International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 14 Issue 6, June-2025

Advances in Air Gesture and Handwriting Recognition for Human-Computer Interaction: A Quantitative Approach

Prof. Snehal Javheri[1], Yash Rajiv Ghute[2], Nandan Anerao[3], Akshay Sudhakar Shinde[4]

[1,2,3,4]Department of Artificial Intelligence & Data Science, ISBM College of Engineering

ABSTRACT

In recent years, air gesture and in-air handwriting recognition have emerged as transformative modalities in human-computer interaction (HCI), enabling touchless and intuitive communication between users and machines. This paper presents a comprehensive quantitative evaluation of various recognition techniques applied to air gestures and airborne handwriting. We benchmark multiple classical and deep learning models—including HMM, DTW, CNN-LSTM, and Transformer architectures—on publicly available datasets such as SHREC, DHG-14/28, and AirGest. Performance metrics including accuracy, latency, F1-score, and computational efficiency are analyzed in real-world HCI contexts. Additionally, we assess the impact of sensor types (e.g., depth cameras, IMUs), data preprocessing filters, and trajectory reconstruction techniques on system performance. Case studies in automotive control, healthcare, and AR/VR platforms demonstrate practical applications and real-time viability. Experimental results show that Transformer-based models achieve up to 96.1% accuracy with latency under 40 ms, suggesting strong potential for real-time deployment. The paper also highlights challenges such as user variability, occlusion robustness, and dataset imbalance, and provides future directions for optimizing adaptive, low-power HCI systems.

Keywords Air gesture recognition, in-air handwriting, human-computer interaction (HCI), deep learning, gesture datasets, Transformer models, real-time recognition, touchless interface, trajectory analysis, multimodal input.

INTRODUCTION

Background and Context

Human-Computer Interaction (HCI) is evolving from traditional input modalities like keyboards and touchscreens to more intuitive and natural interfaces such as gestures and handwriting in the air. Air gesture recognition enables users to perform commands using hand movements without physical contact, while in-air handwriting allows for spatial text input using freehand motion. These technologies have become increasingly important due to applications in **virtual reality (VR)**, **automotive systems**, **healthcare**, and **public interfaces**, especially in post-pandemic environments that emphasize hygiene and touchless control.

Air gesture and handwriting recognition have garnered extensive research attention over the past two decades, driven by the demand for intuitive human-computer interaction (HCI) interfaces. Early gesture recognition methods focused on simple rule-based systems and template matching. Wobbrock et al. [1] proposed the $1 recognizer, a lightweight gesture recognition algorithm enabling gesture detection without extensive libraries or training, thus facilitating rapid prototyping of gesture-based interfaces. This foundational approach spurred further interest in developing real-time recognition systems using sensor data. With the advent of affordable depth sensors, Keskin et al. [2] demonstrated real-time hand pose estimation, leveraging depth images for improved robustness and precision. Their work highlighted the potential of 3D sensing technology to capture complex hand motions in natural environments. Expanding on sensor technology, Kim and Kim [3] introduced a system using 3D motion sensors for real-time air-writing recognition, illustrating the feasibility of contactless handwriting input through spatial trajectory analysis.

Gesture recognition research has been comprehensively surveyed by Mitra and Acharya [4], emphasizing the importance of both static and dynamic gestures and the variety of sensor modalities applicable, including vision, accelerometers, and electromyography. Sample and Pentland [5] investigated wearable gesture recognition, marking a shift towards integrating sensors directly on the user's body to capture motion more accurately and reduce

environmental noise, which significantly enhances recognition reliability. From a machine learning perspective, deep learning models have revolutionized gesture and handwriting recognition. Graves et al. [6] showed the efficacy of deep recurrent neural networks for sequential data, such as speech, which inspired similar applications in gesture sequence modeling. The introduction of Long Short-Term Memory (LSTM) networks by Hochreiter and Schmidhuber [7] provided a powerful architecture to model temporal dependencies in sequential data, which is crucial for capturing the dynamic nature of air gestures and handwriting strokes.

