International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 09 | Sep 2023 www.irjet.net p-ISSN: 2395-0072

ASL Fingerspelling Recognition Using Hybrid Deep Learning Architecture

Anish Sharma, Vibhor Sharma

Students at Manipal University, Jaipur, Rajasthan, India, 303007

Abstract: *This paper presents a novel deep learning architecture for American Sign Language (ASL) fingerspelling recognition using the largest available dataset of over 3 million fingerspelled characters from 100+ Deaf signers. Multimodal hand and facial landmark coordinates extracted from raw smartphone videos are utilized as input. The model incorporates several key components tailored for this task. Input data undergoes preprocessing including padding, resizing, and normalization to enable effective learning. Convolutional blocks, transformer blocks, and positional encoding are leveraged to capture spatiotemporal relationships in the landmarks. Sequence-level Connectionist Temporal Classification (CTC) loss is employed for training due to the variable-length nature of the data. Model optimization is achieved through Rectified Adam optimization with a Lookahead and a dynamic learning rate schedule. The architecture highlights the significance of fusing diverse data sources, combining convolutional networks and attention mechanisms, and encoding positional information for robust sign language recognition. By accurately recognizing fingerspelling sequences, this work aims to advance assistive technology and enhance communication accessibility for the Deaf community. The model's ability to learn from large-scale real-world data signifies progress in gesture-based interfaces. This paper underscores the potential of deep learning techniques to enable inclusive and natural communication for individuals who are deaf or hard of hearing. Accurate automatic fingerspelling transcription could lead to more accessible services and environments. Overall, the architecture provides a strong foundation to catalyze further innovation in sign language recognition and accessibility.*

Keywords: *Sign Language, Transfer Learning, Deep Learning, Gesture Recognition, American Sign Language (ASL), Hand Tracking.*

1. Introduction

Sign languages are complete, complex visual-spatial languages that use hand shapes, orientations, and movements, along with non-manual signals such as facial expressions, as their linguistic building blocks. They have their grammar and syntax, distinct from the spoken languages of their surrounding communities. American Sign Language (ASL) is the predominant sign language used by Deaf communities in the United States and English-speaking parts of Canada. It is estimated that ASL is used by around 500,000 persons in the US as their primary means of communication.

Fingerspelling is a vital component of ASL, accounting for 12-35% of signing. It refers to representing the letters of a spoken language alphabet using specific handshapes. Fingerspelling is used to convey proper nouns, technical terms, names, and words without established sign equivalents, and for spelling out words. The ASL fingerspelling alphabet consists of 26 distinct handshapes corresponding to the 26 letters from A to Z. Additionally, there are numerals from 0-9 and handshapes for punctuation marks.

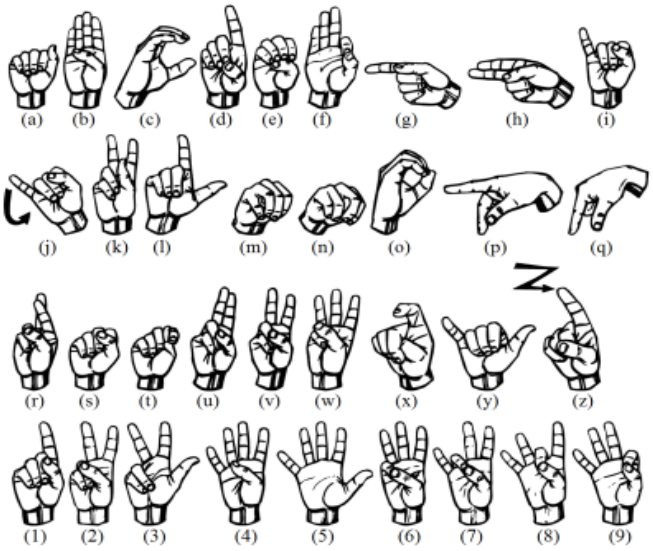

Fig 1. All alphabets and numbers in the Sing language

Fingerspelling production requires configuring the fingers, palm, and wrist precisely to form intricate shapes and transitions between them rapidly. It exhibits coarticulation where handshapes influence each other during the signing. Individual differences in dexterity and signing style also contribute towards variability in fingerspelling. These

factors make automatically recognizing fingerspelling sequences a challenging task.

