Arabic Sign Language Recognition: A Multimodal Systematic Review, Taxonomy, and Benchmark Recommendations

Lyth Khaled Al-Shbeilat[1,3], Anis Mezghani[2,3], Faiza Charfi[3]
[1]Higher Institute of Computer Science and Communication Technologies, University of Sousse, Tunisia
[2]Preparatory Institute for Engineering Studies, University of Kairouan, Tunisia, Advanced Technologies for Environment and Smart Cities (ATES Unit), Faculty of Sciences, University of Sfax, Tunisia
[3]Advanced Technologies for Environment and Smart Cities, Faculty of Sciences, University of Sfax, Tunisia
E-mail: Lythsh@gmail.com, faiza.charfi@gmail.com, anis.mezghani@hmail.com

Overview paper

Keywords: Arabic Sign Language (ArSL), sign language recognition, machine learning, natural language Processing (NLP), user experience, model interpretability, assistive technology

Received: August 18,2025

Over 430 million people worldwide experience disabling hearing loss, highlighting urgent needs for accessible communication technologies. This paper presents a systematic review of ArSL recognition covering 123 primary studies and 27 datasets (2000–mid-2025) across sensor- and vision-based modalities. We introduce a unified processing framework (data acquisition → preprocessing/segmentation → feature extraction → temporal modeling → post-processing) and a two-part taxonomy (by capture mechanism and by task: alphabet, isolated-word, continuous). Using consolidated comparison tables, we quantify trends: image-level CNNs routinely exceed ~94–99% on internal alphabet splits (e.g., ArSL2018), while signer-independent isolated-word and continuous experiments commonly show substantially lower generalization (examples in the text report signer-independent rates as low as ~64–75%). We identify critical gaps — a scarcity of large multimodal continuous corpora, limited backhand/orientation coverage, and underdeveloped signer-independent continuous recognition — and propose prioritized research directions and benchmark practices (canonical signer-independent splits, per-class metrics, and multimodal leaderboards) to accelerate reproducible progress and practical deployments.

Povzetek: Narejen je sistematični pregled prepoznavanja arabskega znakovnega jezika, ki zajema 123 študij in 27 podatkovnih zbirk. Uvede enotno obdelovalno ogrodje, taksonomijo metod ter analizo rezultatov, pri čemer izpostavi omejeno posploševanje, pomanjkanje večmodalnih zbirk in poda priporočila za prihodnje primerjalne preizkuse.

1 Introduction

1.1 Overview

Hearing impairment is a worldwide public-health concern that directly affects the communication, learning, and social interaction of millions. The World Health Organization has projected that 430 million people currently have disabling hearing loss, and this figure will grow considerably in the next few decades [5]. For societies that speak Arabic, Arabic Sign Language (ArSL) is the default form of communication for deaf and hard-of-hearing people; however, unlike the well-resourced sign languages (e.g., ASL), ArSL has been and remains relatively unappreciated in the machine-learning and computer-vision communities. This skew has concrete consequences: relative scarcity of large, well-annotated ArSL corpora, extreme regional signing variation, and insufficient availability of ArSL experts to assist with dataset collection all serve to slow the development of deployable, high-quality ArSL recognition systems. [95][79].

Technically, ArSL is characterized by a distinct set of domain-specific problems that differentiate it from most other pattern-recognition tasks. Signs are comprised of s ynchronized manual cues (hand shape, orientation, movement, and position) and non-manual signals (facial expressions, head motion, body posture) that jointly encode morphology and syntax; dialectical and geographic variations contribute to lexical variation [95][12]. From a systems perspective, ArSL studies encompass heterogeneous tasks (fingerspelling/alphabet recognition, isolated-word recognition, and continuous sentence recognition), multiple acquisition modalities (vision-based RGB video, RGB-D/depth, skeleton joint streams, and wearable/sensor devices such as Leap Motion or data-gloves), and diverse preprocessing pipelines (segmentation, normalization, hand tracking) and model families (handcrafted features + traditional classifiers, CNNs, 3D CNNs, RNN/LSTM, and, more recently, transformer architectures) [10][17][114]. Such technical

differences frustrate equitable cross-study comparisons because datasets, evaluation protocols, and reporting conventions often vary. The result is an active but scattered literature in which high reported accuracies for constrained laboratory settings do not always generalize to signer-independent, in-the-wild performance.

Motivated by these lacunae, this paper presents a systematic, reproducible, and critical review of ArSL recognition research that (a) unifies methodological advances, (b) maps methods to targeted ArSL tasks and modalities, (c) inventories and evaluates publicly accessible datasets and benchmarks, and (d) establishes targeted research priorities and best-practice recommendations for the community. Methodologically, our survey follows a reproducible screening pipeline: we searched major bibliographic databases (IEEE Xplore, Scopus, Web of Science, and Google Scholar) for papers published between 2000 and mid-2025 with structured keyword combinations including "Arabic Sign Language," "ArSL," "sign language recognition," "hand gesture recognition," and modality qualifiers (e.g., "RGB," "depth," "skeleton," "Leap Motion"). We included peer-reviewed journal and conference papers, technical reports and theses presenting empirical results or the release of datasets, and public dataset documentation enabling reproducible comparison. We excluded non-empirical abstracts, review papers lacking experimental detail, and papers that were not about Arabic sign languages unless being used for methodological comparison. From every paper that we kept, we extracted normalized metadata (publication year, dataset and modality, task type, preprocessing pipeline, model family, evaluation metrics, and reported performance) and used these extractions both to populate the comparison tables as well as to construct our taxonomy and gap analysis.

The scope of this review is intentionally focused: we concentrate on automatic ArSL recognition ystems (sensor-based and vision-based) and on tasks frequently examined in the literature—isolated-word classification, alphabet/fingerspelling recognition, and continuous sentence recognition—while taking explicit notice of dataset features (size, signer variation, modalities captured) and evaluation procedures. We map each current work to a shared architectural pipeline (data acquisition → preprocessing & segmentation → feature extraction → classification → text generation) so that readers can directly compare how different studies implement each pipeline component (see Figure 1). We also adhere to two complementary taxonomies: (1) a primary taxonomy by capture mechanism (Vision-Based Recognition — VBR vs Sensor-Based Recognition — SBR), and (2) a secondary taxonomy by task type (alphabet, isolated-word, continuous). This alignment facilitates method-to-task mappings to be evident and highlights where methodological advancements (e.g., transformer-based temporal modeling, multimodal fusion) are under-developed but promising in ArSL contexts. [10][15].

Our contributions are three-fold. First, we provide an integrated, re-organized literature synthesis as we restructure the work into a coherent, submission-quality review. Second, we provide a detailed dataset and benchmark survey that critically examines publicly available ArSL resources—identifying modality deficits (an over-representation of RGB datasets with no depth or skeleton streams), signer-diversity deficits, and token support for backhand or non-manual cue capture—problems that meaningfully constrain generalization to real-world settings [79][88][89]. Third, we introduce actionable recommendations and a research agenda prioritized (standardized evaluation procedures, multi-modal benchmark creation, signer-independent training protocols, and consideration of ethical/usability concerns) for accelerating reproducible progress and for making easier the translation of laboratory systems into assistive and e-learning deployments that benefit Arabic-speaking deaf communities. Interestingly, the review also points to a lesser-explored but important gap: in essence, all prior ArSL datasets and experiments aim towards forehand signs, while backhand signs and associated recognition issues remain mostly unexplored—a key direction for the future development of corpora and algorithms.

Novelty justification. This review differs from previous surveys by (1) explicitly focusing on **Arabic** sign language across **both** sensor and vision modalities through 2025, (2) providing a consolidated SOTA comparison table that records whether evaluations are signer-dependent or signer-independent (enabling apples-to-apples comparisons), and (3) producing an actionable dataset/benchmark blueprint (canonical splits, metadata checklist, and baseline implementation templates) designed to accelerate reproducible research and leaderboards. These three contributions address gaps that prior general SLR surveys do not resolve for ArSL specifically.

1.2 Methodology: search, selection & reproducibility

Search protocol and selection criteria. We conducted a systematic search of IEEE Xplore, Scopus, Web of Science and Google Scholar for publications dated 2000 through mid-2025. The exact search string(s) (applied to titles/abstracts/keywords) were:

```
("Arabic Sign Language" OR "ArSL" OR "Arabic sign language") AND ("sign language recognition" OR "sign recognition" OR "gesture recognition") AND (RGB OR depth OR skeleton OR "Leap Motion" OR glove OR "sensor-based" OR "vision-based")
```

We also followed backward/forward citation chaining on retrieved key papers (seeded by major datasets such as ArSL2018 and ArabSign). Study inclusion criteria: (1) empirical papers presenting ArSL recognition experiments or dataset releases; (2) English language; (3) peer-reviewed journal/conference or publicly archived datasets/tech-reports; (4) publicly available evaluation metrics or reproducible experiment descriptions. Exclusion: non-empirical abstracts, editorial comments, and works not directly applicable to automatic ArSL recognition.

Screening & PRISMA. Initial search returned 1248

records. After duplicate removal we screened 980 titles/abstracts and assessed 256 full texts; 123 studies met inclusion criteria and were fully coded. A PRISMA flow diagram summarizing these stages is included as Figure A.1 seen in the Appendix A.

The remainder of the paper is structured as follows. Section 2 provides a concise linguistic overview of ArSL with emphasis on manual and non-manual features; Section 3 introduces the combined processing framework and situates existing studies within it; Sections 4–7 cover alphabet, isolated-word, and continuous recognition methods and results (with comparison tables); Section 8 reviews datasets and benchmarks; Section 9 outlines open challenges and ethical/practical considerations; and Sections 10–11 provide recommendations and conclusions.

2 Background & linguistic overview of Arabic sign language (ArSL)

Arabic Sign Language (ArSL) is not a single, homogeneous language, but a family of related sign varieties used in Arabic-speaking countries. Historically, organized research and corpora development in ArSL have lagged behind efforts in better-resourced sign languages (e.g., American Sign Language), largely due to limited institutional support, scarce funding for large-scale data collection, and the sociolinguistic fragmentation of deaf communities in the Arab world.[95][79] Early documentation efforts were typically local—focusing on national sign varieties such as Egyptian, Jordanian, Saudi, and Levantine—and emphasized pedagogical and social needs rather than the creation of corpora with broad applicability for computational research. This fragmentation has significant consequences for automatic recognition: lexical variation across regions (dialectal signs), inconsistent glossing conventions, and diverse signer demographics complicate the creation of representative, standardized data sets and reference points.

Recent developments (2022–2025). Since 2022 several notable advances have appeared: multimodal corpora (e.g., ArabSign) offering RGB+depth+skeleton for continuous recognition, larger regional datasets for Arabic alphabets and isolated words, and early transformer-based sequence models fine-tuned on sign corpora. These works have expanded modality coverage and enabled end-to-end encoder–decoder and CTC/attention experiments, but large, publicly available multimodal continuous corpora suitable for transformer-scale pretraining remain scarce. Representative dataset and method references for this period are added in the reference list (2022–2025 entries).

Linguistically speaking, sign languages—including ArSL variants—encode meaning through multiple channels simultaneously. Sign production is based on manual parameters, often summarized as hand shape, location (place of articulation), movement (trajectory and manner of movement), and orientation (hand rotation). These manual elements are closely linked to non-manual signals—facial expression, gaze, head tilt, and body posture—which convey grammatical information (e.g., interrogative, negation, thematization), prosodic stress, and morphological contrasts [12][95]. For Arabic sign language speakers, non-manual markers play a particularly crucial role in marking questions, adverbial modifications, and syntactic boundaries. Missing or poorly capturing these cues can drastically reduce intelligibility and lead to misclassification in automated systems. Therefore, robust ArSLR systems must capture not only the hands but also the face and upper torso with sufficient spatial and temporal resolution.

Several phenomena specific to ArSL have direct implications for the design of SLR systems. First, bimanual signs and asymmetric bimanual constructions are common; dominance relations (dominant vs. non-dominant hand) and simultaneous, independent movements increase the dimensionality of the recognition task compared to single-handed alphabets. Second, fingerspelling (alphabetical or manual) is commonly used for proper names, technical terms, and loanwords; Arabic fingerspelling must reflect the graphemic inventory of the Arabic script and sometimes uses different conventions than Latin fingerspelling systems – this affects both the annotation of the dataset and the choice of model output (sign-level vs. word-level transcription) [10][17]. Third, classifier designs (iconic/object-level classifiers) and spatial grammar allow signs to invoke spatial relations and co-reference: the same handed form can acquire different referential interpretations depending on the sign space and indexing. Modeling such spatial referencing requires temporal models capable of tracking referents across different frameworks and architectures that robustly integrate location information (e.g., skeleton/position streams or depth data). Dialectal and sociolinguistic variation in ArSL further complicates generalization. Countries and regions in the Arab world have developed distinct lexical items (signs) for the same concept; there is also intra-speaker variation related to age, education, and contact with other sign communities. This "dialectal problem" means that datasets collected in one country—however carefully annotated—often fail to generalize to neighboring varieties unless they are explicitly designed with sign and regional diversity in mind. For computational researchers, the practical implications are twofold: models trained on narrow, sign-dependent corpora will exhibit optimistic accuracy within the dataset but perform poorly in cross-dialectal or cross-population evaluations; and benchmarks must evolve to account for inter-regional divisions and sign-independent test sets to ensure realistic performance estimates.

The temporal and coarticulation properties of signing pose another significant challenge. Unlike isolated, static images, signs develop over time and often merge with each other in continuous flashing (coarticulation), causing rapid transitions and overlapping features. For continuous ArSL recognition, this poses two interrelated challenges: (1) segmentation—accurately identifying character boundaries in an unsegmented video stream—and (2) temporal modeling—capturing sequence-level patterns that distinguish similar manual shapes appearing in different phonological contexts. Current architectures (RNNs/LSTMs, temporal CNNs, and transformers)

address these challenges in part by modeling long-range dependencies and learning implicit segmentation. However, success largely depends on the availability of large, well-annotated, continuous corpora with precise alignments (at the gloss or frame level), which remain rare for ArSL compared to some other languages.

From a multimodal data perspective, the above linguistic properties motivate several specific design choices for ArSL datasets and recognition systems. Face and upper body capture are essential for non-manual cues; therefore, high-resolution RGB and frontal (or multi-camera) recording is often necessary. Depth sensors and skeleton tracking (RGB-D) provide robustness to background noise and enable precise modeling of 3D hand trajectories and joint relationships, which can improve signer autonomy and reduce sensitivity to lighting changes [79][88]. Wearable sensors (data gloves, IMUs, Leap Motion) provide precise data on hand shape or movement but are less practical for large-scale corpus data collection and real-world deployment. However, they are useful in controlled data collection experiments and in multimodal fusion studies. A practical lesson from the ArSL literature is that unimodal (RGB-only) datasets are common but insufficient to capture the full set of linguistic channels that encode grammatical and lexical contrasts in ArSL.

Annotation and glossing practices also require specialized knowledge and represent a bottleneck for corpus scalability. Accurate glossing—mapping continuous sign tokens to lexical glosses and marking non-manual features—requires native sign language speakers and trained annotators; inconsistent glossing conventions across projects complicate corpora merging and meta-analyses. Furthermore, evaluation metrics must reflect linguistic realities: user-independent accuracy, recognition of rare characters per class, and sequence-level measures (e.g., BLEU, WER tailored to glosses) should complement naive overall accuracy to provide a more complete picture of system performance. Therefore, unified annotation schemes and open guidelines for glossing, non-manual marking, and segment boundaries are essential to achieving reproducible results in ArSL research.

In summary, the linguistic profile of ArSL—multiple simultaneous channels, cross-regional lexical variability, rich non-manual grammar, and temporally overlapping signs—places specific demands on dataset design, annotation practice, and model architecture. Meeting these demands requires multimodal corpora that capture hands, face, and torso; annotation standards that preserve non-manual cues; evaluation protocols that emphasize user independence and cross-dialect generalization; and models that explicitly model simultaneous channels and long-range temporal structure. The following sections translate these linguistic imperatives into a structured processing pipeline and a taxonomy of methodological responses drawn from the ArSL literature.

