



# **Tribhuvan University**

Institute of Science and Technology

A Project Report on  
**Sign Language Recognition using Deep Convolutional Networks**

*for the partial fulfillment of the requirement for the degree of*  
Bachelor of Science in Computer Science and Information Technology  
**(BSc. CSIT)**

**SUBMITTED BY:**

Sushil Tiwari (20102/075)

Saugat Ghimire (20087/075)

Roshan Basnet (20075/075)

**SUBMITTED TO:**

Patan Multiple Campus

Department of CSIT

Patandhoka, Lalitpur

April 23, 2023

## **SUPERVISOR’S RECOMMENDATION**

I hereby recommend that this project be prepared under my supervision by Sushil Tiwari [20102/075], Saugat Ghimire [20087/075], and Roshan Basnet [20075/075] entitled “Sign Language Recognition using Deep Convolutional Networks” in partial fulfillment of this requirement for the degree of Bachelor in Science in Computer Science and Information Technology (BSc. CSIT) be processed for evaluation

.....

Nawaraj Poudel

Supervisor

## **LETTER OF APPROVAL**

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled “SIGN LANGUAGE RECOGNITION USING DEEP CONVOLUTIONAL NETWORKS” is in partial fulfillment of the requirement for Bachelor in Science in Computer Science and Information Technology (BSc. CSIT) has been well studied. In our opinion, it is satisfactory in the scope and quality of the required degree.

submitted by:

Sushil Tiwari (20102/075)

Saugat Ghimire (20087/075)

Roshan Basnet (20075/075)

## **COPYRIGHT**

The author has agreed that the library, Patan Multiple Campus may make this report freely available for inspection. Moreover, The author has agreed that permission for extensive copying of this report for scholarly purposes may be granted by the supervisor, who supervised the project work recorded herein or, in their absence, by the Head of the Department where this project was done. It is understood that due recognition will be given to the author of this report and the Department of Computer Science and Information Technology, Patan Multiple Campus for any use of the material of this report. Copying or publication or other use of this report for financial gain without the approval of the Department of Computer Science and Information Technology, Patan Multiple Campus, and the authors' written permission is prohibited. Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head

Department of Computer Science and Information Technology

Patan Multiple Campus

Balkhu, Kathmandu, Nepal

## **ABSTRACT**

Sign language is a form of communication commonly used by people with hearing impairment or people with speech impediments. Not all ordinary people understand the language. The translation of sign language into the alphabet/text automatically will facilitate the communication of the deaf with ordinary people. In recent years, deep convolutional networks (DCNs) have shown promising results in sign language recognition, owing to their ability to learn and extract hierarchical features from the input data. This report explores the application of DCNs in sign language recognition and presents an overview of various DCN architectures used in this domain. The report also highlights the challenges and limitations of DCNs in sign language recognition and discusses future research directions in this field. Finally, the report provides a detailed evaluation of the performance of DCNs on different sign language datasets, which demonstrates their effectiveness in accurately recognizing sign language gestures.

Keywords: Sign language, Convolutional network, DCN, datasets

## **List of Abbreviations**

**DCNs** ..... Deep convolutional networks

**DCN**..... Deep convolutional network

**ASL** ..... American Sign language

**HCI** ..... Human-computer interaction

**ONEIROS** ..... Open-finished Neuro-Electronic Intelligent Robot Operating System

**RNNs** ..... Recurrent neural networks

# Table of Contents

<b>CHAPTER I .....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>1</b>
Background .....	1
Objective .....	2
Applications .....	2
1.5. Feasibility Analysis .....	3
1.5.1. Economic Feasibility .....	3
1.5.2. Technical Feasibility .....	3
1.5.3. Operation Feasibility.....	3
1.5.4. Project Timeline.....	3
1.6. System Requirements .....	4
1.7. Technologies Used .....	4
<b>CHAPTER II.....</b>	<b>11</b>
<b>LITERATURE REVIEW .....</b>	<b>11</b>
2.1. Related Works .....	11
<b>CHAPTER III .....</b>	<b>14</b>
<b>SYSTEM DESIGN AND ARCHITECTURE .....</b>	<b>14</b>
3.1. Block Diagram .....	14
3.2. Use Case Diagram.....	15
3.3. Data Flow Diagram .....	16
3.4. Sequence Diagram.....	17
<b>CHAPTER IV.....</b>	<b>18</b>
<b>METHODOLOGY .....</b>	<b>18</b>
4.1. Dataset Collection .....	18
4.2. Algorithm Used .....	19
4.3. Testing and Verification.....	22
<b>CHAPTER V .....</b>	<b>23</b>
<b>RESULTS AND DISCUSSION .....</b>	<b>23</b>
5.1. Output.....	23
5.2. Work Completed .....	26
5.3. Limitations .....	26
5.4. Problems Faced .....	26
<b>CHAPTER VI.....</b>	<b>27</b>

<b>CONCLUSION AND FUTURE ENHANCEMENTS .....</b>	<b>27</b>
6.1. Conclusion.....	27
6.2. Future Enhancements .....	27
<b>CHAPTER VII.....</b>	<b>28</b>
<b>REFERENCES.....</b>	<b>28</b>



# CHAPTER I

## INTRODUCTION

### **Background**

The world can't exist without correspondence, whether or not it appears as contact, discourse, or visual articulation. Text and visual articulations work with correspondence between the hard of hearing and the quiet. Hands and facial highlights are exceptionally critical in offering human viewpoints in confidential correspondence. Various mechanical upgrades and much examination have been directed to help not-too-sharp people. Profound learning and PC vision can likewise be used to affect this reason.

If an individual can't talk or hear, gesture-based communication is the main method for correspondence accessible to them. Fingerspelling is a vital tool in sign language, as it enables the communication of names, addresses, and other words that do not carry meaning in word-level association [1].

Gesture-based communication permits provoked people to offer their viewpoints and sentiments. In this paper, a special gesture-based communication recognizable proof strategy for distinguishing letter sets and movements in communication through signing is proposed.

The problem we are investigating is sign language recognition through unsupervised feature learning. Many systems are developed for sign language recognition and there is no exact system for recognizing the complete signs [2]. Being able to recognize sign language is an interesting computer vision problem while simultaneously being extremely useful for deaf people to interact with people who don't know how to understand American Sign Language (ASL) [3].

## **1.1. Problem Statement**

There are various dumb and deaf people have their form of expressing their feelings through signs. It is very difficult for normal people to understand the exact content of symbolic expressions of these people. It is a very challenging job that has created a communication barrier in real life. Technology is very fast growing and incredible, yet there is not much technological development and improvement for dumb deaf people. So, the purpose of our project is aimed to prevent the misconception and enhance communication and harmony between different types of people

### **Objective**

The objective of our project is to analyze different sign gestures to eradicate the communication barrier between dumb-deaf people and normal people.

### **Applications**

The ability of a computer or machine to understand hand gestures is the key to unlocking numerous potential applications. Potential application domains of gesture recognition system are as follows:

1. Sign language recognition—Communication medium for the deaf. It consists of several categories: fingerspelling, isolated words, the lexicon of words, and continuous signs.
2. Robotics and Tele-robotic—Actuators and motions of the robotic arms, legs, and other parts can be moved by simulating a human's action.
3. Games and virtual reality—Virtual reality enables realistic interaction between the users and the virtual environment. It simulates the movement of users and translates the movement into the 3D world.
4. Human-computer interaction (HCI)—Includes application of gesture control in the military, medical field, manipulating graphics, design tools, annotating or editing documents

## 1.5. Feasibility Analysis

### 1.5.1. Economic Feasibility

We have all the resources available to do the project thus further expenses are not required.

### 1.5.2. Technical Feasibility

It is technically feasible to develop systems for recognizing sign language. Several approaches have been successful in achieving good results in sign language recognition, including the use of computer vision techniques, machine learning algorithms, and wearable devices.

