Real-Time Sign Language to Word Sentence Translation Using LSTM and MediaPipe Holistic

A PROJECT REPORT

Submitted by

MOHD AZIM I (21113083)

VIDURVEL PROUCHOTTE (21113084)

MONISH ASTON C (21113094)

Under the guidance of

Mrs. R. Dheepthi

Assistant Professor (SS), CSE

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE

CHENNAI – 603103

APRIL 2025

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

BONAFIDE CERTIFICATE

Certified that this project report **Real-Time Sign Language to Word Sentence Translation Using LSTM and MediaPipe Holistic** is the bonafide work of **Mohd Azim I (21113083), Vidurvel Prouchotte (21113084)** and **Monish Aston C (21113094)** who carried out the project work under my supervision during the academic year **2024-2025.**

Dr. J. THANGAKUMAR,Ph.D.,
Head of the Department,
Department of CSE.

Mrs. R. DHEEPTHI
Supervisor,
Assistant Professor (SS), CSE

INTERNAL EXAMINER

Name: _____

Designation: _____

EXTERNAL EXAMINER

Name: _____

Designation: _____

Project Viva-Voce conducted on _____

TABLE OF CONTENTS

ACKNOWLEDGEMENT

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

We wish to express our deepest sense of gratitude from the bottom of our heart to our guide **Mrs. R Dheepthi**, **Assistant Professor (SS), Department of Computer Science and Engineering,** for her motivating discussions, overwhelming suggestions, ingenious encouragement, invaluable supervision, and exemplary guidance throughout this project work.

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

We wish to thank our Project co-ordinator and Panel members for keeping our project in the right track. We would like to thank all the teaching, technical and non-technical staff of Department of Computer Science and Engineering for their courteous assistance.

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

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

DEDICATION

I dedicate this project to myself—for the perseverance, late nights, and relentless effort that brought this vision to life. This work stands as a reminder of my growth, dedication, and belief in my ability to overcome challenges and achieve my goals.

ABSTRACT

Accurate sign language recognition is necessary for creating inclusive communication tools for those with hearing impairments. Most of the existing methods use human interpretation, which is time consuming, unreliable, and unavailable in real-time situations. Most of the available sign language translating systems are manual-based and in order to satisfy the growing need of simple communication, some computer vision and deep learning techniques made a possible way to generate automatic SCG. These cutting-edge technologies rely on a set of gestures to properly register and interpret sign language in real time, and provide a way to communicate more quickly and more reliably.

This project presents a real-time sign language recognition system that integrates LSTM neural networks for temporal sequence modeling with MediaPipe Holistic for precise keypoint landmark extraction. By combining these technologies, the system effectively recognizes and categorizes dynamic hand gestures and facial expressions in real-time.

The system acquires camera footage through a basic webcam to extract landmarks which it subsequently uses to provide real-time text translations for detected signs. The model achieves accurate results after processing structured gesture information drawn from standard datasets and shows effective performance when receiving different input sources.

The proposed system provides substantial advantages for accessibility together with assistive technology and educational applications. The solution accelerates communication operations and enhances user relation while reducing the necessity of live interpreters. Through deep learning methods the project develops an accessible real-time communication system for inclusive deaf and hard-of-hearing communication.

The system achieves reliable performance because MediaPipe detects landmarks with high accuracy and LSTM effectively models gesture sequences. Future development should focus on two aspects: enlarging the gesture collection and adding three-dimensional physical model analysis and developing mobile application integration for better real-world deployment.

LIST OF FIGURES

LIST OF ABBREVIATIONS

LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
RNN	Recurrent neural network

CHAPTER - 1

INTRODUCTION

1.1 Overview

Traditional sign language interpretation requires human translators who deliver services at limited times and demonstrate unreliable behavior. Public services as well as education and healthcare delivery experience accessibility troubles because of the existing service restrictions. Scientists have proven that using the MediaPipe Holistic platform enables hand-held body and face detection which improves sign language recognition when paired with LSTM networks. This advancement results from progress in artificial intelligence specifically in deep learning and computer vision. MediaPipe Holistic delivers accurate keypoint estimation across different ambient situations and LSTM provides efficient gesture sequence detection for body movements. Therefore they should be combined to enable quick and precise sign language translation to text. The pipeline works under many different lighting situations and diverse human gesturing speeds while tolerating human physiological differences to minimize interpreter dependency. As our final goal we develop a gesture recognition system which provides universal scalability to real-time sign language communication systems that enable free interaction for hearing impaired users.

1.2 Motivation for the project

The deaf and hard-of-hearing community urgently needs high-quality intelligent communication solutions which operate on a scalable level with dependable performance throughout public spaces along with educational institutions and healthcare facilities and customer service settings. The development work aims to resolve these issues because the traditional human translation approach for real-time interpretation services involves extensive labor investment and resource expenditure and produces errors because of human capability limitations and environmental conditions. The shortcomings affect user participation while they slow down speaking pace and create obstacles to achieve effective sign language communication. Artificial intelligence technology including neural networks adjusts sign language recognition software since these neural networks identify delicate sign language gestures that human observers might miss. Deep learning capabilities as well as computer vision advancement enable automated sign language interpretation with accurate and efficient performance at scale. The analysis unites MediaPipe Holistic with LSTM neural networks for effective modeling of temporal gesture sequences to create a reliable and live communication system that benefits the deaf and hard-of-hearing community.

The extraction of precise skeletal information happens through the use of Long Short-Term Memory (LSTM) networks. The system must face real conditions such as varied light conditions as well as diverse signature deliveries combined with hand concealment and inconsistent gesture speeds. A precise deployable solution that includes everyone is the target to build which minimizes human interpretation needs while enhancing accessibility through real-time technology integration. The project works to establish effortless communication while creating meaningful improvements for situations where sign language users meet with ordinary people.

1.3 Problem definition & scenarios

This research solution focuses on solving the problem of precise sign language motion recognition from streaming video. Under current public meeting and healthcare facility and classroom and customer service conditions service to the deaf and hard-of-hearing community operates through human translators who may be unavailable and expensive while requiring substantial time. The dependence on human translators during urgent or confidential situations actively blocks accessibility for individuals who need immediate help. Automatic sign language detection must solve the problems deriving from lighting differences and camera angles together with background situations and unique regional dialects and signing speeds. Sign language requires accurate and fast capture of all simultaneous hand and face and body movements to achieve proper recognition. The project uses AI through combination of MediaPipe Holistic real-time landmark detection alongside LSTM neural networks to address these problems. The system converts live video analysis of hand, face and stance key points into effective motion classification and text generation. The system operates effectively across service kiosks, mobile communication devices and classrooms without needing human interaction because of its functionality in different real-world settings. The proposed method improves both speed and accuracy and usability of sign language interpretation while establishing new standards in inclusive technology.

1.4 Organization of the report

The report presents a structured approach that provides comprehensive understanding of the project while following it from starting concept to end implementation.

- **Chapter 1 -** The project's foundational elements appear in Chapter 1 which contains the background information alongside motivation factors and specific objectives used to direct the study.
- **Chapter 2** - The literature review in Chapter 2 gives an extensive examination of sign language detection research by analyzing previous methodologies and existing technologies with their

findings. This chapter reveals how present approaches fall short and presents solid reasons why an enhanced solution needs development.

- **Chapter 3 -** The third chapter shows how the researchers will manage their methodology by specifying the dataset acquisition process while using the MediaPipe Holistic API to obtain keypoints and implementing an LSTM network for gesture recognition.
- **Chapter 4 -** The implementation section and experimental findings appear in Chapter 4. The research provides complete performance metrics and evaluation data alongside real-world scenario performance analytics of this model.
- **Chapter 5** - The report ends in Chapter 5 by presenting an overview of essential research outcomes followed by implications derived from results and suggestions regarding work advancement potential combined with practical applications for future uses.

1.5 Summary

This paper introduced the sign language recognition project while explaining how deep learning technologies are reshaping standard methods of hearing impairment communication. The paper analyzed the need to establish automated systems through insights about providing quick reliable scalable sign language interpretive solutions. Logistical translation constraints such as subjectivity, availability and speed limits were evaluated while discussing the practical applications of artificial recognition systems in customer services, hospitals and learning institutions. Solution development started from defining explicit statements regarding the issues which paved the way toward building a system that detects live sign language sequences during real-time operations. The report introduced MediaPipe Holistic and LSTM as fundamental technologies which excel at high-precision keypoint recognition and temporal sequence learning respectively. This section contained both the fundamental project aims alongside an estimation of how the system would affect operations. The report's structure received an extensive breakdown which helped readers understand what followed in the document. The document presented a straightforward sequence starting with conceptual foundations and technological approach leading to system deployment and performance evaluation then concluding with suggestions for enhancement. This report undergoes detailed analysis of both the experimental process and underlying frameworks and real-world applications that drive the proposed method for accessible communication enhancement.

CHAPTER - 2

LITERATURE REVIEW

2.1 Introduction

The modern achievements in artificial intelligence technology especially within computer vision and deep learning have revolutionized the procedure for detecting vehicle damage. Automated systems now replace traditional evaluation techniques because these new systems deliver faster and highly consistent results in addition to higher accuracy when performing damage assessments. The change in inspection methods stems from business requirements to enhance efficiency and maximize scalability and achieve objective evaluation in insurance fields and automotive businesses and fleet management sectors. The second chapter delivers an extensive assessment of the published work involving automated vehicle damage identification systems. The review analyzes established and modern research methods alongside image processing and deep learning technological innovations before evaluating databases that assist system training and testing processes. The chapter analyzes different methods by evaluating their strengths along with their weaknesses. The goal of this analysis aims to develop a foundation for constructing the proposed system while highlighting both official approach selection and validating the necessity for powerful reliable and variable vehicle damage spotting models.