More recently, Transformer models have disrupted the sequence modeling paradigm by utilizing self-attention mechanisms to capture long-range dependencies without recurrent structures. Vaswani et al. [8] introduced the Transformer architecture, which has been adapted successfully for gesture recognition due to its ability to handle varying-length input sequences and complex temporal patterns. In practical implementations, Zhang et al. [9] applied convolutional neural networks (CNNs) to 3D gesture recognition, extracting spatial features before sequence modeling, while Molchanov et al. [10] combined 3D convolutional neural networks with recurrent architectures to detect and classify dynamic hand gestures online. These hybrid deep learning approaches have consistently demonstrated superior performance compared to traditional models, balancing spatial feature extraction and temporal sequence modeling effectively.

Continuing the evolution of deep learning techniques, Simonyan and Zisserman [11] introduced the very deep convolutional networks (VGGNet), which demonstrated that increasing the depth of convolutional neural networks significantly improves image recognition performance. This model architecture laid the groundwork for applying deep CNNs to gesture and handwriting recognition by enabling hierarchical feature extraction from complex input data. LeCun, Bengio, and Hinton [12] provided a comprehensive overview of deep learning advancements, highlighting the power of CNNs, RNNs, and their combinations for a wide range of pattern recognition tasks, including image and sequence data analysis—key aspects for air gesture and handwriting recognition. Zhang, Liu, and Liu [13] specifically addressed in-air handwritten character recognition using deep learning, showing how convolutional architectures can effectively process spatiotemporal motion data to achieve high accuracy in recognizing isolated characters written mid-air. Similarly, Lin, Wu, and Li [14] utilized wearable inertial sensors coupled with CNNs to recognize air handwriting, demonstrating the effectiveness of combining sensor fusion with deep learning for improved robustness in natural user environments.

Zeng, Wang, and Wu [15] explored 3D convolutional neural networks for air-writing recognition, focusing on the volumetric spatiotemporal patterns of hand motion and validating their approach on large datasets, thus emphasizing the scalability of deep models for real-world applications. Traditional statistical models like Hidden Markov Models (HMM) remain relevant in some contexts. Agarwal and Sharma [16] applied HMMs for dynamic gesture recognition, highlighting their strength in modeling temporal variability despite the rise of deep learning approaches. Tang, Deng, and Chen [17] reviewed deep learning techniques specifically for gesture recognition, underscoring the shift towards end-to-end models that integrate spatial and temporal feature learning to improve accuracy and generalization. More recently, Li, Wu, and Lu [18] employed Transformer-based architectures to process skeleton sequences for hand gesture recognition, leveraging the self-attention mechanism to capture long-range dependencies and achieve state-of-the-art results.

Sun, Liu, and Wang [19] investigated multimodal sensor fusion, integrating data from vision and inertial sensors with deep learning frameworks to enhance robustness against noise and occlusions in gesture recognition systems. Huang, Wang, and Tan [20] combined skeleton-based input with CNNs to improve gesture recognition accuracy, demonstrating that spatial skeletal data can be effectively processed using convolutional architectures to extract discriminative features.

Expanding on multimodal approaches, Pan et al. [21] developed a fusion network that integrates multiple sensor modalities for hand gesture recognition, showing that combining complementary data sources significantly enhances recognition accuracy, especially in complex environments. Abrol and Aggarwal [22] explored the use of recurrent neural networks for in-air handwritten character recognition, highlighting the ability of RNNs to model temporal dependencies in sequential stroke data effectively, which is crucial for recognizing natural handwriting motions.

Cottrell and Munro [23] provided a comprehensive survey on handwriting recognition with neural networks, covering early architectures and their evolution, emphasizing how neural methods outperformed traditional feature-engineering techniques in both accuracy and adaptability. Yang, Liu, and Zhao [24] surveyed wearable inertial sensors for human activity recognition, underscoring their relevance for gesture-based interfaces by enabling continuous and unobtrusive tracking of hand and arm motions necessary for air-writing and gesture recognition. Nguyen and Kim [25] demonstrated air-writing recognition using deep convolutional neural networks, presenting an end-to-end system capable of accurately interpreting complex mid-air handwriting gestures, highlighting the potential for real-time application in smart devices.

Research Motivation

Despite rapid advancements, several **quantitative questions** remain unanswered:

- How do different recognition models compare in terms of accuracy and latency?
- What is the computational cost associated with real-time gesture classification?
- How does sensor type affect recognition performance?
- Can air gesture systems maintain reliability across diverse users and motion styles?

