Real-time Sign Language Recognition

A PROJECT REPORT

Submitted in partial fulfillment of the requirement for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Anjali Meena (21103016), Navleen Kaur (21103099)
Rajveer Singh (21103121), Suryamani Patel (21103148)

Under the supervision of

Mr. Rahul Aggarwal
Assistant Professor

Department of Computer Science and Engineering
Dr. B. R. Ambedkar National Institute of Technology Jalandhar
-144008, Punjab (India)

May 2024

Dr. B. R. Ambedkar National Institute of Technology Jalandhar

CANDIDATES' DECLARATION

We hereby certify that the work presented in this minor project report entitled "**Real-time Sign Language Recognition**" in partial fulfillment of the requirement for the award of a Bachelor of Technology degree in Computer Science and Engineering, submitted to the Dr. B R Ambedkar National Institute of Technology, Jalandhar is an authentic record of our own work carried out during the period from July 2023 to May 2024 under the supervision of Mr. Rahul Aggarwal Assistant Professor, Department of Computer Science & Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar.

We have not submitted the matter presented in this report to any other university or institute for the award of any degree or any other purpose.

Date: 30th May, 2024

Submitted by
Anjali Meena (21103016)
Navleen Kaur (21103099)
Rajveer Singh (21103121)
Suryamani Patel (21103148)

This is to certify that the statements submitted by the above candidates are accurate and correct to the best of our knowledge and are further recommended for external evaluation.

Mr. Rahul Aggarwal, Supervisor	Dr. Rajneesh Rani
Assistant Professor	Head and Associate Professor
Deptt. of CSE	Deptt. of CSE

ACKNOWLEDGEMENT

As in any show, many people are hard at work backstage preparing for the final performance to go smoothly, so it is with gratitude to acknowledge that the completion of the "Real-time Sign Language Recognition" project was made possible with the help of many true friends. It would be crucial to note that our team was greatly privileged to obtain this support throughout the course of the project. For this kind of supervision and help we would like to express our heartfelt appreciation as it has been one of our achievements.

To begin with, we are grateful to our project mentor, Mr. Rahul Aggarwal Assistant Professor, who believed in us and in the venture, and offered new ideas when appropriate. One could not have asked for more as he was always supportive and we were able to understudy our challenges.

We also owe special thanks to Dr. Rajneesh Rani, the Head of the Department of Computer Science and Engineering, who directly or indirectly motivated and guided us during this research.

We would therefore like to thank Dr. Aruna Malik, Coordinator Major Projects for availing to us mentors and all the other necessary resources.

We are highly indebted for the constant support and cooperation from all the faculty members of the Department of Computer Science and Engineering. Also, our appreciation goes to all laboratory staff for their useful contribution to this study.

Thank you.

[Anjali Meena, Navleen Kaur, Rajveer Singh, Suryamani Patel]

ABSTRACT

Real-Time Sign Language Recognition is a novel approach to the recognition of sign language presented through a web-based application that applies the concept of deep learning technologies. This project offers a real-life application that can help people who struggle to find professional sign language interpreters due to the geographical location of their communities. The technical specifics of Real-Time Sign Language Recognition is that the product is based on an artificial intelligence architecture that utilizes a CNN embedded with the principles of transfer learning to achieve maximum accuracy and speed of identifying signs.

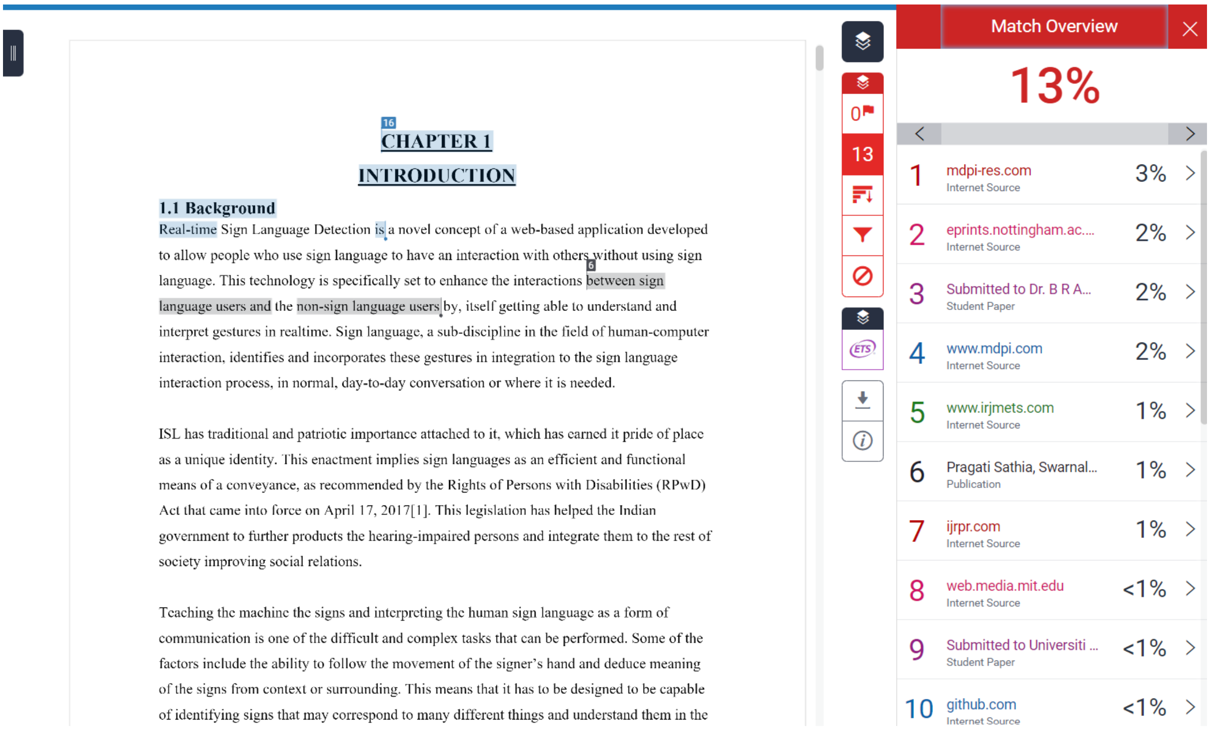
This work precedes from a large hand gesture image set, gathered independently using OpenCV and CVzone hand tracking module. In this dataset, there are different images of hand signs, which go through several steps of data normalization and data augmentation, as the latter provides the model with high-quality images to learn from. Some of the measures that are taken are resizing the images, increasing the quality of the images and applying various techniques such as flips, rotations, and change in light intensity in the images so as to make the model resilient to actual-like variations.

After data preparation, this work applies transfer learning to fine-tune a MobileNetV2 model that has been pretrained. This approach enables the model to build on features learned from a vast array of other images other than sign language imagery and sign language hand gestures; thus promoting generalization from sign imagery. The model training is carried out with extreme care with a particular emphasis on the aspect such as accuracy, F1 and confusion Matrix score which are critical in determining whether the model can accurately classify common signs and any other signs that are unique in nature.

Flask is a popular web framework that accelerates the process of deploying Python applications, and this is the framework that Real-Time Sign's application layer is built upon. This is due to the choice of the python web framework Flask which permits the entwining of back end artificial intelligence methods with a front end graphical user interface whereby users can interact with the tools through a simple web based application. Instead of uploading images, users can utilize their live webcam feed to get immediate predictions with associated confidence scores. This interaction is supported by a backend infrastructure that handles live video data processing, model inference, and result presentation in real time.

PLAGIARISM REPORT

We have checked plagiarism for our Project Report for our project a **Turnitin.** We are thankful to our mentor Mr. Rahul Aggarwal for guiding us at this. Below is the digital receipt. The Plagiarism is approximately 13%.



LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI - Artificial Intelligence

API - Application Programming Interface

CNN - Convolutional Neural Network

CPU - Central Processing Unit

CSS - Cascading Style Sheets

DL - Deep Learning

GPU - Graphical Processing Unit

HTML - HyperText Markup Language

ICT - Information and Communication Technology

ILSVRC - ImageNet Large Scale Visual Recognition Challenge

ML - Machine Learning

PyPI - Python Package Index

TFX - TensorFlow Extended

UML - Unified Modeling Language

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Background

Real-time Sign Language Detection is a novel concept of a web-based application developed to allow people who use sign language to have an interaction with others without using sign language. This technology is specifically set to enhance the interactions between sign language users and the non-sign language users by, itself getting able to understand and interpret gestures in realtime. Sign language, a sub-discipline in the field of human-computer interaction, identifies and incorporates these gestures in integration to the sign language interaction process, in normal, day-to-day conversation or where it is needed.

ISL has traditional and patriotic importance attached to it, which has earned it pride of place as a unique identity. This enactment implies sign languages as an efficient and functional means of a conveyance, as recommended by the Rights of Persons with Disabilities (RPwD) Act that came into force on April 17, 2017[1]. This legislation has helped the Indian government to further products the hearing-impaired persons and integrate them to the rest of society improving social relations.

Teaching the machine the signs and interpreting the human sign language as a form of communication is one of the difficult and complex tasks that can be performed. Some of the factors include the ability to follow the movement of the signer’s hand and deduce meaning of the signs from context or surrounding. This means that it has to be designed to be capable of identifying signs that may correspond to many different things and understand them in the context of a conversation being carried out at that precise moment.

This thus makes sign language users parade a lot of barriers of communication especially in many regions where there are poor healthcare systems or no competent professional sign language Interpreters. In some villages, one may never encounter a single interpreter, especially where Internet connection is a luxury. This can positiviely affect communication and worse case decrease the quality of the communication as well as bring about delays. Quite often, people may not seek the help of an interpreter for some brief encounter leaving them frustrated or unable to convey or receive messages properly.

Real-time Sign Language Detection can be used to tackle these challenges because it does not require people to wait for an interpreter, unlike when using vocal sign languages. The effectiveness is attributed to the ability to use artificial intelligences to pointedly identify and interpret hand gestures from the web camera feeds, making the sign language gesture recognition almost real-time. It helps people to be able to convey messages and ideas across and guarantee that the interaction occurs when required.

Conclusively, the Real-time Sign Language Detection is effective especially for the hearing impaired people especially in areas where they can rarely get someone who understands sign language interpreter. Through integration and application of AI in the area of real time conversation by sign language, this application seeks to improve on the quality of conversation and tend to the needs of the sign language users hence increasing on the level of inclusion of such people.

1.2. Literature Survey

The field of human computer interaction (HCI) is relatively new and has emerged as a key research domain within the last decade especially in sign language interpretation to enhance communication between profoundly hearing impaired friends and their environment primarily using sign language [8]. Two primary methodologies have emerged: Although the research in the field of car tracking can be divided into vision-based and sensor-based techniques [9] .

Vision-Based Methods: These approaches use cameras and depth sensors to map out the gestures and body language and incorporate the use of computer vision algorithms. For instance, Microsoft Kinect utilizes innovative methods of light projection that can capture hand and body gestures. Another approach is the application of markers such as blackened gloves in order to capture hand and body gestures [9] [10] [4] .

Sensor-Based Methods: Up till now, motion based sensing where finger and hand motions are detected using physiological sensors such as the flex sensors and IMUs that are often incorporated in wearables. These sensors give accurate information, mechanical and accelerometer, for hand movement and sign language recognition accurately in sign languages [8] [5].

Commercial Sensor Glove Models: Signifying on commercial data gloves like Cyber-Glove and 5DT data glove, these devices possess multiple sensors with commendable diverse functions of capturing hand movements as well as its orientation. [9] [4] [7] These gloves are different as they utilize neural network based algorithms in order to attain very high recognition rates for sign language.

EMG and Hybrid Methods: EMG sensors alongside inertial sensors can record muscle activity, and offer accurate measurements across multiple axes for hand motion. fusing vision and motion capture sensors provides better gesture recognition than using only one type of sensors evident by the research works with recognition rates of [8], [11], [12], [13], [14], [15] and [16].

Mathematical Models: Many researchers like Thad Starner and Alex Pentland have made enhancement in the ASL recognition system in real time by using Hidden Markov Models (HMM) using different camera sets in order to capture the gestural effects [2] [6] .

Framework and Development: From the recent standardized frameworks, much effort has being put towards enhancing the accuracy and efficiency of sign language recognition. Specifically, wireless systems have incorporated sensors and software architectures to enhance the communication of the hearing-impaired individuals [19] [17] [18].

Innovative Approaches: Dimensions in sign are easily detected with high accuracy due to the use of characteristic markers in images that have been converted to grayscale. Cutting-edge data acquisition techniques include vision-based, glove-based and marker techniques that synchronize with neuro-networks for gesture recognition [3] [4] [5] .

1.3. Problem Statement and its Necessity

The major issues that spurred the development of Real-time Sign Language Detection are as follows:

1. **Few of professional sign language interpreters:** In any developing country there are very few or no sign language interprets available in those remote areas Real-time Sign Language Detection play significant role to cover the communication gap for those peoples who use sign language.

2. **Helping Early Sector Interaction and Connection:** There are many situations where people with hearing problems or the deaf may avoid requesting interpretation services for minor interactions, which costs them important interactions, misunderstandings, and lost opportunities for good communication. Real-time Sign Language Detection can be beneficial as it offers instant help, which means that administrative staff can much better communicate due to this option in the future.

3. **Optimizing Healthcare Translating in Patient Conferences:** Generally, the particular communication that patient-consumer who are sign language users require involves a translator, a process that is complex and takes a lot of time. In Real-time Sign Language Detection, a healthcare provider or a relevant specialist can enter the system to detect gestures to help interpret, and assist a patient, and provide clear communication without the immediate need of an interpreter to be present. This could perhaps be valuable especially in urgent cases when a patient is brought in or during simple clinical visits, where nurses and other doctors get to grasp a few concerns the patient has.

4. **Supporting Local Interpreters and Other Beginners:** In tier 3 or 4 cities, or rural areas of developing countries, there may not be any sign language expert and even if there are, they may not be sufficiently trained In some situations, local healthcare providers or interpreters who are new to this field may be contracted to provide sign language services. Real-time Sign Language Detection can offer these practitioners accurate means of deciphering sign language, aid them in making better decisions concerning patient care as to when to refer such patients to attentive specialized health facilities/ services.

5. **Real-time Sign Language Detection:** Real-time Sign Language Detection enables the hearing-impaired persons to get involved in communication soon after they are in need using other social gatherings, schools, and workplaces. This technology enhances social equality and allows people using sign languages to have equal shots at communication as those who do not have this difficulty.

6. **Improving the Availability of Services:** Public services including government services, transportation centre and customer services agency do not provide sign language interpreters during service deliveries when needed to; making it difficult for the hearing impaired persons to access services. Real-time Sign Language Detection may be implemented in these contexts, where spoken language interpretation may not be sufficient, to enhance interaction while guaranteeing equal opportunities in the receipt of public services.

7. **Prepared for Learning:** Educational facilities and classrooms can benefit from the Realtime Sign Language Detection technology, as children with hearing impediments may require help visualizing lectures and other discussions or presentations. This technology interprets sign languages in real-time, thus applying the technology in classroom observation and learning to help students, especially those who are physically challenged, learn and in turn, be able to engage in an academic discourse fully.

