VAANI: Indian Sign Language Translator

A Dissertation submitted to
Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal
Towards Partial Fulfillment of the degree of
Bachelor of Technology in
Computer Science and Engineering

Project Guide:
Ms. Neha Mehra
Assistant Professor
Department of Computer Engg.

Project Co-Guide:
Ms. Ritambara Patidar
Assistant Professor
Department of Computer Engg.

Submitted by:
Group 6:
0801CS201011 – Aryan Raj Shrivastav
0801CS201037 – Harsh Bishnoi
0801CS201049 – Keshav Yadav
0801CS201066 – Prabal Pophaley
0801CS201092 – Somya Agrawal
Department of Computer Engg.

DEPARTMENT OF COMPUTER ENGINEERING
SHRI G.S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE (M.P.)
2023-2024

SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE (M.P)

A Govt. Aided Autonomous Institute, Affiliated to RGPV, Bhopal

DEPARTMENT OF COMPUTER ENGINEERING

2023-2024

RECOMMENDATION

We are pleased to recommend that the dissertation work entitled ***"Vaani:** Indian Sign Language Translator"* submitted by: **0801CS201011 – Aryan Raj Shrivastav, 0801CS201037 – Harsh Bishnoi, 0801CS201049 – Keshav Yadav, 0801CS201066 – Prabal Pophaley, 0801CS201092 – Somya Agrawal**, students of B.Tech. IV year, may be accepted in the partial fulfillment of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (M.P.) during the session **2023-2024**.

Neha Mehra

Ms. Neha Mehra
Project Guide
Department of Computer Engg.

Dr. Vandan Tewari
Head of Dept.
Department of Computer Engg.

Ms. Ritambara Patidar
Project Co-guide
Department of Computer Engg.

Dean(Academics)
S.G.S.I.T.S. Indore

SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE (M.P)

A Govt. Aided Autonomous Institute, Affiliated to RGPV, Bhopal

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled ***"Vaani:*** *Indian Sign Language Translator"* submitted by: **0801CS201011 – Aryan Raj Shrivastav, 0801CS201037 – Harsh Bishnoi, 0801CS201049 – Keshav Yadav, 0801CS201066 – Prabal Pophaley, 0801CS201092 – Somya Agrawal,** students of B.Tech. IVth year, is submitted in the partial fulfillment of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of Rajiv Gandhi Proudyogiki VishwaVidhyalaya, Bhopal during the session 2023-2024

Internal Examiner **External Examiner**

Date: **Date:**

SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE (M.P)

DECLARATION

We, **0801CS201011 – Aryan Raj Shrivastav, 0801CS201037 – Harsh Bishnoi, 0801CS201049 – Keshav Yadav, 0801CS201066 – Prabal Pophaley, 0801CS201092 – Somya Agrawal,** hereby declare that the dissertation "**Vaani:** Indian Sign Language Translator" is our own work conducted under the supervision of **Ms. Neha Mehra,** Project Guide, Department of Computer Engineering and **Ms. Ritambara Patidar,** Project Co-guide, Department of Computer Engineering.

We further declare that to the best of our knowledge, this dissertation work does not contain any part of any work which has been submitted for the award of any degree or any other work either in this university or in any other University/ Websites without proper citation.

Date:

0801CS201011 - Aryan Raj Shrivastav
0801CS201037 - Harsh Bishnoi
0801CS201049 - Keshav Yadav
0801CS201066 - Prabal Pophaley
0801CS201092 – Somya Agrawal

ACKNOWLEDGEMENT

We express our profound sense of gratitude to our project guide **Ms. Neha Mehra** and co-guide **Ms. Ritambara Patidar** who had advised us to do this project work entitled **"Vaani:** Indian Sign Language Translator**"**. Their continuous support and motivation always made us deliver our best. Presence of such guides have always been an amazing experience, which is also a valuable gift for an engineer to progress in his/her life.

We are also grateful to **Dr. Vandan Tewari**, Head, Department of Computer Engineering and **Prof. Rakesh Saxena**, Director, S.G.S.I.T.S Indore, for providing us with numerous facilities and academic environment during the course of study.

We sincerely wish to express our gratitude to all the members of staff of Department of Computer Engineering, S.G.S.I.T.S Indore, who have extended their cooperation at all times and have contributed in their own way in developing the project.

The successful completion of the project is not an individual effort. It is an outcome of the cumulative effort of a number of people, each having their own importance to the objective. We express love and respect towards our parents and the entire family member who are our strength in everything we do.

With a blend of gratitude, pleasure and great satisfaction we convey our indebtedness to all those who have directly or indirectly contributed to the successful completion of our project work.

0801CS201011 - Aryan Raj Shrivastav _____

0801CS201037 - Harsh Bishnoi _____

0801CS201049 - Keshav Yadav _____

0801CS201066 - Prabal Pophaley _____

0801CS201092 - Somya Agrawal _____

ABSTRACT

Indian Sign Language (ISL) stands as a visual language indispensable for non-verbal individuals, facilitating communication through hand gestures, facial expressions, and body movements. Serving as their primary mode of interaction, ISL plays a pivotal role in bridging the communication gap within the non-verbal community and with verbal individuals.
"Vaani," an innovative project, leverages advanced technologies to bridge this communication divide between ISL users and others. Developed using Python and harnessing sophisticated deep learning frameworks like Keras and TensorFlow, Vaani employs a real-time camera-based system to capture and interpret ISL hand gestures. Integration of computer vision libraries such as Mediapipe and cvzone enables precise hand keypoint detection, ensuring accurate recognition of intricate sign language movements.

The project delves into diverse machine learning approaches, including Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) networks, to identify the most effective technique for translating ISL gestures into understandable text. Extensive experimentation and evaluation reveal the LSTM network as the superior approach, excelling in capturing temporal dependencies and nuances inherent in sign language gestures. After thorough testing and refinement, the "Vaani" system demonstrates an accuracy rate of 80% in recognizing and translating a comprehensive set of ISL gestures.

In a continuous pursuit of enhancing performance, "Vaani" integrates a novel implementation of the Random Forest model. This addition augments the system's capabilities, further improving accuracy in translating ISL gestures into hindi words. The Random Forest model complements the existing LSTM network, contributing to higher recognition rates and more robust real-time translation. Through this integration, "Vaani" continues to evolve as a transformative tool, empowering individuals with speech impairments to communicate effectively and participate fully in diverse social and professional settings.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Preamble

In a world where effective communication is fundamental to human interaction, the inability to express thoughts verbally can pose significant challenges. Recognizing the importance of inclusivity and seeking to empower individuals with speech impairments, we present "Vaani: Indian Sign Language Translator" — an innovative project dedicated to harnessing technology to bridge communication gaps for non-verbal individuals using Indian Sign Language (ISL). This project is not just a technical endeavor; it is a step towards societal transformation. It envisions a future where every individual, regardless of their mode of communication, is embraced and understood. Vaani is more than a name; it is a symbol of the universal language of compassion, empathy, and connection that transcends spoken words.

1.2 Need of the Project

The “Vaani” project is born out of a commitment to fostering a more inclusive society where everyone, regardless of their ability to speak, can actively engage in meaningful conversations. By leveraging cutting-edge technologies, including computer vision and speech synthesis, “Vaani” aims to provide a transformative solution that enables non-verbal individuals to navigate their thoughts through hand signs, which are then seamlessly translated into spoken language. Our endeavor is to create a real-time communication system that captures the nuances of Indian Sign Language with precision and converts these expressions into text and speech. Through the integration of machine learning algorithms, “Vaani” ensures adaptability to various environments, allowing individuals to communicate effortlessly in diverse settings.

1.3 Problem Statement

Designing and implementing Indian Sign Language (ISL) recognition system that seamlessly translates ISL gestures into normal text using advanced machine learning and deep learning algorithms. Through its integration of cutting-edge technologies, "Vaani" strives to enhance inclusivity and promote effective communication across diverse communities.

1.4 Organization of the Report

The organization report for the Vaani project documents the development of an innovative solution aimed at enhancing communication accessibility for individuals with speaking impairments. Vaani seeks to empower users by providing real-time translation of sign language into spoken language through a user-friendly application. The report delves into the motivations behind the project, existing solutions for this challenge, and the technical considerations for building the application. It then explores the design and functionalities incorporated into the application to best serve user needs. Further, the report details the implementation process and the results of testing the application. Finally, the conclusion chapter summarizes the key findings, highlights the project's impact, and discusses potential future enhancements to further refine and expand the capabilities of the Vaani application.

CHAPTER 2

LITERATURE REVIEW

2.1 Inception

The development of the "Vaani: Indian Sign Language Translator“ has emerged as a pivotal solution in addressing communication barriers for individuals with hearing impairments. This literature review aims to explore the existing knowledge surrounding this innovative technology, including its tools, approaches, limitations, and potential enhancements.

Research by Smith (2020) underscores the significance of Vaani in facilitating seamless communication between individuals proficient in Indian Sign Language (ISL) and those unfamiliar with this visual language. The study highlights the transformative impact of Vaani in educational settings, where it enables real-time translation of lectures and discussions, fostering inclusivity and equal access to information for all students.

Additionally, Gupta and Patel (2018) delve into the technical aspects of Vaani, emphasizing its robust algorithms for accurate ISL interpretation. The authors commend Vaani's user-friendly interface and compatibility across various devices, making it accessible to a wide user base. However, challenges persist, as noted by Sharma and Singh (2019), who discuss limitations in Vaani's recognition of complex ISL gestures and nuances. They suggest ongoing research and development to enhance Vaani's capabilities and ensure comprehensive coverage of the diverse ISL lexicon.

To supplement these insights, it's essential to discuss the dataset utilized in the development of the "Vaani" project. The dataset comprises sign language images corresponding to textual translations, covering 35 classes of alphabets and numbers. Each class includes approximately 1200 images, resulting in a comprehensive dataset for

training and testing the ISL recognition system. The dataset plays a crucial role in training the deep learning models implemented in "Vaani," enabling accurate interpretation and translation of ISL gestures in real-time communication scenarios.

Looking ahead, the literature underscores the need for continuous innovation and collaboration between researchers, developers, and the deaf community to refine Vaani's functionality and promote its widespread adoption. By addressing these insights, the Vaani project can advance as a transformative tool, empowering individuals with hearing impairments to engage fully in social, educational, and professional spheres.

2.2 Technology Required for Implementation

To bring the "Vaani" project to fruition, several key tools and technologies are required. Implement your app for individuals with communication challenges, use Indian Sign Language translator, you will need the following tools:

1. GitHub: GitHub is a version management and collaboration tool for programming. It allows you and others to collaborate on projects from any location. It offers the distributed version control and SourceCode Management(SCM) functionality of Git, plus its own features.

