IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

Sign Language to Text Conversion

Aditya Kharat[1], Yash Patil[2], Omkar Jagtap[3], Rajashri Sonawale[4]
Students, Department of Computer Engineering[1,2,3,4]
Mahatma Gandhi Mission's College of Engineering and Technology, Kamothe, Maharashtra, India

Abstract: *Sign language is one of the oldest and most natural forms of communication, but most people do not know sign language and it is very difficult to find an interpreter, so we have developed a real-time method of using neural networks for American Sign Language. In our method, the hand first passes through the filter, and after the filter is applied, the hand passes through a classifier that predicts the class of hand gestures. Our method gives 95.7% accuracy for the 26 letters of the alphabet.*

Keywords: Sign Language, Text Conversion, Computer Vision, ASL, American Sign Language, Hand Gestures, ANN.

I. INTRODUCTION

The most common type of sign language is American sign language. Because D&M people's primary handicap is communication-related, and they are unable to communicate via spoken languages, sign language is their only means of communication. Communication is the process of exchanging ideas and information through a variety of means, including voice, signs, behavior, and pictures. People who are deaf or dumb (D&M) use their hands to communicate with others by making various gestures. Gestures are nonverbally communicated signals that are recognized by eyesight. Sign language is the nonverbal communication of the deaf and dumb. Sign language is a language that uses gestures instead of sounds to convey meaning by combining hand shapes, hands, arms, body orientations and movements, facial expressions, and lip patterns. Contrary to common belief, sign language is not international. These vary by region.

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.

Figure 1: Medium of communication for D&M people

Minimizing the verbal exchange gap among D&M and non-D&M people turns into a want to make certain effective conversation among all. Sign language translation is among one of the most growing lines of research and it enables the maximum natural manner of communication for those with hearing impairments. A hand gesture recognition system offers an opportunity for deaf people to talk with vocal humans without the need of an interpreter. The system is built for the automated conversion of American Sign Language into textual content and speech.

In our project we primarily focus on producing a model which can recognize Fingerspelling based hand gestures in order to form a complete word by combining each gesture. The gestures you want to train are shown in the image below.

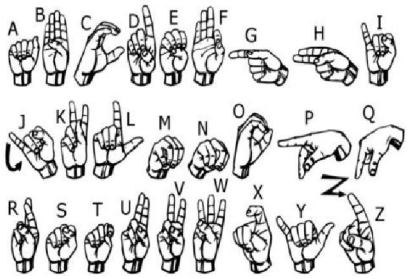

Figure 2: American Sign Language

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

1.1 Motivation

The interaction between a normal person and a D & M person is different from normal text, creating a language barrier as the structure of sign language. Therefore, they rely on visual-based communication for interaction.

With a common interface for converting sign language to text, non-D & M people can easily understand gestures. Therefore, a vision-based interface system has been studied that allows D & M people to enjoy communication without knowing each other's language. The purpose is to develop a user-friendly Human-Computer Interface (HCI) that allows computers to understand human sign language. There are various sign languages all over the world. That is, American Sign Language, French Sign Language, Japanese Sign Language, British Sign Language, Indian Sign Language, and work is done in other languages around the world.

II. LITERATURE SURVEY

In recent years, a great deal of research has been done on hand gesture recognition. Using a literature review, we found that the basic steps for hand gesture recognition are:-

- Data acquisition
- Data pre-processing
- Feature extraction
- Gesture classification

2.1 Data Acquisition

Various approaches to collecting data via hand gestures can be performed in the following ways:

A. Use of Sensory Devices

It uses electromechanical equipment to provide accurate hand composition and position. Information can be extracted using different glove-based approaches. But it's expensive and not user-friendly.

A. Vision Based Approach

In vision-based methods, the computer webcam is the input device for observing the information of hands and/or fingers. The vision-based method requires only one camera, which enables natural human-computer interaction without the use of additional devices, thereby reducing costs. The system tends to complement biological vision by describing artificial vision systems that are implemented in software and/or hardware mechanism. The main challenges of visual hand recognition are dealing with large variations in the appearance of the human hand due to numerous hand gestures, as well as the potential for different skin colors, perspectives, scales, and speed of camera capture. It extends to change scene.

