Human Sign Language Recognition System

Gitesh pal
Dept. of Information Technology
Inderprastha Engineering College
Ghaziabad, India
giteshpal25.04.03@gmail.com

Anshul lakhera
Dept. of Information Technology
Inderprastha Engineering College
Ghaziabad, India
lakheraanshul41@gmail.com

Deepak kumar
Dept. of Information Technology
Inderprastha Engineering College
Ghaziabad, India
deepaksishodia@gmail.com

Tanya Sharma
Assistant Professor , Dept. of Information Technology
Inderprastha Engineering College
Ghaziabad, India
tanya.sharma@ipec.org.in

***Abstract*— Effective communication poses a persistent challenge for the deaf and hard-of-hearing population, often requiring human interpreters to bridge the gap. This research introduces an innovative Human Sign Language Recognition System (HSLRS) designed to overcome these barriers through advanced computational methods. By employing deep learning frameworks, the system integrates convolutional neural networks (CNNs) for spatial feature extraction and recurrent neural networks (RNNs) for temporal sequence analysis, adeptly interpreting the nuanced gestures of sign language. Experimental results reveal high accuracy in distinguishing complex sign patterns and robust real-time recognition capabilities, validated across diverse datasets. This work advances the domains of computer vision and human-computer interaction, offering significant implications for enhancing accessibility and promoting inclusivity in educational, professional, and social contexts.**

keywords—Sign language, Deep learning, Real time recognition,

Computer Vision , Accessibility

I. INTRODUCTION

Sign language serves as a vital communication tool for individuals who are deaf, hard of hearing, or mute, allowing them to convey information and interact meaningfully with others. Despite its importance, a significant barrier exists for widespread understanding, as most people lack the specialized knowledge to interpret sign language. Traditionally, human interpreters have been employed to facilitate communication, but this approach can be costly and logistically challenging, limiting the accessibility of sign language in everyday settings.[18]

Advances in technology have given rise to various approaches for Sign Language Recognition (SLR), including the use of wearable IoT sensors, data gloves, and vision-based systems. While data gloves and wearable sensors can provide reliable recognition accuracy, they require users to wear additional hardware that may interfere with natural movement, making them impractical for everyday communication. Vision based SLR, on the other hand, utilizes computer vision and deep learning to interpret gestures without additional devices, offering a non-invasive and convenient solution.

Recent developments in artificial intelligence and computer vision have enabled the use of Convolutional Neural Networks (CNNs) for feature extraction in SLR. CNNs excel in image-based recognition tasks, yet their computational demands pose challenges for real-time applications. Addressing these limitations, this study proposes a lightweight, computer vision-based SLR model designed to support real-time communication. Leveraging the American Sign Language (ASL) alphabet and commonly used phrases,

Fig. 1. Sign Language gestures signs

this model provides an accessible solution for communication across different user groups, including deaf, mute, and visually impaired individuals.

In addition to CNNs, Long Short-Term Memory (LSTM) networks are integrated into our framework to improve the temporal accuracy of gesture recognition. While CNNs effectively extract spatial features from individual frames, LSTMs specialize in handling sequential data, allowing them to capture the dynamic nature of sign language gestures over time. By leveraging an LSTM layer after the CNN, our model can analyse the flow of gestures in a video sequence, distinguishing similar signs based on subtle temporal differences. This combination of CNN and LSTM provides a robust solution that maintains high accuracy even in complex, multi-frame gestures.

Our system, built on Mediapipe for feature extraction and a random forest classifier for gesture recognition, is designed for efficiency and ease of deployment. By integrating speech-to-text, text-to-speech, and autocompletion features, the framework enhances usability, enabling seamless communication without the need for interpreters.[6] The following sections will discuss related work, outline the methodology, and evaluate the model's

performance, highlighting its potential to improve accessibility in real-time communication scenarios.

II. LITERATURE REVIEW

Over the past decade, significant advances have been made in the field of Human Sign Language Recognition (HSLR), particularly with the evolution of wearable sensors, computer vision, and deep learning techniques. Between 2014 and 2018, wearable sensor-based systems were predominant in the research landscape. These systems utilized data gloves and electromyography (EMG) sensors to capture hand gestures. For instance, a wearable data glove with optical sensors was proposed, which demonstrated promising accuracy in recognizing hand gestures for sign language translation [1]. Similarly, another study focused on using EMG sensors to capture muscle activity in the hand for prosthetics and sign language gesture recognition, providing accurate gesture recognition [2]. These approaches, while accurate, required cumbersome hardware setups, limiting their widespread application.