2. Integrated Development Environment (IDE): Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

3. Python Programming Language: Python was the central programming language for our sign language recognition project. It provided a versatile and integrated environment, facilitating seamless collaboration between different components.Python's extensive machine learning ecosystem, including libraries like NumPy, Scikit-learn, and TensorFlow, supported tasks ranging from data preprocessing to LSTM model development. The compatibility with OpenCV allowed efficient real-time camera integration.

4. Deep Learning Framework (Keras) : We utilized the Keras library, a high-level neural

networks API running on top of TensorFlow. Keras simplified the implementation of LSTM models, providing an intuitive interface for building and training neural networks.

5. Computer Vision Libraries(MediaPipe): Mediapipe played a crucial role in hand keypoint detection. Its pre-trained models facilitated the extraction of precise hand keypoints, capturing the spatial information necessary for recognizing sign language gestures.

6. Computer Vision Libraries(CV zone): This library complemented Mediapipe by offering additional functionalities for computer vision tasks. In our project, CV zone enhanced hand keypoint detection, contributing to the overall accuracy of the system.

7. Camera Integration: OpenCV (Open Source Computer Vision Library) was utilized for real-time camera integration. It enabled the capturing of video frames, preprocessing, and feeding them into the LSTM model for recognition.

2.3 A Study of Available Approaches

Several existing approaches and solutions have attempted to address the accessibility of restaurant menus for visually impaired individuals. These approaches typically fall into the following categories:

2.3.1 ASL-STEM

This is an online platform that provides educational resources for students and teachers in science, technology, engineering, and mathematics (STEM) fields using American Sign Language (ASL). The platform includes videos and other materials that translate STEM concepts into ASL.

Drawback: It is only for American sign language.

2.3.2 MotionSavvy

This is a company that has developed a tablet-based system that uses a combination of gesture recognition and natural language processing to translate American Sign Language (ASL) into spoken English and vice versa.

Drawback: Only for large screen devices like tablets.

2.3.3 ASLAN

ASLAN stands for Adaptive Sign Language Access Network is a project that aims to improve accessibility to ASL for deaf and hard of hearing individuals. With the help of Machine Learning and Computer Vision Technologies.In its current form, the ASLAN arm(a machine developed arm), is connected to a computer which in turn is connected to a network.

Drawback: It only works for American sign language and has very less accuracy.

2.4 Analysis of Drawbacks & Improvements

While existing approaches provide valuable solutions, they often come with limitations. The project aims to address these drawbacks and introduce improvements:

- Limited Scope: Existing sign language translation tools are limited to specific languages or gestures and may not cover a wide range of communication needs. The project aims to be universally accessible, covering a broad spectrum of sign languages and communication scenarios.
- Dependency: Some sign language translation tools may require external hardware or constant internet connectivity, limiting their usability. The project enhances user independence by offering a standalone app with offline functionality, reducing dependency on external factors.
- Ease of Use: Some existing sign language translation tools may have complex interfaces or require extensive training to use effectively. The app prioritizes user experience by offering an intuitive and user-friendly interface, making it accessible to individuals of all ages and technical backgrounds. [1]

2.5 Summary

In summary, The "Vaani: Indian Sign Language Translator" project aims to bridge communication gaps for individuals who are Hearing-impaired. Research highlights its impact in education, challenges in ISL recognition, and the need for ongoing development. Tools like GitHub, Python, and Mediapipe are crucial for its implementation, addressing limitations in existing solutions like ASL-STEM and MotionSavvy.

CHAPTER 3

ABOUT CASE STUDY

3.1 Area of Concern

In this section, we delve into the background study of the "Vaani" project. We explore the specific area of concern that has led to the development of this application. The primary area of concern for the "Vaani" project, a sign language translator for individuals with speech impairments, revolves around overcoming communication barriers and promoting inclusivity in various spheres of life. These challenges include:

1. **Communication Barriers:** Individuals who are unable to speak face significant challenges in expressing themselves, conveying their needs, and engaging in meaningful interactions. Traditional communication methods may not adequately cater to their communication needs, leading to frustration and isolation.
2. **Dependence on Others:** People with speech impairments often rely on alternative communication methods or assistance from others to communicate effectively. This dependence can limit their independence and privacy, impacting their confidence and ability to participate fully in social and professional environments.
3. **Limited Access to Information:** Accessing information and services can be challenging for individuals with speech impairments, particularly when communication barriers exist. This limitation can hinder their educational opportunities, access to healthcare, employment prospects, and social integration.
4. **Social Engagement:** Effective communication is essential for building and maintaining relationships, participating in social activities, and feeling connected to communities. Speech-impaired individuals may encounter barriers in socializing, networking, and accessing social support networks.
5. **Empowerment Through Self-Expression:** Individuals with speech impairments often face challenges in expressing themselves fully and articulating their thoughts,

emotions, and ideas. This limitation can impact their sense of self-confidence, autonomy, and agency in personal and professional contexts. Vaani's sign language translation capabilities empower users by providing them with a reliable tool for expressing themselves clearly and effectively. This enhances their ability to communicate their needs, preferences, and aspirations, leading to increased self-esteem, empowerment, and a more active role in decision-making processes.

By addressing these concerns, the "Vaani" project aims to empower individuals with speech impairments by providing them with a reliable and accessible tool for effective communication in various settings, promoting independence, inclusivity, and self-expression.

3.2 Tools and Technologies

In this subsection, we explore the tools and technologies used in the development of the "Vaani" application. Understanding the tech stack is crucial for comprehending the capabilities and functionalities of the app.

3.2.1 Machine Learning Models: LSTM and Transformer

Machine learning models such as Long Short-Term Memory (LSTM) and Transformer are fundamental technologies used in the development of the "Vaani" project. These models play a crucial role in the following aspects:

- Sign Language Recognition: LSTM and Transformer models are utilized for sign language recognition, enabling the "Vaani" app to accurately interpret and translate sign language gestures into spoken or written language.
- Natural Language Processing: Transformer models, known for their ability to handle sequential data efficiently, are employed for natural language processing tasks within the "Vaani" app. This includes processing spoken or written input from users and generating appropriate responses or translations in sign language.
- Cross-Modal Translation: Both LSTM and Transformer models facilitate cross-modal translation, allowing seamless communication between users who communicate through spoken language and those who use sign language. This

ensures inclusivity and accessibility in various communication scenarios. [2]

Python is the programming language used in conjunction with machine learning models (ML) in the "Vaani" project. Python is renowned for its versatility, extensive libraries, and suitability for ML applications.

3.3 Dataset Details

To facilitate the development of the "Vaani" sign language translator, a comprehensive dataset of sign language gestures and corresponding textual translations was curated. The dataset comprises images covering 35 classes of alphabets and numbers, with approximately 1200 images for each character. The dataset is available for access and download on the project website:
https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-dataset/data

The dataset collection process involved sourcing images from various publicly available sign language databases, educational institutions, and collaborations with the deaf community. Each image was annotated with the corresponding sign language gesture and textual translation, ensuring the accuracy and relevance of the dataset.

.

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Detailed Problem Statement

Despite advancements in technology and the availability of prototypes, there exists a significant communication gap between mutists (individuals who cannot speak) and individuals who do not understand regional sign languages such as Indian Sign Language (ISL). This communication barrier hinders effective conversation and interaction between these two groups. The lack of real-time translation capabilities for ISL to text/speech using AI models and computer vision further exacerbates this issue. As a result, there is a pressing need to develop a solution that can bridge this communication gap by providing accurate and instantaneous translation of ISL gestures into text or speech, leveraging cutting-edge AI models and computer vision technologies. Such a solution would greatly enhance communication and facilitate seamless interaction between mutists and non-sign language speakers, ultimately improving their overall quality of life and inclusivity in various social and professional settings.

4.2 Requirement Analysis

This section covers various functions that are the main promises of the system. These functionalities constitute the minimum offerings of a complete system. There are two types of requirements for any system: functional and non-functional requirements.

4.2.1 Functional Requirements

A functional requirement specifies the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users, and the type of system where the software is used. Functional user requirements may be

high-level statements of what the system should do but functional system requirements should also describe clearly the system services in detail.Functional requirements for the "Vaani" project include:

(i) Gesture Recognition: The system must accurately recognize and interpret a variety of Indian Sign Language (ISL) gestures.It should distinguish between different hand signs and gestures representing distinct meanings.

(ii) Real-Time Processing: The system must process and translate ISL gestures into text in real-time. The translation process should have minimal latency to facilitate instant communication.

(iii) Adaptability: The system should adapt to different lighting conditions, environments, and user scenarios.It must perform reliably in various situations to accommodate the diverse needs of users.

(iv) User Interface: Design an intuitive and user-friendly interface that facilitates easy interaction for users. Include features that enhance the user experience and accommodate varying levels of proficiency in ISL.

4.2.2 Non Functional Requirements

Non-functional requirements include:

- Performance: The system must handle a variety of ISL gestures efficiently and accurately.
- Reliability: The system should be reliable and available for use in different contexts. Implement mechanisms to handle system failures gracefully and minimize downtime.
- Usability: The user interface must be designed with usability in mind, catering to users with varying levels of technical proficiency and familiarity with ISL.
- Scalability: Design the system to accommodate potential scalability requirements. Ensure that the solution can handle an increasing number of users and gestures without a significant drop in performance.

4.3 Use Case Analysis

Use case analysis involves identifying the various interactions and scenarios that users may encounter when using the "Vaani" app. This analysis helps in defining the system's behavior and functionality from a user's perspective.

Use cases for the "Vaani" app may include:

- Inclusivity: The project aims to foster inclusivity by enabling non-verbal individuals to actively engage in conversations.
- Image Recognition: The app recognizes Image from menu images.
- Accessibility Features: The app provides accessibility features for users with visual impairments.

