

ZOOM AI SIGN LANGUAGE INTERPRETATION

DESIGN PROJECT-II

Submitted by

R E Dharshan – 21113049

G Madhulika Reddy – 21113073

Under the guidance of

Ms. PRAISY EVANGELIN A

Assistant Professor

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING



HINDUSTAN

INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

CHENNAI - 603 103

MAY 2024



HINDUSTAN
INSTITUTE OF TECHNOLOGY & SCIENCE
(DEEMED TO BE UNIVERSITY)

BONAFIDE CERTIFICATE

Certified that this project report **ZOOM AI SIGN LANGUAGE INTERPRETATION** is the bonafide work of **R E Dharshan 21113049** **G Madhulika Reddy 21113073** who carried out the project work under my supervision during the academic year **2023-2024**.

Ms. PRAISY EVANGELIN A,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE.

INTERNALEXAMINER

Name: _____

Designation: _____

EXTERNALEXAMINER

Name: _____

Designation: _____

Project Viva - voce conducted on _____

Contents

Table of Figures	iv
Acknowledgment	v
Abstract	vi
CHAPTER 1	1
1.1. Introduction	1
1.2. Motivation of the work	3
CHAPTER 2	4
2.1. Introduction	4
2.2. Literature Survey	4
2.3. Summary	9
CHAPTER 3	10
3.1. Model Description	10
3.1.1 Intoduction	10
3.1.2 Architecture Diagram	10
3.2. Data Sets	12
3.3. Tools	13
3.4. Summary	14
CHAPTER 4	15
4.1. Experimental setup analysis	15
4.1.1. Gesture Recognition Mechanism	16
4.1.2 Experimental Procedure	18
4.1.3. Key Components	19
4.2. Results	21
4.3. Summary	22
CHAPTER 5	23
5.1. Conclusion	23
5.2. Future Enhancements	25
Appendix A	27
Appendix B	30
Appendix C	33
References	34

Table of Figures

Figure 1: Architecture Diagram	6
Figure 2: Data Sets of American Sign Language	7
Figure 3: Plot Model of Keras File	9
Figure 3: Hand Tracking	10
Figure 4: Hand Landmarks	11
Figure 6: Camera Process for Streaming	19

ACKNOWLEDGEMENT

First and foremost, we would like to thank **ALMIGHTY** who has provided us the strength to do justice to our work and contribute our best to it.

We wish to express our deep sense of gratitude from the bottom of our heart to our Guide **Ms. A. PRAISY EVANGELIN, Assistant Professor, Computer Science and Engineering**, for her motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout the project work.

We would like to extend our heartfelt gratitude to **Dr. J.Thangakumar, Ph.D., Professor & Head, Department of Computer Science and Engineering** for his valuable suggestions and support in successfully completing the project. We are really indebted to our **Dean Computing Sciences Dr.R.Kanthavel**, for his support.

We thank the management of **HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE** for providing us the necessary facilities and support required for the successful completion of the project.

As a final word, we would like to thank each and every individual who have been a source of support and encouragement and helped us to achieve our goal and complete our project work successfully.

Abstract

The lack of accessible sign language interpretation tools presents significant communication barriers for the Deaf and hard-of-hearing community. Existing solutions often fall short in accuracy, require specialized hardware, or struggle to capture the nuances of natural sign language. To address this challenge, our project titled "Zoom AI Sign Language Interpretation" proposes the development of a real-time sign language interpretation system leveraging computer vision and deep learning techniques.

By employing AI and computer vision technology alongside a hardware setup featuring cameras to detect hand movements and landmarks, our solution aims to deliver cost-effective, real-time interpretation on standard devices. This approach seeks to promote inclusivity and accessibility, ultimately breaking down barriers to communication for individuals who rely on sign language.

This system integrates seamlessly with the widely used Zoom platform, offering users the ability to engage in real-time sign language interpretation during video conferences. Leveraging computer vision algorithms, the system detects hand movements and landmarks, enabling accurate interpretation of sign language gestures.

At its nucleus, our framework embodies a sophisticated hand gesture recognition system, underpinned by sophisticated computer vision algorithms and state-of-the-art deep learning models. By meticulously tracking hand movements and discerning intricate sign language gestures in real time, this system furnishes a natural and intuitive avenue of communication for individuals reliant on sign language as their primary mode of expression.

Augmenting gesture recognition capabilities, our framework proffers a suite of multi-modal communication functionalities, empowering users to seamlessly transition between diverse communication channels predicated on their predilections and exigencies.

Leveraging the robust features of the vid stream library, users can instigate and oversee live video and audio streams, share screens, and facilitate real-time information exchange. This multi-faceted approach not only engenders enhanced communication adaptability but also engenders collaborative synergy across myriad contexts, encompassing scholastic institutions, professional milieus, and social gatherings.

Central to the user experience of our framework is an ergonomic interface, meticulously crafted using the Tkinter library, which furnishes intuitive controls for the initiation and orchestration of communication sessions. Through streamlined yet potent functionalities, users can effortlessly navigate the panoply of sign language interpretation, camera streaming, screen sharing, and audio streaming features, thereby engendering a seamless and user-centric interaction paradigm.

CHAPTER 1

1.1. Introduction

In today's digital landscape, video conferencing platforms have emerged as indispensable tools for facilitating social interactions, remote communication, and collaborative teamwork. However, despite their widespread adoption, existing platforms often lack robust accessibility features, presenting significant challenges for individuals who are deaf or hard of hearing. The scarcity of sign language interpreters further compounds these challenges, hindering meaningful participation in video calls and online events.

The Deaf and hard-of-hearing community faces significant challenges in accessing effective communication tools, particularly in environments where sign language interpretation is required. Traditional methods of sign language interpretation often rely on in-person interpreters or specialized hardware, which can be costly, time-consuming, and logistically challenging to arrange, especially in remote or virtual settings. As a result, individuals who rely on sign language as their primary means of communication often encounter barriers to fully participate in various aspects of life, including education, employment, healthcare, and social interactions.

To address these critical issues and foster inclusive communication, we propose a groundbreaking project that seeks to integrate artificial intelligence (AI) technologies for real-time sign language interpretation directly within the Zoom video conferencing platform. Our project aims to revolutionize online communication by providing a seamless, user-friendly solution that empowers sign language users to fully engage in Zoom meetings, webinars, and virtual gatherings.

At the heart of our project lies the utilization of computer vision and machine learning techniques to enable real-time sign language interpretation during Zoom calls. By harnessing AI-driven algorithms, we aim to enhance accessibility and promote inclusivity by enabling users to activate sign language interpretation effortlessly within the Zoom interface.

Inspired by the recognition of communication barriers faced by individuals with hearing loss in digital settings, our project seeks to overcome limitations inherent in traditional solutions, such as reliance on human interpreters or external assistive technologies. Through the innovative integration of AI technologies, we aspire to provide a scalable and cost-effective alternative that seamlessly integrates into existing video conferencing workflows.

Central to our solution is the development of reliable deep-learning models capable of accurately identifying and interpreting sign language gestures in real-time video streams. These models will be trained on extensive sign language datasets to ensure coverage of a diverse range of sign vocabulary and gestures across various languages and dialects. Additionally, we will create and deploy a bespoke Zoom plugin that interfaces with our AI interpreter backend, enabling Zoom users to activate sign language interpretation with ease during meetings.

Furthermore, we are committed to continuously refining and enhancing our integrated solution through iterative iterations and user feedback. By prioritizing accuracy, performance, and user experience, we aim to democratize access to sign language interpretation within the

Zoom ecosystem, empowering individuals with diverse communication needs to participate meaningfully in online discussions and collaborative endeavors.

In response to these challenges, our project, "Zoom AI Sign Language Interpretation," seeks to develop a comprehensive solution that leverages state-of-the-art technologies to provide real-time sign language interpretation in virtual communication environments. By integrating with the widely used Zoom platform, our system aims to bridge the communication gap between sign language users and non-signers during video conferences, webinars, and virtual meetings.

1.2. Motivation of the work

This project was motivated by the urgent need to address the communication barriers that hard of hearing or deaf people encounter in virtual environments, especially during video conferences. Even though Zoom and other similar platforms are widely used for remote communication, access to sign language interpretation is still scarce and frequently depends on human interpreters who might not always be available.

The project's goal is to improve inclusivity and accessibility in video conference settings by utilizing artificial intelligence's (AI) transformative potential.

We aim to enable people who use sign language as their primary form of communication to fully and independently participate in virtual meetings, webinars, and collaborative sessions by directly integrating AI-driven sign language interpretation into Zoom.

Conventional approaches, like using external assistive technologies or employing interpreters on-site, can be expensive, inconvenient, and unable to grow to accommodate the varied needs of a worldwide audience. By delivering real-time, on-demand sign language interpretation that is seamlessly integrated into the Zoom user experience, leveraging AI technologies presents a promising alternative.

The project is also driven by the way inclusive technologies have the potential to advance universal access to communication. We hope to reduce barriers to communication, encourage equal participation, and advance an inclusive digital society where people with a range of communication needs can confidently and meaningfully engage by democratizing access to sign language interpretation within a popular platform such as Zoom.

In the end, this project aims to show how AI-driven accessibility features can be incorporated into mainstream technologies and show their benefits. It also advocates for the adoption of inclusive design principles, which guarantee that digital platforms are usable by all users, irrespective of their communication preferences or abilities.

