Journal of Computer Science

Original Research Paper

An Air-Written Real-Time Multilingual Numeral String Recognition System Using Deep Convolutional Neural Networks

[1]Meenal Jabde, [1]Chandrashekhar Himmatrao Patil, [2]Amol D. Vibhute and [1]Shankar Mali

[1]*Department of Computer Science, Dr. Vishwanath Karad MIT World Peace University, Pune, MH, India*
[2]*Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, MH, India*

Article history
Received: 13-07-2024
Revised: 12-08-2024
Accepted: 27-08-2024

Corresponding Author:
Chandrashekhar Himmatrao Patil
Department of Computer Science, Dr. Vishwanath Karad MIT World Peace University, Pune, MH, India
Email: chpatil.mca@gmail.com

Abstract: Air-writing is a modern practice for free-space writing of characters or words with hand or finger movements. In the new era, hand gestures are commonly used for Human-Computer Interaction (HCI) or controlling machines in several applications. Writing with a finger or holding any object in hand and 3-D movement in the air is helpful for several applications. However, it is hard to build simulated environments due to the complex structure of the hand and regulate the joints. Palm detection is another way to reduce the complexity of hand detection. Deep learning-based techniques enhance object detection tasks with excellent performance in this task. Convolutional Neural Networks (CNN) are the powerful frameworks used in detecting palms facing several challenges in this task. Therefore, we propose a novel approach using hand landmark detection to identify palms. Our proposed CNN-based air-writing recognition model is intended to detect and recognize numbers in 2 different languages. We present the CNN-based 2-D model to detect air-writing in real-time video. The proposed models were implemented on our developed datasets for two languages, Devanagari and English, with 99.55% accuracy. Our contribution is focused on the air-writing interface using a fingertip to write the numbers instead of traditional input devices. In addition, different air-writing gestures are introduced to control the writing activity. Therefore, our proposed model integrates detection, recognition, and control of air-writing activity. Hence, we achieved precise detection and management of palm movements and accurate recognition of air-written numbers. The results are promising and valuable for real-time multilingual numeral string recognition.

Keywords: Air-Writing, Palm Detection, Handwriting Recognition, HCI, CNN

Introduction

Like pen writing, finger-touched writing techniques are widely used for human and computer interaction (Lee and Hollerer, 2007; Zhang *et al*., 2013). However, the emerging methods enhance their capabilities in tracking hand and fingertip movements without using physical devices for writing. In this case, the air-writing technique is introduced to provide a feasible alternative solution instead of conventional text inputs using the keyboard or a mouse. It is the best eye-free execution technique compared to virtual inputs using virtual keyboards or similar devices (Chen *et al*., 2016a). The air-writing systems are encouraged by virtual and augmented reality methods that improve Human-Computer Interaction (HCI) systems. It also has replaced traditional systems in several fields with hand gesture-based interaction. The areas like automotive interfaces (Ohn-Bar and Trivedi, 2014) and human activity recognition (Rohrbach *et al*., 2016) have adopted these techniques with several proposed approaches (Molchanov *et al*., 2015).

However, hand motion gesture recognition is a challenging and inadequate task for air-writing. It aims to replace touch panels to achieve more real human and computer interfaces. In recent studies, the HCI approaches in a vision-based system are introduced with different methods, including mid-air finger-writing recognition. Several studies have been conducted in detecting hand movements with some limitations. For instance, fingertip detection using various sensors are previously introduced to track fingertip movement, constant noticing, and identification of air-writing (Amma *et al*., 2012). Similarly, a red marker indicating fingertip was also successfully used in the hand movement recognition structure to identify letters and numbers with various arithmetic operators (Misra *et al*., 2028). In recent research, depth sensors such as spatiotemporal Hough Forest (Chang *et al*., 2016) and leap motion controller (Chen *et al*., 2016b) are used for air-writing recognition without markers or gloves.

On the other hand, air-writing without sensors can be done by writing letters or numbers in free space by hand or finger movement, specifically in front of a webcam without any external devices or sensors. Only some studies have been conducted on air-writing multilingual numeral recognition using only a webcam or without handheld devices. For instance, the study (Rahman *et al*., 2021) presented a sliding window-based method that isolates a small segment of the spatiotemporal input from the air-writing activity for noise removal and digit segmentation. While processing temporal data, Recurrent Neural Networks prove a sizable level of accuracy. English numbers were studied using MNIST and pen digit datasets as well as the experiments' special air-written English numeral dataset (ISI-Air Dataset). The system generated 98.75 and 85.27% accuracy readings for single-digit and multiple-digit English digits, respectively (Rahman *et al*., 2021). The authors Mukherjee *et al*. (2019) introduced a new algorithm for writing hand pose detection to initial air-writing using the faster Region-based CNN (R-CNN) framework for accurate hand detection and hand segmentation. The number of elevated fingers was also counted based on the geometrical characteristics of the hand. Additionally, they propose a dependable fingertip tracking and recognition method based on a recently developed signature function called distance-weighted curvature entropy. They used the suggested air-writing technique and achieved 73.1% accuracy. Using the 0-9 digits and tiny English alphabets from the EMNIST dataset, air-writing character recognition accuracy was 96.11% (Mukherjee *et al*., 2019).

However, earlier studies are inadequate to work on real-world scenarios and not popular due to expensive approaches, techniques, and restricted use. The studies on air-writing multilingual numeral recognition without handheld devices used RNN and CNN methods with adequate performance. Additionally, the previous practices limit the low performance of motion sensing and hand movement tracking with a lack of detection. Natural handwriting patterns are not followed in earlier studies. Changing the user's writing behavior makes it challenging to recognize accurate input. Therefore, the present study proposes a unique approach that uses a CNN method to detect fingertips and identify air-writing in the video frame. Real-world air-writing can be recognized with laptop video cameras or webcams, making it easily accessible to all users. Since the face and hand skin tones are identical, detecting the hand in the video frame is challenging. Thus, we proposed palm detection to reduce the complexity of the system.

