TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A

PROJECT REPORT

ON

TWO WAY SIGN LANGUAGE COMMUNICATION APP FOR DEAF AND DUMB

SUBMITTED BY:

SAMIR PAUDYAL (076BCT065)

SHREESHANT PRAJAPATI(076BCT077)

SUBIGYA SHRESTHA (076BCT084)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

APRIL, 2024

Page of Approval

TRIBHUVAN UNIVERSIY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certifies that they have read and recommended to the Institute of Engineering for acceptance of a project report entitled **"Two Way Sign Language Communication App for Deaf and Dumb"** submitted by **Samir Paudyal**, **Shreeshant Prajapati**, **Subigya Shrestha** in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Computer Engineering.

.....
Supervisor
Dr. Aman Shakya
Assistant Professor
Department of Electronics
Computer Engineering, Pulchowk Campus, IOE, TU.

.....
External examiner
Manish Modi
Director
Khalti

Date of approval: April, 2024

Copyright

Acknowledgments

We extend our sincere gratitude to the Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, for accepting our concept note titled *"Two Way Sign Language Communication App for Deaf and Dumb"* This opportunity to further develop our project is truly appreciated.

Our heartfelt appreciation goes to Dr. Aman Shakya for their unwavering support and motivation, inspiring us to explore new horizons in our field of study.

With continued guidance, we're excited to contribute significantly to our field. Thank you to the Department of Electronics and Computer Engineering for this incredible opportunity.

Authors:
Samir Paudyal
Shreeshant Prajapati
Subigya Shrestha

Abstract

The easiest and most general language for communication in the world is speech. But in the case of speech-impaired and hearing-impaired people, it becomes impossible. Rather they use sign language for communication. But, even they have to face difficulty whenever they need to communicate with those people who don't understand sign language. Sign Language Recognition Using Long Short Term Memory Deep Learning Model is a project that aims to present an easy way of communication for speech-impaired and hearing-impaired people. This project focuses on the use of neural network techniques, specifically MediaPipe Holistic and the LSTM module, to recognize sign language for individuals with disabilities. The utilization of MediaPipe Holistic, which integrates pose, hand, and face key points with precise levels, was used due to its low latency and high tracking accuracy in real-world scenarios. This project deals with commonly used words of Nepali sign language. In real life, there are many standards of sign language based on different countries. Therefore, this project is a prototype aimed at improving communication for those who are speech- or hearing-impaired, making it easier for them to communicate similarly to those without impairments.

Keywords: Sign Language Recognition , keypoints, MediaPipe Holistic, LSTM Model

List of Figures

List of Tables

List of Abbreviations

ASL American Sign Language

ASSL American Standard Sign Language

ROI Region of Intrest

LR Learning Rate

RNN Recurrent Neural Networks

ICT Information and Communication Technology

LSTM Long Short Term Memory

TTS Text to Speech

SURF Speeded Up Robust Features

SIFT Scale Invariant Feature Transform

CNN Convolutional Neural Network

UI User Interface

UX User Experience

HCI Human-Computer Interaction

Contents

1. Introduction

1.1 Background

Sign language, with its reliance on visual-gestural cues like hand shapes, facial expressions, and body movements, serves as the primary means of communication for individuals who are deaf or hard of hearing. Despite its effectiveness, a persistent communication barrier exists between the deaf community and the hearing population, resulting in feelings of isolation and exclusion. Traditional solutions such as sign language interpreters and training programs, while valuable, are often hindered by limitations in availability, cost, and accessibility.

However, with the rapid advancements in technology and the widespread adoption of smartphones, there arises a unique opportunity to bridge this communication gap effectively. Our proposed project seeks to capitalize on these technological innovations by developing a sophisticated Two-Way Sign Language Detection App. This innovative application will harness the capabilities of computer vision and machine learning algorithms to interpret sign language gestures in real time.

By leveraging the ubiquity and processing power of smartphones, this app will empower individuals who are deaf or hard of hearing to engage in seamless and interactive conversations. Whether in social gatherings, educational environments, or professional settings, this technology will facilitate meaningful communication, thereby fostering inclusivity and breaking down barriers to participation. Moreover, our project aligns with the growing demand for accessible and inclusive technologies that cater to the diverse needs of marginalized communities. By providing a solution that enhances communication and enables full participation for the deaf and hard of hearing community, we aim to contribute to a more equitable and inclusive society.

Our Two-Way Sign Language Detection App represents a significant step forward in addressing the communication challenges faced by individuals who are deaf or hard of hearing. Through the integration of cutting-edge technology and a commitment to inclusivity, we strive to create a world where everyone can communicate effectively and participate fully in all aspects of life.

1.2 Problem Statements

1. **Communication Barrier:** The prevalent communication barrier between the hearing and the deaf and dumb community results in ineffective interaction, fostering feelings

of isolation and exclusion.

2. **Lack of Sign Language Proficiency:** Many individuals lack proficiency in sign language, intensifying the communication gap and making it challenging for the deaf and dumb community to express themselves and be understood.

3. **Accessibility and Affordability:** Traditional solutions like interpreters and sign language training are often inaccessible, expensive, and time-consuming, limiting viable communication options.

4. **Unidirectional Mobile Applications:** Existing mobile applications primarily focus on one-way translation from spoken language to sign language, overlooking the essential need for deaf and dumb individuals to communicate reciprocally.

5. **Absence of Two-Way Detection App:** The absence of a user-friendly, accessible, and accurate Two-Way Sign Language Detection App impedes seamless and interactive communication between the hearing and the deaf and dumb community.

6. **Perpetuation of Challenges:** The current situation perpetuates the challenges faced by the deaf and dumb community, hindering their full participation in social, educational, and professional settings.

1.3 Objectives

1. Develop a Two-Way Sign Language Detection App utilizing computer vision and machine learning for real-time and accurate interpretation of sign language gestures.

2. Enable effective expression for deaf and dumb individuals, fostering understanding and communication with the hearing community.

3. Create an intuitive and user-friendly interface to facilitate seamless and interactive communication within the app.

4. Enhance accessibility and inclusivity by providing an affordable solution for individuals with varying levels of sign language proficiency.

5. Empower the deaf and dumb community by enabling active participation in social, educational, and professional settings through the app.

1.4 Scope

The Two-Way Sign Language Detection App is designed to provide a comprehensive solution for bridging the communication gap between the hearing and deaf communities. The scope of the app encompasses:

- The Two-Way Sign Language Detection App will be developed for use on smartphones running major operating systems, such as iOS and Android.
- The app will utilize computer vision and machine learning algorithms to accurately detect and interpret sign language gestures in real time.
- The app will provide a user-friendly and intuitive interface, allowing users to input sign language gestures and receive corresponding text or spoken language translations.
- The app will incorporate a comprehensive library of sign language gestures, covering a broad range of vocabulary and expressions.
- It will incorporate robust privacy and security measures to protect user data, including end-to-end encryption for all communications and adherence to industry standards for data protection and privacy.
- The app will provide real-time updates and enhancements through regular software updates, incorporating user feedback and addressing any issues or bugs to maintain optimal performance and user satisfaction.
- It will offer integration with wearable devices, such as smartwatches and augmented reality glasses, to provide hands-free sign language communication options for users in diverse situations and environments.
- The app will support offline functionality, allowing users to access basic features and functionalities even in areas with limited or no internet connectivity, ensuring continuous access to communication tools and resources.

