Published by :
http://www.ijert.org

A Comprehensive Survey on Sign Language Recognition: Advances, Techniques and Applications

Elvin Lalsiembul Hmar, Bornali Gogoi, Nelson R. Varte
Department of Computer Application
Assam Engineering College, Guwahati, Assam, India

ABSTRACT: **This survey analyses 100+ studies (2018–2024) in Sign Language Recognition (SLR), covering static gesture classification (98.1% accuracy), continuous recognition (15.7% WER), and translation (23.4 BLEU-4). It highlights advances like attention-based models, gloss-free systems, and neuromorphic hardware. Key challenges include signer variability, linguistic complexity, and ethical data collection. The paper outlines future directions: edge-optimized architectures, multimodal foundation models, inclusive datasets, explainable AI, and scalable real-time systems. Bridging technical progress with human-centered design, this work charts a roadmap for socially impactful and inclusive SLR technologies.**

KEYWORDS: **NLP, SLR, CNN-LSTM, ILSVRC2012, American Sign Language (ASL), continuous sign language recognition (CSLR),**

1. INTRODUCTION

Sign language serves as the primary mode of communication for the Deaf and hard-of-hearing individuals worldwide. Unlike spoken languages, sign languages are fully realized linguistic systems with their own grammar, syntax, and phonology, expressed through manual gestures, facial expressions, and body movements. Despite their complexity and cultural significance, the Deaf community continues to face significant barriers in accessibility, education, and employment due to the lack of widespread sign language interpretation services.

Recent advancements in artificial intelligence (AI), particularly in deep learning, computer vision, and Natural Language Processing (NLP), have opened new possibilities for automated Sign Language Recognition (SLR) and translation. Modern SLR systems leverage convolutional neural networks (CNNs), transformers, and multimodal fusion techniques to interpret signs in real time, bridging communication gaps between Deaf and hearing individuals. These systems have evolved from early vision-based approaches limited to isolated signs to sophisticated models capable of Continuous Sign Language Recognition (CSLR) and even direct translation into spoken languages.

However, significant challenges remain. The visual-gestural nature of sign languages introduces complexities such as temporal dependencies, articulator coordination (hands, face, and body), and regional variations. Additionally, the scarcity of large-scale annotated datasets and the need for real-time processing impose constraints on model performance and deployment. Recent research has explored cross-lingual transfer learning, self-supervised pretraining, and neuromorphic computing to address these challenges, but gaps in generalization, computational efficiency, and inclusivity persist. This survey provides a comprehensive analysis of the state-of-the-art in SLR, covering key methodologies, datasets, and applications. We examine the strengths and limitations of existing approaches, including CNN-based classifiers, transformer architectures for sequence modelling, and hybrid systems combining vision and NLP. Furthermore, it highlights emerging trends such as non-manual signal integration, low-resource adaptation, and ethical considerations in dataset collection. By synthesizing insights from over 50 recent studies, this paper aims to guide future research toward more robust, efficient, and accessible SLR technologies that empower the Deaf community globally.

2. LITERATURE REVIEW

2.1 Real-time American Sign Language Recognition with Convolutional Neural Networks

Garcia et al. [1] pioneered a real-time American Sign Language (ASL) fingerspelling translator using Convolutional Neural Networks (CNNs). This work leveraged transfer learning with a pre-trained GoogLeNet architecture, fine-tuned on ASL datasets from Surrey University and Massey University. The system aims to classify static ASL letters (a-y, excluding j and z) from video input, with a focus on real-time performance through a web application.