The primary objectives are: (1) To develop own air-writing dataset for the Devanagari and English languages, (2) To implement the palm detection model for detecting hand landmarks using Mediapipe, (3) To detect hand pose, fingertip, and hand gestures, (4) Digit recognition using CNN-based models, (5) To evaluate the accuracy obtained via CNN model and compare with standard literature.

Literature Review

The recognition of numerals, specifically for Devanagari and English scripts, has diverse applications in human-computer interaction, augmented reality, and wearable technology. This literature review summarizes the modern techniques, datasets, challenges, and future scope in air-writing numeral recognition for Devanagari and English.

There are broadly four techniques and algorithms: Sensor technologies: Various sensors, including accelerometers, gyroscopes, and depth cameras, have been used to capture air-writing gestures. Inertial Measurement Units (IMUs): IMUs, which combine accelerometers and gyroscopes, are popular due to their compact size and ability to capture motion in 3D space (Luo *et al*., 2023). Depth cameras: Devices like the Microsoft Kinect or Leap Motion Controller capture the depth and position of the hand, providing detailed gesture data (Mohammadi and Maleki, 2019). Preprocessing methods: Preprocessing is crucial to clean and normalize the raw data from sensors. Noise filtering: Techniques like Kalman filtering are applied to smooth the motion data (Kumar and Bhatia, 2014). Segmentation: Identifying the start and end points of a gesture to segment meaningful data (Younas *et al*., 2023). Feature extraction: Effective features must be extracted from the raw data to feed into recognition algorithms. Time-domain features: These include the velocity and acceleration of the hand movements (Gan *et al*., 2020). Frequency-Domain features: Fourier transform can be applied to extract frequency components of the gesture data (Arya *et al*., 2015). Spatial features: Trajectory and curvature of the hand movements (Pardeshi *et al*., 2014).

Recognition algorithms: Several machine and deep learning algorithms were employed for air-writing recognition. Hidden Markov Models (HMMs): HMMs have been used to model the sequential nature of gesture data (Ghods and Sohrabi, 2016). Dynamic Time Warping (DTW): DTW aligns time-series data to recognize patterns despite variations in speed (Luo *et al*., 2023). Neural Networks: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown high accuracy in recognizing complex gestures (Alam *et al*., 2019; Rahman *et al*., 2020). Hybrid models: Combining HMMs with neural networks to leverage the strengths of both approaches (Trivedi *et al*., 2018).

Devanagari numeral recognition challenges: Devanagari script has complex strokes and intricate patterns, making numeral recognition more challenging (Gupta and Bag, 2021). Datasets: Publicly available datasets for Devanagari numerals are limited, necessitating the creation of custom datasets (Amma *et al*., 2012). Studies: Research has shown promising results using CNNs for recognizing Devanagari numerals. Preprocessing techniques like stroke extraction and normalization are crucial (Kumar and Bhatia, 2014). English numeral recognition challenges: Variability in handwriting styles and speeds can affect recognition accuracy (Hamad and Kaya, 2016). Datasets: Several datasets, like the MNIST dataset, are widely used for benchmarking numeral recognition systems (Agrawal *et al*., 2021). Studies: CNNs, RNNs, and hybrid models have been extensively used for English numeral recognition, achieving high accuracy. Data augmentation techniques help in handling variations in handwriting (Rahmanian and Shayegan, 2021).

Comparative studies: Comparative studies between Devanagari and English numeral recognition highlight the importance of tailored preprocessing and feature extraction techniques for each script (Qu *et al*., 2018; Patil *et al*., 2023; Patil, 2014; Patil and Mali, 2015-2016; Mali and Patil, 2015). Transfer learning has been explored to leverage models trained on one script to improve performance on another Roy *et al*. (2018).

Challenges and future directions: Real-time recognition: Achieving real-time performance with high accuracy remains a significant challenge (Rahman *et al*., 2020). User variability: Handling the variability in gestures across different users requires robust algorithms (Rahman *et al*., 2020). Dataset diversity: Creating large and diverse datasets for both Devanagari and English numerals is crucial for training robust models (Huang *et al*., 2015; Jabde *et al*., 2024c). Integration with AR/VR: Exploring the integration of air-writing recognition with augmented and virtual reality applications (Lee and Hollerer, 2007).

Materials

Software Tools

Feature extraction and preprocessing were implemented using Python libraries such as Open CV or scikit-learn.

Hardware

Experiments were conducted on a server with NVIDIA GPUs, Intel Core i7 processor.

Methods

The Developed Dataset

In the present study, we have created our dataset (Table 1) based on air-writing videos of 2 scripts, Devanagari and English script using fingertip trajectories. The dataset samples in Fig. (1) show Devanagari 0 and 5 and English 4 and 7.

The air-writing detection system has two stages: Palm detection at the primary stage and writing recognition in the second step (Fig. 2). In the first stage, Palm detection includes fingertip detection and recognizing palm gestures for a control writing activity. Our system can recognize 0-9 digits of 2 selected languages, such as Devanagari and English Script/language. An index finger is used for writing the numbers and then the number is written in any of the two defined languages. In the second stage, a registered number is recognized with each digit and its language. The system can realize the combination of language inputs like English and Devnagari, Gurumukhi and English or like this.

