SIGN LANGUAGE HAND GESTURE RECOGNITION SYSTEM

Major project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By
Aryan Gupta(181331)

Esha Pandey(181336)

UNDER THE SUPERVISION OF

Mr. Deepak Gupta

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234,

Himachal Pradesh, INDIA

CERTIFICATE

This is to certify that the work which is being presented in the project report titled **Sign Language Hand Gesture Recognition System** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Esha Pandey and Aryan Gupta during the period from August 2021 to May 2022 under the supervision of **Mr. Deepak Gupta**, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

Aryan Gupta (181331) Esha Pandey(181336)

The above statement made is correct to the best of my knowledge.

Mr. Deepak Gupta

Asst. Prof. Senior Grade

Computer Science & Engineering
Jaypee University of Information Technology, Waknaghat.

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing that made it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Mr. Deepak Gupta, Asst. Prof. Senior Grade,** Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of machine **Learning To** carry out this project. His Endless Patience, scholarly guidance, continual encouragement, constant and energetic supervision,constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Mr.Deepak Gupta, Department of CSE, for his kind help to finish our project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing,which have developed their convenient help and facilitated my undertaking. Finally, I must acknowledge with due respect the constant support and patience of my parents.

Aryan Gupta(181331) **Esha Pandey(181336)**

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

Hand motion is one of the techniques utilised in communication through signing for non-verb al correspondence. It is most generally utilized by hard of hearing and stupid individuals who have hearing or discourse issues to impart among themselves or with typical individuals. Different communication via gestures frameworks has been created by numerous creators all over the planet yet they are neither adaptable nor savvy for the end clients. Subsequently in this paper presented programming which presents a framework model that can consequently perceive communication through signing to assist tragically challenged individuals with conve ying all the more successfully with one another or ordinary individuals. Design acknowledgm ent and Gesture acknowledgment are the creating fields of examination. Being a critical part in nonverbal correspondence hand signals are assuming a key part in our regular routine.

Hand Gesture acknowledgment framework gives us an imaginative, normal,easy to use method of correspondence with the PC which is more recognizable to the people. By considering as a primary concern the likenesses of human hand shape with four fingers and one thumb, the product intends to introduce a continuous framework for acknowledgment of hand signal on premise of identification of some shape based highlights like direction, Center Of mass centroid, fingers status, thumb in places of lifted or collapsed fingers of hand.

Chapter 01 : INTRODUCTION

1.1 Introduction

The ability to interact naturally with systems is becoming increasingly important in many areas of human-computer interaction. Gesture recognition is one of the modes to break the barrier between human and computer interaction. Gesture recognition is a computer process that attempts to recognize and interpret human gestures using mathematical algorithms.

Hand gesture recognition systems provide a natural , innovative and modern way of non verbal communication. The setup consists of a single camera that captures user-generated gestures and takes this hand image as input to the proposed algorithm. Gesture recognition can be used to recognize not only human hand gestures, but everything from head nodding to various gait.

As we are aware, the vision based innovation of Hand motion recognition is an important component of human-PC connection. Somewhat recently, console and mouse assume a huge part in human PC communication. In any case, attributable to the rapid technological and programming advancements, new sorts of human PC connection techniques have been required. Specifically, advances like discourse acknowledgment and signal acknowledgment get incredible consideration in the field of HCI.

Motion is an image of actual conduct or passionate articulation. It incorporates body signals and hand motion.

Hand signals are one of the nonverbal specialized strategies utilized in communication through signing. It is most ordinarily utilized by challenged people who have hearing or discourse issues to speak with another or with non-hard hearing individuals. Numerous producers all through the world have made different communication via gestures frameworks, but they are neither

versatile nor savvy for end clients. Plan affirmation and Gesture affirmation are the making fields of assessment. Being a basic part in nonverbal correspondence hand signals are expected to be a vital part in our standard daily practice. Hand Gesture affirmation structure gives us an innovative, standard, simple to utilize strategy for correspondence with the PC which is more conspicuous to the people. By considering as a first concern the similarities of human hand shape with four fingers and one thumb, the item means to present a ceaseless system for affirmation of hand movement on reason of acknowledgment of some shape based features like bearing, Center of mass centroid, finger status, thumb in spots of lift or fallen fingers of hand.

Hand gesture recognition has been achieved using both non-visual and vision-based techniques. The location of finger development with a couple of wired gloves is an illustration of a non-vision-based strategy. Vision-based procedures are more normal overall since they don't include the utilization of any hand contraptions. The writing partitions hand motions into two classifications: static and dynamic signals. Static hand signals are those where the hand's situation and direction in space don't change for a while. Dynamic signals are those where there are any progressions inside a particular time period. Hand waving is an illustration of a unique hand motion, while joining the thumb and pointer to shape the "Ok" image is an illustration of a static hand signal.

To start with, the hand area is distinguished from the first pictures from the information gadgets. Then, at that point, a few sorts of highlights are removed to portray hand motions. Last, the acknowledgment of hand signals is cultivated by estimating the likeness of the element information. The skin shading delicate to the lighting condition and component focuses are joined to heartily recognize and portion the hand locale. At the point when the area of interest (ROI, the hand locale for the situation) is distinguished, highlights should have been separated from the ROI district. Shading, splendor, and inclination esteems are generally utilized elements. CRF (restrictive irregular field), and adjusted supporting classifier are prepared to segregate hand signals. Although the acknowledgment execution of these refined classifiers is great, the time cost is extremely high.

In this paper, we present a productive and successful strategy for hand motion acknowledgment. The hand district is identified through the foundation deduction technique. Then, at that point,

the palm and fingers are parted to perceive the fingers. After the fingers are perceived, the hand signal can be ordered through a straightforward standard classifier.

American Sign Language(ASL) is a worldwide, predominantly used sign language which is only used by the people who have the D&M (deaf and dumb) disability. They use American Sign Language to communicate to each other. The process of communicating thoughts and information using a variety of techniques, including voice, signs, behavior, and imagery, is known as communication. Deaf and dumb people communicate with others by making different gestures with their hands. Gestures are nonverbal communications that can be recognised with the naked eye. The nonverbal communication of the deaf and dumb is known as sign language..

ASL is a language totally isolated and unmistakable from English. It contains every key element of language, with its own standards for declaration, word arrangement, and word request. While every language has techniques for hailing different limits, for instance, representing a request as opposed to saying something, tongues change in how this is done. For example, English Speakers could represent a request by raising the pitch of their voices and by changing word demand; ASL clients represent a request by creating an upheaval, growing their eyes, and moving their bodies forward.

ASL started in the mid nineteenth century in the American School for the Deaf in West Hartfords, Connecticut, from a circumstance of language contact. From that point forward, Schools for the deaf and Deaf people's organizations have widely adopted ASL. Despite its widespread use, there has never been a definite census of ASL users. The number of ASL clients in the United States ranges from 250,000 to 500,000, including children of hard of hearing adults.

Similarly likewise with different dialects, explicit methods of communicating thoughts in ASL fluctuate as much as ASL clients themselves. Not with individual contrasts in articulations, ASL has local accents and vernaculars; similarly as specific English Words are expressed contrastingly in various pieces of the country, ASL has provincial varieties in the beat of marking, elocution, shoptalk, and signs utilized. Other sociological elements, including age and

sex, can influence ASL use and add to its assortment, similarly likewise with communicated in dialects.

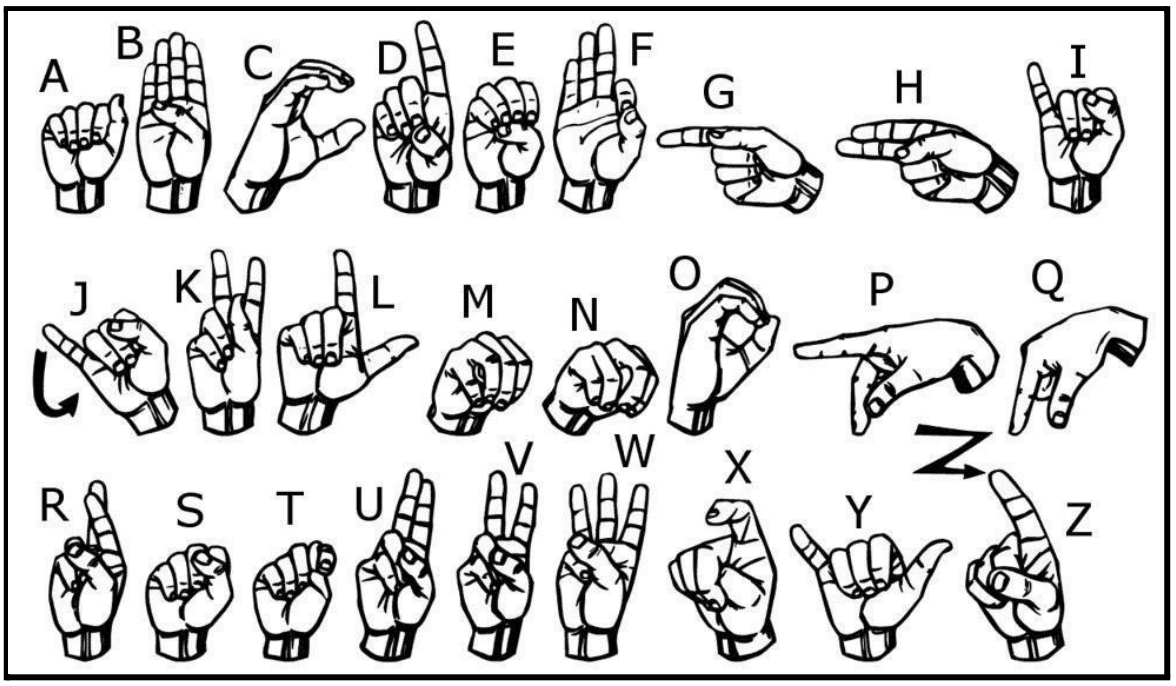

Fig 1: American Sign Language Symbols

1.1.1 Sign Language

Sign language is a visual gestural communication system used by deaf and hard of hearing persons. Hand motions and gestures, which have their own lexicon and grammar, are used to express meaning.

