Enabling gesture-based application interaction on head mounted VR display

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

RISHI KALIS.K (Reg. No - 39110849)
ROHITH.V (Reg. No – 39110859)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Rohith.V(39110859) and Rishi Kalis.K(39110849)** who carried out the Project Phase-2 entitled **"Enabling gesture based application interaction on head mounted VR display"** under my supervision from January 2023 to April 2023.

Internal Guide

Dr. G. KALAIARASI, M.E., Ph.D.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

DEPARTMENT OF CSE

Submitted for Viva voce Examination held on 20.04.2023

Internal Examiner **External Examiner**

DECLARATION

I,**Rishi Kalis.K(Reg.No-39110849),** hereby declare that the Project Phase-2 Report entitled “**Enabling gesture based application interaction on head mounted VR display”** done by me under the guidance of **Dr. G.Kalaiarasi, M.E.,Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

Rishi kalis-K

DATE:20.04.2023
PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.G.Kalaiarasi M.E.,Ph.D,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

With the integration of smartphones, VR head-mounted displays that are affordable (HMDs) have made VR more accessible to the general public and increased its widespread use. However, due to their lack of interaction, these systems frequently have limitations. GestOnHMD, a technique for a pipeline for classifying gestures and gesture-based interaction, is presented in this project. It makes use of the gesture technique in smartphones to detect hand gestures, detecting 21 points on hand onVR headset. Initially, it carried out gesture-elicitation research using a specific headset the Google Cardboard to detect the 21 hand points which detects the hand gestures and display the meaning of the hand gesture. Based on signal detectability and user choices, we then selected different kinds of gestures to input in the dataset. It uses the neural networks to map the hand points and the cam detects the hand gestures and recognizes it and the displays the meaning of each on the VR device's screen. Each gesture can be made specifically for your command and its meaning byshowing the hand gesture in front of the cam and storing it in the dataset. It can be used to read American Signal Language, etc. which makes people to understand hand gestures and might help in future exploring. The output is displayed on the Google Cardboard Virtual Reality device screen which is connected to the laptop.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full form
AR	Augmented Reality
ASL	American Sign Language
CNN	Convolutional Neural Networks
CLAHE	Contrast Limited Adaptive Histogram Equalization
DNN	Deep Neural Networks
FPV	First Person View
HMD	Head Mounted Display
HMV	Head Mounted Viewer
LED	Light Emitting Diode
LCD	Light Crystal Display
MOMF	Multi Output Multi Fidelity
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
SEQ	Suitability Evaluation Questionnaire
VOC	Visual Object Class
VR	Virtual Reality

CHAPTER 1
INTRODUCTIO
N

1.1 CARD BOARD:

Fig 1.1-Google cardboard

Google created Google Cardboard, a low-cost virtual reality device that works with the majority of smartphones. Google launched it in 2014 to demonstrate the idea behind "Cardboard VR". The Google Cardboard may be used to construct a broad variety of intricate applications even if it is less sophisticated than the Oculus Rift or HTC Vive. With a basic understanding of programming, it is relatively simple to create applications for Google Cardboard. With a user-friendly interface and preloaded libraries, gameengines like Unity3D and Unreal help developers create VR applications. The specifications for the headset as well as the instructions for putting it together are publicly accessible via Google's website making it possible for individuals to construct their very own virtual reality (VR) headset using components that are readily available. A cardboardpiece that has been precisely cut into shape, 45 mm-long lenses, magnets or capacitive three pieces of rubber band, tape, and a hook-and-loop fastening are the fundamental components that make up a cardboard viewer. Virtual reality (VR) technology is rapidly advancing, providing users with more immersive and realistic experiences. However, traditional input methods, such as game controllers and keyboards, can be cumbersome and disrupt the immersion in the virtual world. To overcome this problem, hand gestures have been offered to users as a natural and simple manner for them to engage with virtual items. One of the ways Virtual objects can be interacted with by users in the Google Cardboard VR device is through the use

of hand gestures. This study will examine the application of hand gestures to interact with virtual objects in Google Cardboard VR device. Hand gesture recognition technology uses cameras or other sensors to capture and interpret the hand of the user movements, and translate them into actions in the virtual world. This technology has been used in various applications, including gaming, virtual reality, and human-computer interaction.

Hand gesture recognition technology has been around for several decades, with the earliest applications dating back to the 1970s. However, it was not until the advent of more powerful computers and advances in computer vision that this technology became more practical for widespread use. Currently, there are several methods used to capture and interpret hand gestures, including computer vision, Leap Motion, and virtual reality headsets with built-in cameras. Computer vision uses cameras to capture images of the user's hands and uses algorithms to interpret the hand movements. Leap Motion uses infrared cameras and LED lights to track the user's hand movements. Virtual reality headsets with built-in cameras use the headset's cameras to track the user's hand movements using hand gestures as a means of communication in VR has severalbenefits. First, hand gestures are a simple and easy technique to communicate for users to interact with virtual objects. Second, hand gestures can provide a more immersive experience by allowing users to physically interact with the virtual environment. Third, hand gestures can reduce the need for traditional input devices, such as game controllers, which can disrupt the immersion in the virtual world. In this research paper, we'll investigate on how to detect hand gestures using hand gesture algorithms and display the message on the Virtual Reality screen using Trinus Virtual Reality software. We will talk about the most recent developments in devices for recognizing hand gestures, and its applications in Virtual Reality. We will also analyze the benefits and drawbacks of using hand gestures as a method of interaction in Virtual reality and identifyareas for future research.

Cardboard features:

- **Cross-platform support** for both Android and iOS devices.
- **Affordable viewers** that support one-button input and gaze-based UI.

- **Powerful SDKs** that make it easy to create VR experiences in the development environment of your choice

1.2 SUPPORTED DEVICES:

- Cardboard supports Android versions 4.1 or higher.
- Cardboard supports iOS versions 8.0 or higher.

Not all viewers work with every phone. If you see a device incompatible message,you may not have a device with a gyroscope included. To see whether your phonewill work with a specific Cardboard viewer.

1.3 CARDBOARD VIEWERS:

Cardboard viewers make it easy to insert and secure your phone for VR experiences. Cardboard viewers are:

- Affordable
- Lightweight and portable
- Easy-to-use even for first time users with just a single button input
- Immersive, supporting a wide field-of-view

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE SURVEY FOR EXISTING SYSTEM

1) Title: Exploring the Past with Google Cardboard
   Author: Adeola Fabola, Dr Alan Miller, Professor Richard Fawcett

The combination of mobile technologies and digital reconstruction has the potential to enhance visitors' experiences at historic sites. It is possible to use technology that is already in people's pockets to create an immersive on-site exploration of historic sites by designing a mobile app with Google Cardboard.

The paper examined their experience developing a mobile app that serves as a digital guided tour of St. Andrews Cathedral's remains. A tour of one of Scotland's most significant historical sites is provided by the app, which combines traditional media like audio, images, panoramas, 3D video, and 4 Steradian (or 360°) video witha mobile smartphone and Google Cardboard.

The mobile application can be downloaded from iTunes or Google Play, allowing for immediate distribution to a potential audience of millions.

It works in conjunction with the location-aware medieval St Andrews App, which offers a self-guided tour of the entire city of St. Andrews. Even though Google Cardboard isn't available, the app still serves a useful purpose by providing audio commentary and visual content about this historic structure.

2) Title: A Low-cost, Low-latency Approach to Dynamic Immersion in Occlusive Head-Mounted Displays
   Author: Robert W. Lindeman

In this poster, Researchers discuss the idea of controlling how immersive head-mounted displays (HMDs) are by using LCD panels. The paper proposed using LCD panels similar to those found in common active-stereo glasses used in movie theatres to replace the cowling around typical HMDs of the "ski-goggle" variety.

To adjust how much of the real world is visible to the user's peripherals, the panels can be controlled with very basic standalone circuitry or a microcontroller. Because it allows users to see objects in their immediate surroundings (such as the keyboard and mouse), can be used to combat cybersickness by providing natural cues when needed, and introduces no additional latency into the system, this can significantly improve the usability of consumer HMD’s. Using Google Cardboard, we demonstrate several prototypes of our strategy and offer some initial thoughts regarding the systems.

3) Title: Getting Around in Google Cardboard – Exploring Navigation Preferences with Low-Cost mobile VR

Author: Wendy Powell, Vaughan Powell, Phillip Brown, Marc Cook, Jahangir Uddin

The adoption and application of virtual reality (VR) have undergone a paradigm shift in recent years. The development of low-cost VR headsets and advancements in graphics rendering have made VR accessible to everyday consumers. The Google Cardboard VR viewers, which are compatible with the majority of smart phones and only cost a few dollars, allow mobile VR to truly enter the realm of the everyday. However, these headsets are currently mostly used for passive entertainment or 360-degree media viewing, not for active virtual space exploration. Our preliminary evaluation of three travel and navigation strategies is presented in this paper.

4) Title: Genome Gazing: A 360° Stereoscopic Animation for Google Cardboard Author: Kate Patterson

Scientific concepts related to genomics and epigenetics can be abstract and are mostly invisible to the human eye. Communication of these concepts to a lay audience is particularly challenging. Visualizing DNA and associated molecules via an embodied experience offers the user an opportunity to interact with this scientific data in a more meaningful way. The project explored new ways to visualize and represent the major molecular players in genomics and epigenetics using virtual reality via portable head mounted displays.

5) Title: Robust Hand Gestural Interaction for Smartphone based AR/VR Applications Author: Shreyash Mohatta, Ramakrishna Perla, Gaurav Gupta, Ehtesham Hassan,
and Ramya Hebbalaguppe

Hand gestures will dominate user interfaces in the future. For smartphones with limited computational power, this paper explores an intuitive hand gesture-based interaction in this paper. In order to accomplish this researcher, present an effective First Person View (FPV) gesture recognition algorithm that focuses on recognizing a four-swipe model (Left, Right, Up, and Down) for smartphones using a single monocular camera vision. By providing a touch-free interface and real-time performance, this can be utilized in the construction of AR/VR-based automation systems for large-scale deployments with low-cost AR/VR devices like Google Cardboard and Wearality. They effectively address the FPV constraints presented by a wearable by taking into account multiple cues like palm color, hand contour segmentation, and motion tracking. Under the same constraints, we also provide comparisons of swipe detection to existing methods. They demonstrate that our method is superior in both computational time and accuracy for gesture recognition.

6) Title: Personalized Augmented Reality Experiences in Museums using Google Cardboards
Author: Marinos Theodorakopoulos, Nikos Papageorgopoulos, Andriana Mourti,Angeliki Antoniou, Manolis Wallace, George Lepouras, Costas Vassilakis,Nikos Platis

In the paper, they investigate whether the Google Cardboard can be used to deliver individualized cultural experiences. In particular, they created the application and the content necessary to provide highly individualized visits to the Archaeological Museum in Tripolis, Greece.
They also look at problems with the usability of using Google Cardboards. They also outline the next steps based on the promising early results.

7) Title: Hand Gesture based Region Marking for Tele-support using Wearables Author: Archie Gupta, Shreyash Mohatta, Jitender Maurya, Ramakrishna Perla,
Ramya Hebbalaguppe, Ehtesham Hassan

In many applications, wearable augmented reality (AR) devices1 are being investigated for the purpose of visualizing contextual information in real time. Importantly, these devices can also be used for tele-assistance from remote locations when on-field operators need help with troubleshooting from outside experts. Touchless hand gestures are the most natural way to select a Region of Interest (ROI) through a wearable, such as defective machine parts, for effective communication. To localize the ROI in First Person View (FPV), a hand-gestured interaction technique is presented in this paper. The selected region is highlighted to the remote server setup for expert advice using freehand sketching gestures. The proposed method is novel because: (a) it uses a touchless finger-based gesture recognition algorithm that runs on smartphones and can be used with wearable frugal modality like Google Cardboard/Wearality; and (b) it uses an on-board implementation of the recognition module to reduce network latency and achieve real-time performance. They conducted user studies that demonstrate the proposed method's ease of use and usefulness. Furthermore, they utilized the PASCAL Visual Object Classes (VOC) criteria to assess the ROI gesture's efficacy.

8) Title: A Personalized Stereoscopic 3D Gallery with Virtual Reality Technology on Smartphone

Author: Huy Tran, Minh Nguyen, Huy Le, Wei Qi Yan

Due to the recent development of smartphones and low-cost VR headsets, virtual reality (VR) is gaining popularity at an ever-increasing rate. In this piece, an online system that makes stereoscopic 3D galleries for individual use is what we suggest. In addition to receiving stereoscopic images, the system lets users upload two images of an object of interest (such as a personal collectible artifact) taken with their smartphone. The images must be close to a stereo pair (i.e.) before taking the second shot, the camera is slightly shifted to the right. After several corresponding points have been identified, the uploaded images are automatically rectified into an epipolar stereo pair (photos will be horizontally aligned). In order to produce the VR-friendly

stereoscopic images that are viewable in VR, a straightforward stereo matching algorithm is used to determine the average disparity range between the two images. The system then creates a gallery of the collected photos for VR devices like Google Cardboard, Samsung Gear VR, and Google Daydream to display 3D visualization. This system is portable, inexpensive, and simple to set up and use. Users all over theworld could easily visualize and share their collectible items in 3D with this system; which are thought to be beneficial to the VR and social media communities

9) Title: VRNav: A Framework for Navigation and Object Interaction in Virtual Reality Author: Sampath Shanmugam, Vaishak R Vellore, Sarasvathi V

The field of virtual reality (VR) is currently a hot topic of industry and academic research and experimentation. Today, anyone with a smartphone can experience the magic of virtual reality thanks to the introduction of VR headsets like the Google Cardboard, HTC Vive, and Samsung Gear VR. A user must be able to both navigate and interact with objects in the virtual world in order to have a good VR experience. However, neither of these objectives is simple to attain. When external controllers and sensors are used, immersion and freedom are frequently diminished.

