histogram representation of the detected fingers and feature identification. [25]

### **III. De-Sign-Pher: Indian Sign language Detection.**

Indian Sign Language is still a relatively new and untouched field in regards to utilizing it in machine learning. Thus, this has been an inspiration for us to work in this field and use the knowledge of Artificial Intelligence, Machine Learning amalgamated with Deep Learning to help the community of Specially -abled people to make their interaction with the other people easy. Our motivation behind this project is:

- Just because a portion of the population cannot hear or speak, does not mean that their form of communication should not advance along with the rest of the world.
- With this being said, sign language is a two-way street. The deaf community needs to know it, along with everyone else around them.
- Our major aim behind De-Sign-Pher is to minimize, if not eradicate the communication gap between the general population and the section of the population who is hard of hearing.

### **IV. SOFTWARE REQUIREMENTS**

#### **Tensorflow:**

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and

developers easily build and deploy ML powered applications.

TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

#### **Keras:**

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

It is used in cases where we want to quickly build and test the neural network with minimal lines of code. It contains implementations of commonly used neural network elements like layers, objective, activation functions, optimizers, and tools to make working with images and text data easier.

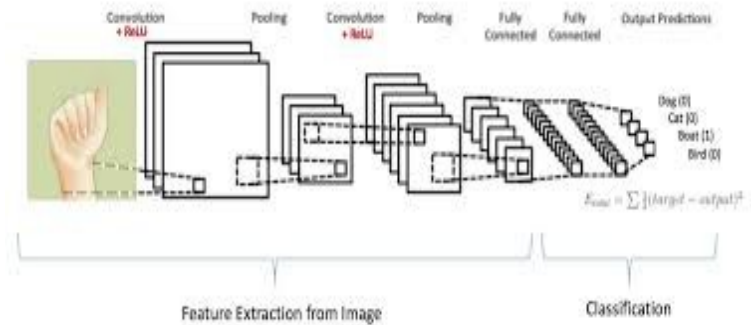
#### **OpenCV:**

OpenCV (*Open Source Computer Vision Library*) is a library of programming functions mainly aimed at real-time computer vision such as video analysis, CCTV footage analysis and image analysis. OpenCV is written by C++ and has more than 2,500 optimized algorithms.

## **V. METHODOLOGY**

### **Feature Extraction and Representation:**

Feature extraction involves reducing the number of resources required to describe a large set of data. The image is represented as a 3D Matrix having dimension as of height and width of the image and value of each pixel as depth ( 1 in case of Grayscale and 3 in case of RGB ).The values obtained from these pixels are used for extracting useful features using Convolutional Neural Network (CNN).



### **Artificial Neural Network(ANN)**

Artificial neural networks are based on the collection of connected nodes, and are designed to identify the patterns. They are part of deep learning, in which computer systems learn to recognize patterns and perform tasks, by analyzing training examples.

Essentially, Artificial Neural Network is a connection of neurons, replicating the structure of the human brain. Each connection of a neuron transfers information to another neuron. Inputs are fed into the first layer of neurons which processes it and transfers to another layer of neurons called hidden layers. After processing information through multiple layers of hidden layers, information is passed to the final output layer.

**They are capable of learning and they have to be trained. There are different learning strategies :**

1. Unsupervised Learning
2. Supervised Learning
3. Reinforcement Learning

### **Convolutional Neural Network(CNN)**

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vision that includes image and video recognition, along with recommender systems and natural language processing (NLP).

Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions:

width, height, depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner.

Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

### 1.ConvolutionLayer:

In convolution layer we take a small window size [typically of length 5\*5] that extends to the depth of the input matrix. The layer consists of learnable filters of window size. During every iteration we slide the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process we will create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position. That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color.

### 2.PoolingLayer:

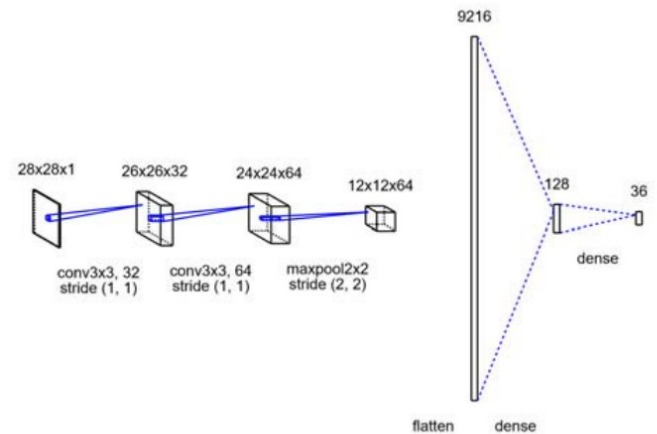
We use a pooling layer to decrease the size of the activation matrix and ultimately reduce the learnable parameters. There are two type of pooling:

a) Max Pooling : In max pooling we take a window size [for example window of size 2\*2], and only take the maximum of 4 values. Well lid this window and continue this process, so we'll finally get an activation matrix half of its original Size.

b) Average Pooling : In average pooling we take the average of all values in a window.