The architecture for a gesture recognition system includes a simple, effective, and accurate technique for converting motions into text or voice. Data capture, data preparation and transformation, feature extraction, classification, and ultimately the outcome are all important aspects of sign language recognition (into text or speech).

Separating the correspondence barrier among persons and gazing back, communication by

gestures acknowledgement Sign language recognition aims to break down communication barriers between individuals by giving hard of hearing and silent persons more opportunities to be heard and recognised consistently.

1.1.2 CNN

Because a computer's picture is nothing more than a matrix, it cannot view things like we do. In CNN, the input to a layer is three-dimensional, with height, breadth, and depth. Convolutional layer, pooling layer, fully linked layer, and final output layer are the main components.

1. Convolution Layer :

The CNN building block is the convolution layer, which is effectively a layer in which items convolve on one another. The learnable filter is a matrix that convolves across a constrained area of the picture represented by another matrix to produce a 2D activation matrix. By the completion of the convolution process, there is an activation matrix with fewer parameters (dimensions) and more distinct features than the original picture.

2. Pooling Layer :

The number of learnable parameters and CPU power required to evaluate the input are lowered as a consequence of the implementation of a pooling layer to minimize the size of the activation matrixx. There are two forms of pooling:

a) Max Pooling :

We take the maximum of all the integers in the pooling area (kernel) and change the pooling region each time to process a different neighborhood of the matrix in this method (input). The resulting matrix is half the size of the activation matrix.

b) Average Pooling :

Average Pooling is a pooling procedure that estimates the average value for features map patches and utilizes it to produce a downsampled (pooled) feature map. After a convolutional layer, it's commonly used. In average pooling, all values in the pooling region are averaged. It introduces a little amount of translation invariance, meaning that minor changes in the image have little influence on the values of most pooled outputs. It extracts smoother characteristics than Max Pooling, whereas Max Pooling extracts more apparent features like edges.

3. Fully Connected Layer :

Until now, some visual characteristics had been retrieved and the image's size had been decreased. This layer signals the start of the picture categorization process.
After the data has been correctly converted, it is flattened into a single column vector and fed into the feedforward neural network, with backpropagation done to each iteration during training.

4. Final Output Layer :

The output of a fully connected layer is sent into a final layer of of neurons [with count equal to the total number of classes] that provides the final output in the form of probability, that is, values between 0 and 1 for the final prediction.

1.2 Problem Statement

There are millions of people in this world who have the disability of speaking and hearing but with an abundance of things to talk about, to communicate to one another they use sign language. Regular people and D&M persons are separated by a linguistic barrier in the form of a sign language framework that differs from conventional text. As a result, they interact through vision based communication.

The issues looked by the not-too-sharp individuals right now and the hardships of their correspondence with typical people started our advantage and drove us to attempt to track down an answer for their challenges and to limit them however much as could reasonably be expected. Since they address a critical piece of society, they need to convey their thoughts in the least difficult manner by straightforward gadgets. So our venture plans to overcome this issue by empowering correspondence between dumb\deaf individuals from one perspective and ordinary individuals on other hand by presenting an economical electronic gadget that makes an interpretation of the fingers presses into the message and discourse.

The signals can be effectively perceived by others assuming there is a standard stage that changes gesture based communication to message. Accordingly, research has been led on a dream based interface framework that will permit D&M people to convey without communicating in a similar language.

With so many individuals relying on sign language to communicate, holding a conference where a handicapped person and an abled person can't communicate and aid each other becomes nearly impossible. It's also worth noting that, unlike spoken languages, which vary considerably from nation to country and are based on various areas, more than 80% of persons with disabilities can communicate and comprehend sign language, even if they come from entirely different places. As a result, a work initiative aiming at overcoming this communication barrier would be extremely beneficial to persons with impairments.

1.3 Objective

The objective of this examination is to concentrate on the exhibition of a Convolutional Neural Network (CNN) perceiving and interpreting into text SL pictures (motions performed by a hand). Given the expansive extent of this undertaking We restricted the extent of the review to American Sign Language (ASL) letters. The goal of this project is to create a user-friendly

human-computer interface (HCI) in which the computer recognises human sign language to bridge the communication gap between the D&M people and regular people.

1. To build a model which captures the images using OpenCV to create the dataset of American Sign Language.

2. The generated dataset is utilized to train the Convolutional Neural Network model, which allows the computer to recognise the symbols.

1.4 Methodology

The system is founded on the concept of vision. All of the signs are made with the hands, which eliminates the need for any artificial equipment for interaction. We used static sign language to generate the dataset for our project. Our project's static sign language data was in the form of photographs. We converted those coloured images to black&white images and then applied a threshold to detect only the border or outlines of the hand. To detect the signs represented by each of these images, we trained the dataset with a Convolutional Neural Network (CNN) model. The proposed model is based on the object recognition benchmark. According to this benchmark, all the tasks related to an object recognition problem can be ensembled under three main components: Backbone, Neck and Head.

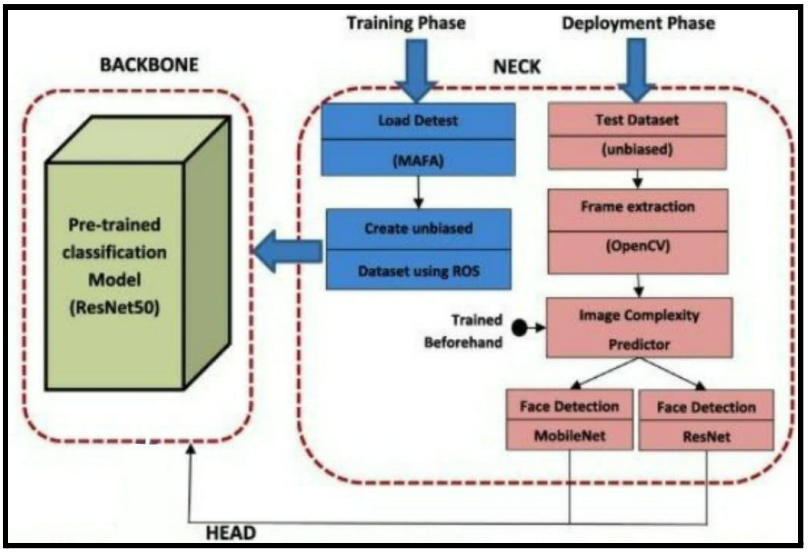

Fig 2: Proposed architecture

The backbone here is a basic convolutional neural network that can extract information from photos and transform it to a feature map. The notion of transfer learning is used on the backbone of the proposed architecture to extract new features for the model by utilizing previously learnt attributes of a powerful pre-trained convolutional neural network.

Recognizing gestures has recently been the subject of extensive investigation. The use of Hidden Markov Models (HMMs) resulted in the detection of context-dependent models being limited. ASL is one of the most extensively used sign languages for establishing communication.

The following are the essential phases in establishing recognised gestures with the use of a literature review:

1.4.1 Data acquisition

When developing a proper data acquisition plan, the key goal is to understand the system's requirements in terms of data volume, diversity, and velocity, and then to choose the optimal instrument to assure the acquisition and necessary throughput.

Types of data acquisition:

1. Sensor based

Hand motion, configuration, and position are extracted using electromechanical devices. To extract information, various glove-based methodologies can be applied. However, it is both costly and inconvenient to use.

2. Vision based

A computer camera is used as the input device in a vision-based technique to receive information from hands or fingers. The acquired data is either a single image frame or a series of images. The fundamental issue of vision-based hand detection is dealing with the huge variability in the look of the human hand due to a large number of hand movements, variable skin color options, and differences in the angles of view, scales, and speed of the camera capturing the image.

1.4.2 Data preprocessing

Data preprocessing is a crucial phase in the data mining process that involves manipulating or removing data before it is utilized to ensure or improve performance. The phrase "garbage in, trash out" applies to data mining and machine learning initiatives in particular.

Fig 3: Data Preprocessing

1.4.3 Feature extraction

Feature extraction is a sort of dimensionality reduction in which a huge number of pixels in an image are efficiently represented in such a way that the image's most interesting sections are effectively captured. Feature selection is a useful technique for minimizing the number of dimensions in high-dimensional data. Feature Extraction is a technique for reducing the amount of features in a dataset by generating new ones from existing ones. The original set of features should then be able to summarize the majority of the information in the new reduced set of features. From a combination of the original set, a summarized version of the original features may be generated.

1.4.4 Gesture Classification

A method for recognising and classifying healthy subjects' hand movements in real time has been discussed. Also discussed is the analysis of selecting the optimal characteristics and classifier for the intended hand movements.