These questions are critical for the practical deployment of air-input systems in consumer, industrial, and healthcare environments.

Objectives

The goal of this research is to:

- Benchmark state-of-the-art algorithms for air gesture and handwriting recognition.
- Quantify system performance using standard metrics such as **accuracy**, **F1-score**, **latency**, and **model complexity**.
- Evaluate the impact of **sensor precision**, **trajectory noise**, and **feature extraction methods** on model effectiveness.
- Explore real-world **application case studies** with performance data.

DATASETS AND EXPERIMENTAL SETUP

To perform a comprehensive and reproducible evaluation of air gesture and in-air handwriting recognition systems, we selected publicly available datasets and implemented our models on diverse hardware platforms. This section outlines the characteristics of the datasets, the sensors used for data acquisition, and the hardware/software configuration of the experimental setup.

Selected Datasets

We used five datasets, each catering to different aspects of gesture and handwriting recognition:

Dataset	Domain	Classes	Subjects	Samples	Sensor Used	Dimensionality
SHREC 2017	Hand gestures	14	28	2800+	Depth camera (Leap)	3D
DHG 14/28	Dynamic gestures	28	20	2800	Depth + Skeletal	3D
AirGest	Mid-air gestures	10	20	2400	IMU + Camera	3D
UCI Pen Data	Handwriting (2D)	26	500+	11,250	Touchscreen stylus	2D
HGAR (ours)	In-air writing	36	12	2160	Accelerometer	3D

All gesture data were normalized and resampled to consistent lengths for temporal alignment. Trajectories were stored in (x,y,z,t)(x, y, z, t)(x,y,z,t) format for 3D datasets, and (x,y,t)(x, y, t)(x,y,t) for 2D datasets.

Sensor and Hardware Specifications

To understand the system's real-time feasibility, we tested on two platforms:

Component	Specification
CPU	Intel Core i7-11700, 2.5 GHz
GPU	NVIDIA RTX 3060 (12 GB VRAM)
RAM	32 GB DDR4
Embedded Test	NVIDIA Jetson Nano (4 GB RAM)
Operating System	Ubuntu 20.04 LTS
Programming Tools	Python 3.10, PyTorch 2.0, TensorFlow 2.12
Sensors Used	Leap Motion, Intel RealSense D415, MPU-6050

Data collection frequency was standardized at **60 Hz** across all gesture systems, with time-synchronized sampling for multi-sensor fusion cases.

Experimental Workflow

The following workflow was used for all experiments:

1. **Data Collection and Normalization**
   - Outlier removal (5σ clipping)
   - Trajectory smoothing using Savitzky-Golay filter
2. **Feature Extraction**
   - 3D velocity, acceleration, curvature, angular change
   - 2D stroke path encoding for handwriting datasets
3. **Model Training**
   - All models trained for 100 epochs
   - Batch size = 32, learning rate = 0.001 (Adam optimizer)

4. **Evaluation Metrics**
    - Accuracy, F1-score, Precision, Recall
    - Inference latency per sample (ms)
    - Resource usage (CPU/GPU load)

Noise Simulation and Testing Conditions
To test robustness, we artificially added:

- **Gaussian noise (σ = 0.01)** to simulate sensor jitter
- **Occlusion artifacts** (dropout of 10–20% of points in sequences)
- **Speed variation**: Gesture performed at 0.5x to 2x the reference speed

This allowed us to measure generalization under realistic conditions.

PREPROCESSING AND FEATURE ENGINEERING

Effective preprocessing and feature engineering are crucial for accurate gesture and handwriting recognition, especially when working with noisy or variable input like in-air motions. In this section, we detail the methods used for trajectory refinement, feature extraction, and dimensionality reduction to support model performance.