1.4. Motivation

- There seems to be a lack of resources especially for the handicapped such as getting resources on how to deal with barriers to communication of a person in sign language to be expensive and only accessible to a professional interpreter.
- Real-time Sign Language Detection was thena introduced as an application that would help people all over the world especially in the rural areas to make sign language recognition more attainable.
- This implies that the possibility of fast and efficient communication can now be achieved with a gesture camera or webcam where common language interpreters may not be needed.
- Currently implemented with latest technological platforms, Real-time Sign Language Detection works to identify and respond to hand gestures at the moment it is made making it easy to interpret sign languages used in the community.
- This unique approach can be helpful for every person who wants to improve his/her communication with others while not having professional abilities or materials to provide top-quality interpreting assistance.

1.5. Feasibility: Non-Technical and Technical

- **Technical**: Real-time Sign Language Detection benefits from the availability of powerful high-level programming languages like Python, coupled with extensive support for machine learning algorithms and graphical processing units (GPUs) accessible through platforms like Google Colaboratory. These resources enable the training of sophisticated models even on low-end personal computers. Additionally, open-source web frameworks such as Flask streamline the development of web applications, while tools like PlantText facilitate the creation of UML diagrams, aiding in the project's technical planning and implementation.
- **Social:** There is currently a lack of widely adopted or mainstream applications addressing the specific problem domain of real-time sign language recognition. Real-time Sign Language Detection fills this gap by offering a novel solution that enhances communication accessibility for individuals who rely on sign language in various social contexts.
- **Economical-Feasibility:** The development expenses for Real-time Sign Language Detection are expected to be minimal, as the project leverages open-source libraries and publicly available datasets for model training. This cost-effective approach ensures that the project remains financially feasible while delivering valuable functionality to users.
- **Scope:** Real-time Sign Language Detection aims to provide a user-friendly and accessible solution for individuals with hearing impairments, enabling them to communicate effectively in real-time using sign language gestures. By offering immediate interpretation and recognition of sign language, the project enhances communication accessibility in diverse settings, including educational institutions, healthcare facilities, public services, and social interactions.

1.6 Research Objectives

- Real-time Sign Language Detection is designed to develop the ability of normal people to understand sign languages with the help of ICT technology to enhance interactions between normal people and the deaf individuals when it comes to social or even job opportunities.
- This is true since Real-time Sign Language Detection offers direct conception sign language Gestures thus giving the hearing impaired an easier time when it comes to voicing themselves and being active in an interactional process or an activity.
- Real-time sign language detection has the capacity of modifying education by offering a greater equality in academic success as well as people who are deaf and dumb can get

to see and understand what is going on in the class alongside being able to respond in real time.

- Real-time Sign Language Detection for Independence creates better communication within healthcare sectors through sign language interpreter services thus making the patient's needs and concerns easily understandable by the service provider hence increasing quality healthcare services and positively affecting the patient's health.
- Exclusively on the interpersonal level: Real-time Sign Language Detection also has social ramifications for communities and organizations in global societies and culture by increasing diverse inclusion and accommodating people with disabilities.

CHAPTER 2

PROPOSED SOLUTION

Real-time Sign Language Detection is a complex development of an online application designed to apply the most effective deep learning techniques for sign language gesture recognition. In the visually impaired sector, this application would help the users to gain interpretations for sign language communication needs at a faster pace. There exist highly sophisticated algorithms to identify and categorize many sign language movements with a significant level of efficiency, and the accomplishments of the application are on the same level, as it provides the interpretations to the users and informs them about the results immediately.

Real-time Sign Language Detection is therefore a process that goes through several phases aimed at arriving at the final predictive model which is followed by the final step of data collection from various sources. This data is next pre-processed to eliminate noise from the data and equalize size of the hand gesture as well as ensuring uniformity of the data collected.

Afterward, the deep learning algorithm is next trained on the pre-processed data set. This involves the ability to use neural network architectures to capture all the fine details of the signs and the motions involved in the sign language, which in turn enhances the development of a reliable classification model. Error-checkingactivities are conducted to verify the model performance and optimize its capabilities during the model development process.

At the end of this training, the model is deployed on to the Real-time Sign Language Detection web application. . it can be used in case the user signs in front of the camera or webcam, which is the way of interaction with the application using sign language gestures. The application uses the trained model in the interpretation of the gestures, giving an immediate response and definition of the interpreted sign language. This capability enables socially benefited and efficient communication for people using sign language to communicate.

As a result, Real-time Sign Language Detection is certainly a source of hope as it helps in enhancing the signs intelligence in order to enhance accessibility for individuals who make use of sign language. Through the provision of efficient, precise, and easily accessible sign language gestures' interpretation, the application enables the users to participate in live interpersonal interactions to their potential, thus, eradicating barriers and promoting mutual respect in numerous sociopolitical and business environments.

We will first provide an overview of the project and then proceed to explain the solution in a step-by-step manner.

- The dashboard of the application activates the webcam and provides the space where the user is supposed to display the signs.

Figure 2.1 Landing view of Sign Language Recognizer

- Once the hands are detected the application detects the skeletal structure of the hands and the proceed to display the label under the web cam feed.

