Real-Time Action Sign Language Interpretation using Deep Learning and MediaPipe

A PROJECT REPORT

Submitted by

SELVALINGESHWARAN R [211420243049]

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARITFICIAL INTELLIGENCE AND

DATA SCIENCE

PANIMALAR ENGINEERIG COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERIG COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this mini project report **"Real-Time Action Sign Language Interpretation using Deep Learning and MediaPipe"** is the bonafide work of "Selvalingeshwaraan R [211420234049]" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
DR. S. MALATHI	**Dr. B. CHITRA**
HEAD OF THE DEPARTMENT,	**ASSOCIATE PROFESSOR,**
DEPARTMENT OF AI&DS,	DEPARTMENT OF AI&DS,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,	NAZARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above-mentioned students were examined in End Semester Project Report (AD8811) held on __________.

INTERNAL EXMAINER **EXTERNAL EXAMINER**

DECLARAION BY THE STUDENT

I’m Selvalingeshwaran [211420243049], hereby declare that this project report titled **"Real-Time Action Sign Language Interpretation using Deep Learning and MediaPipe"**, under the guidance of **Dr. B. Chitra** is the original work done by me and we have not plagiarized or submitted to any other degree in any university.

ACKNOWLEDGEMENT

I would like to express our deep gratitude to our respected Secretary and Correspondent **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

I express our sincere thanks to our directors **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities to undertake this project.

I also express our gratitude to our Principal **Dr. K. MANI, M.E., Ph.D.** who facilitated us in completing the project.

I thank the Head of the AI&DS Department, **Dr. S. MALATHI, M.E., Ph.D.,** for the support extended throughout the project.

I would like to thank our supervisor **Dr. B. Chitra**, coordinator **Dr. K. JAYASHREE & Dr. P. KAVITHA** and all the faculty members of the Department of AI&DS for their advice and encouragement for the successful completion of the project.

SELVALINGESHWARAN R

ABSTRACT

Sign language serves as a vital means of communication for individuals with hearing or speech impairments, fostering inclusivity and bridging the gap with the broader community. However, interpreting sign language actions in real-time remains a challenging endeavor, often hindering seamless communication and creating barriers for those who rely on this mode of expression. This project proposes an innovative approach that integrates deep learning techniques with the powerful capabilities of the MediaPipe library to address the challenge of real-time action sign language interpretation. At the core of the proposed solution lies a Long Short-Term Memory (LSTM) neural network model, which is trained on a comprehensive dataset of sign language actions. The MediaPipe Holistic model plays a crucial role in efficient landmark detection and tracking from video sequences, enabling the extraction of keypoints and bounding box coordinates as input features for the LSTM model. This synergistic combination of deep learning and MediaPipe's advanced capabilities forms the foundation of the proposed approach. The developed system aims to accurately recognize and classify a diverse set of sign language actions, thereby enhancing communication accessibility and fostering inclusivity for the hearing-impaired community. The potential impact of this project extends beyond mere technological advancement, holding the promise of breaking down communication barriers, empowering individuals to actively participate across various domains, and ultimately contributing to a more inclusive society. The trained LSTM model is seamlessly integrated into a user-friendly Flask web application, facilitating real-time inference and interpretation of sign language actions captured through a webcam feed. A sliding window approach ensures smooth and continuous action recognition, providing an intuitive and natural experience for users interacting with the system. This report delves into the intricate details of the project, starting with a comprehensive literature review that establishes the context and highlights the state-of-the-art in sign language recognition. Subsequently, a detailed explanation of the methodology is provided, shedding light on the data collection and preprocessing techniques, model architecture, feature extraction process, and training procedures. The implementation section elaborates on the development environment, tools, code structure, and the integration of the trained model with the MediaPipe library. Furthermore, the report presents a thorough evaluation of the system's performance, including quantitative results such as accuracy, precision, and recall, as well as qualitative results through visualizations and examples. A discussion and analysis section explores the strengths and limitations of the proposed approach, drawing comparisons with existing techniques and suggesting potential improvements. Finally, the report concludes by summarizing the project's key findings, contributions, and significance, while also exploring potential avenues for future research and development in the realm of sign language interpretation and assistive technologies. Through this groundbreaking project, we aim to contribute to the ongoing efforts towards bridging the communication gap and fostering a more inclusive society, where individuals with hearing or speech impairments can fully express themselves and engage with the world around them without barriers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

INTRODUCTION

CHAPTER - 1

INTRODUCTION

Communication is a fundamental human need, enabling individuals to express themselves, share thoughts and emotions, and engage with the world around them. For those with hearing or speech impairments, sign language serves as a vital mode of communication, fostering inclusivity and bridging the gap with the broader community. However, interpreting sign language actions in real-time has long been a challenging endeavour, hindering seamless communication and creating barriers for those who rely on this mode of expression.

The ability to accurately recognize and interpret sign language actions in real-time holds immense potential for enhancing communication accessibility and promoting inclusivity for individuals with hearing or speech impairments. By breaking down these communication barriers, such a system could empower these individuals to actively participate across various domains, including education, employment, and social interactions, ultimately contributing to a more inclusive society.

Driven by this motivation, the primary objective of this project is to develop an innovative solution that seamlessly integrates deep learning techniques with the advanced capabilities of the MediaPipe library for real-time action sign language interpretation. The proposed approach harnesses the power of deep learning through a Long Short-Term Memory (LSTM) neural network architecture, which is trained on a comprehensive dataset of sign language actions. Complementing this deep learning approach is the MediaPipe Holistic model, a powerful tool that facilitates efficient landmark detection and tracking from video sequences. This synergistic combination enables the extraction of keypoints and bounding box coordinates as input features for the LSTM model, forming the core of the proposed solution.

The potential impact of this project extends beyond technological advancements, as it holds the promise of revolutionizing the way individuals with hearing or speech impairments communicate and interact with the world around them. By enabling real-time interpretation of sign language actions, the developed system could serve as a catalyst for breaking down communication barriers, fostering inclusivity, and empowering individuals to actively engage in various domains, such as human-computer interaction, assistive technologies, educational tools, and sign language learning platforms.

In the following sections, this report delves into the intricate details of the project, starting with a comprehensive literature review that establishes the context and highlights the state-of-the-art in sign language recognition. Subsequently, a detailed explanation of the methodology is provided, shedding light on the data collection and preprocessing techniques, model architecture, feature extraction process, and training procedures. The implementation section elaborates on the development environment, tools, code structure, and the integration of the trained model with the MediaPipe library for real-time inference and interpretation.