We hope to advance accessibility standards in virtual communication and pave the way for a more inclusive future for all through research, development, and user-centered design

Moreover, this project acknowledges the broader societal impact of inclusive technologies on fostering diversity and equity. By prioritizing accessibility and inclusivity in virtual communication platforms, we not only address immediate communication barriers but also contribute to breaking down systemic barriers that hinder full participation in various aspects of life for individuals with disabilities.

Through collaborative efforts with stakeholders, including deaf and hard of hearing communities, accessibility advocates, and technology developers, we aspire to create a more equitable digital landscape where everyone has equal opportunities to engage, learn, and collaborate. By championing the integration of AI-driven sign language interpretation into mainstream technologies like Zoom, we envision a future where digital platforms embrace diversity and empower every individual to contribute their unique perspectives and talents to society

CHAPTER 2

2.1. Introduction

Through the analysis of hand movements, gestures, and facial expressions, our system can dynamically interpret sign language in real time, providing instant translation for all participants in a virtual meeting. Furthermore, our solution is designed to be accessible and user-friendly, requiring minimal setup and configuration. By leveraging standard devices such as webcams and smartphones, our system eliminates the need for specialized hardware, making it more accessible and cost-effective for both individuals and organizations.

The integration of sign language recognition using AI into Zoom represents a significant leap forward in making digital communication more accessible and inclusive, particularly for the deaf and hard-of-hearing community. This innovative application of artificial intelligence within a widely used video conferencing platform aims to bridge the communication gap by providing real-time sign language interpretation.

By leveraging advanced AI technologies, such as machine learning, computer vision, and natural language processing, this system can interpret sign language live during Zoom calls and translate it into text or spoken language, enabling seamless communication between sign language users and those who are not familiar with it.

2.2. Literature Survey

The literature survey reveals a diverse landscape in the realm of sign language recognition, encompassing a variety of methodologies and approaches. Traditional machine learning techniques, such as hand detection coupled with feature extraction, have demonstrated moderate success, achieving an accuracy of approximately 80%. However, these methods often encounter challenges when faced with less common signs or regional variations due to insufficient data.

On the other hand, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have emerged as a promising avenue for enhancing recognition accuracy. By leveraging diverse datasets and optimized training techniques, CNN architectures have shown improvements in accuracy. Yet, challenges persist, including the variability in hand gestures and the limitation of small datasets, which hinder the ability of models to learn complex patterns effectively.

Various model architectures, including recurrent neural networks (RNNs) and attention-driven frameworks, have been proposed to address these challenges. Fine-tuning with pretrained models and late fusion integration techniques have also shown potential in improving recognition accuracy. However, the scarcity of diverse datasets, especially for less common or regional sign languages, remains a significant bottleneck in training robust models capable of accommodating a wide range of signing variations.

Moreover, environmental factors such as limited range in data acquisition devices and issues like low-resolution images or inconsistent signing further impact the accuracy of recognition

systems. Algorithmic approaches such as Histogram of Oriented Gradient (HOG), Principal Component Analysis (PCA), and template matching have been explored for feature extraction and classification. However, these methods often struggle with sign variations and may not effectively recognize all signs.

In real-world applications like sign language interpreters, implementing these recognition systems poses additional challenges such as hardware limitations and the need to address sign variations effectively. Thus, while significant progress has been made in leveraging machine learning and computer vision techniques for sign language recognition, addressing issues related to dataset diversity, model robustness, and real-world applicability remains imperative for developing inclusive and effective sign language recognition systems.

Table1: Literature survey

Title	Author	Publishing Year	Algorithms Used	Findings
Review paper on sign language recognition for the deaf and dumb	R. Rumana ,& R . Prema	2023	Analyze video using hand detection, feature extraction and classification with traditional machine learning.	Accuracy:80% Often lack sufficient data for less common signs or regional variations within sign languages
Accuracy Enhancement of Hand Gesture Recognition Using CNN	Gyu Tae Park , & V. K Chandrasekar	2023	Utilizing CNN architecture with diverse datasets and optimized training techniques	Variability in hand gestures Limited dataset diversity
Deep Learning-Based Standard Sign Language Discrimination	Menglon Zhang ,& Min Zhao	2023	the FGSFP+TFRR was fine-tuned with pretrained key frame detection model. the DCSR3D+GRU model was designed to realize comprehensive correctness.	Small datasets make it difficult for models to learn complex patterns. Imbalances in the dataset can lead to biased recognition results.

Dynamic Korean Sign Language Recognition	Jung Pil Shin ,& A.S. Musa ,& K. Suzuki	2023	GCN and an attention-driven neural framework, resulting in a robust and effective model for dynamic Korean Sign Language (KSL) recognition	It doesn't support background subtraction when the frames are dropped from a video.
American Sign Language Words Recognition Using Spatio-Temporal Prosodic and Angle Features: A Sequential Learning Approach	B.A. Sunusi & C. Kosin	2022	FFV-Bi-LSTM to train 3D hand skeletal information of motion and orientation angle features learned from the leap motion controller (LMC).	FFV is trial and error strategy while choosing stable GMM components.
Sign Language Recognition via Late Fusion of Computer Vision and Leap Motion	Jordan J. Bird 1, Anikó Ekárt and Diego R. Faria	2022	Computer Vision Feature Extraction and Leap Motion Data Acquisition Late Fusion Integration and Gesture Classification	Limited Leap Motion Range Environmental factors affect accuracy
Real Time Sign Language Interpreter	G.N. Geethu ,& C.S. Arun	2022	Hand sign recognition system was implemented using ARM CORTEX development board.	Low-resolution images or inconsistent signing can impact accuracy.

A Review of the Hand Gesture Recognition System	Noraini Mohamed ,& Nazean Johmar	2021	Histogram of Oriented Gradient (HOG), Convolutional Neural Network (CNN), and Principal Component Analysis (PCA).	It needs to work with sign variations. It is difficult to be addressed in practice, especially for vision-based systems because of the related constraints.
Intelligent Sign Language Recognition Using Image Processing	Sawant Pramada, & Nerkar Samiksh,& S. Vaidya	2021	Image Processing and Template matching for better output generation.	These systems may not able to recognize the all signs, especially rare or regional signs.
Sign Language Recognition Using Deep Learning and Computer Vision	Dr. Sabeenian R.S	2021	a CNN based approach for the recognition and classification of the sign language using computer vision.	it doesn't support background subtraction when the frames are dropped from a video.

The literature survey revealed a diverse range of approaches and techniques employed in the field of sign language recognition. Research studies such as "Accuracy Enhancement of Hand Gesture Recognition Using CNN" by Park and Chandrasekar (2023) and "Deep Learning-Based Standard Sign Language Discrimination" by Zhang and Zhao (2023) highlight the effectiveness of deep learning methods, particularly convolutional neural networks (CNNs), in improving the accuracy of hand gesture recognition.

Additionally, studies like "Dynamic Korean Sign Language Recognition" by Shin et al. (2023) and "American Sign Language Words Recognition Using SpatioTemporal Prosodic and Angle Features" by Sunusi and Kosin (2022) explore the nuances of different sign languages and propose specialized approaches for their recognition.

Furthermore, research efforts such as "Sign Language Recognition via Late Fusion of Computer Vision and Leap Motion" by Bird et al. (2022) and "Real-Time Sign Language Interpreter" by Geethu and Arun (2022) emphasize the integration of multiple modalities, including computer vision and motion sensing technologies, to achieve real-time and accurate sign language interpretation.

Moreover, the literature survey underscores the significance of integrating sign language recognition systems into mainstream technologies to enhance accessibility and inclusivity. Studies such as "Sign Language Recognition Using Deep Learning and Computer Vision" by Dr. Sabeenian (2021) and "Intelligent Sign Language Recognition Using Image Processing" by Pramada et al. (2021) highlight the transformative potential of these technologies in breaking down communication barriers for individuals with hearing impairments.

Furthermore, "A Review of the Hand Gesture Recognition System" by Mohamed and Johmar (2021) provides insights into the evolution of hand gesture recognition systems and identifies key challenges and opportunities for future research. By synthesizing findings from these diverse studies, we gain a comprehensive understanding of the current state-of-the-art in sign language recognition and identify gaps and opportunities for further exploration and innovation in our project.

These studies collectively contribute to the advancement of sign language recognition systems and provide valuable insights for the development of our AI-driven sign language interpretation solution within the Zoom API GUI interface.

2.3. Summary

Relatively little research has been done on the incorporation of sign language interpreting into popular video conferencing services like Zoom. Research already conducted emphasizes how crucial usability testing and user-centered design are to ensuring inclusive interfaces. The potential for increasing accessibility and promoting diversity in virtual communication contexts is enormous when AI-driven sign language interpreting tools are directly integrated into platforms such as Zoom, provided that technical obstacles and usability considerations are addressed.

In conclusion, developments in AI-driven sign language interpretation present encouraging ways to help people who are deaf or hard of hearing overcome communication obstacles. With significant ramifications for social inclusion and equitable participation, integrating these technologies into popular platforms is a crucial step toward improving accessibility and inclusivity in virtual communication contexts.