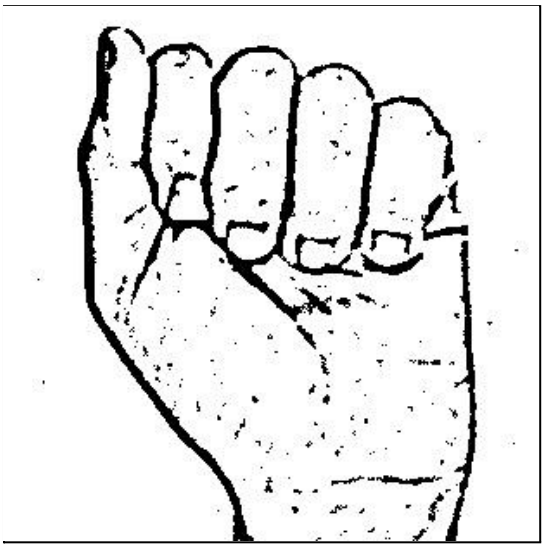

Fig 4: Gesture Classification

1.5 Technical Requirements

Tools

- Anaconda - Jupyter Notebook

  Anaconda is an open-source data science distribution for the Python and R programming languages that attempts to make package management and deployment easier. Data-science packages for Windows, Linux, and macOS are included in the release.

Technology

- Convolutional Neural Network(Machine Learning)

  A convolutional neural network is a type of artificial neural network used in deep learning to evaluate visual information.

Language

- Python

  The scripting language used to create this project is Python. Keras and TensorFlow, two popular Python libraries for creating neural networks, were employed.

Libraries

- OpenCV

  It's an open source collection of functions for image processing, object recognition, feature analysis, and video and image capture.

- Numpy

  NumPy is a Python library that adds support for huge, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to operate on these arrays.

- Pandas

  pandas is a data manipulation and analysis software package for the Python program ming language. It includes data structures and methods for manipulating numerical tables and time series, in particular.

- Tensorflow

  Tensorflow is a library that handles the mathematical processing behind a neural network, such as computing and improving the loss function by modifying weights and biases to produce correct prediction results.

- Keras

  Keras is a wrapper for Tensorflow that is used when a neural network needs to be quickly created and operated in a few lines of code. It includes layers, targets, act-i vation functions, optimizers, and other tools for working with images and text data.

1.6 Data Workflow Diagram

1.6.1 Data flow during model development

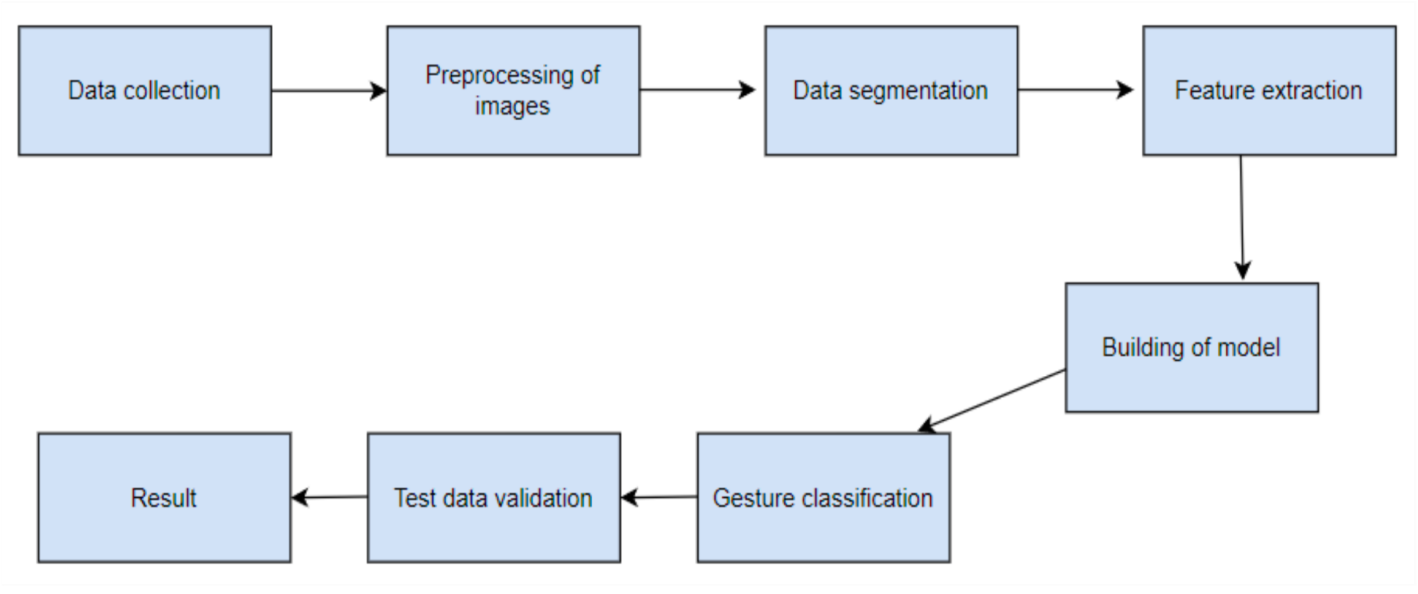

Fig 5: Data flow during model development

1.6.2 Data flow in GUI

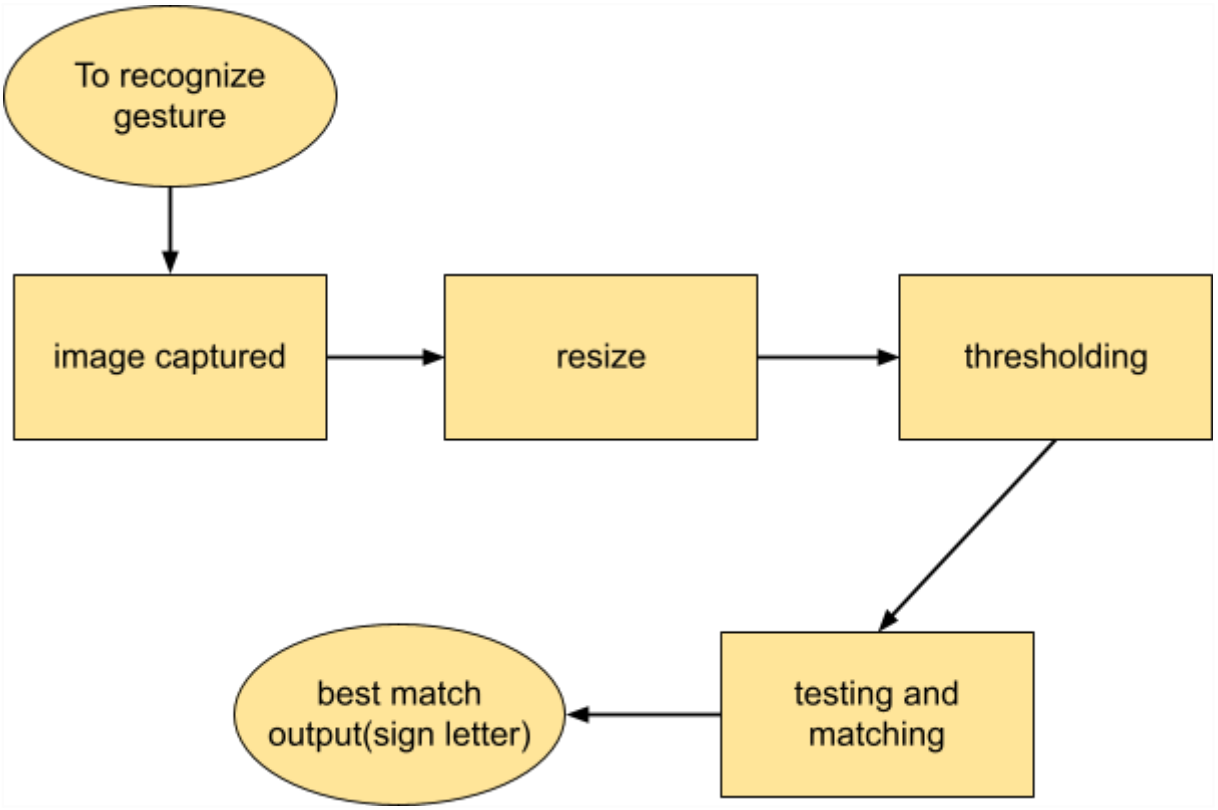

Fig 6: Data flow in GUI

Chapter 02 : LITERATURE SURVEY

2.1 Literature survey

1. Deafs Mute Communication Interpreter - Reviews :

The purpose of this paper is to discuss the various deaf-mute communication translator systems that are currently in use. Wearable Communication Device and Online Learning System are the two broad categories of communication approaches utilized by deaf-mute people. Glove-based systems,keypad methods, and Handycam Contact screen are generally instances of wearable specialized strategies. Different sensors, an accelerometer, a reasonable microcontroller, a text to discourse change module, a keypad, and a touch-screen are utilized in every one of the three sub-partitioned strategies expressed previously. The subsequent choice, an internet learning framework, can take out the necessity for an outer gadget to translate messages between hard of hearing quiet and non-hard of hearing quiet individuals. The Online Learning System utilizes an assortment of procedures. Thin module, TESSA, Wi-See Technology, SWI PELE System, and Web-Sign Technology are the five development draws near.

