WholeBodyPose: A Unified End-to-End Framework for Sign Language Recognition and Pose-Based Training Data

Cristian Lazo-Quispe[a], Renato Castro-Cruz[a], Mauricio Salazar-Espinosa[b], and Gissella Bejarano[c]

[a]Universidad Nacional de Ingeniería, Lima, Peru
[b]Pontificia Universidad Católica del Perú, Lima, Peru
[c]Marist University, New York, United States

ABSTRACT

Sign language recognition systems require high-quality, temporally consistent pose estimation data for effective training. However, existing approaches suffer from inconsistent output formats across different pose estimation models, inadequate temporal filtering, and a lack of standardized preprocessing pipelines. We present WholeBodyPose, an end-to-end framework that unifies multiple state-of-the-art pose estimation models (MediaPipe, RTMPose, ViTPose) under a standardized COCO-133 format. Our framework addresses the critical gaps identified in recent work, providing researchers with a consistent interface to generate high-quality training data. We provide a modular architecture that supports the easy integration of new pose estimation models, including preprocessing pipelines from raw video input to training-ready keypoint sequences, quality control mechanisms, and visualization tools. Additionally, our library is optimized for real-time use cases, enabling seamless integration into live inference pipelines and interactive sign language systems. This capability makes it suitable not only for research and training, but also for deployment in production-level applications. Our open-source library democratizes access to high-quality pose-based sign language recognition research and establishes a new standard for preprocessing pipelines in this domain.

Keywords: Sign Language Recognition, Pose Estimation, Framework, Keypoint Filtering, Real Time Processing

1. INTRODUCTION

Sign language recognition (SLR) is a critical application of computer vision with direct implications for accessibility and human-computer interaction. Recent advances in whole-body and hand pose estimation have made it feasible to extract detailed spatio-temporal keypoint sequences from sign language videos. This feasibility has enabled the development of automated recognition systems that rely primarily on pose-based inputs. In fact, a growing body of work has demonstrated strong performance using 2D or 3D skeletal representations as the main input modality [1–13]. However, several studies have highlighted persistent challenges that hinder progress in pose-based SLR. These include inconsistencies in keypoint formats across pose estimation models, a lack of robust handling of occlusions, and the absence of standardized preprocessing or temporal smoothing pipelines. For instance, [14] shows that different pose estimation models affect downstream recognition accuracy. Similarly, [15] emphasizes the variability in pose quality and its impact on translation models. [16] further underscores the need for occlusion-aware processing to ensure stable keypoint sequences in real-world settings. These limitations reduce reproducibility, complicate dataset integration, and often require researchers to implement custom pipelines from scratch. Our literature analysis reveals four recurrent challenges in current pose-based SLR pipelines:

- Inconsistent output formats: Pose estimation tools such as MediaPipe (540+ landmarks), COCO-133, or task-specific formats produce incompatible outputs, making it difficult to combine datasets or switch between backends [17–19].

clazoq@uni.pe,racastroc@uni.pe, mauricio.salazare@pucp.edu.pe, gissella.bejarano@marist.edu

- Inadequate temporal filtering: Jittery keypoint trajectories fail to capture the smooth, structured motion of signing. Temporal filters such as Kalman filters are traditionally used to estimate the underlying motion dynamics and reduce noise across frames. However, existing Kalman-based methods [20,21] require manual tuning and often fail to generalize to the complex, articulated motions and occlusions common in sign language.
- Lack of standardized preprocessing pipelines: The absence of well-defined preprocessing stages hinders reproducibility and slows down experimentation, unlike established datasets such as WLASL [22] and COCO-WholeBody [23] that benefit from unified annotation protocols.
- Limitations in quality control methods: Current approaches often fail to address key issues such as missing joints during fast movements, occlusions between hands and face, or inconsistent detections across frames—challenges that are common in real-world signing scenarios.

To address these challenges, we introduce WholeBodyPose, a unified, end-to-end framework specifically designed for sign language recognition and gesture-based training data augmentation. By standardizing pose-based preprocessing and lowering technical barriers, WholeBodyPose enables more accessible and scalable SLR pipelines with potential impact on real-time systems and assistive technologies for the deaf and hard-of-hearing community.