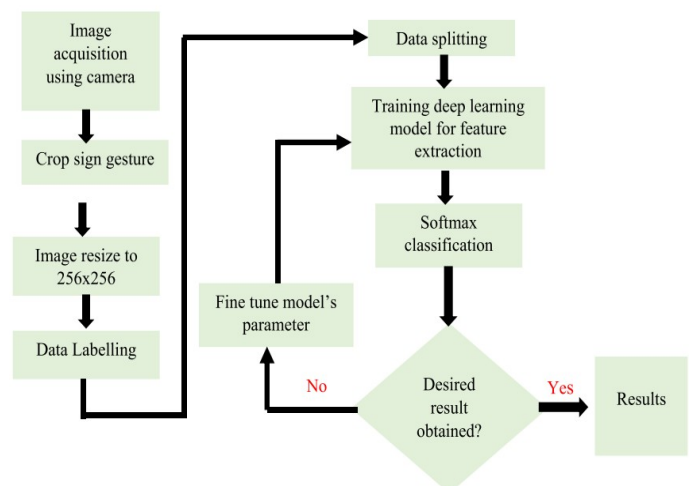
### 3.FullyConnectedLayer:

In the convolution layer, neurons are connected only to a local region, while in a fully connected region, we will connect all the inputs to neurons. After getting values from a fully connected layer, we will connect them to the final layer of neurons [having count equal to total number of classes], that will predict the probability of each image to be in different classes.



## DESIGN APPROACH & DETAILS

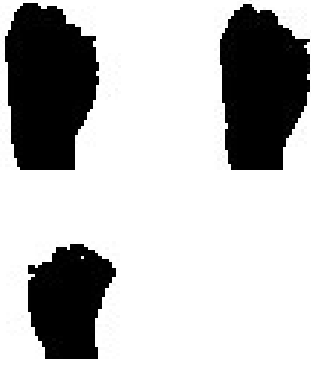
### Flowchart to represent methodology of the project



### DataSetGeneration

We created our own dataset as the required dataset was difficult to find. We used Open computer vision (OpenCV) library in order to produce our dataset. Firstly we captured around 600 images of each of the symbols(0-5) in Sign-Language for training purposes and around 60 images per symbol for testing purposes.

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.



The collection of images for the digit '0'.

Similarly the collection of sign language in respective directories created for each letter and digit will be stored for training and further testing.

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



Gaussian Blur

#### Algorithm Layer 1:

1. Apply gaussian blur filter and threshold to the frame taken with opencv to get the processed image after feature extraction.

2. This processed image is passed to the CNN model for prediction .

#### Algorithm Layer 2:

1. We detect various sets of symbols which show similar results on getting detected. 2. We then classify between those sets using classifiers made for those sets only.

#### CNNModel

1. 1st Convolution Layer :The input picture has a 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

128X128 pixel image, one for each Filter-weights.

2. 1st Pooling Layer : The pictures are downsampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of array. Therefore, our picture is downsampled 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 downsampled again using max pool of 2x2 and is reduced to 30 x 30 resolution of images.

5. 1st 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 .. The output of these layers is fed to the 2nd Densely Connected Layer.

6. 2nd Densely Connected Layer : Now the output from the 1st Densely Connected Layer is 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 (number).

Pooling Layer : We apply Max pooling to the input image with a pool size of (2, 2) with relu activation function. This reduces the amount of parameters thus lessening the computation cost and reduces overfitting.

