BE PROJECT REPORT ON

HAND SIGN GESTURES TO TEXT CONVERSION

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

SAGAR KAPADE	Roll No: 19U518
ABHIJEET KATORE	Roll No: 19U519
PARAG KATRE	Roll No: 19U520
ADITYA RAHANGDALE	Roll No: 19U532

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SMT. KASHIBAI NAVALE COLLEGE OF ENGINEERING

VADGAON BK, OFF SINHGAD ROAD, PUNE 411041

SAVITRIBAI PHULE PUNE UNIVERSITY

2021 -2022

Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitles

"HAND SIGN GESTURES TO TEXT CONVERSION"

Submitted by

SAGAR KAPDE	Roll No: 19U518
ABHIJEET KATORE	Roll No: 19U519
PARAG KATRE	Roll No: 19U520
ADITYA RAHANGDALE	Roll No: 19U532

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Dr. P. N. Railkar** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Dr. P. N. Railkar)
Guide
Department of Computer Engineering

(Prof. R. H. Borhade)
Head,
Department of Computer Engineering

(Dr. A. V. Deshpande)
Principal,
Smt. Kashibai Navale College of Engineering Pune – 41

Place: Pune
Date:

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank our **HOD, Prof. R. H. Borhade**, for providing me an opportunity to do the project work in Smt. Kashibai Navale College of Engineering Pune and giving us all support and guidance which made me complete the project duly. We are extremely thankful to him for providing such nice support and guidance, although he had a busy schedule managing the corporate affairs.

We owe our deep gratitude to our project guide **Dr. P. N. Railkar**, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

SAGAR KAPDE	Roll No: 19U518
ABHIJEET KATORE	Roll No: 19U519
PARAG KATRE	Roll No: 19U520
ADITYA RAHANGDALE	Roll No: 19U532

ABSTRACT

Sign language is among the oldest and also most natural methods of communication; As a result, we created an accurate timing approach for fingerspelling depending on American sign language utilizing neurons or neural networks. The automatic identification of human gestures from camera images is intriguing for vision research. We present a convolution neural network (CNN) approach for recognizing human movements' hand motions from a camera-captured image. The aim is to distinguish the hand movements of human task activities from a camera image. The training and testing data for the CNN are obtained using the skin model, hand position, and orientation. The hand image is first filtered, and then it is sent to a classifier that predicts if the hand gestures belong to any of the classes. The hand position attempts to rotate and translate the hand image into a neutral position. After that, the CNN is trained with the calibrated data images.

***Keywords*— Feature Extraction and Representation, Artificial Neural Networks, Convolution Neural Network**

TABLE OF CONTENTS

J. Qin, J. Ashley and B. Xie, "Hand Gesture Recognition Based on a Nonconvex Regularization," *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2021, pp. 187-192, doi: 10.1109/ICMA52036.2021.9512752.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

As computer technology advances, so does the need for humans and machines to work together. Communication is one of the most crucial components of human development, and technological advancements assist in its improvement. Effective communication is a crucial talent that allows us to comprehend and connect with those around us. It enables us to foster respect and trust, settles conflicts, and promotes long-term development in an environment that fosters problem-solving, caring, and innovative thinking. Poor communication skills are usually to blame for any quarrel or problem in any relationship. Poor communication is characterized by inattention, disagreements, vilification, and a language barrier between communicators. All of these characteristics impact both physically fit and physically challenged people. Investigating the communication barrier between hearing-impaired and hearing people has led to the need for a method of bridging this communication gap.

Hearing-impaired and verbally challenged people utilize sign language, a kind of hand gesture involving visual motions and signs, as a means of communication. However, few people can understand sign languages, creating a serious communication barrier between the deaf community and the rest of society. This document explains how to recognize letters from hand movements using the American Sign Language (ASL). Instead of speaking words, sign language uses hand gestures and movements, body language, and facial expressions to communicate.

Although sign language is used mainly by the deaf and hard of hearing, hearing people who cannot communicate verbally also use and who have difficulty speaking due to a disability or medical problem (augmentative and alternative interaction, or those with deaf family members). Previously, communication with sign language was only possible when both concerned parties understood it or with an outside expert's assistance.

In our project, we focus on producing a model that can recognise Fingerspelling based hand gestures to form a complete word by combining each gesture. The gestures we aim to train are as given in the image below.

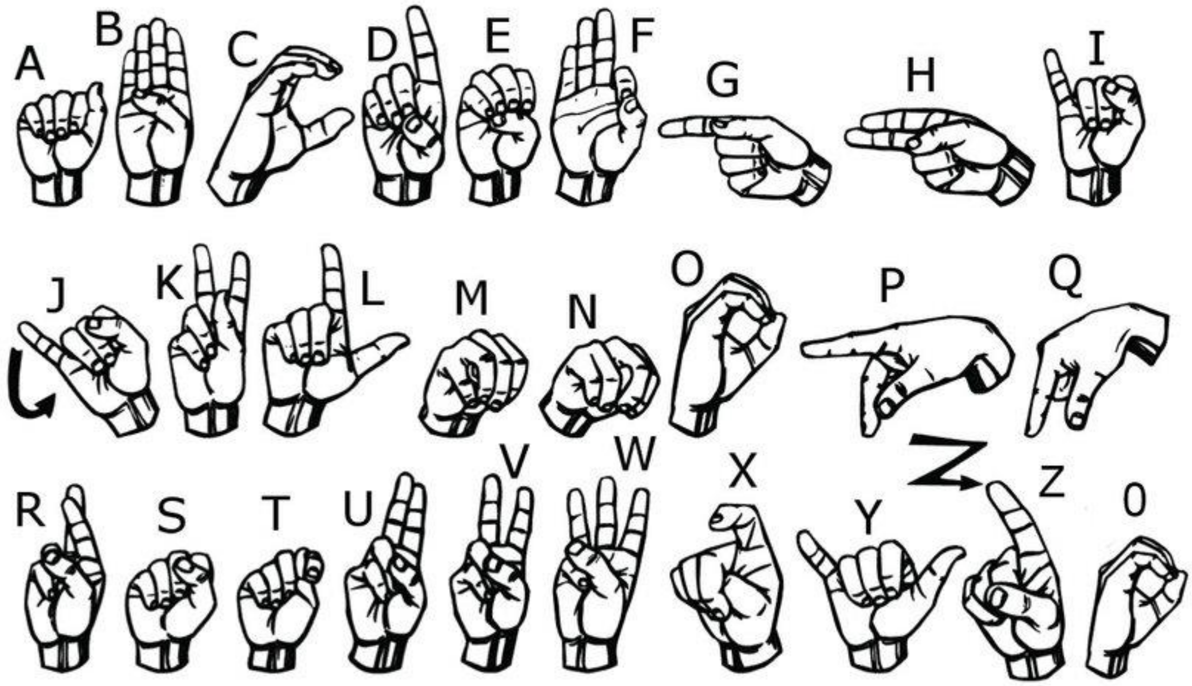


Fig. 1.1 Finger Spelling ASL

1.2 MOTIVATION

While the world has advanced in terms of online interaction through various platforms, the D&M (Dumb and Deaf) community continues to suffer a linguistic barrier. As a result, they rely on vision-based communication to communicate. Other individuals can readily understand the hand movements if the sign language is converted to text using a familiar interface. As a result, work on a vision-based interface system that enables D&M individuals to communicate without speaking the language has begun. The goal is to create a user-friendly sign-to-speech converting software that recognizes human sign language.

American Sign Language (ASL), Japanese Sign Language, British Sign Language (BSL), Indian Sign Language (ISL), and French Sign Language (FSL) are examples of sign languages. Previously, other languages were utilized.

1.3 PROBLEM DEFINITION AND OBJECTIVES

To create computer software and train a model using CNN, which takes an image of a hand gesture and shows the output of the detected sign language in text format.

1.3.1 OBJECTIVES

- To create a completely functional product for those who cannot hear, speak or see so that they can get connected to the world easily.
- To use techniques that improve every aspect compared to similar existing technologies.
- To make this technology readily available at everyone's fingertips, even in the absence of internet connectivity.

1.4 METHODOLOGIES OF PROBLEM-SOLVING

A system is a vision-based approach. All the signs are represented with bare hands, eliminating the problem of using any artificial devices for interaction.

Data Set Generation

All we could find were the datasets in the form of RGB values. Hence we decided to generate a data set. The steps followed to generate the data set are as follows.
We have decided to use the Open computer vision(OpenCV) library to produce our dataset. Firstly, we will capture around 800 images of each of the symbols in ASL for training purposes and approximately 200 images per symbol for testing purposes.
First, we capture each frame shown by the webcam of our machine. In each frame, we define a region of interest (ROI) denoted by a blue bounded square. From this whole image, we extract our ROI, which is RGB, and convert it into a grayscale image. Finally, we apply our gaussian blur filter to our image, which helps us extract various features of our image—the image after applying gaussian blur.

GESTURE CLASSIFICATION:-

Algorithmic Layer:-

1. Apply gaussian blur filter and threshold to the frame taken with OpenCV to get the processed image after feature extraction.
2. This processed image is passed to the CNN model for prediction, and if a letter is detected for more than 50 frames, then the letter is printed and taken into consideration for forming the word. 3. Space between the words is considered using the blank symbol.

Activation Function:-

We have decided to use ReLU(Rectified Linear Unit) as our activation function in each layer. It is a simple function stated as f(x) = max(0,x) for each input pixel.Using these activation functions, we would be able to reduce the required computation power, Gaussian Descent losing problems, etc.

Pooling Layer:-

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

Optimizer:-

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

Fingerspelling sentence formation Implementation:-

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

Key Words and Definitions:

Feature Extraction and Representation:

The representation of an image as a 3D matrix having dimensions as of height and width of the image and the value of each pixel as depth (1 in case of Grayscale and 3 in case of RGB). Further, these pixel values are used for extracting useful features using CNN.