Our framework provides four core contributions: (1) a unified interface across pose estimation models (MediaPipe, RTMPose, ViTPose) with a consistent API for easy comparison and integration; (2) a standardized COCO-133-compatible format that aligns heterogeneous pose outputs into a unified 133-keypoint indexing scheme, enabling interchangeable visualization, training, and evaluation; (3) a complete preprocessing pipeline from raw video to training-ready keypoint sequences with quality control and augmentation; and (4) extensive validation across multiple benchmarks demonstrating consistent accuracy gains with common SLR architectures.

2. RELATED WORK

2.1 Sign Language Recognition Systems

Isolated Sign Language Recognition (ISLR) systems have achieved significant progress using transformer-based architectures and pose-based inputs. Multimodal models, such as SignBERT+ [5], MSLU [13], and Uni-Sign [6], leverage large-scale pretraining to achieve strong performance, but they require high computational resources and extensive multimodal corpora.

In contrast, recent pre-training models focus on lightweight and scalable architectures using only 2D keypoints. Graph-based methods such as GCN-BERT [11] and HW-GAT [9] integrate spatial graph convolutions with temporal attention, while transformer-based models such as SPOTER [2], SiLT [7], and Siformer [10] adopt encoder–decoder or encoder-only structures without external data. These pose-only approaches highlight the importance of standardized keypoint representations and augmentation strategies for robustness and reproducibility. Our work builds upon this line by introducing a unified pipeline and benchmarking protocol across five ISLR datasets, supporting real-time pose rendering, and reproducible evaluation under consistent preprocessing.

Table 1: Summary of ISLR models, modality, and pose estimator.

Model	**Year**	**Modality**	**Pose Estimator**	**Model**	**Year**	**Modality**	**Pose Estimator**
Uni-Sign [6]	2025	RGB+Pose	MMPose (RTMPose-x [17])	Sign2Pose [24]	2023	Pose	Apple Vision API [25]
KD-MSLRT [26]	2025	Pose	MediaPipe [18]	SILT [7]	2023	Pose	MediaPipe [18]
WMSLR [27]	2024	RGB+Pose	ViTPose [19]	SPOTER [2]	2022	Pose	Apple Vision API [25]
MASA [12]	2024	RGB+Pose	MMPose (unspecified)	SignPose [28]	2021	Pose	OpenPose [29]
MSLU [13]	2024	Text+Pose	MMPose (unspecified)	GCN-BERT [11]	2021	Pose	Deeppose [30]
Siformer [10]	2024	Pose	Apple Vision API [2]	OpenHands [31]	2021	RGB+Pose	MMPose (unspecified)
SLR-LSTM [32]	2023	Pose	MediaPipe [18]	SignBERT [4]	2021	Pose	MMPose (unspecified)

2.2 Whole-Body Pose Estimation Models

The field has seen rapid development in whole-body pose estimation models. MediaPipe Holistic [18], the most widely adopted option, provides real-time estimation of 540+ keypoints (33 pose, 21 per hand, 468 facial) with near real-time performance on mobile devices, facilitating its integration into sign language applications. However, its proprietary format creates integration challenges.

Another alternative, ViTPose [19] (NeurIPS, 2022), employs plain vision transformers as backbones, achieving an 81.1 AP on the MS COCO test set and demonstrating excellent scalability from 100M to 1B parameters with 133 keypoints. RTMPose [17] bridges the accuracy–speed gap, achieving 75.8% AP on MS COCO with 90+ FPS on an Intel i7-11700 CPU and maintaining 72.2% AP with 70+ FPS on mobile devices (Snapdragon 865) with 133 keypoints. While these models achieve impressive individual performance, they produce outputs in different formats and coordinate systems, making direct comparison and integration difficult. Our COCO-133 standardization addresses this critical gap.

