SIGN LANGUAGE RECOGNITION SYSTEM USING MACHINE LEARNING

# A PROJECT REPORT

***Submitted by***

**SAI VIKRAM KARTHIKEYAN 312320104119**

# SANJAY GANESH.M 312320104123

***in partial fulfilment of the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



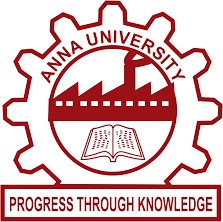
# St. JOSEPH’S COLLEGE OF ENGINEERING

**(An Autonomous Institution) OMR, Chennai 600 119**

# ANNA UNIVERSITY :: CHENNAI 600 025

**MARCH 2024**

**ANNA UNIVERSITY, CHENNAI**



# BONAFIDE CERTIFICATE

Certified that this project report **“SIGN LANGUAGE RECOGNITION SYSTEM USING MACHINE LEARNING”** is the bonafide work of **SAI VIKRAM KARTHIKEYAN (312320104119) AND SANJAY GANESH.M (312320104123)** who carried out the work under my guidance. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Dr. G. Maria Kalavathy M.E.,M.B.A., Ph.D., | Ms.T.Jenitha, M.E |
| Professor & Head of Department | Assistant Professor, |
| Dept. of Computer Science and Engineering, | Dept. of Computer Science and Engineering,, |
| St. Joseph’s college of Engineering, | St. Joseph’s college of Engineering, |
| OMR,Chennai-600 119 | OMR Chennai-600 119 |

# Submitted to Project and Viva Examination held on

**INTERNAL EXAMINER EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

At the outset, we would like to express our sincere gratitude to our beloved **Dr. B. Babu Manoharan M.A., M.B.A., Ph.D., *Chairman, St. Joseph’s Group of Institutions*** for his constant guidance and support to the student community and the Society.

We would like to express our hearty thanks to our respected ***Managing Director* Mr. B. Shashi Sekar, M.Sc.** for his kind encouragement and blessings.

We wish to express our sincere thanks to the ***Executive*** ***Director***

**Mrs. S. Jessie Priya M.Com.** for providing ample facilities in the institution.

We express sincere gratitude to our beloved **Principal Dr*.* Vaddi Seshagiri Rao M.E., M.B.A., Ph.D., F.I.E.** for his inspirational ideas during the course of the project.

We express our sincere gratitude to our beloved Dean ***(Research*) Dr. A. Chandrasekar M.E., Ph.D., Dean (Student Affairs) Dr. V. Vallinayagam M.Sc., M.Phil., Ph.D., and Dean (Academics) Dr. G. Sreekumar M.Sc., M.Tech., Ph.D.,** for their inspirational ideas during the course of the project.

We wish to express our sincere thanks to **Dr. G. Maria Kalavathy M.E.,M.B.A., Ph.D., *Professor* & *Head of the Department***, Department of Computer Science and Engineering, St. Joseph’s College of Engineerin***g*** for her guidance and assistance in solving the various intricacies involved in the project.

We would like to acknowledge our profound gratitude to our supervisor **Ms. T. Jenitha M.E.** for her/his expert guidance and connoisseur suggestion to carry out the study successfully.

Finally, we thank the **Faculty Members** and **our Family**, who helped and encouraged us constantly to complete the project successfully.

# ABSTRACT

Sign language is a technique used for communication purposes by deaf people. American sign language (ASL) is the main communication language of deaf and people who have hard hearing around the world. Communication shouldn’t be a barrier between people as it should be accessible to the specially impaired people (that is deaf and dumb people in our case) as similar to everyone and also it can help in schooling systems to teach those specially impaired people to understand and converse with everyone seamlessly that is the goal of our project. For solving the problem, Sign language recognition is proposed of static ASL using Deep Learning. The contribution consists of two solutions to the problem. The first one is resized with Bicubic static ASL binary images. The second solution is to classify the 26 alphabets of static characters of ASL using Convolution Neural Network and Deep Learning. This proposed model presents a Sign Language Recognition System that utilizes convex hull algorithm techniques for the accurate recognition of sign language gestures. The proposed system uses a convolutional neural network for feature extraction from the input video stream. The extracted features are then fed to a recurrent neural network (RNN) for temporal modeling and classification. The system achieves high accuracy by utilizing a long short-term memory layer, which effectively captures temporal dependencies between sign language gestures.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | ABSTRACT | iv |
|  | LIST OF FIGURES | vii |
|  | LIST OF ABBREVIATIONS | viii |
| **1** | **INTRODUCTION** |  |
|  | * 1. PROBLEM STATEMENT | 1 |
|  | 1.2 HEURISTIC | 1 |
|  | 1.3 SYSTEM OVERVIEW | 2 |
|  | 1.4 SCOPE OF THE PROJECT | 2 |
| **2** | **LITERATURE SURVEY** | 3 |
| **3** | **SYSTEM ANALYSIS** |  |
|  | 3.1 EXISTING SYSTEM  3.1.1 Disadvantages of existing system` | 11  11 |
|  | 3.2 PROPOSED SYSTEM | 12 |
|  | 3.2.1 Advantages of proposed system | 12 |
|  | 3.3 REQUIREMENTS SPECIFICATION | 12 |
|  | 3.3.1 software requirements | 13 |
|  | 3.3.2 hardware requirements | 13 |
|  | 3.4 LANGUAGE SPECIFICATION | 13 |
| **4** | **SYSTEM DESIGN** |  |
|  | 4.1 SYSTEM ARCHITECTURE | 15 |
|  | 4.2 UML DIAGRAM | 16 |
|  | 4.3 STATE DIAGRAM | 16 |
|  | 4.4 SEQUENCE DIAGRAM | 17 |
|  | 4.5 USE CASE DIAGRAM | 18 |
|  | 4.6 BLOCK DIAGRAM | 19 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **5 MODULE DESCRIPTION** |  |
| 5.1 LIST OF MODULES | 20 |
| 5.1.1 camera module | 20 |
| 5.1.2 detection module | 20 |
| 5.4 GESTURE RECOGNITION | 21 |
| 5.5 CONVEX HULL METHOD | 21 |
| 5.6 GESTURE CONTROLS | 22 |
| **6 CONCLUSION AND ENHANCEMENT** |  |
| 6.1 CONCLUSION | 23 |  |
| 6.2 FUTURE ENHANCEMENT | 23 |  |
| **APPENDIX I** | 25 |  |
| **APPENDIX II** | 38 |  |
| **REFERENCES** | 40 |  |

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGENO** |
| **4.1** | SYSTEM ARCHITECTURE | 14 |
| **4.2** | UML DIAGRAM | 14 |
| **4.3** | STATE DIAGRAM | 15 |
| **4.4** | SEQUENCE DIAGRAM | 16 |
| **4.5** | USE CASE DIAGRAM | 16 |
| **4.6** | BLOCK DIAGRAM | 17 |

**LIST OF FIGURES**

# LIST OF ABBREVIATION

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVIATION** | **DEFINITION** |
| **1** | SQL | Structured Query Language |
| **2** | CNN | Convolutional Neural Network |
| **3** | DBMS | Database Management System |
| **4** | ORM | Object Relational Manager |

# CHAPTER 1

# INTRODUCTION

* 1. **PROBLEM OVERVIEW**

There are many applications where hand gestures can be used for interaction with systems like video games, controlling UAVs, medical equipment, etc. These hand gestures can also be used by handicapped people to interact with the systems. Classical interactions tools like keyboard, mouse, touchscreen, etc. may limit the way we use the system.The main objective of this project is to design a software that can help the people understand American sign language through the signs made by deaf and dumb people.This can help in understanding the signs made by hearing impaired people to people in their day to day life.It focuses on the use of this system in education systems to break the difference between those students and new students.Language support is added so that the people can understand the signs through the description shown for each sign of American sign language.