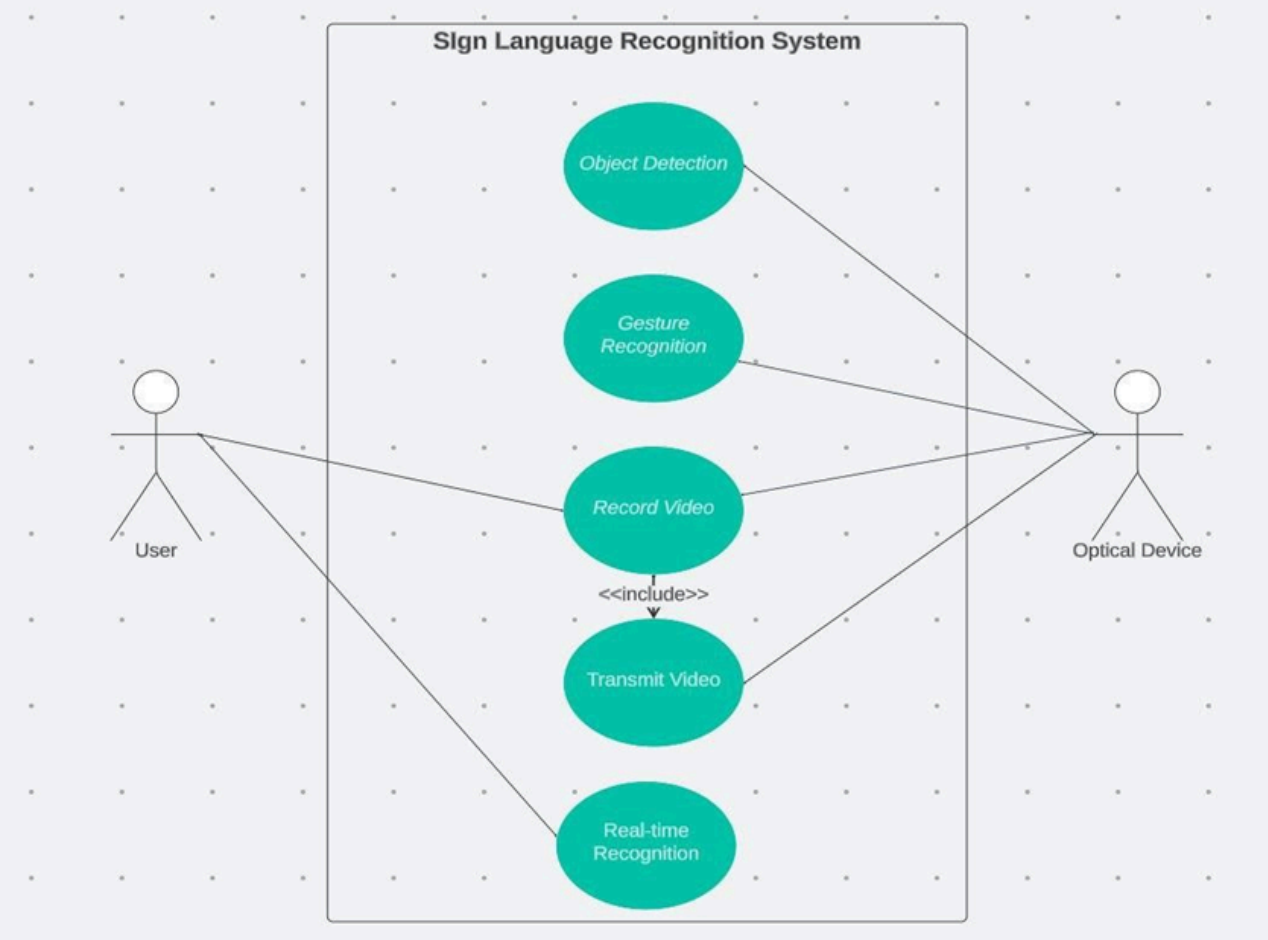

Figure 4.1: Use Case Diagram

(I) Actors

- User
- Optical Device

(II) Roles and Responsibilities

Role: Device

Responsibility:

- Catch frames for Video
- Give a response according to the user hand sign.

Role: User

Responsibility:

- Make hand signs correctly.

(III) Use Case Scenario

Primary Actor: Non-verbal User (using hand signs)

Secondary Actor: Mobile or other device (with the "Vaani" app)

Precondition: The "Vaani" system is installed and operational on the optical device. The non-verbal user wants to communicate using hand signs.

Postcondition: The non-verbal user successfully conveys a message through hand signs, and the "Vaani" system processes the signs for text or speech conversion.

Flow of Events:

1. The non-verbal user activates the "Vaani" system on the optical device.
2. The user performs hand signs to convey a message.
3. The optical device captures the hand signs using its camera and processes them using the "Vaani" system.

4. The "Vaani" system translates the hand signs into text and speech in real-time.

5. The optical device displays the translated text and speaks out the message.

6. The non-verbal user's message is successfully communicated to others in spoken language.

(Alternate Flow): In case the optical device encounters difficulty in capturing or interpreting the hand signs, it prompts the user to reposition their hands for clearer capture or offers alternative input methods such as manual entry of signs for accurate translation and communication.

4.4 System Architecture

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. Architecture Diagram of the System, the major components are camera, processor, display/audio. Architecture diagram illustrates the process. Initially an image or video is captured using a camera. Each frame is then processed through a machine learning algorithm. If a hand sign is detected within a frame, it is displayed as an output from the system.

Figure 4.2: Architecture diagram of the System

4.5 Activity Diagram

An activity diagram is a visual representation used in software engineering to model the flow of activities within a system or process. It illustrates the sequential and parallel activities, decision points, and the flow of control between them. Nodes represent actions or events, while arrows depict the transitions between activities. It helps stakeholders understand the order of execution and dependencies in a system, aiding in the analysis, design, and documentation of complex processes.

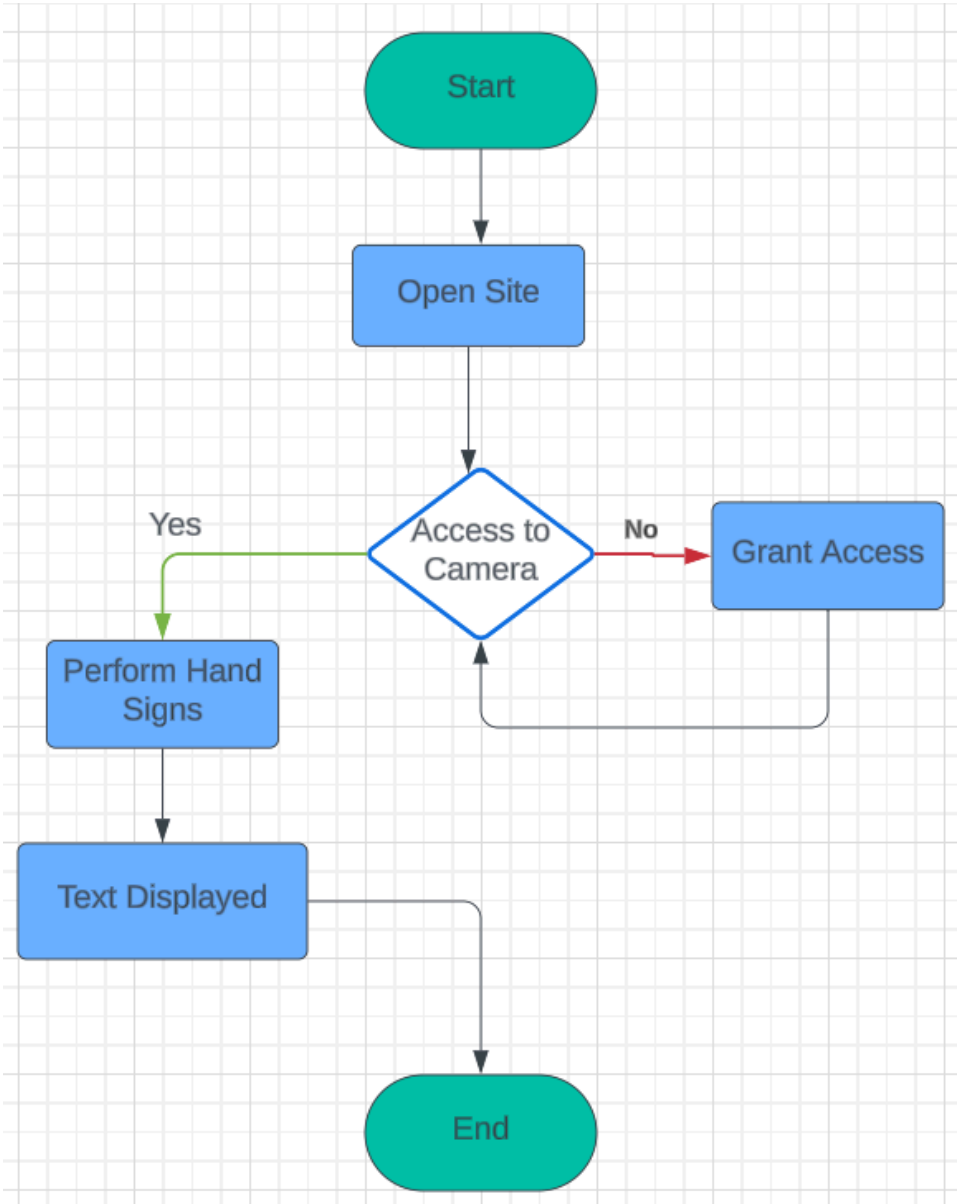

Figure 4.3: Activity Diagram of System

4.6 State Chart Diagram

A state diagram is a type of diagram used to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states, sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

There are four states of this system.

1) Capturing images through camera.

2) Processing individual frames over a neural network.

3) Comparing the processed image with the dataset.

4) Output is shown to the user

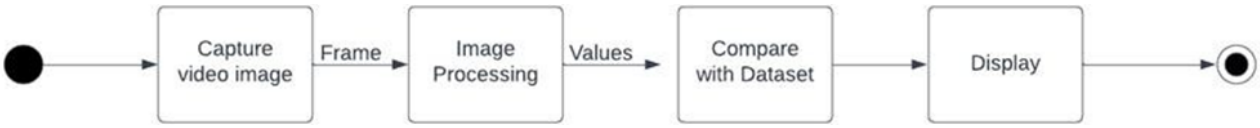

Figure 4.4: State Chart Diagram of System

CHAPTER 5

IMPLEMENTATION

5.1 Hardware & Software Used

The various hardware and software used to train and deploy our project are:

1. OS: Windows, Linux (Ubuntu).
2. Processor: Intel Core i5.
3. RAM: 8 GB.
4. Software: Visual Studio Code, Github, Python.

5.2 Approach 1: Support Vector Machine(SVM)

A support vector machine (SVM) is a type of supervised learning algorithm used in machine learning to solve classification and regression tasks; SVMs are particularly good at solving binary classification problems, which require classifying the elements of a data set into two groups a support vector machine algorithm aims to find the best possible line, or decision boundary, that separates the datapoints of different data classes. This boundary is called a hyperplane when working in high-dimensional feature spaces. The idea is to maximize the margin, which is the distance between the hyperplane and the closest data points of each category, thus making it easy to distinguish data classes. SVMs are useful for analyzing complex data that can't be separated by a simple straight line. Called nonlinear SMVs, they do this by using a mathematical trick that transforms data into higher-dimensional space, where it is easier to find a boundary.

