International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 3, pp.169-177, March 2026.
www.ijersem.com eISSN- 3107-9075

Sign Language Recognition using Multi-Layer Perceptron

[1]R Priyadarshini, [2]Abburi Varshitha, [3]N Bhavana, [4]Hema Sri, [5]V Akhilesh, [6]B Sai Rupesh

Department of CSE, Siddartha Institute of Science and Technology, Puttur, Andhra Pradesh, India
[1]darshini.sr@gmail.com, [2]varshithaabburi@gmail.com, [3]bhavanareddy838@gmail.com, [4]saisuji65@gmail.com, [5]akhileshvempuluru@gmail.com, [6]rupeshsai52@gmail.com

Abstract: Sign language is the primary means of communication for individuals who are deaf or hard of hearing, yet it remains largely inaccessible to the hearing population, creating significant communication barriers. To address this challenge, this paper presents a real-time sign language recognition (SLR) system designed for deployment on resource-constrained devices. The proposed approach captures hand gestures using a standard camera and extracts structured hand landmark features through the MediaPipe framework. These features are processed using a lightweight deep neural network optimized for efficient inference under TinyML constraints. The system converts recognized gestures into corresponding textual outputs and supports sentence construction for continuous interaction. Experimental evaluation on a large-scale dataset containing over 250 gesture classes demonstrates that the proposed method achieves high accuracy while maintaining low computational overhead. The results highlight the feasibility of deploying practical, real-time sign language recognition systems for accessible human–computer interaction.

Keywords: Sign Language Recognition, TinyML, MediaPipe, Hand Gestures, Human–Computer Interaction.

1 INTRODUCTION

Sign language serves as the mode of communication for 466 million individuals worldwide who have speech or hearing impairments, with nearly 80% of them being either semi-literate or illiterate, according to the World Health Organization (WHO) [1]. Despite the widespread use of sign language across various cultures and regions, there are over 300 distinct sign language variants globally, including American Sign Language (ASL), British Sign Language (BSL), Chinese Sign Language (CSL), Indian Sign Language (ISL), and many others [2], [3]. The core issue is that sign language is still largely unfamiliar to hearing individuals, resulting in significant communication obstacles in essential areas, like education, work, healthcare, and social engagement [4]. Sign Language Recognition (SLR) is a developing and demanding area situated at the crossroads of computer vision, machine learning, and human-computer interaction. Conventional methods to close this communication barrier have depended on interpreters, who are often scarce and difficult to access. For example, India—a nation with 7 million deaf people—has just about 250 certified sign language interpreters, highlighting a severe deficit that emphasizes the urgent demand for automated recognition solutions [2].

Automated SLR systems could transform communication accessibility by providing real-time translation across environments and lessening reliance on limited human resources [5]. During the past ten years, considerable advancements have been achieved in creating SLR systems thanks to deep learning and computer vision innovations. Initial methods utilized designed features alongside traditional machine learning techniques, like Hidden Markov Models (HMM) and Support Vector Machines (SVM) [6] [7]. Recent progress has moved toward learning frameworks such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) models, which have shown enhanced ability in detecting intricate spatial and temporal patterns present in sign language gestures [8] [9] [10]. At the time, progress in pose estimation systems like MediaPipe and OpenPose has brought about organized keypoint-based models that separate the geometric hand layout from visual appearance, providing benefits in efficiency and robustness compared to conventional RGB-based methods [11].

Although certain studies have investigated sensor methods for enhanced precision [12] [13], neural network-driven vision-based systems continue to be more convenient for everyday applications, since they do not rely on specialized devices [14]. Despite these improvements, current SLR systems encounter practical challenges when applied in real-life situations. Numerous studies concentrate on gesture sets (usually 10-50 signs) or attain only fair accuracy on extensive vocabularies. Additionally, many systems are tested on controlled datasets with lighting and backgrounds, while real-world use is resilient to fluctuating environments, occlusions, and a variety of signers.