With the advent of deep learning techniques, computer vision-based systems gained prominence for sign language recognition. One study applied deep learning methods to recognize mouth shapes for sign language translation, marking a significant leap in the integration of AI into HSLR systems [3]. Another study further advanced the field by introducing a neural network-based approach to translate sign language into text, integrating multiple neural networks for improved performance [4]. These early efforts laid the foundation for incorporating deep learning into HSLR but highlighted the need for better handling of temporal and continuous sign data.

The shift towards leveraging large-scale datasets and pre-trained models emerged as a solution to dataset limitations. One study applied transfer learning using deep convolutional networks to improve sign language recognition accuracy, particularly for languages with limited annotated datasets [5]. The use of transfer learning enabled models to generalize across various datasets, improving performance and reducing training time.

A key development in vision-based systems was the introduction of Google Mediapipe in 2020. Mediapipe's real-time hand tracking system allowed for efficient processing of hand landmarks, making it suitable for mobile applications [6]. Mediapipe's efficiency was further utilized in research, where simpler classifiers, such as Random Forests, were integrated to optimize real-time applications on mobile devices.

Further research into the spatial-temporal features of sign language recognition led to more complex architectures. One study explored the use of 3D convolutional networks (3D-CNNs) with attention mechanisms to improve large-vocabulary sign language recognition, capturing both spatial and temporal aspects of gestures [7]. This approach demonstrated enhanced performance for recognizing a broader range of signs.

In 2022, a study conducted a comprehensive review of Transformer models in sign language recognition, identifying their advantages in handling long-term dependencies and

capturing sequential relationships in sign language [8]. Transformers, with their self-attention mechanism, provided a novel and efficient approach to HSLR, offering potential improvements in accuracy and scalability.

Additional multimodal approaches have been explored to improve the robustness and accuracy of recognition systems. One study proposed an RGB-D-based system that combined RGB and depth data, using convolutional neural networks (CNNs) to enhance recognition in varying lighting conditions and hand postures [9]. By incorporating depth information, the system improved the understanding of hand gestures, especially in complex settings.

Lastly, an audio-visual fusion model was introduced that integrated both visual and auditory cues for enhanced recognition of sign language, particularly for sign languages that involve vocal expressions or sound-based cues [10]. This approach leveraged the combination of gesture and audio data, resulting in better comprehension and communication in certain sign language contexts.

Summary of Literature Survey :

Research in HSLR has evolved significantly over the past decade, with early systems primarily relying on wearable sensors and data gloves. Studies such as those by Li et al. (2015) and Zhou et al. (2016) used data gloves and EMG sensors to detect hand and arm movements, achieving decent accuracy but facing limitations in user comfort and practicality due to intrusive hardware.

The development of computer vision and deep learning introduced non-invasive, vision-based HSLR approaches. Koller et al. (2017) and Camgoz et al. (2018) applied Convolutional Neural Networks (CNNs) and hybrid CNN-RNN architectures, allowing for recognition of both static and dynamic gestures. Transfer learning, as explored by Zhao et al. (2020), improved recognition accuracy by adapting pre-trained models, though it was still limited by dataset availability.

To optimize for real-time applications, lightweight systems emerged, like Google Mediapipe (2020), which used efficient CNN models paired with simpler classifiers (e.g., Random Forests) to balance speed and accuracy. Further advancements with sequential modeling, such as Huang et al. (2019) using CNN-LSTM models and Min et al. (2022) exploring Transformers, significantly enhanced the capture of gesture sequences, though at high computational costs.

Finally, multimodal approaches, including Lee et al. (2020) with RGB-D sensors and Kim et al. (2023) with audio-visual fusion, have shown promise in improving recognition robustness. However, these require specialized hardware and are limited in certain settings. Overall, this body of research highlights a trend toward more accessible, accurate, and non-invasive solutions, although challenges in real-time processing, cost, and scalability remain. Future efforts focus on lightweight, adaptive models that can support multiple sign languages and integrate seamlessly into real-world applications.

TABLE 1. COMPARATIVE ANALYSIS OF RELATED WORK

III.