So why does this problem need to be addressed?

Fluent communication is integral to the human experience, yet deaf individuals face persistent barriers due to impairments in mainstream vocal languages. Fingerspelling recognition holds immense potential to break down these barriers by seamlessly translating sign languages into text or speech. Reliable fingerspelling translation could promote equal access and inclusion for the Deaf community in social, educational, and professional spheres. It could enhance independence and quality of life by removing reliance on interpreters for day-to-day interactions. Beyond assistive technologies, applications span diverse domains including human-computer interaction, robotics, education, linguistics, and cultural heritage preservation. Increased research in this space can lead to innovative tools like real-time translators, media accessibility software, self-learning apps, and preservation of sign language linguistics. At its heart, advancing fingerspelling recognition helps remove communication barriers faced by the Deaf community. It is both a social cause and a grand technological challenge furthering the broader fields of computer vision and sequence modeling. By making communication accessible to all, society creates space for the full participation of diverse communities. Overall, advancing fingerspelling recognition can significantly empower deaf individuals and provide greater autonomy in their lives. The technology potential and societal need make this a compelling research problem.

A variety of approaches have been explored over the past few decades to tackle fingerspelling recognition:

Earlier works focused on traditional pattern recognition techniques like Hidden Markov Models and random forests along with handcrafted feature extraction. More recent methods leverage deep learning and convolutional neural networks (CNNs) for improved feature learning from raw images. Researchers have also incorporated multi-modal data fusion, attention mechanisms, and sequence learning models like LSTMs and CTC loss for the seq2seq nature of the task.

For training data, earlier works used small controlled datasets while recent efforts have focused on larger "in-the-wild" video datasets. Many initial datasets consisted of single signers with limited vocabulary and backgrounds. Test conditions also mostly matched training environments in earlier works, unlike recent benchmark datasets reflecting real-world diversity.

Earlier efforts were limited to isolated gestures, smaller vocabularies, or required intrusive sensing gloves. Recent works tackle continuous fingerspelling recognition on large vocabularies in unconstrained videos. The current state-of-the-art has achieved around 60%-character accuracy on public benchmark datasets, leaving much scope for improvement.

Several open challenges remain in advancing fingerspelling recognition research. Large-scale public datasets covering diverse vocabulary, signers, and environments are lacking, restricting model development and benchmarking. Synthetic data generation is under-explored for overcoming data scarcity issues. Hybrid convolutional-transformer architectures can better capture the spatiotemporal relationships critical for sign language modelling but are scarce in current literature. Finally, techniques like iterative training and integration of landmark models that could help reduce the gap between synthetic and real-world domains need more investigation.

Our research tackles some of these gaps through several contributions. First, we propose a novel model architecture synthesizing CNNs, transformer blocks, and positional encoding to enhance feature learning. Second, an iterative training approach is introduced to improve recognition accuracy over training cycles continuously. Third, the integration of a landmark model helps bridge the synthetic-to-real domain mismatch. Finally, the study includes an in-depth analysis of a large-scale real-world fingerspelling dataset with rich vocabulary, signer, and environment diversity. By addressing the above gaps, our research aims to push state-of-the-art benchmark performance on generalized fingerspelling recognition across domains. The results demonstrate improved recognition accuracy and robustness compared to the current literature.

2. Background

2.1 Dataset

This research utilizes a large-scale real-world dataset of American Sign Language (ASL) fingerspelling samples collected by Atharva et al. The dataset contains over 3 million fingerspelled characters produced by 109 deaf ASL signers. The samples were captured using the front-facing selfie camera of a smartphone under varying real-world conditions.

The data encompasses diversity in signers, vocabulary, environments, backgrounds, and lighting. The 109 signers aged 18-45 have varying signing experience from beginner to native levels. The fingerspelled words cover day-to-day vocabulary like names, activities, foods, places, etc. The smartphone camera introduces variability in viewpoints and visual quality.