2.2 Data Pre-Processing and 2.3 Feature Extraction for Vision-Based Approach

In [1], the hand detection approach combines threshold-based color detection with background subtraction. We can use AdaBoost face detector to differentiate between faces and hands as they both involve similar skin-color.

We can also extract necessary image which is to be trained by applying a filter called Gaussian Blur (also known as Gaussian smoothing). The filter can be easily applied using open computer vision (known as OpenCV) and is described in [3].

For extracting important photo that is to study we will use instrumented gloves as cited in [4]. This helps reduce computation time for Pre-Processing and gives us more concise and accurate data compared to applying filters on data received from video extraction.

We tried to manually segment the image using a color segmentation technique, but the results of the segmentation I tried were not very good because the skin color and tones are highly dependent on the lighting conditions. In addition, our project requires training many symbols that are similar to each other, such as the "V" symbol gesture and the number "2", so we decided to train a large number of symbols. Instead of segmenting the hands from a random background, there is no need to segment based on skin color to keep the background of the hands stable and monochromatic in a more accurate way to achieve the symbol. This will help us get better results.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

2.4 Gesture Classification

In [1], the hidden Markov model (HMM) is used to classify gestures. This model handles the dynamic aspects of the gesture. Gestures are extracted from a series of video images by tracing the skin-colored mass corresponding to the hand into the facial space of the body centered on the user's face.

The intention is to understand lessons of gestures: deictic and symbolic. The photo is filtered the use of a quick look–up indexing table. After filtering, pores and skin color pixels are collected into blobs. Blobs are statistical gadgets primarily based totally at the location (x, y) and the colorimetry (Y, U, V) of the pores and skin colour pixels that allows you to decide homogeneous areas.

In [2] Naïve Bayes Classifier is used that is an powerful and speedy approach for static hand gesture recognition. It is primarily based totally on classifying the specific gestures in step with geometric primarily based totally invariants which can be acquired from photo statistics after segmentation.

Therefore, in contrast to many other detection methods, this method is skin color independent. Gestures are extracted from each casing of the video with a static foundation. The first step is to segment and label the objects of interest and extract geometric invariants from them. The next step is to classify the gestures using the K-nearest neighbor algorithm supported by the Distance Weighting Algorithm (KNNDW), which provides the appropriate data for the locally weighted naive Bayes classifier.

According to the article on "Human Hand Gesture Recognition Using a Convolution Neural Network" by Hsien-I Lin, Ming-Hsiang Hsu, and Wei-Kai Chen (graduates of Institute of Automation Technology

National Taipei University of Technology Taipei, Taiwan), they built a skin model to extract hands from the image and added the method to apply binary thresholds to the entire image. After obtaining the threshold image they calibrate it about the principal axis in order to centre the image about the axis. They feed this image into a convolutional neural network model to train and predict the output. They have trained their model over 7 hand gestures and using this model they produced an accuracy of around 95% for those 7 gestures.

III. Keywords and Definitions

3.1 Feature Extraction and Representation

Representation of an image as a 3D matrix with dimensions such as the height, width, and value of each pixel as depth (1 for grayscale, 3 for RGB). In addition, these pixel values are used to extract useful features using CNN.

3.2 Artificial Neural Network (ANN)

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

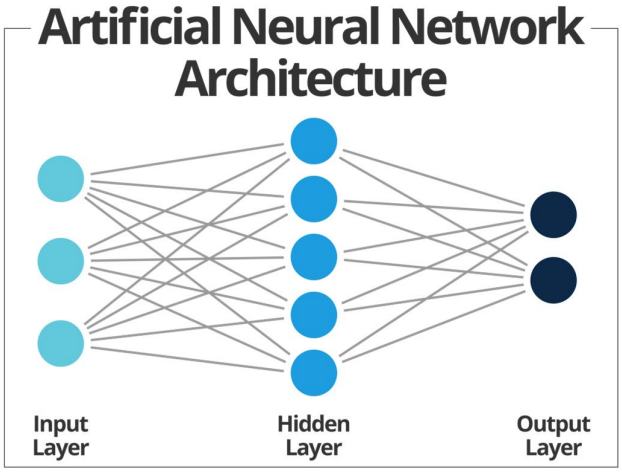

Figure -3 ANN Architecture

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

These are capable of learning and have to be trained. There are different learning strategies:

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

3.3 Convolution Neural Network (CNN)

Unlike a normal neural network, the neurons in the CNN layer are arranged in three dimensions: width, height, and 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. In addition, at the end of the CNN architecture, the final output layer has dimensions (number of classes) to reduce the big picture to a single vector of class scores.