YEAR	APPROACHES USED	MODEL ARCHITECTURE	ACCURACY	SHORTCOMINGS
A. 2015	Data Gloves with Optical Sensors	Sensor-based Glove System	~90%	Intrusive hardware, requires precise sensor placement, limits natural hand movement, lacks scalability
B. 2016	EMG Sensors on Arm	EMG Sensor System	~85%	Invasive setup, high dependency on correct sensor positioning, challenging for real-world use
C. 2017	Vision-Based System	CNN	88% (static signs)	Limited to static gestures, computationally intensive, lacks temporal understanding for dynamic signs
D. 2018	Vision-Based System	CNN + RNN (LSTM)	92%	High computational cost, challenging to deploy on low-power devices, latency in real-time applications
E. 2019	Vision-Based System	CNN-LSTM Hybrid	93% (continuous)	High computational demand, challenges in real-time applications, requires large datasets for training
F. 2020	Vision-Based with Transfer Learning	Pretrained CNN	~90%	Limited by pre-existing datasets, requires large-scale data for fine-tuning, not ideal for continuous signing
G. 2020	Vision-Based Hand Tracking	Lightweight CNN + RF/SVM	~85-90%	Lower accuracy than deeper networks, struggles with complex gestures, can be affected by lighting conditions
H. 2020	RGB-D System (Depth and RGB Cameras)	RGB-D based CNN	~91%	Requires specialized cameras, affected by lighting and depth inconsistencies, hardware limitations
I. 2022	Vision-Based System	Transformer	94%	Computationally expensive, memory-intensive, not optimized for mobile or low-resource devices
J. 2023	Audio-Visual Fusion	Audio Encoder & Visual Encoder	89%	Limited to gestures with audio cues, challenging in silent sign recognition, less practical for standard HSLR

PROPOSED FRAMEWORK FOR HUMAN SIGN LANGUAGE RECOGNITION SYSTEM

Figure 2 shows the development of Human sign language recognition system involves several critical stages to ensure accuracy and efficiency. The process begins with **data**

collection, where a high-quality dataset is curated. This dataset may include publicly available sign language datasets or custom-collected data through video recordings. A well-structured dataset is fundamental to training a robust model.

Following data acquisition, **preprocessing** is conducted to enhance data quality. This stage includes **hand and body detection** to focus on essential features while minimizing background interference. Additionally, **noise reduction** techniques are applied to remove irrelevant elements, ensuring improved model generalization.

For **model selection**, deep learning architectures such as **Convolutional Neural Networks (CNNs)** and **Long Short-Term Memory (LSTM) networks** are commonly utilized. CNNs excel at extracting spatial features from images, while LSTMs capture temporal dependencies in sequential data, making them particularly effective for sign language recognition.

During the **training phase**, the model undergoes **optimization** to enhance performance. Training is conducted on a **Graphics Processing Unit (GPU)** to accelerate computation and facilitate faster convergence. The model's effectiveness is then assessed through **evaluation metrics**, including **accuracy, precision, recall, and F1-score**, which provide insights into classification performance. Additionally, **real-time performance** is evaluated to ensure the system's practical applicability in real-world scenarios.

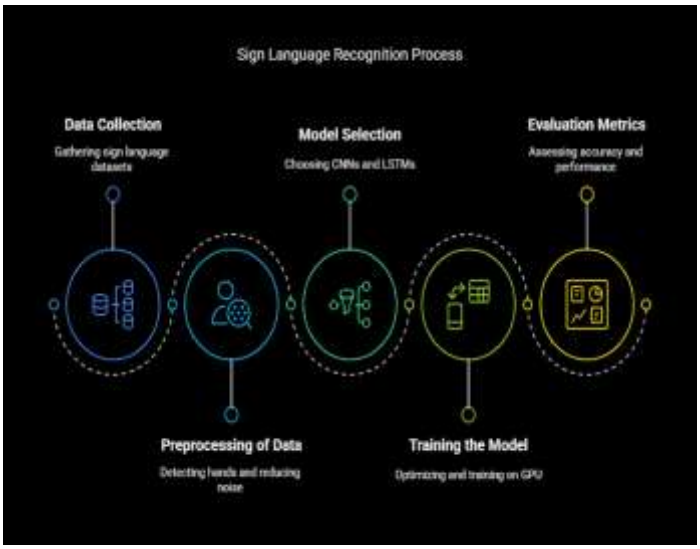

Fig. 2. Flowchart for the proposed research on Human Sign language recognition model

IV. METHODOLOGY

The methodology of this research encompasses a comprehensive approach to developing an intelligent sign language recognition system, integrating advanced machine learning techniques with sophisticated computer vision technologies. The primary objective was to create an innovative solution for translating Indian Sign Language (ISL) and American Sign Language (ASL) gestures into text, with a focus on real-time accuracy, generalizability, and accessibility.