Artificial Neural Networks:

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

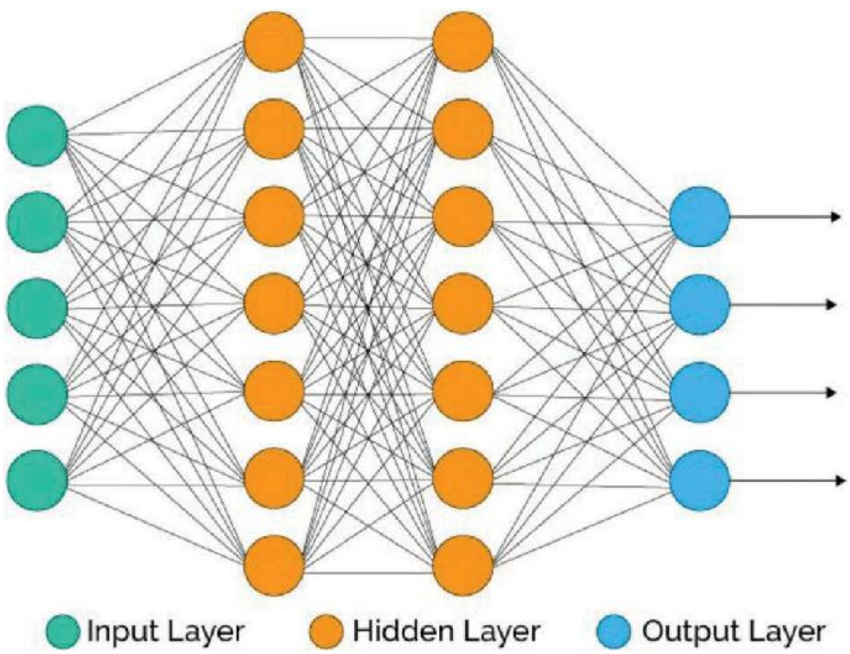


Fig. 1.2 Artificial Neural Network

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

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

Convolution Neural Network:

Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, and depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

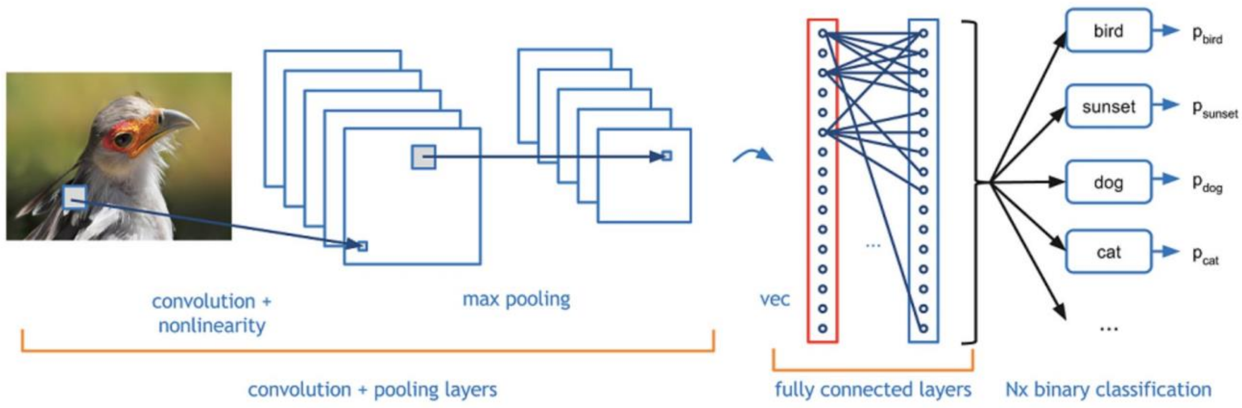


Fig. 1.3 Convolutional (CNN) for Image Classification

1. **Convolution Layer**: In the convolution layer we take a small window size [typically of length 5*5] that extends to the depth of the input matrix. The layer consists of learnable filters of window size. During every iteration, we slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position. As we continue this process well 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 colour

2. **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 well finally get an activation matrix half of its original Size.

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

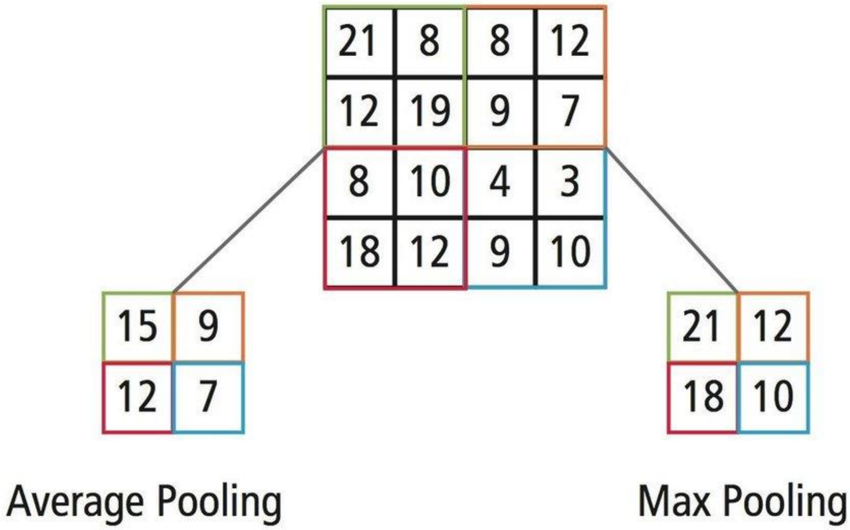


Fig. 1.4 Types of Pooling

3. **Fully Connected Layer:** In convolution, layer neurons are connected only to a local region, while in a fully connected region, well connect all the inputs to neurons.

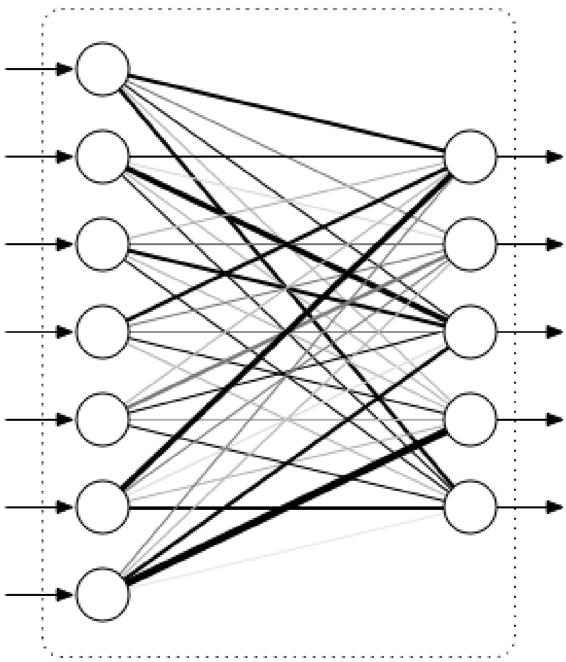


Fig. 1.5 Fully Connected Layer

4. **Final Output Layer:** After getting values from a fully connected layer, well 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.

TensorFlow:

Tensorflow is an open source software library for numerical computation. First we define the nodes of the computation graph, then inside a session, the 13 actual computation takes place. TensorFlow is widely used in Machine Learning.

Keras:

Keras is a high-level neural networks library written in python that works as a wrapper to TensorFlow. It is used in cases where we want to quickly build and test the neural network with minimal lines of code. It contains implementations of commonly used neural network elements like layers, objectives, activation functions, optimizers, and tools to make working with images and text data easier.

OpenCV:

OpenCV(Open Source Computer Vision) is an open source library of programming functions used for real-time computer vision. It is mainly used for image processing, video capture and analysis of features like face and object recognition. It is written in C++ which is its primary interface, however, bindings are available for Python, Java, MATLAB/OCTAVE.

CHAPTER 2
LITERATURE SURVEY

In recent years there has been tremendous research done on hand gesture recognition. With the help of a literature survey that is as follows:-

Paper Name	Publication House and Year	Description
Hand Gesture Recognition Based on a Nonconvex Regularization [1]	Southeastern Center for Electrical Engineering Education, (SCEEE) 2021	Paper stresses on the hand gesture recognition approach to improving the performance using Nonconvex Regularization which can enhance the prediction accuracy marginally.
Hand gesture recognition on python and OpenCV [2]	International Conference on Electrical Energy and Power Engineering, (ICEEPE) 2021	This paper develops hand gesture recognition using Python and OpenCV which can be implemented by applying the theories of hand segmentation and hand detection system which use the Haar-cascade classifier
Real-Time Hand Gesture Recognition[3]	International Journal of Scientific Research in Computer Science, Engineering and Information Technology, (IJSRCEIT) 2021	Discusses on a set of flows for hand gesture recognition (AdaBoost classifier based on Haar feature, Using camshaft algorithm and the hand gesture area is classified by a convolution neural network
Dynamic Hand Gesture Recognition Based on Short-Term Sampling Neural Networks[4]	Institute of Electrical and Electronics Engineers, (IEEE) 2020	This paper uses the short-term sampling neural network model for hand gesture recognition effectively.
A new hand gestures recognition system [5]	Indonesian Journal of Electrical Engineering and Computer Science, (IJEECS) 2020	This paper recognizes six gestures After the recognition step generates the command and passes it to control the computer. The system is implemented with different rotations, scale, lighting and plain or simple background.

Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network [6]	Multidisciplinary Digital Publishing Institute, (MDPI) 2020	This paper presented a new approach to hand gesture recognition using a combination of geometry algorithm and a deep learning method to achieve fingertip detection and gesture
Hidden Markov model-based Sign Language to Speech Conversion System in TAMIL [7]	International Conference on Biosignals, Images, and Instrumentation, (ICBSII) 2018	This paper uses an HMM-based hand gesture recognition system that recognizes dynamic hand gestures and converts them to Tamil text.
Hand Gesture Recognition [8]	International Journal for Research in Applied Science and Engineering Technology, (IJRASET) 2018	This paper compares the power among PCA, OTSU, HMM techniques and uses PCA along with HMM
Hand Gesture Recognition Based on Computer Vision: A Review of Techniques [9]	Journal of Imaging 2020	This paper presented a vision-based hand gesture recognition system which operates on real-time videos on an average PC using low cost cameras
Dynamic Hand Gesture Recognition Using Kinect Sensor [10]	International Conference on Global Trends in Signal Processing, Information Computing and Communication, (ICGTSPICC) 2016	In this paper a technique for representing, identifying, and understanding human gestures utilizing the HMM(hidden Markov model) is developed
Sign Language Recognition System Based on Weighted Hidden Markov Model [11]	International Centre for Settlement of Investment Disputes (ISCID) 2015	this paper utilized Kinect to produce one robust feature consisted of hand shape and hand trajectory.

Below Table 1. is the Gap Analysis of each research paper of recent studies.

GAP ANALYSIS

Existing solution	Prediction Accuracy	Extensibility	Realtime	Storage Efficiency
Hand Gesture Recognition Based on a Nonconvex Regularization [1]	✔	✔	✔	✔
Hand gesture recognition on python and OpenCV [2]	✗	✔	✔	✔
Real-Time Hand Gesture Recognition [3]	✔	✗	✔	✔
Dynamic Hand Gesture Recognition Based on Short-Term Sampling Neural Networks[4]	✔	✔	✗	✗
A new hand gestures recognition system [5]	✗	✗	✗	✔
Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network [6]	✔	✔	✔	✗
Hidden Markov model-based Sign Language to Speech Conversion System in TAMIL [7]	✔	✔	✔	✔
Hand Gesture Recognition [8]	✔	✗	✔	✗
A Precise Technique for Hand Gesture Recognition [9]	✗	✔	✔	✔
Dynamic Hand Gesture Recognition Using Kinect Sensor [10]	✔	✗	✔	✔
Sign Language Recognition System Based on Weighted Hidden Markov Model [11]	✗	✔	✔	✔

Accuracy = prediction acc. of signs, Extensibility = if the model could be extended, Realtime = instantaneous pridiction, Storage Efficiency = work well without making mistakes

Table: 1 Gap Analysis

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

A System Requirements Specification (SRS) is a document or set of documentation that describes the features and behavior of a system or software application. A software requirements specification (**SRS**) describes a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that illustrate user interactions that the software must provide.

Purpose of SRS

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters, and goals. This document describes the target audience of the project and its user interface, hardware, and software requirements.

Requirement of SRS

As mentioned previously, the SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that the design solution can be devised.

3.1.1 PROJECT SCOPE

Applications:

1. Applicable to sorts of hands, irrespective of sizes.
2. This will be the form of nonverbal communication that'd be used in several fields such as communication between deaf-mute people, human-computer interaction (HCI), Home automation, and medical applications.

Features:

1. Infers all types of signs.
2. It helps in the fast processing of information and recognition in a short time.
3. Maintenance of data is very easy as it's based on a file system.
4. Easy identifying the gestures with no clicks and a hassle-free approach

3.1.2 USER CLASSES AND CHARACTERISTICS

Admin: The Admin is the primary component of the system that will manage and handle all the underlying tech stack. Admin is the power user of the system; all control is confined to the admin only.

Client: The end-user just has to use it to communicate, and no internal, underlying access is given to them.