Table 1: The developed dataset with their details

Dataset	Multilingual numeral dataset
Samples	Total 20,000 samples (1000 Samples per digit per language)
Languages	Devanagari, English,
Digits	0-9
Size	1KB for each image

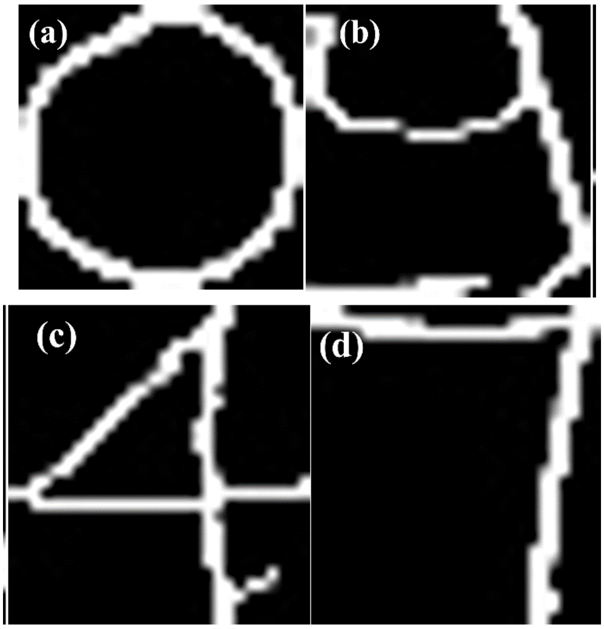

Fig. 1: Air-writing dataset samples in different languages (a) Devnagari 0; (b) Devnagari 5; (c) English 4; (d) English 7

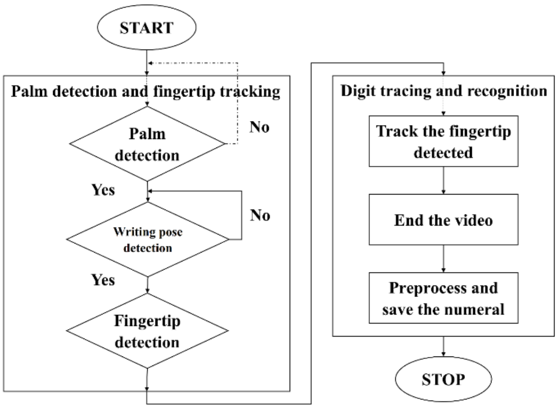
START
Palm detection and fingertip tracking
Palm detection
No
Yes
Writing pose detection
No
Yes
Fingertip detection
Digit tracing and recognition
Track the fingertip detected
End the video
Preprocess and save the numeral
STOP

Fig. 2: The proposed block diagram of air-writing detection and recognition model

Palm Detection Model

Palm detection is a more straightforward framework than a hand detector due to the identical skin tone of the face and hands in video frames (Mittal *et al.*, 2011). Therefore, first, we train a palm detector by estimating bounding boxes of palms and fists using the Mediapipe library (Lugaresi *et al.*, 2019). It needs to detect writing hand gestures and a fingertip with real-time tracking. Then, the air-writing digit is recognized with character trajectory. CNN plays an essential role in these complex and challenging tasks. It can detect palms with the Single Shot Detector (SSD) technique and performs well in more than one palm seen due to smaller palm size. However, we restrict the detection to one palm to reduce confusion in spotting palm gestures. Another palm can be detected only when the first palm disappears in the video frame. The plan of the projected methodology is depicted in Fig. (3).

Hand Landmark Model

The palm detection task is succeeded by the hand landmark detection task (Lugaresi *et al.*, 2019). The accurate localization of hand landmarks is shown in Fig. (4). A model illustrates 21 coordinates, including four landmarks on each finger. However, the model continuously detects all landmarks in real-time video frames by the regression method. A robust model can even catch a partially visible hand in the frame. It represents a hand with 21 landmarks and a hand flag indicating the existence of a hand in the video frame.

We benefited from two-dimensional coordinates to discover real-world samples in the dataset and track the landmark technique (Simon *et al.*, 2017). It comprises relative depth concerning the wrist point. Each landmark represents the detected hand with L_x, L_y, and L_z coordinates. Width and height indicated by L_x and L_y are normalized to [0.0, 1.0], whereas depth z is considered from the wrist as an origin. The value of L_z indicates the closeness of the landmark to the wrist. Though the number of hands has been detected in a single frame, we consider one hand to enhance the accuracy in air-writing detection.

Furthermore, the tracking is enhanced by evaluating the threshold value while handedness is predicted with binary classification to indicate the left or right hand. Real-time GPU and CPU interference are supported on mobile and desktop devices.

CNN-Based Digit Recognition

A 2-D CNN model is designed to detect and recognize air-writing (Fig. 5) with real-time video input. The model has been trained using an efficient dataset of many samples. Our dataset has 20000 various examples. The model splits the dataset into 80% for the training set and 20% for the testing set of datasets (Table 4). Each dataset contains a sample of each digit from each language. The large variety of pieces in the dataset enhances the model's performance.