Keeping two important points in mind, the primary objective of the paper is to develop a framework that makes it possible for users to navigate and interact with virtual world objects without having to use a controller or any other similar touch input-based mechanism used by existing implementations.

First and foremost, the setup needs to be constructed in such a way that these objectives can be accomplished with only a smartphone and a VR head-mounted display and no additional hardware. Second, the setup needs to be kept as affordable as possible. As a result, anyone with a smartphone and a low-cost Google Cardboard headset should be able to play around with virtual reality and use our framework in any virtual world they create.

10) Title: REAL TIME HAND SEGMENTATION ON FRUGAL HEADMOUNTED DEVICE FOR GESTURAL INTERFACE

Author: Jitender Maurya, Ramya Hebbalaguppe , Puneet Gupta

With the resurgence of HMDs, air gestures have emerged as a natural and intuitive method of interaction and communication. HMD's, such as Daqri smart glasses and Microsoft HoloLens, have additional sensors and on-board processors, increasing the device's cost. As a result, our objective is to make Augmented/Virtual reality accessible to a wider audience by widening the area of mobile device interaction: with just low-cost head mounts like Wearality and Google Cardboard and a smartphone. In an egocentric perspective, reliable hand segmentation precedes gesture recognition, which is one necessary step for a feasible human-computer interaction. A real-time method for segmenting hands using a standard mobile device's RGB camera is what they propose. The novelty of our work lies in the development of a filtering method known as the Multi Orientation Matched filter, which can be used for hand segmentation and operates on the device, even when the background is skin-like. On public datasets, they have extensively tested our hand segmentation method. Under the same constraints, they offer hand segmentation comparisons to existing techniques. They demonstrate that our approach outperforms competitors in terms of accuracy and computational time. In addition, they demonstrate that Android smartphones can use our solution to classify zoom gestures in real time.

11) Title: A Framework for Cardboard based Augmented Reality Author: Syed Rameez Rehman, Shahzad Rasool

In the paper, they propose a generic framework for immersive augmented reality experience based on a low-cost head mounted viewer (HMV) such as Google Cardboard. The proposed frame work provides a functional implementation structure thatcan act as the baseline for a large variety of applications. It consists of a camera for capturing the real-world image, a module for frame sorting, which provides a frame-by-frame output for processing and the provision for input from different on board as well as external sensors. A computer vision module processes the incoming video to generate meaningful information that is used together with the input from different sensors by the main processing module to take decisions regarding position and pose of virtual objects with respect to the real scene. The module for placement and positioning of 3D virtual objects performs transformations on the virtual objects such that they are consistently overlaid on the real-world image and rendered on the display screen. This module is responsible for ensuring that the final image rendered is consistent across different head mounted viewers. Furthermore, differentmodalities of interaction are incorporated including direct voice input and using an external controller. Communication with external sensors and other neighboring systems is also supported. As a proof-of-concept a head mounted viewer application has been implemented that provides navigation for drivers in NUST H-12 Campus, Islamabad. A user can select their destination and the application guides the user to their destination through the head mounted viewer display. Using the on-board sensors, the user is routed via the shortest vehicular path by displaying virtual directional arrows superimposed on the real view.

12) Title: Camera-Based Selection with Cardboard
HMDs Author: Siqi Luo, Robert J.Teather

Utilizing the outward-facing camera on modern smartphones, they examine selection strategies for low-cost mobile VR devices like Google Cardboard. Air touch, head ray, and finger ray were the three selection techniques that they compared. According to the initial evaluation, hand-based selection (air touch) was the worst. The selection performance of a ray cast with the tracked finger position was significantly higher. Based on our findings, it appears that ray-based methods can be used for camera-based mobile tracking.

13) Title: The Cardboard VR Game Development
Tool Author: Min-Bin Chen

The head-mounted device's technological advancement allows the user to experience virtual reality. The immersive experience, when combined with the game's gameplay, can offer the player a novel experience. Google Cardboard is inexpensive and simple to use, making it easier to play virtual reality games. Due to the device's

functional limitations, cardboard VR games make game design more difficult, so there are few related games.

However, playing such games is simpler. A successful game design will have a high market value. When students create games that are difficult to design, they can concentrate on game design to create better ones if the pressure to program is reduced. This study creates development tools for Cardboard games to help game designers improve the quality of their creations.

14) Title: Virtual look around: interaction quality evaluation for virtual tour in multiple platforms

Author: Adriel Rodrigues, Jean Felipe Patikowski Cheiran

Virtual Reality (VR) has become widely accessible in recent years through avariety of devices, including desktop and head-mounted displays (HMDs). For instance, 360-degree panoramic photos and videos on social media made virtual tourexperiences popular, allowing people to visit a remote location without having to deal with crowded transportation or places. Also, in the current pandemic, a virtual tour has a lot of potential as a way to escape and even teach remotely. However, they do not have any studies that compare the interaction quality of these virtual tours at various levels of immersion. The purpose of this paper is to compare the interaction quality of a WebXR- developed virtual tour website across a variety of devices. This kind of research is important because it makes it easier for users to find a better balance between costs and benefits. They used WebXR and React 360 to create a 360-degree virtual tour and conducted a user study with 41 undergraduate students on four distinct devices to accomplish our goal: a smartphone, a Google Cardboard headset, a laptop computer, and a Samsung GearVR HMD. By modifying the Suitability Evaluation Questionnaire (SEQ), they assessed users' attitudes and performance by determining how long it took them to complete a set of tasks.

The key findings of our study include the following: (i) the overall self-reported experience of using Google Cardboard is worse than that of using other devices; (ii) the performance of users is fairly comparable between the platforms; and (iii) tests with the smartphone show evidence of unexpected cybersickness symptoms.

15) Title: GestOnHMD: Enabling Gesture-based Interaction on Low-cost VR Head- Mounted Display

Author: Taizhou Chen, Lantian Xu, Xianshan Xu and Kening Zhu

With the integration of smartphones, low-cost virtual reality (VR) head-mounted displays (HMDs) have made VR more accessible to the general public and increased its widespread use. However, due to their lack of interaction, these systems frequently have limitations. GestOnHMD, a technique for gesture-based interaction and a pipeline for gesture classification, is presented in the paper. It makes use of stereo microphones found in common smartphones to detect tapping and scratching gestures on the front, left, and right surfaces of a mobile VR headset. They first conducted a gesture- elicitation study using the Google Cardboard as our focused headset to generate 150 user-defined gestures, 50 for each surface. Based on user preferences and signal detectability, we then selected 15, 9, and 9 gestures for the front, left, and right surfaces, respectively. The deep-learning classification pipeline was trained to detect and recognize these on-surface gestures by constructing a dataset with the acoustic signals of 18 users, Last but not least, in conjunction with thelive demonstration of GestOnHMD, they carried out a series of online participatory- design sessions to collect a set of user-defined gesture-referent mappings that could potentially benefit from GestOnHMD.

16) Title: Sign language translator and gesture recognition

Author: Elmahgiubi, Mohammed, Mohamed Ennajar, Nabil Drawil, and Mohamed Samir Elbuni

A crucial system for the deaf and mute is presented in the paper: an automatic translator of hand signs. Here, the system's anticipated requirements and degree of performance are outlined. The article lists system components and elaborates on how hardware wire connections and component assembly work. The software component of this system is also very sophisticated, with basic setup and recognition algorithms included. The study discusses the difficulties in detecting vague measures and suggests corresponding technical solutions. The outcomes of the trial make it clear that the system has the ability to assist specific people and communities. The issue with the paper is that especially given that the majority of the letters could be recognized by the algorithm is (20 out of 26). Furthermore, the software part of this system is extensively elaborated to include simple system initialization and recognition algorithms. The paper addresses the challenges of identifying ambiguous measurements and proposes respective technical solutions.

17) Title: Sign language recognition using image-based hand gesture recognition techniques.

Author: Nikam, Ashish S., and Aarti G. Ambekar

Only for counting fingers and recognizing numbers, convexity hull algorithms can be used. With the use of the contour analysis procedure, a study akin to this can be carried out on the Marathi and English alphabets. By comparing hand templates and taking contour curve shapes into consideration, recognition can be accomplished. Itis simple to identify hand gestures in contour analysis, by taking into consideration vector values, regardless of scaling and shape. Even by creating an application that converts sign language to voice output, a barrier to communication between people with speech impairments can be lifted. The software's goal is to display a real-time system for recognizing hand gestures based on the detection of a few shape-based criteria, such as orientation, the centroid of the Centre of Mass, the status of the fingers, and the placement of the thumb in relation to raised or folded fingers. By bearing in mind how the human hand, which has one thumb and four fingers, is shaped, this is accomplished. The issue with the paper is that the algorithm used can be replaced with contour analysis process for better results.

18) Title: Hand gesture recognition system for translating indian sign language into text and speech.

Author: Kunjumon, Jinsu, and Rajesh Kannan Megalingam

In order to help deaf individuals and give voice to the voiceless, the paper's main goal is to break down communication barriers between speech-impaired people and able-bodied people. This goal is accomplished by employing the system that is suggested. The gloves convert sign gestures into text in both English and Malayalam and are reasonably priced. The suggested solution offers increased precision, is portable, and uses relatively little power. Flex sensor, microcontroller, and Bluetooth module HC 05 are used to convert Sign language, which is then displayed on an Android phone. The issue with the paper is that the phrases to be recognized is very less and accuracy should be increased. The project can be further developed to converts more words phrases. The dynamic movements can be detected and sensed more accurately. The work can be extended to converts sign language to text and speech.

19) Title: Cost effective portable system for sign language gesture recognition. Author: Khambaty, Yousuf, Rolando Quintana, Mehdi Shadaram, Sana Nehal,

Muneeb Ali Virk, Waqar Ahmed, and Ghayas Ahmedani

The technology described in the paper transforms movements that are static into voice. It also goes by the name GesTALK. The self-designed and reasonably priced Data Glove serves as the main input device. It has two primary modes of operation. While speaking a whole string by concatenating words in the second mode, the system says a single alphabet in the first mode in response to a static gesture performed by the user. The words in a string are spelled out using a variety of alphabet movements. Both the American Sign Language (ASL) and the Pakistan Sign Language (PSL), as well as other language models, have been proved to function with this system. The issue with the paper is that cochlear implants are not a viable option for all HI people and is a frustratingmode of communication. Hiring experienced interpreters is expensive and challenging, people cannot rely on them in daily life.

20) Title: Real-time sign language framework based on wearable device: analysis of MSL, DataGlove, and gesture recognition.

Author: Ahmed, Mohamed Aktham, B. B. Zaidan, A. A. Zaidan, Abdullah Hussein Alamoodi, Osamah Shihab Albahri, Z. T. Al-Qaysi, Ahmed Shihab Albahri, and Mahmood M. Salih.

The purpose of the study was to break down the difference in communication between hearing-impaired individuals and the general population. As a result of the way the sub-modules are built and the particulars of each module, a framework for MSL recognition is therefore described. The framework under discussion is divided into three main sections: the first section, titled "Analysis of MSL Modules," deals withthe extraction of helpful features that can be used to determine system needs and accurately detect complex indications. The development of an ideal DataGlove based on the findings of system experiments and sensor testing is the focus of the second module (DataGlove Modules). In the third module, DataGlove's data collection is preprocessed and categorized (Gesture Recognition Module). This conversation about development might help close the communication gap between hearing- impaired people and others. The issue with the paper is that it recognizes only simplehand gestures and accuracy needs to be improved.

2.2 INFERENCES FROM LITREATURE SURVEY

1) The combination of a smartphone and a lightweight virtual reality headset like the Google Cardboard opens up new possibilities for on-site exploration of heritage sites. It also offers an improved user experience, by tackling some of the challenges that have been uncovered in previous works.

2) A continuous motion system with the ability to control the speed and stop seems to be the best compromise for immersive mobile VR.

3) An Augmented Reality tele-presence framework for highlighting the region of interest from a wearable has been presented. We have demonstrated a touch-less gesture recognition algorithm on a smartphone with Google Cardboard in a dynamic background setting. Real-time performance is achieved by implementing gesture recognition module on-board.

4) The underlying reason for developing this feature is to enable the frugal headsets for AR applications like industrial inspection and tourism. The proposed hand segmentation method performs correctly on-device in real-time without using additional hardware such as depth and IR sensors. It is designed using CLAHE, Gaussian blur with our proposed MOMF.

5) VR enhances the level of media experience and practices human pursuit of realism. Cardboard head-mounted displays can perform well in the use and promotion of medical, educational or gaming applications. Since it can only sense theplayer's head movements by the mobile phone, its fixed-point play and eye-control interaction make it difficult to design a good game.