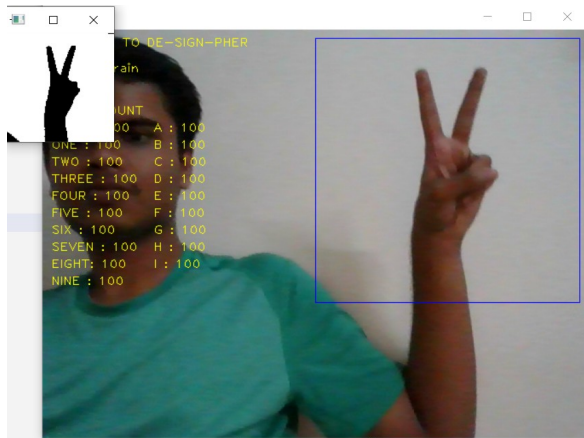
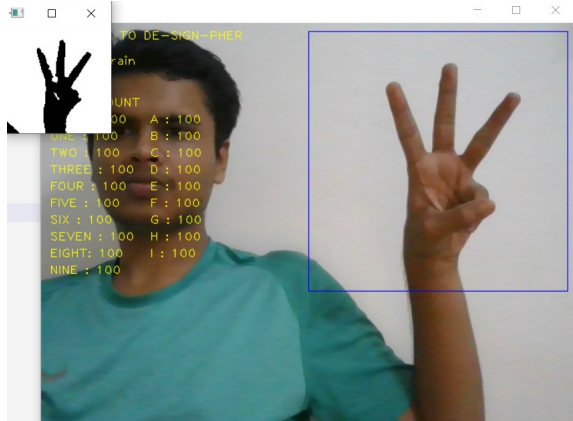
## VI. IMPLEMENTATION

### Testing and Training

We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. We apply an 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.



## VII. RESULT ANALYSIS

In this report, a functional real time vision based Sign Language recognition for specially abled people have been developed for ISL numbers.

We achieved final accuracy of **96.60%** on our data set. We have improved our prediction after implementing two layers of algorithms wherein we have verified and predicted symbols which are more similar to each other.

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

```
100/100 [=====] - 1s 10ms/step - loss: 0.1555 - accuracy: 0.9420
Epoch 12/15
100/100 [=====] - 1s 10ms/step - loss: 0.1617 - accuracy: 0.9400
Epoch 13/15
100/100 [=====] - 1s 10ms/step - loss: 0.1832 - accuracy: 0.9500
Epoch 14/15
100/100 [=====] - 1s 10ms/step - loss: 0.1187 - accuracy: 0.9660
Epoch 15/15
100/100 [=====] - 1s 10ms/step - loss: 0.1266 - accuracy: 0.9540
```

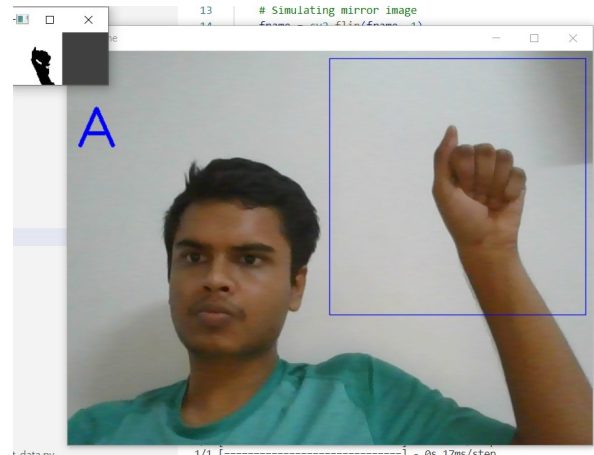
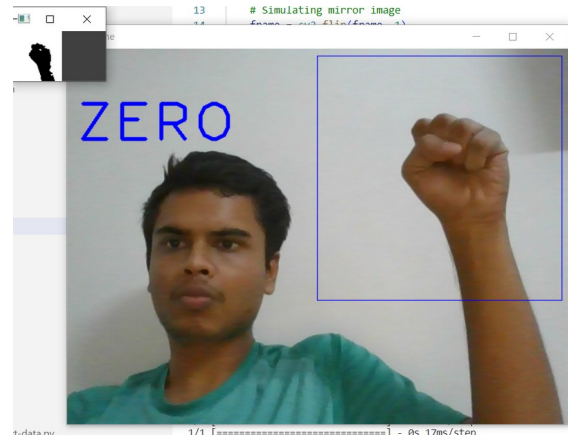


Table I. THE RESULTS OF TESTS (%) PERFORMED RANDOMLY 5 TIMES.

Test 1	Test 2	Test 3	Test 4	Test 5
99.44	99.44	100	99.44	98.89

Table II. THE RESULTS OF REAL-TIME SYSTEM TESTS PERFORMED FOR EACH CHARACTER.

Character	True Result	False Result	Accuracy(%)
0	10	-	100
1	9	1 (Z)	90
2	9	1 (V)	90
3	10	-	100
4	10	-	100
5	10	-	100
6	10	-	100
7	10	-	100
8	10	-	100
9	10	-	100
A	10	-	100
B	10	-	100
C	10	-	100
D	10	-	100
E	10	-	100
F	10	-	100
G	10	-	100
H	10	-	100
I	10	-	100
J	10	-	100

## VIII. CONCLUSION

This project was about hand gesture recognition which helps specially abled people to communicate better. We used CNN algorithm in making our ML model. We created our own dataset and captured images. Training and testing of the model was carried out. We performed the usability testing and our model was ready to be presented.

