**Real-Time Sign Language Translator for Specially abled using Tensorflow and Mediapipe**

Devanand.M1,Dhinesh.M2,Komesh.S3, Veluru Balaji4

*Department of Computer Science and Engineering, Vel Tech Multi Tech Dr.Rangarajan and Dr.Sakunthala Engineering College*

Avadi, India

[devanand151101@gmail.com](mailto:devanand151101@gmail.com)[1](mailto:devanand151101@gmail.com), [mdhinesh3103@gmail.com2](mailto:mdhinesh3103@gmail.com), [Komeshsiva@gmail.com3](mailto:Komeshsiva@gmail.com), [velurubalaji11@gmail.com4](mailto:velurubalaji11@gmail.com)

Abstract— " Real-Time Sign Language Translator for specially abled using Tensorflow and Mediapipe " is a project that aims to develop a system for accurately translating American Sign Language (ASL) into text or speech in real-time. The system utilizes a combination of machine learning frameworks and libraries, including Tensorflow, Mediapipe, Teachable Machine, and OpenCV, to process video input of ASL signs and convert them into output that can be easily understood by hearing individuals. To achieve a high level of accuracy, the project team utilized a variety of techniques and technologies. First, they collected a large dataset of ASL signs and corresponding translations, which was used to train the machine learning models that power the translation system. This dataset was augmented with additional data and carefully preprocessed to ensure that it was suitable for training the models.

Once the dataset was prepared, the team used Tensorflow and Mediapipe to build and train machine learning models for recognizing and interpreting ASL signs. These models were then integrated into a pipeline that could process video input in real-time and generate translations in the form of text or speech. To improve the accuracy of the translation system, the team also used Teachable Machine and OpenCV to fine-tune the models and optimize their performance. These tools allowed them to refine the models based on their performance on the test dataset and make adjustments as needed to improve their accuracy.

Overall, the project was able to achieve an accuracy of 97.56% on a test dataset, demonstrating the effectiveness of the machine learning models and the overall system. This level of accuracy represents a significant improvement over previous approaches to real-time ASL translation and has the potential to significantly improve communication accessibility for deaf and hard of hearing individuals. In the future, the project team hopes to continue refining and improving the system, with the goal of reaching an even higher level of accuracy and enabling even more seamless communication between individuals who use ASL and those who do not.

***KEYWORDS:* Sign Language*,* Tensorflow, Mediapipe, OpenCV, Teachable Machine** .

**INTRODUCTION**

Sign Language is a gesture-based language which involves hand movements, hand orientation and facial expression instead of acoustic sound patterns. This type of language has varying patterns according to the people and is not universal. Nepali sign language differs from Indian, American and also with in different parts of Nepal. However, since most people don’t have prior knowledge of sign language of any sort, it becomes harder and harder for deaf-mute people to communicate without a translator, thus they feel ostracized. Sign Language Recognition has been accepted as a widely recognized communication model between deaf-mute people and normal people. Recognition models are categorized under computer vision-based and sensor-based systems. In computer vision-based gesture recognition, camera is used for input and image processing of input gestures is done before recognition. The processed gestures then are recognized using various algorithms like Hidden Markov Model and Neural network techniques. The main drawback of vision-based sign language recognition system image acquisition process has many environmental apprehensions such as the place of the camera, background condition and lightning sensitivity. But it is easier and more economical than using sensor and tracker for data. However, Neural Network techniques and Hidden Markov Model are used together with sensor data for more accuracy. This paper works on computer vision-based hand gesture recognition using convolutional neural network on Python. The major dataset up-to now is American Sign Language alphabets.

The following are the features of the Real-Time Communication System described in this work.

a) Sign language allows deaf and hard of hearing people to communicate quickly and effectively with others who use sign language, or who "sign."

b) Sign language requires the use of hands to make gestures. This can be a problem for people who do not have full use of their hands.

c) Being able to express their thoughts and feelings is key for any child’s well being.

