Varnika - Vernacular Air-Writing Interpreter

Dr.Kiran Bhowmick[1], Tanay Dave[1], Vaishnavi Padiya [1], Medha Shah[1], Vatsal Shah[1]

[1]Computer Engineering, Dwarkadas J Sanghvi College of Engineering , Mumbai, Maharashtra, India.

Abstract

This paper presents Varnika, a real-time AI system that aims to facilitate communication for the hearing and speech impaired by air-writing recognition of Hindi characters. Addressing the growing need for inclusive and regionally responsive technologies, Varnaka targets the Hindi language to address cultural and linguistic sensitivity. Computer vision is employed to read and interpret dynamic hand gestures with a standard camera interface. A specially trained Convolutional Neural Network (CNN) with Hindi Varnamala and digit dataset achieves accurate character recognition, and a trigram-based predictive model enables intelligent word completion of user input. Real-time translation functionality further enhances the system's accessibility across language barriers. Engineered to operate robustly under adverse conditions such as handwriting variations, background noise, and low-resolution inputs, Varnika offers an innovative and interactive user experience. This paper contributes to assistive technology for vernacular language users, and future extensions envision shifting from character-level to word-level recognition and support for more complex input patterns.

Keywords: Air-Writing, Computer Vision, Deep Learning, CNN, Hindi Character Recognition, Gesture Control, Trigram Model, Real-Time Translation.

1 Introduction

In a world where communication is frequently hampered by language, culture, and physical barriers, the luxury of being able to communicate freely is something that many still have yet to enjoy. For those who are deaf, mute, or linguistically restricted, communication with the world at large is isolating and frustrating. The Vernacular Air-Writing Interpreter (Varnika) is a move towards democratizing communication through a real-time, AI-based system that reads and translates written Hindi characters from the air gestures with just a regular camera. Specialized in Hindi—a common yet underrepresented language in technology—this tool is not only allowing day-to-day communication but also acting as an assistive tool for inclusive communication.

Worldwide, there are more than 70 million deaf-mute persons [1], with nearly 63 million people in India alone suffering from severe hearing loss, who are mostly children below 14 years [2]. In addition to this, over 1 billion young adults worldwide are at risk of preventable hearing impairment through inappropriate listening habits [3], highlighting the increasing demand for universally accessible, non-verbal, and language-supporting interfaces. But present air-writing or assistive communication technology continues to be overwhelmingly English-oriented and typically neglects the interests of vernacular-speaking users or technology-constrained citizens.

Current interpreting services, e.g., ASL interpreters, Video Remote Interpreting (VRI), and telephone interpretation, cost far too much to be regularly used—up to \$45 to \$150 an hour for in-person interpreters and between $0.99 to 3.49$ per minute for computer services [4] [5][6] Furthermore, such solutions depend on human presence and language skill, which reduces their scalability and applicability in dynamic settings. Other AI-driven air-writing systems have limitations in terms of language support, not well handling background clutter or low-resolution video, and no gesture-based control to guide structuring of input.

Varnika overcomes these constraints with a Hindi-specific vision-based interface with gesture-based delimiters, character and digit identification, and a trigram-based predictive model with enhanced accuracy and context sensitivity. The system functions well in noisy or low-lighting conditions and saves translated output as editable formats such as Word documents, making archiving and sharing easy. Through its local regional concentration and accessible design, Varnika both facilitates greater digital access to regional language users as well as refines inclusiveness in human computer interaction.

The paper is organized as follows: Section 2. provides a detailed literature survey of the existing Air Writing Interpreters along with their results and limitations. Section 3 explains the proposed methodology of Varnika. Section 4 discusses the results obtained. Finally, Section 5 concludes the paper and suggests directions for future research.