### 1.5.3. Operation Feasibility

It is concerned with the operating capabilities of the system. This project requires a computer with decent computing capabilities. At least 8 Gigabytes of RAM and a modern CPU are required.

### 1.5.4. Project Timeline

Typically, Schedule feasibility is the estimate of how long the application of the system has taken to develop. The project is scheduled as feasible if it can be completed using some methods like a payback period.

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	11th	12th
Study and Analysis	4 weeks										
Data Collection				2 weeks							
Implementation					4 weeks						
Testing					4 weeks						
Documentation									3 weeks		
Review											
Presentation										2 weeks	

Figure: 1.1: Gantt Chart for overall project completion

## 1.6. System Requirements

### 1. Hardware Specification

- 3 GB of free drive
- 8 GB of RAM
- Quad Core CPU/GPU

### 2. Software Specification

- Operating System – Windows/MAC/LINUX
- Python - For creating source code and algorithm

## 1.7. Technologies Used

### Python

Python is a high-level, interpreted programming language that is known for its simplicity, readability, and versatility. It was first released in 1991 by Guido van Rossum and has since become one of the most popular programming languages in the world. Python is used for a wide range of applications, including web development, data analysis, machine learning, artificial intelligence, scientific computing, and more. Python supports multiple programming paradigms, including object-oriented, procedural, and functional programming. It has a dynamic type system and automatic memory management, which makes it very easy to write and debug code.

Python can be run on a wide range of platforms, including Windows, macOS, Linux, and many others. It also has a large number of third-party libraries and frameworks that make it easy to accomplish complex tasks with minimal coding. Some popular libraries and frameworks include NumPy, Pandas, Django, Flask, PyTorch, TensorFlow, and many others.

### Opencv

OpenCV (Open Source Computer Vision Library) is an assortment of programming capacities designed for the most part for ongoing PC vision. At first, made by Intel, it was subsequently upheld by Willow Garage and Itseez (which Intel accordingly bought). The library is stage autonomous and allowed to use under the BSD open-source permit.

OpenCV's application regions include:

2D and 3D element tool compartments

Egomotion assessment

Facial acknowledgment framework

Signal acknowledgment

Human-PC Association (HCI)

Versatile mechanical technology

Movement getting it

Object recognizable proof

Division and acknowledgment Stereopsis sound system vision: profundity insight from 2 cameras

Structure from movement (SFM).

Movement following

OpenCV was formally begun in 1999 as an Intel Research program to foster CPU-concentrated applications, as a feature of a progression of tests that included constant beam following and 3D presentation dividers. Various streamlining experts from Intel Russia and Intel's Performance Library Team were among the essential supporters of the venture. In the earliest long stretches of OpenCV, the venture's targets were expressed as:

Advance vision research by giving code for key vision foundation that isn't simply open yet additionally streamlined. Don't bother wasting time.

Give a standard establishment for designers to expand on, so that code is all the more promptly reasonable and adaptable, thus spreading consciousness of the objective.

Advance vision-based business applications by making compact, execution-enhanced code unreservedly accessible — with a permit that needn't bother with the source code to be open or free.

OpenCV is created in C++ and its essential connection point is in C++, however, it likewise holds a huge, yet less exhaustive, C connection point. Every single new progression and calculation is reflected in the C++ interface. There are Python, Java, and MATLAB/OCTAVE ties. You might track down the API for these connection points in the web-based documentation. Coverings in various programming dialects have been made to work with more prominent use. In variant 3.4, JavaScript ties for a subset of OpenCV capacities were made accessible for web stages as OpenCV.js.

## **Keras**

Python-based Keras is an open-source library for brain organizations. TensorFlow, Microsoft Cognitive Toolkit, R, Theano, and PlaidML are viable with it. Ease of use, measured quality, and extensibility are focused on to advance speedy trial and error with profound brain organizations. It was created as a feature of the ONEIROS (Open-finished Neuro-Electronic Intelligent Robot Operating System) research venture, and Google engineer Francois Chollet is its essential originator and maintainer. Chollet is likewise the engineer of the Xception model of a profound brain organization. Highlights: Keras contains various executions of regularly utilized brain network building blocks, for example, layers, goals, enactment capacities, and enhancers, as well as an assortment of devices that make it simpler to work with picture and text information to improve on the 11 codings expected for composing profound brain network code. The source code is accessible on GitHub, and the local area support discussions are a GitHub bugs page and a Slack channel. As well as supporting convolutional and intermittent brain organizations, Keras additionally upholds exemplary brain organizations. Standard utility layers like dropout, group standardization, and pooling are upheld. Keras empowers the sending of profound models on cell phones (iOS and Android), the web, and the Java Virtual Machine. Furthermore, it empowers the disseminated preparation of profound learning models on groups of GPUs and TPUs, ordinarily related to CUDA. The Keras applications module is utilized to give pre-prepared models to profound brain organizations. Keras models are utilized for determining, highlighting extraction, and calibrating. This part expounds on the utilization of the Keras application structure. A prepared model comprises model Architecture and model Weights. Since model loads are huge records, we should download the ImageNet data set and concentrate on the component. The following are the absolute most well-known pre-prepared models.

ResNet

VGG16

MobileNet

InceptionResNetV2

InceptionV3

The open-source bundle Keras gives a Python connection point to fake brain organizations. Keras fills in as the TensorFlow library's connection point.

Keras upheld numerous backends before variant 2.3, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML. As of adaptation 2.4, support is restricted to TensorFlow alone. It focuses on ease of use, seclusion, and extensibility to work with quick trial and error with profound brain organizations. It was created as a component of the undertaking ONEIROS (Open-finished Neuro-Electronic Intelligent Robot Operating System) research try, and its important maker and maintainer are Google engineer Francois Chollet. Chollet is likewise the maker of the profound brain network model Xception.

## **NumPy**

NumPy is a Python library that adds support for huge, complex exhibits and grids, as well as an immense number of undeniable level numerical capacities to deal with these clusters. Jim Hugunin and various others constructed the ancestor of NumPy, Numeric, adaptation 12, before all else. In 2005, Travis Oliphant incorporated NumPy by blending parts of the opponent Numarray library into the Numeric library. NumPy is an open-source program with a few supporters. Highlights: NumPy focuses on the CPython reference execution of Python, which is a mediator that doesn't enhance bytecode. Commonly, numerical calculations intended for this variant of Python execute considerably more leisurely than their gathered partners. NumPy settles the gradualness issue to some degree by offering multi-faceted clusters, capacities, and administrators that work really on exhibits, requiring the modifying of specific code, for the most part, inward circles, written in different dialects. NumPy in Python gives an ability identical to MATLAB since both are deciphered and permit the client to build quick projects as long as most of the activities are performed on clusters or frameworks as opposed to scalars.

NumPy is normally incorporated with Python, a more current and complete programming language, while MATLAB has a huge assortment of strengthening tool kits, most quite Simulink. SciPy is a library that adds further MATLAB-like capacity, while Matplotlib is a plotting device that offers MATLAB-like plotting usefulness. Both MATLAB and NumPy depend on BLAS and LAPACK for straight polynomial math tasks. NumPy exhibits are used by the broadly utilized PC vision programming OpenCV's Python ties to store and control information. Since pictures with a few channels are put away as three-layered clusters, ordering, cutting, and concealing with different exhibits are exceptionally productive strategies for accessing specific pixels inside a picture.

The NumPy cluster as the widespread information structure in OpenCV for pictures, removed highlight focuses, channel parts, and a few different information types radically work on the programming process and troubleshooting. Constraints: Inserting or adding components to an exhibit is more troublesome than utilizing Python's rundowns. The exhibit augmentation system `np.pad(...)` creates new clusters with the predetermined structure and cushioning values, moves the given cluster into the new exhibit, and returns the outcome. The `np.concatenate([a1,a2])` strategy in NumPy doesn't really link the two clusters, yet rather returns another exhibit with the components from the two clusters in 13 successions. The components of a cluster may just be reshaped utilizing `np.reshape(...)` if the quantity of exhibit individuals doesn't change. Because of the way that NumPy clusters should be seen on nonstop memory cushions, certain circumstances emerge.

