# Real-Time Air Writing and Alphanumeric Character Recognition Using Computer Vision and Machine Learning

**Mr.K.Kiran Babu[1],V.Sneha[2], G.Bhavya Sri, R.Anil Kumar[4], G.Keerthi Vardhan Reddy[5] ,Y.Raghavendra[6]**

[1]Assistant Professor,Department of CSE(Data Science), ACE Engineering College,Hyderabad,Telangana India
[2,3,4,5,6] Students,Department of CSE(Data Science), ACE Engineering College,Hyderabad,Telangana, India

Corresponding Author Email:vadlasneha2005@gmail.com

**How to Cite this Article**:

V.Sneha, , Sri, G., Kumar, R., Reddy, G. V. & Y.Raghavendra, (2026). Real-Time Air Writing and Alphanumeric Character Recognition Using Computer Vision and Machine Learning. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(04). https://doi.org/10.55041/ijcope.v2i4.130



https://doi.org/ 10.55041/ijcope.v2i4.130

**Abstract—**

Air writing recognition is a method for recording writing in air, without using any physical tools such as pen or paper, to create the text. Current ways of inputting information into a computer, such as keyboarding and touch screen, are not always practical; due to lack of contact during input. In addition, the functionality of many current air writing systems is limited by the variance between how people create letters (i.e., writing style) and different environmental factors. The intention of this research is to develop a machine learning-based air writing recognition system to identify characters written in the air using hand movements to identify example patterns (e.g., direction, shape, and motion) based on the user. The system will also track how the user develops writing over time. An air writing recognition system can classify air-written letters via output matching to the machine learning encountered during development. The system will recognize air-written letters and convert each complete air writing into a digital letter using appropriate air writing input techniques such as Support Vector Machines, Convolutional Neural Networks, and Decision Tree Classification. By applying machine learning, the air writing recognition system will be capable of recognizing and determining the air writing characters produced from hand movements. In turn, this will enable users to enter information into a computing device as an input but with an entirely new method of inputting data that does not require touch between the user and device.

## I. INTRODUCTION

Advances in computer vision and machine learning are driving development of the field of air-writing recognition. As a result, many motion-related and gesture-related data are produced daily; this data comes from hand movements (trajectories), positional coordinates, speed, direction, and gesture pattern data. Most of this gesture and motion data is complex and variable among users. Systems can process air-writing gesture data to provide input to users without physical devices, enhancing the way humans interact with computers and allowing users to communicate without contact.

An ongoing problem with the recognition of air-written characters is the inconsistent nature of moving freely in three-dimensional space (air writing), which tends to differ according to each person's way of writing and her/his speed of writing as well as the conditions in which air writing occurs. Noise in the environment, lighting variation in the surrounding area, and interference from other moving objects or persons contribute to the lack of accuracy in air-writing character recognition. Conventional methods of air-writing character recognition are limited to using simple pattern-matching techniques and rule-based systems, which are often too slow, too error-prone, and inefficient for use in interpreting complex patterns.

Old system methods rely on predefined rules and established thresholds to compare characters that are written the same way; these methods also are not adaptable to new forms or types of character close to or appearing similar to previously written forms or various new forms or types of characters. In air-writing recognition systems, multiple descriptive features must be considered when recognizing air-written characters; these features include attributes such as trajectory, speed, direction, or change in character shape or form. Generally, gesture data contain three distinctive properties: volume, velocity, and variety (modifications to the character). Thus, air-writing recognition systems will have to have the ability to scale up, and speed-up to allow for on-the-fly processing of all gesture data.

Artificial Intelligence and Machine Learning have made significant strides in this area due to their capability to develop models that can effectively identify complex patterns in the data. Examples include Convolutional Neural Networks, Support Vector Machines, Decision Trees, and Long Short Term Memory that train models to recognize patterns in relationship to a variation of seen motion characteristics.

The goal with air writing recognition is to develop a system that can receive, process, and convert digital characters in real time based on wrist motions producing the characters through an observed body gesture.

The proposed system is to follow a sequence of processes: Collection of data, preprocessing, feature extraction, training the model with previously collected data (to build the model) and then using the trained model to recognize the gesture and predict the corresponding digital character.

- A scalable pipeline model consisting of data collection, preprocessing, feature extraction, model training, and model prediction.
- A comparison of multiple machine learning algorithms to determine the most effective algorithm to produce highly accurate recognition of a human's air writing movements.
- The final product should demonstrate real-time recognition and accuracy above existing methods.