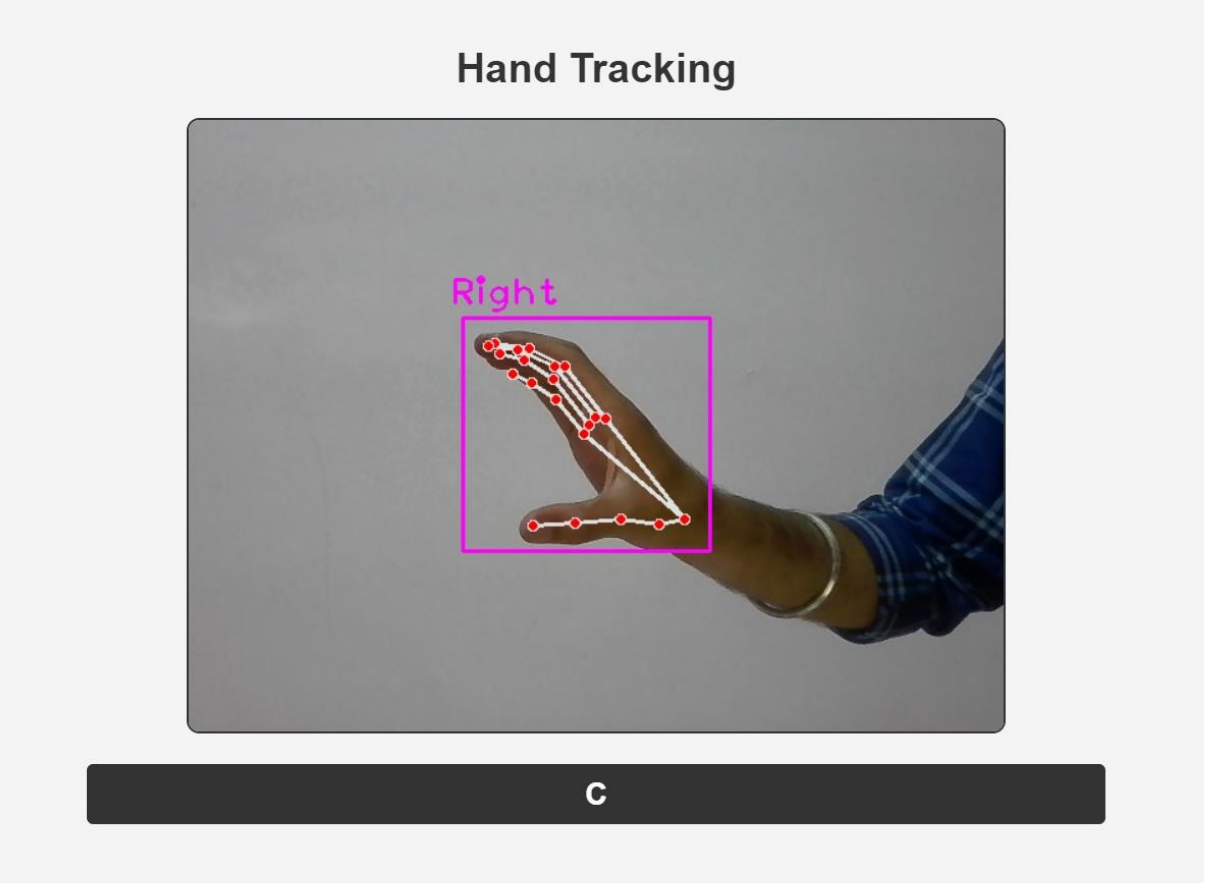


Figure 2.2 View After Recognizing Sign Language

As specified earlier, transfer learning was used to train the model and MobileNetV2 is the pre-trained model that is custom trained.

The following was the procedure followed to train the model:

1. **Data Gathering:** The Data For the model was gathered manually using OpenCV and CVZone Hand tracking model. Data for each individual letter was collected with 300 images for each of the letters.

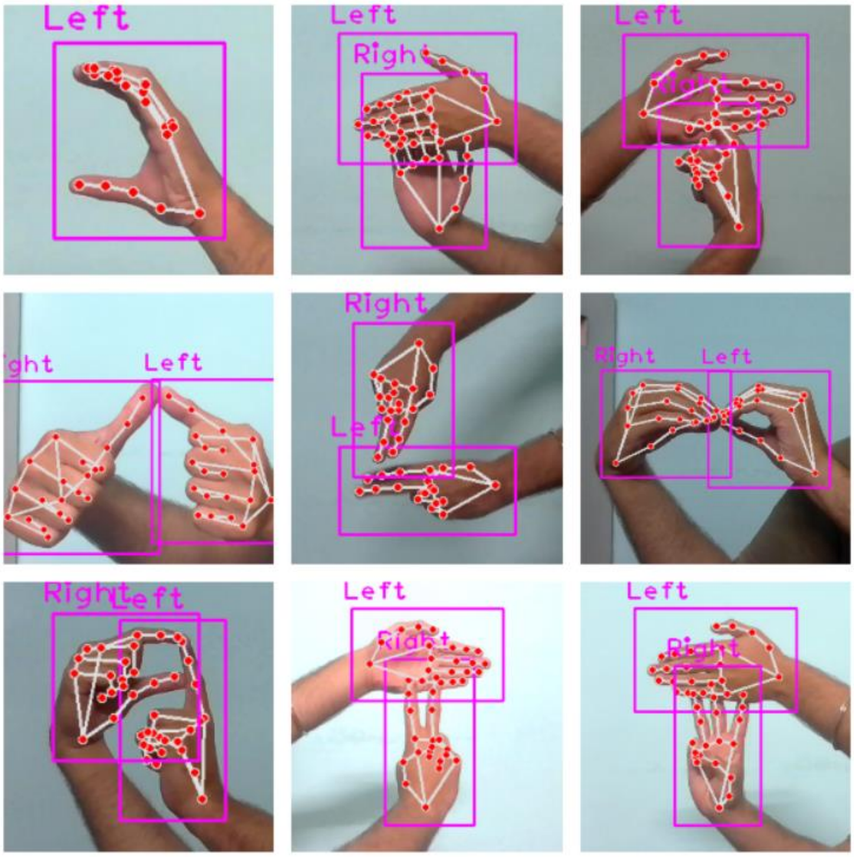


Figure 2.3 Data Collection

The below shown flow diagram specifies the steps which were involved in the collection of data for each individual letter. And the steps involved are:

- The process begins at the "START" point.
- The camera is initialized, and its display is shown.
- The system uses a pretrained cvzone hand detector to analyze the camera feed and detect hands.
- Decision point: The system checks if any hands are detected in the camera feed.
    - Yes: If hands are detected, the flow proceeds to the next step.
    - No: If no hands are detected, the system loops back to continue displaying the camera feed and detecting hands.
- When hands are detected, the captured image is stored in a dedicated folder specific to the letter being recorded.
- Decision point: The system checks if 300 images have been collected.
    - Yes: If 300 images have been collected, the process proceeds to the end.
    - No: If fewer than 300 images have been collected, the system loops back to continue detecting and storing hand images.
- The process ends once 300 images have been successfully collected and stored.

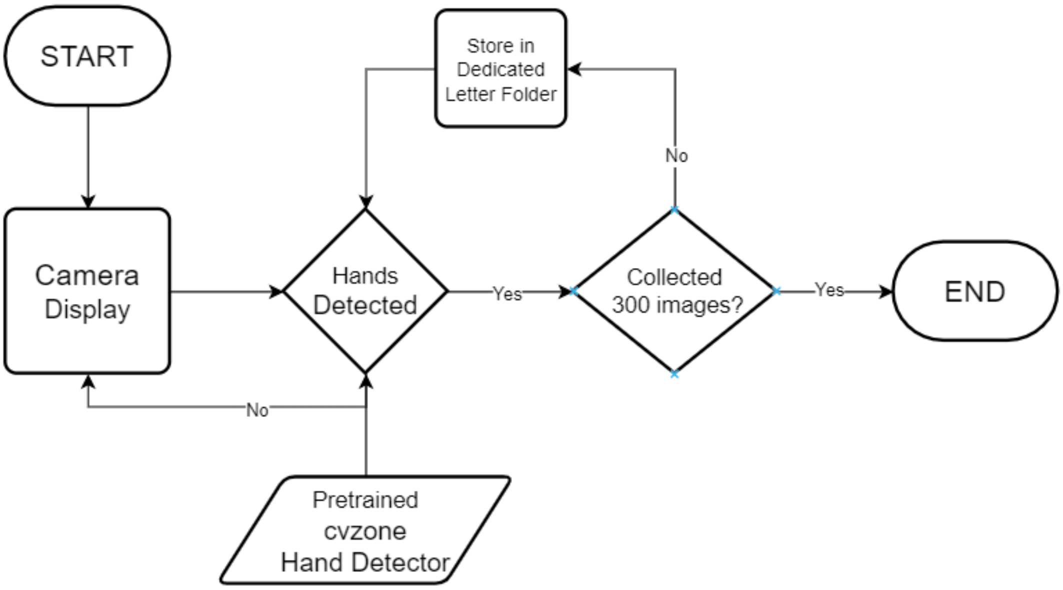


Figure 2.4 Flow Diagram of Data Collection

2. **Data Labeling:** After collecting the data and performing the next step was to label the data so that correct label is recognized while detecting.

```
labels = ["A","B","C","D","E","F","G","H","I","J","K","L","M",
          "N","O","P","Q","R","S","T","U","V","W","X","Y","Z"]
```