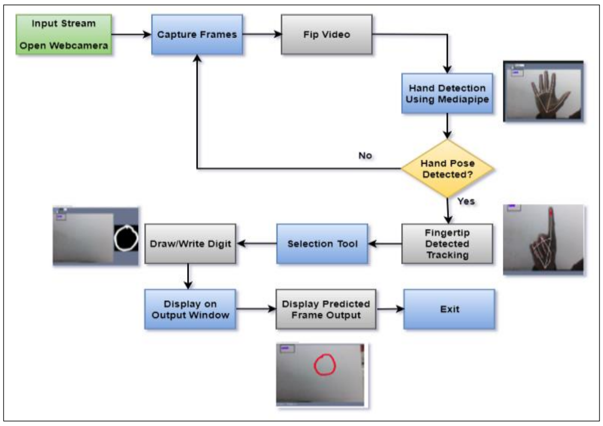

Fig. 3: The workflow of the air-writing recognition system

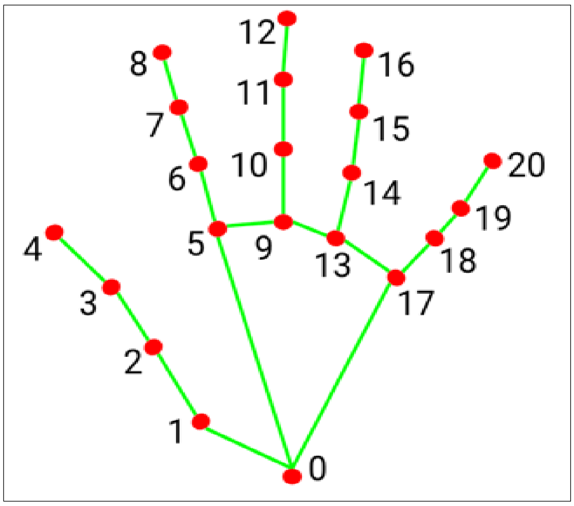

Fig. 4: Hand tracking with hand landmarks

Table 4: Samples in each type of dataset

Dataset	Samples
Train	16000
Test	4000

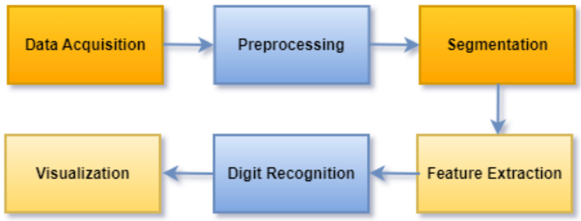

Fig. 5: CNN-based digit recognition steps

CNN model uses kernel size (*x*, *y*) to illustrate digits in two dimensions. The training dataset is applied to the model. Our proposed model uses three filters for different data sizes, including 32, 64, and 128 MB. Each convolutional block contains a convolution layer and maximal pooling, including batch normalization and activation functions. A pooling layer between two convolution layers is intended to reduce the spatial dimensionality and the output is a featured map pooled. We apply the max pooling technique to control feature selection and overfitting of the model. The CNN applies batch normalization after convolution to enhance the network's performance (Ioffe and Szegedy, 2015). It also maintains the stability of the model and then we used the rectified linear unit activation function in the hidden layer.

Data Acquisition

Real-time air-writing is started with video captured with a web camera (Fig. 1). A palm in a frame is recognized and a palm detection model using a media pipe is applied to display hand landmarks. A hand pose with only the index finger open allows air-writing. A hand gesture, combining landmarks on the thumb and middle finger, starts air-writing and tracking of a fingertip. It provides writing digits which are saved with another hand gesture. The number is displayed as an image in a separate window. The CNN-based workflow is provided in Fig. (5).

Preprocessing

An air-writing input is saved as an image and then preprocessed to recognize the numbers. Image masking reduces the image size defining the boundaries of a digit in the frame. Firstly, the digits were detected and the number of pixels was added or removed from the image according to the size and shape of the integer. It is essential to remove unnecessary data from the picture (Rahman *et al*., 2021; Patil *et al*., 2023).

Segmentation

Background Removal

Segmentation of the digit includes background removal in the digit image. Background subtraction enhances the performance of digit detection since it separates foreground elements by generating a foreground mask and removes the background (Zivkovic, 2004). In all cases, CNN uses appropriate thresholds to release the image's dark area and morphological operators remove noises. The remaining is the digit in the picture (Mukherjee *et al*., 2019; Jabde *et al*., 2024a).

Digit Segmentation

For digit segmentation, the contour is created, indicating the centroid of the digit. The borderline with the same padding covers the entire image by applying filters and stride. We use three filters (Mukherjee *et al*., 2019) and compress the image to the appropriate size. The CNN model processes the image with activation functions ReLU and SoftMax. At the same time, max pooling covers a maximum region of the feature map to extract the most protruding features from the image (Goodfellow *et al*., 2016). ReLU function *f(x)* is given in Eq. (1):

$$f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \qquad (1)$$

A dense layer in CNN classifies the input according to the output obtained from the convolutional layer. The SoftMax activation function (Eq. 2) maps the real-value information when applied to the dense outer layer. The prediction probability ranges between [0, 1]:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (2)$$

where, σ is the SoftMax function, *z* is the input vector, while e^{zi} and e^{zj} indicate output vectors with *K* number of classes. A CNN is trained with a batch of *n* samples with parameter set ϕ performing an update iteratively to acquire the finest parameter set matching to the minimal loss function (Hsieh *et al*., 2021) by Eq. (3):

$$\varphi_t = \varphi_{t-1} - \rho \, \nabla_{\varphi} \, l(\varphi; x^{(i:x+n)}; \, y^{(i:y+n)}) \qquad (3)$$

where, target and predicted outputs *x* and *y* of the network are included with gradient operator $\nabla\varphi$ and learning rate ρ. The loss function is also computed using Eq. (4) Bachman (2007):

$$l(\varphi, x, y) = -\sum_{i=1}^{N} x_i \log y_i \qquad (4)$$

In model training, a slower learning rate can cause slow convergence, while too large a learning rate causes variation of the loss function value. Thus, we used the RMS prop optimization algorithm to select a suitable learning rate. Therefore, the model is trained with an appropriate learning rate and minimal loss function to attain model fit. Our model is tested with a test dataset and evaluated further for the performance of the model. It includes loss function, accuracy, and confusion matrix.

Digit Recognition

A processed image then displays a picture of a digit in a paint sheet and stores the location data. A live tracker gives a preview of the digit on the screen. A number is then compared with the multilingual numeral dataset and a pre-trained CNN classifier recognizes the accurate digit and language. The recognized digit is displayed with the name of the language. The output contains digit and language names in English script.

Each number is recognized separately in case of more than one digit as input. Our model can efficiently predict multilingual information in the same frame. If air-writing input contains two numbers in two different languages, they are individually expected with their input language. Then the output displays each digit with its language name separately.

Results and Discussion

Our Dataset is developed with several samples for each digit in different languages. Figure (6) shows a sample graph of the Dataset considering English and Devanagari scripts/languages. In this graph, the number of samples is indicated by several elements. The alphabet defines a digit from 0-9. However, our Dataset includes 1000 elements for each digit.

Table (5) demonstrates the value of the loss function and accuracy related to the validation set. As the value of the validation loss function decreases, accuracy increases and attains the highest accuracy of 0.9955. Also, the time required per step is considerably reduced. The loss and accuracy of the training and testing model are also depicted in Fig. (7).