## **Neural Networks**

A neural network is an assortment of calculations that looks to track down information's hidden connections by repeating the human mind. Brain networks in this setting allude to either natural or counterfeit neuronal frameworks. Brain organizations can adjust to various information sources, permitting them to deliver the best outcome without a reexamination of the result standards. Starting with computerized reasoning, brain networks are acquiring enormous fame in the advancement of exchanging frameworks. A brain network looks like the brain network tracked down in the human cerebrum. A "neuron" in a brain network is a numerical capacity that gathers and sorts input by a predefined design. Amazingly, the organization looks like measurable techniques, for example, bend fitting and relapse investigation.



A neural network is made out of connected hub layers. Like various direct relapses, every hub is a perceptron. The perceptron takes care of the sign got by rehashed straight relapse to a possibly nonlinear initiation work. Perceptrons in a multifaceted perceptron (MLP) are put in layers that are connected. Input designs are gathered by the information layer. Input examples might be planned to the result layer's arrangements or result signals. The information loads are adjusted through secret layers until the brain organization's mistake edge is little. It is proposed that secret layers extrapolate conspicuous info information attributes that have the prescient potential for yields. This makes sense of element extraction, which fills a comparable need as measurable methodologies like head part examination. Spaces of Utilization coming up next are a portion of the uses of ANN. It infers that ANN's turn of events and applications utilize a multidisciplinary approach. Discourse Recognizability Speech plays a significant capacity in human contact. Thus, it is typical for people to expect spoken communications with PCs. People utilize challenging-to-learn, complex dialects to speak with innovation in the current day. Utilizing machine-reasonable communication in language may be a direct method for conquering this correspondence obstruction. Critical advancement has been accomplished around here, albeit such frameworks battle with confined jargon or syntax, as well as the trouble of retraining the framework for changed speakers and settings. ANN assumes a critical part in this field.

## **Deep Learning**

Profound Learning: Deep-gaining brain networks are recognized from single-stowed away layer brain networks by their profundity, or the number of hub layers input should move through in a multistep design acknowledgment process. Prior renditions of brain organizations, for example, the earliest perceptrons, included just a single secret layer between the information and result layers. Profound learning comprises multiple layers (counting info and result). So significant isn't simply a popular expression used to give the appearance that PCs read Sartre and pay attention to cloud groups. An expression with an exact definition connects with various covered layers. In profound learning organizations, each layer of hubs is prepared on a particular arrangement of traits given the result of the former layer. The more layers you add to a brain organization, the more complicated the properties its hubs can perceive, because each layer incorporates and recombines the information from the one beneath. Include ordered progression is a rising intricacy and reflection progressive system. It empowers profound

learning organizations to handle huge, high-layered informational indexes with billions of nonlinear boundaries.

In particular, counterfeit brain networks can find stowed away designs inside unlabeled, unstructured information, which frames the extraordinary greater part of the world's information. Unstructured information, which incorporates photos, texts, video, and sound accounts, is alluded to as crude media. Profound learning is especially skilled at examining and gathering the world's crude, unlabeled media, finding shared traits and peculiarities in material that has never been organized in a social data set or named by an individual. Profound learning can, for example, bunch 1,000,000 photos in light of their likenesses: cats in a single district, conversation starters in another, and photographs of your grandma in a third. This lays the preparation for smart photograph collections. Profound learning organizations, dissimilar to most customary AI calculations, naturally remove highlights without human mediation. Considering that highlight extraction can take groups of information researchers years to achieve, profound learning is a method for dodging the lack of expertise. It upgrades the capacities of small information science groups, which can't scale by their actual nature. While preparing unlabeled information, every hub layer in a profound organization consequently learns highlights by persistently endeavoring to remake the contribution from which it takes its examples and limiting the distinction between its expectations and the likelihood conveyance of the info information. In this design, limited Boltzmann machines, for example, produce alleged recreations. Simultaneously, these brain networks figure out how to find relationships between's particular key perspectives and optimal results — they lay out joins between highlight signs and what those signs show, whether or not a full reproduction or named information is being utilized. A profound learning network that has been prepared on marked information may hence be applied to unstructured information, giving it admittance to undeniably more contribution than AI organizations.

## CHAPTER II

### LITERATURE REVIEW

Sign language is a vision-based language that uses an amalgamation of a variety of visuals like hand shapes and gestures, orientation, locality, movement of hand and body, lip movement, and facial expressions. Like spoken language, regional variants of sign language also exist, e.g., Indian Sign Language (ISL), American Sign Language (ASL), and Portuguese Sign Language. There are three types of sign languages: spelling each alphabet using fingers, sign vocabulary for words, using hands and body movement, facial expressions, and lip movement. Sign language can also be isolated as well as continuous. In isolated sign language, people communicate using gestures of a single word, while continuous sign language is a sequence of gestures that generate a meaningful sentence. A list of some work which is performed in the field of sign language to text translation is as:

#### 2.1. Related Works

- Translation of Sign Language Into Text Using Kinect for Windows v2. This model proposes methods to recognize and translate dynamic gestures of the German Sign Language (Deutsche Gebärdensprache, DGS) into text using Microsoft Kinect for Windows v2. Two approaches were used for the gesture recognition process: sequence matching using the Dynamic Time Warping algorithm and a combination of Visual Gesture Builder along with Dynamic Time Warping. For benchmarking purposes, eleven DGS gestures, which were provided by an expert user from Germany, were taken as a sample data set. The proposed methods were compared based on the computation cost and accuracy of these gestures. The computation time for Dynamic Time Warping increased steadily with an increasing number of gestures in the data set whereas in the case of Visual Gesture Builder with Dynamic Time Warping, the computation time remained almost constant. However, the accuracy of Visual Gesture Builder with Dynamic Time Warping was only 20.42% whereas the accuracy of Dynamic Time Warping was 65.45%. Based on the results, we recommend the Dynamic Time Warping algorithm for small data sets and Visual Gesture Builder with Dynamic Time Warping for large data sets [4].
- American Sign Language Translation Using Edge Detection and Cross Co-relation  
This project is to implement an automated translation system that is capable of

translating ASL to English text using common computing environments such as a computer and a generic webcam. In this project, a real-time hand gesture recognition system using a combination of image processing modalities is implemented. A prototype graphical user interface application for ASL sign capture, processing, collection, and analysis is presented. The approach consists of a gesture extraction phase followed by a gesture recognition phase. An image gesture database is collected through the application and used as training information to be used in the gesture recognition stage. This model aims to provide two different translation paradigms:

- English Characters (alphabet)
- Complete words or phrases

In the method to recognize individual characters, the hand gesture image is processed by combining image segmentation and edge detection to extract morphological information and then processed by the gesture detection stage that recognizes the corresponding alphabet letter

In this feature selection stage, a subset of frames that can represent a particular word or phrase is selected. The collection of frames representing a word or a phrase is then processed using the multi-modality technique used for processing individual characters. Finally, the gesture recognition stage is applied to both approaches using a cross-correlation coefficient-based scheme to detect the expression [5].

- Ronchetti et al. [6] discussed an image processing-based method for extraction of descriptors followed by a hand shape classification using ProbSom which is a supervised adaptation of self-organizing maps. Using this technique, they were able to achieve an accuracy of above 90% in Argentinean Sign Language. The classification technique they used was based on eigenvalue-weighted Euclidean distance. They identified 24 different alphabets of Indian Sign Language with an accuracy of 96%. Kumud and Neha [8] proposed a method for recognizing gestures from a video containing multiple gestures of Indian Sign Language. They extracted the keyframe, based on gradient, to split the video into independently isolated gestures. The features were extracted from gestures by applying Orientation Histogram and Principal Component Analysis.