Figure 2.5 Data Labels

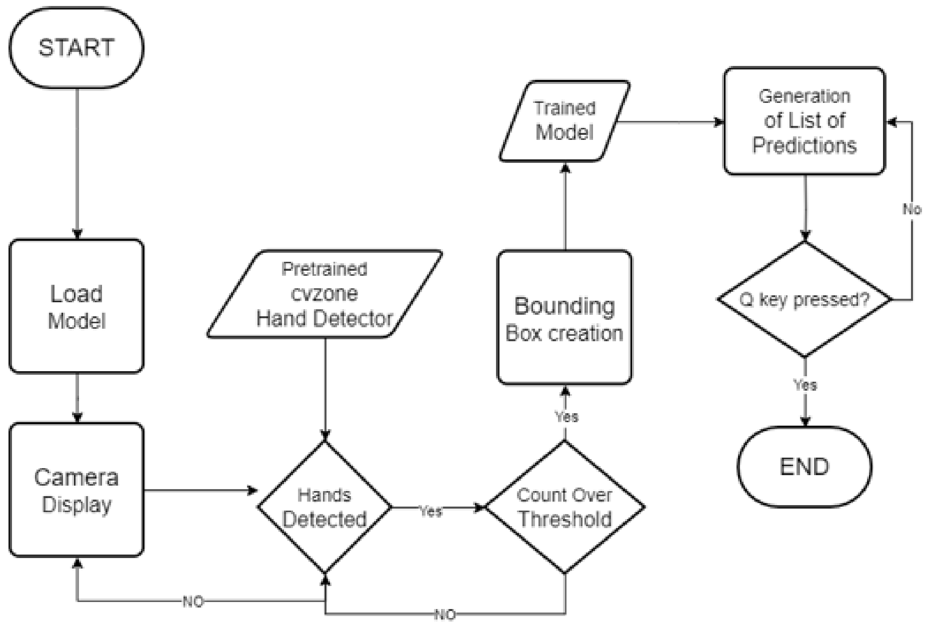
3. **Model Training:** Next Step involved the model training using the collected and labelled data.

The below shown flow diagram specifies the which involves training the model and testing it, and finally running it in real time. And the steps involved are:

- The process begins at the "START" point.
- The system loads the trained model that will be used for making predictions.
- The camera is initialized, and its display is shown.
- The system uses a pretrained cvzone hand detector to analyze the camera feed and detect hands.
- Decision point: The system checks if any hands are detected in the camera feed.
  - Yes: If hands are detected, the flow proceeds to the next step.
  - No: If no hands are detected, the system loops back to continue displaying the camera feed and detecting hands.
- Decision point: The system checks if the count of detected hands is over a certain threshold.

- Yes: If the count is over the threshold, the flow proceeds to the next step.
- No: If the count is not over the threshold, the system loops back to continue detecting hands.

- Once the count of detected hands is over the threshold, bounding boxes are created around the detected hands for further processing.
- The bounding boxes are passed to the trained model to make predictions.
- The trained model generates a list of predictions based on the detected hands within the bounding boxes.
- Decision point: The system checks if the "Q" key has been pressed.
  - Yes: If the "Q" key is pressed, the process ends.
  - No: If the "Q" key is not pressed, the system loops back to continue generating predictions.
- The process ends once the "Q" key is pressed.



`

Figure 2.6 Flow Diagram of Model training, testing and real time running.

4. **Model Evaluation:** The model needs to be evaluated based on evaluation metrics to analyze the correctness and reliability of the model.

   To get the evaluation of the model, following metrics were chosen:

- **Accuracy Graph:** The accuracy plot also given at the end of the particular training epochs or iterations. As such, a graph illustrating the increase, or decrease in the accuracy level of the model during training, is illustrated as follows; The above diagram indicates how the model is handling the information and if there is some indication that they are learning.

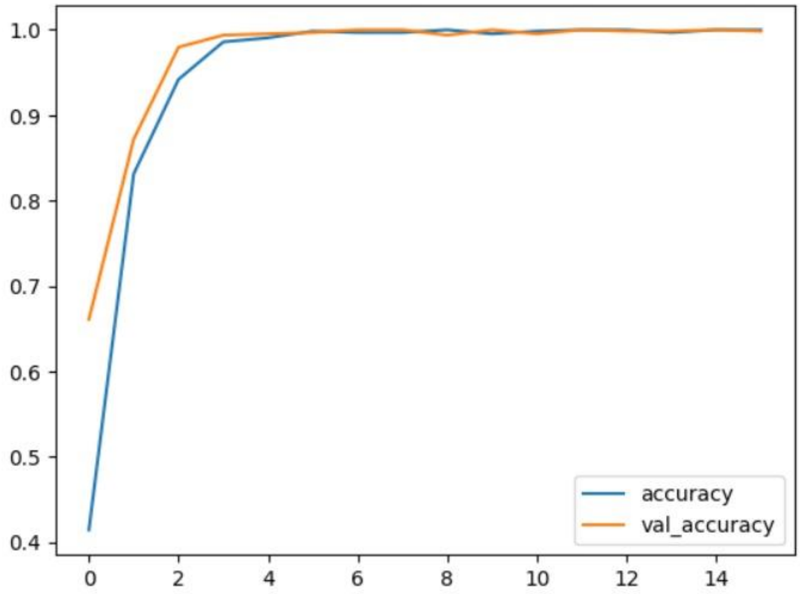


Figure 2.7 Accuracy and Validation Accuracy Graph

- **Loss Graph:** It is mandatory to graph and explain the loss curve down to the training epochs or iteration. Based on the loss graph, disparity which is calculated by subtracting the true labels from the model prognosis can be seen. The output of the loss function represents how well the model is performing in terms of making predictions, and therefore, a positive slope to the curve suggests that the model is becomaing worse, and a negative slope would mean that the model has a better chance of making better predictions.

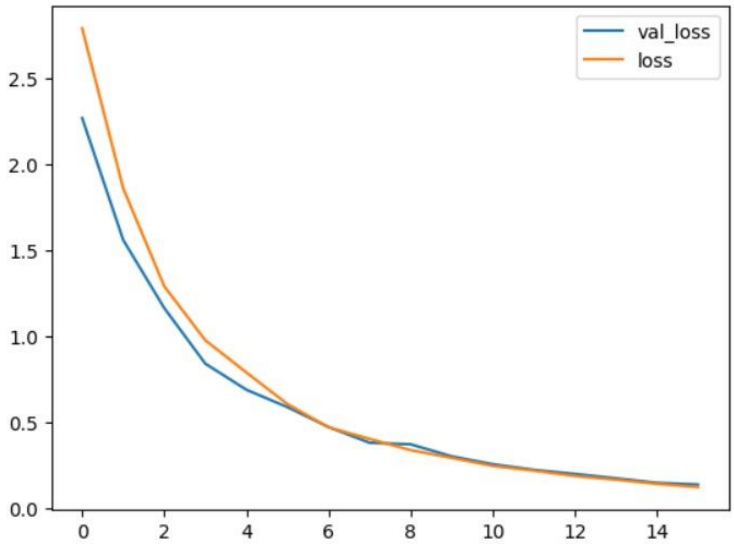


Figure 2.8 Loss and Validation Loss Graph

- **Confusion Matrix:** Next, display the confusion matrix – this is actually a 2 by 2 table that provides detail information on the classification model. It helps in finding out actual positives, actual negatives, over predicted values, under predicted or missed out values for each the classes of the given data. According to the confusion matrix, it is easy to get to know that where the model is wrong and which classes are being mostly mis-classified by the model during the prediction.