6) We developed a virtual tour in UNIPAMPA using WebXR and React 360 technologies, and carried out a case study with 41 participants. Four distinct platforms (a laptop computer with a mouse, a smartphone, a Google Cardboard viewer and a Samsung Gear VR HMD) were alternately used by all participants while performing tasks with different levels of complexity. We conclude that the adapted SEQ scores point out significantly worse interaction quality on the Google Cardboard viewers.

7) We propose GestOnHMD, a deep-learning-based gesture recognition framework to support gestural interaction for mobile VR. We trained a three-step framework of deep neural networks for gesture detection and recognition. The on-PC experiments showed an overall accuracy of 98.2% for both gestures detection and surface recognition, and 97.7% for gesture classification. In addition, our online participatory design studies showed that we provide a sufficient number of design options for a wide range of mobile VR applications.

8) The issue in the project is that especially given that the majority of the letters could be recognized by the algorithm is (20 out of 26). Furthermore, the software part of this system is extensively elaborated to include simple system initialization andrecognition algorithms. The paper addresses the challenges of identifying ambiguous measurements and proposes respective technical solutions.

9) Problem with the problem is that cochlear implants are not a viable option for all HI people and is a frustrating mode of communication. Hiring experienced interpreters is expensive and challenging, people cannot rely on them in daily life.

10) In the project the phrases to be recognized is very less and accuracy should be increased.

11) By bearing in mind how the human hand, which has one thumb and four fingers, is shaped, this is accomplished. The issue with the paper is that the algorithm used can be replaced with contour analysis process for better results.

12) The outcomes of the trial make it clear that the system has the ability to assist specific people and communities. The issue with the paper is that especially given that the majority of the letters could be recognized by the algorithm is (20 out of 26).

13) The issue in the project is one continuous motion with the option to change the speed and stop would appear to be the best compromise because none of the three strategies examined in this study provided an optimum answer for travel withinmobile VR.

14) By delivering natural cues when necessary, it can also be utilized to lessen cybersickness, this can significantly improve the usability of consumer HMD’s. The issue with the paper is that the design is extremely difficult, both financially and technically.

15) The limitations of this paper are that the users may not be able to utilize the system when they are offline and may suffer slow loading times when they are online,which could have a detrimental effect on their user experience.

CHAPTER 3
DESCRIPTION OF PROPOSED SYSTEM

Google Cardboard was envisioned as the cheapest Virtual Reality (VR) solution on the planet, and at this point, nothing else comes close in terms of pricing. However, the low price did not bring about mass adoption, and Google's Android-based VR platform is nothing more than a tech curiosity at this point. In this post, Toptal Technical Editor Nermin Hajdarbegovic leverages his extensive experience in the graphics industry to explain what's keeping Cardboard VR down, and what the platform needs to attract more users, investment, and development.

Imagine if you will, standing on the surface of the moon, overlooking a crater from your lunar rover, listening to mission control chatter. Or don't. Instead of imagining it, just order a cheap Google Cardboard VR set instead, stick your phone in it, and start exploring the solar system, museums, tourist spots, coral reefs and much more. Let the Imagination Technologies GPU in your phone live up to its name and do your imagining for you.

Google Cardboard is hardly a new concept. It was unleashed on the unsuspecting geekosphere at Google I/O 2014, roughly 18 months ago. Since then, Google has tweaked the Cardboard reference design, but the concept hasn't changed; Google Cardboard was envisioned as the cheapest Virtual Reality (VR) solution on theplanet, and at this point, nothing else comes close in terms of pricing.

3.1 SELECTED METHODOLOGY

3.1.1 Display Device

The device that is used in this work is an Android device manufactured by Redmi known as Note 9 pro max. This device runs on an Android 'Snow Cone' (version 12.0.0) mobile operating system. It is capable of running Google VR applications and is compatible with Google Cardboard. The phone has to be configured to

‘Debug Mode’ before installing any new applications to it. Debug mode enables developers to install new applications into the phone and debug any errors if necessary.

Fig-3.1 Note 9 Pro Max

Display	6.67-inch 1080x2400 AMOLED
Processor	Qualcomm Snapdragon 720G Processor
Memory and Storage	RAM: 8GB Internal storage: 128GB
Dimensions	159.3 x 77.8 x 7.3 mm
Weight	178g
Battery	3450 mAh

Table-3.1 Note 9 Pro Max Specifications

3.1.2 Google Cardboard Viewer

The Google Cardboard viewer is designed to work with Android or iOS phones with screen sizes from 4 to 6 inches. It has an interactive button that works with all compatible phones.

In order to create a cardboard viewer, only 6 parts are required:

1. Cardboard
2. Lenses
3. Magnets
4. Two strips of Velcro
5. Rubber band
6. NFC tag (optional)

These parts are easily accessible almost anywhere in the world and can be easily purchased to create a cardboard viewer with simple easy-to-follow instructions from the Google website.

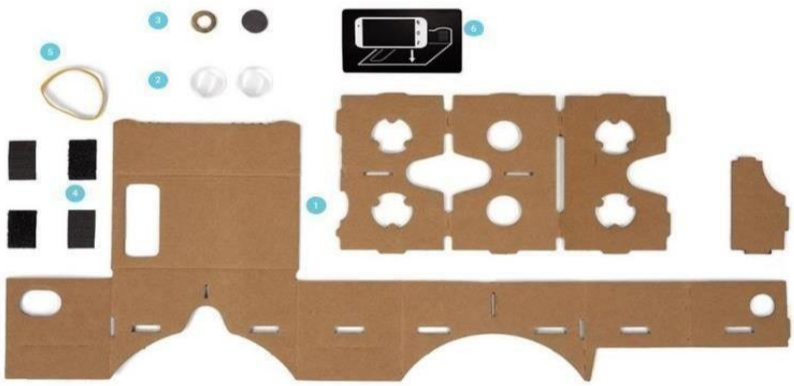

Fig-3.2 Google Cardboard Viewer Part

Lens Diameter	34 mm
Field Of View (FOV)	80 degrees
Magnets	0.75 inches diameter 0.12 inches thickness
Dimensions	5.9 x 3.5 x 2.2 inches
Weight	96g

Table-3.2 Google Cardboard Specifications

3.2 OVERALL DESIGN OF PROPOSED SYSTEM

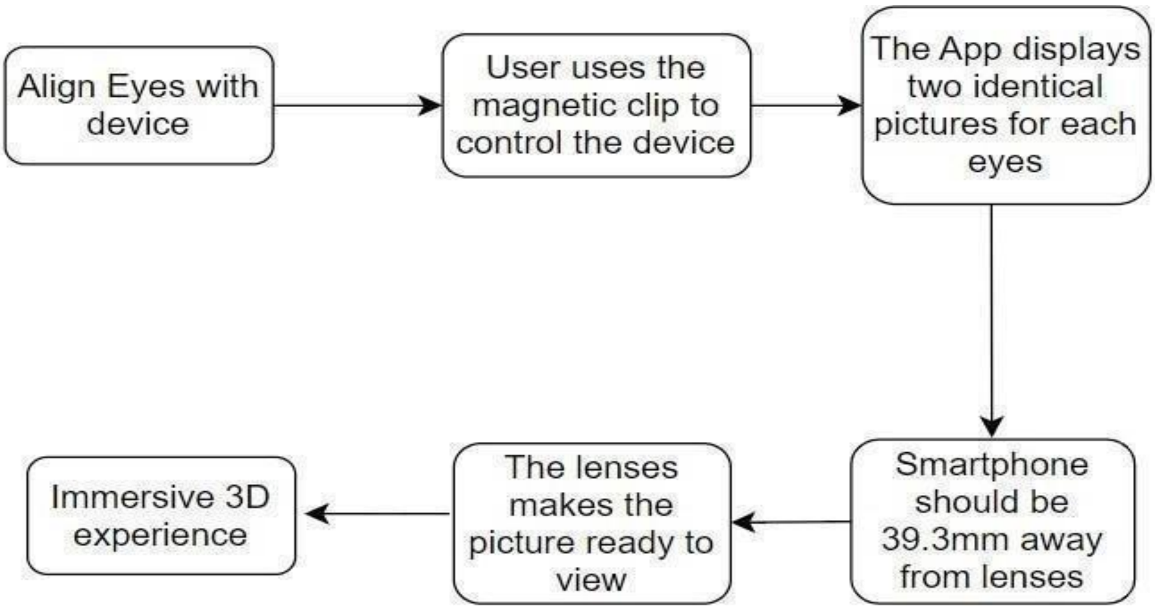

Fig-3.3 Flow diagram for Google cardboard

3.2.1 GOOGLE CARDBOARD'S PHYSIOLOGY

From a mental, experiential perspective, a number of factors determine whether or not Google Cardboard will be used effectively. Primarily, the user needs to be able tosee clearly and process images as well as the ideas in them. The algorithm focuses primarily on the user's personal preference for what images to view and how to view them if those factors are present and fully functional. The algorithm cannot officially include some innovations that are being developed to assist Google Cardboard in simulating the Doppler Effect through spatial audio effects. However, these modifications are not yet available. The steps required to assemble the Google Cardboard are included in the mechanical or app-oriented algorithm, so this section of the algorithm begins after those steps.

The following steps make up Google Cardboard's physiological algorithm: The concave slots in the center of the cardboard device must first be aligned with the users' faces and eyes. The device cannot be used in a conventional manner or at least in the manner that it was intended to be used if the user does not have the mental and physical/visual capacity to see, process, and comprehend the images in the manner that GoogleCardboard intends them to be perceived. The lenses are used to present images from a manipulated, three- dimensional perspective if the user has the mental and physical/visual capacity to see, process, and comprehend the images. The user then controls the flow, duration, and order of the images by using the magnetic installmentson the side of Google Cardboard. The algorithm continues indefinitely if the viewer wishes to continue viewing the images or video; The algorithm comes to a close at this point if the viewer decides not to keep watching the videos or images.

3.2.2 TECHNOLOGICAL PERSPECTIVE

From this point of view, the app's concept behind the scenes is the input component of this process. After the part, the concepts need to be visualized. Apps for Google Cardboard's images are formatted slightly differently than those for other, more standard photo viewing apps. In order to create a three-dimensional impression in a person's mind, the main difference appears as a splint between two similar images. The left and right eyes of humans perceive the two images differently. In order for users to focus on the specific points on the mobile device that is being used in Google Cardboard, designers must create optics for the images. Finally, the images ought to be adjusted in such a way that they can track and respond to a user's head movements. This gives designers the ability to create the illusion of depth and perspective, making the images appear endless and sensationalized.

3.2.3 MECHANICAL PROCESSES

The following steps make up the mechanical algorithm of Google Cardboard (GC). The user first mounts their smartphone on the GC dock. If the dock has an NFC chip that launches the GC app on the smartphone automatically, Velcro add-ons can be used to secure the device. If there is no NFC chip, the user will have to manually launch the GC app on their smartphone before locking it in place. The user then presses the GC against their forehead with the lenses and groove facing them. To move through images, the user finally manipulates the side-mounted magnet.
The user unlocks their smartphone from the dock when the process is finished and, if necessary, closes the GC app.

3.3 HAND GESTURE RECOGNITION

Hand gesture recognition is the process of identifying and classifying the movements and positions of the hand or fingers. It is a subfield of computer vision and machine learning that has gained significant attention in recent years due to its potential applications in a wide range of fields, including robotics, sign language recognition, virtual reality, and human-computer interaction.

Hand gesture recognition systems typically involve three main stages: data acquisition, feature extraction, and classification. In the data acquisition stage, hand movements and positions are captured using cameras, sensors, or other devices. In the feature extraction stage, various features are extracted from the acquired data, such as hand shape, finger position, and hand movement trajectory. Finally, in the classification stage, a machine learning algorithm is used to classify the hand gestures based on the extracted features.

One of the main challenges in hand gesture recognition is the high degree ofvariability in hand movements and positions. The same gesture can be performed in different ways by different individuals, and the same hand movement can have different meanings depending on the context. Therefore, hand gesture recognition systems need to be able to handle this variability and be robust to changes in lighting, background, and hand position.

There are several techniques and algorithms used for hand gesture recognition, including rule-based methods, template matching, and machine learning-based methods. Rule-based methods use a set of predefined rules to identify hand gestures based on specific criteria, such as the position of the fingers or the shape ofthe hand. Template matching methods involve comparing the acquired hand image with a set of pre-defined templates to identify the closest match. Machine learning-based methods involve training a machine learning algorithm using a dataset of handgesture images and their corresponding labels, and then using the trained model to classify new hand gesture images.

There are several software libraries and frameworks available for hand gesture recognition, including Mediapipe, OpenCV, TensorFlow, Python, and NumPy. These libraries provide tools for various tasks related to hand gesture recognition, such as data acquisition, image processing, feature extraction, machine learning, and real-time processing. They can be used individually or in combination to build custom hand gesture recognition systems.

Hand gesture recognition has numerous potential applications, such as sign language recognition for the deaf, virtual reality interactions, and human-robot collaboration. As the field continues to evolve, it is likely that new techniques and algorithms will be developed to improve the accuracy and robustness of hand gesture recognition systems, opening up even more possibilities for its use.

3.4 REQUIREMENTS FOR HAND GESTURE RECOGNITION

Requirements for this project include:

1. Python - 3.88
2. OpenCV - 4.5

To install OpenCV, use "pip install opencv-python".

3. MediaPipe - 0.8.5

To install MediaPipe, use "pip install mediapipe".

4. TensorFlow version 2.5

Run "pip install TensorFlow" to set up the TensorFlow module.