Lionel et al. [6] proposed a system to recognize Italian sign language gestures. They used Microsoft Kinect and convolution neural network (CNN) accelerated via a graphic processing unit (GPU). They achieved a cross-validation accuracy of around 92% on a data set consisting of 20 Italian gestures. Rajat et al. [7] proposed a finely tuned portable device as a solution to alleviate this problem of minimizing the communication gap between normal and differently-abled people.

## CHAPTER III

### SYSTEM DESIGN AND ARCHITECTURE

#### 3.1. Block Diagram

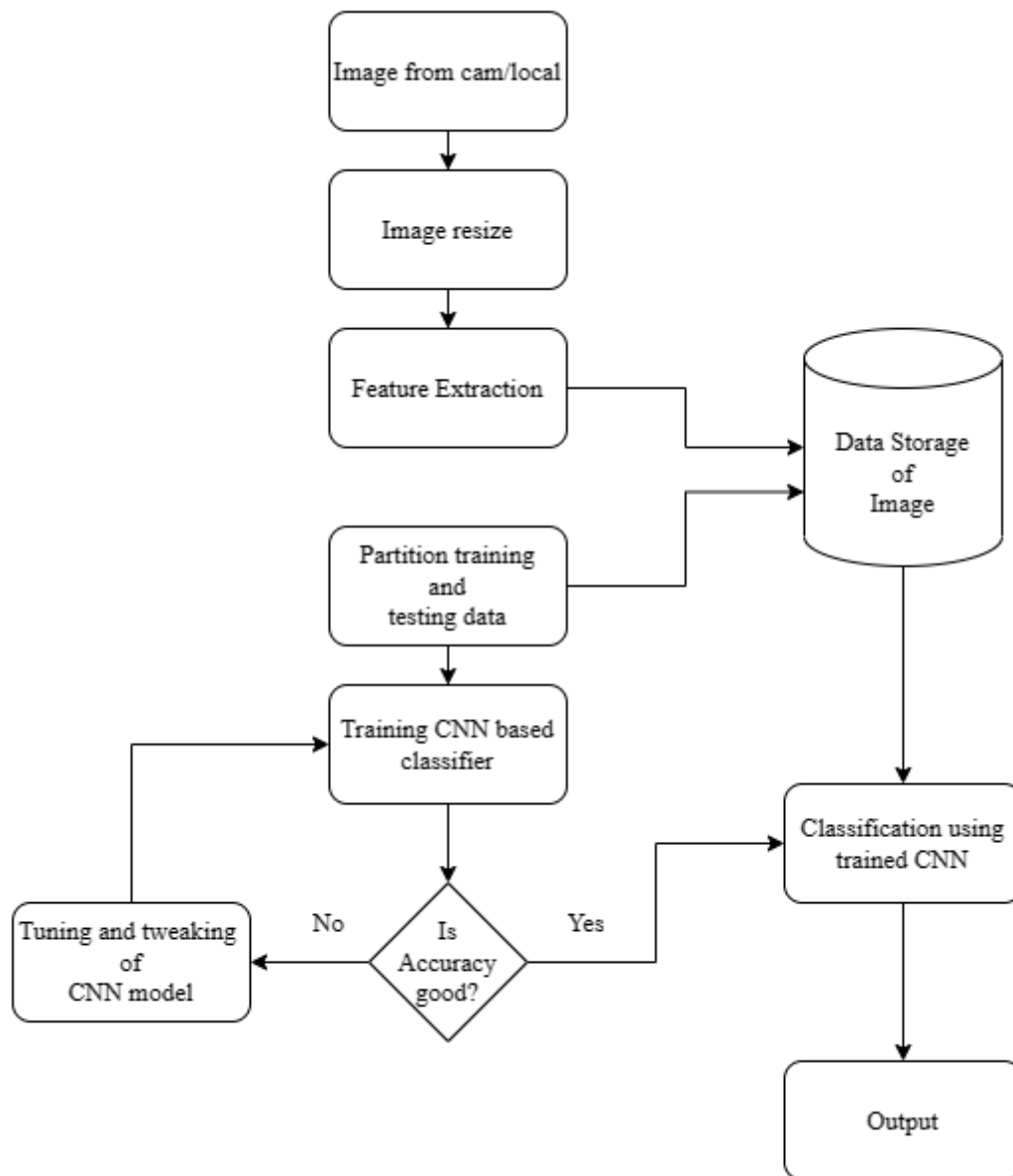


Figure: 3.1: Block diagram of data collection, testing, training, and classification

