**Real-Time Sign Language Detection Using LSTM**

Masters of Technology

in

VLSI and Embedded Systems

by

Shruti Gujar

Niti Agrawal

Under the Guidance of

Prof. Vinaya Gohokar



Department of Electrical and Electronics Engineering

MIT World Peace University

2024-2025

**DECLARATION**

We hereby declare that this written submission is a reflection of our original ideas and work. Wherever we have incorporated the ideas, words, or contributions of others, we have appropriately credited and referenced the sources by academic standards. Additionally, we affirm that we have strictly adhered to the principles of academic honesty and integrity throughout the preparation and presentation of this submission. We understand the importance of these principles and have taken care to ensure that our work upholds them fully.

**ACKNOWLEDGEMENT**

We wish to express our deep sense of gratitude and thank Prof. Vinaya Gohokar ma’am for her guidance, help and useful suggestions, which helped us in our approach. We are also grateful to our batchmates who provided us with the feedback whenever required.

**Project Team**

Shruti Gujar

Niti Agrawal

**ABSTRACT**

This project focuses on the development of a **real-time sign language detection system** using **Long Short-Term Memory (LSTM)** networks, aiming to bridge communication barriers for individuals with hearing impairments by translating American Sign Language (ASL) gestures into text. The core objective of this work is to achieve high accuracy and low latency, making the system practical for real-time applications.

The model leverages the **MS-ASL dataset**, which includes over 1,000 different signs, ensuring robustness in various real-world environments. The **Media Pipe** framework is employed to preprocess video inputs, extracting key hand and finger landmarks as input features to the LSTM model. Evaluation metrics, including accuracy, precision, recall, F1-score, and latency, were applied to validate the system’s performance. The project achieved significant results, underscoring the potential of LSTMs for real-time sign-language recognition. Moreover, the framework is flexible, allowing the future addition of more sign languages and complex gestures, enhancing inclusivity and scalability. The findings of this project demonstrate the feasibility of LSTM networks as a powerful tool for real-time sign language recognition and translation.

**Contents**

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |
| **Chapter 1** | **Introduction** | 6 |
| **Chapter 2** | **Objectives of the project** | 7 |
|  | 2.1 Scope and Applications | 7 |
| **Chapter 3** | **Dataset Description and Detailed Analysis** | 9 |
|  | 3.1 MS-ASL Dataset | 9 |
|  | 3.2 Data Preprocessing | 10 |
| **Chapter 4** | **Algorithm/Model/Methodology** |  |
|  | 4.1 Algorithm | 11 |
|  | 4.2 Methodology | 13 |
|  | 4.3 Why LSTM? | 15 |
|  | 4.4 Literature Review | 16 |
| **Chapter 5** | **Results** | 17 |
| **Chapter 6** | **Evaluation Criteria** | 18 |
| **Chapter 7** | **Conclusion** | 19 |
|  | **References** | 20 |

**Chapter 1**

**INTRODUCTION**

This project aims to develop a cutting-edge real-time sign language recognition system using advanced machine learning techniques. The primary goal is to address communication barriers for individuals with hearing impairments by translating sign language gestures into readable text. Utilizing **Long Short-Term Memory (LSTM)** neural networks, the system excels in recognizing sequential hand gestures from live video streams, effectively capturing the nuances of sign language.

LSTMs are particularly suited for this task due to their ability to learn from time-dependent data, making them ideal for interpreting complex gestures that occur over time. The system processes video input to recognize and convert gestures in real-time, ensuring minimal latency for practical applications.

Furthermore, the model is designed to be flexible and scalable, allowing for the incorporation of additional sign languages and more complex gestures in the future. Ultimately, this project seeks to create an efficient and accurate tool that enhances communication accessibility for individuals who rely on sign language, promoting inclusivity in various social and professional settings.