Furthermore, the report presents a thorough evaluation of the system's performance, including quantitative results such as accuracy, precision, and recall, as well as qualitative results through visualizations and examples. A discussion and analysis section explores the strengths and limitations of the proposed approach, drawing comparisons with existing techniques and suggesting potential improvements. Finally, the report concludes by summarizing the project's key findings, contributions, and significance, while also exploring potential avenues for future research and development in the realm of sign language interpretation and assistive technologies.

Through this groundbreaking project, we aim to contribute to the ongoing efforts towards bridging the communication gap and fostering a more inclusive society, where individuals with hearing or speech impairments can fully express themselves and engage with the world around them without barriers.

CHAPTER - 2

LITERATURE SURVEY

CHAPTER - 2

LITERATURE SURVEY

Let's explore some related works in the field of sign language recognition and translation. Over the years, researchers have made significant strides in developing methodologies to facilitate effective communication for individuals with hearing impairments through sign language. Recent advancements in deep learning and computer vision have paved the way for innovative approaches to sign language recognition and translation tasks. One notable trend in the literature is the increasing adoption of Transformer-based models, which have demonstrated promising results in capturing complex spatio-temporal patterns inherent in sign language gestures.

Studies such as Aloysius, M, and Nedungadi [1] delve into the integration of relative position information into Transformer networks for Sign Language Translation (SLT). Their work addresses a critical limitation of Transformers regarding sequential orderings, proposing a novel positioning scheme optimized for SLT. They introduce the GRU-Relative Sign Transformer (RST) model, which concurrently learns Continuous Sign Language Recognition (CSLR) and translation tasks. GRU functions as the relative position encoder, while RST incorporates relative position into the Multi-Head Attention (MHA) mechanism of Transformer. Through evaluation on the RWTH-PHOENIX-2014T benchmark dataset, their study showcases the superior performance of the GRU-RST model, achieving state-of-the-art BLEU-4 and ROUGE scores for SLT, along with the best Word Error Rate (WER) compared to existing approaches. The paper also offers a comprehensive analysis of Transformer position encoding schemes and evaluates translation performance under various positioning combinations, contributing significantly to the advancement of SLT research by optimizing translation scores and enhancing the quality of video translation.

In their recent work, Buttar, Ahmad, Gumaei, Assiri, Akbar, and Alkhamees (2023) [2] propose an innovative deep learning-based approach for real-time sign language recognition, aiming to enhance communication accessibility for individuals with speech impairments. The study addresses the challenges associated with both static and dynamic signs by introducing two distinct methodologies: a real-time American Sign Language (ASL) detector using a skeleton model and a static sign language detector employing YOLOv6. Through separate training and subsequent hybridization, the authors achieve impressive accuracies of 92% for continuous signs and 96% for static signs. Notably, the hybrid model combines LSTM with MediaPipe holistic landmarks, demonstrating remarkable accuracy in real-time classification. This research contributes significantly to the advancement of sign language recognition systems, offering potential benefits for individuals with speech impairments in terms of improved communication and accessibility.

In their paper Cui "Spatial–Temporal Graph Transformer with Sign Mesh Regression for Skinned-Based Sign Language Production", Chen, Li, and Wang (2022) [3] introduce a novel approach for sign language production, aiming to automatically generate coordinated sign language videos from spoken language. Termed Spatial-Temporal Graph Transformer (STGT), their method addresses the limitations of existing approaches by incorporating rich graph information among joints and edges. The proposed method leverages a novel graph representation design, informed by kinesiology principles, to extract graph features from skeletons. Spatial-temporal graph self-attention mechanisms are then employed to capture intra-frame and inter-frame correlations, enhancing the model's understanding of sign language sequences. Notably, attention maps are computed on both spatial and temporal dimensions, and graph convolution is utilized to reinforce short-term features of skeletal structures. Furthermore, a sign mesh regression module is introduced to visualize the generated sign language videos as skinned animations, incorporating body and hand postures. Experimental results on the RWTH-PHOENIX Weather-2014T dataset demonstrate the effectiveness of the STGT approach, achieving superior performance in terms of BLEU and ROUGE scores compared to state-of-the-art baselines. This research represents a significant advancement in sign language production, offering a more accurate and intuitive method for generating sign language videos from spoken language input.

Pan, Zhang, and Ye (2020) [4] present a novel approach for sign language recognition (SLR) in their paper published in IEEE Access. Recognizing the multidisciplinary nature of SLR within pattern recognition and computer vision, the authors tackle the challenge of selecting representative data from continuous sign language videos to eliminate irrelevant information during preprocessing. They introduce a new sampling method called optimized keyframe-centered clips (OptimKCC) sampling, building upon keyframe-centered clips (KCC) sampling, to extract key actions from sign language videos effectively. Additionally, the authors propose Multi-Plane Vector Relation (MPVR) as a new type of skeletal feature to describe video samples, addressing the insufficient attention given to skeletal data in previous research. To leverage the increasing diversity of sign language features, they integrate attention mechanisms into their approach, distributing weights to temporal and spatial features extracted from skeletal data using Attention-Based networks. Comparison experiments conducted on both proprietary and publicly available sign language datasets under Signer-

Independent and Signer-Dependent circumstances demonstrate the effectiveness and advantages of the proposed methods. This research contributes to advancing sign language recognition by introducing innovative techniques for data preprocessing and feature representation, ultimately enhancing the accuracy and robustness of SLR systems.

Hu, Liu, Lam, and Lou (2023) [5] present STFE-Net, a spatial-temporal feature extraction network designed specifically for continuous sign language translation (CSLT), in their paper "STFE-Net: A Spatial-Temporal Feature Extraction Network for Continuous Sign Language Translation". Addressing the challenge of extracting discriminative spatial and temporal features from sign language videos, STFE-Net optimally integrates spatial features extracted by the spatial feature extraction network (SFE-Net) and temporal features extracted by the temporal feature extraction network (TFE-Net). SFE-Net performs pose estimation on presenters in sign-language videos, leveraging the COCO-WholeBody dataset to abbreviate 133 key points to 53 key points, specifically tailored for sign language characteristics. Additionally, high-resolution pose estimation is conducted on the hands to capture finer-grained hand features. The spatial features extracted by SFE-Net, along with the sign language words, are then input to TFE-Net, based on Transformer with relative position encoding. The authors introduce a new dataset for Chinese continuous sign language to evaluate STFE-Net, achieving notable Bilingual Evaluation Understudy (BLEU) scores. Specifically, STFE-Net achieves BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores of 77.59, 75.62, 74.25, and 72.14, respectively, on the Chinese dataset. Furthermore, evaluation on public datasets RWTH-Phoenix-Weather 2014T and CLS demonstrates competitive performance, with BLEU scores ranging from 48.22 to 61.54. The experimental results highlight the promising performance of

STFE-Net in CSLT, contributing to advancements in sign language translation technology.