System Database: The database system does not store or use public data/sensitive data, i.e., name ID, etc... So no threat to users. The database system follows the files system.

3.1.3 ASSUMPTIONS AND DEPENDENCIES

Assumption:

1. It is assumed that the user places only the Hand/palm inside the camera frame.
2. Background must be clear, without noise.

Dependencies:

1. The application relies on the input that is obtained from the camera.
2. The use of redundant systems will result in a greater cost.

3.2 FUNCTIONAL REQUIREMENTS

3.2.1 System Input

The system will take input as an image. The image will consist of the sign.

3.2.2 System Output

The system will give appropriate words corresponding to the signs.

3.3 NON FUNCTIONAL REQUIREMENTS

3.3.1 Performance Requirements

The system should be able to capture the hand signs in considerable time. It should give the right output every time.

3.3.2 Safety Requirements

The system must be reliable. It should take care of the maximum capacity of handling the hand signs. The system should not crash in any circumstances and show uniform behavior in all cases.

3.3.2 Security Requirements

The application must be secure and does not need any user intervention to start the system. Once it receives any gestures, it starts indicating the signs, and as it is nothing to do with the user's identity its only concern is to detect the signs, so it does not capture user sensitive data

3.3.4 Software Quality Attributes

The system should have QA attributes like adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability.

3.4 SYSTEM REQUIREMENTS

3.4.1 Database Requirements:

Files system.

3.4.2 Software Requirements (Platform Choice):

1. Google Colab

3.4.3 Hardware Requirements:

- External/Internal Camera
- RAM - at least 2GB
- Storage - 64 GB or larger storage device
- Graphics card - Compatible with DirectX 10.
- Display - High definition (720p) display

3.5 ANALYSIS MODELS: INCREMENTAL MODEL

The SDLC Model to be applied is the Incremental model. The incremental build model is a method of software development where the model is designed, implemented, and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements.

3.6 PROJECT TIMELINE

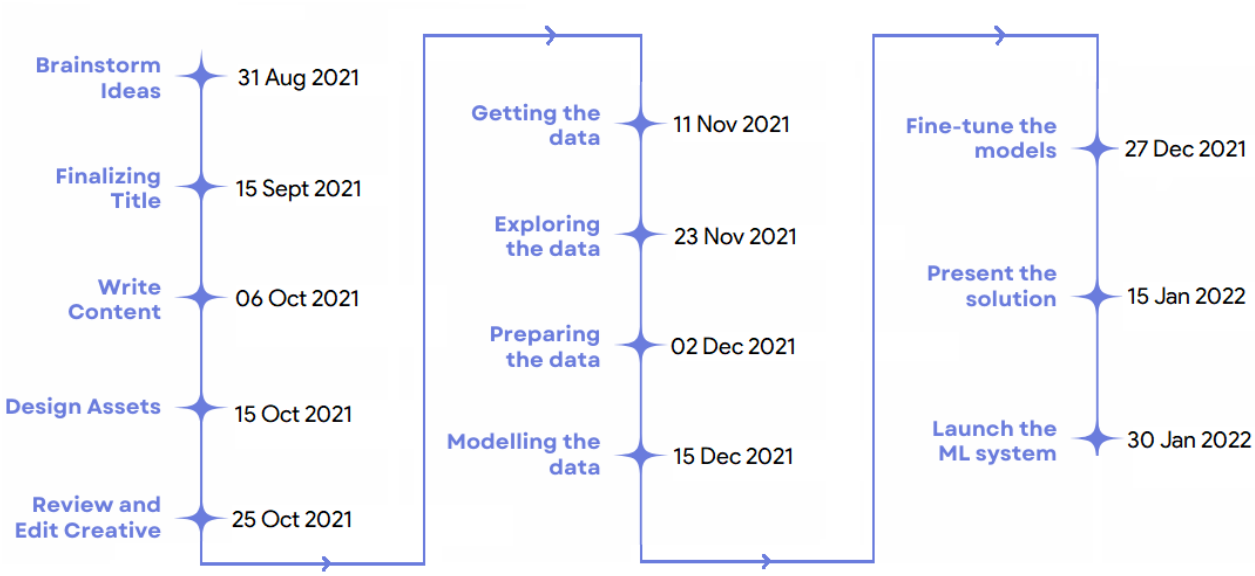


Fig. 3.6.1 Project Timeline

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

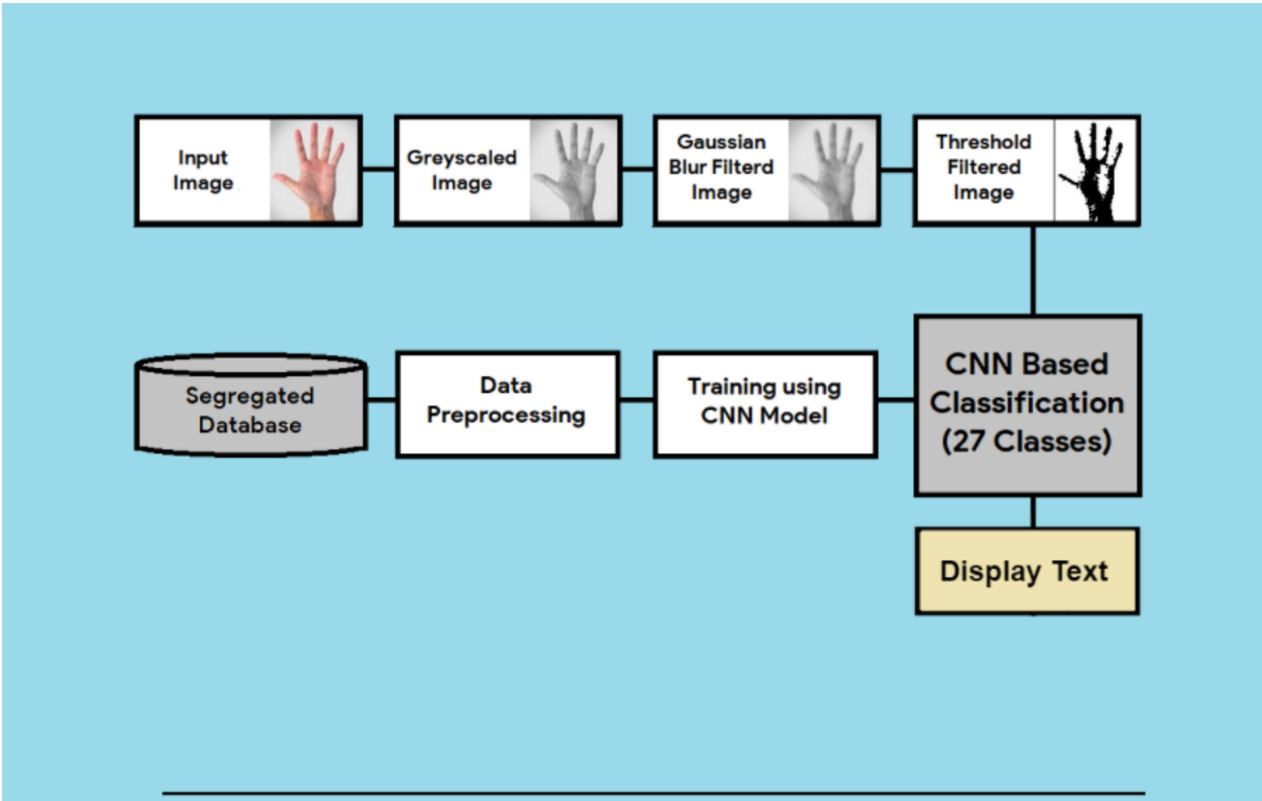


Fig. 4.1 System Architecture

The above figure, 4.1, explains the system architecture of our proposed System.

The inter-connected components are shown with their relationship with each other.

4.2 MATHEMATICAL MODEL

Assumption: Say we have a system for our Application, let it be "**A**"

Input:

func= f_1, f_2 , f_3 ....., f_n F as set of functions to execute commands.

I= i_1, i_2, i_3 …..,I_n I as sets of input frame to the function set

O= o_1, o_2, o_3. O Set of outputs from the function sets,

Terminologies:

A = I, F, O, **I** = Input, **O** = Output

func = Functions implemented to get the output

Layer 1: CNN Model

Layer 2: Classification

Space Complexity:

The space complexity of an algorithm is just the amount of memory that you need to use during the execution of the algorithm.

Time Complexity: O(l*nt*k).

- Where l and k are layers, t is training example and n epochs
- Example:

  To find the time complexity for training a neural network that has 4 layers with respectively i, j, k and l nodes, with t training examples and n epochs.
- The result was O(nt*(ij+jk+kl)).

Note: Note that these operations can greatly be parallelized by GPUs.

Outcome: Result in accordance with the signs will be displayed on the screen.

4.3 DATA FLOW DIAGRAM

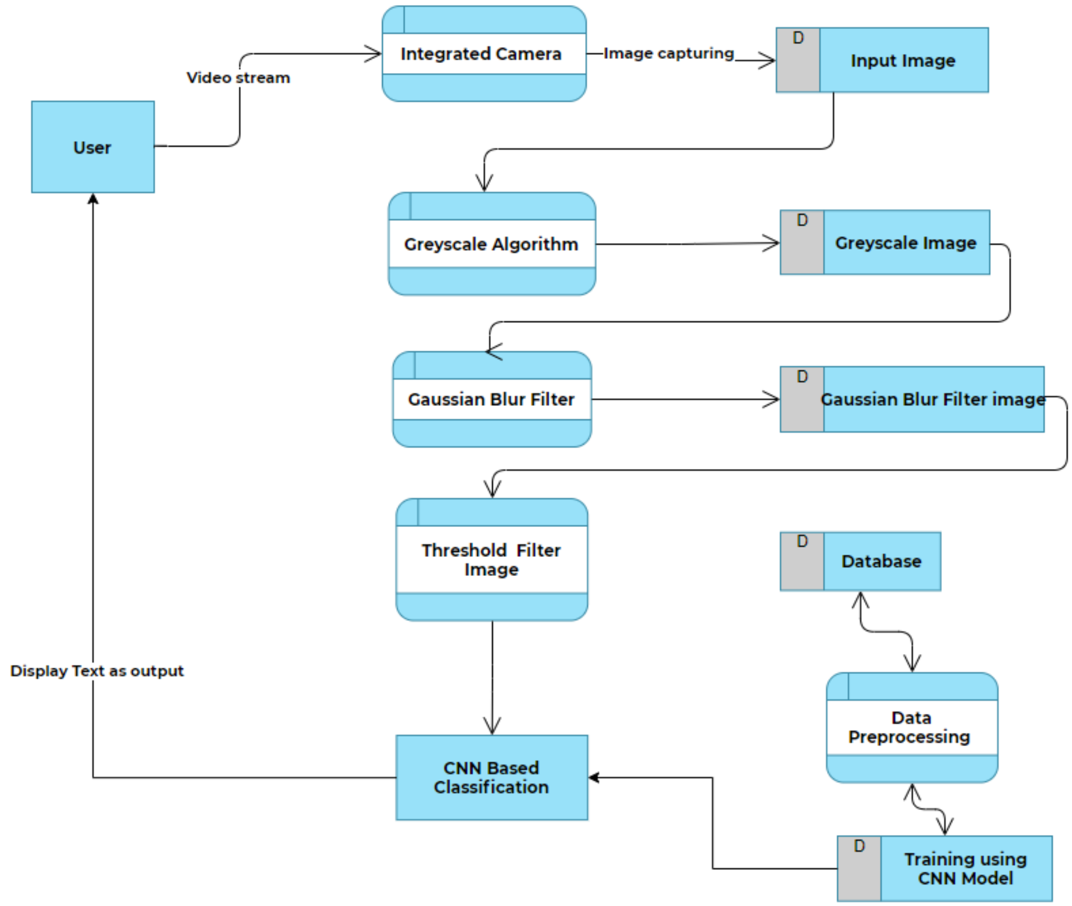


Fig. 4.2 Data Flow Diagram

The above figure, 4.2, explains the data flow diagram of the System".