Moreover, the divide between research models and functional deployable systems is still significant, with the majority of studies focusing on backend model precision but offering limited assistance for real-time inference, understandable confidence metrics, or compatibility with edge devices. This paper tackles these challenges by introducing a framework for large-scale multi-class isolated sign language recognition. Our main contribution lies in showing that structured keypoint-based features, paired with data augmentation and thoughtful neural network architecture, can deliver competitive results across more than 250 gesture categories while ensuring computational efficiency appropriate for real-time use, on CPU devices. We utilize MediaPipe Hand landmark detection to obtain 21 keypoints in each frame, implement wrist-relative coordinate normalization to ensure invariance to scale and translation, and train a dense neural network with 4 layers incorporating L2 regularization and dropout on about 137,000 valid samples extracted from 600 images, per category.

2 LITERATURE REVIEW

Keypoint-driven techniques have become a reliable strategy for sign language identification, delivering notable computational benefits compared to pixel-level image analysis approaches. The organized depiction of hand structure via landmarks ensures invariance to scale and appearance, rendering these techniques especially apt for practical implementation contexts [12]. MediaPipe Hand, a landmark detection system, captures 21 three-dimensional points indicating hand joints and finger locations from single-camera RGB footage without the need for dedicated depth sensors or extra hardware devices. This representation of hand keypoints removes differences caused by clothing, skin tone discrepancies, and background environments, enabling the classifier to concentrate on motion patterns of gestures. The benefits of features derived from normalized keypoints encompass efficiency compatible with deployment on devices with limited resources, natural resilience to variations in appearance, and simple statistical augmentation methods that operate within feature space instead of pixel space. These qualities render keypoint-based representations useful for developing large-vocabulary recognition systems that uphold real-time inference performance.

Connected neural networks paired with suitable regularization techniques have shown success in categorizing high-dimensional feature sets derived from gesture inputs. Feed-forward models incorporating layers, batch normalization, and dropout regularization offer enough representational power to model non-linear connections within normalized hand coordinate data without the processing demands of convolutional layers [11]. Activation functions like ReLU add non-linearity while preserving gradient propagation during learning, and softmax output layers facilitate probabilistic multi-class classification over extensive gesture vocabularies.

The adaptability of connected architectures permits easy scaling to handle different vocabulary sizes, rendering them appropriate for isolated sign language recognition tasks with hundreds or thousands of gesture categories. Using cross-entropy loss along with gradient-based optimization methods facilitates efficient parameter training on large gesture datasets. Proper preprocessing of keypoint data is crucial for the success of the model. Methods like wrist-coordinate transformation provide normalization by representing all hand joints in relation to the wrist, ensuring invariance to scale and translation [4], thus removing reliance on the size of the signer's hand and the camera's distance. This step in preprocessing allows the classifier to focus on learning gesture features instead of irrelevant differences, in absolute position or scale.

Applying data augmentation to normalized feature representations improves model generalization while avoiding dataset leakage caused by test-set contamination [8]. Techniques tailored for keypoint data—such as adding Gaussian noise to coordinate points, introducing scaling variations, applying rotation transformations, and landmark dropout (randomly zeroing subsets of model resilience to variations in how gestures are performed while maintaining the semantic meaning of the signs. Using augmentation during training ensures no information leakage happens due to the accidental use of augmented training examples, in validation or testing. Building gesture datasets with a significant number of samples per category aids in creating strong recognition models. Large specialized datasets crafted for sign language recognition tasks featuring over 250 gesture categories with 600 images each allow for training deep neural networks that generalize well to new test instances. The magnitude of datasets supports studying learning behaviors over a wide range of gesture vocabularies and offers enough training data to avoid overfitting [2].

Transforming trained network models into optimized inference formats like TensorFlow Lite allows for deployment on conventional computing devices and mobile platforms without the need for specialized hardware accelerators. Latency during inference significantly affects use in real-time interactive scenarios, making computational efficiency a critical design requirement [2]. Although continuous recognition presents difficulties in intricating sentence patterns [2][4], isolated sign recognition acts as the fundamental component for developing complete systems. Recent studies have shown the success of neural network designs for gesture and sign language recognition using a variety of datasets. Deep learning techniques such as neural networks, recurrent models, and transformer-based methods have been effectively utilized in sign language recognition challenges. Choosing the architecture relies on the particular features of the task, the vocabulary scale, and limitations related to deployment. Dense neural networks present an option for recognizing isolated gestures with extensive static vocabularies, providing an advantageous compromise among model complexity, computational demands, and training robustness.