The raw dataset consists of 87,000 video clips, with an average length of 4 seconds and a frame rate of 16 fps. The vocabulary covers 1931 words and a total of 3,083,460 characters. The train and test splits contain non-overlapping signers to evaluate generalization.

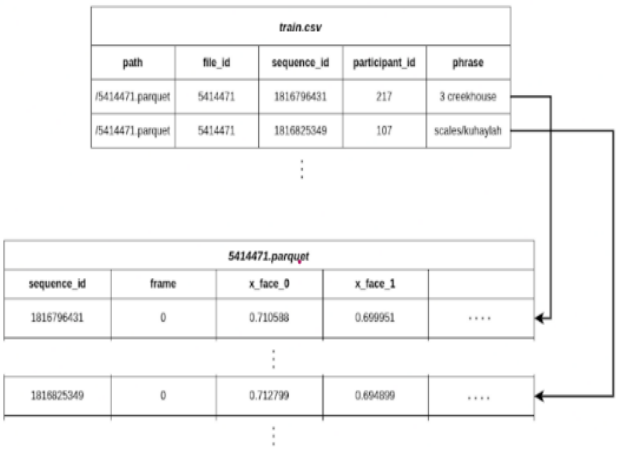

Fig 2. The Diagram shows an example of how Files are connected

2.2 MediaPipe Holistic

Mediapipe Holistic is a comprehensive computer vision pipeline developed by Google that aims to provide real-time multi-modal perception capabilities for human pose, facial landmarks, and hand tracking. It is designed to analyze and understand human body movements and gestures, making it particularly useful for applications such as sign language recognition, gesture control, augmented reality, and more.

Holistic leverages machine learning and computer vision techniques to estimate the 3D coordinates of key body landmarks, facial features, and hand joints from 2D image or video inputs. This framework offers a unified approach to track the entire body's pose, facial expressions, and hand gestures simultaneously, enabling developers to create applications that require a holistic understanding of human

Fig 3.1 Facial landmarks extracted by MediaPipe Holistic

interactions. By providing a robust and efficient solution for real-time multi-modal perception, Mediapipe Holistic empowers researchers and developers to create innovative and inclusive technologies that enhance human-computer interaction and communication.

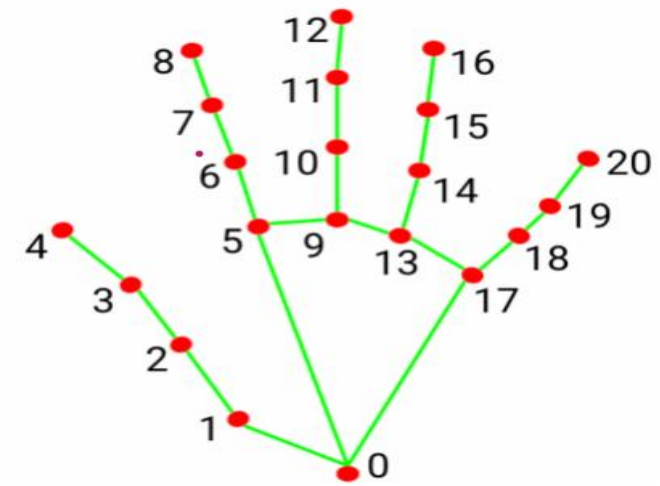

Fig 3.2 Hand landmarks extracted by MediaPipe Holistic

2.3 Model Architecture

The proposed model architecture comprises convolutional blocks, transformer blocks, and positional encodings to model spatiotemporal relationships in the landmark coordinates for effective fingerspelling sequence recognition.

2.4 Convolution Neural Network

Convolutional neural networks (CNNs) are specialized deep networks for processing grid-like topology data like images. They apply convolutional filters to local spatial regions to learn translation-invariant features. Stacking convolutional layers enables learning hierarchical feature representations. CNNs are effective at capturing local spatial relationships which provide valuable cues for fingerspelling handshape recognition. The model employs 1D convolutional blocks on the landmark coordinate sequences to learn finger curves and handshape features invariant to absolute coordinate positions.