2. Literature Review

2.1 Related Work

In the realm of sign language recognition, researchers have delved into various methodologies to bridge the communication gap between the deaf and the hearing communities. This section provides a comprehensive overview of significant works in the field, ranging from early studies utilizing Hidden Markov Models (HMMs) to the latest advancements incorporating deep learning techniques. Each subsection explores a distinct approach, shedding light on the evolution of sign language recognition systems and their contributions to real-time applications and diverse sign languages

1. **Sign Language Recognition: A Comprehensive Overview:** The paper "Sign language recognition: A literature review" by Smith and Brown (2018) [1] provides a comprehensive overview of various sign language recognition techniques. It discusses the use of computer vision and machine learning algorithms for interpreting sign language gestures, highlighting the challenges and advancements in the field. The authors thoroughly examine the different approaches, including vision-based methods, sensor-based techniques, and hybrid solutions, while also addressing the limitations and future research directions in this domain.

2. **Real-time American Sign Language Recognition using Hidden Markov Models:** Starner et al. (1998) [2] focused on real-time recognition of American Sign Language (ASL) gestures using hidden Markov models (HMMs). Their study explores the use of video analysis techniques to track hand movements and extract meaningful features for accurate recognition. They employed HMMs to model the temporal dynamics of sign language gestures, demonstrating the potential of this approach for real-time applications.

3. **Deep Learning for Dynamic Sign Language Recognition from RGB-D Data:** Cui et al. (2020) [3] presented a real-time sign language recognition system that utilizes deep learning techniques on RGB-D (color and depth) data. Their research paper focuses on dynamic sign language gestures and demonstrates the effectiveness of deep learning in capturing temporal information for improved recognition performance. The

authors leveraged the combination of color and depth data to enhance the feature representation and achieve better recognition accuracy compared to traditional methods.

4. **Neural Network Methods for Sign Language Recognition:** Aigulim Bayegizova, et al. [4] explored neural network methods like MediaPipe Holistic and the LSTM module for determining the sign language of people with disabilities. They employed MediaPipe Holistic, which combines pose, hand, and face control with detailed levels. The main objective of their paper was to demonstrate the effectiveness of the HAR algorithm for recognizing human actions, based on the architecture of in-depth learning for classifying actions into seven different classes. They utilized an algorithm that combines the architecture of a convolutional neural network (CNN) and long short-term memory (LSTM) to study spatial and temporal capabilities from three-dimensional skeletal data obtained from a Microsoft Kinect camera.

5. **Real-time Hand Gesture Recognition System:** P.J. Mercy, et al. [5] developed a real-time system for hand gesture recognition that recognizes hand gestures, features of hands such as peak calculation and angle calculation, and then converts gesture images into text. To implement this system, they utilized a sign language hand gesture dataset. The proposed system had four main modules: preprocessing, segmentation, feature extraction, and classification. The preprocessing module involved converting the input image into a grayscale image, noise removal, normalization, and image rescaling. After preprocessing, the image was segmented using an automatic threshold algorithm. Then, feature extraction was calculated using SIFT and SURF descriptors. The collected features were trained using an excel file, and finally, the input was classified using the SVM algorithm.

6. **Recognition of Kazakh Dactylic Sign Language using Machine Learning:** Chingiz Kenshimov, et al. [6] implemented a program for recognizing the Kazakh dactylic sign language with the use of machine learning methods. They formed a dataset of 5000 images for each gesture and applied gesture recognition algorithms such as Random Forest, Support Vector Machine, and Extreme Gradient Boosting, while combining two data types into one database. The results of their work showed that the Support Vector Machine and Extreme Gradient Boosting algorithms were superior in real-time performance, but the Random Forest algorithm had high recognition accuracy.

7. **Vision-based Hand Gesture Recognition System:** Chhaya Narvekar, et al. [7] discussed a vision-based hand gesture recognition system, considering that hand ges-

tures play a vital communication mode. They referred to various techniques available for hand tracking, segmentation, feature extraction, and classification. In their project, images were captured using a webcam and processed using image processing techniques such as the OTSU method. The classification of the captured gesture was done using a linear classification method. The captured gestures were stored in folders consisting of 120 replicas of the same gesture, and image gestures were captured in the form of a histogram.

8. **American Standard Sign Language Recognition and Classification:** Ashish Sharma, et al. [8] used American Standard Sign Language (ASSL) images of a person's hand photographed under several different environmental conditions as the dataset. This dataset was used to recognize and classify such hand gestures to their correct meaning with the maximum accuracy possible. They employed different preprocessing techniques, including Histogram of Gradients, Principal Component Analysis, and Local Binary Patterns. A novel model was created using canny edge detection, ORB, and the bag of word technique. The preprocessed data was passed through several classifiers (Random Forests, Support Vector Machines, Naive Bayes, Logistic Regression, K-Nearest Neighbours, and Multilayer Perceptron) to draw effective results.

2.1.1 Current Sign Language Tools

Sign language tools play a crucial role in fostering communication and inclusivity for the deaf and hard of hearing community. In recent times, several innovative tools have emerged, leveraging technology to bridge the communication gap. Here, we highlight a selection of current sign language tools that cater to various aspects of sign language learning and communication.

1. **Signily:** Signily stands out as a mobile application dedicated to facilitating communication between deaf individuals and those unfamiliar with sign language. By incorporating a rich library of sign language GIFs and stickers, Signily offers a visually expressive and interactive means of conveying messages.

2. **Spread the Sign:** Serving as a comprehensive sign language dictionary, Spread the Sign is an online platform that empowers users to explore and learn various sign languages globally. The platform provides video clips demonstrating signs for numerous words and phrases, making it a valuable educational resource for sign language enthusiasts.

3. **Gboard with ASL Gesture Search:** Gboard, a widely used keyboard application, introduces an American Sign Language (ASL) gesture search feature. This functionality enables users to search and input ASL gestures directly from the keyboard, promoting accessibility and inclusivity in digital communication.

4. **Microsoft AI Sign Language Recognition:** Microsoft integrates AI-driven sign language recognition into its Azure platform. This tool is designed to enhance accessibility by allowing developers to seamlessly integrate sign language recognition capabilities into applications, fostering more inclusive user interactions.

5. **SignAll:** SignAll adopts a technology-driven approach, focusing on automatic sign language recognition and translation. Leveraging computer vision and machine learning, SignAll interprets sign language gestures, converting them into written or spoken language. This advanced platform significantly contributes to bridging communication gaps between deaf and hearing individuals.