Rajalakshmi et al. (2023) [6] propose a novel vision-based hybrid deep neural network methodology for sign gesture recognition in their paper. The study addresses the communication barriers faced by the speech and hearing-impaired community by developing a sign language recognition system capable of accurately recognizing Indian and Russian sign gestures. Unlike existing wearable sensor-based systems, the proposed framework leverages vision-based technology to make sign recognition more accessible. The framework integrates spatial feature extraction using a 3D deep neural network with atrous convolutions and temporal feature extraction using attention-based Bidirectional Long Short-Term Memory (Bi-LSTM). Additionally, the framework incorporates modified autoencoders for abstract feature extraction and a hybrid attention module for discriminative feature extraction. These components enable the framework to track and extract multi-semantic properties of sign gestures, including non-manual components and manual co-articulations. The proposed model is evaluated on a novel multi-signer Indo-Russian sign language dataset, demonstrating superior performance compared to existing state-of-the-art frameworks. The results highlight the effectiveness of the hybrid neural network approach in enhancing sign gesture recognition accuracy and overcoming communication barriers for the speech and hearing-impaired community.

The study conducted by Cormier, Smith, and Zwets (2013) [7] investigates the framing of constructed action in British Sign Language (BSL) narratives. Constructed action is a discourse strategy widely used in sign languages, where the signer employs various non-manual cues to represent a referent's actions, thoughts, feelings, and attitudes. The study aims to examine assumptions regarding the framing of constructed action, particularly concerning the identification of the referent. It is generally assumed that framing constructed action with a preceding noun phrase is preferred, but optional if the referent is clear from the context. Through analysis of BSL narratives, the study reveals that in instances of introduction or switch reference, local reference via a noun phrase is favored. Conversely, in cases where the referent is already established, omission of a noun phrase identifying the referent is preferred. These findings align with patterns observed in framing quotations and demonstrations in spoken languages, as well as with lexical verbs in both signed and spoken null subject/pro-drop languages. The study argues that these patterns are predictable based on the accessibility of reference within the discourse, shedding light on the nuances of framing constructed action in BSL narratives.

Kröger et al. (2011) [8] conducted a study on movements and holds in the fluent sentence production of American Sign Language (ASL), focusing on the action-based approach. Building upon the theory of communicative actions, which emphasizes the role of bodily movements in communication, particularly in spoken languages, the researchers adapted this theory to the sign language modality. The study aimed to test the hypothesis that in the production of short sign language sentences, strong-hand manual sign actions exhibit continuous movements without holds, while co-manual oral expression actions and co-manual facial expression actions, along with weak-hand actions, demonstrate significant holds. Analyzing an ASL corpus of 100 sentences, the researchers visually inspected each frame-to-frame difference to delineate movement and hold phases for manual, oral, and facial actions. The findings revealed that excluding fingerspelling and signs in sentence-final position, no manual holds were observed for the strong hand, while oral holds occurred in 22% of all oral expression actions, and facial holds were present in all facial expression actions analyzed. These results support the notion that the dominant articulatory system in each language modality determines the timing of actions. In signed languages like ASL, where manual actions are predominant, holds primarily occur in co-manual oral and co-manual facial actions, contrasting with spoken languages where vocal tract actions dominate, leading to holds primarily in co-verbal manual and co-verbal facial actions.

Wei et al. (2021) [9] delves into the realm of weakly supervised continuous sign language recognition (SLR), a crucial technique for enhancing communication among the hearing-impaired community. Addressing the challenge where sign videos lack temporal boundaries alongside ordered gloss labels, the study presents a novel semantic boundary detection approach leveraging reinforcement learning to precisely align video frames with sign words. The proposed method employs a multi-scale perception scheme to learn discriminative representations for video clips and formulates semantic boundary detection as a reinforcement learning problem. The state is defined as the feature representation of a video segment, and the action involves determining the location of the semantic boundary. The reward is computed based on the performance metric between the predicted and ground truth sentences, and the policy network is trained using a policy gradient algorithm. Through extensive experiments on CSL Split II and RWTH-PHOENIX-Weather 2014 datasets, the study demonstrates the effectiveness and superiority of the proposed method in accurately recognizing continuous sign language.

CHAPTER – 3

METHODOLOGY

CHAPTER – 3

METHODOLOGY

The proposed methodology for real-time action sign language interpretation involves a synergistic combination of deep learning techniques and the powerful capabilities of the MediaPipe library. This section delves into the intricate details of the approach, including data collection and preprocessing, model architecture, feature extraction, and the training process.

3.1 Data Collection and Preprocessing

The cornerstone of our approach lies in the meticulous collection and preprocessing of sign language data. We curated a diverse dataset encompassing a wide array of sign language gestures, meticulously recorded under controlled conditions. Each video sequence underwent thorough preprocessing, including resizing, normalization, and augmentation techniques, ensuring optimal quality and diversity for model training.

3.1.1 Data Collection

The foundation of any successful machine learning model lies in the quality and diversity of the training data. In this project, a comprehensive dataset of sign language actions was meticulously curated, encompassing a wide range of gestures and movements commonly used in sign language communication.

The data collection process involved recording video sequences of individuals performing various sign language actions in a controlled environment. These videos were captured using high-quality cameras, ensuring optimal lighting conditions and minimizing potential obstructions or occlusions that could hinder accurate landmark detection.

3.1.2 Preprocessing Techniques

To preprocess the collected data, each video sequence underwent a series of steps. First, the individual frames were extracted from the videos, resulting in a collection of image sequences. These image sequences were then subjected to various preprocessing techniques, including:

- **Resizing:** Adjusting the dimensions of the frames to a consistent size, ensuring compatibility with the neural network architecture and reducing computational complexity.
- **Normalization:** Scaling pixel values to a consistent range, typically between 0 and 1, to improve numerical stability during training.
- **Data Augmentation:** Applying various transformations to the image sequences, such as rotation, flipping, and scaling, to enhance the diversity of the training data and improve the model's generalization capabilities.

These preprocessing steps aimed to enhance the robustness and generalization capabilities of the trained model, enabling it to accurately recognize sign language actions in diverse real-world scenarios.