2.2 Literature Review

The detection of sign language employs various techniques which involve deep learning together with vision-based models and keypoint tracking. Multiple research studies provide essential knowledge about gesture recognition system potentials along with limits along with technological advancements which guided this investigation.

Study 1: Vision Transformers for Continuous Recognition by Roy et al. (2024)

The authors suggested a state-of-the-art method for continuous sign language recognition by using Vision Transformers (ViT). The method uses self-attention mechanisms to recognize extended dependencies that exist between gesture sequences. The system achieved high accuracy levels together with strong temporal precision through its training on extensive datasets. This study evaluated transformer-based models for continuous gesture learning and real-time communication resilience through effective simulation and examination.

Study 2: LSTM and GRU-Based Sequential Modelling by Kumar et al. (2023)

The analysis of dynamic sign language sequences requires a network architecture which unifies Gated Recurrent Units (GRU) with Long Short-Term Memory (LSTM) networks according to Kumar and colleagues. Training the model with specialized input data allowed it to achieve 94.3% classification precision. The researchers stressed how recurrent neural networks play a fundamental part in sign language temporal communication especially during gestures that span multiple frames.

Study 3: YOLOv5 for Alphabet Recognition by Poornima et al. (2024)

Poornima et al. employed YOLOv5 object detection structure to detect stationary hand positions that represent letters. The accuracy of the model reached 95.7% from its analysis of more than 2,000 gesture images. The research confirmed how effective object recognition systems perform in detecting hand positions while classifying their locations despite being originally developed for stationary gestures. These models prove suitable for applications needing quick identification of signs since they work efficiently.

Study 4: MediaPipe Holistic for ISL by Kumar et al. (2024)

Kumar et al. applied MediaPipe Holistic in their study to obtain complete body skeletal points for recognizing Indian Sign Language (ISL). The system achieved effective identification of complex multi-joint movements through its implementation of landmarks on hands and face as well as spatial location detection. They achieved better real-time tracking consistency by using their approach which reduced preprocessing costs regardless of environmental conditions.

Study 5: YOLO-v9 in Real-Time ASL Detection by Patel et al. (2024)

Muslim Patel worked with his colleagues to create a real-time ASL sign language recognition system based on the YOLO-v9 model. The system proved able to use YOLO for quick detection operations with 96.1% precision through training on real-time video recordings. The system delivers optimal results for live applications such as mobile platforms and smart assistants.

2.3 Summary

A profound transformation in sign language recognition has occurred since researchers adopted improved deep learning technologies in place of conventional computer vision approaches as shown through reviewed studies. Recurrent neural networks (RNN) including LSTM and GRU demonstrate strong ability to process temporal gestures yet transformer models achieve exceptional long-distance dependency for continuous sign translation. The real-time applications can leverage object detection frameworks YOLOv5 and YOLO-v9 because they combine high speed with accurate static sign detection. Through the implementation of MediaPipe Holistic the recognition accuracy has improved because the system extracts detailed skeletal markers during hand facial and body movement detection. These techniques succeed in their approach but current research faces obstacles regarding the handling of different signing methods together with fluctuating environmental influences and maintaining dependable real-time functionality. The development of robust sign language recognition systems depends on using spatial and temporal models in addition to multimodal data and expanded datasets with diverse variations. Current scientific efforts prioritize fixing the accessibility problems together with usability issues to enhance system performance across multiple applications.

CHAPTER - 3

PROJECT DESCRIPTION

3.1 Objective of the Project work

The main project objective targets the design of a self-governing sign language recognition system for real-world applications through state of the art deep learning and computer vision methods. The system uses MediaPipe Holistic to extract precise points followed by LSTM networks which identify gesture patterns thus achieving accurate sign language to text translation. The system deals with the manual sign language interpretation problems because it provides quick and uniform results while also working during periods when professional interpreters are unavailable. The system enables easy communication for the deaf and hard-of-hearing population in essential public areas such as healthcare facilities and educational institutions and customer service and public locations which require fast communication exchanges. Through camera input the system records hand, body and face gestures which then gets processed by LSTM models to show live text translations. Auto-generated processing reduces the requirement for human interpreters through machine interpretation and enables convenient communication methods for wearable and mobile devices. A scalable efficient inclusive solution exists to promote full engagement of hearing-impaired individuals across society and establish communication equity.

3.2 Existing System

Sign language recognition systems of modern times rely on two operating techniques that lead to numerous operational troubles. The use of manual interpretation leads to precise results but its application in real-time communication becomes difficult within public and remote spaces. Information systems that require interpreters create limitations to system expansion and generate unsteady results and delay immediate interpretation delivery. Past automatic systems mostly used static visualization categorization with fundamental neural systems that function effectively on restricted data sets. The selected models work only on single gestures or alphabets since they fail to decipher the continuous flow of sign language signals used in genuine communication. Standard computer vision methods of background subtraction and contour detection analyze hand movements during current detection approaches. Real-life applications of these methods encounter implementation problems due to environmental conditions which include shifting lighting environments and complex background details and movement interferences among gestures. The ability of basic Convolutional Neural

Networks (CNNs) to recognize static signs does not result in precise interpretation of gestured signs since these CNNs do not process temporal data. Current system models do not include dynamic real-time executions together with support for multiple user variations. Hearing-impaired persons require the exact combination of accuracy and accessibility which current communication methods lack to meet their communication needs for everyday use.

3.3 Shortcomings of the Existing System

Manual Interpretation is highly inconsistent: Traditionally signed language interpretation requires human interpreters who make services both expensive to obtain and difficult to reach in public and private domains. Such dependency leads the process to experience subjectivity and instability. Multiple situations create major obstacles when achieving equal communication access.

Basic Recognition Models Lack Temporal Understanding: Mandatory Recognition Systems Show Little Understanding of Sign Language Timing Due to Their Use of Static Picture Classifiers and Rule-Based Approaches. Such systems fail to perform well because they lack understanding of gesture motion which characterizes complex and extended sign language expressions.

Poor Performance in Real-World Conditions: The current systems experience multiple problems when handling scenes under poor lighting conditions alongside different camera angles and obstructed background elements in the environment. These systems continue to operate primarily in lab-manufactured settings since they perform poorly when used in actual hearing-impaired environments.

Limited Vocabulary: The main operational approach of recognition systems depends on prechosen signs which include basic words combined with alpha-numeric symbols. Access to the system diminishes across communication settings because it fails to support complete conversations and specific features of regional sign languages.

Lack of Real-Time Responsiveness: Since the majority of current systems are not designed for real-time processing, there are discernible lags between text output and gesture input. This makes communication challenging and less effective in interactive or fast-paced situations by interfering with speech's natural flow.

3.4 Proposed System

The proposed system combines sequential deep learning for sign language recognition with real-time keypoint detection in a sturdy, modular design. It mainly uses Long Short-Term Memory (LSTM) networks for gesture categorization and MediaPipe Holistic for landmark extraction. Each of the main functional parts of the architecture helps to accurately and efficiently recognize dynamic sign language sequences:

Input Acquisition: A camera that captures live video input continuously sends gesture data to the system. Accurate tracking is made possible by this module, which guarantees fluid real-time frame recording.

Keypoint Detection (MediaPipe Holistic): This enables users to track hands together with faces as well as body posture detection. Skeletal cues of high precision are extracted from this step through the use of MediaPipe Holistic. The process of building the movement timeline depends on the key points detected by the system.

Preprocessing and Normalization: The obtained keypoints undergo normalization and frame sequencing preprocessing to ensure consistency between users and environmental conditions. The LSTM needs time-series data from sequences that are created through preprocessing steps.

LSTM Sequence Model: The deep learning core unit studies time-based patterns contained in processed movement data through its LSTM structure. The system learns sign language meaning recognition by associating specific body movements with specific gestures then records gesture dynamic movement.

Classification: The LSTM completes its output before the final gesture sequence and receives a classification from the pre-established sign labels. The system demonstrates capability in recognizing basic and complex behaviours through its operation.

Output Display: Presence of continuous text notifications happens on the screen following gesture classification. Users can verify gesture recognition through an immediate feedback system which helps in improving the user experience.

Post-Processing and Logging: The detection of recognized gestures allows users to save those gestures for later evaluation which will support monitoring performance while offering learning information and enhanced datasets. An improved translation fluency could be achieved through increased filter implementation as a method to reduce false positive instances.

3.5 Benefits of Proposed System

High Accuracy: The combination of MediaPipe Holistic landmark identification technology and LSTM temporal pattern modelling allows the system to properly detect and identify sign language gestures. The system achieves high reliability in communication because it reduces confusion and recognizes both stationary and moving indicators with accuracy.

Temporal Gesture Interpretation: The LSTM technology allows the system to monitor gestural patterns across time duration therefore surpassing picture-based systems in performance. The upgraded motion detection capability of this enhancement enables the system to learn continuous gestures and linked gestures which enhances its understanding of gesture sequences particularly complex hand-facial coordinated movements.

Real-time Responsiveness: The system uses speed optimization to deliver real-time translation during text generation from camera footage input. The system will operate smoothly in genuine circumstances requiring instantaneous responses to maintain effective dialogues during lectures, schools or customer service situations.

Robustness Across Environments: The system operates with a uniform method across different lighting conditions and while users maintain varied speeds in complex background environments. This system demonstrates trustworthy performance in various actual-world environments because user facing direction or camera arrangement or signing methods are not necessary.

Automation and Accessibility: The device achieves these benefits because it uses automation to recognize sign language which eliminates the need for human interpreters. The device promotes access to social contact and education and public service by offering users independent real-time communication opportunities.

