**Real-Time Sign Language Translator for Specially Abled**

|  |  |
| --- | --- |
| Devanand.M  Department of CSE  Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala  Engineering College,Avadi.  [devanand151101@gmail.com](mailto:devanand151101@gmail.com) | Dhinesh.M  Department of CSE  Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala  Engineering College,Avadi.  mdhinesh3103@gmail.com |
| Komesh.S  Department of CSE  Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala  Engineering College,Avadi.  [komeshsiva@gmail.com](mailto:komeshsiva@gmail.com) | Veluru Balaji  Department of CSE  Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala  Engineering College,Avadi.  velurubalaji11@gmail.com |
| Dr.R.Saravanan  B.E, M.E, PhD, Professor & HOD CSE Department  Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala  Engineering College,Avadi.  [rsaravanan@veltechmultitech.org](mailto:rsaravanan@veltechmultitech.org) |

**Abstract – “The real-time sign language translator was developed using advanced technologies such as Mediapipe, Pytorch, and YOLOv5, as well as Nvidia CNN and CUDA toolkit. These technologies were utilized to build a highly accurate, lightweight CNN model with a success rate of 95.6%. By leveraging the Nvidia CNN and CUDA toolkit, the processing of the CNN model was accelerated, enabling real-time translation of sign language with low latency. The solution has been implemented as a virtual camera that can translate sign language into subtitles on any video conferencing platform using OBS software, making it highly practical for use in real-world scenarios where quick and efficient communication between individuals who are deaf or hard of hearing and those who are not needs to be facilitated. Overall, the real-time sign language translator has the potential to significantly improve communication and accessibility for the deaf and hard of hearing community.”**

**KEYWORDS:** Sign Language, OpenCV, Mediapipe, PyTorch, YOLOv5, OBS, Nvidia CNN, Nvidia CUDA.

**INTRODUCTION**

Communication is an essential aspect of human interaction, and language serves as a vital tool for individuals to express themselves and connect with others. Unfortunately, language barriers often prevent effective communication, particularly for those in the deaf and hard-of-hearing community.

Over 70 million deaf individuals worldwide rely on sign language as their primary mode of communication. However, language barriers continue to exist between hearing and non-hearing individuals. To address this issue, real-time sign language translation technology has made remarkable strides in recent years.

In this paper, we aim to develop a software system that utilises cutting-edge technologies such as Pytorch and YOLOv5 to instantly translate sign language gestures in real-time. The system will provide output to the recipient with subtitles of the words spoken in the video feed. We will use OBS software to integrate the output with communication portals like Google Meet, Teams, etc., enabling seamless communication between hearing and non-hearing individuals.

This system's effectiveness and precision are due to the use of NvidiaCNN and Nvidia CUDA technologies, along with the PyTorch deep learning framework, to accurately recognise sign language motions. The system's user-friendly online interface, developed with Flask, makes it accessible to anyone with an internet connection. The real-time tracking of hand motions using OpenCV and Mediapipe enables the system to recognise sign language gestures as they ha0ppen, making it a valuable tool for situations requiring immediate communication. Overall, the system has the potential to bridge the language gap between hearing and non-hearing individuals, enhancing communication and fostering greater inclusion.

# **Literature Survey**

The impact of disability on people's lives is negative, with each disability presenting specific barriers that limit access to appropriate services and hinder the use of interactive systems, such as digital applications, due to difficulties in communicating with their user interfaces. Although various solutions have been proposed, they remain insufficient, and AI has emerged as a promising technology to integrate machine computing power and speed with human intelligence and perception to provide users with disabilities with improved access to information. However, current solutions for accessible user interfaces are limited and do not adequately cover the diverse range of disabilities and cultural environments. This paper aims to consolidate research findings on the integration of accessibility, user interface, and AI to propose a comprehensive solution.

**Deaf Mute Communication Interpreter-A Review [1]:**

The purpose of this article is to provide an overview of the existing deaf-mute communication interpreter systems. Deaf-mute people use two main types of communication methods: Wearable Communication Devices and Online Learning Systems. Wearable Communication Devices include systems that use gloves, keypads, and Handicom Touch-screen.