## IX. FUTURE WORK

De-Sign-pher is our small contribution to make sign language communication more widely accessible and make this translation a seamless experience for people who are short of hearing. Due to several limitations, we could implement ISL numbers from 0-10 and A-J in our project. Our dataset was limited so our prediction wasn't 100 % accurate. However, this concept can be incorporated into a fully functional application. We plan to incorporate all alphabets and numbers into this project. Similarly, we hope to work on generating more training data and images for improvement in the accuracy percentage.

We also hope to improve recognition in case of complex backgrounds by trying out various background subtraction algorithms. Further, we

want to improve the preprocessing to predict gestures in low light conditions with a higher accuracy for better results.

## X. REFERENCES

- [1] Kartik Shenoy, Tejas Dastane, Varun Rao, Devendra Vyavaharkar "Real-time Indian Sign Language (ISL) Recognition" 2018.
- [2] Satwik Ram Kodandaram, N Pavan Kumar, Sunil G L "Sign Language Recognition" 2021.
- [3] Y. Lecun; L. Bottou; Y. Bengio; [P. Haffner](#) "Gradient-based learning applied to document recognition" 1998.
- [4]Suharjito, Ricky Anderson, Fanny Wiryana, Meita Chandra Ariesta, [Gede Putra Kusuma](#) "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output" 2017.
- [6]Ruchi Manish Gurav; Premanand K. Kadbe "Real time finger tracking and contour detection for gesture recognition using OpenCV" 2015.
- [7] Ariyawat Chonbodeechalermroong, Thanarat H. Chalidabhongse "Dynamic contour matching for hand gesture recognition from monocular image" 2015.
- [8] Neha Baranwal, Varun Sharma "Literature on Hand GESTURE Recognition using Graph based methods" 2017.
- [9] Md FerdousWahid, Reza Tafreshi, Mubarak Saleh A MAI Sowaidi, Reza Langari "Subject-independent hand gesture recognition using normalization and machine learning algorithms" 2018.
- [10]Paweł Pławiak, Tomasz Sośnicki, Michał Niedźwiecki, Zbysław Tabor, Krzysztof Rzecki "Hand Body Language Gesture Recognition Based on Signals From Specialized Glove and Machine Learning Algorithms" 2016.
- [11] Oyebade K. Oyedotun, Adnan Khashman "Deep learning in vision-based static hand gesture recognition" 2017.
- [12] Jing-Hao Sun, Ting-Ting Ji, Shu-Bin Zhang, Jia-Kui Yang, Guang-Rong Ji "Research

on the Hand Gesture Recognition Based on Deep Learning” 2018.

[13] Gineke A ten Holt, Marcel J T Reinders, Emile Hendriks, Huib de Ridder “Influence of Handshape Information on Automatic Sign Language Recognition” 2009.

[14] Gineke Holt “Sign language detection using 3D visual cues” 2007.

[15] Shivashankara Ss “American Sign Language Recognition System: An Optimal Approach” 2018.

[16] Eleni Efthimiou, Stavroula-Evita Fotinea, Christian Vogler, Thomas Hanke, John Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos and Jérémie Segouat “Sign Language Recognition, Generation, and Modelling: A Research Effort with Applications in Deaf Communication” 2009.

[17] Maria Papadogiorgaki, Grammalidis Nikos, Lambros Makris, Nikos Sarris “VSigns – A Virtual Sign Synthesis Web Tool” 2004.

[18] Ashok Kumar Sahoo, Gouri Sankar Mishra, Kiran Kumar Ravulakollu “Sign language recognition: State of the art” 2014.

[19] Shravani K, Sree Lakshmi A, Sri GeethikaM, Dr.Sapna B Kulkarni “Indian Sign Language Character Recognition” 2020.

[20] Suharjito, Ricky Anderson, Fanny Wiryana, Gede Putra Kusuma “Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output” 2017.

[21] Bhumika Gupta, Pushkar Shukla, Ankush Mittal “K-nearest correlated neighbor classification for Indian sign language gesture recognition using feature fusion” 2016.

[22] Gineke A ten Holt, Jeroen Arendsen, Huib de Ridder, Andrea J van Doorn “Sign language perception research for improving automatic sign language recognition” 2009.

[23] Chan-Wah Ng “Real-time gesture recognition system and application” 2002.

[24] Padmavathi S, Saipreethy M S, Valliammai V “Indian Sign Language Character Recognition using Neural Networks” 2013.

[25] A Ben Hamadou, Y Ben Jemaa, W Mahdi, A Ben Jmaa “Hand Localization and Fingers Features Extraction: Application to Digit Recognition in Sign Language” 2009.