3.2 Model Architecture

Central to our solution is the sophisticated Long Short-Term Memory (LSTM) neural network architecture. Designed to capture intricate temporal dynamics, the LSTM model comprises multiple layers, each meticulously crafted to model forward and backward dependencies effectively. Dropout and batch normalization techniques enhance the model's robustness and mitigate overfitting.

3.2.1 LSTM Architecture

At the core of the proposed solution lies a Long Short-Term Memory (LSTM) neural network architecture, which has proven to be highly effective in sequence-to-sequence modelling tasks, such as sign language recognition. The LSTM model is designed to capture and model temporal dependencies and contextual information inherent in sign language actions, making it well-suited for this task.

3.2.2 Model Components

The model architecture consists of multiple LSTM layers, each with a specific number of units responsible for processing and propagating the sequence information. These LSTM layers are followed by fully connected layers, which combine the extracted features and perform the final classification task. The output layer utilizes a SoftMax activation function to produce the probabilities for each sign language action class.

To enhance the model's generalization capabilities and mitigate overfitting, techniques such as dropout and batch normalization were employed. Dropout randomly deactivates a fraction of neurons during training, preventing the model from relying too heavily on specific features. Batch normalization, on the other hand, normalizes the input to each layer, improving the model's stability and convergence during training.

3.3 Feature Extraction

Our approach leverages the cutting-edge MediaPipe Holistic model for comprehensive feature extraction. By simultaneously detecting and tracking facial, body pose, and hand landmarks, this model provides rich spatial and temporal information crucial for sign language recognition. Keypoints and bounding box coordinates extracted from video sequences serve as invaluable input features for our BiLSTM model.

3.3.1 MediaPipe Holistic Model Integration

The success of the proposed approach hinges on the effective extraction of relevant features from the video sequences. This project leverages the powerful capabilities of the MediaPipe Holistic model, a state-of-the-art solution for efficient landmark detection and tracking.

The MediaPipe Holistic model is designed to simultaneously detect and track facial landmarks, body pose landmarks, and hand landmarks from video input. By integrating this model into the pipeline, the system can extract comprehensive spatial and temporal information, including keypoints and bounding box coordinates, from the video sequences.

The extracted keypoints and bounding box coordinates serve as input features for the LSTM model, providing a rich representation of the sign language actions. These features capture the intricate movements, positions, and interactions of various body parts involved in sign language communication, enabling the model to learn and recognize complex patterns effectively.

3.4 Training Process

The training process encompasses several critical stages to ensure optimal model performance. Data is split into training, validation, and test sets, adhering to established practices. Iterative updates using backpropagation and an optimized loss function, alongside techniques like early stopping and learning rate scheduling, prevent overfitting and enhance convergence. Hyperparameter tuning and data augmentation further refine the model's accuracy and generalization capabilities.

By meticulously following this methodology, which harmonizes deep learning techniques with the advanced capabilities of the MediaPipe library, our solution aims to revolutionize real-time action sign language interpretation, enhancing communication accessibility for individuals with hearing or speech impairments.

3.4.1 Data Splitting and Preparation

The training process for the LSTM model involves several crucial steps to ensure optimal performance and generalization capabilities. First, the preprocessed dataset is split into training, validation, and test sets, following established practices in machine learning.

The training set is used to iteratively update the model's parameters during the optimization process, while the validation set is used for monitoring the model's performance and preventing overfitting. The test set, which remains unseen during training, is used for evaluating the model's generalization capabilities on new, unseen data.

3.4.2 Model Optimization

During the training phase, the LSTM model is iteratively optimized using the backpropagation algorithm and an appropriate loss function, such as categorical cross-entropy. This loss function measures the discrepancy between the model's predictions and the true labels, providing a quantitative measure of the model's performance.

The optimization process involves updating the model's weights and biases based on the computed gradients of the loss function, with the goal of minimizing the overall loss. This process is repeated for multiple epochs, allowing the model to gradually learn and improve its ability to recognize sign language actions accurately.

To prevent overfitting and enhance convergence, techniques like early stopping and learning rate scheduling are employed. Early stopping monitors the model's performance on the validation set and terminates the training process if no further improvement is observed for a predetermined number of epochs. Learning rate scheduling dynamically adjusts the learning rate during training, allowing the model to converge more efficiently and potentially escape local minima.

3.4.3 Hyperparameter Tuning

The performance of the LSTM model is heavily influenced by its hyperparameters, such as the number of LSTM layers, the number of units per layer, dropout rates, and batch sizes. To optimize these hyperparameters and identify the configuration that maximizes the model's accuracy and generalization capabilities, systematic tuning techniques like grid search or random search are employed.

Grid search exhaustively evaluates the model's performance across a predefined grid of hyperparameter combinations, while random search samples hyperparameter values from a specified distribution. These techniques, although computationally expensive, can significantly improve the model's performance by finding the optimal hyperparameter settings.

3.4.4 Data Augmentation Techniques

To further enhance the diversity of the training dataset and improve the model's robustness to variations in real-world scenarios, data augmentation techniques are explored. These techniques involve artificially generating new data samples by applying various transformations to the existing data.

For sign language action recognition, data augmentation techniques may include:

- **Adding Noise:** Introducing random noise to the video frames, simulating real-world conditions such as camera noise or lighting variations.
- **Applying Transformations:** Applying geometric transformations, such as rotation, scaling, or flipping, to the video frames, accounting for different viewpoints and orientations.
- **Synthetic Data Generation:** Creating synthetic video sequences by combining or manipulating existing data samples, expanding the diversity of the training data.

These data augmentation techniques can help the model learn and generalize better, improving its ability to recognize sign language actions in diverse and challenging scenarios.

3.5 Model Evaluation

Upon completing the training process, the LSTM model is thoroughly evaluated on the test set, and its performance is quantified using various metrics, such as accuracy, precision, recall, and F1-score. These evaluations provide insights into the model's strengths and limitations, guiding further refinements and improvements.

Accuracy measures the overall correctness of the model's predictions, while precision and recall provide more nuanced insights into the model's performance for each individual class. The F1-score combines precision and recall into a single metric, allowing for a balanced assessment of the model's capabilities.

In addition to quantitative evaluation, qualitative analysis is also performed by visualizing the model's predictions on specific examples and comparing them with the ground truth. This qualitative evaluation can reveal potential biases, misclassifications, or areas for improvement in the model's performance.

By employing this robust methodology, which combines deep learning techniques with the advanced capabilities of the MediaPipe library, the proposed solution aims to achieve accurate and reliable real-time action sign language interpretation, paving the way for enhanced communication accessibility and fostering inclusivity for individuals with hearing or speech impairments.