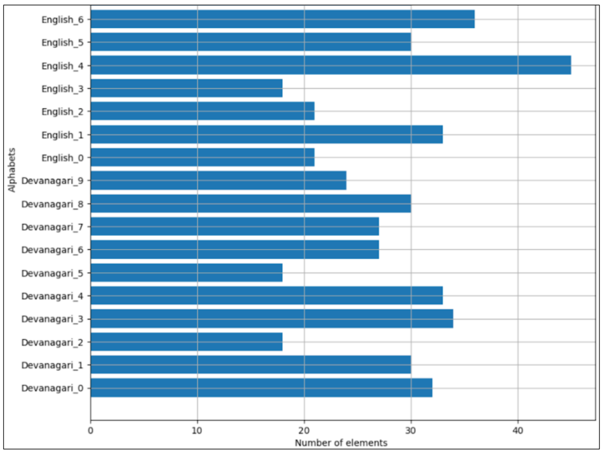

Fig. 6: Graph of training dataset for numerals in English and Devanagari scripts

Table 5: Epoch with loss function and accuracy for the validation set

Sr. No.	Time (s)	Time per step (ms/step)	Loss	Accuracy	Val-loss	Val-accuracy
1	3	92	8.0165	0.1771	2.1286	0.4865
2	1	68	1.8641	0.4888	1.2412	0.6996
3	1	62	0.8162	0.7892	0.3703	0.9260
4	1	59	0.2097	0.9574	0.0907	0.9955

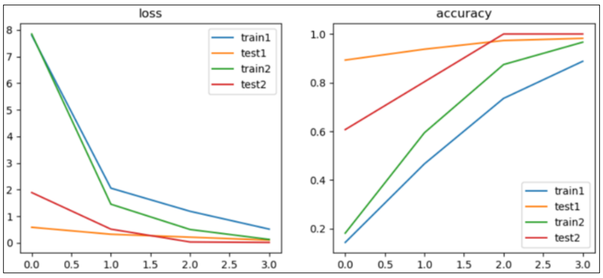

Fig. 7: Loss and accuracy of training and testing of CNN model

One-Digit Prediction

Airwriting with one digit is displayed on a separate paint sheet and the result shows “Language and Digit” in English. Figure (8) shows 3, 9, and 4 in the Devnagari language. The output indicates that the prediction is “Devnagari_3”, “Devnagari_9” and “Devnagari_4”.

A prediction of each digit in any language is indicated in Fig. (9) shows examples of English and Devnagari digits (e.g., 9, 6, 0 in Devnagari and 8, 8, 4, 6, 9, 8 in English). The language is predicted with a digit.

Two-Digit Prediction

Airwriting with two-digit is displayed on a separate paint sheet and the result shows “Language and Digit” for each digit in English. Figure (10) shows 67 and 78 in Devnagari and English language. The output indicates that the prediction for 67 is “Devnagari_6” and “English_7”, and for 78, it shows “English_8”, English_7”.

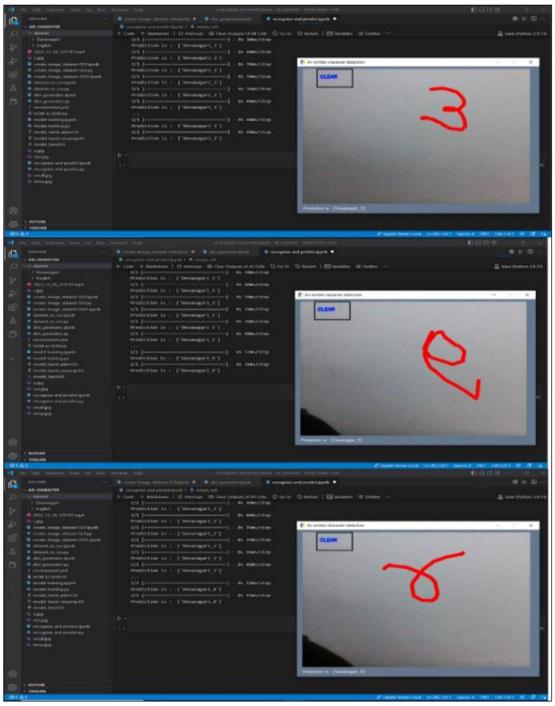

Fig. 8: Prediction of single digits and their languages

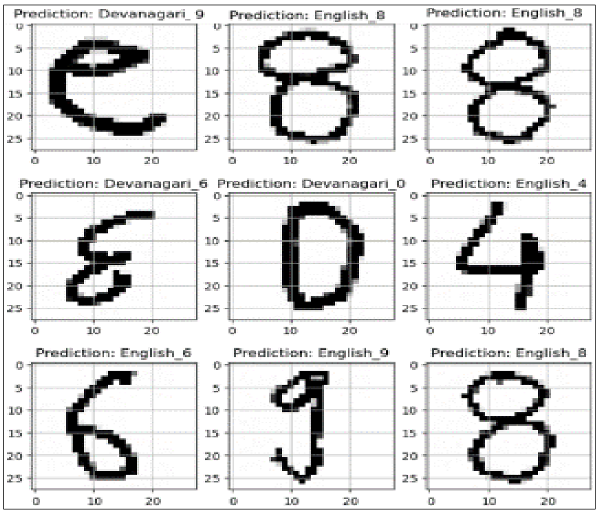

Fig. 9: Sample digit prediction graph

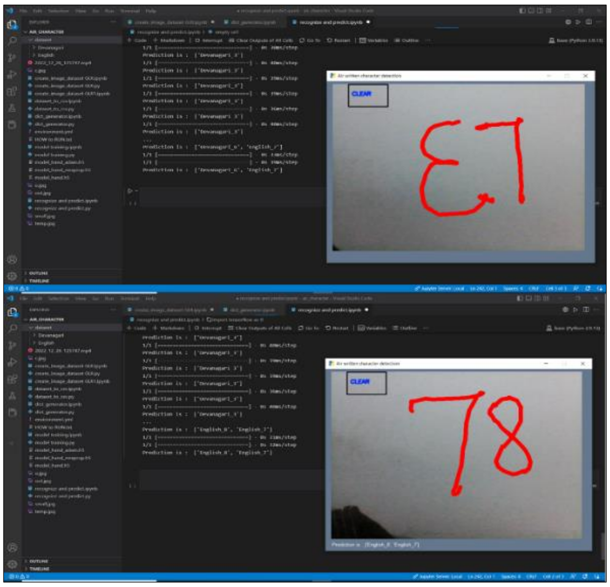

Fig. 10: Prediction of two-digit numbers 67 and 78 in Devnagari and English language