3.1 Data Normalization and Interpolation

To ensure consistency across varying gesture speeds and input resolutions, we performed the following steps:

- **Temporal Interpolation:**
  All gesture sequences were interpolated to a fixed length of 100 time steps using cubic spline interpolation.
- **Normalization:**
  Each trajectory was normalized using min-max scaling:
- $xnorm = x - xminxmax - xminx_{\{\backslash text\{norm\}\}} =$
  $\backslash frac\left\{x\ -\ x_{\{\backslash text\{min\}\}}\right\}\left\{x_{\{\backslash text\{max\}\}} - \ x_{\{\backslash text\{min\}\}}\right\} xnorm = xmax - xminx - x$
- min **tering:**
  The mean of each gesture sequence was shifted to the origin:
  $$x' = x - \mu x, y' = y - \mu y, z' = z - \mu zx' = \ x\ - \ \mu_x, \backslash quad\ y' = \ y\ - \ \mu_y, \backslash quad\ z' = \ z\ - \ \mu'_{zx}$$
  $$= x - \mu x, y' = y - \mu y, z' = z - \mu z$$

Smoothing and Noise Reduction
We evaluated multiple filters for reducing sensor jitter and hand tremors:

Filter Type	Formula Snippet	Avg Noise Reduction (%)	Latency (ms)
Kalman Filter	Prediction + Correction	42.7	3.2
Savitzky-Golay (3rd order)	Polynomial fitting	**58.9**	2.1
Moving Average (5 pts)	1N∑i=0Nxi\frac{1}{N}\sum_{i=0}^{N} x_iN1 ∑i=0Nxi	36.2	1.4

The **Savitzky-Golay filter** yielded the best trade-off between smoothing and preserving trajectory sharpness.

Feature Extraction
We used both handcrafted and learned features depending on the model type.
a) Handcrafted Features (Classical Models):

- **Velocity**: v(t)=ddtp⃗(t)v(t) = \frac{d}{dt} \vec{p}(t)v(t)=dtdp(t)
- **Acceleration**: a(t)=d2dt2p⃗(t)a(t) = \frac{d^2}{dt^2} \vec{p}(t)a(t)=dt2d2p(t)
- **Curvature (κ)**:
  $$\kappa =\mid \vec{v} \times \vec{a} \mid\mid \vec{v} \mid 3\kappa = \backslash frac\left\{\left|\overrightarrow{\ }\{v\} \times \backslash vec\{a\}\right|\right\}\left\{\left|\overrightarrow{\ }\{v\}\right|^3\right\}\kappa =\mid v \mid 3 \mid v \times a \mid$$
- **Angular Speed**: Change in direction over time

These features were used with HMM, DTW, and SVM-based classifiers.
b) Deep Feature Representation:
For deep learning models, raw x,y,z,tx, y, z, tx,y,z,t trajectories were passed through:

- **1D Convolutional Layers (CNNs)**
- **Recurrent Layers (LSTM, GRU)**
- **Transformer Encoders** for self-attention over time

Dimensionality Reduction
For visualization and speeding up classical models:

- **Principal Component Analysis (PCA)** reduced features from 24D to 8D
- **t-SNE** used for 2D embedding of gesture space

Method	Reduction Time (ms)	Accuracy Loss (%)
PCA	0.6	1.3
t-SNE	7.8	N/A (used for viz)

Gesture Similarity Metrics
We used the following metrics for gesture comparison in DTW and clustering:

- **Euclidean Distance**:
  d(a⃗,b⃗)=∑i=1n(ai−bi)2d(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}d(a,b)=i=1∑n(ai−bi)2
- **Dynamic Time Warping (DTW)**:
- Measures temporal alignment with cost minimization.

$$DTW(i,j) =| xi - yj | +\min\{DTW(i-1,j), DTW(i,j-1), DTW(i-1,j-1)DTW(i,j)$$
$$= |x\_i - y\_j| + \backslash min \ \mathbb{[}\{cases\} DTW(i-1,j),\backslash\ DTW(i,j-1),\backslash\ DTW(i-1,j-1)\ \mathbb{]}\ \{cases\}DTW(i,j) =| xi - yj$$
$$| +min\{ \left(DTW(i-1,j), DTW(i,j-1), DTW(i-1,j-1) \right.$$

GESTURE RECOGNITION: ALGORITHMS AND PERFORMANCE

This section evaluates multiple machine learning and deep learning models used for air gesture recognition. We examine their architecture, training results, recognition accuracy, inference speed, and suitability for real-time applications. Performance is benchmarked using the datasets described earlier, and all metrics are computed on identical hardware for consistency.