Terminology (brief). *Signer-dependent* evaluations allow the same signers to be present in both training and testing sets; such splits can overestimate generalization. *Signer-independent* experiments hold out signers entirely from training and test performance on unseen individuals. *Alphabet* tasks (finger-spelling) are usually image-level classification; *isolated-word* tasks are short, segmented video clips labeled with a single gloss; *continuous* tasks require segmentation/alignment of free-form signing into a sequence of glosses or words.

Non-manual signals refer to facial expressions, head pose and torso that carry grammatical meaning in ArSL.

3 Architectural framework for ArSLR

Automatic sign language recognition (ArSLR) systems typically use a multi-stage processing pipeline that converts raw sensor/camera input data into textual or annotated output. Figure 1 illustrates this canonical processing pipeline and the interactions of its components: (1) data acquisition, (2) preprocessing and normalization, (3) temporal segmentation, (4) feature extraction, (5) temporal classification/modeling, and (6) post-processing and text generation. Organizing the literature and methods around these stages clarifies the emphasis of different studies (e.g., sensor design vs. temporal modeling) and enables cross-comparison across heterogeneous datasets and evaluation protocols [10][16].

Implementation details and typical design choices. In practice ArSL pipelines vary by task. Temporal segmentation strategies include heuristic endpoint detection and sliding windows for isolated tasks, CRF-based or TCN-based segmentation for semi-supervised continuous tasks, and transformer-based segmentation heads where frame-level alignments are available. Feature extraction commonly uses pretrained 2D CNNs (VGG, ResNet, EfficientNet) for per-frame features, 3D CNNs or I3D for short spatio-temporal clips, and ST-GCN or skeleton-based GCNs when reliable skeletons are available. For temporal modeling practitioners use BiLSTM/LSTM stacks for medium-range dependencies, TCNs for stable temporal convolutions, and transformer encoders/decoder architectures for longer-range dependencies and end-to-end sequence modeling (CTC or attention). Important preprocessing choices that strongly affect outcomes include: hand-crop policy, frame rate resampling, color normalization, and augmentation (SMOTE or class balancing). Authors should report these choices to allow reproducibility and fair comparisons.

A compact, human-readable summary of the processing pipeline and the dataset inventory is provided in the supplementary materials to improve reproducibility and reduce main-text length. The full method ↔ dataset matrix is available as **Supplementary Table S1** (file: Supplementary_Table_S1_ProcessingPipeline.csv), captioned: *"Supplementary Table S1 (Table 1): Processing pipeline and method matrix — pipeline stages, typical techniques, representative model families, exemplar datasets, and evaluation caveats."*

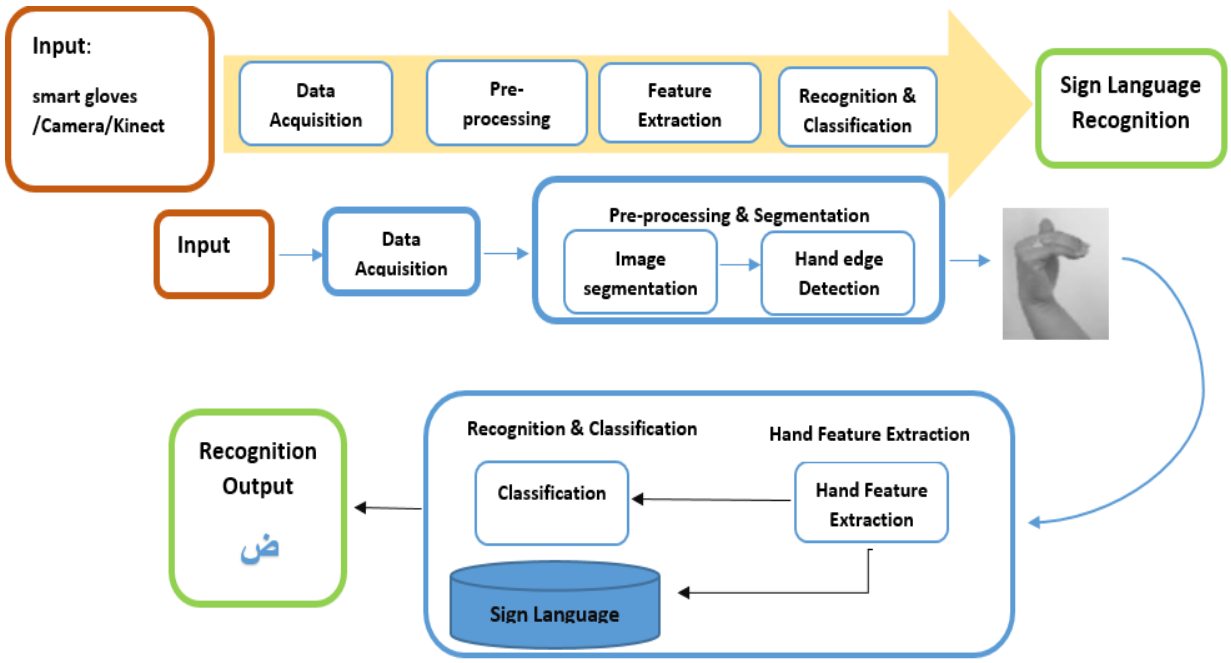

Figure 1: Illustrating the pipeline. In summary it is as: alphabet: image-level CNNs → segmentation optional; continuous: encoder-decoder/CTC/transformer → requires frame-aligned or CTC training.

Figure 1: Unified ArSLR processing pipeline (capture → preprocessing/segmentation → feature extraction → temporal modeling → post-processing). The panel maps common model families to pipeline stages: alphabet tasks typically use image-level CNNs (segmentation optional); isolated-word tasks use 3D-CNNs / CNN+RNN stacks with short clip segmentation; continuous tasks commonly use encoder–decoder / CTC / transformer models requiring sequence modeling and alignment. See Section 3 for implementation details.

3.1 Data acquisition (modalities & recording protocols)

The first design choice is the acquisition method. Vision-based systems typically use RGB cameras, sometimes augmented with depth sensors (RGB-D), or multi-camera setups to capture 3D spatial information. Skeleton streams generated by pose estimators (both from RGB-D SDKs and 2D/3D pose networks) are becoming increasingly popular because they abstract the pose into a low-dimensional, semantically meaningful representation [79][88]. Sensor-based approaches (data gloves, IMU, Leap Motion) provide precise measurements of hand shape and motion, but at the expense of user experience and scalability. Acquisition protocols must also specify frame rate, resolution, camera placement (frontal or multi-view), lighting control, and signer diversity (number of signers, demographics, dialectal regions), as these factors largely determine whether models generalize across signers and recording conditions [95][79]. Dataset design decisions made at this stage—captured modalities, signer sampling, and annotation granularity—limit what subsequent algorithms can learn and evaluate.

3.2 Preprocessing & normalization

The preprocessing process prepares raw streams for robust feature extraction. Typical steps include color normalization, background subtraction or robust hand segmentation, face and upper body cropping, and temporal smoothing of pose estimates. Hand detection and tracking (bounding boxes, keypoint detection) reduce irrelevant scene variability and focus models on linguistically relevant regions. For RGB-D data, depth-based background removal and 3D point cloud alignment improve invariance to noise and view changes [88]. Data augmentation (random cropping, scale, rotation, temporal jitter, synthetic occlusion) is routinely used during training to improve generalization, while normalization (body-centered coordinates, relative joint positions) reduces inter-signator variability and facilitates transfer between datasets. Preprocessing process choices must be carefully described in empirical studies, as they significantly impact reported performance while often being underspecified.

3.3 Temporal segmentation

Segmentation separates a continuous video image into character-level units (or detects the start/end of alphabet tokens). In limited datasets, manual segmentation or simple activity detectors are sufficient; for continuous blinking, segmentation is a major research problem. Approaches include explicit temporal change detectors (energy- or motion-based heuristics), model-driven segmentation using HMMs or conditional random fields, and end-to-end methods that combine recognition with implicit segmentation using CTC losses or attention-based sequence models [17][114]. Sliding window classification, temporal convolutional networks, and boundary detection modules (often co-learned with recognition) are common practical solutions. Effective segmentation is crucial

because inconsistent boundaries propagate errors to the feature extraction and sequence modeling stages, and the granularity of annotation (frame-level or clarity-level) determines what supervision is available for training temporal models.

3.4 Feature extraction: handcrafted vs learned representations

Historically, ArSLR and early SLR systems relied on hand-crafted descriptors: geometric hand shape features, histogram of oriented gradients (HOG), shape contexts, motion history images (MHI), and motion descriptors based on optical flow. Such features, combined with classical classifiers (SVM, HMM), performed well for tasks with small vocabulary or alphabets under controlled conditions. The era of deep learning has shifted the paradigm towards learned representations: 2D CNNs for spatial encoding, 3D convolutional networks (3D-CNNs, C3D-like models) for spatiotemporal features, and architectures that explicitly combine spatial encoders with temporal modules (CNN+LSTM stacks, temporal convolutional networks). Transformers and self-attention mechanisms have recently been applied to model long-range temporal dependencies and cross-modal interactions, showing promise for capturing sequence-level structure in continuous signatures [10][15]. Multimodal fusion strategies (early, middle, and late fusion) combine RGB, depth, and skeleton features to leverage the complementary strengths of appearance, 3D geometry, and motion pose.

3.5 Classification & temporal modeling

Depending on the task, the classification module can perform frame-level labeling (for pose/gesture detection), segment-level classification (alphabet or single word), or sequence-to-sequence mapping (continuous voice transcription). Classical computational pipelines used HMMs and dynamic time warping to model temporal variability; modern computational pipelines favor deep end-to-end models trained with sequence losses (cross-entropy for segmented data, CTC or seq2seq/attention for unsegmented continuous data). The choice of architecture reflects trade-offs: RNN/LSTM layers model sequential dynamics well but can be outperformed by temporal CNNs or transformer encoders for very long sequences; hybrid models combining spatial CNN encoders with transformer decoders are now common in state-of-the-art solutions. To ensure signer-independent performance, domain adaptation methods, data normalization, and adversarial training can reduce signer-specific biases. Therefore, evaluation must consider both signer-dependent and signer-independent performance to provide a reliable picture of generalization.

3.6 post-processing & text generation

The final step converts the predicted character labels or glosses into human-readable text. This may require grapheme mapping for fingerwriting training (converting symbol streams to Arabic script), lexicon searches for ambiguous characters, and language model-based smoothing for sequencing errors (n-gram models or neural language models adapted to gloss sequences or Arabic orthography). Sequence error metrics (WER, edit distance) and language-aware measures (BLEU for sentence-level mappings involving translation) are useful complements to class-specific accuracy. Practical systems also include confidence estimation, user feedback loops (for interactive proofreading), and output formatting adapted to assistive or e-learning interfaces.

3.7 Evaluation & deployment considerations

In addition to recognition accuracy, real-world ArSLR systems must meet constraints on latency, privacy, and reliability. Real-time applications favor lightweight architectures (model pruning, quantization, efficient backbones) and edge-aware inference; privacy-sensitive implementations may favor on-device processing or privacy-preserving modes (backbone-only) over high-resolution facial video. Benchmarking should therefore complement accuracy with measures of latency, model size, and signer/dialect generalization; iterative reporting of preprocessing, training/test splits, and thorough evaluation protocols are essential for fair comparison between studies. In the remainder of this paper, existing ArSLR solutions are mapped within this framework (see Figure 1) and used to indicate which pipeline steps are well-explored and which require urgent attention (e.g., segmentation methods for continuous ArSLR, creation of multimodal datasets, and language-aware postprocessing).

4 Taxonomies of approaches

A clear taxonomy helps to contextualize the diverse ArSLR efforts and compare methods across different datasets, tasks, and evaluation systems. We adopted two complementary taxonomies that together summarize the dominant design approaches in the literature: a primary taxonomy organized by capture mechanism (vision-based recognition—VBR—vs. sensor-based recognition—SBR) and a secondary taxonomy organized by task type (alphabet/fingerliterations, single words, continuous sentence recognition). These taxonomies reflect how researchers collect data and approach the recognition problem, and they also provide a basis for selecting appropriate models and evaluation protocols.

Table 1: Two basic methodologies utilized in sign language recognition

	Sensor	**Vision**
Data collection	Hands movements, finger motions, and trajectories	Videos, Images
Data collection tools	LMC, Kinect, EMG, Data gloves	Different type of cameras
Pre-processing data	No	Yes
extraction Features	Orientation, motions and 3D hand and finger flexes	Concise vectors that encapsulate essential information about hand gestures
Requiring computational power	Relatively less	High
Related work	[14],[15],[16]	[17],[18],[19]

4.1 Primary taxonomy — capture mechanism (Vision-based vs Sensor-based)

Vision-based recognition (VBR)

Vision-based ArSLR relies on cameras (single- or multi-view) and—in more recent work—RGB-D sensors and a pose estimation backend to extract appearance and geometric cues. VBR is attractive because it is non-invasive and naturally captures manual and non-manual signals (hands, face, torso) necessary for the ArSL grammar; however, it is also sensitive to illumination, background clutter, occlusion, and viewpoint variation [10][16]. Bare-hands (no wearable hardware) approaches are widely used for alphabetic and isolated word tasks because they easily scale to aggregation datasets, and modern deep neural networks (including transfer learning with VGG/ResNet/EfficientNet and vision transformers) provide high reported accuracy for static characters and short dynamic characters [23][30][31][35]. RGB-D cameras (e.g., Kinect) and multi-camera setups add depth or multiple perspectives, improving occlusion robustness and enabling precise 3D trajectory modeling; skeleton streams extracted from RGB or depth improve invariance to appearance differences by providing compact connectivity descriptors [75][33][88]. Practical VBR pipelines typically combine hand/face detection and tracking, spatial CNN encoders, and temporal models (LSTM/TCN/transformers) for sequential tasks. However, many high-accuracy results reported in the literature are signer-dependent or dataset-limited, highlighting the need for broader signer/dialect sampling and standardized evaluation partitions.

Sensor-based recognition (SBR)

Sensor-based approaches capture motion or contact signals directly from the signer using gloves, IMU sensors, EMG sensors, Leap Motion controllers, or other wearable devices. SBR provides precise measurements of finger flexion, orientation, and motion trajectory and can perform well in controlled environments with relatively small models and modest computational effort [89][88]. Glove-based systems are excellent at capturing fine details of hand shape and are valuable for technical or laboratory studies, but are invasive and less practical for large-scale body building or daily deployment. Similarly, EMG and IMU signals provide robust motion cues but require specialized hardware and calibration for each user. Hybrid systems that combine sensor streams with cameras attempt to combine the accuracy of SBR with the rich non-manual information available from VBR; however, hybrid datasets are rarer. Table 1 summarizes the relative trade-offs between sensor and vision approaches (data types, tools, pre-processing requirements, and representative related work).

4.2 Strengths, weaknesses, and practical tradeoffs across capture mechanisms

Practitioners choose a capture mechanism based on the intended use case. For large public datasets and field implementations, VBR (especially RGB with pose and skeleton extraction) is typically preferred due to its convenience and ability to capture non-manual signals. For precise hand shape classification or in domains where wearable sensors are acceptable (laboratory studies, sign linguistics research), SBR provides more detailed data on hand articulation. Importantly, because VBR can capture the face and torso, it is often the only viable option when modeling non-manual grammatical markers (e.g., questions, negations). Practical system design must therefore consider accuracy and accessibility, and benchmark reporting should indicate the modality, camera configuration, diversity of sign language users, and whether the assessments are user-dependent or user-independent to ensure interpretable and comparable results.