### 3.2. Use Case Diagram

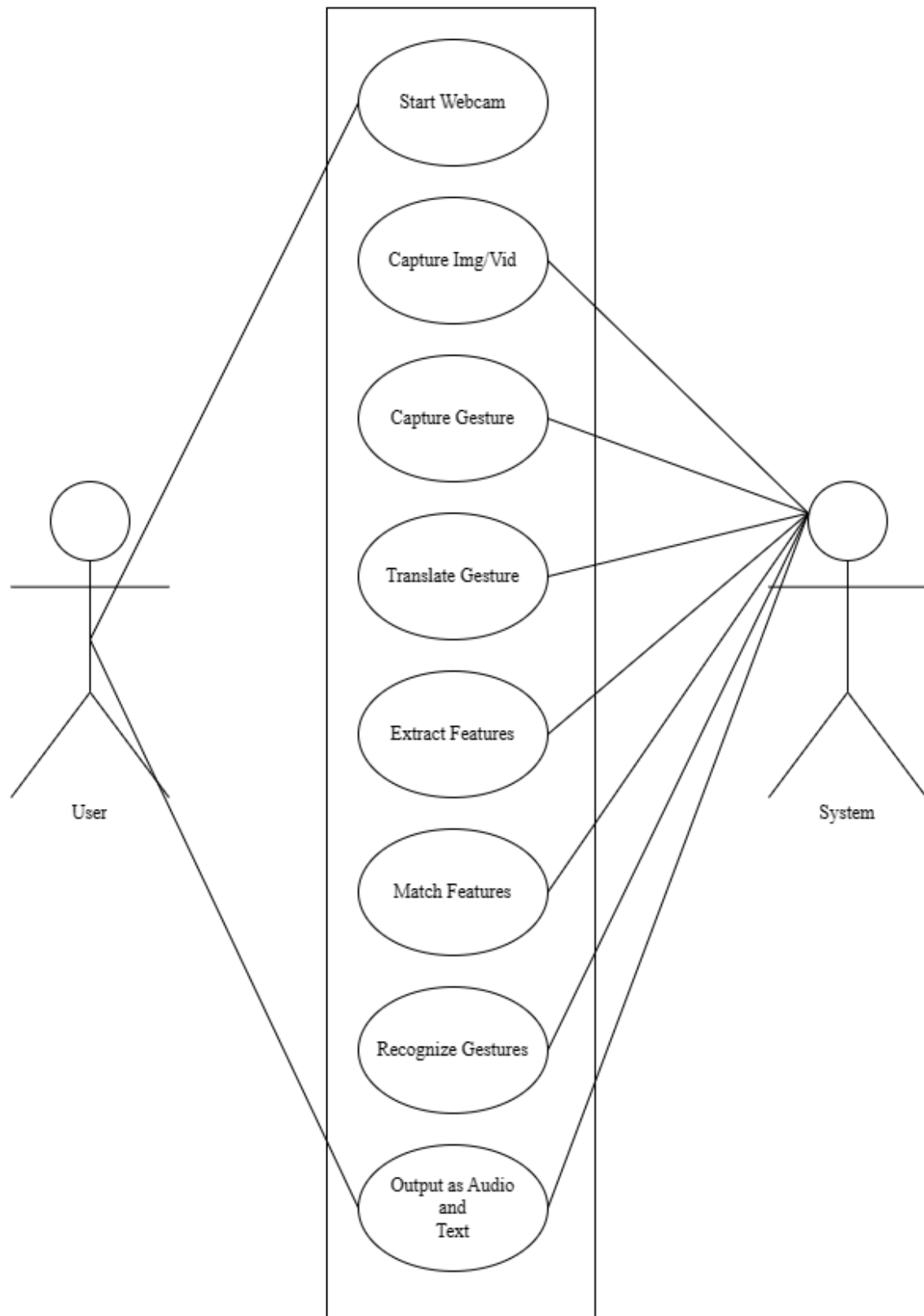


Figure: 3.2: Use-case diagram of Sign Language Recognition

### 3.3. Data Flow Diagram

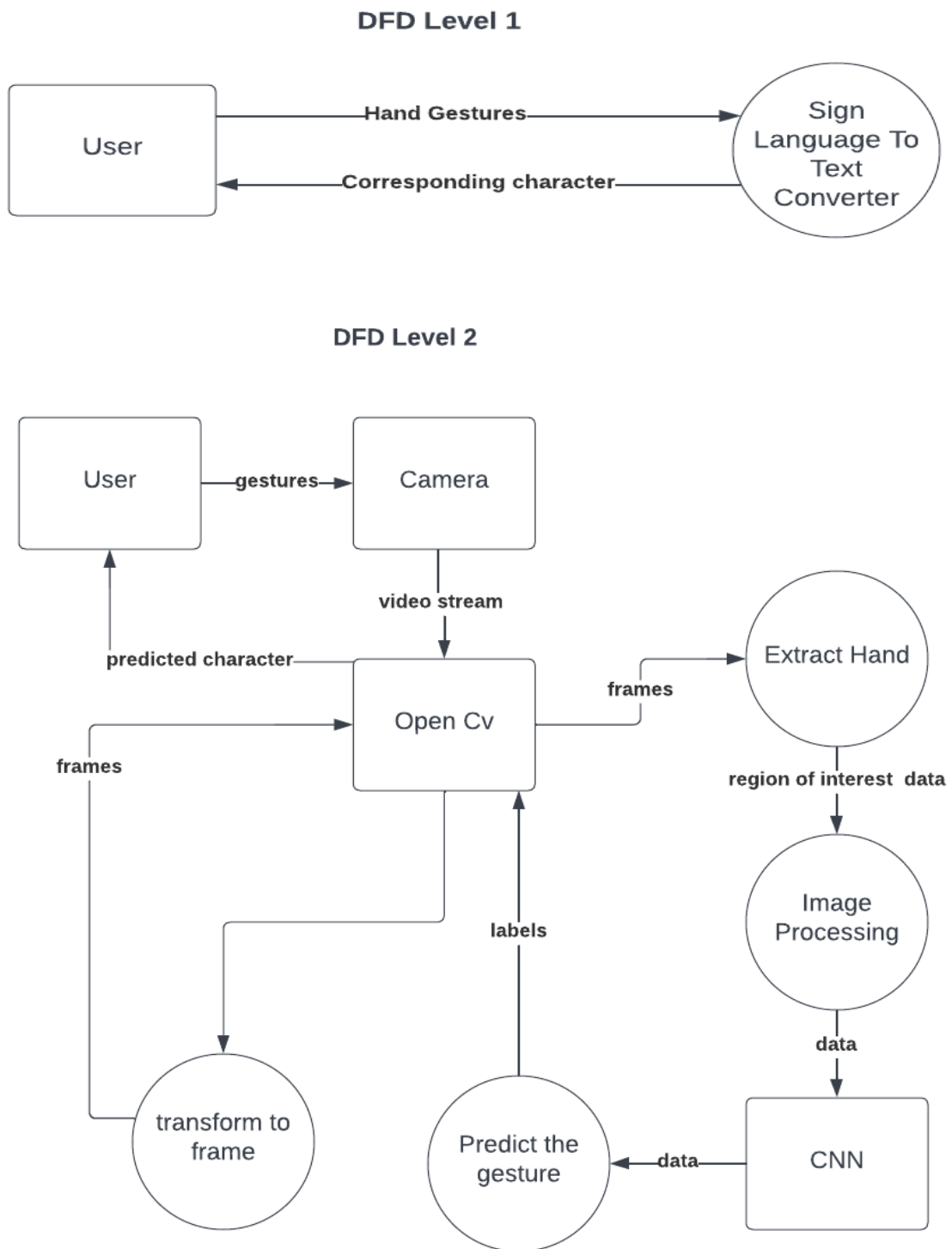


Figure: 3.3: Data flow diagram of Sign Language Recognition



### 3.4.Sequence Diagram

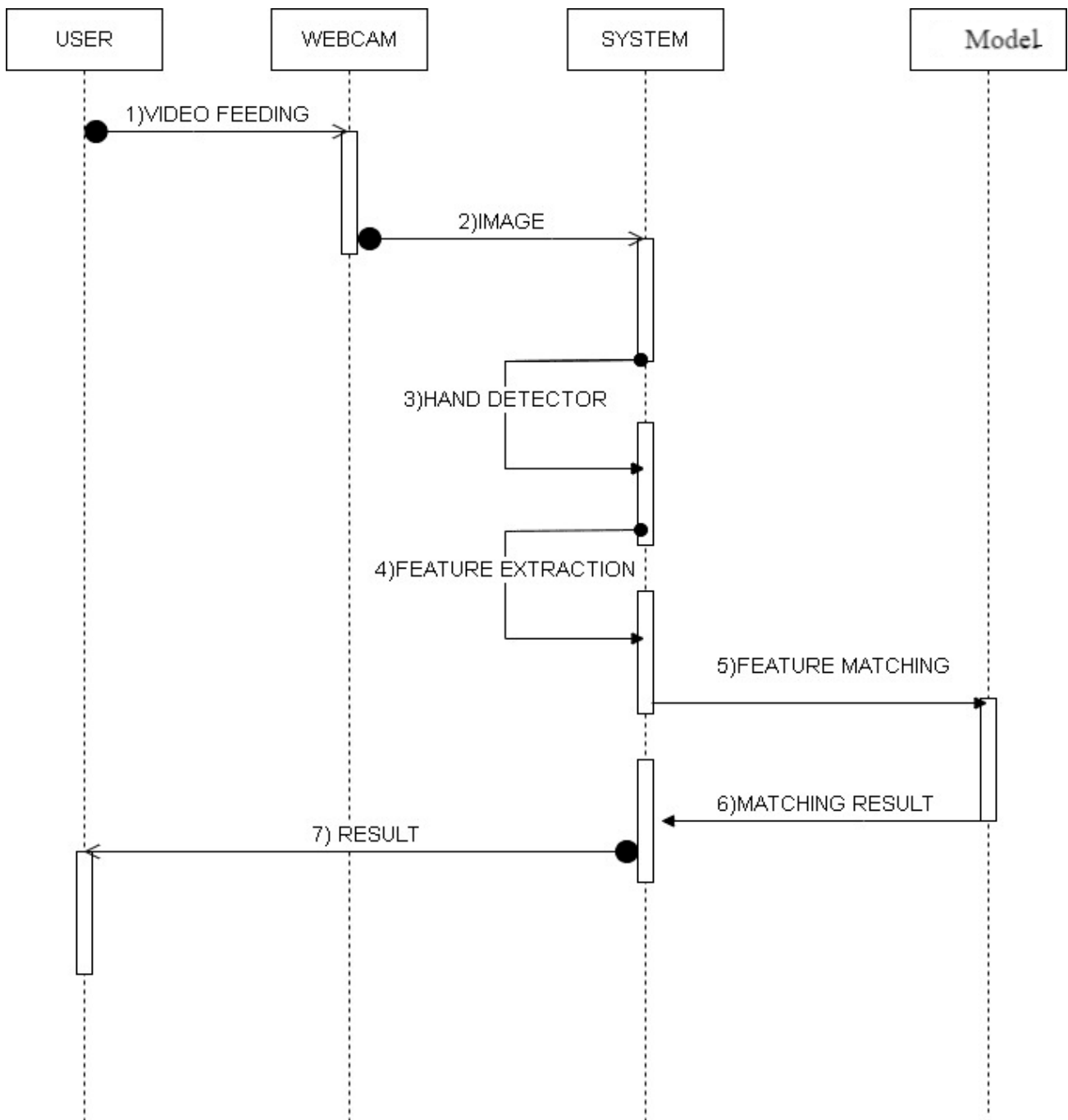


Figure: 3.4: Sequence Diagram of Sign Language Recognition

## CHAPTER IV

### METHODOLOGY

#### 4.1. Dataset Collection

The dataset for American Sign Language (ASL) recognition was obtained by capturing images of hand gestures representing every alphabet using the Mediapipe hand tracking library. The hand-tracking library provided a fast and reliable way to detect hand landmarks and track hand movements in real time. The captured hand gestures were then mapped onto a white background image, which provided a consistent and neutral background for image processing and classification. The resulting dataset can be used to train machine learning models for ASL recognition and assistive technology applications.