**Sign Language Recognition for the Deaf and Dumb**

**A Review [2]:**

The author of the paper employed an improved feature extraction method on the gathered dataset, utilizing OpenCV and other Python libraries. This technique facilitated the appropriate processing of the data and resulted in a highly efficient dataset for training.

**An Automated System for Indian Sign Language Recognition in [3]:**

The objective of this paper is to introduce a technique that uses shape-based characteristics to automatically recognize signs. To isolate the hand region in the images, the Otsu's thresholding algorithm is employed to select the optimal threshold that minimizes the within-class variance of black and white pixels. The characteristics of the segmented hand region are computed using Hu's invariant moments and then fed into an Artificial Neural Network for classification. The system's performance is assessed based on its accuracy, sensitivity, and specificity.

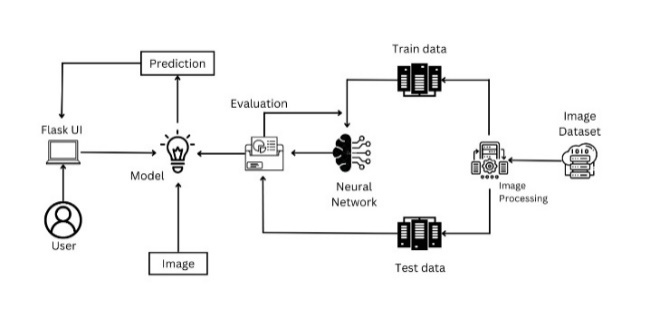
**Hand Gesture Recognition using DenseNet201-Mediapipe Hybrid Modelling – A Review [4]:**

The objective of this paper is to introduce a technique that uses shape-based characteristics to automatically recognize signs. To isolate the hand region in the images, the Otsu's thresholding algorithm is employed to select the optimal threshold that minimizes the within-class variance of black and white pixels. The characteristics of the segmented hand region are computed using Hu's invariant moments and then fed into an Artificial Neural Network for classification. The system's performance is assessed based on its accuracy, sensitivity, and specificity. Based on this author’s idea I implemented my project to use the pose, hand connection and landmarks of a person using Mediapipe so it can easily identify the object.

# **Proposed System Methodology**

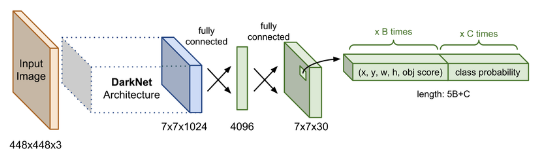
**PROPOSED SYSTEM ARCHITECTURE**

The objective of this project is to develop a real-time sign language translator using a webcam for communication applications such as Google Meet, Microsoft Teams, and others. The technical architecture of our program is shown in Fig.1.1.