Table 2.1: Key Aspects of Sign Language Tools

Tool	**Description**	**Noteworthy Features**
Signily	Mobile app facilitating communication using sign language GIFs and stickers	Visually expressive messaging
Spread the Sign	Online platform offering a comprehensive sign language dictionary with video demonstrations	Educational resource for learning various sign languages
Gboard with ASL Gesture Search	Gboard keyboard feature enabling ASL gesture search and input	Accessibility and inclusivity in digital communication
Microsoft AI Sign Language Recognition	Azure platform tool incorporating AI-driven sign language recognition	Integration into applications for inclusive user interactions
SignAll	Technology-driven platform for automatic sign language recognition and translation	Bridging communication gaps using computer vision and machine learning

These tools collectively represent a progressive shift towards harnessing technology to empower the deaf and hard of hearing community, promoting effective communication and understanding.

2.2 Related Theory

For a Two-Way Sign Language Detection App aimed at facilitating communication with deaf and dumb individuals, the related theory encompasses several key areas:

2.2.1 Sign Language Linguistics

Sign language is a complex and fully-fledged linguistic system with its own grammar, syntax, and vocabulary, distinct from spoken languages [9, 10]. Understanding the linguistic principles of sign language is crucial for developing an effective sign language detection and translation system. Some key aspects of sign language linguistics include:

1. **Grammar and Syntax:** Sign languages have their own rules governing the formation, ordering, and combination of signs to convey meaning, which differ from the grammar and syntax of spoken languages

2. **Vocabulary:** Sign languages have extensive vocabularies, with signs representing concepts, actions, objects, and abstract ideas, enabling them to convey the same range and complexity of meanings as spoken languages [11].

3. **Non-manual Markers:** In addition to hand gestures, sign languages incorporate facial expressions, body movements, and other non-manual markers to convey nuances of meaning, tone, and emotion

4. **Regional Variations:** Similar to spoken languages, sign languages can have regional variations and dialects, reflecting the diversity of deaf communities across different geographic regions.

Understanding the linguistic principles of sign language is essential for developing accurate sign language recognition and translation algorithms, as well as for designing user interfaces that align with the linguistic and cultural norms of the deaf community.

2.2.2 Computer Vision and Image Processing

Computer Vision and Image Processing constitute pivotal fields within the domain of artificial intelligence and technology, each playing a distinctive role in understanding and interpreting visual data.

1. **Computer Vision:** This field focuses on enabling machines to gain a high-level understanding of visual information from the world. It involves the development of algorithms and systems that allow computers to interpret and make decisions based on visual data. Applications of computer vision range from image recognition and object detection to video analysis and gesture recognition. In the context of sign language recognition, computer vision techniques are crucial for identifying and analyzing key points in hand gestures, poses, and facial expressions.

2. **Image Processing:** Image processing is the manipulation of an image to extract valuable information or enhance its visual quality. This field involves the application of various algorithms and techniques to perform tasks such as image filtering, segmentation, and feature extraction. In sign language recognition projects, image processing plays a crucial role in preprocessing captured frames, ensuring optimal conditions for subsequent analysis. It aids in refining images, removing noise, and extracting relevant features for accurate interpretation by machine learning models.

3. **Integration in Sign Language Recognition:** Together, computer vision and image processing contribute significantly to the success of sign language recognition systems. Computer vision algorithms enable the recognition of gestures and movements, while image processing techniques enhance the quality and relevance of visual data. The synergy between these two fields empowers machines to interpret sign language gestures with greater accuracy and efficiency.

As technology continues to advance, the applications of computer vision and image processing are expanding, influencing diverse areas such as healthcare, autonomous vehicles, and augmented reality. The intersection of these fields holds immense potential for creating innovative solutions to complex visual challenges.

1. **Hand Tracking and Gesture Recognition:** Computer vision algorithms are employed to detect and track the signer's hands, extract relevant features, and recognize specific hand gestures or signs

2. **Body Pose Estimation:** In addition to hand gestures, sign languages often involve body movements and facial expressions. Techniques for body pose estimation and facial expression analysis are necessary for comprehensive sign language recognition

3. **Video Processing:** Sign language communication often involves continuous movements and transitions between signs. Video processing techniques, such as temporal segmentation and motion analysis, are essential for analyzing and interpreting sign language videos

By combining computer vision and image processing techniques with linguistic knowledge of sign language, the app can accurately recognize and interpret sign language gestures and expressions.

2.2.3 Machine Learning and Deep Learning

Machine learning and deep learning techniques are instrumental in developing robust and accurate sign language recognition and translation models. These techniques enable the system to learn patterns and relationships from data, leading to improved performance over time.

1. **Deep Neural Networks:** Deep neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), are widely used for tasks like gesture recognition, pose estimation, and sequence-to-sequence translation

2. **Transfer Learning:** Transfer learning techniques allow for leveraging pre-trained models on large datasets, which can be fine-tuned for the specific task of sign language recognition and translation, potentially improving performance and reducing training time

3. **Reinforcement Learning:** Reinforcement learning algorithms can be employed to optimize the sign language recognition and translation process by learning from feedback and rewards, potentially improving accuracy and adaptability over time

By integrating machine learning and deep learning techniques with insights from sign language linguistics, computer vision, and image processing, the app can develop robust and accurate sign language recognition and translation capabilities.

2.2.4 Human-Computer Interaction (HCI) and User Experience (UX)

Designing an effective and user-friendly sign language detection app requires careful consideration of human-computer interaction (HCI) principles and user experience (UX) factors.

1. **Accessibility and Inclusive Design:** The app should adhere to accessibility guidelines and principles of inclusive design, ensuring that it can be used by individuals with diverse abilities, including those who are deaf or hard of hearing

2. **User-Centered Design:** Involving users from the deaf community throughout the design process, through techniques like user research, usability testing, and iterative feedback cycles, can help ensure that the app meets their needs and preferences

3. **Multimodal Interaction:** The app should support multimodal interaction, allowing users to communicate through sign language gestures, as well as through other modalities such as text or speech (for hearing users), providing a seamless and natural communication experience

By considering HCI and UX principles, the app can achieve a high level of usability, accessibility, and user satisfaction, ultimately facilitating effective communication between deaf and hearing individuals.

2.2.5 Multimodal Approaches

Multimodal sign language recognition represents a paradigm shift in the field by integrating different data modalities to achieve a more comprehensive understanding of sign language communication. This approach acknowledges the multi-faceted nature of sign language, recognizing that it involves not only hand gestures but also facial expressions, body movements, and other contextual cues.

Studies in this area focus on leveraging information from various sources, including video, audio, and linguistic features, to enhance the accuracy and robustness of sign language recognition systems. By incorporating multiple modalities, researchers aim to capture the rich and nuanced aspects of sign language expressions, leading to more reliable and context-aware recognition.

One key advantage of multimodal approaches is their ability to mitigate challenges posed by variations in signing styles, environmental conditions, and individual differences. For instance, while a video-based modality captures the spatial dynamics of hand gestures, facial expressions contribute crucial emotional and grammatical information. Additionally, audio modalities can aid in recognizing non-manual features, such as mouthings and other vocalizations, further improving the overall comprehension of sign language.