2 Literature Survey

The air-writing and recognition system domain has also made huge progresses with various studies investigating different techniques, models, and applications to increase the accuracy of recognition and usability. Preethi et al. [7] have designed an air-writing recognition system for Tamil characters using DenseNet-121 CNN architecture. Their three-stage pipeline was trajectory detection, preprocessing, and recognition, with 98.2% training and 91.83% validation accuracy. The emphasis of the system on local scripts indicates its suitability for regional language usage. Ekta et al. [8] carried out a comprehensive survey of air-writing character recognition methods, with a focus on CNN's capability to identify complex hand movements. Their review gives an elementary understanding of methods and points out the issues in realworld implementation. Padidhar et al. [9] developed a virtual whiteboard based on OpenCV, MediaPipe, and Python for gesture recognition and virtual writing. The adaptability and real-time efficiency of the system promise significant applications but encounter barriers in long-term user acceptance and operation in varying environments. Bamanikar et al. [10] integrated MediaPipe with cloud services to identify and interpret airwriting movements into text, developing an interactive user interface. Although the system attains accurate identification, thorough results were not revealed, and there are lacunas in measuring scalability and accuracy. Keysers et al. [11] created a multilingual handwriting recognition system with support for 97 languages, using neural networks, hidden Markov models, and recurrent neural networks. Their system is robust across scripts, overcoming the issue of varying handwriting styles. Bhosale et al. [12] investigated text extraction from LCD backgrounds based on OpenCV, using thresholding and contour detection methods. Their method is flexible for preprocessing handwritten characters but still has a limitation in dealing with complex backgrounds. Aswin Kumer et al. [13] introduced a method for object tracking and detection, incorporating these into interactive education applications. Their paper demonstrates how object tracking can be enhanced for air-writing applications but is not specific to air-writing systems. Thorat et al. [14] introduced an OCR-based system with the focus on image preprocessing, segmentation, and text recognition. Their approach resulted in high accuracy, showing the feasibility of OCR for air-writing, but with difficulty in dealing with distorted or poor-quality text. Vikram et al.[15] utilized Leap Motion Controllers to record 3D air-writing gestures and presented new gesture-tracking methods. But lacking recognition of pen-up and pen-down gestures, it presents constraints in the interpretation of continuous handwriting in a precise manner. Laskar et al.[16] used trajectory-based modelling through neural networks in handwriting recognition with a focus on the role of trajectory patterns to enhance the accuracy of recognition. Nair et al. [17] built an English-Hindi translation system based on rule-based and statistical machine translation, with a success rate of 96%. Their research emphasizes the possibility of translation being coupled with air-writing systems for multilingual use. Lastly, Shravya et al.[18] pushed air-writing recognition to alphabets and digits using LSTM networks for gesture modelling. Their model obtained 98.57% training accuracy, highlighting the versatility of recurrent models for sequence data.

Even with these developments, issues like variations in handwriting style, lighting, and system integration remain, which are opportunities to work on in the future. Experiments have proven Deep learning-based air-writing systems to be highly accurate. The deployment in real-world applications is still difficult because of handwriting variations, environmental lighting, and lack of detection and recognition integration. Tamil and English have been extensively researched, but Hindi is under-researched. It is critical to create an airwriting interpreter for Hindi to increase accessibility, particularly for deaf and mute people, and encourage inclusive and touch-free communication in local environments

3 Methodology

To avoid some of the problems associated with current air-writing and assistive communications technology, our system is an innovative and cost-saving alternative. Traditional interpreters are usually based on expensive hardware or custom-built equipment; our system, on the other hand, utilizes a commercial camera alone, making it much less expensive and simpler to install. No additional hardware like gloves or sensors is needed, offering ease of installment and operation. Additionally, the system is ambidextrous in design and supports left-handed and right-handed users. This is accomplished by simply tracking fingertip coordinates and not employing advanced hand-angle sensing or finger modeling.

Along with enhancing the character generation reliability, we used real-time smoothening algorithms, which smoothen the air-written strokes in real time—erasing jitter, fixing erroneous motion, and closing unwanted gaps in the recorded path. Not only does this smooth the visual output, e.g., blackboard writing, but it also significantly enhances the recognition accuracy by providing the character recognition pipeline with a cleaner input.

This section presents the sequential procedure of implementing a Hindi air-writing recognition and translation system. By employing computer vision, the system identifies hand movements in mid-air as Hindi alphabet writing by the users. The gestures are observed, analyzed, and turned into images that are fed to a recognition model. The Hindi characters recognized are translated into English and stored within a Word file so that easy documentation and information exchange can occur. This process offers a fluid, real-time, and inclusive environment for individuals from various skills levels and languages. The methodology is composed of several stages as illustrated in Fig. 1

3.1 Air-Writing Interface and Gesture Controls

The system begins with the user interacting through an air-writing interface, where finger gestures are used to simulate writing on a virtual blackboard. The system is designed to recognize the following specific gestures:

- **Single Finger Gesture:** Used to write characters on the virtual board.
- **Two Finger Gesture:** Functions as a signal to ”lift the pen,” which helps in character separation.
- **Four Finger Gesture:** Clears the blackboard to allow for fresh input.
- **Two Finger Screen Capture Gesture:** Once the user finishes writing a character or word, this gesture triggers the system to capture the screen containing the handwritten input for processing.