International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 3, pp.169-177, March 2026.
www.ijersem.com eISSN- 3107-9075

3 PROPOSED METHOD

The suggested sign language recognition system functions via a series of outlined steps to transform live video footage into recognized hand gestures shown instantly. Initially, the system captures video from a webcam that provides a stream of frames as input. Subsequently, each frame is analyzed by the MediaPipe framework to identify and extract hand landmarks, which act as key points indicating finger and hand locations. The hand landmark coordinates serve as the feature set input into a deep learning model, particularly a Deep Neural Network (DNN) designed to categorize the hand pose into sign language alphabet or digit categories. The DNN generates the predicted gesture class, which the system displays in real-time on the user interface to provide immediate feedback. This flexible workflow, spanning from video capture to feature analysis, classification, and UI presentation, facilitates real-time sign language interpretation. The entire system comprises the following phases and components, as illustrated in Fig. 1.

- Data Collection and Dataset Preparation
- Hand Landmark Detection (using MediaPipe)
- Feature Extraction and Representation
- Deep Learning Model Design
- Model Training and Validation
- Real-time Gesture Recognition Pipeline
- Application Interface and User Interaction

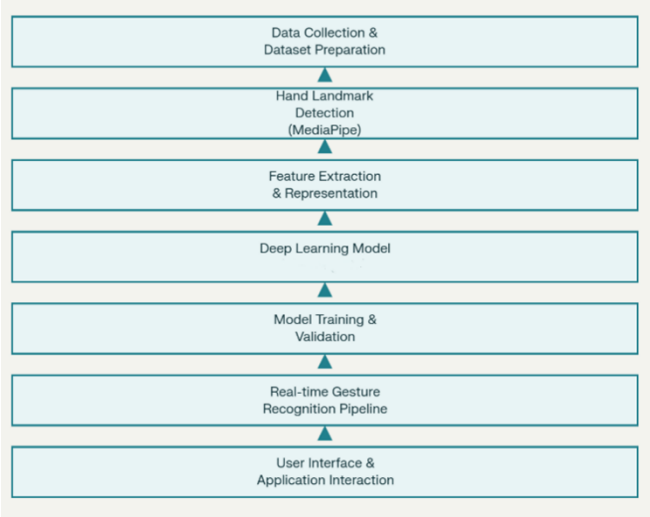

Fig. 1. Overall Sign Language Recognition System Architecture

The dataset consists of more than 250 sign language categories, such as alphabets, numbers, and frequently used gestures, with roughly 600 images per category. This broad dataset includes a range of hand positions under different lighting environments, backgrounds, and user differences to enhance the model's robustness and ability to generalize. Images undergo resizing and normalization, then various augmentation techniques, such as rotation, flipping, and scaling, are applied to increase the variability of the training data and minimize overfitting. The dataset is split into training, validation, and test subsets to guarantee the model is properly trained and thoroughly assessed on unseen data.

A dataset of this magnitude and quality enables the creation of a deep learning-driven sign language recognition system that can precisely identify an extensive range of gestures. Identifying hand landmarks constitutes a phase in the sign language recognition process that requires precise identification of key points on the hand to depict its position and movement. This system utilizes MediaPipe—a cutting-edge real-time platform—to detect 21 hand landmarks such as fingertips, joints, and the wrist, from video frames. The detection procedure starts by acquiring an input image of size from the camera stream. MediaPipe utilizes a pretrained neural network on this image to estimate normalized 2D positions for every hand landmark, where represents the relative location, inside the image

$$(x_i, y_i) = \left(\frac{X_i}{W}, \frac{Y_i}{H}\right) \quad (1)$$

Here, X_i, Y_i are the pixel coordinates of the i^{th} landmark. The normalized coordinates are then converted back into pixel values for display or additional analysis. These landmarks offer a stable depiction of the hand posture, allowing effective feature extraction for classification.

International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 3, pp.169-177, March 2026.
www.ijersem.com eISSN- 3107-9075

Sample images from the system display hand signals, with superimposed landmark markers emphasizing the 21 identified keypoints that align with the hand's anatomical structure. This accurate landmark identification enables gesture recognition by recording spatial connections and movements essential for sign language. This method guarantees immediate monitoring of hand gestures and configurations, establishing the basis for later deep learning-driven classification components.