Comparative Analysis of Results

The confusion matrix has been generated to detect the correctly classified digits shown in Fig. (11). As shown in Table (6), the proposed system using palm detection and CNN architecture for air-written multilingual numeral recognition performance is quite promising as the model accuracy achieved and reported is 99.55%. It is observed that some of the air-written character recognition without sensors reported a maximum recognition accuracy is 96%. Also, numeral recognition is addressed and it said the maximum accuracy was 98.45%.

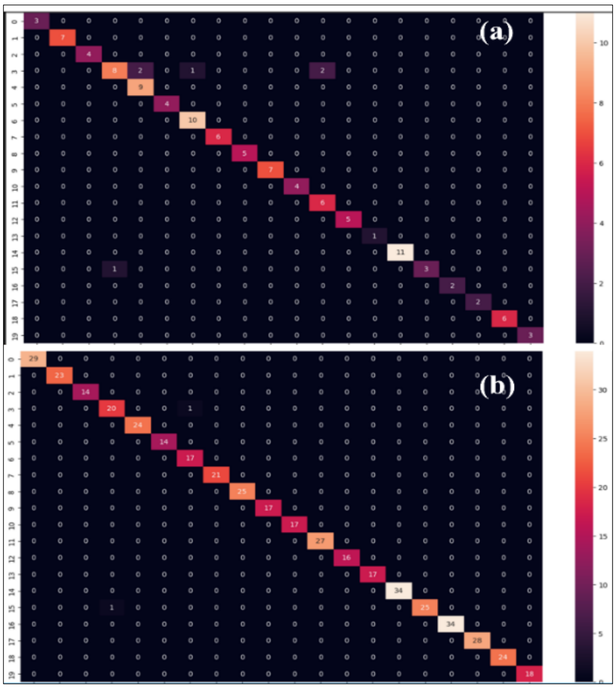

Fig. 11: Confusion matrix generated using CNN model for (a) Devanagari and English one digit and (b) Devanagari and English for two digits

Table 6: Comparative analysis of the proposed system with recent studies for air-writing multilingual numeral recognition without sensors

Datasets	Classification method	Accuracies (%)	Reference
English character-800 video	CNN	96.11	Mukherjee *et al.* (2019)
English, Bengali, Devanagari Numeral-10000 each	CNN	97.7, for numerals	Roy *et al*. (2018)
English numeral dataset-13600 images	CNN	98	Hsieh *et al*. (2021)
English Character Dataset-26 image	HW + EMD	90	Ishida *et al.* (2010)
English Numeral dataset-one digit and multiple digits	CNN	98.45 One English numerals, respectively	Rahman *et al.* (2020)
Developed-Air Dataset in the present study (Devanagari and English 0-9 numeral dataset)	CNN	99.55	Current study

Our CNN-based air-writing recognition model is highly efficient and trained with high accuracy. It can detect digits with a combination of multiple languages. The model has achieved an accuracy of 0.9955 with a minimal loss function value of 0.0907. The training and testing results show that the model is robust and competent to recognize any digit with very little processing time. A fingertip tracking is simplified and used with a novel media pipe framework reducing the need for an expensive and large number of elements. Thus, the webcam can provide precise and significant video input to the system reducing hand detection issues. A CNN is widely accepted as a faster and more accurate prediction with the correct classification (Jabde *et al.*, 2024b). In this case, the forecast of both digit and language is rapidly processed. Hand landmark structure reduced complexity in the detection of the fingertip as well as hand gesture reducing time for input video and eliminating challenges in real-time uncontrolled environments. Thus, our air-writing detection model based on CNN performs better in a real-time environment, while a large multilingual numerals dataset attains a superior collection of samples.

Conclusion

In the present study, a novel dataset of multilingual numeral data is developed with various inputs. A large dataset of 10000 samples in 2 scripts/languages is significant for training the CNN model to achieve the efficient performance of digit recognition. We have added language detection to widen the scope of our model. Another evolution in the proposed system is a real-time fingertip tracker using the Mediapipe library to overcome hand detection issues due to skin color. It reduces the model's size and enhances the speed of palm detection. We used a palm detection technique instead of hand detection, making fingertip detection easier than the technique used for hand detection. Our proposed system has a broad scope with different hand gestures to control air-writing activity. With this implementation, fingertip tracking is made easy, improving the system's efficiency without using a sensor or marker. Digit recognition is a more straightforward process using CNN segmentation and classification techniques. It can easily recognize the digit using novel multilingual datasets created in different languages and predict languages of real-time air-written single or multiple multilingual numbers with higher accuracy. Our model has attained 0.9955 of maximum model accuracy.

Limitations

Real-time digit recognition may affect performance due to changes in the individual writing style and the similarity of the appearance of digits in different languages. However, an adequate number of samples by several individuals are provided in the dataset to reduce writing style impact. A fingertip velocity can affect the tracking performance, thereby affecting a correct prediction. Despite the proposed framework to recognize digits using tools like media pipe for palm detection, it is challenging to control air-writing activity with little hand gestures. A real-time video can face many issues in capturing the right-hand motion to control the start and stop point, saving input in air-writing. Thus, it needs more research on quality output is required.