* 1. **HEURISTIC**

As mentioned previously, skin color models can be used to detect hands and face; the human face won't move so the hands will be the only movable objects in the foreground scene. Thus , background subtraction will be the best solution to find hand area and track hand movements. To get the best background model an average filter is used .Average filter will create an initial background from the first N frame as shown in regions but it's not efficient in complex backgrounds

* 1. **SYSTEM OVERVIEW**

Gesture recognition is an active research field in Human-Computer Interaction technology. It has many applications in virtual environment control and sign language translation, robot control, or music creation. In this machine learning project on Hand Gesture Recognition, we are going to make a real-time Hand Gesture Recognizer using the Media Pipe framework and Tensor flow in Open-CV and Python.

* 1. **SCOPE OF THE PROJECT**

The scope of the project is to develop a gesture recognition system that can identify hand movements in real-time, even in varying lighting conditions. This system will use computer vision and artificial intelligence to interpret the gestures made by a user, and translate them into specific commands or actions.One of the key objectives of this project is to develop a complete system that is easy to use, user-friendly, and does not require any specialized hardware. Instead, the system will use a combination of software and basic hardware components to capture and digitize images of the user's hand movements, which can then be analyzed and classified by the gesture recognition algorithm.To achieve this goal, the system will use a variety of techniques and algorithms to analyze the user's hand movements, including image segmentation, feature extraction, and classification. The system will also need to be able to adapt to different users and different environments, and be able to recognize a wide range of different gestures and hand movements.The system will likely require a significant amount of training and testing to ensure that it can accurately recognize and classify different gestures in real-time. This will involve collecting and labeling large amounts of data, training the gesture recognition algorithm on this data, and then fine-tuning the system based on feedback from users.

# CHAPTER-2 LITERATURE SURVEY

1. **Herzig, Melissa, and Thomas E. Allen. "Deaf children’s engagement with American Sign Language-English bilingual storybook apps." *The Journal of Deaf Studies and Deaf Education* 28.1 (2023): 53-67.**

Designed features of American Sign Language (ASL)-English bilingual storybook apps on tablet computers, based on learning research, were intended to facilitate independent and interactive learning of English print literacy and ASL skill among young learners. In 2013, the Science of Learning Center on Visual Language and Visual Learning introduced the first in a series of storybook apps for the iPad based on literacy and reading research. A study, employing a sample of signing deaf children, examined children’s self-motivated engagement with the various design features presented in the earliest of the apps, The Baobab, and analyzed the relationships of engagement with ASL skill and age of first exposure to ASL, ASL narrative ability, and grade-appropriate English reading ability. Results indicated a robust level of engagement with the app, and a relationship between app pages specifically targeting reading and early exposure and skill levels in ASL. No evidence of relationships between narrative and vocabulary skills and app reading engagement was found. Topics for future research, and strategies for app improvement were discussed.

# Alawajee, Omar. "Exploring the sign language proficiency of university undergraduate students in a preservices preparation program for teachers of deaf students." *Higher Education Pedagogies* 7.1 (2022): 65-87.

The study used an exploratory study design with questionnaires distributed to a convenient sample of female undergraduate students enrolled in a teacher pre- enlistment program for Deaf students. . Research subjects are evaluated based on their mastery of lexical signs, symbolic lexical signs and arbitrary lexical signs. given an arbitrary lexical symbol domain. The study also found that participants' grade point average (GPA) had a significant effect on their overall sign language proficiency score. on participants' overall sign language proficiency scores. This suggests that while these factors may be important, they are not the only factors that contribute to a person's sign language fluency. Based on the findings of the study, the author recommends that future research should focus on developing and improving teacher-student sign language skills. The author also recommends that initial teacher training program designers pay more attention to developing learners' sign language skills, especially in the area of arbitrary lexical symbols. emphasizes the need for continued research and development in this area.Communication through sign language is essential for teachers of deaf students. This study sought to assess and evaluate the sign language proficiency of preservice teachers of deaf students to help preservice teacher preparation program designers identify what aspects of sign language need to be focused on and provide recommendations to improve preservice teachers’ sign language levels. An exploratory research design was used through questionnaires distributed to a convenience sample.

# Ott, Laura E., Linda C. Hodges, and William R. LaCourse. "Supporting deaf students in undergraduate research experiences: Perspectives of American Sign Language interpreters." *Journal of Microbiology & Biology Education* 21.1 (2020): 20.

Deaf undergraduates were eager to engage in research but often felt marginalized due to the lack of appropriate accommodations that allowed for effective communication within heterogeneous research teams consisting of hearing peers and/or mentors. In this case study, four American Sign Language (ASL) interpreters were interviewed, who provided full-time accommodations to teams consisting of one deaf student and two hearing peers during a six-week internship. The interpreters were queried on their role and experiences in supporting the research teams. The findings indicated that the interpreters could be a valuable asset to heterogeneous teams by supporting both deaf and hearing individuals and advocating for the deaf student. That said, interpreters also had to overcome challenges unique to interpreting in the research environment, such as deciding when and how to interpret. The insights provided by the interpreters interviewed here were valuable as undergraduate research programs evaluated how to provide appropriate accommodations to deaf students engaged in research. In addition, they also highlighted the need for research experience coordinators and mentors to consider supporting diverse teams in developing effective communication strategies and applying universal design for learning to the research environment.The inclusion of student populations traditionally underrepresented in the STEM fields was a major emphasis of STEM educational reform (1–3). Undergraduate research was one of many interventions that had been found to be impactful for promoting the success and retention of underrepresented minority (URM) students in STEM fields (4–8).

# Reagan, Timothy, Paula E. Matlins, and C. David Pielick. "Deaf Epistemology, Sign Language and the Education of d/Deaf Children." *Educational Studies* 57.1 (2021): 37-57.