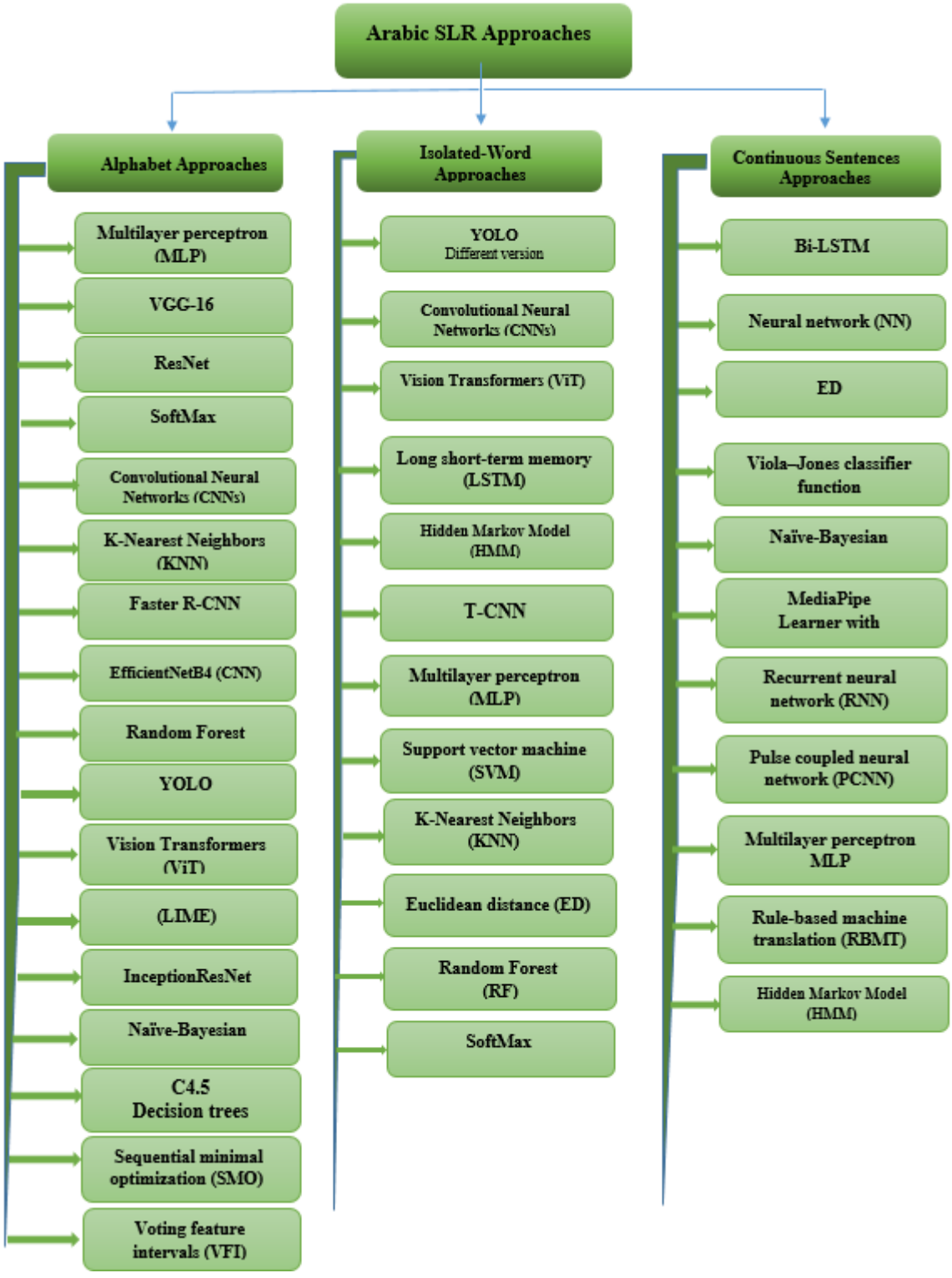

Figure 2: "Classification of ArSL recognition approaches" — here; the figure maps the secondary taxonomy and task categories and is used below to cross-reference Tables 2–4

4.3 Secondary taxonomy — task type (alphabet / isolated words / continuous)

The ArSL literature can also be usefully grouped by the **target recognition task**, which strongly influences dataset design, preprocessing, and model choices:

Alphabet / fingerspelling recognition

Alphabet tasks (static frames for learning fingerspelling) are often the first and most widely studied problem because the class space is limited and the number of gesture instances is relatively short and limited. Many successful systems use image-level CNN classifiers

(transfer learning with VGG/ResNet, AlexNet variants, or lightweight skeletons) and robust hand segmentation preprocessing; reported top-level accuracies on ArSL alphabet datasets (e.g., ArSL2018 and other RGB alphabets) often exceed 90%, and in some studies exceed 97–99% with intensive data balancing or signer-dependent partitioning [23][30][34][35]. Table 2 summarizes representative results on alphabet tasks for different methods and datasets, allowing direct comparison of input modality, model family, and recognition rate.

Isolated-word recognition

Isolated word tasks require modeling short temporal sequences and often rely on spatiotemporal feature extractors: 3D CNNs, CNN+LSTM stacks, or pose-keypoint pipelines combined with sequence models. The performance shown in Table 3 varies more than the alphabetic tasks because datasets differ in vocabulary size, signer variability, and segmentation quality; nevertheless, recent hybrid deep models (CNN+LSTM, temporal attention models) achieve high accuracy within the dataset (often >90%), especially for signer-dependent judgments [46][56][64]. Sensor-based systems (Gloves/Leap Motion) often generate competitive results for isolated words in small controlled datasets, but their generalization remains limited by data availability.

Continuous sentence recognition

Continuous recognition is the most challenging and least mature ArSL task: it requires robust segmentation, long-range temporal modeling, and language-aware post-processing. End-to-end sequential models (attentional encoder-decoder, CTC-trained models, transformer encoders/decoders) are currently the trend, but reported word error rates and BLEU/TER scores indicate significant room for improvement, and many published systems still rely on domain-restricted corpora or intensive manual matching [66][75]. Table 4 provides representative experiments with continuous tasks and shows that even state-of-the-art models produce non-trivial error rates when evaluated in signer-independent or larger-vocabulary conditions—highlighting this task as a priority research gap for ArSL.

4.4 Cross-cutting observations

Two cross-cutting patterns emerge from this taxonomy. First, modality matters: combining complementary streams (RGB appearance, depth geometry, skeleton position, and sensor signals) generally improves robustness but is underestimated because multimodal datasets are still relatively rare. Second, evaluation rigor matters: many reported high accuracies come from signer-dependent splits or small, homogeneous corpora; when signer-independent splits or cross-dialect tests are used, performance often drops significantly, revealing overfitting to capture conditions. Tables 2–4 and Figure 2 together clearly demonstrate these trends, linking tasks, modality, and model families to reported system performance and evaluation.

Consolidated SOTA comparison

A comprehensive SOTA table has been prepared by consolidating the experiments listed in Tables 2–4 of the original manuscript. The full machine-readable SOTA table is provided as `sota_table.csv` in the supplementary files package. SOTA Table consolidates representative ArSL recognition experiments from 2000–mid-2025. For each entry we list: reference, year, task, dataset, modality, model family, reported metric(s) and whether the reported evaluation used signer-dependent or signer-independent splits (when available). Where full details are provided in the original paper we use author splits; otherwise, we mark the split as ‘not specified’. The full machine-readable SOTA listing is provided in Supplementary Table S3 (sota_table.csv).

5 Alphabet recognition (vision & sensor approaches)

Alphabet recognition (finger spelling) is the most mature and widely studied subtask in Arabic Sign Language (ArSLR) recognition. Because alphabet characters are typically short, constrained, and have a fixed set of classes, they provide an easy entry point for both classical pattern recognition processes and modern deep learning classifiers. Early ArSL alphabet systems emphasized hand-crafted features and conventional classifiers (KNN, SVM, decision trees), while more recent work has predominantly focused on convolutional neural networks (CNNs), transfer learning, and lightweight detectors optimized for mobile or real-time use [22][26][30][35]. The availability of the extensive ArSL2018 image corpus (54,049 images in 32 classes from 40 sign characters) accelerated progress in deep learning, enabling robust training of CNN skeletons and transfer learning experiments [35][30][114]. Methodologically, alphabet recognition systems follow two dominant processes. The first is an image-level classification pipeline: hand detection/segmentation (optional), followed by a spatial CNN that maps a cropped hand image or full-frame image to one of the alphabet classes. Representative work in this category includes transfer learning studies that fine-tune VGG-16, ResNet-152, and other ImageNet pretrained networks on ArSL datasets, achieving the highest accuracy when the sampling and augmentation of the datasets are carefully tuned [23][30][114]. The second pipeline augments or replaces the raw images with pose/skeleton or depth features—either as compact input vectors for smaller networks or as auxiliary inputs for multimodal architectures—helping to reduce sensitivity to appearance (illumination, background, skin tone) and, in some cases, improve signer independence [33][79][88]. The chronological picture shows clear progress. Older studies using statistical classifiers or shallow machine learning classifiers on small datasets showed moderate accuracy (e.g., [22] reported a range of 50–90% for several classifiers with limited configurations). With the advent of larger annotated image corpora (notably ArSL2018 [35][30][114] and ArASL [88]), deep CNNs trained from

scratch, and especially transfer-learned models, have achieved significantly higher recognition rates: many works report accuracies in the mid-90s to high 90s on within-dataset test sets (for example, VGG-16 and ResNet152 fine-tuned on a downsampled ArSL2018 showed 99.4% and 99.6%, respectively, in [23]; several other CNN-based studies report 94–98+% depending on preprocessing and sampling strategies) [23][31][34][35][36][114]. These strong results reflect both improved models and the influence of large, relatively homogeneous datasets. However, the apparent jump in accuracy should be interpreted with caution. Many high-performance reports use signer-dependent splits, aggressive data balancing or resampling (subsampling/SMOTE), and controlled acquisition conditions that reduce real-world variability. Noor Azhar et al. [29] show how accuracy can drop when models are evaluated on images captured with a different camera or under different lighting conditions—91.1% on trained images versus 72.5% on newly acquired images—highlighting the gap between within-dataset performance and true generalization to new acquisition conditions. Similarly, studies that explicitly test signer-independent generalization or transfer between datasets show more modest results, indicating persistent overfitting to acquisition and signer conditions unless specific normalization or within-domain adaptation steps are taken [29][33]. Sensor-based approaches—Leap Motion, data gloves, and IMU/EMG systems—offer an alternative path to alphabet recognition by directly measuring hand pose and finger articulation. These approaches excel at precisely distinguishing hand shapes and often demonstrate very high recognition rates on small, inspected bodies [82][89]. For example, experiments based on Leap Motion and glove-based systems have demonstrated high accuracy in laboratory settings, and sensor fusion with a vision system can improve robustness to occlusions and complex backgrounds. However, the trade-off is practicality: wearable sensors reduce scalability for large body collection and hinder user adoption in everyday assistive applications where unobtrusive camera capture is preferred. Therefore, SBR remains a dominant tool in experimental studies and niche applications, rather than widespread implementation. Preprocessing and augmentation strategies significantly impact alphabet recognition results. Typical preprocessing includes resizing (usually to 64×64 or similar), color normalization, median filtering, and manual cropping; many studies convert grayscale input data to three-channel images or use background removal heuristics to reduce unwanted variances [34][35]. Data augmentation (random cropping, inversion, brightness/contrast jitter, synthetic occlusions) and class balancing (undersampling, oversampling SMOTE) improve reported results—a study of ArSL-CNN shows a statistically significant increase in accuracy after applying SMOTE [35]. Detector-based approaches (variants of Faster R-CNN and YOLO) have also been used for hand localization before classification, enabling real-time pipelines and mobile application deployment. Several YOLO-based systems demonstrate strong mapping and classification metrics when combined with robust backbones [28][40–45].

Recently, transformer-based vision models (ViT, Swin) and hybrid CNN-transformer pipelines have entered the ArSL literature. These architectures can be more parameter-efficient and more effectively capture global spatial context, helping to distinguish similar hand shapes. Alharthi and Alzahrani [114] and others describe competitive performance of ViT/Swin models on ArSL datasets, especially when initialized with ImageNet weights and tuned with moderate learning rates and regularization. However, the benefits of transformers are incremental for large, homogeneous image corpuses compared to robust CNN backbones and often depend on careful training recipes. Table 2 below consolidates representative alphabet recognition experiments from the literature. The table shows the range of recognition systems, input data sources, and reported recognition rates—illustrating both the maturity of image-based CNN methods and the variability introduced by the dataset composition and evaluation protocol.

Table 2: Performance comparison of researchers' efforts for alphabet ArSL recognition approaches

Ref.	Year	Recognition System / Techniques	Input Source / Dataset	Recognition Rate
[22]	2017	Compare C4.5, SMO, VFI, MLP, Naïve-Bayes	images/videos	50%, 66%, 75%, 80%, 90%
[23]	2020	VGG-16, ResNet152 (transfer learning)	ArSL2018 (undersampled)	99.4% (VGG-16), 99.6% (ResNet152)
[24]	2020	DBN + SoftMax / SVM	images	83.32%
[25]	2020	CNN	images	90%
[26]	2021	KNN	images	97.548%

[27]	2021	CNN (ArSL2018)	images	96.59% → 97.29% (with SMOTE)
[28]	2021	Faster R-CNN	images	93%
[29]	2022	CNN (mobile app)	trained images / new camera images	91.1% / 72.5%
[30]	2022	VGGNet	ArSL2018	97%
[31]	2022	EfficientNetB4	ArSL2018	95%
[32]	2023	CNN + RNN	proposed dataset, UCF-101	98%
[33]	2021	3D CNN skeleton + 2D point conv	ArSL (video/skeleton)	dynamic 98.39%, static 88.89% (dependent mode)
[34]	2022	CNN (ResNet50 + MobileNetV2 ensemble)	ArSL2018	97%
[35]	2021	ArSL-CNN (CNN)	ArSL2018	96.59% → 97.29% (with SMOTE)
[36]	2023	CNN	ArSL2018	94.46%
[6]	2021	Random Forest	ArSL symbols	92.15%
[7]	2023	CNN	ArSL2018	98%
[37]	2021	CNN (Yemeni SL)	Arabic alphabet dataset	94%

Synthesis and recommendations

Alphabet recognition is now largely a solved problem on limited datasets, with modern neural networks (CNNs) (and transformers) achieving very high accuracy across the dataset. Key challenges include generalization and implementation: (1) collecting multi-camera, multi-illumination, and multi-dialect alphabet datasets to assess their robustness; (2) reporting signer-independent and cross-set results as standard practice; (3) publishing accurate splits of training/testing and preprocessing pipelines to enable repeatable comparisons; and (4) exploring lightweight detectors and quantized models for on-device inference, while preserving privacy. Integrating backbone/depth streams or modest sensor fusion can improve robustness under occlusion and diverse backgrounds, but large-scale multimodal corpora for alphabets are still rare and should be a priority for the community.

6 Isolated-word recognition (vision & sensor) — review and comparative results

Isolated word recognition lies between the relatively limited task of alphabet recognition and the extremely challenging problem of continuous word recognition. Compared to alphabet recognition, isolated word recognition requires modeling short temporal dynamics (hand trajectories, initial/final motion), greater hand shape variability, and often bimanual or asymmetric bimanual coordination. At the same time, many practical datasets provide manually segmented word-level clips, which simplifies supervision and allows the use of segment-level training objectives (cross-entropy on fixed-length clips or collective features). Therefore, the literature offers a wide range of approaches—from classical spatiotemporal descriptors with SVM/HMM classifiers to modern, comprehensive deep models combining spatial encoders and temporal modules (3D CNNs, CNNs+LSTMs/GRUs, temporal convolutions, and attention/transformer-based sequential encoders). This section summarizes representative studies, compares vision-based and sensor-based approaches, discusses preprocessing and augmentation practices, and presents comparative performance results (Table 3).

6.1 Methodological families

Handcrafted spatio-temporal descriptors + classical classifiers

Early and some lightweight modern systems extract motion and shape descriptors (optical flow histograms, motion history images (MHI), HOG over spatiotemporal volumes, trajectory-based shape descriptors) and feed them into HMMs, SVMs, or random forest models. These methods perform well for small vocabularies and limited capture conditions, and remain attractive for low-resource implementations with limited computational budgets. Studies using such pipelines typically emphasize careful segmentation and hand-crafted normalization (body-centering, orientation-invariant features) to maximize inter-signator robustness [46][56].

Spatio-temporal deep networks (3D-CNNs, C3D, I3D)

3D convolutional networks naturally model short-term spatiotemporal patterns and are widely used in isolated word tasks. C3D and inflated 2D skeleton (I3D) networks learn common spatiotemporal filters that capture both appearance and motion signals without explicit optical flow computation. Several isolated word ArSL studies demonstrate strong performance on datasets using 3D skeletons, especially when combined with transfer learning from action recognition datasets or pretraining on multimodal corpora. A weakness of 3D models is their data starvation: they require a large number of labeled clips to avoid overfitting and are sensitive to temporal pruning rules during training and evaluation.