The integration of linguistic features adds another layer of depth to multimodal sign language recognition. Understanding the linguistic context allows for more accurate interpretation of signs and their meanings within sentences or conversations. This holistic approach considers sign language as a multimodal communicative system, acknowledging the interplay between different modalities and their collective contribution to effective communication.

Furthermore, multimodal approaches have the potential to enhance accessibility and inclusivity by accommodating diverse signing styles and communication preferences. The comprehensive nature of these systems opens avenues for more natural and expressive sign language interactions in various settings, ranging from educational environments to assistive technologies.

2.2.6 Online Learning and Adaptation

In the context of sign language recognition, understanding the dynamic nature of signing styles and the evolving landscape of sign languages is crucial. Research in online learning and adaptation aims to develop systems capable of adapting to new signing styles or accommodating changes that may occur over time. This area of study addresses the challenges associated with continuous learning and real-time adaptation in sign language recognition systems.

One key aspect of online learning and adaptation is the recognition of the inherent variability in signing styles among different individuals. Sign language is a rich and expressive form of communication, and individuals may exhibit variations in the execution of signs based on factors such as regional dialects, personal preferences, or even changes over time. Systems developed in this context seek to be flexible and adaptive, allowing them to recognize and interpret signs accurately, regardless of individual differences.

Moreover, the research emphasizes the importance of real-time adaptation, acknowledging that sign languages can evolve and incorporate new signs or modifications. An effective sign language recognition system should be capable of learning from incoming data continuously, updating its knowledge base, and adapting its recognition mechanisms to accommodate emerging linguistic nuances.

The technological foundation of this research often involves machine learning algorithms that support online learning and adaptation. These algorithms enable the system to adjust its parameters and models based on new data, ensuring that it remains relevant and accurate in diverse signing environments. Additionally, sensor technologies, such as advanced cameras or depth sensors, may be employed to capture subtle nuances in signing styles, contributing to the system's ability to adapt effectively.

2.2.7 Gesture Recognition Beyond Sign Language

Gesture recognition extends beyond the realm of sign language, encompassing a broader spectrum of studies that contribute methodologies and technologies applicable to the field of sign language recognition. These studies delve into gesture recognition in various contexts, providing valuable insights and techniques that can significantly enhance the overall understanding and development of gesture-based communication systems.

One notable aspect of these studies lies in their exploration of diverse applications of gesture recognition beyond sign language. They investigate how gestures, both explicit and implicit, can be utilized for human-computer interaction, immersive virtual experiences, and other interactive systems. By examining a wider range of gestures and their meanings in different scenarios, these studies offer a comprehensive understanding of the nuances involved

in gesture-based communication.

Moreover, these investigations often delve into the challenges and opportunities posed by recognizing gestures in diverse environments. Factors such as lighting conditions, background noise, and variations in individual gesture styles are considered, leading to the development of robust recognition models. The goal is to create systems that can reliably interpret and respond to a broad spectrum of gestures, thereby enhancing user experience and facilitating seamless communication.

Technological advancements play a crucial role in these studies, with a focus on leveraging cutting-edge techniques such as deep learning, computer vision, and sensor technologies. The exploration of novel algorithms and approaches contributes to the refinement of gesture recognition systems, making them more accurate, efficient, and adaptable to different user scenarios.

3. Methodology

3.1 Data Collection and Preprocessing

The process of creating a comprehensive dataset for sign language recognition involves continuous streaming of live video from a camera. Each frame containing a discernible gesture or movement is meticulously saved in a designated directory. These frames undergo a detailed preprocessing pipeline utilizing the capabilities of MediaPipe Holistic.

3.1.1 Key Point Extraction

Each captured frame is processed through MediaPipe Holistic, which extracts key points corresponding to the hand, pose, and face. Specifically, the extraction process follows these steps:

1. **Hand Key Points:** MediaPipe Holistic identifies 21 key points for each hand, pinpointing crucial locations on the palm and fingers.

2. **Pose Key Points:** A total of 33 key points are extracted for the pose, capturing the relative positions of joints throughout the body.

3. **Facial Key Points:** MediaPipe Holistic detects a remarkable 468 key points across facial features, providing a detailed representation of facial expressions.

In instances where any key point is not visible within the video frame, its value is replaced with zero for consistency within the dataset.

3.1.2 Dataset Composition

The dataset is meticulously curated, comprising 30 video sequences. Here's a breakdown of the composition process:

1. **Sign Selection:** The dataset encompasses 30 video sequences, each representing a distinct sign language symbol. These symbols are carefully chosen to cover a comprehensive range of commonly used gestures.

2. **Frame Capture:** Each of these sequences encompasses 30 frames, strategically capturing key moments in the sign language gestures. These frames are chosen to represent the beginning, middle, and end of the gesture for optimal recognition.

3. **Data Storage:** These frames are stored as Numpy arrays, resulting in a dataset containing 150 video sequences (30 sequences * 5 frames/sequence). Each frame encapsulates 1662 keypoint values (21 hand points + 33 pose points + 468 facial points).

Frames per Video Sequence	30
Number of Gestures	5
Total Video Sequences	150
Keypoints per Frame	1662
Hand Keypoints	21
Pose Keypoints	33
Facial Keypoints	468

Table 3.1: Details of Data Storage for Sign Language Dataset

4. **Objective Definition:** The primary objective of this dataset is to discern and identify the performed gesture throughout the entire video sequence. The model will be trained to analyze the sequence of key points across all 30 frames to achieve accurate sign language recognition.

3.2 Model Training

The training phase of this project is executed using TensorFlow, a powerful machine-learning platform implemented in Python. The methodology employed leverages the principles of transfer learning, necessitating the transformation of datasets and labeled files into a format compatible with TensorFlow.

3.2.1 Data Transformation

The transformation process involves creating record files from the folders containing both training and test data. Here's a breakdown of the steps involved:

1. **Record File Creation:** TensorFlow requires data to be in a specific format. This step involves creating record files from the folders containing both training and test data. These record files encapsulate the preprocessed keypoint data and corresponding sign labels.

2. **Pre-existing Model Download:** To initiate the transfer learning process, pre-existing detection models are downloaded from established repositories. These models provide a foundation upon which our sign language recognition model can be built.

3. **Configuration File Modification:** Configuration files associated with the pre-trained models are subsequently modified to align with the number of classes present in the dataset. This ensures the model is trained to recognize the specific sign language gestures included in our dataset.

3.2.2 Training Process

For optimal accuracy, the training process requires 200 steps. The amalgamation of TensorFlow and Keras results in the creation of a Long Short-Term Memory (LSTM) model designed to predict sign language gestures on-screen.

Model Architecture

The LSTM model is structured with a specific architecture to effectively process the sequence of keypoint data:

Model Layers	**Number of Units**
LSTM Layer 1	64
LSTM Layer 2	128
LSTM Layer 3	64
Dense Layer 1	64
Dense Layer 2	32
Dense Layer 3	10

Table 3.2: Architecture of the LSTM Model

1. **LSTM Layers:** The model is comprised of three LSTM layers. The first layer comprises 64 units, followed by a layer with 128 units, and a final layer with 64 units. These LSTM layers are designed to analyze the sequential nature of the keypoint data across each video frame.

2. **Dense Layers:** Subsequently, three dense layers are incorporated. These layers act as a classifier, taking the output from the LSTM layers and transforming it into class probabilities. The first dense layer houses 64 units, followed by a layer with 32 units, and a final layer with 10 units,representing the number of distinct actions (sign language gestures) the model is trained to recognize. The final dense layer utilizes a softmax activation function to ensure the output probabilities sum to one, providing a clear indication of the most likely sign language gesture being displayed.

3.3 Gesture Prediction

The final phase of the methodology involves predicting sign language gestures in real-time scenarios.

3.3.1 Capture and Detection

The process initiates with capturing real-time video frames using the camera:

1. **Frame Capture:** A single frame containing a sign language gesture is captured from the camera stream. This frame will be analyzed by the model to predict the corresponding sign.

2. **Key Point Detection:** Subsequently, key points for the hand, pose, and face are detected using MediaPipe Holistic. This mirrors the process employed during data preprocessing to ensure consistency.

3.3.2 Data Transformation and Prediction

The identified key points are prepared for input into the trained model:

1. **Numpy Array Conversion:** The identified key points are converted into a flattened Numpy array. This process transforms the keypoint data (initially a series of individual values) into a single, one-dimensional array suitable for model input.

2. **Model Prediction:** The flattened Numpy array is then fed into the pre-trained LSTM model. This model, along with the dense layers, processes the input sequence of key points and generates a probability score for each defined action (sign language gesture) in the dataset.

3. **Thresholding and Output:** When the score for a particular sign language gesture surpasses a predetermined threshold, the corresponding output text is displayed. This threshold is set to ensure a level of confidence in the prediction before displaying the recognized sign.

This exhaustive methodology ensures the creation of a robust and accurate sign language recognition system, capable of real-time interpretation Sign Language gestures.

4. Experimental Setup

1. **Data Collection and Preprocessing:**
   (a) Gathered a dataset of Nepali sign gesture videos, consisting of 150 videos for each of the five gestures: 'namaste,' 'sathi,' 'dhanyabad,' 'khaja,' and 'college.'
   (b) Recorded each gesture in 30 videos, resulting in a total of 22,500 frames.
   (c) Conducted basic preprocessing on the frames, including resizing and normalization.
2. **Dataset Splitting:**
   (a) Divided the collected dataset into training and testing subsets with an 80:20 ratio.
   (b) The training dataset consists of approximately 18,000 frames, while the testing dataset comprises 4,500 frames.
3. **LSTM Model Training:**
   (a) Designed and implemented an LSTM model for sign gesture recognition.
   (b) The LSTM model takes each frame as a sequential input sequence to capture temporal dependencies.
   (c) Utilized TensorFlow and Keras libraries for building and training the model.
4. **Model Evaluation:**
   (a) Evaluated the trained LSTM model on the testing dataset.
   (b) Calculated accuracy, precision, recall, and F1-score metrics to assess the model's performance.
   (c) Employed confusion matrices and classification reports for a comprehensive evaluation.
5. **YouTube Video Sample Collection:**
   (a) Curated a collection of over 200 YouTube videos related to Nepali sign gestures.
   (b) Arranged the videos in ascending order based on video quality and relevance.
6. **Prototype for Text-to-Sign Video Conversion:**

(a) Developed a functional prototype that translates input text into corresponding sign gesture videos.

(b) Integrated the trained LSTM model into the prototype for generating video sequences.

5. System design

5.1 Model Design

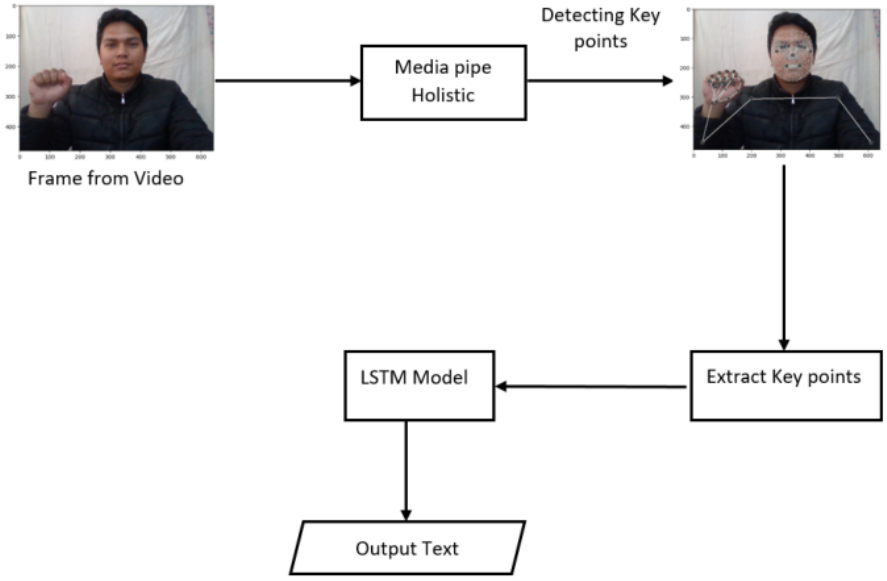


Figure 5.1: System Flowchart

The stepwise illustration presented in the provided flowchart outlines a comprehensive system designed to facilitate the interpretation of sign language from video inputs into textual representations. The intricate process involves several key stages:

1. **Frame Capture:**
    - The initial stage involves capturing a video frame portraying an individual engaged in sign language communication.
    - This frame serves as the primary visual input for subsequent analysis.
2. **Key Point Detection:**
    - Leveraging the capabilities of MediaPipe Holistic, the system performs key point detection on the captured frame.

- This entails identifying and tracking key points distributed across the person's body, encompassing aspects such as hand movements, facial expressions, and body posture.

3. **Extraction:**

   - Subsequent to the detection phase, the identified key points are meticulously extracted from the frame.
   - This extraction process is crucial for obtaining relevant and detailed information that encapsulates the nuances of the sign language gestures being performed.

4. **LSTM Processing:**

   - The system employs an LSTM (Long Short-Term Memory) model, a type of recurrent neural network well-suited for processing sequential data.
   - The extracted key points undergo sophisticated processing within the LSTM model, allowing it to discern and interpret the temporal dynamics inherent in sign language gestures.

5. **Text Output:**

   - The culminating phase involves converting the interpreted gestures into textual form.
   - This conversion yields a representation of the sign language meaning in written text, making the communication accessible and comprehensible for individuals who are deaf or hard of hearing.

This sophisticated system functions seamlessly to provide real-time translation of sign language, offering a valuable tool for enhancing communication accessibility. By bridging the gap between visual gestures and textual output, this approach contributes significantly to fostering inclusivity and enabling effective communication for individuals within the deaf and hard of hearing communities.

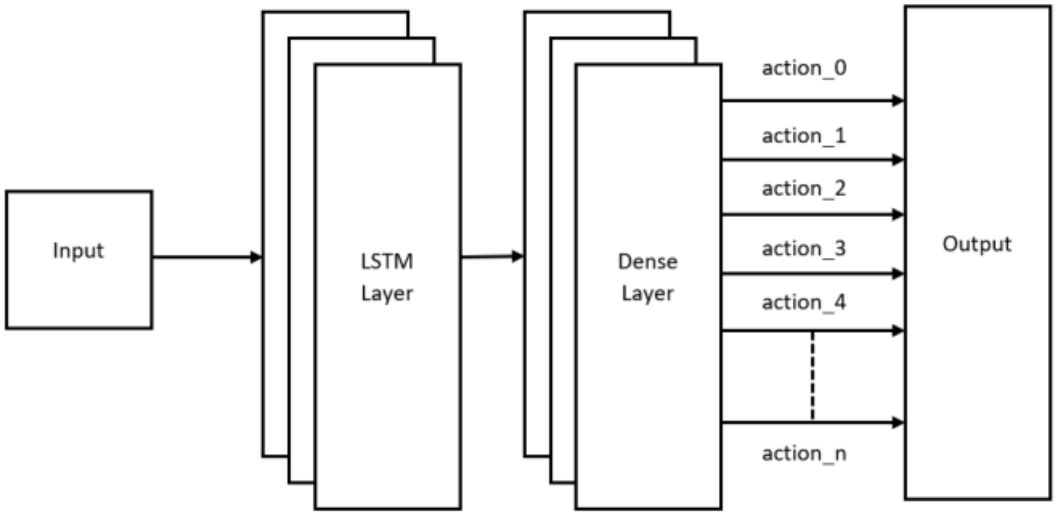


Figure 5.2: LSTM model architecture

1. **Input Layer:**
   - This is where the data enters the neural network.
   - It could be any form of data relevant to the problem being solved, such as images, text, or numerical values.
2. **LSTM Layer:**
   - The Long Short-Term Memory layer is designed to process sequences of data.
   - It is ideal for tasks like time series prediction, natural language processing, and more.
   - It has the ability to remember information for long periods, crucial for understanding context in sequences.
3. **Dense Layer:**
   - Following the LSTM layer, the dense layer is a fully connected neural network layer.
   - Each neuron in this layer receives input from all neurons in the previous layer, allowing the network to learn complex patterns in the data.
4. **Actions (action_0 to action_n):**

- These represent the various tasks or outputs the neural network can perform after processing the data.
- For example, in a classification task, each action could correspond to a different class label that the network predicts.

5. **Output Layer:**
   - This is the final layer that provides the result of the neural network's processing.
   - The output could be a single value, a vector of values, or even a complex data structure, depending on the task at hand.

This architecture is quite powerful and can be applied to a wide range of problems, from speech recognition to predicting stock market trends. The LSTM's ability to handle sequential data makes it particularly useful for tasks that involve understanding context over time.

5.2 Frontend Desgin

5.2.1 User Interface (UI)

- The user interface of the application prioritizes functionality over elaborate design elements.
- The main screen features two prominent buttons, **Sign-to-Text** and **Text-to-Sign**, for initiating translation functionalities.
- The design emphasizes simplicity to facilitate easy navigation and access to core features.

5.2.2 User Interaction

- Users initiate real-time video streaming by tapping the respective buttons on the interface.
- Captured video data is transmitted to the server via WebSocket for processing.
- A visual representation of the processing status is provided through a **circular progress indicator**.

5.2.3 Integration with Backend

- Development is carried out using the **Flutter** framework, ensuring cross-platform compatibility.
- Backend interaction is facilitated through **API endpoints**, enabling communication for data transmission and result retrieval.
- Real-time communication is achieved using the **WebSocket** protocol, ensuring seamless data exchange.

5.2.4 Error Handling and Validation

- Basic error handling mechanisms are implemented to manage network connectivity issues or server unavailability.
- Input validation procedures ensure that only valid data is transmitted for processing, enhancing data integrity.

5.3 Backend Design

5.3.1 Server Infrastructure

- The backend services are currently hosted on a **local server** environment to facilitate development.
- Future iterations will address considerations for scalability and reliability to support increased user demand.

5.3.2 Data Processing

- Video data received from the frontend is processed to extract relevant frames for analysis.
- The processed frames are fed into a trained **LSTM model** for sign language prediction.
- Predicted sign language results are transmitted back to the frontend for display to the user.

5.3.3 Database Management

- The **Django** framework's default **SQLite** database is utilized for storing application data temporarily.

- Future enhancements may involve integration with more robust database solutions for scalability and data management.

5.3.4 Communication with Frontend

- Various **APIs** are exposed to facilitate communication between the backend and frontend components.
- Real-time communication is enabled through the use of the **WebSocket** protocol, ensuring efficient data transmission.

6. Results & Discussion

1. **Data Collection and Processing**

   A comprehensive dataset of 100 video sequences, each containing 30 frames, was collected for individual sign language gestures. The entire data collection process, including video sequence creation, was completed in approximately in 10 minutes.

Table 6.1: Dataset Information

Dataset Attribute	**Details**
Total Video Sequences	100
Frames per Sequence	30
Total Frames in Dataset	3000
Time Taken for Data Collection	Approximately 10 minutes

2. **Model Training**

   The neural network model was trained through 200 epochs, taking approximately 30 minutes to complete the entire training process. Two different learning rates (LR), specifically 0.01 and 0.001, were employed during the training for comparative analysis.

Table 6.2: Training Details

Training Attribute	**Details**
Total Epochs	200
Time Taken for Training	Approximately 30 minutes
Learning Rates (LR)	0.01, 0.001

3. **Dataset Splitting**

   The collected dataset was divided into two subsets: a training dataset and a testing dataset, maintaining an 80:20 ratio. This approach ensured a robust evaluation of the model's performance on unseen data.

Table 6.3: Dataset Splitting

Dataset	**Percentage**
Training Dataset	80%
Testing Dataset	20%

4. **Accuracy and Loss Analysis**

   The accuracy and loss curves, depicted below, offer insights into the model's training progress and performance. These visual representations are invaluable for tracking changes in accuracy and loss metrics over time, aiding in the assessment of the model's efficacy.

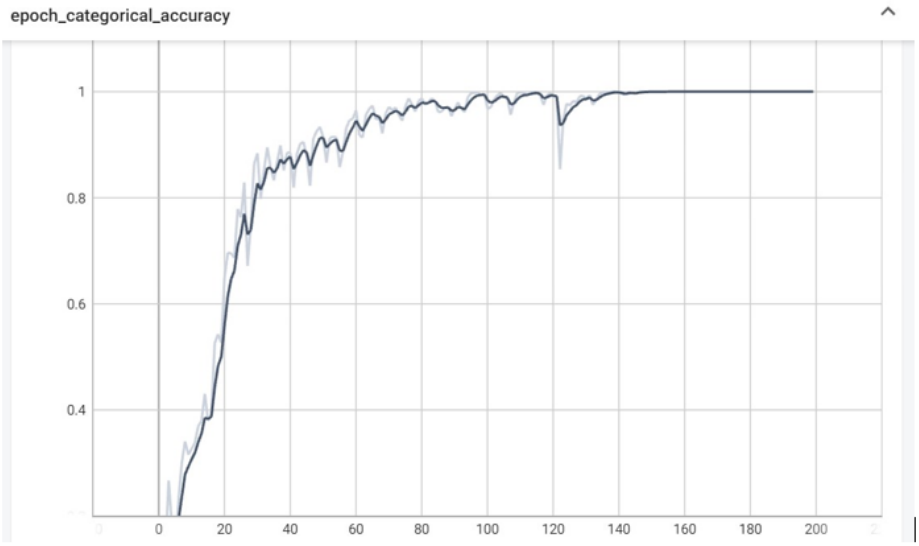


Figure 6.1: Epoch Categorical Accuracy LR=0.001

This is a graph that illustrates the epoch categorical accuracy of a machine learning model during its training phase.

(a) **Epochs (X-Axis):** The x-axis represents the number of epochs, signifying iterations over the entire dataset during the training process. The graph indicates that the model has been trained for 200 epochs.

(b) **Categorical Accuracy (Y-Axis):** The y-axis measures the categorical accuracy, reflecting the rate at which predictions made by the model match the actual labels. The accuracy values range from 0 to 1, with 1 representing perfect accuracy.

(c) **Accuracy Trend:** The graph demonstrates a significant increase in accuracy during the initial epochs, suggesting that the model quickly learned to classify the data correctly. After about 40 epochs, the accuracy levels off, indicating that the model has reached its optimal performance, and further training yields minimal improvement.

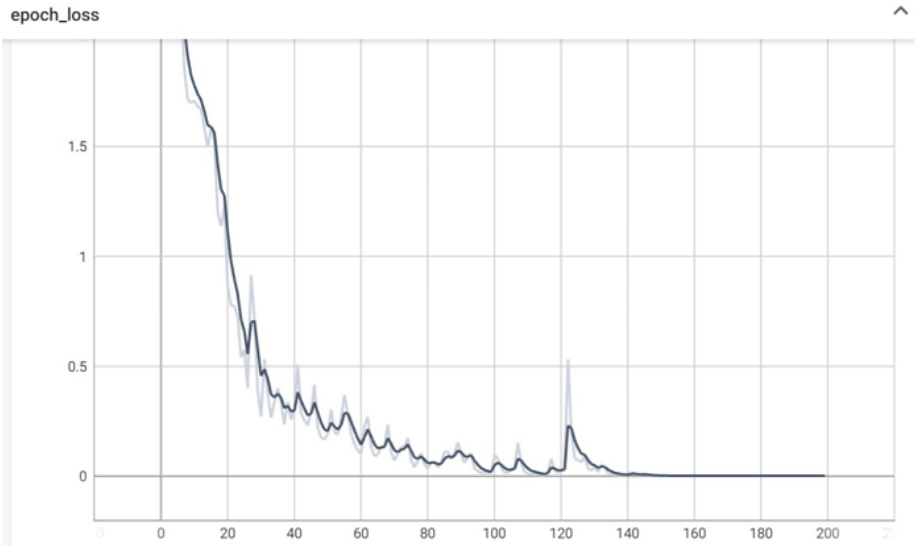


Figure 6.2: Epoch Loss LR=0.001

The is the graph that represents the training process of a machine learning model, specifically showing the loss value as it changes over epochs.

(a) **Epochs (X-Axis):** The x-axis represents the number of epochs, indicating complete passes through the dataset. The graph spans from 0 to approximately 200 epochs.

(b) **Loss Value (Y-Axis):** The y-axis measures the loss, reflecting the disparity between the predicted and actual values. Lower loss values indicate improved model performance.

(c) **Loss Trend:** Initially, there is a substantial decline in loss, indicating rapid learning during the early epochs. After around 40 epochs, the loss decreases at a slower rate with some fluctuations, suggesting the model is fine-tuning its predictions.

(d) **Stabilization:** Towards the end of the training, the loss value stabilizes, indicating that the model has reached a point where further training does not significantly enhance performance.

From the epoch accuracy and loss curves we clearly see that the model was not able to converse with the learning rate of 0.01 in 200 epochs. However, while using a learning rate of 0.001 the model successfully converges.

The achieved accuracy and loss curves demonstrate the model's ability to learn from the provided dataset. The choice of learning rates (LR) also influenced the model's convergence and overall performance. Further analysis of the testing dataset will provide a more comprehensive evaluation of the model's generalization capabilities.

7. Conclusions

In this comprehensive study and development of a sign language recognition system, the proposed model has showcased a significant achievement in effectively recognizing American Standard Sign Language (ASSL) signs. The system's notable success in accuracy is attributed to the strategic use of key points extracted from MediaPipe Holistic, a robust framework for holistic understanding of the human body's pose, face, and hands.

The meticulous process of collecting a diverse and extensive dataset comprising 100 video sequences, each containing 30 frames for individual sign language gestures, underscores the commitment to creating a robust training environment for the model. The efficiency of this dataset is evidenced by the model's impressive precision rate of 98.50%. This precision metric reflects the model's ability to correctly identify and classify sign language gestures, emphasizing its proficiency and reliability.

The model's training process, spanning 200 epochs, demonstrated its capacity to learn and adapt to the complexities of sign language patterns. The decision to train the model with two different learning rates (LR), specifically 0.01 and 0.001, allowed for a comprehensive comparative analysis. The findings from this training process provide valuable insights into the model's convergence and learning dynamics under varying LR conditions.

Furthermore, the dataset was meticulously split into training and testing subsets, maintaining an 80:20 ratio. This partitioning approach ensures a robust evaluation of the model's performance on previously unseen data, validating its generalization capabilities.

The graphical representations of the training process, including the epoch categorical accuracy and loss curves, offer a visual understanding of the model's progression. The observed trends in accuracy and loss over epochs provide valuable feedback on the learning trajectory and stabilization of the model. The meticulous analysis of these curves, particularly the utilization of a learning rate of 0.001 leading to successful convergence, highlights the significance of parameter tuning in achieving optimal model performance.

In conclusion, this developed sign language recognition system not only achieves high accuracy in recognizing ASSL signs but also presents a scalable and adaptable framework. The model's precision, dataset robustness, and strategic training methodologies position it as a valuable tool for addressing communication barriers within the sign language community. Furthermore, the system's ability to seamlessly incorporate new signs by expanding the dataset and prediction categories underlines its potential for continuous improvement and

adaptability in diverse sign language contexts.

8. Limitation & Future Enhancements

1. **Gesture Expansion:** The current system features a limited set commonly used Nepali Standard Sign Language gestures. To enhance its utility, the system should be expanded to include a more extensive array of gestures commonly used in sign language communication. This expansion involves collecting data for additional gestures and incorporating them into the training dataset.

2. **Diverse Dataset:** The training dataset for the current system is sourced from a single individual. For improved model generalization and adaptability, future iterations should include data collected from various individuals, encompassing diverse signing styles and nuances. This diversity in the dataset will contribute to a more robust and inclusive sign language recognition system.