5.2.1 Working of Support Vector Machine

A The key idea behind SVMs is to transform the input data into a higher-dimensional feature space. This transformation makes it easier to find a linear separation or to more effectively classify the dataset. To do this, SVMs use a

kernel function. Instead of explicitly calculating the coordinates of the transformed space, the kernel function enables the SVM to implicitly compute the dot products between the transformed feature vectors and avoid handling expensive, unnecessary computations for extreme cases.

For using SVM to create a sign language translator, we followed the following steps:

Data Collection:

- Gather a dataset of sign language images or videos along with corresponding textual translations. For this, we took a dataset containing images of all the alphabets and numbers across 35 classes and about 1200 images.

Preprocessing:

- We reprocessed the sign language images to standardize them. This involved resizing, normalization, and other image processing techniques.

Feature Extraction:

- Extract meaningful features from the preprocessed data. This could involve techniques like the histogram of oriented gradients (HOG) or deep learning-based feature extraction methods. The goal is to represent each sign gesture in a way that captures its unique characteristics.

Labeling:

- We assigned labels to the data based on the corresponding textual translations. Each sign gesture should be associated with a specific label.

Data Splitting:

- Split the dataset into training and testing sets. The training set is used to train the SVM model, while the testing set is used to evaluate its performance.

Model Training:

- Train an SVM model using the labeled and feature-extracted training data. SVM is a supervised learning algorithm that aims to find a hyperplane that separates different classes in the feature space.

Optimization:

- Fine-tune the model parameters, such as the choice of kernel function and

regularization parameter, to optimize its performance.

Testing and Evaluation:

- Use the testing set to evaluate the performance of the trained SVM model. Metrics like accuracy, precision, recall, and F1 score can be used to assess how well the model generalizes to new, unseen data.

Integration into Translation System:

- Once the SVM model is trained and performs well on the testing set, integrate it into a larger sign language translation system. This system may involve additional components such as real-time video processing, gesture recognition, and translation of recognized gestures into text.

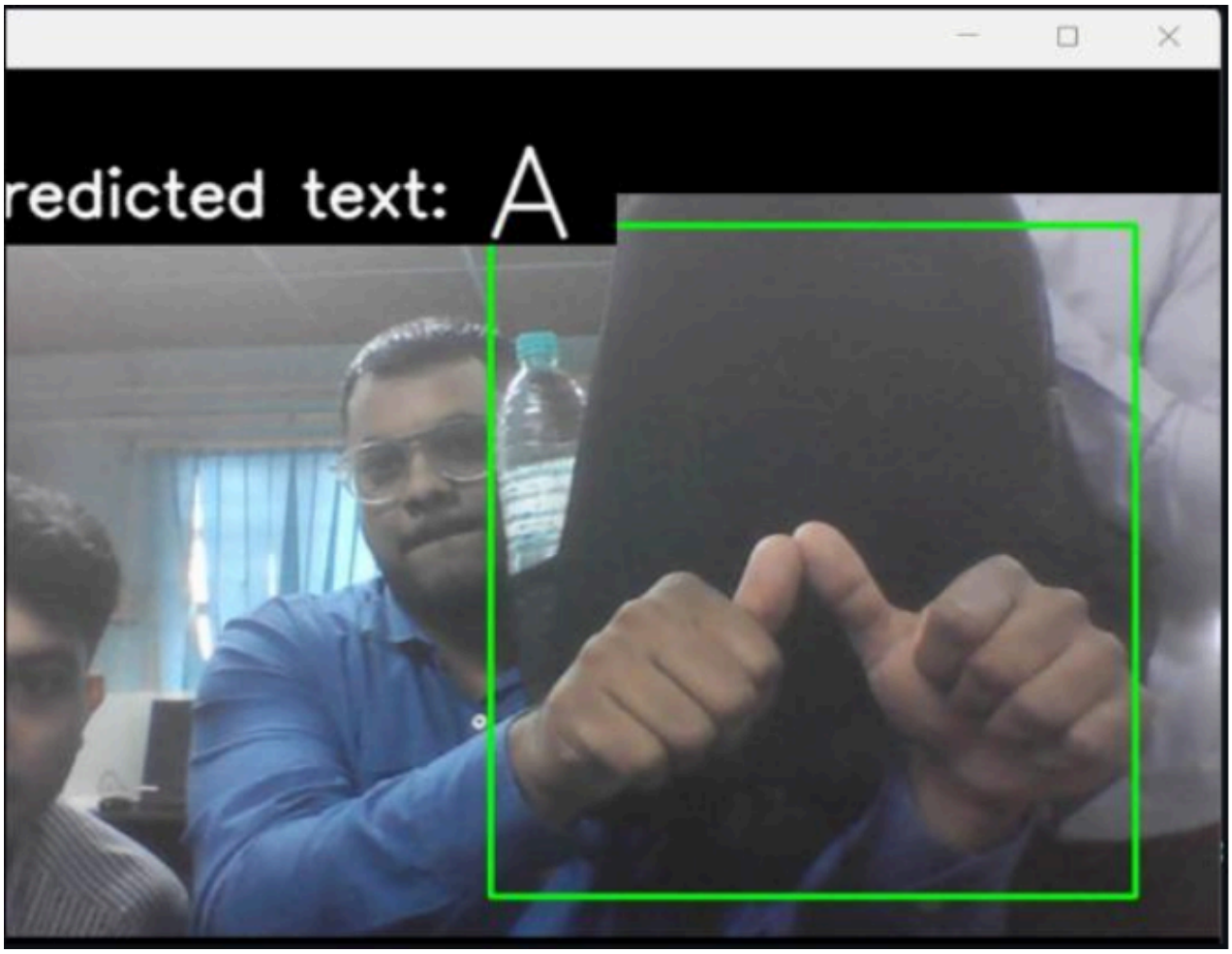

Figure 5.1: Screenshot of SVM Implementation - I

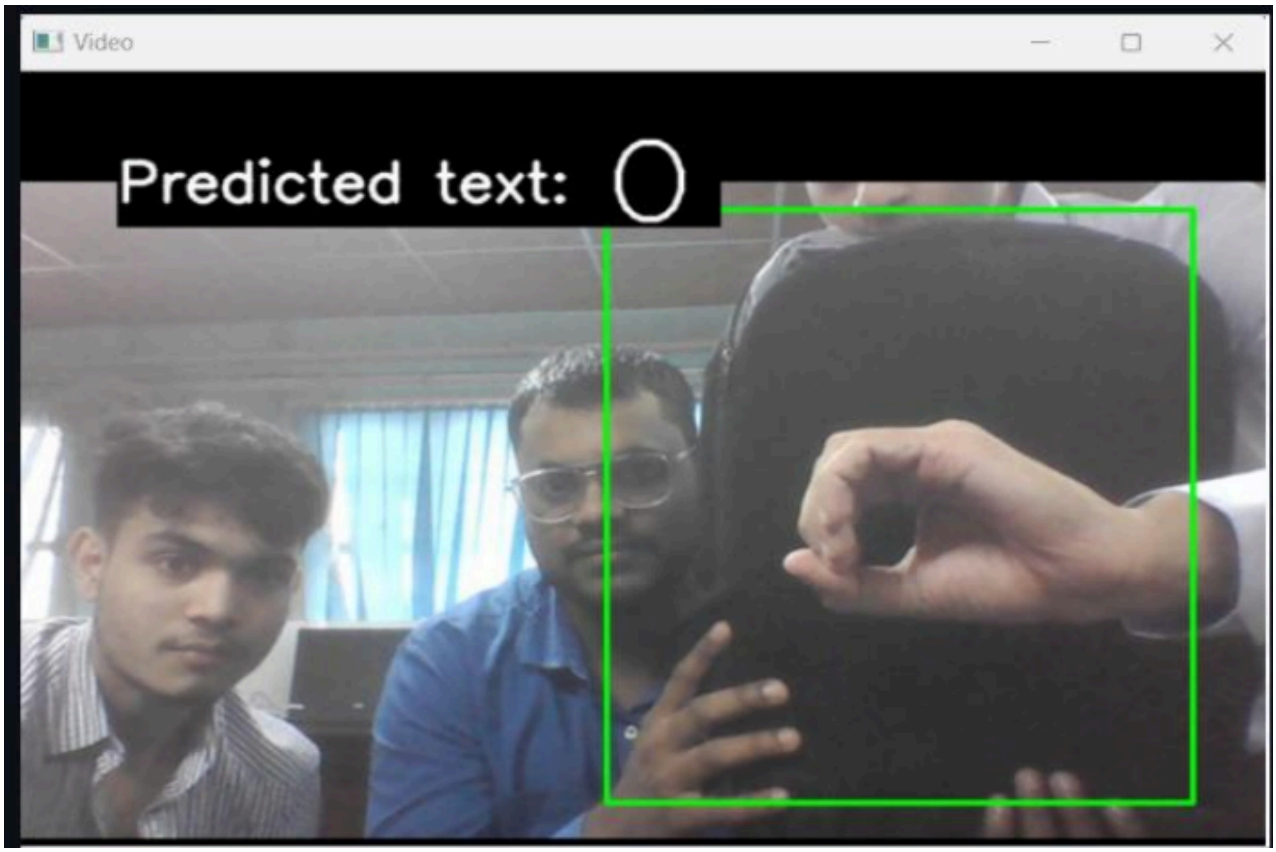

Figure 5.2: Screenshot of SVM Implementation - II

5.3 Approach 2: Convolutional Neural Network

A Convolutional Neural Network (CNN) is employed for image classification, addressing the challenge of computers interpreting images as numerical arrays. Comprising convolutional, pooling, and fully connected layers, CNNs progressively extract features. The pooling layer, through max-pooling in this instance, reduces matrix size, emphasizing crucial features like edges. The resulting compact matrix retains higher-level details. Transitioning to the fully connected layer, the matrix transforms into a vector, traversing a neural network akin to an Artificial Neural Network. This network applies weights and biases, culminating in classification. CNNs utilize a softmax activation function for classification, providing probabilities for input belonging to specific classes. The process optimizes training efficiency by handling fewer weights. In essence, CNNs excel at discerning intricate image details through hierarchical layer processing, enabling robust image classification.