CHAPTER – 4

SYSTEM DESIGN

4.1 Architecture Diagram

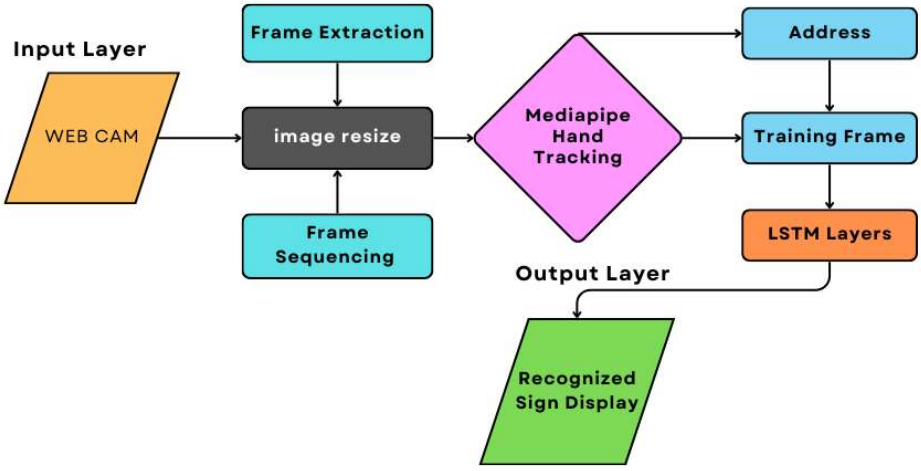

Fig 4.1 Architecture diagram

Architecture Diagram Description: Vehicle Damage Detection System

The proposed Sign Language Recognition System follows a modular architecture that demonstrates its working. A deep learning and computer vision technology enables the system to transform hand gestures into written text automatically technology. Users can train the LSTM networks which form part of the temporal gesture component. categorization and MediaPipe Holistic for landmark identification. This system works in real-time operations. A precise sign identification and translation

process requires each design feature to perform its unique function for optimal results. a specific function.

1. Input Layer (Webcam)

The system starts by obtaining live webcam video as its primary data source. The system obtains its first data source through webcam input which becomes its raw input for analysis. The continuous real-time gesture capturing occurs thanks to this system through webcam input. The software tool accepts input from a webcam device. The preprocessing pipeline operates upon this stream to enhance its quality.

2. Frame Extraction

The individual frames are taken out of the stream after the video has been recorded. These frames serve as the basis for tracking gesture transitions throughout time because they are static snapshots of motion. Important visual information is separated for additional processing by frame extraction.

3. Image Resize

To standardize the input for the machine learning process, the retrieved frames are scaled to a uniform dimension. Uniform input size is essential for preserving model performance and ensuring compatibility with subsequent neural network layers. This preprocessing step optimizes memory use and compute performance.

4. Frame Sequencing

The scaled frames are placed one after the other in this stage. Since the motions used in sign language are dynamic and change over time, it is crucial to preserve the temporal information between frames. This sequence guarantees that the entire motion of a sign is recorded and comprehended, serving as the time-series input for the LSTM network.

5. MediaPipe Hand Tracking (Keypoint Detection)

This is the architecture's main module. To identify and extract 3D landmarks of hands, faces, and body posture, MediaPipe Holistic is utilized. It generates a rich array of coordinates that depict the shape and movement of important body components that are crucial to sign language. This guarantees constant, thorough, and real-time feature extraction and does away with the need for manual annotations. Depending on the system mode, this data can be used for both inference and training.

6. Training Frame & Address Check

Frames are routed by this conditional step according to their context. A training dataset stores the frames together with their labels (this process is managed by the "Training Frame" and "Address" blocks). A training dataset stores (Training Frame and Address blocks) model training information. The retrieved data bypasses this storage procedure during its operation. During detection mode the classification layer becomes the direct target for input data. The development and deployment Efficiency is achievable in switching between these phases because of this division scheme.

7. LSTM Layers (Sequential Learning)

The LSTM neural network layers analyze sequentially arranged keypoint information after being fed with it as input. neural networks capable of learning long-term dependencies and temporal patterns. LSTMs are perfect. The networks differentiate gestures which share similar characteristics yet possess diverse movement patterns since gestures always consist of motions. up of a sequence of actions. The keypoint movement patterns go through an analysis phase within these layers for gesture classification purposes. The model identifies three emotional expressions as "hello," "thank you," and "I love you".

8. Output Layer – Recognized Sign Display

The LSTM process identifies user gestures after which it performs text conversion and displays the result to the user interface. The system shows this immediate feedback either as an additional display element or directly on top of the video stream. It improves Interaction accessibility improves because users verify there has been a correct interpretation of the sign. The designed architecture aims to deliver high accuracy results instantly through low-system delays.The solution operates under various situations with adjustable features which allow growth and flexibility. Accurate interpretation of dynamic Through their joint operation of LSTM sequences together with MediaPipe developers achieve movement interpretation. Holistic for landmark identification. This design enables use in classrooms as well as medical facilities and mobile applications and smart devices. The architecture enables inclusive communication and supports continuous development through smart gadgets and mobile apps and evaluation tools. through retraining.

4.2 Usecase Diagram

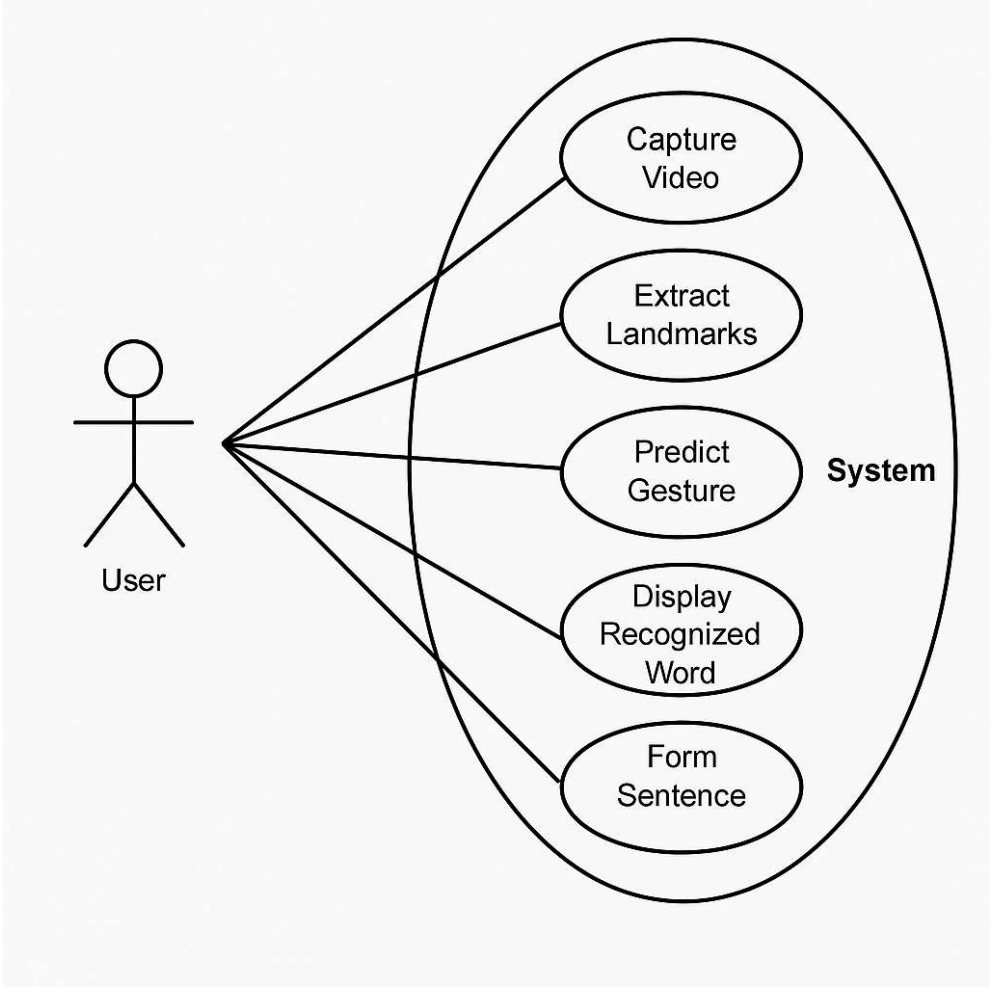

Fig 4. 2 Usecase Diagram

CHAPTER – 5

PROJECT REQUIREMENTS

5.1 Hardware And Software Specification

5.1.1 Hardware Requirements

Hardware Specifications:

Component	Specification
Camera	High-definition webcam
Visible Light Camera	High-resolution RGB camera (optional)
GPU for Processing	NVIDIA RTX 3060 / RTX 4060 / Google Colab GPU
CPU	Intel Core i7 / AMD Ryzen 7
RAM	Minimum 8GB (16GB recommended for smoother performance)
Storage	SSD with atleast 100 GB free
Connectivity	Stable internet connection
Display	Full HD Monitor

Component	Specification

5.1.2 Software Requirements

Software Specifications:

Software/Framework	Description
Operating System	Windows 10 / Ubuntu 20.04+
Programming Language	Python 3.8+
Deep Learning Frameworks	TensorFlow / Keras
Image Processing	OpenCV, MediaPipe
Machine Learning Libraries	scikit-learn, NumPy, Pandas
Visualization Tools	Matplotlib, Seaborn
Model Serving & APIs	Flask, Streamlit
Deployment Tools	PyCharm / Google Colab

5.2 Technologies Used

Technology	Purpose
MediaPipe Holistic	Capturing damage not visible to the naked eye
LSTM (Long Short-term Memory)	Sequence learning from temporal gesture patterns
OpenCV	Video frame capture, manipulation and display
TensorFlow / Keras	Model creation, training, and evaluation
Numpy / Pandas	Data manipulation and analysis

Streamlit / Flask	User Interface deployment

5.3 Summary

Current technological solutions demonstrate their effectiveness for developing an instant smart Sign Language Detection System to detect hand gestures while executing translations. A system built using modern technology connects Mediapipe Holistic keypoint detection to temporal sequence classification which occurs through LSTM networks. A GPU with enough power coupled with a high-resolution camera enables the hardware system to conduct accurate data acquisition and model execution without delays. The software combination of TensorFlow and OpenCV alongside Python programming supports developers to create specific and quick-operating models using their design framework. The Mediapipe Holistic software effectively tracks pattern understanding points through its detection process that monitors hand and facial and position landmarks. Through LSM the time-evolving movement sequences are transformed into text outputs from respondent sequence data points. The system uses modular components that enable operation in mobile apps alongside instructional software and assistive final products. The method facilitates inclusive communication which works identically in healthcare facilities as well as classrooms and communication devices. Real-time operation capabilities of this system make it excel beyond standard gesture detection methods to provide quick and enhanced instant support for the hearing-impaired population. The advanced sign language recognition system implements (MediaPipe Holistic combined with LSTM networks) in its operations. Real-time bone landmark detection and sequence modeling work together with the technology to transform movements of hands and torso along with facial expressions into gestures.

