Group Log: Working Notes, Sketches, and Research

Project: Sign Language to Text Conversion
Group Members (with Roll Nos.):
- Anukalp Rai — 220182
- Anwesha Dwivedi — 220190
- Norah Srivastava — 220736
- Bhavya Sharma — 220297
- Ishan Mukundhan — 220463
Date: November 9, 2025

This log captures the day-to-day progress, rough sketches, research trials, and design decisions. It supplements the main presentation/report by documenting how ideas evolved.

Timeline Snapshots

Date	What we did / learned
Oct 10	Agreed on problem framing (ASL letters A–Z + blank). Mapped success metrics (per-class accuracy, latency, robustness to lighting/background).
Oct 13	Surveyed public datasets; concluded gaps in raw ASL character images; decided to build a **custom dataset** aligned to our gestures and capture setup.
Oct 16	Set up image capture pipeline (webcam). Defined class list (27 symbols). Drafted data collection protocol (plain background, controlled lighting).
Oct 19	First data capture sprint; labeled samples; created train/val splits. Attempted basic thresholding; noise too high.
Oct 22	Implemented **grayscale → Gaussian blur → adaptive/dynamic threshold** preprocessing in OpenCV; feature clarity improved.
Oct 25	Baseline CNN trained on 27 classes; early overfit observed. Added augmentation; tuned LR scheduler; improved generalization.
Oct 28	Designed **two-layer classifier** plan: (1) general 27-class CNN, (2) specialized disambiguation for confusing sets (e.g., D/R/U; S/M/N; I/T/D/K).
Nov 01	Integrated **frame consensus rule** (≥50 consistent frames + dominance gap) to validate a letter; added “blank” to segment words.
Nov 03	Implemented **smart autocorrect** (Hunspell suggest) for word-level refinement; reduced perceived errors in real-time demos.
Nov 05	Measured accuracy on held-out custom set; peak accuracy ~**98%** (clean background, good lighting). Noted degradation in low-light/busy backgrounds.

Date	What we did / learned
Nov 07	Packaging and demo pipeline stabilized (real-time capture → preprocessing → CNN → consensus → word buffer → autocorrect → text stream).
Nov 09	Final presentation prep; compiled risks/limitations and next-steps (background subtraction, low-light robustness).

Working Notes & Research Iterations

Iteration 1 — Baseline (raw frames → single CNN)

Approach: Minimal preprocessing; direct CNN on raw frames.
Findings: Unstable predictions; sensitive to lighting/background; frequent confusions among visually similar letters.

Iteration 2 — Preprocessing stack

Approach: Grayscale + **Gaussian blur** + dynamic threshold; cleaned contours and edges.
Findings: Better separation of handshape; lower noise; improved training stability.

Iteration 3 — Consensus & Word Formation

Approach: Require ≥50 frame-consistent detections and a dominance gap to **validate** a letter; leverage a **blank** class to mark word boundaries.
Findings: Reduced jitter and spurious letters; more readable text output.

Iteration 4 — Two-Layer Disambiguation

Approach: Keep a primary 27-class CNN; route high-confusion sets to targeted classifiers (e.g., D↔R↔U, S↔M↔N, I↔T↔D↔K).
Findings: Noticeable drop in misclassifications within known confusion clusters.

Iteration 5 — Autocorrect Integration

Approach: Plug **Hunspell_suggest** to propose likely words; accept user-selected correction or auto-apply top suggestion.
Findings: Word-level accuracy perceived by users increased; helpful for subtle letter errors.

Design Decisions (Justifications)

- **Custom dataset** to match our camera, backgrounds, and gesture execution styles, ensuring high-fidelity labels and coverage.
- **Gaussian blur** in preprocessing to suppress high-frequency noise without erasing salient hand edges.
- **Consensus over frames** to convert fluctuating logits into stable letter commits.
- **Two-layer classification** to focus model capacity on confusing pairs/triads.
- **Blank token** for natural spacing and sentence construction.

- **Autocorrect** to smooth residual letter-level errors at the word layer.

Rough Sketches (Conceptual)

A) End-to-End Flow (High-Level)

```
Webcam → Preprocess (Gray → Blur → Threshold → ROI) → CNN(27)
     └─ if ambiguous set → Specialist Classifier(s) → Letter logits
        → Frame Buffer (N frames) → Consensus Check → Commit Letter
        → Word Builder (blank handling) → Autocorrect → Text Stream (UI)
```

B) Consensus Window (Timing Logic)

```
Frames:  [f1][f2][f3] ... [fN]
Letter:     A   A   A      A     (dominant)
Counts:     1   2   3 ...  ≥K    → Commit 'A' when count≥K & margin ≥ Δ
```

Parameters: K = 50 (tunable), Δ = dominance gap vs 2nd-best.