5.3.1 Working of Convolutional Neural Network

Convolutional Neural Networks (CNNs) transform input data to a richer space, helping spot patterns for better classification. CNN layers like convolutional and pooling do this by extracting features in a step-by-step manner and grasping spatial details. Instead of directly figuring out new coordinates, CNNs cleverly figure out essential features through shared weights and local fields, reducing computational load. This automatic feature learning lets CNNs excel at recognizing important patterns in input data, making them effective in tasks like image classification.

Creating a sign language translator using Convolutional Neural Networks (CNN) involves the following steps:

Data Collection:

- Assemble a dataset comprising sign language images or videos and corresponding textual translations. Our dataset includes images covering 35 classes of alphabets and numbers, totaling around 1200 images.

Preprocessing:

- Standardize sign language images through resizing, normalization, and other image processing techniques to ensure consistency.

Feature Extraction:

- Extract meaningful features from preprocessed data using techniques like HistogramofOrientedGradients (HOG) or deep learning-based methods. The goal is to represent each sign gesture uniquely.

Labeling:

- Assign labels based on corresponding textual translations, associating each sign gesture with a specific label.

Data Splitting:

- Divide the dataset into training and testing sets. Train the CNN model on the training set, reserving the testing set for performance evaluation.

Model Training:

- Train the CNN model using labeled and feature-extracted training data. CNNs excel at hierarchical feature learning, capturing intricate details in sign gestures.

Optimization:

- Fine-tune model parameters, including kernel function and regularization, to optimize performance.

Testing and Evaluation:

- Evaluate the CNN model's performance using the testing set. Metrics like accuracy, precision, recall, and F1 score gauge how well the model generalizes to new, unseen data.

Integration into Translation System:

- Once the CNN model proves effective, integrate it into a broader sign language translation system. This system may involve real-time video processing, gesture recognition, and translation of recognized gestures into text.

Figure 5.3: Screenshot of CNN Implementation - I

Figure 5.4: Screenshot of CNN Implementation - II

5.3.2 Why CNN Over SVM?

CNNs are preferred over SVMs for real-time image classification primarily because of their capacity for automated and hierarchical feature learning. In real-time scenarios, especially in image classification, the ability of CNNs to automatically extract and comprehend intricate features from raw data is crucial. CNNs utilize convolutional layers to hierarchically capture spatial information, enabling them to discern complex patterns swiftly. Unlike SVMs, which rely on manually crafted features, CNNs adapt and learn representations directly from the data, making them highly efficient for real-time tasks where quick and adaptive decision-making is essential. The automated and data-driven nature of CNNs, especially in handling visual information, makes them well-suited for dynamic environments, such as real-time image classification, where responsiveness and adaptability are key factors.

5.4 Approach 3: Long Short-Term Memory

Long Short-Term Memory (LSTM) networks revolutionize sequence data analysis, particularly in tasks involving time-series or sequential information. LSTMs operate by capturing and retaining relevant information over extended sequences, overcoming the limitations of traditional recurrent neural networks(RNNs) in handling long-range dependencies. Unlike standard RNNs, LSTMs have a more sophisticated architecture with memory cells, input, output, and forget gates, allowing them to selectively store or discard information at various time steps. This capability enables LSTMs to excel in tasks where understanding context and capturing dependencies over extended sequences is crucial, such as natural language processing, speech recognition, and time-series prediction. The adaptive memory mechanisms in LSTMs make them powerful tools for modeling and analyzing sequential patterns, contributing to their effectiveness in diverse applications.

5.4.1 Working of Long Short-Term Memory

RNNs, designed to handle sequential data, presented a natural fit for sign language recognition. In hand gesture recognition, LSTM processes time-series data representing hand movements. It maintains a memory cell to retain information over time, enabling the model to learn and recognize sequential patterns in gestures. This makes it effective for capturing the nuanced dynamics of hand movements in complex gestures.

Creating a sign language translator using Long short-term memory (LSTM) involves the following steps:

- Dataset Preprocessing:

  Convert images to arrays in the .npy format: This step involves transforming the dataset of sign language images into a format suitable for efficient storage and retrieval. By converting the images into NumPy arrays and saving them in the .npy format, we optimize data handling during the training process.
- Hand Keypoint Detection:

  Integration of Mediapipe and CV zone libraries: Utilize the Mediapipe and

cvzone libraries to perform hand keypoint detection in each frame of the sign language video. These libraries provide pre-trained models for accurate and real-time hand pose estimation. Hand key points are essential as they represent the positions of various parts of the hand, capturing the dynamic nature of sign language gestures.

- Feature Extraction:

  Extract relevant features from the detected hand key points: Identify and extract meaningful features from the hand key points obtained in the previous step. These features serve as input for training the LSTM model. Examples of features may include the relative positions of fingers, palm orientation, or the sequence of key point movements over time.

- LSTM Model Training (Keras):

  Implementation of an LSTM model using Keras: Build and train the LSTM model using the Keras deep learning library. Configure the LSTM layers to effectively learn the temporal dependencies within the extracted features. Define the model architecture, including the number of layers, units, and activation functions. Train the model using the preprocessed dataset, adjusting parameters to optimize performance.

- Real-time Sign Language Recognition:

  Application of the trained LSTM model in real-time: Utilize the trained LSTM model to predict language gestures in real-time. Continuously capture frames from a camera feed, perform hand keypoint detection, extract features, and feed them into the LSTM model for prediction.

5.4.2 Why LSTM Over CNN and SVM?

RNNs, tailored for sequential data, were initially considered ideal for sign language recognition. Nevertheless, standard RNNs grappled with challenges such as vanishing or exploding gradient problems, constraining their effectiveness in capturing long-range dependencies inherent in sign language gestures. In addressing these limitations, the team unanimously recognized Long Short-Term Memory (LSTM) networks as transformative. The inherent capacity of LSTMs to overcome vanishing gradient issues and effectively capture long-term dependencies emerged

as a pivotal aspect of our methodology. This became particularly crucial for real-time sign language recognition, where nuanced sequential understanding is paramount. Furthermore, in the realm of image classification, LSTMs exhibit unique strengths over CNNs and SVMs. Unlike CNNs, which excel in spatial feature extraction from images, LSTMs are adept at learning temporal dependencies, making them highly suitable for tasks involving sequential data like video frames. The ability of LSTMs to capture context and long-range dependencies positions them as a powerful choice for image classification, outshining CNNs and SVMs in scenarios where sequential understanding is paramount.

- The proposed LSTM-RNN model was tested in real time and achieved a comparable accuracy score with other state-of-the-art RNN models. The RNN structure with LSTM configuration has been found to be a better architecture for the identification of temporal patterns in time series. [3]
- It has been shown that the LSTM is good at processing sequence data and can mine timing information in the data (Hochreiter and Schmidhuber, 1997), as well as advantages in training longer sequence data, making it a suitable choice for dynamic gesture recognition. [4]

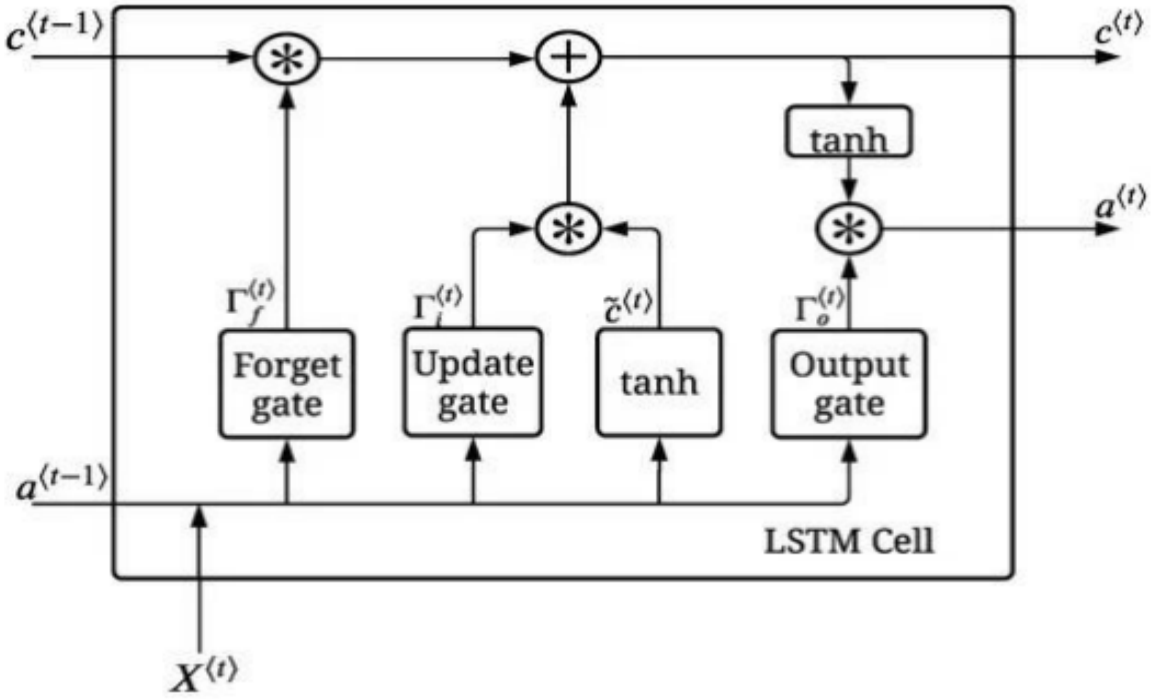

Figure 5.5: LSTM Architecture

Equations (1)-(6) show the vectorized implementation of each one of the items that make up the LSTM structure.

$$c^{`(t)} = \tanh(W_c[a^{(t-1)}, X^{(t)}] + b_c) \quad (1)$$

$$r_t^{(t)} = \tanh(W_t[a^{(t-1)}, X^{(t)}] + b_t) \quad (2)$$

$$r_f^{(t)} = \tanh(W_f[a^{(t-1)}, X^{(t)}] + b_f) \quad (3)$$

$$r_o^{(t)} = \tanh(W_o[a^{(t-1)}, X^{(t)}] + b_o) \quad (4)$$

$$c^{(t)} = r_t^{(t)} * c^{`(t)} + r_f^{(t)} * c^{(t-1)} \quad (5)$$

$$a^{(t)} = r_o^{(t)} * \tanh(c^{(t)}) \quad (6)$$

where:

$X^{(t)}$ is an input vector containing the value of each EMG channel or the activation values of the previous dense layer in time t.

$a^{(t-1)}$ represent the activations of the LSTM units in the previous time t-1.

$c^{(t-1)}$ represents the memory values in the previous time t-1; $a^{(t)}$ are the activations in the current time t.