CNN + sequence models (LSTM/GRU/TCN)

A common architecture is a two-dimensional spatial CNN encoder applied to each frame (or concatenated windows of short frames), followed by a sequence model such as LSTM/GRU or temporal convolutional networks (TCN). This decomposition uses pretrained 2D skeletons and is effective for moderate dataset sizes. Many ARSL studies use this pattern and report that bidirectional LSTMs or stacked temporal layers improve accuracy for characters with subtle temporal features (e.g., slight orientation changes or repetitive motion patterns). Such hybrid models are also easier to regularize and fine-tune than fully three-dimensional networks.

Pose/skeleton-based methods and graph neural networks (GNNs)

Skeletal or pose streams reduce appearance variability by relating the signer to the coordinates of the hand joints and keypoints. Graph convolutional networks (GCNs) and temporal GCNs (T-GCNs) treat joint sequences as spatiotemporal graphs and demonstrate high performance on isolated-word tasks, especially when visual appearance is highly variable (different clothing, backgrounds, or lighting). Skeletal approaches are attractive for signers and privacy-sensitive applications because they avoid high-resolution facial images. However, the quality of skeleton extraction (especially hand joints) is crucial; coarse pose estimators that do not capture finger articulations can limit the recognition of fine-grained hand shapes.

Multimodal fusion and sensor-based pipelines

Multimodal fusion (RGB + depth + skeleton or RGB + Leap Motion/gloves) combines complementary cues: appearance (RGB), 3D geometry (depth), moving pose (skeleton), and precise finger articulation (gloves/Leap Motion). Sensor-based systems that utilize gloves or Leap Motion often report very high accuracy on isolated words in small controlled datasets because they directly capture fine-grained articulation [82][89]. Fusion systems—where practical—often outperform unimodal databases, but the scarcity of large, aligned multimodal corpora limits overall adoption. Hybrid fusion strategies (early feature concatenation, mid-level attention-based fusion, late-score fusion) have been investigated with varying degrees of success; attention-based fusion, which dynamically weights modalities, tends to be more robust when some streams are degraded (occlusion, noise).

6.2 Preprocessing, segmentation, and data augmentation

Isolated word pipelines typically rely on robust hand detection and temporal pruning to ensure the model primarily sees the sign token. Preconditions such as crop center normalization, temporal resampling (uniform length through interpolation or frame skipping), and geometric normalization (rotation, scaling) are commonly used to reduce variance between signers. Data augmentation methods (temporal jitter, synthetic occlusion, spatial perturbations, color jitter) are particularly helpful for moderate-sized datasets. For skeleton- and pose-based methods, normalization to body-centered coordinates and scaling to joints facilitates generalization across signers. Importantly, authors must provide precise segmentation and pruning policies—many performance differences between studies are due to different cropping and temporal normalization policies, not to differences in models.

6.3 Evaluation regimes and common pitfalls

As with alphabet recognition, an important caveat is that reported accuracies are sensitive to the evaluation methods. Signer-dependent splits (random splits at the video level that allow the same signer to appear in training and test) routinely overestimate generalization. Signer-independent splits and cross-set evaluations reveal more realistic performance and often exhibit significant degradation. Other pitfalls include inconsistent reporting of vocabulary size, handling of class imbalance, and whether evaluation is performed for each clip (segment-level accuracy) or for each instance after smoothing the language model. Therefore, when interpreting comparative performance, we prioritize studies that report signer-independent metrics or cross-set transfer results.

6.4 Representative results

Below, we present Table 3, which consolidates experiments on isolated word recognition across modalities and model families. The table highlights the

diversity of datasets, model types, and reported recognition rates in the ArSL literature.

Table 3: Performance comparison of researchers' efforts for isolated-word ArSL recognition approaches.

Ref.	Year	Recognition System	Techniques Classification	Dataset used	Recognition Rate
[46]	2024	vision-based	Ov8, (CNN), (LSTM), and a hybrid CNN-LSTM	ArabSign	% f or YOLOv8, 95.23% for the CNN model, 88.09% % for the LSTM model, and 96.66% for the hybrid model.
[47]	2020	Sensor-based	CNN	Own dataset	90%
[48]	2015	Sensor-based	Hidden Markov Model (HMM)	Own dataset	achieving an accuracy of 80.47%. Singer-independent experiments resulted in an average recognition accuracy of 64.61%.
[55]	2017	vision -based	CNN and Softmax layer	200 videos	Greater than 90%
[56]	2017	vision -based	KNN, SVM, and MLP	Video	98.8 to 99%
[57]	2017	vision -based	ED	450 videos	97%
[58]	2018	Sensor-based	Nearest Neighbour	1200 samples	97.58% for signerdependent and 95.25% for signerindependent
[59]	2018	vision -based	RF	150 videos	55.57%
[60]	2020	Sensor-based	KNN and SVM	30 hand gestures	92.3% for single-hand and 93% for double-hand
[61]	2020	Sensor-based	SVM, RF, KNN	gesture words	83 % for SVM
[62]	2020	vision -based	CNN	Alphabets and word gestures	90%
[63]	2020	vision -based	deep bi-directional LSTM network	3450 videos	89.5% using DeepLabv3+ hand segmentation, 69.0% without hand segmentation
[64]	2021	vision -based	LSTM and Softmax	6748 videos	99.7% for signerdependent and 72.4% for signer-independent modes
[65]	2021	vision -based	CNN and Bidirectional LSTM	ArSL, Jester, and NVIDIA Gesture datasets	85.6%, 95.8%, and 86.6% for ArSL, Jester, and NVIDIA Gesture datasets respectively
[66]	2021	vision -based	2DCRNN and 3DCNN	224 videos	92% for 2DCRNN and 99% for 3DCNN
[67]	2023	vision -based	CNN and RNN	8,467 videos	98% and 92% on validation and testing
[68]	2023	vision -based	A fully connected layer with a Softmax	100 videos	99.74% and 68.2% in signer-dependent and independent modes

6.5 Synthesis and recommended directions

Isolated word recognition has advanced significantly: modern deep spatiotemporal models, backbone GCNs, and multimodal fusion methods often achieve high accuracy within a single dataset, and sensor-based methods can achieve near-perfect performance under controlled conditions. However, the prevailing limitations are analogous to those in other ArSL subdisciplines: (1) many datasets are small, geographically limited, or signer-dependent; (2) multimodal corpora with diverse numbers of signers are rare; (3) temporal segmentation principles and preprocessing choices vary across studies, complicating comparisons; and (4) sensor-based solutions, at the expense of practicality, are less suitable for large-scale assistive implementations. To move this field forward, we recommend (a) building larger, multimodal corpora of isolated words characterized by signer and dialect diversity, (b) standardizing segmentation and temporal sampling protocols, (c) routinely reporting experiments with signer-independent and cross-set data transfer, and (d) exploring lightweight fusion strategies that preserve the advantages of sensors without mandatory portable hardware (e.g., short-term calibration procedures or optional sensor-assisted bootstrapping combined with RGB-only inference). These steps will make reported results on isolated words more informative in terms of real-world performance and more useful for further continuous recognition research.

7 Continuous sentence recognition

Continuous sentence recognition (CSR) is the most challenging and least developed area within Arabic Sign Language (ARSLR). Unlike tasks with the alphabet or isolated words, CSR must handle unsegmented streams where signs are intermingled (coarticulation), where temporal boundaries are ambiguous, and grammatical meaning often depends on simultaneous non-manual cues (facial expressions, head movements, torso shifts). The combination of segmentation uncertainty, long-range temporal dependencies, and the need for language-aware decoding shifts this task from a sequence classification problem to one that combines sequence labeling, intersequence translation, and structural prediction. The following sections organize the main challenges, summarize methodological families and representative studies, discuss popular evaluation metrics and practices, and contextualize Table 4 (comparative results for continuous tasks).

7.1 Core challenges

Co-articulation and blurred sign boundaries. In natural signing, the transitions between successive signs are fluid: handshapes change, trajectories overlap, and prosodic cues are dispersed within signs. Co-articulation complicates the definition of discrete "start" and "end" and complicates naive frame labeling. As a result, segmentation errors are often the dominant source of recognition errors in later text: poorly located boundaries cause feature misalignment with labels and reduce the effectiveness of supervised sequence models.

Trade-offs between segmentation and end-to-end modeling. Historically, researchers approached CSR by first segmenting streams into potential character units (using energy/motion heuristics, hand activity detection, or learned boundary detectors) and then classifying each segment. Although modular pipelines simplify training, they propagate segmentation errors and complicate end-to-end optimization. Current trends favor end-to-end models that implicitly learn segmentation alongside recognition via CTC losses, attention-based encoder-decoder architectures, or transformer models; however, these approaches require significant amounts of data and careful regularization to avoid overfitting.

Temporal Modeling and Long-Range Dependencies. CSR requires capturing dependencies across multiple frames – reference tracking, indexing, and spatial grammar (referencing entities in sign space and then referencing them). RNN/LSTM-based models capture short- and medium-range dependencies, while transformers and temporal convolutional networks (TCNs) offer alternatives better suited to very long sequences. However, effective long-range modeling typically relies on large corpora with aligned annotations at the gloss or sentence level, which are rare in ArSL.

Non-Manual Signals and Multi-Channel Synchronization. Many grammatical distinctions in ArSL rely on facial and torso signals occurring simultaneously with manual articulations. Models that ignore non-manual channels risk systematic errors (e.g., misclassifying a sign question as a declarative statement). Capturing and recording these channels increases the complexity of the data and annotations, but omitting them reduces linguistic coverage.

Data scarcity and annotation costs. CSR training requires precise voting or frame-level matching for supervised learning – expensive to implement because annotators must be proficient in voting and adhere to consistent voting conventions. The lack of large, multi-dialect, continuous corpora for ArSL limits the practical application of complex, data-intensive models and reinforces the need for knowledge transfer, data augmentation, and semi-supervised/self-supervised learning strategies.

7.2 Methodological families

Segmentation and recognition pipelines (modular). Traditional CSR systems employ explicit segmentation (motion/energy detectors, HMM-based boundary models, CRFs) followed by segment classification (HMMs, SVMs, or segment-level CNNs). These pipelines benefit from interpretable error sources and easier supervision in data-poor systems, but their modularity hinders end-to-end optimization and recovery from boundary detection errors. Several ArSL studies have applied this approach to limited corpora where manual segmentation is feasible. [66][75].

CTC-based approaches and weak alignment. Connectionist Temporal Classification (CTC) provides a fundamental method for training sequence models without frame-level matching by summing all valid matches between input frames and label sequences. CTC has been effectively applied to speech and sign recognition when voice-level transcripts exist but precise frame boundaries are lacking. ArSL experiments using CTC (often with CNN encoders + RNN/temporal decoders) are promising, but their performance depends on the availability of diverse training data and language modeling during decoding to disambiguate likely voice sequences.

Seq2Seq with attention and transformer architectures. Attention-based encoder-decoder models (and more recently, full transformer encoder-decoder stacks) learn mappings from frame sequences (or encoded feature sequences) to gloss or sentence tokens. Attention enables implicit matching and can focus on temporally relevant regions for each output token. In ArSL, transformer-based models are attractive for modeling long-range dependencies and for combining inputs from multiple channels (RGB, depth, skeleton, facial framing). However, transformers typically require large-scale pretraining or transfer from related domains (e.g., action recognition, large-scale video-text pairs), and in many ArSL applications, only modest gains are achieved without such pretraining. [15][114].

Hybrid approaches and multi-task learning. Some systems combine explicit boundary detectors with end-to-end sequence models or jointly learn gloss prediction and

auxiliary tasks (character boundary detection, handshape classification, non-manual signal detection). Multi-task signals can regularize models, improve interpretability, and reduce the need for exhaustive annotations for each subtask. Such strategies are useful in ArSL, where some annotations (e.g., coarse sentence transcriptions) are more readily available than detailed frame labels.

Pose- and graph-based temporal models. When backbone or keystone streams are available, graph neural networks (GCN/T-GCN) applied to the temporal joints provide a compact, high-level representation that emphasizes articulatory structure. Position-based models are particularly useful for signer-independent and privacy-preserving implementations because they abstract from appearance; however, they struggle when details about the hand and fingers (essential for many signs) are not reliably extracted. Position-based transformers and spatiotemporal neural networks (GCNs) are becoming increasingly common in ongoing ArSL experiments.

7.3 Post-processing, language modeling and translation

CSR systems often require a language model (LM) for decoding to transform ambiguous voice sequences into well-formed sentences. LM models trained on voice sequences or on the Arabic script (after fingermapping) can be used for decoding by beam search, n-best rescoring, or neural post-editing of the sequences. For systems that aim to directly translate ArSL into Arabic text (rather than just the voices), a translation module—often another seq2seq model—maps the voice sequences into fluent Arabic sentences. Such translation models require parallel corpora of voices and text and have their own evaluation metrics (BLEU, METEOR), in addition to WER and accuracy at the voice level used for recognition.

7.4 Metrics and evaluation practices

Because CSR produces variable-length sequences, standard evaluation metrics extend beyond per-class accuracy. Commonly used metrics include:

- **Word Error Rate (WER)** and **Gloss Error Rate (GER)** (edit distance normalized by reference length) for sequence-level correctness.
- **BLEU** or other translation metrics when systems produce natural language sentences.
- **Segment-level accuracy** (for segmented corpora) when boundaries are available.
- **Detection metrics** (precision/recall/F1) for boundary detection modules.
- **Latency and real-time factor** for deployment assessments.

Crucially, evaluations should report signer-independent splits, cross-dialect tests (when possible), and explicit preprocessing/segmentation rules. Many ArSL studies report promising within-dataset WERs but degrade substantially under signer-independent or cross-corpus conditions, highlighting the need for standardized, harder evaluation splits to measure true generalization [66][75].

Table 4: Performance comparison of researchers' efforts for continuous-sentence ArSL recognition approaches

Ref	Year	Recognition System	Techniques Classification	Input Source	Recognition Rate
[49]	2010	vision -based	Naïve Bayes based on spatio-temporal feature extraction and hidden Markov models	Own dataset Sharjah City for Humanitarian Services (SCHS) [11]	94%
[50]	2023	vision -based	Neural Network	Own dataset Arabic–Arabic sign gloss	with a training accuracy of 94.71% and an 87.04% testing
[51]	2017	vision-based	Euclidean distance classifier (ED)	ArSLRS	recognition rate of 97%
[52]	2024	vision -based	Bi-LSTM	ArabSign	word-recognition rates of 99.8% and 75.3%, respectively
[53]	2014	Sensor-based	Viola–Jones classifier function	SignsWorld Atlas	Dependent =97% and independent =95.28%, respectively.
[54]	2025	vision -based	NN model based (LSTM)	Own dataset	99.80% accuracy during training and 99.40% in testing, with overall accuracy, recall, and F1-score metrics above 99%.
[69]	2010	vision -based	HMM	Video frames	75%.
[70]	2012	vision -based	MLP	Video	Until 70%

[71]	2013	vision -based	PCNN	Video	80%
[72]	2018	vision -based	HMMs	Videos	99.11%
[73]	2020	Sensor-based	RBMT	Images	more than 80%
[74]	2022	vision -based	CNN Softmax	Videos	99%
[75]	2023	Sensor-based	encoder-decoder model Pretrained CNN	sentence by Kinect	WER of 0.50

7.5 Synthesis and recommended directions

Continuous ArSL recognition remains an open challenge for the following reasons: (1) co-articulation and ambiguous boundaries require robust segmentation or powerful end-to-end alignment-capable models; (2) long-range dependency modeling benefits from transformer-style architectures but demands more data or transfer-learning; (3) non-manual channels are linguistically essential yet under-captured in many datasets; and (4) annotation scarcity increases reliance on weakly supervised, semi-supervised, or self-supervised learning strategies.