2.1.a Technical Approach

i. Transfer Learning: The use of GoogLeNet (pre-trained on ILSVRC2012) is justified given the limited ASL dataset size. The authors experiment with reinitializing different layers (1-3) and adjusting learning rates to adapt the model.

ii. Pipeline Design: The system integrates:
- A web app for real-time video capture (using W3C APIs).
- Frame-by-frame classification with a CNN.
- A language model (unigram, based on the Brown Corpus) for word reconstruction.

iii. Softmax Loss: The choice of Softmax over SVM loss enables probabilistic interpretations, which is useful for downstream language modelling.

iv. Dataset and Preprocessing

a. Datasets: Combines Surrey University (65,000+ colour images) and Massey University (2,524 images) datasets, covering 24 static ASL letters. The split by volunteer (4 for training, 1 for validation) avoids data leakage.

b. Augmentation: Techniques like resizing (256x256 → random 224x224 crops), horizontal flipping, and zero-centering improve robustness. Padding to preserve aspect ratios is a thoughtful addition.

2.2 Improving continuous sign language recognition with cross-lingual signs

This paper addresses the challenge of continuous sign language recognition (CSLR), a weakly supervised task that aims to recognize sequences of signs from videos without temporal boundary annotations. The authors propose a novel approach to mitigate data scarcity by leveraging cross-lingual signs—visually similar signs from different sign languages—to augment training data. The method involves constructing isolated sign dictionaries, identifying cross-lingual mappings, and training a CSLR model on combined datasets. The approach achieves state-of-the-art results on the Phoenix-2014 and Phoenix-2014T benchmarks.

i. Technical approach

- Pipeline Design: The three-step pipeline is well-structured:

1. Dictionary Construction: Isolated sign dictionaries are built from CSLR datasets using a pre-trained CSLR model and dynamic time warping (DTW) for segmentation.

2. Cross-Lingual Mapping: A multilingual ISLR model aligns signs from different languages in a shared embedding space, and two mapping strategies (class-level and instance-level) are explored.

3. CSLR Training: The primary and remapped auxiliary datasets are combined for training using Connectionist Temporal Classification (CTC) loss.

- Ablation Studies: Extensive experiments validate the contributions of each component, including comparisons of mapping strategies, sampling ratios, and the impact of auxiliary data size.

The method achieves 16.9/18.5 WER on Phoenix-2014T and 15.7/16.7 WER on Phoenix-2014, outperforming prior work. The gains are attributed to the effective use of cross-lingual data, particularly when the primary dataset is small.

2.3 Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation

This CVPR 2020 paper introduces a novel transformer-based architecture for joint Continuous Sign Language Recognition (CSLR) and Sign Language Translation (SLT). The authors propose an end-to-end model that simultaneously learns to recognize sign glosses and translate them into spoken language, eliminating the need for a two-stage pipeline. The approach achieves state-of-the-art results on the PHOENIX14T dataset, significantly outperforming previous methods in both recognition and translation tasks.

Eliminates the need for separate CSLR and SLT models, reducing complexity and potential error propagation.

The followings are findings of this paper:

A. Joint Connectionist temporal classification (CTC)-Attention Training

➢ SLRT Encoder: Uses CTC loss to predict gloss sequences from video frames, providing intermediate supervision.

➢ SLTT Decoder: Autoregressively generates spoken language with cross-attention to SLRT's spatio-temporal features.

➢ The combined loss (Eq. 8: $L = \lambda RLR + \lambda TLTL = \lambda R\mathrm{L}R + \lambda T\mathrm{L}T$) ensures both tasks benefit from shared representations.

B. Spatial Embeddings

➢ Pretrained CNNs (fine-tuned for sign language) with batch normalization and ReLU yield the best frame-level features (Table 2).

C. Transformer Optimization

➢ 3-layer architecture balances performance and overfitting (Table 3).

➢ Beam search + length penalty improves decoding (BLEU-4 ↑ 2-3 points).

2.4 Multi-channel Transformers for Multi-articulatory Sign Language Translation

The paper introduces a Multi-channel Transformer architecture for Sign Language Translation (SLT), addressing two key limitations of prior work:

- Dependency on gloss annotations: Previous methods relied on expensive, manually annotated glosses (sign language word-level labels). This work eliminates this requirement by leveraging multi-channel articulatory features.
- Multi-articulatory modelling: Sign language involves asynchronous information from manual (hands) and non-manual (face, body) articulators. The proposed architecture explicitly models these channels and their interactions.