**Chapter 2**

**OBJECTIVES OF THE PROJECT**

**The main objectives of this project are:**

* **Design and Implementation:** Create a highly efficient real-time recognition system tailored specifically for American Sign Language (ASL) that effectively translates gestures into text.
* **High Accuracy:** Strive for exceptional accuracy in interpreting sign language gestures, ensuring that the output text is clear and easily understood.
* **Minimal Latency:** Develop the system to operate with minimal latency, facilitating practical, real-time applications that enable seamless communication.
* **Flexible Framework:** Establish a versatile framework that allows for the easy integration of new signs and gestures, supporting further customization and scalability to meet diverse user needs.
* **Adaptability:** Design the system to adapt to the evolving nature of sign languages, ensuring its relevance and effectiveness in various contexts and for different user groups.
* **Promote Inclusivity:** Focus on enhancing communication for individuals with hearing impairments, thereby fostering inclusivity and accessibility in a wide range of environments and situations.

**2.1 Scope and Applications**

* Education: The real-time sign language recognition system can be integrated into educational settings, enabling teachers and students to communicate more effectively with hearing-impaired individuals. This technology can facilitate learning and participation in mainstream classrooms.
* Healthcare: In medical environments, the system can assist healthcare professionals in communicating with patients who use sign language. This can enhance patient care, ensure understanding of medical instructions, and improve overall health outcomes.
* Workplace Integration: Businesses can implement this technology to create more inclusive work environments, allowing employees who are deaf or hard of hearing to communicate seamlessly with their colleagues. This fosters teamwork and collaboration.
* Public Services: Government and public service organizations can utilize the system to provide better access to services for the hearing-impaired community, ensuring that vital information is communicated effectively during emergencies or public announcements.
* Entertainment and Media: The technology can be applied in the entertainment industry, such as in films and television shows, to provide sign language translations or interpretations, making content more accessible to deaf and hard-of-hearing audiences.
* Mobile Applications: The system can be adapted for mobile platforms, allowing users to communicate in sign language through their smartphones. This would facilitate everyday interactions in various social situations.
* Remote Communication: The recognition system can be integrated into video conferencing tools, enabling effective communication in virtual meetings for users who prefer sign language, thus supporting remote work and collaboration.
* Research and Development: This project opens avenues for further research in gesture recognition, machine learning, and artificial intelligence, contributing to advancements in the fields of human-computer interaction and accessibility technologies.

Overall, the real-time sign language recognition system has a broad scope of applications that can significantly enhance communication accessibility and inclusivity across various sectors and communities.

**Chapter 3**

**DATASET DESCRIPTION AND DETAILED ANALYSIS**

**3.1. MS-ASL Dataset**

The **MS-ASL dataset** serves as a foundational resource for developing the real-time sign language recognition system. It contains over **1,000 distinct American Sign Language (ASL)** signs, each represented by multiple video samples. This extensive collection enables large-scale training, allowing the model to learn from a diverse set of gestures performed by various participants under different conditions. The dataset is particularly valuable due to its inclusion of multiple recording environments, which vary in terms of background settings and lighting conditions. This diversity enhances the robustness of the model, enabling it to generalize better to real-world scenarios where variability is common.

In comparison to more recent datasets, such as the **RWTH-PHOENIX-Weather 2014T** and the **ASL Lexicon Video Dataset**, which also contain extensive video samples for sign language recognition, the MS-ASL dataset stands out due to its comprehensive coverage of different ASL signs. The RWTH-PHOENIX dataset focuses on weather-related signs and contains a mix of static and dynamic gestures, while the ASL Lexicon provides video examples of isolated signs but may lack the extensive contextual variety present in the MS-ASL dataset. The breadth of the MS-ASL dataset allows for a more nuanced training of models, ensuring that they are well-equipped to handle the intricacies of sign language interpretation.

**3.2. Data Preprocessing**

Data preprocessing is a crucial step in preparing the dataset for training the LSTM model. The preprocessing phase employs **MediaPipe**, a robust library designed for real-time computer vision tasks, to analyze the video frames and extract critical hand and finger landmarks. These landmarks represent key points of the hands during gesture performance and are fundamental for constructing feature vectors that capture the essence of each sign gesture.

