IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 5, Issue 3, January 2025

Air-Writing Recognition using Deep learning and Artificial Intelligence

Mohsin Arab, Aman Sande, Sushil Deelip Sonawane, Vikas Mahadev Morabagi
Department of Computer Engineering
ISBM College of Engineering Nande, Pune, India
mohsinarab063@gmail.com, aamansande6@gmail.com,
sushilsonawane2002@gmail.com, vikasmorabagi24680@gmail.com

Abstract: *This paper investigates the use of deep learning architectures for recognizing 3D gestures, particularly focusing on air-writing, performed using hand or finger movements. Unlike traditional handwriting, which involves distinct pen-up and pen-down actions, air-writing lacks a defined sequence of such events. Additionally, the absence of visual or tactile feedback and the high dependency among target gestures pose significant challenges for accurate character recognition.*

From the user's perspective, air-writing can be executed in three distinct styles: connected, isolated, and overlapped. In the connected style, users write sequences of characters in free space, similar to writing on paper from left to right. The isolated approach confines each character's gesture within an imaginary 3D box, making segmentation easier but less natural. The overlapped method is the most complex, as it involves stacking adjacent characters within the same imaginary space. Variations in character size and shape during connected writing can impact recognition accuracy, even for the same user, while the constrained isolated style offers more consistent segmentation.

Advancements in Motion Capture (MoCap) technologies and increased computational capabilities have encouraged the exploration of air-writing in human-computer interaction (HCI) systems. Some systems utilize handheld tools or custom gloves for tracking motion trajectories, while low-cost depth sensors, such as Microsoft Kinect, Intel RealSense, and Leap Motion Controller (LMC), offer non-intrusive, real-time tracking solutions. Kinect focuses on long-range 3D posture detection, whereas LMC and RealSense provide millimeter-level precision for tracking hand and finger movements. However, distinguishing actual writing gestures from arbitrary hand movements in continuous motion streams remains a significant challenge, independent of character recognition.

Existing air-writing systems primarily target letter and digit recognition due to the limited availability of word-level training datasets. To address this gap, we propose a solution capable of both detecting and recognizing air-written characters within continuous motion data. Our system, based on the LMC sensor, includes a web interface for capturing 1,200 air-written digits (0–9) while providing real-time visual feedback. An efficient fingertip tracking mechanism simulates pen-up and pen-down states. The air-writing recognition process is framed as a classification task, utilizing both dynamic 3D trajectories (time-series data) and static 2D stroke projections (images) to train deep learning models with convolutional and recurrent architectures. Experimental results demonstrate near-perfect recognition rates achieved within milliseconds, making the system suitable for real-time deployment.

The paper is organized as follows: Section II reviews related works in air-writing. Section III outlines the methodology, including the LMC interface, dataset collection, and employed deep learning models. Section IV describes the experimental setup and evaluation results. Finally, Section V concludes with key findings and future research directions..

Keywords: deep learning architectures

I. INTRODUCTION

Over the past decade, numerous interfaces have been developed to facilitate air-based writing. These systems primarily focus on capturing the spatiotemporal trajectory of hand movements during air-writing. The trajectory of an air-written character can be represented through two core dimensional approaches. Specifically, air-writing recognition can either be analyzed by exploring the temporal dependencies within the trajectory's sequential steps or by examining spatial projections of the motion on vertical and horizontal imaginary 3D planes. These two methods are referred to as the **dynamic** and **static** approaches, respectively.

In the static approach, the air-writing trajectory is treated as a 2D image, resembling conventional handwriting. This enables the application of established techniques from Optical Character Recognition (OCR). Conversely, the dynamic approach represents the motion trajectory's temporal characteristics as a sequence of signal measurements over time. Each frame in the sequence typically contains features such as hand velocity, coordinates relative to the sensor's reference point, and additional calculated metrics like angles between consecutive trajectory points and angular velocities. These are often referred to as **hand-crafted features**. Dynamic gesture recognition methods can thus be adapted for air-writing recognition.

For analyzing static trajectory images, **Convolutional Neural Networks (CNNs)** are widely used. For example, one study introduced a marker-based air-writing framework employing a standard video camera with color-based segmentation to track trajectories. The system then classified static trajectory images using a pre-trained CNN. However, this method was prone to performance degradation under varying lighting conditions due to the reliance on color-based segmentation.

Alternatively, a marker-less approach utilized 3D fingertip coordinates captured by an LMC sensor, which were converted frame-by-frame into consecutive trajectory images. After preprocessing and normalization, these images were used to train a CNN model, achieving an average recognition rate of 98.8%. Despite this high accuracy, deep CNN architectures often include many redundant filters, resulting in significant computational overhead. To address this, a study proposed an efficient pruning algorithm for deep CNN models in air-written Chinese character recognition. This method reduced computational costs with only a 0.17% accuracy drop, compared to a baseline VGG-like CNN achieving 95.33% accuracy.