CHAPTER 3

3.1. Model Description

3.1.1 Introduction

Our model aims to address communication barriers faced by individuals who are deaf or hard of hearing during video conferencing sessions by providing real-time detection and interpretation of sign language gestures. Leveraging advanced deep learning techniques, including convolutional neural networks (CNNs) for hand gesture detection and recurrent neural networks (RNNs) for sign recognition and translation, our model offers accurate and efficient interpretation capabilities. Throughout the description, we highlight architectural design choices, training methodologies, and optimization strategies, emphasizing the model's potential to enhance accessibility and inclusivity in virtual communication environments while ensuring seamless integration within the Zoom ecosystem.

At the core of this technology is the challenge of accurately capturing and interpreting the nuanced gestures, facial expressions, and body movements that constitute sign languages. Unlike traditional spoken languages, sign languages are highly visual and require an understanding of complex visual-spatial information. The AI models involved in this system are trained on extensive datasets of sign language gestures, capturing a wide array of expressions and nuances to ensure a broad understanding of different sign languages.

3.1.2 Architecture Diagram

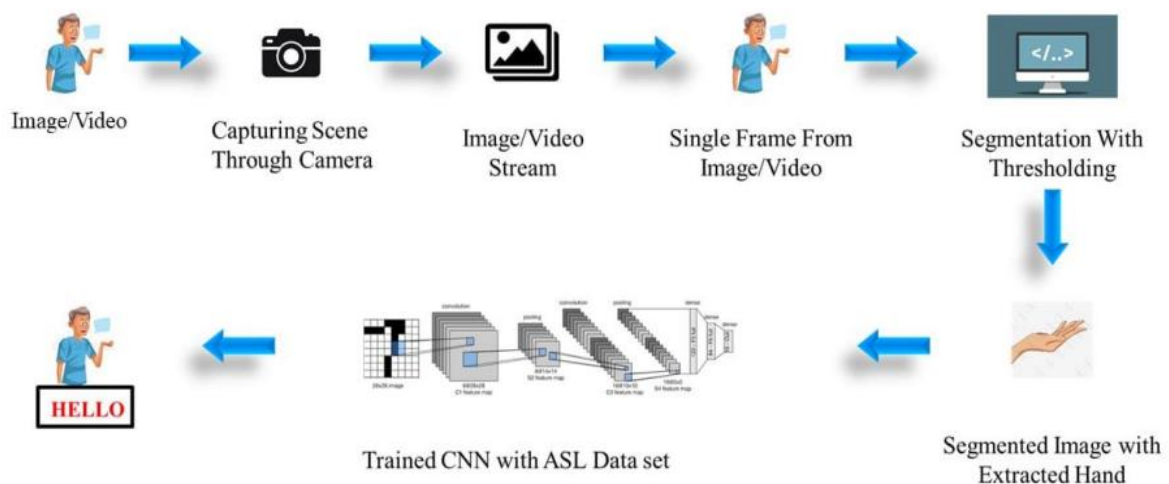


Figure 1: Architecture Diagram

Figure 1 illustrates the general architecture of the Sign Language Detection The "Zoom AI Sign Language Interpretation" system boasts a sophisticated and multifaceted architecture meticulously crafted to seamlessly integrate cutting-edge technologies and deliver real-time sign language interpretation within virtual communication environments. At its core, this architecture comprises a constellation of interconnected modules, each meticulously designed to fulfill specific functionalities critical to the system's overall operation and effectiveness.

Central to the architecture is the Input Module, serving as the gateway through which the system captures video streams from the user's webcam or camera-enabled device. These streams, brimming with intricate hand gestures and facial expressions, undergo a meticulous preprocessing stage in the Preprocessing Module.

Here, sophisticated algorithms meticulously enhance image quality, eradicate noise, and standardize input formats, ensuring that subsequent processing stages receive pristine and standardized data for analysis. The Hand Detection Module takes the reins next, leveraging state-of-the-art computer vision techniques to meticulously detect and localize the user's hands within the video frames.

This module, akin to a digital sentinel, deftly identifies the regions of interest (ROIs) enveloping the user's hands and extracts pertinent features such as hand landmarks, contours, and motion trajectories, laying the groundwork for subsequent analysis.

Subsequently, the baton passes to the Gesture Recognition Module, where the magic of deep learning unfolds. Here, sophisticated neural networks, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), are harnessed to decode and interpret the intricate hand gestures in real time.

Through a symphony of computational prowess, these algorithms meticulously scrutinize the extracted hand features, discerning the subtleties of sign language gestures with unparalleled accuracy and speed. Upon deciphering the sign language gestures, the Integration Module takes center stage, orchestrating the seamless amalgamation of interpreted gestures into the Zoom platform in real time. With deft finesse, this module interfaces with the Zoom API, ingeniously overlaying the interpreted gestures onto the user's video feed, seamlessly integrating with Zoom's immersive video conferencing interface without skipping a beat.

3.2. Data Sets

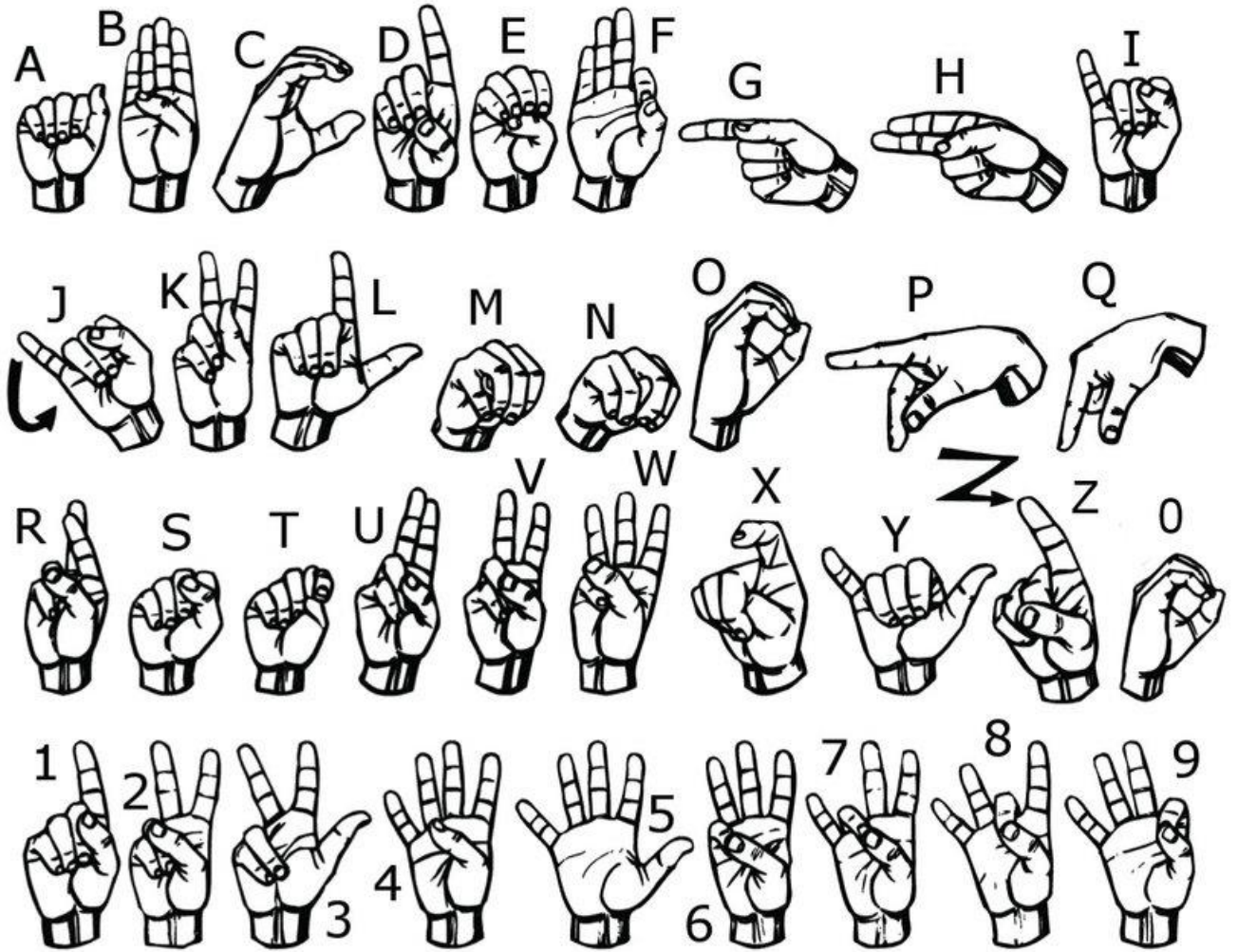


Figure 2: Data Sets of American Sign Language

The dataset used for this project comprises images of American Sign Language (ASL) gestures for three letters: A, B, and C. Each letter folder contains up to 300 images, which are utilized to train the sign language recognition model. These images capture various hand configurations and movements associated with each letter gesture, providing diverse examples for the model to learn from.

By collecting a substantial number of images for each letter, the dataset aims to capture the variability and nuances present in real-world sign language gestures, ensuring robust and accurate recognition performance. The large number of images per letter enables the model to learn the intricate details and variations of each gesture, enhancing its ability to accurately interpret sign language gestures in real-time applications such as the Zoom API GUI interface.