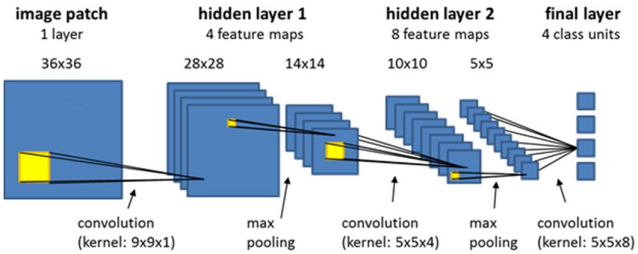

Figure 4: CNN Architecture

A. Convolution Layer

In convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. This layer consists of window-sized learnable filters. During each iteration, we moved the window incrementally [usually 1] to calculate the product of the filter entry and the input value at a particular 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 learns filters to activate when it detects certain visual features, such as edges in a particular direction or spots of a particular color.

B. Pooling Layer

Use the pooling layer to reduce the size of the activation matrix and ultimately the parameters that can be learned. There are two types of pooling:

a. Max Pooling

Max-pooling gets the window size (for example, a window of size 2 * 2) and only gets up to 4 values. Well lid this window and keep this process, so properly eventually get an activation matrix 1/2 of its authentic size.

b. Average Pooling

In average pooling, we take advantage of all values in a window.

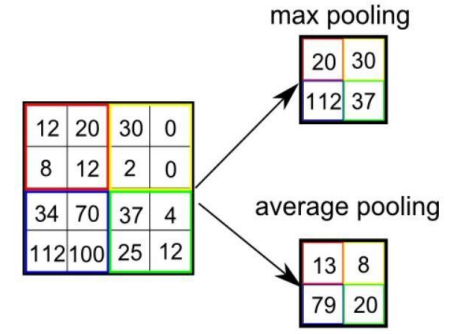

Figure -5 Pooling Layers

DOI: 10.48175/568

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

C. Fully Connected Layer

In convolution layer, neurons are connected only to a local region, while in a fully connected region, we will connect all the inputs to neurons.

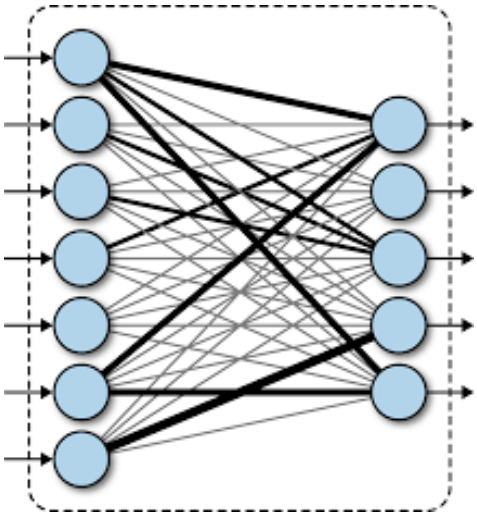

Figure 5: Connections between neurons

D. Final Output Layer

After getting values from fully connected layer, it is connected to the final layer of neurons [having count equal to total number of classes], that will predict the probability of each and every image to be in different classes.

3.4 TensorFlow

TensorFlow is an open-source end-to-end platform for Machine Learning. It contains a flexible ecosystem of tools, comprehensive, libraries and community resources that lets researchers push the state of the art in Machine Learning(ML) and developers easily build and deploy Machine Learning powered applications.

Multiple levels of abstraction is offered by TensorFlow so you can choose the right one according to your needs. Build and train your model using the high-level Keras API to make it easy to get started with TensorFlow and machine learning.

If more flexibility is needed than the, enthusiastic execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distributed Strategy API for distributed training in different hardware configurations without changing the model definition.

3.5 Keras

Keras is a high-level neural network library written in Python that acts as a wrapper for TensorFlow. It is used to quickly build and test neural networks with a minimum of lines of code. It includes implementations of commonly used neural network elements such as layers, targets, activation functions, optimizers, and tools that facilitate the manipulation of images and text data.

