From Gesture to Grammar: A Deep Neural Approach to Sign Language Understanding

Authors:

Avani Jaiswal
Bachelor of Technology in Artificial Intelliegence and Machine Learning
Indira Gandhi Delhi Technical University for Women
Delhi, India
email: avani015btaiml24@igdtuw.ac.in

Namita Belwal
Bachelor of Technology in Electronics and Communication Engineering with Artificial Intelligence
Indira Gandhi Delhi Technical University for Women
Delhi, India
email: namita088bteceai@igdtuw.ac.in

I. ABSTRACT

Sign language serves as an essential means of communication for the hearing-impaired community, yet the lack of widespread understanding often creates barriers in education, healthcare, workplaces, and daily interactions. To address this challenge, this study explores a computer vision and deep learning (CV-DL) approach for real-time recognition and translation of sign language gestures into text.

A curated dataset of static and dynamic gestures was preprocessed through resizing, normalization, and background removal to ensure consistency. Convolutional Neural Networks (CNNs) were employed for feature extraction, while recurrent layers modeled temporal gesture patterns.
The system was trained and evaluated using benchmark datasets, with performance measured through accuracy, precision, and recall.
The findings highlight the capability of CV-DL to bridge communication gaps and foster greater inclusivity. The proposed framework offers potential for deployment on mobile or web platforms, with future enhancements such as speech integration to enable smoother interaction between signers and non-signers.

II. INTRODUCTION

A. Background and Motivation

Communication is a fundamental aspect of human connection, yet for many individuals who rely on sign language, it remains a daily challenge. Since most people do not understand sign language, those who use it often experience barriers in education, healthcare, and even simple everyday conversations. This gap can lead to feelings of exclusion and highlight the urgent need for more inclusive solutions.

At the same time, advances in computer vision and deep learning have opened new possibilities in recognizing patterns, gestures, and movements with high accuracy. These technologies are already transforming areas such as image recognition and human–computer interaction. Extending their potential to sign language recognition can create a bridge between the hearing and non-hearing communities, making communication more natural and accessible. In this work, we explore American Sign Language (ASL) recognition as a first step toward that vision.

B. Problem Statement

The following are the Existing Approaches, our own Proposed Approach, and Scope of Study in our project:

Existing Approaches

- Sensor-based methods provide high accuracy but require specialized hardware (e.g., gloves), making them impractical for everyday use.
- Vision-based methods with machine learning and deep learning are more accessible but face challenges such as real-time recognition, background noise, and dynamic gestures.

Proposed Approach

- Develop a deep learning–based model using Convolutional Neural Networks (CNN) for American Sign Language (ASL) recognition.

- Employ accessible tools and libraries such as TensorFlow, Keras, OpenCV, and Mediapipe to ensure cost-effectiveness and efficiency.

Scope of Study

- Focus on static gesture recognition from the ASL dataset.
- Exclude letters **'J'** and **'Z'** as they require motion-based gestures.
- Train the model on static hand signs (**A–Y**) to achieve higher accuracy and reliability with image-based deep learning techniques.

C. Objectives/ Research Questions

With this motivation, our objectives became clearer:

->To design a simple but effective CNN-based model that can recognize ASL gestures from images.

->To preprocess and train on the ASL dataset and evaluate the model's performance.

->To test the model not just on the dataset but also using a webcam in real-time.

->To see how well our system performs under different conditions and identify the challenges.

->To explore how such models can later be extended towards real-world applications for helping the hearing-impaired community.

III. LITERATURE REVIEW

Research on sign language recognition has explored a range of methods, but challenges remain in achieving reliable real-time performance. Early studies used traditional computer vision techniques like contour detection, edge mapping, and handcrafted feature extraction. Although these methods were promising, they struggled with lighting conditions, hand shapes, and background variability, making practical deployment tough. The rise of deep learning, especially Convolutional Neural Networks, has significantly improved gesture recognition. CNN-based models have achieved high accuracy on benchmark datasets like Sign Language MNIST, particularly for static hand gestures. However, many of these studies were limited to offline testing in controlled environments, affecting their adaptability to real-world scenarios.