Once the landmarks are extracted, they are transformed into feature vectors that serve as input to the LSTM model. This approach allows the model to utilize the spatial and temporal information embedded in the gestures, which is essential for accurate sign language recognition. Given that gestures can vary significantly in duration, the preprocessing pipeline incorporates techniques such as **zero-padding** and **frame interpolation** to standardize the input sequences. Zero-padding adds extra frames to shorter sequences to ensure uniformity, while frame interpolation can be used to increase the temporal resolution of longer sequences, facilitating consistent training across different gesture durations.

When compared to modern datasets like the **Sign Language Recognition Dataset (SLR)** and the **Aerial Sign Language Dataset**, which employ advanced preprocessing techniques such as optical flow and depth estimation, the MS-ASL dataset's approach to preprocessing focuses on landmark detection, emphasizing the relevance of hand gestures in a two-dimensional space. While both methodologies aim to improve the accuracy and efficiency of sign language recognition systems, the choice of preprocessing techniques can significantly impact the model's performance based on the characteristics of the dataset being used.

Overall, the detailed analysis of the MS-ASL dataset and its preprocessing techniques highlights its suitability for developing a robust sign language recognition system while also illustrating how it compares favorably with contemporary datasets in the field. The effective use of landmark extraction and sequence standardization sets a solid foundation for training models that can accurately interpret and translate sign language gestures in real-time.

**Chapter 4**

**ALGORITHM/MODEL/METHODOLOGY**

**4.1. Algorithm**

The core of this project is the Long Short-Term Memory (LSTM) neural network. LSTMs are effective in handling sequential data and are well-suited for tasks requiring temporal understanding, such as gesture recognition.

1. **Initialize System:**
   * Load the trained **LSTM model**.
   * Set up the **camera** or video stream for real-time input.
   * Import the **MediaPipe** library for extracting hand landmarks.
2. **Start Video Capture:**
   * Continuously capture **live video frames** from the camera.
3. **For each frame in the video stream:**
   * **Preprocess Input:**
     + **Convert the frame to RGB format** (required for most computer vision libraries).
     + **Detect hand landmarks** using MediaPipe’s hand detection module.
     + If **no hands** are detected, return to step 3 and continue with the next frame.
4. **Landmark Extraction and Feature Vector Generation:**
   * For detected hand landmarks:
     + Extract **key hand and finger points** (landmarks such as wrist, thumb, index finger, etc.).
     + Convert the hand landmarks into a **feature vector** that represents the spatial positions of the hand.
5. **Create Sequence for Temporal Recognition:**
   * Append the feature vector of the current frame to a **sequence of past feature vectors**.
   * **Ensure the sequence has a fixed length** (e.g., 30 frames). If the sequence is too short, pad with zeros; if it's too long, remove the oldest vectors.
6. **Feed the Sequence into the LSTM Model:**
   * Once a sufficient sequence of frames is collected, input the sequence into the **trained LSTM model**.
7. **Gesture Prediction:**
   * The LSTM model processes the sequence and **predicts the gesture** (e.g., an ASL sign).
   * If the gesture is classified as a known sign, proceed to the next step.
8. **Display Prediction:**
   * **Convert the predicted gesture** into its corresponding text.
   * Display the detected gesture’s text on the screen.
9. **Handle Real-Time Constraints:**
   * Ensure that the system can capture, process, and predict gestures with **minimal latency** (target: less than 500 milliseconds).
10. **Loop Continuously:**
    * Repeat steps 3-9 for each frame in the video stream, allowing for continuous real-time gesture recognition.
11. **End the System:**
    * Stop the video capture and close the system when the user exits.

