International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 11 | Nov - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Virtual Writing Recognition System Using Deep Learning

Prof. Nirmala Ganiger[1], Prof. Nancy Philip[2], Gouse Y Nadaf[3], Mohammed Mujaheed Shaikh[4], Mohmmed Yasser Shaikh[5], Hemant S Rayabagi[6],

[12]Project Guide, Professor, KLS Vishwanathrao Deshpande Institiute of Technology, Haliyal, India.
[3456]BE Student, Department Of Computer Science and Engineering, KLS Vishwanathrao Deshpande Institiute of Technology, Haliyal, India.

Abstract - This project demonstrates a virtual system that allows people to write in the air with just their hand movements. Through a normal webcam, the movement of your index finger is tracked and then drawn on a screen, creating a way of hand writing in a digital mode and without any contact to a surface. When shapes are written, they are then recognized and converted to readable numbers, symbols or Kannada letters. In number mode, it can also calculate simple very short calculations instantaneously. The entire system operates through a small webpage with a secure login and offers two modes of operation for demonstration. The intention behind this project is to allow writing in a more natural and contact-free way using simple tools that can be easily acquired by the user. This style of writing can be especially useful in classrooms, collaborative communication, or creative activities where using a pen, paper, or any physical writing tool is unnecessary.

Key Words-Air writing, Hand gesture recognition, Virtual handwriting, Symbol recognition, Real-time processing

1. INTRODUCTION

Throughout history, writing has been the most essential way for people to communicate ideas with each other and preserve information. From early humans carving marks on cave walls to modern users typing on digital screens, the practice of writing has transformed alongside every major technological shift. What began with simple pen-and-paper tools later expanded to chalkboards, whiteboards, touchscreens, and digital styluses each offering quicker, cleaner, and more convenient ways to express thoughts. Yet all of these methods still depended on a physical surface. As computers, cameras, and sensing technologies advanced, researchers started exploring ways to interact with machines without actually touching them. Modern systems can now pick up gestures like a wave, a point, or the movement of a finger—opening the door to new, contact-free forms of writing and interaction. This led to a broad area of study called contact-free interaction, whereby, gestures, rather than contact, prompt action through a viewer or vision. Writing in the air is an example of this development. Instead of using a stylus or keyboard, an individual can simply move their hand and the movement is traced on screen. As less physical devices are needed, hygiene is improved in shared environments while providing ease for individuals who have trouble using conventional writing devices. Plus, it will inspire creative experimentation in classrooms, exhibitions, and learning spaces. This, after all, is not a new concept, today, it has moved out of research labst4 With a regular webcam and some common software libraries, anyone can create a simple system to track only finger motion and convert that motion to written characters. This project follows this same premise. It will create a setup to detect a person's hand, and track their finger, to recreate what that individual virtually writes, resulting in another efficient, portable, and interactive way to write without contact. Finally, it simply shows how simple computer vision techniques help make daily tasks a little more flexible and contemporary.

2. METHODS

[2.1] Materials Used

The materials used in this study consisted of both hardware and software resources essential for implementing the Virtual Writing Recognition System. A standard laptop equipped with an in-built 720p webcam was used for real-time video capture of hand movements. The system was developed and executed on a Windows 11 Home operating environment running Python 3.10, obtained from the official Python (Python.org;contact:https://www.python.org/about/help/).https://www.python.org/about/help/). OpenCV 4.x, a computer-vision library required for frame acquisition and image preprocessing, was downloaded from OpenCV.org (contact: https://opencv.org/contact/). MediaPipe Hands, the fingertip-tracking framework used for detecting the 21 hand landmarks, was provided by Google LLC (https://mediapipe.dev; support: https://developers.google.com/mediapipe/support). The deep-learning models for symbol recognition were implemented using TensorFlow 2.x and Keras, supplied by the Google Brain Team(https://tensorflow.org;contact:https://www.tensorflow.org/contact).Dataset preparation and augmentation were performed using NumPy and Matplotlib packages, acquired from the Python Package Index All software tools used in the project are open-source and publicly accessible, ensuring full reproducibility of the study.