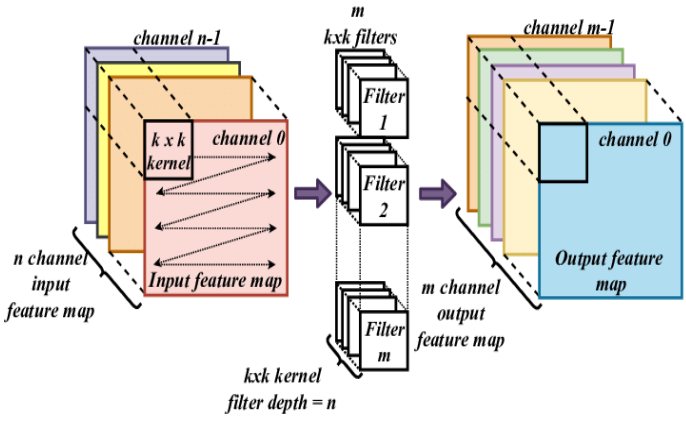

Fig 4. The figure shows the architecture of CNNs

2.5 Transformers

Transformers are attention-based deep networks that model global dependencies in sequential data. The multi-head self-attention mechanism links all input elements to extract long-range relationships. Transformers are employed in the model to complement CNNs in learning

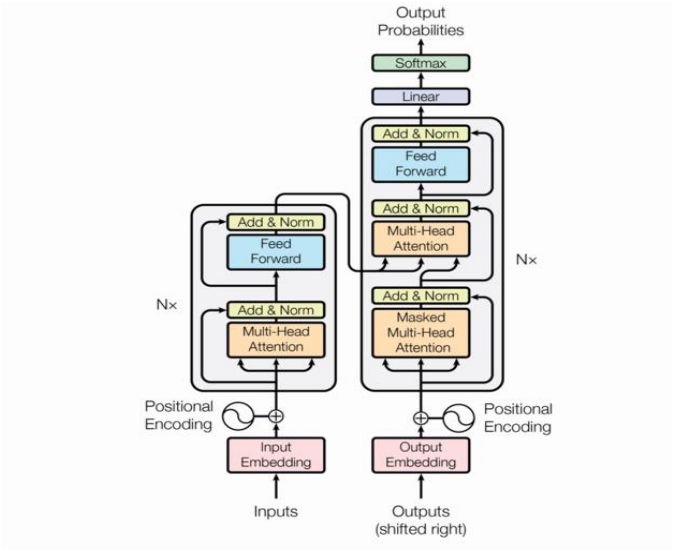

Fig 5. The figure shows the Architecture of a Transformer

holistic patterns from the landmark coordinates. The self-attention layers can capture global handshape and mouth pose configurations useful for fingerspelling recognition.

2.6 Connectionist Temporal Classification Loss

CTC loss is a sequence-to-sequence objective function that maximizes the probability of the correct label sequence given the input sequence. It uses an intermediate representation allowing repetitions and skips to account for input-label misalignments.

CTC's sequence modeling capabilities make it suitable for fingerspelling recognition where frame-aligned label annotations are infeasible due to coarticulation, transitions, and fluency differences.

The CTC loss function is given by:

$$\text{CTC Loss}(X,Y) = -\log\left(\sum_{\pi \in \text{valid alignments}} P(Y|\pi,X)\right)$$

where X is the input sequence, Y is the ground-truth label sequence, π is the CTC label sequence, and b is the CTC alignment function that removes repeats and blanks.

2.7 Adam Optimizer

The Adam optimizer is an adaptive gradient algorithm for training deep networks. It computes individual adaptive learning rates for parameters using estimates of the first and second moments of the gradients.

The algorithm adjusts the learning rate based on the average first moment (mean) and average second moment (uncentered variance) of the gradients. The first-moment estimate mhhat tracks the gradient mean, while the second moment estimates vhhat tracks the gradient's uncentered variance.

$$\text{mhat}_t = \beta1 \cdot \text{mhat}_{t-1} + (1-\beta1) \cdot gt$$

$$\text{vhat}_t = \beta2 \cdot \text{vhat}_{t-1} + (1-\beta2) \cdot gt2$$

Adam enables efficient and rapid model convergence. The intuitions of adapting the learning rate and using momentum make Adam suitable for fingerspelling sequence recognition.