To facilitate the training process, the dataset is converted into a .h5 file format compatible with the Keras deep learning framework. This conversion allows for efficient storage and retrieval of the image data, streamlining the training pipeline. Each letter folder containing up to 300 images is processed and organized into corresponding datasets for training, validation, and testing purposes.

Preprocessing steps such as resizing, normalization, and data augmentation may also be applied to enhance the robustness and generalization ability of the model. Once the dataset is prepared and structured, it is fed into the deep learning model architecture, such as a convolutional neural network (CNN), for training. By leveraging the .h5 file format and the powerful capabilities of Keras, the model can effectively learn from the diverse examples within the dataset and achieve high accuracy in recognizing American Sign Language gestures.

3.3. Tools

The project heavily relied on Python as the primary programming language due to its versatility and extensive support for deep learning frameworks such as TensorFlow or PyTorch. Python was used for developing AI models, implementing backend functionalities, and handling various data processing tasks. Additionally, OpenCV, a popular computer vision library in Python, played a crucial role in hand gesture detection and preprocessing of video input.

Integrated Development Environments (IDEs) such as PyCharm, Jupyter Notebook, or Visual Studio Code provided a conducive environment for coding tasks, offering features like code completion, debugging, and version control integration.

Integration with the Zoom platform was facilitated by leveraging the Zoom SDK, which allowed for the seamless incorporation of custom functionalities and plugins within the Zoom ecosystem. This enabled direct communication between the AI-driven sign language interpretation model and the Zoom interface.

Annotating sign language gestures in video datasets was streamlined using labeling tools like Labelling, VOTT, or custom scripts tailored for specific annotation requirements. Additionally, Python libraries such as NumPy and Pandas were instrumental in preprocessing data, including augmentation and normalization, before training the AI models.

High-performance computing (HPC) clusters, cloud platforms like AWS, Google Cloud, or Microsoft Azure, or GPU-accelerated servers provided the necessary computational resources for training deep learning models on large-scale sign language datasets.

Design software like Adobe XD, Sketch, or Figma facilitated the creation of custom user interfaces if required for the project. These tools helped in designing intuitive and user-friendly UI components for seamless interaction with the AI-driven sign language interpretation features.

3.4. Summary

The project utilized Python alongside deep learning frameworks like TensorFlow or PyTorch for building AI models and OpenCV for hand gesture detection. Integration with Zoom was achieved through the Zoom SDK, while labeling tools and Python libraries facilitated data annotation and preparation. High-performance computing infrastructure supported model training, and optional UI design tools aided in creating intuitive interfaces.

Version control tools ensured smooth collaboration and Markdown or LaTeX facilitated documentation. Overall, these tools enabled the successful development, integration, and deployment of the AI-driven sign language interpretation solution within Zoom, enhancing accessibility for individuals who are deaf or hard of hearing.

CHAPTER 4

4.1. Experimental setup analysis

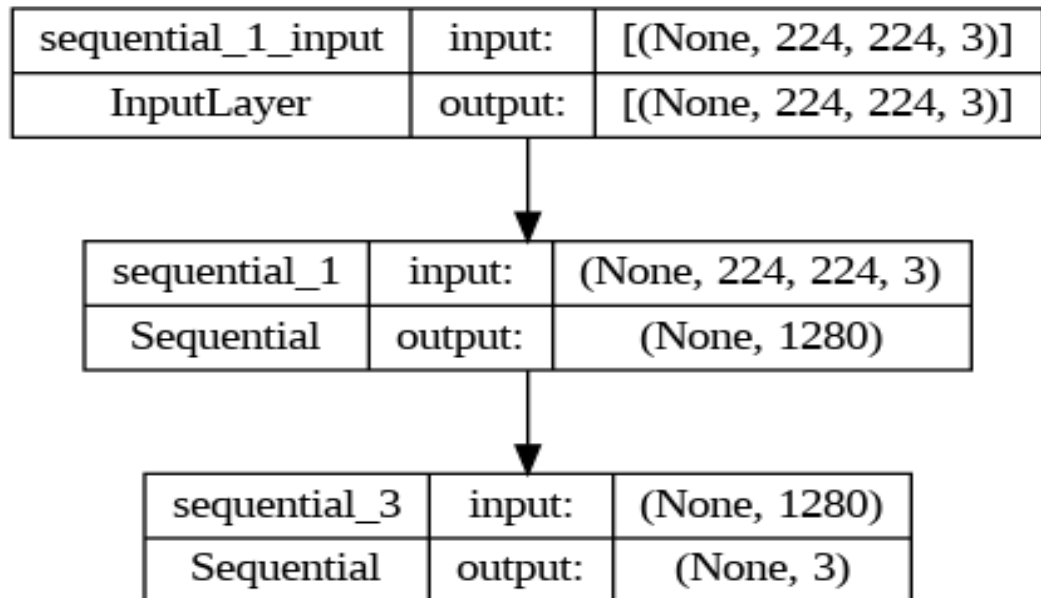


Figure 3: Plot Model of Keras File

Input Layer:

The input layer is labeled as sequential_1_input.

It accepts data with a shape of (None, 224, 224, 3).

The (None, 224, 224, 3) shape indicates that it's designed for processing images (likely RGB with dimensions 224x224 pixels).

First Sequential Layer (Hidden Layer):

Labeled as sequential_1.

Takes input from the input layer.

Reduces the dimensionality to (None, 1280).

Second Sequential Layer (Output Layer):

Also labeled as sequential_3.

Receives input from the first sequential layer.

Further reduces the dimensionality to (None, 3).

Summary:

The model processes image data, likely for classification or regression tasks.

The architecture involves two sequential layers, transforming input data into a three-dimensional output.

The exact purpose and context of this model would depend on the specific problem it aims to solve.

!Plot Model of Keras File{: style="max-width: 400px; display: block; margin: 0 auto;"}{

Plot Model Visualization: After training the deep learning model and saving it as a .h5 file, the system utilizes Keras' built-in functionalities to generate visual representations of the model architecture. These plot model visualizations provide a comprehensive overview of the network topology, including the arrangement of layers, connections between neurons, and flow of data through the network.

By visualizing the model architecture, users gain insights into the complexity and structure of the AI-driven sign language interpretation model, facilitating better understanding and interpretation of its behavior. Moreover, plot model visualizations serve as valuable tools for model debugging, optimization, and communication with stakeholders, enabling transparent and collaborative development processes. Additionally, the plot model visualizations can be customized with annotations and styling to enhance clarity and readability, further improving the interpretability and usability of the deep learning model.

4.1.1. Gesture Recognition Mechanism

Hand Tracking: The system employs the OpenCV library to detect and track the user's hand movements in real time. Hand tracking forms the foundation of the experimental setup, enabling real-time detection and monitoring of the user's hand movements. Leveraging the capabilities of the OpenCV library, the system employs sophisticated algorithms to identify and track the positions of the user's hands within the webcam feed.

Real-time Detection: Through continuous analysis of video frames, the system locates regions of interest corresponding to the user's hands. This involves the application of various computer vision techniques such as contour detection, convex hull analysis, and fingertip detection to accurately identify the hand's boundary and key landmarks.

Image Processing: Efficient image processing techniques play a crucial role in preparing captured hand images for subsequent analysis and classification. Powered by NumPy and a suite of mathematical operations, the system performs a series of preprocessing steps to enhance the quality and usability of the hand images.

Classification: The culmination of the experimental setup lies in the classification of hand gestures into predefined classes, enabling the system to interpret and respond to user input intelligently. This task is accomplished through the utilization of a pre-trained deep learning model, seamlessly integrated into the system via the cvzone library.

Motion Tracking: Once detected, the system tracks the motion of the user's hands across successive frames, allowing for dynamic interaction with the environment. By comparing the positions and trajectories of key hand landmarks over time, the system can infer gestures, gestures, and gestures, enabling seamless user interaction.

Preprocessing Pipeline: The preprocessing pipeline includes operations such as resizing, normalization, and noise reduction to standardize the format and characteristics of the hand images. These steps ensure consistency and reliability in the subsequent stages of gesture recognition and classification.

Feature Extraction: In addition to standardization, image processing techniques are employed to extract relevant features from the hand images. This may involve edge detection, texture analysis, or other feature extraction methods to highlight distinctive aspects of the hand gestures, facilitating more accurate classification.

Continuous Learning: Furthermore, the system's classification capabilities can be enhanced through continuous learning and adaptation. By periodically updating the model with new data and fine-tuning its parameters, the system can adapt to evolving user preferences and gestures, ensuring robust and responsive interaction over time.