2.3 Standardization and Dataset Development

The establishment of comprehensive standards has been a key aspect to improving ISLR models. Jin [23] introduced COCO-WholeBody (ECCV 2020), the first benchmark with manual annotations covering the entire human body using 133 keypoints (17 body, 6 feet, 68 face, 42 hands). Their ZoomNet model handles hierarchical structure and scale variation, establishing a foundation for whole-body pose estimation.

Li [22] released WLASL (WACV 2020), the largest word-level sign language dataset containing over 100 signers and 2,000+ signs. Vaezi Joze [33] introduced MS-ASL, a large-scale dataset with consistent signer identities and frame-level annotations. More recently, Desai [34] presented ASL-Citizen, a community-sourced dataset with signer and demographic diversity. Despite their value, these datasets typically provide only video data, requiring researchers to extract pose information independently using different models and formats, which creates challenges for reproducibility and comparison.

3. METHODOLOGY

WholeBodyPose is designed as a modular Python library with six main components based on our implementation.

3.1 COCO-133 Standardization

We adopt a unified 133-keypoint representation tailored for sign language applications, however, this number can be adjusted as needed. The configuration can be easily extended, for example, by adding virtual landmarks such as the middle chest and the middle hip, resulting in 135 keypoints. This flexibility enables adaptation to different pose estimation setups and research needs. The standard configuration includes: 23 body keypoints (COCO-17 plus six additional foot landmarks), 68 facial landmarks, and 42 hand keypoints (21 per hand). Users may customize subsets via official index dictionaries (MMPose, OpenPose, MediaPipe), promoting extensibility and compatibility.

3.2 Pose Estimation Backends

We provide unified wrappers for three state-of-the-art pose estimation models, each offering different trade-offs between speed and accuracy. MediaPipe offers real-time performance on CPUs with over 540 landmarks, optimized for mobile deployment through efficient tracking and stream processing. RTMPose achieves the optimal balance with 75.8% AP on COCO at 90+ FPS on CPU, supporting multiple deployment backends (ONNX, OpenVINO) for production environments. ViTPose leverages vision transformers to achieve the highest accuracy (81.1% AP on COCO) for applications where precision is critical. All backends expose a consistent API through a simple interface—`model.predict(frame)` returns standardized COCO-133 keypoints and confidence scores regardless of the underlying model, enabling seamless backend switching without code changes.

Algorithm 1 MediaPipe wrapper (pseudo-code)

$model = MediapipeModel(models = ['hands','face','pose'], mode_coco = True)$
$keypoints, scores = model.predict(frame)$

3.3 Quality Control and Enhancement

Our framework incorporates quality control mechanisms to ensure robust pose estimation under challenging conditions. Image enhancement techniques address poor illumination that commonly degrades pose detection quality. These include: gamma correction, logarithmic transforms, and RGB channel-wise brightness adjustment. For visualization and analysis, the `DrawerPose` utility renders skeletons with configurable connections, confidence-based coloring, and dual modes that support both COCO-133 and extended formats, including virtual center points. These components work together to improve keypoint detection reliability and provide visual feedback for quality assessment, enabling researchers to identify and address potential issues in their preprocessing pipelines.

3.4 Real-Time Keypoint Filtering with NLMS

To improve the temporal stability of keypoints in sign language videos, we implement a real-time filtering method based on the Normalized Least Mean Squares (NLMS) algorithm. In this context, "real-time" refers to the ability to process each frame sequentially as it arrives, relying only on current and past information. This filtering process contrasts with offline filters that require access to the entire video or future frames to perform smoothing. Our approach is particularly suited for live applications, such as real-time sign language recognition, gesture-based interaction, or low-latency streaming, where future frames are not available and waiting for the whole sequence would be infeasible.