2. An Efficient Framework for Indian Signs Language Recognition Using Wavelet Transform :

The suggested ISLR system is a patterns recognition technique with two key components: features extraction and classiffication. To recognise sign language, a combination of Discr ete Wavelet Transform (DWT)-based feature extraction and a nearest neighbor classifier is utilized. The experimeental results reveal that utilizing a cosine distance classifier, the proposed hand gesture recognition systems achieves a maximum classification accuracy of 89.23%.

3. Hand Gesture Recognition Using PCA in :

The authors of this study provided a strategy for database-driven hand gesture identification based on a skin color model approach and thresholding approach,as well as an effective te-mplate matching approach, that may be employed in human robots and other applications. The hand region is first segregated using a skin color model in the YCbCr colorspace. Thre sholding is used to distinguish the foreground and background in the next stage. Finally, for recognition,a template-based matching technique is constructed utilizing Principal Component Analysis (PCA).

4. Hand Gesture Recognition System For Dumb People :

The creators showed an advanced picture handling based static hand signal recognition framework. The SIFT strategy is utilized to make a hand motion including vectors. At the edges, SIFT highlights have been registered that are invariant to scaling, turn, and commotion expansion.

5. An Automated System for Indian Sign Language Recognition in :

This paper presents a strategy for consequently perceiving signs utilizing shape-based qualities. Otsu's thresholding calculation is utilized to fragment the hand area from the pictures, which decides an ideal edge to decrease the inside class variety of thresholded highly contrasting pixels. Hu's invariant minutes are utilized to ascertain highlights of the fragmented hand area, which are then taken care of into an Artificial Neural Network for characterization. Precision,Sensitivity and Specificity are utilized to assess the framework's exhibition.

6. Hand Gesture Recognition for Sign Language Recognition: A Review in :

The creators gave an assortment of hand motion and gesture based communication acknowledgment techniques that have recently been proposed by different researchers. Communication through signing is the main method for correspondence for the almost totally senseless. These genuinely incapacitated individuals impart their feelings and considerations to others utilizing communication through signing.

7. Design Issue and Proposed Implementation of Communication Aid for Deaf & Dumb People in:

The author of this research proposed a system to help deaf and dumb persons communicate with normal people using Indian Sign Language(ISL),whereas hand gestures are transformed into relevant text messages. The main goal is to create an algorithm that can turn dynamic gestures into text in real time. Finally, after testings, the system will be integrated on the Android platform and made available as an app for smartphones and tablet computers.

8. Real Time Detection And Recognition Of Indian And American Sign Language Using Sift In

For human-PC collaboration in an assortment of utilizations, the creator fostered an ongoing vision-based framework for hand signal recognizable proof. The framework can perceive 35 distinctive hand motions in Indian and American Sign Language (ISL and ASL) at a much quicker rate and with an undeniable degree of exactness. To diminish the likelihood of incorrect location, a RGB-to-GRAY division calculation was applied. The creators offered an ad libbed Scale Invariant Feature Transform(SIFT),an approach for separating highlights ,which they utilized. MATLAB is utilized to demonstrate the framework. A GUI worldview

has been made to plan an effective and easy to understand hand motion acknowledgment framework.

9. A Review on Feature Extraction for Indian and American Sign Language in :

The research and development of sign language based on manual communication and body language were reported in this paper. Pre-processing, feature extraction, and classifications are usually the three processes of a sign language recognition system. Neural Network (NN), Support Vector Machine(SVM), Hidden Markov Models(HMM), Scale Invariant Feature Transform(SIFT), and other classification approaches are utilized for recognition.

10. SignPro - An Application Suite for Deaf and Dumb in :

The author demonstrated an application that uses sign language to assist deaf and dumb people in communicating with the rest of the world. The real-time gesture to text conversion is a crucial aspect of this technology. Gesture extraction, gesture matching, and speech conversion are among the processing phases. ISSN No.: 2454-2024 (online) International Journal of Technical Research & Science pg. 433 www.ijtrs.com www.ijtrs.org Gesture extraction involves the use of various image processing techniques. Matching histograms, computing bounding boxes, segmenting skin color, and expanding regions are some of the techniques used. Feature point matching and correlation based matching are two techniques that can be used for Gesture matching. Text to gesture conversion and voicing out of text are two further functions of the programme.

11. Offline Signature Verification Using Surf Feature Extraction and Neural Networks Approach :

The use of neural networks for off-line signature detection and verification is proposed in this paper, where the signature is collected and provided to the user in an image format.

12. Mahesh, M., Jayaprakash, A., & Geetha, M. (2017, September) -

The authors proposed an Android application that allows a gesture to be added to a data set and identified. Descriptors are used to identify gestures, and then histogram comparison is used to narrow down the data set for descriptor comparison. The image is initially preprocessed before being used for recognition. At each iteration, the photos in the dataset are loaded one by one. Following that, the photos are pre-processed. The two pre-processed photos are sent into the matcher, which compares the images using histogram and ORB descriptor matching and outputs the image name if a good match is discovered.

13. Byeongkeun Kang , Subarna Tripathi , Truong Q. Nguyen(2015) -

The authors discuss the use of depth sensors to gather extra information. Accuracy is improved by using GPU and CNN, as well as a depth sensor technique. The hand is segmented by first considering the closest area of the image from the camera as the hand and then using a black wristband to create a depth void around the wrist. Hand segmentation is performed by locating related components from the closest region in this depth image.

14. Wu, J. (2017) -

The writers go over the fundamentals of CNN, including the convolution layer, which is the foundation of CNNs. This paper also discusses activation functions, including the most extensively used activation function, the Rectified Linear Unit (ReLU). To add non linearity to the data, the ReLU function is utilised. Pooling layer is another significant component of CNN that reduces the size of the activation matrix and comes in two types: average and maximum pooling.

Chapter 03 : SYSTEM DEVELOPED

3.1 Overview of the Method

The study reported in this paper aims to develop a classification model for hand letter and number gestures as well as a new dataset for hand gesture identification. The model recognises the gesture in each image by detecting local characteristics in the hand images. The photos in the produced dataset are constant size, the hands are centered, and the hand values are normalized. Because all of the preprocessing stages have been completed on the pictures, they are all in the same coordinate system and can be directly compared. The figure below shows the working layout of the project. To begin, the sign language symbol images are captured using the OpenCV library, and the coloured images are converted to black&white images to reduce the background noise and memory.

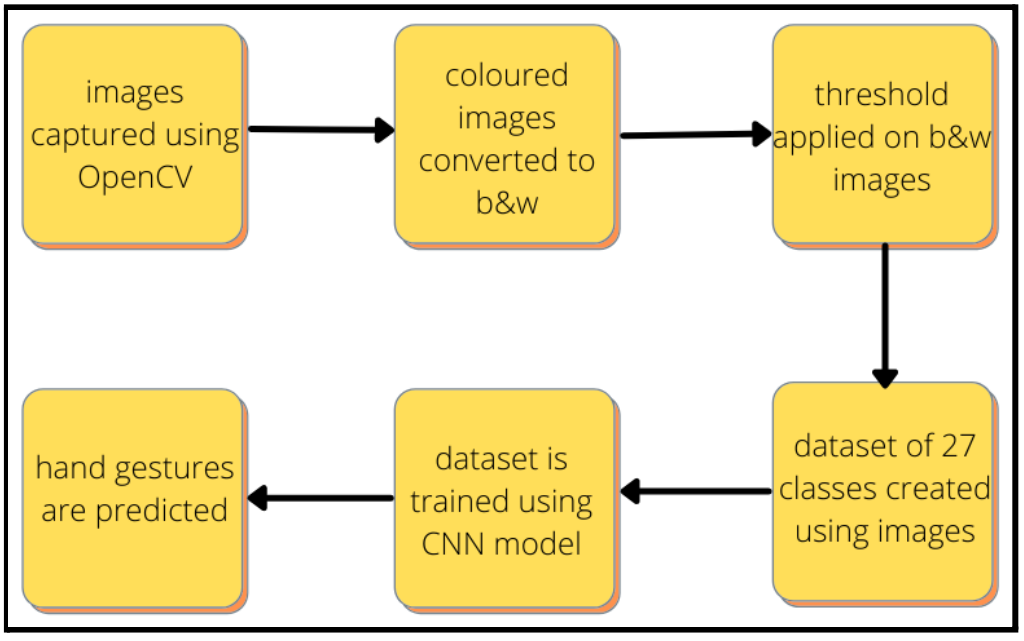

Fig 7: Working layout

Then threshold is applied to the images to select only the outlines of the hand for a more clear dataset. By doing this, 27 classes (26 alphabets and 1 class for blank space) are created. This dataset is not used to train the Convolutional Neural Network(CNN) model. Now our model predicts the ASL alphabets in real-time.

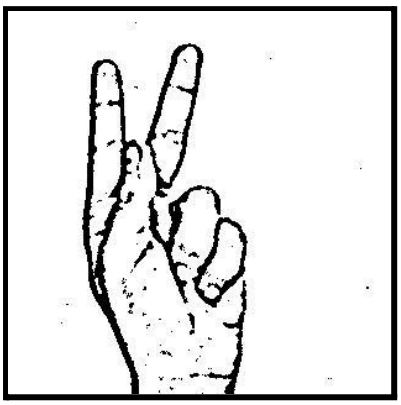

Fig 8: Symbol ‘K’ in ASL after applying threshold to the image

The user's symbols are captured in the first phase of Sign Language Recognition.