C) Two-Stage Disambiguation Router

```
        +-------------------+
Image → |  CNN (27 classes) | → confident → Commit
        +-------------------+
                 |
                 └─ low margin / known-confusion →
                      +------------------------------+
                      | Specialist heads per cluster |
                      |  (D/R/U), (S/M/N), (I/T/D/K) |
                      +------------------------------+
```

D) UI Wireframe (Real-time View)

```
+----------------------------------------------------+
|  Camera Feed (preprocessed preview, optional)      |
|                                                    |
|  [ Confidence bars per top-3 classes ]             |
+--------------------+-------------------------------+
|  Letter Buffer     |  Text Output                  |
|  A  N  U  K  A ... |  "ANUKALP RAI"                |
+--------------------+-------------------------------+
|  Actions: [Backspace] [Space/Blank] [Copy]         |
+----------------------------------------------------+
```

E) Data Pipeline (Capture → Split → Augment)

```
Capture → Label → Split(train/val) → Augment {flip, rotate,
illumination, scale} → Train → Eval → Export (TFLite/ONNX optional)
```

Sketches (Styled like sample Group Log)

1) System Architecture (Block Diagram)

```
+-------------+      +--------------------+      +----------------------
+      +------------------+      +------------------+
|  Camera /   | ---> |  Preprocessing     | ---> |  Base CNN (27 cls)   | ---
> |  Consensus Logic | ---> |  Word Builder    |
|  Webcam     |      |  (Gray, Blur, ROI) |      |  (A–Z + Blank)
|      |  (N frames, Δ)   |      |  + Autocorrect   |
+-------------+      +--------------------+      +----------------------
+      +------------------+      +------------------+
                                                                | (low
margin / confusions)
                                                                v

+----------------------------+
                                                     | Specialist
Classifiers     |
                                                     |  (D/R/U), (S/M/N), (I/
T/K) |

+----------------------------+
```

2) Data Flow (from Capture to Deployment)

```
[Capture] → [Annotate/Label] → [Train/Val Split] → [Augment]
   → [Train (LR sched, early stop)] → [Evaluate] → [Export Model]
   → [Runtime Inference Pipeline] → [Live Text Output]
```

3) UI State Machine (Letter→Word→Sentence)

```
  (Idle)
    │   new frame
    v
 [Predict top-k]
    │  if top1 margin < Δ and class∈confusion → route
    │  else keep
    v
[Frame Buffer Update] --(count≥K)--> [Commit Letter]
```

```
    │                                  │
    │ blank & word non-empty           │ non-blank
    │                                  v
    └----> [Insert Space] <----- [Append to Word]
                    │                      │
                    v                      v
             [Autocorrect]           [Live Text Box]
```

4) Error Handling & Feedback Loop

```
Misclassify → [Backspace] / [Undo last letter] → (optional) [Manual pick from top-3]
→ Add to Hard-Negative set → Next training cycle → Updated model
```

5) Metrics Panel (what we track)

```
- Per-class accuracy (A…Z, Blank)
- Latency (ms per frame) & throughput (FPS)
- Consensus commit rate vs jitter
- Confusion matrix (before/after specialist routing)
- Demo: WER (word error rate) with autocorrect on/off
```

----------------+ Image → | CNN (27 classes) | → confident → Commit +-------------------+ │ └── low margin / known-confusion → +------------------------------+ | Specialist heads per cluster | | (D/R/U), (S/M/N), (I/T/D/K) | +------------------------------+

```
### D) UI Wireframe (Real-time View)
```

+--+ | Camera Feed (preprocessed preview, optional) | | | | [Confidence bars per top-3 classes] | +--------------------+-------------------------------+ | Letter Buffer | Text Output | | A N U K A ... | "ANUKALP RAI" | +--------------------+-------------------------------+ | Actions: [Backspace] [Space/Blank] [Copy] | +--+

```
### E) Data Pipeline (Capture → Split → Augment)
```

Capture → Label → Split(train/val) → Augment {flip, rotate, illumination, scale} → Train → Eval → Export (TFLite/ONNX optional) ```

What Worked / What Didn't

Worked
- Preprocessing (grayscale + Gaussian blur) stabilized features.
- Frame consensus (≥50 + gap) reduced jitter.

- Two-layer specialization lowered confusion among look-alikes.
- Word-level autocorrect improved end-user readability.

Didn’t
- Low-light scenes: accuracy drops; edges merge with background.
- Busy backgrounds: false contours increase; more false positives.
- Over-aggressive thresholding: occasional loss of finger detail.

Decisions & Next Actions

1. Add **background subtraction** / segmentation to decouple hand from scene.
2. Explore **low-light enhancement** and auto-exposure control.
3. Collect **hard negatives** (cluttered scenes) and fine-tune with stronger augmentation.
4. Calibrate **per-class thresholds** and dynamic consensus length by confidence.
5. Evaluate lightweight backbones for on-device use (latency).
6. Add prediction **intervals/confidence bars** to UI.

References & Tools

- Internal presentation slides and notes on ASL alphabet (27 symbols incl. blank), preprocessing, two-layer design, and consensus logic.
- **Python, TensorFlow/Keras, OpenCV, NumPy, Matplotlib, Hunspell** (versions as configured during development).

Sign-offs

- Prepared by: Anukalp · Bhavya · Anwesha · Norah · Ishan
- Reviewed on: November 9, 2025