One of the traditional areas of concern in educational foundations had been philosophy of education, and within philosophy of education, a central focus had been epistemology. Defined as the branch of philosophy dealing with the study of knowledge, in the past half-century approaches to the study of epistemology had evolved in significant ways. The rise of ethno epistemology was perhaps one of the clearest examples of such changes. After a brief overview of the concept of ethno epistemology, we argued for the existence of a distinctive deaf epistemology. We then offered a discussion of some of the ways in which this deaf epistemology was reflected in American Sign Language (ASL). The article concluded with a discussion of the ramifications of deaf epistemology for deaf education.The deaf cultural and linguistic community was referred to in many different ways: it was called the “deaf community,” the “deaf culture,” “Deafhood,” and in ASL, it was referred to using the sign DEAF-WORLD. Historically, the term “deaf” was virtually always used to describe individuals with neurologically impaired hearing. As awareness of sign languages grew, so too did an understanding of sign language users as members of a distinctive linguistic and cultural community. In order to indicate the difference between individuals who were audiologically deaf and those who were culturally and linguistically deaf, in the 1960s and 1970s it became common to adopt a distinction between deaf and Deaf: the former referring to deafness solely as an audiological condition, while the latter referring to Deafness as a linguistic and cultural condition. Although the arguments presented here applied to deaf people in all countries, the specific case with which this article was concerned was that of the deaf community in the United States.

1. **Hussein, Karim Q., and Maha A. Al-Bayati. "Multi-Mode e-Learning System of Reading Skills for Deaf Students Based on Visual Multimedia." *International Journal of Interactive Mobile Technologies* 16.10 (2022).**

The study investigated the processing of written words by ASL-English bilingual deaf middle school students, and found that deaf bilinguals were faster to respond to English word pairs with phonologically related translations in ASL than to English word pairs with unrelated translations, but no difference was found for hearing controls with no knowledge of ASL. The results indicated that co-activation of signs and written words was not the outcome of years of bilingual experience, but instead characterized bilingual language development.Previous studies of deaf and hearing bilinguals using a signed and a spoken language demonstrated that cross-language activation was modality-independent, challenging notions that the primary driver of language nonselective processing is ambiguous word forms such as cognates and interlingual homophones. Signed and spoken languages have very little articulatory or perceptual overlap, and signed languages have no standardized, widely-used written systems.Co-activation among unimodal bilinguals had been studied in hearing adults, and to a lesser extent, children. Importantly, while cross-language activation of spoken languages could be investigated in toddlers, signing bilingual participants had to be sufficiently proficient in a signed language and the written form of a spoken language.The system was developed using the ADDIE model, which consists of five stages: analysis, design, development, implementation, and evaluation. The authors describe the system's architecture, which includes a web-based platform and a mobile application.

# Jones, Gabrielle A., Dawei Ni, and Wei Wang. "Nothing about us without us: Deaf education and sign language access in China." *Deafness & Education International* 23.3 (2021): 179-200.

This empirical qualitative study aimed to understand the Chinese context in promoting accessible high-quality education for deaf communities and to create an opportunity for deaf experts to contribute to sign language research, instruction, interpreting programmes, and deaf education in China. Using a focus group methodology, data was gathered from 48 participants from four different stakeholder groups (10 teachers, 16 administrators/researchers, 6 interpreters, 16 community members) identifying concerns and solutions to achieving educational access. Video recorded discussions were transcribed, analyzed and consolidated into themes.Results showed a fragile trust between deaf and hearing professionals, a need for continued investigation on sign language standardization and preservation, and a desire for worldwide collaboration and inclusion of deaf and hearing scholars in establishing a deaf university in China. This participatory, community-based research method yielded insights toward improving deaf education and sign language training within Chinese special education and toward the design, implementation and establishment of a future university serving only deaf students in China. As with other forms of education, high self-efficacy often encourages greater student confidence and autonomy. While self-efficacy has been shown to have positive effects in face-to-face education, its antecedents and consequences in online distance education are less clear. This study addressed this issue.

# Shoba, S., et al."Sign Language Recognition for Daily Activities Using Deep Learning." Deep Learning Research Applications for Natural Language Processing. *IGI Global*, 2022. 204-217.

Sign language recognition had become a critical research area in the field of computer vision as the need for disability solutions was growing. Sign language acted as a bridge to reduce the communication gap between normal people and deaf and dumb people. However, previous sign language identification systems lacked essential characteristics such as accessibility and cost, which were critical for people with speech disabilities to interact with their daily settings. An attractive solution was initiated for sign languages in terms of words and common expressions for daily activities. This helped to interact with deaf and dumb people by connecting them to the outside world more quickly and easily. The sign gestures obtained were processed through popular machine learning and deep learning models for classification accuracy. This chapter discussed word sign recognition, image processing algorithms for separating the signs from the background, machine learning algorithms, and the complete model set up for sign recognition. Communication was a key mechanism by which information sharing was done among people. People with disabilities needed some way of communication to deal with normal and disabled people. However, this was a challenging and difficult process when normal and deaf and mute people tried to communicate.The best and most effective way of communication for deaf and dumb persons was sign language. Sign language communication not only benefited the communication between the deaf and dumb but also benefited the communication between normal and deaf and dumb people. A deaf and mute person used hand gestures for communication. Also, for normal people understanding their gestures was a complicated task.

# DeCastro, Giulia Zanon, Rúbia Reis Guerra, and Frederico Gadelha Guimarães. "Automatic translation of sign language with multi-stream 3D CNN and generation of artificial depth maps." *Expert Systems with Applications* 215 (2023): 119394*.*

Sign languages have played an essential role in the cognitive and social development of the deaf, consisting of a natural form of communication and being a symbol of identity and culture. However, hearing loss had a severe social impact due to an existing communication barrier, preventing access to essential services such as education and health. A bi-directional sign language translation was considered as the solution to bridging the communication gap between the deaf and the listener, completing a two-way communication cycle. Virtual personal assistants could benefit from this technology by extending how users interacted with the intelligent system. With this idea, a multi-stream deep learning model was developed to recognize signs of Brazilian (BSL), Indian (ISL), and Korean (KSL) Sign Languages. Different types of information were combined for the classification task, using single-stream and multi-stream 3D Convolutional Neural Networks. In addition, considering the largest source of sign data globally – the internet – a depth sensor-free classification method was proposed, with depth maps artificially generated through Generative Adversarial Networks. In order to consider the main parameters that encoded sign languages, the final architecture was composed of a multi-stream network that received the segmented hands, the faces, the distances and speeds of the points of articulation, and the RGB frames associated with artificial depth maps. Finally, a visual explanation was provided to understand which regions were important for model decision- making.