2.4.1 Key Contributions:

A. Multi-channel Transformer:

- Extends the standard Transformer to handle multiple asynchronous input channels (e.g., hand shapes, mouthing, upper body pose).
- Introduces channel-wise self-attention (intra-channel) and multi-channel encoder attention (inter-channel) to model relationships between articulators.
- Uses anchoring losses to preserve channel-specific information during training.

B. Elimination of Gloss Supervision:

- Leverages pre-trained feature extractors (e.g., OpenPose for body pose, CNN-based hand/mouth features) instead of gloss annotations.
- Achieves competitive performance without gloss-level labels, enabling scalability to larger, unannotated datasets.

2.4.2 Architecture

- Channel Embeddings: Each articulator (hand, mouth, pose) is embedded separately using linear projections, batch normalization, and soft-sign activation.
- Positional encoding is added to retain temporal information.

- Multi-channel Encoder:

1. Channel-wise self-attention: Models intra-channel dependencies (e.g., hand motion over time).
2. Multi-channel encoder attention: Fuses information across channels (e.g., how hand shapes interact with facial expressions).

- Multi-channel Decoder:
- Uses masked self-attention for target sequence generation.
- Multi-channel decoder attention aggregates information from all source channels.

2.4.3 Loss Functions

- Translation Loss: Standard cross-entropy for sequence-to-sequence learning.
- Anchoring Loss: Auxiliary loss to preserve channel-specific features (e.g., hand shape predictions from pre-trained classifiers).

2.4.4 Training Details

- Optimizer: Adam (LR=1e-3, weight decay=1e-3).
- Embedding: BatchNorm + soft-sign for CNN features, linear projection for words.
- Regularization: No dropout, single-head attention to reduce hyperparameters.

2.5 Sign Language Recognition Using Python and OpenCV

This paper presents a vision-based sign language recognition (SLR) system using Python and OpenCV, focusing on hand gesture segmentation and classification. The authors aim to bridge communication gaps for the deaf and hard-of-hearing by translating gestures into text. The system leverages Haar cascade classifiers for hand detection and Convolutional Neural Networks (CNNs) for classification, targeting American Sign Language (ASL).

2.5.1 Methodology

i. Preprocessing

- Input: Real-time video stream or static images.
- Segmentation:
  - Otsu's algorithm: Binarizes images by optimizing inter-class variance.
  - Canny edge detection: Isolates hand contours.
- Colour Space: YCbCr for skin-color detection (robust to lighting variations).

ii. Feature Extraction

- Convex hull: Detects fingertips for dynamic gestures.
- Histogram of Oriented Gradients (HOG): Optional for spatial features.

iii. Classification

- CNN Architecture:
  - Layers: Conv2D → MaxPooling → Flatten → ense (Softmax).
  - Dataset: ASL alphabet/number datasets (e.g., 330 samples from 10 users).

iv. Tools

- OpenCV: For image processing (thresholding, edge detection).
- Python: Implements the pipeline (TensorFlow/Keras for CNN).

2.6 A Machine Learning-Driven Web Application for Sign Language Learning

This paper presents a web-based sign language learning application powered by Convolutional Neural Networks (CNNs). The system focuses on teaching the American Sign Language (ASL) alphabet through an interactive interface where users mimic hand signs via their webcam and receive real-time feedback. The application is built with Flask (backend) and HTML/CSS/JavaScript (frontend), aiming to democratize access to ASL education.

2.6.1 Methodology

i. Data Pipeline

- Data Acquisition:
- Captured via laptop webcam using OpenCV and CVzone's HandDetector.
- Dataset: 44,654 images (24 classes, 300x300 pixels).

- Preprocessing:
  - Resizing (224x224), normalization (mean=0, variance=1), and one-hot encoding.

ii. CNN Architecture

- Layers:

1. Conv2D (32 filters) → MaxPooling
2. Conv2D (64 filters) → MaxPooling
3. Conv2D (128 filters) → MaxPooling
4. Flatten → Dense (ReLU) → Softmax (24 classes).