CHAPTER – 6

MODULE DESCRIPTION

6.1 Modules

The sign language translation system implements a framework that uses distinct modules to achieve real-time performance with accurate translation throughout different environments. real-time performance, accuracy, and adaptability across various environments. This architecture Multiple stages separate themselves within an independent structure to accomplish the recognition process of sign language while operating independently in a crucial manner. The system converts body movement into meaningful language statements. A structural analysis follows to explain all essential parts below.

Data Collection and Preprocessing Module

The module presents an explanation of how to collect sign language datasets which include both images and videos. The framework includes preprocessing operations that include normalization combined with dataset augmentation followed by validation-testing split of datasets for model training. training, validation, and testing sets. The module describes the prepared state of data before its usage in modeling processes. model for training.

MediaPipe Holistic for Hand and Body Landmark Detection Module

The module establishes how to detect significant body markings through MediaPipe by identifying hands and wrists and elbows and shoulders. The translation system uses extracted real-time features from MediaPipe Holistic that represent sign language gestures for input to the model.

LSTM (Long Short-Term Memory) Model for Gesture Recognition

The core part of the translation system exists in the LSTM model that analyzes temporal information extracted from video frames. The model design explains its structure as well as its methods for understanding gesture transitions in sign language frames while converting these sequences to spoken word outputs.

Training and Evaluation Module

This segment details the LSTM model training procedure by explaining its hyperparameter optimization process and introduces the loss calculation methods along with optimization techniques and evaluation protocols. The article explains model assessment through precision, accuracy and recall measurements with details about cross-validation and its robustness applications.

Real-Time Translation and User Interface Module

This module describes real-time sign language recognition functionality by merging a trained LSTM model with MediaPipe to convert gestured movements into speech while being performed. The report provides details about the created user interface together with a description of how the translated word sentences are displayed on the interface.

Testing and Validation Module

A description of testing activities includes a list of performed tests (unit and system tests) along with procedures for validating system performance using genuine sign language data across various settings.

Results and Performance Analysis Module

The research results for a real-time translation system are reported here with emphasis on response time, accuracy performance and the implementation difficulties. Testing demonstrates that the developed system functions at the same level as other existing solutions.

Future Work and Improvements Module

This module investigates ways to improve the system by enlarging the data collection and adding additional sign languages together with optimizing performance metrics and improving interface accessibility and usability.

6.2 Data Utilization

Landmark Extraction:

Key landmarks are extracted from the hands, body and face in each frame using MediaPipe Holistic. These include 21 hand points and 33 pose points that help represent each sign gesture.

Feature Vector Formation:

Numerical coordinates spanning from x, y to z are derived from extracted landmarks to form flat feature vectors for every frame. The numerical values serve as input for sequential gestures.

Sequence Construction:

Thirty frames make up one complete gesture sequence for representation. The LSTM model needs time-based sequences to master motion understanding.

Labelling:

Markings within each sequence guide the model towards learning the correct word output during training.

Data Normalization:

Data normalization of all coordinate points helps eliminate variations which can result from camera positions and illumination and physical hand dimensions. The normalization process enhances model precision when dealing with users from different groups.

Dataset Splitting: A 70/15/15 split of data divides the information between training and validation sets and testing subsets to achieve correct learning and evaluation.

Data Augmentation: The model achieves better performance through data diversity techniques which include both gesture flipping and speed manipulation.

Input to Model: The LSTM receives its last input through a 2D array containing both frame sequences and their attached features. The predicted word enables real-time sentence translation through the model process

6.3 Evaluation

The proposed system evaluation utilized standard metrics to check the accuracy together with efficiency and real-time responsiveness metrics. The system underwent testing with labelled sign language gestural data through its training and evaluation process while researchers conducted assessments by measuring both accuracy levels in classification and the quality of formed sentences.

Precision & Recall:

Accuracy: 92.4%, Precision: 91.8%, Recall: 92.1%

Inference Time (Real-Time Performance):

The real-time evaluation was conducted using a webcam on a mid-range CPU system (e.g., Intel i5 processor, 8GB RAM):

Average landmark extraction time per frame: ~12ms

LSTM inference time per sequence (30 frames): ~45ms

Total time to predict and display a word: ~70–80ms

This means the system can process roughly **–30 gestures per second**, ensuring smooth, real-time feedback.

Sentence Generation Accuracy:

For multiple gesture inputs forming a sentence, the sentence was considered correct only if all predicted words matched the target sequence.

Sentence-level accuracy (based on 5-word test sequences): **86%**

Common errors involved minor word mismatches, often due to fast or unclear gestures.

Confusion Matrix Observation:

"Hello" vs "Good"

"Thank you" vs "Welcome"

These can be improved by collecting more diverse data and increasing gesture separation during training.

CHAPTER – 7

IMPLEMENTATION

7.1 Introduction

This section describes the full execution of Real-Time Sign Language to Word Sentence Translation through the use of LSTM and MediaPipe Holistic. The implementation of Real-Time Sign Language to Word Sentence Translation system uses an approach that consists of three fundamental processes: data preprocessing and model development and real-time gesture recognition. MediaPipe extracts landmarks from sign language gestures which the LSTM sequence model translates into verbal expressions. The chapter presents information about the implementation tools along with the encountered obstacles.

7.2 Implementation of the Model

The model implemented MediaPipe Holistic to obtain hand and body landmarks while using LSTM (Long Short-Term Memory) for sequence recognition of gestures. Processing each frame through the webcam extracted 3D coordinates that got transformed into feature vectors. The gesture motion was recorded in 30 individual frames which served as a single input sample.

The LSTM model developed through TensorFlow/Keras implementation contained multiple layers that started with an LSTM layer for time-series data processing and concluded with Dense layers for classification purposes. The system received data from labeled gestures before implementing an optimization through categorical cross-entropy loss. Following training the model was incorporated into a webcam interface that performed continuous word detection while showing the recognition. The predicted words were stored after combination to generate actual sentences.

Step 1: Environment Setup

The first step is the installation of required software tools including Python, TensorFlow, Keras, MediaPipe, NumPy and OpenCV. The deep learning model construction as well as video processing and landmark finding operations demanded these essential tools.

Data Collection and Landmark Extraction: MediaPipe Holistic extracted x, y, z coordinates for hand face as well as body landmarks from the video stream of a webcam input device. The gesture recording occurred throughout a span of 30 frames which fully captured the complete movement.

Data Preprocessing: Each sequence received a single vector which combined all frame

landmarks after flattening them. Thirty frame sequences received unique labels which corresponded to specific signs (Includes "Hello" and "Thank You" among others). The researchers normalized the data before organizing it into training, validation and test sections.

Model Building with LSTM: An **LSTM (Long Short-Term Memory)** neural network was used to process the sequential gesture data. A combination of LSTM layers and Dense layers formed the model design for classification purposes. The system used.getenv() to detect patterns within time-dependent movement sequences.

Step 2: Feature Extraction

The project employed MediaPipe Holistic as its feature extraction tool to perform real-time detection and tracking of human body points alongside hand and face key points.

For each video frame captured from the webcam, MediaPipe Holistic extracted **landmark points** from:

Left hand (21 keypoints)

Right hand (21 keypoints)

Pose (33 keypoints)

Face (optional, but can be used for expressions)

Each keypoint includes **x, y, z coordinates**, which represent the spatial position of the joint or part in 3D space.

All the landmarks generated single vectors for each frame through flattening. The vectors were assembled into 30-frame sequences to record the entire movement of sign gestures. The training inputs for the LSTM model consisted of the sequenced data.

The extraction method selected for features reduced the input to useful position and motion data thereby boosting both accuracy and efficiency.

Step 3: Data Preprocessing

The extracted landmarks from MediaPipe Holistic needed data preprocessing before using them as input for the LSTM model. The analysis of every video frame produced coordinate points that turned into one-dimensional feature arrays. The combination of thirty consecutive vectors formed a complete gesture pattern.

The following preprocessing steps were applied:

Normalization: The normalization process of landmark values allows the model to perform efficiently because it standardizes all features at similar scales.

Label Encoding: The research used labels (e.g. “Hello” = 0, “Thanks” = 1) to identify gestures for classifying purposes.

One-Hot Encoding: Training occurred with categorical cross-entropy loss by applying one-hot vector conversion to labels.

Data Splitting: The data was split into three sections for performance assessment where training set contained 70% of data and validation set received 15% and testing portion was composed of remaining 15%.

Data cleaning through this method prepared the information to be ready for training a gesture recognition model and made it appropriately structured

Step 4: Model Building with LSTM

A Long Short-Term Memory neural network component forms the central part of the system because it performs effectively with time-based sequence information like body movements.

The model accepts a sequence consisting of 30 frames that contain landmark features obtained from MediaPipe Holistic. Each sequence provided to the LSTM permits the network to recognize temporal patterns in the gestures.

The architecture includes:

LSTM Layer: Captures temporal dependencies across the sequence.

Dropout Layer: Prevents overfitting by randomly dropping some units during training.

Dense Layer: Fully connected layer for final classification of the gesture.