It represents the flow of data from the input components to the output components.

4.4 UML DIAGRAMS

4.4.1 USE CASE DIAGRAM

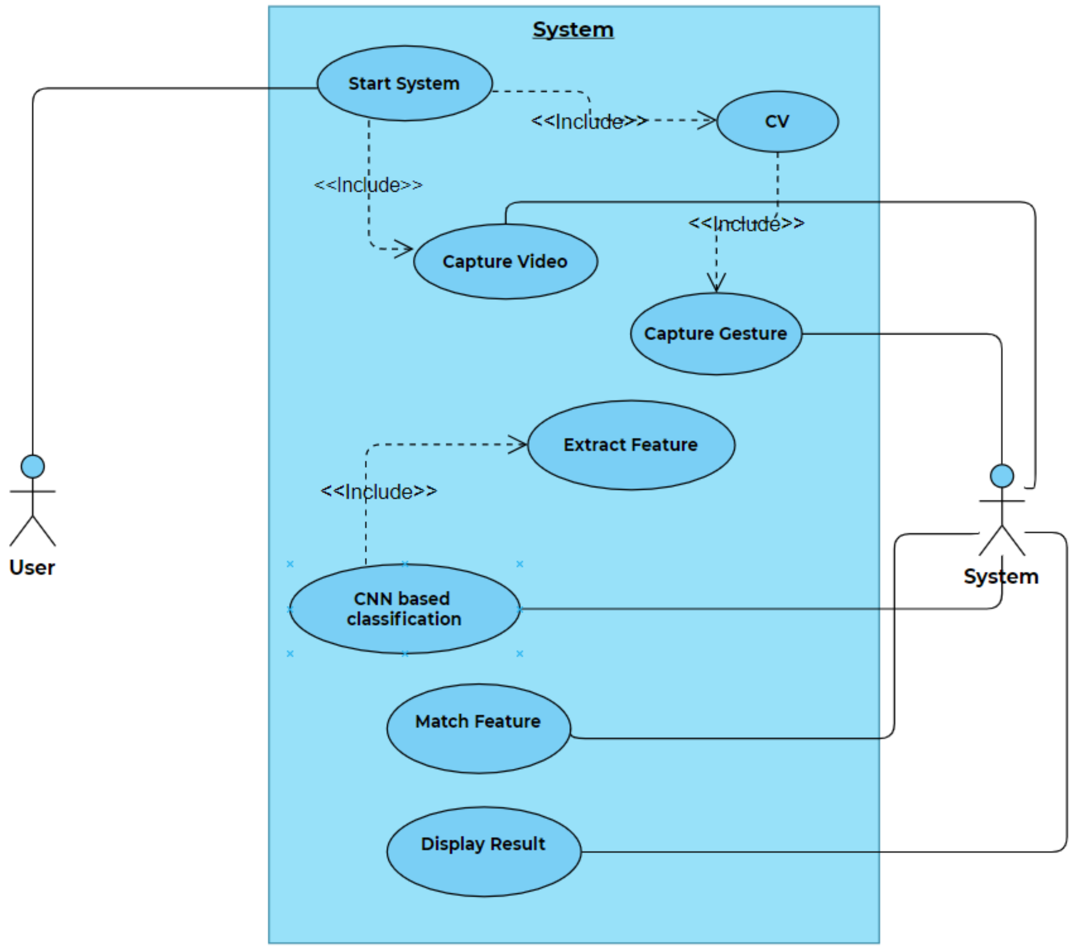


Fig. 4.3 Use Case Diagram

The above figure, 4.3, explains the Use Case diagram of our proposed System".

The diagram explains the components and their functions.

4.4.2 SEQUENCE DIAGRAM:

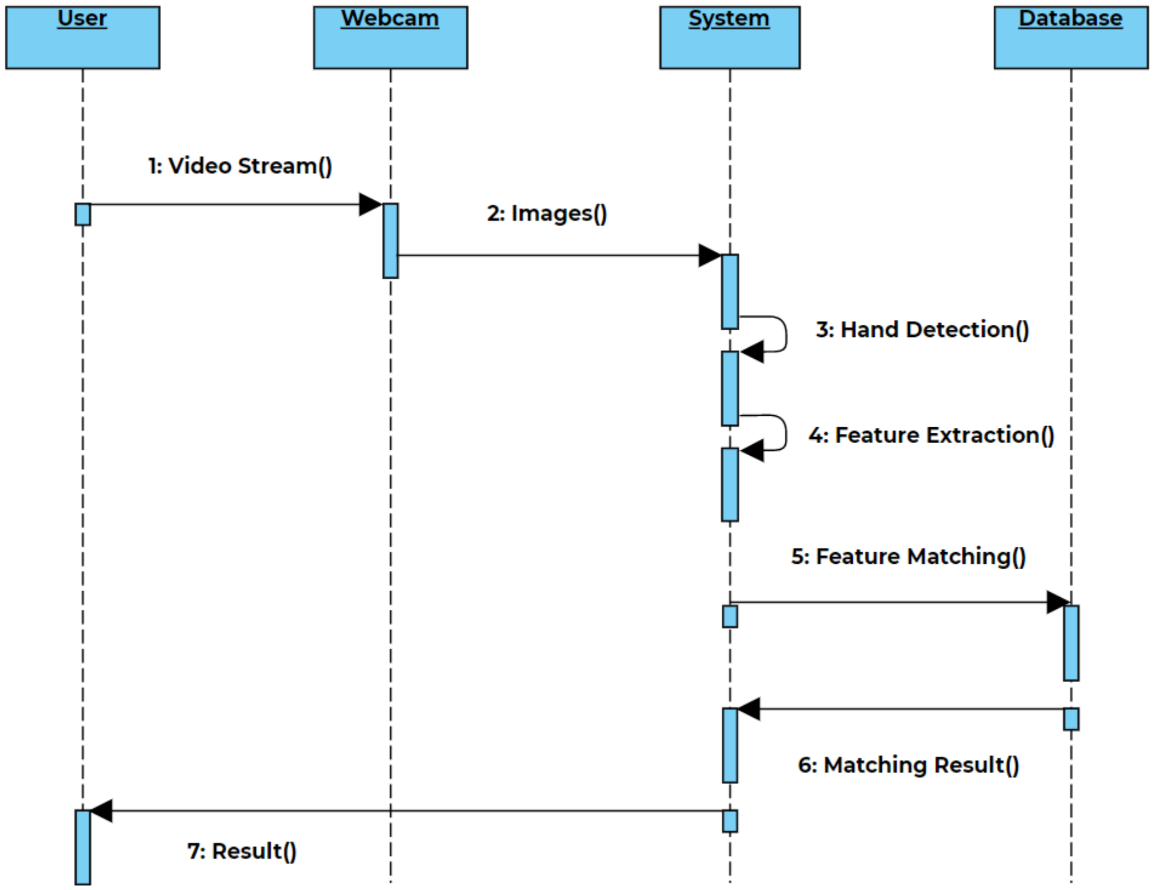


Fig. 4.4 Sequence Diagram

The above figure, 4.4, explains the sequence diagram of our proposed System".

It explains the sequence of the working of the System".

4.4.3 ACTIVITY DIAGRAM:

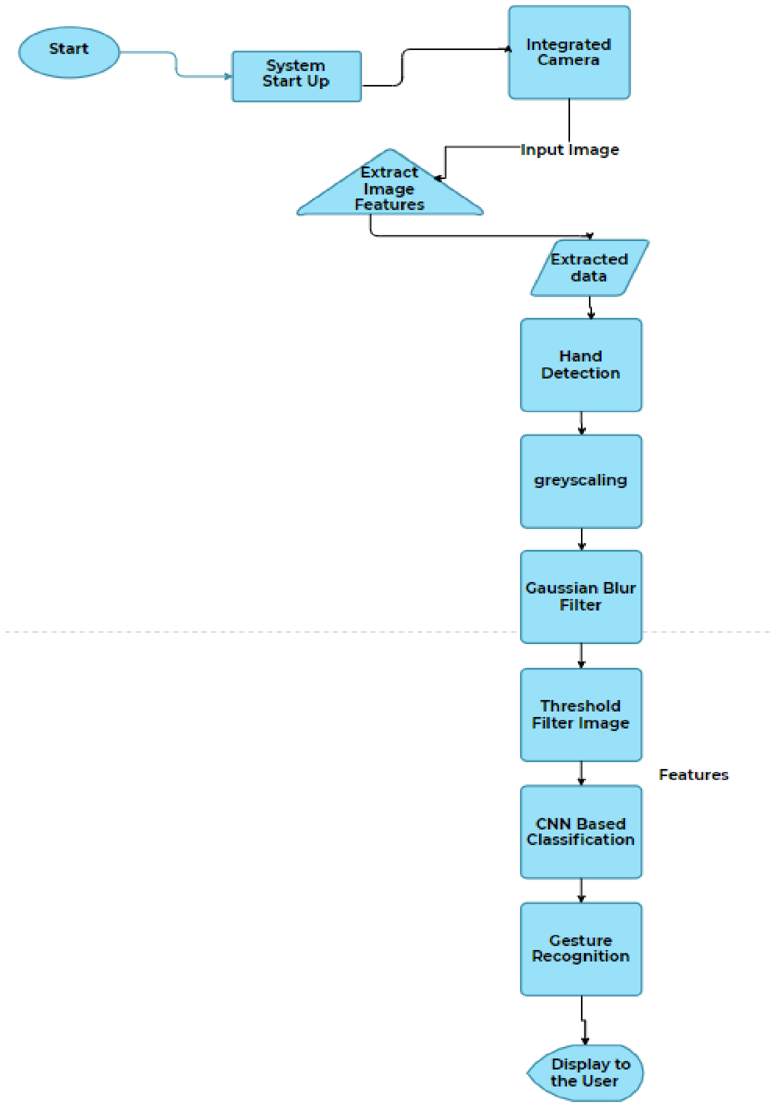


Fig. 4.5 Activity Diagram

The above figure, 4.5, explains the activity diagram of our proposed System".

It explains all the activities involved from the start to the end.