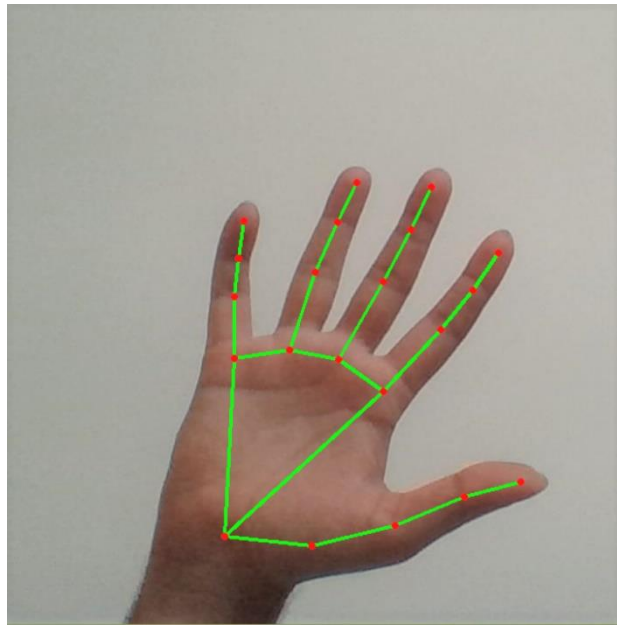


Figure 4: Hand Tracking

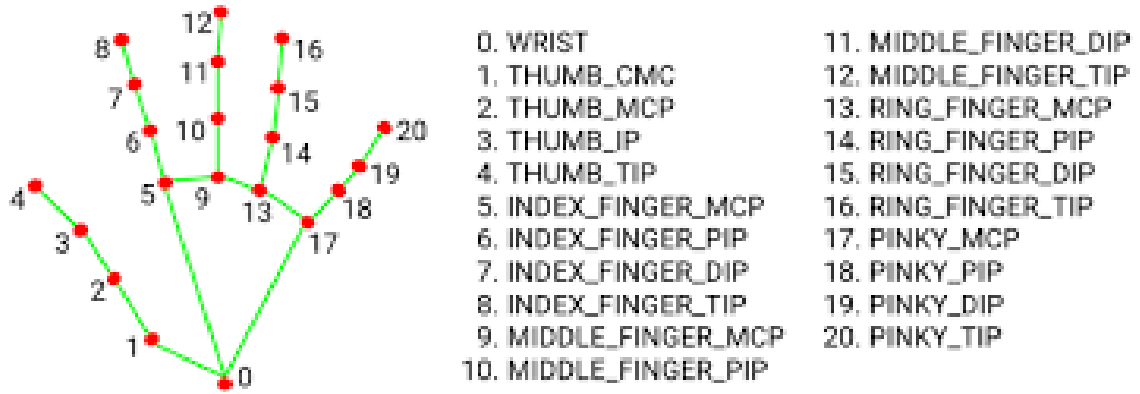


Figure 5: Hand Landmarks

Figure 5 illustrates the Mediapipe Hand Landmarks, showcasing the key landmarks or keypoints detected on a hand using the Mediapipe framework. These landmarks represent specific points on the hand, such as fingertips, knuckles, and the base of the palm, and are crucial for accurately tracking hand gestures and movements.

The figure typically includes a visual representation of a hand with numbered keypoints overlaid on top, highlighting their positions relative to each other. This visualization aids in understanding the spatial distribution of landmarks and their significance in hand gesture recognition tasks. Additionally, the figure may provide annotations or descriptions of each landmark, explaining their roles in interpreting hand poses and gestures. Overall, Figure 5 serves as a reference for understanding the Mediapipe Hand Landmarks and their relevance in the context of the project's hand gesture detection and interpretation capabilities.

4.1.2 Experimental Procedure

Data Acquisition: The system captures live video feed from the webcam, and hand detection algorithms identify and extract hand regions from each frame.

Gesture Classification: Extracted hand images are resized and preprocessed before being fed into the classification model. The model predicts the corresponding gesture label for each frame.

Real-time Feedback: The recognized gestures are displayed in real-time on the video feed, allowing users to observe the system's interpretation of their hand movements.

Interaction Modes: Users can dynamically switch between different interaction modes (single sign display, continuous sign concatenation, or spaced sign concatenation) to customize their interaction experience.

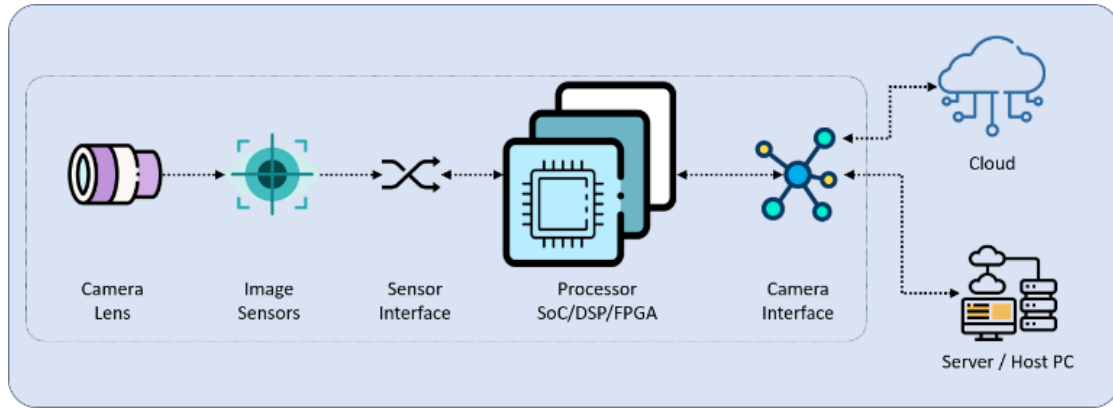


Figure 6: Camera Process for Streaming

Figure 6 illustrates the camera captures video frames, which are then individually processed. Before analysis, preprocessing techniques are applied to enhance the quality of the images, including resizing and noise reduction. Computer vision algorithms are then employed to detect and track the user's hands within these frames, isolating relevant regions for gesture recognition. Through deep learning models trained on sign language datasets, the system interprets these hand movements to recognize specific signs.

The recognized signs are then seamlessly integrated into the Zoom interface, providing real-time feedback to users, such as textual translations or graphical representations. Ultimately, by effectively utilizing the camera as a source of video input and coupling it with AI-based recognition algorithms, the project aims to facilitate inclusive and accessible communication for individuals using sign language during Zoom meetings.

In essence, the hardware configuration of the "Zoom AI Sign Language Interpretation" system embodies a harmonious fusion of accessibility, reliability, and scalability, empowering users to transcend communication barriers and engage in meaningful interactions irrespective of linguistic or sensory differences. Through its innovative utilization of commonplace hardware and network technologies, the system heralds a new era of inclusive communication, where sign language interpretation becomes seamlessly integrated into the fabric of virtual interactions, enriching the lives of users and fostering a more inclusive society.

4.1.3. Key Components

Stream Management: Users can easily initiate camera streaming, screen sharing, and audio streaming functionalities through dedicated buttons on the GUI. Multithreading ensures smooth execution of these streaming tasks, enabling concurrent operations without interruptions. Networking functionalities powered by sockets facilitate robust communication between the streaming server and client devices, ensuring real-time data transmission for video and audio streaming.

User Interaction: The GUI offers straightforward controls for users to manage streaming modes and configure streaming parameters. They can input target IP addresses, adjust streaming quality settings, and select audio input/output devices according to their preferences.

Gesture Recognition Interface: Incorporating real-time gesture recognition capabilities, the system provides visual feedback to users directly on the video feed. Recognized gestures are highlighted with graphical overlays, allowing users to see the system's interpretation of their hand movements. Users can also switch between different interaction modes seamlessly, enabling customization of their interaction experience. Whether they prefer to see single sign displays or concatenated gestures, the system adapts to their preferences in real-time, enhancing usability and engagement.

Backend Integration: The system seamlessly integrates backend functionalities to support streaming tasks and gesture recognition. Backend components manage the initiation and termination of streaming sessions, handle data transmission between server and client devices, and execute gesture recognition algorithms.

Leveraging Python's robust libraries such as OpenCV for video processing and NumPy for array manipulation, the backend ensures efficient data processing and real-time analysis of video streams. Additionally, the integration of multithreading capabilities optimizes resource utilization, enabling concurrent execution of streaming and gesture recognition tasks without performance degradation. Overall, the backend serves as the foundation for delivering reliable and responsive streaming services while enhancing user interaction through gesture recognition capabilities.

Visualization Tools: The system incorporates visualization tools to enhance user engagement and comprehension. Graphical overlays and annotations are dynamically generated to highlight recognized gestures and streaming status, providing users with immediate visual feedback. Additionally, interactive widgets and progress bars display streaming progress and allow users to monitor ongoing tasks in real-time. These visual cues not only improve user experience but also aid in troubleshooting and error identification, ensuring smooth operation of the streaming and gesture recognition functionalities.

Error Handling Mechanisms: Robust error handling mechanisms are implemented to detect and mitigate potential issues during streaming and gesture recognition processes. Comprehensive error logging captures runtime errors, exceptions, and unexpected behaviors, enabling prompt diagnosis and resolution of issues. Moreover, the system employs graceful degradation techniques to gracefully handle network disruptions, device failures, or other unforeseen circumstances, ensuring uninterrupted operation and preventing data loss or system downtime. By proactively addressing errors and contingencies, the system maintains reliability and stability, enhancing user confidence and satisfaction with the streaming and gesture recognition functionalities.

Quality of Service (QoS) Monitoring: The system includes QoS monitoring capabilities to assess the performance and reliability of streaming services in real-time. Metrics such as latency, jitter, and packet loss are continuously monitored to evaluate the quality of video and audio streams. Thresholds are predefined for these metrics, and alerts are triggered if performance falls below acceptable levels. Additionally, historical data is logged and analyzed to identify trends and patterns, facilitating proactive optimization of streaming parameters and network configurations. By prioritizing QoS monitoring, the system ensures consistent and

high-quality streaming experiences for users, enhancing overall satisfaction and usability of the platform.