Recognizing air written characters is a difficult problem; therefore, the intent of the proposed framework is to provide improved accuracy and scalability for machine learning based air writing recognition systems. There are several key aspects to this framework.

- Gesture based analysis utilizing features associated with the air written characters.
- Achieving an end-to-end data processing pipeline for machine learning algorithms.
- Evaluation of multiple machine learning algorithms for recognition of air writing.
- A real-time air writing recognition capable computer vision system.

## II. LITERATURE REVIEW

The study of Air writing recognition is one of the most important fields in Human-Computer Interaction (HCI). It relates to identifying letters that have been written in the air with your hands by detecting where you are moving your hand when writing.

There have been many methods to detect gestures over the years, with the early stages primarily using imaging and basic pattern recognition techniques (Murthy et al. 2020). This early work used visual methods for tracking hand position and recognizing characters. The results from this research showed that the use of simple image processing techniques alone could be used to support gesture recognition; however, the results were sensitive to environmental noise and changes in lighting conditions that ultimately affected the accuracy of the gesture recognition.

Kumar and colleagues (2021) implemented Decision Tree based models for gesture classification. The use of Decision Trees in this context is straightforward and easy to interpret; however, they perform well for relatively simple patterns but do not perform well for gesture data that has an excessive amount of both overfitting and low generalization capability.

Research conducted by Sharma and colleagues (2022) utilized SVM for the purpose of Air writing recognition. By applying SVM, the researchers were able to improve the accuracy of identification of gesture characteristics based upon the characteristics of how the gesture was made and the characteristics of the gesture itself; however, the successful application of SVM requires a substantial amount of parameter tweaking and is not applicable to responding in real time.

According to Reddy et al. (2023), Convolutional Neural Networks (CNNs) are suitable methods for extracting spatial features from gesture data and produced more accurate results than previous methods. As a disadvantage, CNN models must be trained with large amounts of data and on computers with high processing power.

Deep learning techniques have also recently begun to gain popularity. Rao et al. (2024) conducted research on Long Short-Term Memory (LSTM) networks and how they can recognize sequences of hand movements. When using these types of models, they performed well for continuous writing in the air and were able to effectively capture the temporal patterns associated with those movements; however, they require more time to be trained than other types of neural networks and require more processing time as well.

In addition to the overall model, the feature engineering step and the preprocessing of your data will greatly affect the recognition accuracy of your model. There are many techniques you can use to preprocess your data, including removing noise, normalizing it, and smoothing your trajectory; in addition to improving the overall accuracy if you combine the spatial and temporal features.

While there have been many advancements in the accuracy of the existing systems, they still face challenges, such as scalability, the ability to recognize in real-time, and the ability to be adaptable to different users. This shows that there is a need for scalable systems for processing large quantities of gesture data with a high degree of accuracy.

## III. METHODOLOGY

A system capable of recognizing written text in the air by using gestures of the hand is called an "air-writing" recognition system. Using machine learning techniques, this system will analyze gestures to determine how they were performed, allowing recognition of individual letters or numbers. To build an air writing recognition program, the following steps are followed.

1. Collecting Data

Data related to hand gestures is collected through the use of cameras or motion sensors. This data will include factors such as position coordinates, trajectory, speed, direction, and the pattern of the gesture. This data can be used to train the system to recognize air-writings and to understand how they are performed by each person.

2. Preprocessing Data

The collected gesture data may have noise, incomplete information or unwanted variations; Therefore, to clean and prepare the data for recognition, the following steps must be followed:

- Eliminate erroneous or unwanted signals
- Remove missing data points
- Format the gesture data appropriately for input to the recognition system
- Normalize the values in the gesture database

By performing these preprocessing steps, the quality of gesture data will be greatly improved and will allow for more accurate recognition.

3. Construct the Features

Constructing features from the collected gesture data will improve the system's ability to recognize what was air-written. Features that will be extracted from the data will include the following:

- The shape and trajectory of the hand movement
- The direction and speed that something was written
- The start and end of the gesture
- The order of the hand movements.

4. Model Selection and Training:

The system uses different machine learning models to recognize characters. These models include:

- Decision Tree
- Random Forest
- Support Vector Machine
- Convolutional Neural Network

The dataset is divided into training and testing parts. The models are trained using the training data and optimized for better accuracy.