4.4.4 CLASS DIAGRAM:

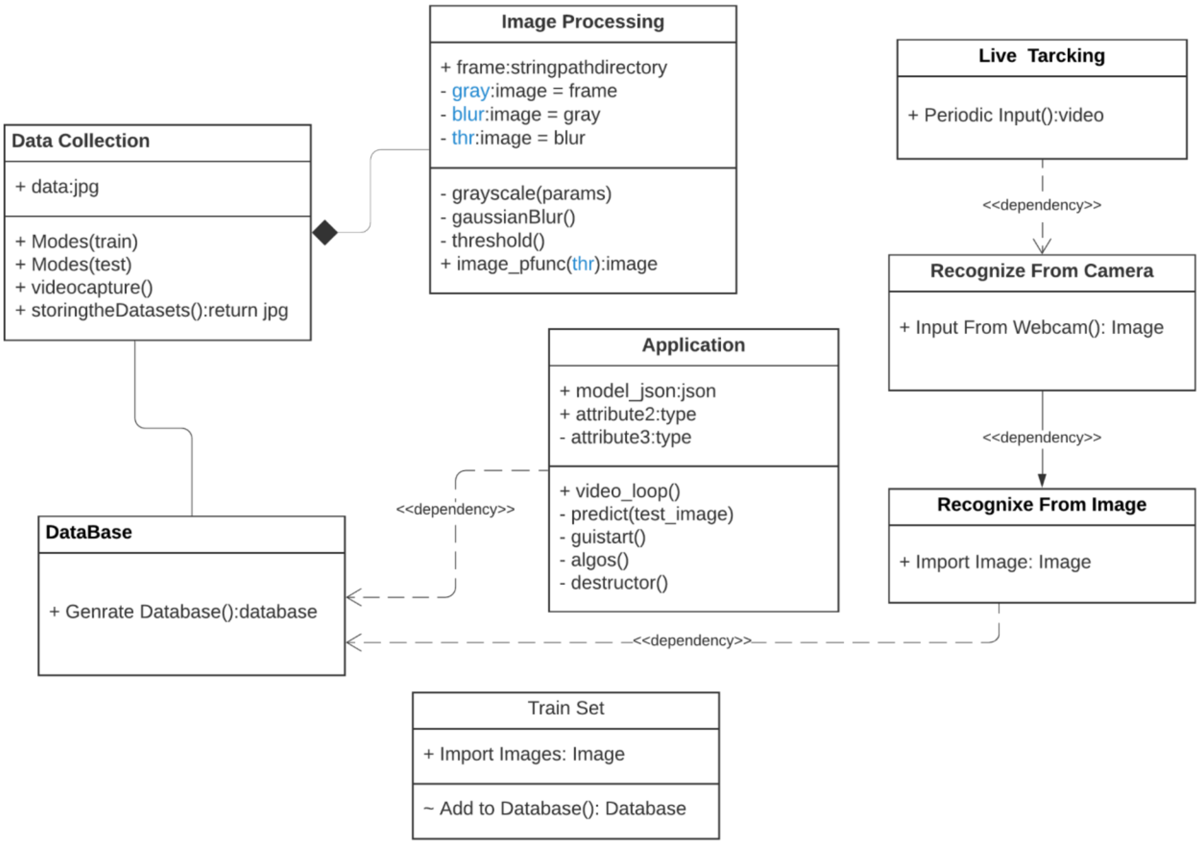


Fig. 4.6 Class Diagram

The above figure, 4.6, explains the class diagram of our proposed project.

It explains all the classes of the system and the functions associated with it.

4.4.5 STRUCTURAL DIAGRAMS

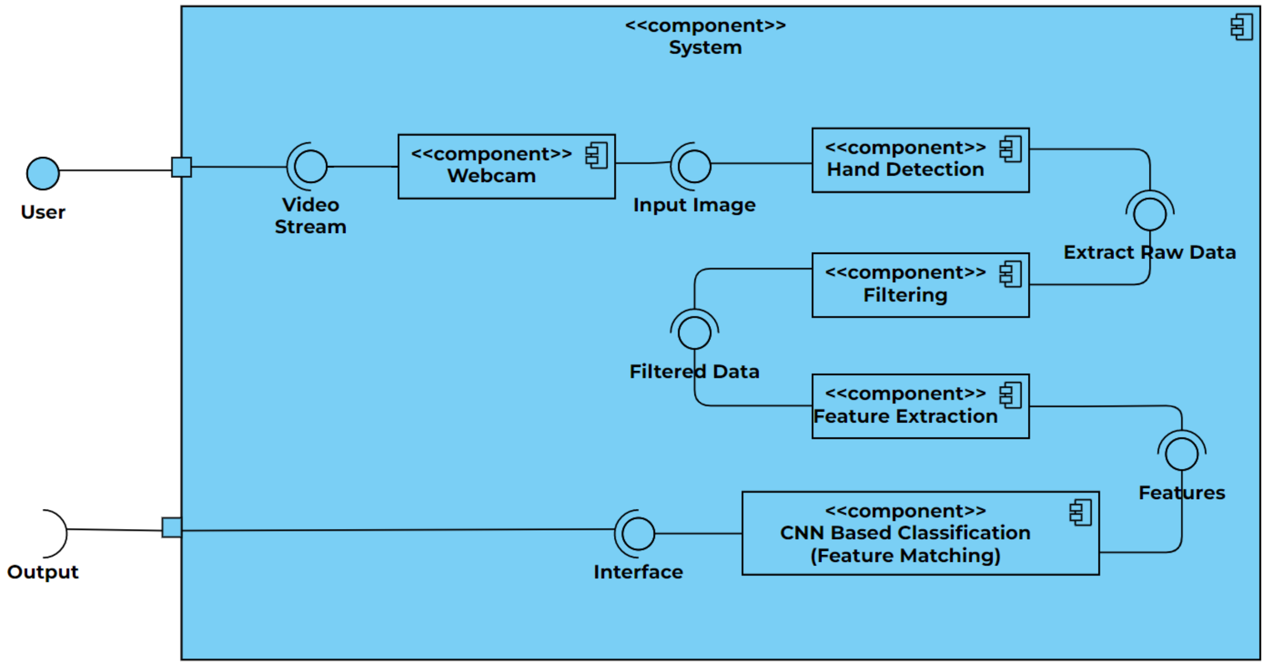


Fig. 4.7 Component Diagram

The above figure Fig. 4.7, explains the Component diagram of our proposed project. It explains how components are wired together to form larger components or software systems that is the model.

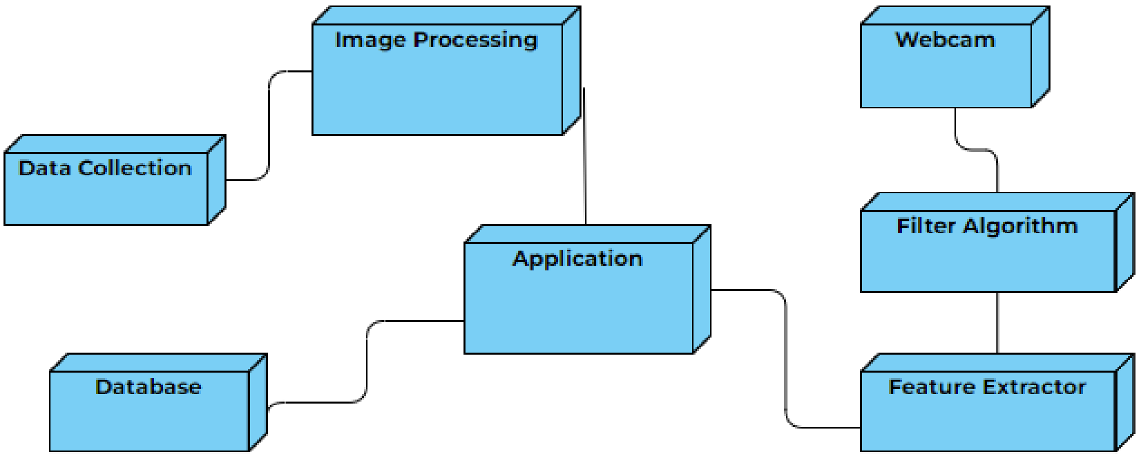


Fig. 4.8 Deployment Diagram

Here in Figure Fig 4.8. the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATE

5.1.1 Reconciled Estimate

Revised the previously completed package through team discussions.

5.1.1.1 Cost Estimates

With an aim to develop the project using open source tools, no major cost consuming factors exist.

5.1.1.2 Time Estimates

The estimated time of our project is 8 months.

5.1.2 Project Resources

- People: Internal Guide, External Guide, Development Team
- Hardware: CPU, GPU, External/Internal Camera
- Software: Windows OS, Python Libraries

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

Risk Identification forms an important part in the Software Development Life Cycle. This gives a view of all the possible risks to the system and whether they can be controlled or not. We may define risks by presenting a questionnaire and answering all the questions relating to it or we also have a table giving risk details and probability of occurrence.

1. Are end-users enthusiastically committed to the project and the system/product to be built? Yes, they will be enthusiastic about using this project since it is very helpful as it helps them to form communication with others.
2. Are requirements fully understood by the software engineering team and its customers? Yes
3. Have customers been involved fully in the definition of requirements? Yes, we are building this project keeping the end user in mind.
4. Do end users have realistic expectations? Yes,
5. Does the software engineering team have the right mix of skills? Yes
6. Are the project requirements stable? Yes
7. Is the number of people working on the project adequate to do the job? Yes

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality in below Table 5.2.2.1

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Underestimation of time required to train model	Low	High	Medium	High
2	No datasets available	High	Medium	High	High

Table no.5.2.2.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	> 75%
Medium	Probability of occurrence is	26 --- 75%
Low	Probability of occurrence is	<25%

Table no.5.2.2.2: Risk Probability Definitions

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following in the Table 5.2.3.1 are the details for each risk.

Impact	Value	Description
Very High	> 10%	Schedule impact or Unaccepted quality
High	5 --- 10%	Schedule impact or some parts of the project have low quantity
Medium	< 5%	Schedule impact or barely noticeable degradation in quality, Low impact on the schedule for Quality can be Incorporated

Table 5.2.3.1: Risk Details

Risk ID	1
Risk Description	Underestimation of time required to train the model
Category	Technology
Source	Identified during early development
Probability	Low
Impact	High
Response	Accept
Strategy	Use of Google Colab Platform for training
Risk Status	Identified

Risk ID	2
Risk Description	No datasets available
Category	Technology
Source	Identified during early development
Probability	High
Impact	High
Response	Mitigate
Strategy	Create your own dataset.
Risk Status	Identified

Table 5.2.3.2 : Risk Impact definitions

5.3 PROJECT SCHEDULE

5.3.1 Project Task Set

Major Tasks in the Project Stages are as follows:

- Task 1 (T1): Creating the dataset.
- Task 2 (T2): Learning and understanding the basics of machine learning.
- Task 3 (T3): Cleaning and pre-processing of the dataset.
- Task 4 (T4): Selection of the highest accuracy algorithm for training the model.
- Task 5 (T5): Creating a client terminal for using the trained model and making various predictions.

5.3.2 Task Network

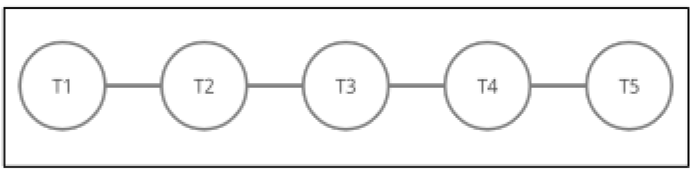


Figure 5.1: Task Network

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

Team consists of 4 members and 1 internal guide

5.4.2 Management Reporting and Communication

There are reviews which take place within the semester. Changes and modifications are suggested in these reviews and acted upon later. Communication takes place through

- Email
- Whatsapp
- Microsoft Teams

CHAPTER 6
PROJECT IMPLEMENTATION

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

6.1 OVERVIEW OF PROJECT MODULES

1. **Data-set Generation:-**

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

   We used the Open computer vision(OpenCV) library to produce our dataset. Firstly we captured about 800 images of every symbol for training and around 200 images of every symbol for testing

   Each frame is shown by the webcam(integrated to the machine), we captured those. We have defined a region of interest (ROI) that is denoted by blue square, in Fig.6.1

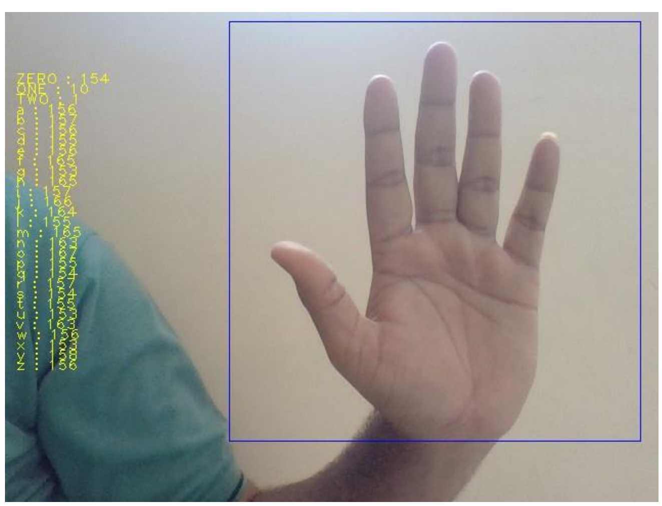


Fig. 6.1 Hand in ROI(region of interest)

We then extract our ROI that is RGB and later convert it into a grayscale image.