Methods for identifying persons and body parts have been extensively researched, and a wide range of applications have been developed. In image processing, neural networks (NN) are frequently employed as recognition models. We offer a basic NN model as well as several modifications that have been applied to various fields. I concentrate on a specific network type, the Convolutional Neural Network (CNN), which is a popular NN in image processing due to its ability to recognise local features. When a user interacts with a NUI system using their body, the identified features are frequently employed as input. This kind of interaction eliminates the need for physical control devices, allowing for natural computer communication. NUI's main goal is to provide techniques for providing a positive user experience while engaging with the body directly and reducing uncertainty.

3.2 Dataset

A proper hand gesture dataset on American Sign Language was not available on the internet, so we prepared our own dataset, around 1300 images (50 for each alphabet) were captured using opencv library in python. Some images had noise and some were just not good for the model, so out of these 1300 images some images were removed manually and 780 images were chosen which are suited best for the model, 30 for each alphabet. Then from all the collected images, the hand sign portion was selected using the 'labelImg' library for each image manually and a ".xml" file was created for each image which stores the pixels for every image. All the collected data is divided into training data and test data. 80% data for each alphabet was randomly selected for training data and the rest 20% data was used in testing.

All the collected images are labeled using the labelimage library in python. Picture marking is the most common way of relegating a specific tag to the specific articles present in our pictures with the goal that our machine can recognize them.

Picture marking is utilized in the PC vision field. Each article discovery framework, picture division model, movement identification, movement tracker models use the picture marking process as an underlying advance while planning information for our model.
For our picture marking task we utilized an open-source python bundle named "labelImg". This bundle can assist us with marking our pictures. This library uses pyqt5 to give us a GUI climate to make our assignment done.

We have used an open-source python bundle labelImg. Picture marking task is manual and tedious. Yet, the greater quality time you will spend in this cycle, the better your model will be.

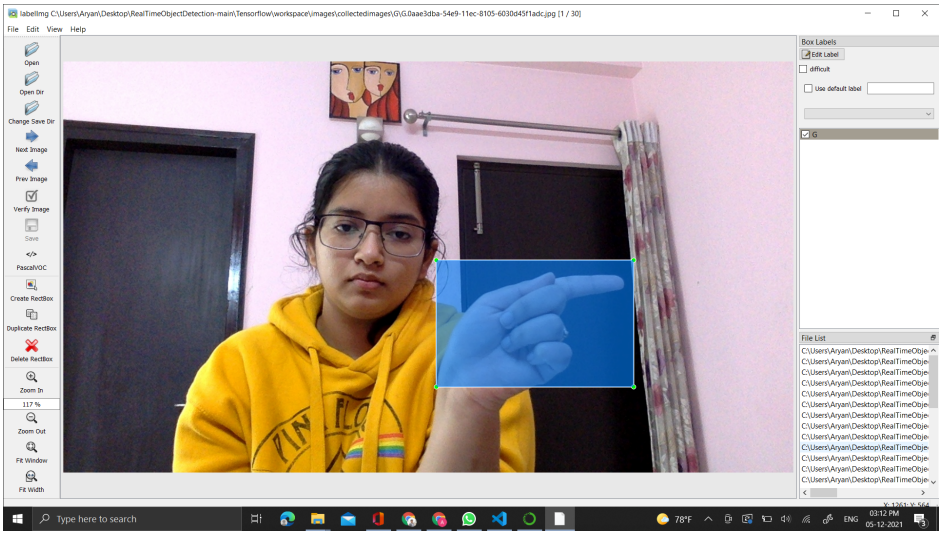

Fig 9: LabelImg

This labelImg library converts the selected part of the image into an xml file which stores all the pixels of the fingers and thumb detected which are further required to train the model, so basically this is one of the most important part because it makes the data readable for the machine so the machine can train the model using this data.

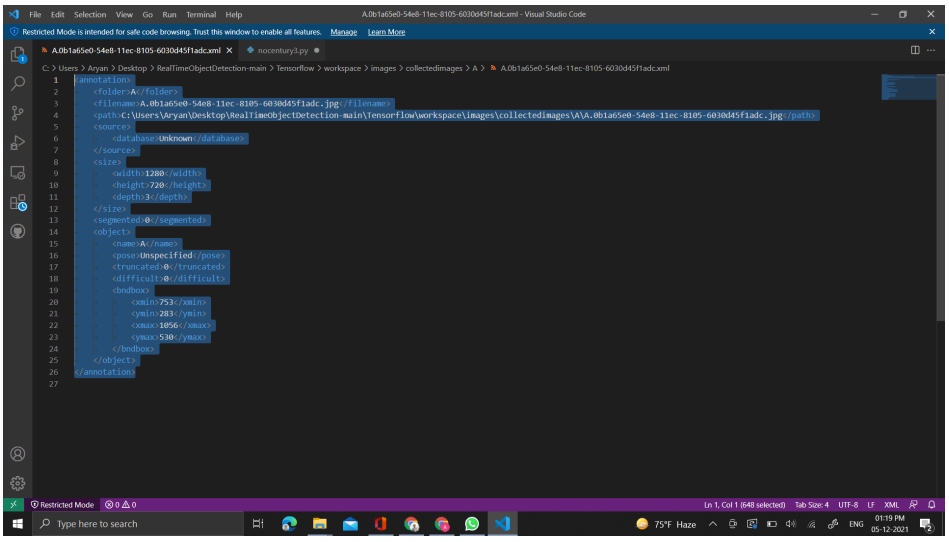

Fig 10: Label code

3.3 Convolutional Neural Networks

Convolutional Neural Network(CNN) is a class of Artificial Neural Network under Deep Learning. It is most applied to projects that are image related. The main goal of this field is to enable machines to see and comprehend the world in the same way that humans do, and to use that knowledge for a variety of tasks such as image and video recognition, image analysis and classification, media recreation, recommendation systems, natural language processing, and so on. Major breakthroughs in Computer Vision with Deep Learning have been built and developed over time, mostly through one algorithm, and that is **Convolutional Neural Network**.

As we have all of our images and we have created the dataset, now it will get trained under the Convolutional Neural Network model for further use.

Convolutional Neural Network is a combination of multiple steps, that are:

1. Input Dataset
2. Feature Learning
3. Classification

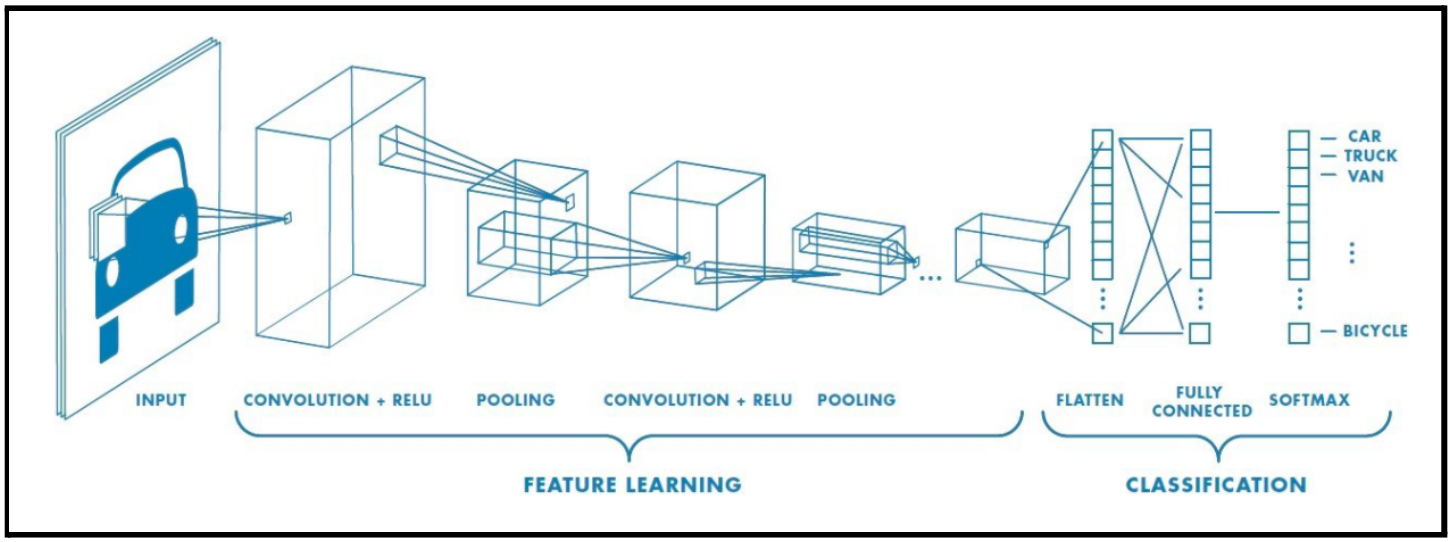

Fig 11: Construction of a typical CNN

3.3.1 Input Dataset

The representation of an image as a 3D matrix with dimensions equal to the image's height and width, as well as the value of each pixel's depth (1 in Grayscale and 3 in RGB). These pixel values are also used by CNN to extract valuable features.