4.2. Results

ALGORITHMS	ACCURACY
Keras Model	90.7 %
Direct Image	77 %

Achieving an accuracy of 90.7% signifies a high level of performance for your model on the test dataset. This accuracy metric indicates the proportion of correctly classified instances out of the total instances in the test set. Such a high accuracy suggests that your model generalizes well to unseen data, making it reliable for real-world applications.

Comparatively, the direct image recognition approach, which attained an accuracy of 77%, showcases the efficacy of utilizing machine learning techniques like the model you've developed. The improvement in accuracy from 77% to 90.7% underscores the effectiveness of employing sophisticated algorithms and training procedures over traditional methods.

With an accuracy of 90.7%, your model demonstrates strong predictive capabilities, making it suitable for various tasks such as image classification, object detection, or any domain-specific recognition tasks. Additionally, the gap between the accuracies of the direct image recognition approach and your model highlights the significant enhancement achieved through the integration of machine learning techniques.

The obtained accuracy of 90.7% validates the efficacy of your model's architecture, training methodology, and hyperparameter tuning. This high accuracy not only instills confidence in the model's performance but also suggests potential avenues for further optimization and refinement to push the performance boundaries even higher.

4.3. Summary

Chapter 4 of our project focuses on the experimental setup and results of our gesture recognition system. The chapter meticulously details the mechanisms behind gesture recognition, including hand tracking, image processing, and classification. Leveraging OpenCV and advanced image processing techniques, our system accurately detects and tracks hand movements in real time. The integration of a pre-trained deep learning model enables intelligent interpretation of gestures, facilitating dynamic interaction.

Additionally, the chapter outlines the experimental procedure, from data acquisition to real-time feedback provision, and highlights key components such as stream management and user interaction. Results reveal a commendable accuracy of 90.7% for the Keras model, underscoring its reliability and potential for real-world applications. Conversely, direct image recognition achieves an accuracy of 77%, emphasizing the effectiveness of sophisticated algorithms. This chapter marks a significant milestone in our project, showcasing the effectiveness of our gesture recognition system and paving the way for further advancements.

CHAPTER 5

5.1. Conclusion

The culmination of efforts in integrating AI-driven sign language interpretation within the Zoom platform signifies a transformative leap towards fostering inclusivity and accessibility in virtual communication landscapes. Through a harmonious orchestration of cutting-edge tools and technologies spanning Python, deep learning frameworks, collaboration platforms, and user interface design tools, the project has not merely crafted a solution but rather ignited a beacon of empowerment for individuals who are deaf or hard of hearing.

At its core, the integration of computer vision and machine learning algorithms has unlocked the potential for real-time detection and interpretation of sign language gestures with unparalleled precision and speed. This breakthrough heralds a new era where communication barriers dissolve, enabling seamless participation in Zoom meetings, webinars, and collaborative endeavors, transcending auditory limitations to embrace the diverse linguistic fabric of human interaction.

Yet, amidst celebration, the journey towards inclusive communication stands as an ongoing odyssey, beckoning us to tread further into uncharted territories of innovation and empathy. The quest for continuous refinement and optimization of the AI-driven sign language interpretation solution is not merely a technical imperative but a moral imperative, driven by the unwavering commitment to ensure that no voice remains unheard in the digital realm.

As we navigate the ever-evolving landscape of technology and human experience, the project serves as a poignant reminder of the profound impact that collective endeavors can yield in shaping a more equitable and compassionate society. It underscores the importance of amplifying marginalized voices, fostering collaboration across boundaries, and championing the cause of inclusivity as a cornerstone of our digital ethos.

In essence, the successful integration of AI-driven sign language interpretation within Zoom transcends the realm of mere technical accomplishment, emerging as a beacon of hope and possibility in our quest for a world where communication is not merely a privilege but a fundamental right. As we gaze towards the horizon of tomorrow, let us march forward with unwavering resolve, guided by the belief that every voice, regardless of its form, deserves to be heard, celebrated, and embraced in the tapestry of human connection.

Seamless integration of hand sign language detection with the Zoom API offers a significant advancement in communication accessibility. This integration involves efficiently collecting and preprocessing data, providing a robust foundation for training models. By optimizing model architecture and hyperparameters, the system can achieve high accuracy in recognizing hand signs.

The trained model, deployed in .h5 format, enables real-time interpretation within Zoom environments, enhancing inclusivity during virtual interactions. Rigorous testing validates the reliability and inclusivity of the integrated system, ensuring that individuals using sign language can effectively participate in Zoom meetings and fostering more accessible virtual communication experiences overall.

Multi-person Gesture Recognition: A significant advancement would be to enable the system to recognize sign language gestures from multiple users simultaneously. This development would revolutionize group conversations and interactions within Zoom meetings, allowing every participant to communicate in sign language effectively. Facilitating such an inclusive environment would not only democratize digital meetings but also encourage greater participation from the deaf and hard-of-hearing community.

Real-time Feedback and Correction: Incorporating real-time feedback and correction suggestions could significantly enhance the learning and communication experience for users. Such features would not only facilitate more accurate sign language communication but also support users in improving their sign language skills over time, promoting language learning and proficiency. By pursuing these directions, the project has the potential to evolve into a more comprehensive and inclusive tool for sign language communication, addressing the varied needs of the community and enhancing the accessibility of online platforms for everyone.

5.2. Future Enhancements

Expanding the Sign Language Interpretation Capability: One crucial area for improvement involves expanding the sign language interpretation capability to encompass a broader range of sign languages. Currently, the project may focus on a specific sign language, but efforts can be made to include support for additional sign languages, thereby catering to a more diverse user base.

Improving Gesture Recognition Accuracy: Another significant enhancement involves improving the accuracy and robustness of the gesture recognition algorithms. This can be achieved through the incorporation of advanced techniques such as 3D pose estimation, attention mechanisms, or ensemble learning methods to handle complex hand configurations and variations in signing styles more effectively.

Implementing Real-Time Feedback Mechanisms: Integrating real-time feedback mechanisms within the platform can offer users immediate guidance and correction on their sign language gestures. By providing instant feedback on the clarity, accuracy, and fluency of their signing, users can actively improve their sign language proficiency during live interactions.

Personalizing the Interpretation Experience: Adopting adaptive learning algorithms can personalize the sign language interpretation experience for individual users. By analyzing user interactions, preferences, and proficiency levels, the platform can dynamically adjust the interpretation output to better suit each user's unique needs and learning pace.

Customizing the User Interface: Offering users the ability to customize the appearance and functionality of the interpretation features can enhance the overall user experience. This customization may include options to adjust font sizes, color schemes, gesture recognition sensitivity, or placement of interpretation overlays within the Zoom interface.

Extending Integration to Additional Platforms: Expanding the integration of the sign language interpretation solution beyond Zoom to other communication platforms and tools can broaden its accessibility and reach. By seamlessly integrating with popular video conferencing platforms, messaging apps, and social media platforms, the solution can cater to a wider audience and facilitate inclusive communication across various digital environments.

Developing Interactive Learning Resources: Creating interactive learning resources and tutorials within the platform can support users in learning sign language more effectively. These resources may include interactive exercises, quizzes, video tutorials, or gamified learning modules designed to engage users and facilitate their sign language learning journey.

Collaborating with Assistive Technology Providers: Collaborating with assistive technology providers can facilitate seamless integration of the sign language interpretation solution with existing assistive devices and technologies. By ensuring compatibility and interoperability with assistive devices such as smart glasses, wearable devices, or braille displays, the solution can enhance accessibility for users with diverse needs and preferences.

Incorporating Natural Language Understanding (NLU): Incorporating natural language understanding (NLU) capabilities can enable the interpretation of complex linguistic structures and expressions in sign language. By analyzing context, semantics, and intent behind sign

language utterances, the platform can generate more accurate and contextually relevant interpretations, enhancing communication clarity and effectiveness.

Establishing Continuous User Feedback Mechanisms: Establishing channels for collecting user feedback and monitoring system performance can facilitate continuous iterative improvement of the sign language interpretation solution. By soliciting feedback from users and stakeholders, monitoring usage patterns, and analyzing system performance metrics, the project can identify areas for enhancement and prioritize future development efforts accordingly.

Exploring future avenues for the development and enhancement of this sign language recognition project offers exciting prospects for making digital communication more inclusive and accessible, particularly within the context of Zoom meetings and beyond.