# CHAPTER 3 SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

In recent decades, due to computer software and hardware technologies of continuous innovation and breakthrough, social life and information technology have a very close relationship in the twenty-first century. In the future, especially the interfaces of consumer electronics products (e.g. smartphones, games and infotainment systems) will have more and more functions and be complex. How to develop a convenient human-machineInterface (HumanMachine Interaction/Interface, HMI) for each consumer electronics product has become an important issue. The traditional electronic input devices, such as mouse, keyboard, and joystick are still the most common interaction way. However, it does not mean that these devices are the most convenient and natural input devices for most users. Since ancient times, gestures have been a major way for communication and interaction between people. People can easily express the idea by gestures before the invention of language

**3.1.1 DISADVANTAGES OF EXISTING SYSTEM**

1. Limited Accuracy: Existing systems may suffer from limited accuracy in recognizing and interpreting sign language gestures, leading to misinterpretations and communication errors.
2. High Costs High-quality sign language recognition systems can be costly to develop and maintain, making them less accessible for smaller organizations or individuals
   1. **PROPOSED SYSTEM**

Most gesture recognition methods usually contain three major stages. The first stage is object detection. The target of this stage is to detect hand objects in the digital images or videos. Many environment and image problems are needed to solve at this stage to ensure that the hand contours or regions can be extracted precisely to enhance the recognition accuracy. Common image problems contain unstable brightness, noise, poor resolution and contrast. The better environment and camera devices can effectively improve these problems. However, it is hard to control when the gesture recognition system is working in the real environment or is becoming a product. Hence, the image processing method is a better solution to solve these image problems to construct an adaptive and robust gesture recognition system. The second

stage is object recognition. The detected hand objects are recognized to identify the gestures. At this stage, differentiated features and effective classifiers selection are a major issue in most research. The third stage is to analyze sequential gestures to identify users’ instructions or behaviors.

* + 1. **ADVANTAGES OF PROPOSED SYSTEM**

1. Reduce external Interface The Advantage of System is to Reduce External Interface like Mouse And Keyboard.
2. High Portability The proposed System reduce the working of external interface like keyboard and mouse so it makes it high portable
   1. **REQUIREMENT SPECIFICATION**

Python, NumPy and OpenCV as they are the prerequisites for this.Image basics (such as pixels, dimensions etc) and some basic operations with images such as Thresholding and Segmentation.

* + 1. **SOFTWARE REQUIREMENTS**

1. Anaconda is an open-source distribution for python and R programming languages. It is utilized for information science, machine learning, profound learning, and so on. With the availability of more than 300 libraries for information science, it turns out to be genuinely ideal for any developer to work on anaconda for information science
2. OpenCV :(Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itself (which was later acquired by Intel). The library is cross-platform and free for use under the open- source BSD license.
3. Numpy is a general-purpose array-processing package. It provides a

high-performance multidimensional array object, and tools for working with these arrays.

1. Py-Charm is an integrated development environment (IDE) used in computer programming, specifically for the Python programming language. Python Integrated Development Environment (IDE) providing a wide range of tools.

**3.3.2 HARDWARE REQUIREMENTS**

* + - Microsoft Windows XP Professional /Windows 7 Professional /Windows 10
    - Processor: 800MHz Intel Pentium III or equivalent
    - Memory: 4 GB
    - Disk space: 1GB of free disk space
    - System Type: 64-bit operating system, x64-based processor
    - Web cam (For real-time hand Detection)
  1. **LANGUAGE SPECIFICATION**

Python is an object-oriented language, meaning that it allows for the creation and manipulation of objects, which can be used to represent real-world concepts or abstract ideas. It also supports functional programming concepts like lambda functions and map/reduce/filter operations.Python has a vast standard library, which includes modules for tasks like web development, data analysis, and scientific computing. Additionally, there is a large and active community of developers who contribute to a rich ecosystem of third-party libraries and tools.

Some of the key features of Python include:

1. Dynamic typing: Variables don't need to be explicitly declared with a data type.
2. Garbage collection: Python automatically frees up memory that is no longer being used by the program.

# CHAPTER 4 SYSTEM DESIGN

* 1. **SYSTEM ARCHITECTURE**

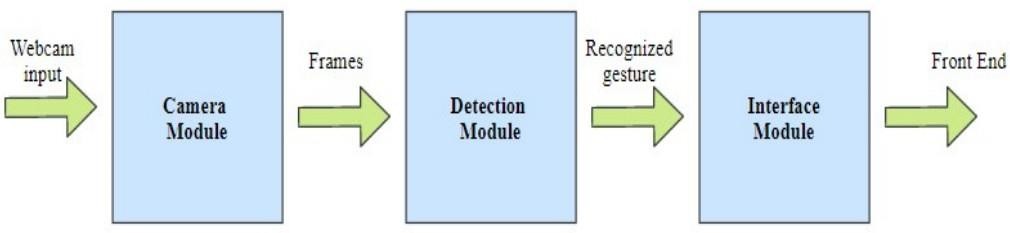


Fig 4.1 System Architecture

Fig 4.1 refers that the system architecture includes a camera module, a detection module, and an interface module. The camera module captures the sign language gesture and converts it into a digital signal, which is then passed to the detection module. The detection module extracts features from the signal, such as hand shape and motion, and analyzes them using machine learning algorithms to recognize the gesture. Once the gesture is recognized, the interface module outputs the corresponding text output to enable communication. This architecture is a key enabler for effective communication between people who use sign language and those who do not.

* 1. **UML DIAGRAM**

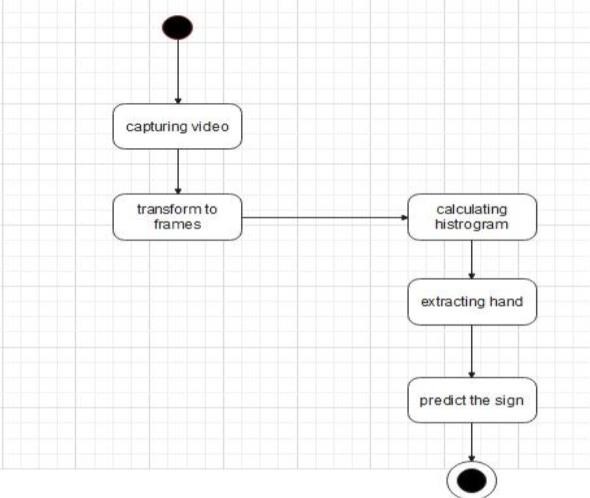


Fig 4.2 Uml Diagram

Fig 4.2 states that sign language recognition system is a technology that enables effective communication between sign language users and non-users. System architecture can be modeled using UML (Unified Modeling Language) diagrams, which include several components such as actors, use cases, classes and their relationships.

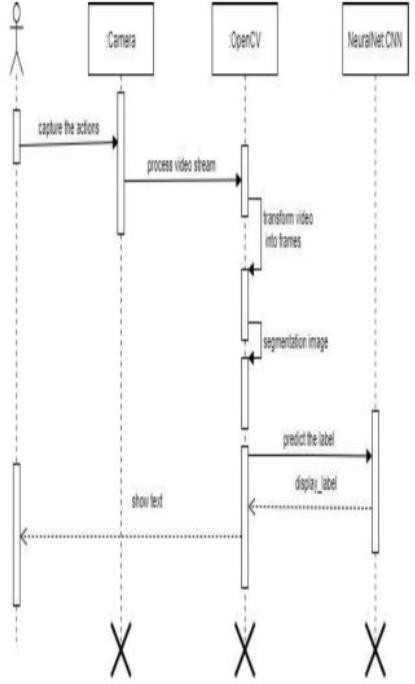
* 1. **STATECHART DIAGRAM**

Fig 4.3 Statechart Diagram

Fig 4.3 refers to a statechart diagram that is useful in modeling the behavior of a sign language recognition system, as it represents the system's states and transitions between them. The statechart diagram can include states such as idle, capturing gesture, analyzing gesture, and displaying output.