In the end, we use our gaussian blur filter on the image which eases the way to extract a variety of features of our image.

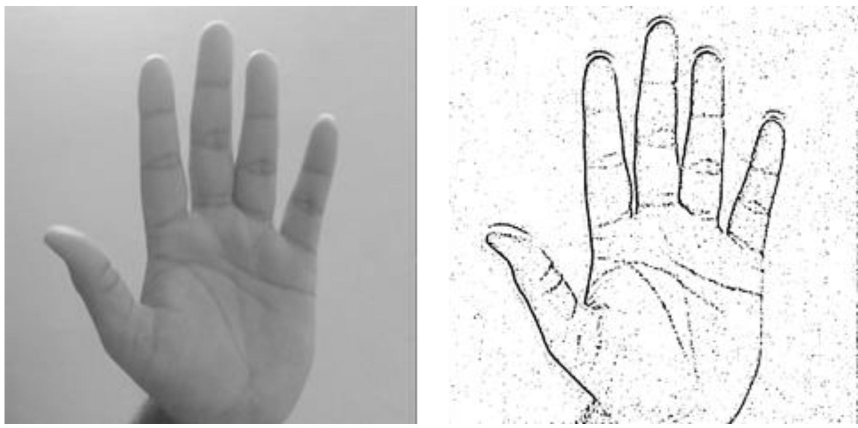


Fig. 6.2. Feature Extracting

1. **Gesture Classification;-**

- Layer 1;- CNN Model
    - **1st Convolution Layer**:- The input picture has a resolution of 128x128 pixels. It is first processed in the first convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 126X126 pixel image, one for each Filter-weights.

    - **1st Pooling Layer**:- The pictures are downsampled using max pooling of 2x2 i.e we keep the highest value in the 2x2 square of the array. Therefore, our picture is down sampled to 63x63 pixels

    - **2nd Convolution Layer**:- Now, these 63 x 63 from the output of the first pooling layer is served as an input to the second convolutional layer.It is processed in the second convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 60 x 60-pixel image.

- **2nd Pooling Layer**:- The resulting images are downsampled again using a max pool of 2x2 and are reduced to 30 x 30 resolution of images

- **1st Densely Connected Layer**:- Now these images are used as an input to a fully connected layer with 128 neurons and the output from the second convolutional layer is reshaped to an array of 30x30x32 = 28800 values. The input to this layer is an array of 28800 values. The output of this layer is fed to the 2nd Densely Connected Layer. We are using a dropout layer of value 0.5 to avoid overfitting.

- **2nd Densely Connected Layer**:- Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 96 neurons.

- **Final layer**:- The output of the 2nd Densely Connected Layer serves as an input for the final layer which will have the number of neurons as the number of classes we are classifying (alphabets + blank symbol).

2. Testing and Training:-

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

We feed the input images after preprocessing to our model for training and testing after applying all the operations mentioned above.

The prediction layer estimates how likely the image will fall under one of the classes. So the output is normalized between 0 and 1 and such that the sum of each value in each class sums to 1. We have achieved this using the softmax function.

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

As we have found out the cross entropy function, we have optimized it using Gradient Descent. In fact, the best gradient descent optimizer is called Adam Optimizer.

3. **Model Deployment:-** Going to deploy our model using Tkinter

6.2 TOOLS AND TECHNOLOGY USED

- Python for implementation.
- Google Colab for model creation.
- Adam optimizer for fast Gradient Descent calculations and adjusting the learning rate effectively.
- ImageDataGenerator for creating a wide variety of data from existing data with data augmentation.
- Open-CV for sign language image collection
- Keras for Deep Learning
- TKinter for frontend

6.3 ALGORITHMS DETAILS

1. **Convolutional neural networks:-** Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, and depth. The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner. Moreover, the final output layer would have dimensions (number of classes) because we will reduce the full image into a single vector of class scores

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

2. **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 well 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.

3. **Fully Connected Layer:** In convolution, layer neurons are connected only to a local region, while in a fully connected region, well connect all the inputs to neurons.

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

CHAPTER 7

SOFTWARE TESTING

The purpose of testing is to compare the outputs from the neural network against targets in an independent set (the testing instances). Note that the testing methods are subject to the project type (approximation or classification).

If all the testing metrics are considered ok, the neural network can move to the so-called deployment phase. Note also that the results of testing depend very much on the problem at hand, and some numbers might be right for one application but bad for another.

7.1 TYPES OF TESTING

The validation methods that need to be used depending on the application type:

- Approximation testing methods
- Classification testing methods
- Forecasting testing methods

As our model is based on classification the method implemented is “Classification testing methods”

7.2 TEST CASES AND TEST RESULTS

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

In the confusion matrix, the rows represent the target classes in the data set and the columns the corresponding output classes from the neural network

The diagonal cells in each table show the number of correctly classified cases, and the off-diagonal cells show the misclassified cases.

For binary classification, positive means are identified, and negative means rejected. Therefore, 4 different cases are possible:

- True positive (TP): correctly identified.
- False positive (FP): incorrectly identified.
- True negative (TN): correctly rejected.
- False negative (FN): incorrectly rejected.

Below is the confusion matrice for our results.

						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 7.2.1 Confusion Matrix

Note that the output from the neural network is, in general, a probability. Therefore, the decision threshold determines the classification. The default decision threshold is 0.5. The output above is classified as positive, and the output below is classified as negative.

Our model gave 99.8% accuracy for Sign Language Detection after training

	precision	recall	f1-score	support
0	1.00	1.00	1.00	117
A	1.00	1.00	1.00	115
B	1.00	1.00	1.00	135
C	1.00	1.00	1.00	124
D	0.98	1.00	0.99	118
E	1.00	1.00	1.00	120
F	1.00	0.99	1.00	142
G	1.00	1.00	1.00	126
H	1.00	1.00	1.00	140
I	1.00	1.00	1.00	115
J	1.00	1.00	1.00	141
K	1.00	1.00	1.00	123
L	1.00	1.00	1.00	132
M	1.00	1.00	1.00	130
N	1.00	1.00	1.00	124
O	1.00	1.00	1.00	120
P	1.00	1.00	1.00	118
Q	1.00	1.00	1.00	121
R	1.00	1.00	1.00	133
S	1.00	1.00	1.00	112
T	1.00	0.99	1.00	151
U	1.00	1.00	1.00	123
V	1.00	1.00	1.00	137
W	1.00	1.00	1.00	127
X	1.00	1.00	1.00	120
Y	1.00	1.00	1.00	125
Z	1.00	1.00	1.00	134
accuracy			1.00	3423
macro avg	1.00	1.00	1.00	3423
weighted avg	1.00	1.00	1.00	3423

Fig. 7.2.2 Allover precision scores

The model has been trained on a python based environment on the Jupyter platform. The model is iterated for a total epoch of 20. Epoch is the number of passes of the entire training dataset the machine learning algorithm has completed The model has attained an accuracy of 99.88 % accuracy on the Validation set. The prescribed model has been evaluated on a Test set where it has attained an accuracy of 99.85% with a loss of 0.60 %.

Below the figure, Fig shows the relations between loss and changes in its trends(variations) as per the epoch, 0 to 20.

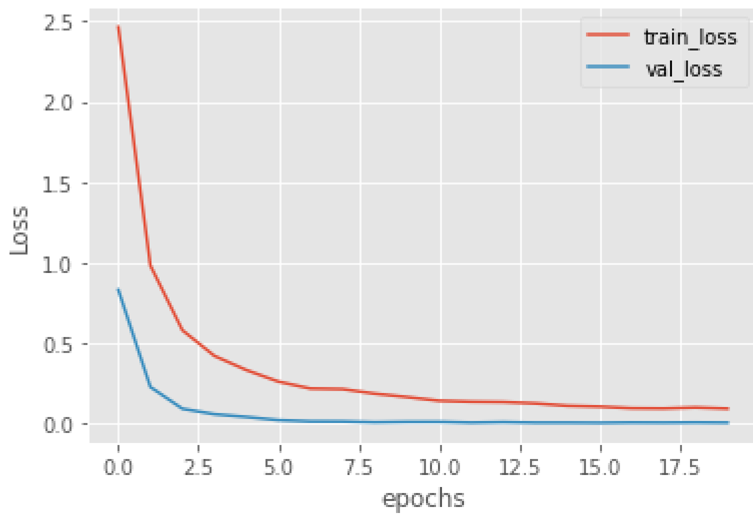


Fig. 7.2.4 Loss curve Plot

Below the figure, Fig shows the Accuracy plot of the model throughout its training journey.

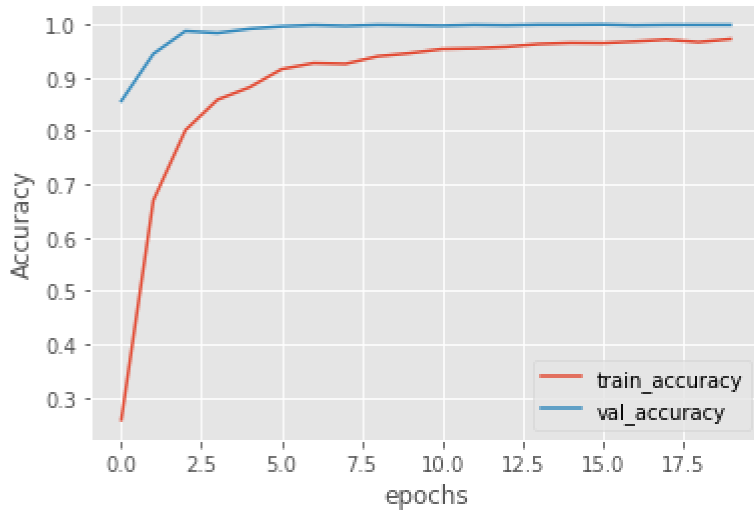


Fig. 7.2.5 Accuracy curve Plot

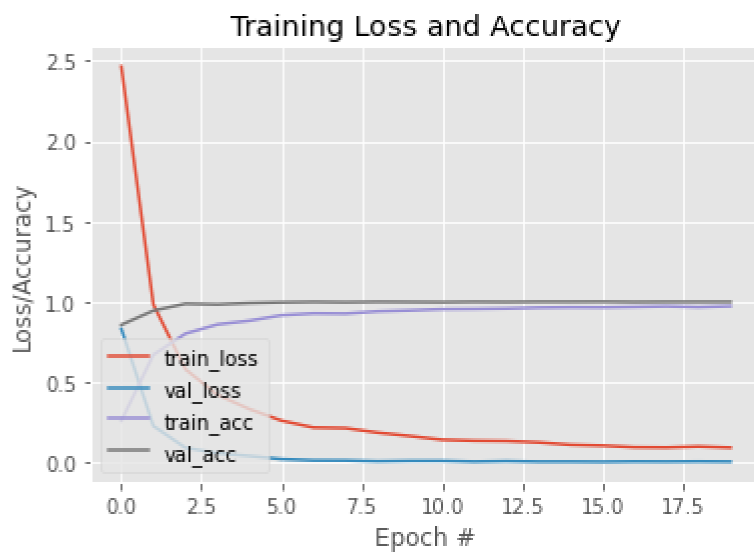


Fig. 7.2.6 Traning Loss and Accuracy

CHAPTER 8

RESULTS

The obtained accuracy was 99.8%. This is higher than current research in American Sign Language. Most research Publications focus on hand detection using technologies such as Kinect. Sensors like the Kinect are not readily available to most users and are expensive. The model uses a camera that is readily integrated into the device. Therefore, it is more accurate and cost-effective. Figure 8.1 represents the interface used to interact with the model; here, the word is generated using the alphabets distinguished after being classified by the CNN model.