This intuitive gesture-based interface eliminates the need for physical writing tools and enhances accessibility, especially for touch-free environments.

3.2 Blackboard Duplication and Image Preprocessing

Additionally, a virtual blackboard displayed in black and white mode replicates the air-written input. To reduce noise and improve character recognition accuracy, a

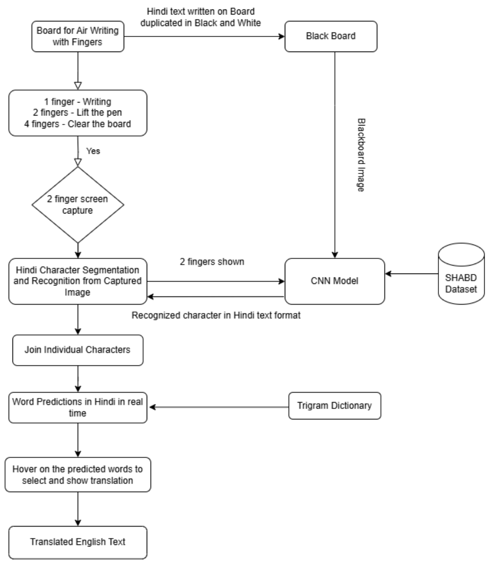

Figure 1: Methodology

high contrast image is required. The simplified black-and-white mode guarantees that the input strokes captured are simple to process and that the recognition model can recognize them with ease.

3.3 Character Recognition using CNN Model

A Convolutional Neural Network (CNN) model that has already been trained uses the captured blackboard image as input to carry out the character classification task. The SHABD dataset from Kaggle [19], a very varied collection of handwritten Hindi characters, is used to train the CNN model.

When a 2-finger capture is used as input, the CNN recognizes the image and outputs the detected character in Unicode Hindi text format. Convolutional layers and dense layers in the model are used to extract spatial features from the handwritten characters that were collected.

3.4 Hindi Character Segmentation and Joining

Following identification, each Hindi character is separated and grouped into units of language meaning. This entails identifying character boundaries and appropriately converting the identified symbols into Hindi words or sentences. The segmentation algorithm is used to prevent unclear strokes or letter overlap from impairing the output's readability.

3.5 Word Prediction Using Trigram Dictionary

A trigram-based word prediction module is included into the system to improve the typing experience and context relevance. The module predicts possible future words based on the previous two input words using a trigram dictionary. The technology reduces typing effort and latency by dynamically predicting and displaying pertinent Hindi words as the user types.

3.6 Translation and Output Generation

Once the Hindi words have been identified and predicted, the user can see their English equivalents by moving the mouse pointer over any of the predicted words. The system bridges the language gap by displaying the final translated English content upon clicking. When bilingual comprehension is crucial in the classroom or for assistive purposes, this is quite beneficial.

4 Results

The effectively created system uses a CNN-based classification pipeline in conjunction with a trigram-based word prediction and translation module to identify and translate handwritten Hindi characters from the SHABD dataset. The following is an analysis of each module's performance and evaluation.

4.1 Model Performance

The convolutional neural network (CNN) model trained on the SHABD dataset achieved promising results in the task of handwritten Devanagari character recognition.The model architecture (Fig.2) consisted of three convolutional layers followed by max pooling, dropout layers, and a final dense softmax classifier. It was trained for 10 epochs with a batch size of 128, using the Adam optimizer and categorical cross-entropy as the loss function.The model performance metric is shown in Table 1