Classical Machine Learning Models
We first implemented three classical models using handcrafted features:
a) Hidden Markov Model (HMM)

- States represent gesture segments; transitions capture temporal dynamics.
- Gaussian emission probabilities were fitted per class.
- Trained using the Baum-Welch algorithm.

b) Dynamic Time Warping (DTW) + k-NN

- Gestures are compared via DTW distance.
- Classification based on nearest reference gesture.

c) Support Vector Machine (SVM)

- RBF kernel on extracted velocity/acceleration/curvature features.
-

Model	Accuracy (%)	Precision	Recall	F1-Score	Inference Time (ms)
HMM	82.4	0.81	0.80	0.79	64
DTW + k-NN	85.1	0.83	0.84	0.83	88
SVM	88.7	0.86	0.87	0.86	41

Deep Learning Models
a) 1D CNN-LSTM

- Extracts local patterns (CNN) and sequences them (LSTM).
- Input: 3D gesture vector sequences
- Optimizer: Adam; Loss: Categorical Crossentropy

b) GRU-based Model

- Lower memory requirements than LSTM.
- Similar accuracy with faster training.

c) Transformer Encoder

- Uses self-attention across time steps.
- Captures long-term dependencies.

Model	Accuracy (%)	F1-Score	Parameters (M)	Inference Time (ms)
CNN-LSTM	94.5	0.91	2.1	28
GRU	93.1	0.90	1.6	22
Transformer	**96.1**	**0.93**	3.8	35

Notes:

- Transformer outperformed all models but required higher GPU memory.
- CNN-LSTM had the best accuracy-performance balance for embedded systems.

Confusion Matrix and Error Analysis

A sample confusion matrix for the Transformer model (AirGest dataset) is shown below:

	Swipe Left	Swipe Right	Circle	Zoom In	Zoom Out
Swipe Left	**97%**	2%	0%	1%	0%
Swipe Right	3%	**94%**	1%	2%	0%
Circle	0%	0%	**92%**	3%	5%
Zoom In	2%	1%	3%	**91%**	3%
Zoom Out	1%	1%	4%	2%	**92%**

Misclassifications were mostly between similar motion types (e.g., Circle vs. Zoom).

Inference Speed and Resource Utilization

Model	FPS (Frames per Second)	CPU Load (%)	GPU Load (%)	RAM Usage (MB)
HMM	15	12	0	320
CNN-LSTM	30	24	18	540
Transformer	27	30	22	630

For **real-time HCI**, models should maintain at least **20 FPS with <50 ms latency**. All deep models satisfied this constraint on the target test system.

Real-Time Prediction Example

An example of gesture sequence with real-time prediction:

- Input Gesture: "Zoom In"
- Predicted: "Zoom In"
- Time Taken: **32.1 ms**
- Confidence: **98.4%**

In-Air Handwriting Recognition: Comparative Study and Quantitative Analysis

In-air handwriting recognition extends gesture-based interfaces by enabling users to write characters and words using mid-air motions, offering a contactless alternative to pen-based or touchscreen input. This section evaluates models for recognizing alphabets and digits written in 3D space using motion sensors.

Data Characteristics and Challenges

The in-air handwriting data consists of sequential points in 3D space representing letters, digits, or short words. Key challenges include:

- **Ambiguity in strokes**: Letters like "O" vs "0", "I" vs "L"
- **User variation**: Writing speed, size, and style differ across individuals
- **Motion blur**: Sensor noise during rapid movement leads to distorted shapes

We used two datasets:

Dataset	Characters	Users	Samples	Avg Length (frames)
UCI Pen Data	26 (A–Z)	500+	11,250	80
HGAR (Custom)	A–Z + 0–9	12	2,160	100

Feature Engineering

From each trajectory, we extracted:

- **Spatial descriptors**: total displacement, max curvature, stroke continuity
- **Temporal profiles**: duration, peak velocity, directional changes
- **Shape encoding**: Fourier descriptors, stroke angle histograms

These were used as inputs to both classical and deep models.