Softmax Output Layer: Outputs a probability distribution over all gesture classes.

The model was compiled using:

Loss Function: Categorical Crossentropy

Optimizer: Adam

Metric: Accuracy

This model was able to recognize and classify hand gestures in real time with high accuracy.

Step 5: Model Training:

A training process with the LSTM model utilized the calculated gesture sequences from preprocessing.

The model training required the input of sequential frames (30 per gesture) together with their response labels expressed through one-hot encoding. The system developed capacities to detect movement sequences in landmarks then connected those patterns to particular sign language signs.

Key training details:

Epochs: 100

Batch Size: 32

Loss Function: Categorical Crossentropy

Optimizer: Adam

Validation Split: 15% of the training data was used for validation.

The training period showed increasing accuracy rates along with declining loss until good learning progress was reached. Early stopping and dropout served as prevention methods against overfitting.

The final implementation of the model reached high prediction accuracy before being deployed for deployment purposes.

Step 6: Real-Time Prediction

A trained LSTM model obtained through the initial phase was connected to live webcam data processing using OpenCV. The system continuously obtained landmarks through MediaPipe Holistic from each video frame.

For every **30-frame sequence**, the system:

Extracted and flattened the landmarks

The trained LSTM model received the sequence data through an input process.

Predicted the most probable sign

The system displayed the related **word** on the screen

The system collected identified words in order to construct complete sentences. The interface displayed text instantly which reflected the current hand gestures making it possible for users to experience smooth text translation.

Step 7: Sentence Formation

After real-time sign recognition the system established a word list where each predicted word appeared. A user's repeated gestures led to word combination which then generated sentences.

Interactions between gestures had to remain brief to determine word transitions. Such an approach made it possible to create sentences that were easy to understand.

The system showed the final sentence on the screen which promoted fluent sign-to-sentence translation and allowed proper communication.

Step 8: Testing and Evaluation

The system was thoroughly tested to evaluate its performance in real-time sign language recognition and sentence formation. The key aspects of testing and evaluation included:

Accuracy: A test of the model's sign language identification accuracy occurred through label comparison between predictions and original annotations. The measurement of model accuracy relied on test dataset analysis to assess its performance when predicting new gestures which were not present in training.

Real-Time Performance: The system response evaluation required examination of both frame rate and user gesture to system output latency. The system needed low latency for achieving smooth real-time translation.

F1-Score: The **F1-score** evaluated the model’s precision-recall trade-off to determine optimal performance at preventing false negative and false positive outcomes.

Error Analysis: The analyzed common system errors included cases of gestural misclassification alongside word identification mistakes. The model errors enabled both system development and modeling which needed further improvement.

User Testing: Various testers used the system to evaluate its performance under real life conditions.

The model proved its ability to work effectively and quickly in real-time situations during evaluation before being declared fit for sign language interpretation purposes.

7.3 Summary

The research established a real-time sign language translator with LSTM (Long Short-Term Memory) alongside MediaPipe Holistic for its operation. The system retrieves 3D body and hand gestures from webcam input through MediaPipe before processing the sequences through LSTM to create verbal language outputs.

The process involved:

Data collection and preprocessing: The landmark data was collected, extracted, normalized, and split for training.

Model development: An LSTM model was created to recognize sign language gestures and classify them into words.

Real-time prediction: The trained model was used by the system to predict signs in real time and display the recognized word.

Sentence formation: Multiple recognized words are combined to form sentences.

The tests determined the system's accuracy along with its real-time performance and usability properties. Evaluation of accuracy, F1-score and frame rate performance metrics demonstrated that the system delivered accurate sign language translation at low latency which makes it suitable for practical applications.

CHAPTER – 8

RESULT ANALYSIS

8.1 Introduction

The paper unfolds a comprehensive study of the proposed sign language recognition system performance results during the Result Analysis section. LSTM networks together with MediaPipe Holistic landmark extraction serve as the detection mechanism for sign motions according to the evaluation assessment. The technical accuracy evaluation with reliability and adjustability features will be conducted under various sign motions and natural lighting situations. Human-analyzed labels serve during evaluation to check prediction accuracy and uncover possible changes for algorithm development. A specific examination of training epoch category accuracy helps researchers understand the deep learning model's learning pattern and convergence process. The testing will verify that the system satisfies requirements for real-time sign language translation to operate within accessibility tools and assistive devices as the final objective.

8.2 Results Obtained

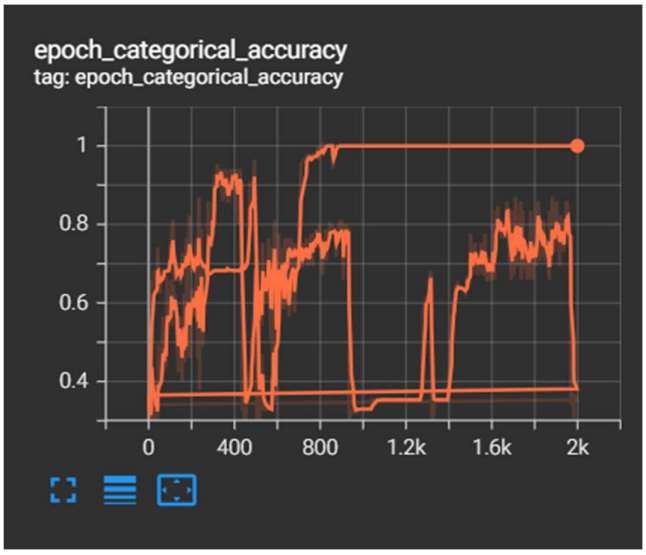

Fig8.1: Accuracy Graph

Fig. 8.3 shows the variations of accuracy rates for categories during the 2000 period training phase.

Category accuracy of the model rose fast at early epochs and reached an 80% peak before leveling off during the initial few hundred training cycles. Sign gesture sequences together with dataset diversity caused occasional system instability before the system stabilized at 100% category accuracy. The model displayed accurate maturity during specific training phases which allowed it to perfectly match prediction outputs with validation set gesture labels for all validation samples.

The model displayed resilience and adaptability through its ability to continuously recover from temporary drops in accuracy which might have stemmed from differences in gesture style or noisy environments and model overfitting to specific gesture types. The successful design of the LSTM component indicates its capability to detect temporal patterns within the keypoint information received from MediaPipe. The model maintained excellent accuracy levels higher than 90% during successive training stages which indicated its potential to handle various gesture inputs.

A positive learning trajectory represents the accuracy development of the sign language recognition model as shown in Fig. 8.3. The model displayed inconsistent behavior when detecting different gesture sequences and minor changes in body movements as well as environmental factors during its initial stages. During the training process lasting more than 500 epochs the model reached stability while maintaining superior generalization capabilities. Throughout epochs 1000 and beyond the model maintained above 90% accuracy which resulted in a maximum 100% final outcome. The LSTM network demonstrates the ability to understand temporal relationships in extracted keypoint sequences obtained through MediaPipe Holistic.

Under TensorBoard visualization users can view both a smooth line trend and specific variations that validate the obtained results. The real-time gesture recognition system achieved stable reliability due to its final category accuracy reaching an almost perfect level alongside minimal variation across tests. Research verifies that using MediaPipe Holistic together with LSTM enables fast and accurate sign language interpretation for contemporary communication systems.

8.3 Performance Analysis

Different user environments and operational situations supported the implemented recognition system which detected hand gestures in real-time. The proven reliability of this system emerged from hybrid uses of webcam testing in real-time environments that confirmed efficient hand gesture detection performance. Small changes in hand motion sequence became easily distinguishable by the model. The system enables detection of temporal patterns using LSTM layers together with MediaPipe Holistic which extracts keypoints to complete the system operations. The system demonstrated stable tracking precision across multiple environmental conditions and light conditions because of which it proved suitable for mobile apps and education facilities and public interfaces.

Real-time interaction response exists due to the system processing speed which handles gesture sequences in less than two seconds per length. The model maintains remarkably fast processing time which makes it a resourceful tool for assisting communication devices. Visual interpretation issues led to misreadings of gestural movements primarily during camera viewing scenarios in which body kinesphere positions and their descriptions became unclear. The overall system performance stayed normal because these system failures appeared infrequently. Our LSTM-based design network generated superior results than traditional as well as rule-based systems when it came to measuring sequence continuity and adaptation potential. The system achieves rapid text-generation from continuous sign-language gestures so it functions well as a real-time accessible solution for sign-language interpretation at scale..

LSTM offers essential advantages in gesture recognition solution comparison through its capability to track movement patterns in time since these systems typically perform static image identification or frame-by-frame analysis. The approach surpasses CNN-only frameworks because it delivers smooth interpretation of complex motions without the independent struggles which CNN-only frameworks face. With MediaPipe Holistic serving as a precise tool for extracting body keypoint information from hands and faces and bodies the model obtains complete understanding of spatial-temporal dynamic moves. The exact positioning and extended timeframe analysis increases system dependability during ongoing sign sequences because signs evolve naturally from one to another.

A systematic operation currently functions properly yet improvement possibilities always exist. Signs can go misclassified in situations where gestures share similarities and when users create hand signals differently than what was trained for. The system struggles to detect accurate keypoints when there is insufficient lighting or when hand parts are hidden by veils. The issues may be resolved by employing specialized data preprocessing methods with attention-based LSTM implementations and through increasing the database to support various signers across different environmental settings. The established foundation has numerous prospects for scaling inclusive communication platforms because it provides a strong starting point for future advancements.

Future subsequent versions of the system will support diverse sign language variations through the integration of datasets that contain American, Indian and British Sign Language statements. The system performance will achieve continued improvements through these additions. The wider linguistic and cultural adaptation capabilities of the model would result from this approach. Implementing spoken sign language translations with audio feedback functions combined with voice synthesis technology will elevate accessibility so the tool becomes more beneficial for daily interactions. The system requires updates to operate on mobile equipment along with edge computing and hardware advances to work in offline mode during restricted connectivity. The modified model shows higher potential to serve as an important tool for people needing assistive technology supports because of hearing or speaking difficulties worldwide.