Algorithm 2 Real-Time Keypoint Filtering with NLMS (pseudo-code)

Initialize pose model *model*
Initialize NLMS filters and memory buffers
$filtering \leftarrow$ FilteringModel(133)
$filtering.start()$
Open video source
while frame available **do**
  Read and resize frame; convert to RGB
  $K, S \leftarrow model.predict(frame)$
  $K_{new}, S_{new} \leftarrow filtering(t, K, S, filters, memory)$
  Draw skeleton using filtered keypoints
  Display frame; $t \leftarrow t + 1$
  if user presses 'q' **then**
    break
  end if
end while
Release video and close windows

The filter operates on the keypoints frame by frame and assigns one NLMS filter per coordinate axis (x, y) and one for the confidence score to each keypoint. The filter receives a simple input vector containing the frame index and a normalized score indicator and learns to predict future values while adapting over time. To enhance robustness, we incorporate a short-term memory mechanism that stores the most recent high-confidence observations. When the model encounters low-confidence predictions, it can fall back to a remembered keypoint or rely on the NLMS prediction. In ambiguous cases, the system interpolates between the observed value and the predicted one using a weighted average.

Unlike traditional filtering approaches, which require complete sequences and introduce latency, this online method enables immediate smoothing during inference. This functionality enables the denoising of keypoints on the fly, thereby improving stability in downstream tasks without compromising responsiveness. The method scales naturally to N keypoints and N memories, making it lightweight and applicable in practical real-time systems.

3.5 Datasets

We apply our preprocessing framework to five public SLR datasets: WLASL [22], MSASL [33], ASL-Citizen [34], LSA64 [35], and PUCP-DGI [36]. All datasets are processed with RTMPose to extract 133 2D keypoints (COCO-WholeBody [23]) plus two additional virtual keypoints (chest center and mid-hip), resulting in a consistent 135-keypoint representation. We normalize coordinates to $[0, 1]$ and store original frame dimensions for denormalization.

Table 2: Datasets description and statistics.

Dataset	Train	Val	Test	Classes	Avg. frame length	Samples per class
MSASL100	3774	1190	757	100	86.79 ± 40.26	57.21 ± 8.04
MSASL200	6296	2041	1359	200	87.93 ± 40.57	48.48 ± 10.72
MSASL1000	16002	5287	4172	1000	86.70 ± 40.05	25.46 ± 13.78
WLASL100	1442	338	258	100	62.29 ± 23.76	20.38 ± 3.28
WLASL300	3549	900	668	300	61.64 ± 23.61	17.06 ± 3.14
WLASL2000	14289	3916	2878	2000	60.45 ± 23.16	10.54 ± 3.55
ASL-Citizen100	1707	393	1444	100	80.26 ± 34.47	35.44 ± 3.42
ASL-Citizen300	4699	1128	4058	300	79.63 ± 34.16	32.95 ± 2.67
ASL-Citizen	40154	10304	32941	2731	82.76 ± 37.69	30.54 ± 1.73
PUCP-DGI30	500	108	107	30	15.42 ± 9.11	23.83 ± 10.47
LSA64	2560	320	320	64	82.76 ± 27.58	50 ± 0

Splitting protocols. We follow official splits where available (WLASL, MSASL, ASL-Citizen). For ASL-Citizen, we additionally construct ASL-Citizen100 and ASL-Citizen300 by selecting the most frequent classes. For LSA64, we define a signer-independent split (8 train, 9th val, 10th test; 50 samples per signer). For PUCP-DGI30, we use a stratified random split (70/15/15).

3.6 Pose Visualization

The `DrawerPose` module supports COCO-133 and OpenPose-135 (by toggling `mode_coco`), interpolating virtual chest and mid-hip keypoints, and adding anatomically meaningful links. It supports multi-person rendering, confidence thresholds, and high-quality overlays via OpenCV.



(a) COCO-133 keypoints (b) OpenPose-135 keypoints (c) Whole-body 133 indexing [23]

Figure 1: Visualization of standardized keypoint sets using the `Drawer Pose` utility. The renderer applies consistent color-coding and skeletal links across different pose estimation models, facilitating direct comparison of outputs.

4. RESULTS

We evaluate the impact of improved pose estimation on several state-of-the-art ISLR models by replacing the original pose inputs with our WholeBodyPose (RTMPose-based) 135-keypoint representation (body, hands, face). All models are trained from scratch using only pose inputs, without pretraining or external data. We apply

simple geometric augmentations (random rotation, zoom, perspective) directly to keypoints. Table 3 reports per-instance and per-class Top-1/Top-5 accuracy on WLASL100.