**4.2. Methodology**

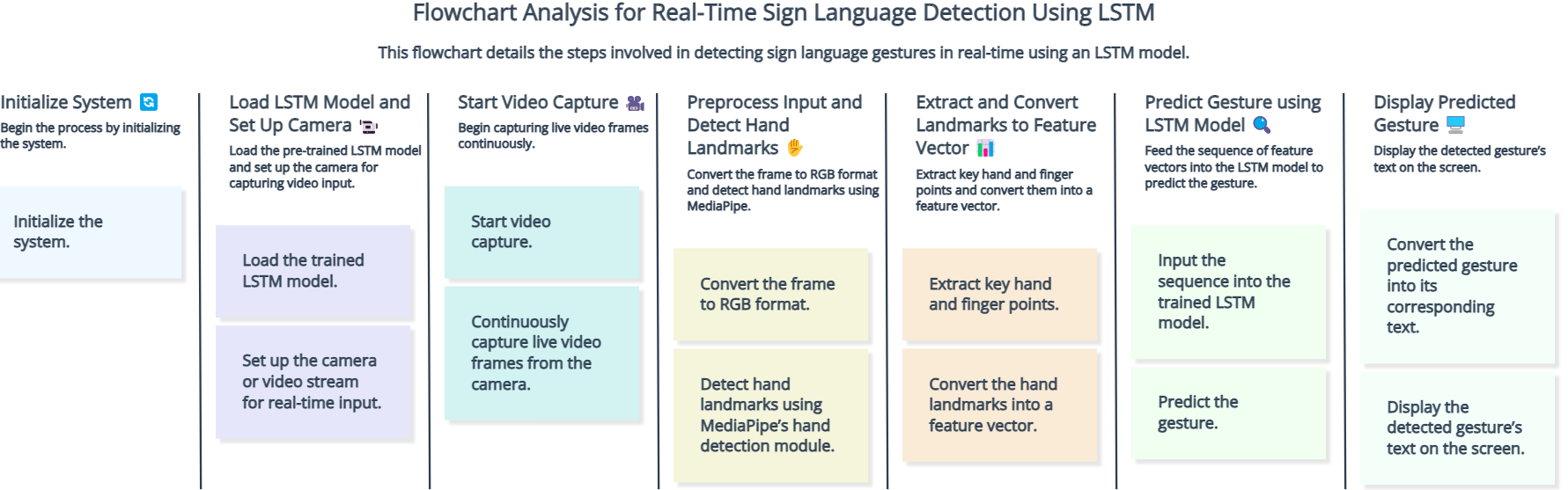


Fig. 4.1 Flowchart of the project working

Dataset Selection:

* The MS-ASL (Microsoft American Sign Language) dataset is used for training. This dataset includes over 1,000 ASL gestures, making it robust for real-world conditions.
* Multiple recording environments are included to ensure the model generalizes well in diverse scenarios.

Data Preprocessing:

* Extract key hand and finger landmarks using the MediaPipe framework. This step reduces the complexity by focusing on essential points of interest.
* Use techniques like zero-padding and frame interpolation to standardize input sequences. These methods ensure that input gestures of varying lengths are properly handled by the LSTM model.

Model Design:

* Long Short-Term Memory (LSTM) is used as the core model due to its ability to handle sequential data and retain contextual information over time.
* Implement input, forget, and output gates to control information flow, ensuring that the model can differentiate between meaningful and irrelevant gestures.

Training and Evaluation:

* Train the model using the MS-ASL dataset, optimizing for high accuracy and low latency.
* Evaluate using performance metrics such as accuracy, precision, recall, F1-score, and latency to ensure that the model meets real-time constraints.

Real-Time Implementation:

* Implement the model in a real-time system that captures live video input, preprocesses the data, and performs gesture recognition in under 500ms.
* Test the system in real-world conditions to validate its performance and make necessary adjustments for robustness.

Performance Tuning:

* Regularly fine-tune the LSTM model with more data or different preprocessing techniques to improve accuracy and reduce latency.
* Consider adding more sign languages or gesture sets for broader applications.