Mediapipe hand tracking library uses a deep learning-based approach to detect and track human hands in real time. The library is based on a lightweight convolutional neural network (CNN) model, which is trained on millions of annotated hand images to learn the hand landmarks and their connections. The network is optimized for efficiency and can run on mobile devices and desktop computers.

The library first detects the hand regions in the input image using a bounding box regression algorithm. It then feeds the detected hand regions to the hand landmark model to estimate the landmarks of each hand. The hand landmarks are a set of 21 2D points that represent the joints and fingertips of the hand.

The hand landmark model is a feed-forward neural network, which takes the detected hand region as input and produces the landmarks as output. The model is trained using a combination of synthetic and real data to generalize to different hand shapes, skin colors, and lighting conditions.

Once the hand landmarks are detected, the library can use them for various applications, such as gesture recognition, hand pose estimation, and augmented reality. The library also provides a set of utilities for visualizing and processing the hand landmarks, such as drawing hand annotations, calculating hand features, and filtering noisy landmarks.

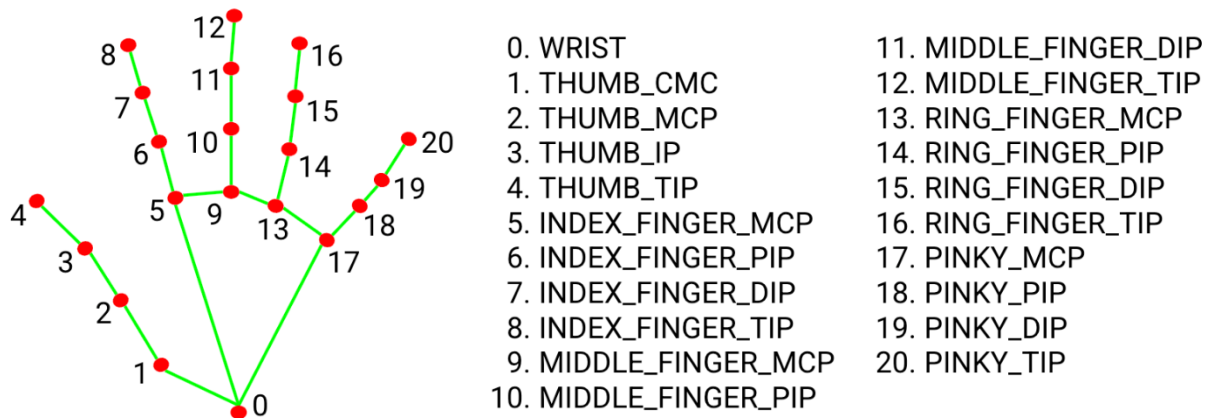


Figure: 4.1: Mediapipe's landmark system

## 4.2. Algorithm Used

### Convolutional Neural Network (CNN):

A Convolution Neural Network (ConvNet/CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area. Convolutional neural networks or ConvNets are great at capturing local spatial patterns in the data. They are great at finding patterns and then using those to classify images. ConvNets explicitly assume that input to the network will be an image. CNNs, due to the presence of pooling layers, are insensitive to the rotation or translation of two similar images; i.e., an image and its rotated image will be classified as the same image. Due to the vast advantages of CNN in extracting the spatial features of an image, we have used the Inception-v3 [8] model of the TensorFlow [9] library which is a deep ConvNet to extract spatial features from the frames of video sequences. Inception is a huge image classification model with millions of parameters for images to classify.

## Convolutional Layer:

In the convolution layer, I have taken a small window size [typically of length  $5 \times 5$ ] that extends to the depth of the input matrix.

The layer consists of learnable filters of window size. During every iteration, I slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position.

As I continue this process 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.

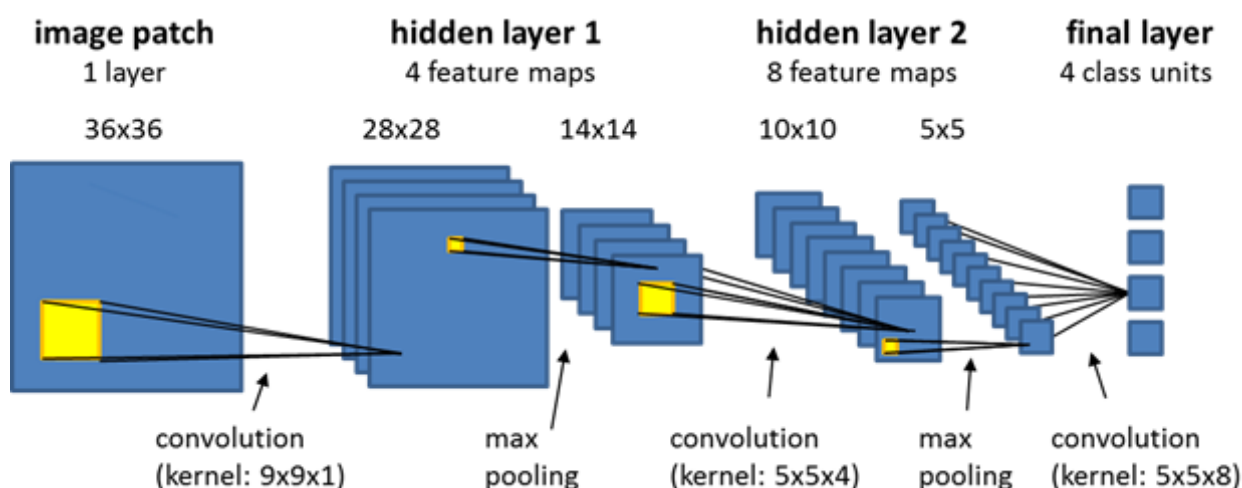


Figure: 4.2: Different convolutional layers

## Pooling Layer:

We use a pooling layer to decrease the size of the activation matrix and ultimately reduce the learnable parameters.

There are two types 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 an average of all Values in a window.