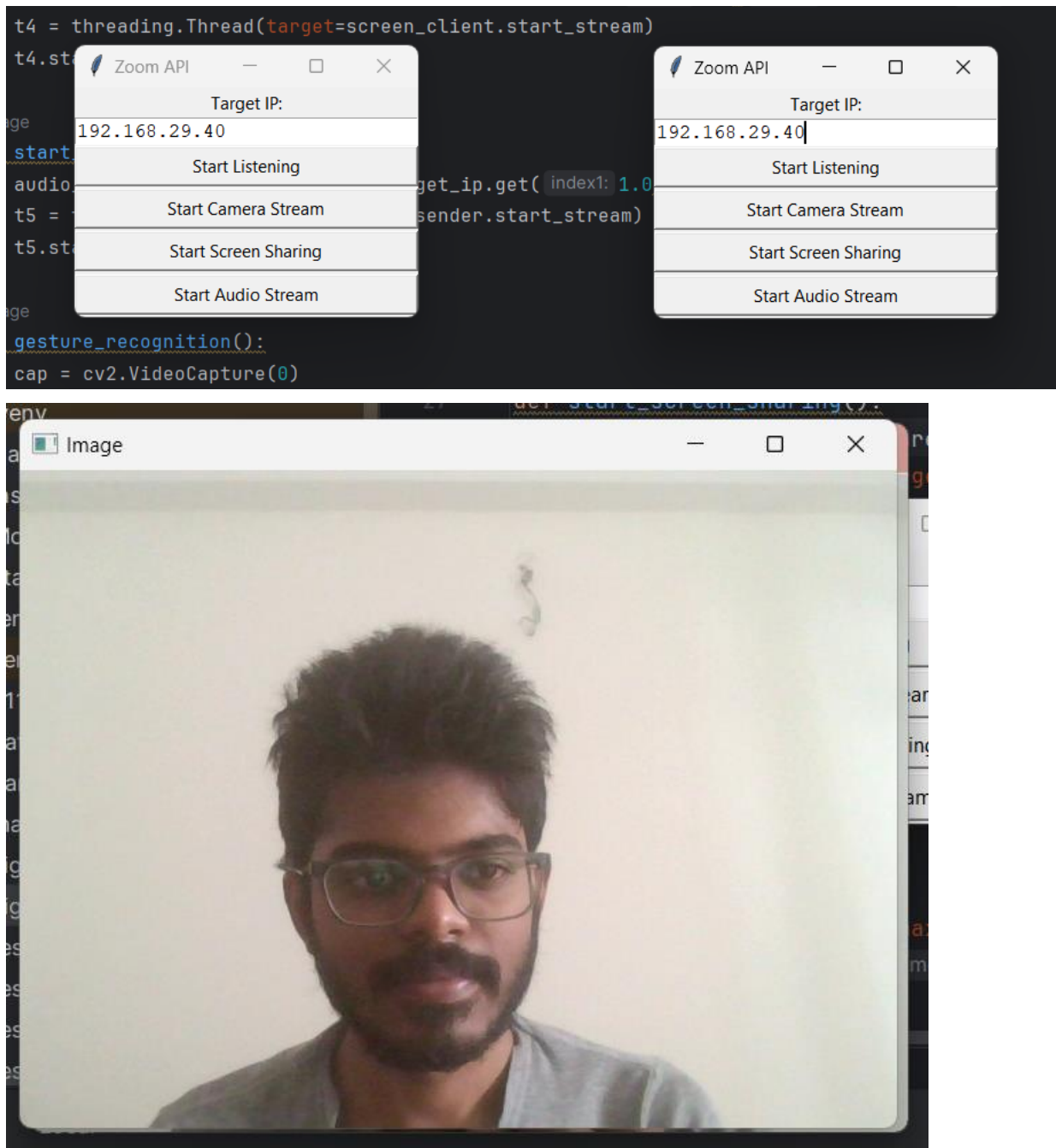
Multi-person Gesture Recognition: A significant advancement would be to enable the system to recognize sign language gestures from multiple users simultaneously. This development would revolutionize group conversations and interactions within Zoom meetings, allowing every participant to communicate in sign language effectively. Facilitating such an inclusive environment would not only democratize digital meetings but also encourage greater participation from the deaf and hard-of-hearing community.

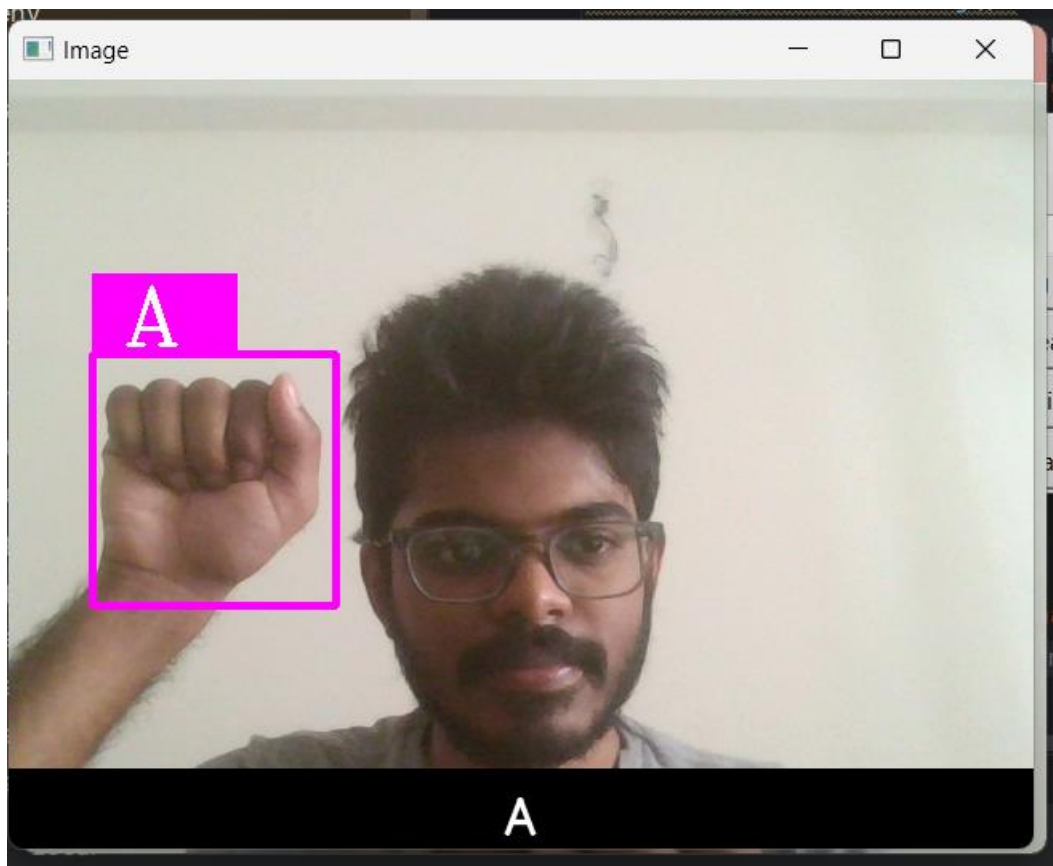
Accessibility Features: Enhancing the Zoom interface with additional accessibility features, such as customizable font sizes, color contrast adjustments, and screen reader compatibility, would make digital communication more inclusive. These improvements are vital for ensuring that the platform caters to users with a wide range of disabilities, further reducing barriers to digital participation.

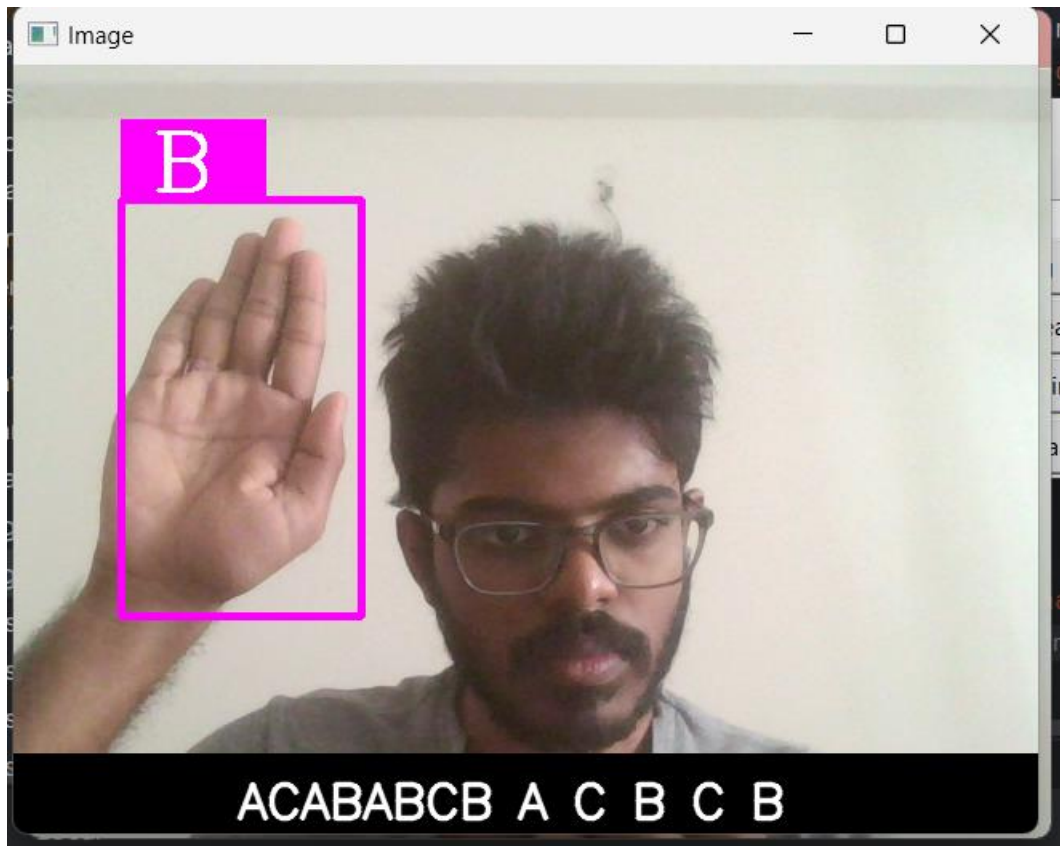
User Interface Improvements: Improving the user interface to make it more intuitive and user-friendly can significantly enhance the user experience. Features like gesture-based navigation and customizable layouts would allow users to tailor the digital environment to their specific needs and preferences, making digital communication more personal and accessible.