**4.3. Why LSTM?**

Sign language is inherently sequential, consisting of ordered hand movements and gestures that convey meaning. Each gesture's interpretation depends on the context provided by preceding movements, necessitating a model that can effectively capture these temporal relationships. LSTMs are uniquely equipped for this task because they retain information across time steps, allowing them to model the progression of hand movements with remarkable accuracy.

One of the most significant advantages of LSTMs is their ability to overcome the **vanishing gradient problem**, which often hinders traditional recurrent neural networks (RNNs) when dealing with long sequences. By maintaining a memory cell that can preserve information over extended periods, LSTMs can effectively learn complex patterns in sign language, enabling them to recognize not just individual gestures but also the nuances of gesture transitions and context.

The architecture of LSTMs includes mechanisms such as **input, forget, and output gates**, which regulate the flow of information and facilitate the retention of important features while discarding irrelevant data. This functionality is crucial in sign language recognition, where the model must differentiate between subtle variations in gestures that may significantly alter meaning.

**4.4. Literature Review**

Recent studies highlight the effectiveness of LSTMs in various applications, particularly in sign language recognition. For instance, research by **Alkhamees et al.** illustrates the benefits of combining LSTMs with complementary techniques, such as the **You Only Look Once (YOLO)** algorithm, which is used for detecting static signs within a video frame. This hybrid approach significantly enhances recognition accuracy by enabling the system to first identify the presence of a sign before processing its dynamic components with the LSTM. Such integrations underscore the flexibility and robustness of LSTMs, which can adapt to recognize both static and dynamic gestures effectively.

Furthermore, other studies have explored the application of LSTMs in combination with **Convolutional Neural Networks (CNNs)** to extract spatial features from hand gestures before feeding the data into the LSTM for temporal analysis. This two-tiered approach not only improves the model's ability to learn from complex visual input but also enhances its overall performance in real-time scenarios.

Additionally, LSTM-based models have shown promising results in different datasets beyond ASL, including **Chinese Sign Language** and **Indian Sign Language**, indicating their versatility across diverse signing systems. Such findings reinforce the notion that LSTMs are not just limited to recognizing American Sign Language but can be adapted to various sign languages worldwide.

Overall, the combination of LSTMs with advanced detection and recognition techniques positions this project to achieve its aims effectively. By harnessing the strengths of LSTM networks, the model is poised to deliver a highly accurate and responsive sign language recognition system, thereby facilitating improved communication for individuals with hearing impairments. The methodologies adopted in this project not only enhance the model's performance but also set a solid foundation for future research and development in sign language recognition technologies.

**Chapter 5**

**RESULTS**

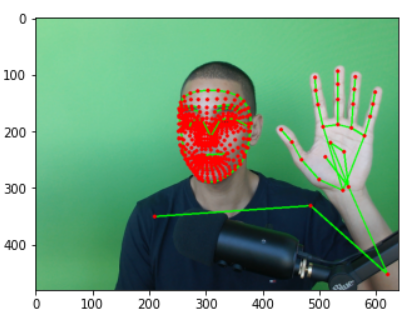
****

Fig. 5.1 Key points extracted using MP

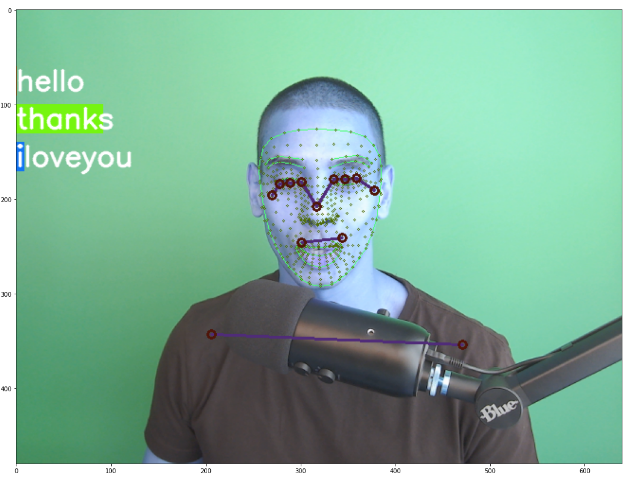
****

Fig. 5.2 Test in real time

**Chapter 6**

**EVALUATION CRITERIA**

The evaluation of the sign language recognition model employs several key metrics to ensure it meets the stringent standards required for real-time applications:

* **Accuracy**: This metric measures the percentage of correctly recognized gestures out of the total presented to the model. A high accuracy rate is crucial for ensuring reliable communication for users depending on the system.
* **Precision, Recall, and F1 Score**:
  + **Precision** assesses the proportion of correctly identified signs among all predicted signs, indicating the reliability of the model's predictions.
  + **Recall** evaluates the model's ability to detect actual signs among all instances, highlighting its effectiveness in capturing relevant gestures.
  + The **F1 Score** combines precision and recall into a single metric, providing a balanced measure that accounts for both false positives and false negatives, making it particularly valuable in cases of class imbalance.
* **Latency**: Given the emphasis on real-time processing, latency is measured as the time delay between the input gesture and the generated output text. The project aims to maintain latency below **500 milliseconds** to ensure smooth and uninterrupted interactions. Minimizing latency is essential for creating a responsive user experience, as delays can disrupt communication flow.

By utilizing these evaluation criteria, the model is rigorously assessed to confirm its capability for effective real-time sign language recognition. This comprehensive evaluation not only demonstrates the model's performance but also identifies areas for potential improvement, ultimately enhancing communication accessibility for individuals with hearing impairments.

**Chapter 7**

**CONCLUSION**

The project successfully demonstrates the feasibility of using Long Short-Term Memory (LSTM) networks for real-time sign-language recognition. The approach developed not only achieves high accuracy but also maintains low latency, making it an effective and practical solution for translating sign language into text in real-world applications. The robust performance of the LSTM model highlights its potential for delivering precise gesture interpretation in dynamic environments, thereby enhancing communication accessibility for individuals who rely on sign language.

Moreover, the methodology employed offers a flexible and scalable framework that can be extended beyond the current scope. This framework can accommodate additional sign languages, facilitating multilingual sign-language recognition, and can be adapted to recognize more complex gestures and nuanced movements. The adaptability of the model ensures that, with further training and refinement, it can continue to evolve alongside advancements in data collection techniques and computing power.

The results of this project underscore the potential of LSTM networks as a foundational technology for sign-language recognition systems, providing a strong basis for further research and development in this field. Through continued improvements in model accuracy, latency, and gesture complexity, this solution could be a transformative tool in bridging communication gaps and promoting inclusivity.

Top of Form

Bottom of Form

**REFERENCES**

[1] Adhikary, A. (2023). Realtime Sign Language Detection Using LSTM Model. GitHub Repository. Available at: <https://github.com/AvishakeAdhikary/Realtime-Sign-Language-Detection-Using-LSTM-Model>

[2] Alkhamees, B. F., et al. (2023). Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs. Mathematics, 11(17), 3729. Available at: <https://www.mdpi.com>

[3] Papers With Code. Sign Language Recognition. Available at: <https://paperswithcode.com/task/sign-language-recognition>

[4] Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2015). Sign Language Recognition Using Convolutional Neural Networks. *European Conference on Computer Vision (ECCV)*. Available at: <https://www.eccv.org>.

[5] Camgoz, N. C., Koller, O., Hadfield, S., & Bowden, R. (2018). Neural Sign Language Translation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Available at: <https://openaccess.thecvf.com.>

[6] Huang, J., Zhou, W., Zhang, Q., Li, H., & Li, W. (2015). Sign Language Recognition Using 3D Convolutional Neural Networks. *IEEE International Conference on Multimedia and Expo (ICME)*. Available at: <https://ieeexplore.ieee.org>.

[7] <https://youtu.be/doDUihpj6ro?si=4Ad76sMsJEAsYAIN>