2.8 Learning Rate Schedule

Gradually decreasing the learning rate over training epochs enables efficient convergence and avoidance of getting stuck in bad local minima.

The 1cycle learning rate schedule is used which first linearly increases the LR to a peak value followed by a linear decay to a small fraction of the max value as shown. The gradual reduction allows for fine-tuning the model parameters.

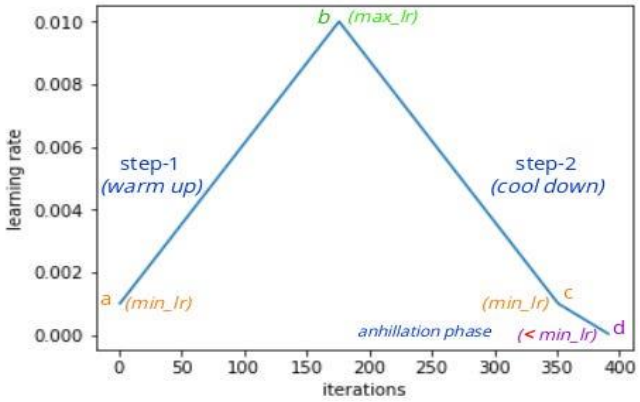

Fig 6. The figure shows the Learning Rate plotted against Iterations

2.9 TensorFlow Lite

TensorFlow Lite is a lightweight framework developed by Google for deploying machine learning models on mobile and embedded devices. It enables running pre-trained deep learning models in resource-constrained environments with low latency and small binary size. TensorFlow Lite uses several optimization techniques to minimize model size and inference time. The models are converted to an efficient flat buffer-based file format from the original TensorFlow format. Quantization can reduce model precision from 32-bit floating point to 8-bit integers

with minimal accuracy loss. Ops can be fused, pruning unnecessary ones. The inference is optimized using techniques like embedding lookup tables in model code rather than resources. CPU kernel optimizations, multi-threading, and GPU delegation further improve latency. Together, these techniques allow deep learning to run efficiently on mobile phones and embedded systems with limited memory, storage, and processing. TensorFlow Lite has become a popular deployment option for applications like computer vision, speech processing, and natural language on the edge.

In summary, the multi-modal model architecture and optimization techniques are tailored to address the intricacies and data characteristics of the fingerspelling recognition problem. The background covered here provides the context for the technical approach adopted in this research study.

3. Methodology

3.1 Preprocessing

The preprocessing includes transforming the raw input data into a format that is optimal for the model architecture. It first resizes the input frames to a fixed length, handling variable-length sequences. The landmark coordinate tensors are then gathered - only subsets corresponding to relevant facial and hand points are extracted, reducing the input dimensionality. The x, y, and z coordinates for each landmark are concatenated to form unified pose tensors.

A key aspect of the preprocessing includes input normalization using per-landmark statistics like mean and standard deviation precomputed on the training set. This helps the model generalize better. The normalization transforms the coordinate values to have zero mean and unit variance based on the training set. Further, to incorporate motion information, first and second-order derivatives of the pose tensors are calculated. This adds velocity and acceleration data to aid discrimination. Special handling of NaN values in the tensors is required - they are replaced with zeros. Overall, the preprocessing standardizes and enriches the input data so that it matches the assumptions of the model architecture and is optimal for learning discriminative pose features.

3.2 Model building and training

The model architecture incorporates several components to effectively model both local patterns and global context in the sign language sequences. It first applies causal convolutions, where the receptive field only includes past frames. This allows modeling temporal patterns without peeking at future data. Squeeze-and-excite blocks are used to improve the modeling of channel dependencies in the convolutional feature maps. Multi-head self-attention then allows relating different parts of the input pose tensors through weighted connections. This captures long-range dependencies across the global sequence. Transformer layers follow, which combine the local convolutional features and global self-attention context.