Based on the surveyed literature and the comparative results in Table 4, we recommend a research agenda that emphasizes: (a) creation of multi-dialect, multimodal continuous corpora with gloss- and sentence-level transcripts; (b) combined use of pose-based and appearance-based encoders with attention-based fusion; (c) pretraining on large-scale related video corpora (action recognition, video–text datasets) followed by fine-tuning on ArSL data; (d) routine reporting of signer-independent WERs and cross-corpus transfer experiments; and (e) incorporation of explicit non-manual detection and language modeling during decoding to improve syntactic and pragmatic translation quality. Pursuing these directions will be essential to move ArSL CSR from constrained laboratory prototypes toward real-world, usable systems that can support Arabic-speaking deaf communities in communication and education.

7.6 Discussion

This Discussion compares the surveyed methods by generalization, signer independence, modality robustness, and deployment readiness. Key points:

1. Alphabet tasks — Image-level CNNs (VGG, ResNet, EfficientNet) and transformer backbones reach very high within-dataset accuracies (>94–99% on ArSL2018 and similar corpora), but most of these results are signer-dependent or within-dataset splits; cross-camera and signer-independent tests often reduce accuracy considerably.

2. Isolated words — 3D-CNNs, CNN+LSTM stacks and skeleton-GCNs perform well for segmented clips; however, performance is highly variable with signer-independent splits, indicating sensitivity to signer sampling, preprocessing and segmentation policies.

3. Continuous recognition — End-to-end CTC/attention/transformer models are promising but remain constrained by the scarcity of large, multimodal, frame-aligned continuous corpora.

4. Modality & fusion — Multimodal fusion (RGB + depth + skeleton + sensors) improves robustness where available, but multimodal continuous datasets at transformer/pretraining scale are still absent.

Implications. To assess algorithmic progress, we recommend canonical signer-independent splits, consistent preprocessing documentation (hand crop policy, temporal resampling), and reporting of at least: top-line accuracy, per-class recall, confusion matrices, signer-independent results and latency/model size for deployment claims.

8 Datasets and benchmarks

ArSL research is constrained primarily by dataset availability, annotation granularity, and modality coverage. Over the last decade the community has produced useful corpora spanning static alphabets, isolated-word video clips, multimodal Kinect-style recordings, and a handful of sensor-based collections, but the landscape remains fragmentary: many corpora are single-site, camera-only, and limited either in signer diversity or in transcript granularity—shortcomings that limit progress on signer-independent and continuous-sentence tasks [79][88][35].

The complete dataset inventory is available as Supplementary Table S2 (file: Supplementary_Table_S2_DatasetInventory.csv), captioned: *"Supplementary Table S2 (Table 5): Dataset inventory — release year, modality, approximate sample/signers counts, typical tasks, reported evaluation splits, and availability/license/consent metadata."*

8.1 Modalities and representative corpora

Current ArSL datasets cluster into three modality groups.

• **RGB (static image) datasets.** The largest publicly circulated corpus is ArSL2018 (54,049 images, 32 classes, 40 signers), which has catalyzed the recent wave of CNN and transfer-learning studies on Arabic alphabet recognition [35][34][31]. Other RGB alphabet datasets of smaller scale (several thousand images) and various capture conditions are also listed in Table 5; these are convenient for training spatial classifiers but lack temporal and 3-D geometry cues needed for continuous or fine-grained handshape tasks [36][91].

• **RGB-D / skeleton video datasets.** A small but growing set of corpora recorded with Kinect-style sensors supply synchronized RGB, depth and skeleton joint streams (for example the ArabSign / ArabSign-benchmark family introduced by Luqman et al.). These multimodal continuous datasets are particularly valuable because depth and skeleton streams permit 3-D trajectory modeling and pose-based methods; however, this highlights that available RGB-D continuous corpora are still modest in scale relative to what modern sequence models (e.g., transformers) typically require for robust generalization [75][73][74].

• **Sensor (wearable) datasets.** Leap Motion, data-glove and IMU/EMG datasets provide high-fidelity measurements of finger flexion and hand orientation and therefore perform very well for fine handshape discrimination in controlled studies [82][81]. The downside is that wearable datasets are typically small, intrusive, and not widely shared, limiting their role as community benchmarks for signer-independent evaluation.

Ethical metadata, licensing and de-identification. For responsible dataset sharing we recommend that every dataset release include a minimal metadata checklist: availability URL/DOI, license, consent statement (written informed consent for recording and sharing), signer demographics (age range, gender, dialect/region), capture hardware and settings (camera model, fps, resolution) and any de-identification steps applied (face blur, skeleton-only release). When full visual release is not possible for privacy reasons, authors should consider releasing skeleton/keypoint data or blurred video versions alongside an access request policy. Table 5 (and Supplementary Table S2) now records availability and license/consent fields where the information was publicly available; entries marked ‘Not specified’ indicate fields that require confirmation from dataset authors.

Table 5 (full dataset inventory provided as Supplementary Table S2: `Supplementary_Table_S2_DatasetInventory.xlsx`): the table lists release year, modality, approximate size and signer counts, evaluation splits, availability and license/consent metadata where publicly reported.

Table 5: “Publicly available ArSL datasets for Arabic Sign Language”

Ref	Year	Dataset Description
[80]	**2011**	Six gestures are used to generate 6,000 different sign images.
[81]	**2012**	The 80-word vocabulary used to construct 40 sentences with no restrictions on syntax or sentence length; this process repeated 19 times.
[77]	**2013**	There are 270 postures that make up the 200 gestures, with 189 postures involving two hands and 81 postures comprising only one hand. Every gesture carried out ten times, each time by a different two person.
[82]	**2014**	There are a total of 2800 frames in the dataset generated from a single user's input of 28 alphabets, with 10 samples of each letter.
[73]	**2015**	The database contains about five hundred static gestures, including "finger spelling, hand movements" (non-manual signs). Lip reading, body language, and facial expressions all play significant roles.
[84]	**2016**	Two sets of static alphabet data exist: 700 instances for each 28 characters written with naked hands and colored gloves.
[78]	**2017**	200 samples taken from the unified ArSL lexicon, with each of the 25 signs being performed by two different signers four times. 125 for training and 75 for testing
[85]	**2018**	Thirty people are serious mobile photographers. Volunteers gesture these 30 ArSL alphabets. There are 900 images spread across 30 letters.
[86]	**2018**	Captured 450 colorful ArSL videos
[87]	**2019**	28 Arabic letters and numerals (0-10) represented by 7869 images for recognition.
[88]	**2019**	The dataset ArSL2018 comprises a collection of 54,049 images, which accurately depict the 32 alphabets and signs of ArSL. These images have been donated by a group of 40 signers.
[89]	**2020**	A total of 44 signs (29 single-handed and 15 double-handed) are executed by a group of 5 signers, where 80% are used for training and 20% for testing.
[76]	**2021**	There are 9240 images of the Arabic alphabet from 10 places and age groups. These images organized in four separate datasets.
[79]	**2021**	There are eleven chapters totaling 502 signs that make up the words in the ArSL lexicon. Three signers used for each sign. There are 75300 samples, the result of 50 repetitions of each sign by each signer.
[90]	**2021**	There are a total of 220000 images in the dataset, split amongst 44 different classes (32 letters, 11 digits (0-10) and 1). There are 5000 images total, taken by various people, of each of the stationary signs.
[91]	**2023**	It contains 7,856 RGB images of ArSL alphabets. Data collected from over 200 people in a wide range of shooting situations (including but not limited to: lighting, background, image orientation, size, and resolution).

8.2 Dataset quality: annotations, balance and metadata

A cluster of quality issues recurs across ArSL corpora:

1. **Annotation granularity.** Many RGB corpora provide only image-level labels (suitable for alphabet tasks) while continuous datasets sometimes provide only coarse segment-level transcripts rather than frame-level alignments. Fine-grained frame or gloss alignments (essential for training CTC and attention models) are rare and costly to produce, constraining end-to-end continuous training [75][74].

2. **Class balance and per-class reporting.** Several datasets show heavy class imbalance; yet many papers report only overall accuracy instead of per-class recall or confusion matrices. We therefore recommend that dataset publishers and authors routinely report macro-averaged metrics and per-class statistics to make skew issues visible [35][27].

3. **Metadata completeness.** Important capture metadata (camera make/model, resolution, fps, capture geometry, lighting conditions), signer demographics (age, region/dialect, handedness) and consent/licensing details are often missing or incomplete in older releases. This lack of standardized metadata reduces reproducibility and complicates cross-dataset comparisons [79][35].

8.3 Gaps identified (evidence-based)

From the survey in Table 5 and our analysis, four critical gaps stand out.

1. **No large, public, multimodal continuous corpus at transformer scale.** ArabSign and related multimodal releases are a major step forward, but their size remains small relative to the pretraining data requirements of modern sequence models; more and larger multimodal continuous datasets are required to close the gap between promising lab models and robust real-world CSR systems [75][73][74].

2. **Limited high-fidelity hand/finger annotation in RGB-D corpora.** Many "RGB-D" datasets provide coarse joint estimates—insufficient to discriminate subtle finger configurations—so GCN/pose approaches are bottlenecked by annotation fidelity rather than model design [60][33].

3. **Geographic/dialect narrowness and signer-dependence.** Numerous datasets were collected at single institutions or within a single country; consequently, models evaluated on these corpora often show degraded cross-dialect or signer-independent performance, a weakness documented across multiple evaluation reports[79][64].

4. **Sparse ethical/privacy metadata.** Camera-based corpora frequently include face imagery yet often lack standardized consent/usage metadata or de-identification guidance—an omission we flag as an urgent ethical and legal concern for open benchmarks and downstream deployment [35][79].

8.4 Recommended benchmarking practices

To make ArSL benchmarks comparable and practically informative we prescribe the following minimal standard (to be included with any dataset or experimental release):

- Publish full capture metadata (camera specs, fps, lighting), signer metadata (region/dialect, gender, handedness) and clear consent/licensing statements. [35][79]

- Provide canonical splits that include signer-independent and cross-dialect test sets. [75][79]

- Supply annotation documentation (gloss conventions, non-manual tags, boundary rules) and at least one frame-aligned subset for continuous corpora. [74][35]

- Report per-class recall, confusion matrices, WER/GER for sequences, and runtime/latency for deployment claims. [35][75]

- Release baseline training scripts and seeds for reproducibility (e.g., simple CNN, skeleton-GCN, CNN+LSTM, and a transformer baseline). [35][75]

8.5 Short- and medium-term dataset priorities

We conclude by prioritizing three dataset investments that would most accelerate ArSLR:

1. **A large, multimodal continuous ArSL corpus** (RGB + depth + reliable hand/finger tracking + face crops) sampled across multiple Arab regions with both gloss- and sentence-level transcripts (scale comparable to hundreds of hours). ArabSign demonstrates the utility of this design but larger, multi-site efforts are needed. [75][74]

2. **High-fidelity hand capture (close-up hand cameras or calibrated multi-sensor rigs)** so that

pose/GNN methods can exploit finger-level cues without resorting to intrusive wearables. [60][82]

3. **Shared benchmark infrastructure and leaderboards** with clear modality tracks (RGB-only, RGB-D, skeleton, sensor-fusion) and canonical signer-independent splits to measure genuine generalization gains. [35][75]

9 Open issues, challenges & taxonomy

Arabic Sign Language Recognition (ArSLR) faces four tightly interlinked challenge clusters—Environment, Language, System, and Gesture—that together determine whether research advances will translate into robust, deployable systems; Figure 3 summarizes these clusters and their sub-issues, and the following executive summary condenses the taxonomy into prioritized, actionable recommendations. Environmental issues (lighting, background clutter, camera viewpoint, occlusion and sensor noise) require that datasets and experimental reports always include capture metadata (camera model, fps, lighting conditions, view angles) and encourage multi-view or multimodal capture where possible; Language-level problems (dialectal variation, non-manual signals, coarticulation and inconsistent glossing) demand explicit dialect labels, joint modeling or annotation of non-manual markers, and standardized glossing conventions to enable cross-dataset comparability. System-level weaknesses (small or homogeneous datasets, expensive frame-level annotation, inconsistent reporting, and missing consent/license metadata) point to three immediate steps: (1) adopt canonical signer-independent splits and publish them with every dataset, (2) include a minimal metadata checklist (availability, license, consent status, signer demographics, capture specs) in dataset releases and Table 5, and (3) require papers to report both accuracy and robustness metrics (per-class recall, confusion matrices, WER/BLEU for continuous tasks) plus deployment metrics (latency, model size).

Backhand signs: dataset evidence & capture recommendations. We confirm that backhand / orientation-sensitive signs are underrepresented across existing ArSL corpora and that their omission reduces orientation robustness in deployed models. To address this we recommend a small, shareable "backhand pilot" dataset: capture ≥10 diverse signers, multi-view camera configuration including ±45° side views plus a close-up hand camera, record at ≥60 fps if possible, and annotate per-frame orientation flags (e.g., palm/backhand/side) and occlusion markers. Optional wrist IMU data can be added to disambiguate orientation under occlusion. Release a public subset with canonical signer-independent splits and explicit orientation metadata to bootstrap orientation-robust model evaluation.

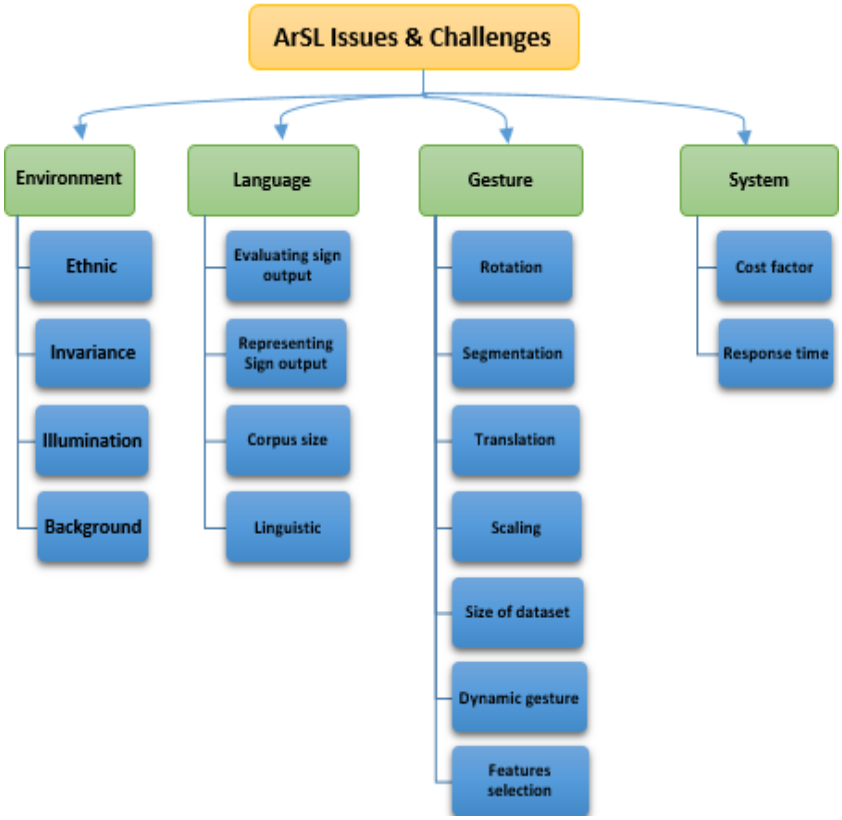

Figure 3: "Taxonomy of ArSLR open issues & challenges"

Figure 3 Taxonomy of key challenges in Arabic Sign Language Recognition: four categories (Environment, Language, System, Gesture) with representative sub-issues and practical implications. Legend: Environment—lighting, viewpoint, occlusion; Language—dialects, non-manual signals, coarticulation; System—dataset scale, annotation, evaluation; Gesture—handshape, orientation/backhand, dynamics. See main text Sections 3–9 for discussion and dataset cross-references.

9.1 Environment (capture & deployment conditions)

Environmental factors determine what signals are available to recognition systems and strongly influence their robustness.

Illumination, background, and viewpoint. Illumination variation, cluttered backgrounds, and non-frontal viewpoints complicate hand detection and image-based feature extraction. RGB datasets collected on phones (e.g., ArSL2018) facilitate handwriting tasks but remain sensitive to lighting and viewpoint changes [35]. Depth and skeleton streams (Kinect-based datasets such as ArabSign and Elpeltagy et al.) mitigate appearance sensitivity to some extent, but many RGB-D corpora are small-scale and provide coarse fidelity for hand joint representation [75][59]. Practical systems must therefore account for domain changes (differences between camera and hardware) through augmentation, domain adaptation, or multimodal fusion.