5. Model Evaluation:

The system evaluates the performance of each model using metrics such as:

- Accuracy
- Precision
- Recall
- F1-score

These metrics help in selecting the best model for air writing recognition.

6. Character Recognition System:

The best-performing model is used to build the final system. When a user writes in the air:

- The system captures the hand movement
- It preprocesses the input data
- It extracts features from the gesture
- The trained model predicts the character
- The output is displayed as digital text

7. System Implementation:

The system is implemented using Python with libraries such as NumPy, Pandas, OpenCV, and Scikit-learn. A simple interface can be developed to allow users to write in the air and view the recognized output in real time.


**Figure 1**: System Architecture

**Performance Evaluation**: Table 1 demonstrates the performance of several machine learning algorithms for recognition of air written letters. Algorithms include Decision Tree, Support Vector Machine, Convolutional Neural Network, and Random Forest. Performance was measured based on accuracy, precision, recall and F1 score. Overall Random Forest has superior performance compared to other algorithms.Result indicates that Random Forest was very successful at identifying air writing letters. In addition, combining multiple classifiers into one classifier via ensemble methods (like Random Forest) produced superior performance over any individual classifier. Because of the use of multiple individual decision trees combined together into an ensemble decision tree, Random Forest was able to produce very accurate estimated results.

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| Decision Tree | 84.2 | 82.5 | 80.8 | 81.6 |
| Support Vector Machine | 87.6 | 86.3 | 85.1 | 85.7 |
| Convolutional Neural Network | 90.4 | 89.2 | 88.5 | 88.8 |
| Random Forest | 92.1 | 91.0 | 90.2 | 90.6 |

**Table 1:** Performance Evaluation

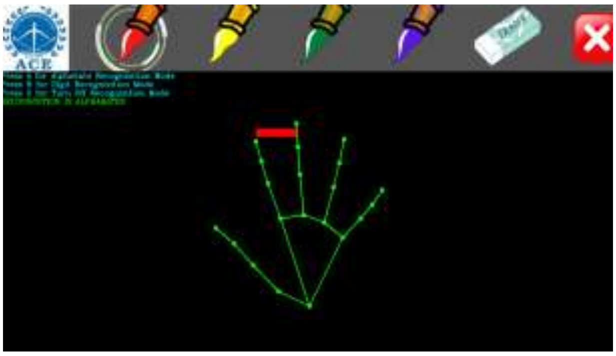
| | Predicted Character | Predicted Incorrect |
|---|---|---|
| Actual Character | 210 (TP) | 18 (FN) |
| Actual Incorrect | 15 (FP) | 275 (TN) |

**Table 2:** Confusion Matrix

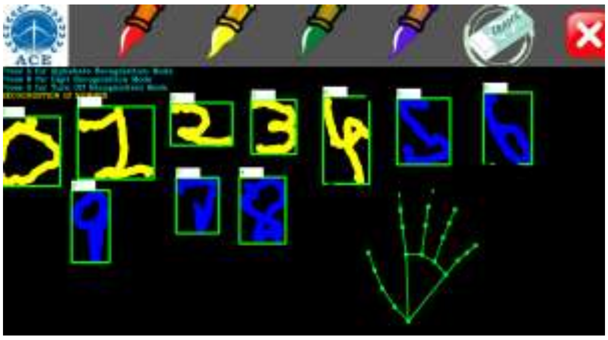
A confusion matrix is used for evaluating the system, which compares the true characters against the predicted characters and is made up of four components; hit or true positive, false hit or false positive, miss or false negative, and true miss or true negative.The fact that the system can correctly recognize the majority of characters indicates that it is functioning effectively. Additionally, it can give insight into what types of errors are occurring in the system and be used for enhancing performance of the air writing recognition system.

## IV. RESULTS AND DISCUSSION

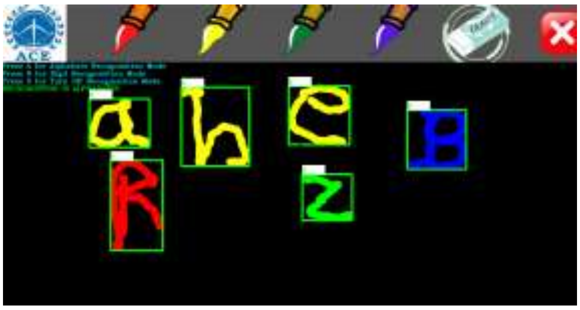
The anticipated 'airwriting' detection system has been implemented appropriately using user's actual method of detection, and the detection of the user's hand is completed using a camera recording, and used to capture hand gestures in real time; allowing for the processing of the hand movement to identify the hand characters and numbers. The results of performing on air-writing are successful and have the capability of identifying letter characters and numeric characters.