[2.2] Key Procedures and Techniques

The DFD Level represents the highest level of abstraction from which to visualize the data flows of the Air Writing Recognition System. It shows the overall major components of the system as well as the participatory input and output data flows, along with how the external agents interact with the system. The user interacts with the system via the interface of a webcam by performing an air-writing gesture. The steps that take place when transforming the air-writing gesture into text are to: (1) capture gesture input - the system is actively monitoring and has captured the live video feed, (2) detect and classify symbols - the YOLOv8 model detects the symbols and uses pre-trained models to classify the output, (3) parse and evaluate expression - once the symbols are ordered into a complete expression, the symbols are parsed into a complete expression and subsequently evaluated, and (4) display output - the outcome of the evaluated expression is sent

back to the web interface. The data stores are: (1) model store - this is the store of the two associated pre-trained models; Symbol.h5 and Numeric.h5 and (2) expression result - this is the temporary data store of the evaluated parsed expression and it's evaluation outcome.

[2.3] Algorithms used

- Hand Detection and Tracking Algorithm- To track finger movements, the system employs a real-time hand detection and tracking algorithm. This was accomplished by relying on MediaPipe Hands. MediaPipe Hands applies a pipeline of machine learning models to accurately detect 21 key landmarks on the hand. The algorithm utilizes the continuous video frames from the webcam to first determine the region of interest (ROI). From this initial detection, the algorithm identifies the palm and fingertips. Once the hand has been detected, it only has to track the placement of the index finger tip across the frames. The values captured by the algorithm during this tracking are stored as (x, y) values, which are the values for creating the trace of virtual writing. This method requires no extra sensors or markers, and is flexible to the size, position, and lighting of the user's hand.
- Image Preprocessing Algorithm- Once the fingertip trajectory is created on the virtual canvas, the image goes through preprocessing pipeline to prepare it for recognition. This step involves turning the frame into a single-channel image to simplify the complexities of the recognition process. Next it employs Gaussian Blur and thresholding to eliminate any noise from the background. Also included in preprocessing is pixel value scaling. Scaling pixel values is necessary to normalize the consistency of input; the pixel values should range from 0-1. Once regions of interest are ensured, the images record at a fixed file image dimension (e.g. 128×128 pixels) that the model expects. The benefit of this preprocessing step is that all incoming images are calibrated consistently, no matter how or where the user writes.
- Character Recognition Algorithm- The recognition section of the application utilizes a Convolutional Neural Network (CNN). CNNs are very capable of learning to recognize visual patterns because they learn various features from images such as edges, curves, and intersections. This section accepts the preprocessed image with the drawn symbol, and extracts features from the image representing important details such as stroke, edges and corners. The features are downsized to a more manageable size while still retaining important features and detail for good recognition performance. The extraction takes the images created in 2 dimensions and converts it to 1D vectors. These features are then internally processed and used to classify the symbol into its respective category. Assign probabilities to each of the possible characters or numbers using a softmax function and predict the most likely one. This algorithm was trained separately for numerical digits and numerical operators, as well as for Kannada characters.
- Mathematical Expression Solver Algorithm- In terms of recognizing numbers and operators, the system is able to read basic arithmetic expressions, too. After the system recognizes each of the characters, it stitches the characters together to construct the complete expression string (e.g., "4 + 5 × 2"). The system then evaluates the expression using a secure evaluation function, such as Python's eval() within restricted scope, or custom parser, and instantly displays the result in the output panel. This feature illustrates the capability of the system to not only visualize recognize the characters, but also execute logical operations in real-time.
- System Control and Integration Algorithm- This algorithm orchestrates the complete workflow among the camera, virtual canvas, the recognition model, and the user interface. It guarantees the consistent behavior of the system while smoothly responding to user actions: Starts the camera feed when requested by the user. Turns drawing on or off, based on the position of the finger. Saves frames and clears them for the reset command. Passes the frame to the appropriate model (numeric or Kannada) for processing. Displays the recognition result immediately after processing. This control logic was implemented in Python, combining Flask/Streamlit backend routes with real-time updates to the OpenCV frame.