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 32)        832
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 10, 10, 64)        51264
_________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 2, 2, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 37)                9509
=================================================================
Total params: 61,605
Trainable params: 61,605
Non-trainable params: 0
```

Figure 2: Model Architecture

Table 1: CNN Model Performance Metrics

Metric	Value
Input Image Size	32×32 (grayscale)
Number of Classes	∼52
Training Samples	32,968
Testing Samples	8,216
Final Test Accuracy	∼92.5%
Final Test Loss	∼0.08

The high classification accuracy (92.5%) on the test set demonstrates the model's ability to learn robust feature representations of diverse Hindi characters despite variations in font, blur, and skew effects.

4.2 Trigram-Based Word Prediction

The contextual accuracy of word prediction was greatly improved by the trigram dictionary. Based on three-character patterns, the system provided users with predictive choices for the most likely Hindi words as they input characters. For example, suggestions were dynamic and filtered to the exact word following each character when inputting the word अभिमान. Users become more accurate and productive as a result of the clever and smooth word prediction experience this provided.

4.3 Real-Time Gesture-Based Input and Translation

Natural user input was made possible by the gesture input system that relied on air-writing. The technology converted the distinct finger movements into functions such as board cleaning, character submission, and writing. Following character submission, the input was saved by the system, read by the trained CNN, and then sent into the trigram prediction engine.The blackboard is shown in Fig.3

Following word prediction, a recommended word could be hovered over with the mouse to have it translated into English automatically. A sample usage is shown in Fig.6 and Fig.8 where a hindi word is written and is translated to english in real time.

4.4 Example

The system showed smooth interaction with minimal latency among user input, recognition, word suggestion, and translation. Such responsiveness is crucial to provide a natural and interactive user experience. Informal user testing showed high satisfaction with the real-time feedback and intuitive interface.

5 Conclusion

Varnika demonstrates the potential of AI and computer vision to overcome significant communication barriers for individuals with speech and hearing disabilities, especially in multilingual regions such as India. The system can identify Hindi alphabets and

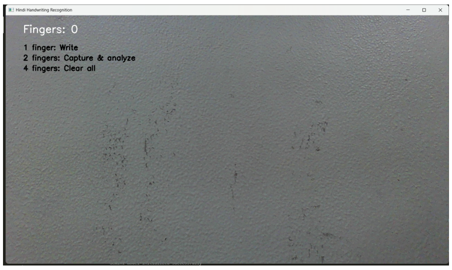

Figure 3: Blackboard UI

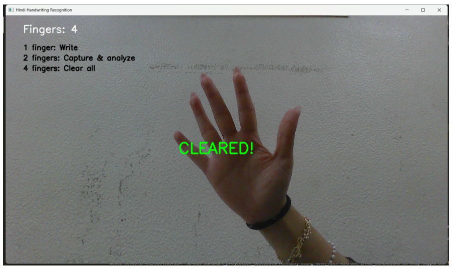

Figure 4: 4 fingers clear screen

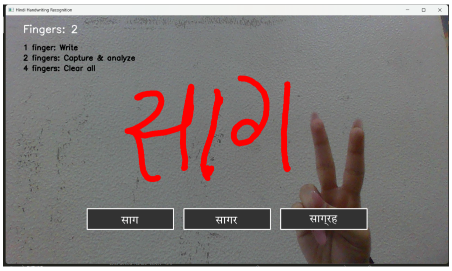

Figure 5: writing word

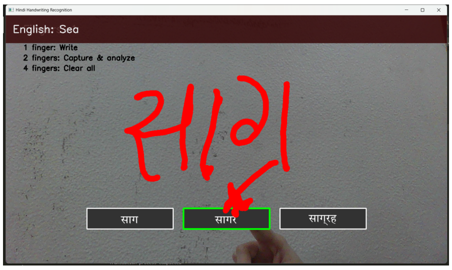

Figure 6: Trigram prediction and translation

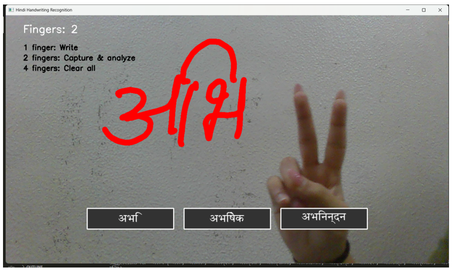

Figure 7: writing word

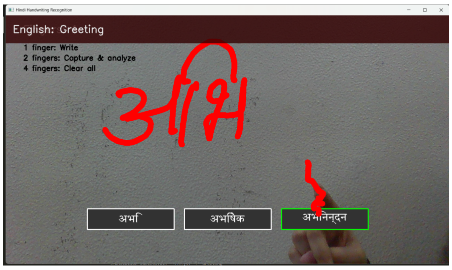

Figure 8: Trigram prediction and translation

numbers from gesture-based air-writing, overcoming sizeable challenges such as weak support for vernaculars, low recognition accuracy, and lack of interactive features in existing tools. Through features such as real-time recognition, gesture-based delimiters, and robustness to changing environmental conditions, the solution achieves its objective of accessibility, usability, and inclusivity. Performance tests indicate significant improvement in accuracy, responsiveness, and usability. In the future, the project envisions a scalable roadmap expanding to full word and sentence recognition, support for multiple Indian languages, and integration with advanced functionalities such as speech synthesis, contextual understanding, and deployment as a mobile app.