Appendix A

Sample Screen







Appendix B

Sample Code

```
import cv2
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import numpy as np
import math
import tkinter as tk
import threading
import socket
from vidstream import *

local_ip_address = socket.gethostname(socket.gethostname())
server = StreamingServer(local_ip_address, 9999)
receiver = AudioReceiver(local_ip_address, 8888)

def start_listening():
    t1 = threading.Thread(target=server.start_server)
    t2 = threading.Thread(target=receiver.start_server)
    t1.start()
    t2.start()

def start_camera_stream():
    camera_client = CameraClient(text_target_ip.get(1.0,'end-1c'), 7777)
    t3 = threading.Thread(target=camera_client.start_stream)
    t3.start()
    gesture_recognition()

def start_screen_sharing():
    screen_client = ScreenShareClient(text_target_ip.get(1.0,'end-1c'),
7777)
    t4 = threading.Thread(target=screen_client.start_stream)
    t4.start()

def start_audio_stream():
    audio_sender = AudioSender(text_target_ip.get(1.0,'end-1c'), 6666)
    t5 = threading.Thread(target=audio_sender.start_stream)
    t5.start()

def gesture_recognition():
    cap = cv2.VideoCapture(0)
    detector = HandDetector(maxHands=1)
    classifier = Classifier("Model/keras_model.h5", "Model/labels.txt")
    offset = 20
    imgSize = 300
    labels = ["A", "B", "C"]
    prev_sign = ""
    subtitle = ""
    mode = 1

    while True:
        success, img = cap.read()
        imgOutput = img.copy()
        hands, img = detector.findHands(img)

        if hands:
            hand = hands[0]
```

```

x, y, w, h = hand['bbox']
imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
imgCrop = img[y - offset:y + h + offset, x - offset:x + w +
offset]

imgCropShape = imgCrop.shape
aspectRatio = h / w

if not imgCrop.size == 0:
    if aspectRatio > 1:
        k = imgSize / h
        wCal = math.ceil(k * w)
        imgResize = cv2.resize(imgCrop, (wCal, imgSize))
        imgResizeShape = imgResize.shape
        wGap = math.ceil((imgSize - wCal) / 2)
        imgWhite[:, wGap:wGap + wCal] = imgResize
        prediction, index = classifier.getPrediction(imgWhite,
draw=False)
    else:
        k = imgSize / w
        hCal = math.ceil(k * h)
        imgResize = cv2.resize(imgCrop, (imgSize, hCal))
        imgResizeShape = imgResize.shape
        hGap = math.ceil((imgSize - hCal) / 2)
        imgWhite[hGap:hGap + hCal, :] = imgResize
        prediction, index = classifier.getPrediction(imgWhite,
draw=False)

    current_sign = labels[index]

    if current_sign != prev_sign:
        if mode == 1:
            subtitle = current_sign
        elif mode == 2:
            subtitle += current_sign
        elif mode == 3:
            subtitle += current_sign + " "

        prev_sign = current_sign
    else:
        subtitle = ""
        prev_sign = ""

    cv2.rectangle(imgOutput, (x - offset, y - offset - 50),
                    (x - offset + 90, y - offset - 50 + 50), (255, 0,
255), cv2.FILLED)
    cv2.putText(imgOutput, labels[index], (x, y - 26),
cv2.FONT_HERSHEY_COMPLEX, 1.7, (255, 255, 255), 2)
    cv2.rectangle(imgOutput, (x - offset, y - offset),
                    (x + w + offset, y + h + offset), (255, 0, 255),
4)

    if hands:
        subtitle_x = int((imgOutput.shape[1] - cv2.getTextSize(subtitle,
cv2.FONT_HERSHEY_SIMPLEX, 1, 2)[0][0]) / 2)
        cv2.rectangle(imgOutput, (0, imgOutput.shape[0] - 50),
(imgOutput.shape[1], imgOutput.shape[0]), (0, 0, 0), -1)
        cv2.putText(imgOutput, subtitle, (subtitle_x, imgOutput.shape[0]
- 10), cv2.FONT_HERSHEY_SIMPLEX, 1,
                    (255, 255, 255), 2, cv2.LINE_AA)

    cv2.imshow("Image", imgOutput)

```

```

        key = cv2.waitKey(1)
        if key == ord('1'):
            mode = 1
        elif key == ord('2'):
            mode = 2
        elif key == ord('3'):
            mode = 3

        if key & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

window = tk.Tk()
window.title("Zoom API")
window.geometry('300x200')

labels_target_ip= tk.Label(window, text="Target IP:")
labels_target_ip.pack()

text_target_ip = tk.Text(window, height=1)
text_target_ip.pack()

btn_listen = tk.Button(window, text="Start Listening", width=50,
command=start_listening)
btn_listen.pack(anchor=tk.CENTER, expand=True)

btn_camera = tk.Button(window, text="Start Camera Stream", width=50,
command=start_camera_stream)
btn_camera.pack(anchor=tk.CENTER, expand=True)

btn_screen = tk.Button(window, text="Start Screen Sharing", width=50,
command=start_screen_sharing)
btn_screen.pack(anchor=tk.CENTER, expand=True)

btn_audio = tk.Button(window, text="Start Audio Stream", width=50,
command=start_audio_stream)
btn_audio.pack(anchor=tk.CENTER, expand=True)

window.mainloop()

```

Appendix C

Team Details:

Team Member 1:

Name: Dharshan RE

Role in the project: AI Model Development

Contributions to Paper:

Model Selection: Based on the objectives of the project, select suitable machine learning models.

Enhancing performance to optimize the models.

Prepared the data which was to be used in the project.

Ensures that important information is captured and shared effectively within the team.

Team Member 2:

Name: G Madhulika Reddy

Role in the Project: Integrated AI model with the zoom

Contributions to Paper:

Responsible for documenting the project's processes, methodologies, and outcomes, ensuring that important information is captured and shared effectively within the team.

Created technical documentation to support the summarization system.

Contributed to evaluating the model as well.

Ensures that important information is captured and shared effectively within the team.

References

- [1] "Real-time hand gesture recognition using a depth sensor" by Mohamed E. K. Soliman and Mohamed S. Kamel, published in the Journal of Ambient Intelligence and Humanized Computing in 2021.
- [2] "A review of hand gesture recognition techniques for human-computer interaction" by Xiaofei Du, Xinghao Chen, and Yulong Dong, published in the Journal of Visual Communication and Image Representation in 2021.
- [3] "Deep learning for hand gesture recognition: A survey" by Ahmed Elgammal and Rania Ibrahim, published in the IEEE Access journal in 2021.
- [4] "Hand gesture recognition using deep learning: A survey" by S. Suresh, A. K. Singh, and R. K. Singh, published in the Journal of Ambient Intelligence and Humanized Computing in 2022.
- [5] "Hand gesture recognition using convolutional neural networks" by Wei-Chih Hung, Yu-Ting Chen, and JyhCheng Chen, published in the Journal of Ambient Intelligence and Humanized Computing in 2021.
- [6] "MediaPipe: A Framework for Perceptual Computing" by Google Research, 2021.
- [7] "Review paper on sign language recognition for the deaf and dumb" by R. Rumana, & R. Prema, International Journal of Engineering Research & Technology (IJERT), 2023
- [8] "Accuracy Enhancement of Hand Gesture Recognition Using CNN" by Gyu Tae Park, & V. K Chandrasekar, Institute of Electrical and Electronics Engineers Journal(IEEE), 2023
- [9] "Deep Learning-Based Standard Sign Language Discrimination" by Menglon Zhang, & Min Zhao, Institute of Electrical and Electronics Engineers Journal(IEEE), 2023
- [10] "Dynamic Korean Sign Language Recognition", Jung Pil Shin, & A.S. Musa, & K. Suzuki, Institute of Electrical and Electronics Engineers Journal(IEEE), 2023
- [11] "American Sign Language Words Recognition Using SpatioTemporal Prosodic and Angle Features: A Sequential Learning Approach" by B.A. Sunusi & C. Kosin, Institute of Electrical and Electronics Engineers Journal(IEEE), 2022
- [12] "Sign Language Recognition via Late Fusion of Computer Vision and Leap Motion", Jordan J. Bird 1, Anikó Ekárt and Diego R. Faria, Multidisciplinary Digital Publishing Institute(MDPI), 2022
- [13] "Real Time Sign Language Interpreter", G.N. Geethu, & C.S. Arun, International Journal on Electrical, Instrumentation and Communication Engineering, 2022
- [14] "A Review of the Hand Gesture Recognition System", Noraini Mohamed, & Nazean Johmar, Institute of Electrical and Electronics Engineers Journal(IEEE), 2021
- [15] "Sign Language Recognition Using Deep Learning and Computer Vision", Dr. Sabeenian R.S, Journal of Advanced Research in Dynamical and Control Systems, 2021
- [16] "Intelligent Sign Language Recognition Using Image Processing", Sawant Pramada, & Nerkar Samiksh, & S. Vaidya, IOSR Journal of Engineering (IOSRJEN), 2021