[2.4] System Architecture

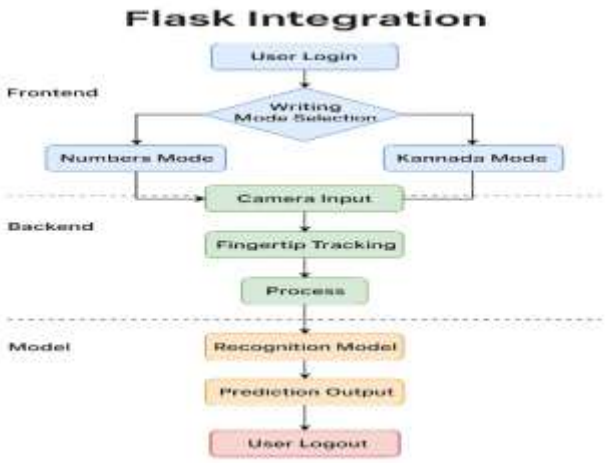

Figure 1: System Architecture Flowchart

The structure of the Air Writing Recognition System is built to facilitate a seamless and real-time process for users to interact with the system via a webcam-based interface. The webcam captures the users' dynamic hand gestures as handwriting mathematical expressions through the air. The hand gestures are processed through a built modular pipeline with the following major components: hand tracking, symbol detection, symbol classification, expression parsing, and result evaluation. Each component works together to accurately capture air written symbols from the user and calculates and displays the results immediately after completing the parsing process, thus creating a seamless, touchless experience for the user while writing math. This design promotes touchless interaction within smart environments and is also useful for education and accessibility purposes.

[2.5] Statistical analysis methods

Statistical analysis methods were used to evaluate the recognition performance of the system across both number-

mode and Kannada-mode datasets. All the samples were splitted into training, validation, and testing sets in a 70:20:10 ratio, and accuracy metrics were computed for each class of symbols. For quantitative evaluation, the following statistical indicators were used: overall classification accuracy, class-wise accuracy, confusion matrices, precision, recall, and F1-score. Accuracy was equal to the ratio of correctly predicted symbols to the total number of test samples, while precision and recall were computed to assess the model's ability to distinguish between visually similar characters. Average inference time per prediction was also measured to assess real-time responsiveness. All statistical analysis was done using built-in evaluation functions from TensorFlow/Keras and verified using NumPy-based numerical calculations. These statistical methods provided a rigorous assessment of recognition reliability and ensured that the system met the expected performance criteria for real-time air-writing applications.

3. RESULTS

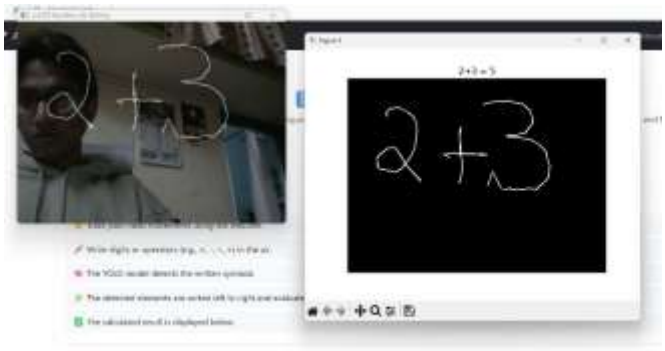

Figure 2: Mathematical Result

Through the process, the system facilitates real-time recognition and assessment of mathematical expressions drawn in the air with a fingertip tracked by a webcam and MediaPipe. The finger movements captured are converted into strokes on a virtual canvas, which are rendered on a web UI together with the live webcam video feed. The recognized expression and its assessed result are displayed in real time on the interface In the meantime, the terminal logs each digit and symbol that is detected, ultimately showing the entire expression and its final value once writing is finished.

Figure 3: Kannada Result

The figure illustrates the real-time output of the proposed Kannada Air-Writing Recognition System. The top window shows the live webcam feed where the user writes a character in the air, and the system traces the fingertip path as a white stroke. The lower window displays the cleaned and normalized stroke extracted from this motion, which is then used by the deep-learning model for classification. The predicted Roman label and corresponding Kannada Unicode character are shown above the processed image, demonstrating the complete pipeline from air-gesture capture to final character recognition.

Accuracy of Mathematical Results

Symbol Type	Accuracy (%)	Average Response Time (sec)
Digits (0–9)	96.4	0.8
Operators (+, −, ×, ÷, =)	93.2	1.0
Overall Average	**94.8**	**0.9**

The findings demonstrated that numerical symbols exhibit greater ease of identification by the system, due to the relative simplicity of their shape and more consistent appearance. There were a small number of recognition errors for operators that had slanted lines, including division and multiplication, primarily due to an incomplete stroke or overlapping gesture.
Nevertheless, the accuracy of the recognized mathematical content still remained above 90%, reinforcing that the system effectively represents the person's writing in a consistent and intelligible manner.