5. Numpy – 1.19.3

3.4.1 PYTHON

Python is a popular programming language that is widely used in machine learning and computer vision, including hand gesture recognition. Here are some of the ways that Python can be used in hand gesture recognition:

Pre-processing and data manipulation: Python provides a large number of libraries and packages for scientific computing, data analysis, and machine learning. These libraries, such as NumPy, Pandas, and Scikit-learn, can be used for pre-processing and manipulating image data, such as resizing, cropping, and normalization.

Feature extraction: Python provides a variety of feature extraction algorithms that can be used to extract features from hand gesture images. For example, the HOG (Histogram of Oriented Gradients) descriptor is a popular feature extraction algorithm that can be used to extract local texture and shape features from images.

Machine learning algorithms: Python provides a variety of machine learning libraries, such as TensorFlow, Keras, and PyTorch, that can be used to build and trainmachine learning models for hand gesture recognition. These models can be trained on labeled datasets of hand gesture images, and can then be used to recognize new hand gestures in real-time.

Real-time processing: Python is well-suited for real-time processing applications, such as hand gesture recognition, due to its high-level syntax and fast performance. Python can be used to implement real-time gesture recognition systems that can be deployed on embedded devices, such as Raspberry Pi or other microcontrollers.

Visualization: Python provides a variety of data visualization libraries, such as Matplotlib and Seaborn, that can be used to visualize hand gesture data, such as histograms, scatterplots, and heatmaps. These visualizations can be used to gain insights into the performance of hand gesture recognition models, and to identify areas for improvement.

Python provides a powerful set of tools for hand gesture recognition, and is widely used in research and industry applications. Its large community and vast number of available libraries make it a flexible and customizable choice for hand gesture recognition projects.

3.4.2 MEDIAPIPE

Mediapipe is an open-source framework developed by Google that provides a comprehensive set of building blocks for building real-time multimedia pipelines. One of the pre-built components in Mediapipe is the Hand Pose Estimation module, which can be used for hand gesture recognition.

The Hand Pose Estimation module in Mediapipe uses a machine learning model to detect the landmarks (key points) of the hand in an image or video. The module takes an input image or video and outputs a set of 21 3D landmark points representing the position of each finger joint and the palm.

Once the landmarks are detected, they can be used to recognize hand gestures. One common approach is to use a machine learning classifier such as a support vector machine (SVM) or a neural network to classify the landmarks into different hand gestures. The classifier is trained on a dataset of hand gestures and corresponding landmark coordinates, and it learns to map the input landmarks to the corresponding

gesture label.
Another approach is to use the relative positions of the landmarks to describe the hand pose and recognize gestures based on the pose. This approach does not require training a separate classifier and can be more robust to variations in lighting and hand shape.

Mediapipe provides a high-level API for building hand gesture recognition applications using the Hand Pose Estimation module. The API abstracts away the details of landmark detection and provides a simple interface for developers to use. Developers can customize the application by specifying the gesture recognition algorithm, the training dataset, and the output format.

3.4.3 TENSORFLOW

TensorFlow is an open-source machine learning library developed by Google, which provides a comprehensive set of tools and APIs for building and deploying machine learning models. TensorFlow is widely used in hand gesture recognition because of its ability to train complex neural network models and run them on a variety of hardware platforms.

TensorFlow can be used in hand gesture recognition in several ways:

Training custom models: With TensorFlow, developers can create custom neural network models for hand gesture recognition. These models can be trained on a dataset of hand gestures and corresponding labels using supervised learning algorithms such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs). TensorFlow provides a high-level API for building these models, as well as low-level APIs for more fine-grained control over the model architecture.

Transfer learning: Transfer learning is a technique where a pre-trained neural network model is used as a starting point for training a new model on a different task. With TensorFlow, developers can use pre-trained models such as Inception, ResNet, or MobileNet as feature extractors for hand gesture recognition. The pre-trained models are trained on large image classification tasks, such as ImageNet, and can be fine-tuned on a smaller dataset of hand gesture images for the recognition task.

Real-time inference: TensorFlow can be used to run trained models on embedded devices, such as smartphones or microcontrollers, for real-time hand gesture recognition. TensorFlow provides optimized libraries, such as TensorFlow Lite, for running models on resource-constrained devices. This allows developers to create hand gesture recognition applications that run entirely on the device without requiring a cloud or server-side processing.

TensorFlow is widely used in research and industry for hand gesture recognition applications. Some examples include recognizing sign language gestures, controlling robots with hand gestures, and controlling virtual reality environments. TensorFlow's flexibility and scalability make it a popular choice for developing hand gesture recognition applications of all types and complexities.

3.4.4 OPENCV

OpenCV (Open Source Computer Vision) is a popular computer vision library that is often used in hand gesture recognition applications. Here are some of the ways that OpenCV can be used for hand gesture recognition:

Image processing: OpenCV provides a variety of image processing functions, such as filtering, thresholding, and edge detection, which can be used to pre-process images of hands before passing them to a hand gesture recognition algorithm. For example, smoothing can be used to remove noise from the image, and thresholding can be used to segment the hand from the background.

Contour detection: OpenCV provides functions for contour detection, which can be used to identify the boundary of the hand in the image. Once the contour is detected, it can be used to extract features from the hand, such as the centroid, area, and orientation.

Feature extraction: OpenCV provides functions for feature extraction, such as the HOG (Histogram of Oriented Gradients) descriptor, which can be used to extract features from the hand. These features can then be used as input to a machine learning algorithm for hand gesture recognition.

Gesture recognition: OpenCV can be used to implement machine learning algorithms for hand gesture recognition, such as support vector machines (SVMs) or convolutional neural networks (CNNs). These algorithms can be trained on a dataset of labeled hand gesture images, and then used to recognize new hand gestures in real-time.

Real-time processing: OpenCV is optimized for real-time image processing, which makes it well-suited for hand gesture recognition applications that require low-latency processing. OpenCV can be used to implement real-time gesture recognition systems that can be deployed on embedded devices, such as smartphones or wearable devices.

OpenCV provides a powerful set of tools for hand gesture recognition, and is widely used in research and industry applications.

3.4.5 NUMPY

NumPy is a popular Python library for numerical computing that is often used in hand gesture recognition. Here are some of the ways that NumPy can be used in hand gesture recognition:

Image processing: NumPy provides a variety of functions for image processing, such as flipping, rotating, and cropping, which can be used to pre-process images of hands before passing them to a hand gesture recognition algorithm. For example, flipping can be used to augment the dataset of hand gesture images, and cropping can be used to remove irrelevant background.

Feature extraction: NumPy provides functions for mathematical operations, such as dot product, which can be used to extract features from the hand, such as the orientation and curvature of the fingers. These features can then be used as input to a machine learning algorithm for hand gesture recognition.

Data manipulation: NumPy provides functions for manipulating multidimensional arrays, such as reshaping and transposing, which can be used to preprocess the hand gesture images and to prepare them for machine learning algorithms.

Machine learning algorithms: NumPy is often used in conjunction with machinelearning libraries such as Scikit-learn, TensorFlow, and PyTorch to build and train machine learning models for hand gesture recognition. NumPy arrays can be used tostore the input data, labels, and weights for the machine learning models.

Real-time processing: NumPy is well-suited for real-time image processing due to its fast array processing speed. NumPy can be used to implement real-time gesture recognition systems that can be deployed on embedded devices, such as Raspberry Pi or other microcontrollers.

NumPy provides a powerful set of tools for hand gesture recognition, and is widely used in research and industry applications. Its speed and efficiency make it a popular choice for real-time image processing and machine learning applications.

3.5 CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional neural networks (CNNs) are a popular type of deep learning model used in hand gesture recognition. CNNs are especially useful for this task because they can learn to recognize features and patterns in images without requiring explicit feature engineering.

CNNs are typically composed of multiple convolutional layers, followed by pooling layers and fully connected layers.

In hand gesture recognition, the input to the CNN is usually an image or video frame of a hand, and the output is a probability distribution over a set of gesture classes.

Here are the steps involved in using CNNs for hand gesture recognition:
Dataset preparation: A dataset of hand gesture images or video frames must be collected and labeled with the corresponding gesture class labels.

Data preprocessing: The images or frames in the dataset must be preprocessed to normalize the pixel values and reduce noise. This may involve resizing, cropping, or augmenting the images to increase the size of the dataset.

Model architecture: A CNN architecture must be designed that is appropriate for the hand gesture recognition task. The architecture may consist of multiple convolutional layers with different kernel sizes and pooling layers to reduce the dimensionality of the output. The final layer is typically a fully connected layer with softmax activation, which outputs a probability distribution over the gesture classes.

Model training: The CNN model is trained on the dataset of hand gesture images or frames using backpropagation and gradient descent. The objective is to minimize the loss function, which measures the difference between the predicted and true labels. During training, the weights of the CNN are updated to improve its performance on the task.

Model evaluation: Once the CNN model is trained, it is evaluated on a separate test set to measure its accuracy and performance. The evaluation may involve computing metrics such as precision, recall, and F1 score, as well as visualizing the confusion matrix to identify the most common errors.

CNNs have been used successfully in a variety of hand gesture recognition tasks, including recognizing sign language gestures, controlling virtual and augmentedreality environments, and controlling robots with hand gestures. With appropriate dataset preparation, data preprocessing, and model architecture design, CNNs can achieve high accuracy and robustness in hand gesture recognition applications.

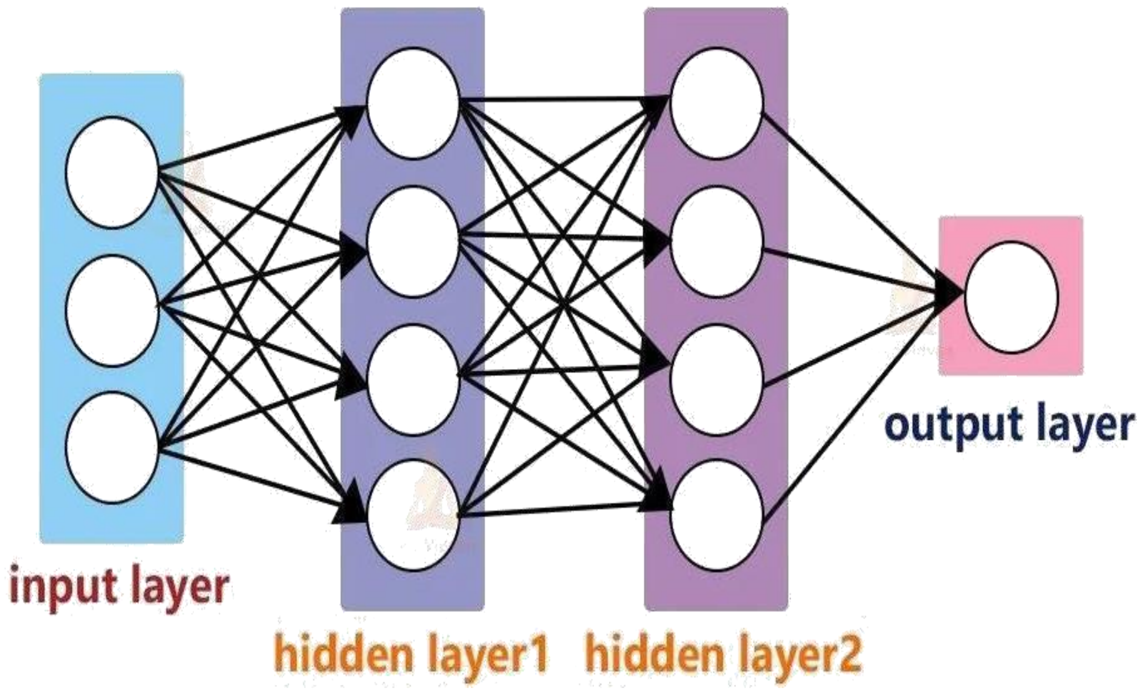

Fig.3.4 Neural Networks with two hidden layers

3.5.1 THE HAND POSE ESTIMATION MODULE

The Hand Pose Estimation module in Mediapipe is a pre-built module that allows developers to detect and track the positions of the hand landmarks in real-time. This module is part of the larger Mediapipe framework, which provides developers with a set of pre-built components for building computer vision and machine learning applications.

The Hand Pose Estimation module uses a deep neural network (DNN) to estimate the positions of 21 key hand landmarks, including the wrist, thumb, and fingers. The DNN is trained on a large dataset of hand images and uses a combination of convolutional and fully connected layers to extract features and classify hand landmarks.

To use the Hand Pose Estimation module in Mediapipe, developers can simply import the pre-built module and pass in the hand image or video stream. The module will then return the estimated 3D positions of the 21 hand landmarks in real-time, which can be used for various applications such as hand gesture recognition, virtual reality interactions, and robotics.

The Hand Pose Estimation module also provides developers with several options for customization and optimization. For example, developers can adjust the input and output resolution, as well as the neural network architecture and training parameters, to optimize performance and accuracy for their specific use case. Additionally, developers can use other Mediapipe modules in combination with the Hand Pose Estimation module, such as the Object Detection module or the Face Detection module, to build more complex computer vision applications.

The Hand Pose Estimation module in Mediapipe provides developers with a pre-built, easy-to-use tool for detecting and tracking the positions of hand landmarks in real-time. This module can be used for a wide range of applications, and its customizable options allow developers to optimize performance and accuracy for their specific use case.