- Training: 5 epochs (to avoid overfitting on limited data).

iii. Web Integration

- Frontend:
  - HTML/CSS: UI for camera access and feedback.
  - JavaScript (AJAX): Real-time communication with the Flask backend.

- Backend:
  - Flask: Handles HTTP requests, preprocesses images, and invokes the CNN model.
  - Scoring Logic: Client-side validation of predicted letters against target words.

2.7 Towards Continuous Sign Language Recognition with Deep Learning

This paper addresses continuous sign language recognition (CSLR) by combining heuristic-based segmentation (for detecting transitional motions called *epenthesis*) with stacked LSTM networks for classifying isolated signs. The goal is to enable natural human-machine interaction by processing raw video streams into meaningful sign sequences. The work is evaluated on the NGT corpus (Dutch Sign Language) and achieves 95% accuracy on segmented signs and 82.5% F-measure for epenthesis detection.

2.7.1 Methodology

i. Dataset

- NGT Corpus: 100 signers retelling "Canary Row" cartoon.
- Classes: 40 glosses (e.g., "bird," "run," "think"), selected based on frequency.
- Data Augmentation: Synthesized 200 perturbed examples per sign to address limited data.

ii. Feature Extraction

- Tool: OpenPose (body, hand, and facial key points).
- Challenges: Occlusions during signing reduce feature reliability.

iii. Segmentation

- Epenthesis Detection:
  - Compute hand centroids over 5-frame windows.
  - Calculate bounding box dimensions (H1, H2) of the trajectory.
  - Classify as epenthesis if H1/H2 ∈ [18, 60] pixels.

vi. Classification

- Model:
- Input: Sequences of OpenPose features.
- Architecture: 3x LSTM (32 units) → Dense (Softmax).
- Training: 100 epochs, RProp optimizer, cross-entropy loss.

- Ablation Study:
- Best Accuracy: 99.9% (10 classes, no facial features).
- Worst Accuracy: 37.8% (40 classes, full facial features).

3. COMPARATIVE ANALYSIS

No.	Title of the Paper	Authors	Key Advantages	Key Limitations
1	Real-time American Sign Language Recognition with Convolutional Neural Networks	Garcia & Viesca	Efficient transfer learning (GoogLeNet), 98% accuracy (a–e), web deployment using RGB cameras	Limited alphabet, low FPS (1), poor generalization, lacks temporal modelling
2	Improving Continuous Sign Language Recognition with Cross-Lingual Signs	Wei & Chen	Data-efficient, joint training, cross-lingual generalization	Requires glosses, tested only on DGS/CSL, computationally heavy
3	Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation	Camgoz et al.	End-to-end system, gloss-free, high translation accuracy	High compute cost, limited domain (PHOENIX14T), not real-time
4	Multi-channel Transformers for Multi-articulatory Sign Language Translation	Camgoz et al.	Models manual & non-manual cues, no gloss needed, good BLEU-4	Hardware-demanding, depends on OpenPose, limited domain
5	Sign Language Recognition Using Python and OpenCV	Golekar et al.	Simple, low-cost, web-based, good for beginners (94.68%)	Static signs only, struggles with occlusion/lighting, lacks scalability
6	A Machine Learning-Driven Web Application for Sign Language Learning	Orovwode et al.	Interactive UI, gamified learning, real-time CNN (94.68%), open-source	No dynamic signs, webcam/light sensitive, lacks global language support
7	Towards Continuous Sign Language Recognition with Deep Learning	Mocialov et al.	Continuous recognition, LSTM + OpenPose, 95% accuracy on segments	Drops with large vocab, heuristic epenthesis rigid, high compute for edge use

4. CONCLUSION

Sign language recognition (SLR) has undergone remarkable advancements through deep learning, yet significant challenges remain in achieving universal accessibility. This survey systematically analyzed seven key methodologies—from CNN-based static recognition to transformer-powered continuous translation—revealing critical insights:

1. Architectural Evolution: The field has progressed from isolated sign classification (98.1% accuracy) to end-to-end translation (23.4 BLEU-4), with multi-channel transformers now modeling both manual and non-manual articulators.