8.1 Outcomes

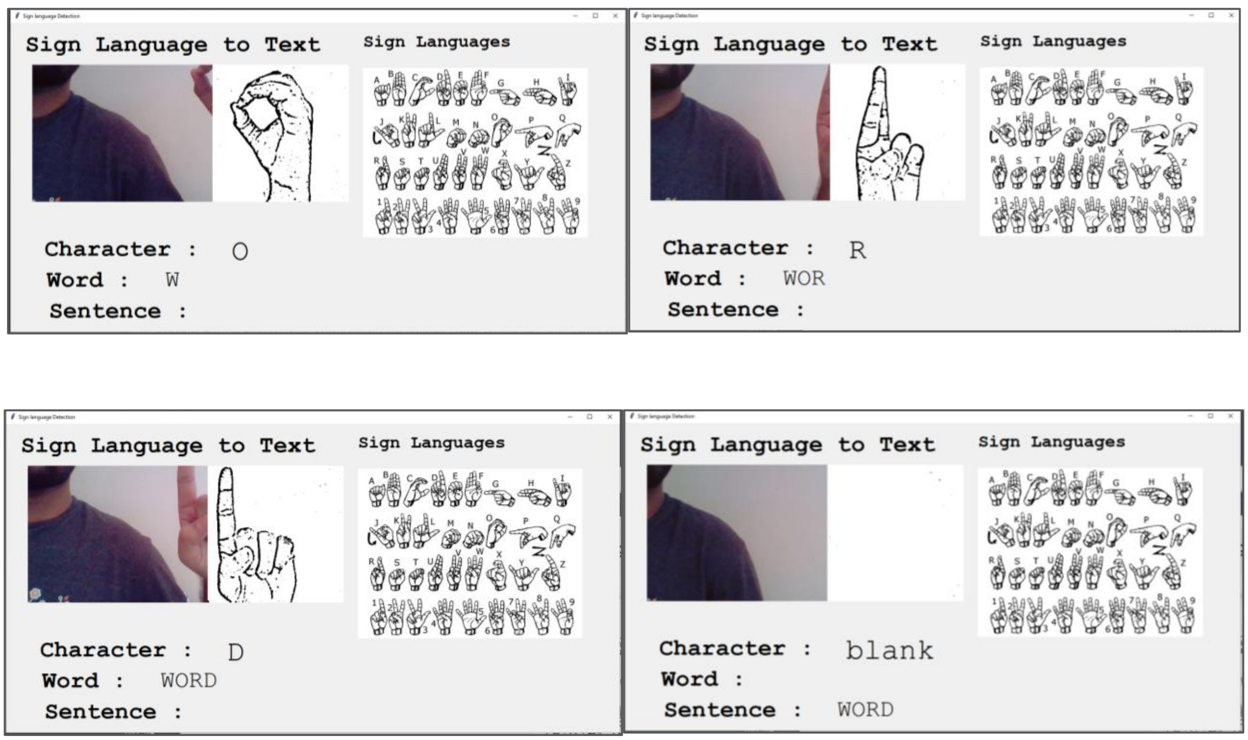


Fig.8.1 Screenshots: Walkthrough

Complete walkthrough of word creation from the symbol and eventually the sentence creation can be seen this these images of Fig 8.1 and Fig 8.2

8.2 Screenshot

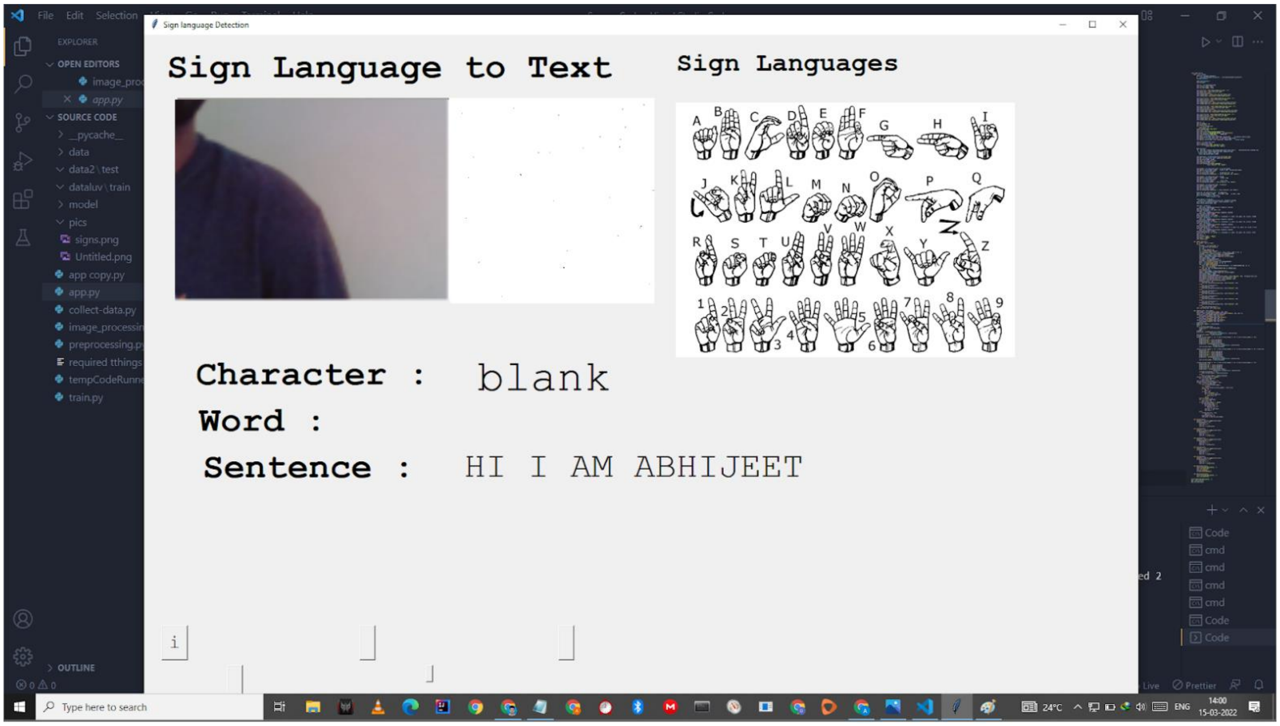


Fig.8.2 Screenshots: Sentence Creation using Finger Spelling

CHAPTER 9

CONCLUSIONS

9.1 CONCLUSIONS

With the use of Training Data, the model will be trained, and it will recognize the input data when the Integrated Camera detects a live hand gesture. The model learns by processing training data to predict the correct output in the testing phase. Using this, the Interaction Barrier between people will be filled by Machine Learning. For ASL alphabets, this paper has created a functional real-time vision-based American Sign Language recognition system for D&M individuals. The final accuracy that was so far made on the usage of the dataset that was used is 98.0 per cent which is perfect. After creating a two-layer algorithm, the improvement can be done on the predictions by identifying and predicting more similar symbols. Almost any symbol can be detected this way, provided there is no background noise, and it is displayed correctly with good lighting

9.2 FUTURE WORK:

We are planning to achieve higher accuracy even in the case of complex backgrounds by trying out various background subtraction algorithms. We are also thinking of improving the preprocessing to predict gestures in low light conditions with higher accuracy. We can provide support for multiple languages using the techniques of natural language processing.

9.3 APPLICATIONS

- It can provide an easy conversation between dumb or deaf people and other people.
- It can be used to teach new learners of sign languages easily and effectively, in a faster way.
- It can greatly help in the effective education of dumb or deaf people so that they can learn and connect with the outer world easily.
- As the application is more portable,it can be used anywhere and at any time in the world without the requirement of the internet being offline.

Appendix A

The class P consists of solvable problems in polynomial time, i.e. solvable problems can solve these problems in time $O(n^k)$ in the worst case, where k is constant.
These problems are tractable, and others are called intractable or super polynomial.
Generally, an algorithm is a polynomial-time algorithm if there exists a polynomial p(n) such that the algorithm can solve any instance of size n in a time O(p(n)).
Problems requiring $\Omega(n^{50})$ time to solve are essentially intractable for large n. Most known polynomial-time algorithms run in time $O(n^k)$ for a relatively low value of k.
The advantage in the class of polynomial-time algorithms is that all reasonable deterministic single-processor models of computation can be simulated on each other with at most a polynomial slow-d.

Time Complexity: O(l*nt*k).

- Where l and k are layers, t is the training example and n epochs.
- Example:
- To find the time complexity for training a neural network with 4 layers with respective i, j, k and l nodes, with t training examples and n epochs.
- The result was O(nt*(ij+jk+kl)).

Thus the problem is P type Problem

Appendix B

Details of paper publication:

Name of the journal: International Journal of Current Science - IJCSPUB (IJCSPUB.ORG)

International Peer Reviewed & Refereed Journals, Open Access Journal

ISSN: 2250-1770 | Impact factor: 8.17 | ESTD Year: 2011

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 8.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly, Indexing in all major database & Metadata, Citation Generator, Digital Object Identifier(DOI)

Dear Author, Congratulation!!!