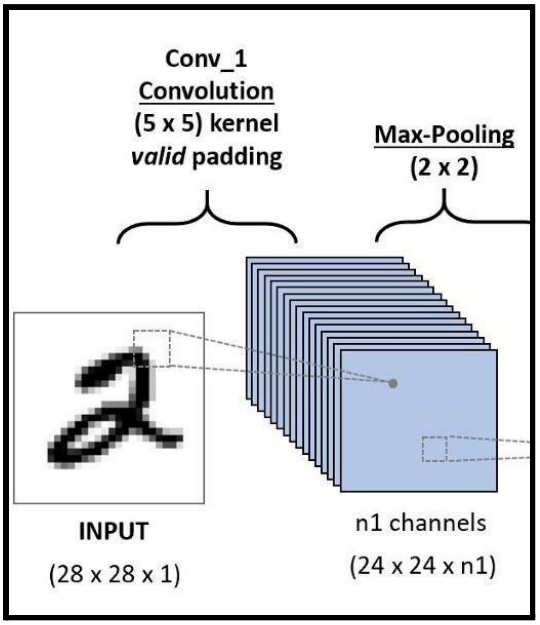

Fig 12: Feature extracted from the image

Every image that was captured using the laptop camera and OpenC library were coloured images. Coloured images are of three coloured channels: Red, Green and Blue (RGB). This makes it more difficult and complicated for our Convolutional Neural Network(CNN) model to get trained, it will consume more time and energy for the model to be trained. To avoid this issue we convert each image of our model into black&white images. Now each image has only one coloured channel.

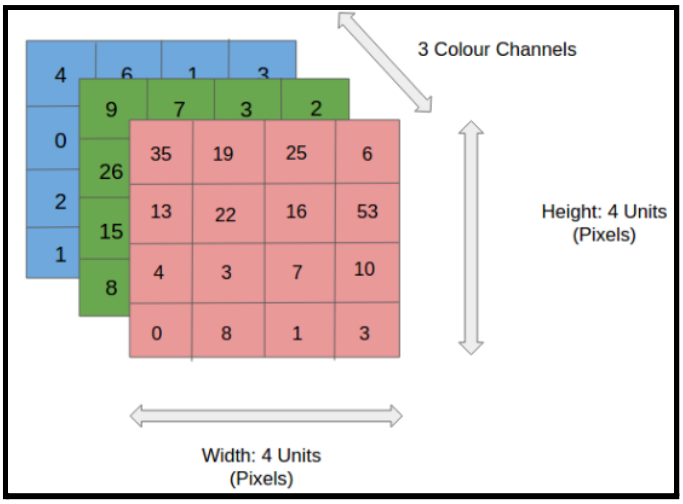

Fig 13: RGB coloured channel

Fig 14: Symbol 'A' in RGB coloured image to black&white to threshold

Now we apply threshold-an image processing feature which helps us to convert a black&white image to a binary image. This is a sort of image segmentation in which the pixels of an image are changed to make it easier to evaluate the image.

3.3.2 Feature Learning

1. Convolution Layer:

We used a window size of 5*5 in the convolution layer that extends to the depth of the input matrix. The layer is made up of learnable window size filters. We slid the window by stride size [usually 1] on each iteration and computed the dot product of filter entries and input values at

each place. Create a 2-Dimensional activation matrix that offers the response of that matrix at every spatial position as we continue this process. For example, the network will learn filters that activate when certain visual features, such as an edge of a certain direction or a splotch of a certain color, are detected.

2. ReLU Layer:

To all of the values in the input volume, the ReLU layer applies the function f(x) = max(0, x). In its most basic form, this layer simply sets all negative activations to 0. This layer improves the model's nonlinear properties and the overall network's nonlinear qualities without impacting the conv layer's receptive fields.

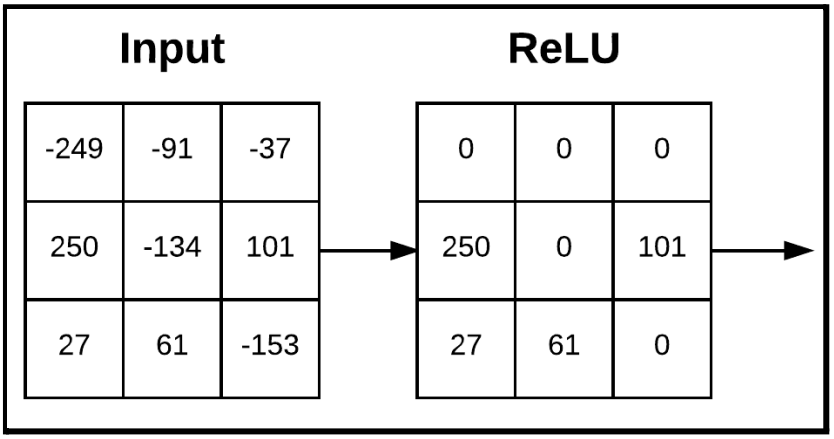

Fig 15: Results after ReLU Layer

3. Pooling Layer:

The pooling layer is used to lower the size of the activation matrix and, as a result, the learnable parameters. Pooling can be divided into two categories:

1. **Max Pooling:**

   We use max pooling to take the maximum of four values from a window [for example, a window of size 2*2]. So, if we close this window and keep going, we'll end up with an activation matrix that's half the size it was before.

2. **Average Pooling:**

We take the average of all values in a window when we use average pooling.

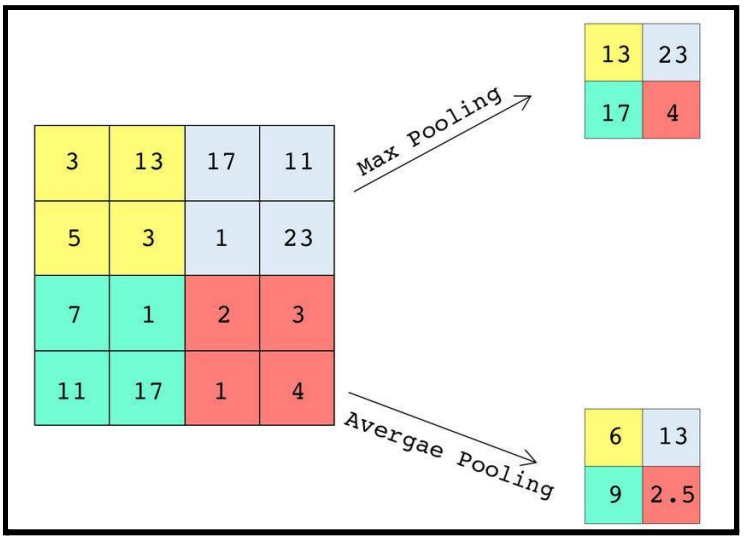

Fig 16: Max Pooling vs Average Pooling

3.3.3 Classification

1. Flatten

To construct a single lengthy feature vector, we flatten the output of the convolutional layers. Flattening is the process of turning data into a one-dimensional array for use in the next layer. It's also linked to the final classification model, which is referred to as a fully-connected layer.

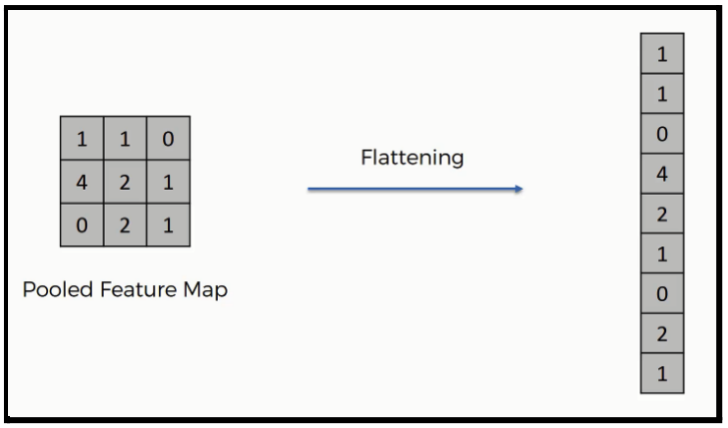

Fig 17: Flattening

2. Fully Connected Layer:

Fully Connected Layers are the network's final layers. Feed forward neural networks are what the Fully Connected Layer is all about. The output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer, is the input to the fully connected layer. A fully connected region is where all the inputs are well connected to the neurons, whereas, in a convolution layer, neurons are only connected to a limited region.