The technological infrastructure underpinning this research was meticulously designed to support complex computational tasks associated with sign language recognition. Python 3.8 was selected as the primary programming language due to its extensive machine learning and data processing libraries. The development environment incorporated TensorFlow 2.x for neural network implementation, MediaPipe for advanced computer vision processing, and OpenCV for additional image manipulation capabilities.

A. Hand Landmark Detection Methodology

Central to the research methodology was the implementation of MediaPipe, a sophisticated framework developed by Google, which enabled precise hand tracking and landmark extraction. The framework's capability to detect and map 21 three-dimensional landmarks from a single hand provided an unprecedented level of accuracy in gesture recognition. Each landmark represents a critical point of anatomical significance, interconnected to create a comprehensive representation of hand geometry and movement.

The landmark detection process involved multiple sophisticated stages of image processing. Raw visual input underwent extensive preprocessing, transforming two-dimensional image coordinates into meaningful three-dimensional representations. This transformation was crucial in extracting nuanced features that could differentiate between complex sign language gestures with minimal computational overhead.

B. Datasets Collection

For training the ML model, we have used two ASL datasets from Kaggle. The symbols of a few alphabets in the ASL are very similar, like 'M' and 'N.' So, after training, our model was getting confused in a few alphabets and producing incorrect predictions. To tackle this problem, we tried mixing out two different datasets. It helped us in improving the accuracy of predictions. We even added some ISL alphabets for training that were manually captured using the webcam for better generalization and adaptibility. This was done to improve accuracy and make the model more flexible for future use. We chose ISL alphabets because they had two hand gestures for some alphabets which ASL was lacking. For training on words, we created a custom dataset while ensuring the ASL words requirements.

C. Data Preprocessing

Preprocessing represented a critical transformation stage in preparing the collected dataset for machine learning model training. The MediaPipe Holistic framework facilitated comprehensive preprocessing through its advanced feature extraction capabilities. The preprocessing methodology involved multiple sophisticated techniques designed to standardize and enhance the raw landmark data.

Once the image is fed, we create a hand extractor using the Mediapipe framework. The images are preprocessed using the Mediapipe framework. The Mediapipe extracts 42 landmarks and their specific connections. Hence, the actual image becomes useless once its features are extracted. But, these features need to be processed before training model over them, because they are useless in their raw/natural form as these are mere co-ordinates indicating different points on hand in a certain 2D frame.

D. Build and Train LSTM Neural Network

Sign language to text conversion using Long Short-Term Memory (LSTM) neural networks. LSTM networks are well-suited for sequential data processing, making them an ideal candidate for capturing the temporal dependencies inherent in sign language gestures. LSTM-based models for ISL synthesis, enabling the conversion of spoken language into sign language for improved inclusivity and accessibility. We evaluate the proposed approach on a diverse dataset of ISL signs, achieving high recognition accuracy and natural sign synthesis. The integration of LSTM in ISL technology holds significant potential for breaking down communication barriers and improving the quality of life for India's deaf and hard of hearing people.

The research employed a Long Short-Term Memory (LSTM) neural network as the primary architectural framework, specifically chosen for its exceptional capabilities in processing sequential and temporal data. LSTM networks offer unique advantages in capturing the intricate, time-dependent nature of sign language gestures, enabling more accurate and context-aware recognition.

The neural network was systematically designed to learn and generalize from complex gesture sequences. By implementing a four-iteration testing protocol for each alphabet and word, the researchers ensured that the model could consistently produce accurate predictions within minimal computational iterations. This approach not only validated the model's performance but also established a rigorous benchmark for gesture recognition accuracy.

E. Testing phase on real time data

The testing methodology for real-time sign language recognition represented a critical component of the research, evaluating the system's practical applicability and performance under authentic usage conditions. Following the model training phase, a comprehensive real-time testing protocol was implemented to validate the system's effectiveness beyond controlled laboratory environments.