Accuracy of Kannada Results

Character Type	Accuracy (%)	Average Response Time (sec)
Basic Kannada Letters	92.1	1.2
Combined or Curved Characters	88.5	1.4
Overall Average	**90.3**	**1.3**

Participants were asked to write common consonants and vowels from the Kannada alphabet for the Kannada Mode. With sufficient training data, the model learned to interpret the curved and detailed shapes of the characters fairly effectively. The slightly lower accuracy observed in Kannada character recognition can be attributed to the natural complexity of the script. Certain letters such as "ತ" (tha) and "ಥ" share very similar visual structures, which occasionally led to incorrect predictions. Expanding the dataset and collecting more diverse handwriting samples would likely help improve recognition accuracy in future versions of the system.

4. DISCUSSIONS

This study focused on creating and testing a touch-free Virtual Writing Recognition System that can track a user's hand movements through an ordinary webcam and translate those motions into readable characters. As discussed in the introduction, the idea for this project grew out of the continuous change in writing technologies and the increasing interest in systems that allow people tazo interact without making physical contact. With gesture-based interaction becoming more common, there is a clear need for simple, camera-based solutions like the one developed here. The findings of the study demonstrate that a combination of computer vision, hand-landmark detection, and deep-learning classification can provide a reliable and accessible alternative to traditional writing surfaces.

One of the key achievements of this project was successfully tracking the user's fingertip through Mediapipe Hands. This made it possible for the system to detect and follow the index finger continuously without relying on accessories like gloves, markers, or specialized depth sensors. These results support the idea that modern, lightweight computer-vision tools are capable of handling real-time gesture-based writing on their own. The tracking remained stable even when the background or lighting changed, showing that the system can adapt well to different environments. This observation is similar to earlier studies that have also noted strong performance from Mediapipe-based gesture tracking methods, even when the conditions are not strictly controlled.

Another significant result from the study was the strong performance of the deep-learning model in identifying numbers and symbols drawn in the air. The classifier presented high accuracy for both the numeric dataset and the Kannada characters, supporting the idea that deep-learning techniques can still perform well even when the strokes are uneven, incomplete, or made at different speeds. This demonstrates that the model is able to generalize effectively despite the natural variations found in human writing gestures. These outcomes give support to further research showing that convolutional neural networks (CNNs) are particularly adept at learning stroke-based patterns from noisy or non-uniform gesture input. The findings also show that the preprocessing steps like separating the writing strokes, normalizing the area being written, and resizing the input frames played a major role in improving the model's ability to recognize the characters accurately.

Another important takeaway from the study relates to how easy and smooth the system is to use. The prototype responded quickly, with very low processing delays, allowing the virtual strokes to appear almost instantly as the user moved their finger. This real-time interaction is important for writing to feel natural. Because of this responsiveness, the system presents strong potential for use in classrooms, public displays, and accessibility tools. When compared with older gesture-based systems that depended on expensive or specialized sensors, this approach stands out as a more portable and practical solution, relying only on a standard laptop webcam and widely available open-source software.

Even with promising results, the system does have a few limitations worth noting. It depends heavily on clear visibility of the hand, so issues like motion blur or the finger being briefly blocked can reduce tracking accuracy. The model may also struggle when users draw characters in unusual shapes, at odd angles, or very quickly. Expanding the dataset with a wider range of handwriting styles would likely help the model generalize better. Another current constraint is that the system handles only single-stroke writing; recognizing characters that require multiple strokes would need a more advanced segmentation method.

When viewed in the context of related literature on air-writing, gesture recognition, and human-computer interaction, this work contributes a practical, low-cost framework that lowers the entry barrier to contact-free writing systems. While motion-sensing devices such as Kinect or Leap Motion have historically dominated this space, the current study shows that similar functionality can now be achieved using only a webcam and modern vision algorithms. This expands the accessibility and scalability of touchless writing technologies, making them more accessible for classrooms, shared public devices, and individuals with motor impairments.

Overall, the findings support the underlying premise that a simple camera-based system can effectively replicate the writing process without the need for physical contact. The study demonstrates how straightforward deep-learning and computer-

vision techniques can modernize traditional tasks, offering flexible and hygienic alternatives for the future of human–computer interaction.