3.6 OpenCV

OpenCV (OpenSource Computer Vision) is an open source library of programming functions used for real-time computer vision. Mainly used for image processing, video recording and analysis of functions such as face and object recognition. Written in C ++, the primary interface, Python, Java, and MATLAB / OCTAVE bindings are available.

IV. METHODOLOGY

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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

4.1 Data Set Generation

Python 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. In the form of RGB values we found the datasets. Hence, we created our own data set. Steps followed for creation of our data set are as follows.

We used Open computer vision (OpenCV) in order to produce our dataset. Firstly, we captured around 800 images of each of the symbol in ASL (American Sign Language) for training purposes and around 200 images per symbol for testing purpose.

First, we capture each frame displayed by the webcam of our system. Each frame defines a region of interest (ROI), which is indicated by a blue contoured square, as shown in the image below.:

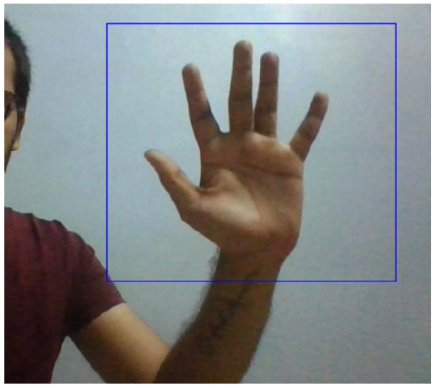

Figure 6: Capturing Data in Region of Interest (ROI)

Then, we apply Gaussian Blur Filter to our image which helps us extract various features of our image. The image, after applying Gaussian Blur, looks as follows:

Figure -7 Gaussian Blur processed data

4.2 Gesture Classification

Our approach uses a two-tiered algorithm to predict the user's final symbol.

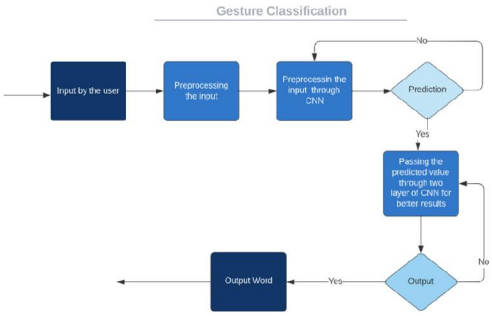

Figure 8

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

Algorithm Layer 1:

1. Apply Gaussian blur filters and thresholds to frames captured by OpenCV to get the processed image after feature extraction.
2. This processed image is passed to the CNN model for prediction, and if the character is recognized over 50 frames, the character is printed and the word formation is taken into account.
3. Space between the words is considered using blank symbol.

Algorithm Layer 2:

1. We detected various sets of symbols which show similar results on getting detected.
2. Then use a classifier created specifically for these sets to classify these sets.

Two-level algorithms are used to validate and predict symbols that are similar to each other so that they can approach when the displayed symbols are recognizable. Testing has shown that the following icons are not displayed correctly, and that other icons are also displayed.

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

To handle the above cases, we created three different classifiers to classify these sets.

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

4.3 Finger Spelling Sentence Formation Implementation

1. Whenever the number of recognized characters exceeds a certain value and other characters do not approach the threshold, print the character and add it to the current string (in this code, the value is 50). And set the difference threshold to 20)
2. Otherwise, clear the current dictionary containing the number of recognitions of the current symbol to avoid the possibility of predicting the wrong character.
3. No spaces are found whenever the number of spaces found (plain background) exceeds a certain value and the current buffer is empty.
4. Otherwise, a space is printed to predict the end of word and the stream is added to the next sentence.

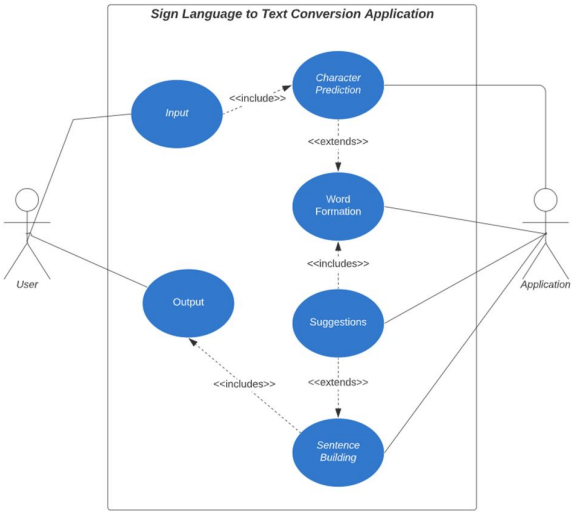

Figure 9: Application Flowchart

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

4.4 Autocorrect Feature

A library called Python Hunspell_suggest suggests the correct alternative for each (wrong) input word and displays a set of words that match the current word. The user can select a word that matches the current additional sentence. This reduces misspelling and predicts complex words.