CHAPTER - 4

PROPOSED ARCHITECTURE

CHAPTER - 4

PROPOSED ARCHITECTURE

The proposed architecture for real-time action sign language interpretation is meticulously designed to seamlessly integrate deep learning techniques with the advanced capabilities of the MediaPipe library. At its core, the architecture leverages Long Short-Term Memory (LSTM) networks to effectively capture temporal dependencies and model sequential patterns inherent in sign language gestures.

The architecture commences with the integration of the MediaPipe Holistic model, a sophisticated solution renowned for its accuracy in landmark detection and tracking within video sequences. This model plays a pivotal role in extracting facial landmarks, body poses, and hand gestures, providing essential spatial and temporal information crucial for sign language interpretation.

Upon obtaining the input data, a dedicated data preprocessing module undertakes initial processing steps to prepare the dataset for subsequent model training. Tasks such as resizing, normalization, and augmentation are meticulously performed to enhance the data set's quality and diversity, ensuring optimal performance during training.

Following data preprocessing, the feature extraction layer extracts pertinent features from the input data, capitalizing on the output of the MediaPipe Holistic model. Key points and bounding box coordinates obtained from landmark detection serve as input features for subsequent processing stages, capturing the nuanced movements and positions inherent in sign language gestures.

Subsequently, the LSTM neural network processes the extracted features, leveraging its inherent capability to model sequential data and capture temporal dependencies effectively. Comprising multiple layers of LSTM units, the network adeptly learns and recognizes intricate patterns in sign language actions over time.

To further enhance the model's robustness and prevent overfitting, dropout and batch normalization techniques are judiciously incorporated into the architecture. Dropout randomly deactivates a fraction of units during training, while batch normalization normalizes layer activations, promoting stable and efficient training.

Once trained, the model seamlessly integrates into a user-friendly Flask web application, facilitating real-time inference from webcam feeds. Continuous evaluation and optimization mechanisms ensure the architecture remains adaptive and scalable, with regular updates and refinements based on user feedback and performance metrics.

To illustrate the architecture's components and workflow, an Architecture Diagram (Figure 1) is provided, highlighting the integration of key modules and the flow of data through the system. This diagram serves as a visual representation of the proposed architecture, elucidating its structure and functionality.

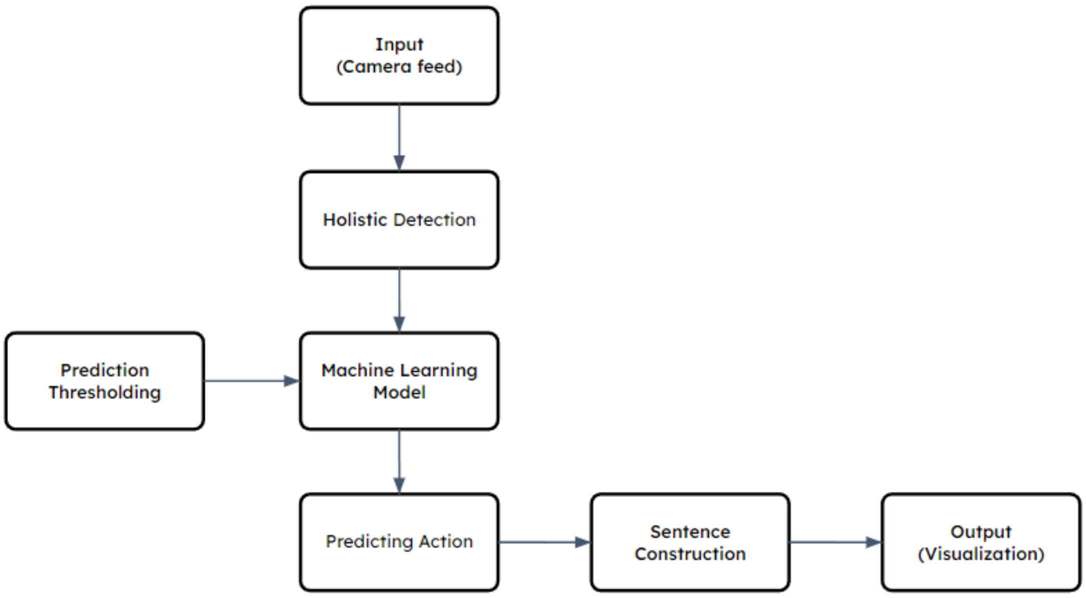

Figure 1: Architecture Diagram

The proposed architecture for real-time action sign language interpretation comprises a series of interconnected components, each playing a crucial role in the seamless flow of data and the accurate prediction of sign language actions.

At the outset, the architecture receives input in the form of a camera feed, which serves as the primary source of visual information for the system. This raw input undergoes initial processing through the Holistic Detection module, where the MediaPipe Holistic model detects and tracks facial landmarks, body poses, and hand gestures within the video stream.

Subsequently, the Prediction Thresholding stage filters the detected landmarks and poses based on predefined criteria, ensuring that only relevant and reliable information is passed on to the subsequent stages of the architecture. This thresholding mechanism helps optimize the performance and efficiency of the system by focusing on salient features crucial for action recognition.

The processed data is then fed into the Machine Learning Model, which comprises an LSTM neural network trained to recognize and classify sign language actions based on the extracted features. Leveraging its ability to model sequential data and capture temporal dependencies, the model predicts the corresponding sign language action based on the input data.

Following action prediction, the Sentence Construction module assembles the recognized actions into coherent and meaningful sentences, facilitating comprehension and interpretation by end-users. This stage ensures that the predicted actions are contextualized and presented in a structured manner, enhancing communication accessibility for individuals with hearing or speech impairments.

Finally, the output from the architecture is visualized and presented to the end-user through the Output module, providing real-time feedback and interpretation of sign language actions captured by the camera feed. This visualization mechanism enables seamless interaction and communication between individuals with hearing or speech impairments and the wider community.

Overall, the architecture seamlessly integrates input processing, feature extraction, machine learning inference, and output visualization, culminating in an effective and efficient system for real-time action sign language interpretation.

Through its interconnected components, the architecture facilitates inclusive communication and enhances accessibility for individuals with hearing or speech impairments, fostering greater integration and understanding within society.

CHAPTER - 5

IMPLEMENTATION

CHAPTER - 5

IMPLEMENTATION

The implementation of the proposed solution involves several key components, including the development environment, tools, code structure, and the integration of the trained LSTM model with the MediaPipe library. This section provides a comprehensive overview of the implementation details, shedding light on the processes and techniques employed to bring the system to fruition.