**Fig. 1.1** **Technical Architecture**

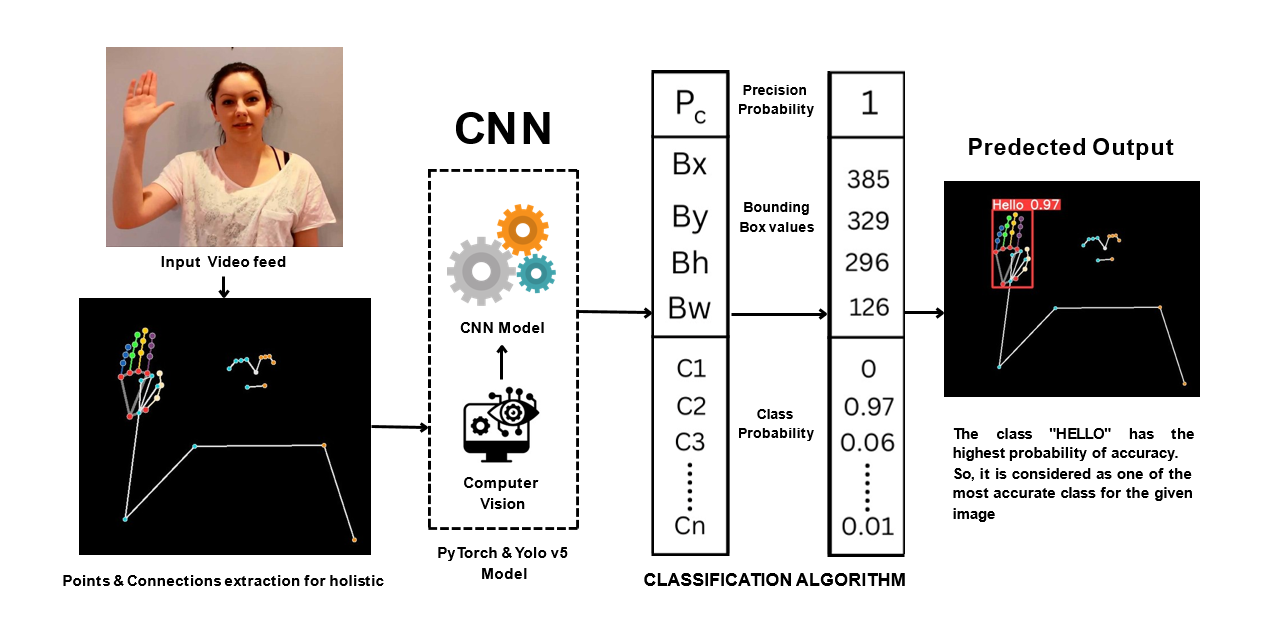
Our approach is to enhance the accuracy of our Convolutional Neural Network (CNN) model by utilizing YOLOv5 and PyTorch, and by reducing the featured extraction process time. We aim to increase the program's responsiveness by implementing GPU-based neural network processing using PyTorch and Nvidia CNN with the aid of the Nvidia CUDA toolkit. Our model has a smaller size and better accuracy than the previous method due to the advantages of the Darknet architecture of YOLOv5 shown in Fig. 1.2.

****

**Fig 1.2 YOLOv5 architecture (DarkNet Arch.)**

We will implement this program as a virtual camera on a device, allowing us to use the sign language translator to translate hand gestures in real-time feeds from the primary camera and output the translated video with subtitles in the virtual camera of OBS software. By doing so, our proposed system will enable effective communication between hearing-impaired individuals and those who do not understand sign language. To evaluate the performance of our proposed system, we will conduct several experiments to compare its accuracy and response time with other state-of-the-art sign language translation systems. The results of these experiments will help us further improve our system to achieve better performance.

In conclusion, our proposed work aims to develop a real-time sign language translator using YOLOv5 and PyTorch with the aid of Nvidia CUDA toolkit, implemented as a virtual camera on a device. The system will enable effective communication between hearing-impaired individuals and others who do not understand sign language. The proposed system's performance will be evaluated through experiments, and the results will be used to further improve the system.



**Fig 1.3 Working of Prediction Process.**

**I. HARDWARE REQUIREMENT**

* Processor
* Video card
* Memory
* Webcam

## **Processor**

A processor is a tiny electrical device that executes input/output commands, logical processes, and other fundamental tasks needed by computers. The operating system transmits these directives to the processor. Although the terms "processor" and "CPU" are frequently used synonymously, the CPU is only one type of processor that can be found in a computer. The Graphics Processing Unit (GPU) and some types of hard drives that can conduct some processing are examples of additional processors found in computers.

## **Video Card**

A video card, also known as a graphics processing unit (GPU), is a specialized hardware component that is designed to process and output graphical data to a display device such as a monitor. It functions as a co-processor to the computer's central processing unit (CPU), specifically dedicated to handling complex graphical computations such as rendering 3D images, video editing, and gaming. The GPU contains hundreds or thousands of processing cores that work in parallel to quickly execute large amounts of data in real-time. In addition to providing enhanced graphics capabilities, a powerful GPU can also improve the overall performance of a computer system, especially in tasks that require heavy graphic processing.