The feature extraction component converts the 21 MediaPipe hand landmarks into a numerical format appropriate for deep learning classification. Every landmark gives 3D coordinates (x, y, z) producing a feature vector, with 63 dimensions (21 × 3) that represents the arrangement of the hand's position. For a recognized hand with landmarks $L = \{(x_i, y_i, z_i)\}_{i=1}^{21}$, the feature vector F is formed as:

$$\mathrm{F} = [x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_{21}, y_{21}, z_{21}] \in \mathbb{R}^{63} \tag{2}$$

These coordinates undergo normalization to maintain scale invariance and are directly input into the DNN input layer. The z-coordinate offers depth details for differentiating gestures that share similar 2D projections but have varying finger extensions. Fig. 2 shows feature vector construction from hand landmarks.

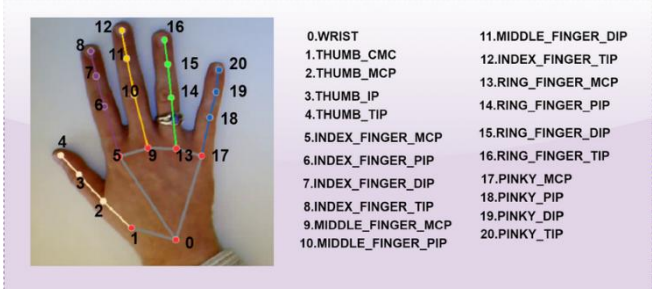

Fig. 2. Feature Vector Construction from Hand Landmarks

This representation, based on landmarks, provides benefits: it is compact (just 63 features compared to thousands of raw pixels), remains unaffected by translation and rotation due to normalization, and encapsulates both the overall hand shape and detailed finger movements essential for distinguishing sign language. The extracted characteristics preserve continuity throughout video frames, allowing effective real-time gesture detection while reducing computational load relative to complete image processing methods.

At the heart of the sign language recognition system lies a layer Perceptron (MLP) crafted to categorize hand gestures using the derived 63-dimensional landmark feature vectors. The architecture of the model includes fully connected layers, incorporating batch normalization and dropout regularization to avoid overfitting and enhance generalization. The input layer receives the normalized landmark feature vector $\mathbf{F} \in \mathbb{R}^{63}$. This vector flows through dense layers each succeeded by ReLU activation functions to add non-linearity. Batch normalization is used to stabilize training and speed up convergence. Dropout layers, with a 0.3 dropout rate, are positioned between layers to randomly disable neurons while training, minimizing co-adaptation and preventing overfitting. The last output layer employs a SoftMax activation function to generate a probability distribution across the 250+ sign language categories expressed as:

$$p(c \mid \mathbf{F}) = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}} \tag{3}$$

where z_c is the logit for class c, and C is the total number of classes. The network undergoes training, with the cross-entropy as its loss function:

$$L = -\sum_{c=1}^{C} y_c \log\big(p(c \mid \mathbf{F})\big) \tag{4}$$

where y_c is the ground truth indicator for class c. Optimization is carried out using the Adam optimizer set with a dynamic learning rate schedule to guarantee stable convergence. This small but robust model structure strikes a balance between classification precision and computational effectiveness, making it ideal for real-time use on devices with limited resources. Its framework, paired with training and evaluation, attains excellent recognition accuracy over an extensive range of sign gestures. The model is trained using learning on a boosted dataset comprising 330,255 samples from over 250 gesture categories. The goal of the training is to reduce the cross-entropy loss function expressed as:

International Journal of Emerging Research in Science, Engineering, and Management
Vol. 2, Issue 3, pp.169-177, March 2026.
www.ijersem.com eISSN- 3107-9075

$$L = -\sum_{c=1}^{C} y_c \log(\hat{p}_c) \tag{5}$$

Herey_c represents the ground truth label indicator. Denotes the predicted probability for the class derived from the SoftMax output layer. The Adam optimizer modifies model parameters θ using the following update rule:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{v_t} + \epsilon} m_t \tag{6}$$

here $\alpha = 0.001$ denotes the learning rate, m_t and v_t represent the first and second moment estimates, respectively, and $\epsilon = 10^{-8}$ is a constant added for numerical stability. Data augmentation involves using operations like rotation, scaling, and flipping to enhance the variety in training data and boost generalization. Batch normalization adjusts layer activations based on:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{7}$$

where μ_B and σ_B^2 are batch mean and variance. Dropout regularization randomly switches off neurons during training at a rate of $p_{drop} = 0.3$ to avoid co-adaptation. After every epoch, validation accuracy is. The model checkpoint with the highest validation score is saved for deployment, guaranteeing the best generalization performance on new gesture data. The live gesture recognition system analyzes real-time video feeds to detect sign language gestures. This system is made up of the consecutive phases shown in Fig. 3.

