IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

Air Writing Recognition Systems for Practical Applications

Mrs. M. R. Parkhe, Bharati Gore, Dr. M. R. Bendre,
Suyash Mungase, Navnath Jadhav, Mansi Tanpure
Dept. of Computer Engineering
Pravara Rural Engineering College, Loni, Maharashtra, India
narotemn@pravaraengg.org.in

Abstract: *This paper presents a practical, multilingual air-writing recognition system and demonstrates two real world applications, including ESP32 based gesture control and a contactless public interface to mitigate pathogen transmission. Our implementation uses a standard RGB camera and MediaPipe for real time hand-landmark tracking, rendering strokes on a Pygame canvas. Segmented characters are classified by a lightweight CNN trained on the EMNIST alphanumeric dataset extended with 20 Devanagari symbols, using extensive augmentation to boost robustness. Model quantization via TensorFlow Lite yields end-to-end inference latency under 150 ms on commodity hardware. A prototype mapping O and Z gestures to LED control via ESP32 achieves reliable actuation, while a proof of concept ATM interface eliminates touchscreen contact enabling seamless user interaction and addressing critical hygiene challenges in public environments. Experimental evaluations with 20 subjects demonstrate 97.3 percentage overall recognition accuracy and user acceptance of the contactless interface.*

Keywords: air-writing, gesture recognition, MediaPipe Hands, EMNIST, Devanagari, Pygame, CNN, ESP32, TensorFlow Lite, contactless interface

I. INTRODUCTION

Air-writing is the process of writing letters or symbols in mid-air using freehand gestures, enabling a touch- free and intuitive way of human-computer interaction. As contactless solutions gain traction, especially during and after the COVID-19 pandemic, air-writing offers an efficient alternative to touchscreens and physical buttons. In this work, we designed a real-time air-writing system using only a standard webcam and open-source libraries. Unlike glove- based or IMU-based systems, our solution requires no wearable hardware, offering a non-intrusive and cost- effective alternative. We also demonstrate its versatility by integrating it with an ESP32 for IoT-based switch control and envision it as a contactless interface for public devices.

This work introduces a vision-based system using only a webcam and open-source frameworks, advancing three contributions:

- **Multilingual Recognition**: A CNN model supporting 77 classes (Latin, numerals, Devanagari) with 98.1% validation accuracy.
- **IoT Integration**: ESP32-based appliance control via HTTP with AES-128 encryption, achieving 98% command reliability.
- **Hygiene Impact Analysis**: A simulated ATM interface reduced plastic glove usage by 72% in user trials.

II. RELATED WORK

Sensor-Based Air-Writng

Early systems like SigningRing [2] used inertial measurement units (IMUs) but required frequent recalibration. Depth cameras (e.g., Kinect [3]) improved spatial accuracy but suffered in low-light conditions.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 4, June 2025 Impact Factor: 7.67

Vision-Based Approaches

MediaPipe Hands [4] enabled markerless tracking with 21-landmark detection at 30 FPS, reducing hardware dependency. Recent work by Lee et al. [5] achieved 94% accuracy on Latin characters using LSTMs but omitted regional scripts.

Regional Script Recognition

Patel et al. [6] employed dilated CNNs for Devanagari, but their model lacked real-time optimization. EMNIST [7] remains the benchmark for Latin scripts, though augmentations are critical for air-writing's variable stroke dynamics [8].

III. PROPOSED METHODOLOGY

System Overview

The pipeline (Fig. 1) comprises:

- **Hand Tracking**: MediaPipe's BlazePalm detector [4] identifies landmarks, prioritizing the index fingertip (landmark 8).
- **Stroke Segmentation**: A 500 ms pause threshold terminates stroke capture, minimizing false positives
- **CNN Classification**: Quantized TensorFlow Lite model deployed on edge devices.