Occlusion and Multi-Person Scenes Hand occlusion, self-occlusion of fingers, and partially visible faces (e.g., during direction changes) are common in real-world signing and reduce classifier confidence. Multi-person or crowded spaces further complicate signer isolation. Sensor fusion (hand proximity cameras, depth) and reliable multi-object tracking mitigate these issues but increase recording complexity and reduce scalability [82][60].

Latency, computation, and boundary constraints. Real-time assistive applications (mobile or wearable) require low-latency inference and compact models. Many state-of-the-art architectures (transformers, large 3D-CNNs) are computationally intensive; therefore, efficient frameworks, pruning/quantization strategies, and benchmarks that report latency and model size in addition to accuracy are needed in this area [35][75].

Privacy and ethics. Camera-based capture raises privacy concerns, especially when capturing facial images. We document inconsistent consent and anonymizing metadata across corpora and call for unified ethics reporting and privacy-preserving options (framework-only benchmarking, on-device inference) to be publicly shared and implemented. [35].

9.2 Language (linguistic structure & annotation)

The linguistic properties of ArSL impose requirements beyond a strict classification of gestures.

Manual and non-manual channels. ArSL encoding relies on simultaneous channels: manual components (hand shape, movement, location, orientation) and non-manual markers (facial expression, gaze, head/body posture), which convey grammar and prosody. Many datasets emphasize hand imagery but underrepresent systematic non-manual annotations, which weakens performance on grammatical distinctions (e.g., questions, negations) dependent on facial/body signals [12][95]. Robust ArSLR requires corpora that capture synchronized facial and upper body data and annotation conventions that label these signals.

Dialectal variation and lexical divergence. The Arabic world contains regional sign varieties with real lexical differences. Single-site corpora generate models that perform well within distributions but generalize poorly across regions. We emphasize the need for multi-regional sampling and cross-dialect testing to reveal real gaps in generalization [79][75]. Voting harmonization is also essential: inconsistent voting schemes prevent corpora from being merged and hinder transfer learning across datasets.

Coarticulation and Segmentation. Continuous corpora are characterized by coarticulation and fuzzy character boundaries, which complicate boundary annotation and model supervision. The taxonomy highlights the tension between modular segmentation and recognition pipelines (which can propagate boundary errors) and comprehensive approaches (CTC, attention), which require large annotated corpora or clever weak supervision strategies [17][114]. Better-adapted continuous corpora and annotation tools are therefore a priority.

Language Modeling and Translation. Translating sign language voices into well-formatted Arabic text requires additional linguistic modeling (morphosyntax, script mapping for the fingerspelling alphabet). Effective pipelines therefore combine recognition with decoding based on a language model and often require parallel corpora of voices and text to ensure high translation performance; e.g. [64][74].

9.3 System (data, benchmarks, evaluation, and reproducibility)

System-level issues determine whether reported research yields replicable progress and real-world systems.

Benchmarking and Canonical Partitions: We document inconsistent evaluation protocols and the sparse use of signer-independent or cross-dialect partitions. Without canonical partitions and baseline implementations, comparison methods are unreliable; the taxonomy therefore emphasizes that standardization of benchmarking (canonical partitions, modality paths, result tables) is essential for scientific progress [35][75].

Annotation Standards and Tools: High-quality frame-level and gloss-level annotations are expensive but essential for supervised CSR. The taxonomy requires common annotation schemes (including non-manual tags), tools to accelerate consistent labeling, and at least a frame-aligned subset of each continuous corpus to enable supervised alignment methods [75][59].

Data Imbalance and Long-Tail Classes Many datasets exhibit skewed class distributions; rare characters are underrepresented and often poorly recognized. System-level responses include dataset curation (balanced sampling), metric hygiene (reporting macrometrics and recall for each class), and algorithmic solutions (learning with small sample sizes, balanced losses for each class). We repeatedly recommend reporting for each class to detect skew effects [35].

Reproducibility, Code, and Baselines. The taxonomy emphasizes publishing training scripts, seeds, and reference implementations for the underlying models (CNN, skeleton-GCN, CNN+LSTM, transformer) so that the results tables reflect repeatable improvements rather than hidden experimental tricks. [35][75].

9.4 Gesture (articulatory complexity & variability)

Gesture-level phenomena shape what models must distinguish.

Bimanual and asymmetric signs. Many ArSL signs use two hands in asymmetric roles (dominant and non-dominant). Recognition architectures must model interhand relationships and temporal synchronization; simple single-hand recognition systems miss crucial information for such signs [33][16].

Concealed and closed-hand configurations. We highlight an underexplored gap: covered signs and other orientations that reveal the dorsal surfaces of the hands or complex finger articulations are rarely captured in existing corpora. These configurations generate different body/appearance cues and require either improved multi-point capture or proximity sensors/cameras of the hands to accurately recognize finger articulations. Accounting for covered signs is crucial for linguistically complete ArSL coverage.

Accurate finger articulation. Distinguishing subtle finger configurations (important for many alphabetic and lexical characters) is a challenge for skeleton extractors and pose estimators. Leap Motion and glove sensors capture such details, but are invasive and limited in character diversity. The taxonomy therefore recommends hybrid data collection strategies (high-fidelity subsets, sensor-assisted calibration sessions) and improved hand pose estimation research focused on finger fidelity [82][60].

Motion dynamics and temporal scale. Characters vary in their temporal range, from short, static hand movements to long, multi-motion structures. Models must handle multiple temporal scales (frame-level micro-movements and sentence-level references), motivating multi-scale encoders (temporal neural networks CNNs, hierarchical transformers), and training programs that expose models to varying temporal granularity.

9.5 Concluding synthesis (actionable priorities)

Figure 3 frames these open issues as interrelated: environmental constraints shape what linguistic channels can be recorded; linguistic complexity requires richer annotations and multimodal capture; system-level repeatability and benchmark design determine whether progress is measurable; and articulatory phenomena at the gesture level define the minimum required detection fidelity. Based on the taxonomy and the presented overview of datasets/methods, we derive immediate priorities: (1) building large, multimodal, multiregion continuous corpora with frame/gloss alignments and non-manual annotations [75][35], (2) adopting canonical benchmark and leader partitions with repeatable baselines [35], (3) investing in finger fidelity capture (proximity cameras/hybrid sensors) focused on underrepresented character types such as backhand configurations [82][60], and (4) incorporating privacy-preserving options and ethical metadata into dataset releases. Implementing these priorities will lay the foundation for ArSLR systems that are both scientifically rigorous and practically useful for Arabic-speaking deaf communities.

10 Practical considerations: real-time systems, ethics, usability, e-learning interface

Designing ArSLR systems for real-world use requires considerations beyond recognition accuracy: computational latency and cost, hardware and implementation costs, user acceptance by deaf communities, inclusiveness across diverse dialects and user abilities, and clear ethical safeguards. Below, we synthesize practical tradeoffs and present a high-quality e-learning user interface design that maintains the pedagogical intentions discussed in this article. All recommendations are based on the dataset, modality, and system constraints described earlier.

Latency, responsiveness, and acceptable response time

Interactive and assistive ArSLR applications demand low perceived latency. A few general engineering targets are useful when designing prototypes and reporting results:

- **End-to-end latency target:** for conversational or tutoring interactions, aim for **≤300 ms** end-to-end from camera frame to rendered caption/feedback; for low-stakes practice (recording and batch feedback) up to 1–2 s is acceptable. Lower latency improves conversational fluency and user comfort.

- **Pipeline choices that affect latency:** heavy 3D-CNNs and large transformer stacks generally increase inference time; lightweight spatial backbones (MobileNet / EfficientNet-lite), skeleton-only models, model pruning/quantization, and on-device GPU/NN-accelerators reduce latency and enable offline use. Skeleton-only inference (if robust hand keypoints are available) is both fast and privacy-

friendly but depends on the fidelity of the pose extractor. Report latency (ms), CPU/GPU used, and real-time factor alongside accuracy in all system papers and releases.

Cost, hardware, and deployment trade-offs

- **Smartphone (RGB)** — lowest barrier to entry for users and dataset collection (ArSL2018 shows the utility of phone capture), easy to scale, but more sensitive to lighting and background. Works well for alphabet and isolated-word tasks and for e-learning where learners use their own phones. [35]

- **RGB-D (Kinect / depth sensors)** — provides depth and skeleton streams improving robustness to background and viewpoint, but hardware cost, setup complexity, and limited mobile availability constrain distribution at scale. Useful for lab evaluations, high-fidelity corpus building (ArabSign, Elpeltagy), and classroom installations. [75][59]

- **Sensors (Leap Motion / gloves / IMU)** — best for capturing fine finger articulation but intrusive and expensive for broad deployment. Their best use is in controlled data collection, teacher/annotator tooling, or as calibration/teacher signals to bootstrap vision-only models. [82][81]

Budget planning should consider device purchases, annotation labor (the primary cost of continuous and manual labeling), and long-term maintenance (privacy, updates). Hybrid approaches—sensor-assisted data collection to create high-quality "teacher" datasets and then distilling the models into pure RGB student models—offer a practical balance between data quality and implementation costs.

User acceptance, participatory design, and inclusivity

User acceptance depends on real involvement of Deaf users and communities at every stage:

- **Participatory design:** engage native ArSL signers, educators, and local communities when specifying functionality, UI layout, and dialect coverage. Co-design increases trust, usability, and cultural appropriateness.

- **Transparency & control:** allow users to choose privacy options (face blur, skeleton-only mode, local processing), control data sharing (opt-in for research), and inspect / correct automatic transcripts. These features materially increase acceptance.

- **Accessibility & inclusivity:** support multiple dialects and user variations (left/right handedness, age, signing style). Provide personalization (signer calibration steps, adjustable feedback sensitivity) and multimodal outputs (text Arabic, gloss, optional synthesized voice) to broaden utility across users and contexts.

Ethics, privacy and data governance

Ethical deployment requires documented consent procedures, clear licensing, and privacy-preserving defaults:

- **Consent & metadata:** dataset releases and deployed systems must include consent metadata, stated permitted uses, and de-identification protocols for face data. Where possible, provide skeleton-only datasets or on-device processing to reduce sensitive image circulation.

- **Bias & fairness:** routinely evaluate signer-independent performance and cross-dialect generalization. Publish per-group metrics (gender, dialect, age) where ethically permissible to surface and mitigate biases.

- **User agency:** provide users with mechanisms to correct transcripts and to delete their data; ensure clear documentation of system limits (e.g., "may fail for backhand signs or low lighting").

Usability & evaluation metrics beyond accuracy

Measure and report: (1) task completion time (for conversational tasks), (2) subjective usability (SUS/UEQ) from deaf testers, (3) confidence and satisfaction surveys, and (4) system robustness metrics (WER across dialects, sign language-independent accuracy). Combine objective and subjective assessment in lab and field tests before large-scale market launch.

High-level e-learning UI: a proposed design (principles + components)

Keep the e-learning concept— an interactive, learner-centered platform that teaches ArSL while providing automated feedback. Key design principles: simplicity, low latency, multi-view feedback, progressive difficulty, and cultural appropriateness.

Core UI components (desktop/mobile responsive):

1. **Main Signer View (camera feed):** central video of the reference signer or the learner's live camera; toggle **Reference** / **Practice** modes. Show optional face crop + hand close-up panels for fine detail.

2. **Real-time subtitle / gloss bar:** live captioning of predicted gloss or Arabic text with confidence color coding (green/high, orange/medium, red/low). Include a "why low confidence" tooltip (e.g., occlusion, low lighting).

3. **Practice mode & instant feedback:** learner performs a prompted sign; system displays immediate frame-level feedback (correct/incorrect), visual overlays (hand skeleton, keypoint markers), and a short corrective tip (e.g., "rotate palm outward"). Store repetitions and progress.

4. **Replay & slow-motion controls:** allow slow playback (0.5x, 0.25x) and frame-step for detailed comparison between learner and reference; helpful for finger-precision.

5. **Assessment dashboard & curriculum:** shows progress, per-sign mastery, common error types, and recommended next exercises; supports multi-dialect lessons and teacher assignments.

6. **Privacy & settings panel:** allow skeleton-only mode, local processing toggle, data-sharing consent, and dialect selection.

7. **Teacher / community tools:** allow instructors to upload custom lesson videos, annotate non-manual cues, and review student sessions with frame-aligned playback.

Back-end suggestions: provide two modes — **Interactive (low-latency)** for live practice using a lightweight on-device model or skeleton stream, and **Analytic (high-accuracy)** for batch uploads processed on server GPUs with heavier models and richer language-model rescoring. This hybrid allows scalable lessons while preserving interactivity.

Closing note — measuring success

Project success should be judged by technical performance and community acceptance, as measured by classroom implementation, teacher satisfaction, student competence growth, and adherence to privacy/consent policies. Combining rigorous technical requirements (latency, user-agnostic metrics) with participatory design and ethical data management will maximize the real-world benefits of ArSLR systems and e-learning tools for Arabic-speaking deaf communities.

11 Recommendations & future directions

In this section, the survey results are translated into specific, actionable research tasks and community-level recommendations to accelerate robust and replicable progress in Arabic Sign Language Recognition (ArSLR). The recommendations are categorized into (A) priority research tasks, (B) dataset and benchmark specifications, (C) best practices for modeling and evaluation, and (D) community, ethics, and implementation actions.

A. Prioritized research tasks (actionable, ranked)

1. **create a backhand / orientation-focused dataset (high priority).**
Motivation: Hand orientations and anomalous orientations are underrepresented in existing corpora and cause systematic errors in recognition in real-world settings [35][75]. Task: Collect a focused character dataset emphasizing backhand, rotated, and occluded hand configurations for multiple subjects and viewing angles (front + ±45° + camera with a close-up hand camera). Consider both individual tokens and sentence fragments to capture coarticulation with backhand shapes. Annotate hand orientation, occlusion flags, and keypoints for each frame. Use sensor-assisted capture (short glove or Leap Motion calibration) for a subset to provide a "teacher" annotation for finger-level ground truth.[82][60].

2. **Large-scale multimodal continuous benchmark (multi-region) (top strategic priority).**
Motivation: Continuous ArSL is underserved for the multimodal, multidialect corpora required by sequential models [75][59]. Task: Coordinate a cross-institutional dataset capturing RGB, synchronized depth (RGB-D), high-frame-rate handheld cameras (optional), and skeletal streams. Broad sampling in Arab regions (Egypt, Levant, Persian Gulf, Maghreb) with at least several dozen—and ideally hundreds—sign language users, balanced gender and age distribution, and examples of both forehand and backhand. Produce glossary-level and sentence-level transcripts and a frame-aligned subset for supervised sequential training. Publish user-dependent, user-independent, and cross-dialect canonical partitionings and license them for research use.

3. **Signer-independent methods & domain adaptation studies (medium priority).**
Motivation: Many publications report high accuracy within a dataset but poor generalization across signers; developing signer-robust algorithms is essential [35].

Aim: To evaluate and develop domain adaptation strategies (adversarial normalization, contrastive pretraining of multiple signers, adaptation at test time), systematic normalization pipelines (body-centered coordinates, per-joint scaling), and small-sample/meta-learning methods for bootstrapping new signers. Comparative methods using strict signer-independent partitions and transfer tests across datasets.

4. **Multimodal Fusion & Distillation Research (practical priority).**
Motivation: sensors give high fidelity but are impractical at scale; hybrid approaches can bootstrap practical models [89][60].
Task: collect small, high-fidelity multimodal "teacher" datasets (sensors + RGB), train multimodal models, and distill to lightweight RGB-only or skeleton-only student models for deployment. Explore attention-based fusion and modality dropout to improve robustness under degraded streams.

5. **Non-manual Signal Modeling & Grammar-aware Decoding (linguistic priority).**
Motivation: grammatical distinctions in ArSL often rely on facial and torso cues; these are frequently ignored in corpora and models [12][95].
Task: annotate non-manual markers (question, negation, topicalization) in corpora and integrate dedicated non-manual detectors into multi-stream encoders. Combine recognition outputs with language-aware decoders (gloss→Arabic mapping, neural rescoring) and evaluate translation metrics (BLEU/METEOR) in addition to sequence WER/GER.