Acknowledgment

The authors would like to thank anonymous reviewers and editors for reviewing the manuscript and for their scientific and valuable comments and suggestions that have improved the quality of this manuscript.

Funding Information

The authors have not received any financial support or funding to report.

Author’s Contributions

Meenal Jabde: Conceptualization, data curation, drafting the manuscript.

Chandrashekhar Himmatrao Patil: Formal analysis, manuscript editing, corrections, evaluation.

Amol D. Vibhute: Scientific analysis, investigation, validation, technical assistance, manuscript review, and final editing.

Shankar Mali: Formal analysis.

All authors read and approved the final manuscript.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

References

Agrawal, A. K., Shrivas, A. K., & Awasthi, V. Kumar. (2021). A Robust Model for Handwritten Digit Recognition using Machine and Deep Learning Technique. *2021 2nd International Conference for Emerging Technology (INCET)*, 1–4. https://doi.org/10.1109/incet51464.2021.9456118

Alam, Md. S., Kwon, K.-C., & Kim, N. (2019). Trajectory-Based Air-Writing Character Recognition Using Convolutional Neural Network. *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*, 86–90. https://doi.org/10.1109/crc.2019.00026

Amma, C., Georgi, M., & Schultz, T. (2012). Airwriting: Hands-Free Mobile Text Input by Spotting and Continuous Recognition of 3d-Space Handwriting with Inertial Sensors. *2012 16th International Symposium on Wearable Computers*, 52–59. https://doi.org/10.1109/iswc.2012.21

Arya, S., Chhabra, I., & Lehal, G. S. (2015). Recognition of Devnagari Numerals using Gabor Filter. *Indian Journal of Science and Technology*, *8*(27), 1–6. https://doi.org/10.17485/ijst/2015/v8i27/81856

Bachman, D. (2007). *A dvanced Calculus Demystified*. McGraw-Hill.

Chang, H. J., Garcia-Hernando, G., Tang, D., & Kim, T.-K. (2016). Spatio-Temporal Hough Forest for efficient detection–localisation–recognition of fingerwriting in egocentric camera. *Computer Vision and Image Understanding*, *148*, 87–96. https://doi.org/10.1016/j.cviu.2016.01.010

Chen, M., AlRegib, G., & Juang, B.-H. (2016a). Air-Writing Recognition Part I: Modeling and Recognition of Characters, Words and Connecting Motions. *IEEE Transactions on Human-Machine Systems*, *46*(3), 403–413. https://doi.org/10.1109/thms.2015.2492598

Chen, M., AlRegib, G., & Juang, B.-H. (2016b). Air-Writing Recognition Part II: Detection and Recognition of Writing Activity in Continuous Stream of Motion Data. *IEEE Transactions on Human-Machine Systems*, *46*(3), 436–444. https://doi.org/10.1109/thms.2015.2492599

Gan, J., Wang, W., & Lu, K. (2020). In-air handwritten Chinese text recognition with temporal convolutional recurrent network. *Pattern Recognition*, *97*, 107025. https://doi.org/10.1016/j.patcog.2019.107025

Ghods, V., & Sohrabi, M. K. (2016). Online Farsi Handwritten Character Recognition Using Hidden Markov Model. *Journal of Computers*, *11*(2), 169–175. https://doi.org/10.17706/jcp.11.2.169-175

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (illustrated Ed.). MIT press.

Gupta, D., & Bag, S. (2021). CNN-based multilingual handwritten numeral recognition: A fusion-free approach. *Expert Systems with Applications*, *165*, 113784. https://doi.org/10.1016/j.eswa.2020.113784

Hamad, K., & Kaya, M. (2016). A Detailed Analysis of Optical Character Recognition Technology. *International Journal of Applied Mathematics, Electronics and Computers*, *4*(Special Issue-1), 244–244. https://doi.org/10.18100/ijamec.270374

Hsieh, C.-H., Lo, Y.-S., Chen, J.-Y., & Tang, S.-K. (2021). Air-Writing Recognition Based on Deep Convolutional Neural Networks. *IEEE Access*, *9*, 142827–142836. https://doi.org/10.1109/access.2021.3121093

Huang, Y., Liu, X., Jin, L., & Zhang, X. (2015). DeepFinger: A Cascade Convolutional Neuron Network Approach to Finger Key Point Detection in Egocentric Vision with Mobile Camera. *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2944–2949. https://doi.org/10.1109/smc.2015.512

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, 448–456.

Ishida, H., Takahashi, T., Ide, I., & Murase, H. (2010). A Hilbert warping method for handwriting gesture recognition. *Pattern Recognition*, *43*(8), 2799–2806. https://doi.org/10.1016/j.patcog.2010.02.021

Jabde, M. K., Patil, C. H., Vibhute, A. D., & Mali, S. (2024a). A Comprehensive Literature Review on Air-written Online Handwritten Recognition. *International Journal of Computing and Digital Systems*, *15*(1), 307–322. https://doi.org/10.12785/ijcds/150124

Jabde, M., Patil, C., Vibhute, A. D., & Mali, S. (2024b). Offline Handwritten Multilingual Numeral Recognition Using CNN. *Intelligent Systems for Smart Cities*, 385–400. https://doi.org/10.1007/978-981-99-6984-5_25

Jabde, M., Patil, C., Vibhute, A. D., & Saini, J. R. (2024c). A Systematic Review of Multilingual Numeral Recognition Using Machine and Deep Learning Methodology. *International Journal of Computing and Digital Systems*, *17*(1), 1–28.