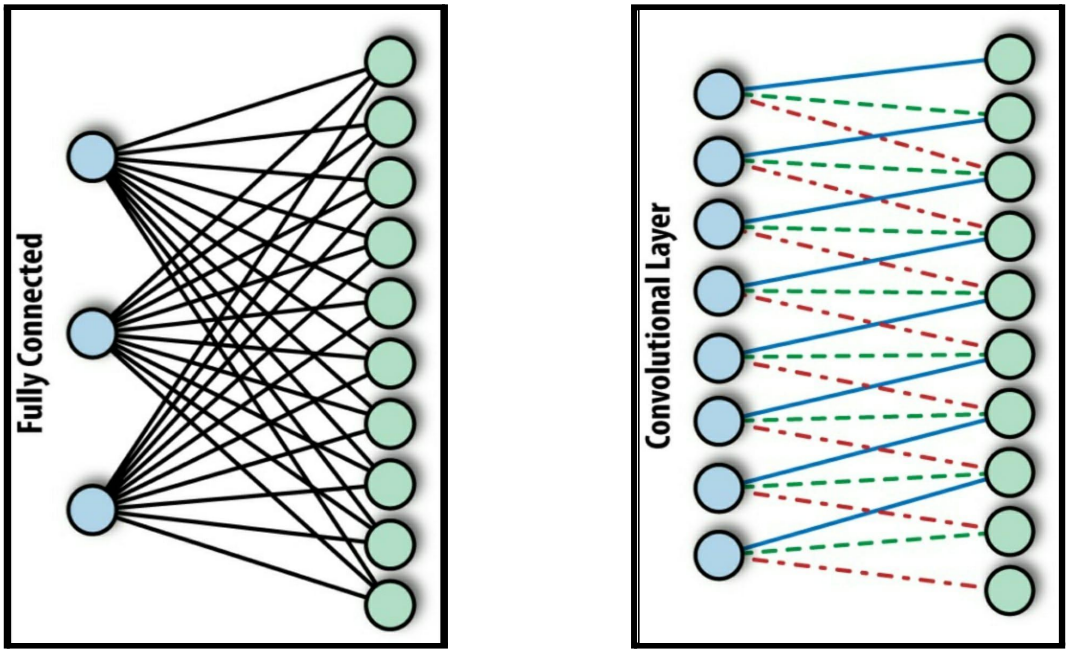

Fig 18: Fully Connected vs Convolutional Layers

3. Final layer:

After collecting values from the completely connected layer, connect them to the final layer of neurons [with a count equal to the total number of classes], which will forecast the likelihood of each image being classified into different classes. It is either **Softmax layer** or **Logistic layer.**

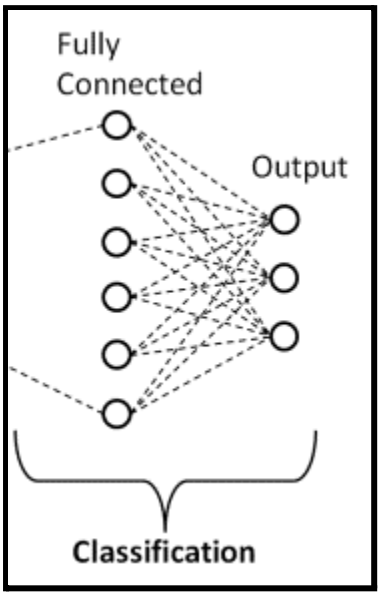

Fig 19: Final Output Layer

3.4 Implementation

3.4.1 Model building

1. Defined DataSet Location and Imported Module

```
import numpy as np
import pandas as pd
import tensorflow as tf
import os
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import cv2
import matplotlib.pyplot as plt
```

```
from tqdm import tqdm
```

```
from google.colab import drive

drive.mount('/content/gdrive')
path = 'gdrive/My Drive/ImagePro/'
```

```
Mounted at /content/gdrive
```

```
files = os.listdir(path)
```

```
files.sort()
```

Fig 20: Import modules and dataset

2. Created two label arrays to hold data, then used OpenCV to extract data such as BGR to RGB image conversion data.

```
files.sort()
```

```
print(files)
```

```
['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', 'Z', '_']
```

```
#image and its lable
image_array = []
label_array = []
```

```
for i in tqdm(range(len(files))):
    sub_files = os.listdir(path+"/"+files[i])
    #print(len(sub_files))
    for j in range(len(sub_files)):
        file_path = path+"/"+files[i]+"/"+sub_files[j]
        #CV2 read image
        image = cv2.imread(file_path)
        #resize 96*96
        image = cv2.resize(image,(96,96))
        #color RGB
        image=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
        image=cv2.cvtColor(image, cv2.COLOR_GRAY2RGB)

        image_array.append(image)

        label_array.append(i)
```

```
100%|██████████| 27/27 [02:54<00:00,  6.47s/it]
```

Fig 21: Using OpenCV to extract data

3. Cleaning the arrays and dividing the data into testing set and training set.

```
image_array = np.array(image_array)
label_array = np.array(label_array,dtype="float")
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,Y_train,Y_test= train_test_split(image_array,label_array,test_size = 0.2)
```

```
del image_array,label_array
import gc
gc.collect()
```

```
111
```

Fig 22: Cleaning and dividing dataset into test set and train set

4. Building model

```
from keras import layers,callbacks,utils,applications,optimizers
from keras.models import Sequential, Model, load_model
```

```
model = Sequential()
pretrained_model = tf.keras.applications.EfficientNetB5(input_shape=(96,96,3),include_top = False)
model.add(pretrained_model)
```

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb5_notop.h5
115269632/115263384 [==============================] - 1s 0us/step
115277824/115263384 [==============================] - 1s 0us/step
```

```
model.add(layers.GlobalAveragePooling2D())
```

```
model.add(layers.Dropout(0.3))
model.add(layers.Dense(1))
model.build(input_shape=(None,96,96,2))
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb5 (Functional)  (None, 3, 3, 2048)       28513527

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 1)                 2049

=================================================================
Total params: 28,515,576
Trainable params: 28,342,833
Non-trainable params: 172,743
_________________________________________________________________
```

Fig 23: Building Model

5. Processing Model

```
model.compile(loss='mse', optimizer='adam', metrics=['mae'])
```

```
ckg_path = "trained_model/model"
model_checkpoint = tf.keras.callbacks.ModelCheckpoint(filepath= ckg_path, monitor="val_mae",mode="auto",save_best_only=True,save_weight_only=True)
```

```
reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(
factor = 0.9, monitor = "val_mae", mode = "auto", cooldown = 0, patience = 5, verbose =1, min_lr= 1e-6
)
```

```
Epoch = 50
Batch_size = 32
```

```
history  = model.fit(X_train,Y_train,validation_data=(X_test,Y_test),batch_size = Batch_size, epochs= Epoch, callbacks=[model_checkpoint,reduce_lr])
```

```
Epoch 1/20
285/285 [==============================] - ETA: 0s - loss: 18.4967 - mae: 2.7166INFO:tensorflow:Assets written to: trained_model/model/assets
285/285 [==============================] - 336s 1s/step - loss: 18.4967 - mae: 2.7166 - val_loss: 2.5886 - val_mae: 1.0511 - lr: 0.0010
Epoch 2/20
285/285 [==============================] - ETA: 0s - loss: 3.0700 - mae: 1.1778INFO:tensorflow:Assets written to: trained_model/model/assets
285/285 [==============================] - 291s 1s/step - loss: 3.0700 - mae: 1.1778 - val_loss: 2.0691 - val_mae: 0.8733 - lr: 0.0010
Epoch 3/20
285/285 [==============================] - ETA: 0s - loss: 2.5544 - mae: 1.0244INFO:tensorflow:Assets written to: trained_model/model/assets
285/285 [==============================] - 292s 1s/step - loss: 2.5544 - mae: 1.0244 - val_loss: 2.2037 - val_mae: 0.7917 - lr: 0.0010
```

Fig 24: Processing Model

6. Resultant Model and Evaluation

```
results = model.evaluate(X_test,Y_test, batch_size=32)
```

```
72/72 [==============================] - 9s 125ms/step - loss: 1.5884 - mae: 0.5064
```

```
model.load_weights(ckg_path)
```

```
<tensorflow.python.training.tracking.util.CheckpointLoadStatus at 0x7fb378e6f850>
```

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model=converter.convert()
```

```
INFO:tensorflow:Assets written to: /tmp/tmpyq660ulz/assets
WARNING:absl:Buffer deduplication procedure will be skipped when flatbuffer library is not properly loaded
```

```
with open("model.tflite","wb") as f:
    f.write(tflite_model)
```

```
prediction_val = model.predict(X_test,batch_size=32)
print(prediction_val[:10])
print(Y_test[:10])
```

```
[[20.777805 ]
 [ 3.9893966]
 [19.925682 ]
 ...
 [25.997211 ]
 [ 1.0067749]
 [10.974005 ]]
[21.  4. 20. ... 26.  1. 11.]
```

Fig 25: Evaluation

```
pre = prediction_val.tolist()
Y_test = Y_test.tolist()

pre prediction_val.tolist() = Y_test
Y_test.tolist()

correct = 0
wrong = 0

for i in range (len (pre)):
if round(pre[i][0])== y_tes[i]:
    correct += 1
else:
    wrong += 1
a = correct/len(pre)*100
b wrong/len(pre)*100 print("Correct", a,"\nWroong", b)
```

```
Correct 98.44389844389845
Wroong 1.556101556101556
```

```
import matplotlib.pyplot as plt

left [1, 2]

height [b,a]

tick_label ['Wrong', 'Correct']

plt.barh(left, height, tick_label = tick_label, color = ['red', 'green'])

plt.ylabel('Predictions')

plt.xlabel('Scale 0-100')

plt.title('Predictions Of Model')

plt.show()
```