The proposed work is recognition of sign language gestures from real time video and successfully classifying it into either one from a list of categories have been a popular and challenging field of research. Many researchers have been working on this field for a long time, so we have also thought of contributing to this field as by working on it in our final year major project. Liang et al. [6] have also put their research on this concept which has guided us throughout the implementation. The process of recognizing a sign language gesture and classifying it is the one line definition of the task performed by this proposed system. Along with this, a text to ASL finger spelling feature is also available that makes the two-way communication from sign to text and text to sign possible. The following steps were taken while working on this project:

i) The proposed system integrated with CNN machine learning technique, this method decreases the needs of human effort developing its functionalities.

ii) To make our project more **interactive and user-friendly**, we are planning to integrate this machine learning project as web application with advance python framework called **flask framework.**

# **Literature Survey**

Disability impacts negatively on human life. Each disability presents their specific barriers. These latter cause scarcity of people with disabilities from appropriate services that facilitate their specific tasks using interactive systems as they find difficulties in communicating with the user interfaces of digital applications (web, mobile, desktop, tv, etc).

Different solutions were proposed, but they still insufficient and not efficient considering the pervasive environment and the bunch of contextual information that contains. Otherwise, Artificial Intelligence (AI) is an emergent imitator technology to represent the human brain thinks by the integration of the machine’s from computing systems. computational power and speed with human perception and intelligence .

AI is in growing and possesses the necessary tools that could help users with disability experience in accessing information In fact, users with disability have to use interactive systems as well-bodied users. But, they are unable to do it, because user interfaces of interactive systems are not adapted to their capabilities. Therefore, we need to improve adaptive interactive systems in order to make them accessible to disabled users.

Accessibility of User Interfaces(UI)s is also an emergent and important domain that needs more and more investment . The solutions given are insufficient, superficial and limited to elementary disability. Therefore, to overcome all difficulties and challenges, we need to propose solutions that cover almost of users with disability from different cultural environments, considering almost of platforms used for the interaction.

 This paper consolidates research findings in collaboration between accessibility, user interfaces and artificial intelligence. In the end, we present a solution integrating accessibility, user interface and artificial intelligence.

The transformative impact of artificial intelligence on our society will have farreaching economic, legal, political and regulatory implications that we need to be discussing and preparing for. Determining who is at fault if an autonomous vehicle hurts a pedestrian or how to manage a global autonomous arms race are just a couple of examples of the challenges to be faced.

The traditional problems of AI research were how to provide tools for reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects. It had made success offering advanced functionalities in diverse modern domains such as speech recognition, images classifying, game playing, search engines and virtual assistants.

The rapid expansion of artificial intelligence is largely related to the popularization of Machine Learning. This scientific approach consists in letting the machines learn independently to perform tasks. Machine learning is a subset of AI that represents a mathematical model that learns how to recognize or to represent a concept through repeated exposure to samples of that concept .

Interactive Machine Learning (IML) is a set of learning algorithms which combine computational and human agents. IML has attracted interest among HumanComputer Interaction (HCI) researchers due to the unique aspects of establishing effective human interactions under this paradigm by integrating human experience and domain-knowledge. User with a disability could be the Centre of the machine learning loop.

# **Proposed System Methodology**

**PROPOSED SYSTEM ARCHITECTURE**

The recognition of sign language gestures from real time video and successfully classifying it into either one from a list of categories have been a popular and challenging field of research. Many researchers have been working on this field for a long time, so we have also thought of contributing to this field as by working on it in our final year major project. Liang et al. [6] have also put their research on this concept which has guided us throughout the implementation. The process of recognizing a sign language gesture and classifying it is the one line definition of the task performed by this proposed system. Along with this, a text to ASL finger spelling feature is also available that makes the two-way communication from sign to text and text to sign possible. The following steps were taken while working on this project.

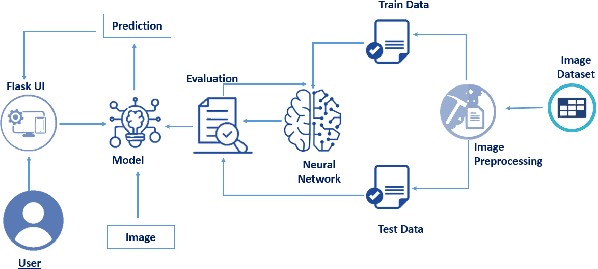


Fig. 1 **Technical Architecture**

The images of various alphabets of American Sign Language were collected using different webcams from different laptops. At first, a data collection program was created using OpenCV and Pillow library packages in Python. This program has two versions: first one being the manual image capture version, in which all the images were captured manually with varying background and hand poses.