Your manuscript with Registration ID: **IJCSP_ 212390** has been **accepted** for publication in the **International Journal of Current Science (IJCSPUB)** | **www.ijcspub.org** | **ISSN: 2250-1770** | **International Peer Reviewed & Refereed Journals, Open Access Journal. IJCSPUB is Peer Review Journal, Refereed Journal, Peer Reviewed Journal Referred Journal and Indexed Journal Open Access Journal Online and Print Journal**

IJCSPUB Impact Factor: 8.17 (Calculated by Google Scholar)

Check your paper status: **https://ijcspub.org/track.php or http://ijcspub.org/Authorhome/alogin.php**

Your Paper Review Report :	
Registration ID:	IJCSP_212390
Title of the Paper:	**SIGN LANGUAGE INTERPRETER USING CONVOLUTION NEURAL NETWORK**
Accepted or Not:	**Accepted**

Criteria	**Points out of 100%**
Continuity	**91%**
Text structure	**96%**
References	**96%**
Understanding and Illustrations	**87%**
Explanatory power	**89%**
Detailing	**92%**
Relevance and practical advice	**92%**

Track Your Paper Details:	
Paper Track Link 1:	**https://ijcspub.org/track.php**
Paper Track Link 2:	**http://ijcspub.org/Authorhome/alogin.php**

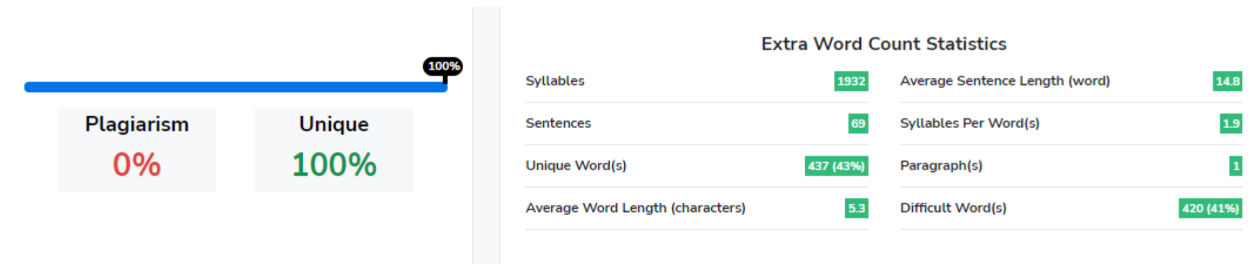
Overall Assessment (Comments.) **Paper Accepted: YES**

Unique Contents: 87% **Paper Accepted**

Reviewer comment store in online RMS system. Paper Accepted complete below Step1 and Step2 process and publish your paper within 1 to 2 day.

Appendix C

Plagiarism Report of the project report in section (each section is of 1000 words and results for the same).

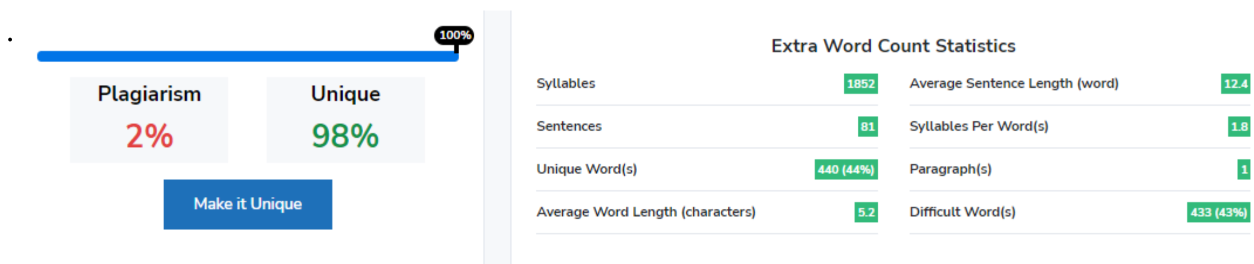


100%

Plagiarism	Unique
0%	100%

Extra Word Count Statistics

Syllables	1932	Average Sentence Length (word)	14.8
Sentences	69	Syllables Per Word(s)	1.9
Unique Word(s)	437 (43%)	Paragraph(s)	1
Average Word Length (characters)	5.3	Difficult Word(s)	420 (41%)



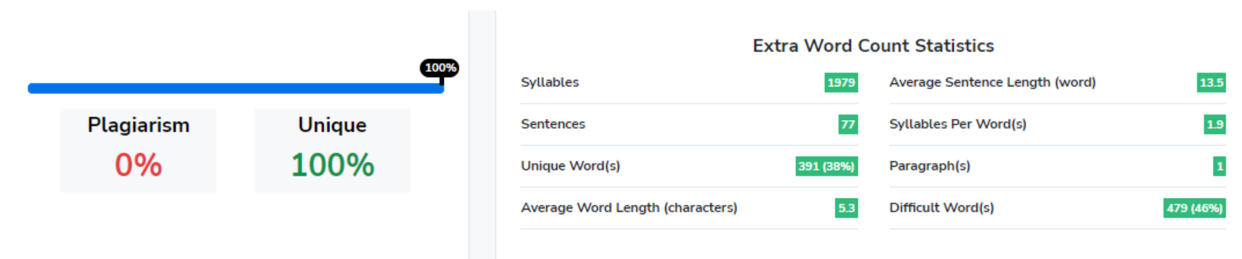
100%

Plagiarism	Unique
2%	98%

Make it Unique

Extra Word Count Statistics

Syllables	1852	Average Sentence Length (word)	12.4
Sentences	81	Syllables Per Word(s)	1.8
Unique Word(s)	440 (44%)	Paragraph(s)	1
Average Word Length (characters)	5.2	Difficult Word(s)	433 (43%)

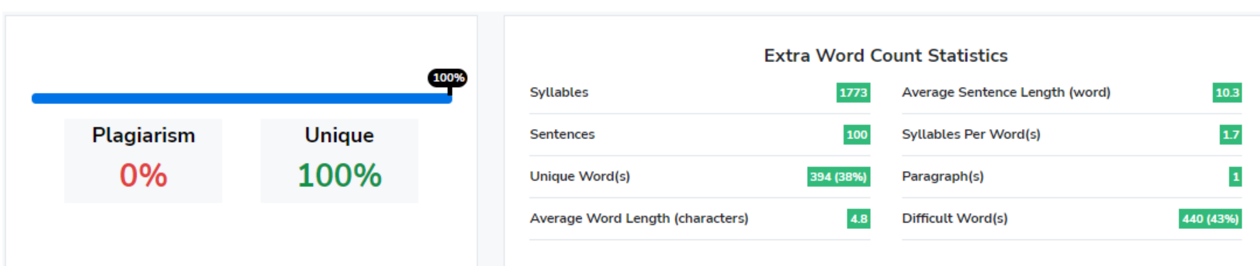


100%

Plagiarism	Unique
0%	100%

Extra Word Count Statistics

Syllables	1979	Average Sentence Length (word)	13.5
Sentences	77	Syllables Per Word(s)	1.9
Unique Word(s)	391 (38%)	Paragraph(s)	1
Average Word Length (characters)	5.3	Difficult Word(s)	479 (46%)



100%

Plagiarism	Unique
0%	100%

Extra Word Count Statistics

Syllables	1773	Average Sentence Length (word)	10.3
Sentences	100	Syllables Per Word(s)	1.7
Unique Word(s)	394 (38%)	Paragraph(s)	1
Average Word Length (characters)	4.8	Difficult Word(s)	440 (43%)

Plagiarism	Unique
2%	98%

Make it Unique

Extra Word Count Statistics

Syllables	1872	Average Sentence Length (word)	9.5
Sentences	110	Syllables Per Word(s)	1.8
Unique Word(s)	428 (41%)	Paragraph(s)	1
Average Word Length (characters)	5	Difficult Word(s)	425 (41%)

Plagiarism	Unique
0%	100%

Extra Word Count Statistics

Syllables	524	Average Sentence Length (word)	16.4
Sentences	18	Syllables Per Word(s)	1.8
Unique Word(s)	177 (60%)	Paragraph(s)	1
Average Word Length (characters)	5	Difficult Word(s)	102 (34%)

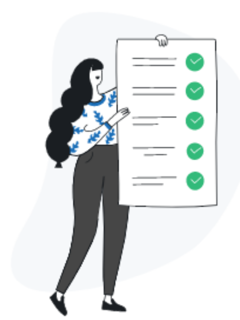
Reading Time 2 mins Speak Time 3 mins

Sentence Creation using Finger Spelling CHAPTER 9 CONCLUSIONS 9.1 CONCLUSIONS With the use of Training Data, the model will be trained, and it will recognize the input data when the Integrated Camera detects a live hand gesture. The model learns by processing training data to predict the correct output in the testing phase. Using this, the Interaction Barrier between people will be filled by Machine Learning. For ASL alphabets, this paper has created a functional real-time vision-based American Sign Language recognition system for D&M individuals. The final accuracy that was so far made on the usage of the dataset that was used is 98.0 per cent which is perfect. After creating a two-layer algorithm, the improvement can be done on the predictions by identifying and predicting more similar symbols. Almost any symbol can be detected this way, provided there is no background noise, and it is displayed correctly with good lighting 9.2 FUTURE WORK: We intend to test several background subtraction techniques to improve accuracy even in the situation of complicated backgrounds. We're also considering upgrading the preprocessing to better predict gestures in low-light situations. We can provide support for multiple languages using the techniques of natural language processing. 9.3 APPLICATIONS ● It can provide an easy conversation between dumb or deaf people and other people. ● It can be used to teach new learners of sign languages easily and effectively, in a faster way. ● It can greatly help in the effective education of dumb or deaf people so that they can learn and connect with the outer world easily. ● As the application is more portable,it can be used anywhere and at any time

Text To Speech

Total Words: 0 Total Characters:

Check Grammar



Congratulation!

No Significant Plagiarism Found

REFERENCES

[1] J. Qin, J. Ashley and B. Xie, "Hand Gesture Recognition Based on a Nonconvex Regularization," *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2021, pp. 187-192, doi: 10.1109/ICMA52036.2021.9512752.

[2] Ahmad Puad Ismail *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* 1045 012043, "Hand gesture recognition on python and OpenCV", *IOP Conf. Series: Materials Science and Engineering*, doi: 10.1088/1757-899x/1045/1/012043

[3] Pranjali Manmode, Rupali Saha, Manisha N. Amnerkar, "Real-Time Hand Gesture Recognition", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN : 2456-3307, Volume 7 Issue 3, pp. 618-624, May-June 2021. Available at doi : https://doi.org/10.32628/CSEIT2173132 Journal URL : https://ijsrcseit.com/CSEIT2173132

[4] W. Zhang, J. Wang and F. Lan, "Dynamic hand gesture recognition based on short-term sampling neural networks," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 1, pp. 110-120, January 2021, doi: 10.1109/JAS.2020.1003465.

[5] AlSaedi, Ahmed & Hanon AlAsadi, Abbas. (2020). "A new hand gestures recognition system." Indonesian Journal of Electrical Engineering and Computer Science. 18. 49. 10.11591/ijeecs.v18.i1.pp49-55.

[6] Tran, Dinh-Son & Ho, Ngoc-Huynh & Yang, Hyung-Jeong & Baek, Eu-Tteum & Kim, S.H. & Lee, Guee. (2020). "Real-Time Hand Gesture Spotting and Recognition Using RGB-D Camera and 3D Convolutional Neural Network." Applied Sciences. 10. 722. 10.3390/app10020722.

[7] V. Aiswarya, N. Naren Raju, S. S. Johanan Joy, T. Nagarajan and P. Vijayalakshmi, "Hidden Markov Model-Based Sign Language to Speech Conversion System in TAMIL," 2018 Fourth International Conference on Biosignals, Images and Instrumentation (ICBSII), 2018, pp. 206-212, doi: 10.1109/ICBSII.2018.8524802.

[8] Dr. Uttara Gogate, Vahaz Kazi, Aakash Jadhav, Tirthesh Bari "Hand Gesture Recognition", *International Journal for Research in Applied Science and Engineering Technology, (IJRASET) 2018*, doi: 10.22214/ijraset.2022.41625

[9] Oudah M, Al-Naji A, Chahl J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. Journal of Imaging. 2020; 6(8):73. doi: 10.3390/jimaging6080073

[10] Pal, Devendrakumar H. and S. M. Kakade. "Dynamic hand gesture recognition using kinect sensor." 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), doi: 10.1109/ICGTSPICC.2016.7955343

[11] Yang, Wenwen & Tao, Jinxu & Xi, Changfeng & Ye, Zhongfu. (2015). Sign Language Recognition System Based on Weighted Hidden Markov Model. 449-452. doi: 10.1109/ISCID.2015.254.