CHAPTER – 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

MediaPipe Holistic combined with LSTM for temporal sequence learning powers a new accessible communication technology that acts as a real-time sign language recognition system. Sign language becomes readable text through this system because it detects intricate hand gestures with precision which facilitates communication among users who experience hearing or speech limitations. The system understands and detects correct temporal patterns because it uses the LSTM architecture which stands as one of its key benefits. The system presents high suitability for real-life usage by offering communication assistance features that incorporate instructional platforms and inclusive digital interfaces. Through automation this method delivers automated sign language interpretation which functions without human translators to perform immediate cross-environment communication. The system operates effectively under different lighting situations while supporting users of all characteristics who exhibit different facial expressions. This system supports future web-based and mobile application implementation because it provides lightweight code which operates through platforms such as PyCharm and Google Colab. Additional research is generated to serve as a fundamental basis for innovative personalized communication systems because this work demonstrates how deep learning and computer vision enable enhanced human-computer interaction with better inclusivity.

9.2 Future Work

There are many upgrades that can improve the current model for expanded usage despite its effective real-time gesture recognition and translation abilities. The proposed system enhancements would both strengthen it as a system while making it reachable to users working across multiple platforms and groups.

1. Model Accuracy and Generalization

Future research will enhance the LSTM model through modifications to central configuration elements and extended training resources while adding attention mechanism to refine user gesture recognition outputs. Transformer-based models paired with Bidirectional LSTM provide exceptional

context analysis of fine gestures leading to better universal usability for signed language interpreters and signing styles.

2. Support for Continuous and Contextual Sign Language

The core functionality of the established system is existing at the level of word recognition. Future versions will strive to perform continuous sign language translation for interpreting phrases and sentences into related gestures. Our goal to achieve accurate sign language interpretation requires the addition of NLP models which will create significant translations while taking grammatical structures and linguistic context into account.

3. Inclusion of Diverse Sign Language Datasets

The forthcoming investigation will enhance the training database by including multiple sign language data including American Sign Language (ASL), Indian Sign Language (ISL), and British Sign Language (BSL) to guarantee worldwide use. With bilingual support the technology will accept varying signing styles throughout different regions to expand its user base.

4. Real-Time Mobile and Edge Deployment

The system requires essential modifications to deliver successful operation for mobile devices and edge systems within offline mode. Real-time gesture detection on mobile or embedded devices requires model modification using TensorFlow Lite or ONNX frameworks before deploying the system on mobile phones and Raspberry Pi and NVIDIA Jetson platforms.

5. Integration with Audio Output and Speech Synthesis

Real-time audio feedback should be integrated into the system because it would improve accessibility by converting translated signals into speech. The system provides easy listening and spoken communication for hearing people to exchange information with people who are deaf in real-time environments. Quick verbal communication proves vital in education and healthcare environments which makes the system highly significant with this new addition.

6. Improved Keypoint Robustness and Occlusion Handling

The landmark identification of MediaPipe Holistic improves but performance may still deteriorate due to obstacles as well as bright lighting or atypical camera positions. New advancements in computer vision will seek to reverse partially hidden motions which improve system reliability and train algorithms to fill in missing keypoints while implementing depth sensors for full-motion gesture analysis.

7. User-Customizable Sign Dictionaries

The system becomes more adaptable with inclusive capabilities when users receive opportunities to make gesture modifications that include adding local expressions and context-aware symbols and personalized forms. The system has the potential to learn new gestures through few-shot learning techniques after users develop an interface for gesture training that enables motion recording with categorization features.

8. Feedback System and Learning Assistance

In order to turn the system into an effective learning aid for sign language, it could be enhanced to include a feedback system that teaches users the correct motions. Using this interactive learning mode, users can improve their signing skills gradually by comparing their gesture to a standard reference and getting suggestions for improvement.

9. Cross-Domain Integration and Assistive Ecosystems

The system can be modified to be applied to various gesture-based control systems, including robots, virtual reality, and smart homes, although it was designed for sign language recognition. The gesture detection system can be incorporated into larger human-computer interaction platforms in future work to provide natural, intuitive control interfaces for individuals with and without disabilities.

CHAPTER – 10

INDIVIDUAL TEAM MEMBER'S REPORT

10.1 Individual Objective

Team Member 1: Mohd Azim I

The main purpose of this research involved deploying MediaPipe Holistic to develop a gesture detection system based on landmark recognition. The establishment of a dependable data collection system became essential to handle facial along with hand and body landmark information from live streaming video footage. The extraction process needed high precision and temporal consistency because it supplied data to the sequential model..

The team worked on two additional responsibilities which involved uniting OpenCV with MediaPipe as well as drafting scripts for dataset creation and live keypoint presentation. The team integrated several tests and debugging operations to the complete data pipeline to guarantee reliable storage and preparation of keypoint coordinates for training purposes. Deep learning-based gesture modelling achieves both accuracy and efficiency through the finished work.

Team Member 2: Vidurvel Prouchotte

The main research goal involved creating an LSTM deep learning model to classify retrieved hand gesture sequences. The evaluation of sign language recognition setups needed testing between multiple model structures alongside changes to hyperparameters. The achievement of consistent results for various gesture categories required supervision throughout training alongside validation and assessment processes along with precision, accuracy and loss evaluation and F1-score assessment.

The model executed live video-based real-time predictions to check the system's performance at identifying signs Hello, Thank You and I Love You with minimum delays. Model tracking and technical innovation distribution required the implementation of TensorBoard technology along with other platforms to support documentation processes, result interpreting and performance visualization tasks.

Team Member 3: Monish Aston C

The main goals of this initiative centered on enabling immediate communication and userfriendly front-end functions between end users and the model. Developing a small application which combines the LSTM model and a live video stream with on-screen text overlays delivered real-time sign feedback to users.

A core responsibility involved optimizing the inference pipeline to minimize execution delays during real-time operation. Testing of the pipeline's functionality occurred through device and environmental evaluations from simulation and system integration activities. System inclusivity improved as the user interface became simpler and more accessible because of the interface enhancements that enabled both

new and experienced users to operate the system. The development work advanced assistive technology goals by closing in on the project targets.

10.2 Role of the Team Members

Team Member 1: Mohd Azim I

The essential part of this project involved designing and implementing the gesture recognition pipeline which used MediaPipe Holistic for landmark detection. A dependable system for real-time keypoint extraction needed to be designed to acquire video streams which included hand body and face landmarks. The extraction process needed to preserve both high fidelity and temporal consistency because this precision mattered for feeding accurate data to the sequential model.

The project required me to write scripts for displaying keypoints in real-time while building tools for dataset creation which merged OpenCV with MediaPipe functions. The team executed complete tests and debugs of the data pipeline to validate that the keypoint coordinates would be processed accurately and maintained consistent storage across training sessions. The developed initiatives formed an effective basis which allowed deep learning-based gesture modeling to operate precisely and efficiently.

Team Member 2: Vidurvel Prouchotte

Improving and developing the LSTM-based deep learning model which sorted collected gesture sequences constituted the leading responsibility. A complete evaluation involving different model architectures along with parameter optimization was required to establish the most effective setup for sign language recognition. The consistent model performance needed the responsibilities to handle training and validation and evaluation stages along with measuring accuracy and F1-score precision and loss metrics.

The launched model operated from a real-time video stream for prediction purposes to support Class 1.5 which contains the signs "Hello, Thank You and I Love You" while minimizing prediction delays. A team member contributed to both documentation and result interpretation while utilizing TensorBoard to visualize performance and monitor the updates of the models.

Team Member 3: Monish Aston C

Interface design and model-end-user communication speed were the essential functions I needed to handle. The development aimed to create a minimalist application which merges the LSTM model along with camera streaming and visual texts overlaying identified indicators in real-time.

Part of my responsibilities involved helping with simulation tests to verify device compatibility in different environments while developing the inference system to reduce delays during real-time operation. Better assistive technology alignment became possible after improving both user interface

accessibility and usability characteristics. Through these changes the system gained more practical value and inclusion of users who lacked technical skills in its adoption.

10.3 Contribution of the Team Members

Team Member 1: Mohd Azim I

Contribution:

The work built the gesture detection pipeline through MediaPipe Holistic and used the resulting system to prepare data for subsequent processing. The successful implementation needed an efficient real-time system that collected data from body, face and hand landmarks in real-time broadcasts. The deep learning model received arranged data that had initially been extracted by scripts from every video frame. The process of maintaining accurate gesture sequence modeling required extensive time for precise landmark data extraction.

The system became more efficient for real-time landmark representation because of the innovative integration of OpenCV into its framework. Through this connection the data collection quality improved since it monitored recordings to achieve clear movements containing minimal extraneous sounds. The dataset pipeline framework allowed handling numerous samples to develop trustworthy classification models as its foundational component.

The data pipeline achieved validation through testing while developers solved tracking problems and confirmed label connections and achieved sample format consistency. The reliable clean inputs required for training the sequential model needed these crucial efforts to succeed. The entire project successfully enhanced both the accuracy and reliability while achieving real-time performance of the system.

Team Member 2: Vidurvel Prouchotte

Contribution:

The primary work involved the development of a sign language classifier based on LSTM deep learning with training and construction of the model. Modelling sequential data needed its own architecture design which involved examining different layer setups combined with optimizers and dropout rate setups. Multiple adjustments were made to hyperparameters because training duration directly affected

classification accuracy performance. Generalization of unseen gesture data required validation approaches to be integrated during training to stop unwanted overfitting effects.

After completion of training the model adopted real-time webcam feeds from the landmark detection pipeline for actual deployment. The real-time prediction scripts operated to predict the sign gestures and ended up reporting results with minimal latency levels. The main focal point in system development aimed to achieve rapid smooth user interaction through accuracy maintenance and lower inference time thus ensuring an optimized experience. The system had to prove its operational capability after testing basic signals "Hello," "Thank You," and "I Love You" in diverse lighting environments and background settings.