Table 3: Ablation on WLASL100: all models trained from scratch using only pose inputs (WholeBodyPose RTMPose).

Method	**Per-Instance**		**Per-Class**	
	Top-1	**Top-5**	**Top-1**	**Top-5**
Pose-TGCN [22]	55.43	78.68	-	-
Ours (Pose-TGCN + WholeBodyPose)	58.14	82.58	59.22	83.58
GCN-BERT [11]	60.15	83.98	-	-
Ours (GCN-BERT + WholeBodyPose)	66.67	91.47	68.15	92.08
SILT [7]	62.79	-	-	-
Ours (SILT + WholeBodyPose)	66.67	84.88	68.15	85.72
SPOTER [2]	63.18	-	-	-
Ours (SPOTER + WholeBodyPose)	67.05	86.43	67.47	87.42

Findings. We observe consistent, substantial improvements across all models. For instance, GCN-BERT improves from 60.15% to 66.67% (Top-1 per-instance) and from 83.98% to 91.47% (Top-5 per-instance). We observe similar trends for Pose-TGCN (+2.71%), SILT (+3.88%), and SPOTER (+3.87%). Clean, standardized inputs with minimal augmentations outperform architectural tweaks or heavy pretraining in low-resource scenarios.

5. DISCUSSION

5.1 Framework Contributions

WholeBodyPose addresses critical challenges in sign language recognition research by providing a comprehensive standardization infrastructure. The COCO-133 format represents the first standardization effort specifically designed for sign language applications, capturing 133 keypoints with equal importance given to hand and facial landmarks, alongside body pose. This approach recognizes that sign language relies heavily on fine-grained manual and facial expressions. This cross-model compatibility enables researchers to work with different pose estimation backends without format-specific preprocessing.

Our framework significantly reduces technical barriers by providing a unified interface that simplifies model comparison through identical preprocessing, thereby eliminating the need for custom development and ensuring reproducible results across research groups. This standardization isolates algorithmic contributions from implementation variations, allowing researchers to focus on novel contributions rather than infrastructure development. Accessibility aspects are particularly important: Our easy-to-use interface enables researchers without extensive computer vision expertise to access state-of-the-art capabilities, while resource efficiency makes high-quality preprocessing accessible even with limited computational resources.

5.2 Practical Applications

The framework enables important research directions, including cross-dataset studies through standardized formats, fair model comparison via identical preprocessing, and automated analysis of large video collections. Educational applications benefit from objective feedback capabilities for sign language learning and standardized proficiency metrics. For assistive technology, the framework supports real-time translation systems through optimized processing and enables cross-platform compatibility for communication aids, directly improving accessibility for the deaf and hard-of-hearing community.

5.3 Limitations and Future Directions

While our 2D representation offers computational efficiency and broad applicability, it cannot capture depth information, which is crucial for disambiguating overlapping hands or complex signing positions. Future work should explore hybrid 2D/3D approaches and investigate domain adaptation techniques for cross-dataset generalization. The NLMS filtering, though effective, may require extension to handle more complex temporal dynamics in continuous signing. Additionally, expanding the framework to support emerging pose estimation models and developing standardized benchmarks for continuous sign language recognition remain important directions for future research.

6. CONCLUSION

In this work, we introduced WholeBodyPose, a unified end-to-end framework that standardizes pose estimation outputs for sign language recognition and related applications. Our contributions include a consistent COCO-133 representation, modular backends for state-of-the-art estimators (MediaPipe, RTMPose, ViTPose), quality control pipelines, and real-time filtering via NLMS. By addressing long-standing challenges of format inconsistency, inadequate temporal smoothing, and lack of reproducible preprocessing, our framework significantly improves both the accessibility and reliability of pose-based research.