4.5 Training and Testing

Converts the input image (RGB) to grayscale and applies Gaussian blur to remove unwanted noise. Apply the adaptive threshold to extract the hand from the background and resize the image to 128x128. The prediction layer estimates the likelihood that an image will fall into one of its classes. Therefore, the output is normalized between 0 and 1 and the sum of all the values in all bins is 1. We achieved this with the SoftMax function.

Initially, the output of the predictive layer is a bit far from the actual value. We trained our network with labeled data for better results. Cross entropy is a performance indicator used in classification. This is a continuous function that is positive for values that are not equal to the labeled value and is zero only if they are equal to the labeled value.

Therefore, we optimized the cross entropy by minimizing the cross entropy to near zero. To do this, adjust the neural network weights at the network layer. TensorFlow has a built-in function for calculating cross entropy. Now that we understand the cross entropy function, we have optimized it for gradient descent using the best gradient descent optimizer called Adam Optimizer.

V. Challenges Faced

There were many challenges during the project. The first problem we faced was with datasets. We wanted to process RAW images and square images like Keras's CNN because it's much more convenient to process only square images.

We couldn't find an existing dataset that matches my needs, so I decided to create my own dataset. The second problem was choosing a filter that could be applied to the image so that it would get the correct features of the image and serve that image as input to the CNN model. We tried various filters including binary threshold, canny edge detection and Gaussian blur etc. but finally settled with Gaussian Blur Filter.

More issues were faced relating to the accuracy of the model we had trained in the earlier phases. This problem was eventually improved by increasing the input image size and also by improving the data set.

VI. Results

Images 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 American sign language.

Most of the research papers focus on using devices like Kinect for hand detection. In [7] they build a recognition system for Flemish sign language using convolutional neural networks and Kinect and achieve an error rate of 2.5%.

In [8] 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 [9] they achieve an average accuracy of 86% for 41 static gestures in Japanese sign language.

Using depth sensors map [10] 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. Sensors like Kinect are not only readily available, but also expensive to buy for most viewers. This model uses a standard laptop webcam. Hence it is great plus point.

VII. Conclusion

In this report, functional real-time American Sign Language visual recognition was developed for D & M people in the asl alphabet. We achieved final accuracy of 98.0% 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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

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

REFERENCES

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

[2]. Pujan _Ziaie, Thomas M uller, Mary El len Foster, and Alois Knoll "A Nai ıve Bayes Munich, Dept. of Informatics VI, Robotics and Embedded Systems, Boltz mannstr. 3, DE-85748 Garching, Germany.

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

[4]. Mohammed Waleed Kalous, Machine recognition of Aus lan signs using Power_Gloves: Towards large_lexicon recognition of sign language.

[5]. aeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural Networks-Part-2/

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

[7]. Pigou L., Diel eman S., Kindermans PJ., Schrauwen B. (2015), Sign_Language_Recognition Using Convolutional Neural Networks. In: Aga pito L., Bronstein M., Rother C. (eds) Computer_Vision - ECCV 2014 Workshop. ECCV-2014. Lecture Notes in ComputerScience, vol-8925. Springer, Cham

[8]. Zaki, M.M., Sha heen, S.I.: Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters 32 (4), 5 72–577 (2011).

[9]. N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," 2017 Nico graph International (Nico Int), Kyoto, Japan, 2017, pp. 19-24. doi: 10.1109/NICOInt.2017 .9

[10]. Byeong keun Kang, Subarna Tripathi, Truong Q. Ngu yen" Real-time sign language fingerspelling recognition using convolutional neural networks from depth map" 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)

[11]. Number_System_Recognition (https://github.com/chasinginfinity/number-sign-recognition)