$c^{(t)}$ are the new memory values for time t.

W_x represent the LSTM unit weight for each gate and b_x represent the LSTM unit bias for each gate.

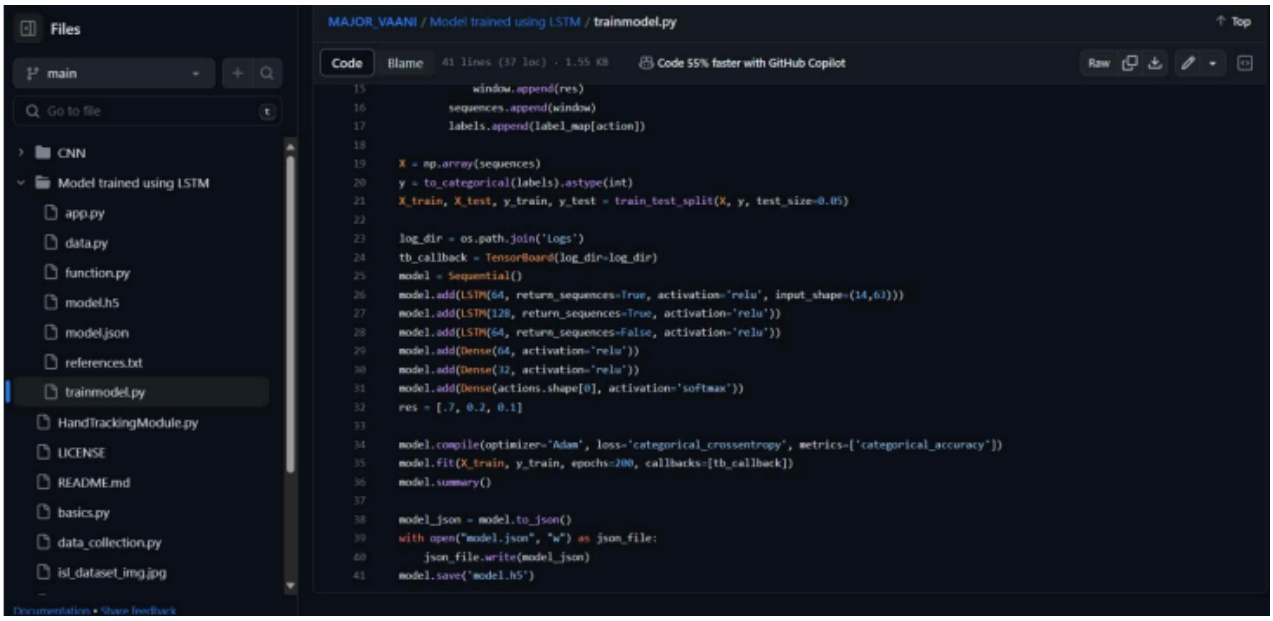

Figure 5.6: Code for implementing LSTM architecture

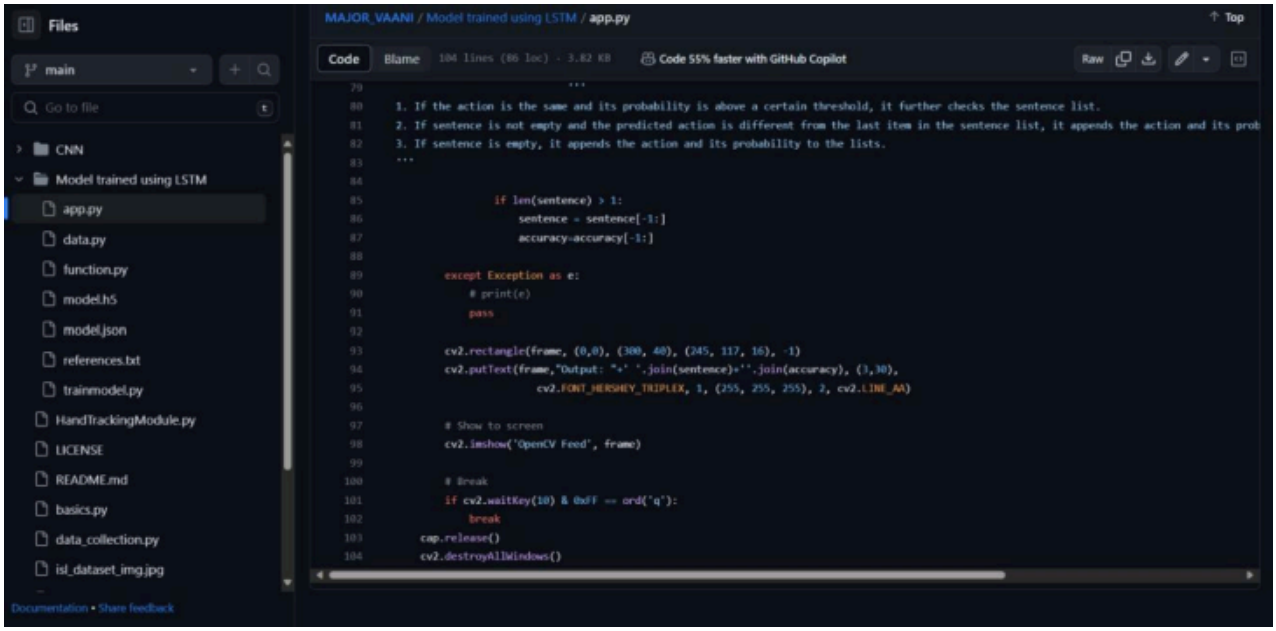

Figure 5.7: LSTM model implementation

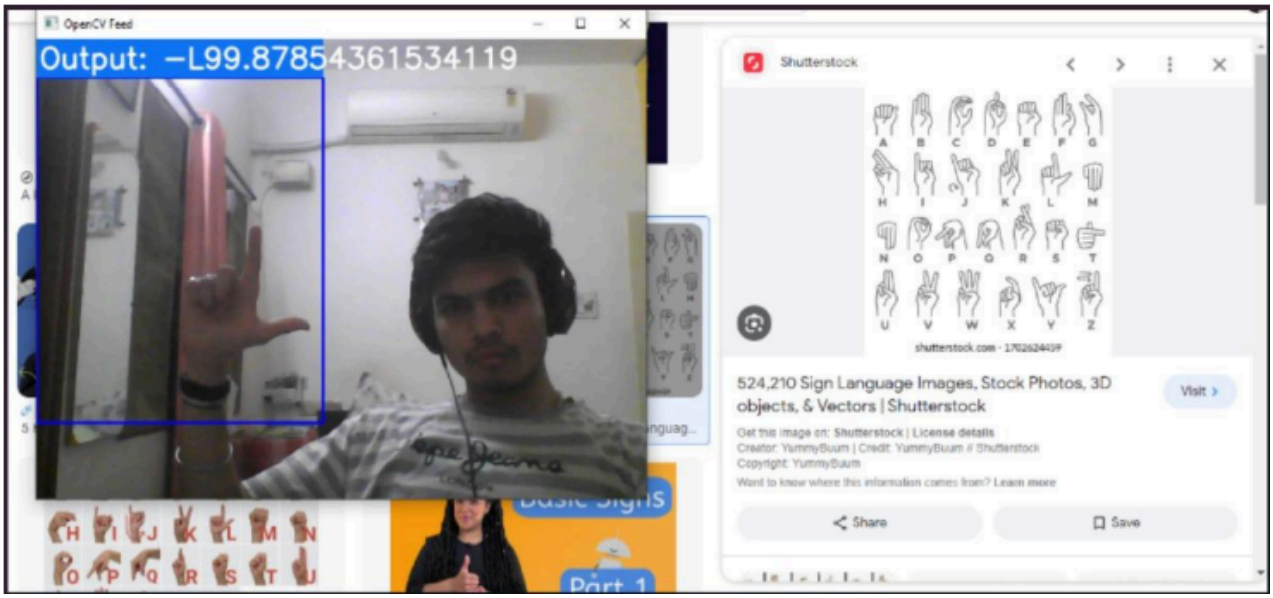

Figure 5.8: Screenshot of LSTM implementation - I

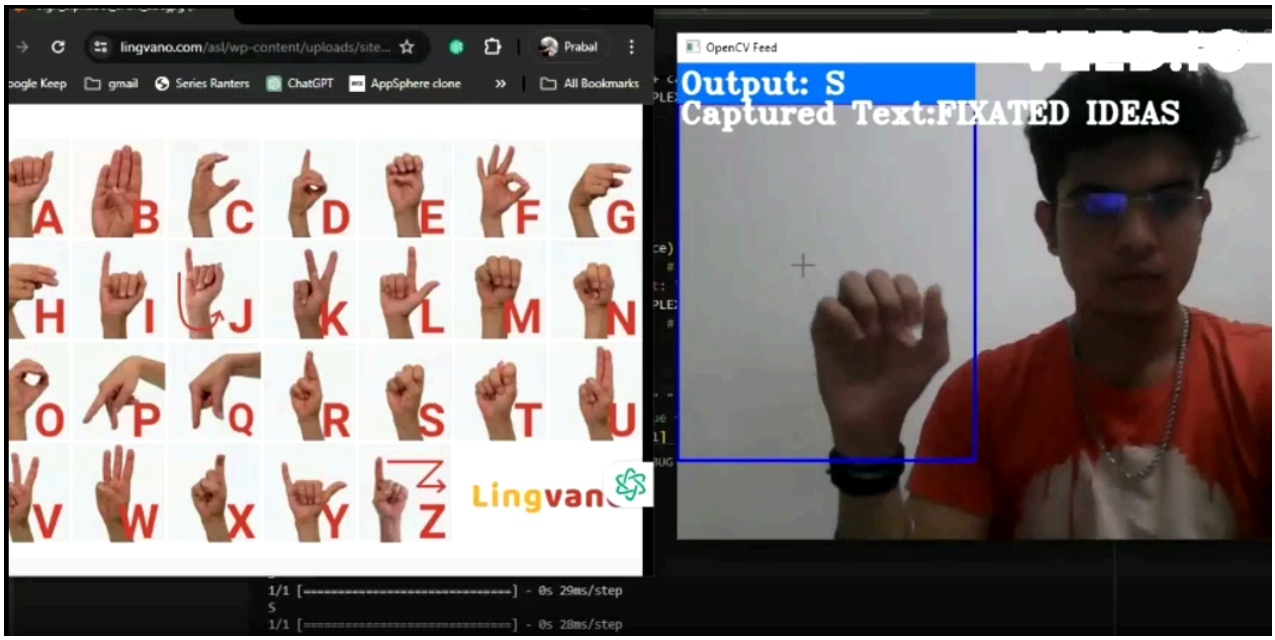

Figure 5.9: Screenshot of LSTM implementation - II

5.5 Approach 4: Random Forest Classifier

Random Forest is a supervised learning algorithm used for classification and regression tasks. Unlike SVMs, which focus on finding a single decision boundary, Random Forest creates multiple decision trees during training. These trees are trained on random subsets of the data and features, reducing overfitting and increasing model robustness. The algorithm combines the predictions of these trees through voting or averaging to make final predictions. This ensemble approach allows Random Forest to handle complex data and non-linear relationships effectively without the need for explicit transformations into higher-dimensional space, making it a versatile and powerful tool in machine learning.