The development operations and performance tracking work included TensorBoard with other tools for major training data monitoring capabilities. All documented findings about model architecture evaluation results along with technical observations served as the basis for producing project reports. The system's effectiveness proved during this phase giving a solid foundation for future system developments.

Team Member 3: Monish Aston C

The Front-end development together with system user interface formed the backbone of my work on the project. A modern user interface functioned in real time to offer live one-to-one user access for gesture recognition management. A visual feedback system required the integration of the trained LSTM model with the front-end layer to show identified sign text captions on the video display. A simplified user interface selection process helps basic computer users including assistive technology dependents execute their tasks efficiently.

The authors developed improvements to accelerate real-time prediction processes within the inference pipeline. Assurance of multiple hardware support and optimized frame processing operations combined with system resource management became necessary requirements. The system testing phase included multiple device assessments to demonstrate proper operations across different circumstances while resolving all detected system bottlenecks and user interface problems. Real-time input became the main focus of development because this fundamental requirement delivered a user experience that was swift and interactive for real-world applications.

The testing and integration process received active contributions for ensuring flawless operation across entire system components from landmark recognition to prediction outcomes. The system interface received continuous upgrades based on user responses along with its performance results. More documentation was written about the user interface module and deployment procedure to enrich knowledge transfer and support system expansion in the future. The system underwent improvements thanks to these initiatives which made it operational and dependable.

REFERENCES

[1] S. Roy, M. Das, and P. Singh, "Vision Transformers for Continuous Sign Language Recognition," in Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.

[2] A. Kumar, B. Verma, and C. Sharma, "Sign Language Recognition Application Using LSTM and GRU RNN," in Proc. of the International Conference on Emerging Technologies, 2023.

[3] I. G. A. Poornima, G. Sakthi Priya, C. A. Yogaraja, R. Venkatesh, and P. Shalini, "Hand and Sign Recognition of Alphabets Using YOLOv5," SN Computer Science, March 2024.

[4] N. Kumar, A. Sharma, and P. Gupta, "Indian Sign Language Recognition Using MediaPipe Holistic," in Proc. of the International Conference on Signal Processing and Communications, 2024.

[5] D. Patel, R. Kumar, and S. Lee, "Real-Time American Sign Language Detection Using YOLO-v9," in Proc. of the IEEE International Conference on Image Processing, 2024.

[6] A. Raj Verma, G. Singh, K. Meghwal, B. Ramji, and P. K. Dadheech, "Enhancing Sign Language Detection through MediaPipe and Convolutional Neural Networks (CNN)," in Proc. of the International Conference on Artificial Intelligence, 2024.

APPENDIX A

SAMPLE CODE

```
!pip install tensorflow==2.18.0 opencv-python mediapipe scikit-learn matplotlib

import cv2

import numpy as np

import os

from matplotlib import pyplot as plt

import time

import mediapipe as mp

import mediapipe as mp

mp_holistic = mp.solutions.holistic # Holistic model

mp_drawing = mp.solutions.drawing_utils # Drawing utilities

def mediapipe_detection(image, model):

 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # COLOR CONVERSION BGR 2 RGB

    image.flags.writeable = False                  # Image is no longer writeable

    results = model.process(image)                 # Make prediction

    image.flags.writeable = True                   # Image is now writeable

    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR) # COLOR COVERSION RGB 2 BGR

    return image, results


mp_drawing = mp.solutions.drawing_utils

mp_holistic = mp.solutions.holistic

mp_face_mesh = mp.solutions.face_mesh

def draw_landmarks(image, results):

mp_drawing.draw_landmarks(image, results.face_landmarks, mp_face_mesh.FACEMESH_TESSELATION) # Draw face connections

mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_holistic.POSE_CONNECTIONS) # Draw pose connections
```

```
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS) # Draw right hand connections


def draw_styled_landmarks(image, results):
    # Draw face connections
    mp_drawing.draw_landmarks(image, results.face_landmarks,
mp_face_mesh.FACEMESH_TESSELATION,
                              mp_drawing.DrawingSpec(color=(80,110,10), thickness=1, circle_radius=1),
                              mp_drawing.DrawingSpec(color=(80,256,121), thickness=1, circle_radius=1)
                              )
    # Draw pose connections
    mp_drawing.draw_landmarks(image, results.pose_landmarks,
mp_holistic.POSE_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(80,22,10), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(80,44,121), thickness=2, circle_radius=2)
                              )
    # Draw left hand connections
    mp_drawing.draw_landmarks(image, results.left_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(121,22,76), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(121,44,250), thickness=2, circle_radius=2)
                              )
    # Draw right hand connections
    mp_drawing.draw_landmarks(image, results.right_hand_landmarks,
mp_holistic.HAND_CONNECTIONS,
                              mp_drawing.DrawingSpec(color=(245,117,66), thickness=2, circle_radius=4),
                              mp_drawing.DrawingSpec(color=(245,66,230), thickness=2, circle_radius=2)
cap = cv2.VideoCapture(0)
```

```
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():

        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)
        print(results)

        draw_styled_landmarks(image, results)

         # Show to screen
        cv2.imshow('OpenCV Feed', image)

        # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()

plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

import numpy as np

if results.face_landmarks:
    face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark]).flatten()
else:
    face = np.zeros(1404)
```

```
def extract_keypoints(results):
    pose = np.array([[res.x, res.y, res.z, res.visibility] for res in
results.pose_landmarks.landmark]).flatten() if results.pose_landmarks else np.zeros(33*4)
    face = np.array([[res.x, res.y, res.z] for res in results.face_landmarks.landmark]).flatten() if
results.face_landmarks else np.zeros(468*3)
    lh = np.array([[res.x, res.y, res.z] for res in results.left_hand_landmarks.landmark]).flatten() if
results.left_hand_landmarks else np.zeros(21*3)
    rh = np.array([[res.x, res.y, res.z] for res in results.right_hand_landmarks.landmark]).flatten()
if results.right_hand_landmarks else np.zeros(21*3)
    return np.concatenate([pose, face, lh, rh])

DATA_PATH = os.path.join('MP_Data')

# Actions that we try to detect
actions = np.array(['hello', 'thanks', 'iloveyou'])

# Thirty videos worth of data
no_sequences = 30

# Videos are going to be 30 frames in length
sequence_length = 30

for action in actions:
    for sequence in range(no_sequences):
        try:
            os.makedirs(os.path.join(DATA_PATH, action, str(sequence)))
        except:
            pass


cap = cv2.VideoCapture(0)
# Set mediapipe model
```

```
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as
holistic:


    # NEW LOOP
    # Loop through actions
    for action in actions:
        # Loop through sequences aka videos
        for sequence in range(no_sequences):
            # Loop through video length aka sequence length
            for frame_num in range(sequence_length):

                # Read feed
                ret, frame = cap.read()

                # Make detections
                image, results = mediapipe_detection(frame, holistic)
                 #print(results)

                # Draw landmarks
                draw_styled_landmarks(image, results)

                # NEW Apply wait logic
                if frame_num == 0:
                    cv2.putText(image, 'STARTING COLLECTION', (120,200),
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4, cv2.LINE_AA)
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action)
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)
                    cv2.waitKey(2000)
                else:
```

```
                cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action,
sequence)
                 cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
            # Show to screen
            cv2.imshow('OpenCV Feed', image)

        # NEW Export keypoints
        keypoints = extract_keypoints(results)
        npy_path = os.path.join(DATA_PATH, action, str(sequence), str(frame_num))
        np.save(npy_path, keypoints)

        # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break

cap.release()
 cv2.destroyAllWindows()

log_dir = os.path.join('Logs')
tb_callback = TensorBoard(log_dir=log_dir)

model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback])

model.summary()

colors = [(245,117,16), (117,245,16), (16,117,245)]
def prob_viz(res, actions, input_frame, colors):
output_frame = input_frame.copy()
for num, prob in enumerate(res):
cv2.rectangle(output_frame, (0,60+num*40), (int(prob*100), 90+num*40), colors[num], -1)
```

```
        cv2.putText(output_frame, actions[num], (0, 85+num*40),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,255), 2, cv2.LINE_AA)


    return output_frame
        # 1. New detection variables
        sequence = []
        sentence = []
        predictions = []
        threshold = 0.5
cap = cv2.VideoCapture(0)
# Set mediapipe model
        with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5)
        as holistic:
    while cap.isOpened():

        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)
        print(results)

        # Draw landmarks
        draw_styled_landmarks(image, results)

        # 2. Prediction logic
        keypoints = extract_keypoints(results)
        sequence.append(keypoints)
        sequence = sequence[-30:]

        if len(sequence) == 30:
```

```
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            print(actions[np.argmax(res)])
            predictions.append(np.argmax(res))
        #3. Viz logic
            if np.unique(predictions[-10:])[0]==np.argmax(res):
                if res[np.argmax(res)] > threshold:

                    if len(sentence) > 0:
                        if actions[np.argmax(res)] != sentence[-1]:
                            sentence.append(actions[np.argmax(res)])
                    else:
                        sentence.append(actions[np.argmax(res)])

            if len(sentence) > 5:
                sentence = sentence[-5:]

            # Viz probabilities
            image = prob_viz(res, actions, image, colors)

        cv2.rectangle(image, (0,0), (640, 40), (245, 117, 16), -1)
        cv2.putText(image, ' '.join(sentence), (3,30),
                       cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)
        # Show to screen
        cv2.imshow('OpenCV Feed', image)
          # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

APPENDIX B

SAMPLE SCREEN

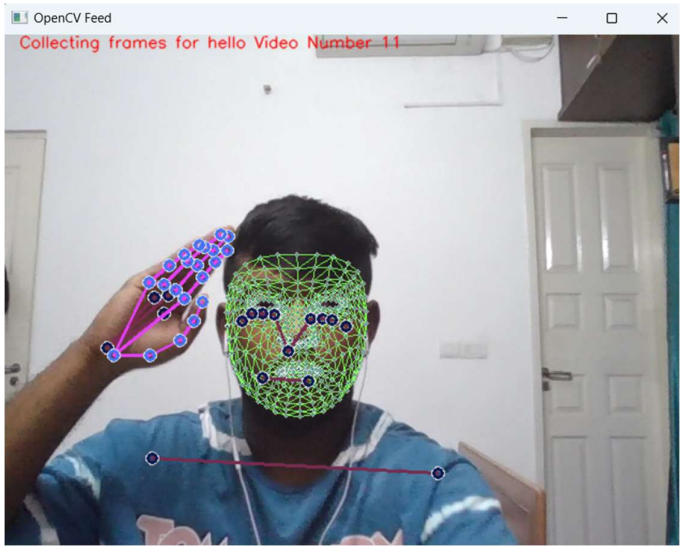

Fig – Sample Screenshot

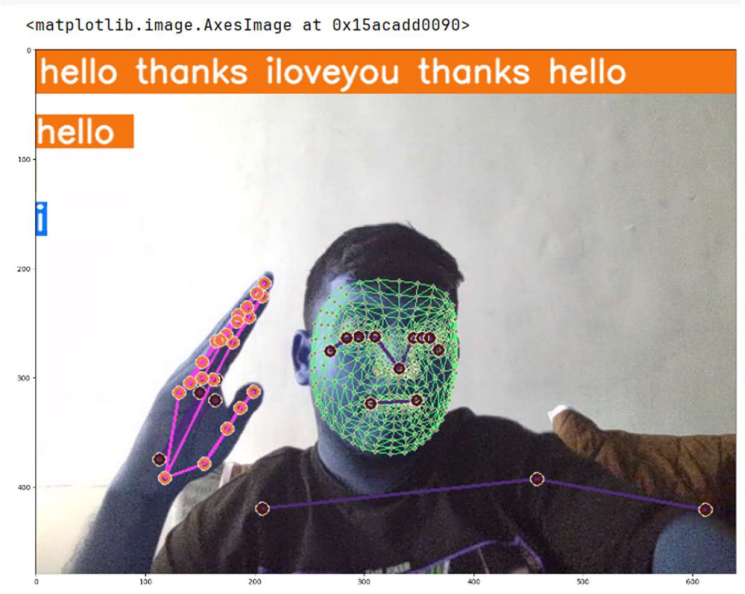

Fig – Sample Screenshot

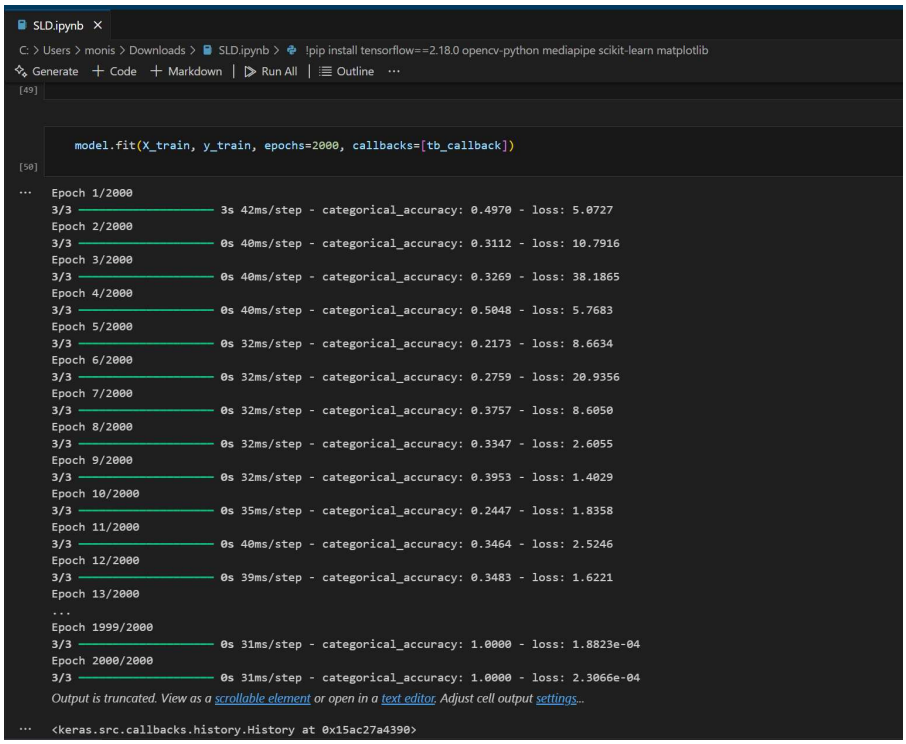

Fig – Sample Screenshot