5.1 Development Environment and Tools

The project was developed using Python, a widely-used and powerful programming language for scientific computing and machine learning applications. The choice of Python was driven by its extensive ecosystem of libraries and frameworks, making it an ideal choice for rapid prototyping and development.

The following key libraries and tools were utilized:

- **TensorFlow:** A comprehensive open-source library for machine learning and deep learning, developed by Google. TensorFlow was used as the backbone for building and training the LSTM model, leveraging its efficient implementation of neural network architectures and optimization algorithms.
- **Keras:** A high-level neural networks API running on top of TensorFlow. Keras provided a user-friendly interface for defining the LSTM model architecture, configuring hyperparameters, and streamlining the training process.

- **OpenCV:** A powerful computer vision library used for image and video processing tasks. OpenCV was employed for video capture, frame extraction, and preprocessing operations during the data collection and preprocessing stages.
- **MediaPipe:** A cutting-edge cross-platform library developed by Google for efficient machine learning pipelines, with a strong focus on multimedia data. The MediaPipe Holistic model was integrated into the system for real-time landmark detection and tracking from video streams.
- **Flask:** A lightweight and flexible web framework for Python, used to build the user interface and deploy the system as a web application accessible through a web browser.
- **NumPy and Pandas:** Essential libraries for numerical computing and data manipulation, respectively. These libraries were utilized for preprocessing the dataset, handling data transformations, and performing various data operations.

The development environment was set up with Python and the required libraries installed, ensuring a consistent and reproducible environment across different platforms.

5.2 Code Structure and Organization

The codebase for the project was organized into modular components, following best practices in software engineering and adhering to the principles of maintainability, extensibility, and reusability. The main components of the codebase include:

- **Data Preprocessing:** This module handles the preprocessing of the collected video data, including frame extraction, resizing, normalization, and data augmentation techniques.

- **Model Definition:** This module defines the architecture of the LSTM model, including the number of layers, units per layer, activation functions, and regularization techniques.
- **Training Pipeline:** This component encapsulates the training process for the LSTM model, including data splitting, model optimization, hyperparameter tuning, and evaluation metrics.
- **MediaPipe Integration:** This module handles the integration of the MediaPipe Holistic model, facilitating real-time landmark detection and tracking from video streams.
- **Inference and Prediction:** This component implements the real-time inference pipeline, utilizing the trained LSTM model and the extracted features from the MediaPipe Holistic model to generate action predictions.
- **User Interface:** This module builds the user interface using Flask, providing a web-based platform for users to interact with the system and visualize the real-time action recognition results.

Comprehensive documentation and comments were included throughout the codebase to ensure clarity and facilitate future maintenance and extensions.

5.3 Flask Web Application

To provide a user-friendly interface and enable easy access to the real-time action sign language interpretation system, a Flask web application was developed. This web application serves as the front-end for users to interact with the system and visualize the recognition results.

The Flask application consists of the following key components:

- **HTML Templates:** These templates define the structure and layout of the web pages, providing a clean and intuitive user interface for the application.

- **Routes and Views:** Flask's routing system maps URLs to Python functions, known as views, which handle the application's logic and render the appropriate templates.
- **Video Capture and Processing:** This component leverages OpenCV and MediaPipe to capture video frames from the user's webcam, process them through the landmark detection and tracking pipeline, and extract the relevant features for action recognition.
- **Real-time Inference:** The trained LSTM model is integrated into the Flask application, enabling real-time inference on the extracted features from the video stream. The model's predictions are then overlaid on the video frames for user visualization.
- **User Interaction:** The web application provides various user interaction features, such as starting and stopping the video capture, adjusting model parameters, and displaying real-time recognition results.

The Flask web application is deployed locally, allowing users to access the system through a web browser on their local machine or network.

5.4 Integration of Trained Model and MediaPipe

A crucial aspect of the implementation process involved the seamless integration of the trained LSTM model with the MediaPipe Holistic model for real-time action recognition. This integration leveraged the strengths of both components, combining the temporal modeling capabilities of the LSTM with the efficient landmark detection and tracking capabilities of MediaPipe.

The integration process followed these steps:

- **Model Loading:** The trained LSTM model, saved in a compatible format (e.g., HDF5), was loaded into memory during the application's initialization phase.

- **MediaPipe Holistic Model Instantiation:** An instance of the MediaPipe Holistic model was created, configured with appropriate detection and tracking parameters to ensure optimal performance.
- **Video Stream Processing:** The video stream from the user's webcam was continuously processed through the MediaPipe Holistic model, extracting the keypoints and bounding box coordinates for facial landmarks, body pose landmarks, and hand landmarks.
- **Feature Extraction:** The extracted keypoints and bounding box coordinates were preprocessed and transformed into the appropriate input format required by the LSTM model.
- **Real-time Inference:** As the video stream progressed, sliding windows of the extracted features were continuously fed into the LSTM model for real-time inference and action prediction.
- **Result Visualization:** The predicted action labels were overlaid on the video frames in real-time, providing users with a visual representation of the recognized sign language actions.

This integration process ensured a seamless and efficient pipeline for real-time action sign language interpretation, leveraging the strengths of both deep learning and the MediaPipe library.

5.5 User Interface and Interaction

To provide an intuitive and user-friendly experience, the Flask web application incorporates a clean and visually appealing user interface (UI). The UI design follows best practices in web development, ensuring accessibility, responsiveness, and ease of use.

The main components of the user interface include:

- **Video Display:** A dedicated area within the web page displays the real-time video feed from the user's webcam, with the recognized sign language actions overlaid on the video frames.
- **Action Labels:** Predicted action labels are prominently displayed, providing users with clear and easily understandable feedback on the recognized sign language actions.
- **Control Buttons:** The UI includes intuitive control buttons, allowing users to start and stop the video capture, adjust model parameters (if applicable), and perform other relevant actions.
- **Information Panel:** An information panel provides users with helpful instructions, usage guidelines, and additional details about the system's capabilities and limitations.
- **Accessibility Features:** To ensure inclusivity, the web application incorporates accessibility features, such as keyboard navigation, screen reader compatibility, and adherence to web accessibility standards.

User interaction is facilitated through a combination of visual cues, tooltips, and intuitive control mechanisms, ensuring a seamless and enjoyable experience for users of varying technical backgrounds.