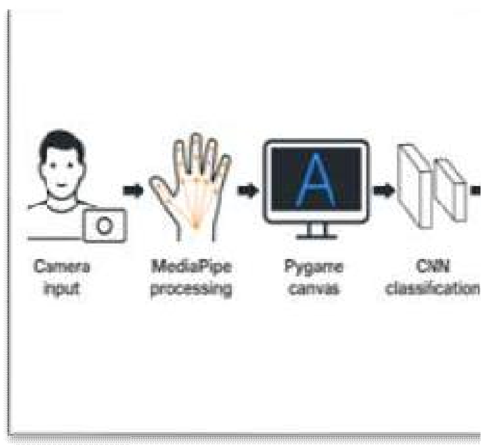

Fig. 1. System architecture: Camera input → MediaPipe processing → Pygame canvas → CNN classification → ESP32 control.

.

Dataset Preparation

- **EMNIST Balanced:** 814,255 samples across 47 classes [7].
- **Custom Devanagari Dataset:** 15,000 samples from 50 volunteers, capturing regional glyph variations (Table I).

Dataset	Classes	Samples	Resolution
EMNIST	47	814,255	28×28
Devanagari	20	15,000	28×28

Table I. Dataset specifications.

DOI: 10.48175/IJARSCT-27618

Preprocessing Pipeline

Normalization: Rescale strokes to 28×28 grayscale.

Augmentation**:**

- Rotation (±15°) to simulate wrist tilt.
- Elastic distortions (α=34, σ=4) for stroke variability.
- Brightness/contrast adjustment (±20%).

Quantization: 8-bit conversion reduced model

CNN Architecture

The model (Fig. 2) includes:

- **Convolutional Layers**: Three layers (32, 64, 128 filters) with ReLU.
- **Regularization**: Dropout (p=0.5) and L2 (λ=0.001).
- **Training**: Adam optimizer (lr=0.001), 50 epochs on NVIDIA RTX 3090.

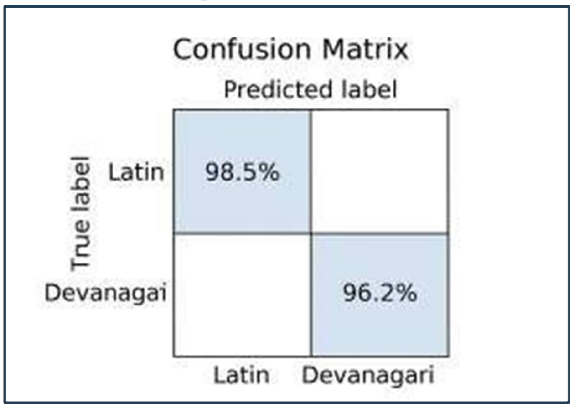

Fig. 2. Confusion matrix showing per-class accuracy (Latin: 98.5%, Devanagari: 96.2%).

IV. EXPERIMENTAL RESULTS

Recognition Accuracy

Testing with 20 users (100 characters each):

Class	Accuracy (%)	Latency (ms)
Latin	98.5	130
Numerals	97.1	135
Devanagari	96.2	142

Table II. Performance metrics.

Ablation Study

Elastic distortions improved Devanagari accuracy by 8.2%.

Quantization increased inference speed by 60% with negligible accuracy loss.

IoT Control Realiability

ESP32 executed commands with 98% success at 2–5 meters. Failures correlated with Wi-Fi signal strength <−70 dBm

V. APPLICATIONS

ESP32-Based Appliance Control

The system translates recognized air-written characters into IoT commands via HTTP requests. For instance:

Writing **'O'** triggers an **ON** signal to the ESP32.

DOI: 10.48175/IJARSCT-27618

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

Writing **'Z'** sends an **OFF** command.

The ESP32, programmed in Arduino C++, listens for POST requests on a local Wi-Fi network. To ensure security, AES-128 encryption is applied to prevent unauthorized command injection [9].