3.6 TRINUS VR SOFTWARE

Trinus VR is a software application that allows users to use their mobile device as a virtual reality (VR) headset by connecting it to a computer. With Trinus VR, users can play their favorite PC games in virtual reality, watch movies and videos in 3D, and experience VR content without having to purchase a dedicated VR headset.

Trinus VR works by using the mobile device as a VR display and the computer as a VR processor. The Trinus VR software consists of two parts: the Trinus VR Server, which runs on a PC or laptop, and the Trinus VR Client, which runs on the mobile device. The Trinus VR Server uses the graphics card and CPU of the computer to render the VR content and then streams it to the Trinus VR Client on the mobile device.

To use Trinus VR, users need to download and install the Trinus VR Server on their PC or laptop and the Trinus VR Client on their mobile device. They then need to connect the Trinus VR Client to the Trinus VR Server via Wi-Fi or USB tethering. Once connected, the Trinus VR software will automatically configure the settings of the VR headset to match the specifications of the mobile device.

Trinus VR supports a variety of VR headsets, including Google Cardboard, Daydream, and Samsung Gear VR. It also supports a variety of VR controllers, including the PlayStation Move and the Wii Remote, which can be used to interact with VR content on the PC.

Overall, Trinus VR provides an affordable and accessible way for users to experience VR content on their mobile devices without having to invest in a dedicated VR headset.

3.7 System Architecture

Steps to follow:

- Import necessary packages.
- Initialize models.
- Read frames from a webcam.
- Detect hand key points.
- Recognize hand gestures

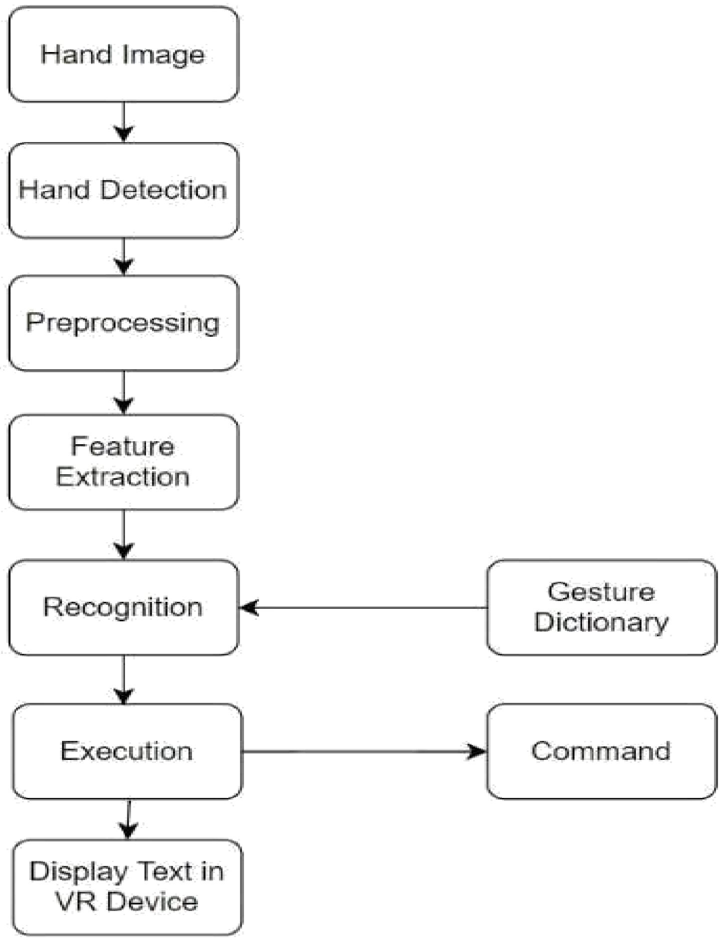

Fig.3.5 Gesture Recognition for google cardboard

Image Acquisition: The first step in hand gesture recognition is to acquire an image or video of the hand gesture. This can be done using various devices, such as webcams, RGB-D cameras, or even smartphone cameras. The quality of the image is critical, as it can significantly affect the accuracy of the recognition system. Thus, the camera should be positioned at an appropriate angle and distance from the hand to capture the gesture accurately.

Hand detection: In this stage, the hand region in the acquired image is detected and extracted using computer vision techniques. This can be done using algorithms such as skin color segmentation, background subtraction, or machine learning-based object detection.

Preprocessing: The acquired image is pre-processed to enhance its quality and remove any noise or unwanted background. This involves operations such as noise filtering, thresholding, and segmentation. Noise filtering removes any random fluctuations in the image caused by lighting conditions or camera movement. Thresholding converts the grayscale image into a binary image, making it easier to identify the hand region. Segmentation isolates the hand region from the background by using techniques such as skin color detection or background subtraction.

Feature Extraction: Once the image is pre-processed, features are extracted from it. These features are typically hand-crafted or learned from the data using machine learning techniques. Hand-crafted features can include geometric features like the position, orientation, and size of the hand, while learned features can be deep features extracted from convolutional neural networks (CNNs). Other features commonly used in hand gesture recognition include shape descriptors, texture features, and motion features.

Recognition: In this stage, a machine learning algorithm is used to classify the hand gesture based on the extracted features. This can be done using techniques such as decision trees, support vector machines, or neural networks.

Gesture dictionary: A dictionary or lookup table is used to map the recognizedgesture to a specific action or command. This can include tasks such as controlling arobotic arm, playing a video game, or navigating a user interface.

Execution: Finally, the recognized gesture is executed, which involves performing the corresponding action or command based on the gesture dictionary. This can be done using software or hardware components such as actuators, motors, or displays.

Classification: The final step is to classify the hand gesture based on the extracted features. This is typically done using machine learning algorithms such as support vector machines (SVMs), decision trees, or neural networks. The classification algorithm is trained on a dataset of hand gestures and corresponding labels, and it learns to map the input features to the corresponding gesture label. During inference,

the input image is passed through the classification algorithm, and the predicted gesture label is outputted.

Trinus VR is a software application that allows users to use their mobile device as a virtual reality (VR) headset by connecting it to a computer. The Trinus VR software consists of two parts: the Trinus VR Server, which runs on a PC or laptop, and the Trinus VR Client, which runs on the mobile device. By using Trinus VR, users can experience VR content on their mobile device without having to purchase a dedicated VR headset.

To display a message in Google Cardboard VR using Trinus VR, you can follow these steps:

Install and configure the Trinus VR Server on your PC or laptop. You can download the Trinus VR Server from the official website and install it on your computer. Once installed, you will need to configure the server settings, such as the resolution and frame rate, to match the settings of your mobile device.

Install and configure the Trinus VR Client on your mobile device. You can download the Trinus VR Client from the Google Play Store or the App Store and install it on your mobile device. Once installed, you will need to connect the Trinus VR Client to the Trinus VR Server on your PC or laptop.

Launch the Google Cardboard VR app on your mobile device. You can download the Google Cardboard VR app from the Google Play Store or the App Store and install it on your mobile device. Once installed, you can launch the app and place your mobile device inside a Google Cardboard VR headset.

Display the message in Google Cardboard VR. To display the message in Google Cardboard VR, you can use the Trinus VR software to mirror the screen of your PC or laptop onto your mobile device. This will allow you to display the message on your PC or laptop and view it in Google Cardboard VR.

To build this Hand Gesture Recognition, we’ll need four packages. So, first import these.

- Mediapipe
- Numpy
- OpenCV
- Tensorflow

Initialize

MediaPipe:

- Mp.solution.hands module performs the hand recognition algorithm. So, we create the object and store it in mpHands.
- Using mpHands.Hands method we configured the model. The first argument is max_num_hands, that means the maximum number of hand will be detected by the model in a single frame. MediaPipe can detect multiple hands in a single frame, but we’ll detect only one hand at a time in this project.
- Mp.solutions.drawing_utils will draw the detected key points for us so that we don’t have to draw them manually.

Initialize Tensorflow:

- Using the load_model function we load the TensorFlow pre-trained model.
- Gesture.names file contains the name of the gesture classes. So first we open the file using python’s inbuilt open function and then read the file.
- After that, we read the file using the read () function.

Read frames from a webcam:

- We create a VideoCapture object and pass an argument ‘0’. It is the camera ID of the system. In this case, we have 1 webcam connected with the system. If you have multiple webcams then change the argument according to your camera ID. Otherwise, leave it default.
- The cap.read() function reads each frame from the webcam.
- cv2.flip() function flips the frame.
- cv2.imshow() shows frame on a new openCV window.
- The cv2.waitKey() function keeps the window open until the key ‘q’ is pressed.

Detect hand keypoints:

- MediaPipe works with RGB images but OpenCV reads images in BGR format. So, using cv2.cvtCOLOR() function we convert the frame to RGB format.
- The process function takes an RGB frame and returns a result class.
- Then we check if any hand is detected or not, using result.multi_hand_landmarks method.
- After that, we loop through each detection and store the coordinate on a list called landmarks.
- Here image height (y) and image width(x) are multiplied with the result because the model returns a normalized result. This means each value in the result is between 0 and 1.
- And finally using mpDraw.draw_landmarks() function we draw all the landmarks in the frame.

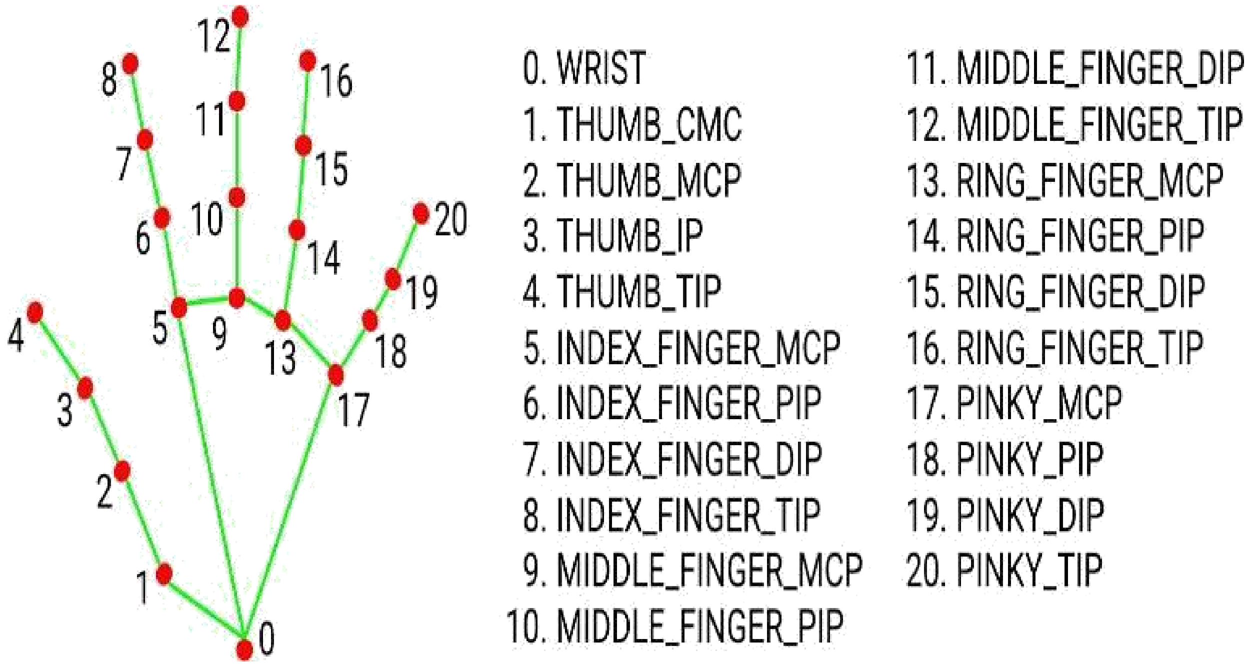

Fig.3.6 Key points to recognize hand gestures

Recognize hand gestures:

- The model.predict() function takes a list of landmarks and returns an array contains 10 prediction classes for each landmark.

  The output looks like this-

  [[2.0691623e-18 1.9585415e-27 9.9990010e-01

  9.7559416e-05 1.6617223e-06 1.0814080e-18 1.1070732e-27

  4.4744065e-16

  6.6466129e-07 4.9615162e-21]]
- Np.argmax() returns the index of the maximum value in the list.
- After getting the index we can simply take the class name from the classNames list.
- Then using the cv2.putText function we show the detected gesture into the frame.

Displaying the hand gesture recognition output on Google Cardboard VR screen using Trinus VR is an interesting idea that can enhance the user experience of hand gesture recognition. By using Trinus VR, users can see the hand gesture recognition output in a more immersive way and interact with it in a natural and intuitive manner.

To implement this idea, the hand gesture recognition system would need to be integrated with the Trinus VR software, which would require some programming and configuration. The Trinus VR software would need to be set up to display the hand gesture recognition output on the VR screen, which could be done using the Trinus VR Server settings.

Once the system is set up, users can wear the Google Cardboard VR headset and see the hand gesture recognition output on the VR screen. They can then interact with the output by making the corresponding hand gestures, which would be detected by the hand gesture recognition system and displayed on the VR screen in real-time.

Overall, displaying the hand gesture recognition output on Google Cardboard VR screen using Trinus VR can provide a more immersive and engaging user experience for hand gesture recognition.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Result

The Google cardboard VR device which was made is used as VR device but has limitations to its use by the users. In order to increase the features and functions performed by the users, we have added hand gesture recognition feature or method that can capture and interpret users hand movements and translate it into the meaning for each hand gesture and display it on the Virtual Reality screen of the Google Cardboard Virtual Reality device and also it is designed to make interaction between people and the virtual things in the virtual world. The user will be able to engage with virtual things in a natural way thanks to its design and also help users with recognition of hand movements. Using Python and OpenCV, we created a hand gesture recognizer. For gesture identification and detection, we used the MediaPipe and Tensorflow frameworks, respectively. The hand gesture recognition is used to help people who don't know sign language and cannot understand it and this willhelp them in understanding sign language and can help converse with others who are dumb and deaf as well. Understanding sign language is just one example here. Hand gesture recognition can be used in many ways for many uses. The result of ourproject is that it recognizes the hand gesture and displays the message on thescreen of the Google Cardboard Virtual Reality device and the user can view the message in Virtual Reality experience.

4.2 Discussion

The addition of hand gesture recognition to the Google Cardboard VR device offers users an enhanced experience by providing the ability to interact with virtual objects in a more natural way. By capturing and interpreting the user's hand movements and translating them into specific hand gestures, the device can display the corresponding action on the virtual reality screen. The use of Python and OpenCV to create the hand gesture recognizer and MediaPipe and TensorFlow for gesture identification and detection respectively, makes the technology accessible to a wide range of developers and users.

One significant application of this technology is its potential to assist people who are deaf or hard of hearing, as it can help them understand and communicate through sign language. Additionally, it can be used to improve accessibility for people with physical disabilities, such as those who may have difficulty using traditional input devices. Furthermore, it can enhance the virtual reality experience for gaming and entertainment, allowing users to interact with objects in the virtual world in a more intuitive way.

However, there are also some limitations to the technology. The accuracy and effectiveness of the hand gesture recognition may vary depending on the lighting conditions and other external factors. Additionally, some users may find the device difficult to use or experience discomfort while wearing it for extended periods of time.

Overall, the addition of hand gesture recognition to the Google Cardboard VR device provides a promising opportunity to improve user experience and accessibility. With further development and refinement, it has the potential to revolutionize the way people interact with virtual reality and with each other.

CHAPTER 4
CONCLUSION AND FUTURE WORK

4.1 Conclusion

The proposed system in this research paper combines several of a webcam and hand gesture recognition software to capture and interpret hand gesture of the user, and translate them into meaning of the hand gesture in the virtual world. The system was created to offer consumers a simple and natural method to interact with virtual objects in Google Cardboard VR device. The proposed system offers several benefits such as providing a more immersive experience, reducing the need for traditional input devices, and being a low-cost solution. However, there are also some drawbacks such as the technology being complex and difficult to set up and calibrate and being affected by the user's physical limitations. The proposed system is still in its early phases, and additional study is required to increase the precision and dependability of the hand gesture recognition technology and to make it more accessible to users with physical limitations. Overall, the proposed system represents a promising solution for natural and intuitive interaction with virtual objects in virtual reality environments and will be beneficial for a variety of purposes such as for understanding sign languages like ASL, BSL, etc... As the technology continues to evolve, it is expected to become more accurate and reliable and can be used in many places for many different uses providing a more immersive and realistic experience for users. The output is displayed in the Google Cradboard VR device screen which is connected to the laptop using Trinus VR software that will make the users see the hand gesture recognition output in a more immersive way and interact with it in a natural and intuitive manner.

4.2 Future Work

- Work on integrating hand gesture recognition completely with the Trinus VR.
- Work on increasing the dataset to a very large amount of hand images which will help in increasing the hand recognition and increase accuracy.

REFERENCES

1) Fabola, Adeola, Alan Miller, and Richard Fawcett. "Exploring the past with Google Cardboard." In *2015 Digital Heritage*, vol. 1, pp. 277-284. IEEE, 2015.

2) Lindeman, Robert W. "A low-cost, low-latency approach to dynamic immersion in occlusive head-mounted displays." In *2016 IEEE Virtual Reality (VR)*, pp. 221-222. IEEE, 2016.

3) Powell, Wendy, Vaughan Powell, Phillip Brown, Marc Cook, and Jahangir Uddin. "Getting around in google cardboard–exploring navigation preferences with low-cost mobile VR." In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, pp. 5-8. IEEE, 2016.

4) Patterson, Kate. "Genome gazing: A 360° stereoscopic animation for Google cardboard." In *2017 IEEE Virtual Reality (VR)*, pp. 439-439. IEEE Computer Society, 2017.

5) Mohatta, Shreyash, Ramakrishna Perla, Gaurav Gupta, Ehtesham Hassan, and Ramya Hebbalaguppe. "Robust hand gestural interaction for smartphone based AR/VR applications." In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 330-335. IEEE, 2017.

6) Theodorakopoulos, Marinos, Nikos Papageorgopoulos, Andriana Mourti, Angeliki Antoniou, Manolis Wallace, George Lepouras, Costas Vassilakis, and Nikos Platis. "Personalized augmented reality experiences in museums using Google Cardboards." In *2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pp. 95-100. IEEE, 2017.

7) Tran, Huy, Minh Nguyen, Huy Le, and Wei Qi Yan. "A personalised stereoscopic 3D gallery with virtual reality technology on smartphone." In *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1-6. IEEE, 2017.

8) Shanmugam, Sampath, Vaishak R. Vellore, and V. Sarasvathi. "VRNav: A Framework for Navigation and Object Interaction in Virtual Reality." In *2017 2nd International Conference on Computational Systems and Information Technology* *for Sustainable Solution (CSITSS)*, pp. 1-5. IEEE, 2017.

10) Syed, Rameez Rehman, and Shahzad Rasool. "A Framework for Cardboard Based Augmented Reality." In *2018 International Conference on Frontiers of Information Technology (FIT)*, pp. 287-292. IEEE, 2018.

11) Chen, Min-Bin. "The cardboard VR game development tool." In *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp. 438-441. IEEE, 2019.

12) Rodrigues, Adriel, and Jean Felipe Patikowski Cheiran. "Virtual look around: Interaction quality evaluation for virtual tour in multiple platforms." In *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pp. 47-56. IEEE, 2020.

13) Chen, Taizhou, Lantian Xu, Xianshan Xu, and Kening Zhu. "Gestonhmd: Enabling gesture-based interaction on low-cost vr head-mounted display." *IEEE Transactions on Visualization and Computer Graphics* 27, no. 5 (2021): 2597-26

14) Elmahgiubi, Mohammed, Mohamed Ennajar, Nabil Drawil, and MohamedSamir Elbuni. "Sign language translator and gesture recognition." In 2015 Global Summit on Computer & Information Technology (GSCIT), pp. 1-6. IEEE, 2015

15) Kunjumon, Jinsu, and Rajesh Kannan Megalingam. "Hand gesture recognition system for translating indian sign language into text and speech." In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 14-18. IEEE, 2019.

16) Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image-based hand gesture recognition techniques." In 2016 online international conference on green engineering and technologies (IC-GET), pp. 1-5. IEEE, 2016

17) Khambaty, Yousuf, Rolando Quintana, Mehdi Shadaram, Sana Nehal, Muneeb Ali Virk, Waqar Ahmed, and Ghayas Ahmedani. "Cost effective portable system for sign language gesture recognition." In 2008 IEEE International Conference on System of Systems Engineering, pp. 1-6. IEEE, 2008.

18) Ahmed, Mohamed Aktham, B. B. Zaidan, A. A. Zaidan, Abdullah Hussein Alamoodi, Osamah Shihab Albahri, Z. T. Al-Qaysi, Ahmed Shihab Albahri, and

Mahmood M. Salih. "Real-time sign language framework based on wearable device: analysis of MSL, DataGlove, and gesture recognition." *Soft Computing* 25,no. 16 (2021): 11101-11122.

APPENDIX

A) SOURCE CODE

```
# import necessary

packages import

cv2 import numpy as

np import mediapipe

as mp import

tensorflow as tf

from tensorflow.keras.models import load_model


# initialize mediapipe

mpHands = mp.solutions.hands

hands = mpHands.Hands(max_num_hands=1,

min_detection_confidence=0.7) mpDraw = mp.solutions.drawing_utils


# Load the gesture recognizer model

model = load_model('mp_hand_gesture')


# Load class names

f = open('gesture.names', 'r')

classNames =

f.read().split('\n')


f.close()

print(classNames

)
```

```
# Initialize the webcam cap

=

cv2.VideoCapture(0)

while True:

    # Read each frame from the

    webcam _, frame = cap.read()


    x, y, c = frame.shape

    # Flip the frame vertically

    frame = cv2.flip(frame, 1)

    framergb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)


    # Get hand landmark prediction

    result = hands.process(framergb)


    # print(result)

    className = ''

    # post process the result

    if

        result.multi_hand_landmark

        s: landmarks = []

        for handslms in

            result.multi_hand_landmarks: for lm in

            handslms.landmark:

                # print(id, lm) lmx
```

= int(lm.x * x)

```
            lmy = int(lm.y * y)

            landmarks.append([lmx,
 lmy]) # Drawing landmarks on frames

         mpDraw.draw_landmarks(frame,handslms,
mpHands.HAND_CONNECTIONS)


         # Predict gesture

         prediction =

         model.predict([landmarks]) #

         print(prediction)

         classID = np.argmax(prediction)

         className = classNames[classID]


   # show the prediction on the frame

   cv2.putText(frame, className, (10, 50),

               cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2, cv2.LINE_AA)


   # Show the final output

   cv2.imshow("Output", frame)


   if cv2.waitKey(1) ==

      ord('q'): break


# release the webcam and destroy all active

windows cap.release()

cv2.destroyAllWindows()
```

B) OUTPUT SCREENSHOTS

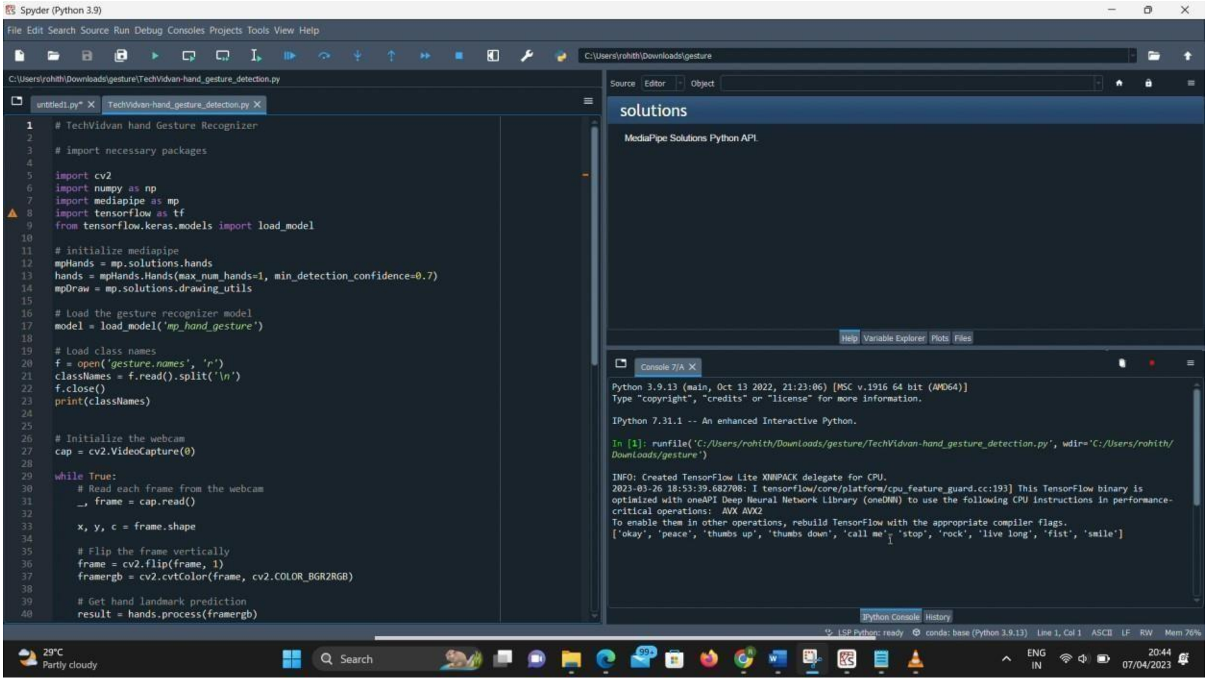

Fig.a. Coding Screen

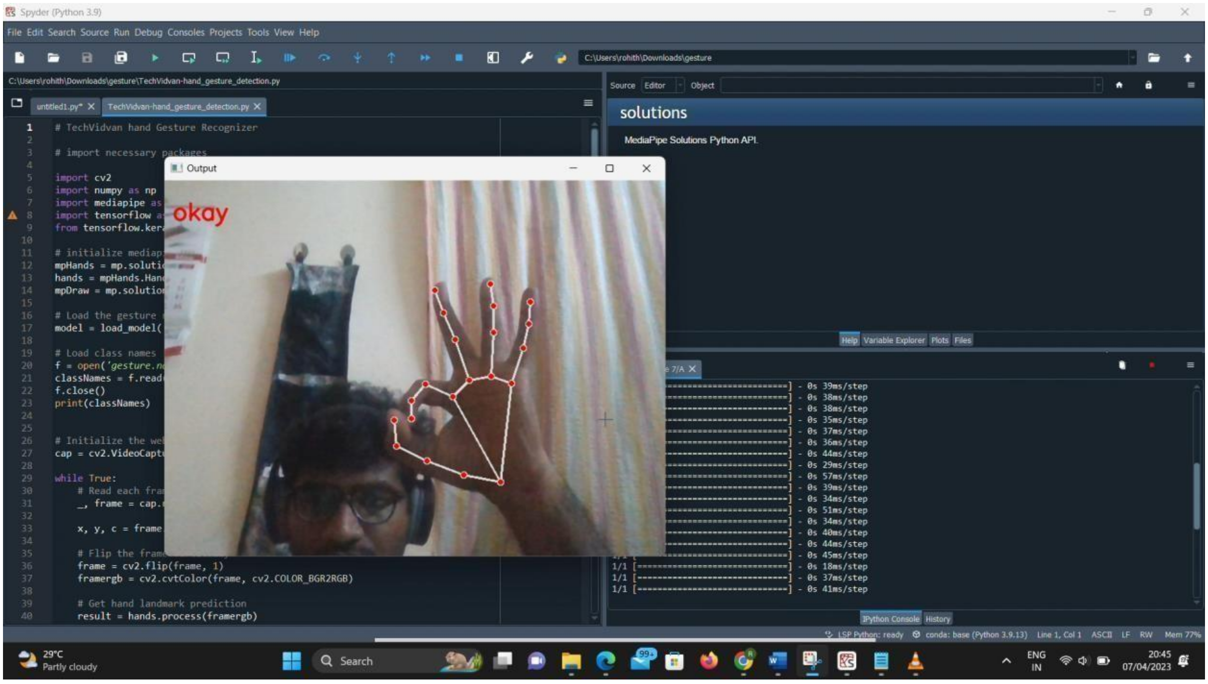

Fig.b. Output Screen

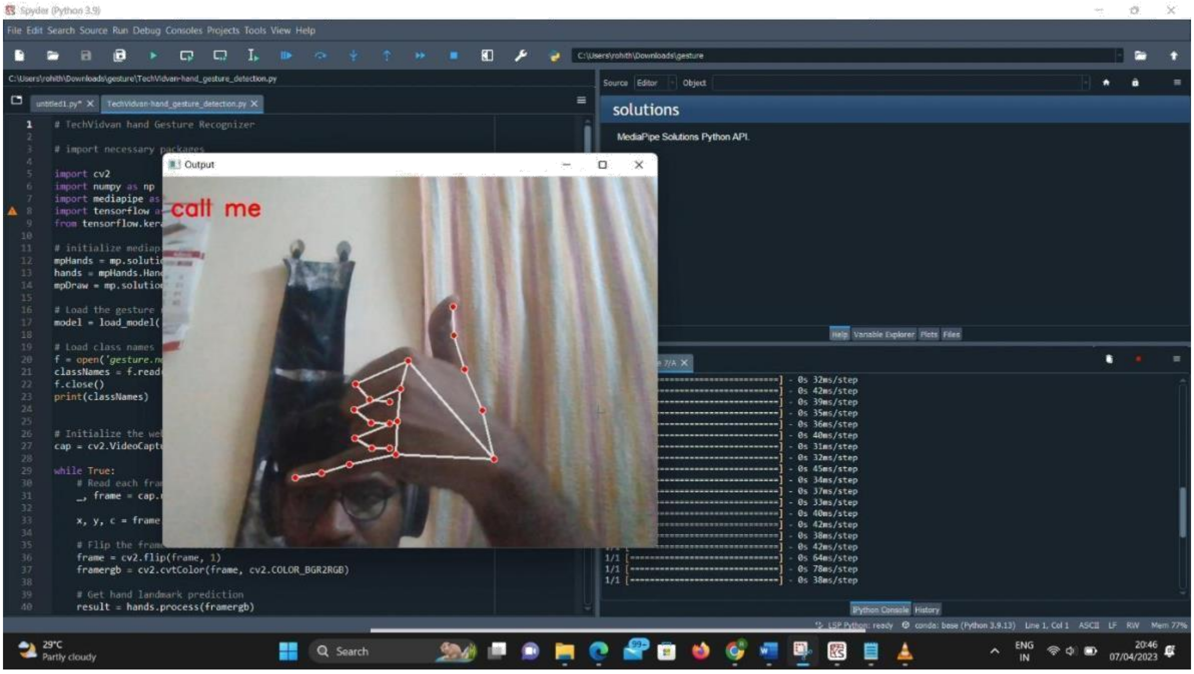

Fig.c. Output Screen

C)RESEARCH PAPER

Enabling gesture-based application interaction on head mounted VR display

Rishi Kalis K[1]
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
rishikalis29@gmail.com

Rohith V[2]
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
rohithrithik123@gmail.com

Dr.Kalaiarasi G[3]
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
kalaiarasi.cse@sathyabama.ac.in

***ABSTRACT*— With the integration of smartphones, VR head-mounted displays that are affordable (HMDs) have made VR more accessible to the general public and increased its widespread use. However, due to their lack of interaction, these systems frequently have limitations. GestOnHMD, a technique for a pipeline for classifying gestures and gesture-based interaction, is presented in this paper. It makes use of the gesture technique in smartphones to detect hand gestures, detecting 21 points on hand on VR headset. Initially, we carried out gesture-elicitation research using our specific headset the Google Cardboard to detect the 21 handpoints which detects the hand gestures and display the meaning of the hand gesture. Based on signal detectability and user choices, we then selected different kinds of gestures to input in the dataset. It uses the neural networks to map the hand points and the cam detects the hand gestures and recognizes it and the displays the meaning of each On the VR device's screen. Each gesture can bemade specifically for your command and its meaning by showing the hand gesture in front ofthe cam and storing it in the dataset. It can be used to read American Signal Language, etc. which makes people to understand hand gestures and might help in future exploring. The output is displayed on the Google Cardboard Virtual Reality device screen which is connected to the laptop.**

Keywords—Google cardboard, Head mounted displays, technique, Virtual Reality, hand gesture, interaction, smartphone, Trinus

I. INTRODUCTION

Google created Google Cardboard, a low-costvirtual reality device that works with the majority of smartphones. Google launched it in 2014 to demonstrate the idea behind "Cardboard VR". The Google Cardboard may be used to construct a broad variety of intricate applications even if it is less sophisticated than the Oculus Rift or HTC Vive. With a basic understanding of programming, it is relatively simple to create applications for Google Cardboard. With a user-friendly interface and preloaded libraries, game engines like Unity3D and Unreal help developers create VR applications. The specifications for theheadset as well as the instructions for putting it together are publicly accessible via Google's website making it possible for individuals to construct their very own virtual reality (VR) headset using components that are readily available. A cardboard piece that has beenprecisely cut into shape, 45 mm-long lenses, magnets or capacitive three pieces of rubber band, tape, and a hook-and-loop fastening are the fundamental components that make up acardboard viewer.

Virtual reality (VR) technology is rapidly advancing, providing users with more immersive and realistic experiences. However, traditional input methods, such as game controllers and keyboards, can be cumbersome and disrupt the immersion in thevirtual world. To overcome this problem, hand gestures have been offered to users as a natural and simple manner for them to engage with virtual items. One of the ways Virtual objects can be interacted with by users in the Google Cardboard VR device is through the use of hand gestures. This study will examine the application of hand gestures to interact with virtual objects in Google Cardboard VR device. Hand gesture recognition technology uses cameras or other sensors to capture and interpret the hand of the user movements, and translate them into actions in the virtual world. This technology has been used in various applications, including gaming, virtual reality, and human-computer interaction. Hand gesture recognition technology has been around for several decades, with the earliest applications dating back to the 1970s. However, it was not until the advent of more powerful computers and advances in computer vision that this technology became more practical for widespread use. Currently, there are several methods used to capture and interpret hand gestures, including computer vision, Leap Motion, and virtual reality headsets with built-in cameras. Computer vision uses cameras to capture images of the user's hands and uses algorithms to interpret the hand movements. LeapMotion uses infrared cameras and LED lights to track the user's hand movements. Virtual reality headsets with built-in cameras use the headset's cameras to track the user's hand movements usinghand gestures as a means of communication inVR has several benefits. First, hand gestures are asimple and easy technique to communicate for users to interact with virtual objects. Second, hand gestures can provide a more immersive experience by allowing users to physically interact with the virtual environment. Third, handgestures can reduce the need for traditional input devices, such as game controllers, which can disrupt the immersion in the virtual world. In this research paper, we'll investigate on how to detecthand gestures using hand gesture algorithms and display the message on the Virtual Reality screen using Trinus Virtual Reality software. We will talk about the most recent developments in devices for recognizing hand gestures, and its applications in Virtual Reality. We will also analyze the benefits and drawbacks of using handgestures as a method of interaction in Virtual reality and identify areas for future research.

II. LITERATURE SURVEY

The combination of mobile technologies and digital reconstruction has the potential to enhance visitors' experiences at historic sites. It is feasible toconstruct an immersive on-site tour of historical locations using the technology that people already carry about with them by designing a mobile app with Google Cardboard. [6] The paper examines ourknowledge creating a mobile application that functions as a virtual tour guide for the ruins of St. Andrews Cathedral. Visiting a prominent Scottish figure famous significant historical sites is provided by the app, which combines traditional media like using a Google Cardboard and a mobile phone, you can view and hear audio, photos, 360° (or four Steradian) videos, 3D video, and panoramas. Together, they enable a self-led exploration of the entire city of St. Andrews with the location-aware medieval St Andrews App. Even though GoogleCardboard isn't available, the app still serves a useful purpose by providing audio commentary and visual content about this historic structure. The limitations of this paper is that the users may not be able to utilize the system when they are offline and may suffer slow loading times when they are online, which could have a detrimental effect on their user experience.

In paper [7], Researchers discuss the idea of controlling how immersive head-mounteddisplays (HMDs) are by using LCD panels. The paper proposes using LCD panels similar to those found in common active-stereo glasses are a common sight in cinemas to replace the cowling around typical HMDs of the "ski- goggle" variety. To adjust how much of the user can see the outside world peripherals, the panels are manageable with very basic standalone a microcontroller or electronic circuits due to the fact that it enables users can use it to see objects in their local environment, such as the keyboard and mouse, and it doesn't increase system latency. By [7] delivering natural cues when necessary, it can also be utilized to lessen cybersickness, this can significantly improve the usability of consumer HMD's. The issue with thepaper is that the design is extremely difficult, both financially and technically.

The adoption and application within virtual reality (VR) have undergone a paradigm shift in recent years. The [8] development of low-cost VR headsets and advancements in graphics rendering have made VR accessible to everyday consumers. The Google Cardboard VR viewers, which are compatible with the majority of smart phones and only cost a few dollars, allow mobile VR needs to become more widely used. But at the moment, theseheadsets are mostly utilized for media consumption or passive pleasure in 360 degrees, not for active virtual space exploration. [8] Our initial assessment of three traveling and navigating strategies is presented in this paper. A smart glove that records hand motions and translates them into legible text. The issue in the paper is one continuous motion with the option to change the speed and stop would appear to be the best compromise because none of the three strategies examined in this study provided an optimum answer for travel within mobile VR.

A crucial system [1] for the deaf and mute is presented in the paper: an automatic translator of hand signs. Here, the system's anticipated requirements and degree of performance are outlined. The article lists system components and elaborates on how hardware wire connections and component assembly work. The software component of this system is also very sophisticated, with basic setup and recognition algorithms included. The study discusses the difficulties in detecting vague measures and suggests corresponding technical solutions. The outcomes of the trial make it clear that the system has the ability to assist specific people and communities. The issue with the paper is that especially given that the majority of the letters could be recognized by the algorithm is (20 out of 26)

Only for counting fingers and recognizing numbers, convexity hull algorithms can be used. With the use of the contour analysis procedure, a study akin to this can be carried out on the Marathi and English alphabets. [3] By comparing hand templates and taking contour curve shapes into consideration, recognition can be accomplished. It is simple to identify hand gestures in contour analysis, by taking into consideration vector values, regardless of scaling and shape. Even by creating an application that converts sign language to voice output, a barrier to communication between people with speechimpairments can be lifted. The [3] software's goalis to display a real- time system for recognizing hand gestures based on the detection of a few shape-based criteria, such as orientation, the centroid of the Center of Mass, the status of the fingers, and the placement of the thumb in relation to raised or folded fingers. By bearing in mind how the human hand, which has one thumb and four fingers, is shaped, this is accomplished. The issue with the paper is that the algorithmused can be replaced with contour analysis process for better results.

In order to help deaf individuals and give voice to the voiceless, this paper's main goal is to break down communication barriers between speech-impaired people and able-bodied people.[2] This goal is accomplished by employing the system that is suggested. The gloves convert sign gestures into text in both English and Malayalam and are reasonably priced. The suggested solution offers increased precision, is portable, and uses relatively little power. [2] Flex sensor, microcontroller, and Bluetooth module HC 05 are used to convert Sign language, which is then displayed on an Android phone. The issue with the paper is that the phrases to be recognized is very less and accuracy should be increased.