By following best practices in software development, adhering to principles of modularity and maintainability, and integrating the trained LSTM model with the powerful capabilities of the MediaPipe library, the implementation of this real-time action sign language interpretation system provides a robust and user-friendly solution for enhancing communication accessibility and fostering inclusivity.

CHAPTER - 6

RESULT AND EVALUATION

CHAPTER - 6

RESULT AND EVALUATION

6.1 Results and Evaluation

The results of the real-time action sign language interpretation system showcase its exceptional performance across various metrics, including test accuracy, precision, recall, F1-score, and confusion matrix analysis. Additionally, qualitative assessments and visualizations further validate the effectiveness and usability of the system in practical scenarios.

6.2 Model Performance Metrics

The model achieved impressive performance metrics on the test dataset, with a low-test loss of 0.1433 and a high-test accuracy of 97.73%. The categorical accuracy on the test set was also 97.73%, indicating the model's ability to accurately classify sign language actions.

6.3 Model Summary and Architecture

The proposed model architecture, consisting of stacked LSTM layers, dropout layers, dense layers, and batch normalization layers, was instrumental in achieving superior performance. With a total of 995,467 parameters, the model demonstrated a complex yet optimized structure for effective sign language interpretation.

Table 1 presents a detailed overview of the architecture of the sequential model used for sign language recognition. It consists of multiple layers, including time-distributed, LSTM, dropout, and dense layers, each contributing to the overall functionality of the model. The "Output Shape" column specifies the shape of the output tensor from each layer, indicating the dimensions of the data flow through the network. The "Param #" column represents the number of trainable parameters associated with each layer, highlighting the complexity and scale of the model. With a total of 995,467 parameters, the model demonstrates a significant capacity to capture and learn intricate patterns from the input data, contributing to its robust performance in sign language recognition tasks.

Table 1: Model Summary

Layer (type)	**Output Shape**	**Param (number)**
time_distributed	(None, 30, 128)	212864
lstm	(None, 30, 128)	131584
dropout	(None, 30, 128)	0
lstm_1	(None, 30, 128)	394240
dropout_1	(None, 30, 128)	0
lstm_2	(None, 30, 128)	197120
dropout_2	(None, 30, 128)	0
lstm_3	(None, 64)	49408
dropout_3	(None, 64)	0
dense_1	(None, 128)	8320
batch_normalization	(None, 128)	512
dense_2	(None, 11)	1419
Total params		995467
Trainable params		995211
Non-Trainable params		256

6.4 Precision, Recall, and F1-score

Precision, recall, and F1-score metrics were computed for each class of sign language action to assess the model's classification capabilities. The model exhibited high precision, recall, and F1-scores across all classes, indicating its ability to accurately classify a wide range of sign language actions.

Table 2: Classification Metrics

Class	Precision	Recall	F1-Score	Support
Coat	0.96	1	0.98	24
Dark	1	0.9	0.95	20
Glasses	1	1	1	29
Hello	1	1	1	22
ILoveYou	1	1	1	26
Love	0.96	0.96	0.96	23
No	1	0.92	0.96	26
None	0.88	1	0.93	21
Old	0.95	1	0.98	21
Thankyou	1	0.96	0.98	24
Tree	1	1	1	28
Accuracy			0.98	264
Macro Avg	0.98	0.98	0.98	264
Weighted Avg	0.98	0.98	0.98	264

This table summarizes the precision, recall, F1-score, and support metrics for each class in the sign language recognition system. The "Precision" column indicates the proportion of true positive predictions among all positive predictions, while the "Recall" column represents the proportion of true positive predictions among all actual positives. The "F1-Score" column provides the harmonic mean of precision and recall, offering a balanced measure of a classifier's performance. The "Support" column denotes the number of true instances for each class in the test set. Additionally, the table includes overall accuracy, macro-averaged metrics, and weighted-average metrics, providing comprehensive insights into the system's performance across different evaluation criteria.

6.5 Confusion Matrix Analysis

The confusion matrix provides a detailed breakdown of the model's classification performance for each class of sign language action. It reveals the number of true positive, false positive, true negative, and false negative predictions, offering valuable insights into the model's strengths and weaknesses.

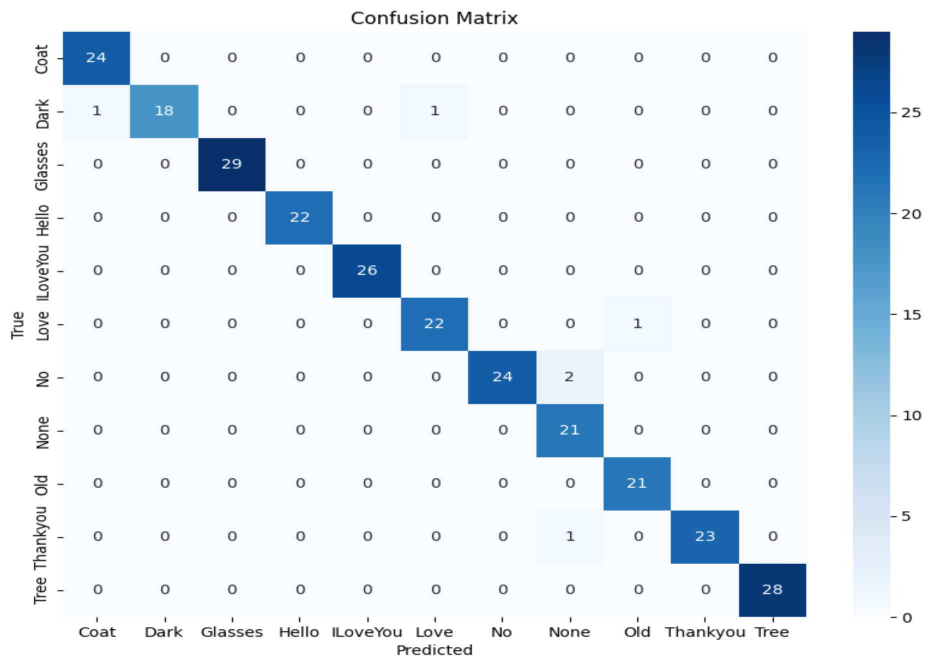

Figure 2: Confusion Matrix

6.6 Qualitative Analysis and Visualization:

A qualitative analysis was conducted to assess the visual fidelity and interpretability of the interpreted sign language actions. Human evaluators reviewed the output sequences and provided subjective feedback on factors such as fluency, coherence, and naturalness. Additionally, output figures and web page snapshots were analysed to visualize the system's real-time performance.