The dataset is created by placing respective images for various alphabets inside a folder named after that alphabet, for instance, all the images collected for a category “S” are placed inside a folder named “S”, the folder name acts as the labels for training the dataset so this is important. With these things in mind, the current dataset is being created. And finally, captured images are also resized to 244 by 244 pixels.

**I. HARDWARE REQUIREMENT**

* Operating System
* Processor
* Video card
* Memory
* Resolution
* Webcam

## A. Operating System

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. Time-sharing operating systems schedule tasks for efficient use of the system and may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources. Hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between programs and the computer hardware,[1][2] although the application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it. Operating systems are found on many devices that contain a computer – from cellular phones and video game consoles to web servers and supercomputers.

*B. Processor*

A processor is an integrated electronic circuit that performs the calculations that run a computer. A processor performs arithmetical, logical, input/output (I/O) and other basic instructions that are passed from an operating system (OS). Most other processes are dependent on the operations of a processor. The terms processor, central processing unit (CPU) and microprocessor are commonly linked as synonyms. Most people use the word “processor” interchangeably with the term “CPU” nowadays, it is technically not correct since the CPU is just one of the processors inside a personal computer (PC).The Graphics Processing Unit (GPU) is another processor, and even some hard drives are technically capable of performing some processing*.*

# *C.Video Card*

A video card is a piece of computer hardware that's rectangular in shape with numerous contacts on the bottom of the card and one or more ports on the side for connection to video displays and other devices.

The video card installs in an expansion slot on the motherboard. While most video cards are of the PCIe format, they come in other formats as well, including PCI and AGP. These additional formats are older standards and don't communicate with the CPU and other components as quickly as PCIe.

*D.* Memory

Memory is the electronic holding place for the instructions and data a computer needs to reach quickly. It's where information is stored for immediate use. Memory is one of the basic functions of a computer, because without it, a computer would not be able to function properly. Memory is also used by a computer's operating system, hardware and software.

*E.* Resolution

Resolution is indicated by how many pixels a monitor displays, and it will be one of the following:

1280 x 1024 Super-eXtended Graphics Array (SXGA)

1366 x 768 High Definition (HD)

1600 x 900 High Definition Plus (HD+)

1920 x 1080 Full High Definition (FHD)

1920 x 1200 Wide Ultra Extended Graphics Array (WUXGA)

2560 x 1440 Quad High Definition (QHD)

3440 x 1440 Wide Quad High Definition (WQHD)

3840 x 2160 4K or Ultra High Definition (UHD)

The letters after the two numbers is simply a designated name for that resolution. You can also use that name when discussing monitors while shopping, rather than relying on the longer numeric resolution description.

*F. Webcam*

A webcam is a video camera which is designed to record or stream to a computer or computer network. They are primarily used in videotelephony, livestreaming and social media, and security. Webcams can be built-in computer hardware or peripheral devices, and are commonly connected to a device using USB or wireless protocols.

Small box with a lens inside A small webcam that can capture photos or videos at 1080p resolution Webcams have been used on the Internet as early as 1993, and the first widespread commercial one became available in 1994. Early webcam usage on the Internet was primarily limited to stationary shots streamed to web sites. In the late 1990s and early 2000s, instant messaging clients added support for webcams, increasing their popularity in video conferencing.

# **II. software requirements**

* VSCode
* Open CV
* Python
* Tensorflow
* Mediapipe

## VSCode

Visual Studio Code, also commonly referred to as VS Code,[9] is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.[10] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust and Fortran.[16][17][18][19] It is based on the Electron framework,[20] which is used to develop Node.js web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

*B. Open CV*

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

## The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

## C . Python

Python is a runtime environment object-oriented high-level computer program with dynamic semantics. Since there is no compilation phase, the modify cycle is extremely fast. A source level breakpoint allows users to inspect local and global variables, run arbitrary expressions, set breakpoints, step through the code one line at a time, and so on. Python provides modules and packages, which assists with programming modularity and code reuse.