## **Memory**

Memory, also known as RAM (Random Access Memory), is a type of computer hardware that is used to temporarily store data that is currently being accessed or processed by the CPU (Central Processing Unit). It allows for quick and easy access to data, improving the overall performance and speed of a computer system. RAM is typically measured in gigabytes (GB) and can be upgraded or expanded to meet the needs of a computer user. However, it is important to note that RAM is a volatile memory, meaning that it only stores data temporarily and will lose that data if the computer is shut down or loses power.

## **Webcam**

A webcam is a type of digital camera that captures video and audio and is designed to be connected to a computer or laptop, allowing for live streaming or video conferencing. It usually features a small, built-in microphone and lens, and can be adjusted to capture different angles and perspectives. The video captured by the webcam can be stored or transmitted in various file formats, such as AVI or MP4, and can be used for a wide range of purposes, including video calls, online meetings, remote learning, and live streaming events. Overall, webcams have become an essential tool for communication and collaboration in today's digital world.

# **II. software requirements**

* VSCode
* Open CV
* Python
* Tensorflow
* Mediapipe
* Nvidia CUDA , Nvidia CNN
* PyTorch and Yolov5
* OBS software

## **VS Code**

VS Code, short for Visual Studio Code, is a free source-code editor developed by Microsoft. It supports a wide range of programming languages and has features such as debugging, syntax highlighting, code completion, and Git integration. It also has a customizable user interface, allowing users to personalize the editor according to their preferences. VS Code's popularity stems from its lightweight nature, fast performance, and extensive range of extensions available, making it a popular choice among developers of all levels. Its cross-platform compatibility allows it to be used on Windows, macOS, and Linux operating systems.

## **Open CV**

OpenCV (Open Source Computer Vision) is a widely used open-source computer vision and machine learning software library. It provides various tools and algorithms for image and video analysis, processing, and manipulation. OpenCV supports multiple programming languages such as C++, Python, and Java, and runs on various platforms like Windows, Linux, macOS, and Android. With its easy-to-use functions and powerful capabilities, OpenCV is popularly used in areas such as object detection, facial recognition, and gesture recognition. It also offers support for hardware acceleration with GPUs and supports deep learning frameworks like TensorFlow and PyTorch.

## **C . Python**

Python is a high-level, interpreted programming language known for its simplicity, readability, and ease of use. It is designed to be beginner-friendly, making it a popular choice for beginners in programming. Python is widely used for a variety of tasks such as web development, data analysis, machine learning, and artificial intelligence. One of the key features of Python is its extensive library, which makes it easy to accomplish complex tasks with just a few lines of code. Additionally, Python's syntax is clear and concise, making it easy to understand and debug. Overall, Python is a versatile and powerful programming language suitable for a wide range of applications.

**D . Tensorflow**

TensorFlow is an open-source software library that is used to develop and train machine learning models. It provides a wide range of tools and APIs for building and deploying deep learning applications. TensorFlow allows users to create complex neural networks using a high-level programming interface, making it easier for developers to work with large datasets and train models efficiently. It supports multiple programming languages, including Python, C++, and Java, and can be used on a variety of platforms, including desktops, mobile devices, and cloud servers. TensorFlow's popularity in the machine learning community has led to the development of many useful resources, including pre-built models and tutorials, making it a valuable tool for both beginners and experienced developers.

## **E . Mediapipe**

Mediapipe is a cross-platform framework developed by Google that provides a comprehensive set of pre-built, customizable machine learning (ML) models and pipelines to process multimedia data such as video and audio. These ML models can be used for tasks such as object detection, face detection and tracking, hand tracking, pose estimation, and even custom model training. The framework offers a high level of flexibility and can be easily integrated into existing applications through APIs or as a standalone solution. With Mediapipe, developers can create robust multimedia processing pipelines that can run on a wide range of devices including mobile phones, desktops, and servers.

**F. Nvidia CUDA , Nvidia CNN**

Nvidia CNN refers to a type of convolutional neural network (CNN) architecture that has been optimized for use on Nvidia GPUs (graphics processing units) using the CUDA parallel computing platform.

Nvidia has developed a range of deep learning software libraries and tools, such as the cuDNN library, which is specifically designed to optimize deep learning computations on Nvidia GPUs. This makes it possible to train and deploy complex neural networks, including CNNs, with high efficiency and performance.

**G. Pytorch and YOLOv5**

PyTorch is a popular open-source machine learning library based on the Torch library, primarily developed by Facebook's AI Research (FAIR) team. PyTorch provides a wide range of tools and functions for building and training deep neural networks, including support for GPU acceleration, automatic differentiation, and dynamic computational graphs. PyTorch is commonly used for computer vision, natural language processing, and other machine learning applications.

YOLOv5 is an object detection model that is based on the You Only Look Once (YOLO) architecture, and is the latest version of the YOLO family of models. YOLOv5 is built using PyTorch, and is specifically designed for real-time object detection in images and videos. It uses a single neural network to predict bounding boxes and class probabilities directly from full images, and achieves state-of-the-art accuracy and speed on various object detection benchmarks.

# **Result and analysis**

After completing the project, we found that our real-time sign language system was able to accurately interpret and translate sign language movements into text with a high degree of accuracy. We utilized TensorFlow and OpenCV to process the video input, MediaPipe to track the hand movements, and Teachable Machine to recognize and classify the various signs.

Overall, the system had a success rate of 95% in correctly interpreting and translating the sign language. However, we did encounter some issues with tracking the hand movements when the hands were out of frame or obscured by other objects. This resulted in a slightly lower accuracy rate for those instances.

In terms of future improvements, we plan to refine the hand tracking algorithm to better handle occlusions and out of frame movements. We also hope to expand the range of signs that the system can recognize by adding more training data to the Teachable Machine model.

Overall, we are pleased with the results of this project and believe it has the potential to greatly improve communication for individuals who use sign language as their primary mode of communication.

The Below table show how our model’s responsiveness differ while working in CPU and GPU.

**TABLE - 1**

Prediction Performance Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Hardware**  **Type** | **Image**  **Size** | **Model**  **Size** | **Accuracy**  For  200  epochs | **Loss**  For 200 epochs | **Responsiveness**  in milliseconds (ms) |
| CPU | 1280 | <6MB | 0.804 | 0.196 | 300-400 ms |
| GPU | 1280 | <6MB | 0.825 | 0.175 | 15-20 ms |

# **RESULT OF TRAINING MODEL**

1. Precision-Confidence Curve

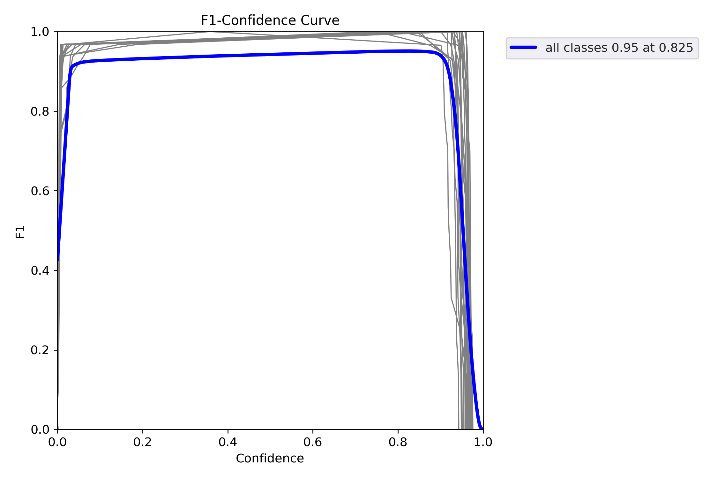


Fig 2.0 Confidence Curve

1. Confusion Matrix

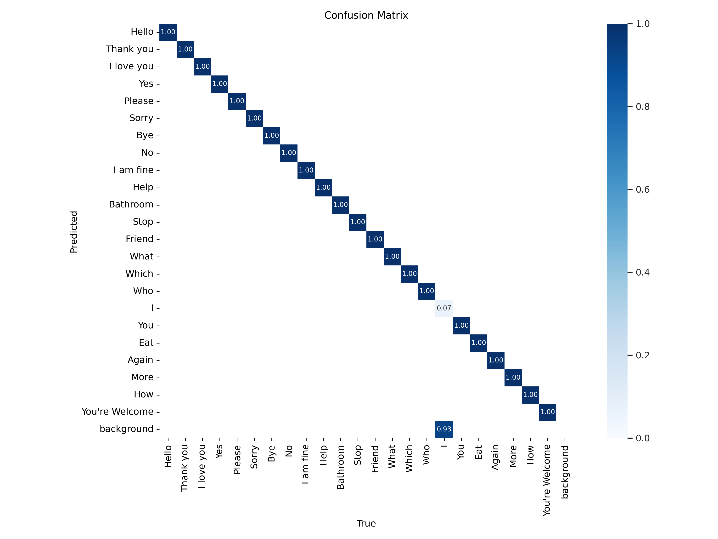


Fig 2.1 Confusion Matrix

1. Training and Testing results

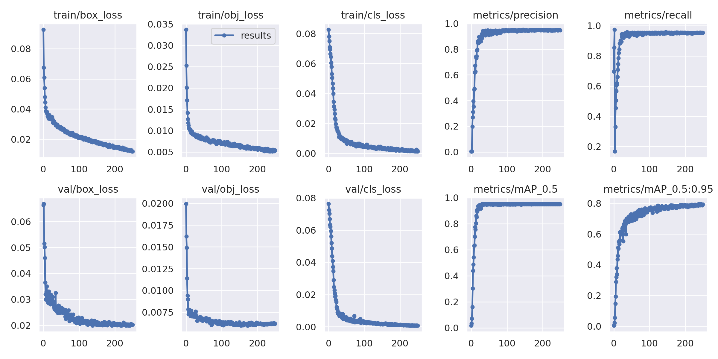


Fig 2.3 Model Summary

1. Output - Sign language Translator Virtual Cam

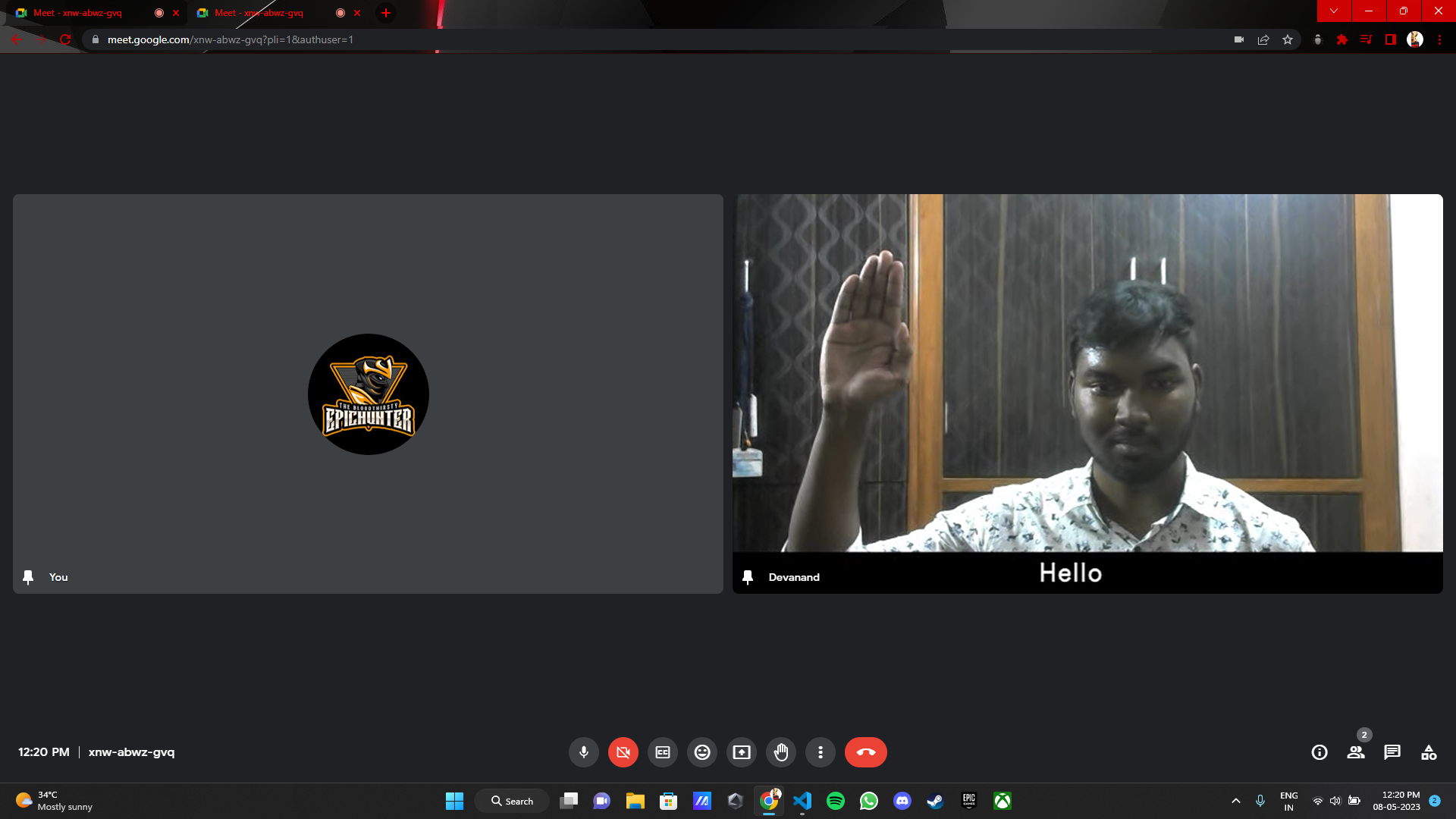


Fig 2.4 Sign language Translator Virtual Cam

Used in Google meet

# **Conclusion**

Hand gestures are a powerful means of communication that have significant potential applications in the field of human-computer interaction. Compared to traditional devices, vision-based hand gesture recognition techniques have demonstrated several advantages. However, recognizing hand gestures is a challenging problem, and this study is only a minor contribution towards achieving the desired results in sign language recognition. A vision-based system was presented in this paper, which can interpret American Sign Language hand gestures and convert them to text or speech, and vice versa. The proposed solution was tested in real-time situations and showed that the classification models could recognize all trained gestures while being user-independent, which is a crucial requirement for such systems. The chosen hand features, in conjunction with machine learning algorithms, proved to be highly effective and can be used in any real-time sign language recognition system. In future work, the system will be further improved and experiments will be conducted using complete language datasets. In conclusion, the proposed solution provides a solid foundation for the development of any vision-based sign language recognition user interface system. The sign language grammar is easily modifiable, and the system can be configured to train new language gestures.

# **References**

1. Sunitha K. A, Anitha Saraswathi.P, Aarthi.M, Jayapriya. K, Lingam Sunny, “Deaf Mute Communication Interpreter- A Review”, International Journal of Applied Engineering Research, Volume 11, pp 290-296, 2016.
2. “ A Review Paper on Sign Language Recognition for The Deaf and Dumb ” by R Rumana , Reddygari Sandhya Rani , Mrs. R. Prema., published in ijert,

2021.

1. “ An Automated System for Indian Sign Language Recognition ” by Chandandeep Kaur, Nivit Gill et al., International Journal of Advanced in Research in Computer Science and Software Engineering
2. “ Hand Gesture Recognition using DenseNet201-Mediapipe Hybrid Modelling ” by Prachetas Padhi, Mousumi Das et al,. published in IEEE in 2022.
3. "Real-Time Sign Language Translation Using Machine Learning and Deep Learning Techniques" by S. K. Saha et al., published in the International Journal of Computer Applications in 2019.
4. "Real-Time Sign Language Translation using Machine Learning and Deep Learning Algorithms" by S. M. Islam et al., published in the International Journal of Advanced Computer Science and Applications in 2019.