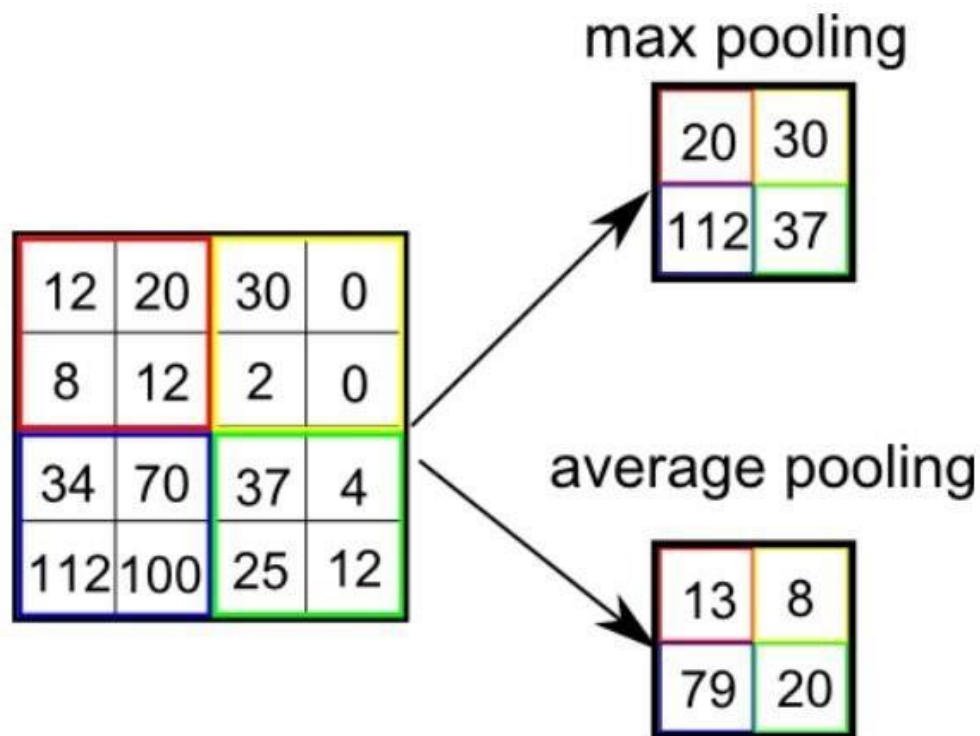


Figure: 4.3: Average pooling and max pooling

### Fully Connected Layer:

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

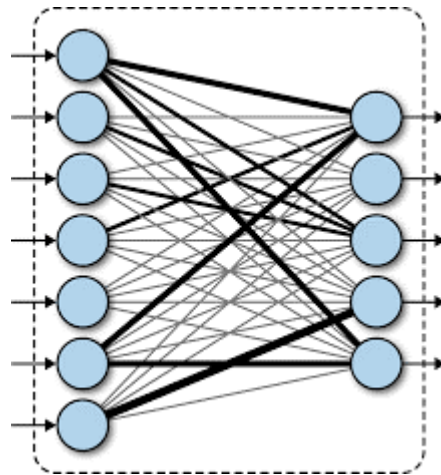


Figure: 4.4: Dense or fully connected layer

### c. Final Output Layer:

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

After getting values from the fully connected layer, we connect them to the final layer of neurons [having a count equal to a total number of classes], which will predict the probability of each image being in different classes.

## 4.3. Testing and Verification

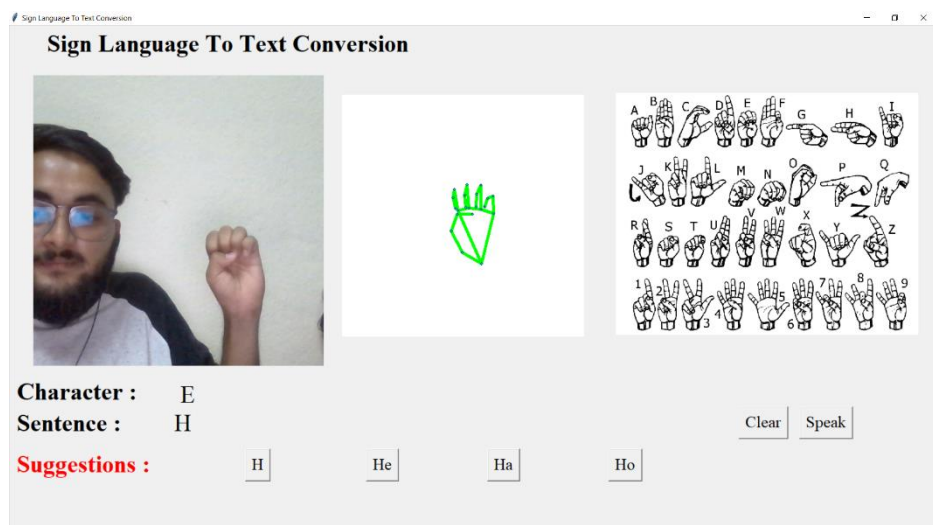
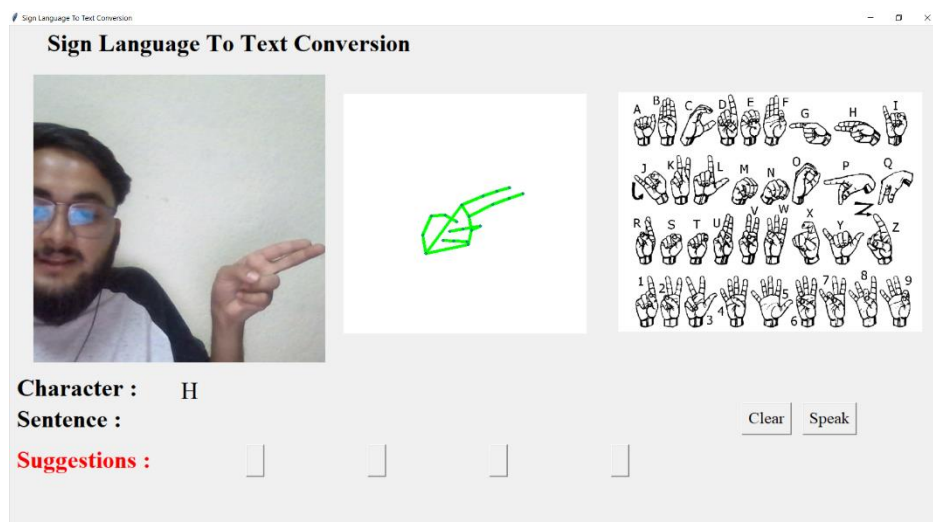
A fraction of the dataset will be used for testing and verification. Once the CNN has been trained, it is tested on a separate dataset (called the test set) to evaluate its performance on data that it has not seen before. This helps to ensure that CNN can generalize its knowledge to new data.

## CHAPTER V



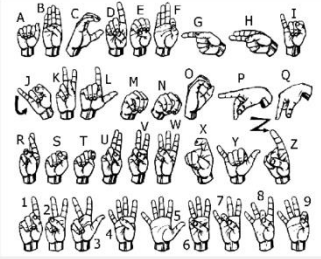
## RESULTS AND DISCUSSION

### 5.1. Output

The output of the project is a machine learning model that can recognize American Sign Language (ASL) alphabets from images of hand gestures and text-to-speech with word suggestions . The model achieved an accuracy of 95% on the test dataset.




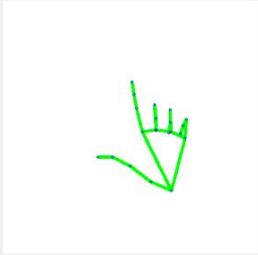
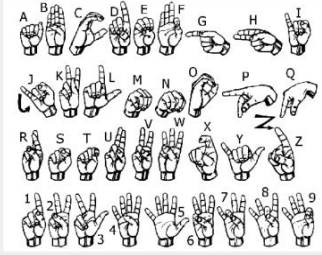
Sign Language To Text Conversion

Character : L  
Sentence : HE

Suggestions :


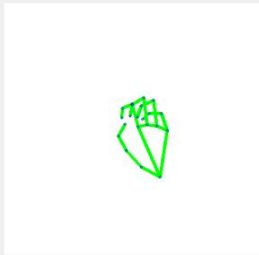
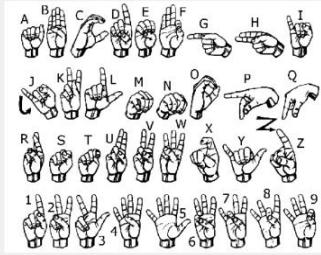
Sign Language To Text Conversion

Character : L  
Sentence : HEL

Suggestions :

Sign Language To Text Conversion

Character : O  
Sentence : HELL

Suggestions :



The graph generated during training with corresponding Model Accuracy and confusion matrix are as shown in figure