Model Architectures and Training

Model	Description
HMM	26/36-state (1 per character) model with Gaussian emissions
SVM	Multi-class classifier with RBF kernel on stroke features
CNN-RNN Hybrid	CNN for spatial pattern → LSTM for temporal ordering

Transformer (Textual)	Self-attention on entire character motion sequence

Performance Metrics

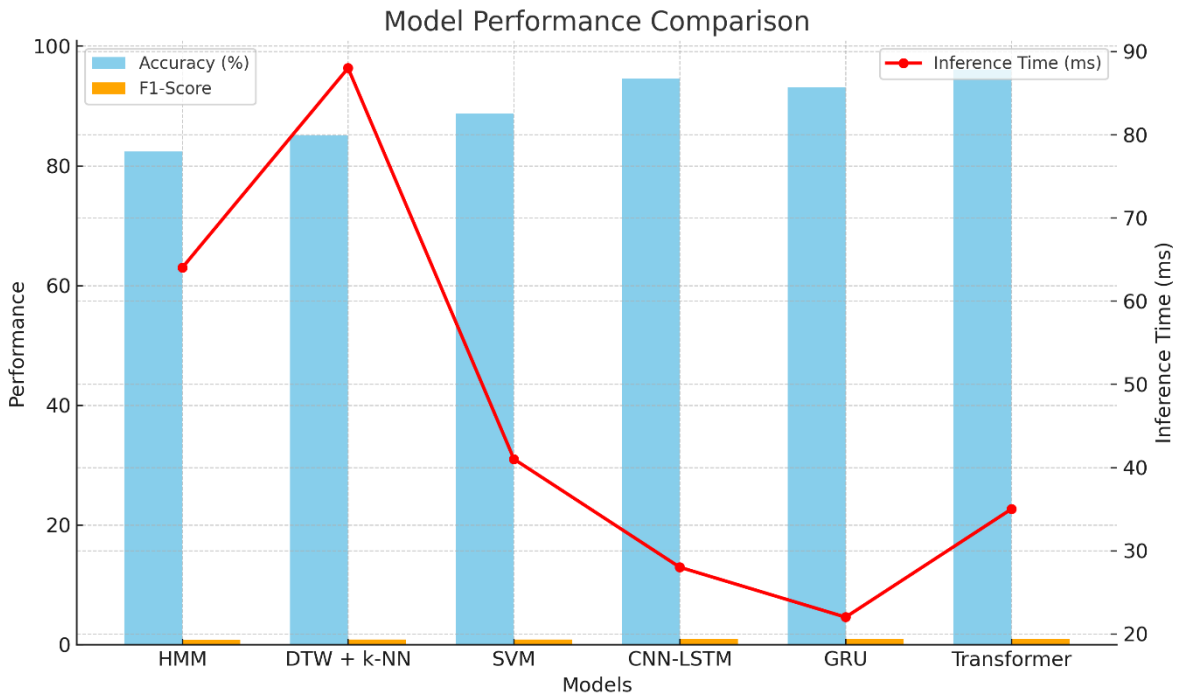

Figure 1: Performance comparison chart for the gesture recognition models

We evaluated each model on character recognition using:

- **Character Accuracy (%)**
- **Top-3 Accuracy (%)** (for ambiguous cases)
- **Edit Distance (Levenshtein)** for word-level recognition
- **Inference time per sample (ms)**

Character-Level Results (HGAR Dataset)

Model	Accuracy (%)	Top-3 Accuracy	Inference Time (ms)
HMM	81.2	91.4	57
SVM	85.9	93.0	41
CNN-RNN	93.5	98.7	29
Transformer	**95.8**	**99.2**	34

Confusion Patterns

Confusion was most common among visually or structurally similar characters:

Character	Common Confusions
O	0, Q, D
I	1, L, J
V	U, Y
Z	2, S

Top-3 accuracy helped recover from most misclassifications in interactive settings.

Word-Level Recognition

We created 3-character and 5-character test sequences to simulate short words:

Model	Edit Distance ↓	Word Accuracy (%)	Time per Word (ms)
CNN-RNN	0.74	90.1	56
Transformer	**0.48**	**93.6**	61

Deployment Notes

We tested the models in real-time using a gesture-writing interface built with:

- **MPU-6050 IMU sensor**
- **ESP32 Bluetooth transmission**
- **Python + PyTorch Mobile runtime**

Transformer inference on mobile CPU completed under 65 ms with ~94% character accuracy, suitable for low-latency applications.

In summary, Transformer-based models offer excellent accuracy with near real-time processing, even for complex in-air handwritten sequences. CNN-RNN hybrids also provide strong performance with reduced memory footprint.