* 1. **SEQUENCE DIAGRAM**

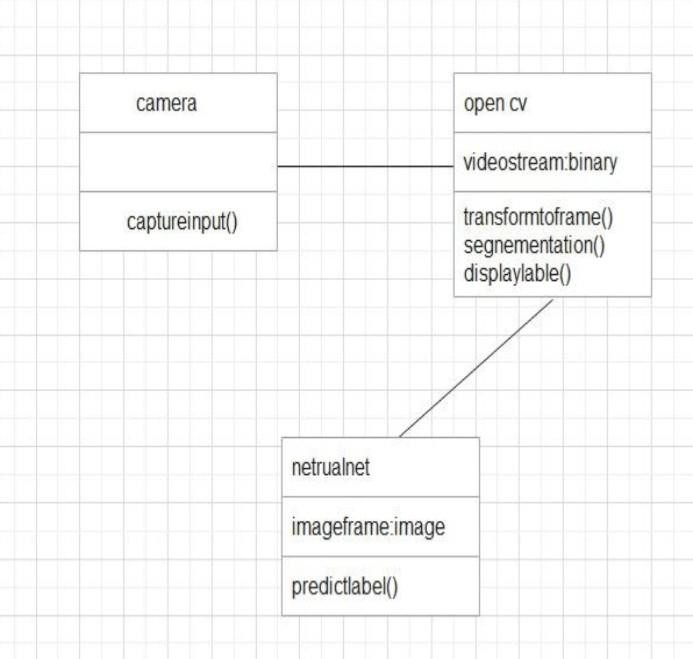


Fig 4.4 Sequence Diagram

Fig 4.4 refers to the sequence diagram that illustrates the sequence of events that occur in a sign language recognition system. It starts with the user starting the system, and the system prompting the user to perform a sign language gesture.Once the user performs a sign language gesture, the system captures the video feed of the gesture and sends it to the gesture recognition module. The gesture recognition module analyzes the video feed and recognizes the gesture, which is then sent to the sign language dictionary.

* 1. **USE CASE DIAGRAM**

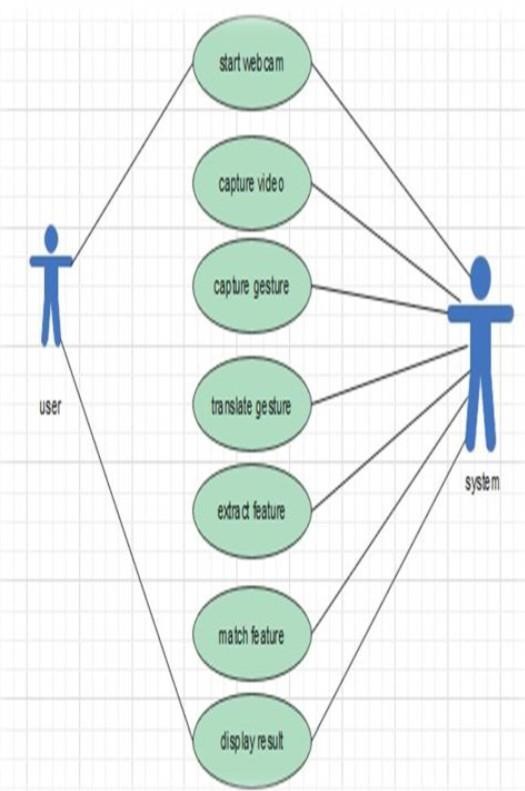


Fig 4.5 Use Case Diagram

Fig 4.5 refers to the use cases diagram describing the main actions that the user can perform with the system. Other use cases, such as configuring the system or updating the sign language dictionary, may also be included in the use case diagram depending on the specific requirements of the system.The use case diagram helps to provide a clear understanding of the system's functionality and the different ways in which the user can interact with it. It can also serve as a starting point for designing the system's user interface and defining the system's requirements.

* 1. **BLOCK DIAGRAM**

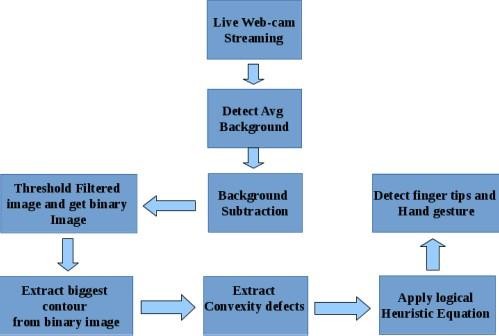


Fig 4.6 Block Diagram

Fig 4.6 refers to the block diagram that shows the different components and how they are connected.The block diagram provides a clear overview of the different components of the sign language recognition system and how they are connected. This can help in the design and development of the system, as well as in troubleshooting any issues that may arise.

# CHAPTER 5 MODULE DESCRIPTION

**5.1 LIST OF MODULES**

**5.1.1 CAMERA MODULE**

This module is subject for interfacing and capturing input through the different sorts of picture markers and sends this picture to the detection module for handling as frames. The generally utilized techniques of capturing and recognizing input are hand belts, data gloves and cameras. In our framework, we use the inbuilt webcam which is financially savvy to see both static and dynamic signs.

**5.1.2 DETECTION MODULE**

This module is liable for the image processing. The output from camera module is presented to different image handling methods, for instance, color conversion, noise removal, thresholding following which the image goes through contour extraction. In the event that the image contains defects, at that point convexity defects are found by which the gesture is identified. In the event that there are no defects, at that point the image is classified utilizing Haar cascade to recognize the gesture.Three methods were susceptible to environmental changes, especially where lighting was involved. This could be viewed as the drawback of using a single RGB camera, which only provides us with the means to differentiate Using color. Even if calibration was used to adjust the thresholds somehow, it is worth noting that lighting can differ even within an environment – a hand beside the window would have a different RGB or HSV from a hand in a darker corner of the same room. Of the three methods, it was determined that background subtraction is the most robust and simplest to implement. The fact that it did not rely on the exact color of the hand meant it was less susceptible to variations in lighting within the environment. It also returned the

segmentation in a binary image, which made the gesture recognition steps much simpler. The primary weakness of this method was dealing with dark areas (i.e. the need to light up the hand), which would be the main constraint moving forward.

**5.2 GESTURE RECOGNITION**

As mentioned earlier, gesture recognition can only come after segmentation is done. Due to the segmentation methods not yielding desired results, this section is not the core focus of this paper. We will explore one method of identifying simple hand gestures, and implement 2 basic gesture controls: cursor movement and mouse click.