B. Dataset & benchmark specifications (practical blueprint)

- **Core modalities:** RGB (frontal), RGB-D (depth + skeleton), optional close-up hand camera (for finger fidelity), and a small sensor subset (Leap Motion / glove) for teacher data. [75][82]
- **Signer sampling:** multi-region, target ≥50 signers for isolated/alphabets and ≥100 signers for continuous corpora where feasible; ensure gender, age, and handedness metadata. [35][75]
- **Annotation:** provide (a) image/segment-level labels for alphabets/isolated words, (b) gloss- and sentence-level transcripts for continuous data, and (c) a frame-aligned subset (≈5–10% of corpus) with non-manual tags.
- **Canonical splits & metadata:** publish signer-dependent and signer-independent test splits, plus cross-dialect test sets; include capture metadata (camera model, fps, resolution), consent/licensing, and per-signer demographics. [35][59]
- **Baseline releases:** provide reproducible baselines (lightweight CNN; skeleton-GCN; CNN+BiLSTM; transformer seq2seq) with fixed seeds and training scripts to seed leaderboards.

C. Modeling & evaluation best-practices

- **Evaluation metrics:** report per-class recall and confusion matrices, macro-F1, signer-independent accuracy, WER/GER for sequences, BLEU/METEOR for translation, latency (ms), and model size (MB). Always include both signer-dependent and signer-independent numbers. [35][75]
- **Robustness checks:** evaluate cross-dataset transfer and camera/hardware shifts (e.g., training on phone-captured data and testing on lab Kinect) to expose domain fragility.
- **Efficient deployment:** invest in model compression (pruning, quantization), skeleton-only fast inference, and small transformer/TCN variants for edge devices. Publish latency and memory profiles.
- **Open protocols:** include data augmentation recipes, preprocessing pipelines (hand crop, normalization), and exact temporal-cropping rules — these must be in paper appendices for reproducibility.

D. Community, ethics, and infrastructure

- **Leaderboards & tracks:** host modality-separated tracks (RGB-only; RGB-D; skeleton-only; sensor-fusion) with canonical splits and shared evaluation scripts. This encourages fair comparison and incremental progress. [35]
- **Ethics & privacy:** require dataset releases to include consent metadata and de-identification recommendations; provide skeleton-only options and on-device inference templates to reduce image sharing. [35]

- **Funding & collaboration:** pursue multi-university, multi-country grants to fund large corpus collection (annotation is expensive). Encourage partnerships with Deaf education centers for community-aligned data collection and participatory design.

- **Tooling:** invest in annotation tools that simplify gloss/non-manual labeling (time-aligned interfaces, semi-automatic keypoint propagation) to reduce labeling cost and improve consistency.

Closing roadmap (short/medium/long term)

- **Short (6–12 months):** publish clear dataset checklist; release baseline reproducible code; begin small backhand-focused and sensor-teacher pilot datasets.

- **Medium (12–24 months):** coordinate and release a first multimodal, multi-region continuous corpus with canonical splits and leaderboards; develop signer-independent baselines.

- **Long (24+ months):** iterate larger corpora, improvements in finger fidelity capture, community adoption in educational deployments, and robust translation pipelines linking ArSL to Arabic text.

12 Conclusion

Supplementary materials. The submission includes three supplementary files: `Supplementary_Table_S1_ProcessingPipeline.xlsx` (full pipeline ↔ methods matrix), `Supplementary_Table_S2_DatasetInventory.xlsx` (full dataset inventory with availability/license/consent metadata) and `sota_table.csv` (machine-readable consolidated SOTA experiment list). These files provide the exhaustive tables referenced in the main text.

This review maps the landscape of Arabic sign language recognition (ArSLR) in terms of both task (alphabet, single word, continuous mode) and capture modality (vision-based, sensor-based), indicating where concrete progress has been made and where work is still urgently needed.

For aphabet-related tasks (finger-spelling), vision-based imagery—especially ArSL2018—and modern CNNs and transformer networks deliver consistently high accuracy within a corpus, demonstrating that image-level classification is currently a relatively mature problem under controlled conditions [35][31]. Sensor pipelines (Leap Motion, data gloves) also achieve excellent results for precise finger articulation in small, controlled studies, but their invasiveness and small number of signers limit widespread application [82].

For single-word recognition, the image is more heterogeneous. Vision-based video approaches using 3D-CNNs, CNN+LSTM stacks, and pose/skeleton encoders perform very well on well-segmented corpora of moderate size, but reported performance varies depending on vocabulary size, sign variety, and segmentation quality [46][64][75]. Sensor-based systems again provide high precision in laboratory settings, and multimodal fusion (RGB + depth + sensors) often yields the best results within a corpus—pointing to fusion as a practical path to robustness when datasets support it [60][89]. Continuous sentence recognition remains the most challenging area. Co-articulation, ambiguous character boundaries, long-range dependencies, and the necessary non-manual grammar make continuous ArSLR sentence recognition a sequential, data-intensive problem. End-to-end methods (CTC, attention-based sequencing, transforms) are promising, but progress is limited by the scarcity of large, multimodal, frame-aligned, continuous corpora and standardized evaluation partitions [75][59][12]. Vision-based continuous systems struggle most with signer independence and cross-dialect generalization; sensor-assisted corpora help, but are not a scalable substitute for diverse multimodal corpora.

Two consistent themes emerge across modalities. First, modality trade-offs: vision-based capture is non-invasive and captures non-manual cues (face, torso) essential for ArSL grammar, while sensors provide better fingerprint fidelity, but at the expense of scalability and signer diversity [35][82][60]. Second, evaluation fragility: many high accuracies reflect signer-dependent or one-sided partitions; Signer-independent and cross-dialect evaluations routinely reveal significant performance degradations [75][35]. To advance this field, we reiterate the paper's priority goals: building large multimodal (RGB + depth/skeleton + high-fidelity hand) and multiregion continuous corpora; collecting targeted subsets of backhand/orientation and finger fidelity; standardizing canonical partitioning and reporting (signer-independent metrics, per-class recall, WER/GER for sequences); and pursuing multimodal distillation so that practical RGB-only systems can inherit sensor-level fidelity where needed [75][35][82]. In short, ArSLR has matured in discrete tasks and controlled setups, but achieving robust, deployable systems for Arabic-speaking communities requires coordinated investments in multimodal data, signer-resistant algorithms, collaborative benchmarking, and community-focused evaluation – combining promising lab results with inclusive real-world impact.

Appendix A

Figure A.1. PRISMA flow diagram summarizing our literature selection process: records identified, screened, full-text assessed and studies included (2000–mid-2025).

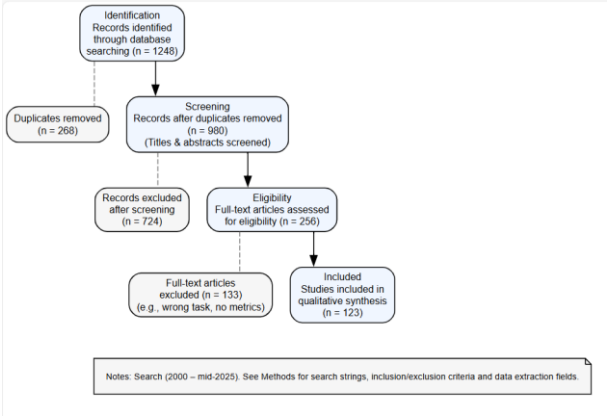

Figure A.1: PRISMA figure counts. This figure is also attached in the supplementary files.

Acknowledgements

Language and editing. The manuscript was revised for clarity and concision; a formal language proof-reading pass was performed to reduce redundancy and improve readability. The authors welcome professional copyediting if required by the journal.

References

[1] J. N. Pérez, “Juan pablo bonet, reducción de las letras y arte para enseñar a hablar a los mudos,” RIFOP: Revista interuniversitaria de formación del profesorado: continuación de la antigua Revista de Escuelas Normales, no. 14, pp. 252–253, 1992.

[2] N. E. Groce, everyone here spoke sign language: Hereditary deafness on Martha’s Vineyard. Harvard University Press, 1985.

[3] L. Ray, “The abbe’de l’epee,” American Annals of the Deaf and Dumb, vol. 1, no. 2, pp. 69–76, 1848.

[4] M. A. Abdel-Fattah, “Arabic sign language: a perspective,” Journal of deaf studies and deaf education, vol. 10, no. 2, pp. 212–221, 2005.

[5] B. S. Wilson and D. L. Tucci, “Addressing the global burden of hearing loss,” The Lancet, vol. 397, no. 10278, pp. 945–947, 2021.

[6] T. Aujeszky and M. Eid, “A gesture recogintion architecture for Arabic sign language communication system,” Multimed. Tools Appl., vol. 75, no. 14, pp. 8493–8511, 2015.

[7] A. RaySarkar, G. Sanyal, and S. Majumder, “Hand Gesture Recognition Systems: A Survey,” Int. J. Comput. Appl., vol. 71, no. 15, pp. 25–37, 2013.

[8] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, 2019.

[9] Y. Ying Wu and T. S. Huang, “Hand modeling, analysis and recognition,” IEEE Signal Process. Mag., vol. 18, no. 3, pp. 51–60, May 2001.

[10] Ahmed, A. M., Alez, R. A., Taha, M., & Tharwat, G. (2016). Automatic translation of Arabic sign to Arabic text (ATASAT) system. Journal of Computer Science and Information Technology, 6, 109122.

[11] R. M. Duwairi and Z. A. Halloush, “Automatic recognition of arabic alphabets sign language using deep learning.” International Journal of Electrical & Computer Engineering (2088-8708), vol. 12, no. 3, 2022.

[12] M. Hassan, K. Assaleh, and T. Shanableh, “Multiple proposals for continuous Arabic sign language recognition,” Sensing and Imaging, vol. 20, no. 1, p. 4, 2019.

[13] M. Mohandes, J. Liu, and M. Deriche, “A survey of image-based arabic sign language recognition,” in 2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14). IEEE, 2014, pp. 1–4.

[14] M. Mohandes, S. Aliyu, and M. Deriche, “Arabic sign language recognition using the leap motion controller,” in 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE). IEEE, 2014, pp. 960–965.

[15] C.-H. Chuan, E. Regina, and C. Guardino, “American sign language recognition using leap motion sensor,” in 2014 13th International Conference on Machine Learning and Applications. IEEE, 2014, pp. 541–544.

[16] K. S. Abhishek, L. C. F. Qubeley, and D. Ho, “Glove-based hand gesture recognition sign language translator using capacitive touch sensor,” in 2016 IEEE international conference on electron devices and solid-state circuits (EDSSC). IEEE, 2016, pp. 334–337.

[17] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, “Hand gesture recognition with 3d convolutional neural networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 1–7.

[18] R. Hartanto, A. Susanto, and P. I. Santosa, “Real time static hand gesture recognition system prototype for indonesian sign language,” in 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE). IEEE, 2014, pp. 1–6.

[19] T.-Y. Pan, L.-Y. Lo, C.-W. Yeh, J.-W. Li, H.-T. Liu, and M.-C. Hu, “Real-time sign language recognition in complex background scene based on a hierarchical clustering classification method,” in 2016 IEEE second international conference on multimedia big data (BigMM). IEEE, 2016, pp. 64–67.

[20] R. Yang, S. Sarkar, and B. Loeding, “Enhanced level building algorithm for the movement epenthesis problem in sign language recognition,” in 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1 8.

[21] “Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming,” IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 3, pp. 462– 477, 2009.

[22] Ahmed, A. M., et al. (2017). Automatic Translation of Arabic Sign To Arabic Text (ATASAT) System. Computer Science & Information Technology, 109.

[23] Saleh, Y., & Issa, G. (2020). Arabic sign language recognition through deep neural networks fine-tuning.

[24] Hasasneh, A. (2020). Arabic Sign Language Characters Recognition Based on A Deep Learning Approach and a Simple Linear Classifier. Jordanian Journal of Computers and Information Technology, 6(3).

[25] Kamruzzaman, M. (2020). Arabic sign language recognition and generating Arabic speech using convolutional neural network. Wireless Communications and Mobile Computing, 2020.

[26] Tharwat, G., et al. (2021). Arabic sign language recognition system for alphabets using machine learning techniques. Journal of Electrical and Computer Engineering, 2021, 1-17.

[27] Alani, A. A., & Cosma, G. (2021). ArSL-CNN: a convolutional neural network for Arabic sign language gesture recognition. Indonesian journal of electrical engineering and computer science, 22.

[28] Alawwad, R. A., et al. (2021). Arabic sign language recognition using faster R-CNN. International Journal of Advanced Computer Science and Applications, 12(3).

[29] Azhar, N. A. N., et al. (2022). Development of Mobile Application for Arabic Sign Language based on Android Studio Software. JOURNAL OF ALGEBRAIC STATISTICS, 13(3), 3152-3160.

[30] R. M. Duwairi and Z. A. Halloush, "Automatic recognition of arabic alphabets sign language using deep learning." International Journal of Electrical & Computer Engineering (2088-8708), vol. 12, no. 3, 2022

[31] M. Zakariah, Y. A. Alotaibi, D. Koundal, Y. Guo, and M. Mamun Elahi, "Sign language recognition for Arabic alphabets using transfer learning technique," Computational Intelligence and Neuroscience, vol. 2022, no. 1, p. 4567989, 2022.

[32] M. M. Balaha, S. El-Kady, H. M. Balaha, M. Salama, E. Emad, M. Hassan, and M. M. Saafan, "A vision-based deep learning approach for independent-users arabic sign language interpretation," Multimedia Tools and Applications, vol. 82, no. 5, pp. 6807–6826, 2023

[33] M. A. Bencherif, M. Algabri, M. A. Mekhtiche, M. Faisal, M. Alsulaiman, H. Mathkour, M. Al-Hammadi, and H. Ghaleb, "Arabic sign language recognition system using 2d hands and body skeleton data," IEEE Access, vol. 9, pp. 59 612–59 627, 2021.

[34] A. Alnuaim, M. Zakariah, W. A. Hatamleh, H. Tarazi, V. Tripathi, and E. T. Amoatey, "Human-computer interaction with hand gesture recognition using resnet and mobilenet," Computational Intelligence and Neuroscience, vol. 2022, no. 1, p. 8777355, 2022.

[35] A. A. Alani and G. Cosma, "Arsl-cnn: a convolutional neural network for arabic sign language gesture recognition," Indonesian journal of electrical engineering and computer science, vol. 22, 2021.

[36] E. Aldhahri, R. Aljuhani, A. Alfaidi, B. Alshehri, H. Alwadei, N. Aljojo, A. Alshutayri, and A. Almazroi, "Arabic sign language recognition using convolutional neural network and mobilenet," Arabian Journal for Science and Engineering, vol. 48, no. 2, pp. 2147–2154, 2023.

[37] B. A. Dabwan and M. E. Jadhav, "A deep learning based recognition system for yemeni sign language," in 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI). IEEE, 2021, pp. 1–5.

[38] I. Hmida and N. B. Romdhane, "Arabic sign language recognition algorithm based on deep learning for smart cities," in The 3rd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2022), vol. 2022. IET, 2022, pp. 119–127.

[39] H. I. Mohammed and J. Waleed, "Hand gesture recognition using a convolutional neural network for arabic sign language," in AIP Conference Proceedings, vol. 2475, no. 1. AIP Publishing, 2023.

[40] Al Ahmadi, S., Mohammad, F., & Al Dawsari, H. (2024). Efficient YOLO-based deep learning model for Arabic sign language recognition. Journal of Disability Research, *3*(4), 20240051.

[41] Batnasan, G., Gochoo, M., Otgonbold, M.-E., Alnajjar, F., & Shih, T. K. (2022). ArSL21L: Arabic sign language letter dataset benchmarking and an educational avatar for metaverse applications. In 2022 IEEE Global Engineering Education Conference (EDUCON) (pp. 1814–1821). IEEE.

[42] Attia, N. F., Ahmed, M. T. F. S., & Alshewimy, M. A. M. (n.d.). Improved deep learning model based real-time recognition of Arabic sign language. [Unpublished manuscript or in preparation].