Dynamic air-writing gesture recognition traditionally employed methods such as **Hidden Markov Models (HMMs)**, **Dynamic Time Warping (DTW)**, and **Support Vector Machines (SVMs)**. However, these techniques are more effective for simple gestures with low inter-class similarities and tend to deliver lower accuracy for complex air-writing tasks. To overcome these challenges, **Recurrent Neural Networks (RNNs)**, particularly those using bidirectional Long Short-Term Memory (BLSTM) units have demonstrated significant performance improvements.

More recently, fusion-based techniques combining CNNs with LSTM or BLSTM classifiers have been introduced. These hybrid networks effectively integrate spatial and temporal feature learning, outperforming earlier methods in air-writing recognition tasks.

II. METHODOLOGY

A. Data Collection

To facilitate the collection of air-written digits (0-9), a user-friendly web interface was developed utilizing the LMC JavaScript API (version 2). The interface was designed to require minimal instructions, making it accessible and efficient. As illustrated in Fig. 1, the LMC device was positioned on a tabletop in front of a display screen, which provided visual feedback to users. Through a calibration process, an air-writing spotting algorithm was implemented to detect the start and end points of gestures dynamically. This approach allowed the screen to serve as a virtual boundary for writing. Additionally, a blue box displayed on the screen specified the active writing area, which could be adjusted by users without affecting the captured frame data.

For the dataset collection, 10 participants (8 males and 2 females) aged between 23 and 50 years were included, with no constraints on stroke patterns or other writing styles. Each participant wrote every digit at least 10 times, resulting in approximately 1,200 samples. Each recording consisted of time-series data representing raw tracking information from the LMC API, sampled at 60Hz. The web interface and the collected dataset are hosted online and are accessible for further research purposes [25].

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 5, Issue 3, January 2025

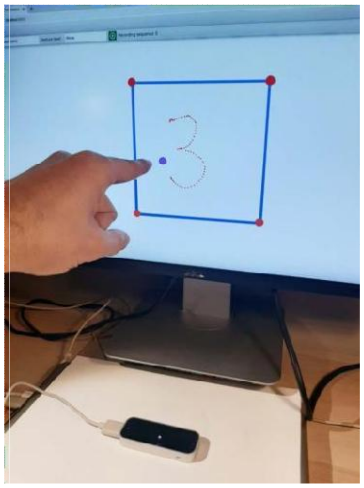

Fig. 1: An example of the LMC orientation and the web interface

B. Architectures

A variety of machine learning methods were explored to address the challenge of digit recognition. These methods were categorized into two main approaches: dynamic and static. Extensive experiments were conducted to compare and, in some cases, combine these approaches for optimal results.

1. Dynamic Approach

In this method, the input was represented as a sequence of 2D points traced by the user's fingertip over time. The goal was to leverage sequential models capable of learning the trajectory and distinguishing between digits effectively. Due to the variability in how digits can be written, robustness was a critical consideration when selecting models. The sequential architectures tested included:

- CNN-LSTM: This architecture added a 1D convolutional layer immediately after the input to extract spatial features. A total of 300 filters with a kernel size of 2 and a stride of 1 were employed. Fig. 3 illustrates this architecture.
- LSTM and BLSTM: These recurrent architectures were employed to capture long-term dependencies in the sequential data. The LSTM used 300 hidden units in each block, while the BLSTM combined outputs from both forward and backward passes. A linear fully connected layer with log-softmax regression classified the outputs into 10 digit classes. The configuration is depicted in Fig. 2.
- TCN-Dynamic: Temporal Convolutional Networks (TCNs) were utilized due to their ability to handle sequential data effectively. A specific configuration with a kernel size of 7, 25 hidden units per layer, and a stack of 6 layers was chosen. The dilation factor doubled with each successive layer, ensuring a receptive field of 64 frames. This configuration was inspired by its success on the Sequential MNIST dataset [27].

In all dynamic models, a "many-to-one" configuration was used, where only the final output was considered for calculating the cross-entropy loss.

2. Static Approach

The static approach involved converting the air-written digits into image representations. Each digit's trajectory was mapped onto a 28x28 binary image by assigning points to corresponding pixels. Min-max normalization was applied to ensure consistency in size and positioning across the dataset. The following architectures were tested:

- CNN: This model comprised two 2D convolutional layers, each followed by batch normalization, a ReLU activation, and max pooling. The first convolutional layer had 16 filters, and the second had 32, with kernel sizes of 5. The encoded representations were passed through a two-layer fully connected network with a softmax activation for classification. Fig. 4 illustrates this architecture.

- TCN-Static: Following the methodology used in the Sequential MNIST dataset [27], the 28x28 binary image was transformed into a 1D sequence by appending rows sequentially. This sequence was used as input for an 8-layer TCN.