Kumar, G., & Bhatia, P. K. (2014). Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition. *International Journal of Advances in Engineering Sciences*, *3*(3), 14–22. https://doi.org/10.13140/RG.2.1.3896.7842

Lee, T., & Hollerer, T. (2007). Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. *2007 11th IEEE International Symposium on Wearable Computers*, 83–90. https://doi.org/10.1109/iswc.2007.4373785

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, Ming Guang, Lee, J., Chang, Wan-Teh, Hua, Wei, Georg, M., & Grundmann, M. (2019). MediaPipe: A Framework for Building Perception Pipelines. In *arXiv:1906.08172*. https://doi.org/10.48550/arXiv.1906.08172

Luo, Y., Ke, W., & Lam, C.-T. (2023). Wearable Real-time Air-writing System Employing KNN and Constrained Dynamic Time Warping. *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6. https://doi.org/10.1109/wcnc55385.2023.10118944

Mali, S. M., & Patil, C. H. (2015). Marathi Handwritten Numeral Recognition using Zernike Moments and Fourier Descriptors. *International Journal of Computer Applications*, 32–34.

Misra, S., Singha, J., & Laskar, R. H. (2018). Vision-based hand gesture recognition of alphabets, numbers, arithmetic operators and ASCII characters in order to develop a virtual text-entry interface system. *Neural Computing and Applications*, *29*(8), 117–135. https://doi.org/10.1007/s00521-017-2838-6

Mittal, A., Zisserman, A., & Torr, P. H. S. (2011). Hand detection using multiple proposals. *Procedings of the British Machine Vision Conference 2011*, 5.

Mohammadi, S., & Maleki, R. (2019). Real-time Kinect-based air-writing system with a novel analytical classifier. *International Journal on Document Analysis and Recognition (IJDAR)*, *22*(2), 113–125. https://doi.org/10.1007/s10032-019-00321-4

Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1–7. https://doi.org/10.1109/cvprw.2015.7301342

Mukherjee, S., Ahmed, Sk. A., Dogra, D. P., Kar, S., & Roy, P. P. (2019). Fingertip detection and tracking for recognition of air-writing in videos. *Expert Systems with Applications*, *136*, 217–229. https://doi.org/10.1016/j.eswa.2019.06.034

Ohn-Bar, E., & Trivedi, M. M. (2014). Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations. *IEEE Transactions on Intelligent Transportation Systems*, *15*(6), 2368–2377. https://doi.org/10.1109/tits.2014.2337331

Pardeshi, R., Chaudhuri, B. B., Hangarge, M., & Santosh, K. C. (2014). Automatic Handwritten Indian Scripts Identification. *2014 14th International Conference on Frontiers in Handwriting Recognition*, 375–380. https://doi.org/10.1109/icfhr.2014.69

Patil, C. H. (2014). Recognition of Handwritten Marathi Vowels using Zone based Symmetric Density Features. *International Journal of Computer Applications*, *108*(4), 1–6. https://doi.org/10.5120/18896-0187

Patil, C. H., & Mali, S. M. (2015). Segmentation of Isolated Handwritten Marathi Words. *International Journal of Computer Applications*, 21–26.

Patil, C. H., & Mali, S. M. (2016). Handwritten Marathi Consonants Recognition using Multilevel Classification. *International Journal of Computer Applications*, 21–30.

Patil, C. H., Zope, R., & Jabde, M. (2023). Comparative Study of Multilingual Text Detection and Verification from Complex Scene. *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 903–910. https://doi.org/10.1109/icaaic56838.2023.10141373

Qu, X., Wang, W., Lu, K., & Zhou, J. (2018). Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network. *Pattern Recognition Letters*, *111*, 9–15. https://doi.org/10.1016/j.patrec.2018.04.001

Rahman, A., Roy, P., & Pal, U. (2021). Air Writing: Recognizing Multi-Digit Numeral String Traced in Air Using RNN-LSTM Architecture. *SN Computer Science*, *2*(1). https://doi.org/10.1007/s42979-020-00384-9

Rahman, A., Roy, P., & Umapada, P. (2020). Continuous Motion Numeral Recognition Using RNN Architecture in Air-Writing Environment. *Pattern Recognition*, 76–90. https://doi.org/10.1007/978-3-030-41404-7_6

Rahmanian, M., & Shayegan, M. A. (2021). Handwriting-based gender and handedness classification using convolutional neural networks. *Multimedia Tools and Applications*, *80*(28–29), 35341–35364. https://doi.org/10.1007/s11042-020-10170-7

Rohrbach, M., Rohrbach, A., Regneri, M., Amin, S., Andriluka, M., Pinkal, M., & Schiele, B. (2016). Recognizing Fine-Grained and Composite Activities Using Hand-Centric Features and Script Data. *International Journal of Computer Vision*, *119*(3), 346–373. https://doi.org/10.1007/s11263-015-0851-8

Roy, P., Ghosh, S., & Pal, U. (2018). A CNN Based Framework for Unistroke Numeral Recognition in Air-Writing. *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 404–409. https://doi.org/10.1109/icfhr-2018.2018.00077

Simon, T., Joo, H., Matthews, I., & Sheikh, Y. (2017). Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4645–4653. https://doi.org/10.1109/cvpr.2017.494

Trivedi, A., Srivastava, S., Mishra, A., Shukla, A., & Tiwari, R. (2018). Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network. *Procedia Computer Science*, *125*, 525–532. https://doi.org/10.1016/j.procs.2017.12.068

Younas, J., Narayan, S., & Lukowicz, P. (2023). Air-Writing Segmentation using a single IMU-based system. *2023 19th International Conference on Intelligent Environments (IE)*, 1–6. https://doi.org/10.1109/ie57519.2023.10179093

Zivkovic, Z. (2004). Improved Adaptive Gaussian Mixture Model for Background Subtraction. *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*, 28–31. https://doi.org/10.1109/ICPR.2004.1333992

Zhang, X., Ye, Z., Jin, L., Feng, Z., & Xu, S. (2013). A New Writing Experience: Finger Writing in the Air Using a Kinect Sensor. *IEEE MultiMedia*, *20*(4), 85–93. https://doi.org/10.1109/mmul.2013.50