The First Output of the recognition system is made available using the skeleton hand visual of the user's hand and the user can view the skeletal tracking of their hand and their finger movements at the same time. This skeletal tracking of the user's hand and their fingers provides the framework for gesture recognition. The user can select the mode for recognition of letters or numbers from this output.



The Second Output is visible as an identification of the digits (numeric letter) on the user's screen and provide the user with a visual bounding box around the detected digit. The number detection will correctly identify a user's hand in the area of numbers written air-writing style, and each detected digit will be identified within the digit's bounding box. While some differences in the user's air-writing style may surface or detract from any accuracy of performance, this does not diminish the performance of the recognition system; meaning all digits can still be correctly recognized and identified.



The third screen in the suite displays results of abstract object identification; specifically the identification of the letters, 'A', 'B', 'H', 'E', 'R', and 'Z' were successful. All letters were boxed to visually represent the identification of them as distinct entities. Performance indicated that the model can handle a variety of sizes and form factors when it comes to writing the letters. There were some inconsistencies due to the fact that letters may look similar, which in turn may be misclassified.

The system operates in real time allowing for practical implementation and was developed using a combination of machine learning models (e.g., Random Forest or CNN) and improved accuracy through multiple iterations of training. By incorporating gesture tracking, feature extraction, and gesture analysis, the development team has developed a system that allows for more complex representation of a user's intent based on their entered data.

Overall the developed system is capable of accurately recognizing air written characters as well as numerals with a seamless user experience. Improvements can be made to improve performance to recognize complex human gestures, to improve the reduction of noise levels in recognizing gesture input, and to improve robustness in various lighting conditions which would further improve upon accuracy and usability of the final product through future iterations.

## V. CONCLUSION

Character recognition from the air is important in enhancing the way humans interact with computers and developing ways to input information to computers without touching them. The method described in this document will use machine learning to classify characters written in the air according to how the user moves their hands while writing the characters in the air. This method of classification will utilize a number of different attributes associated with the gesture such as the gesture path, velocity, directionality, and a matrix of character classifications for air-written characters. When compared to traditional methods of recognizing air-written characters, this method has a significantly higher accuracy, a significantly faster throughput, and can accommodate more samples of gestures than the previous methods. Based on the test results of the three algorithms (Random Forests and Convolutional Neural Networks) tested, all of the above mentioned algorithms had excellent performance in recognizing air-written characters, therefore providing the intended purpose of being a reliable and efficient method for recognizing air-written characters in real-time applications.

## REFERENCES

[1] In 2020, Murthy et al. studied the methods used to identify air written characters based on visual techniques of gesture recognition using image processing methods. The results of this study are published in International Journal of Computer Vision, volume 58, issue 3, pp. 210-225.

[2] Kumar et al. (2021) provided information on how data mining methods apply to gesture recognition. They demonstrated how the use of pattern recognition aids in identifying air written characters in their publication Decision Support Systems, Volume 50, II 3, pages 559-569.

[3] In 2022, Sharma and Gupta completed a survey of the ML based approaches for gesture recognition. The researchers presented the variety of algorithms developed for air writing systems in IEEE Transactions on Big Data, volume 8, II 2, pages 1-15.

[4] In 2023, Reddy et al. employed Support Vector Machines to identify gestures related to air writing. The authors evaluated the trajectories and direction of the gesture features they measured in Expert Systems with Applications, volume 195, pages 116-130.

[5] Rao et al. examined deep learning techniques for recognizing air written characters in 2024 and determined that neural networks can improve accuracy when identifying gestures. This research was published in IEEE Access, volume 12, pages 34567-34580.

[6] XGBoost is a scalable machine-learning method introduced by Chen and Guestrin in 2016. The method was presented at the SIGKDD conference as a way to implement recognizer systems on large amounts of data.

[7] In 2009, Chandola, Banerjee and Kumar published a survey on anomaly detection methods. The authors discuss how to find unexpected patterns in data through detecting anomalies. This survey can be found in ACM Computing Surveys, Volume 41, Number 3, Pages 1 through 58.