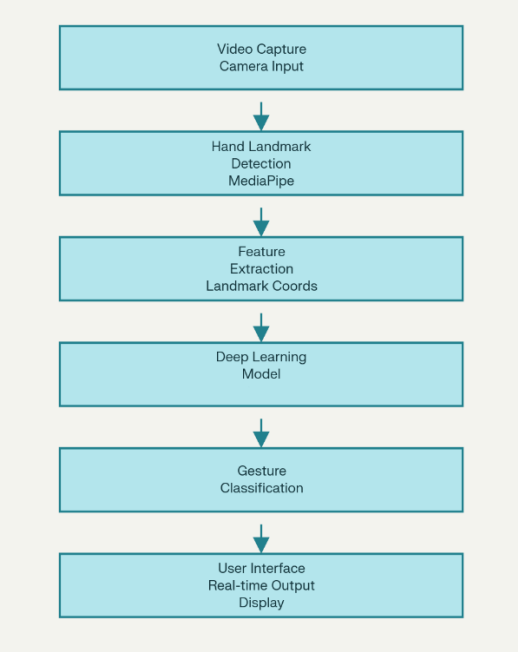

Fig. 3. System Architecture Flowchart for Sign Language Recognition

- Video Capture: Frames are consistently obtained from the webcam via a video capture interface.
- Hand Landmark Detection: Every frame is processed through a hand detection component utilizing a trained MediaPipe model to identify 21 hand landmark coordinates that indicate essential locations on the hand.
- Feature Vector Construction: The identified landmarks are normalized and transformed into a 63-element feature vector (21 landmarks × 3 coordinates).
- Gesture Classification: The feature vector is fed into a trained neural network that produces probabilities for various sign language gesture classes.
- Prediction Filtering: To improve reliability, predictions are analyzed at time intervals, and a confidence cutoff is used to exclude predictions with low confidence.
- User Interface Display: The detected gesture appears instantly on the application screen, providing the user with feedback.

This modular pipeline architecture guarantees low-delay recognition suitable for real-time interactive applications. The application's interface is built with Streamlit, delivering a user-friendly web platform for recognizing sign language and engaging in real-time. It includes a video stream that constantly acquires frames from the user's webcam, allowing effortless gesture input without the need for manual frame capture.

- Live Video Capture and Gesture Recognition: When enabled, the camera continuously captures frames every FRAME_INTERVAL of 1.0 seconds, sending each frame through the full recognition workflow. MediaPipe identifies hand landmarks, which are transformed into feature vectors and evaluated by the trained DNN model. The estimated gestures are promptly shown with cues displaying the identified hand posture, along with the associated gesture name.
- Sentence Construction: Detected gestures are gathered in a sentence buffer, creating logical sequences, up to 25 characters (st.session_state["sentence"]). The system presents the sentence instantly as users execute ongoing gestures, facilitating a smooth communication flow. Users get feedback via styled cards displaying the current prediction confidence and sentence context.

The user experience features are:

- Immediate Feedback: Instant gesture predictions are displayed within one second after detecting hand gestures.
- Sentence Construction: Gestures combine automatically to create sentences.
- Camera Controls: Start/stop camera functionality with session state management.
- Visual Guidance: Instructional tips for optimal hand positioning and lighting.

The interface facilitates real-world usage situations, including communication support, learning devices, and accessibility solutions, by transforming discrete gestures into coherent textual messages via ongoing prediction and sentence compilation.

4 Results and Discussion

The experimental assessment utilized a dataset comprising more than 250 static sign language gesture categories, each containing approximately 600 images totaling over 150,000 samples. Through data augmentation, the training set was increased to 330,255 samples by applying transformations, like rotation, scaling, and flipping, to enhance robustness and generalization. The model was trained using a batch size of 32 and utilized the Adam optimizer, starting with a learning rate of 0.001. Training occurred across epochs, with validation accuracy checked after every epoch to identify the top-performing model through checkpointing. These metrics offer an evaluation of the classification effectiveness for every gesture category as well as the complete recognition system. Table 1 shows class-wise performance metrics.