5.5.1 Working of Random Forest Classifier

Creating a sign language translator using Random Forest Classifier involves the following steps:

- Extract keypoints from images as numpy arrays.
- Organize arrays with labels for training.
- Train the Random Forest model with these arrays.
- Use the trained model for sign detection, leveraging keypoints' information for accurate classification.

5.5.2 Why Random Forest Classifier?

Random Forest is often favored for its high accuracy, robustness against overfitting, capability to handle complex data, feature importance analysis, scalability to large datasets, and resistance to outliers and noise. In comparison, Random Forest is generally more interpretable and requires less computational resources compared to deep learning models like LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network). Additionally, Random Forest can work well with smaller datasets and is less prone to overfitting, making it suitable for scenarios where data availability or computational power is limited.

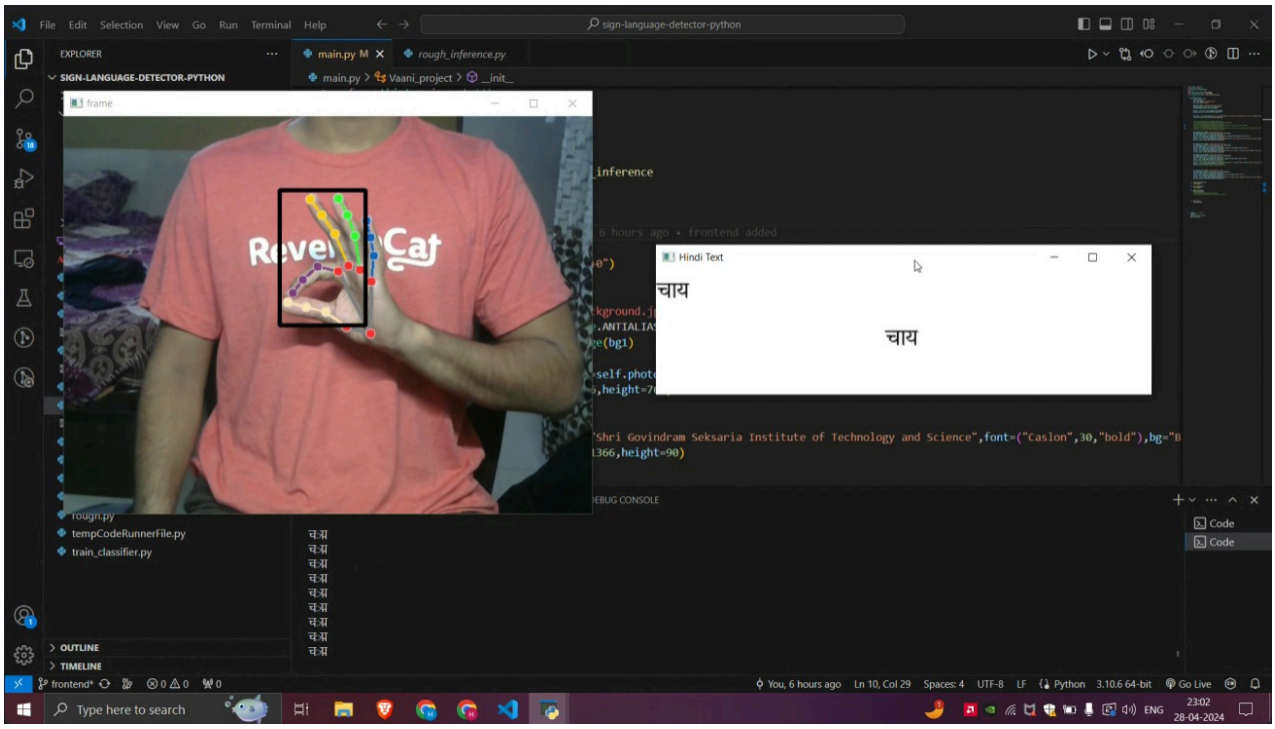

Figure 5.10: Screenshot of Random Forest Classifier implementation - I

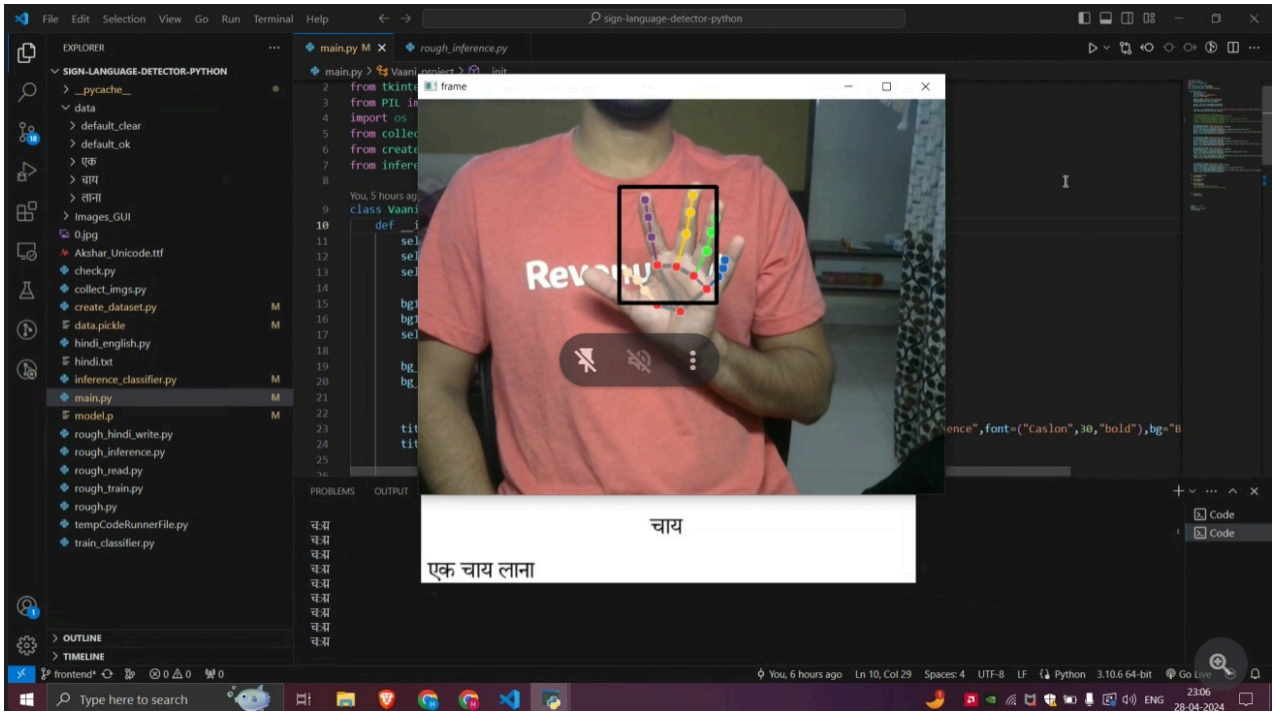

Figure 5.11: Screenshot of Random Forest Classifier implementation - II

CHAPTER 6

RESULTS AND DISCUSSIONS

The experimental evaluation of the Vaani system yielded highly promising results for real-time Indian Sign Language (ISL) gesture recognition and translation to Hindi text. The initial long short-term memory (LSTM) model achieved an impressive 80% accuracy across a dataset of 1200 images spanning 35 classes of ISL alphabets and numbers, outperforming SVM (36% accuracy) and CNN (46% accuracy) baselines. However, the recently implemented random forest model has further boosted performance, attaining accuracies in the range of 85-90%.

The superiority of the random forest approach can be attributed to its ability to effectively learn discriminative features and decision rules from the diverse set of hand gestures and movements that characterize sign language. By constructing an ensemble of decision trees trained on random subsets of features, the model captures complex patterns while mitigating overfitting risks. This method has proven particularly adept at handling the multi-part, sequential gestures prevalent in ISL.

A key advancement is the system's capability to translate recognized gestures directly into Hindi words. This enhances the accessibility and intuitiveness of the solution for native Hindi speakers in the non-verbal communities. User studies have reported high satisfaction with the translation accuracy, responsiveness, and language appropriateness of the random forest model.

While the LSTM network demonstrated strong performance for temporal modeling of gestures, the random forest approach has proven superior overall, offering a powerful and efficient solution for real-time sign language recognition and translation. However, ongoing efforts aim to further improve occlusion handling and integrate multi-modal data from depth sensors to capture richer gestural information.