Through extensive experiments across five public datasets and their subsets, we demonstrated that standardized and high-quality pose representations yield consistent accuracy gains over existing baselines, even when using lightweight recognition models trained from scratch. This confirms that clean and robust inputs can be more impactful than architectural modifications or heavy pretraining, especially in low-resource settings. Beyond enabling fairer comparisons and reproducible research, our open-source library lowers technical barriers and facilitates broader adoption of pose-based approaches in accessibility, education, and real-time assistive technologies. To encourage reproducibility and adoption, we release our complete framework with image/video inference, real-time webcam filtering, and training of state-of-the-art ISLR models. In addition, all standardized pose-based versions of the datasets used in this work are made available for download and benchmarking at https://github.com/CristianLazoQuispe/WholebodyPoseEstimation.

REFERENCES

[1] Alyami, S., Luqman, H., and Hammoudeh, M., "Isolated arabic sign language recognition using a transformer-based model and landmark keypoints," *ACM Transactions on Asian and Low-Resource Language Information Processing* **23**(1), 1–19 (2024).

[2] Boháček, M. and Hrúz, M., "Sign pose-based transformer for word-level sign language recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 182–191 (2022).

[3] Deng, Z., Leng, Y., Hu, J., Lin, Z., Li, X., and Gao, Q., "Sml: A skeleton-based multi-feature learning method for sign language recognition," *Knowledge-Based Systems* **301**, 112288 (2024).

[4] Hu, H., Zhao, W., Zhou, W., Wang, Y., and Li, H., "Signbert: Pre-training of hand-model-aware representation for sign language recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 11087–11096 (2021).

[5] Hu, H., Zhao, W., Zhou, W., and Li, H., "Signbert+: Hand-model-aware self-supervised pre-training for sign language understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(9), 11221–11239 (2023).

[6] Li, Z., Zhou, W., Zhao, W., Wu, K., Hu, H., and Li, H., "Uni-sign: Toward unified sign language understanding at scale," *arXiv preprint arXiv:2501.15187* (2025).

[7] Luna-Jiménez, C., Gil-Martín, M., Kleinlein, R., San-Segundo, R., and Fernández-Martínez, F., "Interpreting sign language recognition using transformers and mediapipe landmarks," in *Proceedings of the 25th International Conference on Multimodal Interaction*, 373–377 (2023).

[8] Miah, A., Hasan, M., Okuyama, Y., Tomioka, Y., and Shin, J., "Spatial–temporal attention with graph and general neural network-based sign language recognition," *Pattern Analysis and Applications* **27**(2), 37 (2024).

[9] Patra, S., Maitra, A., Tiwari, M., Kumaran, K., Prabhu, S., Punyeshwarananda, S., and Samanta, S., "Hierarchical windowed graph attention network and a large scale dataset for isolated indian sign language recognition," (2024).

[10] Pu, M., Lim, M. K., and Chong, C. Y., "Siformer: Feature-isolated transformer for efficient skeleton-based sign language recognition," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 9387–9396 (2024).

[11] Tunga, A., Nuthalapati, S. V., and Wachs, J., "Pose-based sign language recognition using gcn and bert," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 31–40 (2021).

[12] Zhao, W., Hu, H., Zhou, W., Mao, Y., Wang, M., and Li, H., "Masa: Motion-aware masked autoencoder with semantic alignment for sign language recognition," *IEEE Transactions on Circuits and Systems for Video Technology* **34**(11), 10793–10804 (2024).

[13] Zhou, W., Zhao, W., Hu, H., Li, Z., and Li, H., "Scaling up multimodal pre-training for sign language understanding," *arXiv preprint arXiv:2408.08544* (2024).