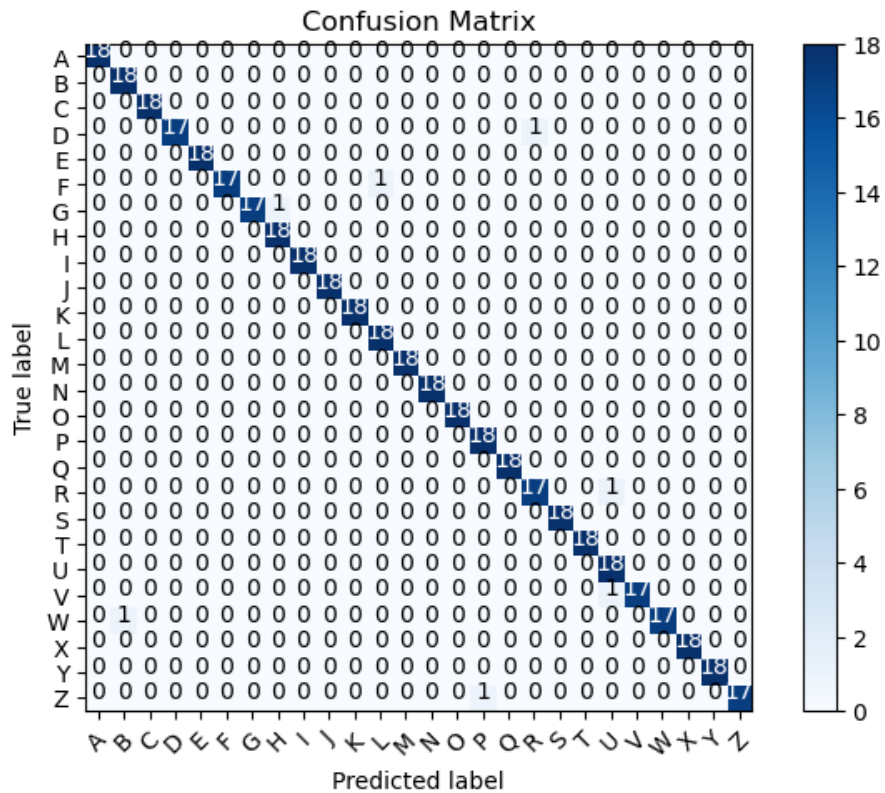


Figure: 5.1: Confusion matrix

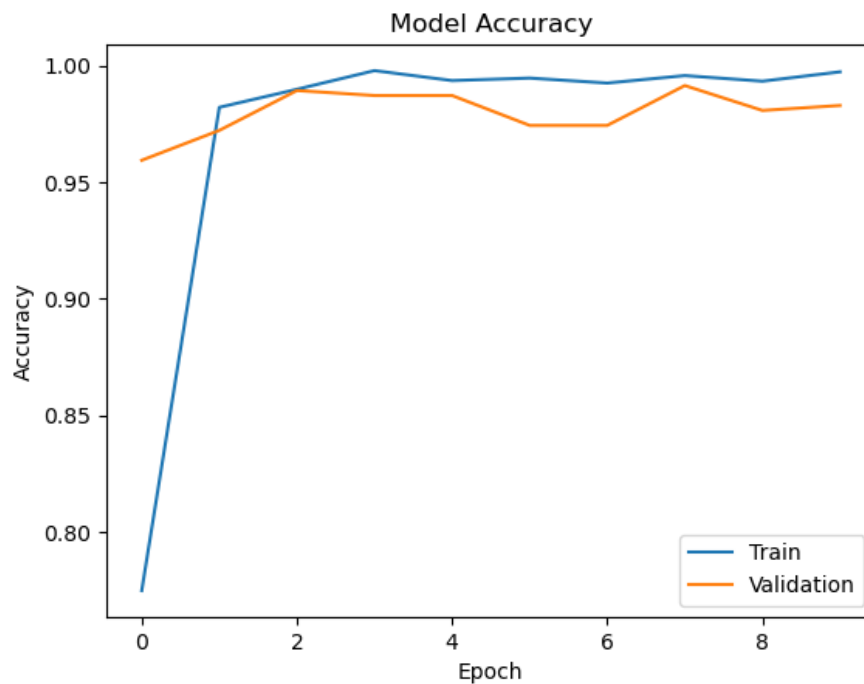


Figure:5.2: Model Accuracy

## 5.2. Work Completed

The following tasks were completed during the project:

- Data collection: We captured images of hand gestures for all 26 ASL alphabets using a camera and the Mediapipe hand tracking library.
- Data preprocessing: The images were preprocessed to remove the background and normalize the hand size and position.
- Model training: We trained a convolutional neural network (CNN) on the preprocessed images to recognize the ASL alphabets.
- Model evaluation: We evaluated the model on a test dataset to measure its accuracy.
- Text-to-speech: Added support for text-to-speech
- Words Builder: Added support for words builder from individual letters

## 5.3. Limitations

One limitation of the project is that it only recognizes ASL alphabets and does not support recognizing words or phrases. Another limitation is that the model's accuracy may decrease in challenging deam lighting conditions or hand poses.

## 5.4. Problems Faced

Some of the challenges faced during the project include:

- Difficulty in capturing high-quality images of hand gestures in challenging lighting conditions
- Complexity in preprocessing the images to remove the background and normalize the hand size and position
- Time-consuming process of training and optimizing the CNN model

## **CHAPTER VI**

### **CONCLUSION AND FUTURE ENHANCEMENTS**

#### **6.1. Conclusion**

In this project, we have presented a system for American Sign Language (ASL) recognition using a deep learning-based approach. Our system uses a convolutional neural network (CNN) trained on a dataset of hand gestures mapped to the ASL alphabet. We have demonstrated that our system achieves high accuracy in recognizing ASL gestures in real-time using live video feeds or recorded videos. Our system can be used to aid people with hearing and speech impairments in communicating with others.

#### **6.2. Future Enhancements**

There is ample scope for future enhancements in our ASL recognition system. Firstly, we can expand the dataset to include a larger number of hand gestures to recognize a wider range of ASL phrases and sentences. Secondly, we can explore the use of more advanced deep learning models such as recurrent neural networks (RNNs) and attention-based models to improve accuracy. Additionally, we can investigate the use of multi-modal input sources such as audio and facial expressions to enhance the recognition capabilities of our system. Finally, we can explore the use of transfer learning techniques to adapt our system to recognize other sign languages used around the world.

## CHAPTER VII

### REFERENCES

- [1] Dhiman, M., 2021. Summer Research Fellowship Programme of India's Science Academies 2017. [online] AuthorCafe.
- [2] T D, Sajanraj & M V, Beena. (2018). Indian Sign Language Numeral Recognition Using Region of Interest Convolutional Neural Network.
- [3] Chen, J., 2021. CS231A Course Project Final Report Sign Language Recognition
- [4] P.Amatya,k.Sergieva & G. Meixener," Translation of Sign Language Into Text Using Kinect for Windows v2".
- [5] A. Joshi, H. Sierra & E. Arzuaga," American sign language translation using edge detection and cross-correlation".
- [6]. Pigou, L., Dieleman, S., Kindermans, P.-J., Schrauwen, B.: Sign language recognition using convolutional neural networks. In: Workshop at the European Conference on Computer Vision 2014, pp. 572–578. Springer International Publishing (2014).
- [7]. Sharma, R., Bhateja, V., Satapathy, S.C., Gupta, S.: Communication device for differently abled people: a prototype model. In: Proceedings of the International Conference on Data Engineering and Communication Technology, pp. 565–575. Springer, Singapore (2017)
- [8]Thomas Noltey, Hans Hansson, Lucia Lo Belloz," Communication Buses for Automotive Applications" In Proceedings of the 3rd Information Survivability Workshop (ISW-2007), Boston, Massachusetts, USA, October 2007. IEEE Computer Society.
- [9] R. S. Pressman, Software Engineering (3rd Ed.): A Practitioner's Approach. New York, NY, USA: McGraw-Hill, Inc., 1992.