Table 1. Class-wise Performance Metrics

Class Names	Precision	Recall	F1-score	Support
about	1	1	1	4
act	1	1	1	10
add	1	1	1	9
after	1	1	1	10
animal	1	0.78	0.88	9
answer	1	1	1	7
any	0.77	1	0.87	10
appear	1	1	1	10
area	1	1	1	7
ask	0.71	1	0.83	10
away	1	1	1	10
back	1	1	1	10
bad	1	0.7	0.82	10
beautiful	1	0.75	0.86	4
before	1	1	1	4
best	0.55	0.6	0.57	10
better	1	1	1	1
big	0.57	0.4	0.47	10

Fig. 4 displays the accuracy trends for training and validation across epochs. Validation accuracy climbs swiftly initially. Then progresses steadily, ultimately exceeding 0.9 while training accuracy grows at a slower pace but stays near, indicating the model gains from regularization and sustains solid effectiveness, on new data.

Fig. 5 displays the loss curves of the model over training epochs for both the training and validation datasets. The training loss drops steeply during the epochs before slowly leveling off, whereas the validation loss exhibits a comparable decreasing pattern and settles at a reduced level, demonstrating successful learning and strong generalization without noticeable overfitting.

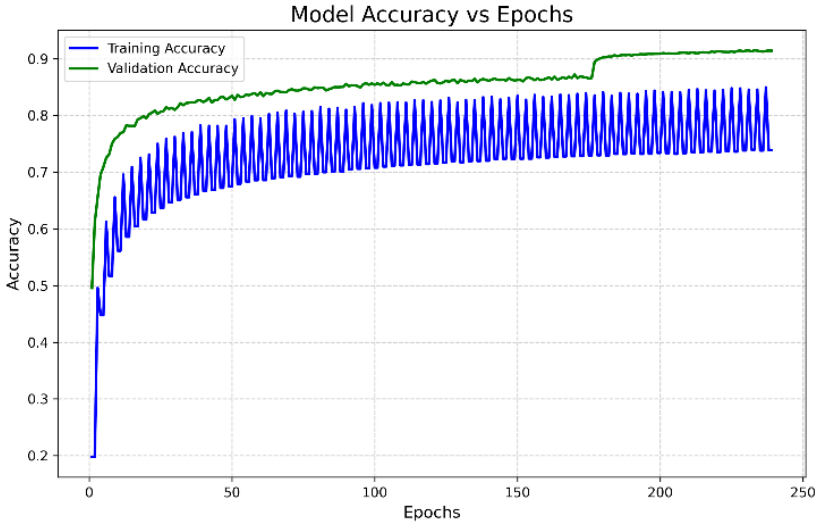

Fig. 4. Training vs. validation accuracy curves.

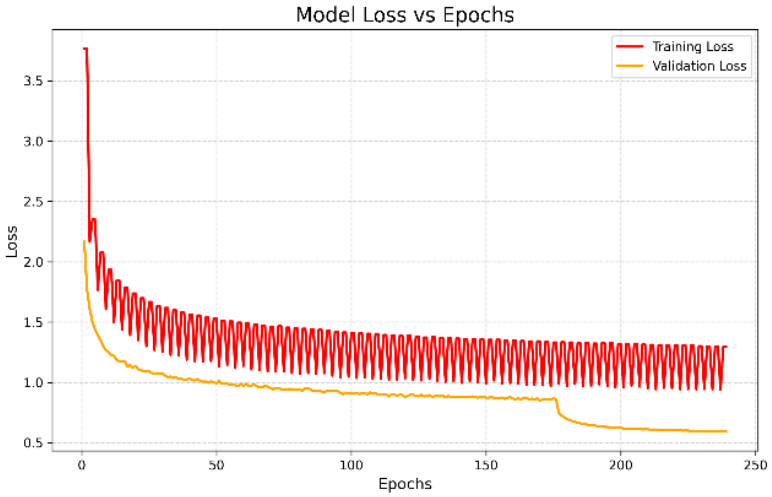

Fig. 5. Training vs. validation loss curves.