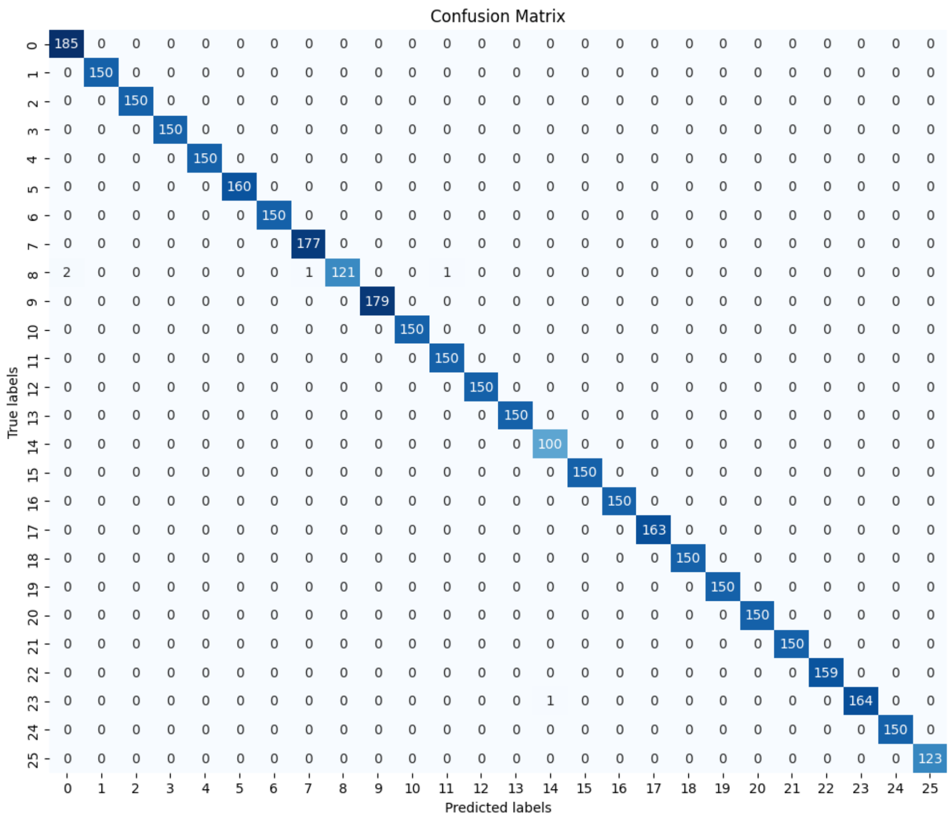


Figure 2.9 Confusion Matrix

5. **Hosting:** Hosting of the machine learning model means placing the model onto a web page where people can access easily. For this task, the Flask framework implemented in the Python programming language was selected. This is made easier by Flask which supports the integration of the model into a web application and allows developers to define API endpoints that will be used by the users interact with the application and create a GUI for the application. Once configured and at the end of the implementation process, the application is published to a web server enabling users to access it. Monitoring and support continue the process of making certain that the application runs efficiently once it has been put into use.

CHAPTER 3
TECHNOLOGY ANALYSIS

3.1. Tech Stack Analysis

The technologies used in the project are:-

3.1.1. Flask

Flask is a micro web framework for Python, hence it is also highly extensible and can be used to develop simple and complex web applications without much stress. ‘Flask’ is an HTTP server framework developed by Armin Ronacher which belongs to “Pallets Projects”. Flask is designed to be minimalistic a developer only gets only the tools necessary to build a web app such as Routing and Request handling. This approach provides a simpler environment for the application developers to manage configurations for their software and flexible installation of only necessary features like database management, form validation or license control. Overall, despite the relative simplicity of Flask, it can accommodate a modular architecture and is useful for both small-scale and complex applications. The fact that Flask is well documented, has a very active developer community and allows using almost all conceivable extensions makes it a perfect choice for creating complex web applications with high scalability, that won’t take much time to code.

Figure 3.1 Flask Web Development

3.1.2. Google Colab

Google Colab, or Colaboratory, is an online interactive environment for Jupyter notebooks on Google Drive that is provided by Google. It ‘s designed to help users with no setup skills write and run Python programs in an online environment. A notable characteristic is the vendor’s provision of free credits to various tough computing resources, such as GPUs and TPUs, even at the fundamental level of the free tier. In this regard, Google Colab is most beneficial for ML and DL because, in most cases, these processes necessitate a considerable amount of working power for model training. These resources can be harnessed in order to train large complicated Neural networks and other forms of ML much more quickly than on standard CPUs, while not requiring any extra expensive purchases in the process. Further, it is compatible with Google Drive, making it easy to save, share and even upload projects from the Google drive Moreover, it has support to most used and highly reputable machine learning libraries like TensorFlow, PyTorch, and Keras. This availability and access make Google colab one of the most used tools among the educators, researchers, and developers in AI data communities.

Figure 3.2 Google Colaboratory

3.1.3. Python

Python is one of the most used versatile high-level programming languages that is most famous for its simple and well-structured brief coding approach. Python is a language developed by Guido van Rossum in 1991 and main emphasis of python is on code readability and it gives programmer liberty to express himself in minimum number of lines of code than compared to other languages due to its smooth language structure. Because of the versatility of procedural, object oriented and functional paradigms and due to the use of several paradigms making it favorable in most types of software development.

The last interesting is the fact that python has a large number of standard libraries as well of libraries to be installed from the PyPI repository. That versatile we can harness different libraries and frameworks to perform a number of operations – web development with Django, Flask; data visualization with Pandas and NumPy, artificial intelligence and machine learning with TensorFlow and scikit-learn.

Python also enjoys such support within the community and also well detailed documentation that makes it rather easy to learn the language and to advance in it. It supports multiple Sectors like research, finance, automation, education, & data science &AI support interoperability with other Languages, build times, & numerical computing support among others. The reasons for its widespread popularity are two: the relative newbies can easily learn how to write code in

Python and use it, and not novices do not want to bother learning the language of working with this tool.

Figure 3.3 Python

3.1.4. TensorFlow

TensorFlow is a machine learning framework which was developed by the Google Brain team and can be used for Artificial Intelligence. It was published in 2015 and allows for the construction and training of sophisticated neural networks; it can also be implemented in CPUs, GPUs, and TPUs. TensorFlow comes equipped with both Keras, which is easy to use and convenient for quick prototyping, and raw APIs if more fine-grained control is required. Its architecture is derived from data flow graphs that support large scale parallel computation. TensorFlow has many applications in research and industry with such derivatives like TensorFlow Lite for mobile deployment, TFX for production pipelines, and TensorBoard visualization tools.

Figure 3.4 Keras & Tensorflow

3.1.5. OpenCV

OpenCV or Open Source Computer Vision Library is an open source computer vision and machine learning software library that originates from the Intel Laboratories. Initially released from Intel in 1999 with Intel Pentium III Campus Implementation the specification is at present governed by the community under the BSD license it is now freely available for both academic and commercial purposes. The toolkit also offers various tools for applications such as object detection, facial recognition, image sectioning, and moving object tracking in several programming languages such as C ++, Python, Java, and MATLAB. It contains more than 2500 algorithms with variable speed and can be used with TensorFlow and PyTorch. Open source and applicable across Windows, Linux, macOS, Android, and iOS, the libraries are utilized in fields such as drone application in robotic vision systems, augmented reality, self-driving cars, and in academia, with abundant developer backing and comprehensive documentation.

Figure 3.5 OpenCV

3.1.6. HTML & CSS

HTML (Hypertext Markup Language) is essential to web development as its role is to define contents and format of web pages by using diverse tags and elements. Structuring can be defined as the determination of the area that a particular piece of information will occupy and most importantly gives that information the semblance of being either easily retrievable or of having some specific form of meaning. Moreover, HTML is a markup language that provides structure, text, and the hypertext links; In contrast, CSS is used to expand HTML where style and appearances are added, for instance, color, fonts and arrangement. CSS makes it easier to manage the graphical user interface of the websites hence makes it easier to apply and maintain the format where needed. HTML and CSS are two fundamental technologies that are at the heart of designing and creating graphic and sessioned webpage on the World Wide Web.