3. Combining Static and Dynamic Approaches

To overcome the limitations of individual methods, a hybrid model combining static and dynamic architectures was implemented. This model integrated an LSTM and a CNN as its primary components. The outputs of these models (10-dimensional vectors) were concatenated and passed through a two-layer fully connected network with a softmax activation function for final predictions. This combination aimed to leverage the strengths of both approaches for improved accuracy.

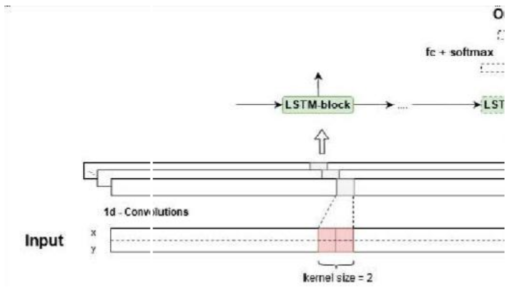

Fig. 3: 1D Convolutional Filtering followed by a LSTM (CNN- LSTM) trained on an oversampled sequence of 2-dimensional digit trail points.

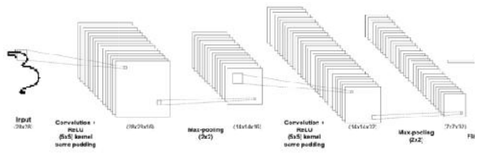

Fig. 4: 2D CNN fed with image representations (28x28) of the drawn digits.

III. EVALUATION AND RESULTS

A. Experimental Setup

During the data collection process, a few erroneous recordings were identified, such as instances where the participant's right hand was misinterpreted as the left hand. A total of 35 such cases were removed from the dataset. After filtering, the dataset was divided into training, validation, and test sets, with 200 samples allocated for testing and 100 for validation. The sequence lengths varied significantly among the recordings. For instance, the digit "1" typically required fewer frames for its recording compared to other digits. To address this, zero-padding was applied to standardize the sequence lengths within each batch, matching the maximum sequence length of that batch. This approach ensured that during training, only the final timestep's output was considered for evaluation.

Each network was developed and trained using PyTorch over 1,000 epochs with a batch size of 64 samples. The training data were shuffled at the start of each epoch, and the validation loss and accuracy were used to determine the optimal model parameters. The experiments were conducted on a system equipped with 16 GB of RAM, an Intel i7-8750H CPU, and a GeForce GTX 1060 GPU with 6 GB of VRAM. The training code is available for access at [25].

B. Results

The best-performing models for each architecture were evaluated on the test set, and their accuracies are summarized in Table I. Both static and dynamic approaches demonstrated impressive performance. Among the architectures, the LSTM achieved the highest accuracy, followed closely by the BLSTM. The CNN ranked third, surpassing the hybrid LSTM-CNN model. The CNN-LSTM and TCN-Dynamic architectures achieved similar results, while the TCN-Static model performed the least effectively. Notably, the TCN models exhibited approximately double the runtime efficiency compared to other architectures, attributed to their smaller number of trainable parameters and the use of dilations.

Analysis of the confusion matrices revealed distinct sources of error for the two top-performing models: the LSTM and the CNN. For the LSTM, only one instance was misclassified, where a "9" was mistaken for a "2." This error can be attributed to the sequential similarity between the trails of these digits. On the other hand, the CNN misclassified three samples. Interestingly, the hybrid LSTM-CNN model resolved the LSTM's single error but retained the CNN's three errors and introduced an additional one.

Increasing the number of convolutional layers in the CNN led to diminished performance, likely due to the simplicity of the 28x28 binary image inputs. Similarly, adding layers or enabling bidirectional configurations for the LSTM did not yield significant improvements and only prolonged training times.

To evaluate the influence of user-specific traits on model performance, a participant cross-validation protocol was employed. In this setup, one participant's data were excluded from the training set and used exclusively for testing. Each fold included 100 samples per participant, 10 from each digit class. Table II highlights a noticeable performance drop across all models in this user-independent setup. Static models, particularly the CNN, outperformed dynamic models, which proved more susceptible to errors. Performance variability across different folds was also more pronounced for dynamic architectures, suggesting the dataset size and diversity were insufficient for building generalized models.

Fine-tuning the models by incorporating samples from the test participant into the training set significantly improved performance. This strategy was applied to the CNN and LSTM models for participants #5 and #8, as shown in Table III. The results consistently demonstrated improved recognition rates with this approach.

IV. CONCLUSIONS AND FUTURE WORK

The strong performance of relatively small models in digit label prediction highlights the effectiveness of the data collection strategy. The use of a vertical plane, such as a computer screen, combined with a distance threshold to indicate breaks in drawing, proved instrumental in achieving these results.