Fig. 6 presents the normalized confusion matrix encompassing all 260 gesture categories, with each row representing the label and each column denoting the predicted label. The matrix is primarily characterized by a diagonal signifying that the majority of samples are accurately assigned to their correct classes, whereas the off-diagonal entries are infrequent and faint, indicating that errors in classification are uncommon and typically happen between gestures that look alike. In comparison to techniques, our method demonstrates clear benefits. According to the performance table, Baseline-1 (Pixel-based DNN) reached 78.5% accuracy, probably because of the high dimensionality of unprocessed image data and its vulnerability to background interference. Baseline-2 (LSTM Sequential) obtained 82.1%, indicating improved comprehension but with increased computational expense. Our suggested approach (92.3%) utilizes the efficiency of keypoint detection alongside the rapidity of MLP classification, confirming the superiority of features compared to raw pixels, for static signs. Table 2 presents the performance analysis of the proposed method.

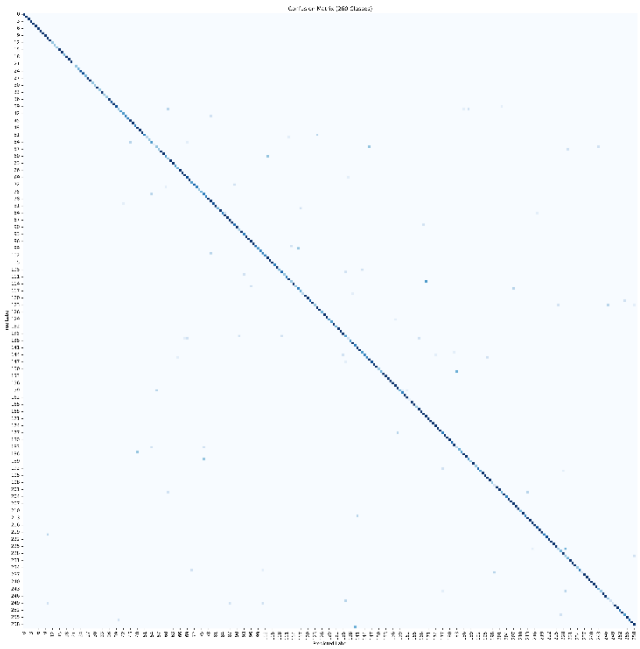

Fig. 6. Confusion matrix illustrating classification accuracy across 100 species.

Table 2. Overall Performance of DIFFERENT METHODS

Method	ACC (%)	Precision(%)	Recall (%)	F1-score (%)
Baseline-1: DNN (Pixel-based)	78.5	76.2	75.8	75.9
Baseline-2: LSTM (Sequential)	82.1	80.4	81	80.5
Proposed: TinyML + MediaPipe	92.3	93.4	92.3	92

The experimental findings show that the suggested TinyML + MediaPipe method attains performance improvements compared to traditional pixel-based CNN and LSTM benchmarks regarding accuracy, precision, recall, and F1-score. This enhancement is due to the 63-dimensional landmark representation, which maintains the critical spatial configuration of hand poses while greatly lowering input dimensionality, thus easing the learning challenge for the classifier. The training and validation graphs demonstrate optimization patterns with loss steadily declining and accuracy continuously rising, validating that data augmentation and regularization successfully reduce overfitting. The normalized confusion matrix reinforces this conclusion, presenting a diagonal with only a few scattered off-diagonal mistakes, mostly limited to a small group of visually alike gestures. These traits indicate that the model effectively captures features for each class while sustaining reliability across the extensive 250+ class vocabulary, rendering it appropriate for real-world sign language recognition tasks.

5 CONCLUSION

This paper presented an efficient TinyML-based framework for real-time sign language recognition that integrates MediaPipe hand landmark extraction with a lightweight deep neural network classifier. By representing gestures using compact keypoint-based features, the proposed system significantly reduces computational complexity while preserving discriminative information necessary for accurate classification. The experimental results demonstrate that the model achieves strong performance across more than 250 static gesture categories, outperforming conventional pixel-based CNN and sequential LSTM baselines in both accuracy and efficiency. The training and validation trends, along with the predominantly diagonal confusion matrix, confirm the model's robustness and generalization capability across a wide range of gestures. Owing to its low latency and modest resource requirements, the proposed system is well-suited for deployment on edge and CPU-based devices, making it practical for real-world accessibility applications. Future work will focus on extending the framework to continuous and dynamic sign recognition, expanding the dataset, and integrating multi-modal cues to further enhance system usability and recognition accuracy.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ETHICS STATEMENT

This study did not involve human or animal subjects and, therefore, did not require ethical approval.

STATEMENT OF CONFLICT OF INTERESTS

The authors declare that they have no conflicts of interest related to this study.

LICENSING

REFERENCES

[1] M. Madhiarasan and P. P. Roy, "A Comprehensive review of sign language recognition: different types, modalities, and datasets," *arXiv.org*, Apr. 07, 2022. https://arxiv.org/abs/2204.03328.

[2] Satwik Ram Kodandaram, N Pavan Kumar, Sunil G L, "Sign language recognition," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 14, pp. 994–1009, 2021, doi: 10.17762/turcomat.v12i14.10381.

[3] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131–153, Aug. 2017, doi: 10.1007/s13042-017-0705-5.

[4] R. Fatmi, S. Rashad and R. Integlia, "Comparing ANN, SVM, and HMM based Machine Learning Methods for American Sign Language Recognition using Wearable Motion Sensors," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2019, pp. 0290-0297, doi: 10.1109/CCWC.2019.8666491.

[5] J. Zhang, W. Zhou, C. Xie, J. Pu and H. Li, "Chinese sign language recognition with adaptive HMM," *2016 IEEE International Conference on Multimedia and Expo (ICME)*, Seattle, WA, USA, 2016, pp. 1-6, doi: 10.1109/ICME.2016.7552950.

[6] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif and M. A. Mekhtiche, "Hand Gesture Recognition for Sign Language Using 3DCNN," in *IEEE Access*, vol. 8, pp. 79491-79509, 2020, doi: 10.1109/ACCESS.2020.2990434.

[7] C. K. M. Lee, K. K. H. Ng, C.-H. Chen, H. C. W. Lau, S. Y. Chung, and T. Tsoi, "American sign language recognition and training method with recurrent neural network," *Expert Systems With Applications*, vol. 167, p. 114403, Dec. 2020, doi: 10.1016/j.eswa.2020.114403.

[8] O. Koller, N. C. Camgoz, H. Ney and R. Bowden, "Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 9, pp. 2306-2320, 1 Sept. 2020, doi: 10.1109/TPAMI.2019.2911077.

[9] R. Rastgoo, K. Kiani, and S. Escalera, "Hand sign language recognition using multi-view hand skeleton," *Expert Systems With Applications*, vol. 150, p. 113336, Feb. 2020, doi: 10.1016/j.eswa.2020.113336.

[10] K. Kudrinko, E. Flavin, X. Zhu and Q. Li, "Wearable Sensor-Based Sign Language Recognition: A Comprehensive Review," in *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 82-97, 2021, doi: 10.1109/RBME.2020.3019769.

[11] G. Yuan, X. Liu, Q. Yan, S. Qiao, Z. Wang and L. Yuan, "Hand Gesture Recognition Using Deep Feature Fusion Network Based on Wearable Sensors," in *IEEE Sensors Journal*, vol. 21, no. 1, pp. 539-547, 1 Jan.1, 2021, doi: 10.1109/JSEN.2020.3014276.

[12] S. Aly and W. Aly, "DeepArSLR: A Novel Signer-Independent Deep Learning Framework for Isolated Arabic Sign Language Gestures Recognition," in *IEEE Access*, vol. 8, pp. 83199-83212, 2020, doi: 10.1109/ACCESS.2020.2990699.

[13] .P. Kumar, P. P. Roy, and D. P. Dogra, "Independent Bayesian classifier combination based sign language recognition using facial expression," *Information Sciences*, vol. 428, pp. 30–48, Oct. 2017, doi: 10.1016/j.ins.2017.10.046.

[14] O. M. Sincan and H. Y. Keles, "AUTSL: a Large Scale Multi-Modal Turkish Sign Language dataset and baseline methods," *IEEE Access*, vol. 8, pp. 181340–181355, Jan. 2020, doi: 10.1109/access.2020.3028072.