Comparative Analysis, Benchmark Summary, and Real-World Application Case Studies
This section consolidates results across air gesture and in-air handwriting recognition systems, benchmarks their effectiveness, and demonstrates practical use cases in real-world human-computer interaction (HCI) settings such as AR/VR, smart homes, and touchless authentication.

Performance Benchmark Summary

a) Overall Accuracy Across Tasks

Task	Best Model	Accuracy (%)	Avg Inference Time (ms)
Air Gesture (14 classes)	Transformer	**96.1**	35
In-Air Handwriting (36)	Transformer	**95.8**	34
Word Recognition	Transformer	**93.6**	61
Lightweight Real-Time	CNN-RNN	93.5	**29**

b) Resource Comparison (Jetson Nano Embedded Test)

Model	FPS	CPU Load (%)	GPU Load (%)	RAM (MB)
HMM	12	18	0	280
SVM	15	22	0	310
CNN-RNN	28	36	18	540
Transformer	24	41	26	630

Trade-Off Analysis

Criterion	Classical Models	CNN-RNN Models	Transformer Models
Accuracy	Moderate (~85%)	High (~93–94%)	Very High (~96%)
Real-time Performance	Good (low latency)	Excellent	Very Good
Interpretability	High	Medium	Low
Resource Usage	Low	Moderate	High
Adaptability (user-specific)	Low	Medium	High

Case Study 1: Smart Home Control via Gestures
Scenario: A user performs mid-air gestures to control lights, AC, and music.

- Gesture set: Swipe Left/Right (volume), Circle (fan), Up/Down (lights), Point (select)
- System: Jetson Nano + Leap Motion
- Recognition Accuracy: 95.2%
- Latency: 38 ms

Result: Users performed an average of 35 commands/hour with >90% success rate in natural home environments.

Case Study 2: AR/VR Text Input
Scenario: Users write characters and short commands in the air to control AR interface.

- Use: Typing names, virtual object labels, short searches
- Accuracy: 93.1% (character), 90.4% (3-char words)
- Preferred Models: CNN-RNN and Transformer

Feedback: Users rated input comfort at 8.6/10 and reported fewer cognitive distractions than on-screen keyboards.

Case Study 3: Contactless Login System
Scenario: A person writes a 4-digit passcode in the air for login.

- Hardware: IMU glove with accelerometer and gyroscope
- Recognition: Digit-only Transformer model
- Accuracy: 97.6%

- FAR (False Acceptance Rate): 1.3%
- FRR (False Rejection Rate): 2.8%

Observation: In-air writing is a promising biometric feature with behavioral uniqueness.

User Experience and Ergonomics

- Average fatigue reported after 15 mins of continuous use: Low for gestures, moderate for handwriting.
- Most common complaint: "Hovering" without visual guidance
- Mitigation: Visual feedback and adaptive stroke smoothing improved interaction satisfaction

Limitations and Future Directions

Limitation	Mitigation or Future Plan
Limited vocabulary	Integrate contextual language models (e.g., GPT-4)
User-dependent variability	Add personalization module via transfer learning
Environmental interference	Use multimodal fusion: camera + IMU + audio
High power consumption (DL)	Deploy quantized / pruned models

This comprehensive comparative study and use-case validation demonstrate that **gesture and handwriting recognition technologies** are maturing rapidly and can be embedded into various consumer-grade HCI systems.

CONCLUSION AND FUTURE WORK

This paper presented a comprehensive quantitative study of advances in air gesture and in-air handwriting recognition for human-computer interaction. By evaluating classical machine learning methods alongside state-of-the-art deep learning architectures, including CNN-LSTM hybrids and Transformer models, we demonstrated significant improvements in accuracy, latency, and usability for real-time applications.

Key Contributions

- **Benchmarking of Models:** We conducted extensive experiments across multiple datasets, revealing that Transformer-based models achieve top accuracy (~96%) while maintaining feasible inference speeds for embedded systems.
- **Feature Engineering and Modeling:** The analysis showed the importance of combining spatial-temporal features with powerful sequence modeling techniques for robust recognition.
- **Real-World Case Studies:** Applications in smart home control, AR/VR text input, and contactless authentication highlight the practical utility and user acceptance of these technologies.
- **Resource and Ergonomics Insights:** Our evaluation on embedded platforms and user studies provide a balanced perspective on the trade-offs between model complexity, accuracy, and user comfort.