**5.3 CONVEX HULL METHOD**

The convex hull method takes the outline of a shape, and identifies the convex and concave (defect) points along the outline. These points provide us with a rough idea of the shape of the object. In the case of a hand, it should have 5 convex points (one for each finger) and 4 defects (one between two adjacent fingers). Using this method, we can identify the number of fingers the user is showing to us by the number of convex and defect points. For example, if there are 2 convex and 2 defect points, it is likely that the user is showing us 2 fingers. In our implementation, we used OpenCV functions such as findContours(), convexHull() and convexityDefects() to obtain the convex hull and defect points. Below is a sample output of the centroid and number of defects for a given detection. Numbers in brackets are (x, y) coordinates. Number on the right is the number of defects .Two issues can be seen in this implementation. The first is that the number of defects is higher than expected, and is likely to be due the shape of the hand not being smooth. The presence of noise in the hand’s contour led to additional defects. The second issue is that the centroid of the shape is affected by all segmented areas, hand or not. In other words, if the arm is part of the image, it could affect the rate of change of the centroid’s location relative to the amount the hand is actually moving.

**5.4 GESTURE CONTROLS**

A gesture control sign language recognition system is a technology that allows individuals who use sign language to communicate more easily with those who do not understand sign language. The system uses sensors or cameras to detect the movements of the hands and body, and then translates those movements into spoken or written language. The basic gesture controls were implemented using the centroid of the detected hull, and the number of defect points. The program tracks the centroid from the moment a hand is detected, and then shifts the cursor based on the movement of the hand. As the centroid moves, the cursor is moved the same number of pixels. Similarly, for the mouse click, the system tracks and stores the number of defect points, compares it with the next number detected.Gesture control sign language recognition systems have the potential to greatly improve communication between sign language users and non-signers. They can be used in a variety of settings, including education, healthcare, and business, to facilitate communication and improve accessibility for people who use sign.

# CHAPTER 6

# CONCLUSION AND ENHANCEMENT

* 1. **CONCLUSION**

In conclusion, a sign language recognition system can greatly improve communication and accessibility for individuals who are deaf or hard of hearing. We have developed this system by using a convex hull algorithm and several other computer vision techniques, these systems can accurately recognize and translate sign language into text in real-time. Moreover, we have collected large datasets of sign language videos for training the system.The proposed system uses a convolutional neural network for feature extraction from the input video stream. The extracted features are then fed to a recurrent neural network (RNN) for temporal modeling and classification.This convex hull algorithm can create a boundary around the hand and remove any irrelevant background, making it easier for the system to recognize and analyze hand gestures accurately with an accuracy rate of 97.3%. By using this algorithm, sign language recognition systems are more viable than ever before. With continued research and development, these systems can help bridge the communication gap between deaf and hearing communities and promote greater inclusion and accessibility for all.

* 1. **FUTURE ENHANCEMENT**

There are several potential future enhancements for sign language recognition systems, including:

* + 1. Incorporating deep learning models: Deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown promising results in image and video recognition tasks. These models could be used to improve the accuracy of sign language recognition systems.
    2. Using multiple cameras: Sign language recognition systems could be enhanced

by using multiple cameras to capture different angles of the signer's movements. This would allow for more accurate tracking of hand and body movements.

* + 1. Incorporating context: Sign language is often dependent on context, so incorporating information about the surrounding environment or topic of conversation could help improve recognition accuracy.
    2. Developing a larger dataset: The availability of a larger dataset of sign language gestures could help improve the accuracy of sign language recognition systems.
    3. Incorporating feedback mechanisms: Feedback mechanisms, such as using haptic feedback or visual feedback, could help signers correct their movements if the system detects errors.
    4. Real-time translation: Real-time translation of sign language could be an enhancement that allows for greater accessibility for those who are deaf or hard-of-hearing. This would require the use of natural language processing and machine learning algorithms to accurately translate sign language to text.

Overall, the future of sign language recognition systems is promising, and there are many potential enhancements that could improve their accuracy and accessibility.

# APPENDIX I

SAMPLE CODE

import cv2

from cvzone.HandTrackingModule import HandDetector from cvzone.ClassificationModule import Classifier import numpy as np

import math import time

import matplotlib.pyplot as plt import seaborn as sns

from keras.models import Sequential

from keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout ,

BatchNormalization

from keras.preprocessing.image import ImageDataGenerator from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report,confusion\_matrix import pandas as pd

train\_df = pd.read\_csv("sign\_mnist\_train.csv") test\_df = pd.read\_csv("sign\_mnist\_test.csv")

y\_train = train\_df['label'] y\_test = test\_df['label'] del train\_df['label']

del test\_df['label']

from sklearn.preprocessing import LabelBinarizer label\_binarizer = LabelBinarizer()

y\_train = label\_binarizer.fit\_transform(y\_train) y\_test = label\_binarizer.fit\_transform(y\_test) x\_train = train\_df.values

x\_test = test\_df.values x\_train = x\_train / 255 x\_test = x\_test / 255

x\_train = x\_train.reshape(-1,28,28,1) x\_test = x\_test.reshape(-1,28,28,1) cap = cv2.VideoCapture(0)

detector = HandDetector(maxHands=1)

classifier = Classifier("Model/keras\_model.h5","Model/labels.txt") datagen = ImageDataGenerator(

featurewise\_center=False, samplewise\_center=False,

featurewise\_std\_normalization=False, samplewise\_std\_normalization=False, zca\_whitening=False, rotation\_range=10,

zoom\_range = 0.1, width\_shift\_range=0.1, height\_shift\_range=0.1, horizontal\_flip=False, vertical\_flip=False)

datagen.fit(x\_train) model = Sequential()

model.add(Conv2D(75 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu' , input\_shape = (28,28,1)))

model.add(BatchNormalization()) model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(50 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu')) model.add(Dropout(0.2))

model.add(BatchNormalization()) model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same'))

model.add(Conv2D(25 , (3,3) , strides = 1 , padding = 'same' , activation = 'relu'))

model.add(BatchNormalization()) model.add(MaxPool2D((2,2) , strides = 2 , padding = 'same')) model.add(Flatten())

model.add(Dense(units = 512 , activation = 'relu')) model.add(Dropout(0.3))

model.add(Dense(units = 24 , activation = 'softmax'))

model.compile(optimizer = 'adam' , loss = 'categorical\_crossentropy' , metrics = ['accuracy'])

model.summary()

history = model.fit(datagen.flow(x\_train,y\_train, batch\_size = 128) ,epochs = 20 , validation\_data = (x\_test, y\_test))

model.save('smnist.h5') while True:

\_, frame = cap.read()

k = cv2.waitKey(1) if k%256 == 27:

# ESC pressed

print("Escape hit, closing...") break

elif k%256 == 32: # SPACE pressed # SPACE pressed

analysisframe = frame showframe = analysisframe

cv2.imshow("Frame", showframe)

framergbanalysis = cv2.cvtColor(analysisframe, cv2.COLOR\_BGR2RGB) resultanalysis = hands.process(framergbanalysis)

hand\_landmarksanalysis = resultanalysis.multi\_hand\_landmarks if hand\_landmarksanalysis:

for handLMsanalysis in hand\_landmarksanalysis: x\_max = 0

y\_max = 0 x\_min = w y\_min = h

for lmanalysis in handLMsanalysis.landmark:

x, y = int(lmanalysis.x \* w), int(lmanalysis.y \* h) if x > x\_max:

x\_max = x

if x < x\_min: x\_min = x

if y > y\_max: y\_max = y

if y < y\_min: y\_min = y

y\_min -= 20

y\_max += 20

x\_min -= 20

x\_max += 20

analysisframe = cv2.cvtColor(analysisframe, cv2.COLOR\_BGR2GRAY) analysisframe = analysisframe[y\_min:y\_max, x\_min:x\_max] analysisframe = cv2.resize(analysisframe,(28,28))

nlist = []

rows,cols = analysisframe.shape for i in range(rows):

for j in range(cols):

k = analysisframe[i,j] nlist.append(k)

datan = pd.DataFrame(nlist).T colname = []

for val in range(784): colname.append(val)

datan.columns = colname

pixeldata = datan.values pixeldata = pixeldata / 255

pixeldata = pixeldata.reshape(-1,28,28,1) prediction = model.predict(pixeldata) predarray = np.array(prediction[0])

letter\_prediction\_dict = {letterpred[i]: predarray[i] for i in range(len(letterpred))} predarrayordered = sorted(predarray, reverse=True)

high1 = predarrayordered[0] high2 = predarrayordered[1] high3 = predarrayordered[2]

for key,value in letter\_prediction\_dict.items(): if value==high1:

print("Predicted Character 1: ", key) print('Confidence 1: ', 100\*value)

elif value==high2:

print("Predicted Character 2: ", key) print('Confidence 2: ', 100\*value)

elif value==high3:

print("Predicted Character 3: ", key) print('Confidence 3: ', 100\*value)

time.sleep(5)

offset = 20

imgSize = 300

folder = "Data/C" counter = 0

labels=["A","B","C","Hello","No","Okay","Yes","Peace","Rock","Thumbs up","Thumbs down"]

while True:

success, img = cap.read() imgOutput = img.copy()

hands, img = detector.findHands(img) if hands:

hand = hands[0]

x, y, w, h = hand['bbox']

imgWhite = np.ones((imgSize,imgSize,3), np.uint8) \* 255 imgCrop = img[y-offset: y + h+offset, x-offset: x + w+offset] imgCropShape = imgCrop.shape

aspectRatio = h / w

if aspectRatio > 1: k = imgSize / h

wCal = math.ceil(k \* w)

imgResize = cv2.resize(imgCrop, (wCal, imgSize)) imgResizeShape = imgResize.shape

wGap = math.ceil((imgSize - wCal) / 2) imgWhite[:, wGap:wCal + wGap] = imgResize

prediction, index = classifier.getPrediction(imgWhite,draw= False) print(prediction,index)

else:

k = imgSize / w

hCal = math.ceil(k \* h)

imgResize = cv2.resize(imgCrop, (imgSize, hCal)) imgResizeShape = imgResize.shape

hGap = math.ceil((imgSize - hCal) / 2) imgWhite[hGap:hCal + hGap, :] = imgResize

prediction, index = classifier.getPrediction(imgWhite,draw= False) cv2.rectangle(imgOutput, (x - offset, y - offset-50),

(x - offset+90, y-offset-50+50), (255, 0, 255), cv2.FILLED) cv2.putText(imgOutput, labels[index], (x , y - 26) ,

cv2.FONT\_HERSHEY\_COMPLEX , 1.7 ,(255 , 255 , 255), 2)

cv2.rectangle(imgOutput, (x-offset, y-offset),

(x + w+offset, y + h+offset), (255 , 0 ,255) ,4)

cv2.imshow("ImageCrop", imgCrop) cv2.imshow("ImageWhite", imgWhite) cv2.imshow("Image", imgOutput)

cv2.waitKey(1)

cap = cv2.VideoCapture(0)

detector = HandDetector(maxHands=1)

offset = 20

imgSize = 300

folder = "Data/Thumbs down" counter = 0

while True:

success, img = cap.read()

hands, img = detector.findHands(img) if hands:

hand = hands[0]

x, y, w, h = hand['bbox']

imgWhite = np.ones((imgSize,imgSize,3), np.uint8) \* 255 imgCrop = img[y-offset: y + h+offset, x-offset: x + w+offset] imgCropShape = imgCrop.shape

aspectRatio = h / w

if aspectRatio > 1: k = imgSize / h

wCal = math.ceil(k \* w)

imgResize = cv2.resize(imgCrop, (wCal, imgSize)) imgResizeShape = imgResize.shape

wGap = math.ceil((imgSize - wCal) / 2) imgWhite[:, wGap:wCal + wGap] = imgResize

else:

k = imgSize / w

hCal = math.ceil(k \* h)

imgResize = cv2.resize(imgCrop, (imgSize, hCal)) imgResizeShape = imgResize.shape

hGap = math.ceil((imgSize - hCal) / 2) imgWhite[hGap:hCal + hGap, :] = imgResize

cv2.imshow("ImageCrop", imgCrop) cv2.imshow("ImageWhite", imgWhite)

cv2.rectangle(imgOutput, (x - offset, y - offset-50),

(x - offset+90, y-offset-50+50), (255, 0, 255), cv2.FILLED) cv2.putText(imgOutput, labels[index], (x , y - 26) ,

cv2.FONT\_HERSHEY\_COMPLEX , 1.7 ,(255 , 255 , 255), 2)

cv2.rectangle(imgOutput, (x-offset, y-offset),

(x + w+offset, y + h+offset), (255 , 0 ,255) ,4)

cv2.imshow("Image", img) key = cv2.waitKey(1)

cv2.imshow("ImageCrop", imgCrop) cv2.imshow("ImageWhite", imgWhite)

cv2.imshow("Image", imgOutput) cv2.waitKey(1)

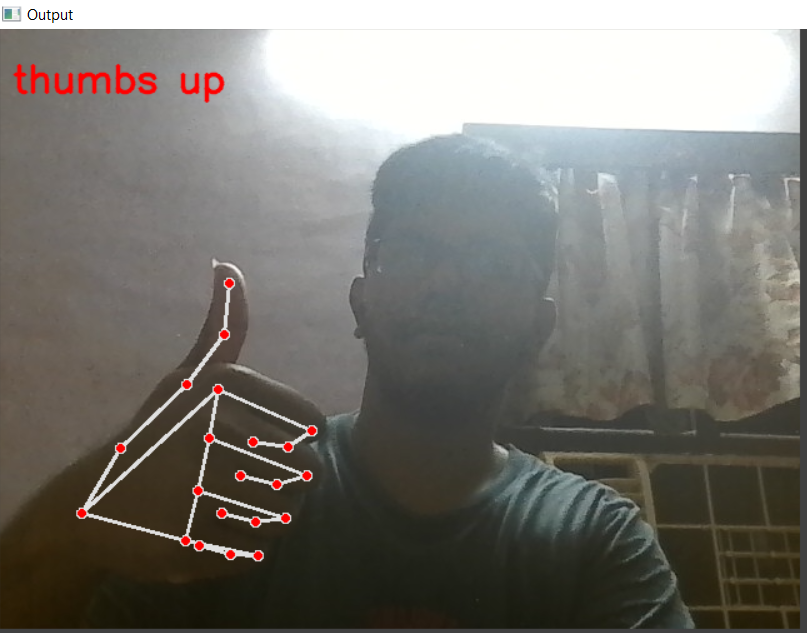
if key == ord("s"): counter += 1

cv2.imwrite(f'{folder}/Image\_{time.time()}.jpg',imgWhite) print(counter)

