

**AR EDUCATIONAL SUITE FOR PRIMARY STUDENTS'
GEOMETRY AND LITERACY**

INDIVIDUAL THESIS

Abhilash K.A.T.W.

(IT22081698)

Individual component: AI Air-Writing Trajectory Tracking

Group ID: R26-IM-005

Report submitted in partial fulfilment of the requirements for the

B.Sc. (Hons) Degree in Information Technology

Specialisation in Interactive Media

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2026

DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. I declare that this thesis is my own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institution of higher learning. To the best of my knowledge, it does not contain any material previously published or written by another person except where appropriately acknowledged in the text. I grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this thesis while retaining the right to use its contents in future work.

Academic integrity and AI disclosure. All use of generative AI, automated writing tools, and AI-supported analysis will be disclosed in accordance with institutional policy. The candidate remains responsible for verifying all citations, code descriptions, datasets, quantitative results, and conclusions. This draft distinguishes group-paper findings from independently verified component results.

Student: Abhilash K.A.T.W. | Student ID: IT22081698

Signature: _____

Date: _____

Supervisor certification. The above candidate carried out research for the bachelor’s degree thesis under my supervision.

Role	Name	Signature / Date
Supervisor	Mr. Aruna Ishara Gamage	_____
Co-supervisor	Mr. Nushkan Nismi	_____

Certification details

ABSTRACT

This individual thesis describes the design, implementation, and preliminary evaluation of AI Air-Writing Trajectory Tracking, Component C3 of the R26-IM-005 Android educational suite for primary geometry and literacy. Conventional letter-tracing activities typically evaluate a visible final mark, whereas free-space air-writing can preserve the order and direction of fingertip movement. The developed prototype uses mobile-camera hand landmarks, a touchless START/FINISH interaction, ordered stroke capture, trajectory normalization, reference-based recognition, selected-letter practice, and child-oriented feedback. The original proposal specified a convolutional neural network; however, the implementation described in the available development record uses a lightweight normalized-trajectory template recognizer. This distinction is maintained to avoid conflating a proposed model with an independently verified deployed model. The integrated research paper reports 25 air-writing trials involving children aged 6–8: 19 correct predictions, three incorrect predictions, and three failed predictions, corresponding to 76% overall correct recognition. The same paper reports mean drawing time of 5.05 seconds and mean session time of 8.35 seconds; the corresponding literacy pre-test and post-test group averages increased from 5/10 to 9/10. These are descriptive findings based on the group-level evidence and do not establish independent model benchmark accuracy, statistically significant learning effects, or complete A–Z generalization. The thesis contributes a documented interaction architecture, implementation rationale, target-versus-free recognition workflow, and evaluation protocol for camera-based early literacy support. Future work should validate the full alphabet with a held-out dataset, obtain detailed paired learning measures, and benchmark recognition models under varied hand positions and lighting.

Keywords: Air-writing, Hand landmarks, Trajectory recognition, Early literacy, Unity, Augmented reality

ACKNOWLEDGEMENTS

I express my gratitude to Mr. Aruna Ishara Gamage, my supervisor, and Mr. Nushkan Nismi, my co-supervisor, for their academic guidance and critical feedback on the research direction. I acknowledge the Department of Information Technology and the Faculty of Computing of the Sri Lanka Institute of Information Technology for providing an environment for the development and investigation of interactive-media applications.

I also acknowledge the contributions of my R26-IM-005 project colleagues, whose complementary work on geometry validation, geometry correction and virtual slicing, and sketch-driven multisensory vocabulary learning established the context in which the Air-Writing component operates. I am grateful to learners, teachers, and guardians who supported the broader pilot study as described in the group research report. Responsibility for this individual component thesis, its interpretation, and its final verification remains with the author.

The acknowledgements should be reviewed against actual participation and assistance records before the final signed submission.

TABLE OF CONTENTS

1 INTRODUCTION	1
1.1 Overall project context	1
1.2 Individual component context	1
1.3 Problem statement	2
1.4 Research gap	3
1.5 Aim and objectives	3
1.6 Research questions or hypotheses	4
1.7 Scope, limitations and assumptions	4
1.8 Individual contribution	5
1.9 Thesis structure	5
2 LITERATURE REVIEW AND RELATED WORK	7
2.1 Review method and scope	7
2.2 Domain and theoretical foundations	7
2.3 Related research	8
2.4 Existing systems, products and techniques	8
2.5 Gap synthesis and conceptual framework	9
3 RESEARCH METHODOLOGY	11
3.1 Research design	11
3.2 Data, participants and study materials	11
3.3 Experimental or study procedure	12
3.4 Measures and evaluation metrics	13
3.5 Data analysis	14
3.6 Ethics, privacy, security and responsible AI	14
3.7 Reproducibility and research artefacts	15
4 PRODUCT AND SOLUTION DESIGN	16
4.1 Requirements	16
4.2 Component architecture within the complete solution	17
4.3 Design alternatives and decisions	17
4.4 Detailed design	18
4.5 Standards and professional practice	20
5 IMPLEMENTATION, INTEGRATION AND TESTING	21
5.1 Development environment and technology stack	21
5.2 Component implementation	21

5.3 Integration with the other components	23
5.4 Verification and software testing	24
5.5 Deployment and operation	25
6 RESULTS AND VALIDATION	26
6.1 Research results	26
6.2 Product and technical results	27
6.3 User or stakeholder validation	27
6.4 Integration results	28
6.5 Summary of findings	29
7 DISCUSSION	30
7.1 Interpretation of findings	30
7.2 Comparison with literature, baselines and existing solutions	30
7.3 Research contribution	31
7.4 Product contribution	32
7.5 Limitations and threats to validity	32
7.6 Implications	33
8 COMMERCIALISATION AND SUSTAINABILITY	34
8.1 Value proposition and target users	34
8.2 Market and competitive position	34
8.3 Business or sustainability model	35
8.4 Scalability, intellectual property and compliance	35
8.5 Contribution to the Sustainable Development Goals	36
9 CONCLUSION AND FUTURE DIRECTIONS	37
9.1 Conclusion	37
9.2 Achievement of objectives	37
9.3 Individual research and product contributions	38
9.4 Future directions	39
REFERENCES	40
GLOSSARY	42
APPENDICES	43

LIST OF FIGURES

Figure 1.1: Conceptual framework for learner-generated air-writing input, interpretation, feedback and reinforcement	6
Figure 3.1: Evaluation workflow from captured session to descriptive technical measures	13
Figure 4.1: Implemented Air-Writing Trajectory Tracking processing architecture.....	17
Figure 4.2: Air-writing touchless session state transitions	20
Figure 4.3: Separation of free letter recognition and selected-letter challenge workflows	20
Figure 5.1: Trajectory preprocessing and label interpretation stages	23
Figure 6.1: Reported recognition outcome distribution for C3 (n = 25).....	26
Figure 6.2: Group-reported C3 pre-test and post-test mean scores (out of 10)	28

LIST OF TABLES

Table 1.1: Thesis scope and ownership	6
Table 2.1: Comparison of input and recognition design alternatives	9
Table 3.1: Available evidence and permitted interpretation.....	12
Table 3.2: Component evaluation measures.....	14
Table 4.1: Key functional and non-functional requirements	17
Table 4.2: Architectural design decisions and reasons.....	18
Table 4.3: Recognition-to-feedback decision table	20
Table 5.1: Implementation stack and role	21
Table 5.2: Principal software classes and responsibilities.....	23
Table 5.3: Suggested regression test matrix	25
Table 6.1: Reported C3 pilot recognition results.....	27
Table 6.2: Available educational outcome and inference boundary	28
Table 9.1: Objective achievement assessment	38
Table A.1: Proposed participant-level record sheet	44
Table C.1: Data checks before quantitative analysis	48

LIST OF ABBREVIATIONS

Abbreviation	Expanded term
AI	Artificial intelligence
AR	Augmented reality
C3	Component 3 (Air-Writing)
CNN	Convolutional neural network
DSR	Design science research
FPS	Frames per second
HCI	Human-computer interaction
KNN	K-nearest neighbours
LTS	Long-term support
ML	Machine learning
RGB	Red-green-blue
SDK	Software development kit
SLIIT	Sri Lanka Institute of Information Technology
SUS	System Usability Scale
UI	User interface
UAT	User acceptance testing

Acronyms used in this thesis

1 INTRODUCTION

1.1 Overall project context

Foundational geometry and literacy require learners to do more than recognize a correct answer. A child must manipulate spatial ideas, produce symbols, and connect a performed action to its meaning. The R26-IM-005 research project, titled AR Educational Suite for Primary Students' Geometry and Literacy, addresses these complementary learning needs through a mobile learning environment built around learner-generated input. Its components interpret drawings, geometric constructions, and letter-writing movements before presenting feedback or augmented-reality reinforcement. The central educational premise is that digital experiences become more meaningful when the learner creates the input rather than merely observes a predefined object [1], [2].

The four research components operate as specialized modules: C1 validates 2D geometric nets and generates corresponding solids, C2 provides fault-oriented correction and virtual slicing, C3 performs camera-based air-writing trajectory tracking, and C4 connects sketch recognition with vocabulary and pronunciation-oriented tasks. Their shared contribution is architectural: capture input, analyze input, determine an instructional state, provide feedback, and enable an appropriate visual response. Component C3 is responsible for the free-space English letter interaction in this architecture. It serves the literacy route and does not claim ownership of the geometry model, net validator, or sketch-to-vocabulary classifier [21].

1.2 Individual component context

Air-writing allows a learner to produce a letter by moving an index finger in front of a camera without touching a screen or holding a pen. A vision system estimates hand landmarks frame by frame, selects the fingertip, and reconstructs an ordered path. Because the path retains temporal order, it includes information about movement and multi-stroke structure that a single static picture cannot provide. This characteristic motivates the individual research focus: the relationship between camera-tracked trajectory capture, letter recognition, and actionable educational feedback [11]–[15].

The implemented Air-Writing prototype was developed in Unity 2022.3 LTS with MediaPipe hand-landmark tracking, touchless interaction zones, trajectory recording, visual line previews, and a lightweight letter recognizer. The learner can enter a free-recognition flow to draw any supported uppercase English letter or choose a target letter through a letter-selection interface. In target-letter practice, the predicted output is checked against the selected character, creating meaningful correct, incorrect, and capture-failed states. The component also maintains a study logger containing predicted and intended labels, points, strokes, confidence-like scores, and timing fields. These implementation details are treated separately from the originally proposed CNN architecture.

1.3 Problem statement

In many conventional writing applications, correctness is assessed by the shape visible at the end of a tracing activity. A child may therefore reproduce a plausible symbol with inconsistent start position, unusual drawing order, or an interrupted stroke sequence without receiving any explanation about the movement. In classroom settings, a teacher cannot always observe each intermediate action of every learner. An accessible camera-based interaction could complement instruction by preserving movement evidence while allowing practice away from paper and touchscreen surfaces.

The individual research problem is formulated as follows: How can an ordinary Android camera support reliable and usable capture of fingertip trajectories for uppercase English air-writing, and how can the resulting ordered strokes be recognized and converted into child-oriented feedback? The question involves technical recognition and interaction design simultaneously. A classifier alone does not solve accidental triggering, fingertip instability, initiation/termination, inconsistent letter scale, or the distinction between unrestricted recognition and a selected-target exercise. These issues must be resolved in the end-to-end design rather than handled as unrelated user-interface details.

1.4 Research gap

Prior air-writing research demonstrates recognition with inertial sensors, fingertip videos, hand-pose estimation, and neural sequence models [11], [13], [14]. These systems establish feasibility, but laboratory recognition quality does not automatically translate to a child-oriented Android interface. A younger learner may move unpredictably, change distance from the camera, use multiple strokes, or pause while deciding where to draw next. Consequently, a direct research-to-classroom translation requires explicit capture boundaries and feedback states in addition to recognition.

The narrower gap addressed by this component is the design and evaluation of an accessible, markerless interaction pipeline that preserves ordered fingertip movement, suppresses unintended sessions, handles incomplete writing, and differentiates a free letter-identification task from a constrained practice task. The broader integrated paper situates C3 within a reusable learner-input-to-feedback contract, but also reports a limited pilot sample; it does not establish a full-alphabet benchmark or a controlled comparison against traditional handwriting instruction [21]. This thesis therefore documents an implementable design and interprets the pilot evidence within its actual limits.

1.5 Aim and objectives

The aim is to design, implement, and evaluate a mobile-camera-based Air-Writing Trajectory Tracking module that can capture English uppercase letter motion without a dedicated input device and return comprehensible recognition or retry feedback. The overall aim is operationalized through four objectives, each corresponding to observable technical evidence rather than an unsupported claim of pedagogical superiority.

- O1: Construct touchless fingertip acquisition and intentional START/FINISH capture control using mobile camera hand landmarks.
- O2: Represent, filter, normalize, and recognize multi-stroke uppercase letter trajectories across different writing locations and sizes.

- O3: Implement two distinct learner workflows—free recognition and target-letter practice—without applying target validation to the free mode.
- O4: Record technical session measurements and interpret pilot recognition and learning observations, acknowledging sample and measurement constraints.

1.6 Research questions or hypotheses

The investigation is organized around four component-level research questions. RQ1 asks how robustly the interface can distinguish intentional writing sessions from incidental fingertip movement. RQ2 asks whether an ordered, normalized trajectory representation enables identification of a supported uppercase character. RQ3 asks whether selected-target practice can issue clear correct/incorrect feedback while preserving the unrestricted behavior of free recognition. RQ4 asks what the available pilot results demonstrate about correct recognition, failure categories, task time, and short-term learning scores.

No causal hypothesis is imposed on the group-level pre/post means because the individual paired observations and a randomized comparison condition are not provided in the source material. Accordingly, the results are interpreted descriptively. This is a deliberate methodological safeguard: an improvement in aggregated mean scores after an activity is useful pilot evidence, but cannot by itself attribute the change uniquely to the proposed trajectory-tracking mechanism [21].

1.7 Scope, limitations and assumptions

The implemented scope described here is uppercase A–Z camera-based trajectory recognition and selected-letter validation in an Android-oriented Unity application. Hand landmark estimates from MediaPipe are used as the primary input; a single-camera system estimates normalized spatial landmarks rather than directly measuring physical three-dimensional coordinates with an external depth sensor. The system relies on adequate hand visibility, usable lighting, a compatible device camera, and relatively controlled interaction distance. These are operational assumptions requiring explicit testing under realistic school conditions.

The original proposal outlined CNN recognition, TensorFlow Lite export, Kalman filtering, and real-time canonical-path correction. Those design intentions are included in the evolution of the work, but they are not represented as independently verified production features where the supplied material does not demonstrate their deployment. Similarly, a fully evaluated lowercase alphabet and comprehensive directional-error accuracy metrics are outside the specific quantitative claims supported by the attached pilot paper. The outcome presented is a bounded, component-specific technical thesis rather than an unsupported statement that every proposed capability has been experimentally validated.

1.8 Individual contribution

The individual contribution is the Air-Writing subsystem and its integration contract. It includes defining a camera-to-fingertip coordinate pipeline, session state transitions, a multi-stroke representation, visual feedback, A–Z recognition modes, selected-target comparison, and developer-accessible study logging. The implementation also addresses practical UI mechanics: start screen actions, letter-selection cards, hold-start and hold-finish affordances, results, retries, and navigation. These are not merely decorative features; each is connected to capture or feedback semantics that affect whether a child's movement can be evaluated correctly.

Other project components are acknowledged only at their integration boundaries. Geometry validation and slicing are outside this individual's algorithms, and the sketch-driven vocabulary activity retains a separate classifier. The individual component supplies or could supply letter-related events to the integrated educational presentation layer, consistent with the architecture described in the group research paper [21].

1.9 Thesis structure

Chapter 2 reviews relevant education, computer-vision, and trajectory-recognition work. Chapter 3 defines the design-science method, data representation, measurement strategy, and research safeguards. Chapter 4 specifies the component

architecture, requirements, interaction states, and design decisions. Chapter 5 explains implementation, integration, verification, and Android operation. Chapter 6 reports the available pilot evidence and distinguishes it from unverified developer logs. Chapter 7 discusses contributions and threats to validity. Chapter 8 considers responsible commercialization and sustainability, and Chapter 9 concludes with a prioritized future-work program. References, glossary, technical test instruments, and academic-integrity disclosure follow in the final sections.



Figure 1.1: Conceptual framework for learner-generated air-writing input, interpretation, feedback and reinforcement

Within C3	Outside the direct C3 research scope
Hand landmark input, motion capture	Mathematical net validation (C1)
Uppercase trajectory recognition	Virtual slicing and net correction (C2)
Selected-target validation and feedback	Sketch-to-object vocabulary classifier (C4)
Session logging and pilot interpretation	Claiming whole-suite results as individual outcomes

Table 1.1: Thesis scope and ownership

2 LITERATURE REVIEW AND RELATED WORK

2.1 Review method and scope

The review draws on the proposal bibliography and the sources cited in the integrated research paper. Priority is given to camera-based fingertip recognition, landmark tracking, trajectory feature modeling, handwriting process information, early-literacy interaction, and design-science evaluation. Literature on AR geometry is used only to explain the shared suite and the educational rationale for interactive feedback. The literature is organized by technical question rather than by a chronological list of unrelated systems, because the central concern is the ability to produce a dependable learning interaction from noisy, user-generated motion.

The primary scholarly groups considered include: free-space handwriting with inertial or camera sensors; on-device hand pose estimation; neural and reference-based sequence recognition; child-facing corrective feedback; and learning-system evaluation. This review does not constitute a systematic meta-analysis. It identifies design affordances and reported limitations relevant to the implementation, and separates findings from another research context from claims that can be made about this specific prototype.

2.2 Domain and theoretical foundations

Handwriting is a sequential motor activity. Its meaning emerges from the relationship between a visual symbol, an ordered hand movement, and a learned language convention. Two trajectories can produce visually similar endpoints yet differ in direction, starting position, speed, and segmentation. Online handwriting recognition captures ordered points as the act unfolds, whereas offline image recognition relies chiefly on a final raster representation. Air-writing belongs to the online family, with the additional challenge that its input surface is unconstrained and must be reconstructed from the camera's observations [11], [13].

A digital learning tool also has a feedback-design problem. Children benefit when instructions relate to an action they can repeat, and when system failures are communicated as opportunities to retry rather than opaque technical exceptions.

Real-time hand tracking supports such interaction only if input states remain predictable: when the system starts listening, when a stroke is recorded, when recognition happens, and what a confidence score means. The implemented session model uses explicit hover zones and a writing board region to make these moments more visible and controllable. AR educational literature emphasizes interactive and age-appropriate experience design rather than visual novelty alone [1], [2], [16].

2.3 Related research

Mukherjee et al. investigated fingertip detection and tracking for air-writing recognition in video, demonstrating the importance of reliable segmentation of fingertip movement [11]. Zhang et al. described MediaPipe Hands as an on-device real-time hand-tracking approach, a practical basis for markerless smartphone interaction [12]. Alam et al. examined deep trajectory recognition using depth sensors, illustrating why temporal representations and spatial normalization are valuable, although their input instrumentation differs from an ordinary RGB-only phone [13]. Watanabe et al. studied 2D camera-based air-writing with pose estimation and a hybrid deep model, a particularly close technical comparison for a markerless implementation [14].

Educational applications add constraints beyond detection performance. Korosidou studied augmented reality for young learners' motivation and alphabet/vocabulary activity [16]. These findings support the relevance of interactive literacy tasks, but they do not prove that a particular air-writing classifier teaches correct handwriting. The distinction is important: recognition accuracy, task completion, engagement, and post-intervention learning scores are separate constructs. The current thesis therefore uses distinct technical and learner-outcome sections. The group's research paper adopted this separation when it reported C3 trial outcomes and aggregate pre/post means independently [21].

2.4 Existing systems, products and techniques

Approaches to air-writing can be classified by sensing method and recognition strategy. Inertial systems capture acceleration and motion from a dedicated sensor

worn or held by the user. Depth-based systems access spatial measurements unavailable to many low-cost devices. RGB camera systems use appearance or detected landmarks to infer a path. After capture, recognition may use image-based CNNs, sequence models, dynamic alignment, nearest-reference methods, or combinations of learned and rule-based features. Each approach makes a different trade-off between dataset needs, interpretability, computing cost, and robustness to unseen writing styles [11]–[15].

The initial project proposal selected a CNN trajectory model and planned TensorFlow Lite deployment. During practical implementation, a reference-template path-recognition method is visible in the development record, with normalized coordinate trajectories compared against stored uppercase examples and a configurable top-K score. A reference approach can be updated quickly through additional recorded examples and is easier to debug in a small prototype, but its reported confidence-like score must not be equated with a calibrated neural probability. Comparative work is needed to establish whether this choice performs better or worse on held-out child writing traces.

Approach	Strength	Constraint for young learners
Wearable inertial capture	Time-series motion data	Extra hardware and calibration
Depth-camera capture	Spatial range information	Cost and device dependence
RGB fingertip tracking	Widely available camera	Occlusion, lighting and jitter
CNN on trajectory features	Learned nonlinear patterns	Requires verified training, evaluation and deployment
Normalized template matching	Interpretable and simple to extend	Sensitivity to style variation and reference coverage

Table 2.1: Comparison of input and recognition design alternatives

2.5 Gap synthesis and conceptual framework

Three themes emerge from the review. First, landmark estimation does not by itself establish a writing interaction, because unintended finger motion may look like part of a character. Second, good classification on controlled data may not generalize to younger children who draw at different scales or interrupt the tracing sequence. Third, a learner's experience depends on the feedback policy as much as on the predicted label: correct recognition in a free task differs from correctness relative to a previously selected target.

The conceptual framework shown in Figure 1.1 links an intentionally captured motion trajectory to interpretation and feedback. The specific implementation closes an engineering gap by making capture and target validation explicit states rather than implicit assumptions. Its scientific contribution remains provisional until the full alphabet, unseen users, and directional validation are tested through an appropriately documented protocol. This caveat strengthens the interpretation of the limited pilot rather than diminishing the usefulness of an implementable system.

3 RESEARCH METHODOLOGY

3.1 Research design

The project follows a design science research orientation in which a technical artefact is iteratively designed, demonstrated, and evaluated against a practical learning problem [19], [20]. The artefact in this individual thesis is the Air-Writing Trajectory Tracking module, including its gesture capture, recognition, user feedback, and data-recording subsystems. The method starts from requirements, constructs an interaction pipeline, verifies component behaviours, and collects limited pilot evidence. Evaluation questions are defined at the level of the component; a correct letter prediction and a successful navigation action are not treated as the same outcome.

The development cycles comprise input acquisition; capture-state design; trajectory normalization; uppercase template collection and recognition; user-interface integration; selected-target validation; and feedback and logging. The available records show iterative Unity work on buttons, panels, recognition, and result handling. These development observations are useful for implementation traceability, but they do not replace a controlled trial. The group paper supplies a separate 25-child C3 pilot result that is reported without treating later developer-side session adjustments as authentic child-user measurements [21].

3.2 Data, participants and study materials

Two classes of data must be distinguished. Recognition reference data comprise labelled trajectories or templates used by the recognizer to make predictions. Evaluation data comprise separately observed sessions containing intended letter, predicted letter or missing output, strokes, points, and timing. A valid evaluation must avoid measuring prediction accuracy on the same templates used to configure the recognizer. Stored project templates support functionality, but their quantity alone is not evidence of generalization or learning gain.

The integrated group paper reports 25 evaluated air-writing trials among children aged 6–8, with the C3 trial sample covering 17 uppercase targets [21]. It does not

provide child-level participant identifiers, detailed sampling structure, held-out training membership, or full per-trial strokes in the uploaded version. The proposal earlier described collecting volunteer trajectories and a larger child dataset, but those were design intentions rather than independently supported completed counts. This thesis therefore interprets the 25-trial published group summary and specifies additional fields and protocols needed for reproducibility.

Evidence source	Content	Appropriate thesis use
C3 group-paper summary	25 trials: 19 correct, 3 incorrect, 3 failed	Pilot descriptive recognition results
C3 group-paper timing	Mean drawing and session durations	Descriptive interaction efficiency
C3 group-paper learning scores	Mean 5/10 before and 9/10 after	Descriptive short-term group difference
Unity development record	Uppercase workflow, state machine, logger	Implementation description
Proposal model design	CNN + canonical corrective overlay plan	Planned architecture and future comparison

Table 3.1: Available evidence and permitted interpretation

3.3 Experimental or study procedure

A repeatable C3 session begins with camera permission and an adequately visible hand. A participant is guided to the interaction screen and observes the START, writing, and FINISH affordances. In free recognition, any supported uppercase letter may be attempted, and the resulting prediction is displayed. In target-letter practice, the child first selects the intended uppercase character from the A–Z grid. The selected target is stored before recognition so that the system can determine correctness relative to the specific task.

To record a stroke, the index fingertip enters the START region long enough to indicate intention. After acceptance, movement inside the writing area is sampled as an ordered path. A multi-stroke symbol may contain pauses and separate segments; the system retains segmentation where available. Entering and holding the FINISH region ends capture and launches recognition. The output is classified as a correct target match, an incorrect target mismatch, or a failed attempt with inadequate or unavailable trajectory evidence. Completion, duration, and point counters are written to the session logger. This is the protocol implemented conceptually; the thesis does not assert that every possible failure state was experimentally measured in the pilot.



Figure 3.1: Evaluation workflow from captured session to descriptive technical measures

3.4 Measures and evaluation metrics

Correct recognition rate is defined as the number of evaluated trials with a correct predicted letter divided by the total number of evaluated trials, including failed-to-recognize trials in the denominator. For the reported C3 pilot this is $19/25 = 0.76$. Incorrect cases are attempts for which a letter was returned that did not match the intended target; failed cases are attempts without a usable letter prediction. The three-category report is more informative than a binary success rate because it separates a wrong prediction from a capture or recognition failure.

Drawing time measures the period attributed to the writing action; total session time additionally captures initiation, completion, and UI overhead according to the implementation's logger. Point count approximates the number of collected trajectory samples, and stroke count records the number of detected stroke segments. A confidence-like score is stored when recognition succeeds, but its relationship to probability calibration depends on the classifier and cannot be assumed. Usability and learning must be evaluated with separate instruments. A pre/post score on a ten-point task provides an educational indicator, while completion time and retry behaviour characterize interaction.

Measure	Definition or data field	Interpretation
Correct recognition	Count(correct) / all tested trials	Descriptive classification performance
Incorrect recognition	Prediction differs from target	Wrong character returned
Failed recognition	No usable returned character (?)	Capture / prediction failure
Mean drawing duration	DrawingTimeSeconds	Time spent producing letter
Mean total duration	TotalSessionTimeSeconds	End-to-end interaction duration
Points / strokes	TotalPoints / StrokeCount	Capture quantity / segmentation
Pre- and post-test	Average marks out of 10	Descriptive task-score change

Table 3.2: Component evaluation measures

3.5 Data analysis

The analysis prioritizes descriptive frequencies, percentages, means, and clearly bounded comparisons. The study outcome is tabulated as correct, incorrect, and failed, and a simple bar chart communicates their relative frequencies. Mean durations are cited directly from the group paper rather than regenerated from invented individual observations. The reported recognition confidence is a summary of recognizer outputs and should not be read as a validated probability. Without class-by-class trial labels, a complete 26-class confusion matrix is not possible from the attached group paper.

For learning scores, the only available values are component-level means, five before and nine after, based on ten-mark assessments. Because the uploaded source does not include participant-level paired test responses, the thesis does not calculate a paired t-test, p-value, standardized effect size, or confidence interval. Similarly, the document does not merge later synthetic or demonstration-oriented CSV records into this pilot sample. Future quantitative work should use a frozen evaluation split, a data dictionary, and an explicit record provenance flag before any inferential model comparison.

3.6 Ethics, privacy, security and responsible AI

Camera-based literacy support involves minors and therefore requires strong safeguards. The group research paper states that institutional ethics procedures were followed, parental or guardian consent and learner assent were obtained, and participant data were anonymized [21]. The project proposal also identifies on-device video processing and minimized retention of personal information as design principles. The individual thesis should be accompanied by the actual clearance document, consent form, and approved storage procedures before submission. The source text alone cannot substitute for signed ethics documentation.

The implementation should avoid retaining raw identifiable hand video when trajectory features suffice for debugging or research. Any exported session files

should be access controlled and use coded identifiers; locally saved debug logs must be treated as research records only when provenance and consent are documented. The classifier may misinterpret children with different motor skills, handedness, or movement styles. A wrong output should therefore be communicated as a retry cue rather than a judgment about the learner's ability. No stored session data should be represented as authentic field observations if they were created or adjusted for interface testing.

3.7 Reproducibility and research artefacts

A reproducible evaluation requires the same Unity scene, Android build, landmark model, capture thresholds, normalized trajectory processing, reference template corpus, and target labels. Version numbers and configuration values must be frozen at test time. The Unity project uses 2022.3 LTS in the documented development context; core components include the AirWritingController, hand-landmark provider, recognizer, canonical-stroke reference, visual line renderer, and UI navigation scripts. Each is a distinct artefact that contributes to end-to-end recognition.

The recommended evidence package consists of the selected code revision, a description of trajectory JSON structure, a snapshot of all recognition templates, app screenshots, test-case sheets, aggregate results, the original unmodified session CSV, and a data dictionary. Demonstration files and simulated rows must be tagged as such and stored separately. The thesis references the group research paper for its published summary and does not assert access to participant-level raw recordings that are not among the provided attachments.

4 PRODUCT AND SOLUTION DESIGN

4.1 Requirements

The component must convert a continuous, noisy hand-detection feed into discrete learning events. Requirements were grouped into input, recognition, validation, feedback, logging, and deployment categories. At the input boundary the system requires a camera feed and a stable index-fingertip estimate. At the session boundary it requires intentional START and FINISH interactions that prevent arbitrary hand movement from being saved as a letter. At the learning boundary it must provide recognizable positive, wrong, and retry feedback to children without disclosing developer-only terminology such as training-template management.

Two workflows are required. In free recognition, the learner starts a generic recognition experience and is told the letter detected; there is no fixed target and the system must not reject an otherwise supported uppercase letter merely because it differs from a previous selection. In the single-letter challenge, the target selected from the A–Z grid is stored and the predicted label is explicitly compared with that target. Incorrect or failed attempts must not silently become success states. Result panels, navigation buttons, and retries preserve these semantics while making state transitions obvious to a young learner.

ID	Requirement	Observable verification
FR-01	Estimate index fingertip from camera	Landmark appears and follows hand
FR-02	Support touchless START/FINISH	Hover begins and ends capture
FR-03	Retain ordered, multi-stroke points	Drawing preview follows strokes
FR-04	Predict uppercase A–Z labels	Free mode displays one predicted letter
FR-05	Check chosen letter in challenge	Mismatched prediction reports wrong
FR-06	Allow retry and panel navigation	Buttons reopen intended mode
FR-07	Record technical session fields	Logger creates structured row
NFR-01	Avoid external input hardware	Camera-only demonstration
NFR-02	Make child-facing controls legible	Visual UI usability review
NFR-03	Keep free and target modes isolated	Mode-specific test cases

Table 4.1: Key functional and non-functional requirements

4.2 Component architecture within the complete solution

The component is organized into an input layer, capture-and-state layer, recognition layer, and user-experience layer. An Android camera or development webcam supplies the frame input to MediaPipe. The hand-landmark runner estimates the hand; the provider translates this output into a fingertip position usable in Unity. A state-aware controller determines whether points belong to the current writing session, while a LineRenderer-based preview provides visible feedback about the path being collected.

The recognizer receives the resulting trajectory as a sequence rather than as an image. Normalization reduces sensitivity to starting location and scale; reference comparison yields a label and score. A validation boundary decides whether the label is merely displayed or checked against a chosen target. The UI layer activates the appropriate success, wrong, failed, or practice screen; the logger stores available technical measures. C3 can participate in the larger suite through shared navigation and literacy progression without introducing dependencies on the geometry components' rule engines. Figure 4.1 presents this modular arrangement.

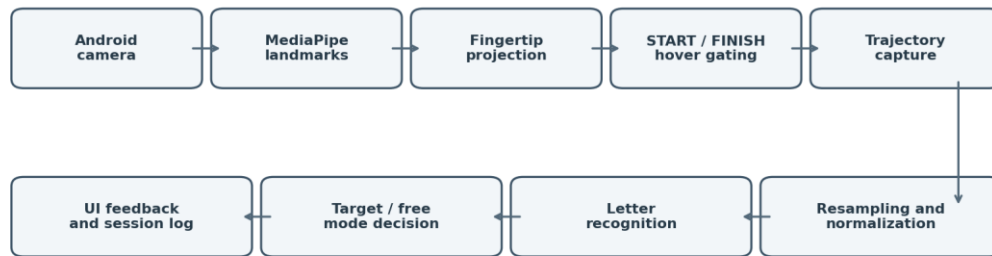


Figure 4.1: Implemented Air-Writing Trajectory Tracking processing architecture

4.3 Design alternatives and decisions

The original plan proposed a CNN model operating on trajectory features with on-device TensorFlow Lite inference. The implementation available in the development

record instead makes use of normalized example templates and a nearest-reference scoring process. For a constrained initial A–Z prototype this choice reduces deployment complexity: adding a new reference can be done through trajectory capture and storage, and the relationship between an input trace and its nearest examples can be inspected. The trade-off is that a limited set of references can overfit to particular styles and does not supply the same learned invariances as a large, well-trained model.

Another design choice is explicit hover gating rather than assuming that any hand movement in view means writing has begun. Without gating, repositioning the phone or the fingertip can produce accidental segments. A visible START state, a central writing board, and a FINISH state establish a user-controlled cycle. A further choice is to maintain recognition mode and target validation as separate rules. The former identifies a produced letter; the latter evaluates whether the produced letter matches the selected assignment. These decisions are reflected in the class responsibilities and test cases, and are central to avoiding regressions when new UI screens are introduced.

Decision	Reason	Remaining limitation
MediaPipe hand landmarks	Markerless camera interaction	Camera visibility and hand occlusion
START/FINISH hover states	Intentional session segmentation	Learning curve for hover timing
Ordered trajectory storage	Retains process information	Frame rate / tracking jitter
Template recognizer for prototype	Small-footprint debuggable recognition	Reference-style bias
Separate free vs target workflows	Prevents unintended validation	State synchronization across screens
Dedicated failed state	Separates insufficient capture from wrong label	Failure thresholds require calibration

Table 4.2: Architectural design decisions and reasons

4.4 Detailed design

The controller's conceptual state machine begins in an idle state, waits for a stable START hover, arms the capture, records movement into one or more strokes, waits for a stable FINISH interaction, then evaluates the assembled path. State transitions are not interchangeable with panel transitions. The system can display a practice panel while recording is not yet armed; conversely, a result panel must appear only after capture is closed. Explicit state tracking prevents issues in which a child sees a

success or wrong message but the underlying recorder still contains points from the last attempt.

A stroke is represented as an ordered sequence of coordinates together with start/end markers or segmented lists. During recognition, the data are translated into a normalized representation. A common procedure removes insignificant points, aligns to a local bounding box, scales to a stable extent, and resamples to a fixed number of positions. The normalized input can then be compared against reference templates using a distance measure that preserves ordering. Several near references may contribute to a score. A missing prediction or insufficient point count is distinguished from a confident but incorrect character.

A selected-letter challenge contains a target label and a predicted label. The comparison is case-normalized for uppercase input; correct means the two are equal, incorrect means a valid different letter was returned, and failed means no valid letter was recovered. In free recognition, the target check is skipped entirely. The distinction is especially important when preserving earlier Start/Uppercase menu behaviour as the child-facing interface evolves. The UI controllers should not independently overwrite the recognizer's semantics: they only display the result of the central session decision.



Figure 4.2: Air-writing touchless session state transitions

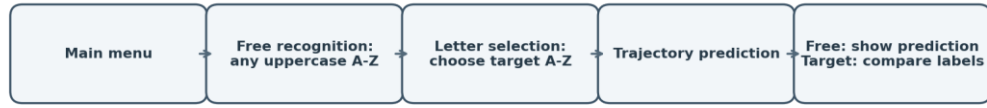


Figure 4.3: Separation of free letter recognition and selected-letter challenge workflows

Mode / condition	Result state	Expected screen
Free mode, valid A–Z prediction	Prediction available	Generic result with detected letter
Target mode, predicted = target	Correct	Congratulations / correct result
Target mode, predicted $\neq$ target	Incorrect	Custom WrongPanel
Either mode, unusable capture	Failed	Retry guidance, no invented letter
Challenge user presses Try Again	Retain target	Fresh capture for same letter
Challenge user presses Cancel	Exit challenge	LetterSelectionPanel
Any mode, presses Home	Reset navigation	StartPanel

Table 4.3: Recognition-to-feedback decision table

4.5 Standards and professional practice

The system design follows conventions relevant to educational computing and research reproducibility. It separates UI concerns from recognition logic, stores data in a structured format, and reserves developer-facing training controls from the child interface. Consistent target-letter naming and data schema definitions reduce errors in analysis. Configuration values such as hover duration, board bounds, and recognition point count should be recorded with a test build, because even a small tuning change can influence observed outcomes. This traceability is necessary for fair comparisons between recognizers and devices.

Accessible interaction requires readable controls and a clear retry pathway. Motion-driven tools should not stigmatize natural hand movement or sensor failures. Anonymized data, documented consent, and explicit AI-use disclosure support ethical practice, which must still be verified through approved field-study procedures.

5 IMPLEMENTATION, INTEGRATION AND TESTING

5.1 Development environment and technology stack

Development uses Unity 2022.3.62f3 LTS and C# in an Android-targeted project with MediaPipe Unity hand-landmark assets. During desktop development, the camera feed was tested through a webcam source; on the target mobile platform it is expected to use Android camera input with the same landmark abstraction. The central `AirWritingSystem` `GameObject` hosts the `AirWritingController` and the relevant provider, renderer, validation, and feedback behaviours. Child-facing panels reside under a Canvas hierarchy containing start, practice, result, letter selection, and custom wrong-result screens.

Unity permits scene-level configuration through Inspector references without hard-coding every UI object name. Serialized fields identify the hand provider, feedback UI, app UI, world camera, line renderer template, and recognition parameters. This supports an iterative workflow in which visual screens can be redesigned while core recognition remains stable. At the same time, the Inspector can introduce null references or stale assignments if scripts are replaced without care. For this reason the engineering practice used for the component is to preserve the working controller and modify one feature at a time, then test both unrestricted and selected-target behaviour before accepting the change.

Technology / module	Role
Unity 2022.3 LTS	Scene, UI, rendering and lifecycle
C#	Controller, recognizer, validation, logging
MediaPipe Hand Landmarker	Estimates hand landmarks from camera frames
Android / webcam camera	Captures learner-hand input
LineRenderer	Previews ordered trace in the practice view
JSON trajectory templates	Reference data for A–Z recognition
TextMeshPro	Dynamic recognition labels and guidance
Structured CSV logger	Captures testing/session measurements

Table 5.1: Implementation stack and role

5.2 Component implementation

The `AirWritingController` is responsible for the capture lifecycle, including waiting for START, arming the first stroke, accumulating coordinate samples, handling

separation between strokes, and accepting FINISH. The conceptual states visible in the implementation include idle waiting, session ready, stroke armed, active recording, and finish-hover completion. Detecting the fingertip in a central writing region does not itself mean that a new session should begin. The START affordance must first indicate deliberate participation, reducing spurious recognition caused by camera initialization and incidental motion.

Hand landmarks are obtained from the provider and transformed into the coordinate system used for writing interaction. The trajectory renderer draws the active path in one colour and completed strokes in another, giving the child immediate evidence that the system has captured their movement. The visual feedback is functional: if the line is not visible, the child has little way to know whether the camera is recording the intended trace. Layering of UI sprites and camera annotations therefore matters. A decorative blackboard rendered as a full-opacity screen overlay can obscure the tracked line; either transparency or an appropriate canvas/rendering order is needed.

The recognizer loads labelled trajectories from an on-disk dataset and normalizes candidate traces before similarity scoring. Stored labels correspond to uppercase A–Z folders. Processing aligns scale and location through path normalization, and a configurable resample count supports comparison between trajectories with different numbers of camera frames. The score is affected by reference coverage and the relative distances among examples. For the available implementation, the term template-based trajectory recognition is the supported technical description; the planned CNN pipeline remains a comparison or future extension until a trained model artefact and matching inference code are supplied.

The selected-letter pathway builds a dynamic A–Z button grid from sprite assets. Each button stores the associated letter and passes it to a LetterSelectionManager. The manager retains the target while the practice screen is opened. On result generation, the controller checks whether target mode is active. If so, it compares the target with the recognized label and routes mismatches to the custom WrongPanel. In free mode the prediction is shown without any fixed-letter constraint. This separation was important during integration because the earlier Start and Uppercase buttons

already supported drawing any recognized character, and the addition of a single-letter challenge was not intended to alter that behaviour.



Figure 5.1: Trajectory preprocessing and label interpretation stages

Class or scene object	Responsibility
AirWritingController	Manages capture states, strokes and recognition results
MediaPipeHandLandmarkProvider	Exposes index fingertip to Unity controller
AirInkTemplateRecognizer	Loads and scores uppercase trajectory examples
AirWritingAppUI	Shows generic start, practice and result screens
LetterSelectionGenerator	Instantiates the A–Z button grid
LetterSelectionManager	Stores selected target and opens challenge practice
WrongPanelController	Displays wrong-result panel and navigation actions
AirWritingStudyLogger	Writes session-level CSV rows

Table 5.2: Principal software classes and responsibilities

5.3 Integration with the other components

In the integrated suite, C3 is the literacy movement-recognition module rather than a dependency of every other component. The geometry pathway consumes sketches or mathematical net structures, not air-written English letters. The vocabulary pathway can optionally consume a verified letter event or share navigation, visual assets, and progression conventions. The group's research paper characterizes the shared architecture as an educational interaction contract: learner input is interpreted before targeted feedback or reinforcement. Therefore, the components need not share a classifier, training dataset, or success metric [21].

Integration testing must check more than compilation. Selecting a single-letter exercise should open the same stable camera capture surface without bringing over a stale free-recognition result. Returning Home should close the wrong and practice panels and restore the start menu. Try Again should retain the chosen target but reset

recording buffers; simply reactivating PracticePanel without resetting the controller can re-use old strokes. The proposed UI classes provide navigation entry points, while the central controller remains responsible for resetting capture state and preserving mode semantics. This separation helps maintain stability as the suite evolves.

5.4 Verification and software testing

Verification focuses on state transitions, recognized output, and non-regression. First, camera initialization must yield a usable fingertip estimate. Second, an unarmed hand moving through the writing board should not automatically generate a result. Third, START hover should arm a new capture, and moving the fingertip should update the stroke preview. Fourth, FINISH hover should close capture once and present a result. Fifth, insufficient data should lead to a failed state rather than a randomly assigned letter. Sixth, selected-target validation should distinguish a matching from a nonmatching prediction without changing generic recognition behaviour.

System verification also requires UI interaction tests. Letter cards must spawn correctly within the scrollable grid, route the selected label to practice, and return to the selection screen through Cancel. A wrong result must display the custom WrongPanel, not an obsolete generic ResultPanel. Home returns to the start screen; Try Again starts a new attempt for the same chosen letter. These tests reflect bugs encountered in practical UI wiring and are recorded as regression cases rather than treated as independent research results. Full evidence would include screenshots, console output, and pass/fail logs for each tested build.

Test ID	Input / action	Expected behavior
T01	Camera starts with hand visible	Landmark tracking active
T02	Hand moves without START	No letter completion
T03	Hold START then draw	New trajectory and visible line
T04	Hold FINISH after enough points	Recognition executes once
T05	Less than minimum points	Failed / retry response
T06	Free mode draws arbitrary letter	Predicted letter displayed
T07	Select B then predict B	Correct screen
T08	Select B then predict A	Custom WrongPanel
T09	WrongPanel Try Again	Same target, cleared capture
T10	WrongPanel Cancel	Letter-selection grid

Test ID	Input / action	Expected behavior
T11	WrongPanel Home	Start screen
T12	Return to free recognition	No target restriction persists

Table 5.3: Suggested regression test matrix

5.5 Deployment and operation

The Android deployment workflow begins with project compilation, device capability checks, and camera permission. The handset should be tested in portrait/landscape orientation consistent with the designed UI; in the development screenshots the interactive canvas was configured for a landscape reference resolution. A correct Unity Editor result is not sufficient evidence that Android camera permissions, image orientation, and tracking performance work correctly. Performance and UI scaling therefore require APK installation and practical testing on at least one supported Android device.

An operator should launch the app, choose a literacy pathway, ensure that the hand is visible, hold the START zone, draw within the indicated region, then hold FINISH. The result view should explain whether the attempt was identified, mismatched the chosen target, or could not be read. In school operation, adult guidance may be needed for first-time setup, camera framing, and interpreting errors. Diagnostic logs should be hidden from children, and downloaded study records should follow the approved research-data process. This operational description is a deployment guide, not a claim of proven performance across all Android devices.

6 RESULTS AND VALIDATION

6.1 Research results

The quantitative C3 pilot result published in the integrated research paper is based on 25 non-empty child-user air-writing trials. Nineteen trials returned the intended letter, three produced incorrect predictions, and three failed to produce a usable letter. Taking all 25 evaluated trials as the denominator yields $19/25 = 76.0\%$ correct recognition; 12.0% incorrect recognition and 12.0% failed recognition. A score computed only over trials that produced a prediction would be $19/22 = 86.4\%$, but that would exclude failed attempts and would not represent the child-user end-to-end recognition success rate. Accordingly, 76.0% is the primary reported value [21].

The paper reports a mean confidence-like value of 0.784 with a standard deviation of 0.314, mean drawing time of 5.05 seconds with a standard deviation of 2.07 seconds, and mean total session time of 8.35 seconds with a standard deviation of 2.65 seconds. These summary values characterize the observed pilot interactions but cannot reveal whether a particular target was more difficult, whether confidence was calibrated, or whether certain learners needed more attempts. The evaluation sampled 17 uppercase target letters; it is not a complete alphabet benchmark. Figure 6.1 visualizes the outcome categories, while Table 6.1 consolidates the statistics.

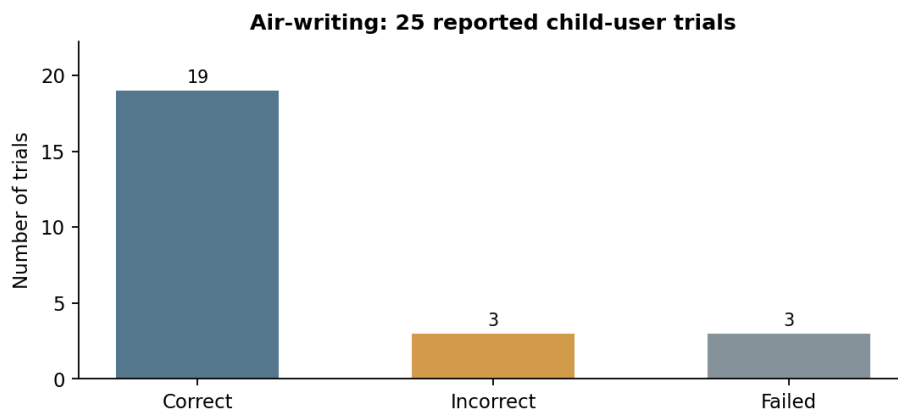


Figure 6.1: Reported recognition outcome distribution for C3 (n = 25)

Indicator	Reported result
Evaluated trials	25
Correct label	19 (76.0%)

Indicator	Reported result
Incorrect label	3 (12.0%)
Failed to produce letter	3 (12.0%)
Mean recognition confidence	0.784 (SD = 0.314)
Mean drawing time	5.05 s (SD = 2.07 s)
Mean total session time	8.35 s (SD = 2.65 s)
Sampled target alphabet	17 uppercase target letters

Table 6.1: Reported C3 pilot recognition results

6.2 Product and technical results

The implemented prototype supports a traceable pipeline from camera landmark input through a state-controlled capture and result display. The interface provides a generic recognition experience and a target-letter challenge, with dynamic A–Z letter selection, a visible air-writing surface, and differentiated result panels. These are product-level features evidenced in the development record. Their availability should be evaluated separately from the reported pilot recognition score: the presence of a button, trained templates, or a debug message does not imply that a complete feature has been validated with children.

The available recognizer configuration and dataset indicate a template-based uppercase recognition pipeline rather than a verified TensorFlow Lite CNN deployment. This distinction affects how performance should be interpreted. The group research paper describes the C3 implementation as CNN-based, reflecting the earlier design documentation, while subsequent development materials describe a normalized template recognizer. The final source-code revision, model inventory, and reproducible build should be inspected to resolve the architecture record before defense. This thesis intentionally documents that discrepancy instead of retroactively presenting a proposal-level neural model as a measured final artefact.

6.3 User or stakeholder validation

The group paper reports that the same 25 learners aged 6–8 took part in both literacy components and that the C3 mean pre-test score was 5 out of 10, increasing to 9 out of 10 in the reported post-test. This is a four-mark increase on a ten-mark scale, equivalent to 40 percentage points of the maximum score. The source does not include the individual paired score data required to establish a standard deviation of

change, a confidence interval, or a causal effect attributable uniquely to the Air-Writing recognition module. Figure 6.2 presents these aggregate means without implying individual-level significance [21].

The observed result is encouraging for a pilot, because children engaged with a camera-based letter task and a high proportion of the recorded trials completed successfully. Nevertheless, acceptance and learning must not be conflated with classification accuracy. Young learners may improve because of familiarity with the test, facilitator explanation, repeated exposure, or other suite elements. The final thesis should append the exact assessment sheets, scoring rubric, participant allocation, and anonymized paired results if available. Without them, the correct scientific description is an observed descriptive improvement rather than a controlled proof of learning effectiveness.

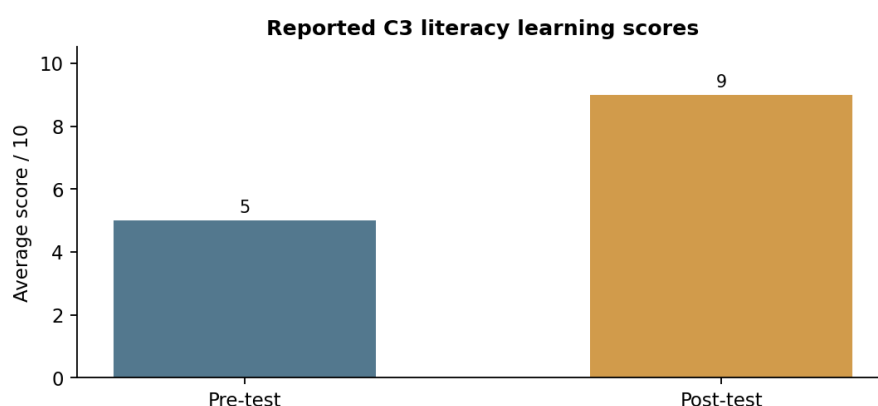


Figure 6.2: Group-reported C3 pre-test and post-test mean scores (out of 10)

Outcome	Supported conclusion	Not supported from available data
Mean pre-test: 5/10	Baseline group average	Individual score distribution
Mean post-test: 9/10	Reported follow-up group average	Paired change variability
Difference: +4/10	Descriptive average increase	Statistical significance or causality
Cohort ages 6–8	Age-specific pilot applicability	All-primary-student generalization

Table 6.2: Available educational outcome and inference boundary

6.4 Integration results

At architecture level, the literacy module's principal integration output is a letter-related instructional state, not an internal trajectory buffer exposed to unrelated

algorithms. A correct or incorrect state can be interpreted by the UI layer and, in an extended integrated flow, by another literacy activity. The group paper states that the suite supports shared navigation and presentation while retaining component-specific processing and evidence. This separation is appropriate because the geometry components use structured net or solid validation, whereas C3 uses ordered motion and character recognition [21].

The integration evidence available in the attachments is primarily design-level and group-level; no standalone C3-to-C4 event trace with timing or end-to-end test logs is supplied. The thesis therefore avoids claiming independently verified cross-component reliability. A recommended integration acceptance test would trace one character from fingertip capture to recognized letter, correct/incorrect feedback, optional vocabulary-event dispatch, and logged outcome, while verifying that geometry functionality remains unaffected. The lack of this test artefact is documented as an evidence gap rather than hidden within the results section.

6.5 Summary of findings

The reported pilot demonstrates that child users could interact with a camera-based free-space letter task and that 19 of 25 trials produced the intended letter under the observed conditions. The remaining six trials comprised equal numbers of wrong-letter predictions and failures to obtain a letter, giving clear priorities for improvement. Timing values suggest that single-letter interaction can be completed within seconds, although they do not establish performance under every device, lighting, or classroom condition.

The distinctive product contribution is a workflow that unifies capture intention, trajectory evidence, recognition, selected-target comparison, child-oriented feedback, and logging. The strongest available quantitative evidence is the 25-trial descriptive C3 summary from the group paper. Conclusions about direction-validation sensitivity, model superiority, complete A–Z performance, and learning causation remain unproven until additional independent data are supplied. These limits guide the discussion and future evaluation plan.

7 DISCUSSION

7.1 Interpretation of findings

The air-writing pilot should be interpreted as an evaluation of an end-to-end interaction rather than of a classifier in isolation. Recognition depends on successful camera acquisition, stable fingertip estimation, adequate motion capture, sensible trajectory normalization, and a decision about which label to display. Any of these stages can prevent a child from receiving a usable result. The distinction between incorrect and failed trials is therefore educationally and technically informative. An incorrect predicted letter suggests a recognition or writing-style mismatch; a failed trial may indicate incomplete capture, insufficient points, or a confidence threshold. The reported 3/25 cases in each category justify separate diagnostic analysis in subsequent iterations [21].

The reported 76% correct recognition is a descriptive pilot rate under specific child-user conditions. It should not be extrapolated to all 26 uppercase characters, both letter cases, or every device because the paper's 25 trials covered only 17 uppercase targets. Similarly, the four-mark average pre/post gain reflects a short sequence of instruction and practice, not proof that the air-writing algorithm alone caused learning improvement. The observed results nevertheless support feasibility: real children could use a camera-based interaction and produce machine-interpretable letter trajectories in a proportion of recorded sessions.

7.2 Comparison with literature, baselines and existing solutions

The closest technical literature shows two main routes to free-space writing recognition: tracking a hand or fingertip with vision [11], [14], and learning sequential movement structure from coordinates or image sequences [13], [15]. The component aligns strongly with the camera-based route because it does not require a glove, inertial marker, or depth sensor. Its START/FINISH gating and child-facing feedback address interaction concerns less visible in classifier benchmarks. Unlike studies that report accuracy on large preprocessed datasets, the present pilot score

includes failed recognition sessions and consequently reflects more of the practical user path [21].

No model-to-model baseline comparison is available in the supplied evidence. In particular, the uploaded materials do not contain a held-out comparison of template matching against a CNN, an LSTM, or a rules-only trajectory classifier under the same evaluation protocol. Therefore, any claim that the implemented method outperforms deep-learning methods would be unjustified. The proposal's CNN architecture is a design candidate, not an established baseline result. A scientifically stronger future comparison will need an identical training/test split, separately identified writers, matched evaluation hardware, and a common end-to-end denominator that retains failed captures.

7.3 Research contribution

The research contribution is a component-level articulation of camera-based air-writing as a full interaction cycle. The problem is not reduced to predicting a character from a stored signal; it includes a deliberate start event, sustained tracking, stroke segmentation, normalization, recognition, target-aware interpretation, and retry guidance. The design preserves the temporal path needed for future direction-sensitive validation. It also specifies the boundary between free recognition and targeted learning, which prevents a pedagogically meaningful selected-letter exercise from contaminating the ordinary identification task.

Methodologically, the thesis makes the outcome denominator explicit and records a three-category recognition result. This improves transparency compared with reporting success only among frames or strokes that produce a label. It distinguishes pilot learning scores from engineering debug traces and treats proposed technical capabilities as planned rather than automatically achieved. This transparency is an important research practice for educational AI, where impressive interface demonstrations can conceal limited experimental support. The design is transferable to other character sets, but transfer must be demonstrated with new reference trajectories and age-appropriate testing.

7.4 Product contribution

For product use, camera-based air-writing offers a low-equipment method of practicing uppercase letters. A learner does not need a stylus or a special wearable; the child raises a finger, uses visible start and finish cues, and receives a visually comprehensible result. The single-letter challenge turns unconstrained recognition into an intentional practice task, while free recognition supports exploration. Both modes reuse the same capture and recognition pipeline, reducing duplicated implementation. A scrollable grid of distinctive letter buttons simplifies navigation among the 26 targets.

The development experience also demonstrates a maintainability requirement: interface redesign should not alter the functioning controller unless a change in recognition semantics is intended. Panel classes activate and deactivate screens, while the central capture system determines whether a particular recognized letter is correct for the current task. This separation helps protect the earlier generic-recognition experience from later enhancements. The study logger extends the product into an evaluation instrument by retaining summary measures, though it cannot substitute for informed-consent and participant-recording procedures.

7.5 Limitations and threats to validity

Construct validity is limited by the difficulty of interpreting recognition confidence-like values as probabilities and by the lack of an independently observed correct-stroke rubric for each pilot trajectory. A correctly recognized letter need not imply that the child's starting point, stroke sequence, or directional motion met a pedagogical writing standard. Consequently, letter classification accuracy should not be labelled stroke-order-validation accuracy. The proposal describes canonical stroke comparison, but the reported child evaluation does not provide separate directional-error detection sensitivity or false-positive measurements.

Internal validity is limited by the short, non-randomized pilot and the availability of aggregate, not paired, learning outcomes. External validity is limited by 25 trials and 17 sampled uppercase targets in the uploaded group paper, and by uncertain representation of age, handedness, camera distance, and classroom lighting within

that small dataset. Engineering reproducibility is constrained because the specific trained model or final code revision used for the published pilot is not included among this turn's attached files. Later interface development may represent a different build from the one described in the article. The final defense should state the tested build and resolve such version differences.

7.6 Implications

For educational practice, the pilot supports investigating camera-mediated writing as a supplementary activity rather than replacing teacher observation of handwriting technique. The strongest practical value is the opportunity to practice independently with a visible motion trace and immediate retry pathway. To progress toward classroom deployment, feedback should explain why an attempt failed when possible, differentiate visibility problems from letter-form problems, and avoid presenting high score values as guarantees of correctness.

For future research, the evaluation design should include separate technical tasks for fingertip tracking, segmentation, letter identification, and pedagogical stroke-order correctness. More data are not automatically better evidence; sample provenance, independence between training and testing, and consistent outcome definitions matter. A properly designed held-out child-user benchmark could determine which characters are systematically confused, how recognition changes with lighting, and whether additional templates or a trained sequence model improve performance. This study offers an implementation framework and an initial descriptive evidence base for that more rigorous programme.

8 COMMERCIALISATION AND SUSTAINABILITY

8.1 Value proposition and target users

The primary beneficiaries are early-primary learners who are developing familiarity with uppercase English letters and teachers seeking additional independent practice tools. Parents and guardians may also use the app at home with an ordinary compatible Android device. The value proposition is an accessible, touchless learning activity that treats the child's own finger movement as the input. Rather than providing another static letter flashcard, it makes gesture, recognition, and corrective or celebratory feedback part of the learning task. This differentiator is credible as a product feature; broad claims of superior learning outcomes require additional comparative research.

For schools, the product would need rapid setup, low support requirements, readable controls, and consistent operation with intermittent connectivity. Use in shared-device environments creates additional privacy considerations because session logs may contain repeated observations associated with children. Teacher-facing analytics should be optional and understandable, and a school should retain control over whether individual data are stored. Any commercial use involving minors should obtain the appropriate parental and institutional consent independently of research consent.

8.2 Market and competitive position

The proposal positioned the project for Sri Lankan schools and parents and compared the concept with touchscreen tracing products. A practical competitive position would emphasize free-space gesture capture using common cameras, separate free and target exercises, and potential adaptation to different educational alphabets. However, the thesis avoids the categorical statement that no comparable app exists worldwide, because the attached materials do not constitute an up-to-date market survey. Existing air-writing research already demonstrates several forms of camera tracking and letter classification [11]–[15].

A careful pilot commercialization approach would therefore be to demonstrate usability and offline operation in a small number of school settings, compare device requirements, gather teacher feedback, and document total deployment cost. Such evidence would support a more defensible value proposition than a purely technical novelty claim. The project can build on the visual identity and navigation conventions of the larger educational suite without implying that all four components use the same recognition engine or are equally mature.

8.3 Business or sustainability model

Several sustainability models could be considered. A public-interest model would release a core literacy practice activity without fees and seek institutional support for maintenance, translations, and accessibility testing. A school-licence model could fund device compatibility updates and teacher dashboards. A freemium family app would provide an essential letter practice pathway at no cost and reserve optional content packs for paid use. None of these models is validated by pricing experiments in the attached evidence; they are possible routes to long-term support, not demonstrated revenue outcomes.

Responsible cost modelling should include user support, Android compatibility, consent and data governance, audio/visual asset licensing, localization, and continuing training-data quality checks. A camera-based product removes the need for a specialized wearable but still depends on smartphones and electricity. Sustainable access would therefore involve lightweight builds, offline support where technically verified, clear hand-position instructions, and teacher-assisted use for children who find motion interaction difficult. Any analytics service would need to justify why data collection is necessary and provide opt-out mechanisms.

8.4 Scalability, intellectual property and compliance

The software architecture can in principle be extended by adding letter reference sets or replacing the recognizer while retaining the camera-to-feedback interface. This modularity supports future lowercase letters and potentially additional scripts, but each new character set requires its own data, ground-truth letter definitions,

interaction tests, and educational validation. Deploying a classifier for Sinhala or Tamil cannot be achieved by changing UI labels alone. Similarly, a teacher dashboard would require new security, identity, and data-retention design rather than being implied by the existing session logger.

Project materials such as icon art, code, pretrained models, training traces, and institutional branding may have different ownership and license conditions. The proposal discusses patent opportunities, but the present thesis does not claim an existing granted patent or a filed provisional application without documentary evidence. Before commercialization, licensing and permissions for MediaPipe, third-party UI art, recorded datasets, and any research-derived content should be reviewed. Child privacy, informed consent, and accessibility requirements are ongoing responsibilities, not steps completed once at release.

8.5 Contribution to the Sustainable Development Goals

The component is most directly aligned with Sustainable Development Goal 4, which promotes inclusive and equitable quality education. A smartphone-mediated writing exercise could support access to supplemental literacy practice, especially when adult feedback time is limited. The contribution should be framed as an enabling technology and a preliminary educational intervention, because the pilot's group means and recognition outcomes do not establish large-scale reductions in learning inequality.

Sustainability also involves inclusion. Young children have diverse movement control, hand sizes, disabilities, and home-device access. A robust next iteration should permit slower capture, accommodate left- and right-handed writing, distinguish camera failure from learner error, and retain a conventional alternative input route when necessary. Classroom-scale evaluation across different communities and device types would provide stronger evidence of equitable benefit. Only after such testing could a wider SDG-related impact claim be supported.

9 CONCLUSION AND FUTURE DIRECTIONS

9.1 Conclusion

This thesis documents the individual AI Air-Writing Trajectory Tracking component of the R26-IM-005 educational suite. The proposed learning interaction uses a common device camera to observe hand landmarks, preserves an ordered fingertip movement trace, recognizes uppercase English letters, and returns result-dependent feedback. The component demonstrates why practical input boundaries, stable stroke capture, target-aware interpretation, and child-facing navigation are essential alongside classification. Its design fits the wider suite through an input–analysis–feedback–reinforcement interaction architecture while retaining an independent technical purpose.

The integrated group paper reports that 19 of 25 evaluated child-user air-writing trials returned the intended character, equivalent to 76% overall correct recognition, with three incorrect outputs and three failed trials. It also reports a descriptive increase in literacy assessment means from 5/10 to 9/10. These results support feasibility but are limited by the small sample, restricted uppercase coverage, and absence of participant-level paired scores in the provided source. The distinction between proposal-stage CNN plans and the current reference-based implementation is maintained as a central technical qualification.

9.2 Achievement of objectives

Objective O1 is addressed by the documented hand-landmark and touchless START/FINISH interaction design. Objective O2 is addressed at a prototype level by ordered trajectory capture and an A–Z normalized-template recognizer, while exhaustive out-of-sample validation remains incomplete. Objective O3 is reflected in the implemented separation of free recognition and selected-letter practice and the custom wrong-result feedback flow. Objective O4 is partially addressed by the logger specification and the descriptive C3 outcomes published in the group paper. Additional raw evidence is needed for independent reproduction and stronger inferential conclusions.

The original proposal envisioned a CNN-based recognizer and finer-grained canonical-stroke direction correction. These ambitions should not be presented as completed solely because they appear in design diagrams. Verification of a trained CNN file, matching inference path, and separate stroke-level evaluation is necessary before claiming those particular objectives as experimentally achieved. This objective-by-objective assessment helps protect the technical credibility of the individual contribution.

Objective	Current supported status	Further evidence required
O1 Landmark tracking and capture control	Implemented prototype workflow	Device-level tracking stability tests
O2 A–Z trajectory representation	Uppercase reference-based prototype	Held-out full-alphabet evaluation
O3 Selected-target feedback	Implemented UI/validation workflow	Logged child usability and regression tests
O4 Technical and learner evaluation	25-trial group-paper summary	Original trial logs and paired pre/post records

Table 9.1: Objective achievement assessment

9.3 Individual research and product contributions

The principal individual product contribution is a working system architecture for interpreting free-space letter movement as a discrete literacy interaction. It comprises touchless state transitions, stroke collection and preview, normalization, letter recognition, target comparison, results, retries, and structured session logging. The learner can explore arbitrary uppercase letters or select one character for a more directed exercise. Clear separation of these workflows is essential to maintaining the integrity of the earlier generic recognition experience while supporting a new challenge mode.

The principal research contribution is the explicit conceptual and methodological account of why the entire capture-to-feedback chain must be evaluated. The available pilot evidence is reported with the correct end-to-end denominator and transparent boundaries. Distinguishing proposed algorithms from implemented ones, genuine user outcomes from test records, and classification from pedagogical stroke accuracy provides a responsible basis for further development and publication.

9.4 Future directions

Future development should prioritize a clean benchmark of all 26 uppercase classes with disjoint participants between recognition-reference collection and evaluation. The benchmark should include multiple traces per character, confidence calibration, varying camera distances, diverse lighting, and balanced representation of writing style and handedness. Capture failures should remain in the overall outcome rate and should be categorized by cause. A character-level confusion matrix and error review would identify the most difficult symbols and guide new template collection or model development.

A second workstream should evaluate actual stroke-order and direction correctness against a documented canonical standard rather than inferring it from recognized labels. A third should implement and independently test the originally proposed CNN or another sequence model, including on-device latency, memory footprint, and power consumption. A fourth should extend the interaction to lowercase characters after verifying suitable reference data and child-friendly feedback. Finally, a larger controlled educational study should retain paired learner scores, assessor rubric, consent and assent documentation, a matched conventional-practice comparator, and delayed retention assessments. These changes would move the system from a feasible pilot toward defensible claims about technical generalization and sustained literacy benefits.

REFERENCES

- [1] H.-Y. Chang, T. Binali, J.-C. Liang, G.-L. Chiou, K.-H. Cheng, S. W.-Y. Lee, and C.-C. Tsai, “Ten years of augmented reality in education: A meta-analysis of (quasi-) experimental studies to investigate the impact,” *Computers & Education*, vol. 191, art. 104641, 2022. doi: 10.1016/j.compedu.2022.104641.
- [2] E. L.-C. Law and M. Heintz, “Augmented reality applications for K–12 education: A systematic review from the usability and user experience perspective,” *International Journal of Child-Computer Interaction*, vol. 30, art. 100321, 2021. doi: 10.1016/j.ijcci.2021.100321.
- [3] M. B. Nadzeri, M. Musa, C. M. Chew, and I. M. Ismail, “The effects of augmented reality geometry learning applications on spatial visualization ability for primary school pupils,” *International Journal of Interactive Mobile Technologies*, vol. 18, no. 16, pp. 104–118, 2024. doi: 10.3991/ijim.v18i16.47079.
- [4] W. Tarng, J.-K. Huang, and K.-L. Ou, “Improving elementary students’ geometric understanding through augmented reality and its performance evaluation,” *Systems*, vol. 12, no. 11, art. 493, 2024. doi: 10.3390/systems12110493.
- [5] J. Wu, H. Jiang, L. Long, and X. Zhang, “Effects of AR mathematical picture books on primary school students’ geometric thinking, cognitive load and flow experience,” *Education and Information Technologies*, vol. 29, pp. 24627–24652, 2024. doi: 10.1007/s10639-024-12768-y.
- [6] A. Uriarte-Portillo et al., “Intelligent augmented reality for learning geometry,” *Information*, vol. 14, no. 4, art. 245, 2023. doi: 10.3390/info14040245.
- [7] C. Walkington et al., “Comparing learning geometry using a tablet to head-mounted display augmented reality: How and when dimensionality matters,” *Education and Information Technologies*, vol. 30, pp. 5397–5426, 2025. doi: 10.1007/s10639-024-13008-z.
- [8] S.-H. Zhang, Y.-C. Guo, and Q.-W. Gu, “Sketch2Model: View-aware 3D modeling from single free-hand sketches,” in *Proc. IEEE/CVF CVPR*, 2021, pp. 6012–6021. doi: 10.1109/CVPR46437.2021.00595.
- [9] L. Zhang, “Hand-drawn sketch recognition with a double-channel convolutional neural network,” *EURASIP Journal on Advances in Signal Processing*, vol. 2021, art. 73, 2021. doi: 10.1186/s13634-021-00752-4.
- [10] A. K. Bhunia et al., “Vectorization and rasterization: Self-supervised learning for sketch and handwriting,” in *Proc. IEEE/CVF CVPR*, 2021, pp. 5672–5681.
- [11] S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, “Fingertip detection and tracking for recognition of air-writing in videos,” *Expert Systems with Applications*, vol. 136, pp. 217–229, 2019. doi: 10.1016/j.eswa.2019.06.034.

- [12] F. Zhang et al., “MediaPipe Hands: On-device real-time hand tracking,” in CV4ARVR, 2020.
- [13] M. S. Alam et al., “Trajectory-based air-writing recognition using deep neural network and depth sensor,” *Sensors*, vol. 20, no. 2, art. 376, 2020. doi: 10.3390/s20020376.
- [14] T. Watanabe et al., “2D camera-based air-writing recognition using hand pose estimation and hybrid deep learning model,” *Electronics*, vol. 12, no. 4, art. 995, 2023. doi: 10.3390/electronics12040995.
- [15] A. Qedear et al., “Real-time air-writing recognition for Arabic letters using deep learning,” *Sensors*, vol. 24, no. 18, art. 6098, 2024. doi: 10.3390/s24186098.
- [16] E. Korosidou, “The effects of augmented reality on very young learners’ motivation and learning of the alphabet and vocabulary,” *Digital*, vol. 4, no. 1, pp. 195–214, 2024. doi: 10.3390/digital4010010.
- [17] J. Belda-Medina and V. Marrahi-Gomez, “The impact of augmented reality (AR) on vocabulary acquisition and student motivation,” *Electronics*, vol. 12, no. 3, art. 749, 2023. doi: 10.3390/electronics12030749.
- [18] H. Lin and W. Wei, “A systematic review on vocabulary learning in AR and VR gamification context,” *Computers & Education: X Reality*, vol. 4, art. 100057, 2024. doi: 10.1016/j.cexr.2024.100057.
- [19] H. Weigand, P. Johannesson, and B. Andersson, “An artifact ontology for design science research,” *Data & Knowledge Engineering*, vol. 133, art. 101878, 2021. doi: 10.1016/j.datak.2021.101878.
- [20] S. G. Sutton, V. Arnold, P. Collier, and S. A. Leech, “Leveraging the synergies between design science and behavioral science research methods,” *International Journal of Accounting Information Systems*, vol. 43, art. 100536, 2021. doi: 10.1016/j.accinf.2021.100536.
- [21] R26-IM-005 research group, “A Learner-Generated Input Framework for AI and AR-Enhanced Primary Education: Design and Pilot Evaluation,” group research paper, SLIIT, 2026 (user-provided manuscript).
- [22] K. A. T. W. Abhilash, “AR Educational Suite for Primary Students’ Geometry and Literacy: AI Air-Writing Trajectory Tracking,” individual project proposal, R26-IM-005, Sri Lanka Institute of Information Technology, March 2026.
- [23] Sri Lanka Institute of Information Technology, “Guidelines on Documentation and Submission of Dissertations,” institutional formatting guidance (user-provided PDF).

GLOSSARY

Air-writing: Producing letters through hand or fingertip movement in free space, reconstructed by a sensing system.

Canonical stroke: An expected reference movement or ordered stroke pattern used to evaluate a written symbol.

Capture gating: Logic that decides when sensor movement belongs to an intended writing attempt.

Confidence-like score: A numerical recognizer output; not necessarily a calibrated probability of correctness.

Failed recognition: An attempt that does not yield a valid predicted label.

Free recognition: A mode that identifies any supported letter without comparing it to a fixed target.

Hand landmark: An estimated position of a defined point, such as the fingertip, on a detected hand.

Normalization: Transformation reducing translation, size and other incidental differences among trajectories.

Selected-target practice: A learning mode in which the recognized character is compared with a chosen letter.

Trajectory: An ordered series of points sampled as the fingertip moves during a writing attempt.

APPENDICES

Appendix A: Reproducible air-writing session protocol

The following protocol is suitable for a future independently documented evaluation. Before each test session, identify the test device, Android version, Unity build identifier, camera orientation, model and reference-data versions, and the environmental conditions. Confirm that a facilitator has the required consent and assent forms. Do not record the learner's identity in the recognition CSV. Use a coded participant ID and store the mapping, if one is necessary, separately under approved access controls.

For each trial, show the intended target or state that it is a free recognition exercise. Ask the child to bring the index fingertip into camera view, hold the START region until the interface indicates readiness, draw the letter using the central region, and hold FINISH. Record whether the system returns the intended label, a different label, or no prediction. Use the same retry protocol for all participants and log the number of attempts. Do not edit a raw record after collection; corrections should be maintained in a separate cleaned-analysis copy with documented reasons.

Field	Example / allowed value	Description
ParticipantCode	C3-XX	Pseudonymous child code
TrialID	Unique value	Distinct recorded attempt
CaptureDateTime	ISO datetime	Recorded time
SessionMode	Free / Target	Type of learning interaction
TargetLetter	A–Z / NA	Expected character if target mode
PredictedLetter	A–Z / ?	Returned prediction or failed state
Outcome	Correct / Incorrect / Failed	End-to-end result
Confidence	Numeric / not available	Recognizer score
TotalPoints	Nonnegative integer	Captured fingertip samples
StrokeCount	Nonnegative integer	Accepted stroke segments
DrawingSeconds	Nonnegative seconds	Drawing interval
SessionSeconds	Nonnegative seconds	Total interaction interval

Table A.1: Proposed participant-level record sheet

Appendix B: Detailed functional acceptance scenarios

B.1 Initialization

Precondition: Launch with camera permission. Action: Place one hand in view. Expected outcome: Hand landmarks are visible without false session completion. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.2 Intent gating

Precondition: Recognition mode with no current stroke. Action: Move fingertip across board without START. Expected outcome: No prediction is generated. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.3 START hold

Precondition: Idle state with visible START target. Action: Hold finger in START for configured interval. Expected outcome: Session becomes ready for recording. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.4 Stroke preview

Precondition: Capture is active. Action: Draw a continuous curve. Expected outcome: On-screen line follows new coordinate points. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.5 Multiple strokes

Precondition: One stroke has ended. Action: Reposition and draw another segment. Expected outcome: Both segments preserved in ordered session. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.6 FINISH hold

Precondition: Sufficient points captured. Action: Hold FINISH until completion. Expected outcome: Single recognition event and clean state closure. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.7 Insufficient capture

Precondition: No meaningful trajectory. Action: Attempt FINISH early. Expected outcome: Failed / retry response rather than invented class. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.8 Unrestricted mode

Precondition: Free recognition selected. Action: Draw any supported uppercase letter. Expected outcome: Predicted label displayed without fixed-target comparison. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.9 Target match

Precondition: B selected in A–Z grid. Action: Draw B and obtain prediction B. Expected outcome: Correct state displayed. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.10 Target mismatch

Precondition: B selected in A–Z grid. Action: Draw other letter predicted as A. Expected outcome: WrongPanel displayed. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.11 Retry semantics

Precondition: WrongPanel open after B mismatch. Action: Press Try Again. Expected outcome: B remains target; old strokes cleared. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A

reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.12 Cancel semantics

Precondition: WrongPanel open. Action: Press Cancel. Expected outcome: LetterSelectionPanel returns. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.13 Home semantics

Precondition: Any result visible. Action: Press Home. Expected outcome: StartPanel returns; other active panels hidden. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

B.14 Navigation regression

Precondition: Free mode after target exercise. Action: Draw arbitrary valid letter. Expected outcome: Former target does not constrain prediction. This case should be marked Pass, Fail, or Blocked by the tester, and the actual observed output should be recorded with the test build, date, and appropriate screenshot or log line. A reproducible failure report includes a clear reproduction sequence and a note on whether the problem occurs in the Unity Editor, Android deployment, or both.

Appendix C: Trajectory data integrity checks

The original raw trajectory and session records should remain immutable. Each cleaned analytical dataset should preserve a record identifier, provenance, source build, and version. For selected-letter practice, a correct result must have a target equal to the predicted label; an incorrect result must contain a valid nonmatching

label; and a failed result must be represented as missing output rather than silently changed to an arbitrary letter. These rules can detect internal inconsistencies, but they do not authorize the invention or alteration of real participant observations to achieve a preferred distribution.

Rule	Check	Action if violated
Unique IDs	No duplicated trial identifier	Resolve using original raw log
Valid target	Uppercase A–Z where target mode	Flag missing context
Correct outcome	Target = prediction	Investigate labeling mismatch
Incorrect outcome	Predicted letter differs from target	Investigate labeling mismatch
Failed outcome	No valid predicted letter	Classify failed; do not impute success
Timestamp order	Session times plausible and traceable	Audit device clock or source
Points/strokes	Nonnegative, within accepted bounds	Compare with raw capture
Dataset provenance	Collected vs synthetic marked clearly	Exclude synthetic from user trial results

Table C.1: Data checks before quantitative analysis

Appendix D: Academic integrity and AI-use disclosure

This thesis draft was prepared with AI-assisted writing support. The assistance covered document structuring, explanation of technical concepts, preparation of schematic figures, and wording aligned with the uploaded proposal, group paper, and institutional template. The author must review and verify every technical description, citation, numerical statement, test result, and software-version claim. AI assistance must not be presented as independent experimental validation. The source documents and the final build, not generated text, remain the evidence of what was designed and actually implemented.

Before submission, the student should complete the institutionally required disclosure with the actual software tools used, the tasks performed, representative prompts where required, personal verification steps, and any sections affected. The researcher should also confirm the final names and signatures, insert ethics authorization and original participant-level records where available, and remove draft-only qualification notes after completing those verification steps. No simulated or revised demonstration logs should be described as collected child-user outcomes.

Appendix E: Source verification items for final submission

- Confirm the exact degree title, full expanded candidate name, co-supervisor spelling, and submission month.
- Supply and inspect the final AirWritingController, recognizer, landmark provider, and model or template artefacts.
- Identify which recognizer build produced the reported 25-trial group-paper result and reconcile CNN/template terminology.
- Provide the original air-writing pilot CSV, protocol, participant split, source timestamps, and signed ethics documentation.
- Provide the individual pre-test/post-test instruments and paired scores if any formal learning-effect calculation is required.
- Provide final Android screenshots and device-test logs for each implementation section.
- Check every reference entry against its original publication before binding or formal submission.
- Update the Table of Contents, List of Figures and List of Tables page numbers after final editing.