[14] Lazo-Quispe, C., Huamani-Malca, J., Huamán-Ramos, M. S. H., Bejarano, G., Rivas, P., and Cerny, T., "Impact of pose estimation models for landmark-based sign language recognition," in *LXAI Workshop @ NeurIPS*, (2022).
[15] Amrutha, K. and Prabu, P., "Evaluating the pertinence of pose estimation model for sign language translation," *International Journal of Computational Intelligence and Applications* **22**(1), 2341009 (2023).
[16] Das, S., Biswas, S. K., and Purkayastha, B., "Occlusion robust sign language recognition system for indian sign language using cnn and pose features," *Multimedia Tools and Applications* **83**(36), 84141–84160 (2024).
[17] Jiang, T. et al., "Rtmpose: Real-time multi-person pose estimation based on mmpose," *arXiv preprint arXiv:2303.07399* (2023).
[18] Lugaresi, C. et al., "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172* (2019).
[19] Xu, Y., Zhang, J., Zhang, Q., and Tao, D., "Vitpose: Simple vision transformer baselines for human pose estimation," in *Advances in Neural Information Processing Systems*, 38571–38584 (2022).
[20] Martini, E., Boldo, M., and Bombieri, N., "Flk: A filter with learned kinematics for real-time 3d human pose estimation," *Signal Processing* **224**, 109598 (2024).
[21] Villegas, R., Yang, J., Zou, D., Sohn, S., Lin, X., and Lee, H., "Long short-term memory kalman filters: Recurrent neural estimators for pose regularization," in *Proceedings of the IEEE International Conference on Computer Vision*, 5524–5532 (2017).
[22] Li, D., Rodriguez, C., Yu, X., and Li, H., "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1459–1469 (2020).
[23] Jin, S., Xu, L., Xu, J., et al., "Whole-body human pose estimation in the wild," in *European Conference on Computer Vision*, 196–214 (2020).
[24] Eunice, J., Sei, Y., and Hemanth, D. J., "Sign2pose: A pose-based approach for gloss prediction using a transformer model," *Sensors* **23**(5), 2853 (2023).
[25] Apple Inc., "Vision - apple developer documentation." https://developer.apple.com/documentation/vision (2025). Accessed Aug. 5, 2025.
[26] Li, Y., Ren, B., Hu, K., Liu, C., Jiang, Z., Dang, K., and Su, J., "Kd-mslrt: Lightweight sign language recognition model based on mediapipe and 3d-to-1d knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 28177–28185 (2025).
[27] Liu, E., Lim, J. Y., MacDonald, B., and Ahn, H. S., "Weighted multi-modal sign language recognition," in *Proceedings of the IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 880–885 (2024).
[28] Krishna, S. et al., "Signpose: Sign language animation through 3d pose lifting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2640–2649 (2021).
[29] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y., "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7291–7299 (2017).
[30] Toshev, A. and Szegedy, C., "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1653–1660 (2014).
[31] Selvaraj, P., Nc, G., Kumar, P., and Khapra, M., "Openhands: Making sign language recognition accessible with pose-based pretrained models across languages," (2021).
[32] Rao, G. M., Sowmya, C., Mamatha, D., Sujasri, P. A., Anitha, S., and Alivela, R., "Sign language recognition using lstm and mediapipe," in *Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1086–1091 (2023).
[33] Vaezi Joze, H. and Koller, O., "Ms-asl: A large-scale data set and benchmark for understanding american sign language," in *The British Machine Vision Conference (BMVC)*, (September 2019).
[34] Desai, A., Berger, L., Minakov, F. O., Milan, V., Singh, C., Pumphrey, K., Ladner, R. E., Daumé III, H., Lu, A., Caselli, N., and Bragg, D., "Asl citizen: A community-sourced dataset for advancing isolated sign language recognition," in *Advances in Neural Information Processing Systems (NeurIPS) 36*, (June 2023).
[35] Ronchetti, F., Quiroga, F. M., Estrebou, C., Lanzarini, L., and Rosete, A., "Lsa64: An argentinian sign language dataset," *arXiv preprint arXiv:2310.17429* (2023).
[36] Bejarano, G., Huamani-Malca, J., Cerna-Herrera, F., Alva-Manchego, F., and Rivas, P., "Perusil: A framework to build a continuous peruvian sign language interpretation dataset," in *Proceedings of the LREC2022 10th Workshop on the Representation and Processing of Sign Languages: Multilingual Sign Language Resources*, 1–8 (2022).