# APPENDIX-II

SNAPSHOT OF THE OUTPUT:

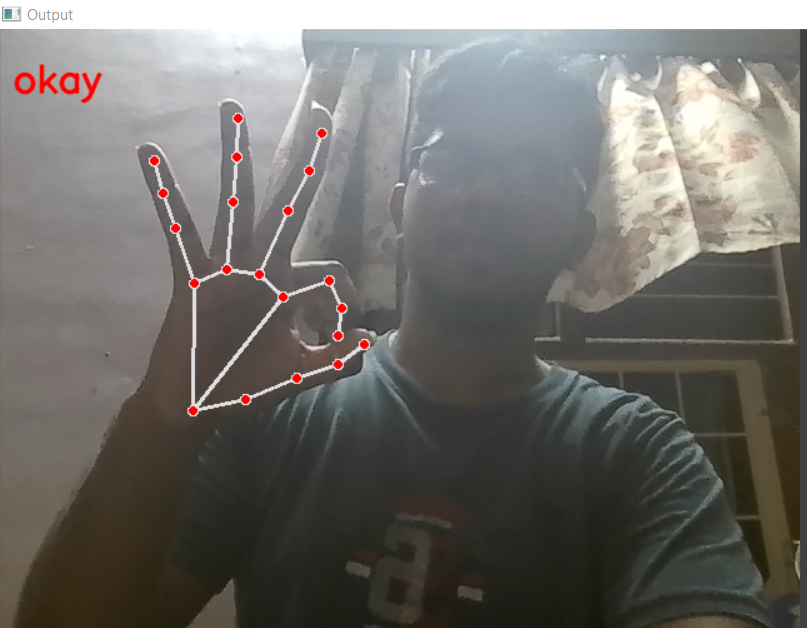
Thumbs up



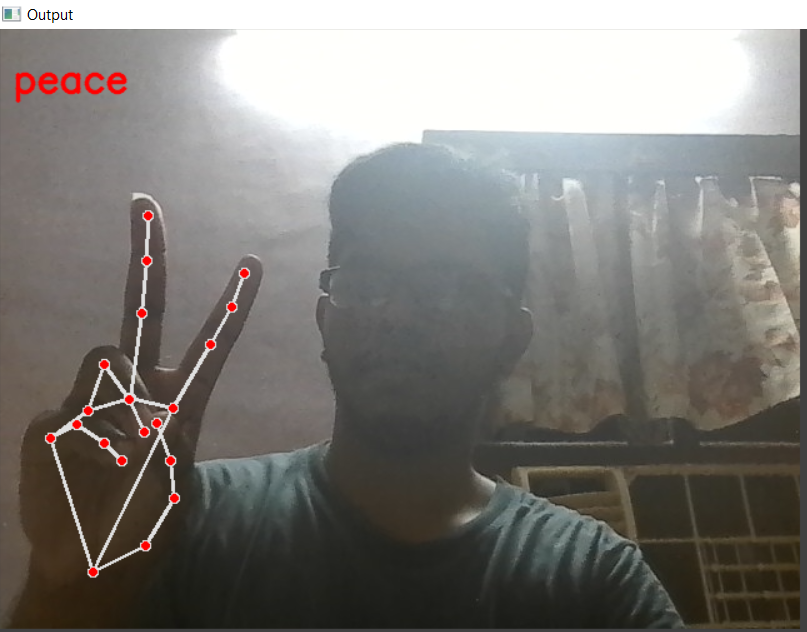
Rock



Okay



Peace



# REFERENCE

1. Bretzner, L., Laptev, I., & Lindeberg, T. (2002). Hand gesture recognition using multi-scale color features, hierarchical models and particle filtering. Paper presented at the Proceedings of fifth IEEE international conference on automatic face gesture recognition.
2. Chen, F.-S., Fu, C.-M., & Huang, C.-L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. Image and vision computing, 21(8), 745-758.
3. Dipietro, L., Sabatini, A. M., & Dario, P. (2008). A Survey of Glove-Based Systems and Their Applications. Ieee transactions on systems, man, and cybernetics, part c (applications and reviews), 38(4), 461- 482.
4. Dong, G., Yan, Y., & Xie, M. (1998). Vision-based hand gesture recognition for human-vehicle interaction. Paper presented at the Proc. of the International conference on Control, Automation and Computer Vision.
5. Garg, P., Aggarwal, N., & Sofat, S. (2009). Vision based hand gesture recognition. World academy of science, engineering and technology, 49(1), 972-977.
6. Gupta, S., Jaafar, J., & Ahmad, W. F. W. (2012). Static hand gesture recognition using a local gabor filter. Procedia Engineering, 41, 827-832.
7. Hasan, H., & Abdul-Kareem, S. (2014). Retracted article: Human–computer interaction using vision based hand gesture recognition systems: A survey. Neural Computing and Applications, 25(2), 251-261.
8. Hasan, M. M., & Mishra, P. K. (2012). Hand gesture modeling and recognition using geometric features: a review. Canadian journal on image processing and computer vision, 3(1), 12-26.
9. O. K. Oyedotun and A. Khashman, Deep learning in vision-based static hand gesture recognition, Neural Comput. Appl., vol. 28, no. 12, pp. 39413951, 2017.
10. Q. Xue, X. Li, D. Wang, and W. Zhang, Deep forest-based monocular visual sign language recognition, Appl. Sci., vol. 9, no. 9, p. 1945, May 2019.
11. S.Jiang,B.Sun,L.Wang,Y.Bai,K.Li,andY.Fu, Skeletonawaremulti modal sign language recognition, 2021, arXiv:2103.08833. [Online]. Available: <https://arxiv.org/abs/2103.08833>
12. A. Sabyrov, M. Mukushev, and V. Kimmelman, Towards real-time sign language interpreting robot: Evaluation of non-manual components on recognition accuracy, in Proc. CVPR Workshops, 2019, pp. 7582.
13. S. Islam, S. S. S. Mousumi, A. S. A. Rabby, S. A. Hossain, and S. Abujar, A potent model to recognize Bangla sign language digits using con volutional neural network, Proc. Comput. Sci., vol. 143, pp. 611618, Jan. 2018.
14. R.-H. Liang and M. Ouhyoung, A sign language recognition system using hidden Markov model and context sensitive search, in Proc. ACM Symp. Virtual Reality Softw. Technol. (VRST), 1996, pp. 5966.
15. B. Saunders, N. C. Camgoz, and R. Bowden, Continuous 3D multi channel sign language production via progressive transformers and mixture density networks, Int. J. Comput. Vis., vol. 2021, pp. 123, May 2021.