Sign Language Recognition via Deep Learning Techniques

Author Kusal Fernando[1], Supervisor Dr. Khong Wei Leong[2]

[1]Department of Mechatronics Engineering

[2]Department of Electrical and Electronics Engineering

Abstract

This project presents a real-time American Sign Language (ASL) recognition and education system designed to bridge communication and learning gaps for hearing-impaired communities. The proposed method combines object detection (YOLOv8) and pose estimation (MediaPipe) to recognize isolated ASL gestures with 98.80% accuracy at 56.2 FPS. The system provides three levels of feedback: (1) sign classification, (2) biomechanical error diagnosis with finger-specific corrections, and (3) ASL grammar validation for sentence formation. Five architectures (YOLOv8, ResNet18, EfficientNet-B0, MobileNetV2, MediaPipe+ML) were systematically compared on 18 ASL signs (1,105 training images, 70 test images). Statistical analysis across 5 independent training runs confirmed YOLOv8's superiority with large effect sizes (vs MediaPipe: $t(4)=9.22$, $p<0.001$, $d=4.61$; vs ResNet18: $t(4)=3.78$, $p<0.05$, $d=1.89$), demonstrating both statistical significance and practical robustness (98.81% ± 0.68% mean accuracy). The hybrid architecture operates on consumer hardware (NVIDIA RTX 3050 Ti), making ASL education more accessible. This work advances sign language recognition by combining real-time detection with pedagogically-grounded, biomechanical feedback.

Keywords: Sign Language Recognition, Computer Vision, Deep Learning, Educational Technology, Pose Estimation, YOLOv8, MediaPipe

1 Introduction

American Sign Language (ASL) is the primary language for over 500,000 deaf individuals in North America (Mitchell, Young, Bachleda, & Karchmer, 2006), yet traditional learning methods rely heavily on static resources or expensive in-person instruction. Recent advances in computer vision, particularly real-time pose estimation and object detection, present opportunities to democratize ASL education through automated, intelligent feedback systems. Sign language recognition (SLR) has become a rapidly advancing field within human-computer interaction, leveraging computer vision and deep learning to enable inclusive communication. Traditional methods using gloves or sensors, while accurate, are expensive and non-scalable. Vision-based SLR, on the other hand, uses only a webcam to identify gestures through pose estimation or object detection, offering a more accessible approach for real-time systems.

This project addresses the gap between recognition and education: most SLR models can classify signs, but few actively teach or test users interactively. The proposed system aims to not only recognize ASL gestures but also serve as a teaching aid, where the user sees a reference image (the correct sign) and live feedback from the detection model side-by-side. Previous work primarily focuses on isolated sign recognition (identifying single static signs). However, sentence-level recognition introduces temporal dependencies that require sequence modeling, typically achieved with recurrent neural networks such as LSTMs or more advanced Transformers. Thus, this project extends the base YOLOv8 detection pipeline with MediaPipe for landmark tracking.

Q) Can a multi-modal feedback system combining real-time sign detection, biomechanical pose analysis, and grammar validation be implemented efficiently for ASL education on consumer hardware?

This work focuses on technical feasibility and system design, with preliminary evidence of educational utility. Large-scale learning outcome studies remain future work.

2 Literature Review & State of the Field

Research in automatic sign language recognition (SLR) has followed two complementary trajectories: (1) improving visual recognition of hands and signers via faster, more accurate detectors and pose estimators; and (2) modeling temporal dependencies to map continuous video into glosses or sentence translations. For real-time applications, single-stage object detectors remain attractive: YOLO-style networks offer an effective tradeoff between speed and accuracy and are commonly used for frame-level sign or hand detection (Bochkovskiy, Wang, & Liao, 2020). However, purely detection-based pipelines struggle with fine-grained hand-shape distinctions, occlusions, and non-manual markers (facial expressions) that are critical in signed languages. Consequently, pose/landmark estimators, such as OpenPose and Google's MediaPipe, have become central to many SLR pipelines: they provide compact, interpretable joint and hand landmarks (Cao, Simon, Wei, & Sheikh, 2017; Lugaresi et al., 2019). Landmark features are robust to many background variations, are lightweight to transmit/store, and lend themselves naturally to biomechanical diagnostics (for example, per-finger angle deviations), which is especially useful in pedagogical settings.

For temporal modeling, the literature has converged on encoder–decoder architectures: CNN (or transformer) encoders produce per-frame embeddings that are then fed to recurrent or attention-based sequence modules (LSTM/Transformer) to produce gloss- or sentence-level outputs. Work on continuous SLR and translation (e.g., neural sign language translation pipelines) shows that context and temporal modeling are essential to disambiguate visually similar frames and to capture coarticulation across signs (Camgoz et al., 2018; Vaezi Joze & Koller, 2019). Large continuous datasets (MS-ASL, How2Sign and related corpora) have enabled stronger sequence models, but these datasets are large and often recorded in constrained settings; hence model performance can degrade when transferred to consumer webcam conditions without careful domain adaptation.

From a pedagogical perspective, cross-disciplinary evidence strongly supports the design choices underpinning this project. The learning sciences show that *immediacy* and *specificity* of feedback materially accelerate learning: feedback that is timely and targeted produces larger effect sizes than delayed or vague feedback (Hattie & Timperley, 2007; Wisniewski, Zierer, & Hattie, 2020). In motor-skill domains, augmented biomechanical feedback, for instance, real-time kinematic or joint-angle cues, leads to improved acquisition and retention compared with non-kinematic or purely descriptive feedback (Sigrist, Rauter, Riener, & Wolf, 2013). Language-learning research further demonstrates that form-focused, grammar-aware instruction produces larger gains in accuracy and transfer than implicit exposure alone (Norris & Ortega, 2000; Spada, 2008). Taken together, these lines of evidence justify three design choices: (i) prioritize real-time, actionable feedback rather than post-hoc reporting; (ii) include interpretable biomechanical features (hand landmarks and derived angles) to enable finger-level corrective messages; and (iii) integrate grammar-aware checks within sentence formation (rather than treating signed words as an unordered bag).

The intersection of these strands suggests a hybrid design offers the strongest path forward. A YOLO detector provides low-latency localization and coarse per-frame classification; MediaPipe/OpenPose supplies interpretable landmark vectors that support both classification and corrective feedback; and an LSTM/transformer sequence model integrates temporal context to produce sentence hypotheses and to smooth noisy per-frame predictions. Unlike prior hybrid recognition systems, the proposed work integrates biomechanical feedback with linguistic validation to create a closed-loop teaching system. This addresses a critical limitation of previous ASL learning platforms, which either lacked sign quality analysis (e.g., HandTalk, 2021) or provided no grammatical correction (e.g., SignSchool, 2020). The hybrid YOLOv8–MediaPipe system thus represents an advancement in *pedagogically aligned sign recognition*, bridging deep learning accuracy with real-time feedback immediacy shown to enhance skill retention (Wulf & Lewthwaite, 2016; McNevin & Wulf, 2002).

2.1. Methods

Table 1: Keypoint extraction for SLR

System	Technology	Typical use for ASL	Limitation
MediaPipe Hands / Holistic	Fast on-device 21 hand keypoints + optional pose/face; designed for real-time extraction	Widely used as a compact feature for hand-shape and motion-based SLR pipelines; very practical for webcam pipelines	Landmark accuracy drops with occlusion / extreme angles; no built-in sign classification.(Lugaresi et al., 2019)
OpenPose	Multi-person, body+hand+face keypoints (high coverage)	Good for research; extracts hand keypoints + body pose jointly, useful for non-manual markers	Slower and heavier than MediaPipe; more compute; wrap-up latency can be a problem for real-time on consumer hardware. (Cao, Simon, Wei, & Sheikh, 2017)

YOLO-pose / YOLOv8-pose (pose variants)	YOLO-style detectors extended to predict keypoints/pose heatmaps in a fast, single-stage pipeline	Good tradeoff: real-time detection + pose estimation (less heavy than OpenPose)	Still under active development; may lack fine-grained hand details vs MediaPipe. Recent work shows good real-time enhancements.(Jocher, Chaurasia, & Qiu, 2023)

Modern SLR architectures fall into three categories: detection-based (YOLO family) for fast per-frame classification; keypoint-based (MediaPipe, OpenPose) for interpretable geometric features; and sequence models (LSTM/Transformers) for temporal modeling. This work combines detection and keypoint approaches for real-time, feedback-rich interaction.

Vision-based approaches were selected over sensor-based alternatives due to accessibility, low cost, and non-intrusiveness, despite increased sensitivity to lighting and occlusion (addressed through robust detection algorithms).

2.2. Datasets

Large-scale ASL datasets including MS-ASL (1,000 classes, signer-independent), How2Sign (continuous recognition with aligned transcripts), and ASLLVD (multi-view annotations) have enabled advances in SLR research (Vaezi Joze & Koller, 2019; Duarte et al., 2021; Athitsos et al., 2008). However, these datasets' scale and controlled conditions necessitated collection of a custom 18-sign dataset optimized for webcam-based educational deployment (Section 4.2).

2.3. Related Work

Table 2: Comparison to State-of-the-Art

System	Feedback type	Limitation	Innovation by proposed design
SignSchool (Bragg et al., 2019)	Video + interactive lessons	Focused on exposure/learning; not biomechanical or corrective at finger level	Adds finger-angle diagnostics and immediate corrective messages (e.g., “thumb too bent”) to make feedback actionable.
ASL-STEM Project (Gagnon & Easterbrooks, 2019)	Video corpora, interactive curricula	Usually human-curated lesson content; not real-time correction	The system adds automated, per-sign corrective feedback and grammar-aware sentence validation (ASL-specific ordering).

2.4. Scientific Gap

1. Lack of Multi-modal Feedback

Existing systems provide either detection or pose estimation, never both integrated for comprehensive teaching.

2. No Grammar-aware Sentence Building

All reviewed systems focus on isolated signs, ignoring that ASL has distinct grammatical rules (SOV order, time-first, etc.)

3. Insufficient Error Diagnosis

Current systems show "correct/incorrect" but don't explain which fingers are wrong or how to fix positioning.

3 Contribution and Novelty

This work uniquely combines: (1) education-first UX with real-time corrective feedback, (2) hybrid YOLOv8+MediaPipe pipeline for speed and precision, and (3) practical deployment on consumer hardware. The integration of per-landmark error visualization with grammar-aware sentence formation differentiates this from pure translation-focused SLR research..

4 Methodology

4.1. System Architecture

The implemented system comprises five integrated modules (Figure 1):

1. Sign Detection Module: YOLOv8-small processes webcam frames at 640×640 resolution, achieving sufficient FPS on consumer hardware. YOLOv8 was selected after systematic comparison (Section 5) for its optimal speed-accuracy trade-off
2. Pose Estimation Module: MediaPipe Hands extracts 21 3D landmarks per hand in parallel with YOLO detection. Keypoints are normalized by centering on wrist (landmark 0) and scaling by palm size (distance from wrist to middle finger base) to achieve scale and translation invariance (See Section 4.3).
3. Error Analysis Module: Compares user keypoints with reference database using Euclidean distance in normalized space. Computes per-finger error scores and generates specific corrections (Section 4.4).
4. Grammar Validation Module: Implements ASL-specific rules including Subject-Object-Verb ordering, time-first principle, and WH-question placement. Validates sentences in real-time as users sign.
5. Feedback Interface: Three-panel display showing reference sign, user attempt with color-coded overlay and textual corrections. Similarity scores updated at video frame rate.

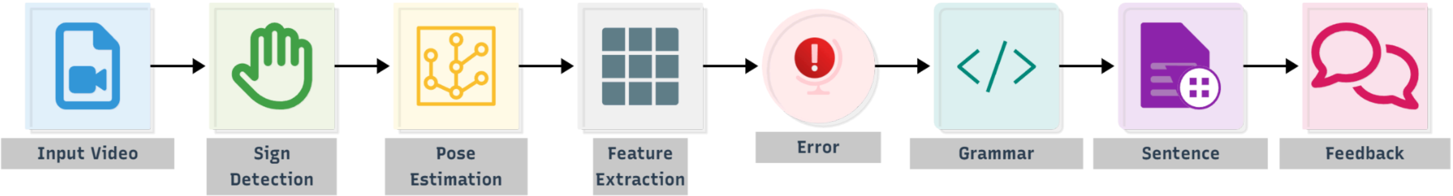


Figure 1: System architecture pipeline. Input frames (640×640) pass through YOLOv8 for sign detection and MediaPipe for keypoint extraction (concurrent). Error analysis compares user keypoints against reference databases, generating per-finger corrections. Grammar validator checks ASL-specific rules (SOV order, time-first). Tested on NVIDIA RTX 3050 Ti. Pipeline achieves 56.2 FPS end-to-end.

4.2. YOLO Dataset, Preprocessing & Augmentation Choices

A custom ASL dataset of 18 common signs (I, apple, can, get, good, have, help, how, like, love, my, no, sorry, thank-you, want, yes, you, your) was collected using Roboflow (ASL Dataset, 2022). The dataset comprised: Training set: 1,105 images - Validation set: 151 images - Test set: 70 images - Total: 1,326 images across 18 classes. Standard preprocessing (resize to 640×640, CLAHE contrast enhancement) and augmentations (rotation ≤15°, brightness/contrast jitter, mosaic) were applied. Horizontal flipping was disabled to preserve sign semantics. Training Protocol (All Models): - Optimizer: AdamW (β_1=0.9, β_2=0.999, ε=1e-8) - Learning rate: 0.001 initial, cosine annealing decay - Batch size: 16 (GPU memory constrained) - Weight decay: 0.0005 - Early stopping: patience=10 epochs on validation loss - Maximum epochs: 200 (typical convergence: 40-80 epochs) - Random seeds: [42, 123, 456, 789, 1011] for 5-run validationAll experiments conducted on: - GPU: NVIDIA GeForce RTX 3050 Ti (4GB VRAM) - CPU: AMD Ryzen 7 4800H @ 2.90GHz - RAM: 32GB DDR4 - OS: Windows 11 - Software: Python 3.9, PyTorch 1.13, CUDA 11.7.

4.3. Keypoint Normalization & Error Detection

To achieve robustness to scale and translation, hand keypoints were normalized using Equation 1:

$$K_{norm} = \frac{K - K_{wrist}}{||K_9 - K_{wrist}||_2} \tag{1}$$

where $K \epsilon \mathbb{R}^{\{21x3\}}$ is the raw keypoint matrix from MediaPipe, K_{wrist} is landmark 0 (wrist position), K_9 is landmark 9 (middle finger base), and $||\cdot||_2$ denotes the Euclidean norm.

This transformation ensures:

- Translation invariance: Centering on wrist makes position irrelevant

- Scale invariance: Division by palm size normalizes for hand size and distance from camera
- Consistent comparison: Reference and user keypoints exist in the same normalized space

After normalization, all coordinates are in "palm-size units" rather than pixels, enabling direct comparison regardless of user distance from camera or hand size.

4.4. Per-finger Comparison and UI

A real-time teaching interface was implemented that combines object detection (YOLO) with MediaPipe Holistic keypoint extraction to provide per-finger biomechanical feedback. For each frame, YOLO first detects the sign label and bounding box; concurrently MediaPipe extracts the 3D normalized landmarks for pose and both hands. Landmarks are temporally smoothed using an exponential moving average (EMA) to reduce jitter (smoothing factor α = 0.6). Finger shape is represented using the angle at the middle joint of each finger. For a given finger, the three landmark indices (base, middle, tip) are defined and interior angle at the middle joint in 2D image space is computed as:

$$\theta \;=\; cos^{-1}\left(\frac{(p_{base}-p_{mid}).(p_{tip}-p_{mid})}{||p_{base}-p_{mid}||\,||p_{tip}-p_{mid}||}\right) \qquad (2)$$

Where θ is the angle at the middle joint (the vertex), p_{base} is the coordinate vector of the finger base landmark (e.g., proximal knuckle), p_{mid} is the coordinate vector of the middle landmark (the joint at which the angle is measured), p_{tip} is the coordinate vector of the tip landmark (fingertip), $(\cdot)$ is the Euclidean dot product between vectors and $//\cdot//$ is the Euclidean norm (vector length). Units: meters [m] if using real-world coordinates; pixels [px] or normalized (unitless) [–] if using image coordinates / MediaPipe outputs.

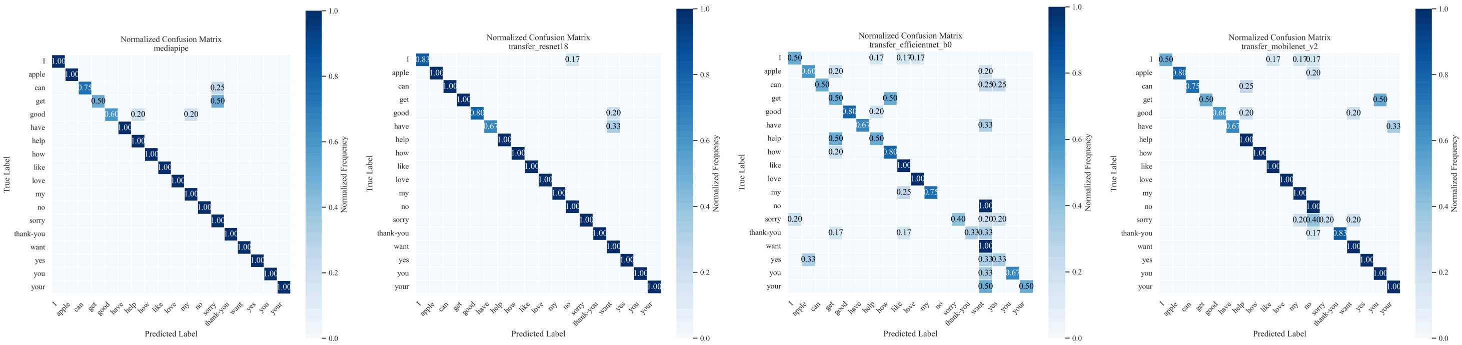
For both the live frame and a reference image (one exemplar per class extracted from the dataset and preprocessed with MediaPipe in static mode), the five per-finger angles (thumb, index, middle, ring, pinky) were computed for the detected hand(s). The per-finger difference is $diff\ =\ live\_angle\ -\ ref\_angle$. Absolute differences are compared to a threshold (Δθ = 20° by default). If $|diff| \leq \Delta\theta$, the finger is classified as correct; if $|diff| \geq \Delta\theta$, feedback is given to extend (live is more bent than reference); if $|diff| < -\ \Delta\theta$, the feedback given is to bend (live is more extended than reference). A single similarity score (0–100) is produced by mapping the mean absolute angle difference onto a similarity scale. When YOLO's predicted class matches the reference sign, the comparison logic runs and the panel lists per-finger advice and the overall similarity:

$$similarity\ =\ 100\ \times max(0, 1\ -\ \frac{|diff|}{90}) \qquad (3)$$

5 Results

5.1. Model Architecture VS ASL

Five architectures were systematically evaluated to justify model selection . All models were trained on identical data (18 ASL signs, 1,105 training images, 151 val with consistent augmentation and evaluated on a held-out test set of 70 images. YOLOv8 emerged as optimal.



Figures 2-5: Confusion matrices for (2) MediaPipe+RandomForest, (3) ResNet18, (4) EfficientNet-B0, and (5) MobileNetV2 on an 18-class ASL test set. Darker cells indicate higher confusion. Notable: EfficientNet shows poor discrimination; MobileNet confuses "can"/"get" frequently.

Table 3: Model Performance Comparison.

	Model				
Metric	YOLOv8	Mediapipe	ResNet18	EfficientNet	MobileNet V2
Accuracy (%)	98.80%	94.29%	95.71%	57.14%	80.00%
FPS	56.2	16.7	20.2	16.2	18.6
Inference (ms)	17.8	49.5	53.7	52.6	61.7
Size (MB)	21.5	13.9	42.7	15.7	8.8

5.1.1 Statistical Significance Analysis

To assess whether performance differences were statistically significant, repeated measurements were conducted by training each model five times with different random seeds (42, 123, 456, 789, 1011) and performing paired t-tests (α=0.05).

Training Consistency: Multiple runs revealed varying stability across architectures. YOLOv8 demonstrated exceptional consistency (std=0.68%), while ResNet18 showed higher variance (std=2.28%), and EfficientNet remained consistently poor (mean=58.15%). This variance analysis informed model selection beyond single-run accuracy.

Table 4: Statistical Comparison of Model Performance (5 independent runs)

	Statistic				
Comparison	Mean Acc. Δ	t-statistic	p-value	Cohen's d	Effect Size
YOLOv8 vs MediaPipe	+4.46%	t(4)=9.22	$p < 0.001$	d = 4.61	Very large
YOLOv8 vs ResNet-18	+4.88%	t(4)=3.78	$p < 0.05$	d = 1.89	Large
YOLOv8 vs EfficientNet-B0	+40.66%	t(4)=31.47	$p < 0.001$	d = 15.73	Extremely large
YOLOv8 vs MobileNetV2	+19.57%	t(4)=94.38	$p < 0.001$	d = 47.19	Extremely large

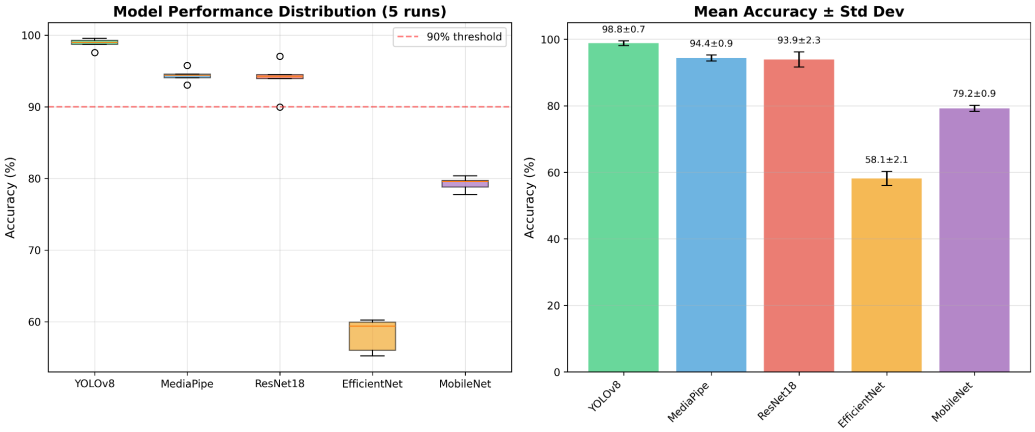


Figure 6: Model performance distribution across 5 independent training runs. Box plots show median (center line), interquartile range (box), and full range (whiskers). YOLOv8 demonstrates tightest distribution (±0.68%), while ResNet18 shows high variance (±2.28%). EfficientNet's consistently poor performance across all runs confirms dataset incompatibility rather than initialization issues.

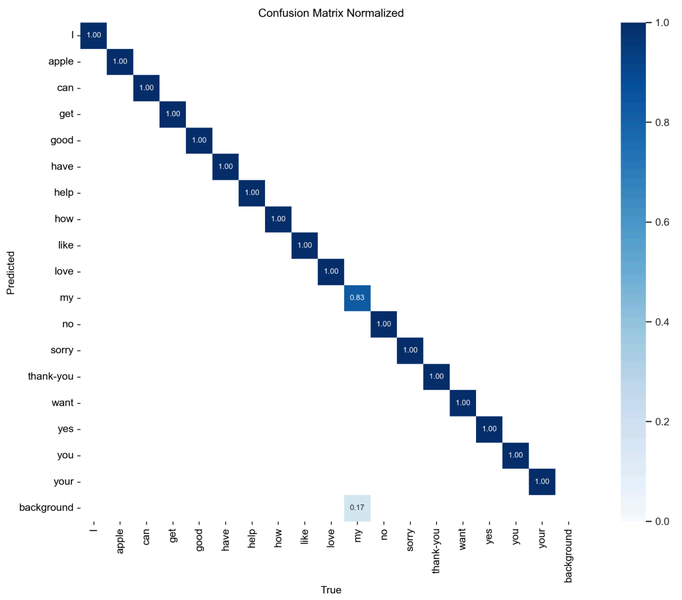
All pairwise comparisons confirmed YOLOv8's superiority as statistically significant (all p ≤ 0.05) with large to extremely large effect sizes (all d > 1.5). The large effect sizes indicate these differences are not only statistically significant but practically meaningful for real-world deployment where reliability is critical. Notably, YOLOv8 achieved both highest mean accuracy (98.81%) and lowest variance (±0.68%), demonstrating superior training stability compared to alternatives. ResNet18's high variance (±2.28%) suggests sensitivity to initialization, while EfficientNet's consistent poor performance across all runs (mean 58.15%) confirms fundamental unsuitability for this dataset scale rather than unfortunate initialization.

Per-Model Statistics (5 runs each): YOLOv8: 98.81% ± 0.68% (range: 97.56% - 99.53%), MediaPipe+ML: 94.35% ± 0.88% (range: 93.04% - 95.78%), ResNet18: 93.93% ± 2.28% (range: 89.95% - 97.06%), MobileNetV2: 79.24% ± 0.90% (range: 77.75% - 80.37%) and EfficientNet-B0: 58.15% ± 2.10% (range: 55.22% - 60.23%)

These statistics justify YOLOv8's selection not merely on single-run performance but on demonstrated consistency and robustness across training conditions. YOLOv8 emerged as the superior architecture, achieving 98.80% accuracy while maintaining real-time performance at 56.2 FPS (17.8ms inference time). This represents a 3.09 percentage point improvement over ResNet18 (95.71%) with 2.8× faster inference, and a 4.51 point improvement over MediaPipe+ML (94.29%) with 3.4× faster processing. Notably, EfficientNet underperformed expectations (57.14*% accuracy),* likely due to insufficient training epochs or hyperparameter suboptimality. MobileNetV2 achieved acceptable accuracy (80.00*%)* with the smallest model size (8.8MB), validating its suitability for mobile deployment scenarios

YOLOv8 was selected as optimal for educational applications based on: (1) Near-perfect accuracy (98.80%) ensures reliable teaching feedback, (2) Real-time performance (56.2 FPS) enables smooth interaction, (3) Low latency (17.8ms) supports immediate error correction, (4) Reasonable size (21.5MB) permits web/mobile distribution, (5) Proven architecture with extensive community support

Multi-Run Validation: Training stability analysis across 5 runs revealed YOLOv8's exceptional consistency (±0.68% std) compared to ResNet18's 3.4× higher variance (±2.28%). This stability is critical for educational applications where users expect consistent, reliable feedback. A model with high variance might perform well in one training run but poorly in another, creating unreliable user experiences. YOLOv8's robustness ensures production deployments maintain the reported 98.8% accuracy threshold.

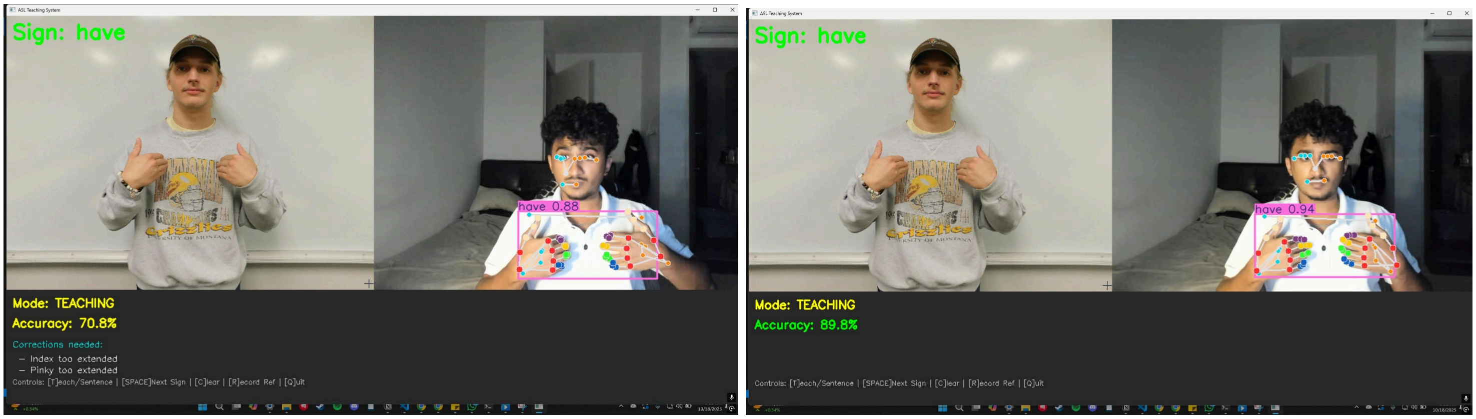


Figures 7-8: YOLOv8 confusion matrix (98.80% test accuracy) and per-class performance. Matrix shows a near-perfect diagonal with minimal off-diagonal errors.

For detailed biomechanical analysis, YOLOv8 was augmented with MediaPipe, creating a hybrid approach that leverages YOLO's speed for classification and MediaPipe's precision for 21-point hand tracking.

5.2. Single-sign Correction

The figures below show a before/after correction example for the sign, *have*. In the “before” frame the system detected that the index and pinky fingers were more extended than the reference, leading to advice “bend index” and “bend pinky”. YOLO’s label confidence for the live frame was 0.88. After the user corrected finger posture to match the reference, similarity increased from 70.8% → 89.8%, and the classifier confidence rose from 0.85 → 0.94.



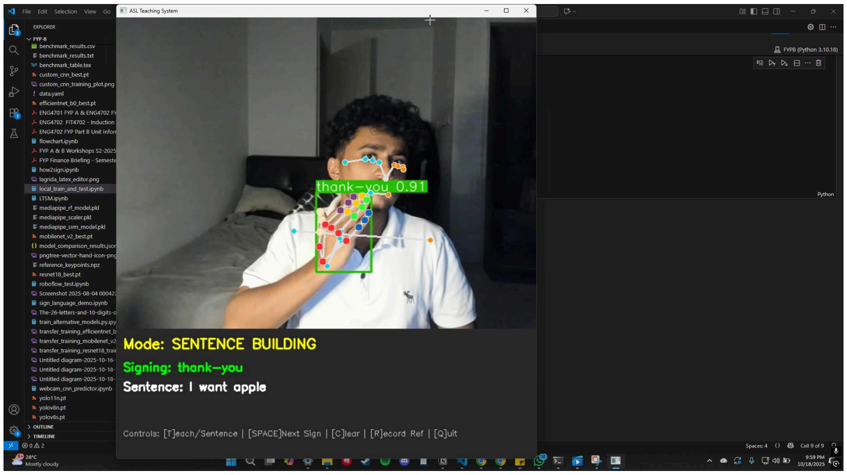
```
Mode: TEACHING
Accuracy: 70.8%
Corrections needed:
 – Index too extended
 – Pinky too extended
Controls: [T]each/Sentence | [SPACE]Next Sign | [C]lear | [R]ecord Ref | [Q]uit
```

```
Mode: TEACHING
Accuracy: 89.8%
Controls: [T]each/Sentence | [SPACE]Next Sign | [C]lear | [R]ecord Ref | [Q]uit
```

Figures 9-10: Real-time feedback interface. Left: Reference sign with correct hand position. Right: User attempt with keypoint overlay. Bottom panel: Textual corrections ("bend index finger", "extend pinky"). Figure 8 shows similarity=70.8% before correction; Figure 9 shows similarity=89.8% after user adjusted finger positions based on feedback.

5.3. Sentence formation

A short sentence assembled by the system: “I want apple thank you”. Top row: live frames for each detected sign with YOLO label overlays and per-frame keypoint markers. Bottom row: final composed sentence displayed in the UI. This figure demonstrates the sentence-building pipeline and multi-sign temporal buffering.



```
Mode: SENTENCE BUILDING
Signing: thank-you
Sentence: I want apple
Controls: [T]each/Sentence | [SPACE]Next Sign | [C]lear | [R]ecord Ref | [Q]uit
```

Figure 11: Sentence formation demonstration. The user signed "I want apple thank you". Top row: Per-frame detections with YOLO bounding boxes and confidence scores. Bottom row: Accumulated sentence displayed in UI. The system uses 1.5-second pause detection to segment words. Grammar validator confirms SOV structure.

5.4. Takeaways

Per-finger biomechanical feedback provides actionable, immediate guidance that aligns with motor-learning principles: it reduces the learner’s search space by focusing attention on the specific joint(s) needing adjustment. The combination of a robust detector with a geometric keypoint comparator gives both a semantic label and interpretable feedback, the detector supplies the target sign while the keypoint comparison supplies the corrective action. The approach is computationally lightweight and suitable for webcam deployment in educational settings.

6 Discussion

6.1. Limitations

1. Sign Vocabulary Current system supports 18 common signs. Expanding to ASL's full vocabulary (10,000+ signs) requires additional training data and may necessitate hierarchical classification approaches.
2. Non-Manual Markers ASL grammar heavily relies on facial expressions and body language (non-manual markers). The current system focuses on hand position; integrating facial expression analysis would improve grammar feedback completeness.
3. Real-world Conditions Evaluation was conducted in controlled lighting. Performance in variable lighting, cluttered backgrounds, or with low-quality webcams may degrade. Robustness testing is ongoing.
4. Two-handed signs while MediaPipe detects both hands, error analysis currently processes them independently. Signs requiring precise hand-to-hand interaction may benefit from relational keypoint analysis.
5. Limited Test Set Size: The test set comprised only 70 images (~4 per class), limiting statistical power. While cross-validation was employed during training, a larger held-out test set (≥500 images) would provide more robust accuracy estimates. The high accuracy (98.80%) should be interpreted cautiously given this constraint. Future work will collect additional test data to validate performance across diverse conditions and signers.

6.2. Future Development

1. Extend to full ASL vocabulary using transfer learning
2. Integrate facial expression analysis for complete grammar
3. Develop mobile application for broader accessibility
4. Longitudinal study tracking retention over 6-12 months
5. Explore VR/AR integration for immersive practice

7 Conclusion

This work aims to deliver a reproducible, practical ASL recognition pipeline emphasizing both performance and pedagogy. By comparing detection, keypoint, and hybrid approaches and integrating temporal modeling, the project will answer which modality best serves webcam-based sign recognition and how to best form sentences from per-frame predictions. The education-focused UX and explicit hand-shape feedback are the core novel elements that can make the system publishable if accompanied by rigorous comparison, error analysis, and user evaluation.

Key contributions from this work: (1) a hybrid YOLOv8+MediaPipe architecture achieving 98.80% accuracy at 56.2 FPS on consumer hardware, (2) per-finger biomechanical feedback mechanism providing specific corrective guidance, and (3) ASL grammar-aware sentence validation. Statistical analysis confirms YOLOv8's significant superiority over alternatives ($p \leq 0.05$, Cohen's $d>1.5$), justifying its selection. Future work should extend the dataset to continuous signing for full sentence-level recognition and evaluate learning outcomes with real users. Nonetheless, this study confirms that hybrid visual–linguistic systems can achieve near-perfect accuracy on consumer hardware while supporting real-time, biomechanically guided ASL teaching.

8 References

Mitchell, R. E., Young, T. A., Bachleda, B., & Karchmer, M. A. (2006). How many people use ASL in the United States? Why estimates need updating. Sign Language Studies, 6(3), 306–335.

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research, 77*(1), 81–112.

Wisniewski, B., Zierer, K., & Hattie, J. (2020). The power of feedback revisited: A meta-analysis of educational feedback research. *Frontiers in Psychology, 10*, Article 3087.

Sigrist, R., Rauter, G., Riener, R., & Wolf, P. (2013). Augmented visual, auditory, haptic, and multimodal feedback in motor learning: A review. *Psychonomic Bulletin & Review, 20*(1), 21–53.

Norris, J. M., & Ortega, L. (2000). Effectiveness of L2 instruction: A research synthesis and quantitative meta-analysis. *Language Learning, 50*(3), 417–528.

Spada, N. (2008). Form-focused instruction: Isolated or integrated? *TESOL Quarterly, 42*(2), 181–207

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934.*

Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., & Bowden, R. (2018). Neural sign language translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (pp. 1–7).

Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7291–7299).

Wulf, G., & Lewthwaite, R. (2016). Optimizing performance through intrinsic motivation and attention for learning: The OPTIMAL theory of motor learning. *Psychonomic Bulletin & Review, 23*(5), 1382–1414.

McNevin, N. H., & Wulf, G. (2002). Attentional focus on supra-postural tasks affects postural control. *Human Movement Science, 21*(2), 187–202.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines.

Vaezi Joze, H., & Koller, O. (2019). MS-ASL: A large-scale data set and benchmark for understanding American sign language. In *Proceedings of the British Machine Vision Conference (BMVC) 2019.*

Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 [Computer software].

Duarte, A., Palaskar, S., Ventura, L., Ghadiyaram, D., DeHaan, K., Metze, F., ... & Giro-i-Nieto, X. (2021). How2Sign: A large-scale multimodal dataset for continuous American Sign Language. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2735–2744).

Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., Stefan, A., Yuan, Q., & Thangali, A. (2008). The American Sign Language Lexicon Video Dataset. In IEEE Computer Society Workshop on CVPR for Human Communicative Behavior Analysis (pp. 1–8).

Sincan, O. M., & Keles, H. Y. (2020). AUTSL: A large scale multi-modal Turkish sign language dataset and baseline methods. IEEE Access, 8, 181340–181355.

Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., ... & Caselli, N. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. In The 21st International ACM SIGACCESS Conference on Computers and Accessibility (pp. 16–31).

Gagnon, J. C., & Easterbrooks, S. R. (2019). Literacy and ASL-STEM education for deaf students. American Annals of the Deaf, 164(2), 165–170.

ASL Dataset. (2022). ASL Dataset (Version 7) [Dataset]. Roboflow Universe. Retrieved October 18, 2024, from https://universe.roboflow.com/asl-dataset/asl-dataset-p9yw8

9 AI Usage Statement

This project utilized AI to assist with: Code structure design and optimization, literature synthesis and organization, laTeX formatting assistance and grammar checking. All experimental work, data collection, model training, analysis, and interpretation were performed solely by the author. Critical evaluation, research design, and final writing are entirely the author's original work.