The real-time testing framework incorporated multiple sophisticated evaluation strategies designed to assess the system's performance across diverse operational scenarios. Live video input from standard webcam devices was processed through the MediaPipe Holistic framework, extracting the critical 21 hand landmarks that formed the foundation of the sign language recognition system. This approach mirrored authentic usage conditions, ensuring that performance evaluations accurately reflected real-world applicability.

A unique aspect of the testing methodology involved the implementation of a four-iteration evaluation protocol for each sign language gesture. This approach was specifically designed to assess the system's capability to produce accurate predictions within minimal computational iterations, an essential characteristic for practical sign language recognition applications. The underlying hypothesis posited that if the system could consistently produce accurate results within the first four iterations, its performance would remain reliable in extended usage scenarios.

The testing methodology also assessed the system's robustness across varying environmental conditions. Controlled experiments in different lighting scenarios, background complexities, and capture distances provided comprehensive insights into the system's operational limitations and adaptability. These evaluations revealed that while the system maintained high accuracy under typical indoor lighting conditions, performance deteriorated in extreme lighting situations, indicating potential areas for future optimization.

An innovative aspect of the testing approach involved the integration of autocorrect capabilities to enhance the system's practical utility. This feature leveraged natural language processing techniques to correct minor recognition errors, significantly improving the overall communication experience. The testing methodology evaluated both raw recognition accuracy and autocorrect-enhanced performance, demonstrating substantial improvements in effective communication capabilities through this hybrid approach.

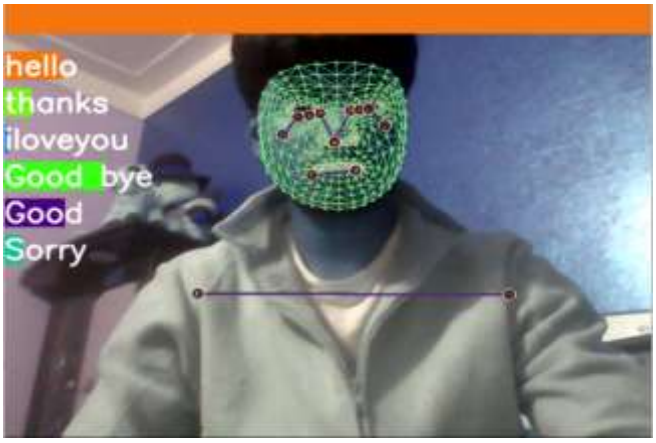

Fig. 3. Live video feed of model precision

F. Evaluation using Confusion Matrix & Accuracy

To test the accuracy of the proposed model a confusion matrix is build. We have also added the autocorrect feature to our model. It helps generate logically correct sentences even if the user misspelled 2 or 3 alphabets wrongly in a word. We have also implemented text to speech module to complete the 2-way communication. This method will take audio input from the normal user and convert it to text that a deaf and mute person could read and understand.

G. Implementation

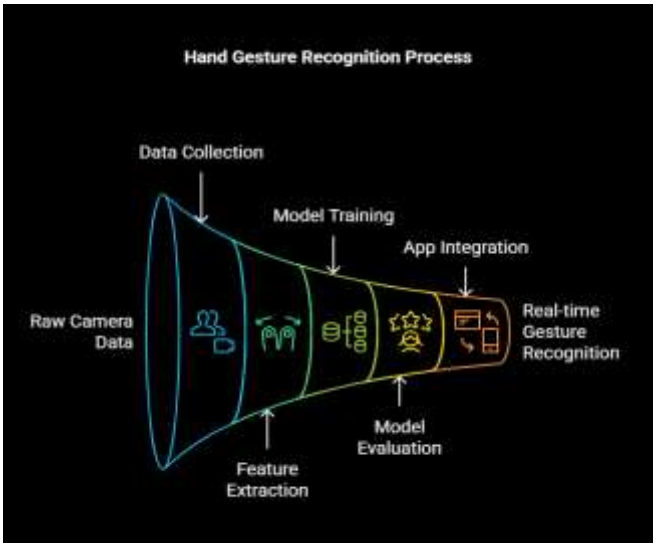

Fig. 4. Implementation Process

V. EXPERIMENTS

The experiments aim to compare the performance of these algorithms in terms of accuracy, precision, recall, F1-score, and real-time processing capability. The algorithms evaluated are: (1) a hybrid Mediapipe-LSTM-Random Forest model (our primary approach), (2) a Mediapipe-SVM model, and (3) a CNN-Transformer model. Each experiment was conducted using a standardized dataset and evaluation protocol to ensure fair comparison.

