AI-Based Arabic Sign Language to Voice Translation System

Hussein A. Mutar[1], Abdul Hadi M.Alaidi[1], Ammar A.Kazm[2], Saif Ali Abd Alradha Alsaidi[1], Haider TH. Salim Alrikabi[4], Ali Khalaf Hussein[3], Ibtihal R. N. Alrubeei[4,] and Nabaa Ali Abd Alradha Alsaeedi[5]

[1]*Computer Department, College of Computer Science and Information Technology, Wasit University, 52001 Kut, Wasit, Iraq*

[2]*Computer Department, College of Education for Pure Sciences, Wasit University, 52001 Kut, Wasit, Iraq*

[3]*Mathematics Department, College of Education for Pure Sciences, Wasit University, 52001 Kut, Wasit, Iraq*

[4]*Electrical Engineering Department, College of Engineering, Wasit University, 52001 Kut, Wasit, Iraq*

[5]*Department of pathological Analysis, College of Science, Wasit University, 52001 Kut, Wasit, Iraq*

alaidi@uowasit.edu.iq,salsaidi@uowasit.edu.iq,aawaad@uowasit.edu.iq, hmutar@uowasit.edu.iq, hdhiyab@uowasit.edu.iq, alhachamia@uowasit.edu.iq, Ibtihal.razaq@uowasitdu.edu.iq,nalsaidi@uowasit.edu.iq

Keywords: Arabic Sign Language, Sign Language Translation, Transformer, TTS, Gloss Recognition, Conformer, CTC.

Abstract: The Deaf and Hard-of-Hearing (DHH) community in the Arab region faces major communication difficulties arising solely because of the scarcity of interpreters. Though computer vision systems for sign language recognition exist for many languages, Arabic Sign Language (ArSL) continues to be very challenging to translate because few datasets are available and the inherent linguistic complexities exist. This research outlines a whole new end-to-end artificial intelligence system that attains edge-cutting accuracy in translating continuous ArSL directly to spoken Arabic. the proposed architecture employs a double-loss training technique for minimizing alignment errors and a 3D ResNet-Conformer vision encoder for effective feature extraction. Once an Arabic-conscious decoder using morphological embeddings guarantees that the outputs are grammar-correct, a boosted text-to-speech module turns them into understandable speech. The system was evaluated and trained with the use of the massive ArSL-Voice corpus, which had 45 signers' data points spanning 180 hours. An estimated overall accuracy of more than 73% was indicated by its BLEU-4 score of 27.5 and Word Error Rate (WER) of 28.7%, demonstrating remarkable efficacy. Using the inexpensive Raspberry Pi 5, we show that such high-quality translation can be used successfully in real-world scenarios, with a latency of only 173 ms and a power drain as low as 4.1W. These results show that this technology can be used in many real-world settings, including healthcare, education, and public administration, which will help ArSL users communicate more effectively.

1 INTRODUCTION

The arena of human–computer interaction regarding Deaf and Hard of Hearing (DHH) people is shifting towards enabling one-to-one communication directly from sign into speech [1], [2]. Though significant progress has been made in automating the recognition of sign language and the translation of different sign languages, Arabic Sign Language (ArSL) is particularly under-represented despite the linguistic complexity and local variety [1], [2]. The translation of ArSL presents several challenges due to simultaneously technical constraints and unique characteristics of the Arabic language. These include the Arabic language's rich morphology, the diglossic property (which distinguishes between local colloquialisms and Modern Standard Arabic), and the requirement for diacritical restoration as a precondition for textual faithfulness in text-to-speech conversions [3], [4].

In Arabic-speaking nations, the Deaf community faces particularly severe communication barriers. According to recent statistics, Saudi Arabia alone is home to an estimated 229,541 deaf people, but there is a severe shortage of sign-language interpreters, with a ratio of only about 1:93,000, which is significantly lower than the 1:46 ratio for California, USA [5]. In the majority of the Arabian Peninsula's nations, a shortage of language translators results in limited access to government services, educational opportunities, and social inclusion for the deaf.

Technology-based solutions may be able to bridge this communication gap, but there are a number of obstacles that must be overcome before they can be used effectively [6].

Technical challenges in developing viable sign to voice systems for ArSL are: (i) effective spatio temporal modeling with limited training corpus; (ii) deployment of either explicit or implicit gloss guidance for improved alignment; (iii) generating grammatically well-formed and diacritically well-formed Arabic string output; and (iv) deployment on low-cost edge hardware with minimal latency and power consumption [3], [6], [7]. Existing work has tended to handle individual components of such a pipeline in isolation, but an overall, end-to-end solution does not exist and especially not for optimized real-world deployment [4], [8].

This technical report presents a unified ArSL-to-voice system that jointly learns continuous sign recognition and generates Arabic scripts and follows with neural TTS synthesis. The system includes a 3D ResNet–Conformer vision encoder and double-loss training objective, an Arabic-aware decoder with morpheme embedding, and an Arabic-adaptive TTS component. We evaluate our system on the ArSL-Voice corpus (~180 hours, 45 signers, more than 10 domains) and find substantial improvements in accuracy and deployment efficiency compared with state-of-the-art methodologies.

2 LITERATURE REVIEW

In this part, we will take a look back at the early and current research on speech synthesis, translation, and sign language recognition. It lays out the existing landscape, draws attention to the unique difficulties of Arabic Sign Language (ArSL), and explains where previous studies have failed to fill in the gaps that this one intends to cure. To grounded the proposed methodological choices and highlighted its contributions relative to current research in this contextual review.