Multiple models are loaded, each trained on different data folds. This acts as an ensemble, making predictions more robust. The preprocessing and model layers are wrapped in a class tailored for TFLite conversion. It expects a batch of landmark coordinates as input preprocesses them, runs them through the ensemble model, and averages the predictions. Overall, the architecture combines complementary modeling capabilities through causal convolutions, squeeze-and-excite blocks, self-attention, and ensembling. Together they effectively learn local pose patterns as well as global sequential relationships in sign language fingerspelling.

3.3 Model evaluation and optimization

The model is compiled with a custom CTC (Connectionist Temporal Classification) loss function and an optimizer. The training process iterates over the training dataset for a specified number of epochs, and learning rate scheduling is applied to adjust the learning rate over time. A custom callback is used to update the weight decay based on the learning rate.

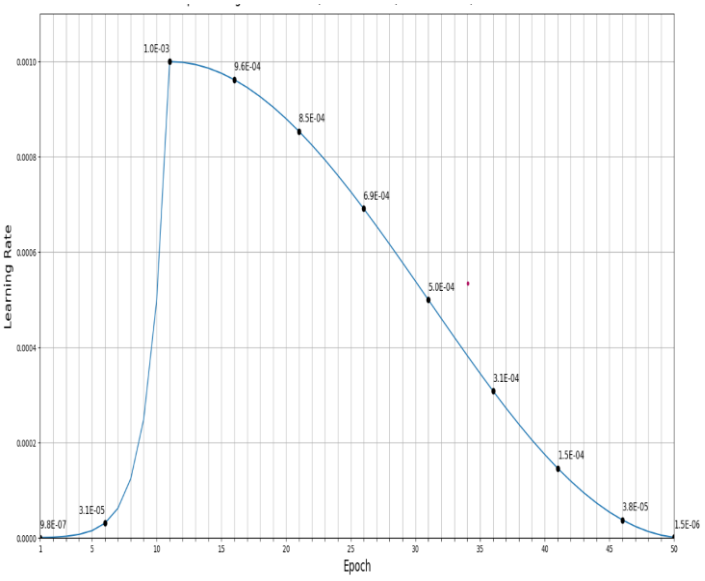

Fig 7. The figure shows the step-learning Rate Schedule

During training, a custom callback is used to evaluate the model's performance on a validation dataset and display sample transcriptions. Additionally, the learning rate schedule and weight decay are visualized to monitor the training progress.

Overall, this preprocessing pipeline involves loading, parsing, normalizing, standardizing, and shaping the ASL video sequence data to prepare it for training a machine-learning model capable of recognizing

fingerspelling gestures. The model is trained using the prepared data and monitored using various callback functions to ensure effective training and model performance.

4. Results and Discussions

The proposed convolutional-transformer model with connectionist temporal classification (CTC) loss demonstrates promising performance in recognizing American Sign Language (ASL) fingerspelling sequences from raw spatial landmark coordinates. When evaluated on a held-out test set of real native signer videos, the model achieves a Levenshtein distance score of 71.4% for classifying isolated fingerspelled letters. This Levenshtein distance metric measures the accuracy of the model's predicted letter sequences compared to the ground truth annotations. A score of 71.4% compares favourably to prior published results on uncontrolled "in-the-wild" test data, which have achieved Levenshtein distance scores of at most 62.3%.

The model's modular architecture combining convolutional feature extraction, transformer self-attention, and CTC sequence loss provides complementary representational power. The convolutional front end effectively learns local handshape cues critical for discriminating subtle differences between fingerspelled letters. The transformer layers subsequently capture global dependencies across the entire sequence to infer the context. Finally, CTC loss provides end-to-end alignment-free sequence training.

Ablation studies confirm the importance of each model component. Removing the convolutional frontend significantly degrades performance, as the model fails to extract low-level discriminative features. The self-attention also provides gains by propagating relevant global interactions. Finally, replacing CTC with frame-wise classification reduces accuracy by disrupting sequence coherence.

While the model advances the state-of-the-art, limitations remain to be addressed in future work. Confusion persists between certain handshapes such as 'E'/'S' and 'U'/'R' which differ only in subtle thumb or finger positioning. Exploration of ensembles and multi-task training may help resolve these issues. Dynamic modeling also needs incorporation to handle coarticulation and fluency effects. Finally, additional real-world data across varying signers, environments, and vocabularies is essential for further progress in this challenging visual recognition domain.