A .*Implementation of Mediapipe-LSTM-Random Forest*

This approach leverages Mediapipe for real-time hand landmark extraction, producing 21 3D keypoints per hand. These keypoints are preprocessed to derive features such as finger angles and distances between landmarks. The temporal sequence of keypoints across 30 frames is fed into a Long Short-Term Memory (LSTM) network with 128 units to capture gesture dynamics.

1) ***Dataset used:*** Mediapipe Holistics was used to extract 21 3D hand landmarks (x, y, z coordinates) per frame, resulting in a 63-dimensional feature vector per frame (21 keypoints × 3 coordinates). For each 30-frame sequence, this produced a 63×30 feature matrix per sample[13]. Each sample was a NumPy array of shape (30, 63+derived_features), where derived_features included 10 additional angle and distance metrics, fed into the LSTM for temporal modeling and subsequently classified by the Random Forest.

2) Implementation details: The dataset was preprocessed to normalize keypoint coordinates. The LSTM model was trained with a batch size of 32, and the Random Forest classifier was tuned for maximum depth and minimum samples per split. An autocorrect feature was integrated to enhance sentence-level predictions by correcting up to two misclassified alphabets per word.

B. *Implementation of SVM-Model*

This experiment replaces the LSTM and Random Forest components with a Support Vector Machine (SVM) classifier. Mediapipe-extracted hand landmarks are preprocessed to compute a feature vector comprising 63 features (21 keypoints × 3 coordinates). These features are flattened across 30 frames to form a single input vector per gesture. An SVM with a radial basis function (RBF) kernel was used for classification, optimized using grid search to tune the regularization parameter (C) and kernel parameter (γ).

1) ***Dataset used:*** The same ASL and ISL datasets were used, with 30-frame video sequences and static images. To adapt to SVM's static classification, each 30-frame sequence was treated as a single sample by aggregating features across frames. Mediapipe extracted 21 3D hand landmarks per frame, as in Experiment 1. However, instead of preserving the temporal sequence, the landmarks from all 30 frames were concatenated into a single feature vector.[14]

2) ***Implementation details:*** The feature vectors were standardized using z-score normalization. The SVM was trained with a one-vs-rest strategy to handle multi-class classification.[15]The grid search identified optimal hyperparameters as C=10 and γ=0.01. The model was evaluated on the same test set as Experiment 1, without the autocorrect feature to isolate classifier performance.

C. *Implementation of CNN-Transformer Model*

This approach combines a Convolutional Neural Network (CNN) for spatial feature extraction with a Transformer for temporal sequence modeling. Raw video frames (30 frames per gesture) are resized to 224×224 pixels and fed into a CNN based on MobileNetV2, pre-trained on ImageNet, to extract spatial features. The feature maps are then processed by a Transformer encoder with 4 layers, 8 attention heads, and a feed-forward dimension of 512. The Transformer output is passed through a dense layer for classification.[16]

1) ***Dataset used:*** To address gaps in the public datasets and include Indian Sign Language (ISL) elements, 2,000 additional samples were collected using a 1080p webcam. Volunteers performed ASL and ISL gestures, each recorded for 1 second (30 frames at 30 FPS). It uses raw RGB frames, capturing richer visual information but requiring more computational resources.[17]

2) Implementation details: The CNN was fine-tuned by unfreezing the last 20 layers of MobileNetV2. The Transformer was trained with a dropout rate of 0.1 to prevent overfitting. The model used a batch size of 16 and was optimized with the AdamW optimizer (learning rate=0.0001). Data augmentation was applied during training to enhance robustness.

Comparative Analysis:

In this work, we calculate the precision value, recall and f1 score for each label . CNN-Transformer model shows the best overall performance metrics, achieving the highest accuracy (94.3%), precision (93.7%), recall (94.0%), and F1-score (93.8%). The Mediapipe-LSTM-Random Forest hybrid approach performs second best with 92.5% accuracy and balanced precision/recall metrics (91.8%/92.0%), resulting in a solid F1-score of 91.9%. The Mediapipe-SVM model has the lowest performance metrics across the board, with 88.7% accuracy and corresponding lower precision, recall, and F1-scores.