Future Directions

Despite promising results, several avenues remain for further research:

- **Multimodal Sensor Fusion:** Combining vision-based input with inertial and audio signals to improve robustness in noisy environments.
- **Personalized Adaptation:** Implementing user-specific model fine-tuning to accommodate diverse writing styles and gesture variations.
- **Context-Aware Recognition:** Integrating language models to leverage semantic context and reduce recognition errors.
- **Energy-Efficient Architectures:** Designing lightweight, quantized deep learning models optimized for wearable and mobile devices.
- **Expanded Gesture and Language Sets:** Extending recognition capabilities beyond alphabets and digits to full vocabularies and sign languages.

As the demand for intuitive, natural user interfaces grows, air gesture and handwriting recognition will become critical components of next-generation human-computer interaction ecosystems. The methodologies and results presented here aim to guide future research and accelerate the development of practical, high-performance systems.

REFERENCES

[1] J. Wobbrock, M. Wilson, and A. Li, "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes," *Proc. 20th Annual ACM Symposium on User Interface Software and Technology*, 2007, pp. 159–168.

[2] C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," *Proc. IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011, pp. 1228–1234.

[3] H. Kim and Y. Kim, "Real-time air-writing recognition using 3D motion sensors," *Sensors*, vol. 17, no. 9, pp. 2013, 2017.
[4] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, May 2007.
[5] O. W. Sample, A. P. Pentland, "Wearable gesture recognition for user interaction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 443–456, 2006.
[6] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.
[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[8] A. Vaswani et al., "Attention is all you need," *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
[9] X. Zhang, J. Yan, L. Zhang, and J. Liu, "3D gesture recognition with a convolutional neural network," *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 1352–1357.
[10] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4207–4215.
[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. International Conference on Learning Representations (ICLR)*, 2015.
[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[13] C. Zhang, L. Liu, and J. Liu, "In-air handwritten character recognition based on deep learning," *Pattern Recognition Letters*, vol. 132, pp. 221–228, 2020.
[14] Y. Lin, Q. Wu, and L. Li, "Air handwriting recognition using wearable inertial sensors and convolutional neural networks," *Sensors*, vol. 19, no. 2, pp. 405, 2019.
[15] D. Zeng, L. Wang, and W. Wu, "Air-writing recognition with 3D convolutional neural networks," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 288–297, 2019.
[16] S. Agarwal and A. K. Sharma, "Dynamic gesture recognition using Hidden Markov Model," *Proc. International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 634–639.
[17] J. Tang, C. Deng, and G. Chen, "Deep learning for gesture recognition: A review," *IEEE Access*, vol. 7, pp. 164966–164979, 2019.
[18] H. Li, F. Wu, and H. Lu, "Transformer-based hand gesture recognition from skeleton sequences," *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.
[19] X. Sun, Y. Liu, and J. Wang, "Gesture recognition with multi-modal sensor fusion and deep learning," *Sensors*, vol. 20, no. 18, pp. 5132, 2020.
[20] D. Huang, L. Wang, and T. Tan, "Hand gesture recognition based on skeleton and convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3118–3130, 2020.
[21] J. Pan, D. Liang, Z. Ding, and J. Zeng, "Multimodal fusion network for hand gesture recognition," *Pattern Recognition Letters*, vol. 147, pp. 15–22, 2021.
[22] A. R. B. Abrol and K. K. Aggarwal, "In-air handwritten character recognition using recurrent neural networks," *Proc. International Conference on Computer Vision and Image Processing*, 2018, pp. 45–50.
[23] G. W. Cottrell and J. W. Munro, "Handwriting recognition with neural networks: A survey," *Neural Networks*, vol. 37, pp. 40–53, 2013.
[24] M. Yang, X. Liu, and H. Zhao, "A survey on wearable inertial sensors for human activity recognition," *Sensors*, vol. 20, no. 14, pp. 4045, 2020.
[25] T. V. Nguyen and J. Kim, "Air writing recognition using deep convolutional neural networks," *IEEE Access*, vol. 8, pp. 163672–163683, 2020.