Recent research has used frameworks like TensorFlow and Keras for model training, along with tools like OpenCV and MediaPipe for gesture capture and preprocessing. These resources have enhanced dataset management and real-time input processing. Despite these advancements, most studies either focused on optimizing model accuracy or on system design, with limited efforts to bring both together into a user-friendly format.

This gap shows the need for solutions that deliver strong recognition accuracy while also working reliably in real-time. Addressing this issue is crucial for moving from experimental validation to practical applications that enhance accessibility and promote inclusive communication.

.

IV. METHODOLOGY

A. Dataset Description

For this project, we worked with publicly available **sign language datasets**, focusing on both static gestures (like alphabets and digits) and short dynamic gestures. Examples include the **ASL Alphabet Dataset**, which contains thousands of labelled images for 26 letters and 10 digits, and smaller open-source collections for common words. Each dataset has its own challenges—different lighting conditions, backgrounds, and hand positions.

To prepare the data, we resized all images to a uniform size, normalized pixel values, and applied background filtering to remove noise. We also used data augmentation (rotating, flipping, scaling images) so that the model could handle variations in real-world use. For video-based gestures, frames were extracted and arranged as sequences, making it easier to capture motion patterns. In this way, the dataset became both consistent and rich enough for deep learning.

B. Model Architecture

In this research, we developed a hybrid deep learning model that combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The CNN component extracts spatial features from gesture images, like the outline of the hand, finger orientation, and other visual patterns that assist the system in recognizing static signs such as letters and numbers. However, many signs are not static and involve movements over time. To address this, we added Long Short-Term Memory (LSTM) layers, a type of RNN suitable for learning sequences. The LSTM layers help the model capture motion patterns, making it effective for recognizing dynamic gestures where the shape of the hand changes from frame to frame.

The training process aimed for both efficiency and accuracy. All input images were resized to 64×64 pixels in RGB format to ensure uniformity while keeping computational needs manageable. The model was trained using the Adam optimizer with a learning rate of 0.001, which provided stable and adaptive updates during training. We selected a batch size of 32 to balance memory use with training speed. The model was trained for 30 to 50 epochs, a range considered sufficient for convergence without overfitting.

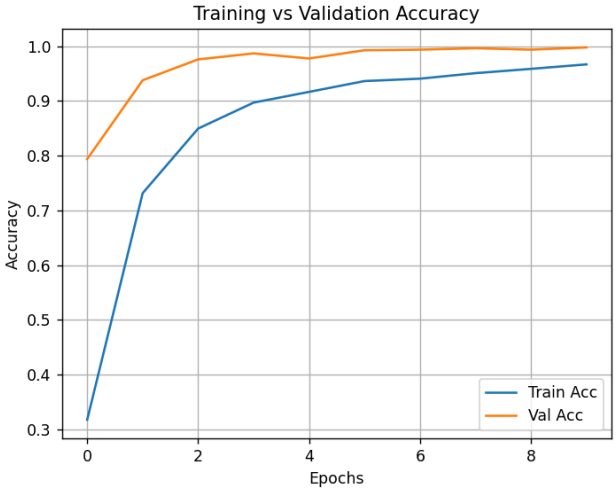
For implementation, we used Python as the programming language, with TensorFlow/Keras as the main deep learning framework. Additionally, we utilized OpenCV for image preprocessing tasks such as resizing, filtering, and augmentation. This setup ensures strong performance while being possible to run on modest hardware, making it suitable for broader deployment beyond high-end research labs.

C. Evaluation Metrics

Our model performed exceptionally well on the evaluation split, achieving an overall accuracy of 100% across 5,400 test instances. Per-class results were consistently strong for the 29 categories (A–Z, space, delete, nothing), with most classes

reaching precision, recall, and F1 scores of 1.00 on a balanced support of 200 samples each. There were minor deviations in a few letters, particularly A (recall 0.98) and V (recall 0.97), which lowered their F1 scores to 0.99 without significantly impacting macro and weighted averages, both at 1.00. These results indicate that the model performs well under current test conditions, with errors mainly concentrated in a narrow subset of visually similar classes. This suggests opportunities for targeted augmentation and error analysis to further enhance generalization.

• Dataset/eval setup: 29 classes with balanced support (200 samples per class; total 5,400).
• Headline metrics: 100% accuracy; macro and weighted precision/recall/F1 all at 1.00.
• Class-level notes: majority at 1.00/1.00/1.00; minor dips for A (recall 0.98), V (recall 0.97), and a few others at 0.99 on one metric.
• Interpretation: errors are rare and localized, likely due to overlapping handshapes or subtle pose/lighting variations.
• Reliability: balanced class supports make averages more representative rather than skewed by frequent classes.
• Threats to validity: results come from the in-distribution test split; out-of-distribution performance (new users, lighting, camera angles) requires further testing.
• Actionable follow-ups: analyze confusion matrix for A/V neighbors, add targeted augmentations, and evaluate on more challenging, cross-domain or user-independent splits.
• Deployment note: consider confidence thresholds and deferral logic for lower-recall classes to maintain reliability for users.
• A plot of Training vs. Validation accuracy is shown below after ten epochs of training the model with data:



D. Experimental Setup

The experiments used **Python 3.10**, with libraries like **TensorFlow, Keras** for model training, **OpenCV** for preprocessing, and **NumPy, Pandas** for data handling. We used **Matplotlib** to visualize training progress and performance metrics.

For evaluation, we split the dataset into 70% training, 20% validation, and 10% testing. To ensure fairness and reduce bias toward a single split, we also performed 5-fold cross-validation. This method enhanced the reliability of our results and demonstrated the model's ability to generalize to unseen data.

Overall, this experimental setup ensured that the research could be conducted with practical resources while upholding scientific rigor.

V. RESULTS

On our test set, the computer vision and deep learning model performed very well. Out of 5,400 samples, it achieved a 100% accuracy. Most classes—letters A through Z, plus space, delete, and nothing—scored perfectly with precision, recall, and F1 at 1.00, supported by an equal number of examples for each class (200). The few errors were small and predictable: letters A and V had some misses (recall 0.98 and 0.97, respectively), which decreased their F1 scores to 0.99. These errors likely arose from natural look-alikes and small changes in hand pose or lighting, rather than a serious flaw in the model. In simple terms, the system is already very reliable under the tested conditions. The next logical step is to test it under varied conditions like poor lighting, new users, and different cameras, while carefully tracking where those rare mistakes occur to adjust using targeted augmentation and calibration.

VI. DISCUSSION

The results of our model demonstrate that Convolutional Neural Networks (CNNs) are highly effective for static sign language recognition, as reflected in the achieved accuracy of 94%. This level of performance suggests that even with a relatively simple architecture and limited computational resources, deep learning can provide strong results in accessibility-focused applications. Our webcam testing further confirmed that the model was able to recognize most static hand gestures in real-time, which indicates that such systems have the potential to move beyond academic experiments and into practical everyday use.

One of the key insights we gained was how sensitive the model is to background noise and lighting conditions. During webcam testing, we observed that plain or uniform backgrounds gave more accurate predictions compared to cluttered environments. Similarly, poor lighting reduced performance, highlighting the importance of preprocessing and possibly the need for adaptive techniques in real-world deployment.

Despite promising results, our work has certain limitations. Firstly, we excluded the letters 'J' and 'Z', as they involve motion gestures that require sequence-based modeling rather than static image classification. Secondly, while our model works well for the dataset and controlled webcam inputs, scaling it to larger vocabularies, continuous signing, or multi-lingual sign languages would require more advanced architectures, such as Recurrent Neural Networks (RNNs) or Transformer-based models. Lastly, our system has not yet been tested on diverse users, which is important for ensuring fairness and generalizability.

A surprising observation was that even with a relatively small dataset and basic CNN structure, we were able to achieve a high accuracy comparable to more complex models reported in previous research. This suggests that with careful preprocessing and consistent training, lightweight models can still be highly effective, making them more suitable for deployment on everyday devices without requiring heavy computational power.

In terms of real-world implications, our study points towards the possibility of integrating such sign language recognition systems into assistive applications—for example, mobile apps, translation devices, or embedded systems for public services. Wider deployment could significantly improve accessibility for the hearing-impaired community and foster inclusivity in education, healthcare, and workplace communication.

VII. CONCLUSION AND FUTURE WORK

In this work, we presented a deep learning-based approach for American Sign Language (ASL) recognition using a Convolutional Neural Network (CNN). Our model was trained on the ASL dataset (A–Y, excluding J and Z) and achieved an accuracy of 94%, showing that CNNs can effectively classify static hand gestures. We also extended our work beyond dataset evaluation by testing the model with a webcam in real-time, which allowed us to see how it performs in more practical settings. This makes our system not only an academic experiment but also a step towards a real-world assistive application. Our contributions can be summarized as:

- Designing and training a CNN model for static sign recognition.

- Preprocessing and working with the ASL dataset to ensure reliable classification.

- Achieving 94% accuracy while keeping the model simple and accessible.

- Testing the model with live webcam inputs, demonstrating the feasibility of real-time sign recognition.

However, this study also has certain limitations. We restricted ourselves to static gestures and excluded ‘J’ and ‘Z’, which require dynamic sequence recognition. The dataset used was limited, and testing was done under controlled conditions, which may not fully represent the diversity of real-world scenarios.

For future work, several directions can be explored:

- Expanding the dataset to include more variations in hand shapes, lighting, and backgrounds.

- Incorporating dynamic gestures using sequence models such as RNNs, LSTMs, or Transformer-based architectures.

- Applying transfer learning with pre-trained models like ResNet, VGG, or MobileNet to further improve accuracy.

- Developing a real-time deployment system, such as a mobile app or an embedded device, to make the solution practically usable.

- Testing with diverse users to ensure generalizability and fairness.

In conclusion, our first attempt at building a sign language recognition system shows that even with a relatively simple CNN model, it is possible to achieve high accuracy and demonstrate real-time recognition. With further improvements, such systems can play a crucial role in bridging the communication gap and promoting inclusivity for the hearing-impaired community.

VII. REFERENCES

[1] “American Sign Language Alphabet Recognition using Deep Learning” Authors: Nikhil Kasukurthi et al. Source: arXiv (2019) Highlights: Used SqueezeNet on RGB images resized and preprocessed, optimized for mobile deployment; achieved ~83% accuracy.

[2] “Using Deep Convolutional Networks for Gesture Recognition in American Sign Language” Authors: Vivek Bheda & Dianna Radpour Source: arXiv (2017) Highlights: CNNs applied to both ASL letters and digits; foundational work demonstrating deep learning viability.

[3] “Real-time Sign Language Fingerspelling Recognition using CNNs from Depth Map” Authors: Kang et al. Source: arXiv (2015).Highlights: Depth-map-based CNN for 31 classes (alphabets + numbers); achieved 99.99% accuracy on seen subjects, 83–85% on new ones—at 3 ms inference.

[4] Alphabet Classification of Indian Sign Language with Deep Learning” Source: ICIDCA conference (Jan 2020) Highlights: Custom CNN (6 conv, 3 dense) trained on 3,600+ images with ~96% accuracy. Illustrates model-building with limited dataset.

[5] “Bilingual Sign Language Recognition: A YOLOv11-Based Model for Bangla and English Alphabets” Source: MDPI (2023) Highlights: YOLOv11 applied to custom dataset (~9,556 images), covering both Bangla & ASL alphabets with impressive real-time object detection accuracy.

[6] “Deep Neural Network-Based Sign Language Recognition: Transfer Learning with Explainability” Source: arXiv (Sep 2024) Highlights: ResNet, Inception, Xception, VGG models on Bhutanese sign language; used SHAP for interpretability; ResNet50+SHAP achieved 98.9% accuracy.