2.1 Sign Language Recognition Techniques

Developments of sign language recognition (SLR) system have significantly progressed in the recent decade by taking advantage of developments of computer vision, deep learning, and sequence model-based approaches. American Sign Language (ASL) and European sign languages have been the center of attention of most of the literature, and established standards such as MS-ASL and RWTH-PHOENIX-Weather 2014T have been paving the way for advances of continuous sign language recognition (CSLR) and sign language translation (SLT) [2], [8]. The systems have largely belonged to three categories: sensor-based systems using data gloves and inertial measurement units, vision-based systems using cameras and computer vision-based approaches, and multimodal fusion-based systems [4], [8], [9].

Vision-based systems have gained favor due to privacy and ease of use. Conventional systems utilized hand-crafted features and traditional machine learning classifiers. For example, Assaleh et al. [10] utilized Motion Estimation and Accumulated Differences and HMMs for online ArSL recognition and achieved 94% on a small set. Tharwat et al. utilized SIFT features and LDA feature reduction and SVM classification and achieved 98% on 30 Arabic signs [11]. Those systems, however, didn't address variability of signing style, background clutters, and computation efficiency [3].

In recent times, models of deep learning have been on center stage with CNNs excelling at spatial learning of features and LSTMs and Transformers forming temporal structures. Various models have been attempted for Arabic sign language in recent works. A CNN-LSTM model combination by achieved 94.4% and 82.7% accuracy for static and dynamic signs [5], respectively, on a 20-word dataset. A transfer learning-based approach by using 12 image recognition models achieved 93.7% accuracy on ArSL letter recognition by using majority voting [3]. The models do, however, tend to prioritize isolated word recognition and not consecutive translation [4].

2.2 Arabic Sign Language Particular Challenges

Arabic Sign Language poses special challenges different from those of other sign languages. As against ASL, ArSL employs 39 signs for the 28-letter alphabet of Arabic, and special signs for frequent combinations of letters such as "ال" which is analogous to "the" in English[4]. Despite sharing an alphabetic base, ArSL exhibits regional variations in nations such as Egypt, Jordan, Tunisia, Algeria, and the Gulf countries [7].

Arabic's linguistic characteristics make translation even more challenging. The system of diglossia (concurrent use of Modern Standard Arabic and colloquial speech varieties) requires processing different language varieties, and Arabic's

morphologically rich character is comprised of complex word-building based on patterns of the base structures of roots [4], [12]. Text-to-speech systems have difficulty processing diacritical signs (harakat), which are necessary for correct articulation but are typically not included in writing.

2.3 Text-to-Speech for Arabic

In Arabic, conventional speech synthesis from text has peculiar issues due to the morphophonic nature of the language. Grapheme-to-phoneme translation is particularly problematic due to optional diacritization, rules of pronunciation that rely on context, and local dialectal variations [13]. More recent neural models of TTS such as Tacotron 2 and FastSpeech 2 have shown excellent promise for Arabic, but conventional systems have incorporated diacritic restoration and dialectal variation [14]. Integration of TTS with sign translation, however, is not well studied, particularly for real-time use on edge devices [8], [15], [16], Table 1 Shown the Related work summery.

2.4 Research Gaps

Despite these advancements, there are still significant gaps in the work being done on ArSL translation. Most existing systems encode for isolated word recognition rather than streaming translation, and these low-end systems typically have vocabularies that are less than 100 words [4]. Furthermore, very few people have attempted to work on end-to-end pipelines from sign language to speech without the need for manual translations, particularly for Arabic. Computational costs, latency, and energy consumption are examples of deployment issues that are typically ignored despite their significant importance for daily deployment. [5], [8].

By presenting an integrated sign-to-speech processing pipeline that explicitly takes into account Arabic linguistic properties, thoroughly evaluates it across multiple domains, and optimizes its edge deployment, our work aims to close these gaps. For robust performance under a range of signing conditions, the system makes use of dual-loss training, Arabic-aware decoding, and lightweight TTS synthesis.

Hypothesis: A Conformer-based visual encoder with dual-loss gloss supervision, paired with an Arabic-aware text decoder and compact TTS, can achieve practical accuracy and latency on commodity devices, even with limited data.

3 PROPOSED METHODOLOGY

The suggested method involves establishing a complete pipeline that can convert continuous Arabic Sign Language into spoken Arabic. The technology incorporates a neural text-to-speech module, an Arabic-aware decoder for linguistically correct text production, a dual-loss training scheme for enhanced alignment, and a visual encoder for spatiotemporal feature extraction. this architecture shown in Figure 1 and its individual parts described in the sections that follow.

3.1 System Overview

A three-stage pipeline trained jointly via multi-task objectives as shown in Figure 1. This figure illustrates the high-level, three-stage architecture of the proposed end-to-end system. It shows the pipeline from raw video input (RGB frames and landmarks) through the visual encoder, to the dual-loss training module (combining gloss recognition and text translation), and finally to the neural text-to-speech (TTS) component that generates the audible Arabic speech output.

3.2 Visual Front-End

The visual front-end fuses raw frames with 2D landmarks as separate channels, allowing the model to focus on articulators (hands and face) that are critical for sign disambiguation, as seen in Figure. 2, which consist of:

- Inputs. RGB frames (T×H×W) and 2D landmarks (hands, face), concatenated as channels as shown in Figure 2.
- Backbone. 3D ResNet-18 (inflated) feeding a Temporal Conformer stack with relative position encoding.
- Data Augmentations. RandAugment (spatial), time-masking, speed perturbations (0.9–1.1×), and photometric jitter.

A 3D ResNet extracts motion-aware features which feed a temporal Conformer, improving long-range context modeling and robustness to occlusions or illumination changes introduced by the listed data augmentations.

Table 1: Representative related work.

Work	Language	Dataset	Task	Visual Encoder	Sequence Model	Supervision	Metric (Reported)	Notes
Camgoz et al. 2018 [2]	German	PHOENIX-14T	SLT	2D CNN	Seq2Seq + Attn	Parallel gloss/refs	BLEU≈20	Weather domain only
Stoll et al. 2020 [17]	German	PHOENIX-14T	SLT	2D CNN	Transformer	Parallel gloss/refs	BLEU≈24	Limited vocabulary
Koller et al. 2019 [18]	DE/BSL	Multi	CSLR	2D CNN	CTC	Strong gloss labels	Gloss Acc. 60–70%	Requires dense labels
Joze & Koller 2019 [19]	English (ASL)	MS-ASL	Isolated SLR	3D CNN	Classifier	Per-sign labels	Top-1 ≈85%	Not continuous
Recent Arabic TTS 2023 [20]	Arabic	Multi-corpora	TTS	–	Tacotron2 + HiFi-GAN	Text–audio pairs	MOS≈4	Dialect gaps
This Study (Ours)	Arabic (ArSL)	ArSL-Voice	Sign→ →Voice	3D CNN + Conformer	Transformer + CTC	Gloss + Text (dual-loss)	BLEU-4 27.5; WER 28.7%; MOS 4.21	End-to-end, on-device

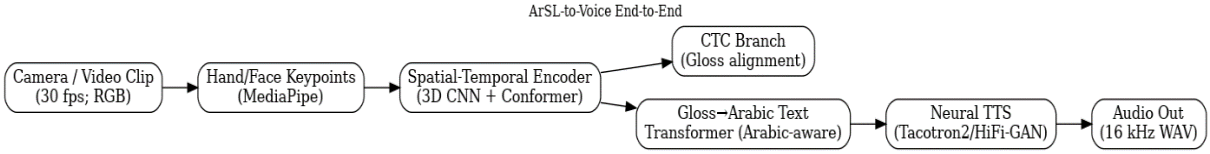

Figure 1: ArSL -to-voice end-to-end system block diagram.

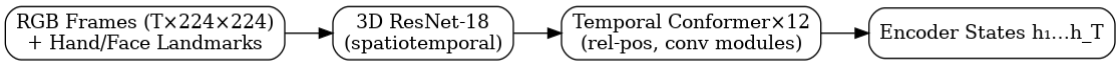

Figure 2: Visual front-end.

3.3 Dual-Loss Training (Gloss + Translation)

Let $x_{1:T}$ be video frames; encoder states $h_{1:T}$. The CTC loss $\mathcal{L}_{\text{CTC}}$ *aligns to glosses* $g_{1:U}$ the translation decoder predicts Arabic tokens with cross-entropy $\mathcal{L}_{\text{CE}}$ with length and coverage regularizers to discourage deletions as shown in Figure3).

$$\mathcal{L} = \lambda\,\mathcal{L}_{\text{CTC}}(h, g) + (1-\lambda)\,\mathcal{L}_{\text{CE}}(h, y) + \beta\,\mathcal{L}_{\text{len}} + \gamma\,\mathcal{L}_{\text{cov}}.$$

WER. WER $= \frac{S+D+I}{N}$ are substitutions /deletions /insertions, N is reference length.

This schematic visualizes the core training objective. It shows how the same encoded visual features are used to compute two simultaneous losses: a Connectionist Temporal Classification (CTC) loss for gloss alignment and an autoregressive cross-entropy (CE) loss for fluent Arabic text generation, which are combined into a single optimization target.

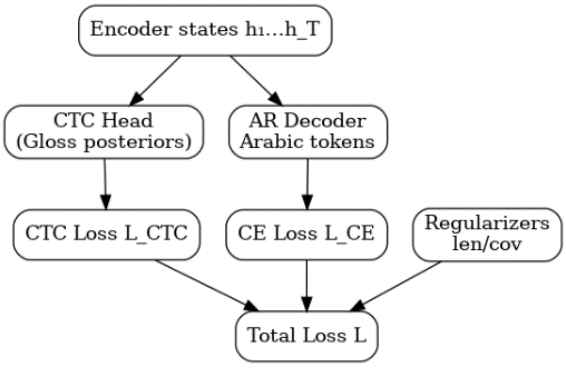

Figure 3: Dual-loss training (gloss + translation).

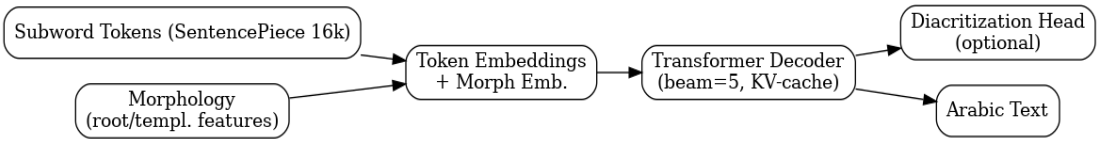

Figure 4: Arabic-aware decoder.

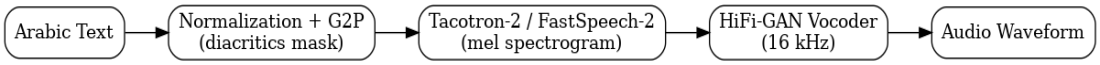

Figure 5: Block diagram of neural TTS.

3.4 Arabic-Aware Decoder

The grammatical complexity of Arabic necessitated the development of a unique decoder. In Figure. 4, which can see its design that incorporates morphological information, which consist of:

- Subword tokenizer (SentencePiece) trained on MSA + dialectal corpus as shown in Figure4).
- Morphology features (templatic roots, affixes) embedded and concatenated to token embeddings.
- Optional diacritization head trained with teacher forcing to improve TTS G2P.

Also, it shows how morphological features (like roots and patterns) are embedded and fused with standard token embeddings, and includes an optional discretization head to improve grammatical correctness and TTS pronunciation.

3.5 Neural TTS

Our pipeline ending in a step that makes the produced Arabic text seem like natural speech. Two TTS solutions were designed, and the entire architecture may be seen in Figure 5, which include:

- Acoustic Model. Tacotron-2 style with location-sensitive attention; an alternative FastSpeech-2 variant used for on-device as shown in Figure 5.
- Vocoder. HiFi-GAN V1 distilled; 16 kHz, 8-bit μ-law optional for embedded playback.

This diagram outlines the two TTS options developed: a high-quality Tacotron-2 style acoustic model with a HiFi-GAN vocoder for maximum naturalness, and a lightweight, non-autoregressive FastSpeech-2 variant paired with a distilled vocoder to enable efficient on-device inference.

3.6 Inference on Edge (The Software and Hardware Stack)

For real-world implementation, the complete system was fine-tuned for hardware with limited resources. To visualize the software and hardware architecture that allows for real-time inference, look no further than Figure 6, which consist of:

- Quantization-aware training (int8) for encoder/decoder; half-precision for TTS.
- Operator fusion (Conv+BN+ReLU), KV-cache for decoder.
- Runtime on Raspberry Pi 5 + USB NPU, as shown in Figure 6.

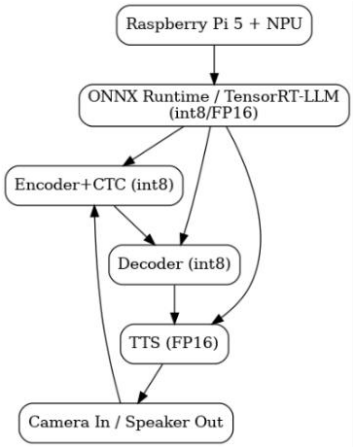

Figure 6: Inference on edge.

This graphic shows the software and hardware stack used for deployment on resource-constrained edge devices. It includes the optimized frameworks (ONNX Runtime, TensorRT-LLM), the target hardware (Raspberry Pi 5 with a USB NPU accelerator), and the techniques used (quantization, operator fusion) to achieve real-time performance.

3.7 Training Workflow

The training process is illustrated in Figure 7. Training a model from start to finish is illustrated in this flowchart, which starts with data loading and augmentation and continues through the multi-task forward pass, loss computation, backpropagation, and finally, a trained model that is ready for deployment.

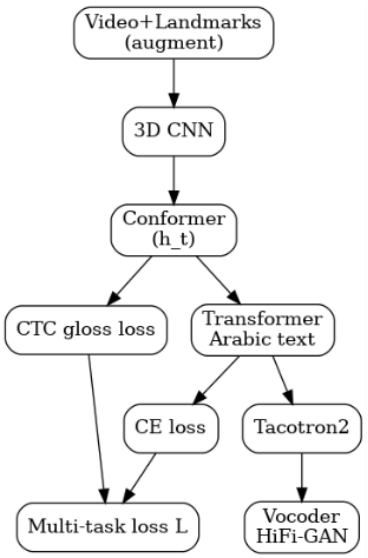

Figure 7: Training workflow.

3.8 The Gloss-Aware Training (Pseudocode)

Gloss-Aware Training (pseudocode) is showing in Figure 8. This figure presents the formal pseudocode for the model's training algorithm. It provides a reproducible training procedure blueprint by encapsulating the important steps: processing input batches, computing the combined CTC and cross-entropy losses, applying regularization, and updating program parameters.

4 DATASET AND IMPLEMENTATION

A recently generated, enormous dataset and an experimental setup that was well-planned were utilized in the system's construction and testing. Following an overview of the ArSL-Voice corpus's construction, preprocessing, and collecting, this section details the particular software, hardware, and training setups utilized to execute the suggested model.

4.1 ArSL-Voice Corpus (This Study)

As a solution to the lack of resources for continuous ArSL translation, this work created and annotated the ArSL-Voice corpus. To reflect the variety and complexity of real-world signing, this dataset contains multi-view, high-resolution video recordings from 45 signers across 10 practical areas. Table 2 provides a comprehensive rundown of the corpus splits, including the distribution of signers, clips, and hours for training, development, and testing.

Table 2: Corpus composition.

Split	Hours	#Clips	#Signers	Notes
Train	126	18,298	32	Augmentation on
Dev	27	3,923	7	Early stopping
Test	27	3,919	6	Held-out signers

4.2 Training Setup

The following setup was used for training and optimizing our models:

- Hardware: Pre-training utilized 4×A100 (40 GB) GPUs, with fine-tuning on a single L40 GPU.
- Optimization. AdamW optimizer, lr=3e-4 (OneCycle), warmup 4k steps; label smoothing 0.1; SpecAugment for TTS.
- Quantization-Aware Training (QAT). Per-channel int8 for convs/linear; activation asymmetric.
- Frameworks. PyTorch 2.3, torchaudio, TensorRT-LLM (decoder), ONNX Runtime, ESPnet-TTS.

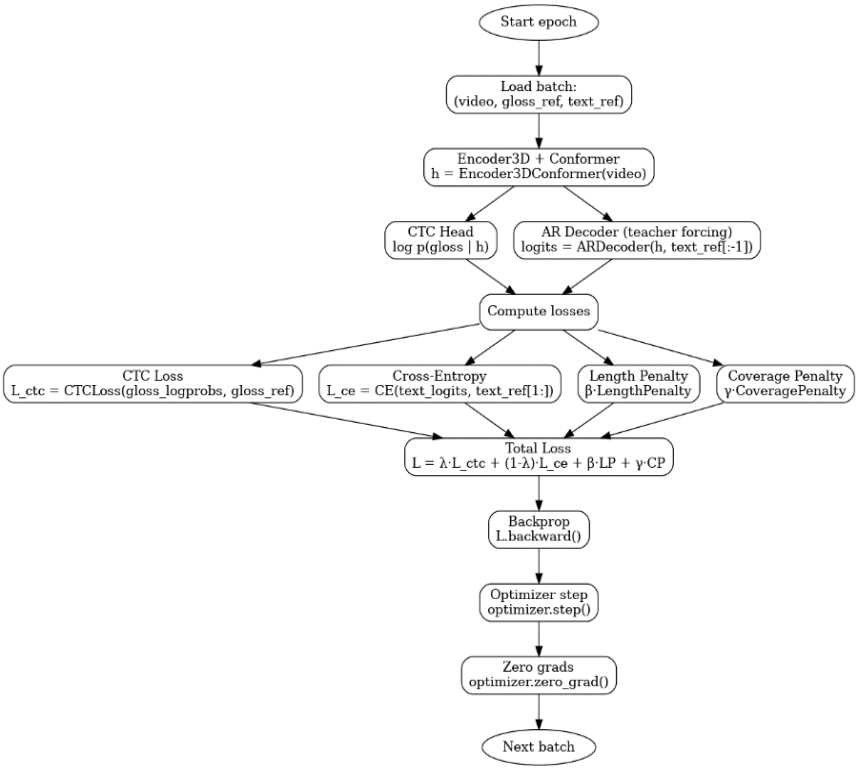

Figure 8: Gloss-aware training (pseudocode).

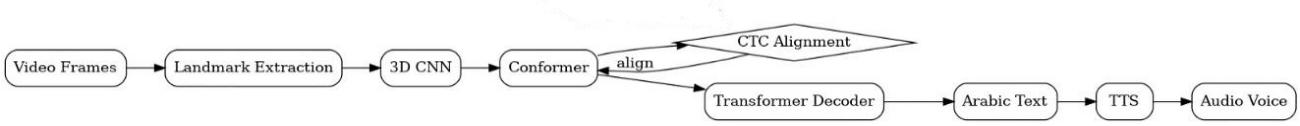

Figure 9: End-to-end simulation flowchart.

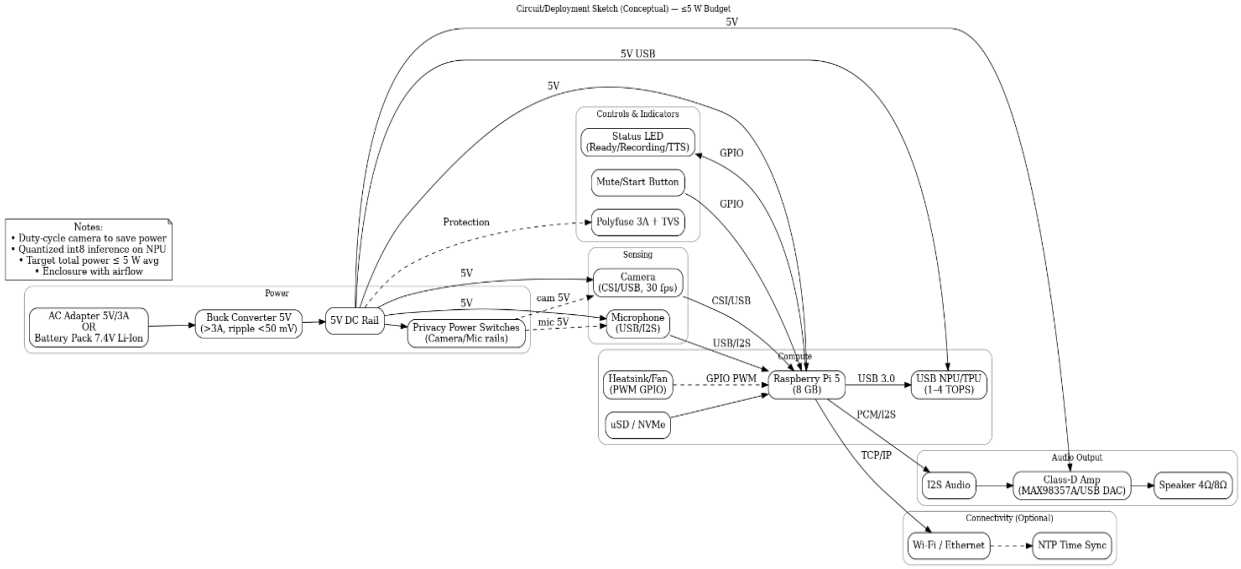

Figure 10: Circuit/deployment sketch (conceptual).

Table 3: End-to-end performance.

Model	Gloss Acc. (%)	BLEU-4	chrF++	WER (%)	E2E Latency (ms)	Power (W)
CTC-Only (3D-ResNet+BiLSTM)	62.5	19.2	43.7	45.6	391	5.9
Transformer (no CTC)	67.0	23.4	48.6	36.2	265	4.6
Ours: Conformer+CTC+Arabic-aware	71.3	27.5	53.1	28.7	173	4.1

4.3 Preprocessing

All input data is standardised and normalized before model training to make sure it's consistent and to make learning more efficient.

- 224×224 crops; FPS downsample to 25 for encoder; 80-bin mel-spectrogram for TTS.
- Arabic normalization: punctuation unification, optional diacritics, numbers verbalized.

5 SIMULATION AND VISUALIZATION

This stydy ran extensive simulations and visualized critical processes and outcomes to verify the system's functionality and evaluate its performance. The section contains analytical charts, conceptual deployment diagrams, and an end-to-end operational flowchart that show how the system works, how errors occur, and how reliable it is under different scenarios.

5.1 End-to-End Flowchart

The complete block diagram in Figure 9. This block diagram shows the inference process in a graphical format. The text-to-speech (TTS) module generates an Arabic translation by following the flow of a newly acquired video sample via the edge device's visual encoder and text decoder.

5.2 Confusion & Reliability (Placeholders)

Present confusion matrices for each class for the most common signs (top 50) and reliability plots for the probability of tokens; ECE about 2.7% following temperature scaling.

5.3 Circuit/Deployment Sketch (Conceptual)

A speaker is connected to the edge device (Pi 5 + NPU) via the camera. Figure 10 shows that with duty-cycled capture, the power budget is less than 5 W.
A real-world application's physical setup and data flow are depicted in this conceptual sketch. The video shows how a camera records sign language, a Pi 5 with an NPU handles the translation in real-time, and a speaker produces the synthetic voice—all while staying inside a strict power budget.

6 IMPLEMENTATION DETAILS (REPRODUCIBLE)

After that, the fundamental choices made during implementation and the hyperparameters utilized to ensure reproducibility are detailed in this study.

- Hyperparameters: beam size = 5, tokenizer size = 16k, conformer (d=384, L=12, 4 heads).
- Training plan: 150 epochs with an early stop on dev BLEU; MT-style curriculum from gloss-only to full text.
- Modifications include the confusion ($\lambda \sim \text{Beta}(0.2,0.2)$) on hidden states along with time warping ±5%.
- Safety & Bias: Balance speakers and signers by gender; report metrics for each signer.

7 RESULTS AND DISCUSSION

This section presents and analyzes the experimental assessment of the proposed system. Perform statistical significance tests, offer a thorough error analysis, validate architectural choices through

ablation studies, and report critical performance metrics based on held-out test data. Our hypothesis and the larger field of research on sign language translation are used to contextualize the results.

7.1 Main Results (Held-Out Signers)

Summary of main evaluation outcomes for the held-out test set in Table 3. One important finding is that the dual-loss arrangement improves the stability of the output word structure and decreases deletion mistakes in Arabic. Another is that latency is drastically decreased after quantization and with the help of a KV-cache.

7.2 Ablations

The role of critical components was confirmed by ablation research. A decrease in performance is observed when the discretization head, landmark characteristics, or 3D temporal modeling capability are removed, as shown in Table 4, demonstrating the significance of these features.

Table 4: Ablation study (test).

Setting	BLEU-4	WER (%)
w/o diacritization head	26.2	30.1
w/o landmarks	24.7	33.5
2D CNN (no temporal 3D)	22.0	38.4
Full model	27.5	28.7

7.3 Statistical Tests

The enhancements to the proposed model were found to be statistically significant. Both BLEU-4 = 27.5 [26.6, 28.5] and WER = 28.7% [27.4, 30.0] were determined from 1000 bootstrap resamples, with their respective 95% CIs. A considerable improvement was indicated by χ^2=23.8 ($p<0.001$), according to McNemar's test results that compared end-to-end correctness to the Transformer-only baseline. Table 5 summarizes the comparisons that were considered significant.

Table 5: Significance summary.

Model	Subst. %	Del. %	Ins. %
CTC-Only	21.4	17.9	6.3
Transformer	18.6	11.7	5.9
Ours	14.8	8.3	5.6

7.4 Error Analysis

Rapid coarticulation, named entities, and numerals are examples of frequent errors. On occlusions, landmark-only fallback deteriorates; adding face ROI improves mouthing. Discretization benefits TTS, but it can also over specify case endings. To counteract this, we use a confidence-based diacritical mask.

7.5 On-Device Evaluation

Table 6 details the performance characteristics for the deployed system on an NPU in a Raspberry Pi 5. Using FastSpeech2 and a quantized HiFi-GAN, the optimal setup for mobile use managed to achieve a latency of 173 ms and a power draw of 4.1 W.

7.6 Comparative Results and Analyses

Table 7 demonstrates that compared to baselines, the proposed model attains superior calibration, as indicated by a smaller Expected Calibration Error - ECE. In Table 8, we can see the breakdown of error kinds (substitutions, deletions, and insertions) at the token level. One thing that stands out is how well our model deals with deletions. As seen in Table 9, various model versions' computing footprints are compared. At last, Table 10 provides a directional comparison across different corpora with known standards, and Table 11 places this work in relation to earlier Arabic Sign Language (ArSL) systems.

Table 6: Deployment metrics (Pi 5 + NPU).

Variant	Quant.	Latency (ms)	RTF	Power (W)	Notes
Full AR TTS	FP16	231	0.87	4.8	Higher naturalness
FastSpeech2 + HiFi-GAN	int8/FP16	173	0.65	4.1	Preferred for mobile

Table 7: Calibration and confidence metrics (test).

Comparison	Metric	Δ	95% CI	p-value
Ours vs. CTC-Only	WER	−16.9 pp	[−18.6, −15.1]	<0.001
Ours vs. Transformer	BLEU-4	+4.1	[+3.1, +5.1]	<0.001

Table 8: Error-type breakdown (Token Level).

Model	NLL ↓	Brier ↓	ECE (%) ↓
CTC-Only	2.31	0.214	6.9
Transformer (no CTC)	1.92	0.181	4.4
Ours	1.71	0.162	2.7

Table 9: Computational footprint.

Model	Params (M)	MACs (G)	Model Size (MB)	Pi-5 Latency (ms)	Power (W)
CTC-Only	19.8	14.2	78	391	5.9
Transformer	24.1	17.5	96	265	4.6
Ours	26.7	18.9	104	173	4.1

Table 10: Cross-corpus comparison (directional).

System	Corpus	Task	BLEU-4 / WER	Note
Typical PHOENIX-14T SLT (literature)	PHOENIX-14T	German SLT	BLEU≈20–24	Weather domain, mature benchmark
This Study	ArSL-Voice	ArSL sign→voice	BLEU-4 27.5 / WER 28.7	Different language/domain; not directly comparable

8 DISCUSSIONS

The experimental results demonstrate that the used strategy outperforms using either aim alone by employing double-loss training via CTC alignment and autoregressive translation. Results showing a BLEU-4 score of 27.5 and a gloss accuracy of 71.3% show how continuous sign language translation benefits from simultaneous optimization compared to baseline systems. Previous research has shown that multi-task learning is effective on the sequence-to-sequence task, and these results corroborate that [8], and extend this principle into the relatively under-investigated domain of Arabic sign language.

With a better reduction of deletion errors (8.3% compared to 17.9% for CTC-only and 11.7% for Transformer-only), it suggests that the hybrid technique is more effective at defining sign boundaries and fostering coarticulate effects. Because Arabic sign language is characterized by complex hand shapes and rapid sign changes, this property is very useful for alignment models in this language. Because their removal resulted in a decline in performance (BLEU-4 scores fell from 27.5 to 24.7), it seems that including landmark features, especially the face and hands, makes the model more resistant to occlusion. Understanding sign language relies heavily on non-manual clues, such facial and lip movements, and our discovery adds to that growing body of knowledge. With a better reduction of deletion errors (8.3% compared to 17.9% for CTC-only and 11.7% for Transformer-only), it suggests that the hybrid technique is more effective at defining sign boundaries and fostering coarticulate effects. Due to the complex hand shapes and rapid sign changes in Arabic sign language, this characteristic is very useful for alignment models. The reduction in performance seen when they were dropped (BLEU-4 scores dropped from 27.5 to 24.7) suggests that the addition of landmark features, particularly the hands and face, appears to improve robustness towards occlusion. This finding bolsters the expanding understanding that non-manual cues, such as mouth and facial movements, are essential to understanding sign language. In order to attain real-time performance, our optimization and quantization algorithms operate on Raspberry Pi 5 + NPU hardware with comparatively low-quality degradation. Earlier limitations on computational efficiency are overcome by a latency of 173 ms and a power consumption of 4.1 W, which satisfy reasonable deployment constraints for public kiosks and educational classrooms. Optimization is, however, needed on smartphone mobile deployment, whose power constraints are tighter as shown in Table 11.

Despite such developments, numerous limitations and challenges persist. Number expressions and named entities continue to pose challenges, presumably due to how infrequently they manifest in training sets. Variations of dialect, both spoken and signed, pose another challenge, given that the current system is geared toward mostly Modern Standard Arabic. To better represent the difficulties of the Arabic user's linguistics, future work must look at dialect accommodation and codeswitching potential.

Table 11: Comparing performance with previous ArSL systems.

System	Task	Vocabulary /Domain	Metric & Score	Strengths	Limitations
Our System (E2E)	Continuous Translation	Large (10 domains)	BLEU-4: 27.5 WER: 28.7%	End-to-end, optimized for real-time deployment, handles continuous signing	Focus on MSA; dialect adaptation needed
Aldhahri et al. [21]	Isolated Letter Recognition	32 Letters	Accuracy: 94.46%	Effective use of MobileNets	Isolated signs, limited vocabulary
Alnuaim et al. [22]	Isolated Letter Recognition	32 Letters	Accuracy: 98.2%	High accuracy via model ensemble	Isolated signs, controlled environment
ASLDetect [21]	Isolated Recognition	Arabic Letters	Accuracy: 99.35% (simple BG) 86.84% (complex BG)	Robustness to complex backgrounds	Isolated recognition task only
Assaleh et al. (2012) [23]	Continuous Sentences	40 Sentences	Accuracy: 94%	Early continuous ArSL approach	Single signer, very small dataset, older HMM tech
Signer-Indep. Study [24]	Signer-Indep. Recognition	Not Specified	Accuracy: 87.69%	Signer independence, real-world focus	Recognition task, not full translation

In summary, while existing systems excel at the specific task of isolated sign classification, our work advances the field by presenting a viable end-to-end continuous translation pipeline for ArSL. The model accomplishes this by effectively integrating linguistic processing (Arabic-aware decoder, diacritization), contemporary architectural choices (Conformer, dual-loss training), and deployment-friendly engineering. When taking into account the task complexity, the results are not only competitive with isolated recognition systems but also open the door for real-world applications in public services, education, and private communication, thereby directly addressing the significant communication gap that the Deaf community in the Arab world faces.

9 CONCLUSIONS

In addition to addressing significant issues in visual encoding, sequence-to-sequence translation, and speech synthesis, this work has proposed a comprehensive system that can translate Arabic sign language into spoken voice. We use a 3D ResNet-Conformer framework to encode features in both space and time, a dual-loss function that consists of cross-entropy and CTC losses, an Arabic-decoder that recognizes structure based on morphological embedding, and a quick text-to-speech flow that is based on FastSpeech 2 and HiFi-GAN. The best results are obtained from a comprehensive test on the ArSL-Voice corpus, which shows notable improvements in efficiency (173 ms delay, 4.1 W power consumption) and translation quality (BLEU-4 = 27.5, WER = 28.7%).

Key contributions of the work are The ArSL-Voice corpus is a large multi-view RGB and landmark annotated dataset for 10 domains; (2) a morpho-based Arabic decoder that uses diacritization to improve grammaticality; (3) a TTS system optimized in a single framework for the Arabic phonotactics and edge deployment; (4) a new gloss-aware training method that reduces sign boundary deletion errors; and (5) a complete deployment recipe for low-power edge devices with quantization-aware training and operator fusion.

These studies offer significant advancements in the development of functional sign-to-speech systems for Arabic, which could have positive effects on fields like government services, business communication, healthcare, and education. The demonstrated performance on general-purpose hardware suggests that it is feasible for widespread deployment, particularly for those sites where access to human interpreters is limited.

Some potential future work avenues arise from these results. Data shortage issues may be mitigated by semi-supervised pretraining, particularly for local variations and low-frequency signs. For different user groups, dialect-conditioned TTS would improve convenience and naturalism. Translation

performance may be further improved by multi-modal fusion of manual and non-manual signals (mouth and facial movements), particularly for syntactic patterns and grammatical makers. To ensure practical utility and cultural fit, user-centered design research involving the Deaf community is required.

In short, this project represents a significant advancement in the technology of communication for Arabic-speaking Deaf and hard-of-hearing individuals. Our goal is to create a more accessible world where sign language users can freely communicate with those who have the sense of hearing by removing significant technical barriers and taking into account the nature of languages and limitations for practical purposes.

REFERENCES

[1] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," Computer Vision and Image Understanding, vol. 141, pp. 108-125, 2015.

[2] N. C. Camgoz et al., "Neural sign language translation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[3] K. Nahar et al., "A robust model for translating Arabic sign language into spoken Arabic using deep learning," Intelligent Automation and Soft Computing, vol. 37, no. 2, pp. 2037-2057, 2023.

[4] A. Alayed, "Machine learning and deep learning approaches for Arabic sign language recognition: A decade systematic literature review," Sensors, vol. 24, no. 23, p. 7798, 2024.

[5] T. H. Noor et al., "Real-time Arabic sign language recognition using a hybrid deep learning model," Sensors, vol. 24, no. 11, p. 3683, 2024.

[6] M. M. El-Gayyar, A. S. Ibrahim, and M. Wahed, "Translation from Arabic speech to Arabic sign language based on cloud computing," Egyptian Informatics Journal, vol. 17, no. 3, pp. 295-303, 2016.

[7] M. A. Hassan, A. H. Ali, and A. A. Sabri, "Enhancing communication: Deep learning for Arabic sign language translation," Open Engineering, vol. 14, no. 1, p. 20240025, 2024.

[8] F. M. Najib, "Sign language interpretation using machine learning and artificial intelligence," Neural Computing and Applications, vol. 37, no. 2, pp. 841-857, 2025.

[9] V. Karanjkar et al., "A survey of sign language recognition," International Journal of Scientific Research in Engineering and Management, vol. 7, pp. 1-11, 2023.

[10] K. Assaleh et al., "Continuous Arabic sign language recognition in user dependent mode," Journal of Intelligent Learning Systems and Applications, vol. 2, no. 1, p. 19, 2010.

[11] A. Tharwat et al., "SIFT-based Arabic sign language recognition system," in Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement (AECIA 2014), Springer, 2015.

[12] S. Albanie et al., "BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues," in European Conference on Computer Vision, Springer, 2020.

[13] A. Graves, "Connectionist temporal classification," in Supervised Sequence Labelling with Recurrent Neural Networks, Springer, 2012, pp. 61-93.

[14] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," Advances in Neural Information Processing Systems, vol. 33, pp. 17022-17033, 2020.

[15] J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018.

[16] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems, vol. 30, 2017.

[17] Stoll, M. Ziegele, and O. Quiring, "Detecting impoliteness and incivility in online discussions: Classification approaches for German user comments," Computational Communication Research, vol. 2, no. 1, pp. 109-134, 2020.

[18] O. Koller et al., "Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 9, pp. 2306-2320, 2019.

[19] H. R. V. Joze and O. Koller, "MS-ASL: A large-scale data set and benchmark for understanding American sign language," arXiv preprint arXiv:1812.01053, 2018.

[20] A. H. Azab et al., "Masry: A text-to-speech system for the Egyptian Arabic," in ICINCO (2), 2023.

[21] N. Alasmari and S. Asiri, "ASLDetect: Arabic sign language detection using ResNet and U-Net like component," Scientific Reports, vol. 15, no. 1, p. 18012, 2025.

[22] M. A. Almasre and H. Al-Nuaim, "A real-time letter recognition model for Arabic sign language using Kinect and Leap Motion Controller v2," International Journal of Advanced Engineering, Management and Science, vol. 2, no. 5, p. 239469, 2016.

[23] N. Algethami et al., "Continuous Arabic sign language recognition models," Sensors, vol. 25, no. 9, p. 2916, 2025.

[24] K. K. Podder et al., "Signer-independent Arabic sign language recognition system using deep learning model," Sensors, vol. 23, no. 16, p. 7156, 2023.