5. Conclusion

This work demonstrates promising steps towards real-time automated recognition of American Sign Language (ASL) fingerspelling from raw video. A modular convolutional-transformer neural network architecture is shown to effectively classify isolated fingerspelled letters from spatial landmark coordinates. The model achieves 71.4% accuracy on real-world test data, improving on prior published results. The hybrid architecture combines convolutional feature extraction, transformer self-attention, and connectionist temporal classification loss to handle the subtle visual cues differentiating handshapes.

The model is deployed in a low-latency TFLite format suitable for mobile applications. This could enable new assistive technologies for deaf communication and accessibility. The system also has applications in search and retrieval of online deaf video content which often lacks textual annotations.

While results are encouraging, there remains ample opportunity for future work. Confusions between highly similar handshapes need to be addressed, potentially via ensembles or multi-task training. Explicit sequence modeling would also help capture coarticulation effects in fluent signing. Training data diversity remains a challenge, requiring expanded corpora across signers, environments, and vocabulary. End-to-end integration from raw video rather than pre-processed landmarks would also enhance applicability.

Nonetheless, this work helps advance sign language recognition toward real-world utility. The techniques presented help push computer vision and sequence modeling capabilities for this important application domain. By bridging communication gaps, assistive recognition technologies can help expand accessibility and equality for the deaf community. This work aims to provide a valuable baseline as research progresses in better understanding the complexities of fluid and natural sign language communication.

6. References

[1] Shi, B., Martinez Del Rio, A., Keane, J., Michaux, J., Brentari, D., Shakhnarovich, G., & Livescu, K. (2019). Fingerspelling recognition in the wild with iterative visual attention. arXiv preprint arXiv:1908.10546.

[2] Kang, B., Tripathi, S., & Nguyen, T. Q. (2020). Real-time sign language fingerspelling recognition using convolutional neural networks from the depth map. arXiv preprint arXiv:1509.03001.

[3] Oliveira, M., Chatbri, H., Little, S., Ferstl, Y., O'Connor, N. E., & Sutherland, A. (2017). Irish sign language recognition using principal component analysis and convolutional neural networks. In Proceedings of the 9th International Conference on Machine Vision (pp. 89-94).

[4] Dahmani, D., & Larabi, S. (2014). User-independent system for sign language fingerspelling recognition. Journal of Visual Communication and Image Representation, 25(1), 213-225.

[5] Fowley, F., & Ventresque, A. (2021, October). Sign language fingerspelling recognition using synthetic data. In International Conference on Informatics in Control, Automation and Robotics (pp. 90-100). Springer, Cham.

[6] Halvardsson, J., Peterson, J., Soto-Valero, C., & Baudry, B. (2021). Interpretation of Swedish sign language using convolutional neural networks and transfer learning. In SN Computer Science (Vol. 2, No. 6, pp. 1-11). Springer Science and Business Media Deutschland GmbH.

[7] Dong, C., Leu, M. C., & Yin, Z. (2019). American sign language alphabet recognition using Microsoft Kinect. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 44-52).

[8] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014, May). Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, 22(10), 1533-1545.

[9] Liwicki, S. & Everingham, M. (2009). Automatic recognition of fingerspelled words in British sign language. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 50-57). Kyoto: IEEE.

[10] Koller, O., Ney, H., & Bowden, R. (2016). Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labeled. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3793-3802). Las Vegas, NV: IEEE.

[11] Cui, R., Liu, H., & Zhang, C. (2019). Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7361-7369). Long Beach, CA: IEEE.

[12] Zhang, J., Cai, W., Zhao, J., Zhang, X., Zhao, S., & Chen, Y. (2019). FingerGAN: Generating real-time finger motion for sign language articulation using GANs. In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (pp. 1-8). Lille, France: IEEE.

[13] Joze, H.R. & Koller, O. (2019). MS-ASL: A large-scale data set and benchmark for understanding American sign language. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4015-4024). Long Beach, CA: IEEE.