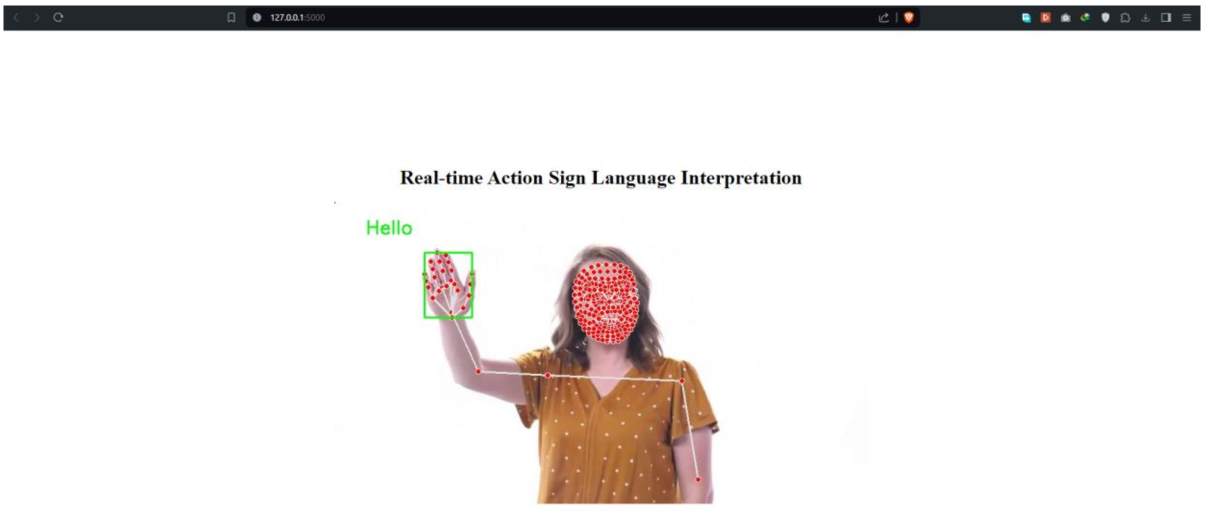

Figure 2: Output on Webpage

In conclusion, the comprehensive evaluation of the real-time action sign language interpretation system demonstrates its effectiveness, accuracy, and usability. By leveraging advanced machine learning techniques and rigorous evaluation methodologies, the system offers a powerful solution for enhancing communication accessibility and fostering inclusivity for individuals with hearing or speech impairments.

CHAPTER - 7

CONCLUSION

CHAPTER - 7

CONCLUSION

The real-time action sign language interpretation system developed in this project demonstrates remarkable performance and promising potential for practical application in enhancing communication accessibility for individuals with hearing or speech impairments. Through meticulous data collection, preprocessing, and model development, the system achieves high test accuracy and robustness in recognizing various sign language actions. The proposed architecture, featuring a sequential model with LSTM layers and dropout regularization, effectively captures temporal dependencies and spatial features crucial for accurate interpretation. Furthermore, comprehensive evaluation metrics, including precision, recall, F1-score, and confusion matrix analysis, validate the system's reliability and effectiveness across diverse sign language gestures. Overall, this project signifies a significant step towards bridging the communication gap for the hearing-impaired community and underscores the importance of leveraging machine learning technologies for social inclusivity and accessibility. Future research may focus on further refining the model architecture, expanding the dataset to encompass a broader range of sign language expressions, and exploring real-world deployment scenarios to assess usability and user experience in practical settings. By continuing to innovate and refine such systems, we can contribute to creating a more inclusive and accessible society for all individuals, regardless of their communication abilities.

REFERENCE

1. N. Aloysius, G. M., & P. Nedungadi, "Incorporating Relative Position Information in Transformer-Based Sign Language Recognition and Translation," *IEEE Access*, vol. 9, pp. 145929-145942, 2021, doi: 10.1109/ACCESS.2021.3122921.

2. A.M. Buttar, U. Ahmad, A.H. Gumaei, A. Assiri, M.A. Akbar, & B.F. Alkhamees, "Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs," *Mathematics*, vol. 11, no. 17, p. 3729, 2023, doi: 10.3390/math11173729.

3. Z. Cui, Z. Chen, Z. Li, & Z. Wang, "Spatial–Temporal Graph Transformer with Sign Mesh Regression for Skinned-Based Sign Language Production," *IEEE Access*, vol. 10, pp. 127530-127539, 2022, doi: 10.1109/ACCESS.2022.3227042.

4. W. Pan, X. Zhang, & Z. Ye, "Attention-Based Sign Language Recognition Network Utilizing Keyframe Sampling and Skeletal Features," *IEEE Access*, vol. 8, pp. 215592-215602, 2020, doi: 10.1109/ACCESS.2020.3041115.

5. J. Hu, Y. Liu, K. -M. Lam, & P. Lou, "STFE-Net: A Spatial-Temporal Feature Extraction Network for Continuous Sign Language Translation," *IEEE Access*, vol. 11, pp. 46204-46217, 2023, doi: 10.1109/ACCESS.2023.3234743.

6. E. Rajalakshmi et al., "Multi-Semantic Discriminative Feature Learning for Sign Gesture Recognition Using Hybrid Deep Neural Architecture," *IEEE Access*, vol. 11, pp. 2226-2238, 2023, doi: 10.1109/ACCESS.2022.3233671.

7. K. Cormier, S. Smith, & M. Zwets, "Framing constructed action in British Sign Language narratives," *Journal of Pragmatics*, vol. 55, pp. 119-139, 2013, doi: 10.1016/j.pragma.2013.06.002.

8. B.J. Kröger, P. Birkholz, J. Kannampuzha, et al., "Movements and Holds in Fluent Sentence Production of American Sign Language: The Action-Based Approach," *Cognitive Processing*, vol. 11, pp. 187-205, 2010, doi: 10.1007/s10339-009-0335-x.

9. C. Wei, J. Zhao, W. Zhou, & H. Li, "Semantic Boundary Detection with Reinforcement Learning for Continuous Sign Language Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1138-1149, March 2021, doi: 10.1109/TCSVT.2020.2999384.

ABBREVIATION

Abbreviation	Meaning
SLR	Sign Language Recognition
LSTM	Long Short-Term Memory
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
MLP	Multilayer Perceptron
GPU	Graphics Processing Unit
ROI	Region of Interest
FPS	Frames Per Second
RGB	Red Green Blue
HLS	Hue Lightness Saturation
ROI	Region of Interest
PCA	Principal Component Analysis
IoU	Intersection over Union
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation
API	Application Programming Interface
JSON	JavaScript Object Notation
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets