International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Study of Vision-Based Air-Writing for Word Recognition and Contactless Application Interaction

Tanvi Gawas[1], Srawani Beldar[2], Suman Panigrahy[3], Dr. Saloni Bhushan[4]

[123]MCA Department, D. Y. Patil Deemed to be University, School of Humanities & Sciences, Navi Mumbai, India
[4]Associate Professor, School of Humanities and Sciences, D. Y. Patil Deemed to be University, School of Humanities & Sciences, Navi Mumbai, India

---

Abstract - *Air-writing lets a user enter text through hand or fingertip movement in free space, without touching a device. Published systems recognise isolated characters and digits with high accuracy, but the recognised text is usually the end of the pipeline rather than the start of an interaction. This paper presents a webcam-based prototype that tracks the index fingertip, normalises the trajectory, recognises a constrained vocabulary, maps the recognised word to an application-level intent, and exposes a confidence-gated execution control. The prototype was evaluated in-set with a stored-template recogniser on 130 uppercase-letter samples (26 letters × 5 variations) and a 13-word vocabulary. It recognised 129 of 130 letters (99.2%; 95% Wilson interval 95.8–99.9%) and 13 of 13 words (100%; 77.2–100%). Mean recognition-stage latency was 0.91, 3.99, and 6.24 ms for one-, four-, and six-letter input. Weaknesses were measured as well: only 4 of 7 invalid strokes were rejected, and accuracy degraded above about 6% trajectory noise. These are preliminary in-set results. They do not establish user-independent accuracy, intent-classification quality, or camera-to-action latency, and a participant-level evaluation protocol is specified for the next stage.*

Key Words: air-writing, fingertip trajectory, MediaPipe, word recognition, intent recognition, human–computer interaction, contactless interaction.

1. INTRODUCTION

Touchless input is useful when keyboards, touchscreens, or wearable devices are inconvenient, for example when the hands are occupied or contact with shared surfaces is undesirable. Air-writing addresses this by letting a user write a character or word with free-space hand motion. The recognition problem is hard because there is no writing surface and no pen-up/pen-down signal, and because the trajectory varies with writing speed, scale, camera position, lighting, hand pose, and individual style.

Early statistical work modelled air-written characters, words, and the motions connecting letters using six-degree-of-freedom motion signals, and later studied how to detect writing activity in a continuous stream of motion data [1], [2]. Vision-based approaches followed, starting with fingertip detection and tracking in video [3] and continuing with deep networks on depth-sensor trajectories [4], webcam trajectories [5], and interpolated fixed-length trajectories [6]. Other sensing modalities such as ultrasonic [7] and radar [8] signals have also been explored, and CNN–LSTM models have been designed specifically for trajectory data [9]. Watanabe et al. combined MediaPipe hand-pose estimation with a CNN–BiLSTM model and reported accuracy above 99% for alphabets and digits [10]. Tan et al. addressed continuous air-writing with an end-to-end Transformer and reported a character error rate of 29.86%, which shows that continuous recognition remains difficult and is already an active research direction [11]. Temporal convolutional networks have likewise reached accuracy above 99% on public trajectory datasets [12].

The present work therefore does not claim novelty in character recognition, in continuous recognition, or in the use of MediaPipe, CNN, or LSTM components. It concentrates on the stage that follows recognition: recognised word → intent → confidence decision → application action. The guiding question is whether an ordinary webcam can support a controlled and measurable path from free-space writing to application-level interaction. This paper reports what the current prototype measurably achieves, states what it does not yet show, and specifies the experiments needed to close the gap.

2. RELATED WORK

The reviewed literature (17 studies, Table 1) falls into four groups.

Motion and non-camera sensing. Chen et al. modelled letters and connecting motions with hidden Markov models on motion-sensor data, extending air-writing from characters to words [1], and separately addressed segmentation of writing activity in continuous streams [2]. Later systems used ultrasonic transceivers with CNN and ConvLSTM models [7], FMCW radar with a multistream CNN [8], millimetre-wave radar that extends to words [13], and wearable inertial sensors for letter and person recognition [16]. These approaches can be accurate but depend on dedicated hardware.

Vision-based tracking and recognition. Mukherjee et al. studied fingertip detection and tracking in video [3]. Alam et al. used a depth sensor with CNN/LSTM models [4], and Hsieh et al. used a webcam trajectory with a CNN [5]. Abir et al. showed that the interpolation method used

to fix trajectory length affects accuracy, and reported a drop from 91.34% (user-dependent) to 85.59% (user-independent) on one dataset [6]. Watanabe et al. [10] and Li and Hsieh [17] use MediaPipe hand keypoints, for air-written characters and dynamic gestures respectively.

Sequence modelling. TARNet combines a CNN and LSTM for lightweight trajectory recognition [9]; a temporal convolutional network models raw 3D sequences [12]; and a Transformer performs end-to-end continuous recognition [11].

Surveys. Jabde et al. review air-written online handwriting and highlight dataset and protocol differences [14]; Arora et al. review handwritten Devanagari recognition and its dataset diversity and variability issues [15].

Table -1: Summary of reviewed studies (values as stated by each source)

Ref.	Year	Method / sensing	Target	Reported result	Limitation relative to this work
[1]	2016	6-DoF motion sensing + HMM	Characters, words	Word-level modelling from letter and connecting-motion models	Specialised motion sensor
[2]	2016	Motion-stream segmentation	Writing activity	Detection and recognition in continuous motion data	Motion-sensor input
[3]	2019	Fingertip detection and tracking in video	Air-writing in video	Trajectory tracking for recognition	Tracking-centred; no word-to-intent stage
[4]	2020	Depth sensor + CNN/LSTM	Digits, characters	99.32% (6DMG)	Depth hardware
[5]	2021	Webcam trajectory + CNN	Air-written characters	96.11% mean accuracy	Isolated characters; segmentation and generalisation issues
[6]	2021	Interpolation + 2D CNN	Digits, characters	Up to 99.76% (RTD); 85.59% user-independent (Smart-band)	Character/digit focus
[7]	2021	Ultrasonic transceivers + CNN/ConvLSTM	Characters	Several models above 97%	Dedicated ultrasonic hardware
[8]	2022	FMCW radar + multistream CNN	Digits / gestures	F1 = 0.9425 (about 94.2% accuracy)	Radar hardware
[9]	2022	CNN + LSTM (TARNet)	Digits, characters	High benchmark accuracy reported	Recognition only
[10]	2023	Webcam + MediaPipe + CNN–BiLSTM	Alphabet, digits	Above 99% for both	No word-to-intent execution
[11]	2023	End-to-end Transformer	Continuous air-writing	CER 29.86%	Continuous recognition remains difficult
[12]	2024	Temporal convolutional network	Digits, characters	99.50% (RTD), 99.56% (RTC); 10-fold CV	User-independent test left to future work
[13]	2024	mmWave radar (AirWrite)	Letters, digits, words	Trajectory tracking and recognition system	Radar hardware
[14]	2024	Literature review	Air-written online handwriting	Review (no single result)	Dataset and protocol differences
[15]	2024	Literature review	Handwritten Devanagari	Review (no single result)	Offline handwriting, not air-writing
[16]	2025	Wearable IMU + ML	Letters; person identity	About 93% person recognition	Wearable hardware
[17]	2025	MediaPipe keypoints + Transformer	Dynamic hand gestures	Up to 100% on evaluated gesture sets	Gestures, not written words

Results come from different datasets, sensors, and protocols and are not directly comparable with each other or with the results in Section 6.

Three observations follow from Table 1. First, high character-level accuracy is common on constrained datasets. Second, reported accuracy depends strongly on sensor and protocol; the user-dependent to user-independent drop in [6] and the open user-independent question in [12] show that in-set accuracy should not be read as generalisation. Third, none of the 17 reviewed studies evaluates a webcam pipeline from written word to intent to confidence-gated action under a single protocol. This conclusion is limited to the studies reviewed here: the survey contains no dedicated work on command or intent classification or on CTC decoding, and extending it to those areas is part of the planned work.

3. RESEARCH GAP AND CONTRIBUTIONS

Character and digit recognition are well represented, continuous air-writing has been demonstrated [11], and several sensing modalities are available. The less-developed layer is the explicit evaluation of recognised air-written words as application commands, including confidence handling and rejection of invalid input, using only a low-cost webcam. Latency is also reported inconsistently across studies, so the stage-wise and camera-to-action definitions used here are stated explicitly. The contributions of this paper are deliberately limited to what the evidence supports:

(1) a webcam-only prototype pipeline from fingertip trajectory to word, intent, and a confidence-gated execution control;

(2) a measured in-set baseline for constrained letter and word recognition, reported with uncertainty intervals together with invalid-input rejection, mild-noise behaviour, and recognition-stage latency; and

(3) an evaluation protocol that keeps recognition accuracy, intent quality, robustness, rejection behaviour, and end-to-end latency separate, to be applied with participant-level splits.

3.1 Hypothesis Questions

Three research questions guide the evaluation of the proposed pipeline.

RQ1: How well does the recogniser generalise to participants who are absent from training?

RQ2: How accurately are recognised words mapped to intents, and how reliably are invalid or low-confidence inputs rejected before execution?

RQ3: What is the camera-to-action latency of the full pipeline, and how is it distributed across stages?

4. SYSTEM DESIGN

4.1 Pipeline Overview

The system is a sequential pipeline (Fig. 1). A webcam supplies the video stream; hand tracking locates the index fingertip; successive fingertip coordinates form a trajectory; the trajectory is normalised and resampled; a recogniser produces a word; the word is mapped to an intent; a confidence check decides whether the action may be executed; and the application action is triggered. Only the recognition stage (shaded in Fig. 1) is evaluated quantitatively in this paper.

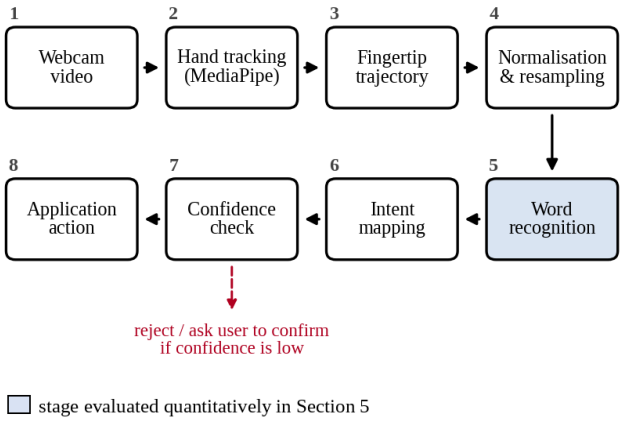


Fig -1: Architecture of the proposed word-to-intent interaction pipeline. The dashed branch shows the intended handling of low-confidence input.

4.2 Hand Tracking and Trajectory Extraction

MediaPipe Hands [18] estimates hand landmarks in each video frame. The index-fingertip landmark is taken as the writing point, and its two-dimensional image coordinates are accumulated into a trajectory $T = \{(x_t, y_t)\}$, $t = 1, \ldots, M$. The prototype interface also provides a canvas mode, which is the mode shown in Fig. 2; the example in that figure contains 268 raw points recorded over 8.5 s.

4.3 Trajectory Normalisation

To reduce dependence on the writing position and the size of the written word, the trajectory is translated to its starting point and divided by its largest extent s, defined as the larger of its horizontal and vertical ranges:

$$x'_t = (x_t - x_1) / s, \; y'_t = (y_t - y_1) / s \qquad (1)$$

$$s = \max\{ x_{max} - x_{min},\ y_{max} - y_{min} \} \quad (2)$$

The normalised trajectory is then resampled to a fixed length of 128 points, giving the 128 × 2 tensor reported by the interface (Fig. 2).

4.4 Template-Based Recognition and Word Formation

The recogniser evaluated in this paper is a baseline that matches the normalised trajectory against one stored template per uppercase letter; letters are concatenated to form words. A learned temporal model (CNN-LSTM or CNN-BiLSTM) is a planned extension and is not part of any reported result, which is consistent with the "model not yet trained" badge visible in Fig. 2. In the noise test, the prototype's alternative-letter substitution option was enabled; Section 7 discusses how this can affect the interpretation of word-level results.

4.5 Intent Mapping and Confidence-Gated Execution

The current vocabulary includes CHROME, EDGE, EXCEL, GOOGLE, and EMAIL, with shortcuts C, E, X, G, and M. Commands open the selected browser or Excel, google.com, or the default mail client. The interface shows the recognised word, intent, and target. Execution requires user confirmation and a one-time 30 s token; a fixed allowlist and local host/origin checks add safeguards. These controls have not been independently tested.

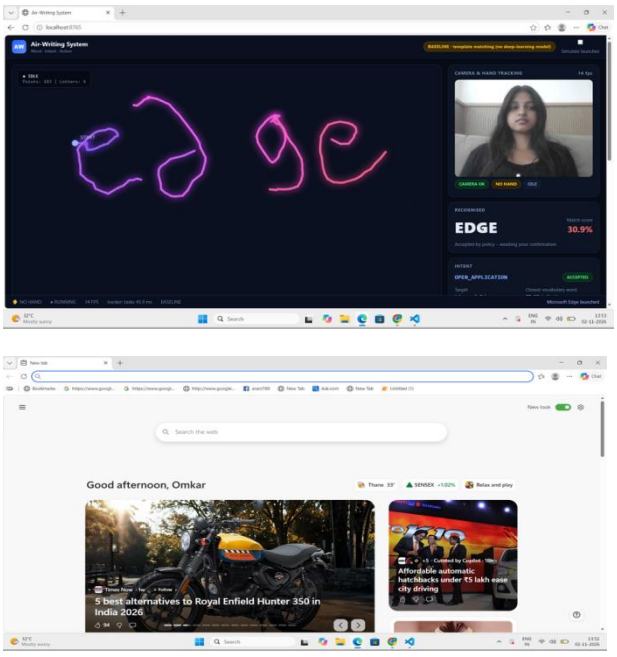


Fig -2: Updated author-supplied screenshots: (a) Air-Writing System showing EDGE, a 30.9% match score, OPEN_APPLICATION intent, and an awaiting-confirmation message; the dashboard footer also displays an Edge-launched message.

(b) Microsoft Edge desktop view. The mixed dashboard states are retained as captured and are not treated as proof of successful execution or measured performance.

5. EXPERIMENTAL SETUP

Measurements used the stored-template recogniser and an in-set evaluation: 130 letter samples, 13 words, two commands under 4% trajectory noise, seven invalid strokes, and recognition-stage timing. The current interface adds EXCEL, GOOGLE, EMAIL and shortcuts C, E, X, G, M; these additions have no new accuracy results. Participant and capture details were not recorded.

Accuracy is the ratio of correct to total samples. Because the sample sizes are small, 95% Wilson score intervals [19] are reported alongside the point estimates. Latency is reported as mean and 95th-percentile (P95) values for the recognition stage only.

Table -2: Evaluation set and documentation status

Item	Current record
Letter classes / samples	26 uppercase letters; 5 variations each; 130 samples
Word vocabulary	13 words (including CHROME and EDGE)
Trajectory representation	Normalised and resampled to a 128 × 2 tensor
Invalid-input set	7 line / diagonal strokes
Noise test	4% additive noise; CHROME, EDGE
Prototype software	Web-based "Air-Writing System" interface (word · intent · action) run in Google Chrome on Windows (Fig. 2)
Input mode shown	Canvas mode in Fig. 2; webcam input is the first pipeline stage listed in the interface
Example trajectory	268 raw points over 8.5 s (about 31.5 points/s) in Fig. 2
Participants tested	50 users tested; responses collected via Google Forms

Table 2 summarises the original in-set evaluation. Fig. 2 shows the current webcam-based EDGE interface; expanded commands are implementation updates and were not included in the original 13-word accuracy test.

6. RESULTS

6.1 Recognition Accuracy

Table 3 and Fig. 3 summarise the measured accuracy. One of 130 letter samples was misclassified. The interval for

letter accuracy is narrow (95.8–99.9%), whereas the 13-word result has a lower bound of 77.2%: a sample of 13 words cannot distinguish a perfect recogniser from one that is correct about four times in five.

Table -3: Measured results with 95% Wilson intervals

Task	Correct / total	Rate	95% interval
Letter recognition	129 / 130	99.2%	95.8–99.9%
Word recognition	13 / 13	100%	77.2–100%
Invalid-input rejection	4 / 7	57.1%	25.0–84.2%

Intervals computed from the counts above [19].

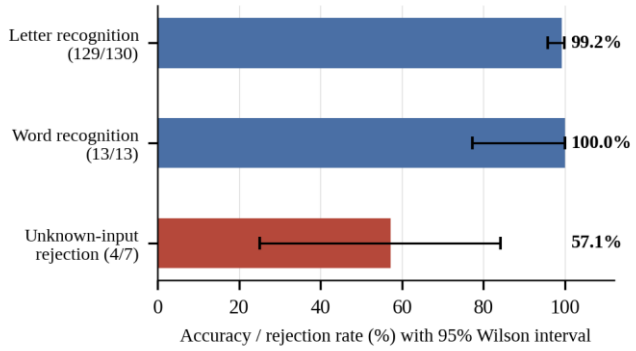


Fig -3: Measured recognition accuracy and invalid-input rejection rate with 95% Wilson intervals.

6.2 Similar-Letter Analysis

The only letter error was a P/R confusion: one of the five P/R variations was misclassified. The groups C/D/G/O, E/F, I/J, M/N, U/V, and V/Y showed no confusion in the tested variations. This describes the evaluated sample and does not show that these letters are generally separable.

6.3 Noise and Invalid-Input Behaviour

At 4% additive trajectory noise, CHROME and EDGE were both recognised correctly, with alternative-letter substitution enabled; accuracy was observed to degrade above about 6% noise. The noise definition and the number of noisy trials were not recorded, so this supports only a statement about mild noise, not a robustness claim.

Rejection of invalid input is the more consequential result for application control. Only 4 of 7 invalid line and diagonal strokes were rejected, and 3 were accepted as valid input (Table 3). With an interval of 25.0–84.2%, the true rejection rate could be well below one half, so the current prototype cannot be considered safe for automatic execution.

6.4 Recognition-Stage Latency

Table -4: Recognition-stage latency

Input	Mean (ms)	P95 (ms)	Mean per letter (ms)
Single letter	0.91	1.33	0.91
Four-letter word	3.99	4.35	1.00
Six-letter word	6.24	6.89	1.04

Mean per letter is computed from the mean latency divided by word length.

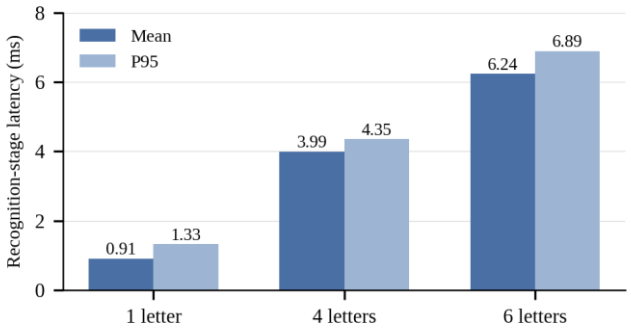


Fig -4: Recognition-stage latency for different input lengths (mean and 95th percentile).

Latency grows close to linearly with word length, at roughly 1 ms per letter. These figures cover the recognition stage only. They exclude camera acquisition, hand tracking, preprocessing, intent mapping, the confidence check, and action execution, and the measurement hardware was not recorded. They must therefore not be reported as camera-to-action latency.

7. DISCUSSION

The measurements show that the prototype recognises a constrained set of uppercase letters and short words in-set. They are evidence of working prototype behaviour and not of general performance: the word result rests on 13 words, the letter result on one stored template per letter, and none of the samples is documented as coming from unseen writers. Word accuracy (100%) exceeding letter accuracy (99.2%) should not be over-interpreted. A vocabulary-constrained decoder or the alternative-letter substitution option may correct letter errors at word level, and an ablation with and without substitution is needed to separate the two effects.

The accuracies in Table 1 cannot be ranked against these results because the datasets, sensors, and user-dependent or user-independent protocols differ. The contribution of this work lies in the interaction layer and its evaluation protocol rather than in character accuracy, which is already above 99% in several camera-based and trajectory-based studies [10], [12].

The principal risk is false command execution: three of seven invalid strokes were accepted. The EDGE screenshot shows an OPEN_APPLICATION intent, a 30.9% match score, and an awaiting-confirmation message; the companion image shows the Edge desktop. These are interface examples, not measured accuracy or proof of successful execution. Confirmation, an allowlist, and a short-lived token add safeguards, but need dedicated security and usability testing.

Finally, the recognition-stage latency (at most 6.24 ms here) is small compared with the 33 ms frame interval of a typical 30 fps camera and with the multi-second writing time (8.5 s in Fig. 2). This suggests that recognition is unlikely to be the dominant delay, but only a camera-to-action measurement can confirm it.

8. LIMITATIONS

1. The system recognises only 13 predefined words, limiting unrestricted sentence formation and recognition of new commands.
2. Variations in handwriting gestures, writing speed, finger movement and individual writing patterns may affect recognition accuracy.
3. Camera angle, lighting conditions, background interference and hand occlusion may affect fingertip tracking.
4. Recognition results are based on a constrained evaluation and do not establish performance on completely unseen users.
5. The system depends on continuous and accurate fingertip detection, making tracking interruptions a potential source of recognition errors.
6. Similar-looking alphabet gestures may cause confusion between characters with closely related writing trajectories.
7. Changes in writing size, direction and starting position may affect trajectory consistency despite the normalisation process.
8. The system has limited ability to distinguish intentional writing gestures from random or accidental hand movements.

9. OBSERVATIONS

This section consolidates what was actually observed in the experiments reported in Section 6. Every observation describes the evaluated sample only; none comes from unseen participants, and all should be read together with the limitations in Section 8.

Letter recognition. Of the 130 uppercase-letter samples (26 letters with five variations each), 129 were recognised correctly, giving 99.2% accuracy with a 95% Wilson interval of 95.8–99.9%. The single error was a confusion between P and R, in which one of the five variations of that pair was misclassified. The remaining similar-letter groups examined (C/D/G/O, E/F, I/J, M/N, U/V, and V/Y) showed no confusion in the tested variations. Because the recogniser matches each trajectory against one stored template per letter, this result indicates that the normalised trajectory and template matching separate the tested single-stroke uppercase shapes well, but only for these samples and not for other writers or writing styles.

Word recognition. All 13 words of the evaluation vocabulary were recognised correctly (100%), but with only 13 words the 95% interval extends from 77.2% to 100%. The result is therefore consistent with correct word formation from recognised letters in a constrained vocabulary, and it cannot be taken as evidence of performance on an open vocabulary. The word-level rate (100%) being higher than the letter-level rate (99.2%) should not be over-interpreted, since the alternative-letter substitution option or a constrained vocabulary may have corrected the single letter error at word level; this was not isolated experimentally.

Noise and invalid input. With 4% additive trajectory noise, both tested words, CHROME and EDGE, were recognised correctly, with alternative-letter substitution enabled, and accuracy was observed to degrade above about 6% noise. This indicates tolerance to mild noise only: two words were tested at one noise level, and the exact noise definition and number of trials were not recorded. The behaviour on invalid input was weaker. Of seven invalid line and diagonal strokes, four were rejected (57.1%, 95% interval 25.0–84.2%) and three were accepted as valid input. The prototype therefore cannot yet separate arbitrary strokes from intended letters, and this is the observation with the most direct consequence for command safety.

Confidence and intent mapping. The current interface supports CHROME, EDGE, EXCEL, GOOGLE, and EMAIL, plus shortcuts C, E, X, G, and M. The supplied dashboard shows EDGE, a 30.9% match score, OPEN_APPLICATION, and awaiting confirmation; the companion capture shows the Edge desktop. These images demonstrate the updated interface, not a new recognition trial. The original 13-word evaluation remains the only reported word-accuracy result. The confirmation token, allowlist, and local host/origin checks have not undergone formal security evaluation.

Recognition-stage latency. Mean recognition-stage latency was 0.91 ms for a single letter, 3.99 ms for a four-letter word, and 6.24 ms for a six-letter word, with 95th-percentile values of 1.33, 4.35, and 6.89 ms. Dividing by word length gives about 0.91, 1.00, and 1.04 ms per letter, so latency grows close to linearly with the number of letters. These values cover the recognition stage only. Camera acquisition, hand tracking, preprocessing, intent mapping, the confidence check, and action execution are excluded, and the measurement hardware was not recorded, so they must not be read as camera-to-action latency.

Summary of the evidence. Taken together, the observations support three statements: the template-based recognition stage is accurate and fast on the constrained, in-set samples evaluated; it tolerates mild trajectory noise; and its handling of invalid input is currently inadequate for automatic command execution. They do not support claims of user-independent accuracy, robustness to speed, scale, or position, intent-classification quality, improvement over learned baselines, or real-time end-to-end performance. These claims depend on the experiments identified in the Conclusion.

10. CONCLUSIONS

The webcam-based prototype links fingertip trajectories to words and application intents. In the original in-set evaluation, it recognised 129/130 letters (99.2%) and 13/13 words (100%), and rejected 4/7 invalid strokes. The current interface adds CHROME, EDGE, EXCEL, GOOGLE, and EMAIL commands with confirmation and allowlist safeguards. Expanded-command accuracy and end-to-end security remain to be evaluated.

The next stage is to document the dataset, apply a participant-level split, implement the baseline models, evaluate robustness and intent classification, add an explicit reject class, and measure camera-to-action latency. The final contribution and claims of the work should be set by the evidence from these experiments.

ACKNOWLEDGEMENT

The authors thank the faculty and the department for guidance and feedback during the development and evaluation of this work.

REFERENCES

[1] M. Chen, G. AlRegib, and B.-H. Juang, “Air-writing recognition Part I: Modeling and recognition of characters, words, and connecting motions,” IEEE Trans. Human-Mach. Syst., vol. 46, no. 3, pp. 403–413, 2016, doi: 10.1109/THMS.2015.2492598.

[2] M. Chen, G. AlRegib, and B.-H. Juang, “Air-writing recognition—Part II: Detection and recognition of writing activity in continuous stream of motion data,” IEEE Trans. Human-Mach. Syst., vol. 46, no. 3, pp. 436–444, 2016, doi: 10.1109/THMS.2015.2492599.

[3] 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 Syst. Appl., vol. 136, pp. 217–229, 2019, doi: 10.1016/j.eswa.2019.06.034.

[4] M. S. Alam, K.-C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, “Trajectory-based air-writing recognition using deep neural network and depth sensor,” Sensors, vol. 20, no. 2, p. 376, 2020, doi: 10.3390/s20020376.

[5] C.-H. Hsieh, Y.-S. Lo, J.-Y. Chen, and S.-K. Tang, “Air-writing recognition based on deep convolutional neural networks,” IEEE Access, vol. 9, pp. 142827–142836, 2021, doi: 10.1109/ACCESS.2021.3121093.

[6] F. A. Al Abir, M. A. Siam, A. Sayeed, M. A. M. Hasan, and J. Shin, “Deep learning based air-writing recognition with the choice of proper interpolation technique,” Sensors, vol. 21, no. 24, p. 8407, 2021, doi: 10.3390/s21248407.

[7] B. Saez-Mingorance, J. Mendez-Gomez, G. Mauro, E. Castillo-Morales, M. Pegalajar-Cuellar, and D. P. Morales-Santos, “Air-writing character recognition with ultrasonic transceivers,” Sensors, vol. 21, no. 20, p. 6700, 2021, doi: 10.3390/s21206700.

[8] S. Ahmed, W. Kim, J. Park, and S. H. Cho, “Radar-based air-writing gesture recognition using a novel multistream CNN approach,” IEEE Internet Things J., vol. 9, no. 23, pp. 23869–23880, 2022, doi: 10.1109/JIOT.2022.3189395.

[9] M. S. Alam, K.-C. Kwon, S. M. Imtiaz, M. B. Hossain, B.-G. Kang, and N. Kim, “TARNet: An efficient and lightweight trajectory-based air-writing recognition model using a CNN and LSTM network,” Hum. Behav. Emerg. Technol., 2022, doi: 10.1155/2022/6063779.

[10] T. Watanabe, M. Maniruzzaman, M. A. M. Hasan, H.-S. Lee, S.-W. Jang, and J. Shin, “2D camera-based air-writing recognition using hand pose estimation and hybrid deep learning model,” Electronics, vol. 12, no. 4, p. 995, 2023, doi: 10.3390/electronics12040995.

[11] X. Tan, J. Tong, T. Matsumaru, V. Dutta, and X. He, "An end-to-end air writing recognition method based on Transformer," IEEE Access, vol. 11, pp. 109885–109898, 2023, doi: 10.1109/ACCESS.2023.3321807.

[12] A. K. Singh and D. Koundal, "A temporal convolutional network for modeling raw 3D sequences and air-writing recognition," Decis. Anal. J., vol. 10, p. 100373, 2024, doi: 10.1016/j.dajour.2023.100373.

[13] C. Lin, Z. Sun, A. Ahmad, X. Fan, Y. Wang, L. Wang, X. Fan, and G. Wu, "AirWrite: An aerial handwriting trajectory tracking and recognition system with mmWave," IEEE Trans. Mobile Comput., vol. 23, no. 12, pp. 13325–13341, 2024, doi: 10.1109/TMC.2024.3425709.

[14] M. K. Jabde, C. H. Patil, A. D. Vibhute, and S. Mali, "A comprehensive literature review on air-written online handwritten recognition," Int. J. Comput. Digit. Syst., vol. 15, no. 1, pp. 307–322, 2024, doi: 10.12785/ijcds/150124.

[15] S. Arora, L. Malik, S. Goyal, D. Bhattacharjee, M. Nasipuri, and O. Krejcar, "Devanagari character recognition: A comprehensive literature review," IEEE Access, vol. 13, pp. 1249–1284, 2025, doi: 10.1109/ACCESS.2024.3520248.

[16] H. Kunt, Z. Yetgin, F. Gözükara, and T. Çelik, "Letter and person recognition in freeform air-writing using machine learning algorithms," IEEE Access, vol. 13, pp. 23142–23155, 2025, doi: 10.1109/ACCESS.2025.3536609.

[17] H.-H. Li and C.-C. Hsieh, "Dynamic hand gesture recognition using MediaPipe and Transformer," Eng. Proc., vol. 108, no. 1, p. 22, 2025, doi: 10.3390/engproc2025108022.

[18] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device real-time hand tracking," arXiv:2006.10214, 2020.

[19] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," J. Amer. Statist. Assoc., vol. 22, no. 158, pp. 209–212, 1927.