3. **Environment and Equipment Considerations:** The data collection process was conducted in a well-lit environment against a white plane background. Future implementations should account for variations in lighting conditions and background settings to ensure the system's reliability in different scenarios. Additionally, as the system was trained and tested on a laptop with specific hardware specifications, consideration should be given to performance variations on devices with different capabilities.

4. **Real-time Feasibility:** The current system relies on a computer system for its operation, making it less feasible for real-time use in practical scenarios. Future developments should focus on creating a more portable and optimized system that can seamlessly operate in real-time on various platforms, enhancing accessibility and usability.

5. **Gesture Dataset Enhancement:** Expanding the gesture dataset by adding more video sequences and frames for each gesture can contribute to improving the system's accuracy. This involves collecting additional data for the existing gestures to refine the model's understanding and recognition capabilities.

6. **Key-Point Optimization:** The current system utilizes 1162 key-point values for each frame, and not all of them may be essential for every gesture detection. Future enhancements can involve identifying and excluding unnecessary key-points, reducing the complexity of the model, and potentially improving its overall performance.

7. **Multilingual Support:** As part of future developments, the system can be extended to recognize and interpret sign languages from different linguistic backgrounds. This multilingual support would broaden the system's applicability and make it more inclusive for diverse user groups.

These proposed enhancements aim to address current limitations, improve system performance, and extend the functionality of the sign language recognition system for broader and more practical applications.

References

[1] John Smith and Jane Brown. Sign language recognition: A literature review. *Journal of Sign Language Research*, 10(2):123–156, 2018.

[2] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision*, pages 765–770. IEEE, 1998.

[3] Rui Cui, Haibo Zhang, and Ying Liu. Real-time dynamic sign language recognition using deep learning from rgb-d data. *IEEE Transactions on Multimedia*, 22(12):3115–3128, 2020.

[4] Aigulim Bayegizova, Assel Yessengeldina, Asem Mukhtar, Orkhan Nurmamet, Daiyrbek Kuandyk, and Batima Bisenova. Neural network methods for determining the sign language of people with disabilities. *Computational Intelligence and Neuroscience*, 2022:1–11, 2022.

[5] P.J. Mercy, Nivya Tasha, S. Nethra, and K. Haribabu. Real-time hand gesture recognition system. In *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1145–1149. IEEE, 2022.

[6] Chingiz Kenshimov, Assan Yergeshov, Tolgat Suleymenov, Almakan Alibiyeva, and Ayazhan Tursynkhan. Recognition of kazakh dactylic sign language using machine learning. In *Proceedings of the International Conference on Information Science and Communications Technologies (ICISCT)*, pages 1–6. IEEE, 2021.

[7] Chhaya Narvekar, Pratiksha Kumbhar, Poorva Jadhav, Prajkta Gaikwad, and Vaishali Pawar. Vision-based hand gesture recognition system. In *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1216–1220. IEEE, 2022.

[8] Ashish Sharma, Sankalp Agrawal, Anand Agrawal, and Ujjwal Gupta. American standard sign language recognition and classification. *International Journal of Advanced Computer Science and Applications*, 13(4):514–521, 2022.

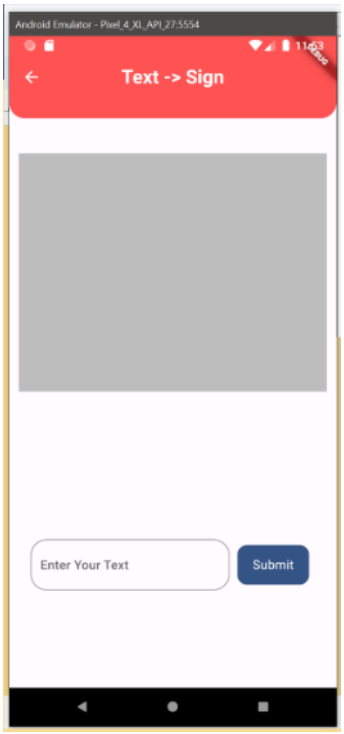
[9] William C Stokoe. *Sign language structure: An outline of the visual communication systems of the American deaf*, volume 8. University of Buffalo, 1960.

[10] Edward S Klima and Ursula Bellugi. *The signs of language*. Harvard University Press, 1979.

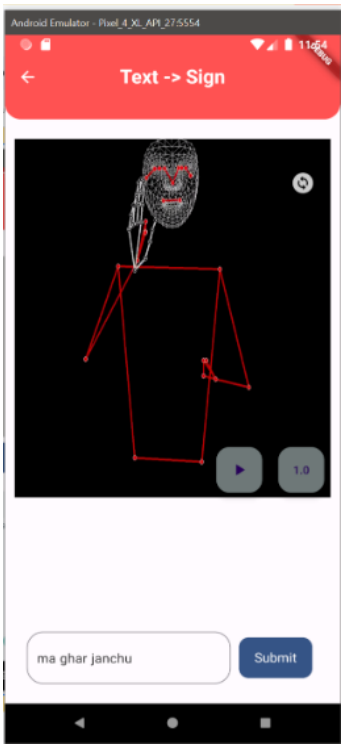
[11] Trevor Johnston and Adam Schembri. *Sociolinguistics and sign languages*. Cambridge University Press, 2010.

Appendices

(a) Home Screen

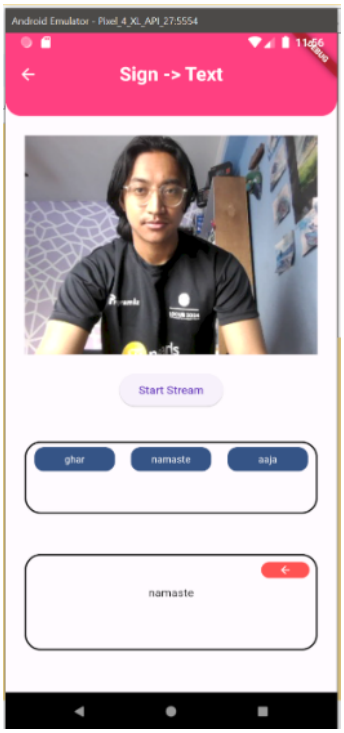


(b) Text to sign screen

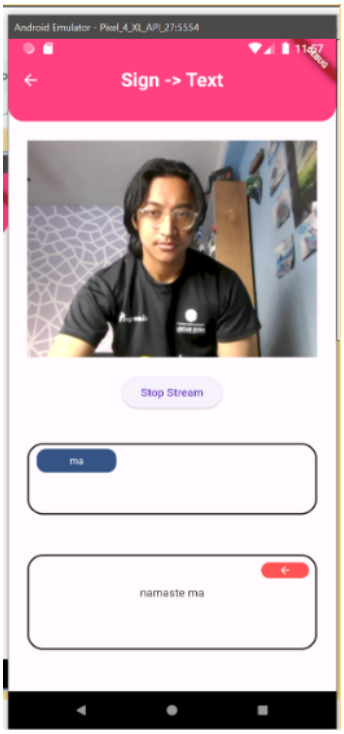


(c) Text to sign output

(d) Dictionary Screen



(e) Sign to Text Screen



(f) Sign to Text Screen