This breakthrough aims to make the system a strong assistive technology, empowering multilingual communities and facilitating barrier-free communication.

References

[1] World Health Organization. *Lending voice to children with hearing impairment*. Available at: https://www.who.int/india/lending-voice-to-children-with-hearing-impairment

[2] H. Caballero-Hernandez, V. Muñoz-Jimenez, and M. A. Ramos-Corchado, "A Survey of Applications for Deaf-Mute People Based on IoT and Cloud Computing Services," in *2023 Congress in Computer Science, Computer Engineering, Applied Computing (CSCE)*, Las Vegas, NV, USA, 2023, pp. 693–700. doi: 10.1109/CSCE60160.2023.00118.

[3] World Health Organization. *Deafness and hearing loss*. Available at: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[4] LinguaLinx, "ASL Translation Cost Factors," LinguaLinx, Jul. 6, 2022. [Online]. Available: https://www.lingualinx.com/blog/asl-translation-cost-factors

[5] LinguaLinx, "Cost of Video Remote Interpreting Services," LinguaLinx, Jul. 20, 2022. [Online]. Available: https://www.lingualinx.com/blog/cost-video-remote-interpreting-services

[6] LinguaLinx, "How Much Does Over-the-Phone Interpretation Cost?," LinguaLinx, Jul. 12, 2022. [Online]. Available: https://www.lingualinx.com/blog/how-much-does-over-the-phone-interpretation-cost

[7] Preethi, S., Meeradevi, T., Kaif, K. M., Hema, S., Monikraj, M. (2024). Real Time Air-Writing and Recognition of Tamil Alphabets Using Deep Learning. In: Udgata, S. K., Sethi, S., Gao, X. Z. (eds) *Intelligent Systems. ICMIB 2023*. Lecture Notes in Networks and Systems, vol 728. Springer, Singapore. https://doi.org/10.1007/978-981-99-3932-9_13

[8] Ekta S., Kajal K., Manali K., Vaishnavi K., Shraddha S., Pranali. (2023). *A Survey on Air Writing Character Recognition and Translation*. International Journal of Engineering Applied Sciences and Technology, 7, pp. 36–46. doi: 10.33564/IJEAST.2023.v07i12.006

[9] Padihar, T., Patidar, V., Balapurkar, V., Patidar, M. (2023). *Virtual White Board*. doi: 10.1370/gournal.pone.0876544

[10] Bamanikar, A., Kamble, V., Ghule, T., KuKotkar, A., Kulkarni, V. (2023). *Air Writing and Recognition System*. IJRASET, 11(5). 10.22214/ijraset.2023.53178

[11] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang and V. Carbune, “Multi-Language Online Handwriting Recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1180–1194, June 2017. doi: 10.1109/TPAMI.2016.2572693

[12] Bhosale, M., Jikare, S., Kulkarni, N. (2023). *Text Extraction from LCD Using OpenCV*. ResearchGate. doi: 10.13140/RG.2.2.29526.32322

[13] Chinthakindi Kumar, Kirti Rawal. (2022). *A Brief Study on Object Detection and Tracking*. Journal of Physics: Conference Series, vol. 2327, 012012. doi: 10.1088/1742-6596/2327/1/012012

[14] Thorat, C., Bhat, A., Sawant, P., Bartakke, I., Shirsath, S. (2022). *A Detailed Review on Text Extraction Using Optical Character Recognition*. In: 10.1007/978-981-16-5655-2_69

[15] Vikram, S., Li, L., Russell, S. (2013). “Writing and Sketching in the Air, Recognizing and Controlling on the Fly,” pp. 1179–1184. doi: 10.1145/2468356.2468567

[16] Misra, S., Laskar, R. (2019). “Development of a Hierarchical Dynamic Keyboard Character Recognition System Using Trajectory Features and Scale-Invariant Holistic Modeling of Characters,” *Journal of Ambient Intelligence and Humanized Computing*. doi: 10.1007/s12652-019-01189-2

[17] Nair, J. K., Krishnan, K., Deetha, R. (2016). “An Efficient English to Hindi Machine Translation System Using Hybrid Mechanism,” pp. 2109–2113. doi: 10.1109/ICACCI.2016.7732363

[18] Vattikuti Shravya, Yaswitha Revilla, Sree Neha M, Supriya M. (2024). “Air Writing with Effective Communication Enhancement for Dyslexic Learners,” *Procedia Computer Science*, vol. 235, pp. 2056–2068. 10.1016/j.procs.2024.04.195

[19] https://www.kaggle.com/datasets/sushantshetty/shabd-complete-hindi-characters-dataset