The experiments underscored the strengths and limitations of static and dynamic approaches, as well as the neural architectures employed. While static methods are traditionally favored for handwriting recognition, this study showcased the potential of sequential modeling for the task.

Future work could involve expanding the Leap Motion air-writing dataset to include letters. Additionally, a comparison with other publicly available datasets, such as the 6DMG motion and gesture dataset, would provide further insights. Exploring innovative ways to combine dynamic and static approaches to leverage their respective advantages is another promising direction. Finally, real-time implementation of the recognition system could be pursued, enabling sequential architectures to predict digit labels at shorter time intervals, even before the completion of a drawing.

REFERENCES

[1] K. Kritsis, M. Kaliakatsos-Papakostas, V. Katsouros, and A. Pikrakis, “Deep convolutional and lstm neural network architectures on leap motion hand tracking data sequences,” in 2019 27th European Signal Processing Conference (EUSIPCO), Sep. 2019, pp. 1–5.

[2] J. Pirker, M. Pojer, A. Holzinger, and C. Gutl, “Gesture-based inter-¨ actions in video games with the leap motion controller,” in HumanComputer Interaction. User Interface Design, Development and Multimodality, M. Kurosu, Ed. Springer, 2017, pp. 620–633.

[3] A. Zlatintsi, I. Rodomagoulakis, P. Koutras, A. C. Dometios, V. Pitsikalis, C.S. Tzafestas, and P. Maragos, “Multimodal signal processing and learning aspects of human-robot interaction for an assistive bathing robot,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 2018, pp. 3171–3175.

[4] M. Simos and N. Nikolaidis, “Greek sign language alphabet recognition using the leap motion device,” in Proceedings of the 9th Hellenic Conference on Artificial Intelligence. New York, NY, USA: Association for Computing Machinery, 2016.

[5] H. Wu, Y. Wang, J. Liu, J. Qiu, and X. L. Zhang, “User-defined gesture interaction for in-vehicle information systems,” Multimedia Tools and Applications, vol. 79, no. 1, pp. 263–288, 2020.

[6] M. Chen, G. AlRegib, and B. Juang, "Air-writing recognition—part i: Modeling and recognition of characters, words, and connecting motions," IEEE Transactions on Human-Machine Systems, vol. 46, no. 3, pp. 403– 413, June 2016.

[7] B. Yana and T. Onoye, "Fusion networks for air-writing recognition," in MultiMediaModeling, K. Schoeffmann, T. H. Chalidabhongse, C. W. Ngo, S. Aramvith, N. E. O'Connor, Y.-S. Ho, M. Gabbouj, and A. Elgammal, Eds. Cham: Springer, 2018, pp. 142–152.

[8] P. Roy, S. Ghosh, and U. Pal, "A cnn based framework for unistroke numeral recognition in air-writing," in 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Aug 2018, pp. 404–409.

[9] C. Amma, M. Georgi, and T. Schultz, "Airwriting: a wearable handwriting recognition system," Personal and Ubiquitous Computing, vol. 18, no. 1, pp. 191–203, feb 2013.

[10] Kinect for windows. Accessed April 2020. [Online]. Available: https://developer.microsoft.com/en-us/windows/kinect/

[11] Intel realsense depth and tracking cameras. Accessed April 2020. [Online]. Available: https://www.intelrealsense.com/[12]Ultraleap: Digital worlds that feel human. Accessed April 2020. [Online]. Available: https://www.ultraleap.com/

[13]S. Poularakis and I. Katsavounidis, "Low-complexity hand gesture recognition system for continuous streams of digits and letters," IEEE Transactions on Cybernetics, vol. 46, no. 9, pp. 2094–2108, Sep. 2016. [14]R. Aggarwal, S. Swetha, A. M. Namboodiri, J. Sivaswamy, and C. V. Jawahar, "Online handwriting recognition using depth sensors," in 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Aug 2015, pp. 1061–1065.

[15] M. Chen, G. AlRegib, and B. Juang, "Air-writing recognition—part ii: Detection and recognition of writing activity in continuous stream of motion data," IEEE Transactions on Human-Machine Systems, vol. 46, no. 3, pp. 436–444, June 2016.

[16] M. S. Alam, K.-C. Kwon, M. A. Alam, M. Y. Abbass, S. M. Imtiaz, and N. Kim, "Trajectory-based air-writing recognition using deep neural network and depth sensor," Sensors, vol. 20, no. 2, p. 376, Jan 2020.

[17] J. Gan and W. Wang, "In-air handwritten english word recognition using attention recurrent translator," Neural Computing and Applications, vol. 31, no. 7, pp. 3155–3172, nov 2017.

[18] Y. LeCun, "Neural networks and gradient-based learning in ocr," in Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop, Sep. 1997, pp. 255–255.