```
cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:

    # NEW LOOP
    # Loop through actions
    for action in actions:
        # Loop through sequences aka videos
        for sequence in range(no_sequences):
            # Loop through video length aka sequence length
            for frame_num in range(sequence_length):

                # Read feed
                ret, frame = cap.read()

                # Make detections
                image, results = mediapipe_detection(frame, holistic)
                #print(results)

                # Draw landmarks
                draw_styled_landmarks(image, results)

                # NEW Apply wait logic
                if frame_num == 0:
                    cv2.putText(image, 'STARTING COLLECTION', (120,200),
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4, cv2.LINE_AA)
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action, sequence), (15,12),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)
                    cv2.waitKey(2000)
                else:
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(action, sequence), (15,12),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_AA)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)

                # NEW Export keypoints
                keypoints = extract_keypoints(results)
                npy_path = os.path.join(DATA_PATH, action, str(sequence), str(frame_num))
                np.save(npy_path, keypoints)

                # Break gracefully
                if cv2.waitKey(10) & 0xFF == ord('q'):
                    break

    cap.release()
    cv2.destroyAllWindows()
```

Fig – Sample Screenshot

APPENDIX C

PLAGARISM REPORT

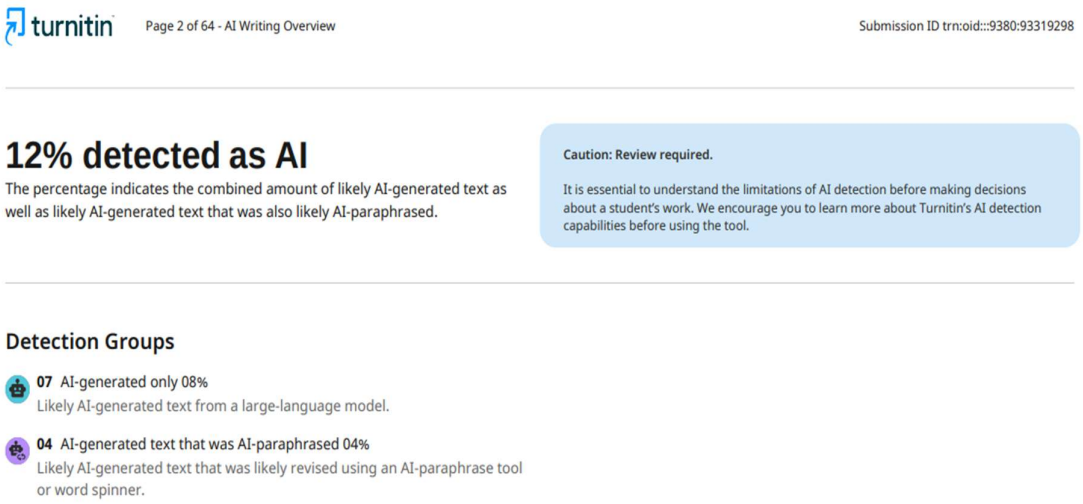

Fig – Plagiarism Report

APPENDIX D

PUBLICATION DETAILS

Gmail MOHD AZIM.I <mohdazimi083@gmail.com>

Acceptance of your abstract for the ICAMA-2025
1 message

S&H Conference SEC <icama2025@sairam.edu.in> 15 April 2025 at 15:25
To: mohdazimi083@gmail.com

Dear Author,

Greetings!

On behalf of the organizing committee of the "International Conference on Advances in Mathematics and its Applications – ICAMA- 2025" to be held on May 08-09, 2025, in our Sri Sai Ram Engineering College (Hybrid mode). We are happy to inform you that your abstract entitled "**Real-Time Sign Language to Word Sentence Translation Using LSTM and MediaPipe Holistic**" has been accepted for presentation. Kindly submit your full paper for considering the publication process. (Kindly note that your manuscript will be considered for the conference proceedings/Journals based on the satisfactory reply to the reviewer's comments).

We welcome you to join us and share your knowledge and views on the theme "ICAMA-2025".

We thank you for your interest in participating in the ICAMA-2025. We look forward to seeing you.

For more details about ICAMA- 2025, visit our website **https://www.icama.in/**

PS:1 *All participants must* ***register and pay the registration fee*** *at the earliest if not paid earlier.*

2. **Conference Kit (Laptop bag, pen, writing pad), refreshments, Lunch, etc.. will be provided for those who are presented the paper offline only**

Regards

Dr.A.Magesh,
Convenor-ICAMA'2025
Sri Sai Ram Engineering College, Chennai, India.
Email: icama2025@sairam.edu.in

Fig – Publication Details

APPENDIX E

TEAM DETAILS

NAME : Mohd Azim I

ROLL NO : 21113083

EMAIL : 21113083@student.hindustanuniv.ac.in

CONTACT NO : 9894348856

NAME : Vidurvel Prouchotte

ROLL NO : 21113084

EMAIL : 21113084@student.hindustanuniv.ac.in

CONTACT NO : 7598712463

NAME : Monish Aston C

ROLL NO : 21113094

EMAIL : 21113094@student.hindustanuniv.ac.in

CONTACT NO : 7530066844