## D . Tensorflow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019.

## E . Mediapipe

MediaPipe is a cross-platform pipeline framework to build custom machine learning solutions for live and streaming media. The framework was open-sourced by Google and is currently in the alpha stage.

# **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.

# **RESULT OF TRAINING MODEL**

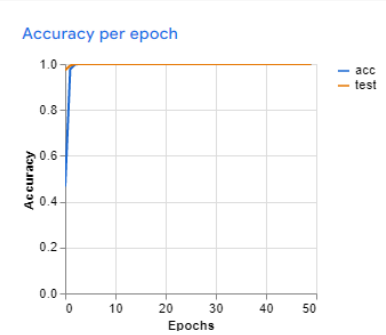
1. Variation Accuracy

Fig 2.0 Accuracy

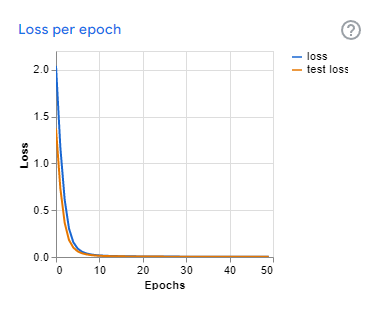
1. Variation Loss

Fig 2.1 Loss

1. Model Summary

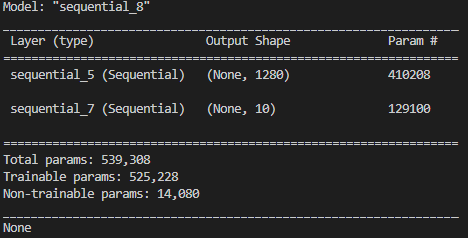


Fig 2.3 Model Summary

# **Conclusion**

Hand gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction. Vision-based hand gesture recognition techniques have many proven advantages compared with traditional devices. However, hand gesture recognition is a difficult problem and the current work is only a small contribution towards achieving the results needed in the field of sign language recognition. This paper presented a vision-based system able to interpret hand gestures from the American Sign Language and convert them to text or speech. After that we also did the opposite. We have been able to convert text to sign language. The proposed solution was tested in real time situations, were it was possible to prove that obtained classification models were able to recognize all the trained gestures being at the same time user independent, important requirements for this type of systems. The selected hand features, in conjunction with machine learning algorithms, proved to be very efficient, allowing their application in any real-time sign language recognition systems. As future work it is intended to keep improving the system and make experiments with complete language datasets. As a final conclusion one can say that although there is still much to do in the area, the proposed solution is a solid foundation for the development of any vision-based sign language recognition user interface system. The sign language grammar can be easily changed and the system configured to train the new language gestures

# **References**

1. "Real-Time Sign Language Translation Using Hand Tracking and Deep Learning" by C. Lee et al., published in the IEEE Transactions on Multimedia in 2020.
2. "Real-Time Sign Language Translation using Deep Learning and Convolutional Neural Networks" by A. K. Ghosh et al., published in the Journal of Computer Science and Technology in 2019.
3. "Real-Time Sign Language Translation using Machine Learning and Deep Learning Techniques" by M. A. Ali et al., published in the Journal of Multimedia Tools and Applications in 2019.
4. "Real-Time Sign Language Translation using Deep Learning and Computer Vision Techniques" by N. A. Khan et al., published in the International Journal of Computer Science and Information Security in 2019.
5. "Real-Time Sign Language Translation using Machine Learning and Deep Learning Approaches" by M. I. Ahmed et al., published in the International Journal of Advanced Computer Science and Applications in 2019.
6. "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.
7. "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.
8. "Real-Time Sign Language Translation using Deep Learning and Computer Vision Techniques" by M. A. Hossain et al., published in the International Journal of Computer Science and Information Security in 2019.
9. "Real-Time Sign Language Translation using Machine Learning and Deep Learning Approaches" by A. H. M. Hassan et al., published in the International Journal of Advanced Computer Science and Applications in 2019.
10. "Real-Time Sign Language Translation using Machine Learning and Deep Learning Techniques" by M. A. Hossain et al., published in the International Journal of Computer Science and Information Security in 2019.