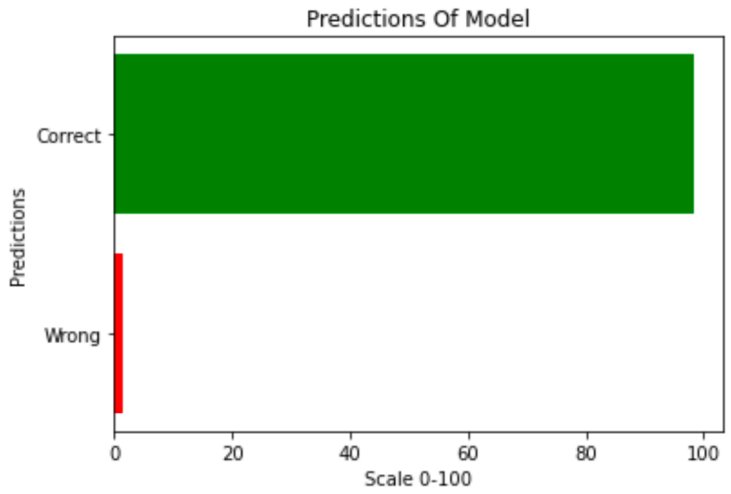

Fig 26: Result

Chapter 04: PERFORMANCE ANALYSIS

4.1 Analysis

Hand signal acknowledgment fills in as a key for defeating numerous challenges and giving accommodation to human existence. The capacity of machines to comprehend human exercises and their significance can be used in a huge swath of uses. One explicit field of interest is gesture based communication acknowledgment. This paper gives a careful survey of cutting edge procedures utilized in ongoing hand motion and gesture based communication acknowledgment research. The methods investigated are appropriately arranged into various stages: information obtaining, pre-handling, division, highlight extraction and grouping, where the different calculations at each stage are explained and their benefits analyzed. Further, we likewise examine the difficulties and impediments looked by motion acknowledgment research as a general rule, just as those restrictive to gesture based communication acknowledgment. Generally, it is trusted that the review might furnish perusers with an exhaustive presentation into the field of robotized motion and gesture based communication acknowledgment, and further work with future exploration endeavors around here.

At first, the recognition rate was only around 59%, then we cleaned the dataset and selected proper images manually to improve the performance of the model.

Experimental result shows the average recognition rate of 89.83% on the test data from the proposed model

4.2 Evaluation Method

We utilized the picture normal strategy for looking at pictures in a dataset. This technique depended on estimating contrasts of pictures. At the point when an info picture was taken care of to the framework the class in the information base that had the most un-normal blunder was the perceived signal .If all pictures of each class in the dataset are comparable, this method accomplishes high precision since picture distinction is insignificant. In the "great" case: when all pictures in each class are something similar, when looking at the information picture the distinction of every pixel is 0 and subsequently the normal blunder is 0. Adding more varieties (various pictures) to the dataset, the blunder increments, and consequently, with high difference, the strategy falls flat. To assess the fluctuation of the datasets and the speculation they accommodate preparing purposes, I utilized the picture normal strategy. Assuming this strategy accomplishes high precision it implies that pictures for preparing are comparable, thus, the variety the dataset offers is restricted.

In any case, assuming the method performs ineffectively it could be a direct result of two reasons:

(1) the dataset isn't right and pictures are not accurately named, or

(2) pictures for each class have high varieties to address each signal

Chapter 05 : CONCLUSION

5.1 Conclusion

A technique for hand signal acknowledgment is presented in this venture. The hand locale is identified from the foundation by the foundation deduction technique. Then, at that point, the palm and fingers are fragmented. Based on the division, the fingers in the hand picture are found and perceived. The acknowledgment of hand signals is refined by a basic standard classifier. The presentation of our technique is assessed on an informational index of 780 hand pictures. The test results show that our methodology performs well and is good for continuous applications.

The presentation of the proposed strategy profoundly relies upon the aftereffect of hand location. Assuming there are moving items with the shading like that of the skin, the articles exist in the aftereffect of the hand identification and afterward corrupt the presentation of the hand motion acknowledgment. Be that as it may, the AI calculations utilized like convolutional neural networks can segregate the hand from the foundation.

With so many individuals relying on sign language to communicate, holding a conference where a handicapped person and an abled person can't communicate and aid each other becomes nearly impossible. It's also worth noting that, unlike spoken languages, which vary considerably from nation to country and are based on various areas, more than 80% of persons with disabilities can communicate and comprehend sign language, even if they come from entirely different places. As a result, a work initiative aiming at overcoming this communication barrier would be extremely beneficial to persons with impairments.

The issue of Sign Language (SL) acknowledgment utilizing pictures is as yet a test. Closeness of signals, client's emphasis, setting and signs with various implications lead to uncertainty. These are a few motivations behind why past work utilized restricted datasets. In this composition, We utilized Random Decision Forests (RDF) and Neural Networks (NN) to direct a fundamental report

about SL acknowledgment utilizing profundity pictures. This review gave data about the requirements of the issue. We reasoned that a dataset to prepare and assess the framework should have adequate motion varieties to sum up every image. We stretched out the normal technique to decide whether a dataset has enough motion varieties and We carried out a Convolutional Neural Network (CNN) for perceiving hand signals in pictures of American Sign Language letter and number images.

5.2 Results

The model is able to predict all the 26 alphabets of sign language properly in real time. The Prediction/Accuracy of the model comes out to be 98.4% . The confusion matrix of the results on the test set is given below which shows the accuracy level of the model.

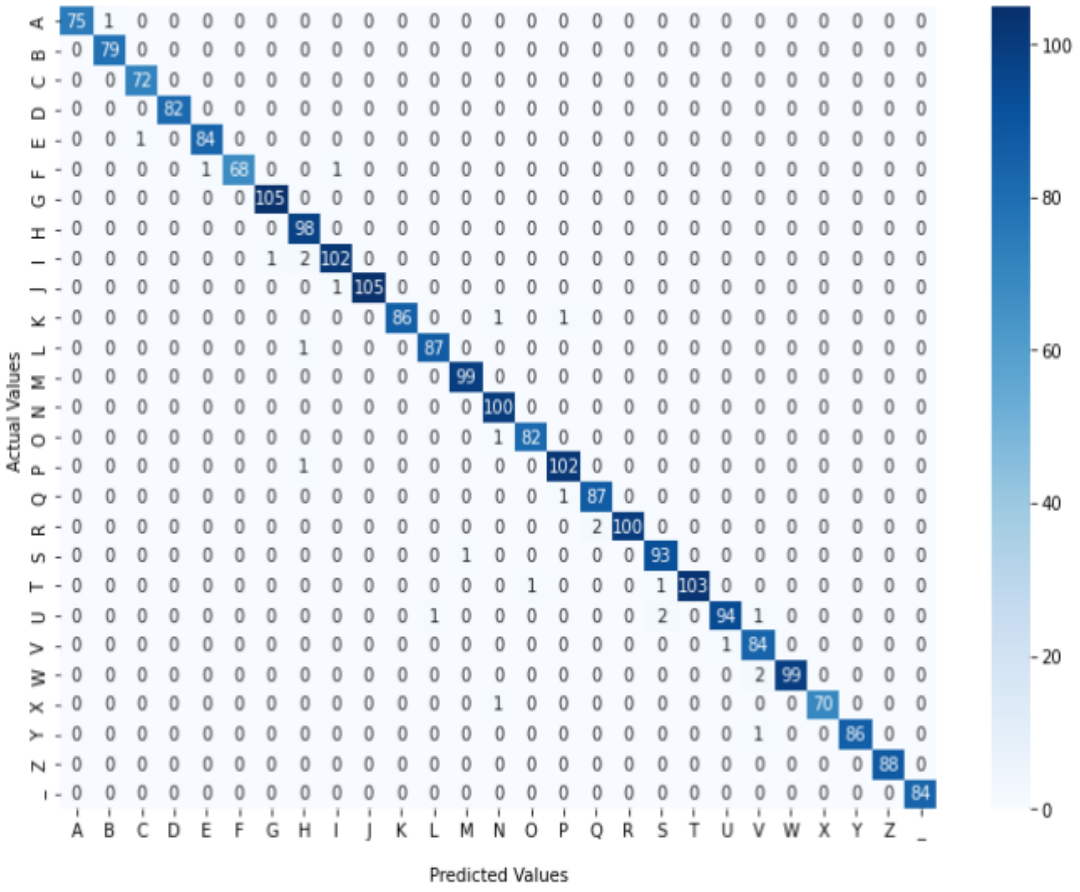

Fig 27: Confusion matrix

5.3 Limitations

At this moment, the model can only recognise static symbols, or motions that do not include any hand motion, and so no video processing is possible. However, the bulk of real-world sign language discussions take place through gestures specified by specific motions. For fast and flawless talks in real life, identifying static gestures is insufficient.

Images of several diverse skin color hues, as well as varied backgrounds and lighting situations, may be added to the dataset to increase the effectiveness of accurately identifying sign language symbols in the model.

References

[1, 2017] Hand Gesture Recognition for Human Computer Interaction by Aashni Haria, Archanasri Subramanian, Nivedhitha Asokkumar, Shrishti Pddar, Jyothi S. Nayak
https://www.researchgate.net/publication/320437730_Hand_Gesture_Recognition_for_Human_Computer_Interaction

[2, 2018] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53]

[3, 2019] Hand Gesture Recognition and Implementation for Disables using CNN'S
https://ieeexplore.ieee.org/abstract/document/8697980

[4, 2020] A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition by Adithya V., Rajesh R. 2020
https://www.sciencedirect.com/science/article/pii/S1877050920312473

[5, 2021] CNN based feature extraction and classification for sign language by Abul Abbas Barbhuiya, Ram Kumar Karsh & Rahul Jain. 2020.
https://link.springer.com/article/10.1007/s11042-020-09829-y

[6] Real time Indian Sign language recognition by Paranjoy Paul, Dr. G N Rathna
http://reports.ias.ac.in/report/19049/real-time-indian-sign-language-recognition