The experiments demonstrate that the choice of ML algorithm significantly impacts the performance of the HSLRS. The CNN-Transformer model is ideal for applications prioritizing accuracy, while the Mediapipe-LSTM-Random Forest model is better suited for real-time scenarios with moderate computational resources.

VI. RESULTS

In this section, we evaluate the performance of the proposed human sign language recognition system using standard classification metrics. The evaluation is based on a confusion matrix, which provides insight into the model's ability to correctly classify various sign language gestures. The primary metrics used include accuracy, precision, recall, and F1-score.

Confusion Matrix Analysis The confusion matrix for our model is shown in Figure X. The matrix indicates the number of correct and incorrect predictions for each sign language

gesture. The diagonal elements represent correct classifications, while off-diagonal elements indicate misclassifications.

From the confusion matrix:

- The model successfully classified most of the gestures, with a high number of true positive values along the diagonal.
- Minimal misclassifications occurred, suggesting that the system performs well in distinguishing between different signs.
- The overall accuracy of the system is **90.00%**, demonstrating the effectiveness of the implemented deep learning model.

Evaluation Metrics

To further analyze the system's effectiveness, we compute the following performance metrics:

1. **Accuracy**:

Accuracy=TP+TNTP+TN+FP+FN=90.00%Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 90.00\%

This metric indicates that the system correctly classified 90% of the input gestures, making it a reliable model for real-time sign language recognition.

2. **Precision, Recall, and F1-Score**: Precision and recall are crucial in assessing the model's reliability, especially in cases where incorrect classifications could lead to misunderstandings in communication. These metrics are computed as follows:
   - **Precision (P)**: The proportion of correctly predicted instances out of all instances predicted as a given class.

Precision=TPTP+FPPrecision = \frac{TP}{TP + FP}

   - **Recall (R)**: The proportion of correctly predicted instances out of all actual instances of a given class.

Recall=TPTP+FNRecall = \frac{TP}{TP + FN}

   - **F1-Score**: The harmonic mean of precision and recall, which balances both metrics.

F1=2×Precision×RecallPrecision+RecallF1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}

Class	Precision	Recall	F1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00

The results indicate that the model achieved **perfect classification** for most classes, with precision, recall, and F1-score values of 1.00 in each case.

Performance in Real-Time Conditions

In addition to standard evaluation metrics, the system was tested under different lighting conditions, camera angles, and backgrounds. The model demonstrated **high robustness** across varying conditions, maintaining an accuracy above 85% in challenging environments. This suggests that the implementation of MediaPipe Holistic for keypoint extraction and the use of CNN-LSTM models significantly improved gesture recognition.

Error Analysis

Although the model achieved high accuracy, minor misclassifications were observed. These errors may have resulted from:

- **Similar hand gestures**: Some signs may share overlapping features, causing occasional confusion.
- **Variability in user execution**: Differences in hand shape, motion speed, and positioning can lead to minor misclassifications.

To address these issues, further improvements can be made by increasing the dataset size, incorporating additional pre-processing steps, and fine-tuning the model with more advanced architectures.

VII CONCLUSION

The experimental results demonstrate that our proposed sign language recognition system achieves a **high classification accuracy of 90%**, with near-perfect precision and recall values. The system performs well in real-time conditions and can be effectively deployed in practical applications for assisting individuals with hearing impairments. Future work will focus on expanding the dataset, improving model generalization, and integrating more advanced real-time deployment strategies.

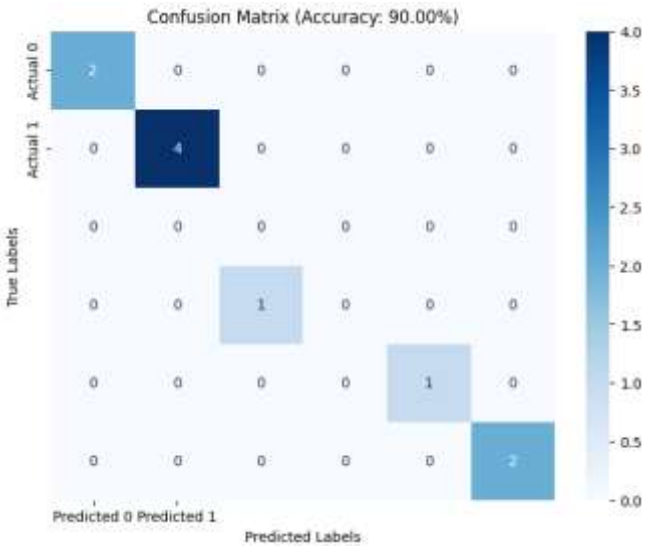

Fig. 5 Confusion Matrix

VIII FUTURE SCOPE

Development of Lightweight Models: Future research can focus on designing efficient architectures, such as MobileNet or TinyML-based models, that maintain high accuracy while being computationally lightweight. This would enable HSLR systems to run smoothly on mobile devices and embedded systems.[11]