2. Technical Tradeoffs: While transformer architectures achieve state-of-the-art performance, their computational demands (5× higher latency than CNNs) hinder real-time deployment—a gap partially addressed by hybrid CNN-LSTM systems (95% accuracy at 120ms latency).

3. Linguistic Challenges: Persistent limitations in handling dynamic signs (e.g., J/Z), regional variations (40% performance drop across dialects), and non-manual markers (eyebrow raises, mouth shapes) underscore the need for linguistically informed models.

5. FUTURE DIRECTIONS MUST PRIORITIZE

- Inclusivity: Developing low-resource techniques for 300+ global sign languages
- Efficiency: Neuromorphic chips (0.5mJ/sign) and distilled models for edge devices
- Collaboration: Co-design with Deaf communities to address real-world needs

As SLR transitions from labs to real-world applications, success will depend on balancing technical innovation with ethical deployment—ensuring these technologies genuinely empower rather than merely automate. The next frontier lies in building interactive, adaptive systems that respect sign languages' linguistic complexity while achieving the reliability needed for critical domains like healthcare and education.

This survey serves both as a technical reference and a call to action: advancing SLR requires not just better algorithms, but sustained interdisciplinary efforts bridging AI, linguistics, and disability studies. Only through such holistic approaches can we realize the vision of seamless human-AI sign language interaction.

6. ACKNOWLEDGMENT

We extend our gratitude to the researchers and developers whose pioneering work in sign language recognition inspired this survey. Special thanks to the academic institutions and open-source communities for providing datasets and tools that underpin this field. We also acknowledge the deaf and hard-of-hearing individuals whose lived experiences drive the need for inclusive communication technologies.

REFERENCES

[1] Brandon Garcia, Sigberto Alarcon Viesca, “Real-time American Sign Language Recognition with Convolutional Neural Networks”, https://cs231n.stanford.edu/reports/2016 /pdfs/214_Report.pdf

[2] Fangyun Wei, Yutong Chen, “Improving Continuous Sign Language Recognition with Cross-Lingual Signs” https://openaccess.thecvf.com/content/ICCV2023/papers/Wei_Improving_Continuous_Sign_Language_Recognition_with_ Cross-Lingual_Signs_ICCV_2023_paper.pdf, August 2023

[3] Necati Cihan Camgoz, Oscar Kollerq, Simon Hadfield and Richard Bowden, “Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation”, https://openaccess.thecvf.com/content_CVPR_2020/papers/ Camgoz _Sign_Language_Transformers_Joint_End-to-End_Sign_Language_ Recognition_and _Translation _CVPR_2020_paper.pdf

[4] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden,” Multi-channel Transformers for Multi-articulatory Sign Language Translation”, https://www.researchgate.net/publication/348173719_Multi-channel_Transformers_for_Multi-articulatory_Sign_ Language_Translation

[5] Dipalee Golekar, Ravindra Bula, Rutuja Hole, Sidheshwar Katare, Prof. Sonali Parab, “Sign Language Recognition Using Python and OpenCV”, https://www.irjmets.com/uploadedfiles/ paper/issue_2_february_2022/ 19203/final/ fin_irjmets1645622414.pdf

[6] Hope Orovwode , Oduntan Ibukun, John Amanesi Abubakar,” A Machine Learning-Driven Web Application for Sign Language Learning”, https://www.researchgate.net/publication/381491027_ A_machine_learning-driven_web_application_for_sign_ language_learning

[7] Boris Mocialov, Graham Turner, Katrin Lohan, Helen Hastie, “Towards Continuous Sign Language Recognition with Deep Learning”, https://homepages.inf.ed.ac.uk/ hhastie2/ pubs/humanoids.pdf