The technology described in the paper transforms movements that are static into voice. It also goes by the name GesTALK [4]. The self-designedand reasonably priced Data Glove serves as the main input device. It has two primary modes of operation. While speaking a whole string by concatenating words in the second mode, the system says a single alphabet in the first mode in response to a static gesture performed by the user. The words in a string are spelled out using avariety of alphabet movements. Both the American Sign Language (ASL) and the PakistanSign Language (PSL), as well as other language models, have been proved to function with this system. [4] The issue with the paper is that cochlear implants are not a viable option for all HI people and is a frustrating mode of communication. Hiring experienced interpreters is expensive and challenging, people cannot rely on them in daily life.

The purpose of the study was to break down the difference in communication between hearing-impaired individuals and the general population. As a result of the way the sub-modules are built and the particulars of each module, a framework for MSL recognition is therefore described. [5] The framework under discussion is divided into three main sections: the first section, titled "Analysis of MSL Modules," deals with the extraction of helpful features that can be used to determine system needs and accurately detect complex indications. The development of an ideal [5] DataGlove based on the findings of system experiments and sensor testing is the focus of the second module (DataGlove Modules). In the third module, DataGlove's data collection is preprocessed and categorized (Gesture Recognition Module). This conversation about development might help close the communication gap between hearing-impaired people and others. The issue with the paper is that it recognizes only simple hand gestures and accuracy needs to be improved.

III. PROPOSED

SYSTEM A. Display

technology

An Android device made by Xiaomi is the tool that was used in this project known as Note 9 pro. The mobile operating system for this smartphone is Android 'Snow Cone' (version 12.0.0). In addition to working with Google Cardboard, it can run Google VR applications. Before adding any new programs to the phone, "Debug Mode" must be set up. Developers have access to the phone's debug mode, which lets them install new programs and, if necessary, troubleshoot bugs.

Fig.1.Note 9 pro

B. Viewer for Google Cardboard

Android or iOS smartphones with screens between 4 and 6 inches can use the Google Cardboard viewer. All phones that are compatible with it can use the interactive button.

There are just six parts needed to assemble a cardboard viewer:

1. Cardboard
2. Lenses
3. Magnets
4. Two strips of Velcro
5. Rubber band
6. NFC tag (optional)

These components are widely available and inexpensive practically anywhere in the world. The Google website offers simple, step-by-step directions for building a cardboard viewer using these components...

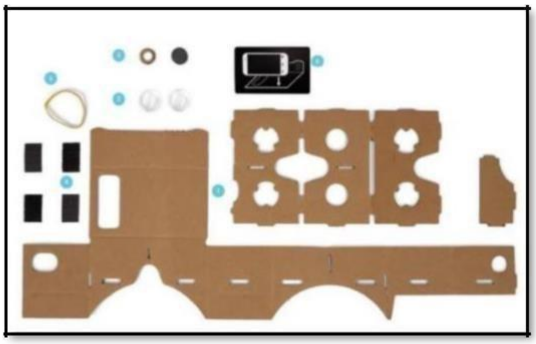

Fig.2.Google cardboard viewer

C. Compatible gadgets

1. Android 4.1 and later are compatible with Cardboard.
2. iOS 8.0 or later are compatible with Cardboard.

Not all viewers are compatible with all mobile devices. You might not own a gadget with a gyroscope if you encounter a device incompatibility notice.

Fig.3. Flow diagram for Google cardboard

Gesture recognition is a topic of current study in the area of Human-Computer Interaction technology. In addition to translating sign language and operating robots, it can also be used to make music in virtual settings. Using Tensorflow and the MediaPipe framework in OpenCV and Python, we will create a real-time Hand Gesture Recognizer. Google created MediaPipe, a framework for adaptable machine learning solutions. It is a lightweight, cross- platform framework that is available under an open-source license. MediaPipe includes a number of machine learning (ML) pre-trained solutions, including face and object detection, pose estimation, hand and object recognition, and more. TensorFlow is an open-source machine learning and deep learning library developed by the Google Brains team. Although it is applicable to many different applications, deep neural networks are given special attention. Artificial neural networks are referred to as neural networks are another name for them. The brain of deep learning algorithms, itis a subset of machine learning. The human brain served as inspiration for the development of neural networks. It resembles the communication betweenneurons in biological systems. Node layers, which comprise input, one or more neural networks are made up of hidden and output layers.

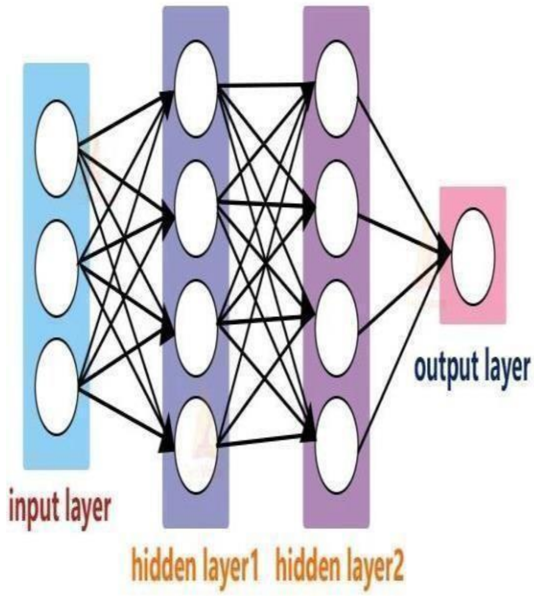

Fig.4. Neural Networks with two hidden layers

To identify the hand and the hand critical points, we will first use MediaPipe. Each hand that MediaPipe detects returns a total of 21 critical points. In order to identify the hand stance, these crucial details will be sent into a network of pre-trained gesture recognizers.

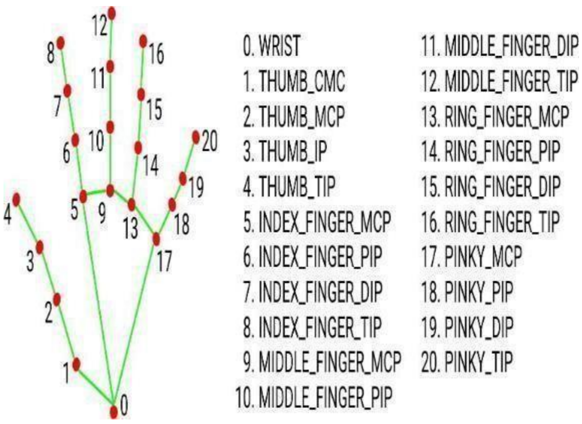

Fig.5. Key points to recognize hand

gestures Requirements for this project include:

1. Python - 3.88
2. OpenCV - 4.5

To install OpenCV, use "pip install opencv-python".

3. MediaPipe - 0.8.5

To install MediaPipe, use "pip install mediapipe". 4. TensorFlow version 2.5
Run "pip install TensorFlow" to set up the TensorFlow module.

5. Numpy – 1.19.3

Steps to follow:

- Import necessary packages.
- Initialize models.
- Read frames from a webcam.
- Detect hand key points.
- Recognize hand gestures.

Fig.6. Gesture recognition for google cardboard

When the hand is gestured in front of the webcam, the process begins and it starts plotting the hand with the key points and storing data on the hand gesture. After taking data key points it is then checked with the dataset which is pre trained model and then it compared with it to recognize what handgesture was gestured in front of the webcam. The output message shows the meaning of the hand gesture done in front of the webcam. The output is displayed in the Google Cardboard Virtual Reality device which is connected to the laptop via Software called Trinus Virtual Reality. The Trinus Virtual Reality Server is installed in the laptop and the Trinus Virtual Reality Client is installed in the Google Cardboard Virtual Reality device. The laptop and the Google Cardboard

Virtual Reality device can be connected via Universal Serial Bus (USB) wire or via Wi-Fi in which both the laptop and the Virtual Reality device should be connected. For better performance and avoiding buffering, the laptop and the Virtual Reality device are connected via USB wire. When Trinus Virtual Reality Server and Trinus Virtual Reality Client are started the connection of the laptop and the Virtual Reality device is successful. Now the user wearing the Google Cardboard Virtual Reality device can view the output in a Virtual Reality experience.

IV. RESULT

The Google cardboard VR device which was made is used as VR device but has limitations to its use by the users. In order to increase the features and functions performed by the users, wehave added hand gesture recognition feature or method that can capture and interpret users hand movements and translate it into the meaning for each hand gesture and display it on the Virtual Reality screen of the Google Cardboard Virtual Reality device and also it is designed to make interaction between people and the virtual things in the virtual world. The user will be able toengage with virtual things in a natural way thanksto its design and also help users with recognition of hand movements. Using Python and OpenCV, we created a hand gesture recognizer. For gesture identification and detection, we used theMediaPipe and Tensorflow frameworks, respectively. The hand gesture recognition is used to help people who don't know sign language and cannot understand it and this will help them in understanding sign language and can help converse with others who are dumb and deaf as well. Understanding sign language is just one example here. Hand gesture recognition can be used in many ways for many uses. The result of our project is that it recognizes the hand gesture and displays the message on the screen of the Google Cardboard Virtual Reality device and the user can view the message in Virtual Reality experience.

V.CONCLUSION

The proposed system in this research paper combines several of a webcam and hand gesture recognition software to capture and interpret hand gesture of the user, and translate them into meaning of the hand gesture in the virtual world. The system was created to offer consumers a simple and natural method to interact with virtual objects in Google Cardboard VR device. The proposed system offers several benefits such as providing a more immersive experience, reducing the need for traditional input devices, and being a low-cost solution. However, there are also some drawbacks such as the technology being complex and difficult to set up and calibrate and being affected by the user's physical limitations. The proposed system is still in its early phases, and additional study is required to increase the precision and dependability of the hand gesture recognition technology and to make it more accessible to users with physical limitations. Overall, the proposed system represents a promising solution for natural and intuitive interaction with virtual objects in virtual reality environments and will be beneficial for a variety of purposes such as for understanding sign languages like ASL, BSL, etc... As thetechnology continues to evolve, it is expected to become more accurate and reliable and can be used in many places for many different uses providing a more immersive and realistic experience for users.

ACKNOWLEDGMEN
T

I would like to thank Dr. G. Kalaiarasi for helping me and providing valuable feedback throughout the project.

REFERENCES

[1] Elmahgiubi, Mohammed, Mohamed Ennajar, Nabil Drawil, and Mohamed Samir Elbuni. "Sign language translator and gesture recognition." In 2015 Global Summit on Computer & Information Technology (GSCIT), pp. 1-6. IEEE, 2015

[2] Kunjumon, Jinsu, and Rajesh Kannan Megalingam. "Hand gesture recognition system for translating indian sign language into text and speech." In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 14-18. IEEE, 2019.

[3] Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image-based hand gesture recognition techniques." In 2016 online international conference on green engineering and technologies (IC-GET), pp. 1-5. IEEE, 2016.

[4] Khambaty, Yousuf, Rolando Quintana, Mehdi Shadaram, Sana Nehal, Muneeb Ali Virk, Waqar Ahmed, and Ghayas Ahmedani. "Cost effective

portable system for sign language gesture recognition." In 2008 IEEE International Conference on System of Systems Engineering, pp. 1-6. IEEE, 2008.

[5] Ahmed, Mohamed Aktham, B. B. Zaidan, A. A. Zaidan, Abdullah Hussein Alamoodi, Osamah Shihab Albahri, Z. T. Al-Qaysi, Ahmed Shihab Albahri, and Mahmood M. Salih. "Real-time sign language framework based on wearable device: analysis of MSL, DataGlove, and gesture recognition." *Soft Computing* 25, no. 16 (2021): 11101-11122.

[6] Fabola, Adeola, Alan Miller, and Richard Fawcett. "Exploring the past with Google Cardboard." In *2015 Digital Heritage*, vol. 1, pp. 277-284. IEEE, 2015.

[7] Lindeman, Robert W. "A low-cost, low-latency approach to dynamic immersion in occlusive head-mounted displays." In *2016 IEEE Virtual Reality (VR)*, pp. 221-222. IEEE, 2016.

[8] Powell, Wendy, Vaughan Powell, Phillip Brown, Marc Cook, and Jahangir Uddin. "Getting around in google cardboard–exploring navigation preferences with low-cost mobileVR." In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, pp. 5-8. IEEE, 2016.

[9] Patterson, Kate. "Genome gazing: A 360° stereoscopic animation for Google cardboard." In *2017 IEEE Virtual Reality (VR)*, pp. 439-439. IEEE Computer Society, 2017.

[10] Mohatta, Shreyash, Ramakrishna Perla, Gaurav Gupta, Ehtesham Hassan, and Ramya Hebbalaguppe. "Robust hand gestural interaction for smartphone based AR/VR applications." In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 330-335. IEEE, 2017.

[11] Luo, Siqi, and Robert J. Teather. "Camera- based selection with cardboard HMDs." In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1066-1067. IEEE, 2019.

[12] Syed, Rameez Rehman, and ShahzadRasool. "A Framework for Cardboard Based Augmented Reality." In *2018 International Conference on Frontiers of Information Technology (FIT)*, pp. 287-292. IEEE, 2018.

[13] Chen, Min-Bin. "The cardboard VR game development tool." In *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pp. 438-441. IEEE, 2019.

[14] Rodrigues, Adriel, and Jean Felipe Patikowski Cheiran. "Virtual look around: Interaction quality evaluation for virtual tour in multiple platforms." In *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pp. 47-56. IEEE, 2020.

[15] Shanmugam, Sampath, Vaishak R. Vellore, and V. Sarasvathi. "Vrnav: A framework for navigation and object interaction in virtual reality." In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pp.1-5.IEEE,2017