[43] Alamri, F. S., Rehman, A., Abdullahi, S. B., & Saba, T. (2024). Intelligent real-life key-pixel image detection system for early Arabic sign language learners. PeerJ Computer Science, *10*, e2063.

[44] Aiouez, S., Hamitouche, A., Belmadoui, M. S., Belattar, K., & Souami, F. (2022). Real-time Arabic sign language recognition based on YOLOv5. In IMPROVE (pp. 17–25).

[45] Mazen, F., & Ezz-Eldin, M. (2024). A novel image-based Arabic hand gestures recognition approach using YOLOv7 and ArSL21L. Fayoum University Journal of Engineering, *7*(1), 40–48.

[46] Gochoo, M., Batnasan, G., Ahmed, A. A., Otgonbold, M.-E., Alnajjar, F., Shih, T. K., Tan, T.-H., & Wee, L. K. (2023). Fine-tuning vision transformer for Arabic sign language video recognition on augmented small-scale dataset. In 2023 IEEE International Conference on

Systems, Man, and Cybernetics (SMC) (pp. 2880–2885). IEEE.

[47] Mohamed, M. M. (2020). Automatic system for Arabic sign language recognition and translation to spoken one. International Journal, *9*(5), 7140–7148.

[48] Sarhan, N. A., El-Sonbaty, Y., & Youssef, S. M. (2015). HMM-based Arabic sign language recognition using Kinect. In 2015 Tenth International Conference on Digital Information Management (ICDIM) (pp. 169–174). IEEE.

[49] Assaleh, K., Shanableh, T., Fanaswala, M., Amin, F., & Bajaj, H. (2010). Continuous Arabic sign language recognition in user dependent mode. Journal of Intelligent Learning Systems and Applications, *2*(1), 19.

[50] Mosa, D. T., Nasef, N. A., Lotfy, M. A., Abohany, A. A., Essa, R. M., & Salem, A. (2023). A real-time Arabic avatar for deaf–mute community using attention mechanism. Neural Computing and Applications, *35*(29), 21709–21723.

[51] Boufelfel, R. (2024). An Arabic sign language recognition system for word-level generation and translation [Doctoral dissertation, University of Guelma].

[52] Almaazmi, M., Elkadi, S., Elsayed, L., Salman, L., & Shanableh, T. (2024). Motion images with positioning information and deep learning for continuous Arabic sign language recognition in signer dependent and independent modes. IEEE Access. Advance online publication.

[53] Shohieb, S. M., Elminir, H. K., & Riad, A. M. (2015). Signsworld atlas; a benchmark Arabic sign language database. Journal of King Saud University-Computer and Information Sciences, *27*(1), 68–76.

[54] Briones Cerquín, A. D., Tumay Guevara, J. A., & Ovalle, C. (2025). Mobile application for continuous recognition and classification of sign language images through deep learning. International Journal of Interactive Mobile Technologies, *19*(7).

[55] ElBadawy, M., et al. (2017). Arabic sign language recognition with 3d convolutional neural networks. Paper presented at the 2017 Eighth international conference on intelligent computing and information systems (ICICIS).

[56] Luqman, H., & Mahmoud, S. A. (2017). Transform-based Arabic sign language recognition. Procedia Computer Science, 117, 2-9.

[57] Ibrahim, N. B., et al. (2017). An Automatic Arabic Sign Language Recognition System (ArSLRS). Journal of King Saud UniversityComputer and Information Sciences.

[58] Abdel-Samie, A.-G. A.-R., et al. (2018). Arabic sign language recognition using kinect sensor. Research Journal of Applied Sciences, Engineering and Technology, 15(2), 57-67.

[59] Elpeltagy, M., et al. (2018). Multi-modality-based Arabic sign language recognition. IET Computer Vision, 12(7), 1031-1039.

[60] Hisham, B., & Hamouda, A. (2021). Arabic sign language recognition using Ada-Boosting based on a leap motion controller. International Journal of Information Technology, 13, 1221-1234.

[61] Almasre, M. A., & Al-Nuaim, H. (2020). A comparison of Arabic sign language dynamic gesture recognition models. Heliyon, 6(3).

[62] Mohamed, M. M. (2020). Automatic system for Arabic sign language recognition and translation to spoken one. International Journal, 9(5).

[63] Aly, S., & Aly, W. (2020). DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. IEEE Access, 8, 83199-83212.

[64] Luqman, H., & El-Alfy, E.-S. M. (2021). Towards hybrid multimodal manual and non-manual Arabic sign language recognition: MArSL database and pilot study. Electronics, 10(14), 1739.

[65] Abdul, W., et al. (2021). Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM. Computers and Electrical Engineering, 95, 107395.

[66] Boukdir, A., et al. Isolated video-based Arabic sign language recognition using convolutional and recursive neural networks. Arabian Journal for Science and Engineering, 1-13.

[67] Balaha, M. M., et al. (2023). A vision-based deep learning approach for independent-users Arabic sign language interpretation. Multimedia Tools and Applications, 82(5), 6807-6826.

[68] Alyami, S., et al. (2023). Isolated Arabic Sign Language Recognition Using A Transformer-based Model and Landmark Keypoints. ACM Transactions on Asian and Low-Resource Language Information Processing.

[69] Assaleh, K., et al. (2010). Continuous Arabic sign language recognition in user dependent mode.

[70] Tolba, M., et al. (2012). A proposed graph matching technique for Arabic sign language continuous sentences recognition. Paper presented at the Informatics and Systems (INFOS), 2012 8th International Conference on.

[71] Tolba, M. F., et al. (2013). Arabic sign language continuous sentences recognition using PCNN and graph matching. Neural Computing and Applications, 23(3-4), 999-1010.

[72] Sidig, A. a. I., et al. (2018). Arabic sign language recognition using optical flow-based features and HMM. Paper presented at the Recent Trends in Information and Communication Technology: Proceedings of the 2nd International Conference of Reliable Information and Communication Technology (IRICT 2017).

[73] LUQMAN, H., & ELALFY, E. (2022). Utilizing motion and spatial features for sign language gesture recognition using cascaded CNN and LSTM models. Turkish Journal of Electrical Engineering and Computer Sciences, 30(7), 2508 2525.

[74] Luqman, H., & Mahmoud, S. A. (2020). A machine translation system from Arabic sign language to

Arabic. Universal Access in the Information Society, 19(4), 891-904.

[75] Luqman, H. (2023). ArabSign: A Multi-modality Dataset and Benchmark for Continuous Arabic Sign Language Recognition. Paper presented at the 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG).

[76] Tharwat, G., et al. (2021). Arabic sign language recognition system for alphabets using machine learning techniques. Journal of Electrical and Computer Engineering, 2021, 1-17.

[77] Elons, A. S., et al. (2013). A proposed PCNN features quality optimization technique for pose-invariant 3D Arabic sign language recognition. Applied Soft Computing, 13(4), 1646-1660.

[78] ElBadawy, M., et al. (2017). Arabic sign language recognition with 3d convolutional neural networks. Paper presented at the 2017 Eighth international conference on intelligent computing and information systems (ICICIS).

[79] Sidig, A. A. I., et al. (2021). KArSL: Arabic sign language database. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 20(1), 1-19.

[80] Nagi, J., et al. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. Paper presented at the 2011 IEEE international conference on signal and image processing applications (ICSIPA), Kuala Lumpur, Malaysia.

[81] Assaleh, K., et al. (2012). Low complexity classification system for glove-based arabic sign language recognition. Paper presented at the Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part III 19.

[82] Mohandes, M., et al. (2014). Arabic sign language recognition using the leap motion controller. Paper presented at the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE).

[83] Shohieb, S. M., et al. (2015). Signsworld atlas; a benchmark Arabic sign language database. Journal of King Saud University-Computer and Information Sciences, 27(1), 68-76.

[84] Ahmed, A. M., et al. (2016). Automatic translation of Arabic sign to Arabic text (ATASAT) system. Journal of Computer Science and Information Technology, 6, 109-122.

[85] Alzohairi, R., et al. (2018). Image based arabic sign language recognition system. International Journal of Advanced Computer Science and Applications (IJACSA), 9(3), 185-194.

[86] Ibrahim, N. B., et al. (2018). An automatic Arabic sign language recognition system (ArSLRS). Journal of King Saud UniversityComputer and Information Sciences, 30(4), 470-477.

[87] Hayani, S., et al. (2019). Arab sign language recognition with convolutional neural networks. Pape presented at the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco.

[88] Latif, G., et al. (2019). ArASL: Arabic alphabets sign language dataset. Data in brief, 23, 103777.

[89] Alnahhas, A., et al. (2020). Enhancing the recognition of Arabic sign language by using deep learning and leap motion controller. Int. J. Sci. Technol. Res, 9, 1865-1870.

[90] Ismail, M. H., et al. (2021). Static hand gesture recognition of Arabic sign language by using deep CNNs. Indonesian Journal of Electrical Engineering and Computer Science, 24(1), 178-188.

[91] Al-Barham, M., et al. (2023). RGB Arabic Alphabets Sign Language Dataset. arXiv preprint arXiv:2301.11932.

[92] S. Bilal, R. Akmeliawati, A. A. Shafie, and M. J. E. Salami, "Hidden Markov model for human to computer interaction: a study on human hand gesture recognition," Artif. Intell. Rev., vol. 40, no. 4, pp. 495–516, 2011.

[93] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," Comput. Vis. Image Underst., vol. 141, pp. 152–165, 2015.

[94] N. El-Bendary, H. M. Zawbaa, M. S. Daoud, A. E. Hassanien, and K. Nakamatsu, "ArSLAT: Arabic Sign Language Alphabets Translator," in 2010 International Conference on Computer Information Systems and Industrial Management Applications, CISIM 2010, 2010, pp. 590–595.

[95] M. A. Abdel-Fattah, "Arabic Sign Language: A Perspective," J. Deaf Stud. Deaf Educ., vol. 10, no. 2, pp. 212–221, 2005.

[96] A. Almohimeed, M. Wald, and R. I. Damper, "Arabic Text to Arabic Sign Language Translation System for the Deaf and Hearing-Impaired Community," in Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies, 2011, pp. 101–109.

97] E. Aghajari and D. Gharpure, "Real Time Vision-Based Hand Gesture Recognition for Robotic Application," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 4, no. 3, pp. 2277–128, 2014.

[98] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," Eng. Appl. Artif. Intell., vol. 22, no. 8, pp. 1141–1158, Dec. 2009.

[99] S. Kausar and M. Y. Javed, "A Survey on Sign Language Recognition," in Frontiers of Information Technology (FIT), 2011, pp. 95–98.

[100] H. Hasan and S. Abdul-Kareem, "Human–computer interaction using vision-based hand gesture recognition systems: a survey," Neural Comput. Appl., vol. 25, no. 2, pp. 251–261, 2014.

[101] Y. Zhou, G. Jiang, and Y. Lin, "A novel finger and hand pose estimation technique for real-time hand gesture recognition," Pattern Recognit., vol. 49, pp. 102–114, 2016.

[102] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," IEEE Comput. Graph. Appl., vol. 22, no. 6, pp. 64–71, 2002.

[103] J. Triesch and C. Von Der Malsburg, "A system for person-independent hand posture recognition against complex backgrounds," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 12, pp. 1449– 1453, 2001.

[104] M. a Berbar, H. M. Kelash, and A. a Kandeel, "Faces and Facial Features Detection in Color Images," in Geometric Modeling and Imaging – New Trends (GMAI06), 2006, pp. 209–214.

[105] Ahmed, A. M., et al. (2017). Automatic Translation of Arabic Sign To Arabic Text (ATASAT) System. Computer Science & Information Technology, 109.

[106] Tharwat, G., et al. (2021). Arabic sign language recognition system for alphabets using machine learning techniques. Journal of Electrical and Computer Engineering, 2021, 1-17.

[107] Tubaiz, N., et al. (2015). Glove-based continuous Arabic sign language recognition in user-dependent mode. IEEE Transactions on Human-Machine Systems, 45(4), 526-533.

[108] Assaleh, K., & Al-Rousan, M. (2005). Recognition of Arabic sign language alphabet using polynomial classifiers. EURASIP Journal on Advances in Signal Processing, 2005(13), 507614.

[109] Tharwat, A., et al. (2015). Sift-based arabic sign language recognition system. Paper presented at the Afro european conference for industrial advancement.

[110] Bauer, B., & Hienz, H. (2000). Relevant features for video-based continuous sign language recognition. Paper presented at the Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on.

[111] Abdel-Wahab, M. S., et al. (2006). Arabic sign language recognition using neural network and graph matching techniques. Paper presented at the Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications.

[112] Sidig, A. A. I., et al. (2021). KArSL: Arabic sign language database. ACM Transactions on Asian and Low Resource Language Information Processing (TALLIP), 20(1), 1-19.

[113] G. Latif, J. Alghazo, N. Mohammad, and R. Alghazo, "Communicating with the deaf and hard of hearing through automatic arabic sign language translator," in Journal of Physics: Conference Series, vol. 1962, no. 1. IOP Publishing, 2021, p. 012055.

[114] N. M. Alharthi and S. M. Alzahrani, "Vision transformers and transfer learning approaches for arabic sign language recognition," Applied Sciences, vol. 13, no. 21, p. 11625, 2023.

[114] N. M. Alharthi and S. M. Alzahrani, "Vision transformers and transfer learning approaches for arabic sign language recognition," *Applied Sciences*, vol. 13, no. 21, p. 11625, 2023.

[115] Latif, G., Mohammad, N., Alghazo, J., AlKhalaf, R., & AlKhalaf, R. (2019). ArASL: Arabic Alphabets Sign Language Dataset. Data in Brief, 23, 103777. https://doi.org/10.1016/j.dib.2019.103777

[116] Luqman, H. (2023). ArabSign: A Multi-modality Dataset and Benchmark for Continuous Arabic Sign Language Recognition. In Proceedings of the 2023 IEEE International Conference on Automatic Face and Gesture Recognition (FG 2023). https://doi.org/10.1109/FG57933.2023.10042720 . (Dataset & code repo: https://github.com/Hamzah-Luqman/ArabSign ; arXiv preprint: arXiv:2210.03951).

[117] Al-Barham, M., Alsharkawi, A., Al-Yaman, M., Al-Fetyani, M., Elnagar, A., Abu SaAleek, A., & Al-Odat, M. (2023). RGB Arabic Alphabets Sign Language Dataset (AASL). arXiv preprint arXiv:2301.11932. https://doi.org/10.48550/arXiv.2301.11932

[118] Alsulaiman, M., Faisal, M., Mekhtiche, M., Bencherif, M., Alrayes, T., Muhammad, G., Mathkour, H., Abdul, W., Alohali, Y., & Alqahtani, M. (2023). Facilitating the communication with deaf people: Building a largest Saudi sign language dataset. Journal of King Saud University — Computer and Information Sciences, 35(8), Article 101642. https://doi.org/10.1016/j.jksuci.2023.101642

[119] Abbas, S., Alahmadi, D., & Al-Barhamtoshy, H. (2024). Establishing a multimodal dataset for Arabic Sign Language (ArSL) production. Journal of King Saud University — Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2024.102165

[120] Noor, T. H., Noor, A., Alharbi, A. F., Faisal, A., Alrashidi, R., Alsaedi, A. S., Alharbi, G., Alsanoosy, T., & Alsaeedi, A. (2024). Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model. Sensors, 24(11), 3683. https://doi.org/10.3390/s24113683

[121] Uthus, D., Tanzer, G., & Georg, M. (2023). YouTube-ASL: A Large-Scale, Open-Domain American Sign Language — English Parallel Corpus. arXiv preprint arXiv:2306.15162. https://doi.org/10.48550/arXiv.2306.15162

[122] Moryossef, A., & Jiang, Z. (2023). SignBank+: Preparing a Multilingual Sign Language Dataset for Machine Translation Using Large Language Models. arXiv preprint arXiv:2309.11566. https://doi.org/10.48550/arXiv.2309.11566

[123] Deng, Z., Leng, Y., Chen, J., Yu, X., Zhang, Y., & Gao, Q. (2024). TMS-Net: A multi-feature multi-stream multi-level information sharing network for skeleton-based sign language recognition. Neurocomputing, 572, 127194. https://doi.org/10.1016/j.neucom.2023.127194