Figure 3.6 HTML & CSS

CHAPTER 4

ECONOMIC ANALYSIS

All the apps and platforms that we have included in our solutions were developed based on free and secure technologies, including APIs, datasets, dependencies. All of these are open source, it basically requires support and the features to adjust to any changes. Therefore, we will be able to guarantee that the use of our application does not result in any expenses, and all the details necessary for the program's functioning require no payment.

- Our goal here is to develop adequate solutions that are affordable, easy to use, and would possess basic functionalities capable of solving most problems.
- The following development stacks which we use are open source and requires internet connectively.
- In terms of free platforms, there is Google Colab which offers a free tier GPU for training your model.
- We particularly chose google colab for the free tier GPU and VS Code for coding the entire rest of the project.
- We designed this plan in such a way that we created a real-time sign language recognition system with zero cost where the models are trained on google colab and the development environment is VS code.

CHAPTER 5

RESULT AND DISCUSSION

5.1. App Usage Instructions

Hardware and software requirements:

- To ensure compatibility, users can access the real-time sign language recognition web application using any laptop with internet connectivity. Upon opening the application, users will encounter a user-friendly interface featuring a live webcam feed. Initially inactive, the webcam becomes operational when users display their hands to the camera.

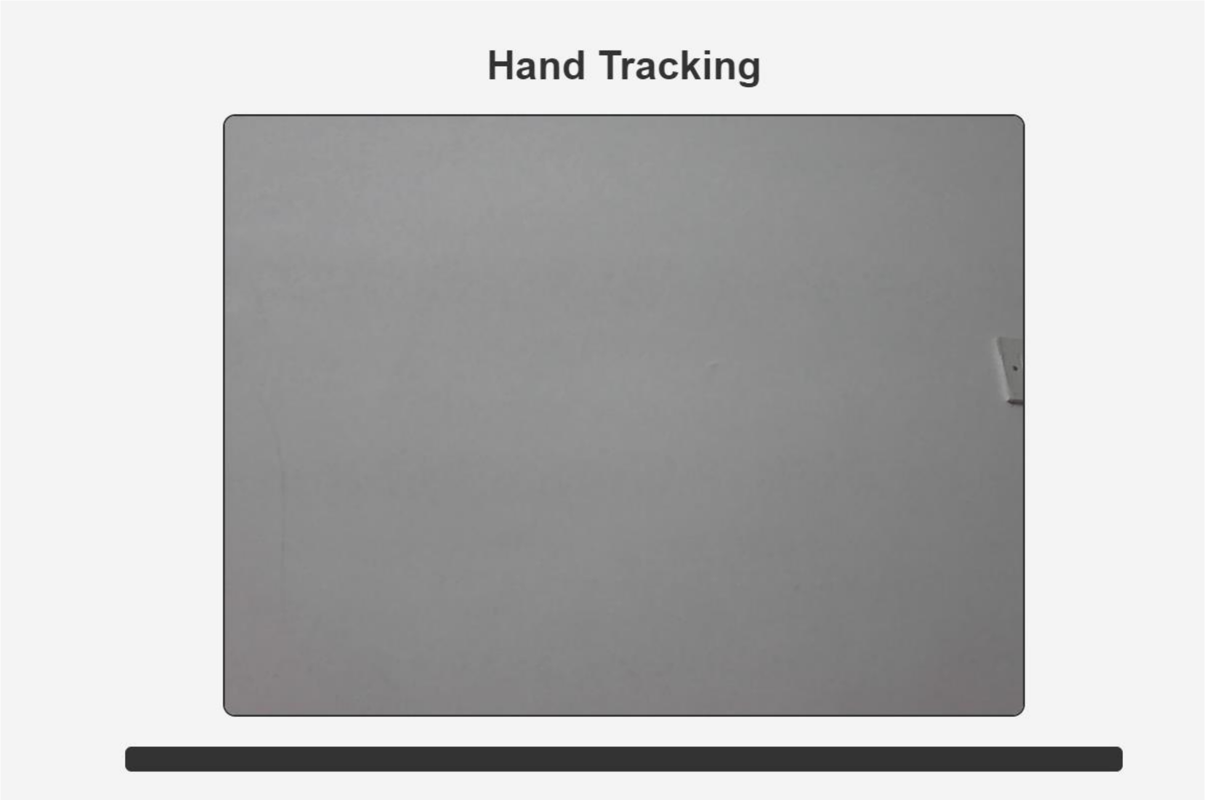


Figure 5.1 Landing view of Sign Language Recognizer

- At this point, the model makes predictions based on the displayed sign language gestures. Subsequently, the predictions are promptly displayed below the webcam feed, facilitating seamless communication for individuals using sign language. This intuitive design prioritizes accessibility and ease of use, aligning with the application's goal of enhancing communication for users worldwide.

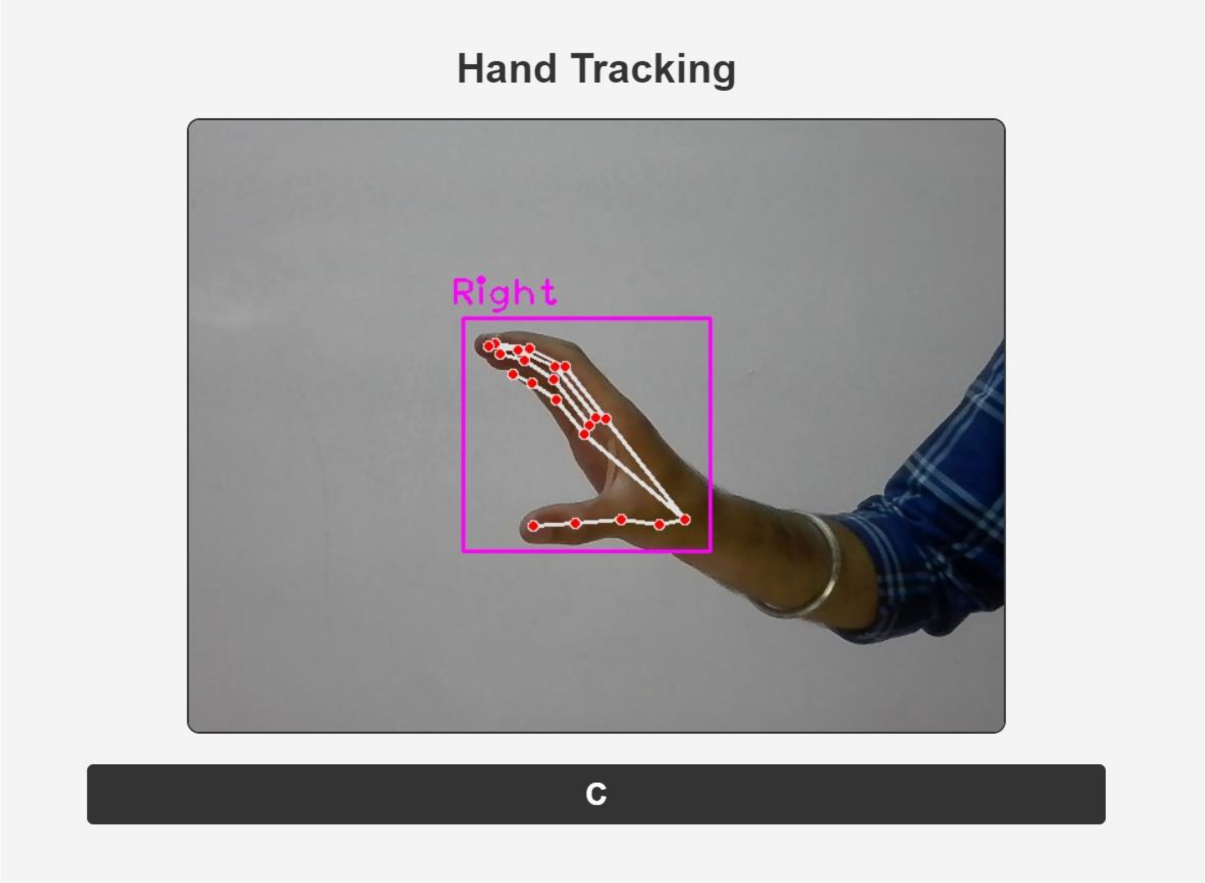


Figure 5.2 View After Recognizing Sign Language

5.2. Current Limitations

1. **Background Dependency:** This model again is limited by the backgrounds which are required to be clear and in certain scenarios this could pose a hindrance in practical implementation.