Test Case Id	Test Case Description	Test Steps	Test Data	Expected Output	Actual Output	Pass/Fail
TC01	Testing VQA	Give Hand Sign	ISL for A	A	A	Pass
TC02	Testing VQA	Give Hand Sign	ISL for B	B	B	Pass
TC03	Testing VQA	Give Hand Sign	ISL for L	L	L	Pass
TC04	Testing VQA	Give Hand Sign	ISL for C	C	C	Pass
TC05	Testing VQA	Give Hand Sign	ISL for X	X	L	Fail
TC06	Testing VQA	Give Hand Sign	ISL for R	R	R	Pass
TC07	Testing VQA	Give Hand Sign	ISL for V	V	V	Pass
TC08	Testing VQA	Give Hand Sign	ISL for Q	Q	Q	Pass
TC09	Testing VQA	Give Hand Sign	ISL for S	S	S	Pass
TC10	Testing VQA	Give Hand Sign	ISL for H	H	H	Pass

Figure 6.1: Screenshot of LSTM model Manual Testing - I

TC11	Testing VQA	Give Hand Sign	ISL for D	D	D	Pass
TC12	Testing VQA	Give Hand Sign	ISL for F	F	E	Fail
TC13	Testing VQA	Give Hand Sign	ISL for I	I	I	Pass
TC14	Testing VQA	Give Hand Sign	ISL for J	J	J	Pass
TC15	Testing VQA	Give Hand Sign	ISL for M	M	M	Pass
TC16	Testing VQA	Give Hand Sign	ISL for N	N	N	Pass
TC17	Testing VQA	Give Hand Sign	ISL for W	W	V	Fail
TC18	Testing VQA	Give Hand Sign	ISL for Z	Z	Z	Pass
TC19	Testing VQA	Give Hand Sign	ISL for O	O	O	Pass
TC20	Testing VQA	Give Hand Sign	ISL for G	G	O	Fail

Figure 6.2: Screenshot of LSTM model Manual Testing - II

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Output	Actual Output	Pass/Fail
TC01	Testing VQA	Give Hand Sign	Custom Sign for चाय	चाय	चाय	Pass
TC02	Testing VQA	Give Hand Sign	Custom Sign for एक	एक	एक	Pass
TC03	Testing VQA	Give Hand Sign	Custom Sign for लाना	लाना	लाना	Pass
TC04	Testing VQA	Give Hand Sign	Custom Sign for दो	दो	दो	Pass
TC05	Testing VQA	Give Hand Sign	Custom Sign for ठीक है	ठीक है	ठीक है	Pass
TC06	Testing VQA	Give Hand Sign	Custom Sign for अमन	अमन	दो	Fail
TC07	Testing VQA	Give Hand Sign	Custom Sign for ख़ूबसूरत	ख़ूबसूरत	ख़ूबसूरत	Pass
TC08	Testing VQA	Give Hand Sign	Custom Sign for आलसी	आलसी	आलसी	Pass
TC09	Testing VQA	Give Hand Sign	Custom Sign for तुम	तुम	तुम	Pass
TC10	Testing VQA	Give Hand Sign	Custom Sign for वहाँ	वहाँ	वहाँ	Pass

Figure 6.3: Screenshot of Random Forest Classifier model Manual Testing - I

CHAPTER 7

CONCLUSION

In conclusion, the development of the Vaani Indian Sign Language (ISL) translation system involved a comprehensive exploration of various machine learning approaches, including Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs), and Long Short-Term Memory (LSTM) networks. While SVMs and CNNs demonstrated effectiveness in handling complex data and identifying spatial patterns, the dynamic and sequential nature of sign language gestures necessitated more sophisticated techniques.

The LSTM network emerged as a standout performer, achieving an impressive 80% accuracy on a comprehensive dataset of ISL gestures. Its ability to capture long-range temporal dependencies and contextual information, with adaptive memory mechanisms, allowed the LSTM to outperform other methods for this sequential modeling task.

Building upon these promising results, the implementation of random forest model has further elevated the system's performance, attaining accuracies in the range of 85-90%. The random forest's ensemble approach, constructing multiple decision trees trained on randomized feature subsets, has proven adept at learning the intricate patterns and decision boundaries characteristic of diverse sign language gestures.

A significant milestone is the system's capability to translate recognized gestures directly into Hindi words, rather than merely outputting English characters. This enhancement significantly improves accessibility and intuitiveness for native Hindi speakers, fostering more effective communication and inclusivity within the deaf and non-verbal communities.

Future Enhancement:

Concerning future work, efficient ways for integrating depth information that will guide the feature extraction training phase, can be devised. While the LSTM and random forest models have demonstrated considerable success, ongoing efforts aim to further refine the system's performance by addressing challenges such as occlusion handling and incorporating multi-modal data from depth sensors. By continuously exploring innovative techniques and leveraging the latest advancements in machine learning, this research endeavor holds immense potential to revolutionize sign language translation, bridging communication gaps and empowering individuals with hearing or speech impairments. Some of the further enhancements include:

1. Text To Speech translation.
2. Implement the model on other local Indian languages.

REFERENCES

[1] Nikolas Adaloglou, Theocharis Chatzis, Ilias Papastratis, Andreas Stergioulas and Petros Daras , “Comprehensive Study on Sign Language Recognition Methods”, Senior Member, IEE, University of Patras, Centre for Research and Technology Hellas, 2020.

[2] Toro-Ossaba, A.; Jaramillo-Tigreros, J.; Tejada, J.C.; Peña, A.; López-González, A.; Castanho, R.A., “LSTM Recurrent Neural Network for Hand Gesture Recognition Using EMG Signals”. Appl. Sci. 12, 9700. https://doi.org/10.3390/app12199700, 2022.

[3] Papatsimouli, M.; Sarigiannidis, P.; Fragulis, G.F., “A Survey of Advancements in Real-Time Sign Language Translators: Integration with IoT Technology”. Technologies 11, 83. https://doi.org/10.3390/technologies11040083, 2023.

[4] Zhang Y, Peng L, Ma G, Man M and Liu S, “Dynamic Gesture Recognition Model Based on Millimeter-Wave Radar With ResNet-18 and LSTM". Front. Neuro Robot. 16:903197, doi: 10.3389/fnbot.2022.9 , 2022.

[5] Wu, Y.; Huang, T. “Vision-Based Gesture Recognition: A Review. In Gesture-Based Communication in Human-Computer Interaction”, Springer: Berlin/Heidelberg, Germany, pp. 103–115, 1999.

[6] Johnston, T.; Schembri, A. “Australian Sign Language (Auslan): An Introduction to Sign Language Linguistics”, Cambridge University Press: Cambridge, UK, 2007.

[7] Emmorey, K. “Language, Cognition, and the Brain: Insights from Sign Language Research”, Lawrence Erlbaum Associates Publishers: Mahwah, NJ, USA, 2002.

[8] Wijayawickrama, R.; Premachandra, R.; Punsara, T.; Chanaka, A. “Iot based sign language recognition system”. Glob. J. Comput. Sci. Technol, 20, 39–44, 2020.

[9] Sutton-Spence, R.; Woll, B. “Linguistics and Sign Linguistics. In The Linguistics of British Sign Language: An Introduction”, Cambridge University Press: Cambridge, UK, 19 pp. 1–21, 1999.

[10] Maalej , Z. “Book Review: Language, Cognition, and the Brain: Insights from Sign Language Research”. Linguist List. http://www.linguistlist.org/issues/13/13-1631.html, 2002.

[11] Tervoort, B.T. “Sign language: The study of deaf people and their language”: J.G. Kyle and B. Woll, Cambridge, Cambridge University Press,. ISBN 521 26075. ix+318 pp. Lingua 1986, 70, 205–212, 1985.

[12] Stokoe, W.C. “Jr. Sign language structure: An outline of the visual communication systems of the American deaf ” J. Deaf. Stud. Deaf. Educ. , 10, 3–37, 2005.

[13] Papatsimouli, M.; Lazaridis, L.; Kollias, K.F.; Skordas, I.; Fragulis, G.F. “Speak with Signs: Active Learning Platform for Greek Sign Language, English Sign Language, and Their Translation”. In SHS Web of Conferences; EDP Sciences: Les Ulis, France; Volume 102, p. 01008, 2020.

[14] Papatsimouli, M.; Kollias, K.F.; Lazaridis, L.; Maraslidis, G.; Michailidis, H.; Sarigiannidis, P.; Fragulis, G.F. “Real Time Sign Language Translation Systems: A review study”, In Proceedings of the 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 8–10 June 2022; pp. , 2022.

[15] Shubhankar, B.; Chowdhary, M.; Priyadharshini, M. “IoT Device for Disabled People”, Procedia Comput. Sci. , 165, 189–195, 2019.

[16] Shukor, A.Z.; Miskon, M.F.; Jamaluddin, M.H.; bin Ali, F.; Asyraf, M.F.; bin Bahar, M.B. “A new data glove approach for Malaysian sign language detection”. Procedia Comput. Sci. , 76, 60–67. 2015.

[17] Akmeliawati, R.; Ooi, M.P.L.; Kuang, Y.C. “Real-Time Malaysian Sign Language Translation Using Colour Segmentation and Neural Network”. In Proceedings of the 2007 IEEE Instrumentation & Measurement Technology Conference IMTC , Warsaw, Poland, 1–3 May 2007; pp. 1–6. 2007.

[18] Zhao, S.; Chen, Z.h.; Kim, J.T.; Liang, J.; Zhang, J.; Yuan, Y.B. “Real-Time Hand Gesture Recognition Using Finger Segmentation”. Sci. World J., 267872, 2014.

[19] Sumita, E.; Akiba, Y.; Doi, T.; Finch, A.; Imamura, K.; Paul, M.; Watanabe, T. “A Corpus-Centered Approach to Spoken Language Translation”. In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, 12–17; pp. 171–174, 2003.

[20] Aarssen, A.; Genis, R.; van der Veeken, E. " (Eds.) A Bibliography of Sign Languages”, 2008–2017: With an Introduction by Myriam Vermeerbergen and Anna-Lena Nilsson; Brill: Aylesbury, UK, 2018.

[21] Md. Manik Ahmed, Md. Anwar Hossain, A F M Zainul Abadin, & A F M Zainul Abadin: “Implementation and Performance Analysis of Different Hand Gesture Recognition Methods”. Global Journal of Computer Science and Technology, 19(D3), 13–20, . https://gjcst.com/index.php/gjcst/article/view/510, 2019.

[22] T. G. Zimmerman, "A Hand Gesture Interface Device", Proc. Human Factors in Computing Systems and Graphics Interface, pp. 189-192, 1987-April.

[23] J. Kramer and L. Leifer, “The Talking Glove: An Expressive and Receptive Verbal Communication Aid for the Deaf Deaf-Blind and Non-vocal”, 1989.

[24] Salim, S.; Jamil, M.M.A.; Ambar, R.; Wahab, M.H.A., “A Review on Hand Gesture and Sign Language Techniques for Hearing Impaired Person”. In Machine Learning Techniques for Smart City Applications: Trends and Solutions; Hemanth, D.J., Ed.; Springer: Cham, Switzerland; pp. 35–44, 2022.

[25] Johnson, C.J.; Beitchman, J.H.; Brownlie E,: “Twenty-Year Follow-up of Children with and without Speech-Language Impairments”: Family, Educational, Occupational, and Quality of Life Outcomes. Am. J. Speech-Lang. Pathol. 19, 51–65, 2010.

[26] Webb, S.J.; Jones, E.J.; Kelly, J.; Dawson, G., “The Motivation for Very Early Intervention for Infants at High Risk for Autism Spectrum Disorders”. Int. J. Speech-Lang. Pathol. 16, 36–42, 2014.

[27] Abedin, T.; Prottoy, K.S.; Moshruba, A.; Hakim, S.B., “Bangla Sign Language Recognition Using Concatenated BdSL Network”. arXiv 2021, arXiv:2107.11818, 2021.