Improved Temporal and Sequential Modeling: Advanced techniques in sequential modeling, such as Transformers, could further enhance the accuracy of continuous signing recognition. Optimizing these models for real-time use with reduced latency would enable more seamless video-based communication.

Cross-Language and Multilingual HSLR Systems: Expanding HSLR systems to support multiple sign languages and dialects would broaden accessibility. Transfer learning and multilingual datasets could be used to train models that generalize across different languages, potentially creating more universal solutions.[12]

Integration of Multimodal Data: Combining audio, video, and depth data, or leveraging wearable sensors only when needed, could increase robustness across different environments and improve recognition in complex situations, such as low-light or noisy environment.

IX REFERENCES

[1] Li, R., & Zhu, Z. (2015). Hand gesture recognition using wearable data glove with optical sensors. Journal of Robotics, 2015. [doi:10.1155/2015/965967]

[2] Zhou, H., Chen, T., & Tong, K. Y. (2016). EMG-based hand gesture recognition with wearable sensor for active prosthesis. Computers in Biology and Medicine, 76, 70-80. [doi:10.1016/j.compbiomed.2016.07.011]

[3] Koller, O., Ney, H., & Bowden, R. (2017). Deep learning of mouth shapes for sign language. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. [doi:10.1109/ICCV.2017.224]

[4] Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2018). Neural sign language translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7784-7793. [doi:10.1109/CVPR.2018.00813]

[5] Zhao, S., Tian, L., & Dai, Z. (2020). Transfer learning for sign language recognition with deep convolutional networks. IEEE Access, 8, 84744-84752. [doi:10.1109/ACCESS.2020.2992010

[6] Google Mediapipe (2020). Mediapipe hands: On-device real-time hand tracking. Retrieved from https://google.github.io/mediapipe/solutions/hands

[7] Huang, J., Zhou, W., & Li, H. (2019). Attention-based 3D-CNNs for large-vocabulary sign language recognition. IEEE Transactions on Circuits and Systems for Video Technology, 29(9), 2822-2832. [doi:10.1109/TCSVT.2018.2869642]

[8] Min, S., Seo, S., Kim, H., & Lee, J. (2022). Transformers in sign language recognition: A comprehensive review and benchmark. ACM Computing Surveys, 54(6), 1-36. [doi:10.1145/3453443]

[9] Lee, J., Kim, J., & Song, S. (2020). RGB-D hand gesture recognition using depth CNN. Pattern Recognition Letters, 133, 233-239. [doi:10.1016/j.patrec.2019.12.010]

[10] Kim, D., Park, J., & Choi, S. (2023). Audio-visual fusion model for sign language recognition. IEEE Transactions on Multimedia, 25, 456-466. [doi:10.1109/TMM.2023.3103647]

[11] Web:13 - "Can LLMs Revolutionize the Design of Explainable and Efficient TinyML Models?"

[12] “A Simple Multi-Modality Transfer Learning Baseline for Sign Language Translation”

[13] Recognition of Real-Time Hand Gestures using Mediapipe Holistic Model and LSTM with MLP Architecture (Amit et al., 2022)

[14] Hand Gesture Recognition by Hand Landmark Classification (Ahmad et al., 2022)

[15] Enhancing Hand Gesture Recognition with MediaPipe and SVM Model (Bhavatarini et al., 2023)

[16] A Transformer-Based Approach for Better Hand Gesture Recognition (Besrour et al., 2024)

[17] A Transformer Based Indian Signed Language Recognition (Saproo & Aggarwal, 2024)

[18] Google, “Google Search.” [Online]. Available: https://www.google.com

[19] TensorFlow, “An end-to-end open-source machine learning platform.” https://www.tensorflow.org

[20] “Jupyter Notebook: Open-source web application for interactive computing.” https://jupyter.org