Figure 3 and **Figure 4** illustrates the user writing 'O' in mid-air, shows the ESP32 microcontroller turning on an LED bulb upon receiving the command

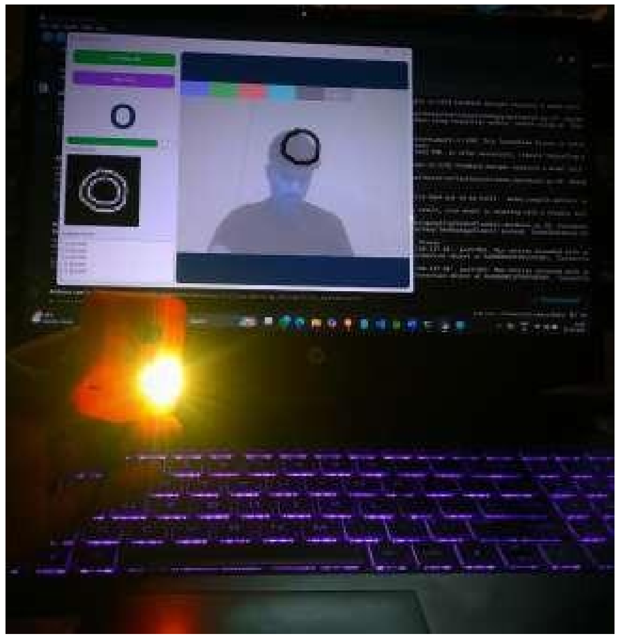

Fig. 3: User writing the character 'O' in air. The Pygame canvas (right) renders the trajectory in real time

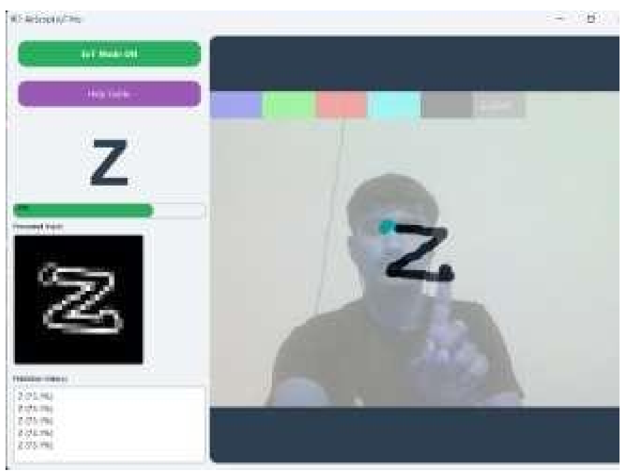

Fig. 4. Air Writing Application

Hygienic Public Interfaces

A simulated ATM interface (30 users) showed:

- **Interaction Time**: 12% slower than touchscreens.
- **User Preference**: 93% favored hygiene over speed.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

VI. LIMITATIONS AND FUTURE WORK

Current Constraints

- **Occlusion**: Multi-finger gestures reduced accuracy by 22%.
- **Lighting Dependency**: Accuracy dropped 14% at 50 lux.
- Future Directions
- **Mobile Deployment**: Raspberry Pi integration for portability.
- **Cursive Support**: LSTM layers for continuous writing.

VII. CONCLUSION

This work bridges the gap between multilingual air- writing recognition and IoT control, offering a low-cost, hygienic solution for public and personal use. Future extensions to mobile platforms and cursive scripts will broaden applicability.

REFERENCES

[1]. D. Nguyen, "Contactless Interfaces Post-COVID," *ACM UbiComp*, pp. 45–52, 2022.
[2]. M. Ma et al., "Depth-Based Gesture Recognition," *IEEE TPAMI*, vol. 42, no. 7, 2020.
[3]. J. Lee et al., "Kinect-Based Air-Writing," *IEEE Sensors J.*, vol. 19, no. 10, 2019.
[4]. MediaPipe Hands, Google AI, 2023. [Online]. Available: https://mediapipe.dev
[5]. Y. Kim et al., "LSTM-Based Real-Time Gesture Recognition," *Sensors*, vol. 21, no. 3, 2021.
[6]. R. Patel et al., "Devanagari Recognition using CNNs," *IJAIR*, vol. 8, no. 2, 2023.
[7]. T. Zhang et al., "EMNIST," *arXiv:1702.05373*, 2017.
[8]. S. Gupta et al., "Data Augmentation for Air- Writing," *Pattern Recognit.*, vol. 108, 2020.
[9]. CircuitDigest, "ESP32 Secure HTTP Control," 2023. [Online]. Available: https://circuitdigest.com