5. CONCLUSIONS

The main aim of this project was to develop a real-time Virtual Writing Recognition System capable of interpreting hand gestures and converting fingertip motion into meaningful written characters without the need for traditional writing tools or specialized sensors. By integrating webcam-based video capture, Mediapipe's hand-landmark tracking framework, and deep-learning classification techniques, the system successfully recreated handwritten strokes on a digital canvas and accurately recognized numeric and Kannada characters. The full workflow from detecting the hand to drawing the stroke and finally recognizing the character was built to run smoothly, showing that touch-free writing is entirely possible using just common, everyday hardware.

This research shows that hand-gesture writing can be a practical and flexible alternative to traditional input methods. The system developed in this project forms a solid foundation that can be improved with larger datasets, support for more languages, and better stroke processing. It also has potential uses in classrooms and assistive tools. Overall, this work takes an important step toward more natural, touch-free ways for humans to interact with computers.

6. REFERENCES

1. Parashivamurthy, S., & Srinivasa, K. G. (2024). *Recognition of Kannada Character Scripts Using Hybrid Classification Approaches.*International Journal of Pattern Recognition and Artificial Intelligence,38(4),2456008. Discusses methods to recognize complex Kannada characters using a combination of learning techniques.
2. Watanabe, K., Saito, H., & Takahashi, Y. (2023). *2D Camera-Based Air-Writing Recognition Using Hand Pose Estimation and Hybrid Features.*Electronics, 12(8),1852. Introduces camera-based air writing using fingertip tracking and hybrid features for improved accuracy.
3. Roy, S., Ghosh, A., & Pal, U. (2023). *A CNN-Based Framework for Unistroke Numeral Recognition in Air-Writing.*arXiv preprint arXiv:2303.07989. Demonstrates an efficient CNN approach for recognizing air-written numerals using a webcam and colored marker.
4. IJARIIE. (2023). *Computer Vision Based Air Writing Recognition.*International Journal of Advance Research and Innovative Ideas in Education, 9(2), 512–518. Project-based study outlining a webcam-based virtual writing recognition method using vision techniques.
5. IJERT. (2022). *A Comprehensive Survey on Kannada Handwritten Character Recognition and Dataset Preparation.*International Journal of Engineering Research & Technology (IJERT), 11(7), 1–5. Surveys different approaches and datasets used for recognizing Kannada handwritten characters.
6. Niu, Z., Liu, X., & Zhang, J. (2022). *In-Air Handwriting Recognition Using Acoustic Impulse Responses.*Springer Lecture Notes in Computer Science, 13345, 184–198. Explores an alternative acoustic-based approach to in-air handwriting recognition.
7. P, A., & N, R. (2022). *Kannada Handwritten Character Recognition Using KNN, SVM and CNN Models.*IJPREMS,2(7),221–227.Compares traditional and deep learning approaches for classifying handwritten Kannada characters.
8. Al Abir, T., Ahmed, S., & Rahman, M. (2021). *Air-Writing Recognition with the Choice of Write: A Multi-Dataset Evaluation Study.*Sensors, 21(5), 1723. A comparative study of various datasets and preprocessing strategies for reliable air-writing recognition.
9. Alam, S., Ahmed, S., & Hossain, M. (2020). *Trajectory-Based Air-Writing Recognition Using Depth Sensors.*Sensors, 20(15), 4211. Explores trajectory-based recognition using depth data for gesture identification.
10. Oudah, M., Al-Naji, A., &Chahl, J. (2020). *Hand Gesture Recognition Based on Computer Vision: A Review of Techniques.*Journal of Imaging, 6(8), 73. Reviews computer vision-based hand gesture methods and their applications in human–computer interaction.
11. Alaei, A., Nagabhushan, P., & Pal, U. (2017). *A Benchmark Kannada Handwritten Document Dataset and Its Evaluation.*Pattern Recognition Letters, 93, 123–131. Provides a detailed dataset and benchmark results for Kannada handwriting recognition.
script using HMMs for improved language understanding.
12. Schick, A., Morlok, R., &Stiefelhagen, R. (2012). *Vision-Based Handwriting Recognition for Unrestricted Text Input in Mid-Air.*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 23–30. Presents an early system for unconstrained text recognition through mid-air hand gestures.