2. **Model Complexity:** The easy form of the first model suggests that, while it will be faster in terms of computational processing, it may not be as precise. , however, using more sophisticated models may lead to increased computational demands and time necessary to perform the training stage.

3. **Custom Dataset Requirement:** The requirement for a dataset collected and annotated for this purpose also presents problems in the data gathering and labeling. Therefore, choosing appropriate methods for data augmentation, to guarantee both the quality and variability of datasets needed to avoid severe restricts linked to the scarcity or bias of the data, is crucial.

5.3 Future Scope

1. **Enhanced Model Complexity:** This is because the current model, based on its design, is relatively simple and require sonly few changes to be implemented. Proposing denser models may help increase the level of precision and resistance in sign language recognition in a real-world application.

2. **Advanced Background Handling:** A major problem with the current model is its inability to operate accurately on images that display complex backgrounds. Future developments can concern with the efforts to explore methods of reducing affect of background clutter on the model performance.

3. **Dataset Augmentation and Quality Enhancement:** In-built datasets can be generated as per need, which are the basis for training the model. But, future works, associated with the current work, should focus on ways of increasing the size and quality of the dataset that is used for training and testing the model so as to enhance the efficiency of the designed model in enhancement of the deep learning model.

5.4 Risk Analysis

Every project that is designed and developed is bound to have some risks, but it is with the right strategies of identifying, analyzing, and preventing the risks, their impacts can be avoided.

1. **Stakeholder Acceptance:** They include the likelihood of stakeholders not embracing the system fully due to various factors. To address this risk, there have been numerous consultation meetings with the technical/non-technical stakeholders so as to agree on the scope/requirements of the project at every development phase.
2. **Application Crashes Due to High Load:** One more possible issue that can be faced is the application failure in case of high load of users. But, this risk can be managed properly by the right scalability measures like when the application is deployed on a cloud platform where scalability issues can be handled easily.

Additional Risks:

1. **Data Security Breaches:** That is why data security breaches may occur which would lead to breach of user privacy and confidentiality. This risk may, however, be managed through sound security practices like use of encryption and restriction of access.
2. **Model Performance Degradation:** Therefore, there is always the danger that the machine learning model is no longer optimal in its predictions due to changes in data distribution or environment. This risk can be managed efficiently through constant supervision and model recalibration at regular intervals.

CHAPTER 6

CONCLUSION

The absence of sign language interpreters to signing impairments is a stagnation in communication for those with hearing lost. Real-time sign language detection considered to offer practical sign language recognition to promote more availability and applicability of the sign language in daily life, educational, and healthcare settings and otherwise. But first of all, it is crucial to mention that Real-time Sign Language Detection is not intended to replace human interpreters but be a helpful addition that may support them in their work or give preliminary data. This is why the human expert and his acute distinction of a specific moment compared to the automated detection algorithms should always be preferred, because no machine can replace a person and especially in such sensitive and context-dependent processes as communication.

REFERENCES

[1] http://timesofindia.indiatimes.com/home/education/news/realising-the-importance-of-indian-sign-language-dictionary/articleshow/86221166.cms

[2] T. Starner and A. Pentland, "Real-time American sign language recognition from video using hidden markov models", Technical Report, M.I.T Media Laboratory Perceptual Computing Section, Technical Report No.375, 1995.I.

[3] Rajam, Subha & Ganesan, Balakrishnan. (2011). Real time Indian Sign Language Recognition System to aid deaf-dumb people. International Conference on Communication Technology Proceedings, ICCT. 10.1109/ICCT.2011.6157974.

[4] Noor Adnan Ibraheem, RafiqulZaman Khan, 2012, Survey on Various Gesture Recognition Technologies and Techniques, International Journal of Computer Applications, Volume 50, No. 7

[5] Akash Deep, Litoriya, Ingole, Asare, M Bhosle, Dr. Pathak ,2022, "Realtime Sign Language Detection and Recognition. 2nd Asian Conference on Innovation in Technology(ASIANCON) ,10.1109/ASIANCON55314.2022.9908995

[6] Thad Starner and Alex Pentland Room E15-383. The Media Laboratory Massachusetts Institute of Technology 20 Ames Street, Cambridge M A 02139 thad, sandy@media.mit.edu

[7] Chong, T.W.; Lee, B.G. American sign language recognition using leap motion controller with machine learning approach. Sensors 2018, 18, 3554.

[8] Appenrodt, J.; Al-hamadi, A.; Elmezain, M.; Michaelis, B. Data gathering for gesture recognition systems based on single color-, stereo color- and thermal cameras. Int. J. Signal Process. Image Process. Pattern Recognit.2010, 3, 37–50.

[9] Lee, B.G.; Lee, S.M. Smart wearable hand device for sign language interpretation system with sensors fusion. IEEE Sens. J. 2018, 18, 1224–1232.

[10] Kinect for Windows. Available online https://developer.microsoft.com/en-us/windows/kinect (accessed on 1 October 2020).

[11] Molchanov, P.; Gupta, S.; Kim, K.; Pulli, K. Multi-sensor system for driver's hand-gesture recognition. In Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, Slovenia, 4–8 May 2015; pp. 1–8.

[12] Sharma, V.; Kumar, V.; Masaguppi, S.; Suma, M.; Ambika, D. Virtual Talk for Deaf, Mute, Blind and Normal Humans. In Proceedings of the 2013 Texas Instruments India Educators' Conference (TIIEC), Bangalore, India, 4–6 April 2013.

[13] Abdulla, D.; Abdulla, S.; Manaf, R.; Jarndal, A. Design and implementation of a sign-to-speech/text system for deaf and dumb people. In Proceedings of the 2016 5th International Conference on Electronic Devices, Systems and Applications, Ras AL Khaimah, United Arab Emirates, 6–8 December 2016.

[14] Kadam, K.; Ganu, R.; Bhosekar, A.; Joshi, S. American sign language interpreter. In Proceedings of the 2012 IEEE Fourth International Conference on Technology for Education (T4E), Hyderabad, India, 18–20 July 2012.

[15] Ibarguren, A.; Maurtua, I.; Sierra, B. Layered architecture for real- time sign recognition. Int. J. Comput. 2009, 53, 1169–1183.

[16] Munib, Q.; Habeeb, M.; Takruri, B.; Al-Malik, H. American sign language (ASL) recognition based on Hough transform and neural networks. Exp. Syst. Appl. 2007, 32, 24–37.

[17] Shriwas, N.V. A Preview Paper on Hand Talk Glove. Int. J. Res. Appl. Sci. Eng. Technol. 2015. Available online: https://www.ijraset.com/fileserve.php?FID=2179 (accessed on 21December 2021).

[18] Al-Ahdal, M.; Nooritawati, M. Review in sign language recognition systems. In Proceedings of the 2012 IEEE Symposium on Computers & Informatics (ISCI), Penang, Malaysia, 18–20 March 2012.

[19] Verma, S.K. HANDTALK: Interpreter for the Differently Abled: AReview. IJIRCT 2016, 1, 4.